Chapter 4

JOINT MULTITARGET PARTICLE FILTERING

C. M. Kreucher

General Dynamics Michigan Research and Development Center

Christopher.Kreucher@gd-ais.com

Mark Morelande

The University of Melbourne

m.morelande@ee.unimelb.edu.au

Keith Kastella

General Dynamics Michigan Research and Development Center

Keith.Kastella@gd-ais.com

A. O. Hero III

The University of Michigan

hero@umich.edu

1. Introduction

In this chapter we review the Joint Multitarget Probability Density (JMPD) for target tracking and sensor management applications and show how it can be evaluated using particle filtering methods. The JMPD is the belief state, i.e. the posterior density, for multiple targets. Specifically, it specifies the probability that: 1) there are T targets present, $T = 1, \ldots, \infty$; and 2) given T, the

targets are in states $\mathbf{x}_1, \ldots \mathbf{x}_T$. Because it captures the probability distribution across target number, target state and target type, the JMPD serves as the essential starting point for any multiple target tracking and sensor management task. Indeed, as discussed in Chapter 3 and Chapter 4, respectively, the JMPD specifies the crucial information required for implementation of POMDP or information theoretic sensor management algorithms.

Consider an Intelligence, Surveillance and Reconnaissance (ISR) system tasked with detecting, tracking and identifying all ground vehicles within a 100 square kilometer region. The system consists of one or more Ground Moving Target Indicator (GMTI) radars deployed on a set of airborne, near-space or space platforms. The system has agile radar beams that can be selectively pointed and operated in a number of different resolution modes. The sensor manager must automatically determine the best beam pointing directions and resolution modes. When the system is first deployed there is great uncertainty about how many targets are present as well as their locations and identities, so it operates in a low-resolution wide-area search mode. This wide-area search mode produces a number of tentative detections which are revisited using a higher resolution mode that has improved signal-to-noise+clutter ratio (SNCR). This reduces the uncertainty regarding the number of targets but the system still has great uncertainty regarding their locations and identities. Once targets are detected, they will be tracked and localized. Then the sensors will be tasked to produce high range resolution profiles or inverse synthetic aperture radar (ISAR) images that are used for automatic target identification. The JMPD captures the uncertainties of target positions and number of targets given past measurements.

At each stage in this process the sensor manager deploys the sensors so as to minimize the uncertainty regarding the targets in the scene. It may appear that this problem can be neatly solved by moving through a sequence of operations: detect, localize, track, identify. However, in reality one has a complex task mixture. Some tracking operations may also provide detection information on new targets; identification tasks include localization information and so on. Additionally, the system must constantly search for new targets that may have entered the scene. This requires a single entity that captures the information quality for all aspects of the problem: this is the role of the joint multi-target probability density. If we can develop numerically tractable methods to evaluate this JMPD, then its information content can be evaluated using methods developed in Chapter 4 and we can predict how the JMPD information content will change on average for a wide set of alternative sensing actions (alternative modes, beam pointing angles and so on). The optimal sensor manager is then the one that produces the largest expected increase in information for each action.

There are two features of the JMPD belief state approximation developed in this Chapter that should be emphasized: 1) the permutation symmetry of the JMPD has significant impact on algorithm design; and 2) the ability of the JMPD to perform multi-target tracking without the need for data association. To see why we require that the JMPD be symmetric under permutation of target indices, consider the scenario that a jeep is near hill A and a tank is near hill B. When we construct a probability density to describe this situation, the ordering of the arguments in the density cannot effect its value. If $\mathbf{x}_1 = \{jeep, x_1\}$ and $\mathbf{x}_2 = \{tank, x_2\}$, where the x_i are position and velocity states of the two objects, then we require that the probability density satisfy $p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_2, \mathbf{x}_1)$. Proper treatment of the this permutation symmetry has a significant impact on the development of efficient particle sampling methods for the JMPD.

Most target tracking solutions developed from about 1960 to the mid-1990's relied heavily on the use of the Kalman filter, linear Gaussian state dynamics, linear Gaussian measurements, and data association methods for assigning different measurements to different tracks. Recent progress on this approach is described in Chapter 11. Unlike methods depending on Kalman filtering and data association, developed in Chapter 11, the particle filtering approximation to the JMPD belief state described in this Chapter is a fully Bayesian approach that can handle non-Gaussian state or measurements and multiple targets-per-measurement without using data association heuristics.

Until recently, the literature in multi-target tracking was focussed on Kalman filtering-based techniques such as multiple hypothesis tracking (MHT) and joint probabilistic data association (JPDA) [33, 13, 14], discussed in Chapter 11. The fully Bayesian perspective on multiple target tracking adopted in this Chapter is more recent. Stone [223] developed a mathematical theory of multiple target tracking from a Bayesian point of view and Srivistava, Miller [176], and Kastella [130] did early work in this area. The issue with fully Bayesian approaches is computation of the JMPD - which suffers from the curse of dimensionality as the number of targets increases. The particle filtering approximations to the JMPD developed in this chapter is one way to reduce the complexity of the fully Bayesian approach.

Particle filters have been previously applied by others to extend MHT multi-target tracking approaches to non-linear and non-Gaussian target dynamics. In [119], Hue introduces the probabilistic multiple hypothesis tracker (PMHT), which is a blend between the traditional MHT and particle filtering. Others have blended JPDA and particle filtering ideas [127, 40].

The use of particle filters in fully Bayesian multi-target tracking problems also has a recent history. By directly approximating the JMPD, the BraM-

BLe [120] system, the independent partition particle filter (IPPF) of Orton and Fitzgerald [189] and the work of Maskell [172] eliminate the need for data association while retaining the flexibility of non-linear and non-Gaussian modeling.

The approach described in this Chapter builds on these fully Bayesian particle filtering approaches and adds a significant innovation with respect to numerics: the adaptive particle proposal method. By utilizing an adaptive sampling scheme that exploits independence when present, the particle filtering JMPD method described here provides significant computational advantages over brute-force methods.

In our work (as in [189]) each particle encapsulates multiple targets simultaneously. Put another way, instead of using one particle per target we use *one particle per scenario*. That is, a particle encodes a hypothesis about the entire multi-target state – which includes the number of targets and the state (position, velocity, etc.) of each target.

The fully Bayesian approach is distinguished from traditional approaches of MHT and JPDA as well as the approaches of Hue [119, 118] and others [127, 213, 79], which require thresholded measurements (detections) and a measurement-to-track association procedure. Further, by estimating the joint multi-target density rather than a many single target densities, our method explicitly models target correlations. These two features together, combined with the tractable numerical implementation discussed here, make the JMPD method a quite broadly applicable approach.

2. The Joint Multitarget Probability Density

The JMPD is the probability density on a jump-diffusion system of the type introduced for target-tracking applications in [223, 176]. The basic building block is the single target state space s. For ground target applications the target inertial state is $\mathbf{x} = [x, \dot{x}, y, \dot{y}]' \in s = \mathcal{R}^4$. At any instant the inertial state of the surveillance volume is determined by $\mathbf{X} \in \mathcal{S} \equiv \emptyset \cup \bigcup_{T=1}^{\infty} s^T$ where $\emptyset$ is a system with no targets and s^T is a T-target system characterized by the concatenated target state $\mathbf{X} = [\mathbf{x}_1', \ldots, \mathbf{x}_T']'$, where $\mathbf{x}'$ denotes the transpose of vector $\mathbf{x}$. The non-negative integer T is the number of targets in the scene. For fixed T the target motion element of the dynamics undergoes a diffusive evolution obeying an Itô equation of the type used in many nonlinear filtering applications. Since T can vary in time, it makes discrete jumps as targets enter or depart the scene. This is the "jump" part of the jump-diffusion dynamics.

For tracking and sensor management applications we construct the posterior density conditioned on a set of observations $\mathbf{y}^i$ occurring at times τ^i. Then $\mathbf{Y}^k$ is the collection of measurements up to and including time τ^k, $\mathbf{Y}^k = \{\mathbf{y}^1, \mathbf{y}^2, ...\mathbf{y}^k\}$. Each observation or "scan" $\mathbf{y}^i$ may be a single measurement or a vector of measurements made at time i. The posterior $p(\mathbf{X}^k, T^k|\mathbf{Y}^k)$ is what we mean by the "Joint Multitarget Probability Density" (JMPD). Notionally, in the absence of any measurements (i.e., at initialization) the JMPD is uniform in target number and uniform in target state conditioned on target number for all target numbers. In practice, this may be modified by prior information such as the location of roadways, waterways, and knowledge of terrain.

For simplicity, we will typically suppress time indices when they are not important to the discussion. Furthermore, when it is clear by context, we may write the JMPD $p(\mathbf{X}, T|\mathbf{Y}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_T|\mathbf{Y})$ as simply $p(\mathbf{X}|\mathbf{Y})$ and similarly for transition densities and likelihood functions. For example,

- $p(\emptyset|\mathbf{Y})$, is the posterior probability density for no targets in the surveillance volume
- $p(\mathbf{x}_1|\mathbf{Y})$, is the posterior probability density for one target with state $\mathbf{x}_1$
- $p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3|\mathbf{Y})$, is the posterior probability density for three targets with respective states $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$

The subset of $\mathcal{S}$ with fixed T is referred to as the T-target sector of $\mathcal{S}$. The state vector $\mathbf{x}_t$ corresponding to a single target is a partition of $\mathbf{X}$. The JMPD permutation symmetry discussed in the introduction can now be made precise: the JMPD is symmetric on each sector under partition permutation,

$$p(\mathbf{x}_1, ..., \mathbf{x}_T|\mathbf{Y}^k) = p(\mathbf{x}_{\pi(1)}, ..., \mathbf{x}_{\pi(T)}|\mathbf{Y}^k), \tag{4.1}$$

where π is one of the $T!$ permutations of the T labels, $\pi : i \to \pi(i)$.

We can gain insight into the role of this permutation symmetry by examining the entropy of the JMPD compared with an unsymmetrical density. As shown in Appendix 1, the entropy is

$$H(p) \equiv \sum_{T=0}^{\infty} \int d\mathbf{x}_1 \cdots d\mathbf{x}_T p(\mathbf{X}|\mathbf{Y}) \ln(p(\mathbf{X}|\mathbf{Y})). \tag{4.2}$$

Consider the case of two well-localized targets restricted to the real line with separation $2d$ given by the Gaussian sum

$$p(x_1, x_2) = \delta_{2,T}(1/4\pi)\big[\exp\big(-[(x_1-d)^2+(x_2+d)^2]/2\big) + \exp\big(-[(x_1+d)^2+(x_2-d)^2]/2\big)\big], \quad (4.3)$$

where $\delta_{i,j}$ is the Kronecker δ-function. When $d \gg 1$ (5.3) gives two well-separated peaks near $(x_1, x_2) = \pm(d, -d)$ and the entropy can be approximately evaluated as $H_{d\gg 1} \approx 1 + \log(4\pi)$. On the other hand as $d \to 0$ the peaks coalesce to form a single mono-modal distribution, reducing the entropy to $H_{d=0} = 1 + \log(2\pi)$.

If we model this 2-target problem using the permutation non-symmetric expression $p_{NS}(x_1, \ldots, x_T) = \delta_{2,T} 1/2\pi \exp\big(-[(x_1-d)^2+(x_2+d)^2]/2\big)$ direct calculation shows that the entropy is $H_{NS} = 1 + \log(2\pi)$ independent of the target separation. For the properly symmetrized density there is a relative reduction in entropy when targets are close together. This corresponds to a reduction in the relevant phase space for the density in order to develop efficient particle sampling schemes. To treat this aspect of the problem, we have developed sampling techniques that explicitly account for the permutation symmetry. This is one of the key features of the JMPD approach that differentiates it from other popular trackers such has the Multiple Hypothesis Tracker (MHT) or Joint Probability Density Association (JPDA) tracker.

2.1 General Bayesian Filtering

Construction of a filter to update the posterior density as measurements come in proceeds according to the usual rules of Bayesian filtering. Define the aggregate target state at time k as the vector $[\mathbf{X}^k, T^k]$. Under a Markovian model, the conditional density of the current target state given the past states is given by $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ and will be referred to as the kinematic prior (KP). The kinematic prior describes probabilistically how the state of the multi-target system evolves over time. It includes models of target motion, target birth and death, and any additional prior information that may exist such as terrain- and road-constraints. The time-updated density is computed via the *time update* equation:

$$p(\mathbf{X}^k, T^K|\mathbf{Y}^{k-1}) = \sum_{T^{(k-1)}=0}^{\infty} \int d\mathbf{X}^{k-1} p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1}) p(\mathbf{X}^{k-1}, T^{k-1}|\mathbf{Y}^{k-1}). \quad (4.4)$$

Observe that the number of targets in the scene can change with time update when $T^k \neq T^{(k-1)}$ to capture the effect of target birth and death. The *measurement update* equation uses Bayes' rule to update the posterior density with

a new measurement $\mathbf{y}^k$:

$$p(\mathbf{X}^k, T^k|\mathbf{Y}^k) = \frac{p(\mathbf{y}^k|\mathbf{X}^k, T^k)p(\mathbf{X}^k, T^k|\mathbf{Y}^{k-1})}{p(\mathbf{y}^k|\mathbf{Y}^{k-1})}. \tag{4.5}$$

2.2 Non-Linear Bayesian Filtering for a Single Target

Non-linear Bayesian filtering generalizes linear filtering equations that govern the Kalman filter for linear Gaussian dynamics and measurements. A convenient starting point is the continuous-discrete non-linear filtering theory developed in Jazwinsky [123, Ch. 5]. To apply this theory, let $\mathbf{x}_\tau$ denote the vector of inertial coordinates (position, velocity) of one of the targets indexed over the continuous time variable τ. In this notation the discrete time state vector $\mathbf{x}^k = \mathbf{x}_{\tau_k}$ is the time sample of $\mathbf{x}_\tau$ at discrete time $\tau = \tau_k$, $k = 1, 2, \ldots$. We assume that the continuous time state vector $\mathbf{x}_\tau$ evolves according to the Itô stochastic differential equation

$$d\mathbf{x}_\tau = \mathbf{f}(\mathbf{x}_\tau, \tau)d\tau + \mathbf{g}(\mathbf{x}_\tau, \tau)d\beta_\tau, \;\; \tau \geq \tau_0, \tag{4.6}$$

where $\mathbf{f}$ is a known vector valued non-linear diffusion function, $\mathbf{g}$ is a matrix valued noise scaling function, and $\beta_\tau, \tau \geq \tau_0$ is a vector valued Brownian motion process with covariance $E[d\beta_\tau d\beta_\tau^\top] = \mathbf{Q}(\tau)d\tau$. Note that both $\mathbf{f}$ and $\mathbf{g}$ can be nonlinear functions of $\mathbf{x}_\tau$ in which case the target state evolves as a non-Gaussian process.

Using the Itô state equation model (5.6) and measurement update equation (5.5) the continuous time posterior density $p(\mathbf{x}_\tau|\mathbf{Y}^k)$ for $\tau \in [\tau_k, \tau_{k+1}]$ of the state can be obtained by solving the Fokker-Planck Equation (FPE)

$$\frac{\partial}{\partial \tau} p(\mathbf{x}_\tau|\mathbf{Y}^k) = L\left(p(\mathbf{x}_\tau|\mathbf{Y}^k)\right), \qquad \tau_k \leq \tau < \tau_{k+1}, \tag{4.7}$$

with initial condition

$$p(\mathbf{x}_\tau|\mathbf{Y}^k)\Big|_{\tau=\tau_k} = p(\mathbf{x}^k|\mathbf{Y}^k),$$

obtained from (5.5). In (5.7) the linear differential operator L is specified by the functions $\mathbf{f}$ and $\mathbf{g}$.

$$L(p) \equiv -\sum_{i=1}^{n} \frac{\partial(\mathbf{f}_i p)}{\partial \mathbf{x}_i} + \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial((\mathbf{g}\mathbf{Q}\mathbf{g}^\top)_{ij} p)}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, \tag{4.8}$$

with i, j indexing components of $\mathbf{x}_\tau$.

In the case that the measurements $\mathbf{Y}^k$ are Gaussian and linear in the state $\mathbf{x}^k$, and the Itô diffusion functions $\mathbf{f}$ and $\mathbf{g}$ are linear functions, the posterior density is Gaussian and the measurement and time update equations of the Bayesian filter can be solved by the linear Kalman filter for estimating the state [123].

2.3 Accounting for Target Birth and Death

The FPE operator (5.8) determines how the single target probability density evolves in time due to the stochastic target kinematics. The time evolution of the full multi-target JMPD is simply a linear superposition of FPE operators acting on the partitions of each T-target segment of the concatenated inertial state vector $\mathbf{X}$,

$$\frac{\partial}{\partial t}p(\mathbf{X})\Big|_{FPE} = \sum_{t=1}^{T} L_t\left(p(\mathbf{X})\right). \tag{4.9}$$

The other aspect of temporal evolution that must be addressed is the effect of changes in the number of targets, sometimes referred to as target birth and death. Here we adopt a Markovian birth-death process model. Births of targets occur with birth rate $\Lambda^+(\mathbf{x},t)$, where the arguments $\mathbf{x}$ and t denote the locations and times at which a target emerges. For simplicity assume that the birth rate is constant over time so that $\Lambda^+(\mathbf{x},t) = \Lambda^+(\mathbf{x})$.

Target birth must maintain the JMPD permutation symmetry property. This requires explicit symmetrization through summation over all T-target permutations

$$\begin{aligned}\frac{\partial}{\partial t}p(\mathbf{X})\Big|_{T-1\to T} &= \frac{1}{T!}\sum_{\pi} p\left(\mathbf{x}_{\pi_1},\ldots,\mathbf{x}_{\pi_{T-1}}\right)\Lambda^+(\mathbf{x}_{\pi_T}) \qquad (4.10)\\ &= \frac{1}{T}\sum_{\rho} p\left(\mathbf{x}_{\rho_1},\ldots,\mathbf{x}_{\rho_{T-1}}\right)\Lambda^+(\mathbf{x}_{\rho_T}),\end{aligned}$$

where ρ is the subset of permutations $\rho_i : (1,\ldots,T) \to (1,\ldots,i-1,T,i+1,\ldots,T-1)$, $i = 1,\ldots,T$. Since $p\left(\mathbf{x}_1,\ldots,\mathbf{x}_{T-1}\right)$ is permutation symmetric, only T terms are required in this summation. Target birth also contributes a *loss term* in the T-target density through transitions from T-target states to $(T+1)$-target states

$$\frac{\partial}{\partial t}p(\mathbf{X})\Big|_{T\to T+1} = -\sum_{t=1}^{T}\int d\mathbf{x}_t \Lambda^+(\mathbf{x}_t) p\left(\mathbf{x}_1,\ldots,\mathbf{x}_T\right). \tag{4.11}$$

Target death is treated similarly defining the state-space dependent death rate $\Lambda^-(\mathbf{x})$. The T-target JMPD sector decreases when the T targets are

present and one dies while it increases when there are $T+1$ targets and one dies. These contribute time dependencies

$$\begin{aligned} \frac{\partial}{\partial t} p(\mathbf{X})\Big|_{T \to T-1} &= -\sum_{t=1}^{T} \Lambda^{-}(\mathbf{x}_t) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_T\right) \\ &= -T\Lambda^{-}(\mathbf{x}_T) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_T\right), \end{aligned} \tag{4.12}$$

and

$$\begin{aligned} \frac{\partial}{\partial t} p(\mathbf{X})\Big|_{T+1 \to T} &= +\sum_{t=1}^{T+1} \int d\mathbf{x}_t \Lambda^{-}(\mathbf{x}_t) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_{T+1}\right) \\ &= (T+1) \int d\mathbf{x}_{T+1} \Lambda^{-}(\mathbf{x}_{T+1}) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_{T+1}\right). \end{aligned} \tag{4.13}$$

Combining these terms leads to the full multi-target Fokker-Planck equation

$$\begin{aligned} \frac{\partial}{\partial t} p(\mathbf{X}) &= \sum_{t=1}^{T} L_t\left(p(\mathbf{X})\right) \\ &+ \frac{1}{T} \sum_{\rho} p\left(\mathbf{x}_{\rho_1}, \ldots, \mathbf{x}_{\rho_{T-1}}\right) \Lambda^{+}(\mathbf{x}_{\rho_T}) - \sum_{t=1}^{T} \int d\mathbf{x}_t \Lambda^{+}(\mathbf{x}_t) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_T\right) \\ &- T\Lambda^{-}(\mathbf{x}_T) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_T\right) + (T+1) \int d\mathbf{x}_{T+1} \Lambda^{-}(\mathbf{x}_{T+1}) p\left(\mathbf{x}_1, \ldots, \mathbf{x}_{T+1}\right). \end{aligned} \tag{4.14}$$

The posterior JMPD, $p(\mathbf{X}^k|\mathbf{Y}^k)$, is propagated to $p(\mathbf{X}^{k+1}|\mathbf{Y}^k)$ using the same multi-target Fokker-Plank equation with initial condition $p(\mathbf{X}_\tau|\mathbf{Y}^k)\big|_{\tau=\tau_k} = p(\mathbf{X}^k|\mathbf{Y}^k)$, analogous to eq. (5.7).

2.4 Computing Renyi Divergence

Chapter 4 discuses several alternative information theoretic metrics that can be used in sensor management applications. The expected Renyi divergence can be evaluated for JMPD and used to predict the information gain expected for a set of alternative sensing actions. For sensor management the relevant quantity is the divergence between the predicted density $p(\mathbf{X}^k, T^k|\mathbf{Y}^{k-1})$ and the updated density after a measurement is made, $p(\mathbf{X}^k, T^k|\mathbf{Y}^k)$ given by

$$\begin{aligned} &D_\alpha\left(p(\cdot|\mathbf{Y}^k)||p(\cdot|\mathbf{Y}^{k-1})\right) = \\ &\quad \frac{1}{\alpha-1} \ln \sum_{T^k=0}^{\infty} \int d\mathbf{X}^k p(\mathbf{X}^k, T^k|\mathbf{Y}^k)^\alpha p(\mathbf{X}^k, T^k|\mathbf{Y}^{k-1})^{1-\alpha}. \end{aligned} \tag{4.15}$$

We follow the principle that measurements should be selected that maximize the information gain, i.e., the divergence between the post-updated density, $p(\mathbf{X}^k, T^k|\mathbf{Y}^k)$, and the pre-updated density, $p(\mathbf{X}^k, T^k|\mathbf{Y}^{k-1})$. To do this, let a ($1 \leq a \leq A$) index feasible sensing actions such as sensor mode selection and sensor beam positioning. The expected value of equation (5.14) can be written as an integral over all possible outcomes $\mathbf{y}_a$ when performing sensing action a:

$$\langle D_\alpha \rangle_a = \int d\mathbf{y} p(\mathbf{y}|\mathbf{Y}^{k-1}, a) D_\alpha \left(p(\cdot|\mathbf{Y}^k) || p(\cdot|\mathbf{Y}^{k-1}) \right). \tag{4.16}$$

2.5 Sensor Modeling

Implementing the Bayes update (5.5) requires evaluating the measurement likelihood function $p(\mathbf{y}|\mathbf{X})$. We use an association-free model instead of the more common associated measurement model. In the associated measurement model, e.g. JPDA, an observation vector consists of M measurements, denoted $\mathbf{y} = (y_1, \ldots, y_M)$ where $\mathbf{y}$ is composed of threshold exceedances, i.e., valid detections and false alarms. The model usually assumes that each measurement is generated by a single target (the valid measurements) or by clutter and noise (false alarms). The valid measurements are related (possibly nonlinearly) to the target state in a known way. If measurement m is generated by target t, then it is a realization of the random process $y_m \sim H_t(\mathbf{x}_t, w_t)$. False alarms have a known distribution independent of the targets and the targets have known detection probability P_d (often modeled as constant for all targets). The origin of each measurement is unknown so a significant portion of any algorithm based on the associated measurement model goes to determining how the measurements correspond to possible targets either through some sort of likelihood weighting (MHT and PDA) or a maximum likelihood assignment process (e.g. multidimensional assignment).

The associated measurement model is widely used and a number of successful tracking systems are based on it. Its practical advantage is that it breaks the tracking problem into two disjoint sub-problems: data association and filtering. While the data association problem is challenging, filtering can be performed using a linearized approach, such as the extended Kalman filter, which is quite efficient. However, there are two disadvantages to the associated measurement model. First, it is based on an artificial idealization of how sensors work in that it assumes each valid detection comes from a single target. This makes it challenging to treat measurement interactions amongst close targets. Second, it requires solution of the data association problem, which usually consumes a large amount of computing resources.

While most data association systems ultimately rely on some variant of the Kalman filter, the use of nonlinear filtering methods such as the particle filter frees us to explore new approaches such as association-free methods for computing the full Bayesian posterior density. This type of model has been used in track-before-detect algorithms, such as the "Unified Data Fusion" work of Stone et. al. [223] and in [130]. There are several advantages to the association-free method. First, it requires less idealization of the sensor physics and can readily accommodate issues such as merged measurements, side-lobe interference amongst targets and velocity aliasing. Second, it eliminates the combinatorial bottleneck of the associated-measurement approach. Finally, it simplifies the processing of unthresholded measurements to enable improved tracking at lower target SNR.

The starting point for developing an association-free model is the recognition that nearly all modern sensor systems produce multidimensional arrays of pixelized data in some form. The sensor measures return energy from a scene, digitizes it and performs signal processing to produce measurements consisting of an array real or complex amplitudes. This can be 1-dimensional (a bearing-only passive acoustic or electronic sensing measures system), 2-dimensional (an electro-optical imager), 3-dimensional (the range, bearing, range-rate measurements of a GMTI system), or higher dimensional data.

The measurement likelihood $p(\mathbf{y}|\mathbf{X})$ describes how amplitudes in the pixel array depend on the state of all of the targets and background in the surveillance region. To be precise, a sensor scan consists of M pixels, and a measurement $\mathbf{y}$ consists of the pixel output vector $\mathbf{y} = [y_1, \ldots, y_M]'$, where y_i is the output of pixel i. y_i can be an integer, real, or complex valued scalar, a vector or even a matrix, depending on the sensor. If the data are thresholded, then each y_i will be either a 0 or 1. Note that for thresholded data, $\mathbf{y}$ consists of both threshold exceedances and non-exceedances. The failure to detect a target at a given location can have as great an impact on the posterior distribution as a detection.

In the simulation studies described below we model pixel measurements as conditionally independent, yielding

$$p(\mathbf{y}|\mathbf{X}) = \prod_i p(y_i|\mathbf{X}). \tag{4.17}$$

Let $\chi_i(\mathbf{x}_t)$ denote the indicator function for pixel i, defined as $\chi_i(\mathbf{x}_t) = 1$ when a target in state $\mathbf{x}_t$ projects into sensor pixel i (i.e., couples to pixel i) and $\chi_i(\mathbf{x}_t) = 0$ when it does not. A pixel can couple to multiple targets and single target can contribute to the output of multiple pixels, say, by coupling through side-lobe responses. The indicator function for the joint multi-target

state is the logical disjunction

$$\chi_i(\mathbf{X}) = \bigvee_{t=1}^{T} \chi_i(\mathbf{x}_t) \tag{4.18}$$

The set of pixels that couple to $\mathbf{X}$ is

$$i_{\mathbf{X}} = \{i | \chi_i(\mathbf{X}) = 1\} \tag{4.19}$$

For the pixels that do not couple to any targets the measurements are characterized by the background distribution, denoted $p_0(y_i)$ (this can generally depend on where the pixel is within the scene but here we assume a constant background). With this, (5.17) becomes

$$p(\mathbf{y}|\mathbf{X}) = \prod_{i \in i_{\mathbf{X}}} p(y_i|\mathbf{X}) \prod_{i \notin i_{\mathbf{X}}} p_0(y_i) \propto \prod_{i \in i_{\mathbf{X}}} \frac{p(y_i|\mathbf{X})}{p_0(y_i)} \tag{4.20}$$

In the last step of (5.20) we have dropped the $\mathbf{X}$-independent factor $\prod_i p_0(y_i)$ since it makes no contribution to the JMPD measurement update.

To completely specify the measurement likelihood we must determine how targets couple to the individual pixels. In our simulations the sensor response within pixel i is uniform for targets in i and vanishes for targets outside pixel i. This is equivalent to modeling the point-spread function as a boxcar. It is convenient to define the occupation number $n_i(\mathbf{X})$ for pixel i as the number of targets in $\mathbf{X}$ that lie in i. The single target signal-noise-ratio (SNR), assumed constant across all targets, is denoted λ. We assume that when multiple targets lie within the same pixel their amplitudes add non-coherently. Then the effective SNR when there are n targets in a pixel is $\lambda_n = n\lambda$ and we use $p_n(y_i)$ to denote the pixel measurement distribution. In this model the measurement distribution in pixel i depends only on its occupation number and (5.20) becomes

$$p(\mathbf{y}|\mathbf{X}) \propto \prod_{i \in i_{\mathbf{X}}} \frac{p_{n_i(\mathbf{X})}(y_i)}{p_0(y_i)}. \tag{4.21}$$

The effect of the sensor on the measurement likelihood can be determined by detailed modeling, e.g. studying the radar ambiguity function and radar processing noise statistics for different waveforms as in Chapter 11. We adopt a simple approximation here which reduces the effect of the sensor to two scalar parameters: the range-azimuth-elevation cell resolution and the SNR. In particular, we assume a cell-averaged scalar Rayleigh-distributed measurement corresponding to envelope detected signals under a Gaussian noise model. Such a model has been used to model interfering targets in a monopulse radar system [36, 233] and to model clutter and target returns in turbulent environments

[101]. Rayleigh models are also often used for diffuse fading channels. In the unthresholded case, the likelihood function is

$$p_n(y) = \frac{y}{1+n\lambda} \exp\left(-\frac{y^2}{2(1+n\lambda)}\right). \tag{4.22}$$

When the tracker only has access to thresholded measurements, we use a constant false-alarm rate (CFAR) model for the sensor. The background false alarm rate is set to a level $P_f \in [0,1]$ by selecting a threshold η such that:

$$P(y > \eta | \text{clutter alone}) = \int_\eta^\infty p_n(y)dy = P_f. \tag{4.23}$$

Under the Rayleigh model the detection probability is

$$P_{d,n} = P_f^{\frac{1}{1+n\lambda}}, \tag{4.24}$$

where n is the number of targets in the cell. This extends the usual relation $P_d = P_f^{\frac{1}{1+\lambda}}$ for thresholded Rayleigh random variables at SNR λ [33].

Note that this simple thresholded Rayleigh model can be easily extended to account for other sensor characteristics, e.g. its non-Gaussian noise, amplitude saturation, or other non-linearities. Once the likelihood function has been specified, as in (5.20), the posterior density (JMPD) can be updated with new measurements via (5.5). However, as the likelihood function (5.20) is not Gaussian this requires some form of function approximation. In the next section the particle filtering approximation to the JMPD is described.

3. Particle Filter Implementation of JMPD

We begin with a brief review of the Sampling Importance Resampling (SIR) particle filter for single targets. This can be generalized directly to produce a SIR JMPD particle filter for multiple targets. The salient feature of the SIR filter is that it uses the kinematic prior as the so-called importance density used to propose new particle states in the time update step.

The SIR filter is a relatively inefficient particle filter and the JMPD SIR filter requires a very large number of particles to track even a modest number of targets. This is largely due to the high dimensionality of the multi-target phase space. Also, when targets are close together we have seen that the entropy of the JMPD is reduced relative to that of a permutation non-symmetric multi-target density. The smaller entropy means that the relevant phase space region is further reduced, placing a great premium on efficient sampling methods.

While the kinematic prior automatically preserves permutation symmetry, care is required when using more sophisticated schemes and has led us to develop Independent Partition (IP), Coupled Partition (CP), Adaptive Partition (AP) and Joint Sampling (JS) methods detailed below.

3.1 The Single Target Particle Filter

To implement a single target particle filter, the single target density of interest, $p(\mathbf{x}|\mathbf{Y})$, is approximated by a set of N_{part} weighted samples (particles) [10, 98]:

$$p(\mathbf{x}|\mathbf{Y}) \approx \sum_{p=1}^{N_{part}} w_p \delta_D(\mathbf{x} - \mathbf{x}_p), \tag{4.25}$$

where δ_D is the Dirac delta function.

The time update (5.4) and measurement update (5.5) are simulated by the following three step recursion of Table 5.1. First, the particle locations at time k are generated using the particle locations $\mathbf{x}_p$ at time $k-1$ and the current measurements $\mathbf{y}^k$ by sampling from an importance density, denoted $q(\mathbf{x}^k|\mathbf{x}^{k-1}, \mathbf{y}^k)$. The design of the importance density is a well studied area [78] since it plays a key role in the efficiency of the particle filter algorithms. It is known that the optimal importance density (OID) is $p(\mathbf{x}^k|\mathbf{x}^{k-1}, \mathbf{y}^k)$, but this is usually prohibitively difficult to sample from. The kinematic prior $p(\mathbf{x}^k|\mathbf{x}^{k-1})$ is a simple but suboptimal choice for the importance density.

The second step in the implementation of the particle filter is to update particle weights according to [10].

$$w_p^k = w_p^{k-1} \frac{p(\mathbf{y}^k|\mathbf{x}_p^k) p(\mathbf{x}_p^k|\mathbf{x}_p^{k-1})}{q(\mathbf{x}_p^k|\mathbf{x}_p^{k-1}, \mathbf{y}^k)}. \tag{4.26}$$

When using the kinematic prior as the importance density, the weight equation reduces to $w_p^k = w_p^{k-1} p(\mathbf{y}^k|\mathbf{x}_p^k)$.

Finally, particle resampling is performed to prevent particle degeneracy. Without resampling, the variance of the particle weights increases with time, yielding a single particle with all the weight after a small number of iterations [77]. Resampling may be done on a fixed schedule or adaptively, based on variance of the weights. The particle filter algorithm that uses the kinematic prior as the importance density $q(\mathbf{x}_p^k|\mathbf{x}_p^{k-1}, \mathbf{y}^k)$ and resamples at each time step is called the sampling importance resampling (SIR) algorithm.

Table 4.1. SIR Single Target Particle Filter (Table I from [149] which is ©2005 IEEE - used with permission)

1 For each particle p, $p = 1, \cdots, N_{part}$,
   (a) **Particle proposal**: Sample $\mathbf{x}_p^k \sim q(\mathbf{x}^k|\mathbf{x}^{k-1}, \mathbf{y}^k) = p(\mathbf{x}|\mathbf{x}_p^{k-1})$
   (b) **Particle weighting**: Compute $w_p^k = w_p^{k-1} p(\mathbf{y}|\mathbf{x}_p)$ for each p.
2 **Weight normalization**: Normalize w_p^k to sum to one over p
3 **Particle resampling**: Resample N_{part} particles with replacement from $\mathbf{x}_p$ based on the distribution defined by w_p

3.2 The Multitarget Particle Filter

To implement the particle filter for approximating the multiple target JMPD we must sample from the surveillance volume belief state defined by (5.5). We approximate the joint multi-target probability density $p(\mathbf{X}|\mathbf{Y})$ by a set of N_{part} weighted samples. For $p = 1, \ldots, N_{part}$, particle p has T_p targets and is given by

$$\mathbf{X}_p = [\mathbf{x}'_{p,1}, \ldots, \mathbf{x}'_{p,T_p}]'. \tag{4.27}$$

Defining

$$\delta(\mathbf{X} - \mathbf{X}_p) = \begin{cases} 0 & T \neq T_p \\ \delta_D(\mathbf{X} - \mathbf{X}_p) & \text{otherwise} \end{cases}, \tag{4.28}$$

the particle filter approximation to the JMPD is given by a set of particles X_p and corresponding weights w_p as

$$p(\mathbf{X}|\mathbf{Y}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p), \tag{4.29}$$

where $\sum w_p = 1$. This is analogous to a multiple hypothesis tracker in that different particles in the sample may correspond to different hypotheses for the number T_p of targets in the surveillance region.

With these definitions the SIR particle filter extends directly to JMPD filtering, as shown in Table 5.2. This simply proposes new particles at time k using the particles at time $k-1$ and the target kinematics model (5.6) while (5.21) is used in the weight update. Target birth and death given in (5.14) corresponds to probabilistic addition and removal of partitions within particles.

Table 4.2. SIR Multitarget Particle Filter (Table II from [149] which is ©2005 IEEE - used with permission)

1 For each particle p, $p = 1, ..., N_{part}$,
   (a) **Particle proposal**: Sample $\mathbf{X}_p^k \sim q(\mathbf{X}, T|\mathbf{X}_p^{k-1}, T_p^{k-1}, \mathbf{y}^k) = p(\mathbf{X}, T|\mathbf{X}_p^{k-1}, T_p^{k-1})$
   (b) **Particle weighting**: Compute $w_p^k = w_p^{k-1} p(\mathbf{y}|\mathbf{X}_p^k)$ for each p.
2 **Weight normalization**: Normalize w_p^k to sum to one over p
3 **Particle resampling**: Resample N_{part} particles with replacement from $\mathbf{X}_p^k$ based on w_p^k

3.3 Permutation Symmetry and Improved Importance Densities for JMPD

The probability of generating a high-likelihood particle proposal using the kinematic prior of the SIR filter decreases as the number of partitions in a particle grows. This is due to the fact that the likelihood of multi-target proposal is roughly the product of the likelihoods for each partition on its own (this is made precise below). As a result, if a few partitions fall in low-likelihood regions the entire joint proposal likelihood is low. This suggests that improved JMPD importance densities can be developed by first generating high-likelihood proposals for each partition and then combining high-likelihood partition proposals to generate multi-target proposals. The key challenge is that, due to the JMPD permutation symmetry, there is no unique ordering of the partitions in the JMPD particles: the single target state vector corresponding to a particular target can appear in different partitions in different particles. Even if we initialize a filter with ground truth so that all of the particles correspond to the same target ordering, the order generally changes in some particles as targets approach each other and there is measurement to target association uncertainty (we refer to this as "partition swapping"). This is not a pathology of the approach and is not to be regarded as symptomatic of an incorrect formulation. It is, in fact, a reflection of the physics of the underlying problem. The key to addressing this issue is to impose a particular ordering on the partitions in all of the targets. Then the correspondence between partitions in different particles is well defined, allowing us to develop partition-based proposal schemes that significantly improve the efficiency of the JMPD-PF. This partition ordering does not violate the permutation symmetry requirement of the JMPD if we approximate it by an appropriately symmetrized function of the sample particles.

To gain more insight into this issue, consider the JMPD for two targets in one dimension introduced in (5.3). For widely separated, well-localized targets

the JMPD has two distinct peaks in the (x_1, x_2) plane. Samples from $p(x_1, x_2)$ can fall in either the upper or lower diagonal half-plane. For convenience we can choose to approximate p using samples with a particular permutation symmetry, say only those in the upper half plane $x_2 \geq x_1$, explicitly symmetrizing as needed. Now as the targets approach each other, the peaks of the JMPD approach the diagonal $x_1 = x_2$. Even if we initially approximate the density with particles that only lie in the upper half-plane, when the targets approach each other two peaks coalesce along the diagonal. This effect leads to the entropy reduction noted in Section 2. If we generate random samples from the JMPD where the targets are close together, some samples will lie in the upper half-plane while others are in the lower half-plane. This suggests the following strategy: When the targets are widely separated we can impose a fixed partition ordering on the particles and use that ordering to construct efficient proposals. When the targets are close together, we must account for the fact that different samples may have different partitions as we construct proposals. Proposals that are more efficient than kinematic proposals can be constructed, but we incur additional overhead when the targets are close together.

3.4 Multi-target Particle Proposal Via Individual Target Proposals

While the kinematic prior is simple to implement, it requires a large number of particles (see Figure 5.5 below). The kinematic prior does not account for the fact that a particle represents many targets. Targets that are far apart in measurement space behave independently and should be treated as such. Another drawback to kinematic prior is that current measurements are not used when proposing new particles. These considerations taken together result in a very inefficient use of particles and therefore require large numbers of particles to successfully track.

To overcome the deficiencies mentioned above, we have employed alternative particle proposal techniques which bias the proposal process towards the measurements and allow for factorization of the target state when permissible. These strategies propose each partition (target) in a particle separately, and form new particles as the combination of the proposed partitions. We describe several methods here, beginning with the independent partitions (IP) method of [189] and the coupled partitions (CP) method. The basic idea of both CP and IP is to construct particle proposals at the target level, incorporating the measurements so as to bias the proposal towards the optimal importance density. We show that each has benefits and drawbacks and propose an adaptive partition (AP) method which automatically switches between the two as appropriate.

The permutation symmetry of the JMPD must be carefully accounted for when using these sampling schemes. The CP method proposes particles in a permutation invariant manner, however it has the drawback of being computationally demanding. When used on all partitions individually, the IP method is not permutation invariant. Our solution is to perform an analysis of the particle set to determine which partitions require the CP algorithm because they are involved in partition swapping and which partitions may be proposed via the IP method. This analysis leads to the AP method of proposal which is permutation invariant.

3.4.1 Independent-Partition (IP) Method. Summarized in Table 5.3, the independent partition (IP) method of Orton [189] is a convenient way to propose particles when part or all of the joint target posterior density factors. The Independent-Partition (IP) method proposes a new partition independently as follows. For a partition t, each particle at time $k-1$ has it's t^{th} partition proposed via the kinematic prior and weighted by the measurements. From this set of N_{part} weighted estimates of the state of the t^{th} target, we select N_{part} samples (with replacement) to form the t^{th} partition of the particles at time k.

Note that the importance density q is no longer simply the model of target kinematics $p(\mathbf{X}^k|\mathbf{X}^{k-1})$ as in the SIR Multitarget particle filter. Therefore the weight given by the weight equation (5.26) does not simply become the likelihood $p(\mathbf{y}^k|\mathbf{X}^k)$. There is a bias added which emphasizes partitions with high likelihood. To account for this sampling scheme, the biases corresponding to each particle for each target, $b_{p,t}$, are retained to use in conjunction with the likelihood $p(\mathbf{y}^k|\mathbf{X}^k)$ when computing particle weights.

The IP method is predicated on the assumption that partition t in each particle corresponds to the same target. Therefore, the partitions in each particle must be identically ordered before this method is applied. If IP is applied to particles that have different orderings of partitions, multiple targets will be grouped together and erroneously used to propose the location of a single target.

3.4.2 Coupled Partition (CP) Proposal Method. When the marginal posterior target distributions for different targets begin to overlap, the corresponding partitions are coupled and the IP method is no longer applicable. This situation requires a Coupled Partition (CP) scheme. We proceed as follows (see table 5.4). To propose partition t of particle p, CP generates R possible realizations of the future state using the kinematic prior. The R proposed future states are then given weights using the current measurements and

Table 4.3. Independent Partition Particle Filter (Table III from [149] which is ©2005 IEEE - used with permission)

1 For each partition, $t = 1 \cdots T_{max}$,

   (a) **Partition Proposal:** Propose partition t via Independent Partition Subroutine

2 **Particle weighting:** Compute $w_p^k = w_p^{k-1} * \frac{p(\mathbf{y}|\mathbf{X}_p)}{\prod_{t=1}^{T_p} b_{p,t}}$

3 **Weight normalization:** Normalize w_p^k to sum to one over p.

Independent Partition Subroutine for Target t:

1 For each particle $p = 1, ..., N_{part}$,

   (a) **Particle partition proposal:** Sample $\mathbf{X}_{p,t}^* \sim p(\mathbf{x}|\mathbf{X}_{p,t}^{k-1})$

   (b) **Particle partition weighting:** Compute $\omega_p = p(\mathbf{y}|\mathbf{X}_{p,t}^*)$

2 **Partition weight normalization:** Normalize ω to sum to one over p.

3 For each particle $p = 1, ..., N_{part}$,

   (a) **Index selection:** Sample an index j from the distribution defined by ω

   (b) **Particle partition selection:** Set $\mathbf{X}_{p,t} = \mathbf{X}_{j,t}^*$

   (c) **Bias balancing:** Retain bias of sample, $b_{p,t} = \omega_j$

a single representative is selected. This process is repeated for each particle until the t^{th} partition for all particles has been formed. This is an auxiliary particle filter of the type suggested in [194] where the multiplicity R plays the role of the auxiliary variable. As in the IP method, the final particle weights must be adjusted for biased sampling.

Stated in the language of genetic algorithms, the difference between CP and IP is that CP "maintains pedigree," i.e., all of the partitions in a new proposed particle must have come from a common ancestor while IP permits cross-breeding from different ancestors. Target birth and death are included in both CP and IP algorithms by adding or deleting partitions as determined by the target birth and death rates.

3.4.3 Adaptive Particle Proposal Method.

IP and CP can be combined adaptively to provide a scheme that delivers the speed advantage of IP for partitions that correspond to widely separated targets (usually the majority of targets) together with improved tracking for coupled targets. The Adaptive-Partition (AP) method analyzes each partition separately. Partitions that are sufficiently well separated according to a given metric (see below)

Table 4.4. Coupled Partition Particle Filter (Table IV from [149] which is ©2005 IEEE - used with permission)

1 For each partition, $t = 1 \cdots T_{max}$

(a) **Partition proposal:** Propose partition t via Coupled Partition Subroutine

2 **Particle weighting:** Compute $w_p^k = w_p^{k-1} * \frac{p(\mathbf{y}|\mathbf{X}_p)}{\prod_{t=1}^{T_p} b_{p,t}}$

3 **Weight normalization:** Normalize w_p^k to sum to one over p

Coupled Partition Subroutine for Target t

1 For each particle $p = 1, ..., N_{part}$,

(a) **Particle partition proposals:** For each proposal $r = 1, ..., R$

i Sample $\mathbf{X}_{p,t}^*(r) \sim p(\mathbf{x}|\mathbf{X}_{p,t}^{k-1})$

ii Compute $\omega_r = p(\mathbf{y}|\mathbf{X}_{p,t}^*(r))$

(b) **Proposal weight normalization:** Normalize ω to sum to one.

(c) **Index selection:** Sample an index j from the distribution defined by ω

(d) **Partition selection:** Set $\mathbf{X}_{p,t} = \mathbf{X}_{p,t}^*(j)$

(e) **Bias balancing:** Retain bias of sample, $b_{p,t} = \omega_j$

from all other partitions are treated as independent and proposed using the IP method. When targets are not well-separated, the CP method is used.

To provide a criterion for partition separation, we threshold based on distance in sensor space between the estimated state of the i^{th} partition and the j^{th} partition. Denote by $\hat{\mathbf{x}}_i$ the estimated x and y positions of the i^{th} partition (5.42). Notice only the spatial states are used (i.e., velocities are neglected), as these are the states that measure distance in sensor space. We have computed the distance between two partitions using a Euclidian metric between the estimated centers, and the Mahalanobis metric (5.30), where $\hat{\Sigma}_j$ is the covariance associated with the estimate of the j^{th} partition (5.43).

$$r^2 = (\hat{x}_i - \hat{x}_j)' \hat{\Sigma}_j^{-1} (\hat{x}_i - \hat{x}_j). \tag{4.30}$$

We have studied the use of a nearest neighbor criterion, where partitions are considered coupled if any sample from partition i is closer to the center of partition j then any sample from partition j. In practice, we find that the Euclidian distance is less computationally burdensome and provides similar performance.

Table 4.5. Adaptive Proposal Method (Table V from [149] which is ©2005 IEEE - used with permission)

1 For each partition $t = 1 : T_{max}$

(a) **Distance computation:** $d(t) = \min_{j \neq t} ||\hat{\mathbf{x}}_t - \hat{\mathbf{x}}_j||$

(b) **Partition proposal:** if $d(t) > \tau$

Propose partition t using IP method

else

Propose partition t using CP method

2 **Particle weighting:** For each particle $p = 1, ..., N_{part}$
$w_p^k = w_p^{k-1} * \frac{p(\mathbf{y}|\mathbf{X}_p)}{\prod_{t=1}^{T_p} b_{p,t}}$

3 **Weight normalization:** Normalize w_p^k to sum to one.

3.5 Multi-target Particle Proposal Via Joint Sampling

In the IP, CP and AP methods as described above, samples are drawn independently for each target. Recall that this approach is motivated by the approximate factorization of the JMPD for well-separated targets. However, this approximate factorization does not hold for closely-spaced targets. As a result, if multiple targets are in close proximity drawing samples independently for each target leads to many particles being proposed in undesirable parts of the multi-target state space. Instead, particles for closely spaced targets should be drawn jointly, conditional on the previous target states and the current measurement.

Therefore a joint sampling refinement to the method promises to improve performance. In this method, those partitions that are deemed to be coupled are clustered according to the method of section 3.3. This results in "partitions" that contain multiple targets – some with 2 targets, some with 3 targets, etc. Then instead of proposing each target individually, the clustered pairs (triplets, etc.) of targets are proposed all at once. This method is summarized in Table 5.6. Note that the idea of a partition containing multiple targets is also present in the work of Orton [189], although adaptively deciding partition boundaries and partition clustering is new to this work.

To understand how this joint sampling method works, Fig. 5.1 illustrates a scenario with two targets moving along the real line in opposite directions with equal speed. The importance density should propose as many particles as possible with the correct arrangement of targets, in this case one target in each of the cells.

Table 4.6. Modified Adaptive Proposal Method

1 **Partition clustering:** Construct non-overlapping target clusters $C_1, \ldots, C_s$, $s \leq r$ such that $\cup_{l=1}^{s} C_l = \{1, \ldots, T\}$, and if $i \in C_l$ then $\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\| < \Gamma \Rightarrow j \in C_l$ where Γ is a threshold.

2 For each cluster, $l = 1 \cdots s$

(a) **Cluster proposal:** if cluster C_l has one entry,

Propose group c using IP method

else

Propose group c using CP method

3 **Particle weighting:** Compute the weights $w_p^k = w_p^{k-1} \frac{p(\mathbf{y}|\mathbf{X}_p)}{\prod_{l=1}^{s} b_{p,l}}$

4 **Weight normalization:** Normalize w_p^k to sum to one.

Let $\mathbf{x}_i^k = (\rho_i^k, v_k^i)$ denote the state of target i with ρ_i^k the position and v_i^k the velocity. The weight given to a particular arrangement of targets can be measured by the probability

$$I(V_1, V_2) = \int_{V_1} d\rho_1^k \int_{-\infty}^{\infty} dv_1^k \int_{V_2} d\rho_2^k \int_{-\infty}^{\infty} dv_2^k q(\mathbf{X}^k | \mathbf{X}^{k-1}, \mathbf{y}^k). \quad (4.31)$$

where the V_i are cell volumes. Ideally we would have $I([0,1],[-1,0]) = 1$, i.e., all samples for each target would be placed in the cell occupied by that target. If both targets are detected, the joint sampling importance density has $I([0,1],[-1,0]) = 0.977$ while the independent sampling importance density has $I([0,1],[-1,0]) = 0.754$. Thus, the joint sampling density places a higher proportion of particles in the correct cells compared to the independent sampling density. The benefits of joint sampling become even more apparent if only one target is detected. In this case $I([0,1],[-1,0]) = 0.692$ for joint sampling and $I([0,1],[-1,0]) = 0.003$ for independent sampling.

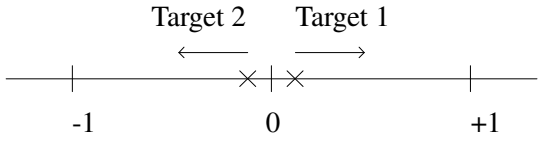

Figure 4.1. Crossing target scenario for demonstration of joint target proposals

Joint sampling of particles is more computationally expensive than independent sampling. In particular, the computational expense of jointly drawing samples for a group of targets increases exponentially with the number

of targets. Fortunately, this intractable increase in computational expense can be avoided by exploiting the approximate factorization of the JMPD for well-separated targets. This approximate factorization is illustrated in Fig 5.2 which shows the probability of correct placement of the particles plotted against target separation for joint and independent sampling. Results are given for the cases where both targets are detected and only one target is detected. It can be seen that for a sufficiently large separation the independent sampling density will almost certainly place particles in the correct location. In such cases the computational expense of joint sampling is unnecessary and should be avoided.

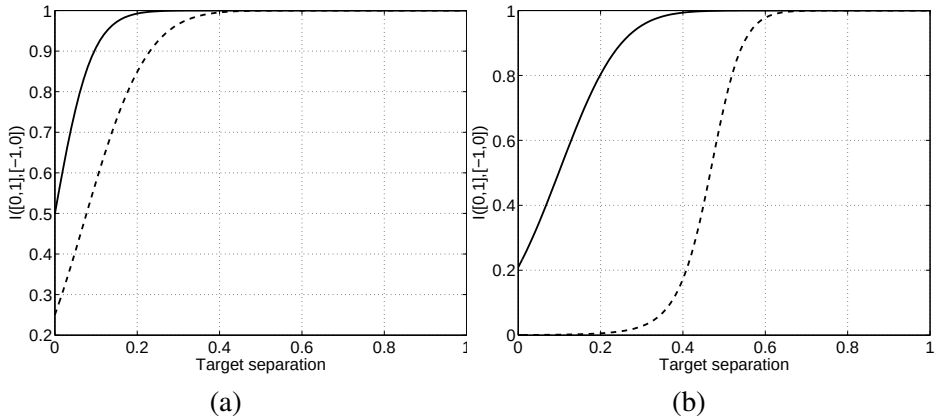

Figure 4.2. Probability of correct placement of particles for joint sampling (solid) and independent sampling (dashed) plotted against target separation when (a) both targets are detected and (b) only one target is detected.

In certain circumstances, the optimal importance density can be even more efficiently approximated than the sample based approach discussed here. In particular, if target dynamics are linear/Gaussian and measurements are made on a grid, the optimal proposal involves sampling from truncated normals [178]. In this case, a similar AP approach is used wherein partitions are first separated into groups that are uncoupled and then each group is treated by sampling from multidimensional truncated normals. This results in optimal sampling from each of the clusters giving even more efficient particle utilization.

3.6 Partition Ordering

As we have seen, the permutation symmetry associated with the JMPD discussed in Section 2 is inherited by the particle filter representation of the

JMPD. Each particle contains many partitions (as many as the number of targets it hypothesizes to exist) and the permutation symmetry of JMPD manifests through the fact that the relative ordering of targets may change from particle to particle.

The fact that partitions are in different orders from particle to particle is of no consequence when the object of interest is an estimate of the joint multi-target density. Each particle contributes the correct amount of mass in the correct location to the multi-target density irrespective of the ordering of its partitions.

However, the IP scheme gains efficiency by permitting hybridization across particles and so requires that particles be identically ordered. Furthermore, estimating the multi-target states from the particle filter representation of JMPD must also be done in a way that is invariant to permutations of the particles. Therefore, when estimating target states, we permute the particles so that each particle has the targets in the same order. We use a K-means [107] algorithm to cluster the partitions of each particle, where the optimization is done across permutations of the particles. In practice, this engenders a very light computational burden. The are two reasons for this. First, partitions corresponding to widely separated targets are not coupled and so remain ordered. Second, since ordering is applied at each time step, so that those coupled partitions are always nearly ordered and so one iteration of the K-means algorithm is usually enough to find an optimal permutation.

As shown in Table 5.7, the K-means algorithm consists of the following. Under the permutation π_p, the particle

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \cdots, \mathbf{x}_{p,T_p}], \tag{4.32}$$

is reordered as

$$\mathbf{X}_p = [\mathbf{x}_{p,\pi_p(1)}, \mathbf{x}_{p,\pi_p(2)}, \cdots, \mathbf{x}_{p,\pi_p(T_p)}]. \tag{4.33}$$

The mean of the t^{th} partition under the permutation π is

$$\bar{\mathbf{X}}_t(\pi) = \sum_{p=1}^{N_{parts}} w_p \mathbf{X}_{p,\pi_p(t)}, \tag{4.34}$$

where it is understood that the summation is taken over only those particles that have partition t. Further, define the χ^2 statistic

$$\chi^2(\pi) = \sum_{p=1}^{N_{parts}} \sum_{t=1}^{T_p} w_p (\mathbf{X}_{p,\pi_p(t)} - \bar{\mathbf{X}}_t(\pi_p))^2. \tag{4.35}$$

To reorder the particles, the goal is to find the set of permutations π that minimize χ^2, i.e.,

$$\hat{\pi} = \min_{\pi} \chi^2(\pi). \tag{4.36}$$

The K-means algorithm is a well known method of approximately solving problems of this type. An initial π is assumed and perturbations about that value are made to descend and find the best (local) π.

Table 4.7. K-means Algorithm Optimizing Over Partition Orderings (Table VI from [149] which is ©2005 IEEE - used with permission)

1 **Initialize ordering:** Initialize with π = current ordering of partitions

2 **Compute means:** Compute $\bar{\mathbf{X}}_t(\pi)$ for $t = 1 \cdots T_p$ using (5.34)

3 **Permute particles:** For each particle p, permute the particle (update π_p) to yield

$$\pi_p \leftarrow \arg\min_{\pi_p} \sum_{t=1}^{T_p} (\mathbf{X}_{p,\pi_p(t)} - \bar{\mathbf{X}}_t(\pi_p))^2$$

4 **Attempt termination:** If no particles have changed permutation, quit. Otherwise set $\pi = (\pi_1, \cdots, \pi_p, \cdots, \pi_{N_{part}})$ and go to 2

3.7 Estimation

Particle estimates of various interesting quantities can be computed. The central quantity of interest for sensor management is the expected Renyi Divergence. Using the particle filter representation for the JMPD and inserting that into (5.16) yields

$$\langle D_\alpha \rangle_m = \frac{1}{\alpha - 1} \sum_{y=0}^{1} p(y) \ln \frac{1}{p(y)^\alpha} \sum_{p=1}^{N_{part}} w_p p(y|\mathbf{X}_p)^\alpha \tag{4.37}$$

To compute the probability of exactly n targets in the surveillance volume, first define the indicator variable $I_p(n)$ for $p = 1...N_{parts}$,

$$I_p(n) = \begin{cases} 1 & \text{if } T_p = n \\ 0 & \text{otherwise} \end{cases} \tag{4.38}$$

Then the probability of n targets in the surveillance volume, $p(n|\mathbf{Y})$, is given by

$$p(n|\mathbf{Y}) = \sum_{p=1}^{N_{part}} I_p(n) w_p \tag{4.39}$$

The estimate of the probability that there are n targets in the surveillance volume is the sum of the weights of the particles that have n partitions. Note that the particle weights, w_p, are normalized to sum to unity for all equations given in this section.

To compute the estimated target state and covariance of target i, we first define a second indicator variable $\tilde{I}_p(i)$ which indicates if particle p has a partition corresponding to target i. This is necessary as each particle is a sample drawn from the JMPD and hence may have a different number of partitions (targets):

$$\tilde{I}_p(i) = \begin{cases} 1 & \text{if target } i \text{ exists in particle } p \\ 0 & \text{otherwise} \end{cases} \tag{4.40}$$

Note that the sorting procedure of Section 3.3 has already identified an ordering of particles to allow $\tilde{I}_p(i)$ to be determined. Furthermore, we define the normalized weights to be

$$\hat{w}_p = \frac{w_p \tilde{I}_p(i)}{\sum_{l=1}^{N_{part}} \tilde{I}_l(i) w_l} \tag{4.41}$$

So $\hat{w}_p$ is the relative weight of particle p, with respect to all particles tracking target i. Then the estimate of the state of target i is given by

$$\hat{\mathbf{X}}(i) = E[\mathbf{X}(i)] = \sum_{p=1}^{N_{part}} \hat{w}_p \mathbf{X}_{p,i} \tag{4.42}$$

which is simply the weighted summation of the position estimates from those particles that are tracking target i. The covariance estimate is

$$\hat{\mathbf{\Lambda}}(i) = \sum_{p=1}^{N_{part}} \hat{w}_p (\mathbf{X}_{p,i} - \hat{\mathbf{X}(i)})(\mathbf{X}_{p,i} - \hat{\mathbf{X}(i)})' \tag{4.43}$$

3.8 Resampling

In the traditional method of resampling, after each measurement update, N_{part} particles are selected with replacement from $\mathbf{X}_p$ based upon the particle weights w_p. The result is a set of N_{part} particles that have uniform weight which approximate the multi-target density $p(\mathbf{X}|\mathbf{Y})$. The particular resampling that was used in this work is systematic resampling [10]. This resampling strategy is easily implemented, runs in order N_{parts}, is unbiased, and minimizes the Monte Carlo variance. Many other resampling schemes and modifications are presented in the literature [78]. Of these methods, we have found that adaptively choosing at which time steps to resample [163] based on the number of effective particles leads to improved performance while reducing compute time. All results presented herein use the method of [163] to determine which times to resample and use systematic resampling [10] to perform resampling. We have also found that Markov Chain Monte Carlo (MCMC) moves using a Metropolis-Hastings scheme [78] leads to slightly improved performance in our application.

4. Multitarget Tracking Experiments

In this section we present performance results for the particle filtering techniques described above, focusing on tracking applications. Chapter 6 presents sensor management results obtained using these techniques. We begin with a set of experiments that shows the impact of the different proposal methods on tracking accuracy and numerical requirements. We then present a few results showing the effect of thresholding on tracker performance and the performance gain that can be achieved from using prethresholded measurements in our association-free approach.

We illustrate the performance of our multi-target tracking scheme by considering the following scenario. Targets move in a $5000m \times 5000m$ surveillance area. Targets are modeled using the four-dimensional state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$. The target motion in the simulation is taken from a set of recorded data based on GPS vehicle measurements collected as part of a battle training exercise at the Army's National Training Center. This battle simulation provides a large number of real vehicles following prescribed trajectories over natural terrain. Based on an empirical fit to the data, we found that a nearly constant velocity model was adequate to model the behavior of the vehicles for these simulation studies and is therefore used in all experimental results presented here. (Estimation performance can be improved with a moderate increase in computational load using a multiple model particle filter with

modes corresponding to nearly constant velocity, rapid acceleration, and stationarity [145].)

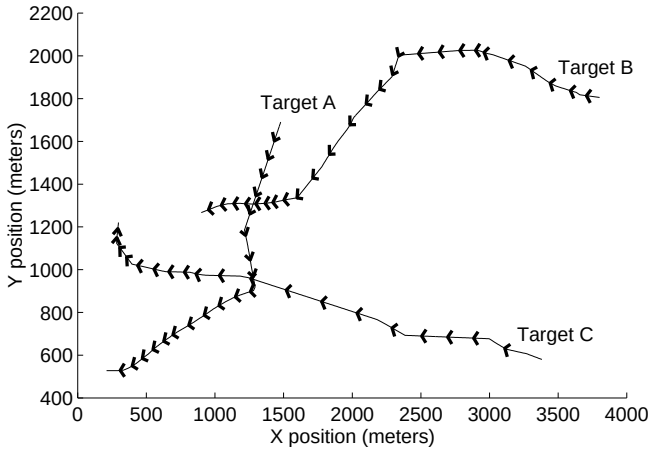

Figure 4.3. A schematic showing the motion of three of the ten targets in the simulation scenario. The target paths are indicated by the lines, and direction of travel by the arrows. There are two instances where the target paths cross (i.e., are at the same position at the same time). (Fig. 1 from [149] which is ©2005 IEEE - used with permission)

We use the simple scalar sensor model described in Section 2.5. The sensor scans a fixed rectangular region of 50×50 pixels, where each pixel represents a $100m \times 100m$ area on the ground plane. The sensor returns Rayleigh-distributed measurements in each pixel, depending on the number of targets that occupy the pixel. Unthresholded measurements return energy according to (5.22) while thresholded measurements behave according to (5.24).

4.1 Adaptive Proposal Results

In Figure 5.4, we compare the performance of the Independent Partitions (Table 5.3), Coupled Partitions (Table 5.4), and Adaptive Partitions (Table 5.5) with the traditional method of sampling from the kinematic prior (Table 5.2), in terms of RMS tracking error. In this example we use 3 targets with motion taken from recorded ground vehicle trajectories. The targets remain close in sensor space for about 50% of the time. Thresholded measurements with $P_d = 0.5$ are used and the SNR parameter λ is varied from 1 to 21. Each proposal scheme uses 100 particles to represent the JMPD. The filter is initialized with truth. Each point on the curve is an average of 100 runs. Table 5.8 shows the rough computational burden of the 4 methods, obtained using the "FLOPS" command of MatLab.

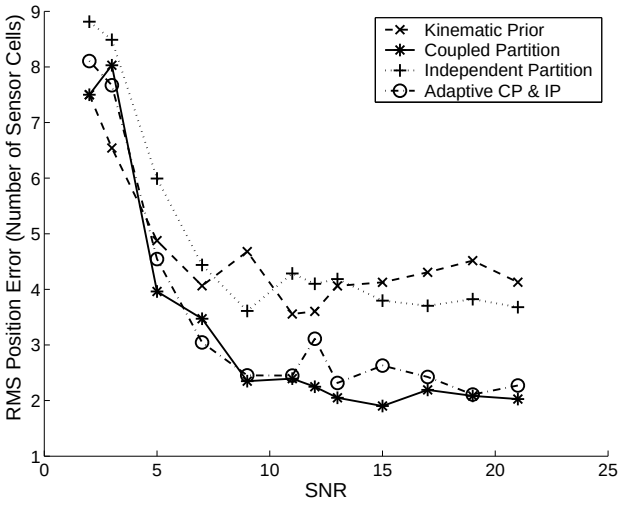

Figure 4.4. The Performance of the Coupled Partitions (CP), Independent Partitions (IP), and Adaptive Partitions (AP) schemes in comparison to simply using the kinematic prior. Performance is measured in terms of RMS position error. The units in the figure are sensor cells. For this simulation, we have extracted 3 targets from our large database of real recorded target trajectories. The targets were chosen so that they spent approximately one-half of the simulation in close proximity. The IP algorithm used alone is inappropriate during target crossings and so performs poorly here. The CP algorithm is always appropriate, but computationally demanding. The AP algorithm adaptively switches between IP and CP resulting in good performance at reduced computation (Figure 2 from [149] which is ©2005 IEEE - used with permission).

At low SNR all of the methods provide similar tracking performance. As SNR increases, the performance of all for methods improves (lower rms error), but CP and AP provide consistently better estimation performance. The performance of KP is relatively low since it does not use the measurements in the proposal. IP adversely effected due to its failure to account for the correlation effects of nearby targets. The CP method makes no assumption about the independence of the targets and therefore performs very well, but at significantly higher computational cost. Most importantly, the adaptive method, which uses IP on partitions that are independent and CP otherwise, performs nearly as well as the CP method. AP achieves approximately a 50% reduction in computational burden (measured by floating point operations) as compared to the CP method alone.

To gain further insight into the relative performance of IP, CP and KP methods we simulate linear Gaussian target motion for five well-separated targets, and examine the estimation performance as a function of the number of particles used in the different filters, shown in Figure 5.5. In this model problem the Kalman filter is optimal and can therefore be used as a basis for compari-

Table 4.8. FLOPS for KP, CP, IP, and AP Methods (Table VII from [149] which is ©2005 IEEE - used with permission)

Method	Flops
Coupled Partition	1.25e+8
Independent Partition	6.74e+6
Adaptive Partition	5.48e+7
Kinematic Prior	6.32e+6

son. We use the nearly constant velocity motion model for both the simulation of target motion and the filter. In each case, the filter is initialized with truth and run until it achieves steady state where the mean track error is measured. The interesting result is that IP and CP achieve the (optimum) Kalman filter performance bound with 200 and 1000 particles respectively while KP still has not quite met the bound with 20,000 particles. Thus even for this idealized problem for which the Kalman filter is optimal, KP performs poorly due to the high dimensionality of the multi-target state space.

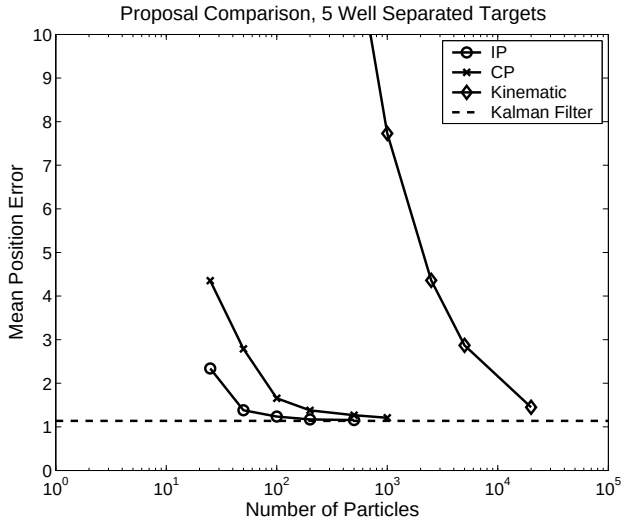

Figure 4.5. The performance of the CP and IP Proposal Schemes, in comparison to sampling from the Kinematic Prior. For the purposes of this example, we consider well separated targets with linear motion and linear state-to-measurement coupling. Therefore for the purposes of this simple example, the Kalman filter is optimal and is shown as a performance bound (Fig. 3 from [149] which is ©2005 IEEE - used with permission).

4.2 Partition Swapping

Figures 5.6 and 5.7 illustrate how partition swapping occurs and how it is reduced by using the K-means partition sorting algorithm. Partition swapping leads to poor performance when it occurs while using the Independent Partition proposal algorithm. Figure 5.6 shows several snapshots of the particle distribution from a crossing target example without partition sorting (the number of particles has been reduced for illustration purposes here). Initially, the targets are well separated and identically ordered (e.g. Time=44) and the IP method is used for particle proposal. When the targets cross (Time=60), partition swapping occurs and the CP method must be used. The targets remain in the same detection cell for several time steps. Without partition sorting using the k-means algorithm of Section 3.3, this swapping persists even after the targets separated and the CP method must be used even at Time=84. This results in an inefficient algorithm, as the CP method is more computationally demanding.

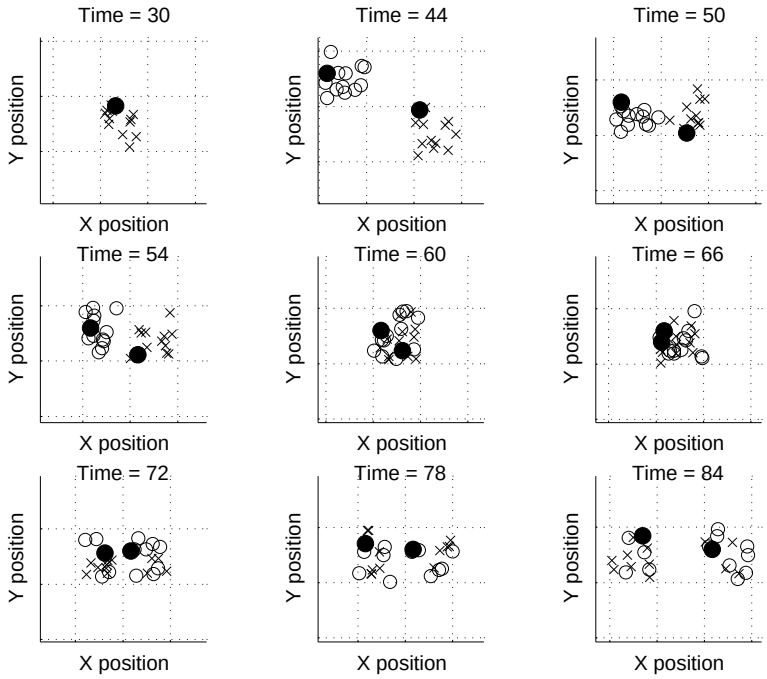

Figure 4.6. This figure illustrates the phenomenon of partition swapping that occurs in direct particle filter implementation of the SIR filter for JMPD. True target locations are indicated by a solid circle. The two partitions for each particle, plotted with $\times$ and $\circ$, are well separated at time 44. From time 60 to 66, they occupy the same detection cell. At time 84, some partition swapping has occurred, indicated by the fact that there are mixtures of $\times$ and $\circ$ corresponding to each target location (Fig. 10 from [149] which is ©2005 IEEE - used with permission).

Figure 5.7 is analogous to Figure 5.6, but this time we utilize the partition sorting algorithm outlined in Section 3.3 at each time step. While the CP method must still be used when the targets are occupying the same detection cell, when they separate (Time=72) the IP method may be used again. The partition sorting allows for the more computationally efficient IP method to be used for proposal by reordering the particles appropriately.

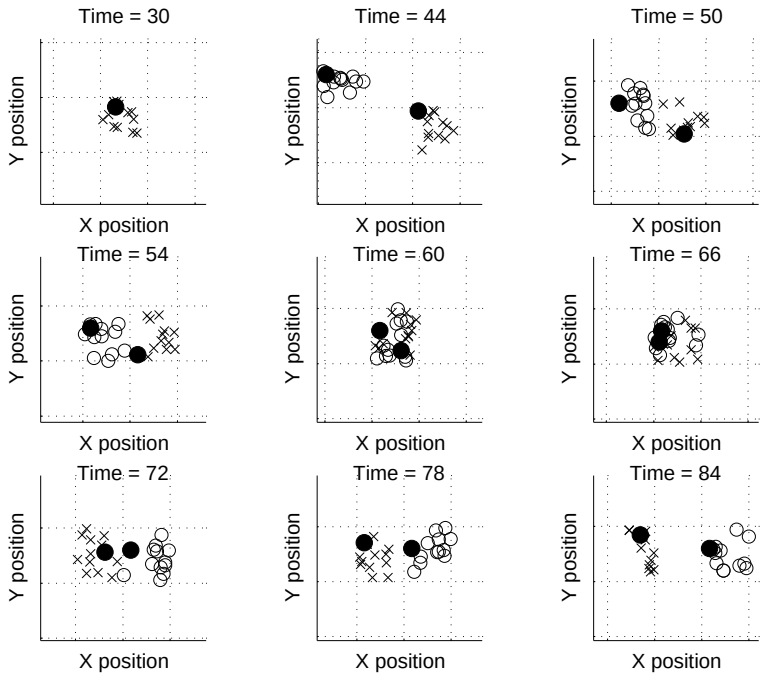

Figure 4.7. An example of the behavior of the particle filter based multi-target tracker during target crossing when partition sorting is employed. The nine time sequential images focus on one of the ten targets that the filter is tracking. The ground truth location of the target (projected into the XY plane) is denoted by a solid circle. The partitions associated with the two targets are denoted by $\times$ and $\circ$. The sensor cells are given by the gridlines. As discussed earlier, the sensor measures on a grid and receives energy from the target density if the cell is occupied or the false alarm density if the cell is empty. Initially (before time 50), this target is well separated from all of the others. At these points the IP algorithm is being used for proposal. During times 50 to 66, a second target is crossing (coming within sensor resolution) of the target of interest. Near time 72, the target complete their crossing and again move apart (Fig. 11 from [149] which is).

4.3 The Value of Not Thresholding

One of the strengths of association-free methods is its ability to use non-thresholded measurements without modification. Intuitively, using non-thresholded measurements should improve performance targets that are missed due to thresholding might be detected without thresholding. Here we quantify the relative performance of the tracker for a set of scenarios in which the target motion and the underlying measurement probability density functions are identical. The only difference is that in one case we threshold to obtain binary measurements with distribution (5.24) while in the other case we use unthresholded measurements and the envelope detected amplitude (5.22) is input into the tracker.

If one is using thresholded measurements, then the first step is to optimize the threshold. For Rayleigh measurements with a given SNR, setting the threshold is equivalent to setting the false alarm probability (see (5.23)). To assess performance, we again use the 3-crossing target data shown of Figure 5.3. The filters are initialized with ground truth and we assess the number of targets in track at the conclusion of the vignette. Figure 5.8 shows a contour plot of the result as a function of P_d and SNR that increases with SNR and is broadly peaked around $P_d = 0.4$.

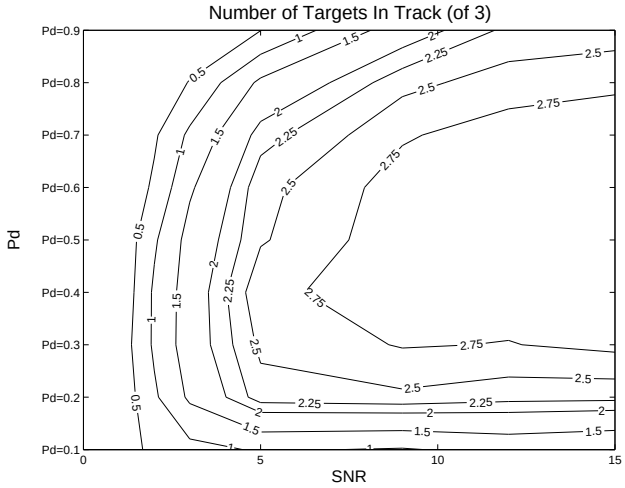

Figure 4.8. A contour plot showing the number of targets successfully tracked in this three target experiment versus P_d and SNR when using thresholded measurements (Fig. 4 from [149] which is ©2005 IEEE - used with permission).

Figure 5.9 shows the performance of the algorithm using optimized thresholded measurements at $P_d = 0.4$ and the non-thresholded measurement algorithm. We see that non-thresholded measurements provide similar tracking

performance at an SNR of 1 as the thresholded measurements provide at an SNR of 5, for a gain of about 7dB from not thresholding the measurements.

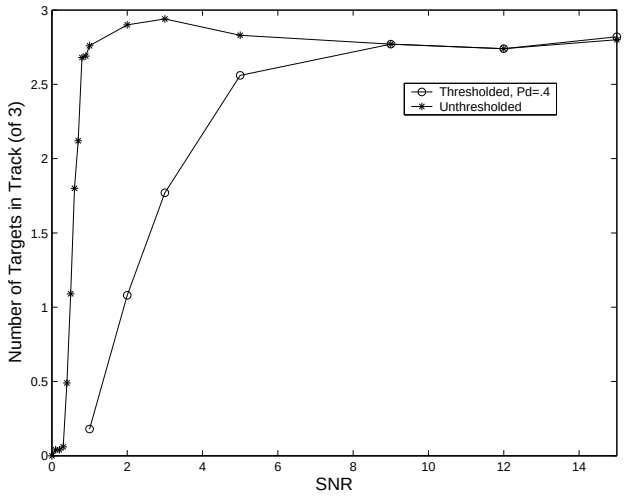

Figure 4.9. A plot of the number of targets successfully tracked in the three target experiment for thresholded measurements and unthresholded measurements as a function of SNR (Fig. 5 from [149] which is ©2005 IEEE - used with permission).

4.4 Unknown Number of Targets

The ability of the JMPD joint particle filtering algorithm to determine the number of targets is illustrated in Figures 5.10 for the same data as in Figure 5.3. There are three targets in this simulation. We initialized the filter uniformly in target number space, allocating one-sixth the probability to $0, 1, \cdots, 5$ targets. Over time, the filter is able to accurately estimate the number of targets in the surveillance region. As the SNR improves, the time until correctly determining the number of targets decreases.

5. Conclusions

There is a need to develop sensor management techniques that are applicable across a broad range of Intelligence, Surveillance and Reconnaissance assets. A critical component to sensor management is the accurate approximation of the joint posterior density of the targets, i.e., the belief state. The belief state is captured by the Joint Multitarget Probability Density and it quantifies the effect of uncertainty within the scene and permits the use of information-based methods to quantify this uncertainty and select most informative sensing actions.

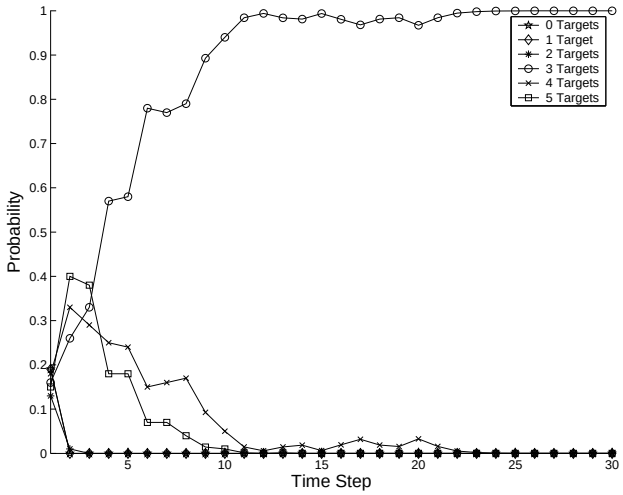

Figure 4.10. The estimate of number of targets in the surveillance region versus time with SNR=4. The filter is initialized with probability uniform for $0, 1, \cdots, 5$ targets. Measurements taken over time allow the filter to properly estimate the number of targets in the surveillance area (Fig. 6 from [149] which is ©2005 IEEE - used with permission).

The need to treat nonlinear target motion and target birth and death motivates the use of nonlinear techniques such as particle filtering. The inherent permutation invariance of the JMPD must be treated properly in developing proposal methods for the particle filter, leading to the IP, CP, AP and OID methods described in the chapter. These can be implemented with a moderate computational cost and can provide order-of-magnitude reductions in the amount of sensor resource required to achieve a given level of tracker performance. A side benefit is that these nonlinear methods lead naturally to association free-trackers that can make use of unthresholded measurements, providing further improvement and effectively increasing the target SCNR by 3-5 dB.

The future directions for this work include expanding the information based sensor scheduling to long term planning discussed elsewhere in this volume (Chapters 3, 7). Another area requiring further work is in optimizing proposals during the track initiation phase as the work reported here focuses on optimizing proposals for firm tracks. The studies presented here have been performed in the context of air- and space-based sensors tracking ground moving targets. These results are readily extendable to other domains such as missile defense or tracking air targets.

In the next Chapter the joint particle filtering methods introduced here will be applied to several sensor management scenarios.