

# Fixed Lag Particle Filtering for Target Tracking

Pete Bunch

August 2011

A report on the first year's progress



UNIVERSITY OF  
CAMBRIDGE

---

---

# *Contents*

---

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Literature Review</b>	<b>7</b>
2.1	Particle filters . . . . .	7
2.1.1	Sequential importance sampling . . . . .	8
2.1.2	Markov chain Monte Carlo . . . . .	9
2.1.3	Coping with dimensionality . . . . .	11
2.1.4	Marginalised particle filters . . . . .	11
2.2	Tracking . . . . .	12
2.3	Probabilistic data association . . . . .	12
2.4	Multi-hypothesis tracking . . . . .	14
2.5	MCMC data association . . . . .	14
2.6	Probability hypothesis density methods . . . . .	14
<b>3</b>	<b>Mathematical foundations</b>	<b>15</b>
3.1	Bayes of our lives - what is filtering? . . . . .	15
3.2	Keep Kalman carry on - the Kalman filter and its extensions . . . . .	16
3.2.1	The basic filter . . . . .	16
3.2.2	The extended filter . . . . .	17
3.2.3	The Kalman smoother . . . . .	17
3.3	Tough as old bootstraps - a review of particle filter . . . . .	18
3.4	Particular details - mathematical basics of the particle filter . . . . .	18
3.4.1	Auxiliary sampling . . . . .	19
3.4.2	Degeneracy and resampling . . . . .	19
3.4.3	Importance distributions . . . . .	20
3.5	MCMC in da house - series vs. parallel . . . . .	20

<b>4</b>	<b>Models for Tracking</b>	<b>21</b>
4.1	Why do we track? . . . . .	21
4.2	Models for tracking . . . . .	21
4.2.1	Data association . . . . .	22
4.3	Algorithms for tracking . . . . .	22
<b>5</b>	<b>Fixed Lag Estimation for Tracking</b>	<b>23</b>
5.1	Why do now what we can do later? . . . . .	23
5.2	A mathematical framework for fixed lag estimation . . . . .	24
5.2.1	Artificial conditionals . . . . .	25
5.2.2	History Proposals . . . . .	25
5.3	Applying the fixed lag method to the tracking model . . . . .	25
5.3.1	Importance distributions . . . . .	25
<b>6</b>	<b>Fixed Lag Particle Filters</b>	<b>29</b>
6.1	Sequential Importance Sampling and Resampling . . . . .	29
6.1.1	Coping with dimensionality . . . . .	29
6.2	Markov Chain Monte Carlo . . . . .	29
6.3	Marginalised Particle Filters . . . . .	29
<b>7</b>	<b>Plan Of Future Research</b>	<b>31</b>

## *Introduction*

---

Everyone loves to track things. Yay!



## *Literature Review*

---

Two spheres of literature will be of interest here: the developments of models and algorithms for tracking, and the studies of the numerical inference schemes which underly our method. We begin with the latter.

### **2.1 Particle filters**

Filtering is the operation of inferring the state of an evolving latent random process given a set of imperfect observations up until the current time. In a Bayesian estimation scheme, we would like to calculate a posterior probability distribution for such a state at each point in time. The difficulty with such a procedure is that the complexity of the state distribution will tend to compound over time. In the particular case where both the state transition and observation processes are linear transformations with Gaussian noise, it is possible to derive an analytic expression for the state distribution with constant complexity, the fabulous Kalman filter (KF) of (Kalman 1960). However, as many problems are often non-linear or non-Gaussian, and as no other such analytic and generally applicable cases have been discovered, (Daum 2005), we must resort to numerical techniques.

The particle filter (PF) approximates a probability distribution using a set of discrete samples or “particles” drawn from it. This will be no trivial task - the distribution cannot be sampled directly, just as it cannot be expressed analytically. Instead, at each time step, the cloud of particles representing the previous distribution is propagated forwards to approximate the next, taking into account the latest observation. The most common PF algorithm in use is known as Sequential Importance Sampling with Resampling (SISR) (or some permutation of this initialism) - in fact the terms SISR and PF are often used interchangeably.

### 2.1.1 Sequential importance sampling

#### *Importance sampling*

The first modern implementation of the SIS algorithm by (Gordon, Salmond & Smith 1993). At the heart of the algorithm is an importance sampling step. Each particle from the old distribution is propagated forward by sampling from a proposal or ‘importance’ distribution. The particles are then assigned an “importance weight” to account for the discrepancy between the proposal distribution and the desired posterior. A significant strength of the algorithm is that we need only be able to calculate the posterior probability up to a normalising constant. This may be easily accounted for by normalising the weights of the discrete sample set.

#### *Resampling*

Over time, the variance in these weights is liable to grow, as unlikely particles deviate completely from the correct state and are assigned a negligible weight. The eventual result is a degenerate sample with all the weight on one or a few particles and most contributing nothing to the approximation. The solution to this degeneracy problem is to introduce a resampling step before the importance sampling, in which particles are sampled from the particle approximation. Thus, heavily weighted particles are multiplied while low weight particles are discarded. Many algorithms for resampling exist, a summary of which is given by (Doucet & Johansen 2009). This resampling step introduces a degree of Monte Carlo error, and so (Liu & Chen 1995) introduce a measure of particle diversity to assess the degree of degeneracy. This measure, the effective sample size (ESS), is an estimate of the equivalent number of equal-weighted particles in the approximation, and is based on a calculation of the sample weight variance. Resampling is only used if the ESS falls below a chosen threshold.

#### *Importance distributions*

The filter proposed by (Gordon et al. 1993), known as the bootstrap filter, used the state transition density as the importance distribution, as did many other early implementations, (Blake, Isard et al. 1998, Kitagawa 1996). However, this can result in poor performance if the observation process is informative. In other words, if we know that the observation is always very close to the state, then there is no point in proposing particles far away which will have a very low weight. It was suggested in, amongst others, (Liu & Chen 1995) that a better proposal could be constructed by considering the position of the new observation. In particular, the “optimal” (in the sense that it minimises the weight variance) proposal is the conditional distribution of the new state given the state history and the new observation (see (Doucet, Godsill & Andrieu 2000) for proof). This is rarely available analytically or samplable, so Gaussian approximations may be used (Doucet et al. 2000).

#### *Auxiliary particle filtering*

In the description above, resampling was introduced as an additional procedure to rejuvenate a degenerate particle distribution. However, it is possible to view it as a more integral part of the sampling procedure: At each step we construct a set of particles by proposing both a new state and a history for each. These histories are selected from the previous particle distribution. The



resampling fulfils the role of a “history proposal”. On a step with resampling we propose histories from the weighted particle distribution from the previous frame. On a step without resampling, we propose histories from the unweighted particle distribution, which can of course be done by simply keeping the same set of particles! In this new paradigm the obvious generalisation is to allow history proposals from the previous set of particles with arbitrary weights. A correction is made in the importance weight calculation to account for the proposal weights. This is the innovation of the auxiliary particle filter introduced by (Pitt & Shephard 1999). The proposal weights may be assigned so as to favour particles which give rise to more likely states.

### *Resample-move*

The objective when selecting the “optimal” proposal distribution or in choosing auxiliary proposal weights is to minimise the variance of the importance weights, and thus reduce the need for resampling. Another important particle filtering technique is the resample-move method of (Gills & Berzuini 2001), which aims instead to replenish the particle diversity after resampling. Rather than beginning the next IS step with multiple copies of the same particle, the current and previous states of each are adapted using one or more Metropolis-Hastings moves. Although this method improves the representation of the state distribution, it requires significant additional computation for each particle.

### 2.1.2 *Markov chain Monte Carlo*

#### *Metropolis-Hastings*

Markov chain Monte Carlo (MCMC) algorithms, like SISR, make use of a set of samples to approximate a probability distribution. Unlike SISR, MCMC is not specifically designed to address problems of sequential inference, although, as we shall see, they may be applied to these problems as an MCMC particle filter.

The basic MCMC method was proposed by (Metropolis, Rosenbluth, Rosenbluth, Teller, Teller et al. 1953) and extended by (Hastings 1970), after which the Metropolis-Hastings (MH) algorithm is named. Rather than producing independent samples from a distribution, MH uses a Markov chain. A reversible, ergodic Markov chain has a stationary distribution. By careful design of the transition properties of the chain, we can make the stationary distribution equal to some desired target distribution. For our inference problems, this target will be the state posterior.

A MH move is made by sampling a value of the state from a proposal distribution. With some probability, the new value is accepted, otherwise the previous value is maintained. The acceptance probability is given by:

$$\alpha = \min \left( 1, \frac{P(X_{\text{new}})q(X_{\text{old}}|X_{\text{new}})}{P(X_{\text{old}})q(X_{\text{new}}|X_{\text{old}})} \right) \quad (2.1)$$

where  $q(\cdot|\cdot)$  is the proposal distribution, and  $P(\cdot)$  the target.

The chain must be initialised at some value, and this is unlikely to be a sample from the target (if we could sample it, we would not need MCMC). Thus, there will be a period over which the sampler converges to the target distribution. This period is known as the “burn in” and the samples should be excluded from the particle approximation.

The choice of proposal does not affect the target distribution, although it will affect the rate at which the algorithm converges. If the proposal distribution has a low variance then successive samples will be close together, and it may take many moves to explore the whole distribution. If the variance is large, then the acceptance probability may be very low and again convergence may be slow.

A complete and excellent discussion of MCMC design considerations, convergence properties, etc. can be found in (Gilks, Richardson & Spiegelhalter 1996), and will not be repeated here.

### *Block sampling*

When MCMC algorithms are applied to multivariate distributions it is not necessary to change every variable in every move. If the state is divided into two blocks,  $X_i$  and  $X_{-i}$ , and a move is proposed only on  $X_i$ , then the acceptance probability is now given by:

$$\alpha = \min \left( 1, \frac{P(X_{i,\text{new}}|X_{-i})q(X_{i,\text{old}}|X_{i,\text{new}}, X_{-i})}{P(X_{i,\text{old}}|X_{-i})q(X_{i,\text{new}}|X_{i,\text{old}}, X_{-i})} \right) \quad (2.2)$$

A special case of this scheme occurs when  $q(X_{i,\text{new}}|X_{i,\text{old}}, X_{-i}) = P(X_{i,\text{new}}|X_{-i})$ . This is the Gibbs sampler devised by (Geman & Geman 1984). With this choice of proposal distribution, the acceptance probability cancels out to 1. Thus, all samples are accepted. The difficulty with this formulation is that the required conditional distribution may not be available, or may not be samplable.

MH moves which alter only a subset of the variables are commonly known as Metropolis-within-Gibbs (MwG) moves.

### *Reversible jump Markov chain Monte Carlo*

MCMC methods are not restricted to distributions of fixed dimension. The Reversible Jump MCMC (RJMCMC) method devised by (Green 1995) allows MH moves to jump between state spaces with different dimensionality. This is especially useful when trying to choose between models while simultaneously estimating parameters, or for estimating the order or number of components in a model, for example in a Gaussian mixture or autoregressive model.

RJMCMC moves may have very low acceptance probabilities. When the sampler jump from one state space to another, there may be no obvious function for generating likely values in the new state. For such cases, (Al-Awadhi, Hurn & Jennison 2004) propose using a system of compound moves, where a model change move is followed by a number of normal MH moves designed to seek out a likely set of state values. A single acceptance probability is then calculated for the set of moves.

### *Sequential Markov chain Monte Carlo*

The interest in MCMC for this project lies in their application to sequential inference problems. Such ‘MCMC particle filters’ were first proposed by (Khan, Balch & Dellaert 2005), although this first implementation suffered from computation issues, and (Golightly & Wilkinson 2006), and have recently been applied to a range of problems including target tracking, (Septier, Pang, Carmi & Godsill 2009), group tracking, e.g. (Carmi, Godsill & Septier 2009), tracking of a correlated stock prices, (Pang, Godsill, Li & Septier 2011) and music transcription, (Bunch & Godsill 2010).

MENTION POLSON'S PRACTICAL FILTER SOMEWHERE - try to understand it first.

The MCMC particle filter runs a Markov chain for each frame which targets the posterior state distribution at that time. The particles are thus unweighted and are generated in series rather than in parallel, with each particle depending on the last. The state space of the Markov chain consists of the entire history of the hidden variables as well as the latest values. Two types of MH move are thus required: History-moves involve proposing a new history from the particles of the previous particle distribution. Current-moves involve a proposal of the latest variables given the history and data. The two types of move are analogous to the resampling and importance sampling steps of the SISR algorithm respectively.

MCMC particle filters have a significant advantage over their SISR counterparts for complex, high dimensional problems. As noted in (Pang, Li & Godsill 2008), we can use MwG moves, changing only a few variables at once, to explore the posterior state distribution. In contrast, with an SISR particle filter, we must propose a new value for every variable at once. In some cases this enables the MCMC particle filter to give a better approximation of the state posterior, as demonstrated in (Pang et al. 2011).

### 2.1.3 *Coping with dimensionality*

It is a commonly noted problem that the efficacy of particle filters falls as the number of state dimensions increases. This is a problem for both SISR and MCMC filters, and indeed for any Monte Carlo approximation. There is a strong intuition for this phenomenon - it will clearly take more particles to adequately sample a square than it would a line, and more still for a cube. An analysis of this effect, including numerical studies, was conducted in (Daum & Huang 2003).

The severity of the “curse of dimensionality” will depend on the correlation between the state variables, and the ability to exploit such correlations by the design of effective proposal distributions. Methods to improve performance of MCMC algorithms include Hybrid Monte Carlo (HMC) (Duane, Kennedy, Pendleton & Roweth 1987), which introduces “momentum” variables to the state in order to build better proposals and the Riemann manifold methods of (Girolami & Calderhead 2011).

### 2.1.4 *Marginalised particle filters*

Monte Carlo methods are effective for tackling nonlinear, non-Gaussian problems when analytic methods cannot be found. However, they are computationally expensive and so should be avoided when an analytic solution does exist. In some problems, the state can be partitioned into two parts, one which behaves in a linear-Gaussian way conditional on the other. In such a case a particle filter can be used for inference of the nonlinear part, and a Kalman filter for the linear-Gaussian part. Such a strategy reduces the variance of the state estimates, in accordance with the Rao-Blackwell theorem. See, for example, (Casella & Robert 1996). The only change required is that likelihood calculation for the particle filter will now require the linear-Gaussian part of the state to be marginalised. Such schemes may be used with both SISR and MCMC particle filters and are known as *marginalised* or *Rao-Blackwellised* particle filters.

## 2.2 Tracking

In a target tracking situation we aim is to trace the trajectory of an object over time from a set of discrete observations. For the simplest case, with a single target under observation, which is detected in every frame, and with no false alarms, this is a simple state-space inference problem. It can be approached with a Kalman filter or a basic particle filter. The phenomenon which makes tracking problems more challenging is association ambiguity. In a given frame, we may not be guaranteed to detect a target, and we may pick up false alarms or clutter measurements. In general there is no way to establish with certainty which observation arose from which target. Finally, the number of targets in the scene may also be unknown. We are thus faced with a three-fold problem:

- Detect how many targets are present
- Estimate which observation arose from each target
- Estimate the state of each target

In this chapter, we review a number of strategies for tackling such a target tracking problem. Note that although we divide up the algorithms into sections, there is significant overlap between them, which we endeavour to highlight.

## 2.3 Probabilistic data association

The earliest works on target tracking with data association used a combination of Kalman filters and heuristics. (Sea 1971) suggests using only the observation with the minimum Mahalanobis distance from the Kalman prediction, i.e. a maximum likelihood estimate of the association. Such nearest neighbour algorithms can easily be caused to lose track by a single unfortunate false alarm.

A probabilistic approach was introduced by (Bar-Shalom & Tse 1975), in the form of the probabilistic data association filter (PDAF). Rather than select a single observation for each target in each frame, for the PDAF a posterior probability of each possible association is calculated, given the previous target states. A Kalman update is then evaluated for each of the possible associations and the results added together, weighted by the corresponding association posterior. Thus, the final update takes into account multiple (indeed, all) observations. Thus the PDAF cannot be so easily upset by single clutter measurement, although additional error is introduced in situations where a nearest neighbour method would have picked the correct association, because of the influence of false alarms which now contribute to the estimate.

PICTURE - demonstrating PDAF

When more than one target is present in the scene, a separate PDAF can be run for each one. However, this leads to errors. When two tracks pass close together, they may both have a high association probability with the same observation(s). The result is that the estimates converge onto the same path, and one of the tracks is lost.

PICTURE - demonstrating track merging

The shortcomings of the PDAF are addressed by extending the filter to consider the joint association posteriors. This is the joint PDAF (JPDAF) of (Fortmann, Bar-Shalom & Scheffe 1983), excellently reviewed in (Bar-Shalom, Daum & Huang 2009). Rather than calculating association probabilities for each target in isolation, the joint association hypotheses are assessed. We can now include as prior information the fact that no observation can be associated with

two targets. Thus the problem of tracks following the same observations is reduced. The price of this improved performance is the need to calculate an association probability for every joint hypothesis. The number of these is of combinatorial complexity in the number of targets and observations in a frame. Gating is used to address this issue: a target-observation association is only considered possible if the Mahalanobis distance between the two is below some threshold. The choice of threshold governs the probability of excluding the correct hypothesis, and thus trades off performance and complexity (Sea 1971).

Both the PDAF and the JPDAF suffer from a problem of bias and track coalescence when targets pass close to each other. This was observed and quantified by (Fitzgerald 1985). The measurements generated by the second target ‘pull’ the estimate of the first away from its correct position. When two targets are travelling along parallel paths this can lead to *track coalescence*. The estimates of both target states move together, midway between the two correct locations. Solutions to this problem often resort to reintroducing nearest-neighbour methods, such as the nearest-neighbour-JPDAF (Fitzgerald 1986) or the set-JPDAF (Svensson, Svensson & Willett 2009).

So far we have outlined how the JPDAF approaches the problems of data association and state estimation. It remains to consider how target detection may be incorporated. One of the first methods proposed was that of (Bar-Shalom, Chang & Blom 1989) which used the interacting multiple model (IMM) algorithm in parallel with the JPDAF. When a the possibility of a new target existed, two filters were run in parallel, one assuming the existence of the new target, and one assuming its non-existence. Decision logic was included to choose between the two possibilities. A less computationally demanding approach was formulated by (Musicki, Evans & Stankovic 1994, Musicki & Evans 2004), in which each target was labeled with a probability of existence, allowing potential new targets to be assessed in parallel with tracking.

The JPDAF requires the dynamics of the targets and the observation process to be linear and Gaussian, or that an appropriate EKF-like linear approximation can be made. However, even given this requirement, the estimation process still involves a sub-optimal step. The final posterior state distribution is constructed by adding a weighted sum of distributions, each made with a different association hypothesis. The individual components are Gaussians, and so the posterior should be a sum of Gaussian. However, in the JPDAF, this is collapsed into a single Gaussian with a matched mean and variance. This collapsing step may throw away important information if there is more than one significant component, and is responsible for the track coalescence effect noted before. It is possible to construct a filter which maintains the complete or a partially complete Gaussian sum posterior, (Singer, Sea & Housewright 1974, Salmond 1990), at the expense of increased complexity. In fact, this produces an algorithm broadly equivalent to the multi-hypothesis tracker (Blackman 2004)!

A final note about the JPDAF. The algorithm assumes that the only interaction between targets is through the associations - no two targets can be associated with the same observation. Thus, once the association probabilities have been calculated, each target may be updated independently using a Kalman filter to give a marginal state distribution. Interactions between targets are difficult to accomodate because we do not work in the joint state space of all targets.

- 2.4 Multi-hypothesis tracking**
- 2.5 MCMC data association**
- 2.6 Probability hypothesis density methods**

# *Mathematical foundations*

---

## 3.1 Bayes of our lives - what is filtering?

Many tasks in signal processing, science in general, and indeed life, require us to make some estimate of an unknown quantity from indirect, incomplete, or inaccurate observations. By constructing a model to explain how these observations depend on the underlying state, we can infer something about that state. We will express this observation model in terms of a likelihood function:

$$P(Y|X) \tag{3.1}$$

where  $X$  is the state and  $Y$  the observations. This is not the whole story - in many cases we are not estimating our unknown state “from scratch”. Previous experience, prejudice, and prior knowledge can also contribute to our estimates. The likelihood and prior terms can be combined through our friend, Bayes rule (Bayes & Price 1763, Laplace 1774), to calculate the posterior probability of the state, i.e. the probability of the state given the observations:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.2}$$

This is the basis of the process of inference. Mathematically, we can assign a probability distribution to the state space of  $X$ . By applying Bayes rule, we are updating our belief about the values of  $X$  using the information in  $Y$ .

Often the quantity in which we are interested,  $X$ , is changing over time, and we would like to estimate its value at each point in time given only the observations received so far. In this case we will generally assume that the state is Markovian, i.e. that  $X_t|X_{t-1}$  is independent of  $X_{1:t-2}$ , giving us a hidden Markov state-space model. This is the traditional filtering problem. By constructing a model for the evolution of the unknown state, we can now derive our prior information from our estimate at the previous time step. In discrete time, we now write:

$$P(X_t|Y_{1:t}) = \frac{\int P(Y_t|X_t)P(X_t|X_{t-1})P(X_{t-1}|Y_{1:t-1})dX_{t-1}}{P(Y_t|Y_{1:t-1})} \tag{3.3}$$

where the subscript indicates the time and ranges are notated by : in the MATLAB style.  $P(X_t|Y_{1:t})$  is called the filtering distribution. Equation 3.3 describes the ideal Bayesian filter.

Instead of marginalising the previous state, we may sometimes want to consider the joint state distribution over all time instances. This may similarly be expanded as:

$$P(X_{1:t}|Y_{1:t}) = \frac{P(Y_t|X_t)P(X_t|X_{t-1})P(X_{1:t-1}|Y_{1:t-1})}{P(Y_t|Y_{1:t-1})} \quad (3.4)$$

The filtering distribution may then be obtained by marginalising out the previous states.

So far, we have expressed the problem entirely in terms of distributions. The same models may be expressed in terms of difference equations of random variables. In the most general form:

$$X_t = f_t(X_{t-1}, V_t) \quad (3.5)$$

$$Y_t = g_t(X_t, W_t) \quad (3.6)$$

where  $f_t$  and  $g_t$  are known deterministic functions and  $V_t$  and  $W_t$  and random variables, known as the process and observation noise respectively.

## 3.2 Keep Kalman carry on - the Kalman filter and its extensions

### 3.2.1 The basic filter

In a few simple cases the filtering set-up permits the derivation of closed form posterior distributions at each time instant. Most notably, the Kalman filter (KF) (Kalman 1960) is an analytic filter for models with continuous state and observation variables, in which both transition and observation models are linear transformations with Gaussian innovations.

$$x_t = Ax_t + v_t \quad (3.7)$$

$$y_t = Cx_t + w_t \quad (3.8)$$

where  $v_t$  and  $w_t$  are now Gaussian random variables with zero mean and covariance matrices  $Q$  and  $R$ . We use lower case variables here to emphasise that these are “nice”, continuous vectors. (As we shall see, our state variable will later be sets or lists).

Kalman’s solution for the linear-Gaussian case is given by:

$$P(x_t|y_{1:t}) = \mathcal{N}(x_t|\mu_t, \Sigma_t) \quad (3.9)$$

$$P(x_t|y_{1:t-1}) = \mathcal{N}(x_t|\hat{\mu}_t, \hat{\Sigma}_t) \quad (3.10)$$

where  $\mu_t$ ,  $\Sigma_t$ , etc. are given by the following recursions.

Time Updates:

$$\hat{\mu}_t = A\mu_{t-1} \quad (3.11)$$

$$\hat{\Sigma}_t = A\Sigma_{t-1}A^T + Q \quad (3.12)$$

Measurement Updates:

$$z_t = y_t - C\hat{\mu}_t \quad (3.13)$$

$$S_t = C\hat{\Sigma}_tC^T + R \quad (3.14)$$



$$K_t = \hat{\Sigma}_t C^T S_t^{-1} \quad (3.15)$$

$$\mu_t = \hat{\mu}_t + K_t z_t \quad (3.16)$$

$$\Sigma_t = (I - K_t C) \hat{\Sigma}_t \quad (3.17)$$

The KF is delightful because it not only provides us with a closed-form analytic solution, but the complexity of that solution does not increase as we receive additional measurements. This is a consequence of the fact that the Gaussian distribution is its own conjugate prior. Unfortunately, no other such convenient cases have been discovered (Daum 2005). Analytic solutions to non-linear, non-Gaussian filtering problems generally require unacceptable conditions, such as zero process noise  $Q = 0$  (Daum 2005).

### 3.2.2 The extended filter

Given the loveliness of the KF, the instinct when faced by an intractable non-linear filtering problem is to linearise it. This produces the Extended Kalman Filter (EKF), and is achieved by replacing the  $A$  and  $C$  matrices in equations 3.11 through 3.17 above with Jacobians:

$$A_t = \left. \frac{\partial f}{\partial x_t} \right|_{\mu_{t-1}} \quad (3.18)$$

$$C_t = \left. \frac{\partial g}{\partial x_t} \right|_{\hat{\mu}_t} \quad (3.19)$$

### 3.2.3 The Kalman smoother

The KF gives us an optimum estimate of  $P(x_t|y_{1:t})$ . However, once more data has arrived, we can improve this estimate. For a given set of data,  $y_{1:T}$ , we can estimate the optimum estimates for all previous state distributions,  $P(x_{1:T}|y_{1:T})$  using a Rauch-Tung-Striebel (RTS) smoother, (Rauch, Tung & Striebel 1965). This begins with a normal KF, followed by a backward filtering pass which propagates information to earlier time instances. This backward pass is implement by the following recursions:

$$\tilde{\mu}_t = \mu_t + \Sigma_t A^T \hat{\Sigma}_{t+1}^{-1} (\tilde{\mu}_{t+1} - \hat{\mu}_{t+1}) \quad (3.20)$$

$$\tilde{\Sigma}_t = \Sigma_t + [\Sigma_t A^T \hat{\Sigma}_{t+1}^{-1}] (\tilde{\Sigma}_{t+1} - \hat{\Sigma}_{t+1}) [\Sigma_t A^T \hat{\Sigma}_{t+1}^{-1}]^T \quad (3.21)$$

giving us

$$P(x_t|Y_{1:T}) = \mathcal{N}(x_t|\tilde{\mu}_t, \tilde{\Sigma}_t) \quad (3.22)$$

For a full derivation, see (Rauch et al. 1965). There exist other ways to implement Kalman smoothing in a fixed-interval sense, such as the forward-backward smoother, and in a fixed-lag sense, but they will not be used in this work.

### 3.3 Tough as old bootstraps - a review of particle filter

In general we will not be so lucky as to have a problem with linear-Gaussian dynamics. In this case, a particle filter (PF) may be the best alternative. The PF is based on the idea that if we cannot represent a probability distribution analytically then we may approximate it as a collection of discrete samples or “particles” drawn from the distribution. Often, it will not even be possible to sample from the desired distribution. We therefore use importance sampling (IS) or Markov chain Monte Carlo methods to generate these samples. The principle of a particle filter is that we recursively generate such a set of particles to approximate  $P(X_{1:t}|Y_{1:t})$  given a previous set representing  $P(X_{1:t-1}|Y_{1:t-1})$ .

The particle filter

### 3.4 Particular details - mathematical basics of the particle filter

With a PF, we approximate a probability distribution with a set of (weighted) samples drawn from that distribution.

$$P(X) \approx \frac{1}{N} \sum_m W^{(m)} \delta_{X^{(m)}}(X) \quad (3.23)$$

where  $\delta(x)$  represents a unit probability point mass at a point  $x$ , and  $\sum_m W^{(m)} = 1$ . Such a method allows us to represent any probability distribution of arbitrary complexity, including multidimensional, multimodal, mixed distributions. As the number of particles increases, the accuracy of the approximation improves at the expense of computational complexity. Thus we have the required tool for estimation in non-linear, non-Gaussian scenarios.

We still face the problem of how to generate these samples. The conventional method for this is Sequential Importance Sampling with Resampling (SISR). For a more complete and traditional introduction to this method, see (Cappé, Godsill & Moulines 2007) or (Doucet & Johansen 2009). Here we follow an outline similar to that used for the derivation of the auxiliary particle filter of (Pitt & Shephard 1999).

Suppose we have a particle approximation to the joint posterior distribution from the previous frame,  $\hat{P}(X_{1:t-1}|Y_{1:t-1})$ . Each particle represents a path through time,  $X_{1:t-1}^{(m)}$ , and has an associated weight,  $W_{t-1}^{(m)}$ . Let us imagine also that the unweighted particles are samples from another distribution:

$$\mu(X_{1:t-1}|Y_{1:t-1}) \approx \frac{1}{N} \sum_m W^{(m)} \delta_{X_{1:t}^{(m)}}(X_{1:t}) \quad (3.24)$$

Thus for a given particle,

$$\hat{P}(X_{1:t-1}^{(m)}|Y_{1:t-1}) = W_t^{(m)} \mu(X_{1:t-1}^{(m)}|Y_{1:t-1}) \quad (3.25)$$

We would like to generate a new particle set which includes the current time instance. We propose a new set of extended tracks from a factored proposal distribution  $X_{1:t} \sim q(X_{1:t}|Y_{1:t}) = q(X_t|Y_t, X_{1:t-1})q(X_{1:t-1}|Y_{1:t})$ . The two factors are the proposal probabilities for the new state value,  $X_t$  and the history,  $X_{1:t-1}$ , respectively. Particles are then weighted to take account of the difference between the targeted posterior distribution and the importance distribution:

$$W_t^{(m)} = \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{q(X_{1:t}^{(m)}|Y_{1:t})} \quad (3.26)$$

The simplest choice for the history proposal is  $q(X_{1:t-1}|Y_{1:t}) = \mu(X_{1:t-1}|Y_{1:t-1})$ , which can be implemented by simply keeping the same set of paths as the previous particle set. This is equivalent to an ordinary IS step with no resampling, and importance weights are given by:

$$W_t^{(m)} = \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{q(X_{1:t}^{(m)}|Y_{1:t})} = \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{\mu(X_{1:t-1}^{(m)}|Y_{1:t-1})q(X_t^{(m)}|X_{t-1}^{(m)}, Y_t)} \approx \frac{W_{t-1}^{(m)}P(Y_t|X_t^{(m)})P(X_t^{(m)}|X_{t-1}^{(m)})}{q(X_t^{(m)}|X_{t-1}^{(m)}, Y_t)} \quad (3.27)$$

Alternatively, we could use  $q(X_{1:t-1}|Y_{1:t}) = \hat{P}(X_{1:t-1}|Y_{1:t-1})$  as the history proposal, i.e. sample from the weighted particle distribution which approximates the previous posterior. This is equivalent to an IS step preceded by resampling. Importance weights are now given by:

$$W_t^{(m)} = \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{q(X_{1:t}^{(m)}|Y_{1:t})} \approx \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{P(X_{1:t-1}^{(m)}|Y_{1:t-1})q(X_t^{(m)}|X_{t-1}^{(m)}, Y_t)} \approx \frac{P(Y_t|X_t^{(m)})P(X_t^{(m)}|X_{t-1}^{(m)})}{q(X_t^{(m)}|X_{t-1}^{(m)}, Y_t)} \quad (3.28)$$

#### 3.4.1 Auxiliary sampling

We can generalise the form of our history proposal distribution by weighting the particles from the previous posterior distribution with any arbitrary set of weights.

$$q(X_{1:t-1}|Y_{1:t}) = \frac{1}{N} \sum_m V_t^{(m)} \delta_X(x_{1:t}^{(m)}) \quad (3.29)$$

Now we have

$$\hat{P}(X_{1:t-1}|Y_{1:t-1}) = \frac{W_{t-1}^{(m)}}{V_t^{(m)}} q(X_{1:t-1}^{(m)}|Y_{1:t-1}) \quad (3.30)$$

giving a general form for the importance weights

$$W_t^{(m)} = \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{q(X_{1:t}^{(m)}|Y_{1:t})} = \frac{P(X_{1:t}^{(m)}|Y_{1:t})}{\mu(X_{1:t-1}^{(m)}|Y_{1:t-1})q(X_t^{(m)}|X_{t-1}^{(m)}, Y_t)} \approx \frac{W_{t-1}^{(m)}}{V_t^{(m)}} \times \frac{P(Y_t|X_t^{(m)})P(X_t^{(m)}|X_{t-1}^{(m)})}{q(X_t^{(m)}|X_{t-1}^{(m)}, Y_t)} \quad (3.31)$$

#### 3.4.2 Degeneracy and resampling

REWRITE THIS ENTIRELY

Which of the two choices of history proposal should we use? The first,  $\mu(X_{1:t-1}|Y_{1:t-1})$ , is the simplest to implement, because we can simply keep the same set of paths for  $1 : t - 1$ . However, the recursive form of the importance weights means that the variance of these weights

will increase over time. The result is the well-documented particle degeneracy effect, whereby all the weight coalesces in one or a few particles, and the weights of the rest tend to zero. Intuitively, this is a poor representation of the distribution - we may as well not have the zero-weight particles! The solution is to use resampling, or in the formulation posed above to use the second form of history proposal,  $\hat{P}(X_{1:t-1}|Y_{1:t-1})$ . This biases the sampling of histories towards those with high weights, encouraging those with low weights to be discarded. Because the importance weights for this type of proposal are not calculated recursively, the variance is reduced.

Particle degeneracy can be quantified ESS

Resampling strategies - systematic, multinomial, etc.

### 3.4.3 Importance distributions

It remains to choose the the importance distribution for the current state,  $q(X_t|X_{t-1}, Y_t)$ . In the original “bootstrap filter” of (Gordon et al. 1993), this was set equal to the transition density,  $P(X_t|X_{t-1})$ , which leads to cancellation in the expression for importance weights. This is simple but not necessarily optimal. For example if the process noise is high, the samples of  $X_t$  will be widely spread. If, however, the observation noise is comparatively low, many or most of the samples will be far from the observation and will have a low weight. This is undesirable, as discussed in section ??.

An improved choice of - Optimal importance dist

Add a discussion of the problem of dimensionality.

## 3.5 MCMC in da house - series vs. parallel

S-MCMC (Godsill) approach

# *Models for Tracking*

---

## 4.1 Why do we track?

In a target tracking scenario, the aim is to trace the trajectory of an object over time from a set of discrete observations. In the previous chapter we saw how we can carry out inference tasks using the recursions of the Kalman filter or the particle filter. Surely now we are done!? What else remains? The additional difficulty in target tracking is the nature of the observation process. Our imperfect sensors may not detect every target present in the scene in every scan. Furthermore, there may be some number of false alarms arising from sensor errors or clutter. Our task thus becomes three-fold: firstly to detect what targets are present in the scene, secondly to work out which observations were generated by each target, and finally to estimate the states of the targets. The basic Kalman and particle filters address only the third of these tasks.

Research in target tracking emerged from military applications such as radar and sonar. However, similar problems emerge in many diverse branches of science: the tracking of people, vehicles or animals in video sequences; of molecules or cells in microscopy data; or of notes in a piece of music. Each can be reduced to a similar underlying model. In this chapter we will formulate such a basic model, keeping it as general as possible rather than focusing on any specific application.

## 4.2 Models for tracking

In this section we introduce the basic models and notation for the object tracking problem which will be used throughout this report.

For our first assumptions, we will consider our targets to exist at singular points in space, and that targets move independently of each other. Each target will have be characterised by a state vector  $x_t$ . This state is composed of continuous-valued coordinates and evolves according to a hidden Markov model (HMM).

$$x_t = f(x_{t-1}, v_t) \tag{4.1}$$

where  $v_t$  is a random vector.

The targets are observed by a sensor which detects each with a probability  $P_D$ . If detected, the sensor returns a point observation at a location  $y_t$  given by:

$$y_t = g(x_t, w_t) \quad (4.2)$$

where  $w_t$  is another random vector, independent of  $v_t$ .

In addition to the target-originating observations, the sensor also detects a number of false alarms. Throughout this report we assume that these are generated by a Poisson process, with uniform intensity over the observation area.

We denote the set of targets present as  $X_t = \{x_{1,t}, x_{2,t}, \dots, x_{K_t,t}\}$ , and the set of observations as  $Y_t = \{y_t^{(1)}, y_t^{(2)}, \dots, y_t^{(M_t)}\}$ .

#### 4.2.1 Data association

In order to conduct inference on our multi-target system, we would like to calculate the likelihood,  $P(Y_t|X_t)$ . In order to evaluate this term, we need to hypothesise some assignment between the observations and the process which generated them, whether a particular target or clutter. We introduce an association variable for each target,  $\lambda_{j,t}$ , which indicates which of the observations in that frame was generated by this target. If the target is not detected, then  $\lambda_t$  is set to 0. We denote the set  $\Lambda_t = \{\lambda_{1,t}, \lambda_{2,t}, \dots, \lambda_{K_t,t}\}$ . As each observations is generated by one target or clutter, no two elements of  $\Lambda_t$  may take the same value, unless 0.

We can now expand the likelihood as:

$$P(Y_t|X_t) = \sum_{\Lambda_t} \prod_{j=1}^{K_t} P(y_t^{(\lambda_{j,t})} | x_{j,t}) \quad (4.3)$$

where the summation is over all feasible values of  $\Lambda_t$ . The number of terms in this summation is  $M_t(M_t - 1) \dots (M_t - K_t)$ , which may be prohibitively large if there are many targets or observations in the scene.

If it is necessary to calculate the likelihood very often, as in a particle filter, it may be preferable to estimate  $\Lambda_t$  instead of marginalising it.

### 4.3 Algorithms for tracking

JPDAF MHT MCMCDA Particle filters. Especially Vermaak RBPF

# *Fixed Lag Estimation for Tracking*

---

## 5.1 Why do now what we can do later?

In chapter 3, we looked at filtering methods, that is, estimation of a hidden state  $X_t$  given observations up until this time  $Y_{1:t}$ . For such a task, Kalman filters and particle filters are our tools of choice. However, for difficult target tracking problems, there may simply not be enough information available to make a good estimate. Consider a target with a low detection probability and high clutter. There is likely to be multiple observations with which the target could be associated in each frame, or the target could disappear entirely for a period. The result is that there is a very large number of possible routes through the observation space. A conventional particle filter will have to maintain particles on each of the possible routes until such time as they become negligibly unlikely. The correct route may at first only have a few particles on it, and when it is identified as the correct route in later frames the particle diversity may be poor. In the worst case, the correct route may at first look so unlikely (e.g. consecutive missed detections), that no particles follow it, and the track is lost.

This problem can be addressed by allowing not only a proposal of a new state  $X_t$  on the end of each path, but also the re-proposal of the preceding states as well. Thus, particles which have followed an incorrect route may be “redirected”, to follow a more probable course. This helps maintain particle diversity. If few or no particles follow the correct route at first, particles may be diverted later once the course becomes more apparent.

In an ordinary particle filter, we estimate the joint posterior,  $P(X_{1:t}|Y_{1:t})$  at each time instant. Thus, our estimates of the state distribution at previous times are updated at each step, but only by resampling. Diversity in the marginal distributions for previous states only decreases as time progresses. Conversely, in using such a system of re-proposals we allow diversity to be maintained in previous states. To limit the computational complexity, we constrain the region in which re-proposals may occur to a lagging window of  $L$  time steps.

## 5.2 A mathematical framework for fixed lag estimation

We now consider a mathematical framework for fixed lag estimation. This method was devised in (Doucet, Briers & Sénécal 2006) and (Briers, Doucet, Maskell & Horridge 2006).

As before, the target posterior distribution in which we are interested is the familiar  $P(X_{1:t}|Y_{1:t})$ . However, the proposal mechanism now becomes more complex, because we will be replacing states in an existing particle. We first propose a particle from which to take the state “history”, that is  $X_{1:t-L}$ . This may be chosen from the particle approximation from any of the previous  $L$  processing steps. However, we get more of the path than we need, because each particle at lag  $s$  is a set of states  $X_{1:t-s}$ . The final  $L-s$  states will be replaced when by a new “tip”,  $X'_{t-L+1:t}$ , drawn from an importance distribution. The complete proposal is thus

$$\{X_{1:t-L}, X'_{t-L+1:t}\} \sim q(s) \int q(X_{1:t-s}|Y_{1:t-s})q(X'_{t-L+1}|X_{1:t-s}, Y_{t-L+1:t})dX_{t-L+1:t-s} \quad (5.1)$$

where  $q(X_{1:t-s}|Y_{1:t-s})$  is a proposal distribution using the arbitrarily-weighted particles from  $\hat{P}(X_{1:t-s}|Y_{1:t-s})$ . We cannot evaluate this. In general, the integral will be intractable. If we restrict our proposals to depend only on the history, i.e. use  $q(X_{t-L+1}|X_{1:t-L}, Y_{t-L+1:t})$ , then the proposal becomes

$$\{X_{1:t-L}, X'_{t-L+1:t}\} \sim q(s)\hat{P}(X_{1:t-L}|Y_{1:t-s})q(X'_{t-L+1}|X_{1:t-s}, Y_{t-L+1:t}) \quad (5.2)$$

However, with this proposal, in order to evaluate importance weights or acceptance probabilities we still need to calculate

$$P(X_{1:t-L}|Y_{1:t-s}) = \frac{P(Y_{t-L+1:t-s}|X_{t-L}^{(m)})P(X_{1:t-L}|Y_{1:t-L})}{P(Y_{t-L+1:t-s}|Y_{1:t-L})} \quad (5.3)$$

The problematic term is:

$$P(Y_{t-L+1:t}|X_{t-L}^{(m)}) = \int P(Y_{t-L+1:t}|X_{t-L}^{(m)}, X_{t-L+1:t})P(X_{t-L+1:t}|X_{t-L}^{(m)})dX_{t-L+1:t-s} \quad (5.4)$$

In general, this will be intractable. The exception is the case where  $s = L$ , but this will in general not work well - such a requirement implies ignoring all the estimation we have done since the start of the window!

The solution to this tractability problem proposed in (Doucet et al. 2006) is to augment the dimension of the target distribution to include the discarded tracks. The new target is:

$$P(X_{1:t-L}, X'_{t-L+1:t}|Y_{1:t})\rho(X_{t-L+1:t-s}|X_{1:t-L}, X'_{t-L+1:t}) \quad (5.5)$$

Once a particle approximation has been generated for this target the required posterior distribution may be obtained by marginalisation. As we are simply going to discard the old track sections  $X_{t-L+1:t-s}$ , the choice of  $\rho(\cdot)$  does not alter the distribution of the posterior. However, the variance of weights/acceptance probabilities may be affected.

With an expanded space for the target distribution there is no need to marginalise the previous states. The proposal distribution now becomes:

$$\{X_{1:t-s}, X'_{t-L+1:t}\} \sim q(s)q(X_{1:t-s}|Y_{1:t-s})q(X'_{t-L+1}|X_{1:t-s}, Y_{t-L+1:t}) \quad (5.6)$$



### 5.2.1 Artificial conditionals

We use  $\rho(\cdot) = q(\cdot)$ . Why?

### 5.2.2 History Proposals

“Conservative” resampling as a history proposal.

## 5.3 Applying the fixed lag method to the tracking model

Erm... i’ve kinda said what I was going to say here in the previous section

### 5.3.1 Importance distributions

The dimensionality of a fixed lag particle filter will generally be very large. Consider a problem where each target has a 4-dimensional kinematic state (position and velocity in  $x$  and  $y$ ) and an observation-association index. This gives us five dimensions per target per time step. If we use a lag window with length  $L = 5$  and 5 targets we will have a 125 dimensional state. If we used a basic, bootstrap approach to such a problem, the chances of even a single particle following the correct path are negligible. Instead we must exploit the strong correlations between states arising from the structure of the problem. We first factorise the proposal thus:

$$\begin{aligned} q(x_{t-L+1:t}, \lambda_{t-L+1:t} | x_{t-L}, Y_{t-L+1:t}) \\ &= q(\lambda_{t-L+1:t} | x_{t-L}, Y_{t-L+1:t}) \\ &= q(x_{t-L+1:t} | x_{t-L}, \lambda_{t-L+1:t}, Y_{t-L+1:t}) \end{aligned} \quad (5.7)$$

Thus we can sample first the association variables then the state variables. Each of these terms can be further factorised over time, as we shall see. If we were to replace each of the factors with its corresponding posterior distribution then this factorisation would recreate the “optimal” importance distribution.

#### Association proposals

The proposal for the association variables is a discrete distribution over the possible observations with which the target could be associated. Within the factorisation above, the “optimal” form of the proposal is:

$$\begin{aligned} q(\lambda_{t-L+1:t} | x_{t-L}, Y_{t-L+1:t}) \\ &\propto P(Y_{t-L+1:t} | \lambda_{t-L+1:t}, x_{t-L}) P(\lambda_{t-L+1:t} | x_{t-L}) \\ &= \prod_{\substack{k=1:L \\ tt=t-L+k}} P(Y_{tt} | Y_{tt+1:t} \lambda_{tt:t}, x_{t-L}) P(\lambda_{tt}) \\ &= \prod_{\substack{k=1:L \\ tt=t-L+k}} \int P(Y_{tt} | x_{tt}, \lambda_k) P(x_{tt} | Y_{tt+1:t}, \lambda_{tt+1:t}, x_{t-L}) dx_{tt} P(\lambda_{tt}) \end{aligned} \quad (5.8)$$

This suggests a convenient sequential sampling procedure, starting with  $\lambda_t$  and working backwards in time. The normalisation constant is not known, but as this is discrete distribution we can enforce normalisation by dividing by the sum.

First we consider a factor from this expression with  $\lambda_{tt} = 1$ , i.e. a proposal that in a particular frame the target is not detected. In this case, the observation density is independent of the state of the target, and we have:

$$P(Y_{tt}|Y_{tt+1:t}\lambda_{tt:t}, x_{t-L})P(\lambda_{tt}) = V^{-1}P(\lambda_t = 0) \quad (5.9)$$

When the target is detected, the factors of the proposal distribution of equation 5.8 may be calculated analytically for the linear-Gaussian case. For other cases, the EKF approximations may be used. As this is only a proposal distribution, such an approximation will not affect the distribution of the particle distribution generated. The state distribution term over  $x_{tt}$  is problematic, requiring  $k$  integrals to calculate. Although this is will be analytic with Gaussian dynamics, its complexity may become unmanageable. This term represents the probability of the state given the set of observations associated with this target at later times. We can render this calculation more manageable by replacing the whole set of future observations with just one.

$$P(x_k|Y_{k+1:t}, \lambda_{k+1:t}, x_{t-L}) \approx P(x_k|Y_{k+d}, \lambda_{k+d}, x_{t-L}) \quad (5.10)$$

For cases with low observation noise, where an observation gives us significant information about the state of the target, this substitution will have little effect. Later observations cannot add much additional information. Again, as this is a proposal distribution, such a substitution will not affect the validity of the resulting particle distribution.

In general we will use  $d = 1$ , as the closest observation in time will give us the most information about  $x_{tt}$ . However, if the target is not detected at time  $tt + 1$ , then we can increase  $d$  to find the next detection of the target.

Using this approximation, for  $\lambda_{t-L+k} \neq 0$ , we have

$$P(Y_{tt}|Y_{tt+1:t}\lambda_{tt:t}, x_{t-L})P(\lambda_{tt}) \propto \mathcal{N}(y_{tt}^{(\lambda_{tt})}|m_{tt}, S_{tt}) \quad (5.11)$$

where usually

$$S_{tt} = [I - R^{-1}C_{tt}\Sigma_{tt}C_{tt}^TR^{-1}]^{-1} \quad (5.12)$$

$$m_{tt} = SR^{-1}C_{tt}\Sigma_{tt}[(A^d)^TC_{tt+d}^TR_d^{-1}y_{tt+d}^{\lambda_{tt+d}} + Q_k^{-1}A^kx_{t-L}] \quad (5.13)$$

$$\Sigma_{tt} = [C_{tt}^TR^{-1}C_{tt} + (A^d)^TC_{tt+d}^TR_d^{-1}C_{tt+d}A^d + Q_k^{-1}]^{-1} \quad (5.14)$$

$$Q_d = \sum_{l=0}^{d-1} A^lQ(A^l)^T \quad (5.15)$$

$$R_d = R + C_{tt+d}Q_d(C_{tt+d})^T \quad (5.16)$$

We will need a different expression for the case when  $tt = t$ , because no future associations have yet been proposed. Similarly, if  $tt < t$  but the future associations have all been proposed as missed detections, then there are no future observations to guide us, whatever choice of  $d$  we use. In these cases we have:

$$S_{tt} = R_d \quad (5.17)$$

$$m_{tt} = C_{tt} A^k x_{t-L} \quad (5.18)$$

For full derivations, see Appendix.

Finally, substituting for the association prior terms, we have:

$$q(\lambda_{t-L+1:t} | x_{t-L}, Y_{t-L+1:t}) \propto \prod_{\substack{k=1:L \\ tt=t-L+k}} \begin{cases} P_D \mathcal{N}(y_{tt}^{(\lambda_{tt})} | m_{tt}, S_{tt}) & \lambda_{tt} = 0 \\ (1 - P_D) \mu_C V^{-1} & \lambda_{tt} \neq 0 \end{cases} \quad (5.19)$$

This gives us a complete sequential mechanism for proposing the associations.

#### State proposals

Once the associations are fixed, the states can be proposed. When the state space model is linear-Gaussian, we can propose directly from the “optimal” importance distribution for the states using the forward-filtering-backward-sampling algorithm of (Chib 1996), as suggested in (Doucet et al. 2006). For nonlinear models, we can use EKF approximations, as for the associations. Once again, we factorise the proposal:

$$\begin{aligned} q(x_{t-L+1:t} | x_{t-L}, \lambda_{t-L+1:t}, Y_{t-L+1:t}) \\ &= P(x_{t-L+1:t} | x_{t-L}, \lambda_{t-L+1:t}, Y_{t-L+1:t}) \\ &= P(x_t | \lambda_{t-L+1:t}, Y_{t-L+1:t}, x_{t-L}) \prod_{k=t-L+1}^{t-1} P(x_k | \lambda_{t-L+1:k}, Y_{t-L+1:k}, x_{t-L}, x_{k+1}) \end{aligned} \quad (5.20)$$

where

$$P(x_k | \lambda_{t-L+1:k}, Y_{t-L+1:k}, x_{t-L}, x_{k+1}) \propto P(x_{k+1} | x_k) P(x_k | \lambda_{t-L+1:k}, Y_{t-L+1:k}, x_{t-L}) \quad (5.21)$$

The distributions  $P(x_k | \lambda_{t-L+1:k}, Y_{t-L+1:k}, x_{t-L})$  are given by a Kalman filter, and are Gaussian with mean  $\mu_k$  and covariance  $\Sigma_k$ . Thus the complete state proposal is given by:

$$q(x_{t-L+1:t} | x_{t-L}, \lambda_{t-L+1:t}, Y_{t-L+1:t}) = \mathcal{N}(x_t | \mu_t, \Sigma_t) \prod_{k=t-L+1}^{t-1} \mathcal{N}(x_k | m_k, S_k) \quad (5.22)$$

where

$$S_k = [A^T Q^{-1} A + \Sigma^{-1}]^{-1} \quad (5.23)$$

$$m_k = S_k [A^T Q^{-1} x_{k+1} + \Sigma^{-1} \mu_k] \quad (5.24)$$



## *Fixed Lag Particle Filters*

---

### **6.1 Sequential Importance Sampling and Resampling**

#### *6.1.1 Coping with dimensionality*

### **6.2 Markov Chain Monte Carlo**

### **6.3 Marginalised Particle Filters**



## *Plan Of Future Research*

---

radar, insects, crowds, cells, jump-diffusions, cursor tracking...





# *Bibliography*

---

- Al-Awadhi, F., Hurn, M. & Jennison, C. (2004). Improving the acceptance rate of reversible jump mcmc proposals, *Statistics & Probability Letters* **69**(2): 189 – 198.  
URL: <http://www.sciencedirect.com/science/article/B6V1D-4CT4DWS-2/2/cab144a2f70e81f189a0e819c4517adc>
- Bar-Shalom, Y., Chang, K. & Blom, H. (1989). Automatic track formation in clutter with a recursive algorithm, *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on*, IEEE, pp. 1402–1408.
- Bar-Shalom, Y., Daum, F. & Huang, J. (2009). The probabilistic data association filter, *Control Systems Magazine, IEEE* **29**(6): 82 –100.
- Bar-Shalom, Y. & Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association, *Automatica* **11**(5): 451–460.
- Bayes, M. & Price, M. (1763). An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs, *Philosophical Transactions* **53**: 370.
- Blackman, S. (2004). Multiple hypothesis tracking for multiple target tracking, *IEEE Aerospace and Electronic Systems Magazine* **19**(1): 5–18.
- Blake, A., Isard, M. et al. (1998). *Active contours*, Vol. 8, Springer London.
- Briers, M., Doucet, A., Maskell, S. & Horridge, P. (2006). Fixed-lag sequential monte carlo data association, *Proceedings of SPIE*, Vol. 6236, p. 62360S. Paper copy from British Library.
- Bunch, P. & Godsill, S. (2010). Transcription of musical audio using poisson point processes and sequential mcmc, *Proc. of 7th International Symposium on Computer Music Modeling and Retrieval*.
- Cappé, O., Godsill, S. & Moulines, E. (2007). An overview of existing methods and recent advances in sequential monte carlo, *Proceedings of the IEEE* **95**(5): 899–924.

- Carmi, A., Godsill, S. & Septier, F. (2009). Evolutionary mcmc particle filtering for target cluster tracking, *IEEE 13th DSP Workshop and the 5th SPE Workshop*, Marco Island, Florida.
- Casella, G. & Robert, C. P. (1996). Rao-blackwellisation of sampling schemes, *Biometrika* **83**(1): 81–94.  
URL: <http://biomet.oxfordjournals.org/content/83/1/81.abstract>
- Chib, S. (1996). Calculating posterior distributions and modal estimates in markov mixture models, *Journal of Econometrics* **75**(1): 79 – 97. Details forward-filtering-backward-sampling algorithm.  
URL: <http://www.sciencedirect.com/science/article/B6VC0-3VWT1TW-7/2/b66dae744497c95ff3eddb744f301a6c>
- Daum, F. (2005). Nonlinear filters: beyond the kalman filter, *Aerospace and Electronic Systems Magazine, IEEE* **20**(8): 57 –69.
- Daum, F. & Huang, J. (2003). Curse of dimensionality and particle filters, *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, Vol. 4, pp. 1979–1993.
- Doucet, A., Briers, M. & Sénécal, S. (2006). Efficient block sampling strategies for sequential monte carlo, *Journal of Computational and Graphical Statistics* **15**(3): 693–711.
- Doucet, A., Godsill, S. & Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering, *Statistics and Computing* **10**: 197–208. 10.1023/A:1008935410038.  
URL: <http://dx.doi.org/10.1023/A:1008935410038>
- Doucet, A. & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later, in D. Crisan & B. Rozovsky (eds), *The Oxford Handbook of Nonlinear Filtering*, Oxford University Press.  
URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.772&rep=rep1&type=pdf>
- Duane, S., Kennedy, A. D., Pendleton, B. J. & Roweth, D. (1987). Hybrid monte carlo, *Physics Letters B* **195**(2): 216 – 222.  
URL: <http://www.sciencedirect.com/science/article/pii/037026938791197X>
- Fitzgerald, R. (1985). Track biases and coalescence with probabilistic data association, *Aerospace and Electronic Systems, IEEE Transactions on* **AES-21**(6): 822 –825.
- Fitzgerald, R. J. (1986). Development of practical pda logic for multitarget tracking by micro-processor, *American Control Conference, 1986*, pp. 889 –898.
- Fortmann, T., Bar-Shalom, Y. & Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association, *Oceanic Engineering, IEEE Journal of* **8**(3): 173 – 184.
- Geman, S. & Geman, D. (1984). Gibbs distributions, and the bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(2): 721–741.

## BIBLIOGRAPHY

---

- Gilks, W. R. & Berzuini, C. (2001). Following a moving target monte carlo inference for dynamic bayesian models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(1): 127–146.  
URL: <http://dx.doi.org/10.1111/1467-9868.00280>
- Gilks, W., Richardson, S. & Spiegelhalter, D. (1996). *Markov chain Monte Carlo in practice*, Chapman and Hall.
- Girolami, M. & Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**(2): 123–214.  
URL: <http://dx.doi.org/10.1111/j.1467-9868.2010.00765.x>
- Golightly, A. & Wilkinson, D. (2006). Bayesian sequential inference for nonlinear multivariate diffusions, *Statistics and Computing* **16**: 323–338. 10.1007/s11222-006-9392-x.  
URL: <http://dx.doi.org/10.1007/s11222-006-9392-x>
- Gordon, N., Salmond, D. & Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation, *Radar and Signal Processing, IEE Proceedings F* **140**(2): 107–113.
- Green, P. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination, *Biometrika* **82**(4): 711.
- Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications, *Biometrika* **57**(1): 97–109.  
URL: <http://biomet.oxfordjournals.org/content/57/1/97.abstract>
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Journal Of Basic Engineering* **82**(Series D): 35–45.  
URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.6247&rep=rep1&type=pdf>
- Khan, Z., Balch, T. & Dellaert, F. (2005). Mcmc-based particle filtering for tracking a variable number of interacting targets, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**(11): 1805–1819.
- Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models, *Journal of Computational and Graphical Statistics* **5**(1): pp. 1–25.  
URL: <http://www.jstor.org/stable/1390750>
- Laplace, P. S. (1774). Memoir on the probability of the causes of events, *Statistical Science* **1**(3): pp. 364–378.  
URL: <http://www.jstor.org/stable/2245476>
- Liu, J. S. & Chen, R. (1995). Blind deconvolution via sequential imputations, *Journal of the American Statistical Association* **90**(430): pp. 567–576.  
URL: <http://www.jstor.org/stable/2291068>
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. et al. (1953). Equation of state calculations by fast computing machines, *The journal of chemical physics* **21**(6): 1087.

- Musicki, D. & Evans, R. (2004). Joint integrated probabilistic data association: Jipda, *Aerospace and Electronic Systems, IEEE Transactions on* **40**(3): 1093 – 1099.
- Musicki, D., Evans, R. & Stankovic, S. (1994). Integrated probabilistic data association, *Automatic Control, IEEE Transactions on* **39**(6): 1237 –1241.
- Pang, S., Godsill, S., Li, J. & Septier, F. (2011). Sequential inference for dynamically evolving groups of objects, in D. Barber, A. Cemgil & S. Chiappa (eds), *Inference and Learning in Dynamic Models*, CUP. To Appear.
- Pang, S. K., Li, J. & Godsill, S. (2008). Models and algorithms for detection and tracking of coordinated groups, *Aerospace Conference, 2008 IEEE*, pp. 1 –17.
- Pitt, M. K. & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters, *Journal of the American Statistical Association* **94**(446): pp. 590–599.  
URL: <http://www.jstor.org/stable/2670179>
- Rauch, H., Tung, F. & Striebel, C. (1965). Maximum likelihood estimates of linear dynamic systems, *AIAA journal* **3**(8): 1445–1450.
- Salmond, D. (1990). Mixture reduction algorithms for target tracking in clutter, *Proceedings of SPIE*, Vol. 1305, p. 434.
- Sea, R. G. (1971). An efficient suboptimal decision procedure for associating sensor data with stored tracks in real-time surveillance systems, *Decision and Control, 1971 IEEE Conference on*, Vol. 10, pp. 33 –37.
- Septier, F., Pang, S. K., Carmi, A. & Godsill, S. (2009). On mcmc-based particle methods for bayesian filtering: Application to multitarget tracking, *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*, pp. 360 –363.
- Singer, R., Sea, R. & Housewright, K. (1974). Derivation and evaluation of improved tracking filter for use in dense multitarget environments, *Information Theory, IEEE Transactions on* **20**(4): 423 – 432.
- Svensson, L., Svensson, D. & Willett, P. (2009). Set jpda algorithm for tracking unordered sets of targets, *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pp. 1187 –1194.