The Particle Filter

Ben Lowery

February 22, 2022

Overview

Real world data has long had a tendency to be messy. This is even more so the case when estimating unknown quantities, even if we know some prior information about the data. For example, say we wish to figure out the location of a target, whose movement is being governed by factors such as position and velocity. And then consider our only reading to base a decision on the location being the bearings of the object. Weave into this noisy, irreverent and nonsense data, commonplace with the real world, and what remains is a cacophony of disruption. However, we do know the prior locations of the object, as well as some intuition and knowledge of an underlying system that is governing the movements. The result of this rather convoluted problem set up is that it could not be solved exactly, lacking what is known as mathematical tractability. This pertains to the idea that a solution could be reasonably found in a closed-form, without causing too much hindrance to the solver.

The origins of such a dilemma came from the work of Gordon et al. (1993). They wished to solve problems like the aforementioned conundrum in which we deal with a large number of different non-linear variables (high-dimensional) and data that does not follow a process we know has a lot of nice statistical properties (it was non-Gaussian distributed). In general, they aimed to solve a more general class of problems, with commonality exhibited in dealing with observations which arrive in an on-line fashion, where sequential observations are attained and used to dynamically update the state of the system. They found the state of the art solution at the time, known as the Extended Kalman Filter, to be insufficient at dealing with highly complex models in general, as they didn't take into account the intricacies of the data and its underlying statistical properties. Instead, a heuristic - which is a method that provides "good" but not perfect results - was presented. Which is undeterred by assumptions of mathematical tractability and could be applied in the broadest setting to most complex models. Their Particle Filter, which went under the guise of the Bootstrap Filter, is the main subject of this paper. It allows for inference on non-linear, non-Gaussian, and high-dimensional problem settings, with a primary focus on solving the Filtering Problem. The Filtering Problem involves finding the state of the unobserved system (think position and velocity in the tracking example), when a new point is observed (its location based on its bearings).

The Particle Filter belongs to a family known as Monte Carlo methods, which are based on solving problems through random number generation. The Particle Filter also has foundations stemming from ideas of importance sampling, a method that involves finding information of a probability distribution by taking information from a different probability distribution. The Particle Filter encompasses a wide array of methods, too stretched to talk about in this short paper. The primary scope is to view Particle Filters in the context of generic problems which all share commonality in possessing observed data conditional on some unobserved states. However it can also be used in solving issues of smoothing (a problem parallel to filtering) and estimating the key parameters of the system.

1 Introduction

Working on methods for the tracking of a moving target given only their bearings, which resulted in observations of the targets location being partial and noisy; Gordon et al. (1993) presented a method known as the Particle Filter (or Sequential Monte Carlo in some literature) which used recursive Bayesian estimation to estimate the posterior mean of the tracked target. The author's proposed "Bootstrap Filter" could be applied to any state-space model and allowed for online inference on nonlinear models, where beliefs need to be quickly updated and incorporated into the state of the system as new observations arrive. The heuristic was found to be far superior to the already existing Extended Kalman Filter (See Terejanu 2009, for tutorial) and could be applied to any state estimation problem. This then leads to the method becoming very appealing to a wide range of settings. Since this initial work, other Particle Filter algorithms have developed, with the review article by Fearnhead & Künsch (2018) exploring both classic and recent methods. In this paper, the Bootstrap Filter will serve as the primary interest and is formulated in Section 3, with an implementation and example for a simple random walk model. Before that, we first consider the underlying model in which Particle Filters are often applied to, and mathematical formalisation of the problem they are solving in the first place.

2 State-space models and filtering

The state-space model, which can also go by Hidden Markov Model, is a type of probabilistic model consisting of two parts: (1) observable, noisy data and (2) unobserved, latent states describing the true state of the underlying system. These latent states ($\mathbf{X}_t$) are modelled as a Markov process, with the observed data ($\mathbf{Y}_t$) assumed conditionally independent to the process. An example of a state-space model, represented by a directed acyclic graph, is given in Figure 1.

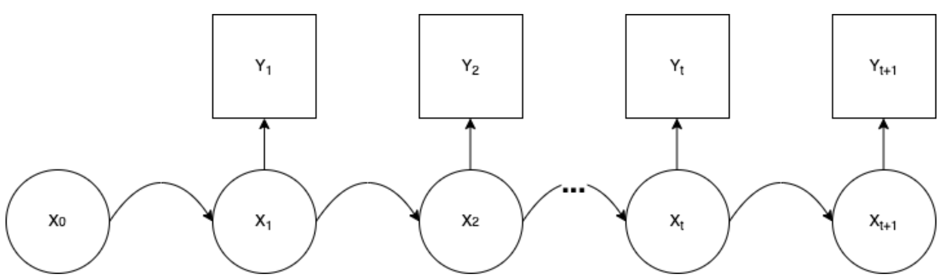


Figure 1: Graphical representation of the state-space model.

For brevity, a sequence of time steps can be written as $\mathbf{x}_{0:t} = \{x_0, x_1, .., x_t\}$ and $\mathbf{y}_{0:t} = \{y_0, y_1, .., y_t\}$. Quantities we are interested in are the transition probability density function (PDF) f and and the observation PDF g. In Equations (1) and (2), the respective transition and observation functions are given:

$$X_t|(X_{t-1} = x_{t-1}) \sim f(x_t|x_{t-1}), \tag{1}$$

$$Y_t|(X_t = x_t) \sim g(y_t|x_t). \tag{2}$$

Note, at time $t = 0$, the initial PDF for the Markov process is $X_0 \sim \mu(x_0)$. The aim is to calculate an estimate for the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, in particular the marginal distribution $p(x_t|\mathbf{y}_{1:t})$ which goes by the filtering distribution. Using Equations (1) and (2), an expression for the posterior at any time t is formulated using Bayes' rule,

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} = \frac{\mu(x_0)\prod_{t=1}^{T} g(y_t|x_t)f(x_t|x_{t-1})}{\int \mu(x_0)\left[\prod_{t=1}^{T} g(y_t|x_t)f(x_t|x_{t-1})\right] \mathrm{d}\mathbf{x}_{0:t}}. \tag{3}$$

A recursive expression for the posterior can be recovered and is given by Equation (4):

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{g(y_t|x_t)f(x_t|x_{t-1})}{p(y_t|y_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}), \tag{4}$$

where,

$$p(y_t|y_{1:t-1}) = \iint g(y_t|x_t)f(x_t|x_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1}\mathbf{x}_t.$$

The filtering distribution is expressed in two stages - the prediction and update step - as follows,

$$\text{Prediction: } p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int f(x_t|x_{t-1})p(x_{t-1}|\mathbf{y}_{1:t-1})\mathrm{d}x_{t-1}, \tag{5}$$

$$\text{Update: } p(x_t|\mathbf{y}_{1:t}) = \frac{g(y_t|x_t)p(x_t|\mathbf{y}_{1:t-1})}{\int g(y_t|x_t)p(x_t|\mathbf{y}_{1:t-1})\mathrm{d}x_t}. \tag{6}$$

Another interest is the expected value of the state-space model,

$$I = \mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[h(\mathbf{x}_{0:t})] = \int h(\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t}, \tag{7}$$

for some arbitrary function we are wanting to evaluate (for example the mean $h(\mathbf{x}_{0:t}) = \mathbf{x}_{0:t}$).

Estimating the filtering distribution revolves around tackling the filtering problem. This is the prominent issue amongst literature surrounding the Particle Filter and pertains to tracking the latent states as a new observation streams in. The filtering distributions are usually difficult to compute, as they require calculation of high-dimensional integrals. One traditional way to sample from complex distributions is the use of MCMC methods, which are an iterative class of methods. However, the state-space models' posterior distributions being expressed recursively makes MCMC an unsuitable method (Doucet et al. 2001). Instead, a variant of Importance Sampling can be applied to the recursive setting to estimate the posterior and filtering distributions.

3 The Bootstrap Filter

A classical Monte Carlo method for estimating properties of complex, hard to evaluate, distributions is Importance Sampling (IS). As a motivating illustration of the technique, say we wish to evaluate the expected value of the state-space model (Equation 7), the fundamental idea is to draw samples from a proposal distribution ($\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$) which has the same support as the target distribution ($p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$), the target being known only up to proportionality. The proposal distribution is deliberately chosen to be tractable and easy to sample from, and with the use of simple algebra, an expression for the the expected value is given as follows,

$$I = \mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[h(\mathbf{x}_{0:t})] = \frac{\int h(\mathbf{x}_{0:t})\tilde{w}(\mathbf{x}_{0:t})\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t}}{\int \tilde{w}(\mathbf{x}_{0:t})\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t}}, \tag{8}$$

where $\tilde{w} = \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}$ are known as the importance weights. Sampling x_t N-times, where each sample is known as a 'particle', a Monte Carlo estimate for the expectation is derived,

$$\tilde{I}_N = \frac{\frac{1}{N}\sum_{i=1}^{N} h(\mathbf{x}_{0:t}^{(i)})\tilde{w}(\mathbf{x}_{0:t}^{(i)})}{\frac{1}{N}\sum_{j=1}^{N} \tilde{w}(\mathbf{x}_{0:t}^{(j)})} = \sum_{i=1}^{N} h(\mathbf{x}_{0:t}^{(i)}) w_t^{(i)}, \tag{9}$$

with the normalised importance weights $w_t^{(i)} = \frac{\tilde{w}(\mathbf{x}_{0:t}^{(i)})}{\sum_{j=1}^{N}\tilde{w}(\mathbf{x}_{0:t}^{(j)})}$. For some problem settings this formulation is an adequate technique for estimation, however it is not particularly applicable for a recursive scenario. As a new data point y_t is observed, the entire chain $\mathbf{y}_{0:t-1}$ and importance weights need to be recalculated; subsequently leading to an increase in computational complexity, as well as issues when taken to higher dimensions (Robert & Casella 2004).

3.1 Sequential Importance Sampling

Following on from IS, a method to fix the computational complexity problems arrives with Sequential Importance Sampling (SIS). Here, the proposal distribution is selected to have the following recursive structure,

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(x_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = \pi(x_0)\prod_{k=1}^{t}\pi(x_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}), \tag{10}$$

and the (normalised) importance weights recalculated as,

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{g(y_t|x_t^{(i)}) f(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}. \tag{11}$$

A simple but effective choice of proposal distribution for this scheme is the prior,

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}) = \mu(x_0)\prod_{k=1}^{t} f(x_k|x_{k-1}). \tag{12}$$

This allows the weights to satisfy $w_t^{(i)} \propto w_{t-1}^{(i)} g(y_t|x_t^{(i)})$, and the soon to be stated Bootstrap Filter assumes a prior proposal distribution.

3.2 SIS with Resampling

While SIS appears to be a novel and efficient method to calculate the recursive posterior distributions, it is also prone from suffering the same problems that occur in Importance Sampling and it can be shown that Importance Sampling inefficiencies when taken to high dimensions is analogous to SIS when t is increased (Doucet & Johansen 2009). This results in the weight distribution possessing a heavy skew and most weights tending towards 0, we call this Weight Degeneracy. To avoid the problem, researchers added a resampling step to the algorithm.

In this extra phase, we eliminate weights with low importance and reward those with high importance. One such sampling technique is Multinomial Resampling, where we obtain samples $N_t^{(1:N)}$ from a multinomial distribution, with the probability parameter given by the weights $w^{(1:N)}$. The method can be sampled from efficiently in $\mathcal{O}(n)$ operations (Doucet & Johansen 2009). Another popular (and widely used) technique is systematic resampling (Kitagawa 1996), however, for the purpose of this paper - and later example - Multinomial Resampling suffices. Incorporating this resampling step to SIS results in the Particle Filter method known as the Bootstrap Filter of Gordon et al. (1993). The steps of this algorithm is given by Algorithm 1.

Algorithm 1: Bootstrap Filter

Input: N particles, T rounds

1. Initialisation: $t = 0$

for $i = 1, ..., N$ **do**

  Draw sample $x_0^{(i)} \sim \mu(x_0)$

end

for t=1,..,T **do**

  2. Importance Sampling:

  for $i = 1, .., N$ **do**

    Draw sample $\tilde{x}_t^{(i)} \sim f\left(x_t | x_{t-1}^{(i)}\right)$ and set $\tilde{x}_{0:t}^{(i)} = \left(x_{0:t-1}^{(i)}, \tilde{x}_t^{(i)}\right)$

  end

  for $i = 1, .., N$ **do**

    Importance weights: $\tilde{w}_t^{(i)} = g(y_t | \tilde{x}_t^{(i)})$.

  end

  Normalise: for each i, $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_i w_t^{(i)}}$

  3. Resampling: Resample (with replacement) each particle $\left(x_{0:t}^{(i)} : i = 1, ..., N\right)$ from $\left(\tilde{x}_{0:t}^{(i)} : i = 1, ..., N\right)$ with probability given by their respective importance weight.

end

Note, the omission of Step 3 is tantamount to simply implementing SIS and, once finished, the algorithm yields the filtering distribution $p(x_t | \mathbf{y}_{1:t})$ for each time step. The algorithm can also be used to calculate the expected value of some given quantity $h(x)$, with the calculation of Equation (9) carried out just before the resampling step.

3.3 Theoretical Results of Particle Filters

Much effort has been dedicated to deriving convergence properties of the Particle Filter. Chopin (2004) showed that, for general Particle Filter methods (referring to these as SMC's in their paper), if Multinomial Resampling was used at ever step, then a Central Limit Theorem is established for both the expectation estimation (Equation 9) and normalising constant of the posterior. That is, as the number of particles N approaches infinity, the Particle Filter follows a Normal Distribution centered on its true value. It has also been shown in various scenarios that the asymptotic variance in Particle Filter methods are orders of magnitude smaller than that of IS and SIS (Doucet & Johansen 2009).

3.4 Example: Latent Gaussian process

We consider a simple example of a linear Gaussian model. As this is a tractable model, the filtering distribution can be solved analytically with the well known Kalman Filter (Kalman 1960), in turn this is then used to evaluate the performance of both SIS and the Bootstrap Filter. For an unobserved process $\mathbf{X}_t$ and noisy observations $\mathbf{Y}_t$, consider the random walk model,

$$X_t = X_{t-1} + \epsilon_t^{(1)}, \quad Y_t = X_t + \epsilon_t^{(2)}, \quad (\epsilon^{(1)}, \epsilon^{(2)}) \sim \mathcal{N}(0,1). \tag{13}$$

With the initial distribution $X_0 \sim \mathcal{N}(0,1)$. Converting to a state-space model, Equation (13) can be simply reformulated as,

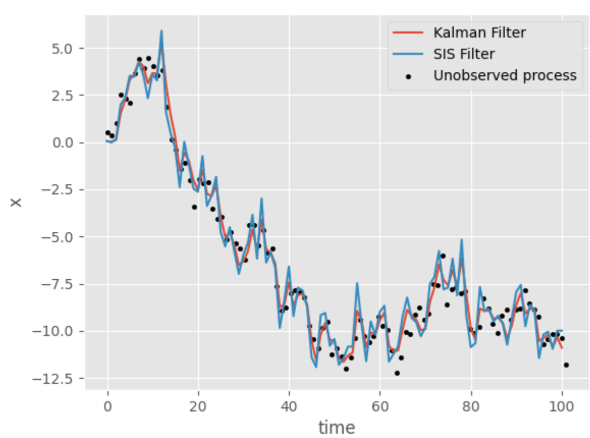
$$X_t|(X_{t-1} = x_{t-1}) \sim \mathcal{N}(X_{t-1}, 1) \quad (\textit{transition PDF}), \tag{14}$$

$$Y_t|(X_t = x_t) \sim \mathcal{N}(X_t, 1) \quad (\textit{observation PDF}). \tag{15}$$

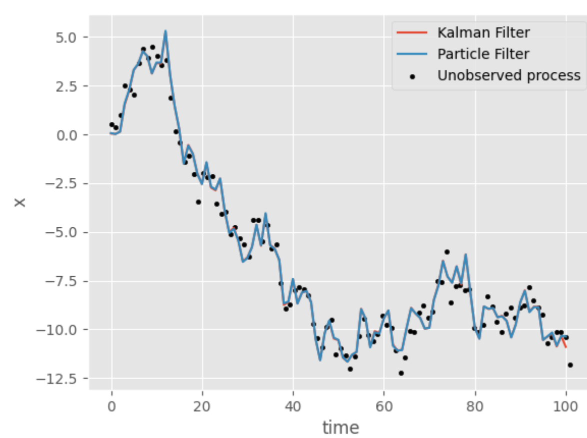
We now apply Algorithm 1 without the resampling stage (SIS) and then with resampling (Bootstrap Filter) using $N = 200$ particles. To compare to the Kalman Filter's exact results for the posterior mean, the estimated posterior mean - given in Equation (16) - is calculated at each time step:

$$\mathbb{E}_{p(x_t|\mathbf{y}_{1:t})}[x] = \sum_{i=1}^{N} w_t^{(i)} x_t^{(i)}. \tag{16}$$

Figure 2 gives the resulting trajectory of the estimated posterior means of SIS and the Bootstrap Filter. These estimates are both close to the true value, but the Particle Filter clearly performs better even in this one-dimensional setting, having an almost identical trajectory path to the Kalman Filter. For a slightly crude quantitative measurement of performance, the mean-squared error for SIS and the Bootstrap Filter is given by 0.386 and 0.009 respectively.



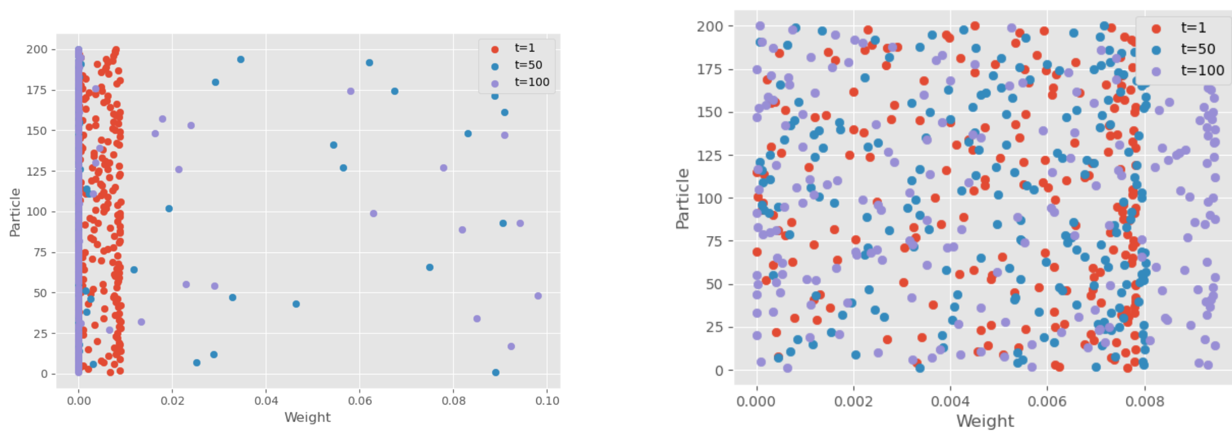
(a) SIS



(b) SIS with resampling (Bootstrap Filter)

Figure 2: Latent Gaussian process with 200 particles.

To see where SIS is failing, Figure 3 gives the weighted value of each particle at various iterations for the two methods. In Figure 3b the particle weights for the Particle Filter at three time intervals are well mixed. This is in contrast to the SIS particle weights, where the number of weights with significant values diminish over time and by $t = 100$ there are few weights greater than 0. This supports the theoretical discussion in Section 3 of SIS's shortcomings with regards to Weight Degeneracy.



(a) SIS Weights

(b) Bootstrap Filter Weights

Figure 3: Each dot gives the weight value on the x-axis, for each individual particle on the y-axis. Colours represent which time step this is.

4 Conclusions and further research

This paper introduced a popular method used to estimate posterior distributions for complex state-space models, the Particle Filter. Combining Sequential Importance Sampling with a resampling step, it allows for Bayesian inference on the estimation of posterior distributions, in particular estimating the filtering distribution and expected values. The main method reviewed was the Bootstrap Filter, which is easy to implement and belongs to a family of Particle Filters which all have attractive theoretical properties and advantages over alternative methods such as the Extended Kalman Filter and Sequential Importance Sampling.

Regarding applications, Particle Filters are also readily applied beyond the areas covered in this paper. Montemerlo et al. (2002) proposed a modified variant of the Particle Filter to localise and map a robots surroundings. While its usage in motion tracking has been implemented to scenarios such as the movements of football players (Dearden et al. 2006, Kataoka et al. 2011). The fields of Chemometrics[1] and mathematical psychology also utilise Particle Filters in a variety of problem settings, with papers by Oppenheim et al. (2008) and Speekenbrink (2016) delving into the applications to their respective areas.

For future investigations, attention can be turned to a problem closely related to filtering, smoothing. Here, one is interested in the state of the system at each point once all data has been made available. Some Particle Filtering resources also look into this problem (e.g. Doucet & Johansen 2009, Fearnhead & Künsch 2018). They find that while Particle Filter methods can be implemented for smoothing, they suffer significant performance issues for large time sequences. Investigating why this is, as well as the particle smoothing algorithms that have proposed thus far, would be a worthwhile endeavour for further research. Another avenue to investigate which concerns more recent advancements in the field, is Particle MCMC, which embeds a Particle Filter inside a MCMC algorithm, and can be used for problems such as Parameter Inference (Andrieu et al. 2010, Dahlin et al. 2014).

[1]The science of extracting information from chemical systems by data-driven means.

Data Availability

Code and plots used in this report can be accessed at the following GitHub link: https://github.com/BenSLowery/MResCode/tree/main/601/RT1.

References

Andrieu, C., Doucet, A. & Holenstein, R. (2010), 'Particle markov chain monte carlo methods', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**(3), 269–342.

Chopin, N. (2004), 'Central limit theorem for sequential monte carlo methods and its application to bayesian inference', *Annals of Statistics* **32**(6), 2385–2411.

Dahlin, J., Lindsten, F. & Schön, T. B. (2014), 'Second-order particle mcmc for bayesian parameter inference', *IFAC Proceedings Volumes* **47**(3), 8656–8661.

Dearden, A., Demiris, Y. & Grau, O. (2006), 'Tracking football player movement from a single moving camera using particle filters'.

Doucet, A., de Freitas, N. & Gordon, N. (2001), *An Introduction to Sequential Monte Carlo Methods*, Springer New York, New York, NY, pp. 3–14.

Doucet, A. & Johansen, A. (2009), 'A tutorial on particle filtering and smoothing: Fifteen years later', *Handbook of Nonlinear Filtering* **12**.

Fearnhead, P. & Künsch, H. R. (2018), 'Particle filters and data assimilation', *Annual Review of Statistics and Its Application* **5**(1), 421–449.

Gordon, N., Salmond, D. & Smith, A. (1993), 'Novel approach to nonlinear/non-gaussian bayesian state estimation', *IEE Proceedings F (Radar and Signal Processing)* **140**, 107–113(6).

Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Transactions of the ASME–Journal of Basic Engineering* **82**(Series D), 35–45.

Kataoka, H., Hashimoto, K. & Aoki, Y. (2011), Player position estimation by monocular camera for soccer video analysis, *in* 'SICE Annual Conference 2011', pp. 1985–1990.

Kitagawa, G. (1996), 'Monte carlo filter and smoother for non-gaussian nonlinear state space models', *Journal of computational and graphical statistics* **5**(1), 1–25.

Montemerlo, M., Thrun, S., Koller, D. & Wegbreit, B. (2002), Fastslam: A factored solution to the simultaneous localization and mapping problem.

Oppenheim, G., Philippe, A. & Rigal, J. (2008), 'The particle filters and their applications', *Chemometrics and Intelligent Laboratory Systems* pp. 87–93.

Robert, C. P. & Casella, G. (2004), Monte carlo statistical methods, *in* 'Springer Texts in Statistics'.

Speekenbrink, M. (2016), 'A tutorial on particle filters', *Journal of Mathematical Psychology* **73**, 140–152.

Terejanu, G. (2009), Extended kalman filter tutorial.