Sequential Monte Carlo Techniques and Bayesian Filtering : Applications in Tracking

Samarjit Das

Abstract—Visual tracking is a very important issue in various applications involving robotics and/or computer vision. The basic idea is to estimate a hidden state sequence from noisy observations under a given state space model. The state space model comprises of a system model and an observation model. The system models that we are going to discuss here has the markov property and the corresponding state space model is basically known as hidden markov model. Due to non-linearities in the system and the observation model, bayesian filtering (also known as particle filtering) is an ideal candidate for the purpose of tracking. Particle filtering is basically a non-linear, non-gaussian analog of kalman filter. It is based on sequential Monte Carlo techniques. In this article we are going to discuss Monte Carlo simulation techniques to solve numerical problems, especially to compute posterior probability distribution which involves complicated integrals. Then we are going to extend these ideas to sequential Monte Carlo techniques and develop algorithms leading to particle filtering. We demonstrate applications of particle filtering in areas such as target tracking in a wireless sensor network.

Index Terms—Monte Carlo simulation, sequential Monte Carlo techniques, particle filtering, visual tracking

1 Monte Carlo Methods

Monte Carlo methods (also referred to as Monte Carlo simulation) are a class of computational algorithms that rely on repeated random sampling to compute their results. These methods are quite often used to simulate physical and mathematical systems. They are most suitable for computation by computers as they rely on repeated computation and random or pseudo random numbers. Monte Carlo methods tend to be used when it is infeasible or impossible to compute an exact result with deterministic algorithm. A simple example is numerical computation of complicated integrals using Monte Carlo simulation.

Consider the following case. Say we want to compute,

$$I = \int_a^b \psi(x)dx$$

where $\psi(x)$ be a function with a very complicated closed form expression making it impossible to be integrated in a closed form. We can solve this problem numerically using Monte Carlo technique. First, consider a random variable X distributed over the interval $[a, b]$ with a probability density function (pdf) $f_X(x)$. The closed form expression of f_X is known and we can draw samples w.r.t. $f_X(.)$. We draw N (sufficiently large) samples from the pdf,

$$x^i \sim f_X(x), \quad \text{i=1, ..., N}$$
$$\hat{f}_X(x) \triangleq \sum_{i=1}^{N} \frac{1}{N}\delta(x - x^i)$$

here $\delta(.)$ denotes a dirac delta function. Thus $\hat{f}_X(x)$ is the discretized PMF (probability mass function) approximation of the continuous pdf f_X. Now the integral can be written as,

$$\begin{aligned} I &= \int_a^b \psi(x)dx \\ &= \int_a^b \frac{\psi(x)}{f_X(x)} f_X(x)dx \\ I &= E_{f_X}[\frac{\psi(X)}{f_X(X)}] \end{aligned}$$

where $E_{f_X}[.]$ denotes the expected value w.r.t the distribution f_X. Now, we can estimate the expectation and thus the integral using the discretized version of the pdf. Or, $\hat{I} \approx E_{\hat{f}_X}[\frac{\psi(X)}{f_X(X)}]$ which ca be computed as follows,

$$\begin{aligned} \hat{I} &= \int_a^b \frac{\psi(x)}{f_X(x)} (\sum_{i=1}^{N} \frac{1}{N}\delta(x - x^i))dx \\ \text{Or, } \hat{I} &= \frac{1}{N}\sum_{i=1}^{N} \frac{\psi(x^i)}{f_X(x^i)} \end{aligned} \quad (1)$$

Using the law of large numbers it can be shown that $\hat{I} \to I$ as $N \to \infty$. Thus by drawing a large number of sample in the interval $[a, b]$ we are able to numerically approximate the integral I using Monte Carlo technique. The process of sampling from the distribution f_X can be extended to *importance sampling* technique by which we can estimate the properties of a particular distribution , while only having samples from a different distribution (known as importance density or importance function) rather than the distribution of interest. This will be discussed in detail while developing the particle filtering algorithm.

• *This article was submitted as a term paper for the course ComS 577, Fall 2008 at Iowa State University, Ames, IA 50010.*

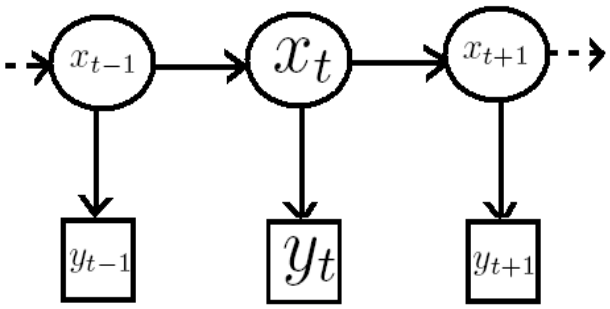

Fig. 1. This figure shows the hidden markov model structure of the state space. The state sequence x_t is a markov process and the observations y_t, $t = 1, 2, ...$ are conditionally independent given the state at t

2 Bayesian Filtering problem

Consider a state-space model with $t = 0, \ldots, T$,

$$x_t = \phi(x_{t-1}) + n_t \quad (2)$$
$$y_t = g(x_t) + v_t \quad (3)$$

where x_t and y_t denote the state and the observation at the current instant respectively. In real-life applications, for example, the state can be the position of a target while the observation is noisy the sensor data about the current position and our goal could be to extract the true 'hidden' state information using the observations and the state dynamical model (the system model, eq (2)). The functions $\phi(.)$ and $g(.)$ can be any linear or non-linear function. The following things can be assumed to be known. The initial state distribution $p(x_0)$, the state transition density $p(x_t|x_{t-1})$ and the observation likelihood $p(y_t|x_t)$. Here, $p(.)$ is used to denote probability density function. The transition density and the observation likelihood can be determined form the know distributions of i.i.d noise sequences n_t and v_t. The state space is assumed to follow a hidden markov model (HMM). In other words, the state sequence x_t is a markov process and the observations y_t, $t = 1, 2, ...$ are conditionally independent given the state at t. The states are hidden (i.e. not observable); all we can know about the system is through the observations known at each instant. The graphical model of the HMM is shown in Fig. 1. Under HMM assumption, $p(x_t|x_{t-1}, past) = p(x_t|x_{t-1})$ and $p(y_t|x_t, past) = p(y_t|x_t)$ e.g. $p(y_t|x_t, y_{t-1}) = p(y_t|x_t)$.

The goal of particle filtering (or bayesian filtering) is to estimate the joint posterior state distribution at time t i.e. $p(x_{1:t}|y_{1:t})$ or quite often its marginal $p(x_t|y_{1:t})$. Here, $x_{1:t} = \{x_1, x_2, ..., x_t\}$. Finally, we would like to compute

$$I_t = E_{p(x_t|y_{1:t})}(f(X_t)) = \int f(x_t)p(x_t|y_{1:t})dx_t \quad (4)$$

When the posterior can be assumed to be gaussian and function $\phi(.)$ and $g(.)$ to be linear the same problem can be recursively solved using Kalman filter [1]. To tackle non-linearity of ϕ and $g(.)$, the Extended Kalman filter [1] can be used, but it still assumes the gaussianity of the posterior distribution. But in many practical problems the posterior is highly non-gaussian (quite often multimodal) together with non-linear $\phi(.)$ and $g(.)$. Under such circumstances, sequential Monte Carlo technique based particle filtering algorithm gives us a way to solve this posterior estimation problem. In the next few sections, we shall develop the particle filter starting with Monte Carlo sampling for pdf approximation. We also discuss sequential Monte Carlo method (especially sequential importance sampling) and how it leads to particle filtering.

3 Derivation of the Particle Filter

Similar to eq (4), say we are trying to compute the expectation w.r.t the joint posterior distribution $p(x_{1:t}|y_{1:t})$

$$E_{p(x_{1:t}|y_{1:t})}[f(X_{1:t})] = \int f(x_{1:t})p(x_{1:t}|y_{1:t})dx_{1:t} = ? \quad (5)$$

In order to solve this problem, we have to look into two very important issues. a) Do we have a closed form expression for $p(x_{1:t}|y_{1:t})$? b) Can we sample from $p(x_{1:t}|y_{1:t})$? Depending on these two issues we can use three different methods two solve this problem namely, simple Monte Carlo sampling, importance sampling and bayesian importance sampling. The method involving bayesian importance is our main focus as it exactly is what bayesian filtering/particle does. These methods are discussed below.

3.1 Simple Monte Carlo Sampling

Consider the case when we can sample from the joint posterior distribution $p(x_{1:t}|y_{1:t})$. In that case we can approximate the distribution as a discretized version (as done in Sec. 1) in terms of the samples drawn from that distribution. Or,

$$x^i_{1:t} \sim p(x_{1:t}|y_{1:t}), \quad \text{i=1, } \ldots, \text{ N}$$
$$\hat{p}(x_{1:t}|y_{1:t}) \approx \sum_{i=1}^{N} \frac{1}{N}\delta(x_{1:t} - x^i_{1:t}) \quad (6)$$

Now, we can approximate the expectation in the equation (5) as follows,

$$E_{p(x_{1:t}|y_{1:t})}[f(X_{1:t})] \approx E_{\hat{p}(x_{1:t}|y_{1:t})}[f(X_{1:t})] = \frac{1}{N}\sum_{i=1}^{N} f(x^i_{1:t})$$

This is the most ideal case. In reality, almost always we cannot sample from the posterior but we might have a closed form expression for the distribution. This leads to the second method known as importance sampling.

3.2 Importance Sampling

Importance sampling can be used to estimate the properties of a particular distribution, while only having samples from a different distribution (known as importance density or importance function) rather than the distribution of interest. Thus when we have a closed form expression for $p(x_{1:t}|y_{1:t})$ but cannot sample from it, we use an importance density to draw the samples. We choose the importance density such that it has a convenient closed form expression and can easily draw samples from it. Say, this importance density be given as $\pi(x_{1:t}|y_{1:t})$. Now, the problem of posterior expectation computation can be solved as follows,

$$\begin{aligned} I_t &= \int f(x_{1:t})p(x_{1:t}|y_{1:t})dx_{1:t} \\ &= \int f(x_{1:t})\frac{p(x_{1:t}|y_{1:t})}{\pi(x_{1:t}|y_{1:t})}\pi(x_{1:t}|y_{1:t})dx_{1:t} \\ &= E_{\pi(x_{1:t}|y_{1:t})}[f(X_{1:t})\frac{p(X_{1:t}|Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}] \end{aligned}$$

Now, draw samples from $\pi(.)$ and get its discretized version $\hat{\pi}(.)$. Then the computation of posterior expectation boils down to,

$$\begin{aligned} x^i_{1:t} &\sim \pi(x_{1:t}|y_{1:t}), \text{ i=1, ..., N} \\ E_{p(x_{1:t}|y_{1:t})}[f(X_{1:t})] &\approx E_{\hat{\pi}(x_{1:t}|y_{1:t})}[f(X_{1:t})\frac{p(X_{1:t}|Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}] \\ &= \frac{1}{N}\sum_{i=1}^{N} f(x^i_{1:t})\frac{p(x^i_{1:t}|y_{1:t})}{\pi(x^i_{1:t}|y_{1:t})} \end{aligned} \quad (7)$$

The corresponding approximation to the joint posterior distribution is given as follows,

$$\begin{aligned} p(x_{1:t}|y_{1:t}) &\approx \hat{p}(x_{1:t}|y_{1:t}) \\ &= \sum_{i=1}^{N} w^i_t \delta(x_{1:t} - x^i_{1:t}) \text{ where} \\ w^i_t &= \frac{1}{N}\tilde{w}^i_t, \text{ with } \tilde{w}^i_t = \frac{p(x^i_{1:t}|y_{1:t})}{\pi(x^i_{1:t}|y_{1:t})} \end{aligned}$$

Thus we can see that even if we cannot sample from the joint posterior, still we can get around the problem by performing importance sampling from a conveniently chosen probability distribution $\pi(.)$.

3.3 Bayesian Importance Sampling

It turns out that in most real-life situations, we can neither sample from the posterior (i.e. $p(x_{1:t}|y_{1:t})$) nor we have any closed form expression of it. Thus computing the integral in equation (5) becomes challenging. Bayesian importance sampling is a variant of the standard importance sampling technique to tackle this problem. The numerical methods developed for solving this problem leads to sequential importance sampling (SIS) and then finally to particle/bayesian filtering. Now, under the problem statement we do not have a closed form expression for the posterior $p(x_{1:t}|y_{1:t})$. But it can be expressed in the following manner,

$$\begin{aligned} p(x_{1:t}|y_{1:t}) &= \frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})} \text{ thus} \\ p(x_{1:t}|y_{1:t}) &\propto p(x_{1:t}, y_{1:t}) \end{aligned} \quad (8)$$

It turns out that we can at least compute $p(x_{1:t}, y_{1:t})$ in closed form recursively. This follows from the hidden markov model (HMM) assumption on the state space. The recursion can be performed as $p(x_{1:t}, y_{1:t}) = p(x_{1:t-1}, y_{1:t-1})p(y_t|x_t)p(x_t|x_{t-1})$ with $p(y_t|x_t)$ and $p(x_t|x_{t-1})$ known. We shall utilize this fact to solve the problem of computing the posterior expectation in a recursive fashion i.e. starting with a known $p(x_0)$ and using $p(y_t|x_t)$, $p(x_t|x_{t-1})$, keep computing the approximate posterior $\hat{p}(x_{1:t}|y_{1:t})$ and $E_{p(x_{1:t}|y_{1:t})}[f(X_{1:t})]$ for $t > 0$.

The bayesian importance sampling is performed as follows. The posterior expectation $E_{p(x_{1:t}|y_{1:t})}[f(X_{1:t})]$ can be written as,

$$\begin{aligned} I_t &= \int f(x_{1:t})p(x_{1:t}|y_{1:t})dx_{1:t} \\ &= \int f(x_{1:t})\frac{p(x_{1:t}, y_{1:t})}{p(y_{1:t})}dx_{1:t} \\ &= \frac{\int f(x_{1:t})p(x_{1:t}, y_{1:t})dx_{1:t}}{p(y_{1:t})} \\ &= \frac{\int f(x_{1:t})p(x_{1:t}, y_{1:t})dx_{1:t}}{\int_{x_{1:t}} p(x_{1:t}, y_{1:t})dx_{1:t}} \end{aligned} \quad (9)$$

Now, let us consider the importance density to be $\pi(x_{1:t}|y_{1:t})$ from which we are going to draw samples. We choose $\pi(.)$ in such a way that we at least recursively know how to compute the expression for $\pi(x_{1:t}|y_{1:t})$. The importance density $\pi(.)$ is assumed to have certain properties which are crucial to the development of the recursive algorithm for approximating the posterior distribution and expectation. These will be discussed soon. Now, let us get back to the problem of computing I_t. From equation (9) it follows that

$$\begin{aligned} I_t &= \frac{\int f(x_{1:t})\frac{p(x_{1:t}, y_{1:t})}{\pi(x_{1:t}|y_{1:t})}\pi(x_{1:t}|y_{1:t})dx_{1:t}}{\int_{x_{1:t}} \frac{p(x_{1:t}, y_{1:t})}{\pi(x_{1:t}|y_{1:t})}\pi(x_{1:t}|y_{1:t})dx_{1:t}} \\ &= \frac{E_{\pi(.)}[f(X_{1:t})\frac{p(X_{1:t}, Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}]}{E_{\pi(.)}[\frac{p(X_{1:t}, Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}]} \end{aligned} \quad (10)$$

Now we can sample from $\pi(.)$ as, $x^i_{1:t} \sim \pi(x_{1:t}|y_{1:t})$, $i = 1, \ldots, N$ and then get its discretized version $\hat{\pi}(.)$. Finally, I_t can be estimated as follows,

$$\begin{aligned} \hat{I}_t &= \frac{E_{\hat{\pi}(.)}[f(X_{1:t})\frac{p(X_{1:t}, Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}]}{E_{\hat{\pi}(.)}[\frac{p(X_{1:t}, Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}]} \\ &= \frac{\frac{1}{N}\sum_{i=1}^{N} f(x^i_{1:t})\tilde{w}^i_t}{\frac{1}{N}\sum_{j=1}^{N} \tilde{w}^j_t}, \quad \tilde{w}^i_t = \frac{p(x^i_{1:t}, y_{1:t})}{\pi(x^i_{1:t}|y_{1:t})} \\ &= \sum_{i=1}^{N} f(x^i_{1:t})w^i_t, \text{ with } w^i_t = \frac{\tilde{w}^i_t}{\sum_{j=1}^{N} \tilde{w}^j_t} \end{aligned} \quad (11)$$

The corresponding approximation to the joint posterior distribution is given as,

$$\hat{p}(x_{1:t}|y_{1:t}) \triangleq \sum_{i=1}^{N} w_t^i \delta(x_{1:t} - x_{1:t}^i) \tag{12}$$

where $\{w_t^i\}_{\forall i}$ are called the normalized importance weights. Now, clearly w_t^i's cannot be computed directly at a given instant. So, we develop a recursive way to compute them. Once this is done, we have a computationally feasible method for computing the posterior distribution. In order to do that we first make sure the following assumption holds for the importance density $\pi(x_{1:t}|y_{1:t})$.

$$\pi(x_{1:t}|y_{1:t}) = \pi(x_{1:t-1}|y_{1:t-1})\pi(x_t|x_{1:t-1}, y_{1:t}) \tag{13}$$

Since $p(x_{1:t}, y_{1:t}) = p(x_{1:t-1}, y_{1:t-1})p(y_t|x_t)p(x_t|x_{t-1})$, we can develop a recursive way of computing the importance weights as,

$$\begin{aligned} \tilde{w}_t^i &= \frac{p(x_{1:t}^i, y_{1:t})}{\pi(x_{1:t}^i|y_{1:t})} \\ &= \tilde{w}_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{\pi(x_t^i|x_{1:t-1}^i, y_{1:t})} \text{ where,} \\ x_t^i &\sim \pi(x_t|x_{1:t-1}^i, y_{1:t}) \text{ and } x_{1:t}^i = [x_{1:t-1}^i, x_t^i] \end{aligned} \tag{14}$$

Thus we can recursively compute the estimates of the posterior distribution and the corresponding expectations starting with the initial distribution. The entire algorithm is summarized in the next section which is known as sequential importance sampling (SIS).

3.4 Sequential Importance Sampling (SIS)

Before we give the algorithm for SIS, it is important to choose the importance density $\pi(.)$. There could be many choices. The simplest one is to use the state transition density as the importance density [2]. Or,

$$\pi(x_t|x_{1:t-1}, y_{1:t}) = p(x_t|x_{t-1}) \tag{15}$$

$$\text{This gives, } \tilde{w}_t^i = \tilde{w}_{t-1}^i p(y_t|x_t^i) \tag{16}$$

The optimal importance density [3], [4] is the one which minimized the variance of the importance weights conditioned on the observations and previous state samples. It can be shown that $\pi_{opt}(.) = p(x_t|x_{t-1}, y_t)$ under the hidden markov model (HMM) assumption. The derivation of $\pi_{opt}(.)$ is beyond the scope of this article. All our discussions will stick to the case where transition density is used as the importance density (given in equation (15)). Finally, the recursive algorithm for estimating the posterior density is summarized below.

SIS Algorithm

1) Sample from the initial distribution. $x_0^i \sim p(x_0)$, assign $\tilde{w}_0^i = w_0^i = \frac{1}{N}$, $i = 1, ..., N$
2) For $t > 0$ and $i = 1, ..., N$,
   a) Sample $x_t^i \sim p(x_t|x_{t-1}^i)$ and $x_{1:t}^i = [x_{1:t-1}^i, x_t^i]$. Compute weights as, $\tilde{w}_t^i = \tilde{w}_{t-1}^i p(y_t|x_t^i)$
   b) Get the normalized importance weights w_t^i and finally, $\hat{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^N w_t^i \delta(x_{1:t} - x_{1:t}^i)$
3) Set $t + 1 \leftarrow t$ and go back to step 2.

The SIS algorithm gives us a way to recursively estimate the discretized posterior distribution. But this algorithm has one major problem which makes it almost ineffective for practical applications. It turns out that the variance of the importance weights conditioned on the observations only increases over time. As a result, after a few iterations, only a few samples (also called *particles*) will have non-zero normalized importance weights. This is known as degeneracy of weights and it ends up wasting a lot of particles making it very inefficient. The particle filter comes into picture to solve this problem. The basic particle filtering algorithm is discussed in the next section.

3.5 The Basic Particle Filter

The particle filter basically modifies the SIS algorithms to prevent the degeneracy of weights. In order to do that at each time step, the particles (i.e. the samples) are resampled w.r.t their normalized importance weights. In other words, $\{w_t^i\}_{i=1}^N$ is used as a probability mass function (PMF) to sample the existing particles again. It is denoted as $x_t^i \sim PMF[\{w_t\}]$ and the new importance weights are assigned as $w_t^i = \frac{1}{N}$. The basic particle filtering algorithm is summarized below.

Particle Filtering (PF) Algorithm

1) Sample from the initial distribution. $x_0^i \sim p(x_0)$, assign $\tilde{w}_0^i = w_0^i = \frac{1}{N}$, $i = 1, ..., N$
2) For $t > 0$,
   a) Sample $x_t^i \sim p(x_t|x_{t-1}^i)$ and Compute weights as, $\tilde{w}_t^i = p(y_t|x_t^i)$, $i = 1, ..., N$
   b) Get the normalized importance weights w_t^i as $w_t^i = \frac{\tilde{w}_t^i}{\sum_{j=1}^N \tilde{w}_t^j}$
   c) Resample the particles as, $x_t^i \sim PMF[\{w_t\}]$ and reassign $w_t^i = \frac{1}{N}$
   d) Get the estimated posterior as, $\hat{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^N \frac{1}{N} \delta(x_{1:t} - x_{1:t}^i)$ where $x_{1:t}^i = [x_{1:t-1}^i, x_t^i]$
   e) Finally, compute $I_t = E_{p(x_t|y_{1:t})}[f(X_t)] \approx E_{\hat{p}(x_t|y_{1:t})}[f(X_t)] = \frac{1}{N} \sum_{i=1}^N f(x_t^i)$
3) Set $t + 1 \leftarrow t$ and go back to step 2.

3.5.1 A Simple PF Example

Consider a very simple linear state space model,

$$\begin{aligned} x_t &= 0.5x_{t-1} + w_t, \;\; w_t \sim \mathcal{N}(0, \sigma_w^2) \\ y_t &= 0.4x_t + v_t, \;\; v_t \sim \mathcal{N}(0, \sigma_v^2) \end{aligned}$$

Given initial distribution $p(x_0) = \mathcal{N}(0, \sigma_0^2)$. Here $\mathcal{N}(\mu, \sigma^2)$ denotes a gaussian distribution with mean μ

and variance σ^2. The algorithm with $N = 1000$ particles is implemented as follows,

1) Sample $x_0^i \sim \mathcal{N}(0, \sigma_0^2)$, $i = 1, \ldots, N$
2) For $t > 0$,
   a) Sample $x_t^i \sim \mathcal{N}(0.5x_{t-1}^i, \sigma_w^2)$, $i = 1, \ldots, N$
   b) Compute weight : $\tilde{w}_t^i \propto e^{-\frac{(y_t - 0.4x_t^i)^2}{2\sigma_v^2}}$
   c) Compute the normalized importance weights $w_t^i = \frac{\tilde{w}_t^i}{\sum_{j=1}^N \tilde{w}_t^j}$
   d) Resample the particles as, $x_t^i \sim PMF[\{w_t\}]$ and reassign $w_t^i = \frac{1}{N}$, $x_{1:t}^i = [x_{1:t-1}^i, x_t^i]$
   e) Compute $\hat{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^N w_t^i \delta(x_{1:t} - x_{1:t}^i)$ and $I_t \approx E_{\hat{p}(x_t|y_{1:t})}[f(X_t)] = \frac{1}{N}\sum_{i=1}^N f(x_t^i)$
3) Set $t + 1 \leftarrow t$ and go back to step 2.

4 Applications : Target Tracking in Sensor Network

Consider a network coverage area of $200m \times 200m$ with 200 randomly distributed sensors. The track-length was of 60 times steps. Let $X_t = [x_t, y_t]$ be the target state (i.e the position of the target in the network) at time t with corresponding velocity $V_t = [v_{x,t}, v_{y,t}]$. The state dynamics equation used was,

$$\begin{aligned} x_t &= x_{t-1} + v_{x,t-1} \text{ and } y_t = y_{t-1} + v_{y,t-1} \\ v_{x,t} &= v_{x,t-1} + e_{1,t} \text{ and } v_{y,t} = v_{y,t-1} + e_{2,t} \end{aligned} \quad (17)$$

And the sensor model (i.e. the measurement/observation model) was,

$$z_t = tan^{-1}(\frac{x_t - x_s}{y_t - y_s}) + e_t,$$

where (x_s, y_s) is the position of the current sensor leader.

The system and observation noises used were as follows: $e_{i,t} \sim \mathcal{N}(0, 0.22)$, $i = 1, 2$ and $e_t \sim \mathcal{N}(0, 0.052)$ were assumed to be white, stationary and independent of each other. The ground truth (i.e. the original track) was generated by assuming the initial position of the target to be $[5, 5]$ and corresponding initial velocity to be $[2, 1.5]$. While implementing tracking we assumed that we have an estimate of the initial position of the target. The current sensor leader was chosen using the minimum distance criteria w.r.t the current target location. A total of 4000 particles were used. At each step, using the observations of the current sensor leader, we estimate the state of the target by averaging over the resampled particles. The results are shown in Fig. 2.

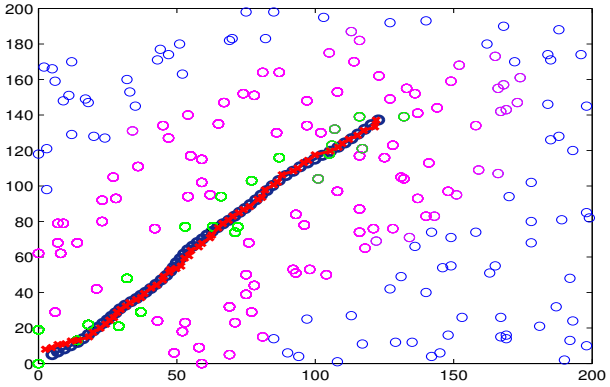

Fig. 2. This figure shows target tracking in a wireless sensor network. The blue path denotes the ground truth i.e. the true trajectory and the red $\times$ denote the tracked version. The blue, magenta and green circular patches are the sensors. The green color indicates a leader node while a magenta color indicates a neighboring node.

5 Conclusion

This article gives an overview of sequential Monte Carlo techniques leading to bayesian/particle filtering. We have shown how Monte Carlo techniques can be efficiently used to compute numerical solutions to problems which are often impossible to solve in closed form. We have derived the particle filtering algorithm starting with simple Monte Carlo sampling and then importance sampling followed by bayesian importance sampling. We have seen that particle filtering can be used in problems with non-linear state-space together with non-gaussian and multimodal posterior for which we do not have a closed form expression. Such algorithms can give us a discretized estimate of the posterior distribution in terms the particles (i.e. samples) and their corresponding weights. This also enables us to approximate the posterior expectation of any function (equation (5)). We have demonstrated the working of a basic PF algorithm for a simple scalar case. Finally, we give application of the particle filtering for target tracking in a wireless sensor network. Particle filters has numerous other applications in areas such as image processing and computer vision, robotics, finance and even for spacecraft trajectory tracking. In computer vision, PF has been successfully used to track deforming shape sequences leading to articulated body tracking. It has also found applications for deformable contour tracking, also know as condensation [5]. More about bayesian/particle filtering and its applications can be found in [2], [4], [3].

References

[1] G. Welch and G. Bishop, "An introduction to Kalman Filters," *SIGGRAPH*, 2001.
[2] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/nongaussian bayesian state estimation," *IEE Proceedings-F (Radar and Signal Processing)*, pp. 140(2):107–113, 1993.
[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.
[4] A. Doucet, "On sequential monte carlo sampling methods for bayesian filtering," in *Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering*, 1998.
[5] M. Isard and A. Blake, "Condensation: Conditional Density Propagation for Visual Tracking," *Intl. Journal of Comp. Vision*, pp. 5–28, 1998.