# 1 Particle filter

## 1.1 The basis of Particle filter

It has been well known that a large amount of realistic data analysis tasks consist of estimating unknown quantities from some given observations. Indeed, if the data are modeled as a linear Gaussian state-space model, it is possible to derive a desired analytical expression to compute the evolving sequence of posterior distributions by implementing Kalman filter, besides, it's still reachable to develop an analytical solution for the data are modeled as a partially observed, finite state-space Markov chain. However, real data can be very complicated, typically involving elements of non-Gaussianity, high dimensionality and nonlinearity, which conditions usually preclude analytic solution. To deal with these sophisticated problems, many of approximation schemes such as Grid-based filters and Gaussian sum approximations have been proposed to solve these problems. One of these proposals named Sequential Monte Carlo(SMC) methods(is equivalent to Partical filter) are a set of simulation-based methods which can be used to compute the posterior distributions. Compared to other schemes, SMC is more flexible, easy to implement, parallelisable and applicable in very general settings.

## 1.2 The implementation of particle filter

Particle filter can be implemented to solve a large number of practical application. A simple localization problem is one of the example. Localization is the process of determining the position of an object with various onboard sensors from which the aircraft can obtain the knowledge of its velocity and its altitude. A simple model of the aircraft motion is provided by a (discrete-time) integrator

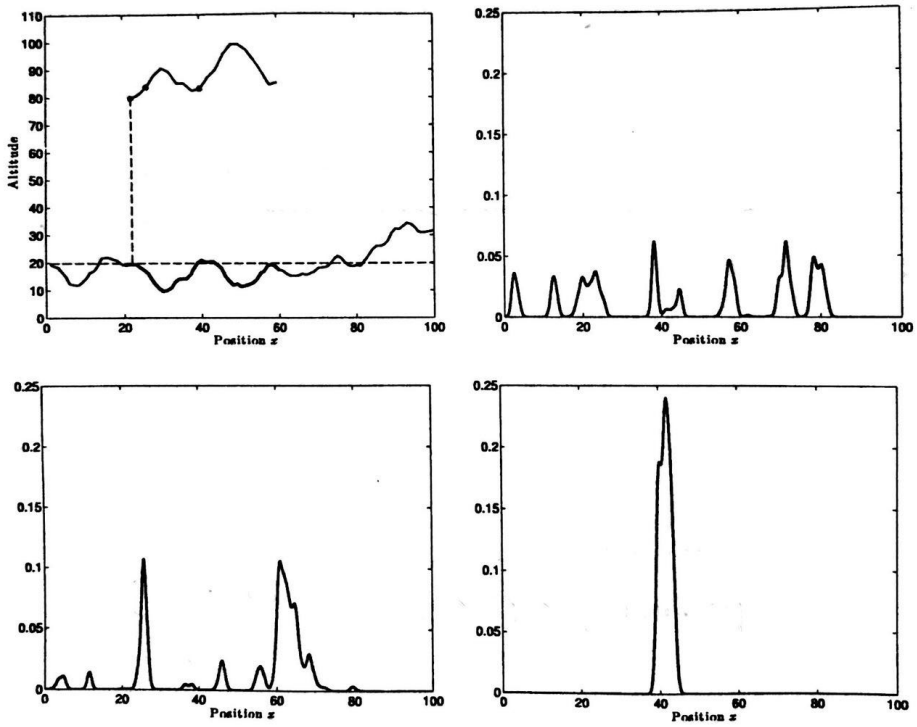
$$x_{t+1} = x_t + v_t + w_t, \qquad w_t \sim \mathcal{N}(0, 5) \tag{1}$$

where $x_t$ denotes the postion and $w_t$ denotes process noise. The velocity can only help in establishing a relative position, which means that additional information is needed to solve the localization problem. One solution is to make use of a map of terrain elevation and downward-facing radar, measuring the distance between the aircraft and the ground. The corrsponding measurement equation is

$$y_t = h(x_t) + e_t, \qquad e_t \sim \mathcal{N}(0, 1) \tag{2}$$

where $y_t$ denotes the distance over ground measured by the radar, $h()$ denotes a look-up in the map encoding the terrain elevation and $e_t$ denote the measurement noise. However, the measurement equation (2) is nonlinear

and the function $h()$ is only defined in discrete points(according to the resolution of the map). But it fits the Particle filter(PF) perfectly, since it can deal with nonlinear functions and a varying number of possible hypotheses.




The result of implementing the PF with N = 200 particles is shown in the above picture. At time 1, the figute(upper left) illustrates the situation that the aircraft is flying at an altitude of roughly 80 m at position $x_1 = 22$ m,where the terrain elevation is 20 m. Now, compare these positions to the PDF $\hat{p}^N(x_1|y_1)$ provided by the PF after the first measurement has been received and processed(upper right plot). As we can see from the upper right plot, the PF provides several possible dominating modes representing the estimated position. One dominating mode is located in $x = 22$, which is corresponding with the upper right plot. When more measurement and information have been received and processed, PF can estimate the more accurate position. The principle illustrated in this example of using a PF to solve the localization problem by combining the information from sensors and maps has been successfully used to solve many different localization problems, including for example underwater vessels, ships, cars and people moving around in both indoor and outdoor environments.

### 1.3 The algorithm of bootstrap Particle filters

Here, we introduce the bootstrap particle filters which is the simple representation of Particle filters. The nonlinear filtering problem amounts to computing the filtering PDFs $\{p(x_t|y_{1:t})\}$ Sequentially in time. One principle solution is implemented by Forward filtering

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})},$$

(3)

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})d_{x_{t-1}}, \tag{4}$$