Chapter 2

Bayesian State Estimation

Most of the localization, mapping and SLAM approaches have a probabilistic formulation. In this chapter, we revise the Bayesian state estimation framework, and in particular the *filtering problem*, in order to provide the reader with background on the used mathematical tools, and for introducing the notation.

The filtering problem can be expressed as estimating the state x of a dynamic discrete system (see Figure 2.1), given:

- the analytical knowledge of the **state transition function** f_t and the statistical knowledge of the **state noise** w_t,
- the analytical knowledge of the **output function** h_t and the statistical knowledge of the **observation noise** v_t,
- a realization of the system output $z_{1:t}$ up to time t.

A probabilistic filter for a dynamic system is a mathematical tool, whose goal is to estimate a distribution of the possible system state histories given the measurements:

$$p(x_{0:t} \mid z_{1:t}) \tag{2.1}$$

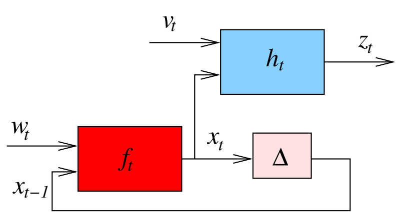

Figure 2.1: Generic System Model. The Δ block is a one step delay.

Here $x_{0:t}$ are the system states from the instant 0 to the instant t, while $z_{1:t}$ is the history of measurements z from 1 to t.

In most of the cases, one is interested in evaluating only the marginal distribution of the current state given the observations:

$$p(x_t|z_{1:t}) \tag{2.2}$$

instead of the full state history.

Several on line and off line techniques for solving the filtering problem have been proposed [35; 21; 1; 54]. Most of them rely on the assumption that the process being observed is Markovian. A process is Markovian if the current measurement is independent from the past ones, given the current state: $p(z_t|z_{1:t-1}, x_t) = p(z_t|x_t)$.

In the context of the SLAM problem, for the Markov assumption to hold, no moving objects unknown to the robot can populate the environment. This imposes obvious restrictions to the application domains. However, in moderately dynamic environments most of the techniques proposed in this section have shown to work. In case of big violations of the Markov assumption, a typical approach consists in pre-processing the filter input in order to skip the sensor readings generated by dynamic objects.

In the rest of this chapter we describe a wide range of useful tools for Bayes filtering, which is a framework that can be used for estimating the state of Markovian systems. In particular we will describe:

- The Kalman Filter (KF), which is an exact, closed form filter working with linear systems, affected by zero mean Gaussian noise.
- The Particle Filters (PF), which are Monte Carlo methods suitable for the state estimation of non linear non Gaussian dynamic systems.

2.1 Bayes Framework

Now we shortly present the key filtering equations, and formalize the problem. Let $p(z|x)$ be the *observation model* , that is the density of the measurement z, given that the system state is x, and let $p(x_t|x_{t-1})$ be the *evolution model*[1]. If the Markov assumption holds, the posterior of the state chain up to time t

[1] In the following sections, the evolution model is also referred to as *motion model*, since it is used for describing the change of the robot state after motion.

is

$$\begin{aligned} p(x_{0:t}|z_{1:t}) &= \frac{p(z_t|x_{0:t}, z_{1:t})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad \text{[by Markov assumption]} \\ &= \frac{p(z_t|x_t)p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|x_t)p(x_t|x_{t-1})}{p(z_t|z_{1:t-1})} p(x_{0:t-1}|z_{1:t-1}) \end{aligned} \tag{2.3}$$

If one is interested in estimating the current state distribution, the filtering equation becomes the following:

$$\begin{aligned} p(x_t|z_{1:t}) &= \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1}}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_t}{\int p(z_t|z_{1:t-1}, x_t)p(x_t|z_{1:t-1})dx_t} \\ &= \eta p(z_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \end{aligned} \tag{2.4}$$

here η is a normalization factor ensuring that Eq. 2.4 correctly represents a probability distribution.

Usually, the evaluation of Eq. 2.4 is done in two steps: **prediction** and **update**. In the prediction step, the result of the state transition from x_{t-1} to x_t is computed. In the update step, the last observation z_t is incorporated in the previously computed probability density. Referring to Eq. 2.4, one can argue that the predict step consists in computing the integral term. The update step is performed by weighting the predicted belief $\int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1}$ with the last observation likelihood $p(z_t|x_t)$. These two steps can be found in all of the filters described in the reminder of this chapter.

Bayes filtering in this form is exact and can be used on any kind of system for which the Markov assumption holds. Unfortunately, in the above equations there are some integrations over the state space. In many cases the state space is high dimensional, and the Bayes filtering cannot be directly implemented. For instance, in the SLAM problem the dimension of the system state is the sum of the robot location dimension and the map space dimension, which can easily be in the order of the hundreds or thousands. A straightforward evaluation of Eq. 2.4 would require an integration over the entire state space. For this reason approximated techniques are needed.

2.2 Kalman Filter

The Kalman Filter (KF) [35; 72] is an exact filter that can be derived directly by Equation 2.4 under the assumptions that the system is linear and the noise is Gaussian. Under these linearity hypotheses the system can be described by

$$\begin{aligned} x_t &= F_t x_{t-1} + w_t \\ z_t &= H_t x_t + v_t \end{aligned}$$

The system noise $w_t \sim \mathcal{N}(0, \Sigma_{w_t})$ and the observation noise $v_t \sim \mathcal{N}(0, \Sigma_{v_t})$ are zero mean normally distributed. The key advantage of the Kalman Filter is that it represents the distributions in closed form, in terms of means and covariance matrix. The update of the Kalman filter can be carried out in the time of a matrix multiplication ($O(n^3)$, where n is the state dimension).

The iterative algorithm of the filter is the following:

- predict:
$$x'_t = F_t x_{t-1} \qquad \Sigma'_t = F_t \Sigma_{t-1} F_t^T + \Sigma_{w_t}$$
- update:
$$K_t = \Sigma'_t H_t \left(H_t \Sigma'_t H_t^T + \Sigma_{v_t} \right)^{-1}$$
$$x_t = x'_t + K_t \left(z_t - H_t x'_t \right) \qquad \Sigma_t = \left(I - K_t H_t \right) \Sigma'_t$$

Unfortunately, in the mobile robot domain, the evolution model, as well as the observation model are non linear, thus the noise cannot be considered Gaussian. However, for mild evolution laws, a non linear extension can be used: the Extended Kalman Filter (EKF) [72], in which local linearizations of the state transition function f and the observation model h are performed. The extended Kalman filter algorithm can be expressed as

- predict:
$$x'_t = f_t(x_t) \qquad \Sigma'_t = F_t \Sigma_{t-1} F_t^T + \Sigma_{w_t}$$
- update:
$$K_t = \Sigma'_t H_t \left(H_t \Sigma'_t H_t^T + \Sigma_{v_t} \right)^{-1}$$
$$x_t = x'_t + K_t \left(z_t - h_t(x'_t) \right) \qquad \Sigma_t = \left(I - K_t H_t \right) \Sigma'_t$$

here $F_t = \nabla_x f_t|_{x_t}$ and $H_t = \nabla_x h_t|_{x_t}$.

The key limitations in the use of extended Kalman filter lies in the strong assumptions that have to be done on the estimated system, namely: Gaussian noise, and linearizability. In most of the robotic systems used for localization and SLAM, the uncertainty is not expressible, nor approximable as a Gaussian distribution, being multi modal and irregularly shaped. When more modes are present in a distribution, dealing with multiple hypotheses is needed, while the Kalman Filter works on their mean. In such a situations, its use is prone to failure. Moreover, the linearization of the system can introduce some systematic error in the estimate. Finally, some systems cannot be linearized (being their 1st order derivatives null), thus the extended Kalman Filter cannot be applied. In these contexts, a second order extension to the Kalman filter: the Unscented Kalman Filter (UKF) has been proposed in [70]. While the UKF in general behaves better than the Kalman filter the hypotheses of Gaussian noise is still required to hold.

Despite the above outlined limitations, the Kalman Filter is one of the most used tools in localization and SLAM, due to its simplicity. Moreover, when the underlying hypotheses hold, it exhibits a strong convergence rate if compared with other filtering techniques.

2.3 Particle Filters

A particle filter is a Bayes filter that works by representing a probability distribution as a set of samples (particles):

$$p(x) \simeq \frac{1}{N} \sum_i \delta_{x^{(i)}}(x). \tag{2.5}$$

where $\delta_{x^{(i)}}(x)$ is the impulse function centered in $x^{(i)}$. The denser are the samples $x^{(i)}$ in a region, the higher is the probability that the current state falls within that region.

In principle, in order to maintain a sampled representation of the feasible system state histories, one should draw the samples $\{x_{0:t}^{(i)}\}$ form the probability distribution $p(x_{0:t} \mid z_{1:t})$ of the current state given the observation history. Such a distribution is in general not available in a form suitable for sampling. However, the **Importance Sampling** (IS) principle ensures that if one can:

- evaluate point wise and draw samples from an arbitrarily chosen importance function $\pi(x_{0:t} \mid z_{1:t})$, such that $p(x_{0:t} \mid z_{1:t}) > 0 \Rightarrow \pi(x_{0:t} \mid z_{1:t}) > 0$, and
- evaluate point wise $p(x_{0:t} \mid z_{1:t})$,

then it is possible to recover a sampled approximation of $p(x_{0:t} \mid z_{1:t})$ as

$$\hat{p}(x_{0:t} \mid z_{1:t}) \propto \sum_i w^{(i)} \delta_{x_{0:t}^{(i)}}(x_{0:t}). \tag{2.6}$$

Here $\{x_{0:t}^{(i)}\}$ are samples drawn from $\pi(x_{0:t} \mid z_{1:t})$ and $w_t^{(i)} = \frac{p(x_{0:t}^{(i)} \mid z_{1:t})}{\pi(x_{0:t}^{(i)} \mid z_{1:t})}$ is the importance weight related to the ith sample that takes into account the mismatch among the target distribution $p(x_t \mid z_{1:t})$ and the importance function.

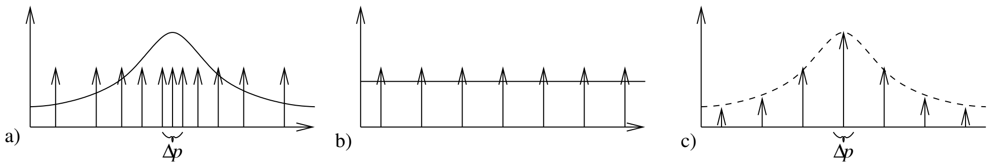

Figure 2.2: The Importance Sampling principle. In picture a) the goal distribution and some samples drawn from it are depicted. The more dense are the samples in a region, the higher is the probability density in that region. If we do not have a closed form for the goal distribution we are not able to draw samples from it. However, we can draw samples from another distribution, for instance the uniform, shown in picture b). Picture c) shows the samples, weighted according to the Importance Sampling. We compute the probability mass falling in an interval Δp by summing the weights of the samples falling in the interval. If using a high number of samples the probability mass computed from a) and c) are similar, becoming equal as the number of samples goes to infinity.

Observe that, in case we are able to draw samples from the target distribution, such that $p(x_{0:t}^{(i)} \mid z_{1:t}) \propto \pi(x_{0:t}^{(i)} \mid z_{1:t})$ then all of the weights are the same, and the variance of $w^{(i)}$ is 0. An intuitive explanation of how the importance sampling principle works is given in Figure 2.2.

Sequential Importance Sampling By restricting to the set of Markovian systems, and in particular focusing the choice on a particular class of importance functions, such that

$$\pi(x_{0:t} \mid z_{1:t}) = \pi(x_t \mid x_{0:t-1}, z_{1:t})\pi(x_{0:t-1} \mid z_{1:t-1}) \tag{2.7}$$

it is possible to recursively computing the importance weights, without revising the past generated trajectories, since

$$\begin{aligned} w_t^{(i)} &= \frac{p(x_{0:t}^{(i)} \mid z_{1:t})}{\pi(x_{0:t}^{(i)} \mid z_{1:t})} \\ &= \eta \frac{p(z_t \mid x_t^{(i)})p(x_t^{(i)} \mid x_{t-1}^{(i)})}{\pi(x_t^{(i)} \mid x_{0:t-1}^{(i)}, z_{1:t})} \frac{p(x_{0:t-1}^{(i)} \mid z_{1:t-1})}{\pi(x_{0:t-1}^{(i)} \mid z_{1:t-1})} \\ &\propto \frac{p(z_t \mid x_t^{(i)})p(x_t^{(i)} \mid x_{t-1}^{(i)})}{\pi(x_t^{(i)} \mid x_{0:t-1}^{(i)}, z_{1:t})} w_{t-1}^{(i)} \end{aligned} \tag{2.8}$$

Where $\eta = 1/p(z_t \mid x_{0:t-1}^{(i)}, z_{1:t-1})$ is a normalization factor. Several approaches select the importance function to be the transition model $p(x_t|x_{t-1})$. Accordinging to the importance sampling principle the weights $w_t^{(i)}$ can be computed as follows:

$$\begin{aligned} w_t^{(i)} &= \frac{p(x_t^{(i)} \mid z_{1:t})}{p(x_t^{(i)} \mid x_{t-1}^{(i)})} = \\ &= \xi \cdot \frac{p(z_t \mid x_t^{(i)})p(x_t^{(i)} \mid x_{t-1}^{(i)})p(x_{t-1}^{(i)} \mid z_{1:t-1})}{p(x_t^{(i)} \mid x_{t-1}^{(i)})} \\ &\propto w_{t-1}^{(i)} p(z_t \mid x_t^{(i)}). \end{aligned} \tag{2.9}$$

Sampling Importance Resampling The direct use of a Sequential Importance Sampling filter requires a huge number of samples, since as the system evolves all of the particles but one will have a high weight. For this reason the Sampling Importance Resampling (SIR) filter [21] has been introduced.

A SIR filter, sequentially processes the observations z_t and the proprioceptive sensings u_t as they are available, by updating a set of samples representing the estimated distribution $p(x_{0:t} \mid z_{1:t}, u_{0:t})$. This is done by performing the following three steps:

1. *Sampling*: The next generation of particles $\{x_t^{(i)}\}$ is obtained by the previous generation $\{x_{t-1}^{(i)}\}$, by sampling from a proposal distribution $\pi(x_t \mid x_{0:t-1}^{(i)}, z_{1:t}, u_{0:t})$.

2. *Importance Weighting*: An individual importance weight $w^{(i)}$ is assigned to each particle, according to the IS principle

$$w^{(i)} = w_{t-1}^{(i)} \frac{p(x_t^{(i)} \mid x_{0:t-1}^{(i)}, z_{1:t})}{\pi(x_t^{(i)} \mid x_{0:t-1}^{(i)}, z_{1:t})}. \tag{2.10}$$

   The weights $w^{(i)}$ account for the fact that the proposal distribution π in general is not equal to the true distribution of successor states.

3. *Resampling*: Particles with a low importance weight w are typically replaced by samples with a high weight. This step is necessary since only a finite number of particles are used to approximate a continuous distribution. Furthermore, resampling allows to apply a particle filter in situations in which the true distribution differs from the proposal one.

Please note that a SIR filter here described, assumes the proposal to be suitable for sequential estimation. This means that $\pi(\cdot)$ satisfies Eq. 2.7.

In principle, one would evaluate point wise and sample from the target distribution $p(x_t \mid z_{1:t}, u_{0:t})$, but this is hard, due to the following reasons:

- it is usually not available in closed form and,
- it depends on the whole input history $z_{1:t}$ and $u_{0:t}$, up to time t.

In the next of this section, we discuss a crucial problem affecting the particle filters: particle depletion. Such a problem can be detected by the analysis of some indicators of the filter behavior, while it can be bounded by a proper choice of the proposal distribution.

Particle Depletion While the resampling step is needed for concentrating the computational effort of the filter in state space regions having a high likelihood, it introduces additional problems. This problem becomes evident when using an uninformed proposal distribution, while the observations are affected by a small noise. Most of the samples generated in the uninformed sampling step will be rewarded with a low weight in the importance weighting phase, and they are likely to be suppressed by resampling. In some degenerated situation, after the resampling step only one particle is retained. This behavior can lead the filter to converge to the wrong solution, since the multimodal posterior is not adequately covered by the samples. This problem is known of as particle depletion [70].

Two are the techniques for lessening particle depletion: sampling from a proposal which is closer to the target distribution, and adding artificial noise to the observation model. The first solution is structural, since the choice of a better proposal distribution makes the importance weights value to be similar for all of the particles, in fact limiting the particle suppression in the resampling stage. The second solution can lead to a working filter, but the introduction of artificial noise decreases the accuracy of the estimate with respect to the one achievable by using the original observation model. Moreover, when using a better proposal distribution particle depletion can be lessened by reducing the number of resampling actions.

Some alternated filtering schemes have been proposed, in order to lessen particle depletion, like the auxiliary particle filter By Pitt and Shephard [54], however the improvements achievable using these techniques are orthogonal to the selection of an improved proposal.

Number of Effective Samples Liu [39] introduced the so-called effective number of particles N_{eff} to estimate how well the current particle set represents the true posterior. This quantity is computed as

$$N_{\mathit{eff}} = \frac{1}{\sum_{i=1}^{N} \left(w^{(i)}\right)^2}. \tag{2.11}$$

The intuition behind N_{eff} is as follows. If the samples were drawn from the true posterior, the importance weights of the samples would be equal to each other, due to the importance sampling principle. The worse is the approximation the higher is the variance of the importance weights. Since N_{eff} can be regarded as a measure of the dispersion of the importance weights, it is a useful measure to evaluate how well the particle set approximates the true posterior.

Choice of the optimal proposal distribution The optimal sequential importance function has been introduced in [39]:

$$\pi(x_t|z_{1:t}) = p(x_t|x_{t-1}^{(i)}, z_t). \tag{2.12}$$

The optimality has to be intended as minimizing the variance of the importance weights.

If drawing from the optimal proposal distribution, the importance weight $w^{(i)}$ for each particle i is computed according to Eq. (2.10):

$$\begin{aligned} w_t^{(i)} &= \frac{p(x_{0:t}^{(i)} \mid z_{1:t})}{\pi(x_{0:t}^{(i)} \mid z_{1:t})} = \qquad (2.13) \\ &= \frac{p(x_{0:t}^{(i)} \mid z_{1:t})}{p(x_t^{(i)} \mid x_{t-1}^{(i)}, z_t)} = \text{[using Eq.2.8]} \\ &\propto \frac{\cdot p(z_t \mid x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)})}{\frac{p(z_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{\underbrace{\int p(z_t \mid x_t) p(x_t|x_{t-1}^{(i)}) dx_t}_{p(z_t|x_{t-1}^{(i)})}}} w_{t-1}^{(i)} \\ &= w_{t-1}^{(i)} p(z_t \mid x_{t-1}^{(i)}). \qquad (2.14) \end{aligned}$$

Reducing the Sampling Space Dimension Due to the extensive representation of the probability distribution used by the particle filters, the number of particles required for a good approximation of a filtering distribution, grows exponentially with the dimension of the state to estimate. In this case, however, it is possible to address the problem by Rao-Blackwellization [11]. This technique can be applied when the dynamic Bayesian network of the system has a particular structure [12]. It consists in partitioning the state space X in two subspaces X^a and X^b. The partition has to be made according to the structure of the system, ensuring that, given a state $x_t \in X$, its projection $x_t^a \in X^a$ depends only on the previous state x_{t-1}, the current command u_t and the current observation z_t. The x projection $x_t^b \in X^b$ should be updated analytically and efficiently once x_t^a is known. This allows to sample only from X^a, in fact decreasing the effective sampling space dimension.