AA274A: Principles of Robot Autonomy I
Course Notes

Previous chapters introduced some of the fundamental concepts related to robotic perception, and specifically techniques for sensing the environment and extracting useful semantic information. While these techniques provide *local* information that is crucial for robots to navigate autonomously, additional *global* information is often required. For example, distance measurements from a laser rangefinder might be useful for detecting objects in an environment, but they only provide information *relative* to the robot's current position. Alternatively, object detection via computer vision only provides information about what is in the robot's *current* view. Robotic autonomy, in particular autonomous decision making and planning, generally requires more than just local information to answer questions such as "have I seen this object before?" and "have I been here before?". These new challenges, associated with building a global understanding of the environment from local measurements, are often referred to as *localization and mapping*[1].

14 Introduction to Robot Localization

The problem of *localization* is to endow the robot with the ability to understand its current position with respect to its environment in a global sense. One of the main classes of techniques for robot localization are *map-based*, where the robot explicitly localizes its position with respect to a *map* of the environment. For example, consider the floor plan (the environment map) in Figure 1: before a robot can navigate to a particular room it must know where in the building it is currently located.[2]

There are two primary components to map-based localization: map representation and belief representation. This chapter focuses on belief representation, which addresses the problem of how to best represent the robot's belief of its current position with respect to the map. One simple approach would be to simply store a best guess of the robot's position (in some map-based coordinate system). However in practice localization information is often *uncertain*, and representing the belief by only a best guess does not capture this important

[1] Localization and mapping is the component of the "think" part of the "see, think, act" cycle that connects with robotic perception.

[2] Other approaches to navigate in environments include *behavioral* approaches, which rely on a specified set of behaviors that will result in a desired global behavior without the need for explicit mapping or localization. An example of this approach would be to have a left-wall following behavior for movement about a building.

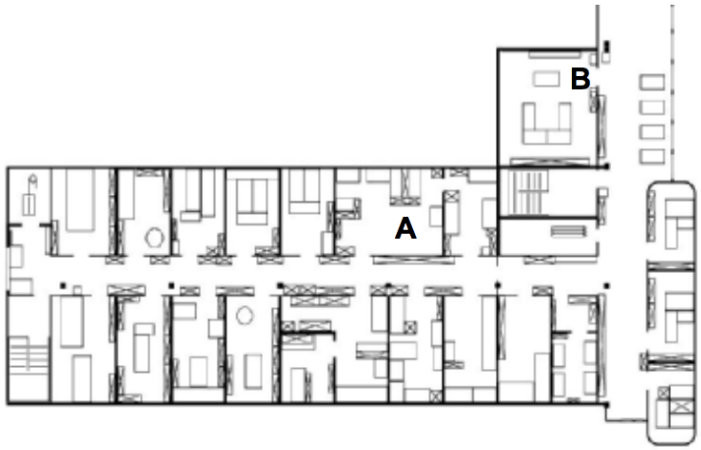

Figure 1: An example environment where localization is crucial for robotic autonomy [SNS11]. For a robot to move from location A to location B it must first understand which room it is in, and that the only path to B is through the hallways. Extracting such global information about the environment from local measurements (e.g. from a range sensor) requires specialized algorithms.

fact. Therefore one common approach is to use a *probabilistic* representation of the robot's belief since probability distributions can be used to model uncertainty (and extract best guesses, for example by finding the mean of a unimodal distribution). A variety of probabilistic representations can be used, for example singe-hypothesis or multiple-hypothesis representations as well as continuous or discrete representations. A few examples showing the differences between these types of representations are given in Figure 2. Some representations are more expressive than others, but there is usually a trade-off with computational complexity of the resulting algorithms that support the representation. Algorithms based on these different probabilistic representations will be presented in this chapter and in subsequent chapters.

14.1 Basic Concepts in Probability

Before discussing different types of robot localization algorithms it is useful to provide a review of some of the fundamental concepts from probability.

14.1.1 Random Variables

Uncertain quantities such as sensor measurements, robot state, and environment variables can be modeled as discrete or continuous *random variables*.

Definition 14.1 (Discrete Random Variable)**.** *A discrete random variable X is a random variable that can only take on values from a countable set. Discrete random variables are*

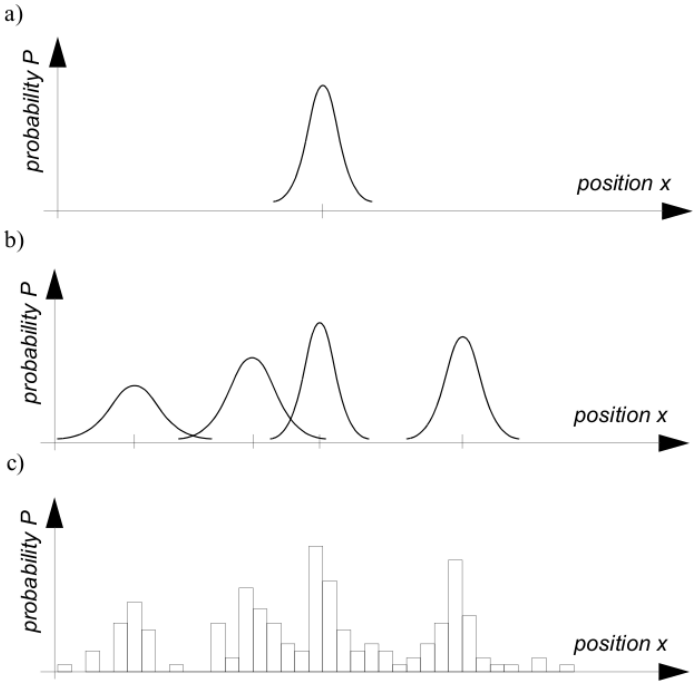

Figure 2: A graphical representation of different types of probabilistic representations: (a) a continuous single-hypothesis belief (e.g. from a single Gaussian distribution), (b) a continuous multiple-hypothesis belief (e.g. a mixture of Gaussians), (c) discrete representation with a finite number of possible values [SNS11].

characterized by a probability mass function $p(x)$ (which can be read as $p(X = x)$, "the probability that X takes on value x") that satisfies:

$$\sum_x p(x) = 1,$$

where the summation is over all possible values of X.

Definition 14.2 (Continuous Random Variable)**.** *A continuous random variable X is a random variable that can take on values from a continuous range. Continuous random variables are characterized by a probability density function $p(x)$ that satisfies:*

$$\int_{-\infty}^{\infty} p(x)dx = 1.$$

The probability of the random variable taking on a value in the interval $[a, b]$ is similarly defined as:

$$P(a \leq X \leq b) = \int_a^b p(x)dx = 1.$$

A common example of a discrete random variable is the result of a coin flip, which can only take on two values: heads or tails. In robotics, a common example of a continuous random variable may be the position of the robot, which could take on an infinite number of values.

14.1.2 Joint Distributions, Independence, and Conditioning

Many applications of probability theory rely on more than one random variable. In these instances it is useful to be able to quantify probabilities associated with multiple random variables at the same time. One of the most fundamental tools when dealing with multiple variables is the *joint distribution*.

Definition 14.3 (Joint Distribution)**.** *The joint distribution of two random variables X and Y defines the probability associated with both taking on specific values at the same time. This is denoted mathematically as $p(x, y)$, which can be read as $p(X = x \text{ and } Y = y)$.*

It is also useful to determine whether different random variables have any relationship to each other. In particular, two random variables that do not have any influence on each other in a probabilistic sense are considered to be probabilistically independent.

Definition 14.4 (Independence)**.** *Two random variables X and Y are independent if and only if:*

$$p(x, y) = p(x)p(y). \tag{1}$$

Independence holds when the occurrence of one value of a random variable does not affect the probability of another random variable taking on a specific value.

Another useful tool that relates two random variables is the conditional probability, which defines the probability of a random variable when the value of a second random variable is *known* or *fixed*.

Definition 14.5 (Conditional Probability)**.** *The conditional probability of a random variable X taking on a value given that a second random variable Y has a specific value is defined as:*

$$p(x \mid y) \coloneqq \frac{p(x, y)}{p(y)}. \tag{2}$$

This can be read as "the probability of X taking on value x conditioned on the fact that Y has taken on value y".

Notice that if the random variables X and Y are independent, then the conditional probability definition simplifies to $p(x \mid y) = p(x)$, which suggests that knowing that Y has taken on value y has provided no new information about the random variable X (which of course is in line with the definition of independence). Additionally, another notion of independence can be defined based on whether or not two random variables are independent when *conditioned* a third random variable.

Definition 14.6 (Conditional Independence)**.** *Two random variables X and Y are conditionally independent given the value of a third random variable Z if and only if:*

$$p(x, y \mid z) = p(x \mid z)p(y \mid z). \tag{3}$$

It is important to note however that conditional independence does not imply independence, and vice versa.

14.1.3 Law of Total Probability

The law of total probability defines a relationship between probabilities, joint probabilities, and conditional probabilities.

Definition 14.7 (Law of Total Probability)**.** *For discrete random variables X and Y the law of total probability states that:*

$$p(x) = \sum_y p(x, y) = \sum_y p(x \mid y)p(y).$$

Similarly, for continuous random variables this law is given by:

$$p(x) = \int p(x, y)dy = \int p(x \mid y)p(y)dy.$$

In words, this law says that the probability of a random variable X taking on a value x can be found by looking at the joint probabilities between X and Y and accounting for *all* possible values of Y. The second part of the law is a direct result of applying the definition of conditional probabilities.

14.1.4 Bayes' Rule

The joint probability $p(x, y)$ between two random variables X and Y can be related to the conditional probabilities $p(x \mid y)$ and $p(y \mid x)$ via the definition of conditional probabilities (2). In particular, since the joint probability can be equivalently expressed in two ways it can be seen that:

$$p(x, y) = p(x \mid y)p(y) = p(y \mid x)p(x).$$

This relationship is commonly referred to as Bayes' rule:

Definition 14.8 (Bayes' Rule)**.** *For discrete random variables X and Y, Bayes' rule states that:*

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}. \tag{4}$$

Bayes' rule is useful as it provides a relationship between the "inverse" conditional probabilities $p(x \mid y)$ and $p(y \mid x)$. This is particularly important for *probabilistic inference*, which is the problem of inferring the value of a random variable from another.

For example, suppose you had a good initial guess of the probability distribution $p(x)$ for a random variable X (the distribution $p(x)$ in this case is often called the *prior*, because it is the guess that comes before any new information is taken into account). Then, suppose some new information regarding the value of a random variable Y is obtained. Using Bayes' rule it is possible to update your belief about the probability distribution of X based on this new information. In particular, the new belief is the conditional probability $p(x \mid y)$ (which is often called the *posterior* because it comes after new information is introduced). These two distributions are related by Bayes' rule!

Bayes' rule can also be extended to cases with additional random variables. For example with three random variables X, Y, and Z, Bayes' rule is:

$$p(x \mid y, z) = \frac{p(y \mid x, z)p(x \mid z)}{p(y \mid z)}.$$

14.1.5 Expectation and Covariance

Probability distributions define in a vary precise way the probability associated with any particular value of a random variable. However, sometimes it is useful to aggregate this information into more practically useful metrics. Two of the most commonly used metrics are the *expected value* and the *covariance*.

Definition 14.9 (Expected Value)**.** *The expected value for a random variable X is denoted as $E[X]$. For discrete random variables the expected value can be computed by:*

$$E[X] = \sum_x xp(x).$$

Similarly, the expected value for a continuous random variable can be computed by:

$$E[X] = \int xp(x)dx.$$

The expected value can be thought of as the average result of an experiment over an infinite number of trials, and is also sometimes referred to as the *first moment* of the distribution. Additionally, expectation is a *linear operator*, such that:

$$E[aX + b] = aE[X] + b,$$

for any values a and b. In the case that the random variable X is a vector-valued random variable, the expectation of the random vector is simply the vector of expectations of each element.

Definition 14.10 (Covariance)**.** *The covariance between two random variables X and Y is denoted $cov(x, y)$ and is computed by:*

$$cov(x, y) = E[(X - E[X])(Y - E[Y])^T] = E[XY^T] - E[X]E[Y]^T$$

Covariance is a metric used to describe the relationship between random variables and is positive if greater values of one variable generally corresponds to greater values of the other (and same for lesser values). Similarly, it is negative if the variables tend to show opposite behavior of each other. If there is no general relationship between the two then their covariance is zero (e.g. independent random variables have zero covariance).

14.2 Markov Models

Recall from previous chapters the kinematic and dynamic models that were developed to describe the physical behavior of a robot. These models consisted of a robot state $\boldsymbol{x}$, and a set of equations that described how $\boldsymbol{x}$ varied in time given some control inputs $\boldsymbol{u}$. In this section another type of model will be developed that is based on these same core ideas. These new models, referred to as *Markov models*, are commonly used in robotics for localization tasks as well as higher level planning tasks.

14.2.1 States, Measurements, and Controls

Similar to previous chapters, the state $\boldsymbol{x}$ is a collection of variables that contains information required to define the physical state of the robot. However unlike previous chapters, the state might also include information about the environment (this state has a higher-level perspective). In the context of robotics, the state may include the robot pose (i.e. location and orientation information), velocity, as well as locations and features of surrounding objects in the environment. Note that in general the state discussed in this section might be different from the state defined for robot kinematics and dynamics (even if the robot is the same). This is because the choice of model is usually specific to the task at hand, and while the kinematic and dynamic models are useful for control, they may not strictly be necessary (or sufficient) for use in localization and planning tasks.

A discrete time formulation is also used in this context, where the state is specified for discrete time instances and denoted by $\boldsymbol{x}_t$ (rather than $\boldsymbol{x}(t)$, as was done in previous chapters). The models developed in this section then describe the changes in the state between time steps, for example between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. It is also useful to define the notation $\boldsymbol{x}_{t_1:t_n} \coloneqq \boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}, ..., \boldsymbol{x}_{t_n}$ for describing a sequence of states between times t_1 and t_n.

The robot interacts with the environment through control actions and by gathering information through measurements[3]. In this context, the measurement data collected at a time t will be denoted as $\boldsymbol{z}_t$, and the control data is denoted as $\boldsymbol{u}_t$. Similar to the state, a useful notation for representing a sequence of measurements or controls is given by $\boldsymbol{z}_{t_1:t_n} \coloneqq \boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, ..., \boldsymbol{z}_{t_n}$ and $\boldsymbol{u}_{t_1:t_n} \coloneqq \boldsymbol{u}_{t_1}, \boldsymbol{u}_{t_2}, ..., \boldsymbol{u}_{t_n}$. In general, the measurements can come from any number of the sensors discussed in previous sections on robotic perception, including cameras and laser rangefinders.

[3]In the context of robot localization, measurements increase the robot's knowledge and control actions tend to result in a loss of knowledge.

14.2.2 Model

The kinematic and dynamic models from previous chapters (expressed as a set of ordinary differential equations) were deterministic models. However, to leverage a probabilistic framework for robot localization it is typically required that the model also be probabilistic. In the most general sense a probabilistic model can be defined by:

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}), \tag{5}$$

which defines a probability distribution over the possible current state $\boldsymbol{x}_t$ given the state, measurement, and control histories. Note that here the convention that will be used is that the robot executes control $\boldsymbol{u}_t$ first, and then the measurement $\boldsymbol{z}_t$ can be made based on the resulting state $\boldsymbol{x}_t$. A general probabilistic measurement model can also be defined as:

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_{0:t}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}). \tag{6}$$

In many cases however, the state is defined such that it is *complete*. A state $\boldsymbol{x}_t$ is complete if no variables prior to $\boldsymbol{x}_t$ can influence the future states. In other words, $\boldsymbol{x}$ contains a sufficient amount of information that the history is not important. This is also known as the *Markov property*. If the Markov property holds, the probabilistic model (5) can be simplified to:

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{u}_t), \tag{7}$$

and the measurement model (6) can be simplified to:

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_t). \tag{8}$$

The resulting overall model with the Markov property, consisting of the state transition probability (7) and the measurement model (7) is referred to as a Bayes network model or a hidden Markov model. Graphically this model can be represented as shown in Figure 3, where the sequencing of the control and measurements are more clearly shown (first control, then measurement).

14.3 Bayes Filter

Given a Bayes network model defined by a state transition model (7) and a measurement model (8), the next task is to determine a way to use this information for robot localization. In particular, the desired task is to estimate the current robot state $\boldsymbol{x}_t$ given the measurement and control information that is available. In the probabilistic framework this estimate is referred to as a *belief distribution*, which is a probability distribution over $\boldsymbol{x}$. This distribution assigns a probability to each hypothesis with respect to the true state. Mathematically the belief distribution is denoted as $bel(\boldsymbol{x}_t)$ and is defined as:

$$bel(\boldsymbol{x}_t) := p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}). \tag{9}$$

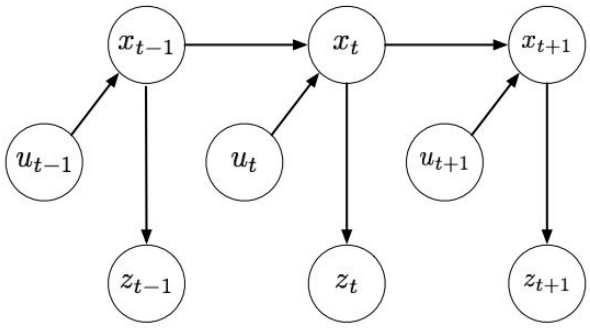

Figure 3: Graphical representation of the Bayes network model (hidden Markov model). Note that the sequencing assumes that the control is applied, and then a measurement is taken.

In other words, the belief $bel(\boldsymbol{x}_t)$ is a posterior probability distribution over the state variables conditioned on the available data. A similar distribution, known as the *prediction* distribution, can also be defined as:

$$\overline{bel}(\boldsymbol{x}_t) := p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}), \tag{10}$$

which is does not include the most recent measurement $\boldsymbol{z}_t$. The process of computing a belief from a predicted belief (i.e. the process of accounting for the new measurement $\boldsymbol{z}_t$) is called a *correction* or *measurement update*.

The most general algorithm for computing beliefs $bel(\boldsymbol{x}_t)$ (which leverages Bayes network models that satisfy the Markov property) is known as the *Bayes filter*. This filter is a recursive algorithm that consists of a prediction step for computing $\overline{bel}(\boldsymbol{x}_t)$ and a correction step for computing $bel(\boldsymbol{x}_t)$ given a new measurement $\boldsymbol{z}_t$.

14.3.1 Algorithm

The Bayes filter algorithm is given in Algorithm 1. In this algorithm, the probability associated with each potential state $\boldsymbol{x}_t$ is updated via a prediction and a correction. The term η in the correction step is simply a normalization constant that ensures the resulting posterior $bel(\boldsymbol{x}_t)$ satisfies the requirements of a probability density function[4]. This algorithm is typically initialized with a prior distribution $bel(\boldsymbol{x}_0)$ that may come from a best guess or simply a uniform distribution. Note that the prediction step is essentially just using the state transition model (7) to guess what might happen to each state for the given control $\boldsymbol{u}_t$. The correction step is then modifying the prediction to actually account for what was observed in the real world.

[4]In fact this normalization constant comes from the denominator in Bayes' rule.

Data: $bel(\boldsymbol{x}_{t-1}), \boldsymbol{u}_t, \boldsymbol{z}_t$
Result: $bel(\boldsymbol{x}_t)$
foreach $\boldsymbol{x}_t$ **do**
  $\overline{bel}(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{x}_{t-1}) bel(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1}$
  $bel(\boldsymbol{x}_t) = \eta p(\boldsymbol{z}_t \mid \boldsymbol{x}_t) \overline{bel}(\boldsymbol{x}_t)$
end
return $p_{k,t}$

Algorithm 1: Bayes Filter Algorithm

14.3.2 Derivation

Recall that the belief distribution is defined as (9), which can be expanded using Bayes' rule to yield:

$$\begin{aligned} bel(\boldsymbol{x}_t) &\coloneqq p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}), \\ &= \eta p(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}), \end{aligned}$$

where

$$\eta = \frac{1}{p(\boldsymbol{z}_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})}.$$

The Markov property can then be leveraged to simplify $p(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{z}_t \mid \boldsymbol{x}_t)$ and the definition of the prediction belief can be used to give:

$$bel(\boldsymbol{x}_t) = \eta p(\boldsymbol{z}_t \mid \boldsymbol{x}_t) \overline{bel}(\boldsymbol{x}_t),$$

which is precisely the second step of the Bayes filter algorithm. Now the derivation of the prediction can be given by again starting from its definition and leveraging the law of total probability:

$$\begin{aligned} \overline{bel}(\boldsymbol{x}_t) &= p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}), \\ &= \int p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) p(\boldsymbol{x}_{t-1} \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) d\boldsymbol{x}_{t-1}. \end{aligned}$$

Again the Markov property can now be used to simplify $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$, and additionally the structure of the model makes it possible to remove the $\boldsymbol{u}_t$ term from the prior distribution $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})$ since the control $\boldsymbol{u}_t$ has no impact on the state $\boldsymbol{x}_{t-1}$ (see Figure 3). Therefore the expression above can be simplified to:

$$\overline{bel}(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{u}_t) bel(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1},$$

since by definition $bel(\boldsymbol{x}_{t-1}) = p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t-1})$. This result is precisely the prediction step from the Bayes filter algorithm.

14.3.3 Practical Considerations

The Bayes filter is a great starting point to derive many useful algorithms, but is itself often not practical to implement. In particular it is generally not reasonable to assume that the integrals in Algorithm 1 can be computed, and if they could be approximated via a numerical scheme this may computationally still be challenging.

14.4 Discrete Bayes Filter

The discrete Bayes filter is a discrete version of the Bayes filter previously introduced. This filter can be applied to problems where the state space is finite (i.e. only a finite number of values of $\boldsymbol{x}$ are possible). This makes the Bayes filter approach more tractable because the integrals do not need to be computed over an infinite set.

In the discrete Bayes filter the belief $bel(\boldsymbol{x}_t)$ is represented using a probability mass function rather than a probability density function (as is the case with the *continuous* Bayes filter). In particular, this probability mass function is simply a finite collection of probabilities $\{p_{k,t}\}$ where $p_{k,t}$ is the probability associated with state k at timestep t. The algorithm generally follows the exact procedure as the Bayes filter in Algorithm 1, but with summations replacing the integrals. In particular, the discrete Bayes filter algorithm is provided in Algorithm 2.

Data: $\{p_{k,t-1}\}, \boldsymbol{u}_t, \boldsymbol{z}_t$
Result: $\{p_{k,t}\}$
foreach k **do**
  $\overline{p}_{k,t} = \sum_i p(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{x}_i) p_{i,t-1}$
  $p_{k,t} = \eta p(\boldsymbol{z}_t \mid \boldsymbol{x}_k) \overline{p}_{k,t}$
end
return $p_{k,t}$

Algorithm 2: Discrete Bayes Filter Algorithm

References

[SNS11] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT press, 2011.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.