

Lecture 10: Object Localization and Detection

Scribes:

10.1 Introduction

We finished covering the material for Homework 2 and we are moving on to the third topic of this class, which is localization. The goal of localization is to use information gathered from the sensing and perception modules in order to determine where the robot is with respect to the world.

Localization incorporates the use of probabilistic methods, as there will always be uncertainty in a robot's position. Therefore, lecture 10 will cover an introduction to probability, including belief representation and Bayes' rule. We will use these topics to introduce the Bayes' filtering algorithm, a theoretically sound but potentially impractical algorithm that uses the state transition model and sensor measurements to generate the belief state.

10.2 Localization

10.2.1 Behavioral Approach

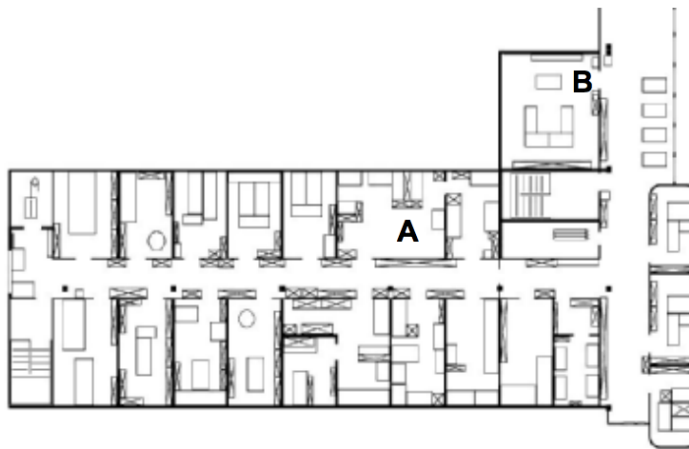


Figure 10.1: Siegwart Figure 5.6. An environment with rooms A and B

Behavioral approach attempts to design a set of behaviors that result in desired robot motion and trajectory. This method avoids mapping the robot's environment for localization. For example, consider Fig. 10.1, where a robot wants to navigate from room A to room B. Using behavioral approach, a robot may find its way to room B by following the left wall, incorporating other procedures such as collision

avoidance and identification of room B. However, compared to map-based approach, this method of navigation has difficulty scaling to different problems or environments. Depending on situations, techniques like following the left wall can also be suboptimal.

10.2.2 Map-based Approach

Map-based localization uses information about the robot's surrounding to identify its position *with respect to* the environment. To gain knowledge of its environment, the robot must actively gather and process sensor data of the surrounding. The most popular sensor is LIDAR, which can provide 360° coverage of the robot's surrounding. This approach can scale to any kind of environment, because a robot can use same techniques and sensors to map its surrounding.

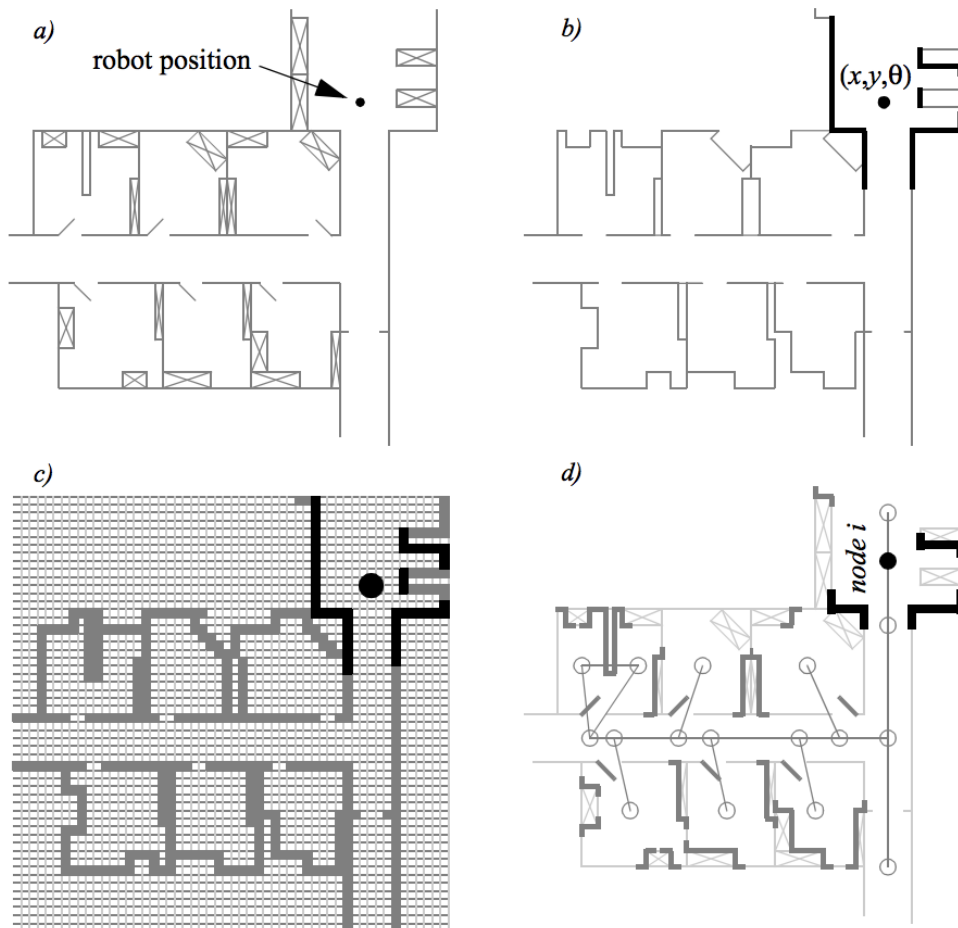


Figure 10.2: Siegwart Figure 5.10. (a) Real map. (b) Map where continuous lines represent walls. (c) Same map discretized into 3000 grid cells. (d) Topological map with nodes and edges.

10.2.3 Belief representation

Due to sensor noise, a robot cannot localize itself with absolute certainty. To compensate for the error, we can represent the estimate of robot's location as a probability distribution across the environment. There are different ways of representation. For example, the belief of robot's position can be *continuous*, which can be expressed by continuous variables (x, y, θ) Fig. 10.2(b).

On the other hand, the belief can be *discrete*. For instance, the map of the environment may be discretized into $N \times N$ cells as in Fig. 10.2(c), and the belief can assign probabilities of being in different cells. The other possible discrete representation is a topological map in Fig. 10.2(d), which maps the environment using nodes and edges. We can then assign probabilities of robot being in certain nodes.

The belief representation can also be classified as either *single-hypothesis belief* or *multiple-hypothesis belief*. *Single-hypothesis belief*, as in Fig. 10.3(a), is useful when describing a unique estimate of a robot's position. On the other hand, *multiple-hypothesis belief* in Fig. 10.3(b, c, d) can be used to describe multiple possible locations of a robot.

Continuous belief representation typically incorporates single-hypothesis belief (ex. state vector $[x, y, \theta]$) that is compact and easy to compute. However, discrete belief representation typically incorporates multiple-hypothesis belief. Because discrete method assigns the probabilities of robot's presence across different cells or nodes in the map, we see that the accuracy of estimate is limited by the resolution of discretization. It may also involve heavy computation, because a robot must keep track of probabilities associated each cell [2].

10.3 Probability Basics

10.3.1 Random Variable

Quantities such as sensor measurements, states of a robot and its environment are modeled as random variables. A random variable is a variable whose possible values are outcomes of a random phenomenon. In the case of sensor measurements, the source of randomness could be sensor noise. Random variables fall under one of two categories: discrete or continuous.

10.3.1.1 Discrete Random Variable

Discrete random variables can take on only a countable number of values. For example, if we flip a coin twice and the random variable X represents the number of coin flips resulting in "heads", X can only take on values of 0, 1 or 2 (and similarly for the case of "tails"). A discrete random variable is characterized by a probability mass function (PMF) $p(X = x)$ (or $p(x)$) where

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

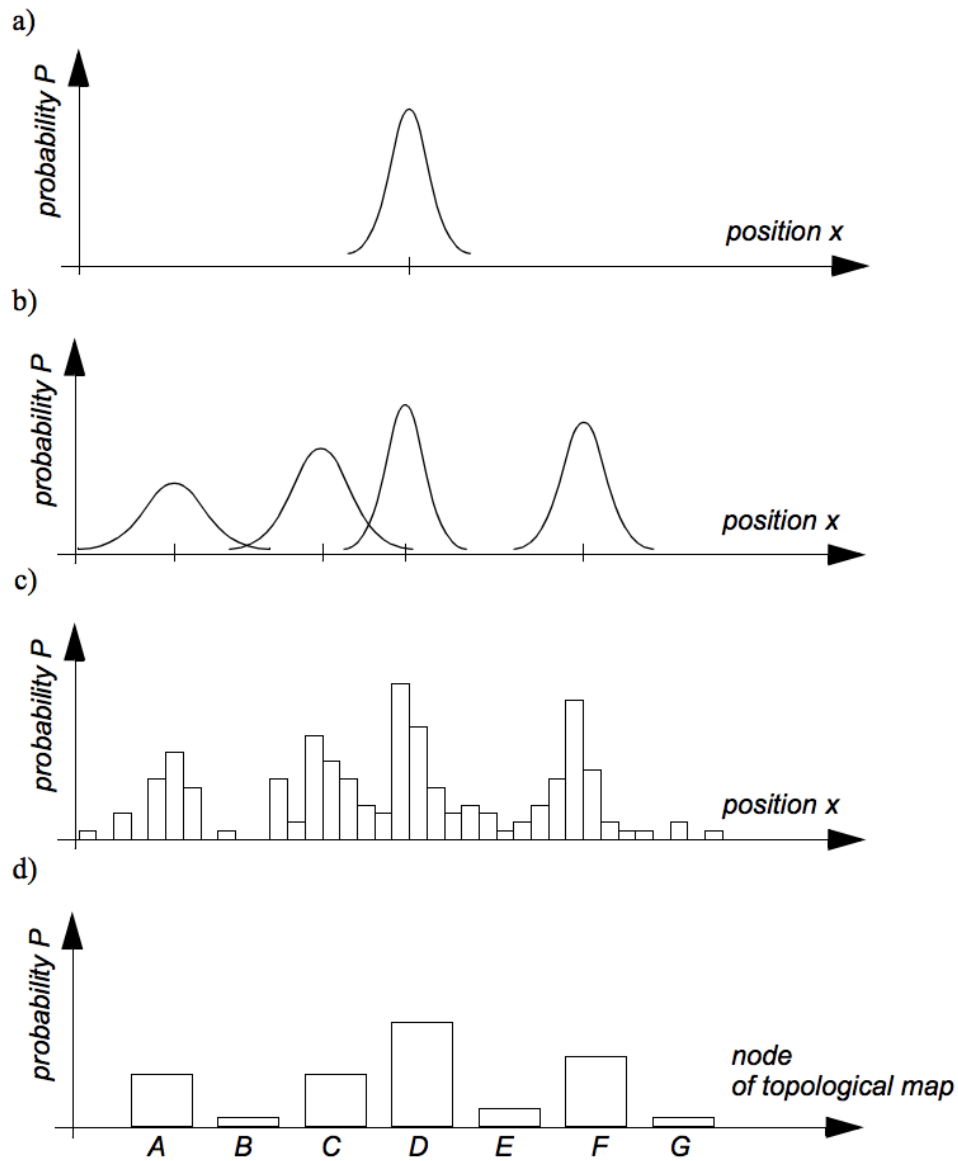


Figure 10.3: Siegwart Figure 5.9. (a) Continuous single-hypothesis belief. (b) Continuous multiple-hypothesis belief. (c) Discrete probabilities for grid-based map. (d) Discrete probabilities for topological map.

10.3.1.2 Continuous Random Variable

In contrast with a discrete random variable, a continuous random variable can take on an infinite number of values. A continuous random variable is characterized by a probability density function (PDF), $p(x)$, where the probability of a random variable being contained within the interval $[a, b]$ is

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

where

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

10.3.2 Joint Distributions, Independence and Conditional Probability

10.3.2.1 Joint Distributions

The joint distribution of two random variables X and Y is denoted as

$$p(x, y) := p(X = x \text{ and } Y = y).$$

10.3.2.2 Independence

If X and Y are random variables such that,

$$p(x, y) = p(x)p(y)$$

then X and Y are said to be independent.

10.3.2.3 Conditional Probability

Conditional probability is the probability of an event happening given that another event has occurred. That is, suppose that we know that the event $Y = y$ occurs with probability $p(y) > 0$. The probability of the event $X = x$ occurring given that $Y = y$ has occurred, or alternatively stated as the conditional probability of X given Y , is given by

$$p(x|y) := \frac{p(x, y)}{p(y)}.$$

Thus if X and Y are independent, $p(x|y) = p(x)$. Also, note that the above definition of conditional probability is indeed a definition and not theoretically derived.

In order to better understand the concept of conditional probability, consider Fig. 10.5. In this figure, the sample space Ω contains the set of all possible outcomes for a random variable. Consider the case where there are two possible outcomes, $B1$ and $B2$ which may occur with an equal probability of 0.5. These events correspond to the orange and blue regions of Ω , respectively. An event A that occurs with probability 0.5 can also be defined, which corresponds to the centered shaded region in Fig. 10.5.

Suppose that we wish to determine the probability of A occurring and that a sample from Ω is observed to result in the event $B1$. Given that $B1$ has occurred, we restrict our attention to the region $B1$. In this restricted region, the area corresponding to A occurring is the intersection of A and $B1$. Since region A overlaps with half of the region $B1$, we have the result that $p(A|B1) = 0.5$. A slightly different but equivalent interpretation of this figure, as given during lecture, is that we rescale the probability of event A occurring by the probability of the event that has happened. Mathematically, this is expressed as

$$p(A|B1) = \frac{p(A \cap B1)}{p(B1)} = \frac{0.25}{0.5} = 0.5.$$

Finally, note that if X and Y are independent $p(x|y) := p(x)$.

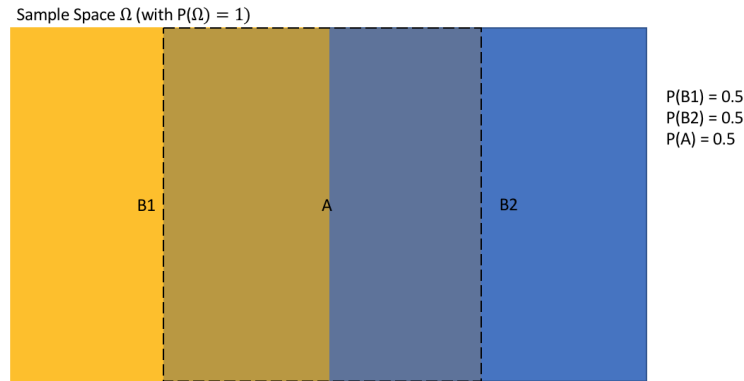


Figure 10.4: Sample space representation. Note that region A is centered and overlaps B1 and B2 equally.

10.3.2.4 Conditional Independence

If X , Y and Z are random variables such that,

$$p(x, y | z) = p(x | z)p(y | z)$$

then X and Y are said to be conditionally independent given Z .

However:

$$p(x, y | z) = p(x | z)p(y | z) \not\Rightarrow p(x, y) = p(x)p(y)$$

Nor:

$$p(x, y) = p(x)p(y) \not\Rightarrow p(x, y | z) = p(x | z)p(y | z)$$

10.3.3 Theorem of Total Probability

Once again consider Fig. 10.5. If we wished to find the probability of A without any assumptions regarding conditioning, we can do so by calculating

$$p(A) = p(A \cap B1) + p(A \cap B2).$$

This idea can be generalized for discrete random variables into what is known as the theorem (or law) of total probability as

$$p(x) = \sum_y p(x, y) = \sum_y p(x|y)p(y)$$

where the definition of conditional probability was used to obtain the last expression. For continuous random variables, the summations simply turn into integrals:

$$p(x) = \int p(x, y)dy = \int p(x|y)p(y)dy$$

10.3.4 Bayes' Rule

To facilitate the explanation of Bayes' Rule please note: a sample space is a set containing all the possible outcomes of an experiment and an event is a subset of elements in a sample space. Imagine that there is an event A whose set of outcomes is contained in the sets of events B_i where $i = 0, 1, \dots, n$ of a sample space. For example in Fig. 10.5 event A 's set is contained in the sets of events B_1 and B_2 . Then,

$$p(B_i|A) = \frac{p(B_i \cap A)}{p(A)}.$$

Using the formula for conditional probability $p(B_i \cap A)$ and $p(A)$ can be expressed as:

$$\begin{aligned} p(B_i \cap A) &= p(A|B_i)p(B_i). \\ p(A) &= \sum_{i=0}^n p(B_i \cap A) = \sum_{i=0}^n p(A|B_i)p(B_i). \end{aligned}$$

Combining these two expressions to derive Bayes' Rule for discrete sample spaces:

$$p(B_i|A) = \frac{p(A|B_i)p(B_i)}{\sum_{i=0}^n p(A|B_i)p(B_i)}.$$

For a continuous sample space Bayes' Rule is:

$$p(B|A) = \frac{p(A|B)p(B)}{\int p(A|B=b')p(B=b')db'}.$$

Note that $p(A)$ is independent of which event B_i is selected. So the denominator of Bayes' formula can be considered a constant for all events B_i with event A . Furthermore, by knowing the probabilities $p(A|B_i)$ and $p(B_i)$, we can determine $p(B_i|A)$. As an example, a brewery buys equal amounts of hops from two farmers, B_1 and B_2 . The brewery has a strict policy of using only one type of hops in each batch of beer it brews. Furthermore, the brewery uses half of all its hops from both sources to produce Pilsner beer. Bayes' Rule can be applied to determine, given that a batch of Pilsner is produced, what is the probability that the batch was brewed using hops from farmer B_1 . Note this example uses the sample space described in Fig. 10.5

$$p(B_1|A) = \frac{p(A|B_1)p(B_1)}{\sum_{i=0}^1 p(A|B_i)p(B_i)} = \frac{p(A|B_1)p(B_1)}{p(A|B_1)p(B_1) + p(A|B_2)p(B_2)} = \frac{0.5 \cdot 0.5}{(0.5 \cdot 0.5 + 0.5 \cdot 0.5)} = \frac{0.25}{0.5} = 0.5.$$

Bayes' Rule can be extended with the use of additional random variables. In the three variable case, given $X = x$, $Y = y$ Bayes' Rule can be conditioned on an additional variable $Z = z$ as follows:

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

This is the probability that event $Z = z$, and event $X = x|Y = y$ occur, which describes $Z = z$ influence on joint event $X = x|Y = y$.

10.3.5 Expectation and Covariance

The expectation value of a random variable $E(X)$ is the average result of an experiment over an infinite number of trials. It is also known as the first moment of the distribution. For the discrete case:

$$E(X) = \sum_{i=0}^n x_i p(X = x_i).$$

For the continuous case:

$$E(X) = \int_{-\infty}^{\infty} x' p(X = x') dx'.$$

The expectation value of a constant is a constant and its calculation is a linear operation.

$$E(aX + b) = aE(X) + b.$$

In the case of a vector of random variables. The expectation of the vector is the vector of each variables expectation value.

Covariance is a calculation used to measure the relationship between random variables. For two random variable vectors $X = x, Y = y$, the covariance is:

$$\text{cov}(X, Y) = E[(X - E(X))(Y - E(Y))] = E(XY^T) - E(X)E(Y^T)$$

If the covariance is positive this implies that X tends to increase at the same time as Y. If the covariance is negative then this implies that X tends to decrease as Y increases. If changes in X and Y have a random relationship, then the covariance will be close to zero. If two variables have a covariance of zero they are called uncorrelated. If two variables are independent they will be uncorrelated. However, two variables which are uncorrelated are not always independent.

10.4 Model for robot-environment interaction[5]

There are two fundamental types of robot-environment interaction:

1. The robot can acquire information about its environment using its sensors.
2. The robot can also influence its environment through its actuators.

10.4.1 State

We define the state(x) as the collection of all aspects of the robot and its environment that can impact the future. x_t denote the state at time t. In addition, we define notation: $x_{t_1:t_2} := x_{t_1}, x_{t_1+1}, \dots, x_{t_2}$. State variables include:

- Robot pose (e.g., robot location and orientation relative to a global coordinate frame)

- Robot velocity
- Location and features of surrounding objects in the environment, etc

A state x_t will be called complete if it is the best predictor of the future. The future may be stochastic, but no variables prior to x_t may influence the stochastic evolution of future states, unless this dependence is mediated through the state x_t . Temporal processes that meet these conditions are only known as Markov chains (Markov property).

10.4.2 Measurement and control data

Environment measurement data provides information about a momentary state of the environment. The measurement data at time t will be denoted z_t . We define notation: $z_{t_1:t_2} := z_{t_1}, z_{t_1+1}, \dots, z_{t_2}$, denotes the set of all measurements acquired from time t_1 to time t_2 , for $t_1 \leq t_2$.

Control data carry information about the change of state in the environment. In mobile robotics, a typical example of control data is the velocity of a robot. Control data will be denoted u_t . The variable u_t will always correspond to the change of state in the time interval $(t-1; t]$. We define notation: $u_{t_1:t_2} := u_{t_1}, u_{t_1+1}, \dots, u_{t_2}$, denotes the sequences of control data from time t_1 to time t_2 , for $t_1 \leq t_2$.

Environment perception provides information about the environment's state, hence it tends to increase the robot's knowledge. Motion, on the other hand, tends to induce a loss of knowledge due to the inherent noise in robot actuation and the stochasticity of robot environments.

10.4.3 State equation

The evolution of state and measurements is governed by probabilistic laws. The probabilistic law characterizing the evolution of state might be given by a probability distribution of the following form: $p(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t})$. Notice that through no particular motivation we assume here that the robot executes a control action u_1 first, and then takes a measurement z_1 .

If the state x is complete then it is a sufficient summary of all that happened in previous time steps. In particular, x_{t-1} is a sufficient statistic of all previous controls and measurements up to this point in time, that is, $u_{1:t-1}$ and $z_{1:t-1}$. From all the variables in the expression above, only the control u_t matters if we know the state x_{t-1} . That is :

$$p(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

10.4.4 Measurement equation

The probability of getting a measurement z_t is also given by a probabilistic law. Again, if we assume that the state x is complete, then we can say that state x_t already encodes the effect of all previous states $x_{0:t-1}$, previous measurements, $z_{1:t-1}$, and previous controls $u_{1:t}$. This allows us to arrive at the equation:

$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

10.4.5 Overall stochastic model

It is important to keep the order of occurrence of different events in the model consistent. In our notation, we assume that the control effort u_t is determined after taking into account the state x_{t-1} and measurement z_{t-1} . The control effort u_t causes the robot to transition to a new state x_t where a new measurement z_t is taken, which in turn allows the determination of a new control effort u_{t+1} . This network model is a stochastic model known as a Markov process because the new state x_t only depends on the previous state x_{t-1} and the previous control u_t . In addition, since we cannot directly observe the current state, but instead estimate it through measurements, this model is further classified as a hidden Markov model.

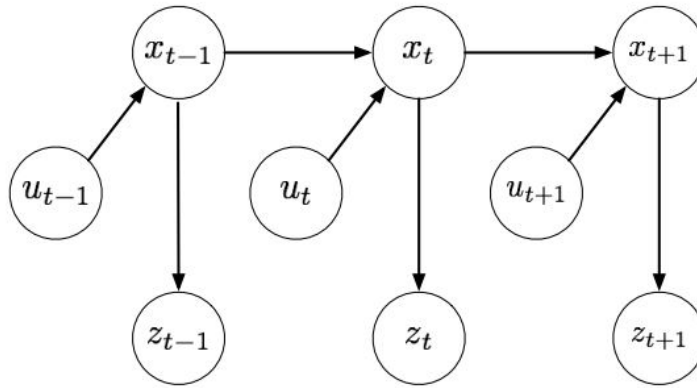


Figure 10.5: Overall dynamic Bayes network model (Hidden Markov Model)

10.4.6 Belief distribution

We can encode our internal knowledge of the state in a belief distribution, which is a probability distribution that has a value for each possible hypothesis of the true state. Formally, our belief distribution at a particular state is the probability of being at that state given all the measurements and control efforts we've seen so far.

$$bel(x_t) := p(x_t \mid z_{1:t}, u_{1:t})$$

Note that this probability distribution can be unimodal or multimodal. A useful tool to calculate our belief distribution is an intermediate belief distribution known as the prediction distribution. The prediction distribution is the probability of being at a state given all the past measurements (except the one we took at this new state) and the previous control efforts.

$$\overline{bel}(x_t) := p(x_t \mid z_{1:t-1}, u_{1:t})$$

Therefore, the first step to calculate our final belief distribution is to calculate the prediction distribution. Next, we add the latest measurement to sharpen the belief. This second step of calculating the true belief from the prediction belief is called the correction step or the measurement update. This leads us to the Bayes filter algorithm.

10.5 Bayes' filter algorithm

The Bayes' filter algorithm is the most general algorithm for calculating state beliefs. The key assumption made in this algorithm is that the state x_t is complete, i.e. only depends on the previous state and control effort, x_{t-1} and u_t . Note that the Bayes' filter algorithm is the most general algorithm but is not necessarily achievable. The algorithm assumes a continuous state space, continuous time, and requires that we be able to compute all the beliefs in an arbitrarily dimensionally large space. Though not practical, the Bayes' filter algorithm is still a great starting point to derive more practical algorithms that can be used in different scenarios.

10.5.1 Overview of algorithm

This is a recursive algorithm. It is initialized with some guess of the initial state, i.e. an initial belief $bel(x_0)$. This can be a uniform distribution of density or point masses if we have no prior information. The first step is to carry out the prediction step to compute $\overline{bel}(x_t)$ using the previous control effort u_t used to get to state x_t . This is simply the integration over all possible previous states x_{t-1} , of the product of the prior belief $bel(x_{t-1})$ (where we think we were before) and the transition model given that we know our last control effort $p(x_t | u_t, x_{t-1})$. This prediction belief is an encoding of how likely it is that we're now at x_t .

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

The second step is the measurement update where the most recent measurement is incorporated to sharpen our belief. This involves taking our prediction belief and multiplying it by the probability of seeing the measurement that we did z_t if we were actually at our predicted state x_t . This allows us to correct our prediction belief and find our true belief.

$$bel(x_t) := \eta p(z_t | x_t) \overline{bel}(x_t)$$

After these two steps, a control effort u_{t+1} is taken, we transition to a new state x_{t+1} , and the process of prediction and measurement update is repeated. This is what makes the Bayes' filter algorithm a recursive algorithm.

10.5.2 Derivations

The second step of the Bayes' Filter Algorithm involves the measurement update of our belief model. Here we will go into further detail regarding the derivation of the measurement update.

We begin with our belief

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

We then expand this using Bayes' Rule to get:

$$bel(x_t) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t | z_{1:t-1}, u_{1:t})}{p(z_t | z_{1:t-1}, u_{1:t})}$$

Where the denominator is the normalizer and is equal to η^{-1} . Next, we use the Markov Property to get:

$$bel(x_t) = \eta p(z_t | x_t) p(x_t | z_{1:t-1}, u_{1:t})$$

Where the right term equates to the prediction distribution $\overline{bel}(x_t)$. Next we derive the Correction Update that we perform on that prediction distribution

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})$$

Using the law of total probability, we expand to get:

$$\overline{bel}(x_t) = \int p(x_t | x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1}$$

Next we use the Markov Property to simplify the left term:

$$\overline{bel}(x_t) = \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1}$$

For general output feedback policies, u_t does not provide feedback additional information on x_{t-1} and so:

$$\overline{bel}(x_t) = \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | z_{1:t-1}, u_{1:t-1}) dx_{t-1}$$

The right term then simplifies so that we get our result:

$$\overline{bel}(x_t) = \int p(x_t | x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1}$$

10.5.3 Discrete Bayes' filter algorithm

For problems with finite state spaces, a Discrete Bayes' Filter can be implemented. Initially, we represent the belief state, $bel(x_t)$, as a probability mass function $\{p_{k,t}\}$. The pseudo code for the algorithm is shown below:

```

Data:  $\{p_{k,t-1}\}, u_t, z_t$ 
Result:  $\{p_{k,t}\}$ 
initialization;
foreach  $k$  do
     $\bar{p}_{k,t} = \sum_i p(X_t = x_t | u_t, X_{t-1} = x_i) p_{i,t-1}$ ;
     $p_{k,t} = \eta p(z_t | X_t = x_k) \bar{p}_{k,t}$ ;
end
return  $p_{k,t}$ 

```

Algorithm 1: Discrete Bayes' Filter Algorithm

10.6 Additional Resources

There are some additional resources to which you can refer. Chapter 2 in [5] provided basic probabilistic concepts that are frequently used in robotics, the formal model of robot interaction, introduction to *Bayes filters* and implementing issues associated with that.

If you want to dive deeper into *Bayes filters*, [4] is a comprehensive book on Bayesian filtering and smoothing and applications of *Bayes filters* such as *Kalman filters* (including extended and unscented *Kalman filters*), *Gaussian filters* and *Particle filters*. As you know, Bayes recursive filter is not a practical algorithm, which means it cannot be implemented on a computer. Therefore, it is important to understand those implementations using the principle of *Bayes filters*.

If you are interested in the original paper on *Kalman filters* by R.E.Kalman, you can refer to [3]. This famous paper described a recursive solution to discrete filtering problem. With the advance of computer and robotics, this paper inspired a lot of researches and applications.

There is a great online article [1] which provides an intuitive explanation of what is a *Kalman filters* and how it works. This article approaches *Kalman filters* in a slightly different way (more from a linear algebra point of view). Nevertheless, it will help you build insightful intuitions of *Kalman filters* and is very interesting to read through.

References

- [1] How a kalman filter works, in pictures. <http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>, Feb. 2019.
- [2] C. Clark. Lecture 9—Markov Localization. E160:Autonomous Robot Navigation, Harvey Mudd College, 2016.
- [3] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [4] S. Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.
- [5] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

Contributors

Winter 2019: Akshay Gupta, Ian Vorbach, Orien Zeng, Somrita Banerjee, Zhengyu Chen
Winter 2018: Junjie Ke, Ryan Wong, Amine Elhafsi, Samuel Sowell, Peter Zachares