Institute for Aerospace Studies
UNIVERSITY OF TORONTO

G M T D'Eleuterio

ROB301H1 F

Introduction to Robotics

the Concept of Bayesian Localization

1 The Tourist Robot

The localization procedure we have been looking at is based on straightforward quantitative measurements and we determine the pose of the robot in terms of clear Cartesian or polar coordinates. But when we say that we'll meet a friend in the foyer of the Galbraith Building, we're not giving a set of detailed coordinates. Coordinates can sometimes be inconvenient. Maybe you'd just like to tell a robot to deliver a package to Professor Robotham's office and all the information you can provide is that it's on the second floor, third door on the left down past the north wing corridor. How can the robot accomplish the task?

This is a little like the tourist's problem. You're trying to find out where you are by identifying the landmarks around you. Imagine a robot dropped somewhere in North America. It comes across a building that, according to its stored information, looks a lot like University College. So it might think that it on King's College Circle. But, wait a minute. The robot also knows that there's a similar structure in Washington, DC, the main building of Smithsonian Institute. It doesn't have enough detailed information to distinguish between the two. In fact, in its database, the robot discovers several other candidate buildings that it could be. However, during its travels around the city, it spots a tower. Well, there's the CN Tower in Toronto and the Washington Monument in Washington. There's also the Space Needle in Seattle but there's no candidate building in Seattle. The robot decides it must be in Toronto or Washington. Now the robot has also noticed in its wanderings a large domed white building. As far as it knows, there's no such building in Toronto but there is the Capitol in Washington. It must be in Washington, DC, in front of the Smithsonian Institute.

Figure 1. University College (Toronto) and Smithsonian Institute (Washington, DC)

What the robot was doing, as we often do in our own travels as a tourist, is reason probabilistically what the landmarks were telling it. The robot is constructing a belief structure of its location and continually interrogates that structure, shaping it as the robot acquires new information. How can we codify this into a localization procedure?

2 The Essential Concept

We shall imagine that we have a map. However, this map does not have to be a *geometric map*; it may be a *topological map* as shown in Figure 2. Localization in a topological map means identifying the node at which the robot is located at the time. Another possibly is an *occupancy grid map*, in which the world of the robot is discretized into cells (Figure 3). This discretization can also be applied to the *configuration manifold* of the robot, where the range for the pose, x, y and θ, say, is divided into cells.

Denote by Λ the set of all possible locations for the robot. Furthermore, let $x_k \in \Lambda$ be the state, *i.e.*, location, of the robot at time k; let u_k be a control input (or an array of inputs) to the robot and z_k a measurement (or an array of measurements) made of the environment. We can assume that u_k belongs to a finite set Υ of control

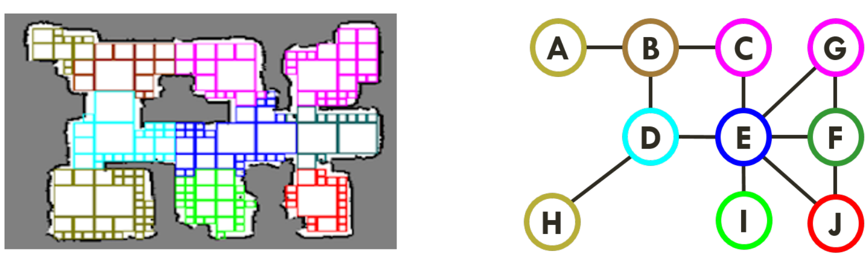

Figure 2. A geometric map (left) and a corresponding topological rendering (right)

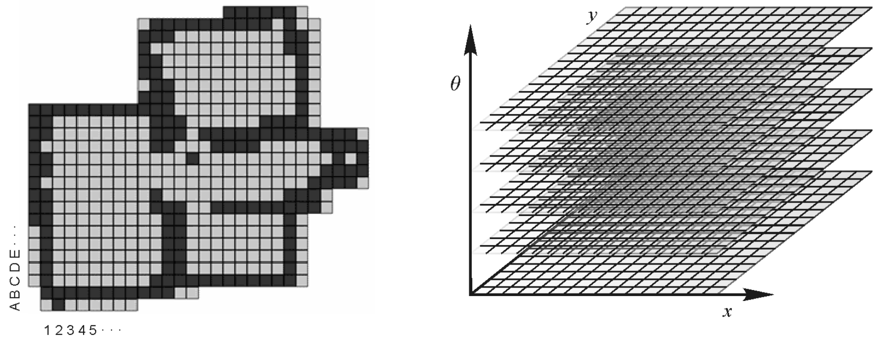

Figure 3. An occupancy grid map (left) and a discretization of the configuration manifold (right)

options and z_k to a finite set Σ of sensor readings.

State Prediction. Let $z_{0:k}$ be the sequence of measurements made up to and including time k, *i.e.*, $z_{0:k} = z_0, z_1 \cdots z_k$; that is, $z_{0:k}$ is the history of all the robot's measurements. Also, let $p(x_{k+1}|z_{0:k})$ be the probability that the robot is at x_{k+1} given the sequence $z_{0:k}$. In fact, $p(x_{k+1}|z_{0:k})$ is the probabilistic prediction of the state x_{k+1}. Now $p(x_{k+1}|z_{0:k})$ is determined by

$$p(x_{k+1}|z_{0:k}) = \sum_{\upsilon_k \in \Upsilon} p(x_{k+1}|\upsilon_k, z_{0:k}) p(\upsilon_k|z_{0:k})$$

where $p(x_{k+1}|\upsilon_k, z_{0:k})$ is the conditional probability that the application of control υ_k, with history $z_{0:k}$, will deliver the robot to x_{k+1} by the next time step. If $\upsilon_k = u_k$ is chosen with certainty, $p(u_k|z_{0:k}) = 1$ and the probabilities of all the other options for υ_k is zero. Thus

What You Need to Know

Probability	$p(x)$
Joint probability	$p(x, y)$
Conditional probability	$p(x\|y)$

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

$$p(x) = \sum_{\forall y} p(x|y)p(y)$$

$$p(x|z) = \sum_{\forall y} p(x|y, z)p(y|z)$$

Bayes's rule

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

$$p(x_{k+1}|z_{0:k}) = p(x_{k+1}|u_k, z_{0:k})$$

We can write this probability conditionally in terms of the previous state x_k of the robot; that is,

$$p(x_{k+1}|z_{0:k}) = p(x_{k+1}|u_k, z_{0:k}) = \sum_{x_k \in \Lambda} p(x_{k+1}|x_k, u_k, z_{0:k}) p(x_k|z_{0:k}) \tag{1}$$

But we make the assumption here that

$$p(x_{k+1}|x_k, u_k, z_{0:k}) = p(x_{k+1}|x_k, u_k) \tag{2}$$

We defend this simplification by arguing that the current location x_k and the control action u_k is sufficient to probabilistically determine x_{k+1}. Accordingly (1) becomes

$$p(x_{k+1}|z_{0:k}) = \sum_{x_k \in \Lambda} p(x_{k+1}|x_k, u_k)p(x_k|z_{0:k}) \tag{3}$$

The conditional probability $p(x_{k+1}|x_k, u_k)$ essentially describes the state model for the robot. Equation (3) represents the state prediction for the robot but it is not corroborated by any additional sensory data at time $k+1$ yet. Let's look at that next.

State Update. Just as in the case of Kalman filtering, we now update the state prediction based on the new measurement z_{k+1}. We seek therefore the probability $p(x_{k+1}|z_{0:k+1})$ and, using Bayes's rule, we have

$$p(x_{k+1}|z_{0:k+1}) = p(x_{k+1}|z_{0:k}, z_{k+1}) = \frac{p(z_{k+1}|x_{k+1}, z_{0:k})p(x_{k+1}|z_{0:k})}{p(z_{k+1}|z_{0:k})}$$

In a move similar to (2), we assume that

$$p(z_{k+1}|x_{k+1}, z_{0:k}) = p(z_{k+1}|x_{k+1}) \tag{4}$$

That is, the measurement at time $k+1$ depends only on the state at $k+1$ and not on the measurement history. Equations (2) and (4) reflect a Markov-like property.

We finally have then

$$p(x_{k+1}|z_{0:k+1}) = \frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|z_{0:k})}{p(z_{k+1}|z_{0:k})} \tag{5}$$

The utility of this result depends on knowing $p(z_{k+1}|x_{k+1})$, which may readily be interpreted as the measurement model. Intuitively, this conditional quantity describes the probability of getting a particular measurement based on where the robot is. It's the sort of thing that can obtained, for example, through experimentation or a repertoire of compiled data. Indeed, $p(z_{k+1}|x_{k+1})$ is sometimes referred to as the *certainty matrix.*

What You Need to Know

Markov Property. *The conditional probability of future states of a stochastic process $x_0, x_1, x_2 \cdots x_k$ depends only on the present state and not any of the past states,* i.e.,

$$p(x_{k+1}|x_k, x_{k-1} \cdots x_0) = p(x_{k+1}|x_k)$$

We are also faced with another problem, namely, determining $p(z_{k+1}|z_{0:k})$ which would be quite awkward to evaluate. Fortunately, though, we never have to do so. The update (5) must be calculated for all $x_{k+1} \in \Lambda$ and we must have

$\sum_{x_{k+1} \in \Lambda} p(x_{k+1}|z_{0:k+1}) = 1$. But the denominator does not depend on the state; it is therefore constant and we can account for it by simply normalizing:

$$p(x_{k+1}|z_{0:k+1}) = \frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|z_{0:k})}{\sum_{\xi_{k+1} \in \Lambda} p(z_{k+1}|\xi_{k+1})p(\xi_{k+1}|z_{0:k})} \tag{6}$$

If we're happy just to have relative measurements, then we don't even have to bother with the normalization and just accept that the sum of $p(x_{k+1}|z_{0:k+1})$ over $x_{k+1} \in \Lambda$ is not necessarily unity.

The method we have outlined is often called *Markov localization* but *Bayesian* or *probabilistic localization* suits it better. The algorithm is briefly summarized in Table 1.

Table 1. Bayesian localization alogrithm

At step $k+1$, assume as given $p(x_k|z_{0:k}), u_k, z_{k+1}$	
State Prediction	$p(x_{k+1}|z_{0:k}) = \sum_{x_k \in \Lambda} p(x_{k+1}|x_k, u_k)p(x_k|z_{0:k})$
State Update	$p(x_{k+1}|z_{0:k+1}) = \frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|z_{0:k})}{\sum_{\xi_{k+1} \in \Lambda} p(z_{k+1}|\xi_{k+1})p(\xi_{k+1}|z_{0:k})}$
Rinse and repeat	

Observations. We have noted that $p(x_{k+1}|x_k, u_k)$ may be regarded as the state model. The computation of the state prediction described by (3) is thus comparable to the *a priori* state estimate in Kalman filtering given by $\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_k\hat{\mathbf{x}}_{k|k} + \mathbf{B}_k\mathbf{u}_k$. Correspondingly, the state update formula (6) takes the role of the *a posteriori* estimate $\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{W}_{k+1}(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k})$.

What we don't have is something like the covariance matrix $\mathbf{P}_{k|k}$. But then we don't need anything because we calculate the entire distribution $p(x_k|z_{0:k})$ over the variable x_k. A summary of the correspondence between the Kalman filtering and the Bayesian approach is given in Table 2.

3 An Example

As an example, let's return to the problem of finding Professor Robotham's office. A geometric map and its topological representation are shown in Figure 4(a). For the state model $p(x_{k+1}|x_k, u_k)$, we will assume three control options: $u_k = -1$ indicating that the robot is commanded to move west; $u_k = 0$ where it receives no input

Table 2. Bayes *vs.* Kalman

Bayes	Kalman
$p(x_{k+1}\|x_k, u_k)$	$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{v}_k$
$p(x_k\|x_k)$	$\mathbf{z}_k = \mathbf{D}_k\mathbf{x}_k + \mathbf{w}_k$
$p(x_{k+1}\|z_{0:k})$	$\hat{\mathbf{x}}_{k+1\|k}$
$p(x_{k+1}\|z_{0:k+1})$	$\hat{\mathbf{x}}_{k+1\|k+1}$

command; and $u_k = +1$ where it is commanded to move east. We can imagine that nominally the robot will be commanded to move only one node (either to the east or west) or stay where is was. However, noise can infiltrate the system causing the motion to result in another location. We shall take the state model as being independent of where the robot is at any given time; that is, given that the robot is at $x_k = X$, where $X \in \Lambda = \{A, B, C \cdots J\}$, we then assume for $u_k = +1$, say, there's a nonzero probability that at the next time step it will end up at $x_{k+1} \in \{X - \chi, X, X + \chi, X + 2\chi\}$. We use $X + \chi$ as a shorthand to indicate the letter location one beyond X, $X + 2\chi$ two letter locations beyond X and so on.

For example, if the robot is at node C at step k then there is a probability $p(B|C, +1)$ that it will move backward to B, despite the robot being commanded to move east. The robot will stay at C with probability $p(C|C, +1)$, which physically means that the robot didn't respond to the input control. With probability $p(D|C, +1)$, it will move to D as intended and so on. Moreover, these probabilities

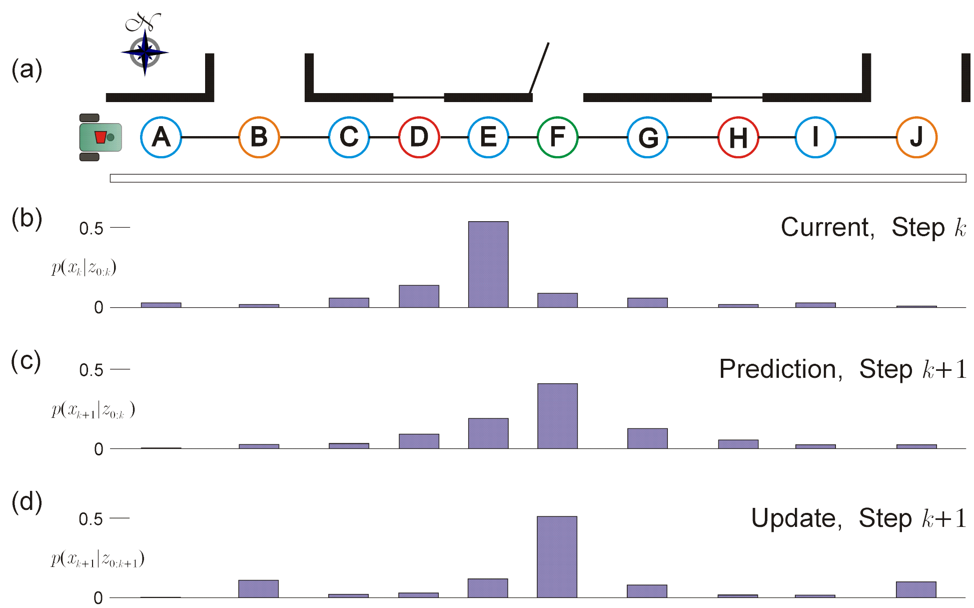

Figure 4. Hallway scenario with probability distributions

are assumed the same for any relative movement, *i.e.*, $p(A|B,+1) = p(B|C,+1) = p(C|D,+1)$, $p(A|A,+1) = p(B|B,+1) = p(C|C,+1)$ and so on.

We take, in this example, the state model to be given by Table 3 below.

Table 3. State model $p(x_{k+1}|x_k = X, u_k)$

x_{k+1} | $u_k =$	-1	0	$+1$
$X - 2\chi$	0.10		
$X - \chi$	0.70	0.10	0.05
X	0.15	0.80	0.15
$X + \chi$	0.05	0.10	0.70
$X + 2\chi$			0.10

In the measurement model $p(z_k|x_k)$, we can account for much more than simple quantitative measure of location. If the robot has a rich sensor, such as a camera or lidar, we take its reduced data to imply that it has detected one of $\Sigma = \{$HALLWAY, WALL, OPEN DOOR, CLOSED DOOR, NOTHING$\}$. ("Nothing" refers to a null measurement—oops!) How the image processing or other data processing, which may result from a fusion of several sensors, will not be our concern here.

If the robot sensor is facing a closed door, say, the sensor system, which introduces noise, will not necessarily agree that it "sees" a closed door. The sensors may with nonzero probability report that it is an open door or a wall. In the model, x_k represents the node location of the robot but given the map there is a correspondence between a node and one of the "descriptions" in $\{$HALLWAY, WALL, OPEN DOOR, CLOSED DOOR$\}$, for which we'll use the designation $x_k \sim$ WALL, for instance. We observe significantly that the number of sensor readings and the number of state descriptions do not have to agree; moreover, the actual measurements do not directly relate to the actual states $x_k \in \Lambda$ in the same way that in Kalman filtering the sensors do not directly have to measure any one of the states.

We summarize the measurement model in Table 4. Note that in no case will the sensor system report the correct state with overwhelming probability.

Table 4. Measurement model $p(z_k|x_k)$

z_k | $x_k \sim$	HALLWAY	WALL	OPEN DOOR	CLOSED DOOR
HALLWAY	0.65	0.10	0.20	0.05
WALL	0.05	0.45	0.10	0.20
OPEN DOOR	0.20	0.10	0.50	0.15
CLOSED DOOR	0.05	0.30	0.15	0.55
NOTHING	0.05	0.05	0.05	0.05

Now let us run through one step in the Bayesian localization algorithm given the current probability distribution $p(x_k|z_{0:k})$ (Figure 4b) and having applied $u_k = +1$. At time $k+1$, the sensor system reads HALLWAY. The subsequent calculations are summarized in Table 5; the predicted localization probabilities are shown in Figure 4(c) and the updated probabilities in Figure 4(d).

Table 5. One step in the Bayesian localization algorithm with $u_k = +1$

$\boldsymbol{x_k}$	Current $p(x_k\|z_{0:k})$	Prediction $p(x_{k+1}\|z_{0:k})$	Measurement z_{k+1}	Update $p(x_{k+1}\|z_{0:k+1})$
A	0.030	0.006	HALLWAY	0.003
B	0.020	0.027		0.109
C	0.060	0.033		0.021
D	0.140	0.092		0.029
E	0.540	0.190		0.118
F	0.090	0.409		0.509
G	0.060	0.127		0.079
H	0.020	0.056		0.017
I	0.030	0.025		0.016
J	0.010	0.025		0.099

Note that if we imagine the robot to have actually moved to node F at step $k+1$, then the sensor system incorrectly reported what it detected, *i.e.*, a hallway instead of an open door. Yet, because of the prior probability distribution $p(x_k|z_{0:k})$, the new localization given by $p(x_{k+1}|z_{0:k+1})$ has the robot most probably at node F. The algorithm is tolerant to mistakes owing to the accumulation of information just we are. . . we hope.