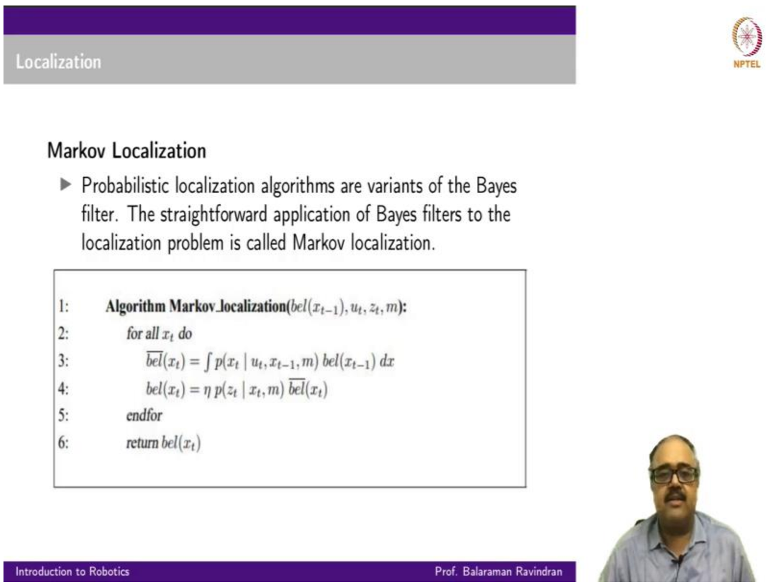
Introduction to Robotics
Professor. Balaraman Ravindran
Department of Computer Science
Indian Institute of Technology, Madras
Lecture No. 42
Markov Localization

(Refer Slide Time: 0:14)

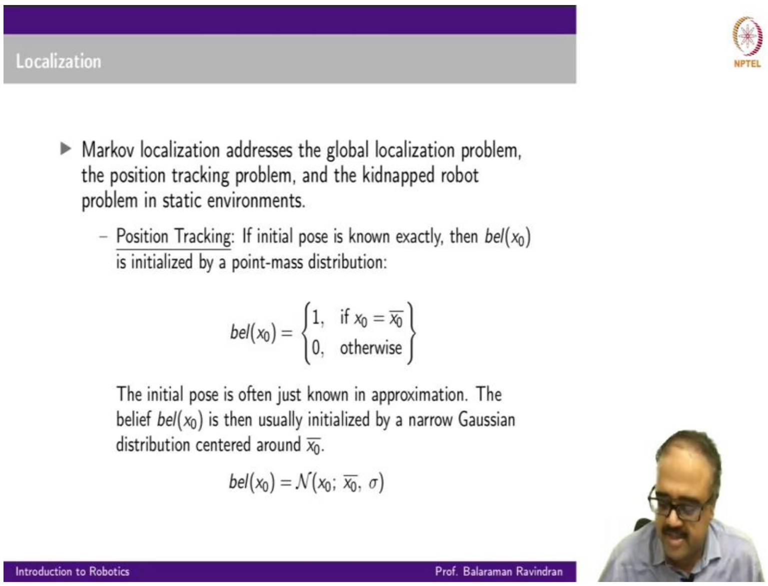


So, in last lecture we looked at the taxonomy of the localization problems. And, this lecture we will look at a very simple, the Markov Localization algorithm. So, if you look at this, this is exactly the Bayes filter algorithm except that instead of looking at, just the state, I also have to look at the map here. So, now we already looked at, looking, the, the motion model with respect to the map and we also looked at the measurement model with respect to the map.

So, it is nothing more than, there is running the original Bayes filter algorithm where both motion model and the measurement model are going to use the knowledge of the map. So, if you look at their, the, the Markov localization algorithm, it takes as input your previous belief state, your current action, your current measurement and the map. And, it could be potentially the current map. If, if you have, if you have a time-varying map.

And, then basically we do the same thing, for all X we obtain bel bar which is taking into account the motion models. So, this is the prediction update. And, then we also get bel by accommodating the measurement. So, this is the correction or the measurement update. And, this gives me the position of X taking into account the knowledge of the map, M. And, then after I have done this update for all the states, it return the new belief state. Just the Bayes filter algorithm. And, this is known as the Markov localization problem.

(Refer Slide Time: 01:44)

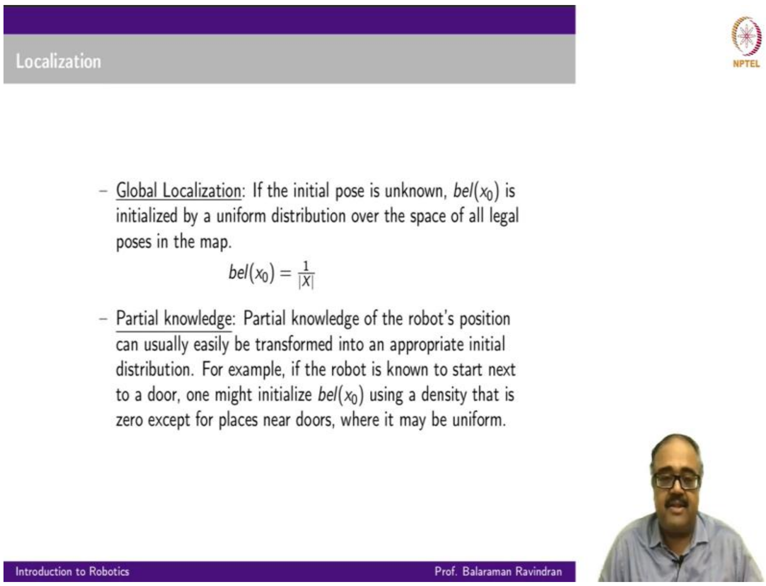


And, in fact, if you roughly think about it, the Markov localization problem can address any of the three local and global problems we talked about. So, the first one is the Global localization problem. We just say that the belief distribution is unique. So, or if it is the position tracking problem and I will set the belief distribution to be something very very focused. If initial pose is completely known, then I will set my bel X naught to be 1, if X naught equal X naught bar, which is where the, which is the right initial position.

So, so I know that the robot is in X naught bar, therefore I will set bel X naught, bel X naught bar to 1. And, bel X naught if it is other than X naught bar, I will set it to 0. So, this is the easy way of handling the position tracking problem. And, if you do not know the exact initial position, I only know the initial position around the small window and then I can just treat it like a narrow Gaussian, where I have my sigma naught, which is a very very narrow initial belief. And, my X naught is, X naught bar is the mean.

So, in the first case I assumed it was exactly at X naught bar. In this case I am assuming that okay, it is somewhere around X naught bar, not too far away. Then how far away is determined by sigma. And, then again, this is assuming that there is just one correct location and then I do the position tracking after that just the normal belief update gives us the position tracking equation.

(Refer Slide Time: 03:25)

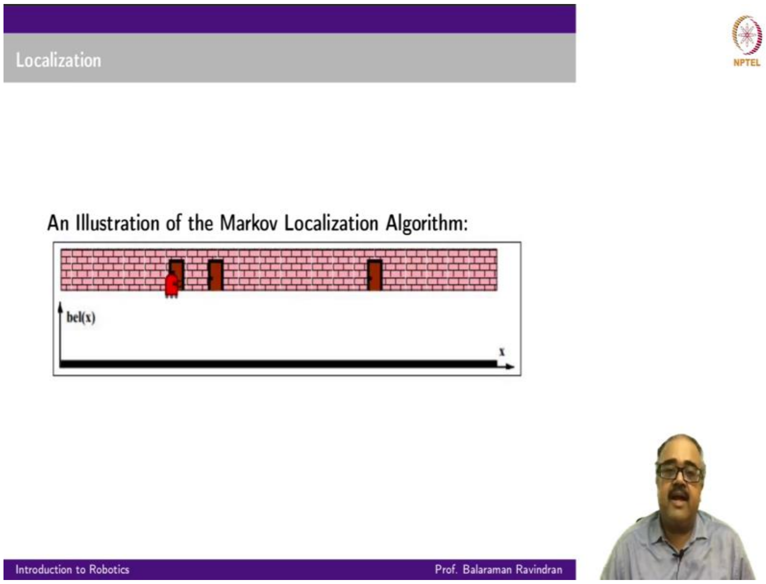


So, let us suppose I want to do global localization. I just say, it is just the uniform distribution. My bel X naught is a uniform distribution. So, I just say that it is the, one over size of X and starts from there. And, the regular updates will essentially give me the global localization equation. And, if I have a belief distribution like the particle filter case which can take care of multiple hypotheses, then I can do proper global localization.

And if I have a Gaussian filter, then it will be a little tricky because it going to very quickly narrow down on a single hypothesis, which might be wrong. I mean, you still might have some noise around the hypothesis but it will, find it hard to actually fit the right distribution quickly. And, then what about the kidnapped robot problem. So, it is some kind of a partial knowledge. So, like bel X naught can actually be something arbitrarily.

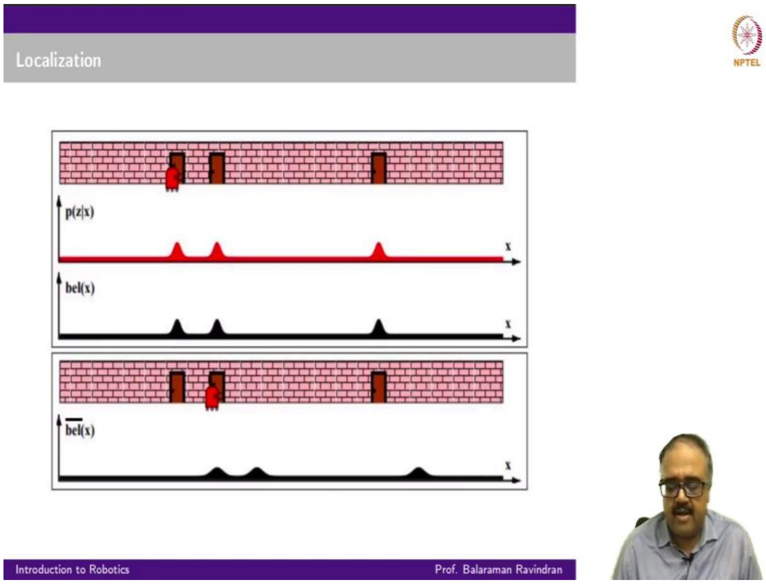
So, so, in fact I can, if you remember this figure I was showing you, I was in a world with 3 doors. And, if I say that my first sensor reading is that I am next to one door. So, I can always say that I will start of near a door, in this case I can say, the density has some value near the door except in other places it is all 0. So, that could be a way of accommodating partial knowledge.

(Refer Slide Time: 04:47)



And, so if I want to accommodate the kidnapped robot question, I have to make sure that my belief distribution is such that my updates, updates never make the belief anywhere about to 0. So, here is the example of the Markov localization algorithm. This is running in a global localization setting. Therefore, when it starts, the belief is anywhere in the world. So, that is the uniform distribution over the span of the world.

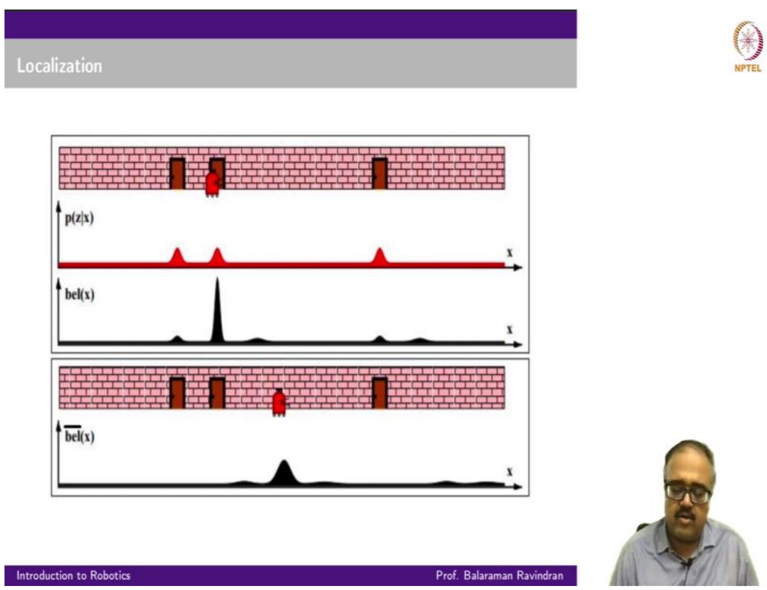
(Refer Slide Time: 05:14)



Next, what happens, what happens, it senses that it is next to a door and therefore the belief distribution becomes one of these 3 locations because only those 3 places are likely to activate the door sensor. This is the probability of the door sensor getting activated in these locations and therefore, I basically put into these 3 locations. And, I am able to do this mainly because I have the map. Because I know where the doors are in the map and therefore as soon as I sense the door, I can put myself in these three places where the doors are in the map.

And, then what I do is I move. So, I move to the right. So, that is the action that has happened and I move to the right. If you remember, we saw this already earlier, in the, in the case of the filter problem. But, now I am just telling you that that it is a localization problem as well. There we did not think about the map. Here we have to have the map. So, the bel bar tells me that I move. So, basically what happens is these 3 peaks, just move to the right by some amount. And, then they also get, spread out because my motion model has some noise.

(Refer Slide Time: 06:20)



And, then finally I sense a door again and the door model has not changed. The door model still is the same thing. And, so, given that these were the three places where my motion model put me. And remember, we started off with the uniform distribution. This has actually not gone to 0. There is still some, some probability that I could be anywhere in the world. Because I want to make sure that I can allow for the kidnapped robot problem.

And, therefore now I make a measurement. These are the 3 places where there is a door. And, then what happens is I combine the bel bar and the measurement. And, now I am more or less sure that this is where I am right. And, these places, you know, kind of get dampened down. So, now what happens is the robot start moving further and further. And, none of these measurements are enough for me to make any refinement. These measurements essentially just tell me a wall. I mean, and wall could be anywhere.

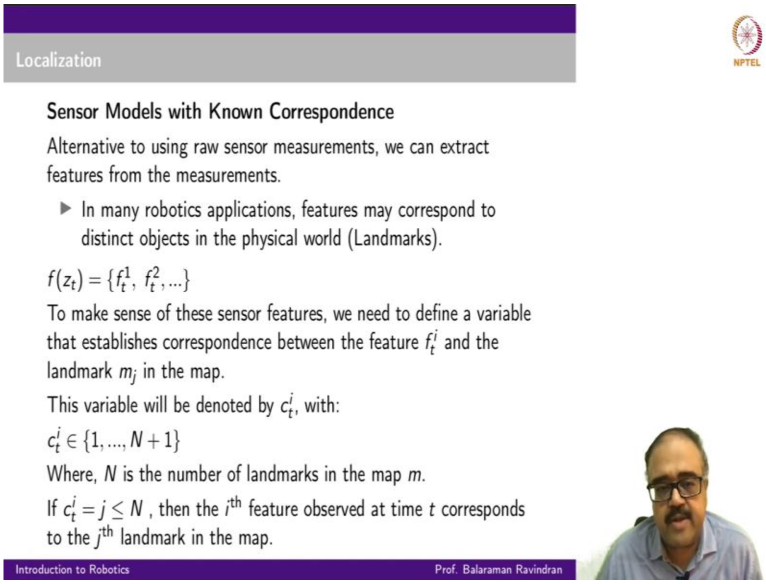
There are many, many places where I could sense the wall and so it is not really telling me much. And, essentially it is just that my movement model noise getting added. Therefore, from a very sharp distribution here, I basically go down to a distribution that is kind of more spread out and not as peak. Because my motion model keeps diffusing that track distribution a little by little. So, this is essentially the localization problem.

It is slightly different from the Markov, from the Bayesian filter problem because localization here is done with respect to the map. And, it does not look very different for you because the sensor model accommodates the map already. And, the movement model and sensor model are going to accommodate the map. And, here the movement model really does not depend on the map.

So, because I have not tried to open the door. So, that is, that is basically the Markov filter algorithm in operation here. So, if you think about it, we talked about multiple kinds of maps. So, we talked about feature based maps and we also talked about location based maps. And, where we looked at occupancy grid maps as a location based map. If you remember, feature based maps; they were collection of features or objects and their properties; where the properties could include a position as well, right other than various other features of the object.

And, so the localization problem, the algorithms as we have been seeing them so far; the Bayesian filter algorithm, as we have seen so far are amenable for working with location based map, especially grid based map. They are great for working with occupancy grid maps. When we start moving to feature based maps, we have to do a little bit more work in order to do localization. Sometimes, actually the feature based maps are more powerful for it to the localization because you can localize yourself with respect to these features.

(Refer Slide Time: 09:19)



In such cases what we do is instead of looking at the raw sensor measurements, we try to extract features from the measurements. So, we, we take the raw sensor measurements and we try to extract certain features from the measurements. So, the features could be something like, what is the location of the door? Or what is the location of the table in the environment? So, I know that there is, the map consists of 5 tables, 3 chairs, 4 doors and 2 windows or something like that.

Now, instead of saying that, I have these in in in like a grid model and I am going to localize, I could potentially just use these in terms of the landmark or or or feature based model, where each object is like a landmark. And, I have these features corresponding to these landmarks. Now, if I know for sure, what sensor feature corresponds to what landmark, it is much easier for me. So, if I know that, the reading coming from sensor 3 is actually sensing door 2. How could this happen? Let us say that doors have numbers on them and my sensors are cameras. So, I could look at the door and say, that is door 2.

So, I have this correspondence. So, I know exactly or or or my features could be a meeting beac…, like, like a radio beacons, or Bluetooth beacons. They are emitting signals in the, in the environment and as soon as they receive a signal, I know which Bluetooth beacon is a meeting that signal. So, I am going have multiple features. If you remember, I was going to have multiple features, and then what I do is, I get the sensor reading, which is Zt. From the Zt, I am going to compute these feature values.

And, to make sure that I am assigning the right feature value or the right post to the right landmark, I am going to maintain what is called the correspondence. So, the correspondence here is something like this. So, cit means that the the ith sensor feature I have computed at time t. I am computing f1t, f2t, bla bla bla. So, the ith feature I compute at time T with the value it takes tells me what is the landmark. We remember, we could have one to n landmarks on the map, right. Each landmark has this feature corresponds it.

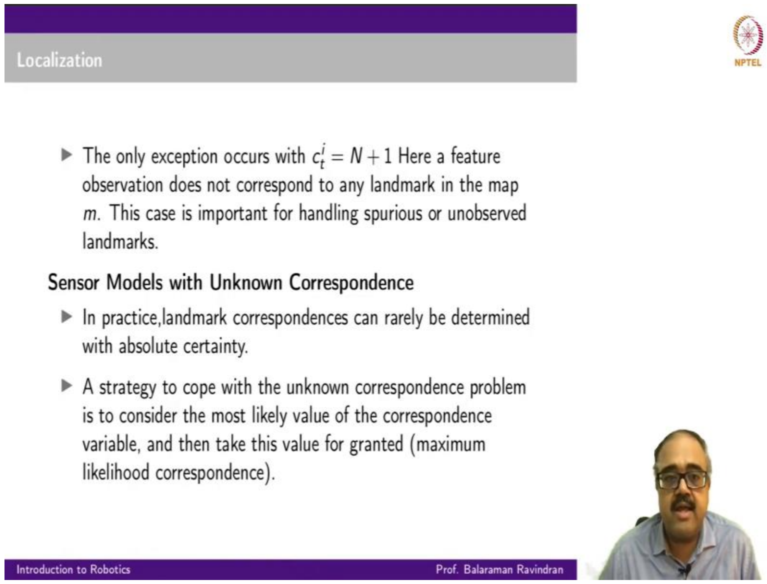
Let us say that I computed distance to a door is 5 meters. Let us say, so, may be our distance to a door is 1 meter. Let us say, I have computed distance to a door is 1 meter. Which door is this? Is it door 1, door 2 or door 3? So, the first distance measurement I computed is distance to the door is 1 meter. And, suppose this is door 3, my c1t will be 3. So, what is, what are we doing here? So, my f1t, my f1t is 1 meter. My c1t is 3. What this mean is, my distance to door number 3 is 1 meter. Is it clear?

So, it is not always the case that my first sensor reading gives me distance 2 to door 3. It could be some other time I come in opposite direction, say, my fifth sensor reading might be giving the distance to door 3. In which case my c5t will be 3 and c5t would be, say 1 meter, 2 meter, or whatever is the distance to the door. So, capital N here is the number of landmarks in the map and the N plus 1, the N plus 1, is essentially to say, for whatever reason I do not know what ,what this feature corresponds to. There is some landmark that I am not able to map this feature to. And, therefore I will put it at the N plus 1th value.

Suppose, there are 10 landmarks and if my c variable says that its value is 11, that means, that for that particular feature say, c5t is 11. That means the fifth feature; I do not know what it has computing. So, that is what I mean by assigning 11 to it. So, this is exactly what this line is saying. If, if the value that cit takes, say some j, is actually less than or equal to N, then ith feature at time T corresponds to the jth landmark in the map.

Suppose I have 3 doors. Let us say that third feature, of first feature c and c1t is 3. Then the first feature corresponds to the third door.

(Refer Slide Time: 14:10)



And, when cit is equal to N plus 1, I do not know what it is. So, this is essentially the problem of the correspondence. Now, if I know the correspondence, right, there is a hardly straight forward adaptation of the Bayesian filter algorithm. And, if we look at the book, they have given you all the, the worked out the full example with extended Kalman filter on how to accommodate these correspondence values into your localization model.

So, basically the, the, the way you compute the belief, updates changes. So, you do this with respect to the, the observation, the features that you have computed and the distance to those features and so on. So, some kind of a triangulations is what you do to accommodate the measurements. The motion model, more or less stays the same. But then what happens if I cannot give you the correspondence? Here I am assuming somebody has given me the correspondence, in the, the first part.

If I cannot give you the correspondence, it is challenging. But, this is usually the case, right. I mean correspondence can rarely be determined with certainty. So, (corres), sometimes, sometimes, I will think, I will be thinking this is door 3. If the doors do not have numbers on them then I am doomed. So, I do not know whether it is door 1, door 2, or door 3. I do not know what it is. Or Bluetooth beacons without, I mean, not Bluetooth beacons; just some bouncing bombs of my ultrasound, then I really do not know what is actual identity of the landmark.

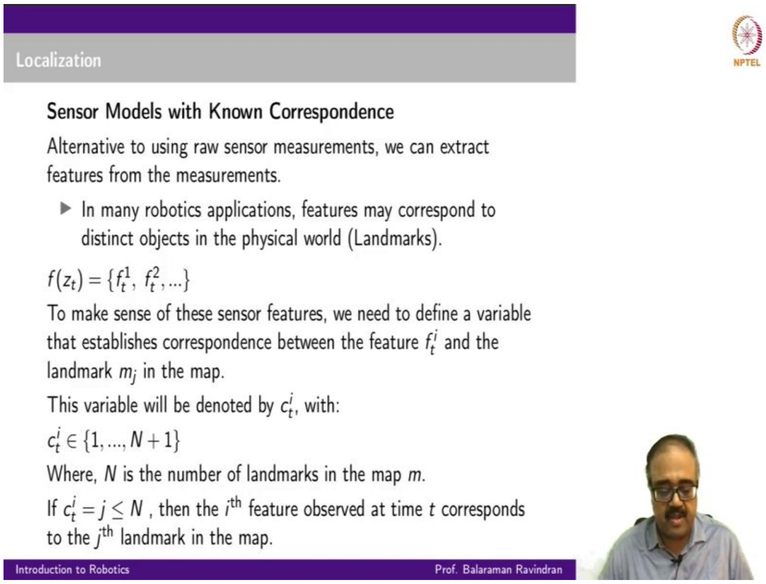
In such cases, you also have to estimate the values of the cit’s. So, you not only have to look at the value of X, but you also have to have some kind of an estimate for the values of cit. And, so, there are the multiple ways in which you can do it. You can also, you can have a distribution over cit and use that for updating your belief, bel Xt. Or you could estimate a point value, say, something like a maximum value estimate.

So, it is the most likely value of the correspondent and then you say that, this is the value; I am not going to look at the noise. And, then use that for making your updates. Now, you can see why kidnapped robot problem can become a reality. Suppose, I say, I think it is door 2, now correspondence assignment tells me it is door 2. But it is actually door 3. Or maybe I have made a mistake in my estimating the correspondence variable. I might have actually think I am in a very different part of the world for a few updates.

And, I might suddenly for sure know that it is door 2 then I know as far as the robot is concerned, 'hey, what? I was getting readings from door 3 all this while; suddenly I am getting a reading from door 2. I, do not know where I am?". So, you might want to account for these kinds of egregious errors. So, so, that is a reason why you look at the kidnapped robot problem as a special case of global localization.

So, this gets, it gets slightly more involved. Because you need to have a separate estimation procedure that is running for the correspondence problem as well. And, so once the, once you estimate, once have some kind of, weather point estimate or a distributional estimate for the correspondence, you can then use that in the previous algorithm that we spoke about in order to estimate the location, in order to update the belief actually.

(Refer Slide Time: 17:39)



So, notice that once I know the correspondence, it is going to affect how I go from my bel bar to bel. So, bel bar is the motion model. So, that gives me some noise in terms of the prediction that I am making. And, to correct the prediction, I am going to use these correspondence values and then map my location with respect to the known features of these landmarks. These landmarks, remember in the map, if I know that I am so far away from landmark X, and so I am 1 meter away from the door, then there is only certain part of the state I could be in. Because I know exactly where the door is.

So, the noise is in that 1 meter path, how far away I am from the door. So, these are essentially some small modifications that you make to the Kalman filter of the extended Kalman filter algorithm in order to accommodate this feature based map. But the bigger challenge is when we have to look at the unknown correspondence problem.