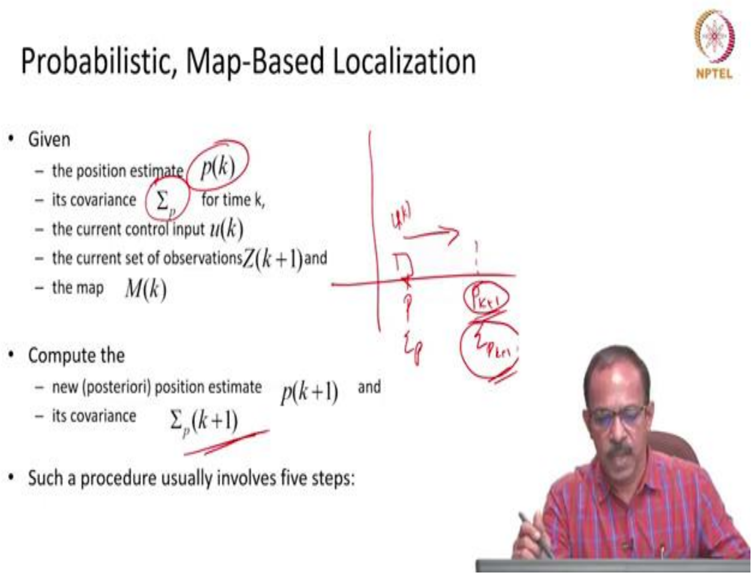
Wheeled Mobile Robots
Prof. Asokan Thondiyath
Department of Engineering design
Indian Institute of Technology, Madras

Markov Localisation
Lecture - 5.3
Map based localisation

(Refer Slide Time: 00:13)



Probabilistic, Map-Based Localization

- Given
  - the position estimate $p(k)$
  - its covariance Σ_p for time k,
  - the current control input $u(k)$
  - the current set of observations $Z(k+1)$ and
  - the map $M(k)$
- Compute the
  - new (posteriori) position estimate $p(k+1)$ and
  - its covariance $\Sigma_p(k+1)$
- Such a procedure usually involves five steps:

Hello everyone welcome back, so we will continue the discussion on Map based localisation. In the last class I explained the logic of having a map based localisation and how we do this map based localisation using a 2 step process, I mean 2 updates primarily a prediction update and a perception update.

So, in the prediction update we use the odometric information and update the position of the robot. And then at the new position the robot will use it is sensors to collect information from the surroundings and use that information to match it with the map information and then based on this they will actually update the position and it is covariance. So, that is basically known as a perception update.

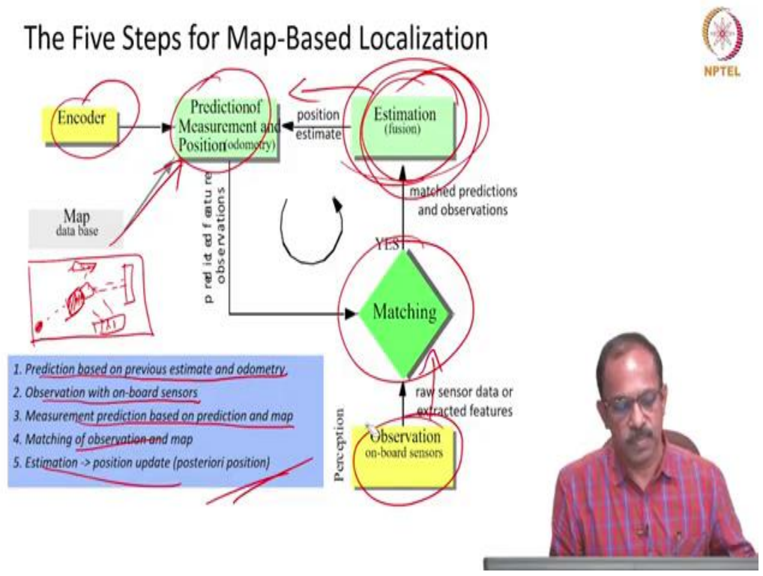
So, there are two steps perception prediction update and a perception update in map based localisation and I explained. How it works basically explaining using a scenario I explained how this actually works. So, in this lecture we will look at more it is mathematical formulations, how algorithms can be used for this mapping matching of the information and then updating the position.

So, just to give you update that what we discussed in the previous class. So, we have a position p(k) the previous position estimate and it is covariance given. So, the robot is initially at one location and it is position p(k) and it is covariance Σ_p is known, so p and Σ_p is known.

And from this position robot is given a control input u k and it moves to the new position p(k) + 1. So, this is the p(k+1) and it will be having a covariance $\Sigma_p(k+1)$, so that is basically this one ok. So, this is the new estimate that we need to make ok.

So we finally need to have the position p(k+1) and $\Sigma_p(k+1)$ and to do this to estimate this position p(k+1) and $\Sigma_p(k+1)$. So finally, when we say localization we want the robot to know it is position and the uncertainty at that position. So, this is basically the localization requirement and if you have to do this we need to go through the five steps which I already mentioned in the one of the previous classes.

(Refer Slide Time: 02:50)



The Five Steps for Map-Based Localization

1. Prediction based on previous estimate and odometry
2. Observation with on-board sensors
3. Measurement prediction based on prediction and map
4. Matching of ~~observation and~~ map
5. Estimation -> position update (posteriori position)

Encoder
Prediction of Measurement and Position (odometry)
position estimate
Estimation (fusion)
Map data base
predicted feature observations
matched predictions and observations
YES
Matching
raw sensor data or extracted features
Perception
Observation on-board sensors
NPTEL

But actually you can see here the five steps are basically you have a prediction based on the previous estimate and odometry. So, first you use the encoder information and the previous position and get the new position predicted and then you observe with the onboard sensors.

So, you have observation, so you observe with the onboard sensors collect the information and then use the map information and use this information to match the whatever you observe. Whatever is observed is match so a measurement prediction based on the prediction and map.

So, what actually it says that at a particular position where the robot has moves robot will predict what it can see and then it will see what actually it is seeing and then match these two and then based on the fusion estimate the new position and then get a new position estimate. So, that is our how you get the p(k+1) and Σ_p(k+1).

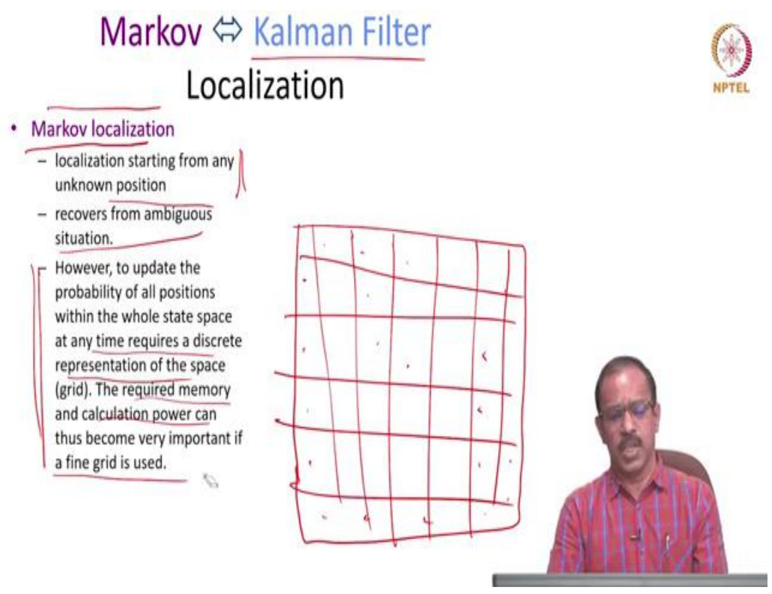
So, we need to have these five steps in doing this. The challenge is basically how do we actually fuse this information? So, you now suppose the robot is in one location, suppose this is the arena and the robot has started from here suppose there are some features here. So, the robot has actually moved to here ok, the new position and its estimate.

Now, based on the map information the robot says that you will be able to see something here, you will be able to see something here and you will be able to see something here. So, that is what actually the robot will predict based on the observation; based on the map information.

If it is at this position it should be able to see these things and then the robot will actually use it sensors to see what actually it is seeing and if they are same what it is seeing and what actually it is supposed to see both are same. That means, the predicted position is the correct position the position the robot is reached is the correct position there is no error.

If there are they are different there is some difference between what actually the robot is seeing and what did you suppose to see there is an error and that error can be used through some fusion algorithms to correct it is position and get a new position estimate which is the p(k+1) and Σ_p(k+1). So, this is what actually we need to do in the estimated in the localization using my best methods.

(Refer Slide Time: 05:23)



Markov ⇔ Kalman Filter Localization

- Markov localization
  - localization starting from any unknown position
  - recovers from ambiguous situation.
  - However, to update the probability of all positions within the whole state space at any time requires a discrete representation of the space (grid). The required memory and calculation power can thus become very important if a fine grid is used.

So, there are multiple algorithms and multiple methods available for this, we will not go through all the method. But we will just discuss about the two methods; one is known as the Markov localization, the other one is the Kalman Filter based localization. So, these two are the most commonly used localization methods which actually uses the map information which uses the map information and the sensor information to localize.

So in Markov Localization can see here, the localization is starting from an unknown position. So, it can actually start from any position the robot need not know where it is currently, it can be anywhere in the arena ok, we have a room or some other place where the robot is working. So, we want the robot to localize itself you need not know its initial position, so completely unknown. So, the robot will believe that it could be anywhere in this arena and it can start localizing based on that information.

And then it recovers the from any ambiguous position, since the robot need not know it is current position. But the challenge here is that when we have this kind of a situation what we normally do is that we actually suppose we have a room like this and we want to localize the robot in this room, what we do is we actually divide this into multiple grids ok.

You can actually have any number of grids and then say that the probability that the robot is in any grid is equal in the beginning that is the robot could be here, here, here or anywhere. So, it is an equal probability suppose there is 100 grids we will say it is a 0.01

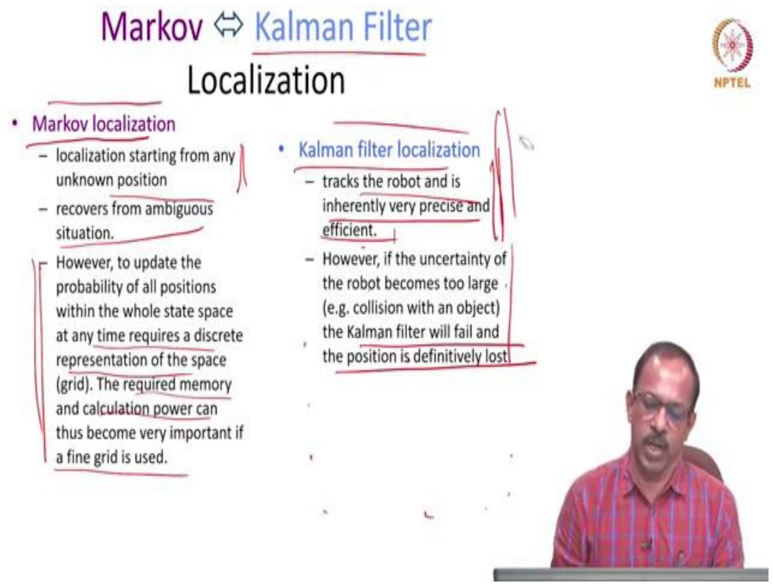
probability is 0.01 and total probability will get 1. So, you will say that it is 0.01 and all the grids.

Now, the robot is given a ah control input u t, now based on that control input it will update what will be the possibility that the robot is here, here or here whatever it is. So, it will update the probability of all the grids once it robot starts moving and then it will look which one has got the highest probability, based on that it will find out what will be the approximate position of the robot.

So, at every instant we need to update the probability of all the grids in this method. So, there is one of the difficulties with Markov localization, but it is it suppose the advantage is that we do not need to know where the robot is, even if the robot lost its position somehow or we the robot is not completely uncertain about it is position still it can start again and then recover its position. So, that is the advantage.

So, this particular one since it you have to update it every time. So, the memory and calculation power can become very important and if fine grid is used. So, if you are using very fine grids then the update will become very difficult, because you need to have lot of memory and lot of computation power. So, that is one difficulty with the Markov based localization.

(Refer Slide Time: 08:42)



Markov ⇔ Kalman Filter Localization

- Markov localization
  - localization starting from any unknown position
  - recovers from ambiguous situation.
  - However, to update the probability of all positions within the whole state space at any time requires a discrete representation of the space (grid). The required memory and calculation power can thus become very important if a fine grid is used.
- Kalman filter localization
  - tracks the robot and is inherently very precise and efficient.
  - However, if the uncertainty of the robot becomes too large (e.g. collision with an object) the Kalman filter will fail and the position is definitively lost

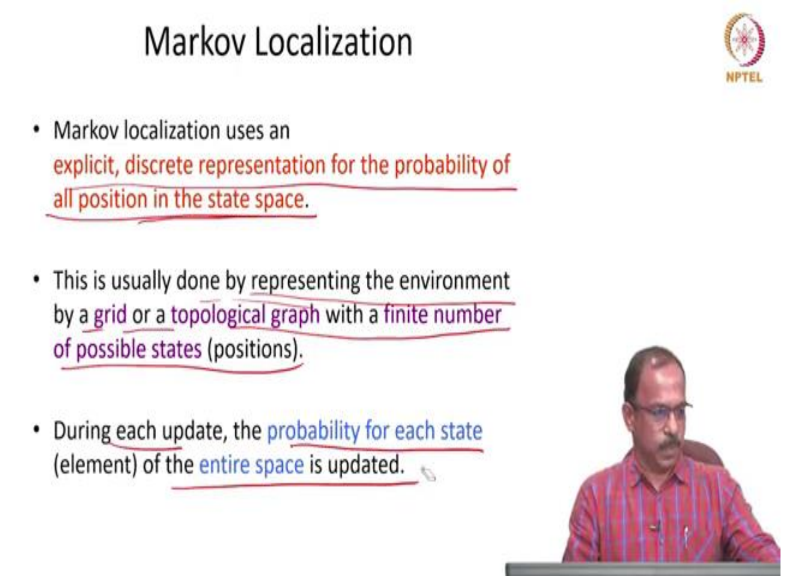
So, we will see how the Markov based localization works. But before that I will just tell you what is Kalman Filter Localization also. So, Kalman Filter Localization again a map based localization method, but it tracks a robot and it is inherently very precise and efficient.

It will be very it is a very precise and efficient method the only thing is that we need to know where the robot is currently. If you have to implement the Kalman Filter Localization the robot should know it is current position and it is uncertainty and then from there it can actually starts.

A completely unknown position the robot will not be able to start the localization. So, that is the difficulty with the Kalman Filter. So, if the uncertainty of the robot becomes too large, then the Kalman Filter will fail and the position is definitely lost. So, you will not be able to recover the position of the robot. So, that is basically the Kalman Filter Localization.

So, this is very efficient and practical implementations are there for Kalman Filter Localization. But Markov Localization works for some applications where you do not have too much of fine grid and still you will be able to computation power is sufficient to get the localization done, then Markov is a good option. But Kalman Filter is the most commonly used or you will see lot of literature on Kalman Filter based localization ok.

(Refer Slide Time: 10:07)



Markov Localization

- Markov localization uses an explicit, discrete representation for the probability of all position in the state space.
- This is usually done by representing the environment by a grid or a topological graph with a finite number of possible states (positions).
- During each update, the probability for each state (element) of the entire space is updated.

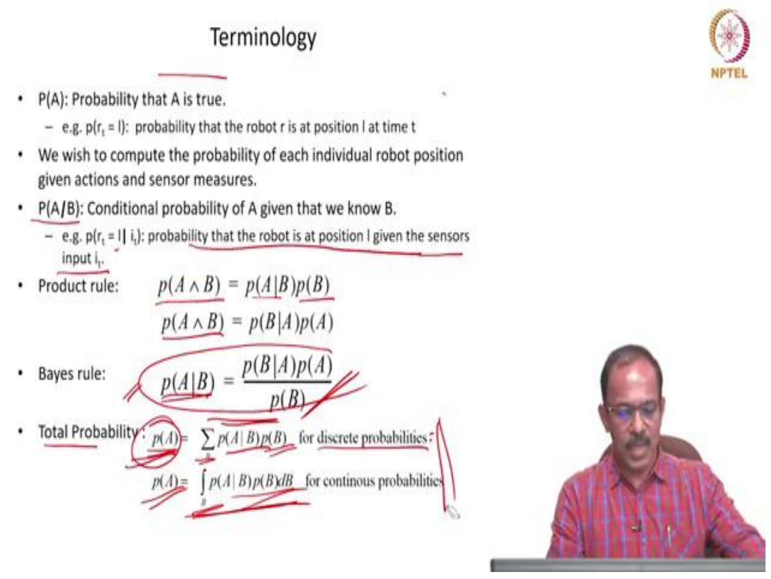
So, let us look at the Markov Localization. So, as I mentioned it uses an explicit discrete representation of the probability of all position in the state space. So, state space basically if you have an x y thη has the robot position and then for every grid you need to have the x y and thη for that grid it is probability of that; it is probability for that grid.

So, the probability of all position in the state space is needed with the discrete representation. So, this is done by representing the environment by a grid or a topological graph with finite number of possible states. So, you will have finite number of possible states like x y thη and then you will have grids which grid or a topological graph.

And then you look at each grid and then see what is the probability that the robot is at that location and what is the probability of it is x position y position thη position and then update it at every instant and then keep on doing this and try to find out at every instant which one gives you the highest probability. So, that is the Markov Localization method.

And then during each update the probability of each state of the entire space is updated. So, the entire space entire grid need to be updated and for each state. So, you will be having x y and thη and then for each grid you need to update all this information. So, it becomes bit complex to practically implement, but still it is a proven method of localization and the advantage as I mentioned that any ambiguous position the robot will be able to recover ok.

(Refer Slide Time: 11:53)



Terminology

- P(A): Probability that A is true.
  - e.g. $p(r_t = l)$: probability that the robot r is at position l at time t
- We wish to compute the probability of each individual robot position given actions and sensor measures.
- P(A/B): Conditional probability of A given that we know B.
  - e.g. $p(r_t = l | i_t)$: probability that the robot is at position l given the sensors input i_t.
- Product rule: $p(A \wedge B) = p(A|B)p(B)$
  $p(A \wedge B) = p(B|A)p(A)$
- Bayes rule: $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$
- Total Probability: $p(A) = \sum_B p(A|B)p(B)$ for discrete probabilities
  $p(A) = \int_B p(A|B)p(B)dB$ for continous probabilities

So, to understand the Markov Localization we need to have a bit of background in probability. So, I hope most of you are having this background, but just want to tell you, so these are the terminology that we may be using. So, P(A) is the probability that A is true and P(A|B) it is the conditional probability of A given that we know B. So, that is known as P A by B that is the conditional P(A|B) ok.

So, probability that the robot is at position I given the sensor input it is given that $p(r_t = i_t)$ where we have probability that the robot is at position i given the sensor input i_t. So, we know something and based on that we can actually get the probability. So, that is known as the conditional probability.

And then there is a product rule and the product rule is basically a union P(A|B) P(B) and this Bayes rule is P(A|B) that is conditional probability of a given B is known can be obtained as P(B|A)P(A) and P(B) by divide by P(B) is the Bayes rule.

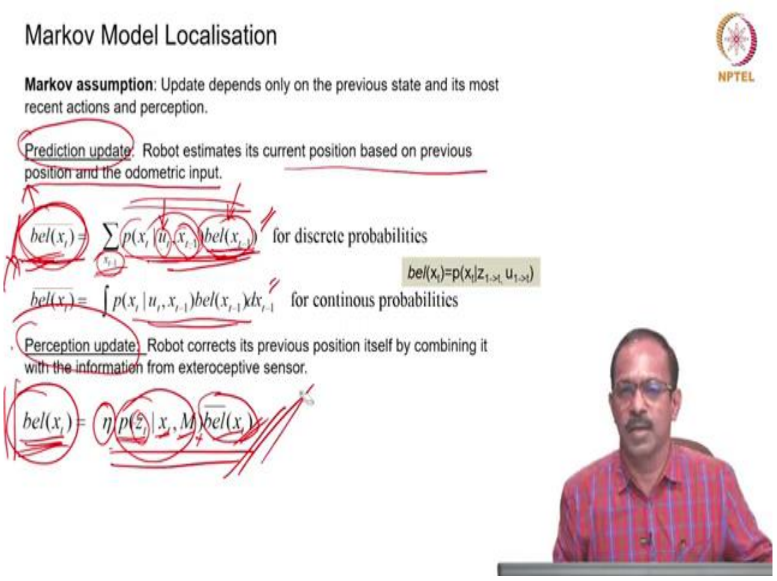
So, this Bayes rule will be used in localization, because we will be using the conditional probability that the probability that the robot is at a particular position given a control input or a map input. And if we know that we will be able to use that one information and the Bayes rule in order to calculate the conditional probability.

So, we will be using the Bayes rule for estimating the robot position and the total probability is given as P(A) that the robot is A is P(A|B) given P(B)*P(B) for discrete probabilities.

So, you can actually multiply these probabilities to get the total P(A) that is probability conditional P(A|B)P(B) will give you the total probability A. So, that is the that is for discrete and if it is for continuous you can go for the integration of P(A|B) to P(B) and P(B) will be the total probability.

So, in case the probabilities are given in discrete manner, then you can actually use this multiplication and then the summation over B Σ or you can do the integration or B and get the total probability. So, this is we will be using the total probability principle and Bayes rule in getting the map based localization done.

(Refer Slide Time: 14:53)



Markov Model Localisation

Markov assumption: Update depends only on the previous state and its most recent actions and perception.

Prediction update: Robot estimates its current position based on previous position and the odometric input.

$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t \mid u_t, x_{t-1}) bel(x_{t-1})$ for discrete probabilities

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

$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$ for continous probabilities

Perception update: Robot corrects its previous position itself by combining it with the information from exteroceptive sensor.

$bel(x_t) = \eta\, p(z_t \mid x_t, M)\, \overline{bel}(x_t)$

So, let me take a example of Markov Model Localisation. So, we do this Markov localisation is on two steps as I mentioned we will be having a prediction update and then having a perception update. So, prediction update is basically to get the $\overline{bel}(X_t)$ that is the position of the robot or the belief state of the robot before getting the sensor input that is the prediction update.

The robot estimate it is current position based on previous position and the odometric input. So, based on the previous position and the control input or odometric input, what is the new position that is basically the prediction update. So, the prediction update is the $\overline{bel}(X_t)$ can be obtained as total probability of the robot being at X_t provided U_t is given and the previous position X_t minus is known and the previous belief state X_{t-1} is also known.

So, this is the belief state that is the probability that the robot is at X_{t-1} is known and then the probability that the robot will be at X_t provided the given the control input U_t and from X_{t-1} from the previous position. If we know this probability and then we know this belief state and our all possible X_{t-1} if you sum this, then you will be getting the prediction update. So, that is basically the prediction update without any control without the sensory input.

So, what we do is we look at the possibility that the robot has moved to X_t from it is previous position X_{t-1} using a control input U_t. And then what was the probability that the robot was at X_{t-1} the previous position and you multiply this and for all possible X_{t-1}, then you will get the belief state X_t or X_{t-1} which is the one prediction updates ok.

If it is continuous probability you go for the integration, it is the same thing only for discrete we use this or continuous we use the integration. So, what we are trying to do is to find out the new position estimate of the robot at X_t or the probability that the robot is at X_t and that is obtained by multiplying the probability that the robot has reached X_t with the control input U_t from it is position X_{t-1}, multiplied by the probability that the robot was at X_{t-1} in the previous instant.

So, these two if you do multiplication you will get the total probability $\overline{bel}(X_t)$. Now, you have the $\overline{bel}(X_t)$, so next one is to get the perception update. So, you have a $bel(X_t)$ now that the robot is at X_t is the probability is known. Now, use the sensor information and update it and that is obtained by $bel(X_t)$.

So, $bel(X_t)$ is basically $\overline{bel}(X_t)$ that is the previous estimate position. And then observation Z_t from X_t, that position X_t given the map information multiplied by the previous belief state will give you the new belief state ok. So, that is how you get it sorry yeah.

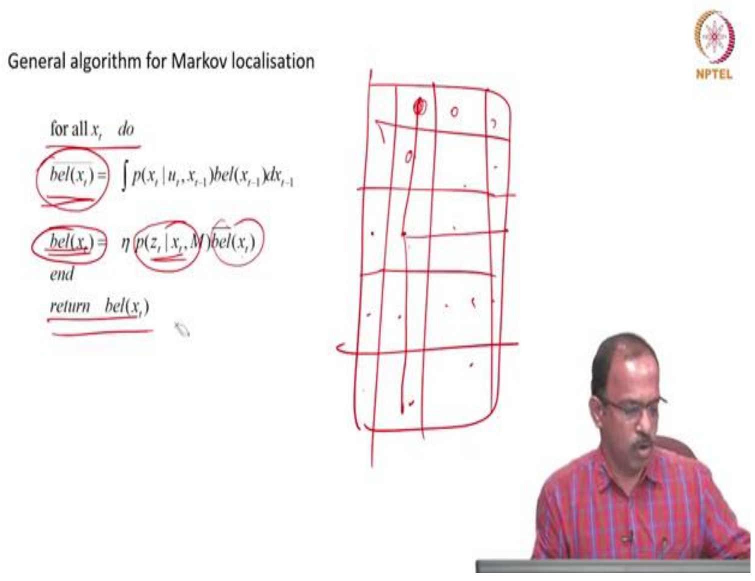
So and then there will be an η, so this η is basically a multiplication factor to ensure that the belief state or the total probabilities 1, otherwise β η does not have any significance. But it is used to make sure that the total probability is 1, because now we have a $bel(X_t)$calculated from the control input.

So, that is used here that multiplied by this probability of the based on map information at from X_t the robot is seeing the data or the robot is getting the feature information Z_t. These two multiply these two probabilities are multiplied to get the new belief state that is the $bel(X_t)$.

So, this is what actually we do in the Markov localisation. So, first step is find out all the probabilities that the robot is at X_{t-1} that is $bel(X_{t-1})$, then multiply that with the probability that the robot is at X_t when a control input is given to the robot. So, that will be the $\overline{bel}(X_t)$.

And then from that belief state then you multiply that with the sensor information or the probability that the robot is seeing an object Z_t from X_t from the current location X_t and that will give you the new $bel(X_t)$. So, this is basically the Markov model localisation. I will take an example and I explained this to you, so it will be much more clear to you how do we actually get this probability estimate or the belief state ok.

(Refer Slide Time: 20:04)



General algorithm for Markov localisation

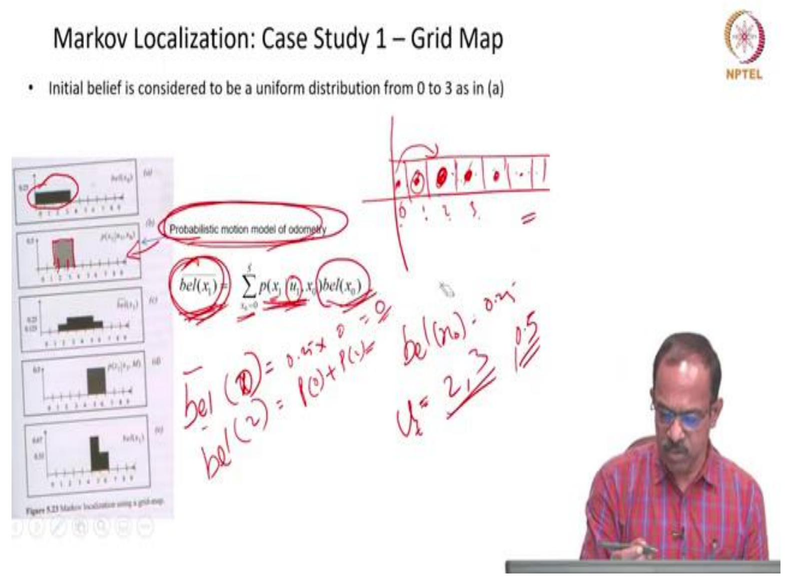
```
for all $x_t$ do
    $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$
    $bel(x_t) = \eta\, p(z_t \mid x_t, M)\, \overline{bel}(x_t)$
end
return $bel(x_t)$
```

So, general algorithm is for all X_t, so when you have multiple grids. So, you have multiple grids, so these are all possible X_t. So, the robot maybe could be anywhere here for all X_t find out the $bel(X_t)$ that is the prediction update. So, from here if the robot is here or here or here or here wherever it is and it is given a control input of 2 to move forward. So, it can actually move to here or it can move to here or it can move to here or it can move to here.

So, for all these X_t you find out the $\overline{bel}(X_t)$. So, you find out the possibility that the robot is here or here or here you find out all the possibilities because of the control input. And then based on the information collected from the sensor you recalculate the belief state, because now this is known and this will be given by the sensor.

So, that probability multiplied by the belief state will give you the $bel(X_t)$ and then that will be the return $bel(X_t)$. So, this is how you implement the algorithm for Markov localisation.

(Refer Slide Time: 21:19)



Markov Localization: Case Study 1 – Grid Map

- Initial belief is considered to be a uniform distribution from 0 to 3 as in (a)

NPTEL
Probabilistic motion model of odometry
$\overline{bel}(x_1) = \sum_{x_0=0}^{3} p(x_1 | u_1, x_0) bel(x_0)$
$bel(x_0) = 0.25$
$u_1 = 2, 3$
0.5
$\overline{bel}(0) = 0.25 \times 0 = 0$
$\overline{bel}(2) = P(0) + P(1)$
0 1 2 3
Figure 5.23 Markov localization using a grid map.

So, I will take a simple example before we close this session, I will take an example and then explain to you how we can actually use the Markov Localization for a practical case. So, I will take a very simple case where the robot is actually moving in a one dimensional grid.

So, assume that this is the grid where the robot is moving. So, I have actually divided this into multiple and I said that the initially the robot could be anywhere here, so these are the four initial states I believe that it is could be anywhere here in 0 1 or 2 0 1 2 3 grids.

So, I will give an equal probability of 0.25, so as that we can see here an equal probability of 0.25 for the belief state. So, the $bel(X_0)$ that is before we start the $bel(X_0)$ = 0.25 for all the grids, so it could be anywhere in the 4 grids. Assume that the robot is moving forward and we say that the robot is given a control input. And the sensor says that yes I can actually measure the how much is the moment I have made.

But when the robot say the sensor says that I am not sure how much it is exactly, but it could be 2 or 3. So, I say that my measurement says that the U_t could be 2 or 3 with equal probability, I would have moved 2 steps or 3 steps I do not know the robot does not know and that probability is equal probability of 0.5. So, there is an equal probability that the robot has moved 2 steps or 3 steps and that is given by the motion model.

So, we say that it could be 2 steps or 3 steps moved forward with an equal probability of 0.5, so that is the $P(x_1|u_1x_0)$. So, that is from x_0 control input u_1 is given and that could have moved 2 steps or 3 steps forward. So, the robot does not know how much exactly but it could be 2 or 3.

Now, so we have 2 information one is basically the $bel(x_0)$ we already have the $bel(x_0)$and then we have the probability that the robot has moved 2 or 3 it is also given here, from the x_0 with a control input the robot could have moved 2 or 3.

So, x_1 that is the position that the robot could have reached with the control input u also is given $P(x_1|u_1x_0)$. So, that is the probabilistic model with the probability of 0.5 could have moved 2 steps or 3 steps from $x_0$with a control input. So, that is basically the motion model for odometry.

Now, with these two we need to calculate what is the prediction update? So, the prediction update is basically. So, we have now x_0 could be from 0 to 3. So, it could be 0 ,1, 2, 3, so $x_0$to 0 to 3. So, it could be then initially at 0 1 2 or 3 we do not know. So, from 0 to 3 $P(x_1|u_1x_0)$. $bel(x_0)$ if you get this total probability that is basically the prediction update $bel(x_1)$.

So, what we are doing? We are finding out suppose the robot was in 0 and it has move 2 what will be the probability, where will be the robot or just moved 3 where will be the robot. Similarly if the robot was initially at 1 and it has moved 2 what will be the probability that the robot is there location.

So, like this we will be able to we need to find out the probability of all the grids and find out what is the actual probability. So, this is basically what we are trying to do ok. So, what we do? We look at ok, now assume what is the probability that the robot is at 0, so that is the belief.

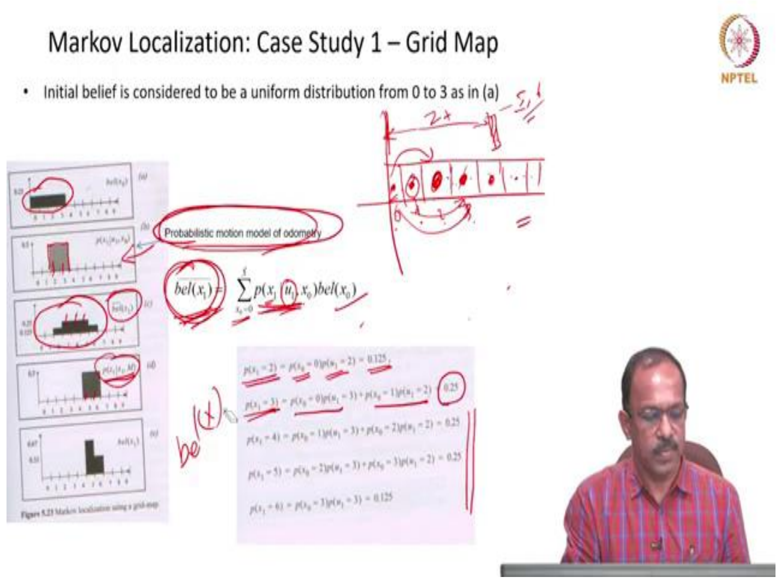
Now, I want to predict the update. So, what is the probability that the robot is at 0? And we know that the robot cannot be at 0 because the probability that the robot was initially at 0 is 0.25 ok. And it has there is no way that the robot could have been 0, because it already moved 2 or 3. So the robot cannot be at 0, therefore the probability here will be 0 the robot 0 grid is 0.

Similarly, what is the probability that the robot is at 1? Initially if the robot was at 0 so there are what are the ways in which the robot can actually reach 1? So, there is no way that robot can reach 1, because if it is initially at 0 it would have moved to 2 or 3. And if it is initially at 1 it would have moved to 3 or 4 because of 2 or 3 steps. So, the probability that the robot is at 1 is also 0.

Now, what is the probability that the robot is at 2. So, the only probability is that the probability that the robot was at 0 plus the probability that the robot has travelled 2 units only the way it can be in 2. So, the probability of robot being at 2 is initially it was at 0 and then the robot travelled 2 steps that is the only condition under which you can actually get the robot and grid 2.

So, we will try to find out what is that probability. Similarly, we tried to find out what is the probability that the robot is at 3. What is the probability that the robot is at 4 and like that we will try to out find out and find get the probability estimates. So, this is what actually we do to get the estimate.

(Refer Slide Time: 27:15)



Markov Localization: Case Study 1 – Grid Map

- Initial belief is considered to be a uniform distribution from 0 to 3 as in (a)

Probabilistic motion model of odometry

$$\overline{bel}(x_1) = \sum_{x_0=0}^{5} p(x_1 \mid u_1, x_0)\, bel(x_0)$$

$p(x_1=2) = p(x_0=0)p(u_1=2) = 0.125$

$p(x_1=3) = p(x_0=0)p(u_1=3) + p(x_0=1)p(u_1=2) = 0.25$

$p(x_1=4) = p(x_0=1)p(u_1=3) + p(x_0=2)p(u_1=2) = 0.25$

$p(x_1=5) = p(x_0=2)p(u_1=3) + p(x_0=3)p(u_1=2) = 0.25$

$p(x_1=6) = p(x_0=3)p(u_1=3) = 0.125$

Figure 5.23 Markov localization using a grid-map

So, you can see here now the probability that the robot is at 2. So, the probability that x_1 is equal to 2 is that the robot is x_0 is equal to 0 multiplied by the probability that control input was 2. So, that was the only condition under which it can be 0 in 2 the robot could be at 2.

So, you will see that the probability is 0.125. Similarly, the probability that the robot is at 3 grid 3 is initially it was at 0. And then travelled 3 units or initially the robot was at 1 and travelled 2 units these are the only two possibilities that the robot could be at 3.

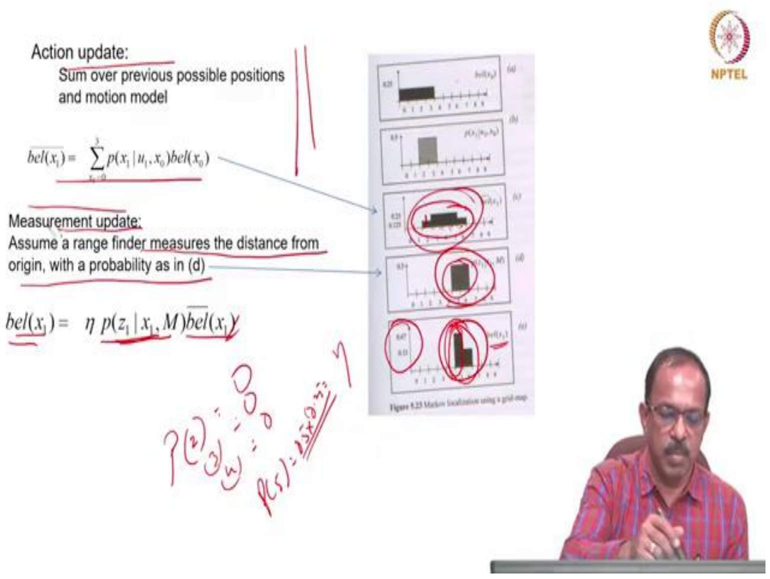
So, we will find out this plus this x_0 = 0, u_1 = 3, x_0 = 1, u_1 = 2. So probability is 0.25. Same way you calculate the all the probabilities and then you will see that this will be the distribution of probability. So, now $bel(x_1)$ the prediction update says that the robot could be in 2, 3, 4 or 5 or 6 with the large high possibility that the robot could be at 3, 4 or 5. So, the that probability is 0.25 others are 0.125, so that is $bel(x_1)$.

Now, we assume that we have a sensor and the sensor is measuring some distance from the current position of the robot to this, it is measuring using a sensor. Assume that there is some sensor which actually measures the distance and then give that information as Z_t, says that from my position I can say that I could be at 5 or 6. So, I am actually measuring some distance from here to the starting position and I am actually seeing it as 5 or 6 units away from the starting position. So, that is the sensor information.

So, the possibility the probability that it could be 5 or 6 are equal, so you have 0.5 probability. So, P(z_1| x_1,M) given the map information and the sensor data z_1 says that it could be 5 or 6 away from the standard with the probability of 0.5.

Now, what we do? We use this information and this information and then find out the total probability for the robot to be in these locations ok. So, that is what actually the next step where we calculate the belief state. So, we want to find out the $bel(x)$ now.

(Refer Slide Time: 29:51)



Action update:

Sum over previous possible positions and motion model

$$\overline{bel(x_1)} = \sum_{x_0=0}^{3} p(x_1 \mid u_1, x_0) bel(x_0)$$

Measurement update:

Assume a range finder measures the distance from origin, with a probability as in (d)

$$bel(x_1) = \eta\, p(z_1 \mid x_1, M)\overline{bel}(x_1)$$

So, what we do? We do the action update ok. So, the action update is this $bel(x_1)$ is there. So, the measurement update, so action update we already saw. So, the measurement update is a range finder measure the distance from the origin with the probability as in d, so this is the probability 0.5.

Now, we find out the total probability $bel(x_1)$ as P(z_1| x_1,M) multiplied by $bel(x_1)$. So, for every probability here p 2, 3, 4, 5, 6 we find out what is the probability. So, we will try to find out what is the probability that robot is at 2, because it was 2.

Now, we say that this $bel(x_1)$ the probability we I know what is the probability it is 0.125. But this one the probability is 0, because it has already only 5 and 6 has got 0.5 all these are 0. So, 2 will be 0, then 3 will be 0, then 4 will be 0 because 5 and 6 only available.

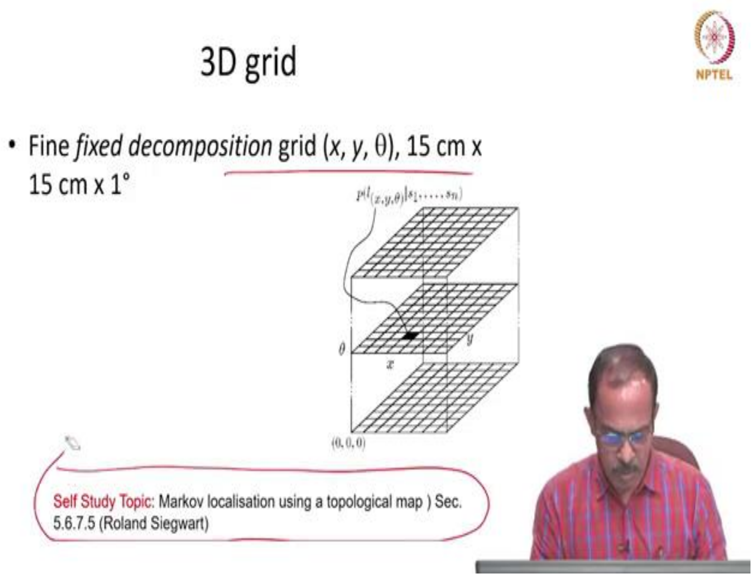
Now, we look at these 5 and 6 what is the probability that. So, this probability that the robot is at 5 will be this 0.5 multiplied by 0.25 so that will be the probability. Similarly, at 6 also you will be able to get the probability and then if you do this you will see that the $bel(x_t)$ belief at the final belief state of the robot will be like this ok.

So, now if you do 0.5 multiplied by 0.25 it will be a small number. So, we will actually multiply with an η, so that we will be actually getting the total probability as 1. So, that is why we use the η multiplication factor and then we find that which is the highest probability. And then now we can say that the robot could be in 5 with a very high probability. We can say that the robot could be at 5; the only other possibility is that it could be at 6.

So, from this initial uncertainty the uncertainty has come down and it has become much more clear that it could be at 5 or 6, then 2, 3, 4, 5 or 6. So, this is the way how we actually update the information update the localization information using the measurement update.

So, the prediction update actually showed lot of uncertainty, but a measurement updates reduce the uncertainty and gave you much better estimate of the robot position and its uncertainty. So, the position and uncertainty are actually given this. So, the position and uncertainty are actually given this. So, this is the basic principle of Markov Based Localisation.

(Refer Slide Time: 32:30)



3D grid

- Fine *fixed decomposition* grid (x, y, θ), 15 cm x 15 cm x 1°

$p((x,y,\theta)|s_1,\ldots,s_n)$
θ
x
y
$(0,0,0)$

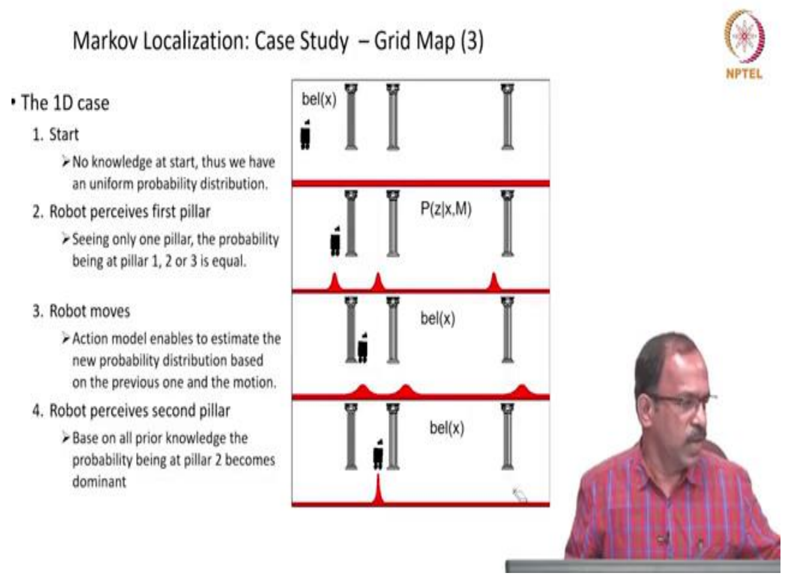
Self Study Topic: Markov localisation using a topological map) Sec. 5.6.7.5 (Roland Siegwart)

So, this can be applied for 3D grid also. So, if you have an x y θ and 15 by 16 15 by 15 grid, then we can see that you will need to have a large grid and then you need to update this x y θ for every grid and then do this computation.

For every moment you need to compute the probability for every grid for every state and update the database and find out which one has the highest probability that will be the position of the robot or that much distribution you will be able to find out. So, that is the basic principle of Markov localisation ok.

So, there are lot of literature available and the textbook also there are lot of discussion on Markov Localisation using topological map. So, you can refer to this and then if you are interested you can go through literature and a lot of resources are available for you to study this ok.

(Refer Slide Time: 33:34)



Markov Localization: Case Study – Grid Map (3)

- The 1D case
  1. Start
     - No knowledge at start, thus we have an uniform probability distribution.
  2. Robot perceives first pillar
     - Seeing only one pillar, the probability being at pillar 1, 2 or 3 is equal.
  3. Robot moves
     - Action model enables to estimate the new probability distribution based on the previous one and the motion.
  4. Robot perceives second pillar
     - Base on all prior knowledge the probability being at pillar 2 becomes dominant

So, we will stop here we will discuss the Kalman Filter based localization in the next class.

Thank you.