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

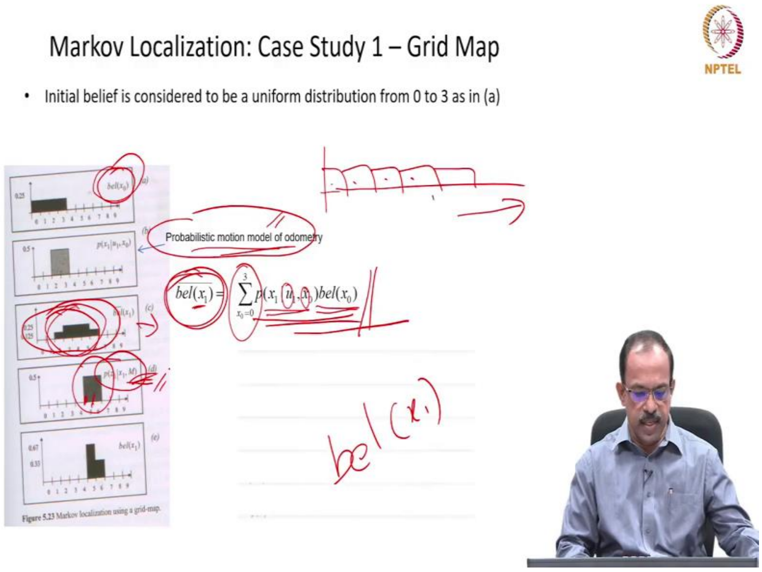
Lecture - 5.4
Kalman Filter Localisation

Hello everyone, welcome back. So, we are discussing the localisation methodologies, basically map based localisation where we have a probabilistic approach for localizing the robots. And we discussed that there are two methods; one is Markov localisation, the other one is Kalman filter based localisation. And all these both the methods use the five step process.

What we discussed in the last class, where we have a prediction update based on the encoder data and the previous position. And the prediction update will give you a mean position and the covariance of that estimate. And then we go for a perception update. In perception update, we will get the data from the sensors, the features on in the map will be collecting through sensors or the features in the environment will be collected using the sensors, and then this should be compared with the map data.

And then based on this matching of these features, we will update the position of the robot that is the perception update. So, the advantage of perception update is that the prediction update errors will be drastically reduced by their perception update. So, this is what actually happens in the map based localisation.

(Refer Slide Time: 01:37)



And we saw that a Markov localization is one of the methods used in mobile robotics where we try to divide the whole area into grids, and then we will try to assign equal probability for all the grids, so that the robot does not know where it is initially, and then try to update the probability of each grid as the robot moves.

So, when the robot is moving, it will try to update the probability that the robots in a particular grid using the prediction update and perception update. And then whichever grid has got the highest probability or the distribution of probability within the grids will be assessed and accordingly the robots position will be estimated, so that is what actually happens in Markov localization.

And we saw one example where I explained about the methodology used in Markov localization. For example, if you have a one-dimensional travel a grid in one dimension. So, you have many grids in this direction x-direction motion. And we assume that initially the robot could be anywhere in the 4 grids 0, 1, 2 3, 4 with an equal probability of 0.25.

And then we will consider the control input u which has got a value of 2 or 3 with equal probability that the robot say the encoder says that the robot has moved two-steps or three-steps with equal probability that means, there is an uncertainty in the movement of the robot. So, it has was moved two or three-steps and that is basically the motion model of odometry.

So, you have this initial probability $bel(x_0)$, and then here this motion model of odometry. And based on this previous $bel(x_0)$ and the motion model of odometry, we will estimate the new position which is $\overline{bel}(x_1)$ which is the prediction update by using this rule where we will try to add all the probabilities that the robot is at x_1.

And then this x_0 = 0 to 3 all the possible positions of the robot in the previous stage. And then we will try to find out the use conditional probability that the robot is at a particular x_1 when the control input u_1 is given from the initial position x_0, and the previous $bel(x_0)$. So, we will take this probability and find out what is the probability that the robot is at 0, 1, 2, 3, 4, 5, 6, or whatever the grid, so that is what actually we do.

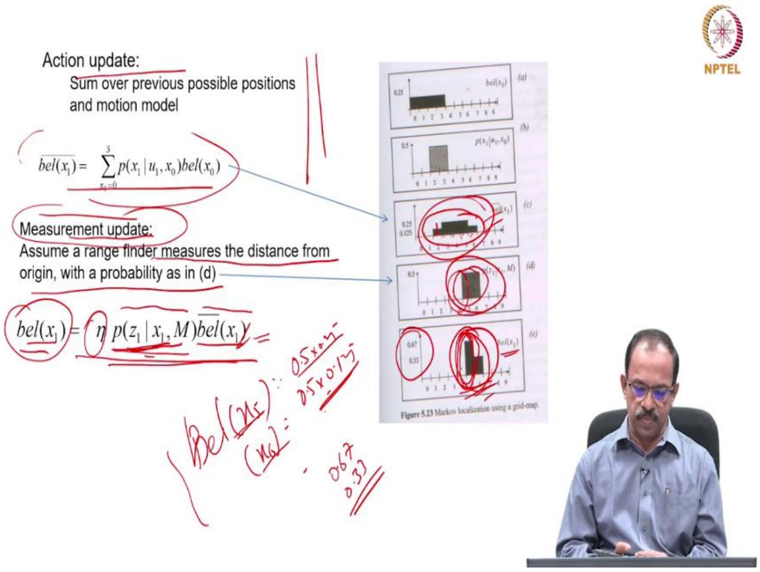
And when we do this, we will get a distribution like this. So, after the first movement from x_0 to x_1, the robot has got a estimate of its position in this format where actually it says that it could be in grid-2 with a probability of 0.125 and could be in 3, 4 or 5 with a in equal probability of 0.25 and it could be in point grid-6 with a probability of 0.125. So, that is a there is a large uncertainty in the position of the robots. So, that is the prediction update.

Now, we go for the perception update where we get the information from the sensor, sensor array, sensor will measure the distance from the starting point to the current position and then use that information to update the position. Now, the sensor also is not accurate. So, sensor will tell that it could be 4 units or 5 units away from the starting point.

So, there is an uncertainty. It says that it could be in four or 5th, 4th, sorry the 5th grid or the 6th grid based on the information. So, there is an uncertainty in the sensor measurement also, but still it says that ok this is the equal probability that you have. So, this is your sensor data or the map information collected from the sensors.

Now, using this perception update, perception data and the $bel(x_1)$, we will try to find out the actual belief state $bel(x_1)$, so that is what actually the perception updates. And this can actually be obtained by using these two probabilities the $\overline{bel}(x_1)$ and the perception or the sensor uncertainty, or the measurement model we can get use that one to get the probability estimated and that is what actually we do.

(Refer Slide Time: 06:15)



So, here we can see that. So, you can see that this is the $bel(x_1)$ and the measurement update we have. And then using the measurement update, we will be able to calculate the $bel(x_1)$. So, we will get the probability of the measurement, and the previous belief state or the $\overline{bel}(x_1)$, and get the total probability.

And find out what is the probability that the robot is at 2, 3, 4, 5 or 6 based on this. And then if you do this multiplication of probabilities $P(z_1|x_1,M)bel(x_1)$, you will be able to get the probability at these two locations 5 and 6. So, you can actually set the belief state the belief that the robot is at x_5 or x_6 can be obtained by this one. So, you have 0.5 the robot is at 5 is 0.5 from the sensor, and 0.25 from the prediction.

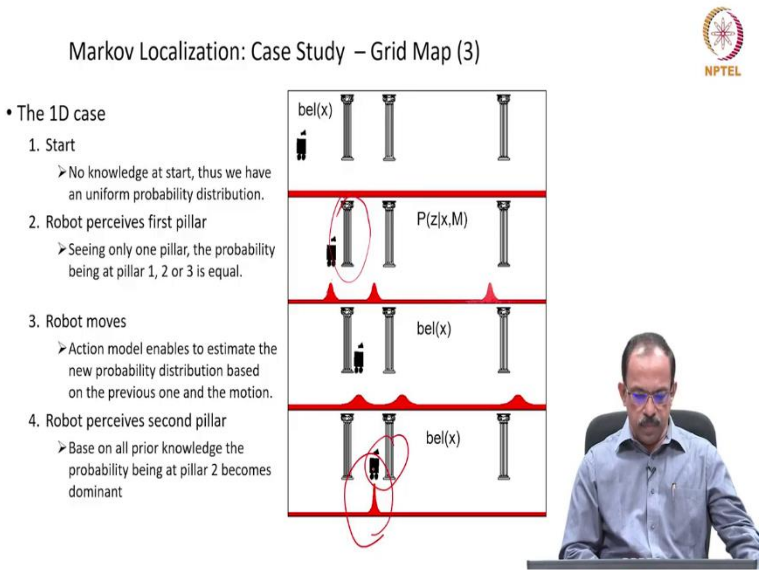
So, 0.5 into 0.12 0.5 into 0.5, so, that will be 0.25, sorry 0.5 into 0.25, so this is 0.5 into 0.25. And at 6, it is 0.5 into 0.5 multiplied by 0.125. So, this is the probability that the robot is at x_5 or x_6. Now, these two, if you add these two, it will not become 1. So, we need to have the total probability is equal to 1. So, if you use a eta a multiplication factor and if you do that multiplication factor, you will get it as 0.67 and 0.13, 0.33, so that is the way how you get the probability the robot is at 5 or 6.

Now, we know that after this perception update, the robot is much more certain that it is in 5, the robot is currently at the grid 5 rather than any other grid 0, 1, 2, 3, 4, or 7, 8, 9.

And there is a small probability that it could be at 6 also, but it is very high probability that it is at point 6, I mean at the grid 5.

So, you can see that the uncertainty of the position estimate from the prediction has drastically increased with a perception update. So, you will be getting that the belief the robot is at 5 is much higher than any other grid. So, this is the way how the perception update helps to increase the accuracy of localization. So, this is the Markov method of localization.

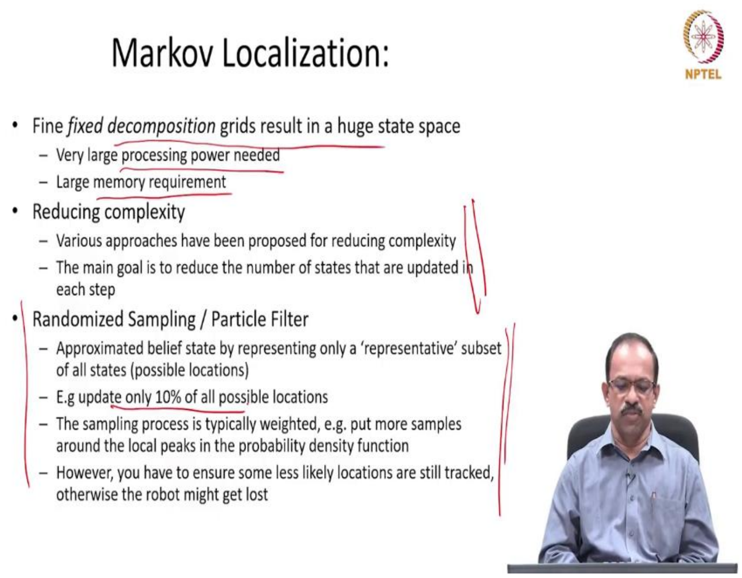
(Refer Slide Time: 08:53)



And I mentioned about the 3D grid also. So, now, you can see the previous example which I mentioned you can see that it actually uses the same principle here because so it seeing a image in a pole here and, so it increases the uncertain I mean increases the probability. And then again it sees here, it actually again comes here it increases the probability.

So, you will be able to get a very high probability that the robot is at this point. So, this is the way how the Markov localization helps to improve the prediction, and the actual localization of the robots using the prediction update and perception updates ok, so that is about the Markov localization.

(Refer Slide Time: 09:37)



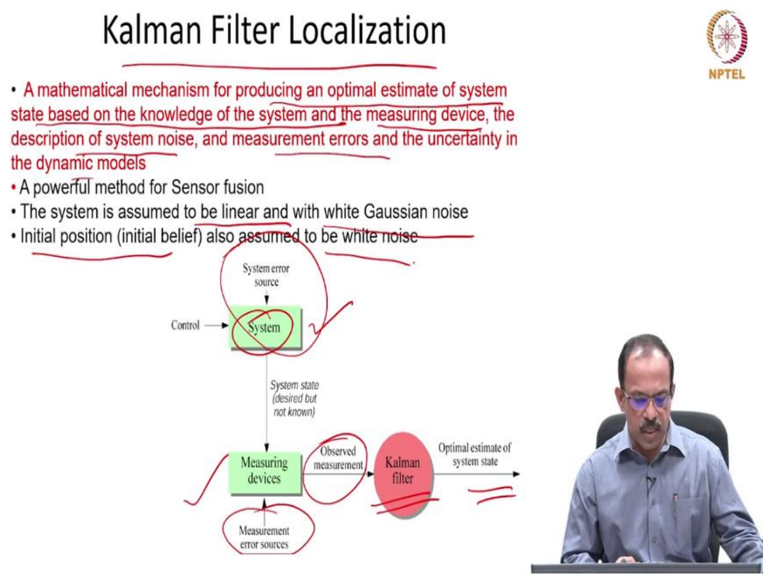
So, what we need to have is a fine fixed decomposition grids which actually results into a huge state space. So, when you have a fine grid and I mean large area and fine grid, and we have the state also is more number of states to be observed, then it becomes a very huge state space, so very large processing power and large memory. So that is the main problem with the Markov method of localization.

And there were many methods suggested in the literature for reducing the complexity, how to reduce the number of states, how to reduce the number of grids, and how to reduce the number of computations needed at each step. So, there are methods like randomized sampling, particle filter, etcetera, etcetera. So, we are not going into all those methods here.

You can, if you are interested, you can actually go through the literature and you can find out that. People have been trying to improve the Markov localization methodology using many strategies, and there are lots of papers in the literature about these methods. So, in some cases you know you update only 10 percent of all possible locations.

So, do not update all the locations, only 10 percent of all possible locations can be updated, so that is these are the strategies adopted by various researchers ok. So, that is about the Markov localization.

(Refer Slide Time: 11:03)



So, now let us look at the Kalman filter localization. So, the difficulties with the Markov is known that lot of computations needed. You need to update the all the grid at every update updates stage.

But the advantage is that you do not need to know the exact location of the robot, the robot could be anywhere. So, it can actually update its location wherever it is; or even if it is does not know the location, it can identify its location based on the updates that it gets through perception and prediction, prediction and perception.

So, to simplify that process over we do not need to go for that kind of a complex computation. So, the Kalman filter based localization was proposed. So, Kalman filter is a it is a very useful and highly utilized filter update in many fields it is not only in for localization, you will see that it is applicable in many other fields also.

So, it actually takes that the whole system and then we assume that there is a error in the system, and then you have a measurement from the so using the sensors an error using the measuring devices. And then it takes these two information from the system state and the observation, and then uses a filter to fuse these two information and grid the optimal estimate, so that is basically the principle of Kalman filter localization.

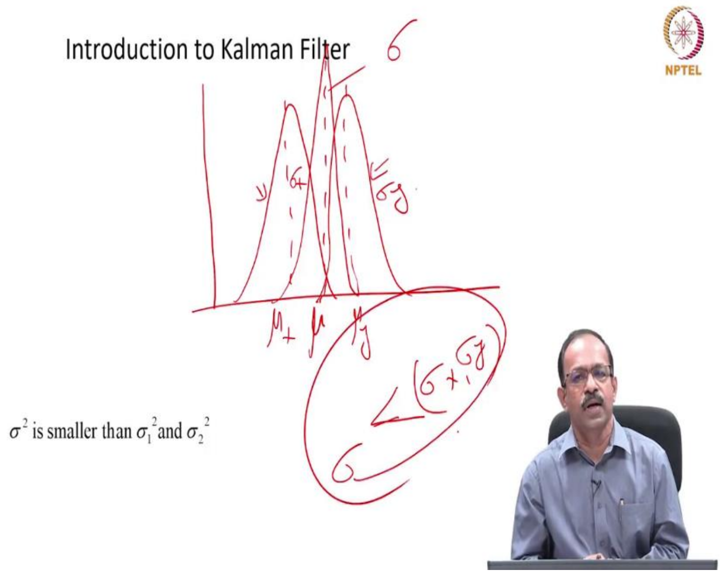
So, it is a mathematical mechanism for producing an optimal estimate of system state based on the knowledge of the system and the measuring device. So, you have the knowledge of the system and the measuring device, and to some extend we know the system noise, and the measurement errors, so that means, every system has got its own errors and the measurement system also has got its own errors.

So, we know these two the measurement errors and the uncertainty in the dynamic models. And once you know this, then you can actually fuse this information to get a better estimate of the system, so that is basically the principle. So, it is the system state can be estimated using a measurement and the previous the knowledge of the previous position.

So, the in this localization, what we assume that we know the previous position and its uncertainty and from there, we start with the new calculation. So, Kalman filter basically recreates the previous state and its uncertainty to do the updates. Now, how does it work ok? It is a powerful method for sensor fusion and this system is assumed to be linear. So, we assume that it is linear and with the white Gaussian noise.

So, we assume that the distribution of noise is white Gaussian that is a normal distribution which can be represented using a probability density function, so that is basically the understanding here. Now, what is the requirement? The initial position and that also is assumed to be a white noise. So, the uncertainty in the initial position also is considered to be a white noise. And the system is also to be assumed to be a white noise including the measurement system.

(Refer Slide Time: 14:41)



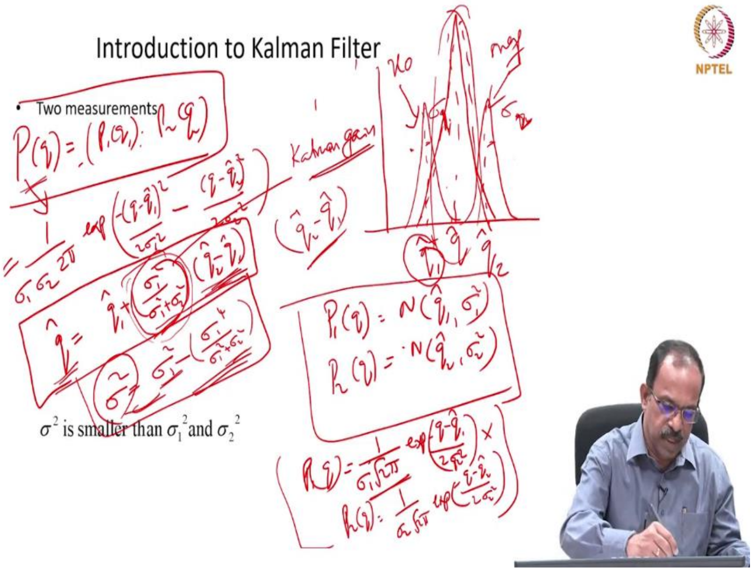
Now, what is the benefit of doing this, or how does it work? So, if you want to know the basic principle of the Kalman filter, we need to look at the way how it works. So, now suppose you have a is of the basic principle is that suppose you have two measurements; one is this one, and another is this one ok. So, we assume that this is one measurement. So, we say that this μ_x, and this is μ_y.

Now, this can say this distribution, so we have one distribution here and another distribution here. So, this is a x and y some distribution you have. And we have that this has got a σ_x and this has got a σ_y. So, these are the two distributions. Now, the Kalman filter strategy is that when you fuse these two, you will be always able to get a distribution which is more accurate and less uncertainty.

So, we can actually get a new mu by combining these two distributions, you will be able to get a new distribution where you will be getting a new σ, and this σ will be always smaller than σ_x or σ_y, so that is the basic principle. So, you can actually get a new estimate where the covariance will be smaller than the two of these two signals or the systems.

So, you have two distributions with its own mean and covariance. And if you combine these two using a Kalman filter, you will be able to get a distribution or an estimate based on these two which will be having a covariance which is less than the covariance of these two. So, you will be always getting a better estimate using this fusion of these two data, so that is basic principle of the Kalman filter. So, how is it coming or what is the mathematical foundation for that? We will just look into that.

(Refer Slide Time: 16:53)



So, as I mentioned, so let us consider these are two distributions. So, I will call this as q_1 and q_2. So, this is and this σ_q and σ_q σ_{q_2}, or σ_1 σ_2 we can call. This is σ_1; this is σ_2. So, it has got a standard deviation of σ_1 and σ_2, these are two σ_s.

So, let me write this $P_1(q)$ as $N(q_1, \sigma_1^2)$ is the covariance. So, we will write this q_1, q_2 as the mean value of this. And σ is the σ_1 and σ are the standard deviation. So, σ_1^2 is the covariance. Similarly, $P_2(q)$, so that the distribution the probability function can be written as $N(q_2, \sigma_2^2)$. So, that is the these two distributions what we have P_1 and $P_2(q)$.

Now, suppose we want to combine these two, then we can use the Bayes rule, and we can write it as P(q) = $P_1(q_1)P_2(q_2)$ that is the Bayes rule of this probability. Now, if we write this $P_1(q)$, so $P_1(q)$ can be written as the distribution can be written because we are assuming it is a white noise and the distribution. So, we can write this as $\frac{1}{\sigma_1\sqrt{2pi}} exp\frac{-q-q_1}{2\sigma_1^2}$

So, this way we can write $P_1(q)$, So, similarly $P_2(q)$ also can be written as $\frac{1}{\sigma_2\sqrt{2pi}} exp\frac{-q-q_2}{2\sigma_2^2}$. So, this is the distribution $P_1(q)$ and $P_2(q)$, so that is the assumption that we have it has a white noise distribution.

Now, this P(q), when you do this Bayes rule application, we will be getting it as, so we can actually multiply these two we will be getting it as 1 by ok. So, if you multiply these two, we able to write it as 1 by, so this can be written as $\frac{1}{\sigma_1\sigma_2\sqrt{2pi}}\exp(\frac{(-q-q_1)^2}{2\sigma_1^2}-\frac{(-q-q_2)^2}{2\sigma_2^2})$. So, we will be able to get this P(q) as this format.

Now, we know that P(q) is also assumed to be a white noise when you have the total probability that also will be a white noise. So, if that is the case, so if P(q) can be written in this format, then we can again assume that what is the corresponding σ of P can be obtained from this relationship. So, if you do that, we will be getting this as q that is the mean of this estimate P(q).

So, if you have a new estimate using these two, you can actually say that this is the q. So, this q, q can be obtained as $\hat{q}$ can be obtained as $\hat{q}_1+\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}(\hat{q}_1-\hat{q}_2)$ that is the new mean value of the estimate. So, we have two estimates initially P_1 and P_2. Now, a new estimate by combining these two, you will be getting it as P(q).

And the mean value of q is given as $\hat{q}_1$ plus that is this is $\hat{q}_1$ this plus some value, and that value is something multiplied. So, this thing is there this is multiplied with the difference of these two means, so that is $\hat{q}_2$ - $\hat{q}_1$. So, you have this $\hat{q}_2$ - $\hat{q}_1$ as the difference of the two means of the previous estimates – the two estimates, that multiplied by a factor and that is added to $\hat{q}_1$ will be the new q.

So, the q will be the new mean will be more than $\hat{q}_1$, and that will be a function of the difference of the $\hat{q}_1$ and $\hat{q}_2$ also. So, that is the new mean value for the estimate. And it is σ^2. So, variance is given as $(\sigma_1^2-\frac{\sigma_1^4}{\sigma_1^2+\sigma_2^2})$. So, this will be the new covariance of the estimate. So, the previous estimate is what σ_1 and σ_2 as the variance.

Now, σ the new variance will be $(\sigma_1^2-\frac{\sigma_1^4}{\sigma_1^2+\sigma_2^2})$. And this will be always a positive quantity. And therefore, you will be seeing that this will be always smaller than σ_1^2. So, it is a covariance will be always smaller than the σ_1^2 and the σ_2^2 also. So, both the new covariance will be always smaller than σ_1^2 or σ_2^2, and that is the basic principle of Kalman filter estimate.

And this factor is known as the Kalman gain. So, we call this $\frac{\sigma_1^2+\sigma\sigma_1^2}{\sigma_1^2+\sigma_2^2}$ is known as the Kalman gain. So, this is basically the Kalman gain. So, we call this as the Kalman gain ok. So, this is the principle of Kalman filter. So, we assume two distributions $P_1(q)$ and $P_2(q)$with its own mean value $\hat{q}_1$ and σ $\hat{q}_2$, and its variance σ_1^2 *and* σ_2^2.

And then we combine these two using the Bayes rule, and find out what is the new mean and covariance, the new mean and mean and variance. And you will see that the new mean is the first mean plus Kalman gain multiplied by the difference of the two gains, two means, and then you have the covariance which is smaller than the two of the previous estimates. So, that is basically the principle of Kalman filter.

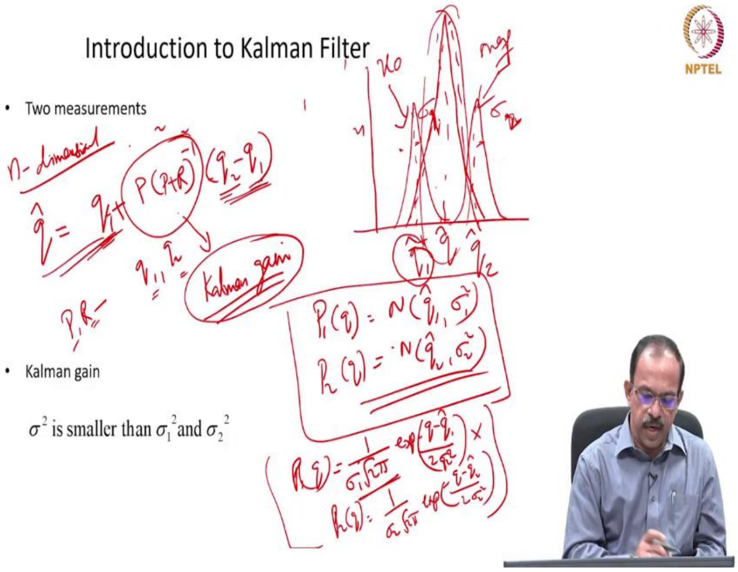
So, the same can be used for position estimate also. If we have the old position as x_0 and then a position estimate based on the map or the sensor information is this one, you can combine these two and get a new estimate which will be having a lesser I mean it will be having a better mean value and a better variance. So, that is the principle of Kalman filter localization.

So, the principle is basically to have two measurement two distributions and then combine them to get a better estimate of the robot position ok. So, the two measurements will be there. And then we have this Kalman gain which is $\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$.

And then as you ensure that, these two will actually the σ square that is the estimated variance will be less than the previous two variance. So, you will be always getting a better estimate of the position, so that is basically the Kalman filter gain, Kalman filter based localization principle.

Now, if it is a, so if this n-dimensional case, so we would not be able to use this as σ square. So, we have to go for the covariance matrix. So, in the matrix form, we can write this as, so the measurement will be in for an n-dimensional case.

(Refer Slide Time: 26:17)

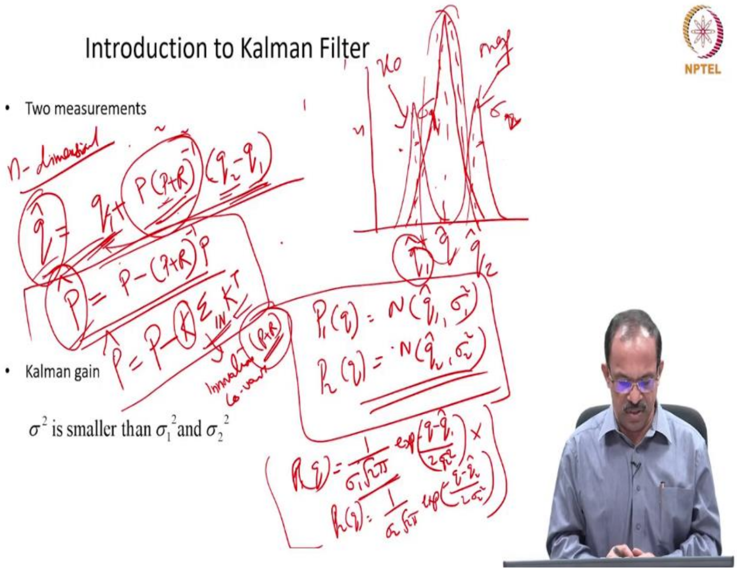


So, if you consider it as an n-dimensional estimate, so for single dimension or one dimension one dimension we normally use the σ and the mean and the σ values – the standard deviation values. But for n-dimensional case, what we do is we write this new q that is the mean value of the estimate can be written as $q_1 + P(P+R)^{-1}(q_2 - q_1)$, where q_2 and q_1 are the mean values of the previous estimates the two estimates; one is the measurement and the other position estimate.

And q_1 is the first one. So, you have this q_1, q_2 - q_1. And here P is the covariance, P and R, so P and R are the covariance of the two measurements. So, covariance of q_1 and q_2. So, P, P and R are the covariance of q_1 and q_2. So, the first measurement has got a covariance of P_1; second one has got a covariance of R, and a mean of q_1 and q_2. So, P and R are the covariances of q_1 and q_2. And this P + $P(P+R)^{-1}$ is this is known as the Kalman gain.

In the previous case, we represented in terms of the σ, but here since they are matrices $P(P+R)^{-1}$ will be the Kalman gain. So, now, you can see that the new covariance sorry the new mean will be the first mean plus Kalman gain multiplied by the difference of these two measurements q_2 a mean into q_1, so that is the new value of q. And the covariance of the new estimate, so P and R are the covariance of the measurements.

(Refer Slide Time: 28:13)



Now, the new covariance $\hat{P}$ can be written as $\hat{P} = P - P(P+R)^{-1}P$. So, this is basically the new covariance. So, we have P as the covariance of the first measurement q_1, and then R as the second one. So, $P - P(P+R)^{-1}P$ which can be written as $P - K\sum_{in} K^T$, where K is the Kalman gain which is $P(P+R)^{-1}$.

And $\sum_{in}$, in is known as the innovation covariance, we call this as the innovation covariance which is defined as P + R. So, this is if P + R is known as the innovation covariance that is the covariance of P q_1 1 and q_2 are added, and that is known as the innovation covariance P + R.

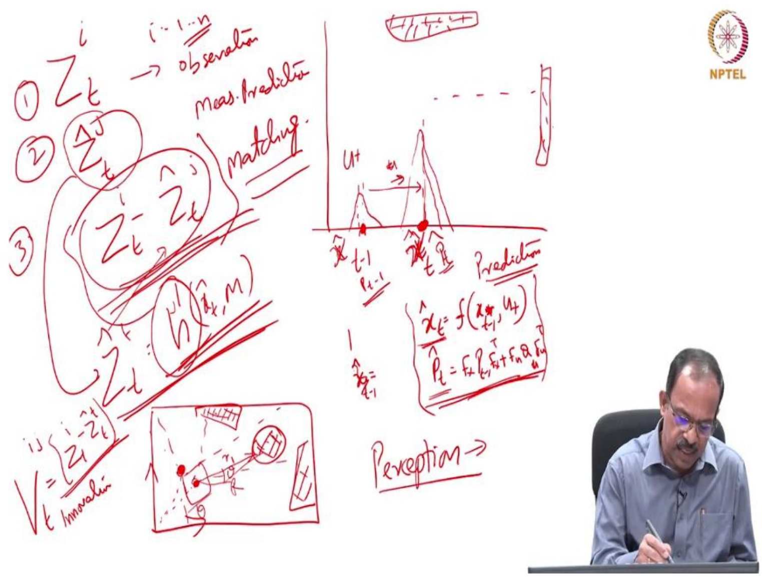
So, the new covariance of the estimate will be $\hat{P}$ = $P - K\sum_{in} K^T$, where K is the Kalman gain given by $P(P+R)^{-1}$, and σ innovation is the innovation covariance which is nothing but P + R, and then K transpose will be the new covariance.

So, when you have two one initial position and a measurement, you can combine these two, you can fuse this two information two data and get a new estimate with a mean of $\hat{q}$ t and a covariance of $\hat{P}$ and that is given by this relationship. So, what we need to know is the mean value of the estimate and its covariance, and the new mean value of the measurement and its covariance.

Once you have these two information, you will be able to get a Kalman filter based estimate which will be having a new mean and covariance. And this new mean will be better than the previous two measurements, and the new covariance will be smaller than the two measurements.

So, you will be having a much better estimate or you will be having much less error in the estimated position of the robot, so that is basically how we do the Kalman filter based localization. I hope you understood this. So, let us see how this can be implemented in a real mobile, I mean the mobile robots ok.

(Refer Slide Time: 31:05)



So, what we do here is assume that the robot is at a position initially. So, we take this as the robot position. And the robot is at a position which is given as previous position X_{t-1}. So, assume that the robot is at t - 1. So, this is X_{t-1}. Now, from this position, so X_{t-1} has got a μ value μ and it is a deviation P.

So, we call this as μ and P that is the mean value and its covariance is known for that particular X_{t-1} . And from there, you give a control input u, and the robot moves to t, X_t. So, this is the control input given u t is given here. So, the robot moves to the X_t ok. So, now we have a prediction update that the robot is it has moved from X_{t-1} to X_t using the control input.

So, this X_{t-1}. So, the at this position the mean value x t minus 1 can be the new value X_t, sorry, the new X_t the new position X_t can be estimated as a function of X_{t-1} plus sorry

X_{t-1} with the control input u_t, so that is basically the position. So, X_t, that is the new position X_t, its mean value of x can be obtained as a function of X_t and u_t.

So, we use the control input to calculate the new position X_t. And it will be having a covariance. So, its P_t can be obtained using the error propagation rule that we saw in the previous case, where we use this as $F_x P_{t-1} F_x^T + F_u Q_u F_u^T$ So, this was the error propagation model that we saw.

So, we will be able to move from X_{t-1} to X_t in the prediction stage. So, this is the prediction stage. In the prediction stage, we will see that the robot has actually moved from X_{t-1} to X_t. And the mean value of X_t and the covariance P_t can be obtained using this relationship.

Because we know based on the robot kinematics you will be able to find out what is the new position of X_t if we know the control inputs. And the covariance the uncertainty in that position can be obtained using the error propagation model. We saw this for the for a differential drive robot how do we actually calculate the position and its uncertainty, so that is the prediction update for the about.

So, now, we have this prediction. So, initially X_{t-1} was known, and P_{t-1}, so I call this as P_{t-1} is the uncertainty here. So, the position x, so it called this position X_{t-1} and the P_{t-1} is it is a uncertainty the t - 1. Now, from here, we have moved to with the prediction. So, initially assume that it was like this with a $\hat{x}$ and P_{t-1}. Now, it has moved to here.

So, you have a higher uncertainty. So, you can say that this is the uncertainty. So, now, it has the uncertainty has increased because of the sensor errors, and the previous P_{t-1}. So, the new $\hat{P}_t$ which is the prediction update will be given by this, and the new position X_t will be given by this. So, this is X_t, $\hat{X}_t$ the new position at t is given by this, and its uncertainty is P_t, and that two can be calculated using this. So, now, the robot is this position with the prediction update.

Now, we go for the next stage which is the perception update. So, in the perception update, what we do we will look at using the sensors and then get all the information coming from the sensor as Z_t. So, assume that, the robot is here. And it has got a sensor, and it measures something from the surroundings ok. So, I am just representing it here, there is a wall here, or there is a wall here something. So, there may be many things.

So, it can actually the robot can actually see or the sensor can see n number of objects in the vicinity ok, so that is why we call it as Z_t, so observation, so the sensor is seeing Z_{ti}, where i = 1 : n. So, it can actually see n objects, so that is basically the observation of the sensor. So, we call this as the observation, observation using the sensors.

And then the next step, so the perception, the first step is basically observation using the sensor. So, you will get see there are 5, 6 objects which is seen in the robot or seen by the sensors which is attached to the robots. So, these are the features Z at t location. Now, we check with the map that is the next one which is basically the measurement prediction, we call this as the measurement prediction.

So, measurement prediction is the robot will check with its map, and then see look at the map and then see if the robot is at this location at t, what are the things it is supposed to see ok, so that is basically the Z_t. So, we call this as $\hat{Z}_t$ ok. We call it as j $\hat{Z}_{gj}$, and j = 1 : m. There can be many things that the robot is supposed to see, but whether it is all are seen or not is not it is not known.

But if it is here, it should see these two ok that is what the map says ok. It should see these two or more also depending on where the robot is it is current position. Some of them may not be visible to the robot, sensor, some of them may be visible to the sensor. So, what it will do? It will check what is the measurement prediction we call it as Z_{tj}.

So, one is the observation from this using sensor, one is the prediction using the map. So, from the map, the robot will be able to predict ok, I am able to see 5 objects. Sensors, see tells ok, I am able to see 6 objects ok. It could be like that or sensor is saying I am able to see 4, but the as (Refer Time: 38:00) map, it says no, no, if you are actually in that position, you should have seen 5 objects. So, this way there will be a difference.

And then we try to find out what is the difference between these two measurements. So, we will try to find out are they matching? Z_t and the $\hat{Z}_t$ i and j for all i and j try to find out what is the mapping between these two, are they able to are they matching properly or not? If they are matching exactly, what is there in the map, and what is there in the what the robot is seeing? That means, the robot predicted position is it is accurate. There is no error in the predicted position of the robot. So, it exactly at t ok.

So, this can actually be explained using this way. Suppose, this is the these are the objects on the in the environment, and the robot is travelling from here this location. So, it is moving in this direction ok. And the robot predicts that it is in this position. And using the sensor, it should be able to see all the things what are there in the vicinity ok.

It will say that my position is this and from there, I can actually see this object. But in reality, the robot may be somewhere here. We do not know the robot has actually reached here, instead of here, the predicted position is not correct. So, the robot is actually here, and it is able to see this object this many objects.

So, there is a difference between what the actually the robot is seeing, and what the robot is supposed to see from that location, so that is basically the difference between these two. So, we try to map these two and see what is the difference in this what is seen and then what is predicted, so that is basically the second stage where we do the match, the third stage where we do the matching.

So, the matching cannot be done directly, because what all the map is with respect to a reference frame. So, the map will be having a reference frame. And the features are actually given with respect to the reference frame ok. And the robot is seeing from using the sensor, it is mounted onto the sensor platform at the robot platform. And robot will be having its own coordinate frame. So, what it is seeing, it will say I am seeing a object at a distance R_1 and at an angle theta with respect to my reference frame.

But this object is defined with respect to a map as this is the distance to the R and this is the theta. So, we need to if you want to compare these two objects, we need to move everything into a single coordinate frame, so we will move all this to a common coordinate frame and then do the matching. So, first we will represent all those features with respect to the common frame either the robot frame or the map frame, and then find out the difference between these two that is the mapping stage.

So, we need to transfer this to the common frame, and then find out the difference between these two. So, that is basically the difference. So, what we will do here is we try to find out the each one all of these this will be transferred to the map frame. So, all this will be transferred to the I mean all the map features will be transferred to the robot frame using a transformation matrix h, we call this as a h_j. So, this Z_t will be transferred to the map frame using a transformation h_j at x_t, m.

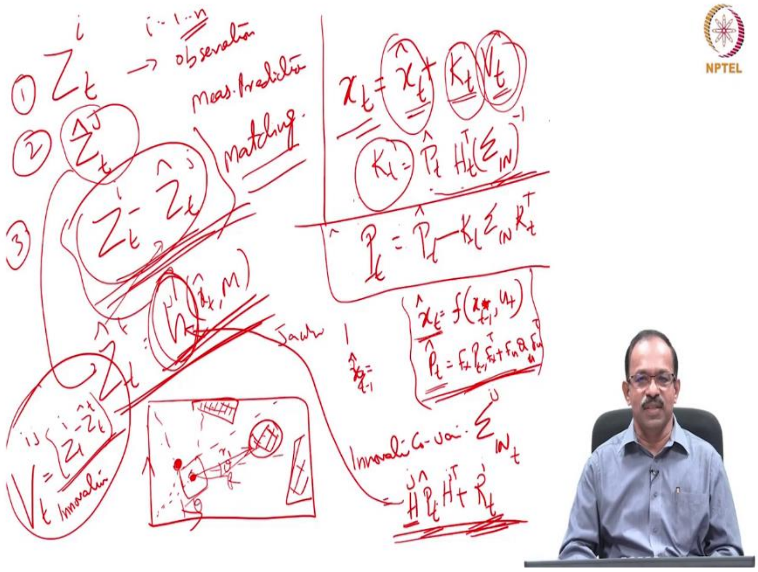
So, we will find out the transformation matrix and then we transfer all these parameters all these to the Z the frame the robot frame using a transformation matrix. So, this h is the transformation matrix. All those feature that is seen from the map or which is predicted from the using the map at x_t will be transferred to the robot coordinate frame, and then we get this Z_{tj}.

And then that Z_{tj} will be used for the comparison. It is not the directly what the map is giving, but after transforming to the transferring to the robot frame you will find out the difference, so that is that Z_{tj} ok. So, once a that matching is done, then we will find out how much is the difference between these two I mean the between i and i, or i j because there can be different numbers.

So, we will try to map each feature, feature with what is supposed to see, and then find out which one is matching, and then find out how much is the difference. And this difference is known as the V_{tij}. So, we call this as the V_{tij} that is what is the difference between ith and j th observation and the measurement prediction, and that we call it as the innovation.

So, the difference between this is known as the innovation and that is given as Z_{tj} - $\hat{Z}_{tj}$ t ok. So, this is basically the innovation that the difference between these two the features are known as the innovation. So, we get this innovation, so that is the mapping matching stage we will get the innovation. And once we get this innovation, we will try to find out the innovation covariance.

(Refer Slide Time: 43:35)



So, that is what actually we do. We will try to find out the difference, and then we will try to find out the innovation covariance. So, we call this as V_{ti} is the innovation, and then we will try to find out the innovation covariance which can which is written as Σ_{in} at the t for i j, so that is the innovation covariance.

So, this innovation covariance is obtained as because we have this measurement sensor uncertainty is known, position uncertainty is known because the robot is measuring from its position using a sensor. So, sensor has got uncertainty, position also has got uncertainty. So, the measurement in covariance or the innovation covariance can be written as this is equal to H_j $\hat{P}_t$ - H^T H_j^T + R_{ti}, where H is the Jacobian of this transformation h j.

So, we have the transformation h_j which transformed from the one coordinate frame to the other coordinate frame. So, H is the Jacobian of this. So, this is the Jacobian of h_j is h_j. And P_t is the covariance at t, so that is what we calculated here from x_{t-1} to x_t moved, so there is a covariance $\hat{P}_t$, so that is the $\hat{P}_t$. And R_{ti} is the covariance of the sensor.

So, you are using a sensor to get the information. So, the sensor also will be having a covariance its uncertainty. So, R_{ti} is the sensor measurement noise or sensor covariance. So, now, we have this innovation and innovation covariance. Innovation is basically the difference between these measurements, and the innovation covariance is the uncertainty in the measurement also.

So, once we have these two, then we can actually get the an estimate of the new position using the Kalman gain, Kalman gain principle which can be written as $hx_t = \widehat{X_t^-} + K_t V_t$ ok. So, $\widehat{X_t}$ is the position here that is the predicted position using the control input. And the this $\widehat{X_t^-}$ is the new estimated position after the measurement that is $\widehat{X_t^-} + K_t$ which is the Kalman gain and V_t is the innovation.

Innovation is this one that is the innovation, V_t is the difference between the measured and the observed sense and the predicted and observed features that is the difference is the innovation. So, we get this innovation V_t from the sensor and the map information, and then we have this K_t which is the Kalman gain, and x_t is the previous the predicted position.

And K_t is given as here the K_t will be $\hat{P}_t$ that is the previous covariance, H_t which is the Jacobian transpose and the Σ innovation inverse, so that is the Kalman gain. So, we will be getting this K_t as $P_t \hat{h}_t^T \Sigma_{in}^{-1}$. Now, we know this, we know this and we know this P_t, and $\hat{P}_t$ is known H_t is the transformation σ innovation covariance is known. So, you will be able to get K_{tr}. So, Kalman gain also obtained here. Once you have this, you will be able to get the new position estimate x_t.

Similarly, once you have this position you can get the new covariance also. So, the new covariance can be obtained as that is P_t. So, we what we have is P_t here. So, what we are interested in now the new covariance P_t can be written as $\hat{P}_t$ that is the previous covariance minus $K_t \widehat{K}_t^T K_t^T$. So, this will be the new covariance.

So, you have the new position estimate and its covariance using the Kalman gain. So, that is the basic principle of Kalman gain or Kalman filter based localization of mobile robots. So, there are a few steps involved in it. First we do the prediction, and then we use the sensors to find out all the features. Then we use the map to check what are the features supposed to see.

And then we try to match them by transferring all these information to a single coordinate frame, we try to match them and find the differences between this seen and supposed to see. And using that, we will find out the innovation and innovation covariance. And once you know innovation, innovation covariance we will find out the

Kalman gain. And use the Kalman gain and innovation, to get the new position predicted, so that is basically the principle of Kalman filter based localization.

So, I will stop here. Please go through this lecture, and then we will discuss this again tomorrow, I will explain it and then explain the way in which it can be implemented for the mobile robot also, so that we will discuss in the next class.

Thank you.