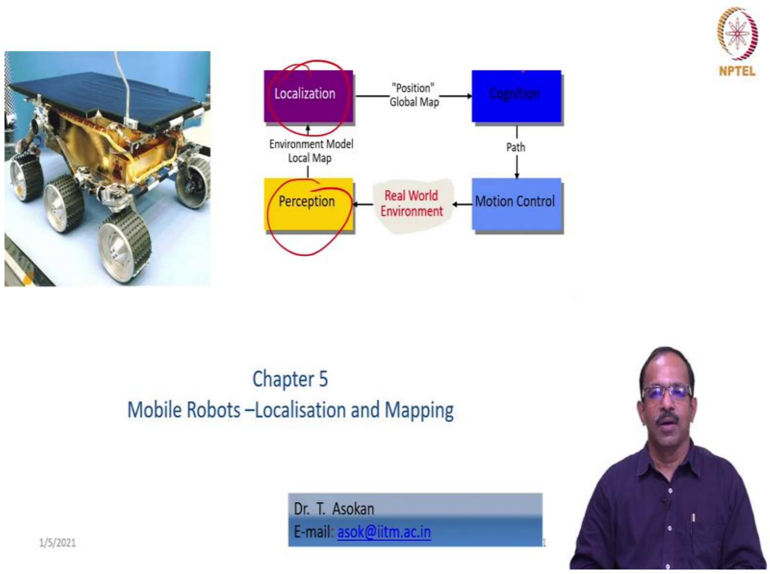
Wheeled Mobile Robots
Prof. Asokan Thondiyath
Department of Engineering design
Indian Institute of Technology, Madras

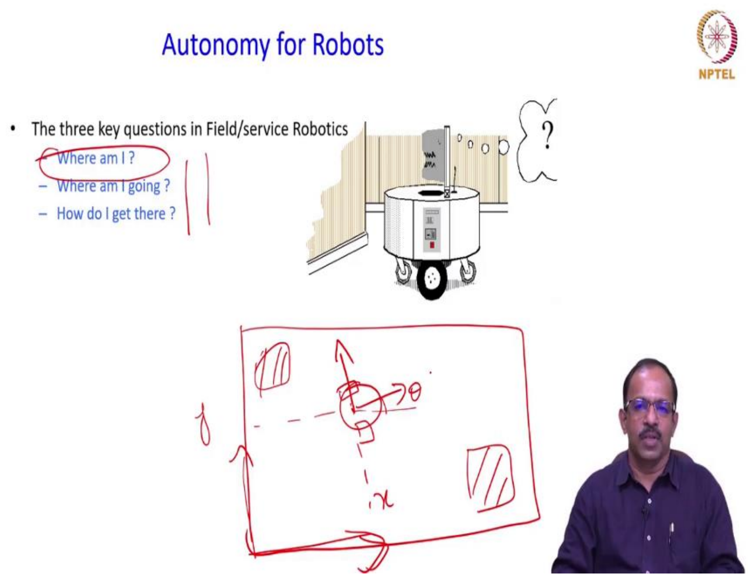
Mobile Robot Localisation
Lecture - 05
Mobile Robots - Localisation and Mapping

(Refer Slide Time: 00:13)



Hello everyone, welcome back. Today, we will start the discussion on Mobile Robots Localisation and Mapping. In the last few classes we talked about sensors and sensing importance of sensing. Now, we will see how we can use these sensors to do the localisation of a robots as well as to create maps. So, that is the mapping, we need to have the capability of localisation and mapping in mobile robots specially, when the robot is autonomous.

(Refer Slide Time: 00:49)

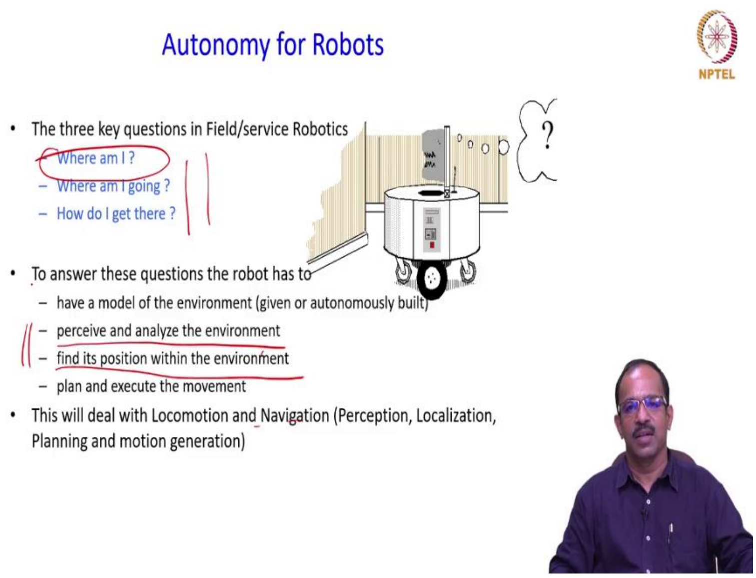


So, as I mentioned in one of the classes, autonomy for robots comes with lot of challenges and as I mentioned, the important requirements are the robot need to know where is it and where is it going and how would how can the robot reach there. So, these are the three questions the robot needs to answer. And the first question is where am I? And that can be answered through localisation.

The robot can localize itself in a given map or a created map; that means the robot knows where it is. So, if there is a large area where the robot is moving and this map is already given to the robots or the robot has created a map on its own based on the information. Now, within this the robot should be able to know where is it. Suppose, this is the robots, robot should know what is its x and y position with respect to a reference frame.

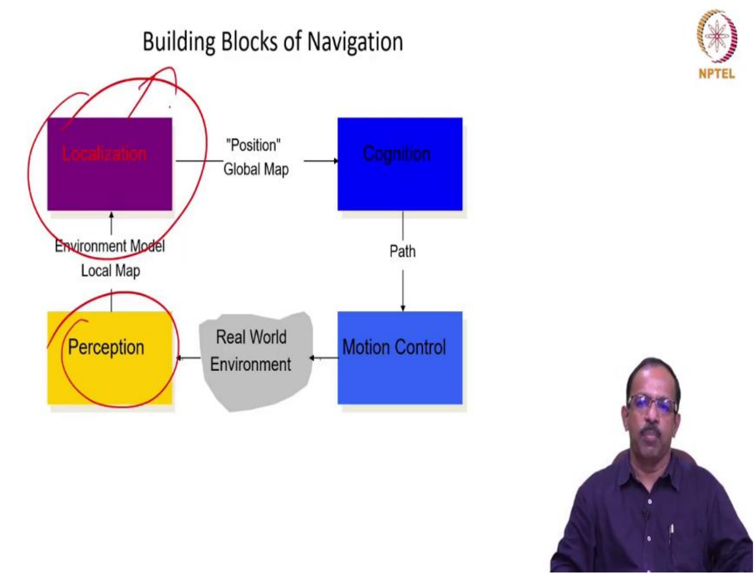
So, if there is a reference frame then with respect to this reference frame the robot should be able to know what is its x position and what is its y position and of course, its orientation θ also. So, that also is important. So, if the robot can actually get this information its x, y and θ, somehow through the use of sensors then we say that the robot is localised itself. So, that is basically the localisation requirements ok.

(Refer Slide Time: 02:21)



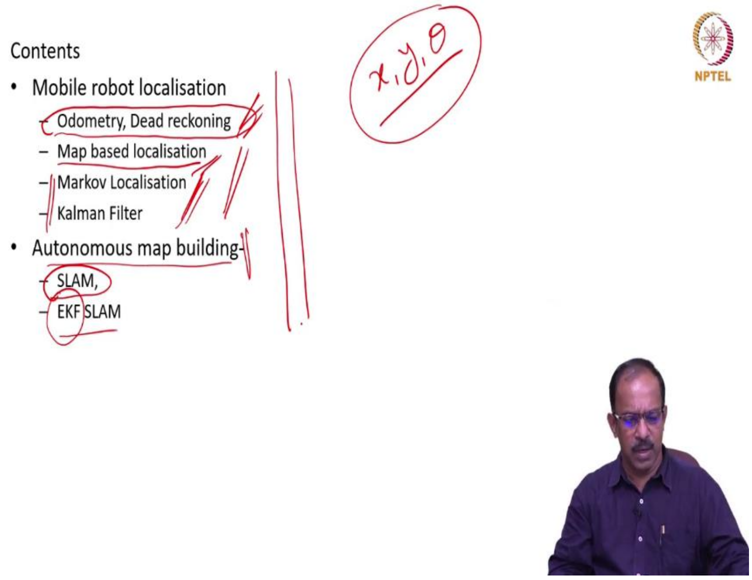
So, I already mentioned about so, we need to pursue and analyze the environment and find its position within the environment and then of course, it has to plan and execute. So, we will talk about this part only, the localisation parts in this discussion.

(Refer Slide Time: 02:41)



So, localization basically requires the perception to be done. So, basically the sensors will go and create get the data, sensors will collect the data and that data will be processed with the information available with the robot either the map is given or the map is created and based on that it do the localization that is what actually happens in the robots.

(Refer Slide Time: 03:07)



Now, in order to do that we need to understand few things that is basically how do the robot do the localisation ok. So, the first one to do localisation method is basically the odometry or we call it as the dead reckoning, that is the robot uses its sensors available on the robot and get the position and orientation and that is basically known as the odometry based localisation or a dead reckoning based localisation.

So, the robot needs to know its x, y and θ and the odometry will provide you the velocity information. So, the encoders attached to the robot can actually provide the velocity information and once you have the velocity information you can integrate it to get the position information and based on that you will be and again using the sensors which can be used for getting the orientation. You can get the orientation also θ.

So, we saw that we can use compass and other sensors to get the orientation and odometry can be used to get the position. So, that is the simplest way of getting the localisation of a robots, but it has got lot of disadvantages it cannot be used for a long duration. It can be used for a very short duration, because of the errors accumulator will create problem. And therefore, we go for the next method called map based localisation.

Map based localisation is method by which we use the odometry information and the map information and then localise that is basically the map based localisation. And then in web based localisation we can have go for the probabilistic localisation methods and these two methods are commonly used; one is known as the Markov localisation and the Kalman filter localisation.

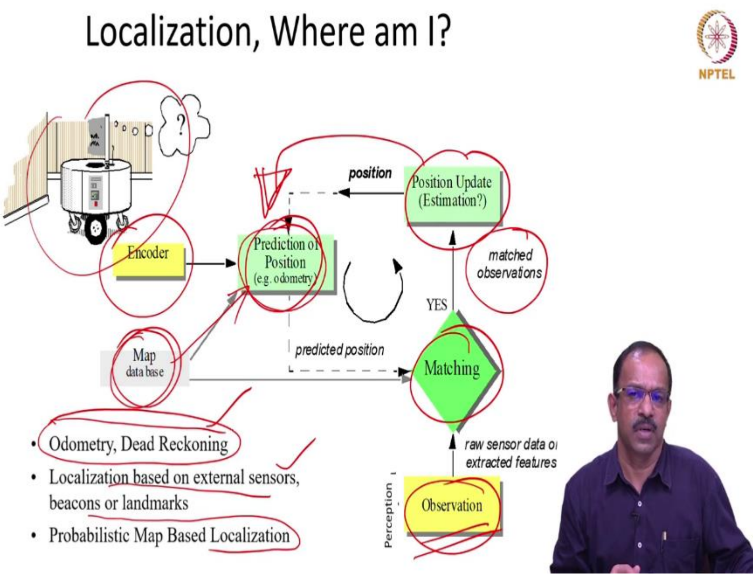
So, both are basically on map based localisation. So, you need to have the map information and then map information plus the odometry information plus the sensor information will be combined in order to get the localisation of the robots. So, these are the algorithms Markov and Markov localisation algorithm and Kalman filter algorithms are there which can be used for map based localisation.

Where we use the, we fuse the information from the odometry we from the other sensors, exteroceptive sensors and then the algorithms will be used to fuse this information and then estimate the position of the robot. So, that is basically the map based localisation. And as the robot do the localisation sometimes their map may not be available, so in that case the robot needs to create a map of the environment as it do the localisation.

And this is known as simultaneous localisation and mapping or slam; simultaneous localisation and mapping means slam. So, the robot will be able to create a map and localise itself within the map simultaneously ok. So, the map is not given. So, the robot does not know the map in advance, but using the sensor information it will try to create a map and then using the fusion algorithms it will try to localise itself within the map that is basically known as slam.

And then there is an advanced version of a extended Kalman filter slam where the slam can actually be achieved using extended Kalman filters. So, these are the topics that we will be discussing in this chapter. So, first we look at ok so, before going to the details, let me explain the scenario where the map based localisation or the localisation is done where we use the data fusion and algorithms in order to estimate the position of the robots.

(Refer Slide Time: 06:59)



So, if the robot is moving in the environment, it will be having its own encoders. So, the using the encoder data we can actually use the odometry to get the localization done that is the primary localization. So, it will predict the position of the robot using odometry and then it will be having a map given to this assuming that, that map is given, then it will check with the map this one and then see what is the predicted position if the mapping features are known.

So, it will be having a sensor and then using the sensor it will observe things and from the map also it will be getting information. It will check these two information. So, map says that I am supposed to see a wall at 5 meters, but I am seeing a map at 6 meter sorry, I am seeing a wall at 6 meters.

So that means, I am actually seeing the same objects, but the predicted position and the actual position are not matching; that means, there is a mismatch between the predicted position where the robot is currently and where it has to do, I mean actually where it is currently, actual position of the robot.

So, there is a difference between the actual position of the robot and the predicted position of the robot that is why we are seeing the sensor is seeing at particular distance and map says that it should be at a different distance that means there is a mismatch. So, it will do a matching and then it will try to recalculate the position, it will update the position and then that updated information will be stored.

So, at every instant as the robot moves it will keep on doing this the these activities. Every instant the encoder data will be used to predict the position and then using that position it will check in the map what should be what are the things that the robot should be able to see and it will use the sensors to really check whether the if the robot is able to see those things.

And they are not able to see or they are seeing at different distances or different locations; that means, there is an error between predicted position and the actual position of the robots. And that will be matched using some standard algorithms or they will be using some fusion algorithms to fuse this information and then get the new estimate of the robots.

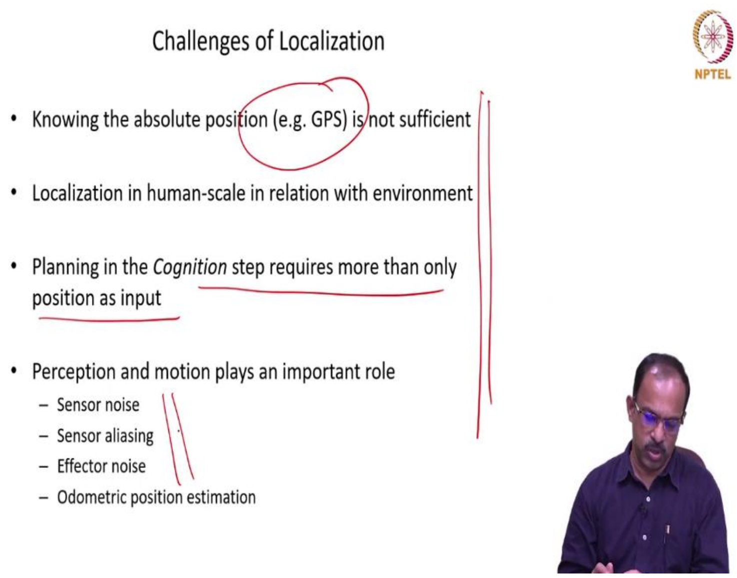
So, this is the way how the localization works in the autonomous robots. So, every autonomous robots will be having these features in the system odometry and the external sensors and then fusing this information and correcting the position of the robot that is what actually the how the localization works based on map based probabilistic approaches ok.

So, we have odometry, then we have localization based on external sensors, and then we use the probabilistic localization methods to recalculate the position of the robot and then estimate the new position of the robots. So, that is basically the process of localization in mobile robots.

Especially, the probabilistic model, where the localization uses the probability of the robot being at a particular location and the probability of the robot seeing an object at a location and then combining them and getting the new position of the robots; so, this localization principle will be used in almost all the robots, which depend on the map information as well as the sensor information for localization.

We will see these steps one by one. First we will look at the odometry based localization, how can we localize the robot based on the odometry information? So, that is the first step where we can use the robots sensor information and then use that information to estimate the position ok.

(Refer Slide Time: 11:19)

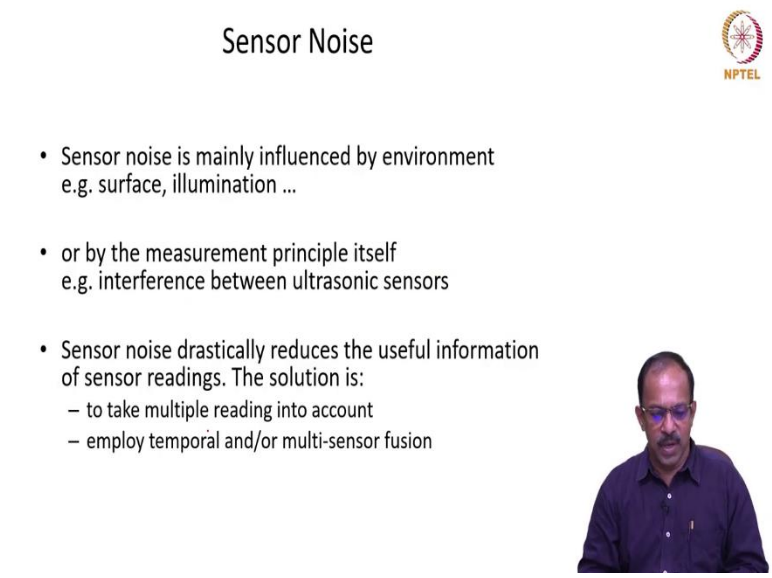


So, there are many challenges in localization as I mentioned, because the robot needs to know its local position not an absolute GPS position is not sufficient. GPS says that you are at this particular latitude and longitude that it is not sufficient for the robot. The robot needs to know within a room or within a human environment where it is that is the information with reference to a particular frame.

Therefore, the GPS information may not be useful in this case ok. And then it may require more than one position as a more than position, only position as a input it may require many other information for it to plan the activities. So, just in local position is not sufficient needs to know how far the objects and how to avoid obstacle as it moves forward.

There are many things the robot needs to do. So, that itself is a challenge. And then as I mentioned the sensor noise sensor other things are also a problem in localization. We mentioned about this in the previous lectures about sensor aliasing effector nodes, etcetera, etcetera. So, all those things will actually lead to lot of errors in the estimated position of the robots ok.

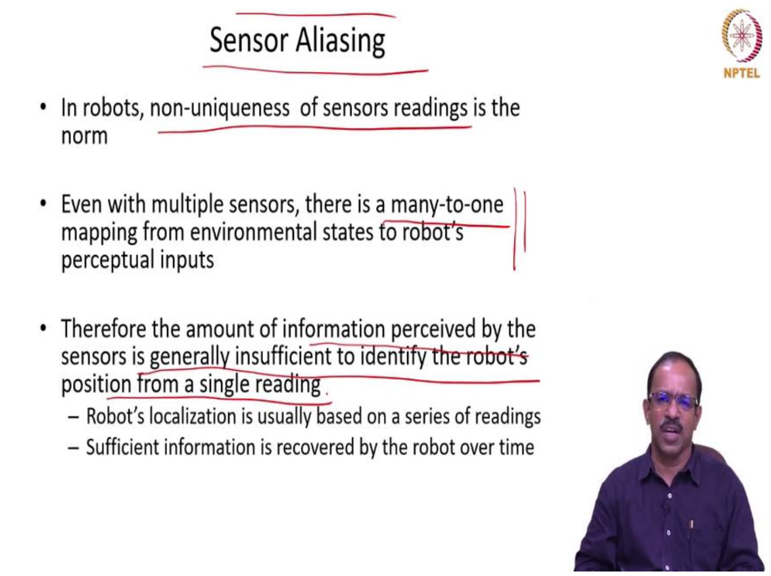
(Refer Slide Time: 12:39)



Sensor noise as I mentioned earlier is influenced by the surface illumination etcetera and the measurement principle also it may affect and because of this it drastically reduces the useful information. So, the useful information that is coming to the user or to the robot will be very much less, because of all these reasons.

And to do that we may have to take multiple to overcome we may have to take multiple readings or employ temporal and or multi sensor fusion. So, these things will be required in order to reduce the errors.

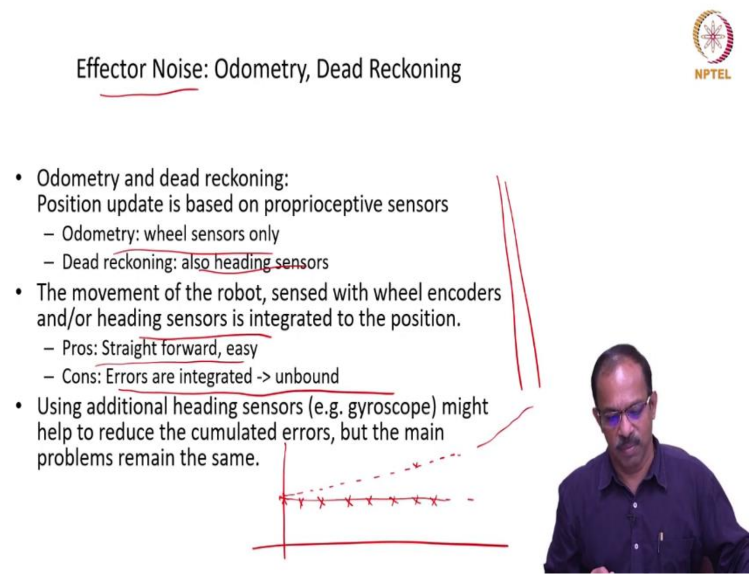
(Refer Slide Time: 13:19)



Sensor aliasing is another one which actually causes problem, because it is known as the non uniqueness of sensors readings where the readings the robot will tell that ok there is an door in front of the robots, but it can be a door, it can be a cupboard, or it can be a cut out, or it can be anything.

So, many to one mapping is possible. So, it can actually map many things to a one an object that is basically known as many to one mapping in the environment ok. And moreover, the information pursed by the sensor is generally insufficient to identify the robots position from a single reading. So, a single reading may not be sufficient to estimate the position of the robots.

(Refer Slide Time: 14:13)



So, these are all challenges in using sensors for localization. And of course, there will be noise from the effector that is the actuators; the motors will be having their own errors. Though we have many things that the wheel encoders and or heading sensors when you integrate it may be easy to do this dead reckoning, but there will be lot of errors, because errors are getting integrated unbound.

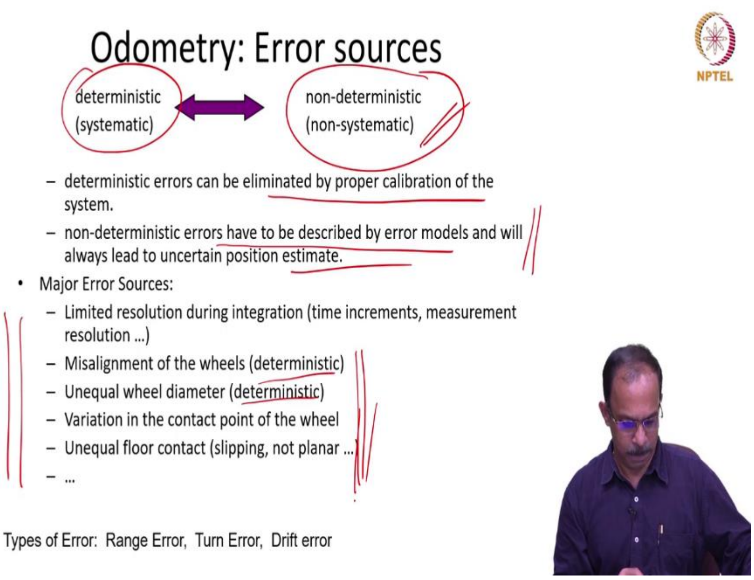
Because when we use an accelerometer to get the velocity information and then the velocity information is used to get the position information. We are actually doing an integration so any error in the accelerator accelerometer is actually integrated and then the position error will be much more than the actual information that we are we are interested in.

So, these are actually challenges in using sensors ok, but unfortunately, we do not have too many solutions for this we need to live with this kind of issues and therefore, we need to have some methods to reduce the error as the robot moves in a environment. As I have mentioned, if you are using only odometry and along with that we are using the dead reckoning with heading sensors that may not be really sufficient, because it actually keeps on adding errors.

So, suppose the robot is moving in a environment suppose, the robot is moving like this and we assume that the robot is moving straight both the encoders are giving you the same reading, but the encoder itself is having some kind of an error. So, though we calculate the position based on this at every instant we try to calculate the position then we will see that actual position is calculated here, but the robot may be going like this.

Calculation will clearly show that your position is here, because your encoder data is coming properly and we are just calculating, but the actual position of the robot may be here and as the time passes this error increases to very large levels and then the estimation will not be of any use.

(Refer Slide Time: 16:25)

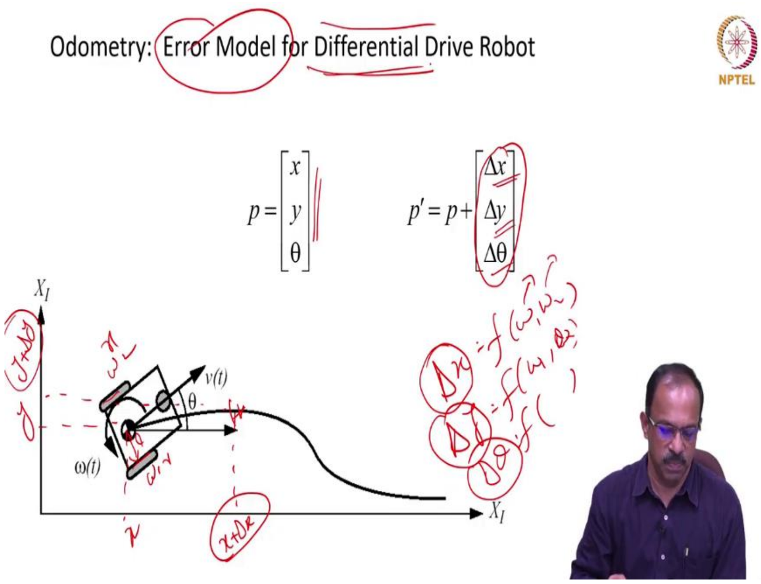


So, this is something which we need to reduce, but reducing this one I already mentioned about the error sources. We will be having deterministic sources and determining errors and non deterministic errors. So, these deterministic errors can be eliminated by calibration, but non deterministic errors need to be modelled by some methods. So, we use the probabilistic model as I mentioned earlier. So, we need to use the probabilistic model and then use that one to model the non deterministic errors.

And there are many other sources of error, because your terrain may not be flat, wheels may not be having the same diameter, the measured wheel speed may not be correct. So, all those things will also add to array. So, these are basically some of them are deterministic which can be compensated, some of them are non deterministic which cannot be eliminated I mean cannot be modelled.

So, like for contact, unequal for contact slippage, all those things cannot be really modelled. So, that need to be taken into account in the estimation of the position of the robot. In the localization we need to be aware of these things and to ensure that we have some methods to compensate for these errors ok.

(Refer Slide Time: 17:41)



So, let us look at the error model for differential drive robots. So, we know that suppose, the robot is at this position. So, you have a robot and the robot is having two wheels as you can see here. Now, we need to develop a model for localization of these robot and we need to know how the model, the error in the estimated position will vary, because of the error in the encoders, because we will be using encoders to estimate the position of the robots.

Assume that the robot is initially at $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$. So, at the initial position is given as $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$. So, this is the x position, this is the y position, and this is the orientation θ. Now, as the robot moves, you will see that the new position will be suppose, the robot we are here so now, the robot is at here at next position.

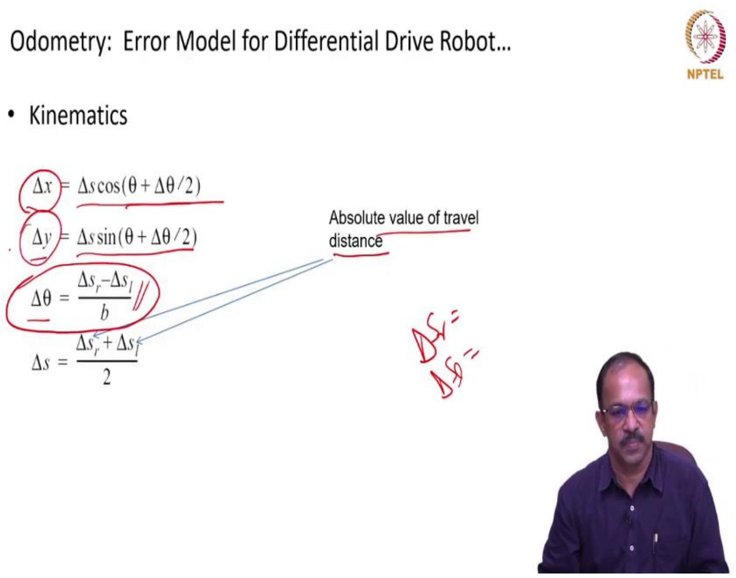
So, we say that this is $x + \Delta x$ and this is ok, this may be $y + \Delta y$. We need to know how do we actually get this $\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$ and then how much will be the error in $\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$, or the total estimated position how much will be the error in the estimated position also.

Assuming that initially there was no error it was exactly $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ and now, we calculate $\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$ from this ω_1 and ω_2. So, we have ω_1 ω_2. So, we know that $\Delta x = f(\omega_1, \omega_2)$ and $\Delta y = f(\omega_1, \omega_2)$ and similarly, $\Delta \theta$ also ok, but there are other parameters like the radius of the wheel and then the distance to the centre. So, all those things also affect.

So, you might have you have you have learnt about the kinematics. In the kinematics we have method to estimate the position of the robot Δ, we can get find out Δx and Δy using this using ω_1 ω_2. So, if we know the error in ω_1, we need to know ω_1 and ω_2, we need to know how much will be the error in Δx Δy and Δθ or the total estimated position that is basically known as the error model for different robots.

So, we will take a example of a differential drive robot in order to make it simple. So, we do not consider the other robot for the time being, we just take the differential robot, we assume that we know some of the things about differential robot and then we use that one to get the estimation.

(Refer Slide Time: 20:45)

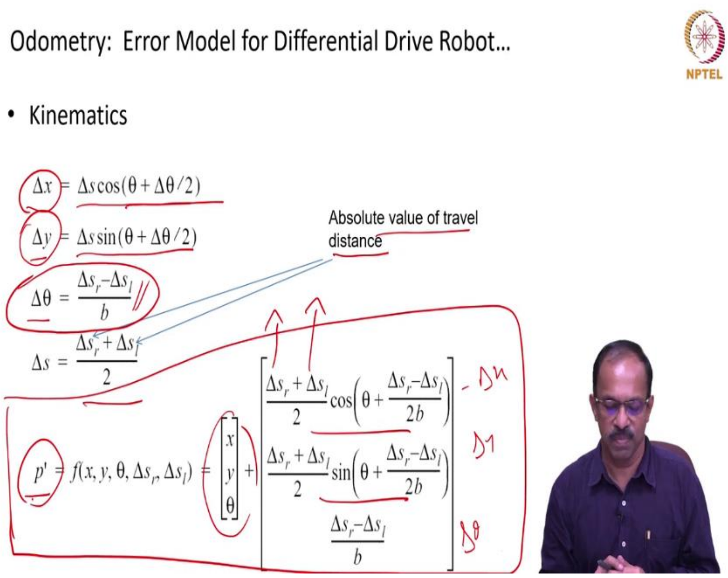


So, if you have this kinematics known. So, we know that $\Delta x = \Delta s\cos(\theta + \Delta\theta/2)$, $\Delta y = \Delta s\sin(\theta + \Delta\theta/2)$, and $\Delta\theta = \frac{\Delta S_r - \Delta S_l}{b}$ and $\Delta s = \frac{\Delta S_r + \Delta S_l}{2}$. What is ΔS_r? They are the absolute value of travel distance.

So, we assume that the ΔS_r is the wheel travel distance for the right wheel and ΔS_l is the distance travelled by the left wheel and that is obtained from the encoders. So, we assume that we use the encoder ΔS this encoder to get the wheel speed, because it as a differential drive robot θ is controlled by the two wheels. So, when there is a variation in ΔS_r and ΔS_l, we will be having a Δθ also.

So, you can see that the $\Delta\theta = \frac{\Delta S_r - \Delta S_l}{b}$ where b is the distance between the wheels that is wheel distance is given as b. So, $\Delta\theta = \frac{\Delta S_r - \Delta S_l}{b}$ and then once we have $\Delta x = \Delta s\cos(\theta + \Delta\theta/2)$ and then $\Delta y = \Delta s\sin(\theta + \Delta\theta/2)$. So, that is the way the kinematics can be used.

(Refer Slide Time: 22:19)



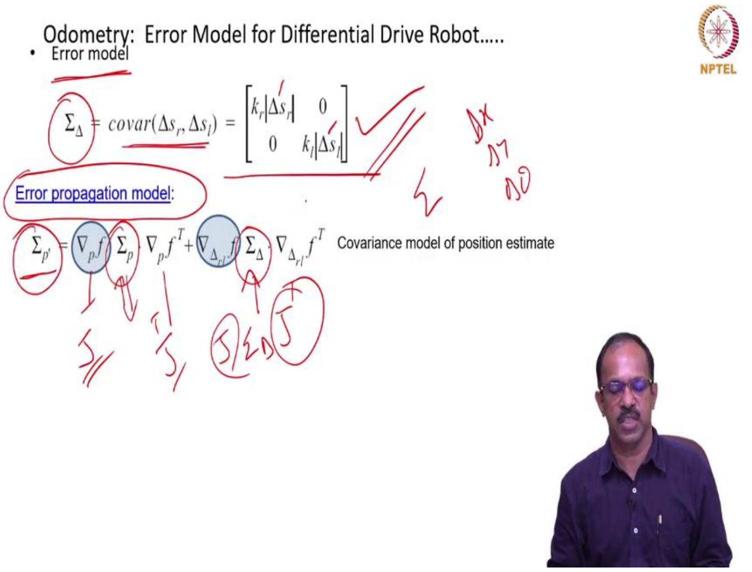
So, you will be able to get Δx and Δy using this information ok. Now, the question is so, the new position of the robot

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

So, the new position p' can be obtained like this.

Now, the question is; if there is an error in ΔS_r, ΔS_l, what will be the error in this new position p' ? That is what we are interested in, that is basically the error model for the differential drive robots.

(Refer Slide Time: 23:05)



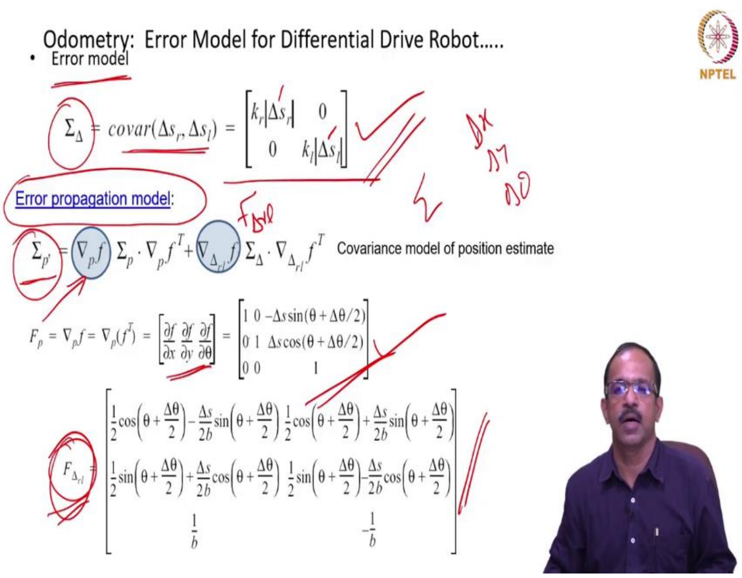
To do this, we need to look at the error propagation model. So, the error model for the sensor, we said that the sensor has got a error and that sensor error can be actually represented as a covariance ΔS_r, ΔS_l. So, we assume that the covariance in sensing the sensor covariance. It is can be given as $\Sigma_\Delta = covar(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix}$ assuming that it is diagonal matrix and it is proportional to ΔS_r and ΔS_l.

So, we assume that this is known. Now, if the sensor information is known then we want to know how much will be the error in x and y, Δx, Δy, and Δθ or we want to know what is the covariance in position estimates. So, for that we use the error propagation model. So, the covariance in the position estimate, new position estimate, it is given by this relationship ∇f position covariance p and then sensor covariance Δ.

So, assuming that this initial covariance of the position estimate is known and the sensor covariance is also known, we will be able to find out the error covariance of the position estimate using this where ∇f is basically nothing but the Jacobian and this is the covariance and this is the Jacobian transpose and this is the Jacobian of the sensor.

And then this is the covariance and then their transpose sorry, this is the Jacobian Δ their position and then the sensor relationship Jacobian. So, once we have this Jacobians, then we will be able to get the covariance of the position estimates. And how do we get this?

(Refer Slide Time: 24:59)



So, the ∇f is obtained. So, this Jacobian Fp for the position estimate Jacobian we can get it as ∇f which can be obtained by taking the partial derivative of f with respect to x y and θ, because f is a function of this x y θ. Therefore, we will be able to get this as partial derivative here,

$$F_p = \nabla_p f = \nabla_p(f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}.$$

Basically, we are taking the partial derivative of this previous function, the this function, we are taking this partial derivative of this function and then writing it here as the Jacobian here. So, this is basically the Fp. So, this is F∇rl which is the Jacobian of the sensor, because sensor also has got the relationship here that can be obtained like this, if ∇rl is equal to taking the partial derivative of the relationship between x and Δ.

So, you will be getting the Jacobian F∇rl also. And for every step we can actually calculate this, because the θ, Δθ all this information is available. So, we can find out this Jacobian and then estimate the position ok. So, I will stop here. We will continue this discussion, ok.

I will explain this once again in the next class and then we will continue the discussion on localization using other methods also.

Thank you.