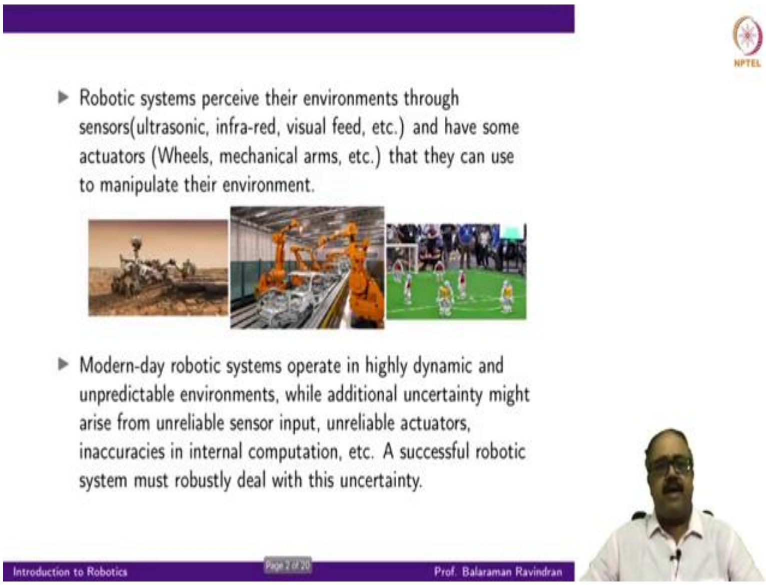
Introduction to Robotics
Professor. Balaraman Ravindiran
Department of Computer Science
Indian Institute of Technology, Madras
Lecture 7.1
Introduction to Probabilistic Robotics

Hello everyone, and now we are going to be looking at the Computer Science module for the Introduction to Robotics course. My name is B. Ravindiran, and I am a faculty in the Computer Science Department, in IIT Madras.

(Refer Slide Time 00:35)



So, during the next few weeks, we will be looking at various issues that have to do with how robotic systems perceive their environments through sensors, and how they act and affect the environment through various actuators. Sensors could be as you have looked at, over the course, already, the sensors could be things that are, you know, from ultrasound, could be infrared, could be cameras, could be more touch sensors, bump sensors.

So, there are variety of ways in which the robots look and sense their environment. In fact, in some ways, the amount of sensory information that a robot can get is more rich than what a typical human looks at. And it becomes more challenging to operate with this kind of rich sensory information as well.

And then you have a set of actuators, it could be wheels, as you can see this humanoid robots, it could be limbs, or it could be more articulated mechanical arms, like in factory assembly lines. So, there are a variety of circumstances under which different kinds of actuators are used. And these are typically used to manipulate the environment.

Now this is one of the core things in robotics. There is a system that perceives environment through sensors, and then operates and affects environment through their actuators that are part of the robotic system. So, what makes a lot of robotics, the algorithmic make aspects of robotics challenging is that, there is a significant level of uncertainty associated with both of these parts of it, there are sensors and their actuators, and the sensors could be highly unreliable.

So, you cannot say that the robot knows exactly where it is just by looking at the sensory input. There are two reasons for this uncertainty. So, one of these reasons is that the sensory information is not complete. So, for example, say I am using an ultrasound sensor, that ultrasound, let us say have 8 ultrasound sensors pointing around 8 different directions. So, in each of those directions, the ultrasound sensor is going to tell me where is my nearest obstacle.

But that is usually not enough for me to localize or locate myself within a room, I do not know exactly where I am, just because I have this sensory information that is coming in to me. So, the sensory information is incomplete, and therefore it is unreliable and that may. Another reason it is unreliable is due to a variety of different disturbances or different noise sources that might be there in the environment. The sensor itself, the actual electronics, actual mechanics that goes into the building of the sensor could have some amount of stochasticity, some amount of noise in it.

Therefore, even though I make the same measurement, I stand in the same place, and I measure distances again and again, I might get slightly different distance readings, it is not that it is going to be stable all the time. The second thing, there are stochastic disturbances in the environment itself. So, I am trying to measure the distance to a wall and it could very well be that somebody just walks in front of the wall for a brief while. And if I measure the distance exactly at that instance, the distance might be much shorter to the wall than it is actually.

And I do not really have a mechanism by which I can include all such unmodelled disturbances in the environment, and then make decisions out of them. So, I typically, you know roll them up into what we call noise or stochasticity and then we try to, you know accommodate for all these disturbances by looking at these kind of noisy models. So, and this is just, just in figuring out where you are.

And if anyone has tried anything with robots, or even trying a simple line following robot, you know that the actuators that are there even a simple like a gear system that turns wheels, is not completely accurate. It is going to take, you know quite a bit of calibration before we figure out, if I say move 1 meter forward, or 1 foot forward, how much does the robot actually move? So that is going to take a little effort to figure out.

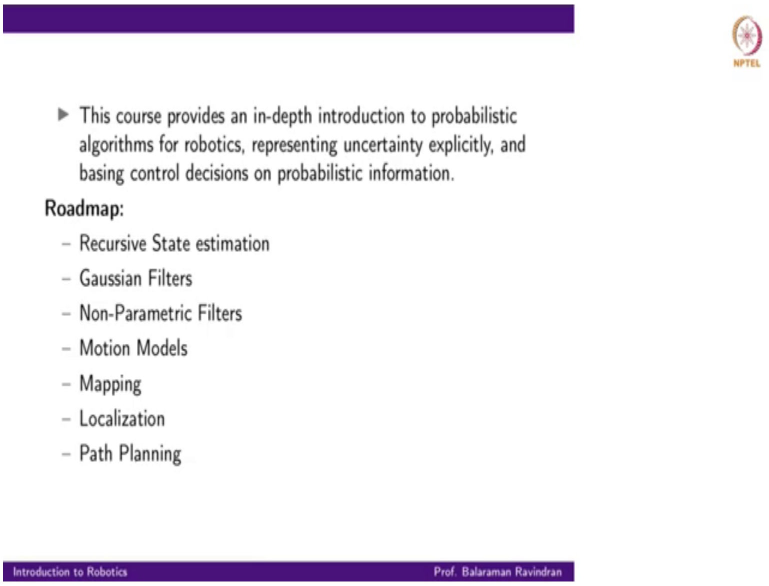
And again, there are many, many sources of noise and it could be that like a gear slips a little bit when you are trying to move, and maybe depending on how much you have lubricated it and things could work slightly differently. And so this, the accuracy of this actuators also, if I say do something, whether the robot actually did it, is not guaranteed. So, we have to account for those kinds of inaccuracies and those kinds of noise as well. Now there is a story that I would like to tell my class.

So, we were working with a very simple like a robotic platform. So, the goal of this platform is that I have a ball in the middle of the platform, and I have multiple degrees of freedom, I can tilt the platform in X-Y and the opposite directions. And the goal is to make sure that the ball stays in the middle of the platform. So, it turns out that we come up with a very good controller, and then we start executing it.

But after a while, the controller starts failing, then I took as a, I mean we were really confused, what the heck was happening, it works fine for a bit, and then suddenly it starts failing. It turned out that the gear assembly that we were using had very high quality plastic gears. And after several hours of operation, the plastic gears started wearing, and therefore the controller needed to be adjusted to account for that. And since we were not getting a very accurate feedback, in fact, we had no way of measuring the wear on the gear.

And because we were not able to model that, so even though we had a good controller, it kept failing every so often. And we were able to redesign the controller based on the measurements that we could make, but then that again, did not stay stable. So, there are many, many such complications that arise. And so what we are going to be looking at through the next few weeks, is how to build robotic systems that can robustly handle this kind of noise and uncertainty.

(Refer Slide Time 06:54)



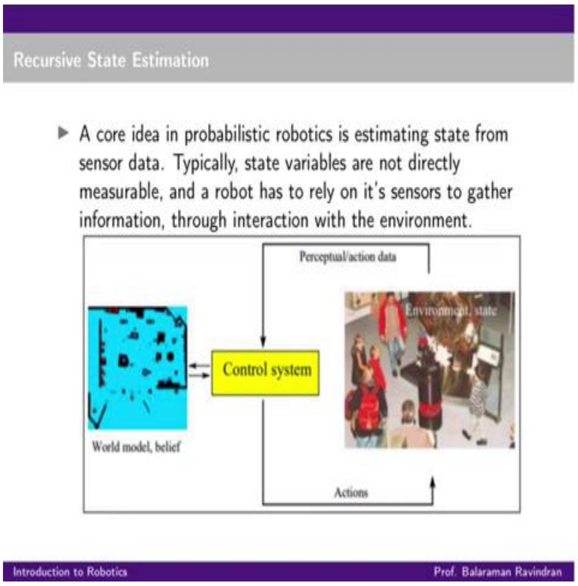
So, I would say that, we are going to be looking at what some people call as Probabilistic Robotics, or Probabilistic Algorithms of Robotics. And the textbook there is mentioned on the webpage is again called Probabilistic robotics and that is going to be dealing with these issues in detail. So, the textbook is very extensive so I am not going to be covering the entire book, it is just not possible in the few weeks that we have available to us. So, what I will do is I will touch upon these topics, and I will look at representative approaches for each of these topics.

In some cases, I will go in fab detail. In other cases, we will stay at a more higher intuition level, so that you can later on our follow up either by reading this book, or by doing additional courses that take you in depth. So, remember this is after all an introduction to robotics course and we are already packing in a lot of material in this course. So, the first topic that I look at is what we call recursive state estimation. And then under recursive state estimation, so we will look at a variety of different approaches.

And so the two main things that I will be looking at are filters for state estimation based on certain Gaussian assumptions, which we called Gaussian filters. And then we will also look at a one class of nonparametric filters that allows us to go away from making any specific assumptions about the system. And then, we will look at both motion models that model the noise in the motion as well as sensing models that model the noise in the sensory systems. Again, I will be looking at very specific examples in both of these and also try to introduce you to the general principles based on these specific examples.

And then we will look at two related problems, one is mapping, mapping is basically trying to figure out how the environment around you looks like. So, if you, after we look at path planning, I will also very briefly I will not put it down here in the roadmap, because it is going to be a very brief introduction of learning with robots, learning on robot, robotics platforms. And I will specifically talk about a paradigm called reinforcement learning, but this will be a very, very brief introduction to reinforcement learning itself.

(Refer Slide Time 09:27)



So, the first thing we will start looking at is this problem of recursive state estimation. Recursive state estimation is one of the core problems in robotics, so if you look at how a robot behaves in its environment, you can see that there is a robot here, it is in a very complex environment, there are obstacles around it and there are people here who are moving obstacles, and the robot is going to get inputs from all kinds of sensors.

You can see a ring of sonar here, and there is a camera on the top and there are bumps sensors, if somebody actually, you know, touches at the bot, it will tell you that there is contact, so all of these, that is a rich set of sensory input that is coming into the robot. And so, with this data, so indoor, it basically has to figure out where it is in the world. So, this is like a world model that the robot has. And it has to figure out exactly where in this world model is the bot?

So, so there are obstacles, and there are like walls that are modelled and the robot has to figure out exactly where in this the bot is and in what direction it is moving, at what angle, what orientation it is facing. So, you have to have things like the X, Y coordinates for the bot and the theta the angle at which it is facing, and the velocity with which it is moving. If it is accelerating, what is acceleration. And if it has an arm on it, what are the various angles at which the arm is positioned, and so on, so forth. And so, all of these are information that the robot needs to decide what it is to do next.

So, this information, so the information that the robot needs to reliably make decisions about what to do next or how to behave in this environment. So, we call these as our state information or state variables. And sometimes the state information is not directly measurable, as we saw here, so I really need to know the X-Y coordinates of the bot and the orientation it is facing. And even if I have, even if I have a GPS sensor, because I am indoors, GPS sensor is going to be pretty inaccurate.

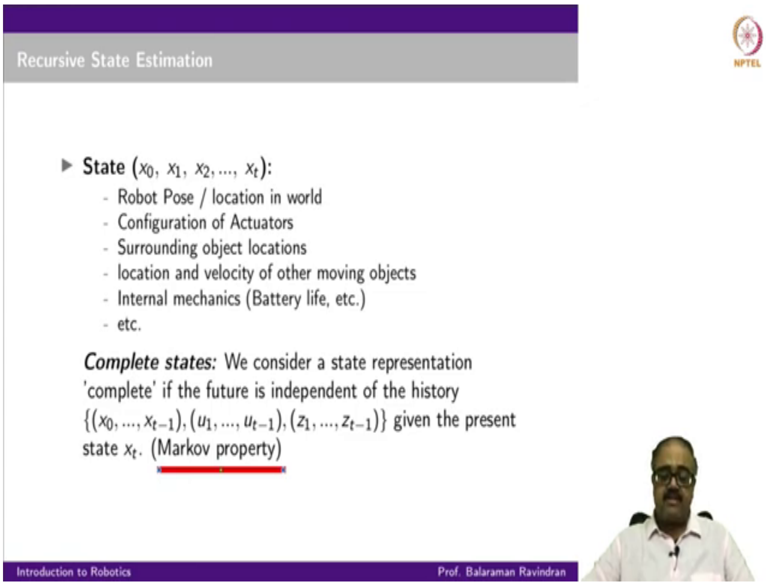
It is just going to tell me within a very broad region where I am, and it is not going to tell me exactly where I am with relation to various obstacles, or humans and other people that I am interacting with in the environment. And therefore, we need a mechanism, where we look at all the sensory data. Suppose I do not have a GPS data, because it is indoors, and it is very noisy, I would have to use all kinds of sonar information that I get and as well as any video information, any visual information that I have, in order to figure out what my status, so this is a huge challenge.

So, the recursive state estimation problem looks like this. So, given all the sensory information I have, and knowledge of what is it that I am doing in the world, the sequence of actions that I have tried to take in the world, can you tell me what is my exact state right now. So that is the recursive state estimation, it is recursive, because I start off with making an estimate for what my status, then I make an action. And then I make new measurements of my world, I make new measurements of where I am.

So, I have an original estimate of my state, so I am in this particular location, I am in this particular location, and I am facing north, let us say that is my initial estimate and then I say, I am going to move 3 meters north, and then I make another measurement. Now, I have to figure out where exactly am I, based on my previous estimate, the measurement that I made and the movement that I have made. So, all of these together helps me estimate, re-estimate my state. So, this is where the recursive part comes in, I use the previous estimate of my state and the action, and the new measurements in order to get a second estimate of the state.

So, this is the recursive state estimation problem. And even if you start off with a very noisy estimate of where you are in this world, what exactly is your state, as you do a few iterations of this, you typically end up refining your estimate about your current exact location and orientation, other state variables, and that makes you, that allows you to make better and better decisions over time.

(Refer Slide Time 14:24)



And so move on, trying to make this a little bit more formal in the next few slides. And then we will go to the next lecture for the actual algorithms for estimating this. So, the state consists of many, many variables, here I am, I have marked x0 to xt, which is a history of a state here. So, x0 is the state at time 0, x1 is the state of time 1 and x2 is the state at time 2, all the way to xt, which is the state at time t. And so, what will each of these x's consists of, here is an example of things that I already mentioned a few more here. The first is the robot pose or location in the world; the x, y and theta.

And if there is more than x, y, I mean, if you can also move in the third dimension, it probably is x, y, z and theta and then I have the configuration of actuators. So how much has the wheel turned, I mean, if there is any kind of other batteries that could, any motors that could move so what is the angle of the motors. Or if I am looking at arms and links in the arms, what are the different, you know relative angles of these limbs of the arms and so on and so forth, so that is another part of the state.

Remember all of these constitute one x, so x naught could potentially have many, many, many components, x 1 could have many, many components, and so on and so forth. And the next category of items that can go into a state description is the object location, the surrounding object, where is a table, is there a table next to me, and where is the wall, and so on, so forth. These object locations could be static, in which case, sometimes you just represent them as part of the map. But these object locations could also be moving around, in which case, you would like to put it as part of your state itself.

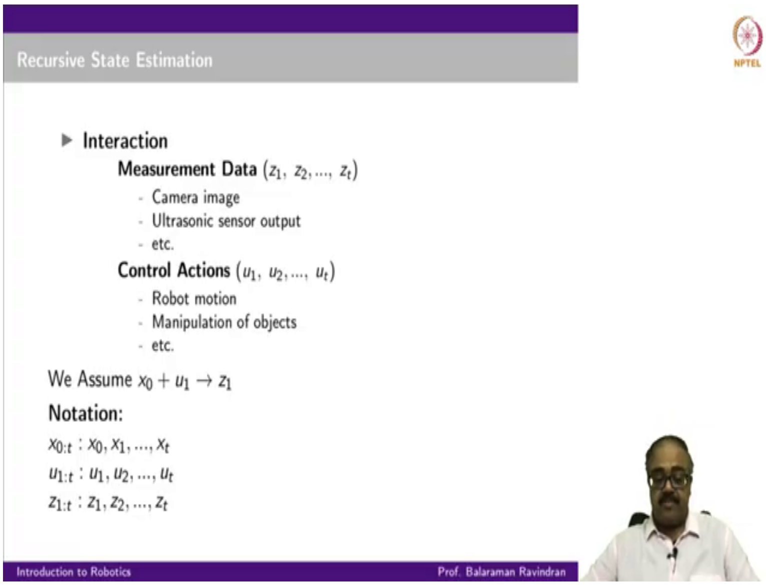
And likewise, not just the locations of the surrounding objects, you could have the velocity as well, if the objects are moving. So, in which case, you have to keep continuously updating the location, and possibly the velocity of this other objects, and any kind of internal measurements that the robot is making. So, things like the internal health of the robot, could be things like battery life, could be things like time to service, could be wear on motors, or wear on gears, and so on, so forth.

So, there are a variety of different things you could think of that essentially would be required for you to make decisions about your behaviour in the work so that you are going to accomplish whatever goal it is that you have set out to reach. So, we would consider, in this case, if a state representation is complete, if it has all the information that you need for making decisions, and we will also assume that the state is going to have what we call the, what we call as the Markov property.

We will assume that the state has what we call the Markov property and this means that the state xt has enough information for me to make decisions without having to worry about everything that went before xt. So that means I do not have to worry about x naught, x1 all the way up to xt minus 1, if I know what is xt. Likewise, u's as we will see in the next slide, u is used correspond to the actions that I take, so I do not have to worry about what are all the past actions I took, as long as I know that I have ended up in state xt.

And likewise, I do not have to worry about all the past observations, all the measurements, z measurements, I do not have to worry about all the past measurements I have made, given that I have ended up in state xt. So, this we call as the Markov property. And so, we are going to assume that our states are complete, and that we have enough information for us to make decisions, and that our states satisfy the Markov property, so that we do not have to worry about the past.

(Refer Slide Time 18:19)



And so, making the rest of it formal, so the interaction, so we are going to assume that z1 to zt are about the measurement data. So it could be the camera image, it could be ultrasonic sensor outputs or it could be bumps sensor readings, contact sensor readings, it could be wheel encoder readings that tells you how much distance you have travelled, it could be tachometer readings that tell you how fast motors are rotating, it could be speedometers odometers, whatever is the measurement data that you have, all of this could go into one z.

Likewise, as we saw with the state, z1 is the set of measurements I make at time 1, z2 is the set of measurements I make at time 2, so each one of z1, z2, all the way to zt, each one of this could be a set of measurements. And likewise, I am going to assume that you have a set of control actions that you could do. So, it could be a moving the wheels of a robot, it could be a robot motion, in whatever mechanism it is, it could be actions about manipulation of objects, or it could be actions that just, you know, flex joint or something like that. It could be at some granularity of grasping objects; it could be at the granularity of just flexing joints.

There is a variety of things that you could do with respect to these control actions. So likewise, you have control actions. So, u1 to ut so remember that x denotes the state, z denotes measurement data, and u denotes the actual action that I take.

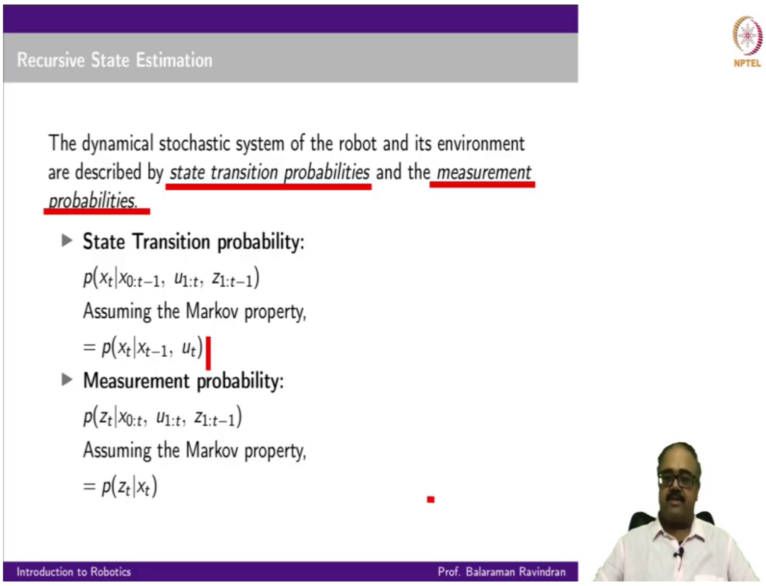
And each of these set of control actions u1, u2, etc. could consists of multiple actions that you are doing, it could have something to do with the robot motion, it could have something to do with manipulation of objects or manipulation of the joint, it could be flexing the joint, or it could be the torque that you supply to motors, it could be about grasping an object, or it could be about moving specific joints in the hand, so the actions could be at multiple different granularities.

And you do not have to perform all possible actions, action components at every time step. So, you have the sequence of actions, again, u1 is the action you took at time 1, u2 is the action or set of actions you took at time 2 all the way up to ut. So, one small notational thing here, I am going to assume that the first state you start out is x naught. So, you know that you are in x naught, then you take action u1 and then after that action you make a measurement z1 because x naught plus u1 would have moved you to x1.

So, this is how the dynamic is going to be, I start in state x naught, I am not making any measurement here, there is some known state typically, some state at x naught that I am going to start at, or sometimes I do not, but we are not measuring anything, we first make an action and then we move to x1, but I do not know what x1 is, all I get from the robot point of view, all I get is a set of measurements, which I call z1. So, from our estimation point of view, the first set of measurements I get or z1 and the first action I have performed is u1, but the state has started out was at x naught.

So, z 1 corresponds to measurements made at state x1, is it clear? So our state's notation is going to run from x0 to xt, where state notation is going to run from x0 to xt, while the action notation will run from u1 to ut, and the measurement notation will also run from z1 to zt. And some, when I want to denote the entire set of measurements, x0 to xt, I will then use this notation x0 colon t. Or if I say x0 colon t minus 1, that means all the way up till t minus 1 but does not include xt, so that is the notation that we will use. Likewise, just like we did for x, we do that for u, and for z as well.

(Refer Slide Time 22:27)



And so, all of this is fine. Now we have the notations for x, u and z but then we really need to model the system itself, so the stochastic system, in fact, we call this as a dynamical stochastic system of the robot is going to be described by two quantities. So, the first I call as the state transition probabilities, the first or the state transition probabilities and then the second set of quantities we are looking at or what are called the, the measurement probabilities. So, we have the state transition probabilities, and we have the measurement probabilities.

So, what do the state transition probabilities tell us, the state transition probabilities tell you, given that I have gone through x0 to xt minus 1 already, this is all the states that I have already visited, 0 to t minus 1. And given that I have taken these actions u1 to ut, I have taken actions u1 to ut and I have made measurements z1 to zt minus 1. Please notice that z1 to zt minus 1, these are all the measurements I have made in the past, these are all the actions I have taken and this is the all the states that I have visited. What is the probability that I will be in a specific state xt?

So, let us assume that the state xt, when the state variable basically measures whether I am in a particular room or not. So, if I say in the past, I was not in the room, and now my action is enter the room, then the probability that I am in the room now should be very high. But there is also some chance that I might not be in the room because the door could be closed or something could have just stopped me from entering the room or the door could be too narrow, whatever it is, so there might be some small chance.

So, I can say the probability that I am in the room, given that I was not in the room and I entered the room is say 0.9 or something like that. Now, if you remember the Markov property, the Markov property says that the history is not important. Therefore, the state transition probability that we have becomes something much simpler, instead of looking at the entire history of states that I have visited, and the actions and the observations I will only look at the last state I am at, xt minus 1.

Notice the difference here, here it was x0 colon t minus 1, which means the entire history from time 0, here it is only xt minus 1, so that means it is the just preceding state and the action ut. Remember, when I am in x0, I take u1, I go to a x1 that is what we, that is what we thought about. So likewise, here, so when I am at t minus 1, I take action ut, so I will end up in xt. So, this state transition probability, so this expression tells us, this expression tells us what is the probability I am going to transition from xt minus 1 to xt when I perform action ut, assuming the Markov property, this is the trace transition probability.

So here is one question that you might have. Hey, why is it that my zt does not figure in this expression, why does zt not figure in this expression because zt really does not cause the transition. If I am in xt minus 1 I take ut I go to xt, I mean a zt is useful for us to estimate what xt should be but zt does not really cause xt. So, the state transition probabilities and the measurement probabilities that we are going to talk about next, describe the system itself, describe what is actually happening in the system.

While zt is useful for me to estimate what is happening in the system, so there is a difference, so I do not worry about the estimation problem right now, I am just defining the system equations. Therefore, in the state transition probabilities, zt does not figure out, is it clear. So, state transition probability is probability of xt given xt minus 1 and ut, so that is what we will use, we will assume that things are Markov throughout.

Even though we will go back to the non-Markov case and then when we are simplifying things, we will make the assumption of Markov property. And therefore, things fall out, just pointing out to you that most of the development that we will do for the rest of the course, assume that your system is Markov. So, the second component that we spoke about, that we mentioned earlier it the measurement probability. So, what is the measurement probability, it is a probability that I will make a specific measurement zt given the history of states x0 to t, the history of actions you want ut, and the history of observations z1 to zt minus 1.

So, this is basically my measurement probability, this tells me, what is it that I will see once I have gone from x0 to xt and taken actions u1 to ut and made the previous observations, z1 to t minus 1. So, I am using observations and measurements interchangeably. So, bear with me. And so that is basically the other measurement probabilities, that is the second component of our system. Assuming that you have the Markov property, we know that the observation that you make at time t depends only on the state at time t, it does not depend on what action I did to come to xt.

So, even ut is no longer relevant for looking at the probability of zt because ut's impact on zt is only through what xt occurs. Once I know what is xt, I do not need to know what action bought me to xt and I certainly do not need to know the previous observations and the previous states and the previous actions I took. So, I can simplify the measurement probability significantly into probability of zt given xt.

So, notice that we have now written out our system dynamics in terms of a state transition probability, which is probability of xt given xt minus 1 and the action you took ut, and the measurement probability, which is probability of zt given xt. And we will be repeatedly using both of these system equations in order to derive multiple quantities in the next few lectures. Thank you.