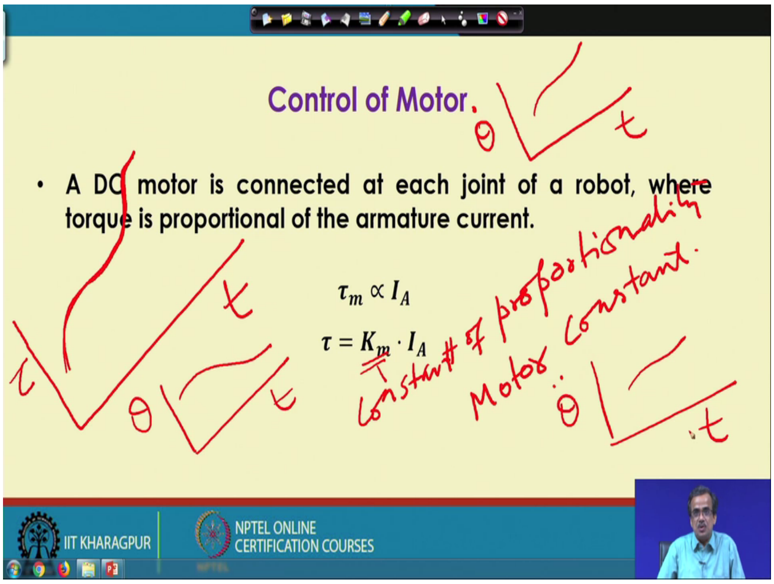
Robotics

Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 30
Control Scheme

We are going to start with a new topic that is topic 5, it is on Control Scheme. Now we have seen starting from the kinematics like how to derive the expression for the joint torque and at each of the robotic joint, we put a motor and this particular motor is going to supply that particular the torque.

(Refer Slide Time: 00:44)



We generally use the DC motor and this particular DC motor is connected at each of the robotic joint. And here the torque generated that is proportional to the armature current.

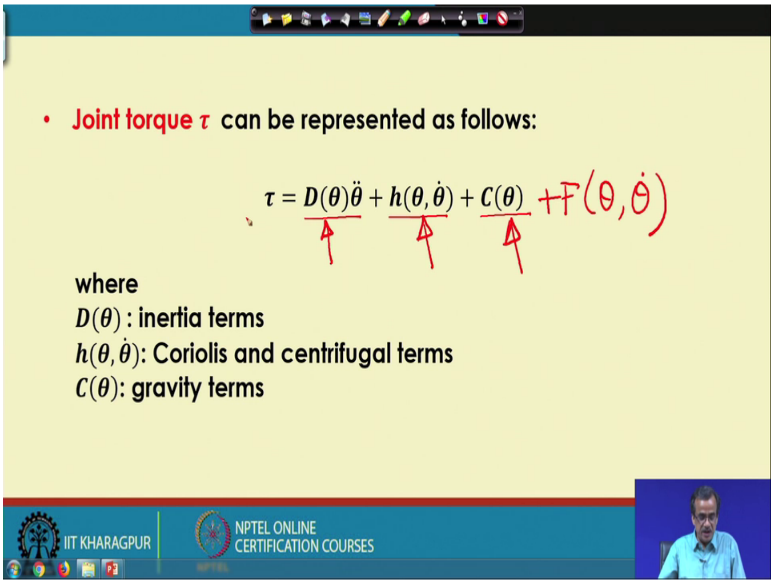
Now, if you see the motor torque that is denoted by say tau m and the armature current is I A. So, tau m is proportional to I A that is the armature current and this can be written as so this can be written as your tau equals to K m multiplied by I A. Now this particulate K m is nothing, but the constant of proportionality. So, this is nothing, but the constant of proportionality and this is also known as your the motor constant. So, this is also known as the motor constant.

So, this tau m is nothing, but K m multiplied by I A. Now here actually what will have to do is at each of the robotic joint will have to generate this particular tau, that is the torque as a function of time for example, if I plot for E particular robotic joint. So, this is the joint torque, so this particular joint torque as a function of time so that will have to generate. Now supposing that this particular distribution is something like this a very random distribution I have considered.

And at the same time what will have to ensure is the joint, the angle that is theta as a function of time. So, there must be some continuous curve something like this and at the same time the first time derivative of theta that is your theta dot as a function of time and your theta double dot that is acceleration as a function of time so, some plot will have to find out some distribution will have to find out.

Then how to ensure that these particular the motor, the DC motor is going to generate this amount of torque with time. This is what is required if I want to or create some angular displacement at the robotic joint in a particular the cycle time, so will have to ultimately generate this particular theta as a function of time, theta dot as a function of time, theta double dot as a function of time. Now I am just going to discuss how to ensure that this particular DC motor is going to generate so this torque within this cycle time.

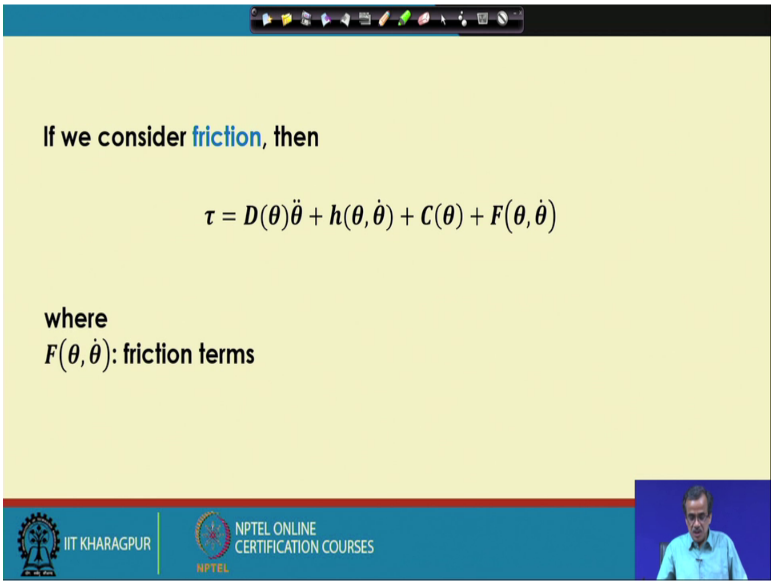
(Refer Slide Time: 03:42)



Now let us see how to generate this particular torque. Now to generate this particular torque let me once again go back to the expression of the joint toque which I have already discuss while discussing the dynamics. That this particular joint torque tau is nothing, but D theta theta double dot plus h theta theta dot plus C theta. Now I have already discuss that this is nothing, but the inertia term this is Coriolis and centrifugal term and this is the gravity term and if I add the friction term.

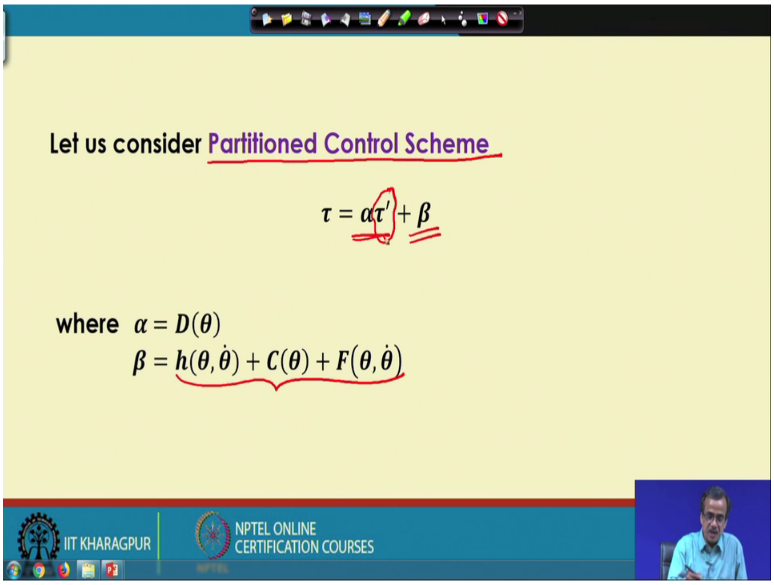
So, this frictional terms is to be added here, which I am not adding for simplicity, but we can add this particular the friction term. Now this particular torque has to be generated by the motor so how to generate that I am going to discuss.

(Refer Slide Time: 04:38)



Now, to generate this particular torque, so what will have to do is.

(Refer Slide Time: 04:43)

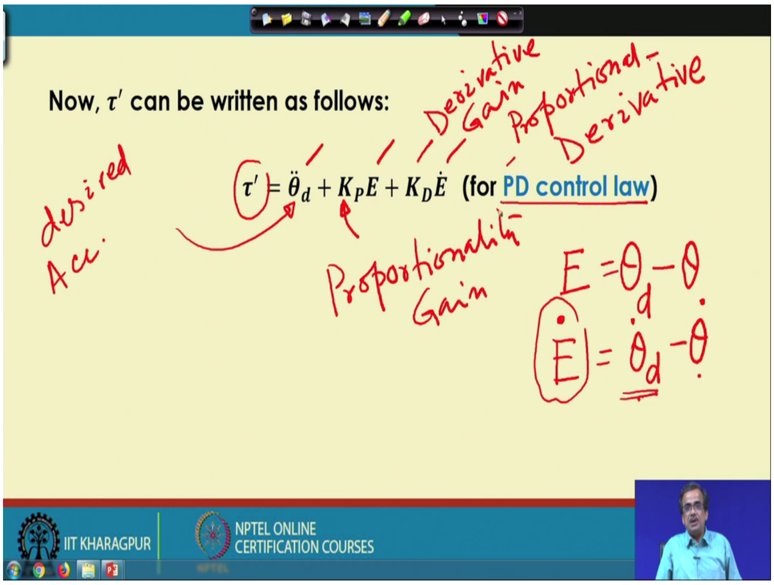


So, will have to consider a particular control scheme will have to take the help of a control scheme. Now here if you see the literature we have got a few very popular the control scheme and out of all such control scheme this is the most important one, and this is known as the partition control scheme.

Now, here in partition control scheme what I do is the torque to be generated that is tau that is distributed that is divided into two parts one is nothing, but alpha into tau prime so this is one part and another is the beta. Now if you say that expression of the joint torque so this h theta theta dot plus C theta plus F of theta theta dot, so this particular terms taken together we call it beta and then alpha is nothing, but D theta, D theta is nothing, but that inertia terms ok. And alpha primes or this particular tau prime that will have to generate with the help of a controller. Now each of this particular motor is having its own controller.

Now, let us try to explain how can it generate, so that particular tau prime with the help of its controller and this particular controller is inbuilt. So, let us try to explain that how to generate that particular the required tau prime.

(Refer Slide Time: 06:15)

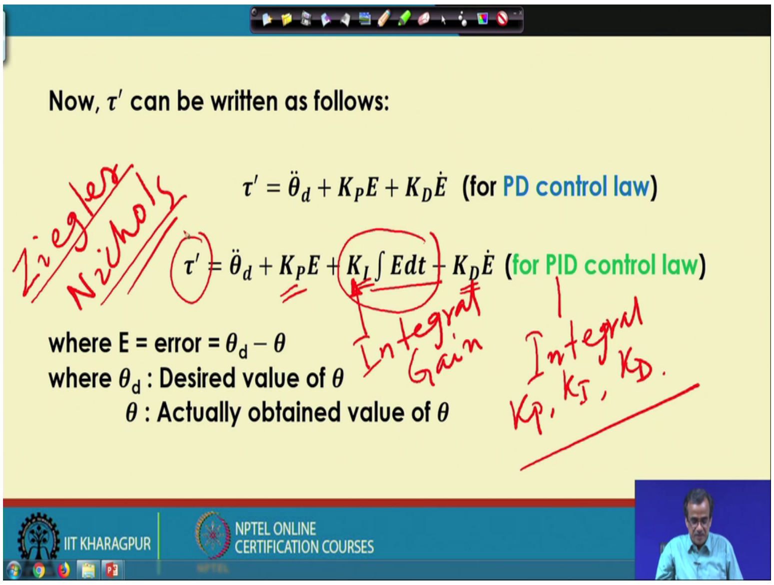


Now, this tau prime can be written as your theta d double dot plus K P into E plus K D into E dot, if I consider PD control law. Now PD control law means Proportional Derivative control law. So, proportional derivative control law, derivative control law that is called the PD control law. Now here I am using two symbols one is called this K P now K P is nothing, but actually the proportionality gain.

So, this is nothing, but proportionality gain value and this K D is nothing, but the derivative gain value, derivative gain. Now here I am also using the terms like E, E is nothing, but the error and that is the difference between the desired theta and the theta which is actually created with the help of the motor. So, this is nothing, but theta and this particular difference is nothing, but the error E and E dot that is your that rate of change of this particular error or we can do something like this, we can find out the deviation or the difference between the angular the velocity that is theta d dot minus theta dot.

So, this is nothing, but actually what I do is, so this is the desired angular velocity and this is the actually obtained angular velocity and this particular difference is nothing, but is your E dot. And what is this theta d double dot? Theta d double dot is nothing, but the desired acceleration so this is nothing, but the desired acceleration. Now let us say how to generate this particular the tau prime using this PD control law.

(Refer Slide Time: 08:44)

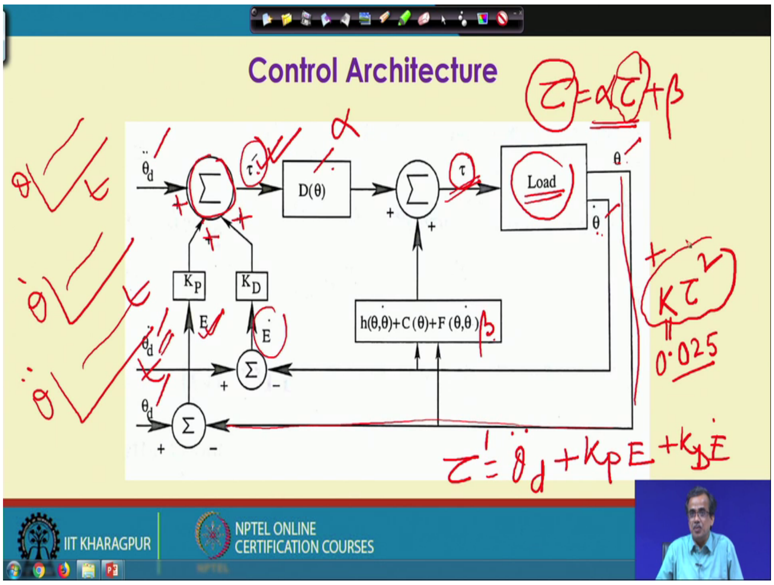


Now before I go for that let me just tell you that if I use PID controller in place of the PD controller, then tau prime will become equal to theta d double dot plus K P into E plus K I into integration Edt plus K D E dot for the PID controller that is Proportional Integral Derivative controller. So, P stands for promotional, I stands for integral so this is integral. So, proportional integral and derivative control law and here we are adding one extra term that is your K I multiplied by your that integration Edt.

Now, here this K I is nothing, but your integral grain. So, this is nothing, but integral gain value. Now the values for this particular K P K I and K D are in fact, can be determine mathematically also, and there is one well known method that is called Ziegler Nichols rule, that is called Ziegler Nichols rule.

Now using this Ziegler Nichols rule, we can find out what should be the numerical value for this particular K P, K I and K D. Now once I have determine the values for this K P K I and K D those values are kept constant those are not altered. So, they are using this Ziegler Nichols method we can find out all such K P, K I and K D values and once you have got that now I can implement, so this particular the tau prime.

(Refer Slide Time: 10:39)



Now, let us see how to implement so this particular your how to get this particular tau prime. Now let us see the controls architecture and this is the block diagram of the control architecture. Now this particular D theta as I mentioned, so this is nothing, but is your alpha and this tau as you mentioned according to this partition control rule this is alpha tau prime plus beta. So, alpha is nothing, but is your D theta and this particular thing whatever you have written here h theta dot C theta plus F theta theta dot this is nothing, but the beta ok.

So, this is alpha this is beta now I will have to generate this particular the tau prime. Now the way it is done is as follows, we take the help of some set of the closed loop control system. So, initially there could be some error, but this particular error will be compensated. Now here I have mention theta d double dot that is nothing, but the desired acceleration, then comes your theta d dot that is the desired velocity and theta d that is nothing, but the desired your the displacement or the angular displacement. Now here with the help of these we are just going to generate this particular the tau prime. Let us see how to generate this particular tau prime.

Supposing that say this is the summing junction say I am passing, so this particular the tau prime now how to determine this particular tau prime according to this PD control rule? Now as I have discussed that this particular, the according to the PD control rule so this particular tau prime is nothing, but E dot multiplied by K D plus your this K P multiplied by E plus theta d double dot. So, let me write it here, so this particular tau prime is nothing, but theta d double dot. Now this is the summing junction and we can see I am putting three such plus sign here. So, theta d double dot is coming from their plus K P multiplied by the error.

So, K P multiplied by the error, so this particular thing next is your K D multiplied by E dot, so this K D multiplied by E dot so those things are sum term here and that is nothing, but is your tau prime. And once you have got this particular tau prime we multiply tau prime by alpha so I will be getting alpha tau prime plus beta so I will be getting the complete tau. Now here we have got a load, load means this is the mechanical load so, if it is a robot now actually the load to be carried to generate that particular the angular displacement that is nothing, but the mechanical load. And supposing that I am using a DC motor, DC motor is generating this particular the torque. That means, it is generating theta theta dot at least theta dot you forget about theta double dot for the .

So, the moment I am just applying I am just putting that particular motor on. So, what will happen is it will try to generate this particular the torque. That means, you are the physically how to, how to realize that the torque has been implemented will be getting the theta theta dot and all such things. And using some sensor we can measure this particular theta and theta dot, how to measure theta and theta dot that I will be discussing after sometime.

Now, supposing that we are able to measure so this particular theta and theta dot, so this theta and theta dot. So, that will be brought here to this junction for the purpose of comparison. So, theta will be brought here for the purpose of comparison with theta d that is the desired theta and I will be getting this particular error, and this error will be multiplied by K P and it will be added here.

Similarly, whatever theta dot we are getting that we can measure. So, this will be brought here for the purpose of comparison at this particular the summing junction. And it will be compared with your theta d dot and will be getting this E dot and this E dot is multiplied by K D and that is also be summed up here. And I will be getting this particular the tau prime. So, this process will go on and go on. So, in the first cycle we may not get the accurate theta and theta dot, but as I told that we are going to use the closed loop control system. So, there will be error compensation and at this particular robotic join we are going to generate so, this particular theta theta dot accurately as a function of time.

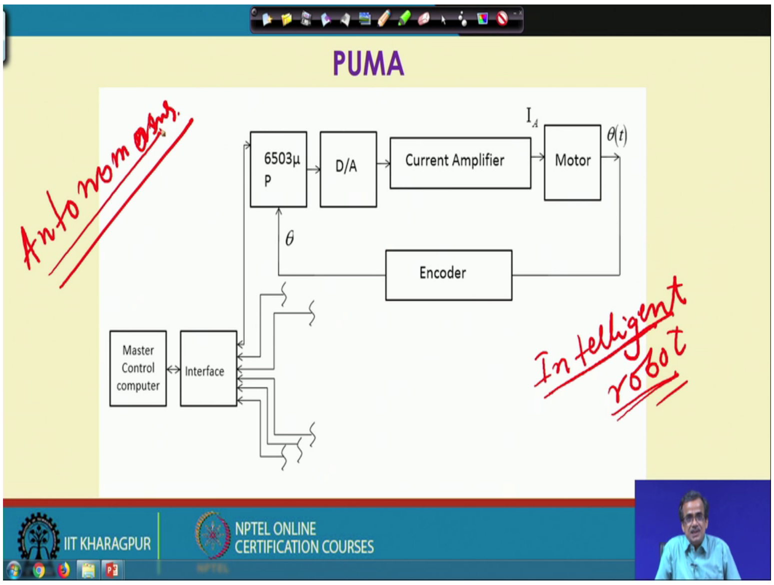
And that is why as I mentioned that will be getting the theta as a function of time, theta dot as a function of time and of course, you will be getting theta double dot as a function of time ok. So, will be getting sum distribution here ok. So, this particular joint torque will be realized in the form of theta theta dot and theta double dot ok.

So, this is the way actually we can generate the desired motion at the robotic joint with the help of your the DC motor, but this particular DC motor is having some loss. So, whenever we try to calculate the power rating for this particular DC motor which you are going to put so that particular loss will have to consider. Now, if I know the torque history, and if I know this particular your theta dot as a function of time, so, very easily we can find out what should be the power rating.

And if I know this particular power rating so we can prepare the specification of this particular the motor which I am going to put at the robotic joint, but as I told while preparing the specification will have to consider that one is the torque required and another is the loss of toque, and this particular loss of torque is generally we try to calculate loss of torque is nothing, but K tau square. So, what about torque is required to generate this theta dot and theta double dot and all such things after that I will have to add this particular the loss that is K tau square, tau is nothing, but the torque and K is the constant.

Generally we considered a small value for the DC motor. So, 0.025 or so and we try to find out this particular loss, and then we decide what should be the power rating for this particular the motor. So, this is the way actually we control the different joints of the robots with the help of DC motor.

(Refer Slide Time: 18:02)



For example say if you consider the puma the way we control puma. So, here I am just going to discuss briefly the control architecture for this particular puma that is a programmable universal machine for assembly. Now here there are 6 joints all 6 are rotary joints and at each of the joint actually we have got this type of the control system. So, if we see the control architecture for each of this particular joint.

So, this is actually the block diagram for the control architecture for a particular the joint. Now here you can see that we have got that 6503 microprocessor. So, to control a particular joint, so I have got a motor and to control that actually we use this type of control architecture or the control scheme. So, we have got the microprocessor here then this digital to analogue that conversion.

Then we have got the current amplifier because this armature current is going to enter and will be getting the joint torque. And this torque will be generated and it will be realized in the form of your theta as the function of time that is the joint angle. Now here I am using one encoder so this particular encoder optical encoder is nothing, but the feedback device.

Now, with the help of this optical encoder we can determine what should be the joint angle and so this particular this is compared with the desired value. So, if there is any such error so that particular error will be amplified here and will be getting some current here armature current, and once again it will generate and this particular error will be compensated. So, at each of the robotic joint will be getting very accurate movement with the help of this particular the motor.

Now here, for this puma there are six motors, now this is the control scheme for a particular the joint. Similarly, for the second joint I have got another such control scheme, third joint, fourth joint, fifth and sixth joint. And all such movement of the joints that will be controlled by one centralized computer and that is your the master control computer.

So, here to control this particular puma, we have got one controller or the director and we can use one master control computer. So, with the help of this master control computer the all the movement of all the joints actually it can be the control. So, this is the way actually we control this particular the puma. Now here so till now whatever I have discuss so let me tell you in short like your till now starting from the kinematics. So, we have discuss how to carry out the dynamic analysis and we have seen how to generate that particular the torque with the help of your say DC motor. And the motor will be equipped with one controller, and generally we use either PID controller or PI controller or PD controller and once you know the gain values of this particular controller so we can control the movement at the different robotic joint.

So, till now actually that that the robot is ready and we have already discuss how to teach a particular robot. So, if I just want to give a task to the robot, so we are in a position to give task to the robot and the robot will try to follow that and try to perform that particular the task. So, till now so whatever we have discuss is this, but one thing we have not a discus can a robot take decision, how can you make the robot capable of taking decision?

So, that particular thing we have not yet discuss that means, how to make a robot intelligent? How to make a robot autonomous? What do you mean by robot intelligence? Those things we have not yet discuss, we have not yet discuss in this course.

So, gradually actually we are moving towards that how to make that particular robot intelligent. Now before I go for that particular intelligent issues in robotics that is actually the fourth module or the last module of robotics. Now let me tell you what do you mean by an intelligent robot. So, by an intelligent robot actually what is mean is your so intelligent robot will be able to take the decision as the situation demand.

That means, in a varying situation in a varying environment and intelligent robot should be able to take the decision, and there is a another term that is called the autonomous robot. Now this autonomous robot is actually, this autonomous robot is actually your a robot which has got the permission to perform as the situation demands.

So, if it is having if the robot is having the ability to perform in a varying situation that may be called an intelligent robot. However, if it if it is having the permission to perform in the intelligent way or take the decision in the varying situation. So, if it is having that permission then only it is called the autonomous robot. Now all the autonomous robot should be intelligent robot, but all the intelligent robot may not be autonomous.

Now let me take a very simple example just to find out the difference between the intelligent system and autonomous system. Now, if you see that under one university there are 10 engineering colleges. Now these engineering colleges will have to follow the rules and regulations of this particular the university. Now at each of the engineering colleges there could be intelligent people, faculty members, students, but there unable to take any decision they will have to depend on the university rules.

So, they are intelligent, but they are not autonomous on the other hand if you see the institutes like IIT, NITS they are intelligent at the same time autonomous, there having the capability to take the decision and their having the permission also to take the decision, they are intelligent they are autonomous. In robotics actually we call it is intelligent and autonomous robot, now how to design and develop and intelligent and autonomous robot.

So, those issues actually will be discussing in details one after another and as I told that we copy everything from human being in robotics we also try to copy the intelligence. That means, the way we collect information, the way we take the decision the way we implement all such decision all such things in fact, we are going to copy in the artificial way in the intelligent and autonomous robot. Now if we see we collect information with the help of sensors, but the robot doesn't have any such sensor.

So, will have to put some sensors for example, say will have to put some camera will have to put some sensors and with the help of this sensors and cameras, the robot will be able to collect information of the environment. And once it is got that particular information, now it will try to do the analysis try to use some sort of motion planning algorithm which I am going to discuss, what should be the course of action.

And depending on the course of action now the movement should be there the movement of the wheels of the robot should be there or the movement of the different links or the limp should be there. Now how to move it actually we take the help of motor, we take the help of controller just to get that particular the motion implemented. So, here actually to make it intelligent, so all such issues will have to discuss one after another. So, we are going to discuss in details how to make the robot intelligent.

Thank you.