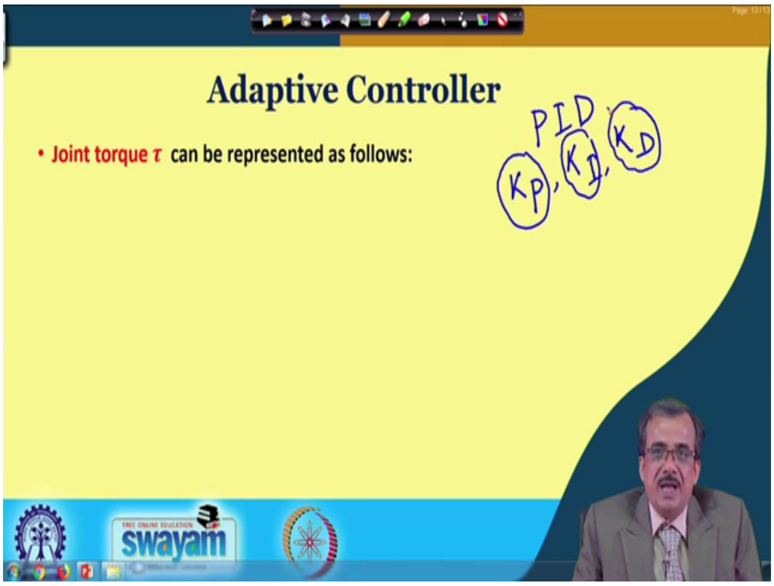
Fuzzy Logic and Neural Networks
Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 40
A Few Applications (Contd.)

(Refer Slide Time: 00:15)

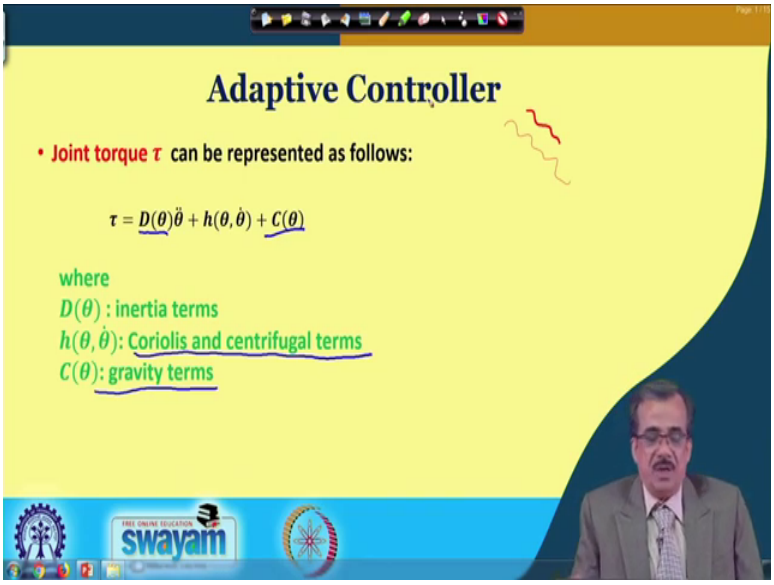


Now, we are going to discuss how to design and develop adaptive controller. Now, in robotics what we do is at each of the joint we use some DC motor and to control the motor we use some controller. Now, I have already mention about this PID controller that is your proportional, integral and derivative controller. Now this PID controller actually what we do is we try to find out some gain values like K P then comes your K I and K D. So, K p is nothing but the proportional gain, K I is the integral gain and K D is nothing but the derivative gain and I have already mentioned that using some principle. So, mathematically we can find out like what should be the values for these particular your K P, K I and K D.

Now, before I discuss further let me tell you that how to determine and what are the different components of a robotic torque, the joint torque. Now, let me let me tell you one thing like if you have not taken any course on robotics or if you do not have any fundamental information of the robotics you need not worry much. Now, the things which I am going to discuss is how to use the principle of fuzzy logic and neural network to design and develop the adaptive controller.

So, you simply try to understand how to use the principle of fuzzy logic and neural network just to design and develop the adaptive controller. You need not understand 100 percent the working principle of a PID controller, but let me spend some time on that so that you can get some idea. But my main emphasis is how to design and develop the fuzzy logic based or the neural network based your the adaptive controller.

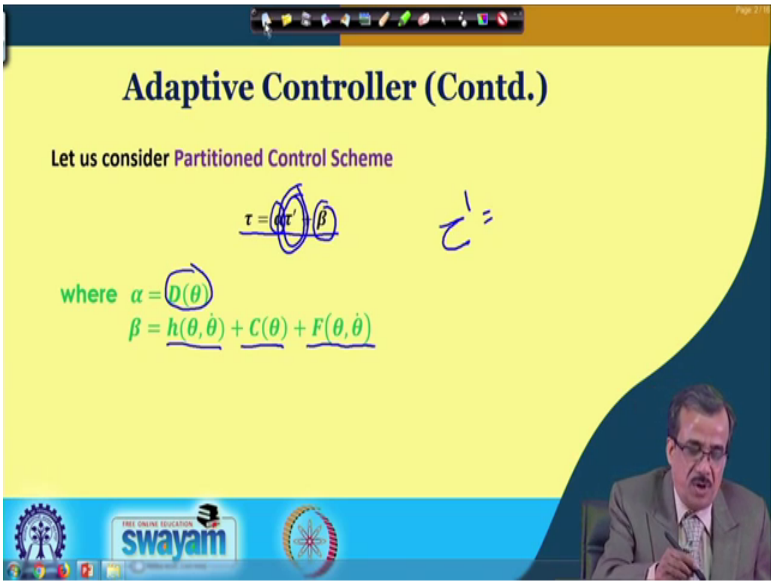
(Refer Slide Time: 02:26)



Now, if you see the joint torque of a robot it has got some components like your, like tau is nothing but your D theta theta double dot plus h theta theta dot plus C theta. Now, if you see this particular expression this is actually the joint torque, like if I want to operate the robotic joint with the help of a DC motor, the motor will have to generate this amount of torque. Now, it has got 3 components. Now, the components are as follows, like your the D theta is nothing but the inertia terms then comes h theta theta dot is nothing but the coriolis and centrifugal term and C theta is nothing but the gravity terms.

Now, I am not going for the derivation of these particular terms, actually you will be getting a very big expression. So, that we discuss in robotics course not in these particular course, but the thing which I am going to tell you that how to generate so this particular tau with the help of a motor using the principle of your adaptive controller. So, that I am going to discuss.

(Refer Slide Time: 03:50)

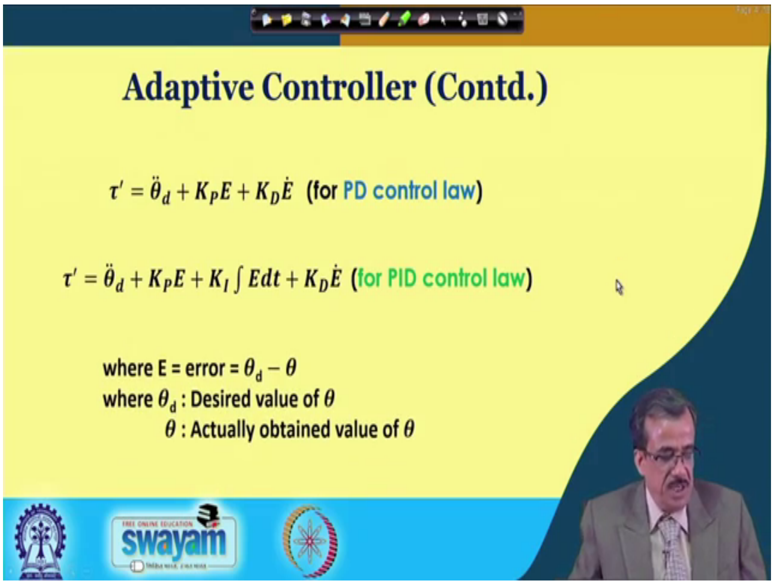


Now, if you see generally we use different types of control scheme, but out of all the control scheme actually the most popular one is the partitioned control scheme. Now, in partitioned control scheme actually what we do is your we use these particular formula that is tau is actually the torque to be generated by the motor mounted at the robotic joint and that is nothing but alpha tau prime plus beta. Now, this alpha is nothing but the inertia term that is D theta which depends on the cross section of the robotic link, it depends on the moment of inertia of this particular the robotic link. So, I am not going for detailed discussion on this.

Now, this beta is nothing but your Coriolis component then comes your gravity terms which depends on the acceleration due to gravity and this is nothing but F theta theta dot is the friction torque. Now, let us try to concentrate or let us try to find out how to determine these particular tau prime that is actually our main aim.

So, how to generate these particular the tau prime so that the motor mounted at the robotic joint can generate that particular the motion that is the joint angle that is the theta accurately and it can provide that particular the torque, the required torque.

(Refer Slide Time: 05:28)

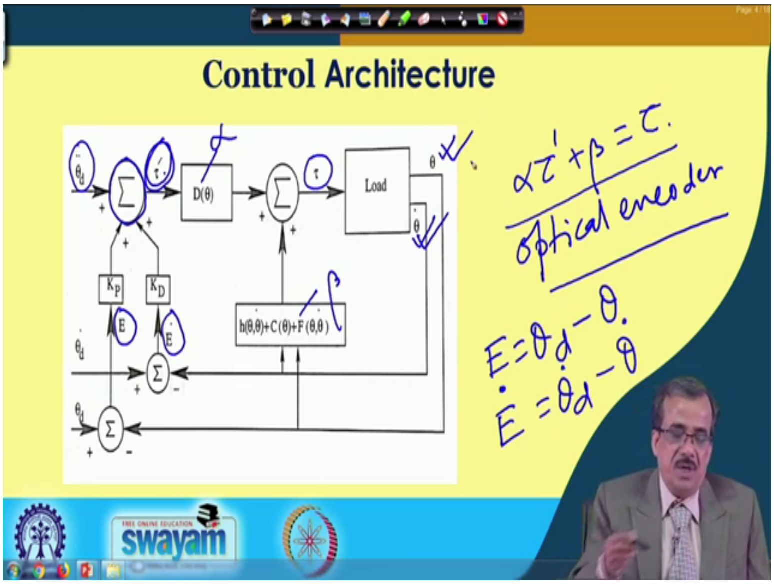


Now, let us try to concentrate on your the control scheme like how to develop or how to implement that particular your the tau prime. Now, f i is the PD control law; that means, your proportional and derivative control law then tau prime is nothing but theta d double dot plus K P multiplied by E, E is nothing but the error plus K D multiplied by E dot that is actually your you can say the rate of change of error.

Now, here you can see your. So, this tau prime is nothing but theta d double dot plus K P multiplied by E plus K D multiplied by E dot. Now, theta d double dot is nothing but is your the desired angular acceleration. Now, let us see how can it generate. So, these particular things or how to generate these particular your the tau prime. Now, if I use the PID controller that is proportional integral derivative. So, tau prime is nothing but theta d double dot plus K P into E plus K I integration E d t plus K D E dot.

Now, this K I is nothing but the integral gain, K P is the proportional gain and K D is nothing but your the derivative gain. Now, let us see how to implement. Now, if you see your the architecture of these particular the controller.

(Refer Slide Time: 07:08)



So, you can see that. So, this is actually the control architecture. Now, here so, our aim is to generate. So, this theta d double dot that is the desired acceleration and here we have got one summing junction.

So, we have got a summing junction here and these tau prime will have to generate and the tau prime actually you can see that we have got these particular D theta. So, these D theta is nothing but alpha as I discussed and your h theta theta dot plus C theta plus F theta theta dot is nothing but the beta and if you remember it is nothing but alpha tau prime plus beta is nothing but is your the torque.

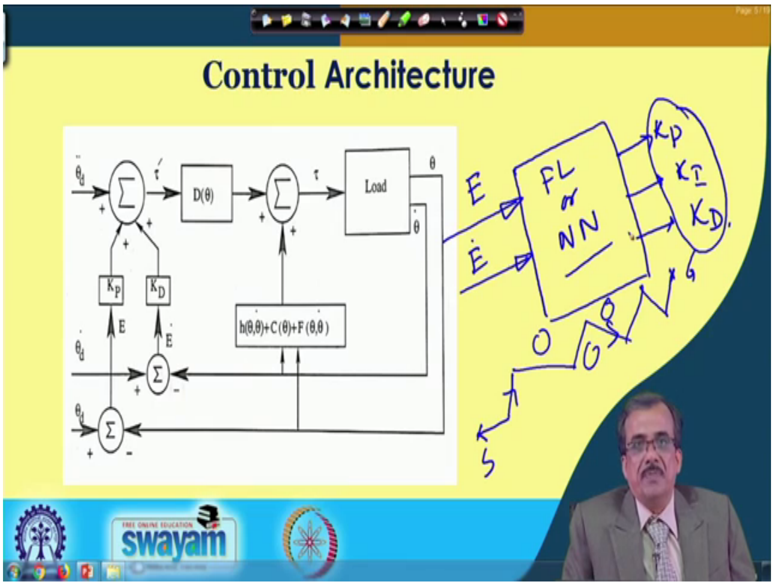
Now, here if I know these particular torque that is your alpha tau prime alpha multiplied by tau prime plus beta is nothing but the torque. So, these particular torque has to be generated, now by the motor and the controller is going to help. Now, if I put one load here, the mechanical load. So, I will be getting some angular displacement like theta and angular velocity like theta dot on the output of that particular the motor. Now, what we will have to do is we will have to measure. So, these particular theta and theta dot for example, we can use some optical encoder.

So, some optical encoder you can use just to find out what should be these particular your theta and theta dot also you can measure. Now, these particular theta and theta dot will be brought to these particular junction. And so, theta will be compared with theta d and will be getting what is these particular the error. So, these error is nothing but is your theta d minus your theta and similarly here you can find out your E dot and this E dot is nothing but is your theta d dot minus your theta dot.

So, this particular theta dot will be brought here for the purpose of comparison and we will be getting these particular your E dot then E dot multiplied by K D plus your E multiplied by K P plus theta d double dot is actually your what is these particular your tau prime. So, we can find out these tau prime and we use the closed loop control system. So, you will be getting these particular the accurate the joint angle and your theta dot.

So, this is the way actually so this control architecture works. Now, my question is how to use the principle of fuzzy logic and neural network just to design and develop the adaptive controller. So, that I am going to discuss now. Now, to design that adaptive controller so that it can generate your the gain values K P, K I and K D in an adaptive way.

(Refer Slide Time: 10:38)



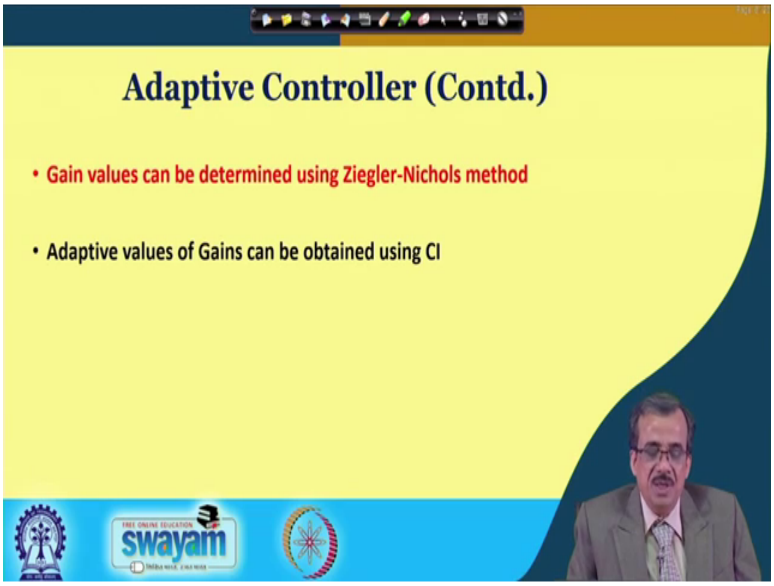
So, what you will have to do is you can design one fuzzy logic system or one say multi layered neural network you can utilize. So, what you can do is here the inputs will be your E that is the error and your E dot that is you can say rate of change of these particular error sort of thing and what should be your the gain values.

So, if I use PID. So, here there will be K P, K I and K D. Now, depending on these particular error and E dot your you can find out what should be the K P, K I and K D.

Now, supposing that the robot is starting from here, say point A and it is going to reach the goal. Now, there could be a possibility that in one time step I am using one set of K P, K I and K D for another time step another set of K P, K I and K D, another set of K P, K I and K D. And by doing that say it is going to reach that particular the goal and supposing there could be some obstacle here just to avoid the obstacle might be this is the path and to move along these particular path supposing that we need some sort of adaptive values for this K P, K I and K D.

Now, if I use fuzzy logic base system or say neural network base system there is a possibility I will be getting the adaptive values for these gain; that means, I will be able to develop some sort of adaptive controller for these particular the motor.

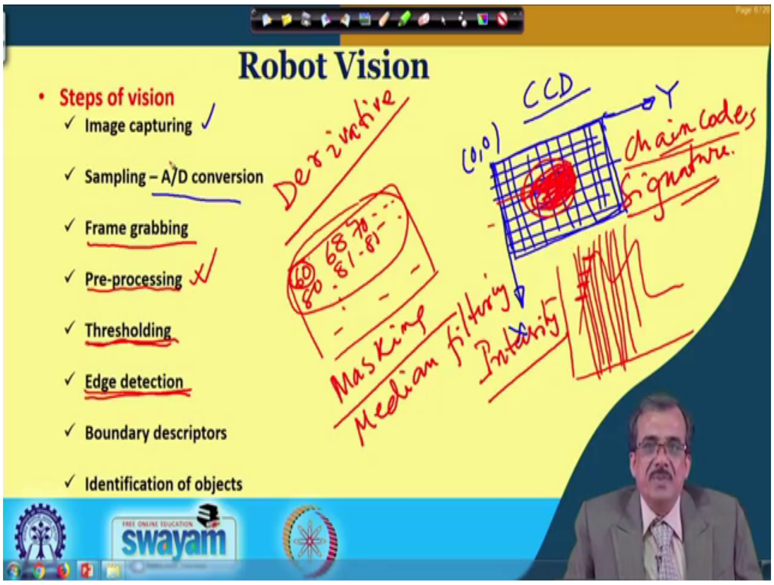
(Refer Slide Time: 12:17)



Now, this is the way actually we can design and develop the adaptive controller for these particular the motor. And that is the purpose and the traditional method I have already discussed that is the Ziegler Nichols method, but here we will be getting only a set of fixed values for the gain.

Now, if you want to go for the adaptive values for this particular gain. So, what you will have to do is. So, you will have to go for some sort of tuning using the computational intelligence that is CI or you can develop some sort of adaptive controller using either fuzzy logic or neural networks.

(Refer Slide Time: 12:57)



So, this is the way actually we can design and develop your the adaptive controller for the robot. Now, I am just going to discuss like how to use the concept of the fuzzy logic and neural networks in robot vision or the computed vision. Now, once again let me tell you that if you do not know the fundamentals of robot vision or computed vision, you need not worry. Although, I am just going to tell you the different steps used in the robot vision in short you need not have the proper knowledge of these particular digital image processing or robot vision or computer vision, the thing which I am going to concentrate how to use the principle of fuzzy logic and neural network.

So, that we can design and develop very efficient your the computer vision system or the robot vision system. Now, before I go for that let me try to understand the physical problem the physical problem is as follows with the help of camera. So, the robot is collected some information. Now, it will have to interpret whether it is object 1 or object 2 or if it is another planning robot. So, that type of information it will have to collect of these particular environment.

So, how to do it now what you do is we follow a certain steps for these robot vision or the computer vision. Now, the steps are as follows. So, what do you do is we take the help of some CCD camera and we go for your image capturing. Now, I will be discussing each of these particular steps in brief. Now, let me let me just try to spend some time on discussing the fundamentals of this, then I will try to find out where are the areas where you can take the help of fuzzy logic and neural networks just to develop more efficient robot vision system or the computer vision system.

So, the first step is your image capturing. So, what I do is we used one CCD camera like charged couple device camera just to collect information of these particular the environment. Now, the quality of these particular image depends on a number of parameters for example, it depends on the level of illumination while taking that particular the picture or the snap. It depends on the calibration of these particular the camera, it depends on the angle of vision like the angle through which I am looking at that particular the object and so on.

So, there are many other factors also which are going to control the quality of the captured image. So, there is enough fuzziness in the quality of these particular your the image the collected image. So, there is a chance that we can inject the principle of fuzzy sets just to tackle so that particular the fuzziness in the captured image. Now, once you have got these particular the image collected with the help of camera then we go for some set of analog to digital conversion, because ultimately we will have to write down the computer program and computer does not know anything of the your these statement.

So, you will have to use some numbers. So, what I do is based on the collected information with the help of camera. So, what you do is. So, supposing that that particular image is actually put on the computer screen and in the computer screen. So, I have got one coordinate system say X and Y this is nothing but the origin say it is 0 0.

Supposing that this is nothing but the computer screen so the whole computer screen that is actually divided into a large number of small elements and these elements are known as actually the picture elements or the pixels and supposing that here. So, I have got one your the object. So, let me consider that here I have got one object sort of thing. Now, here although I am showing it is a color one let me concentrate that this is nothing but the black and white object. Supposing that whatever I have drawn the background say this is in say white and these particular rate supposing this is the black.

So, this is nothing but the black and white object say another computer screen. So, I will be getting these particular the object, now what you do is. So, this is divided into a large number of pixel. So, I do the scanning, I do the scanning along Y direction X direction. So, there is a possibility. So, I will be getting some distribution of the light intensities.

So, this is nothing but the light intensity value, now if it is the black object the light intensity will be less and if it is white object the light intensity will be more.

Now, here in sampling actually or the A D conversion. So, what we do is we do the sampling and so, that corresponding to the different pixel I should be able to find out what is the numerical value; that means, corresponding to these light intensity. So, I am just going to find out what should be your the light intensity values; that means, corresponding to this image. So, I will be getting one matrix of your the light intensity values. So, might be the values could be 60, 68, 70 and so on 80, 81, 85 and so on. So, this type of numerical values we will be getting and these are nothing but the light intensity values at the different pixels and this is what do you mean by the frame grabbing.

Now, what is our aim? Our aim is your the robot should be able to identify that this is object 1, this is object 2 lying in these particular the environment. So, we are here. So, corresponding to these particular image. So, we have got some number in the matrix form and these are all light intensity values, then we take the help of your pre processing, as I told several time that these particular collected image or the collected light intensity values will have some imprecision. There will be a lot of errors and now on there could be lot of noise.

Now, we want to remove these particular errors or the noise and that is why we take the help of your pre-processing, there are many standard methods of pre-processing like your one is call the masking operation, then we have got the neighborhood averaging, then median filtering we can use. So, median filtering so these are all standard methods of pre processing and once again. So, this pre processing can be tackled using the principle of your the fuzzy logic.

Now, then comes your the thresholding. The purpose of thresholding is actually to differentiate or to find out the difference between the object from your the background. So, we try to find out the difference between the object and background and this is what we mean by your thresholding. So, there will be some thresholding value for light intensity and using that particular thresholding value for light intensity. So, we can find out the difference between the object and these particular the background.

So, the computer screen. So, you will be able to see there are some black spots or black region if it is black and white type of the image processing problem. So, after the thresholding is over. So, approximately we will be getting some sort of the image, some sort of object on the computer screen and that actually finds the difference between the object from it is background. Now, once you have got approximately those objects lying on the background now we can go for some sort of edge detection.

Now, in edge detection actually we take the help of some derivative operator. So, we use derivative operator just to find out the rate of change. So, the by derivative we means rate of change such the same thing we use here just to find out, just to detect the edge of these particular object and to find out the edge between the object and these particular the background. And once again there could be a lot of applications. Using the fuzzy logic and neural networks for these handle the edge detection problem. And once you have got the edges, now we will have to describe the boundary so that we can do further processing of these particular the object.

Now, for carrying out the further processing so we will have to express these particular boundary and this boundary can be represented using some there are some standard methods like one is called the chain code method. So, we can use the chain code method or we can use some sort of signature. So, mathematically you can represent the boundary of these particular your the descriptor and once you have explained or once you have expressed the boundary of these particular obstacles or that the objects with the help of the mathematical expression like signature or with the help of this chain codes.

Now, we can do some sort of further processing and then we will be able to identify this is object a, this is object b which are lying in these particular the environment. This is in short the principle of the robot vision or the computer vision so if the robot to ones to collect information of these particular the environment. So, all such steps are to be followed within a fraction of second which is bit difficult because each of these particular steps we will take some CPU time and it will be computationally expensive.

Now, actually what you can do is we can use the principle of these fuzzy logic or we can train some network. So, just to implement each of these particular modules so that once it is trained within a fraction of second. So, these particular the robot will be able to identify. So, this is object a, this is object b, this is object c and so on. Now, here I just want to mention that a few studies have been reported on each of these particular areas using the principle of your the fuzzy logic and neural networks, but there is enough scope for further development. And what you need is actually one package and these particular package will contain your the fuzzy logic based or the neural network based modeling of each of the steps of these particular the computer vision system or the robot vision system.

Now, once it is developed using the principle of your the fuzzy logic and neural networks and once it is properly trained there is a possibility the robot will be able to visualize the environment with the help of. So, this type of your the tools and techniques within a fraction of second and once it can visualize that that particular object and environment then, it will be in a position to go for actually your the motion planning.

Thank you.