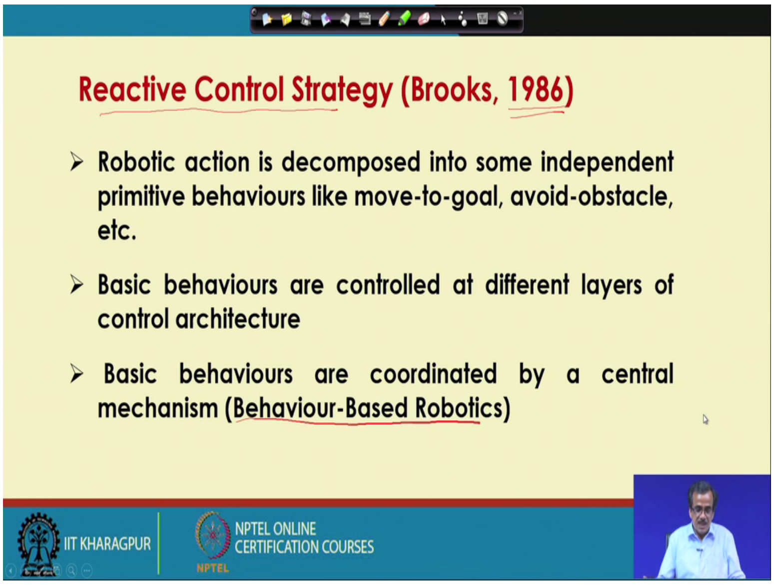
Robotics
Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 40
Robot Motion Planning (Contd.)

Now I am going to discuss the principle of another very popular the motion planning algorithm, which is known as the reactive control strategy.

(Refer Slide Time: 00:23)



Now this reactive control strategy it was proposed in the year 1986 by Brooks. Now here a robotic action is divided into a large number of independent primitive or basic behaviours. For example, say supposing the robotic action is in a soccer plying robot is says to score goal.

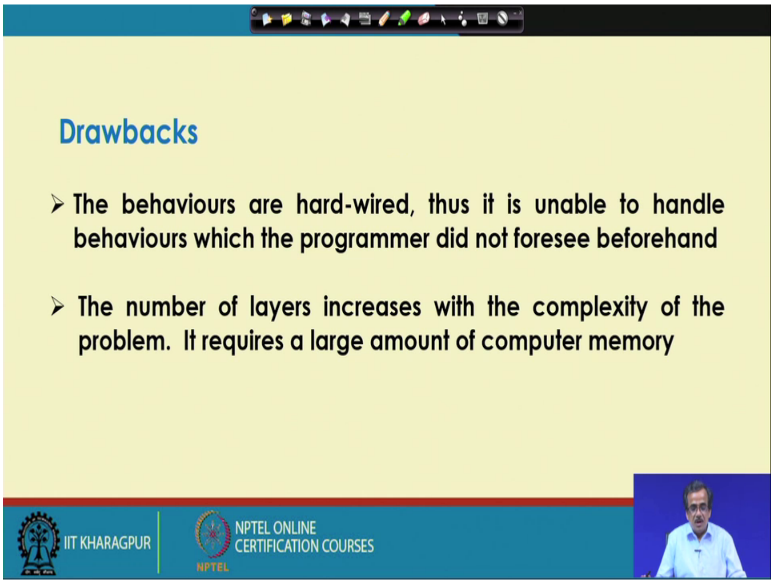
Now to score goal; so this particular robotic action is divided into a large number of the basic behaviours, and each of these basic behaviours is controlled at a particular layer of the control architecture now if I just consider a complicated robotic action; now this complicated robotic action if I want to control. So, I will have to take the help of a large number of layers in the controlled architecture. And here there must be one supervisory the control; that means, there must be one the main computer which is going to control the control activity of the different layers.

And therefore, to handle a complicated task; so, we will have to take the help of a large number of layers and the computation of complexity is going to increase and moreover it requires a large amount of computer memory.

Now, this particular control strategy now it becomes very famous and a particular scheme in robotics was proposed that is knows as the Behaviour-Based Robotics. Now this behaviour based robotics could reach the popularity and many people used in different forms, they modified also and they could solve a number of problems related to the motion planning algorithm.

Now here this reactive control scheme has got a few drawbacks. For example, say if I consider the a complicated robotic task as I told that there could be a large number of layers and consequently.

(Refer Slide Time: 02:36)

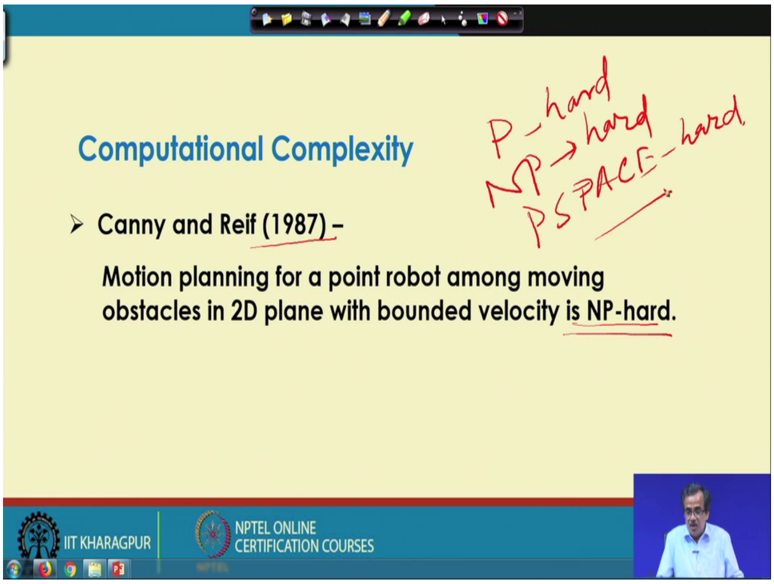


So, this particular control architecture will become complicated it requires a large amount of computer memory and this particular process will become slow. There is another drawback now supposing that the designer say could not for see a few behaviours while designing that particular the controller.

Now if this particular robot is going to face; so, that type of complicated scenarios which was not consider during the design of that particular controller, the robot will not be able to handle; so, that particular complicated situation very efficiently. And there is a possibility that the robot is going to fell to tackle that type of scenario.

Now these are all drawbacks of these particular your the reactive strategy.

(Refer Slide Time: 03:38)



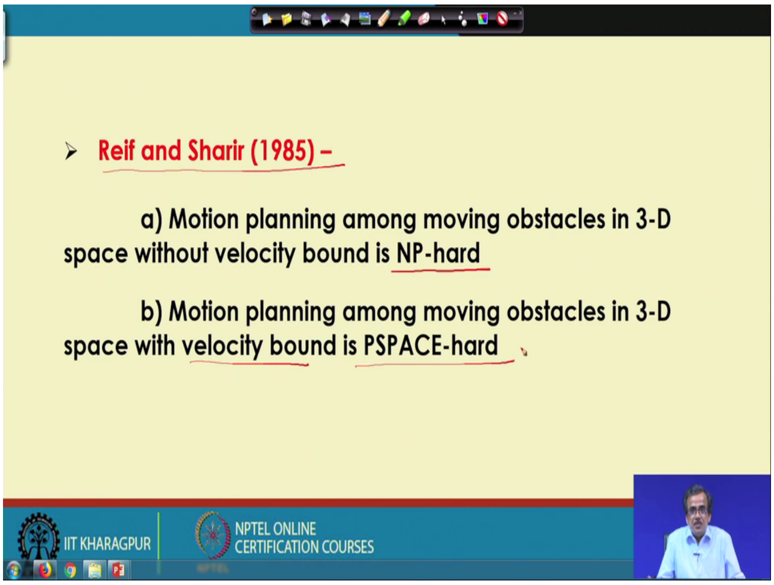
Now if I see the computational complexity of the various that the motion planning algorithms used in robotics. So, this particular the motion planning algorithms are computationally very expensive and thus these algorithms could not be actually implemented online.

For example, say the computational complexity of this motion planning algorithm was studies in details by Canny and Reif. Now, in the year 1987 they studied that computational complexity of the motion planning algorithm and according to them the motion planning for a point robot; moving among moving obstacles in 2 D plane with bounded velocity is found to be NP-hard; so, this is computationally very expansive.

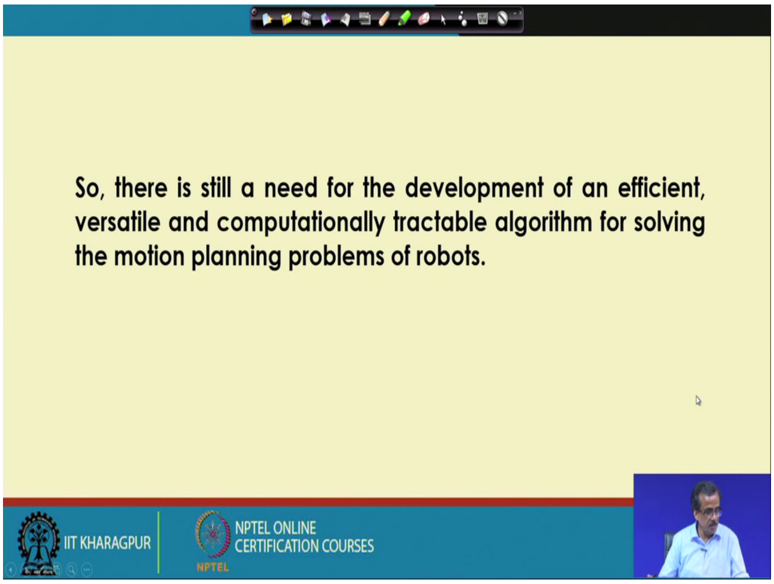
Now the computational complexity of this particular algorithm is expressed in terms of like the hardness values like NP-hard, P-hard then P space hard and all such things. For example, say for example, this hardness is represent as P-hard then comes your NP-hard that is not polynomial hardness. And then comes your P Space Hardness; P space hardness and that is very hard and this particular hardness is the exponential hardness.

Now, here in 2 D plane and for a very simple scenario like a point robot; moving in the presence of some moving obstacle with some bounded velocity is found to be NP-hard.

(Refer Slide Time: 05:25)



(Refer Slide Time: 05:27)

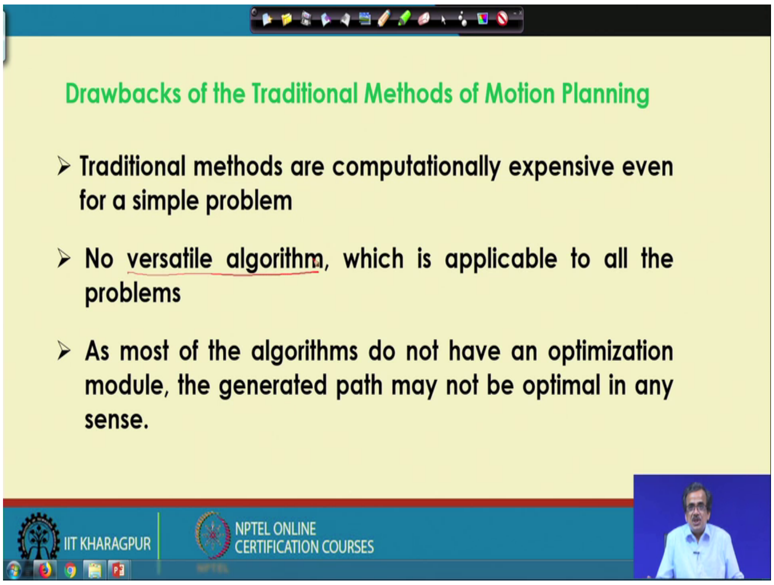


Now, similarly the similar study was carried out by Reif and Sharir in the 1985. And according to them; so they studied this computational complexity in 3-D space for a point robot in 3-D space.

Now this particular motion planning algorithm with the velocity bound is found to be NP-hard. On the other hand, the motion planning problem of a point robot in 3-D space with the velocity bound is found to be the PSPACE-hard. And that is why; so, this type of traditional methods for robot motion planning like could not be implemented online because these are all computationally very expensive.

Now if you see this particular the drawbacks of this particular traditional tools for the motion planning.

(Refer Slide Time: 06:21)



Now we have seen that these particular traditional tools for robot motion planning are computationally very expansive. Thus we could not implement online to solve this dynamic motion planning problem. And moreover now each of these particular traditional tools for motion planning is suitable to solve a particular the problem and that is why these algorithms are not versatile.

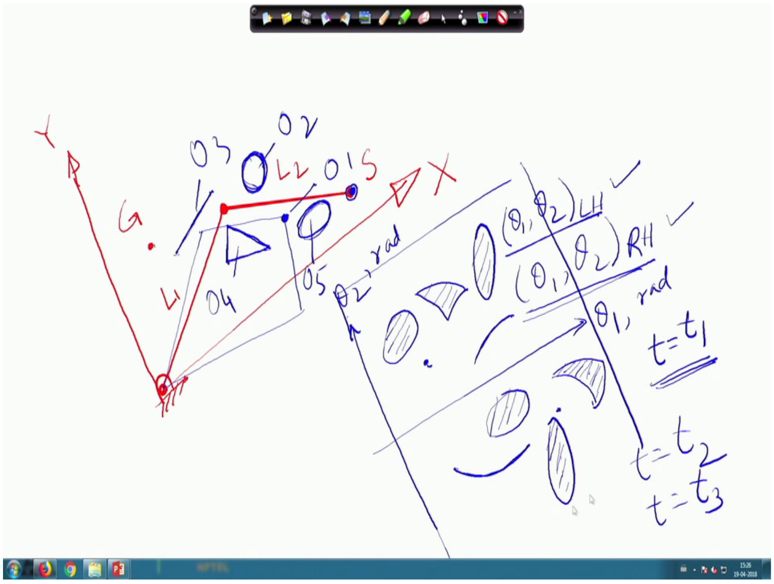
So, for different problems we will have to use the different algorithms and that is why in fact, so we will have to find out some sort of versatile algorithm or the robust algorithm which can be implemented to solve a verity of problems.

Now then comes your; so this particular traditional tools are not equipped with any such optimization module. And that is why the planned path or the generated path may not be optimal in any sense. Now these are all actually the drawbacks of the traditional methods of your the robot motion planning.

Now here if you see so, this particular the drawbacks due to these drawbacks; so, we could not implement all such traditional tools for robot motion planning in a very efficient way. And that is why there is a need for the development of an efficient the robust and computationally tractable algorithm which can be implemented online to solve these particular the robot motion planning in a very efficient way.

Now, to understand the computational complexity of this particular robot motion planning; let me let me try to take one very simple example. Now if I take this particular example we will understand that this particular the robot motion planning problem is very complicated.

(Refer Slide Time: 08:32)



Now for simplicity let me just try to consider one path planning problem for a very simple manipulator. Now supposing the; this is the Cartesian coordinate system like X and Y. And I have got a serial manipulator having say 2 degrees of freedom; say this is the serial manipulator having 2 degrees of freedom.

So, this is the length of the first link and this is the length of the second link say. So, this is the length L 1 and the length L 2 and supposing that the starting point for this particular manipulator is nothing, but S and the goal point of this manipulator supposing that is denoted by G. Now starting from the point S; it will have to reach the goal and while moving so this particular tip of the manipulator should not collide with some static obstacle.

For example, this is a point obstacle; now, similarly I can also consider; so this is the point obstacle. I can consider there could be a line obstacle here there could be some sort of triangular obstacle, there could be some sort of circular obstacle here or say elliptical obstacle here. Now this particular tip of the manipulator it will start from the point S at it will reach the point G; that is the goal.

Now while moving the tip of the manipulator should not collide with all such static obstacle. Now how can we ensure; so this type the movement; the collision free movement? Now, to solve this actually we can do it analytically in a very easy way for example, say the moment this particular the tip of the manipulator is going to touch the point obstacle.

So, this could be actually your the configuration this is one configuration. Similarly, there could be another configuration with the help of which the tip of the manipulator can touch this particular; the point obstacle. And this we can solve the joint angles the moment; it is going to touch. So this particular point obstacle; so, I can find out the joint angles like your theta 1, theta 2, the left hand solution and theta 1 theta 2 and that is nothing, but the right hand solution.

So, corresponding to this particular point; so, I will have that 2 sets of values for these particular the joint angle. Now if I concentrate on this particular the line obstacle; so, these line obstacle will be nothing, but a combination of so many such points. So, we consider a large number of points lying on this particular the line obstacle.

And corresponding to each of these particular point; so, I can find out the 2 sets of theta values. So, I will be getting; so a large number of; the large sets of theta values corresponding to these particular the line obstacle. Now, if I want to ensure the collision free path for this particular tip, for this particular triangular obstacle.

So, this tip is going to trace the boundary of the triangle and I can also find out the sets of theta values or the joint angle values. Similarly the moment is going to trace, the boundary of this particular circular obstacle. So, I can also find out what should be the theta 1, theta 2; the sets of theta 1 theta 2 values. And following the same principle, the moment this particular tip of the manipulator is going to trace the boundary of this particular the elliptical obstacle. So, I can also find out the combination of this particular the theta values.

Now, if I want to ensure the collision free movement of this particular tip. So, what I will have to do is; so, I will have to plot, so this particular theta 1 and theta 2 like the joint angles. And these are in say radian theta 1 and theta 2 are in radian. And there is a possibility for one point I will be getting 2 points here.

And for this particular the straight line the obstacle; so, I will be getting one curve line here and there is a possibility I will be getting another curve line here. Then corresponding to this particular triangular obstacle so, there is a possibility I will be getting one curve line, another curve line, another curve line.

Similarly here I will be getting one curve line, another curve line, another curve line. Then corresponding to these particular your circular obstacle; so, there is a possibility I will be getting one elliptical and here I will be getting another elliptical step. And corresponding to this particular elliptical so, I will be getting one distorted ellip sort of thing; so, might be another distorted ellip sort of thing.

Now on this particular theta space; so, these are nothing, but the forbidden zone; that means, if I want to ensure the collision free path for the tip of this particular manipulator. So, what I will have to do is; I will have to select theta 1 and theta 2 in such a way that your theta 1 theta 2 should not lie on these particular the forbidden zone.

So, these are all forbidden zone this is nothing, but a forbidden point another forbidden point this is nothing, but a forbidden curve another forbidden curve ok. So, this is the way actually we can ensure the collision free movement of this particular the tip of the manipulator.

Now this is a very simple the motion planning problem or the path planning problem. Because here we have considered that these particular obstacle that is your obstacle 1, then comes obstacle 2, obstacle 3, then comes obstacle 4 and obstacle 5; these are all stationary obstacle and this is in 2 D.

Now if I consider that these particular obstacles are moving; then how to ensure the collision free path for this particular the tip of the the manipulator? Now, this problem will become more complicated; now this will become more complicated because the positions of these particular obstacles are going to change with time.

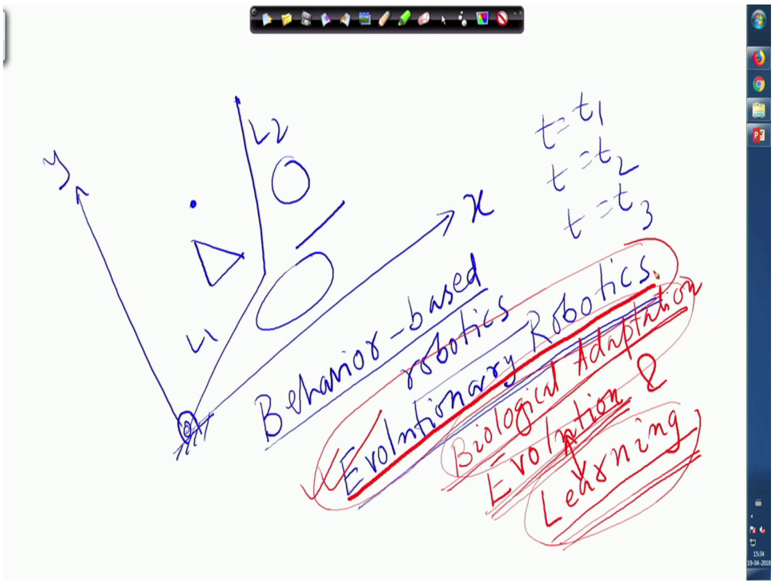
So, might be at time t equals to t 1; so, I will be getting. So, this is the scenario this is actually your the infeasible the region at times a t equals to t 1. Now if it is a problem of dynamic motion planning; so, there is a possibility.

So, at time t equals to t 2; so, this particular feasible and infeasible the zones that is going to vary. And similarly at time t equals to t 3; so, I will be getting another combination of this particular the feasible and infeasible zone. And through this particular feasible zone supposing that; so, this is nothing, but the total area.

Now this white portion is nothing, but the white portion is nothing, but your the feasible zone. And through this particular feasible zone actually what I will have to do is; so, I will have go for I will have to go for. So, this type of I will have to find out your the feasible and the infeasible zone.

Now, once gain let me consider the same example here.

(Refer Slide Time: 16:42)



So so if I consider so, x and y and so this is nothing, but the robot this is L 1 and L 2 and we consider some static obstacle like point obstacle, line obstacle, triangular obstacle, circular obstacle, elliptical obstacle something like this and these obstacles are all stationary.

Now, as I consider in the next time the these obstacles are moving in 2 D; the obstacles are moving. So, as I mentioned at time t equals to t 1, at time t equals to t 2, at time t equals to t 3; the scenario that is the feasible and the infeasible scenario; so, that is going to vary.

The problems becomes much more difficult; now if I just make it more complex. Now supposing that so, I am just going to add another dimension to these particular the obstacle that means, your obstacles will become 3 D and if I consider that these particular obstacles are moving.

And if I consider a manipulator having say 6 degrees of freedom. So, if I consider that my hand is a manipulator; the serial manipulator and staring from; so, a particular point. So, tip of this particular finger; so, I just wanted to move this particular the goal point and while moving from here to this particular point; so, supposing that the robot is started moving.

And while moving now there are some moving obstacles which are going to come in between. And this particular the tip of the manipulator we will have to find out the collision free path online. And this particular problem becomes very difficult to handle because here the motion planning algorithm will have to take the decision within a fraction of second.

And to solve this types of problem actually the traditional motion planning algorithm is going to face a lot of problem. Because these algorithms are computationally expansive so, we cannot take the decision online within a fraction of second and there is a chance that it is going to collide that the tip of the manipulator is going to collide with the moving obstacle.

Particularly whenever it is working in the 3 D space and whenever we are working with the robot with 6 degrees of freedom and if it is working in the 3 D space with the moving obstacle; it becomes difficult to find out the collision free movement for the tip of this particular the manipulator.

So, this particular motion planning algorithms are very complex and finding the online solution is very difficult. And that is why nowadays there is a trend to replace this particular the behaviour based robotics, the principle of behaviour based robotics which I have already discussed using the reactive control scheme.

Now this particular behaviour based robotics are actually will not be able to solve; so, this type of difficult situation in a very efficient way. And that is why nowadays there is a trend that in place of these particular behaviour based robotics; we go to another, the motion planning algorithm and that is called actually your evolutionary robotics.

Now this particular evolutionary robotics, we take the principle of evolutionary principle. So, I am just going to tell you in short the principle of this particular evolutionary robotics, but I am not going to discuss in details of this particular; the principle of evolutionary robotics because this is not is not there within the scope of this particular the course.

But I am just going to tell you the philosophy behind this particular evolutionary robotics and this particular evolutionary robotics could be a possible solution to solve this type of very complicated problem particularly the motion planning in 3 D space for a manipulator having 6 degrees of freedom.

And if we consider the moving obstacle; so, there is a possibility that this principle of evolutionary robotics is going to help us to find out a feasible solution. Now let us see how does it work? Now here in this evolutionary robotics actually what you do is; we use the principle of the biological adaptation. So, this biological adaptation if you see; so, there are, but 2 principle of the biological adaptation.

Now, those are nothing, but the evolution, evolution and the learning. So, this evolution and learning are going to help in biological adaptation. Now if you see this particular evolutional and the leaning; so, these operators are working on 2 different time scale.

For example, say evaluation is working through a large number of iterations. On the other hand; so landing takes space in once life time and these 2 operator are going to help each other. For example, say if you see the principle of learning; now while learning we use the principle of optimization. And most of the optimization tools actually works using the principle of the evolution.

So, this particular the principle of learning or the principle of optimization works through a large number of evaluation. And if I can learn some good things throughout my life; so, I am just going to pass this particular good information to my next generation.

And there is a possibility say due to this particular; the good information there is a possibility the rate of evolution is going to increase. So, the evaluation is going to help this particular learning and learning is going to help this particular the evaluation.

So, they are going to help each other and there is a possibility; it is going to increase the rate of this biological adaptation. Now this particular principle of biological adaptation has been copied in evolutionary robotics. Now here in evolutionary robotics actually what we do is; we try to design and develop some motion planning algorithm using the principle of the evolution and that of learning.

So, what you do is we try to use some evolutionary tool like some sort of biologically inspired optimization tool. For example, it could be genetic algorithm particle some optimization and so on. And we use some learning tool like some sort of neural networks, some sort of fuzzy reasoning tool and we try to evolve the more efficient; the motion planning algorithm for this particular the robots.

And that is actually the principle of; so these particular the evolutionary robotics. So, evolutionary robotics is going to solve the, this particular motion planning, the problem in a very efficient way. And nowadays actually there are many such applications to solve the motion planning problem; using the principle of your, this evolutionary robotics.

Now as I told that this particular principle of evolutionary robotics is beyond this scope of this particular course. So, I am not going to discuss in much more details the principle of this particular the evolutionary robotics but as I told that this is one of the possible way to solve this particular the motion planning, the problem of the robots; particularly for the mobile robots in a very efficient way.

Thank you.