Robot Motion Planning
Prof. Ashish Dutta
Department of Mechanical Engineering
Indian Institute of Technology – Kanpur

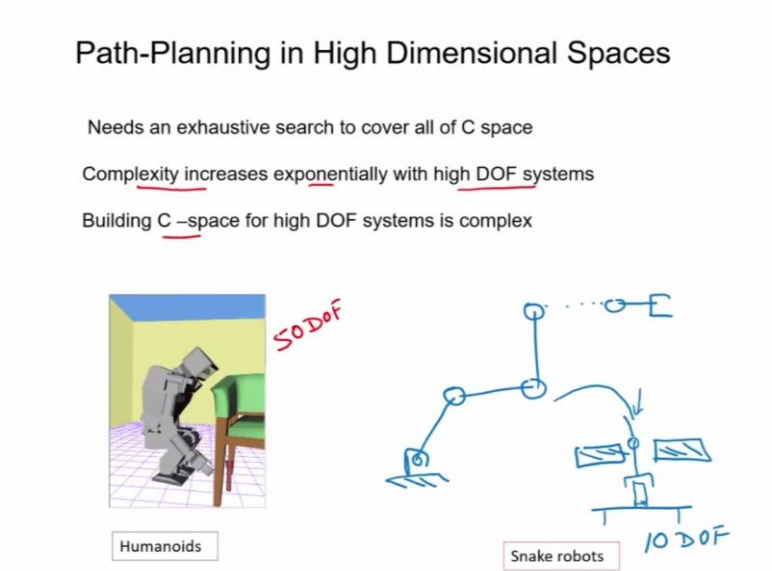
Lecture – 12
Sampling Based Planning

Hello and welcome to lecture number 12 in the course Robot Motion Planning. In the last class we looked at cell decomposition methods where you take the whole work space then you divided into small, small cells, find the connectivity between the cells and then try to find the path from the initial point to the goal point. Now, if the work space is very large of the complex obstacles.

Then what would happen in the cell decomposition method may take a lot of time because you have to explore the full area the full work space. So, in such cases what we do is we use the next method which we will be looking at today that is called sampling based methods that means we do not need to search the whole work space, but we just need to search in areas which is of interest to us or search in such a way that you can go from initial point to the goal point without searching the whole work space.

So, today we will look at sampling based methods. Today we start off with sampling based methods and as I was mentioning that in such methods we do not need to explore the whole work space because it takes a lot of time.

(Refer Slide Time: 01:15)



Now also in case of systems having higher degrees of freedom the very large degrees of freedom. For example look at this humanoid robot on the left hand side. This robot is picking up an object from under the table. Now how many degrees of freedom does it have? It can have very large degrees of freedom somewhere around so maybe 50 degrees of freedom. Now if you have 50 degree of freedom configuration space obviously you understand how complex that is going to be.

You may not be able to even draw it in three dimensional space and then to search for a path to take the hand and catch that object and take it up. There would be infinite solutions again. So, if you need to do that this can actually become a problem which you cannot solve by other methods by drawing configuration space and things like that. The complexity increases exponentially with high degrees of freedom system that we have seen.

So, we started off this course by talking about very simple mobile robot, circular mobile robot or rather point robot and the point was moving around in space and avoiding obstacles, but as the degree of freedom starts going up the complexity starts increasing and it can increase exponentially. Now building C space for high degree of freedom system is very complex.

And for example in the humanoid robot case here it would be virtually impossible to draw this and then exactly figure out to find the path pick up that object and take it up. So, this is one example. There can be other examples let us have a look at another example which is what we call a snake robot or hyper-redundant manipulator having very large degrees of freedom so maybe 10 degrees of freedom.

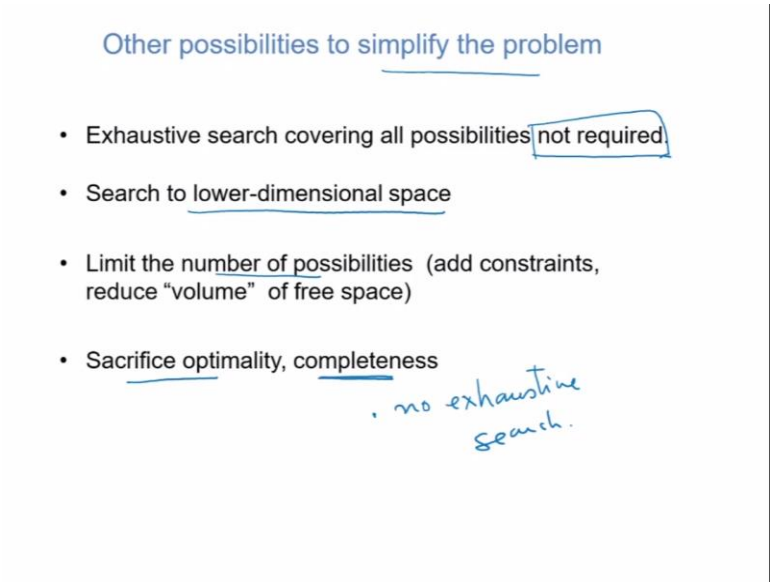
So, you can imagine that we have robot here very often called as snake robot and the snake robot is having 10 degrees of freedom so here there is one more degree of freedom and it has to go inside some position may be there is a hole here and there is an object which is kept here on the table. So, let us say this is an object which is kept on the table. Now, it has to go inside this hole from here.

And then go and catch that object and take it out then you have 10 degrees of freedom. Now, again this is the case where you have such high degrees of freedom that it will be very difficult to draw the configuration space and actually try to figure out what is the best way to

go. For example it can go in from here like this catch this object. Position itself accurately and then come out, but that will be a complex task if you are thinking in terms of C space.

So, this is another example which would be very, very difficult in case of C space based planning. One way to go ahead for such problems is to simplify the problem. Now exhaustive search covering all possibilities is not required. For example in the previous case we looked at the humanoid robot which has 50 degrees of freedom. Now there will be infinite solutions.

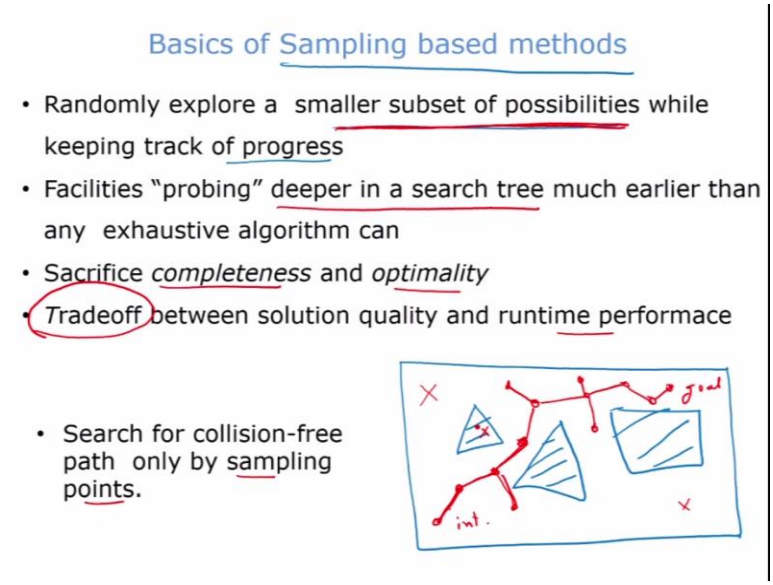
(Refer Slide Time: 04:07)



Now if you want to have an exhaustive search which we will look at all the possibilities and then try to find which is the best possibility that would be time consuming and may not be even possible infinite time. So, exhaustive search is not required so that is one. So, we can search in a low dimensional space and we can search in a low dimensional space and limit the number of possibilities either by adding constraints or by reducing the volume in the free space.

Now, if you do that what will happen is you are going to sacrifice what is completeness in terms of not covering not having exhaustive search, so no exhaustive search. So, this is something that will happen the second is you may sacrifice optimality that your solution may not be optimal in the global sense. It could be locally optimal, but need not be globally optimal because we are only looking at finite set of solutions, but by doing this we can simplify the problem.

(Refer Slide Time: 05:06)



Now what is the basic way by which we perform this problem is that we use the sampling based method and as the name indicates we randomly explore a small subset of possibilities while keeping track of progress say for example let me draw a work space here so this is my work space and these are obstacles let us say these are obstacles and you want to go from one point to another point you want to go from this point initial point to goal point.

So, one way of searching for this is instead of looking everywhere searching from the whole space I do a sampling based search say for example from the initial point I go to the next point which is here then next point here next point here. So, I sample these points possible point say for example a point is not possible inside the obstacle so there is no point there. So, I sample a few points and then I make these network of paths again.

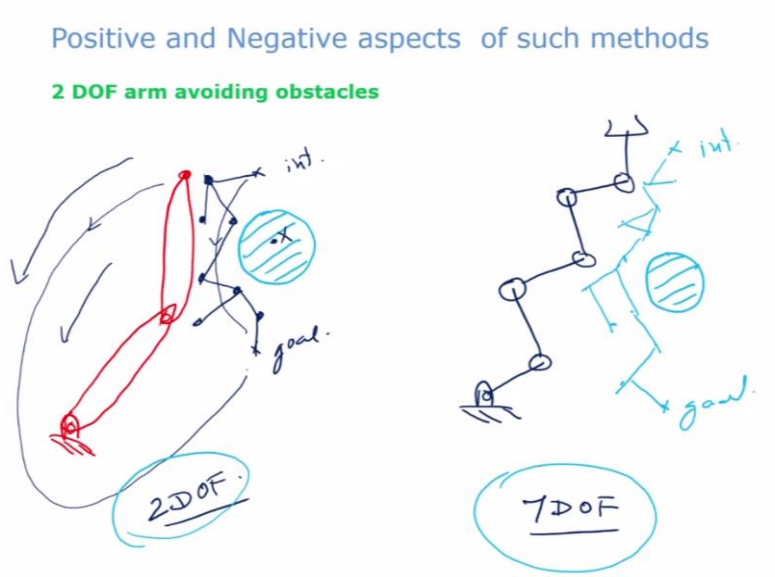
So, these are my sample points and I can see that I can go from one point to another point by sampling in the space. So, what has happened is I have not gone here, I have not gone there. So, I have randomly explore a smaller subset of possibilities. Now obviously it has to be directed if it is completely random it can go in any direction. So, if you know the goal is in this direction or if you know the obstacle is this side then we can sample accordingly.

We will see this as we go along. So, first of all we can randomly explore a smaller subset of possibilities which keeping track of progress and you can probe deeper into the search tree much earlier than any exhaustive search can. Say for example this is my search tree so I started from here I went there, I went there, I went there, I went there, I went here. So, this is

my search tree and I can probe deeper in the search tree much earlier than any other exhaustive algorithm. Now we have sacrificing completeness and optimality here.

So, there is basically a tradeoff between solution quality and run time performance. So, this is basically how we do our sampling based methods by searching for collision free paths only by sampling at once. So, as I was explaining first you sample points possible points that means they are now inside in obstacles then we connect these points to a possible paths and make these network and then try and go from the initial point to the final point.

(Refer Slide Time: 07:30)



Now, for example, let us take the case of this two degree of freedom arm and we have two degree of freedom arm which is going to avoid obstacles. Now this is a two degree of freedom arm which is going to avoid obstacles let us say there is an end effector there then it becomes three degree so let us give a two degree only. So, this is my point and there is an obstacle which is there.

So, this is my obstacle now I want to go from one point to another point without hitting this obstacles I have to go from this point to this point. So, this is my initial point and this is my goal point. How many solutions are there? There can be many solutions we can go like this, we can go from this side. So, these are all possible solutions, but instead of trying to search for solutions everywhere that means we have been trying to look at these solutions which I do not need.

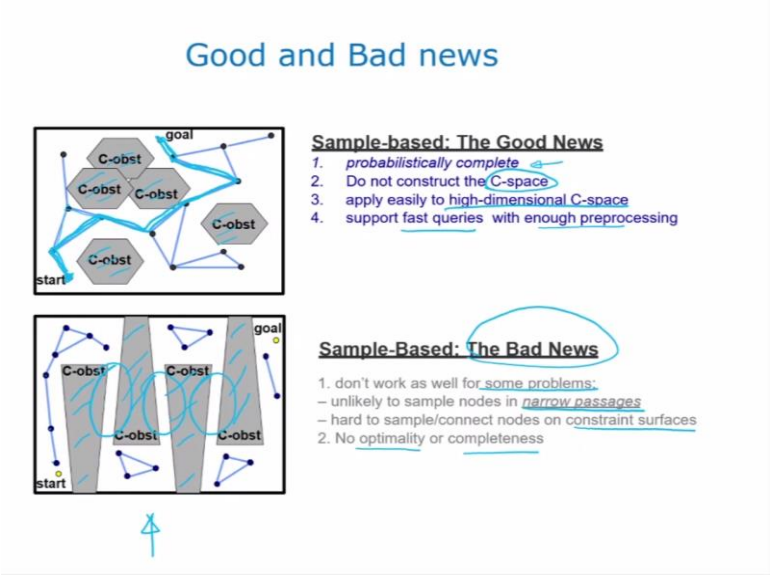
I can simply look at solutions near the obstacles which will take me from the initial point to the goal point. So, basically I can do sampling at these points or maybe these point, this point, this point, this point these are points some sampling which are possible. There is no possibility of point for inside the obstacles so I don't sample there and then what I do is I connect it with paths.

So, these are my paths which are connecting and can see whether I can go from one point to another point from the initial point to the goal point by connecting to these points. Now obviously we are not looking at the complete search space this may not be an optimal solution but is a solution and there might be no solution also. So, as you can imagine very easily here that if your sampling is not dense enough you may miss the initial point and the goal point itself or if a path exist you may not be able to get the path.

Similarly, this is two degree of freedom which is much simpler. Now if you have a 7 degree of freedom system it is a very large 7 degree of system. So, we have 7 degrees of freedom and we want to avoid an obstacle again similar case where there is an obstacle here and you want to go from one point to another point I want to go from this point initial point to goal point. So, what I do is I sample at these points which are possible near the obstacle.

And I connect them by paths making a network of paths and then what we do is we try and see if I can connect the initial point to the goal point to this set of path and if a path exists then it can take you from the initial point to the goal point. Now in this case there was infinite solutions in this case also there is a very large number of solutions. So, we are sampling such that we will get only a few solutions not all the solutions.

(Refer Slide Time: 10:08)



So, there is some good news and bad news as shown in the example here. So, the C obstacles are the obstacles. So we have sampled at these points and these are the sample points if you can call nodes and we can and we connect them by edges or paths and then I connect the start point to this the nearest node and I connect the goal point to this lowest node and then I see if I can find the path like this, like this, like this, like this.

Now the good news is that it is probabilistically complete we are not constructing the C space. So, for higher dimensional space like redundant manipulator or humanoid robot this is fine because we are not constructing the C space. This can be very easily applied to high dimensional C spaces because we do not have to search the whole space we can tune ourselves such that we cover a very small part of the space.

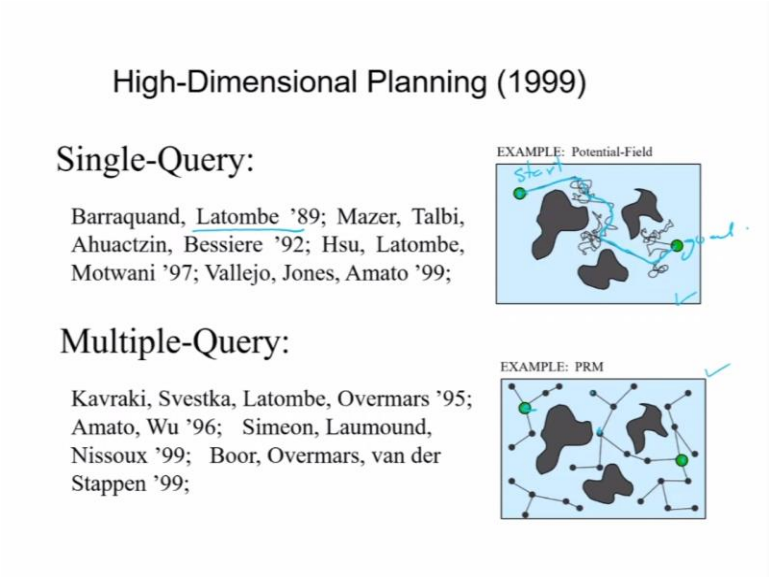
Now it supports fast queries and with enough pre-processing. Now what is the meaning of fast queries, for example, if I have a start point here and the goal point there and I want the query that the way we can reach the goal from the start point is very easy you just search in this way this is a query and then you can go to the goal point. Now what is the bad news look at this figure here. Now these are obstacles C obstacles.

Now I am samples and after getting this samples I am getting these network of paths which are not connected why because there are this narrow passages there. As you can imagine I did not sample there so there is no connection of path there. Similarly there is narrow passage here and narrow passage here. Now, if I want to go from the start point to goal point I will not be able to go there is no connected path.

So, as you can understand it depends on how you have sampled whether you are sampling is dense enough or not whether you have covered the whole space. So, the bad news here is that the good news is that it is faster it is you do not need to construct the C space, but the bad news here is that it does not work well for some problems especially these kind of problems where there are narrow passages and you have no sample there.

Now it is also hard to sample connect nodes on constraints surfaces and no optimality or completeness. So, this is the plus point and the minus point of this method that we are using.

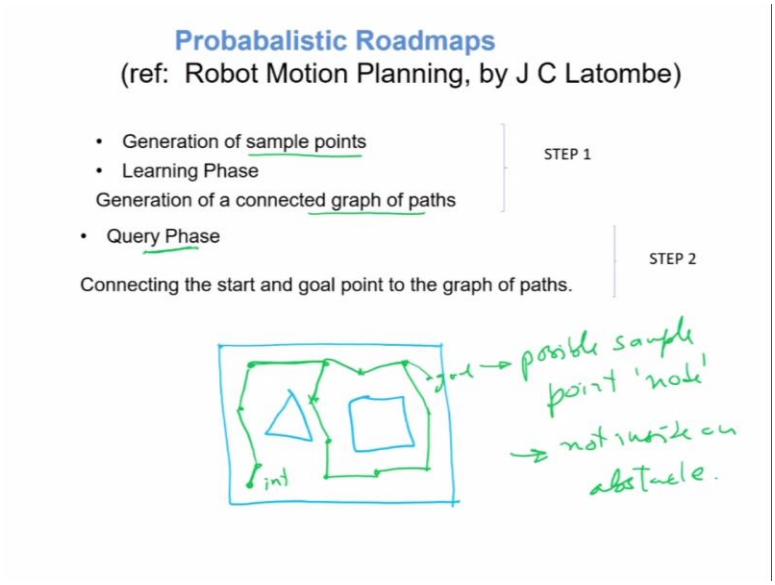
(Refer Slide Time: 12:28)



Now this is basically where the study started off from high dimensional planning 1999 single query and multi query. So, there are these famous papers Latombe in others. So, single query for example you are at the start point here and you want to go to the goal point. So, in the potential field method basically which will be coming to little bit later is that you start going towards the goal and it was an obstacle you get pushed away from the obstacles and you keep going towards the goal avoiding the obstacle and then finally able to go the goal.

This is a single query that means we started off with one query and went off in that direction. In case of multiple query we have start point here, goal point here. You start off with multiple points and start branching out and then see if you can connect the start point to the goal point. So, basically there are two ways single query and multi query ways of looking at this problem.

(Refer Slide Time: 13:24)

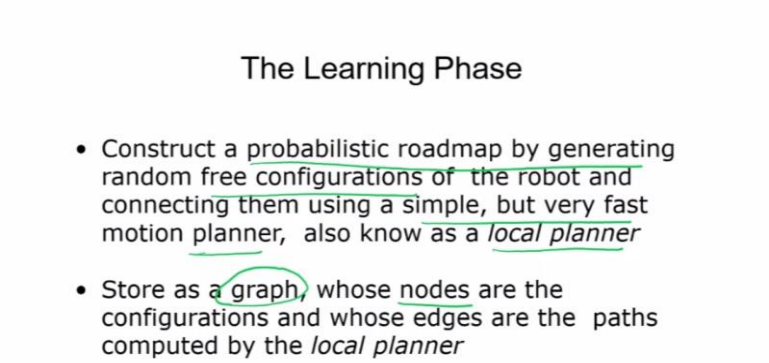


Now, let us look at few more in the exact procedure that we are going to follow the step one and the step two. So, the step one is in such cases we have to generate the first thing is we need to generate the points possible points. So, this is my work space these are triangles. So, we are going to generate the sample points so these are sample points which was generated possible sample points.

So, a possible sample point means it is not inside an obstacle. So, possible sample point now what we can call a node is the one that is not inside an obstacle and then what we do is we have a learning phase in which we connect the nodes by edges. So, we are connecting this nodes with edges like this then generation of a connected graph of paths which is a query phase we are connecting all the nodes by what we call it edges maybe I have one more, one more here, one more here like this.

Then so first sample points are generated then we are connecting them with set of paths and then there is a query phase which is going to take me from suppose this is my initial point and that is my goal point. So, I want to go from initial point whether it is possible to go from initial point to the goal point that is basically what we call the query phase. Connecting with the start and goal points to the graph of paths and whether you can go from the initial point to the goal point.

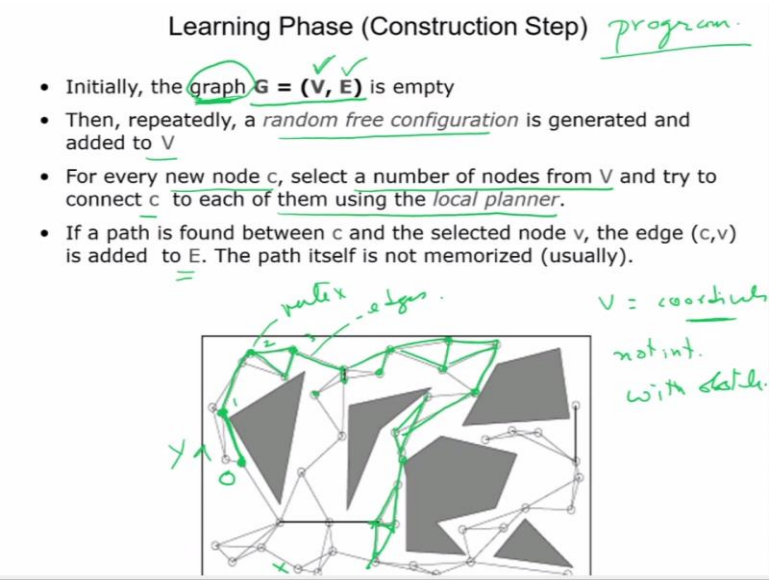
(Refer Slide Time: 15:06)



Now, so this is the first way in which it goes the second is the learning phase after generating your sample points. So, you construct a probabilistic road map by generating random free configurations of the robot and connecting them using simple, but very fast motion planner this is also called the local planner. So, you generate these points and connect them by means of a local planner.

Now store this as a graph whose nodes are the configuration so this is my nodes are the configurations and whose edges are the paths computed by the local planner. So, first we create the nodes then we store this as a graph such that the configurations where the nodes are the configurations and whose edges are the paths local planner.

(Refer Slide Time: 15:50)



Now let us look at this way here. So, initially so we have in terms of a program let me discuss this in the terms of a program now. So, at the end of the day a path planning or motion planning is executed by a program. So, this is a program and how is the program operate so it is a computer program. So, initially it starts off by this graph having a graph of certain G=(V,E) because V is the vertex and E is an edge.

So, each of the points is a vertex so this point is a vertex and this connections are edges. So, we are making this graph and to start off with graph we first make initially this is a V and E is empty because it is in free space only the obstacles are there and we have to start generating this graph then repeatedly a random free configuration is generated and added to V. So, suppose I am here just for example.

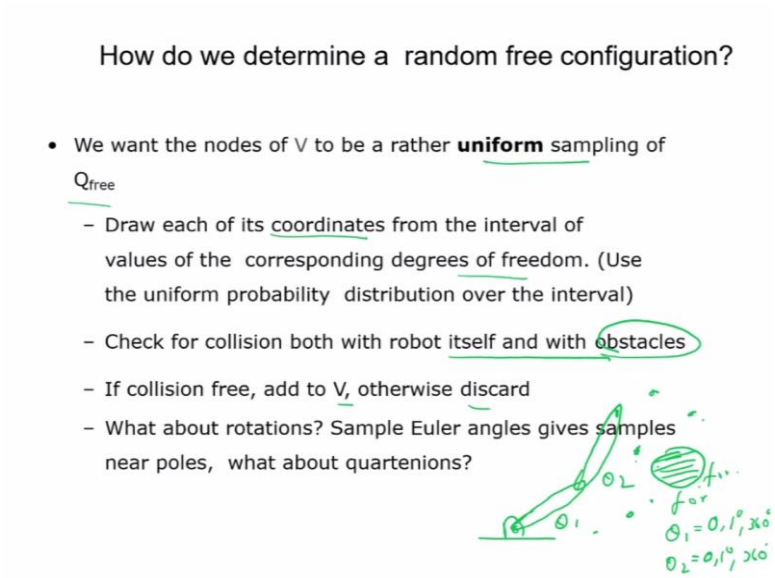
I am starting a random new configuration which is here and this is taken as node this is my node 0 this is my node 1 this is my node this is like a node, this like a node 2, 3 and this is added to the set V here. So, these are nodes. Now for every new nodes select a number of nodes from V and try to connect C to each of them using a local planner. So, I have generated my nodes I have kept this set.

Now what would be the V equal to this would be coordinates. So, this is my X axis suppose this is my X axis and that is my Y axis then these two are going to be node coordinates. Now, if a path is formed between C and the selecting node V the edge (c ,v) is added to E that means if the path exist between this point and this point then that edge is added to E here now. So, this is how we are generating a graph we start off V and E are initially empty.

Now we start off by calculating the V vertices and vertices means basically the coordinates of those points which are possible not intersecting with the obstacle. So, coordinate not intersecting with obstacle and then we connect between the points between these two points by edges and we connect this graph and then we keep adding on to the graph we start calculating the values of V and E where E are the edges is the V is the vertices.

So, now we have all these edges which are connected like this. So, we have made our set of connected graphs on network of paths.

(Refer Slide Time: 18:38)

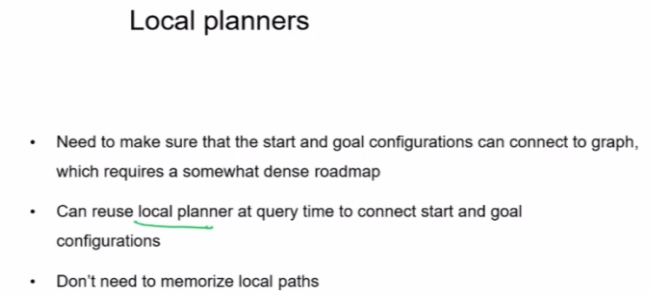


Now how to determine a random free configuration. So, we want the nodes of V to be rather uniform sampling of Q free. So, draw each of its coordinates from the intervals or values of the corresponding degrees of freedom. So, for example, let us take this example of two degree of freedom manipulator. So, I want to have some kind of uniform sampling so I take θ_1 and θ_2 here and I can sample θ_1 as from 0^0 to 1^0 to 360^0.

Similarly I sample θ_2 0^0 to 1^0 and 360^0 and then see where the end factor is so those are my possible samples. So, there is an obstacle here wherever it hits the obstacle it is not going there. So, basically if I have a for loop, two loops are there. So, one will vary θ_1 from 0^0 to 360^0 intervals of 1^0 or 2^0 see which sample points are possible and then this is how we get a uniform sampling of Q free.

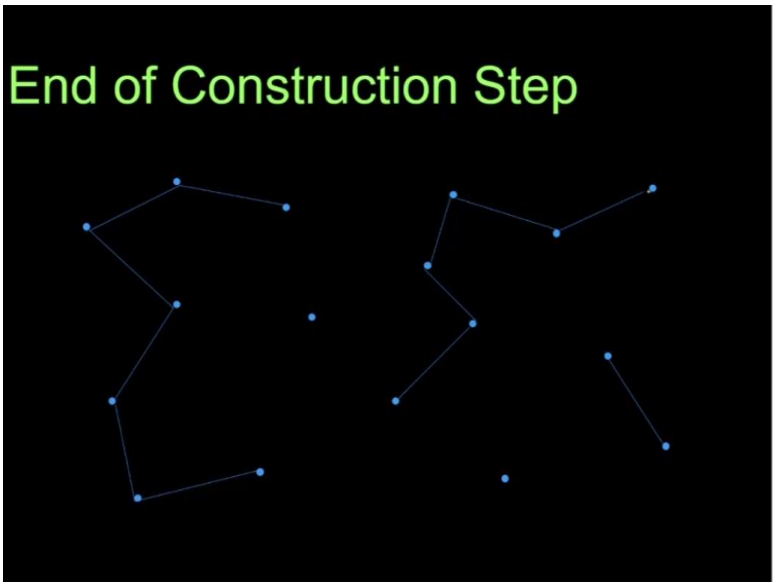
Now check for collision both with robot itself and with obstacles and not only must you check for collision with the obstacles, but even with the robot for example the link should not hit the other link so that is also an obstacle for itself. So, if collision is free add to V otherwise discard that is how we make our point that is how we sample our points. This is the way we go ahead.

(Refer Slide Time: 20:08)



Now after this we have made the points sample points we need local planner now we need to make sure that the start and goal configurations can connect to graph which requires a somewhat dense road map. We can reuse the local planner to connect the various points and generate the edges that is what the local planner does.

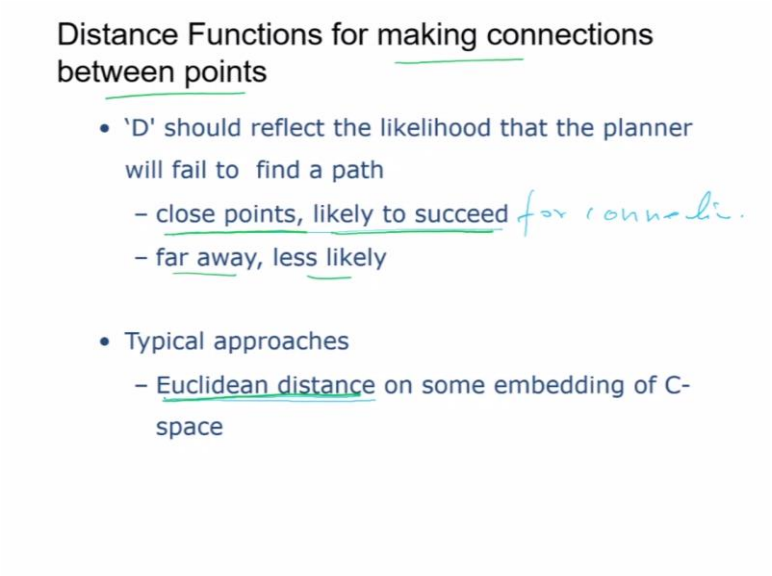
(Refer Slide Time: 20:26)



Now let us see how this is done here, so these are the points I have generated sample points are generated which are not hitting any obstacle there is an obstacle here so these are my sample points which are generated. Next is where I connect the points by means of an edge so this point is connected to this one this is connected to this one and I proceed that way. So, I choose a point connected to the nearest point so these are points which I am connecting.

By doing this what we get is a set of paths now the paths may not be connected. So, this is end of construction step. So, I have two networks, but they are not connected.

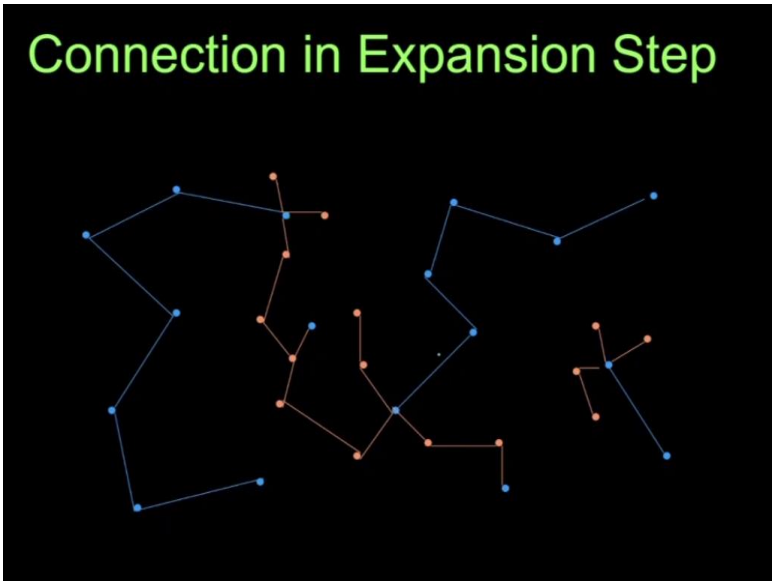
(Refer Slide Time: 21:03)



Then what we need to do is we need to make the connection between them. So, we use a distance function for making connections between points. Now D should reflect the likelihood that the planner will fail to find a path. So, choose points which are likely to succeed for connection, for example, so typically we can look at Euclidean distance on some embedding of C.

For example let us come back here let me just go there. So, what we basically do is I want to make my connections now. So, when I am making the connections from one point we should choose points which are closer to that point which is likely to succeed and not to choose points which are far away that is how I am trying to make my connections now and a typical approach could be Euclidean distance on some embedding of C.

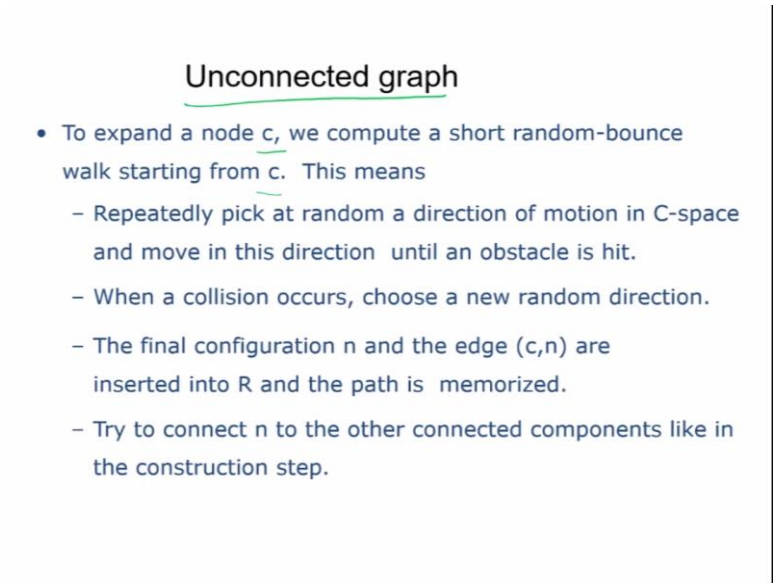
(Refer Slide Time: 22:00)



For example just look at so here I choose points now I have had in the previous case if you remember the blue lines they may not connect it I do not want to connect them. So, what I do is from here I choose points which are near to that the Euclidean distance is near to this point and not far away. Suppose I choose a point there then the possibility of connecting this to that is less and is likely to fail.

So, basically I choose points which are near here and then I try and make a connection. So, I have made my additional points here which are near to this and now I made my connection now I find that the two graphs are connected. So, let us come back here and look at this.

(Refer Slide Time: 22:37)

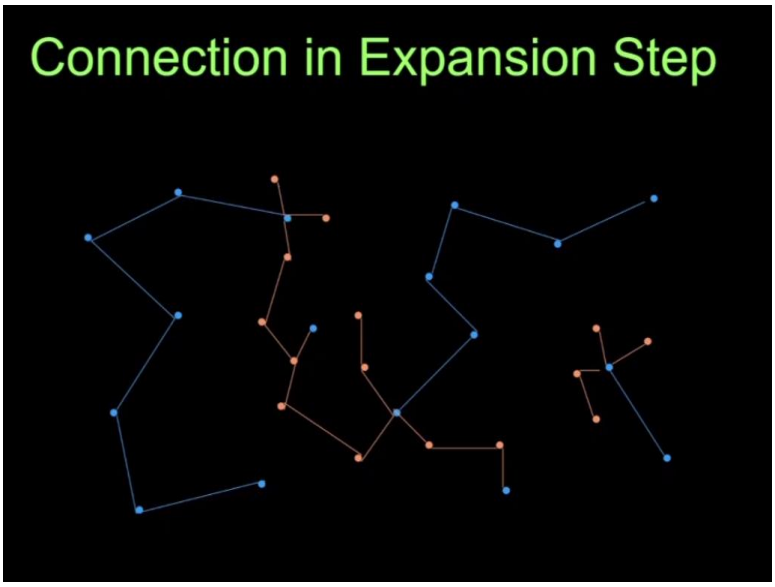


So, if you have an unconnected graph like we had to expand the node c we compute the shortest random bounce walk starting from c means repeatedly pick at random a direction of

motion in C space and move in this direction until an obstacle is hit. When a collision occurs choose a new random direction. The final configuration n with the edge c, n are inserted into R and the path is memorized.

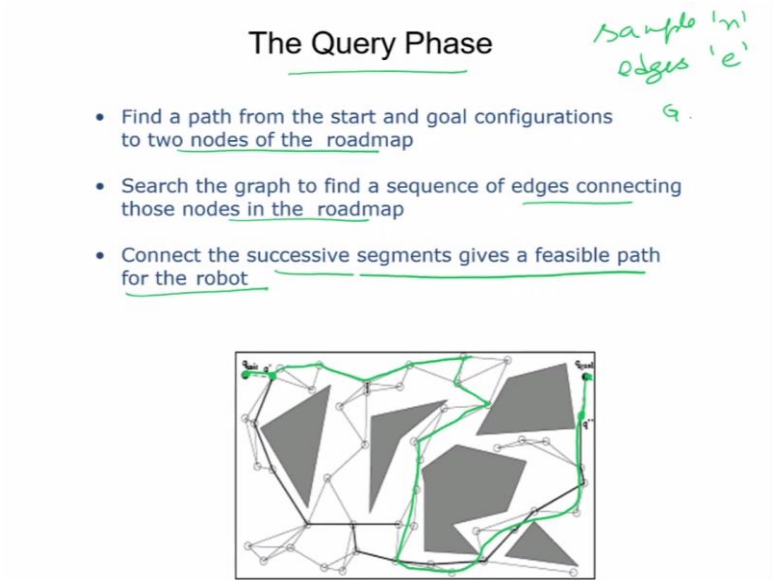
What it means is that here you start with a random direction here and c and go until you hit an obstacle. Now, if you do not hit an obstacle now that means this is a possible path so we add this to your edge in node graph again and proceed like that.

(Refer Slide Time: 23:19)



Now we will see that there can be different ways by which you can these branches can shoot of, but this is the basic idea of how make the connections or the network of paths.

(Refer Slide Time: 23:32)

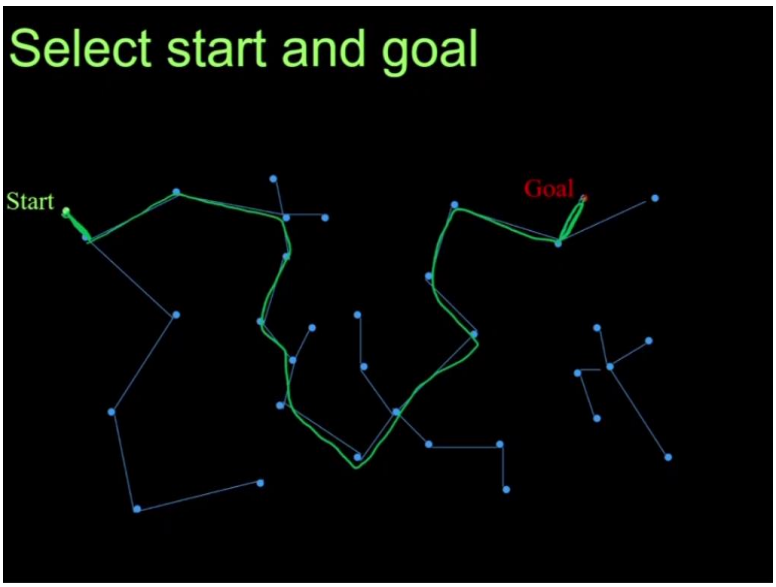


So, after you made your connections so first we have done sample that means we created a nodes n then we got edges. So, we have got edges e and we have made our graph G connection of pathways as shown here. Now next is the query phase where you have to find a path from the start and the goal configurations two nodes at the road map. Search the graph to find a sequence of edges connecting those nodes in the road map.

So, for example, I have my initial point here and my goal point here. So, what I need to do now is find a path from the start and goal configurations to two nodes on the road map. So, for example, I find to this again I can use Euclidean distance which is nearest and from here I find this one. So, now we have connected to the road map. Now connect the successive segments gives the feasible path for the robot for example like this, like this, like this, like this coming this way, this way, this way, coming all this way and then coming here.

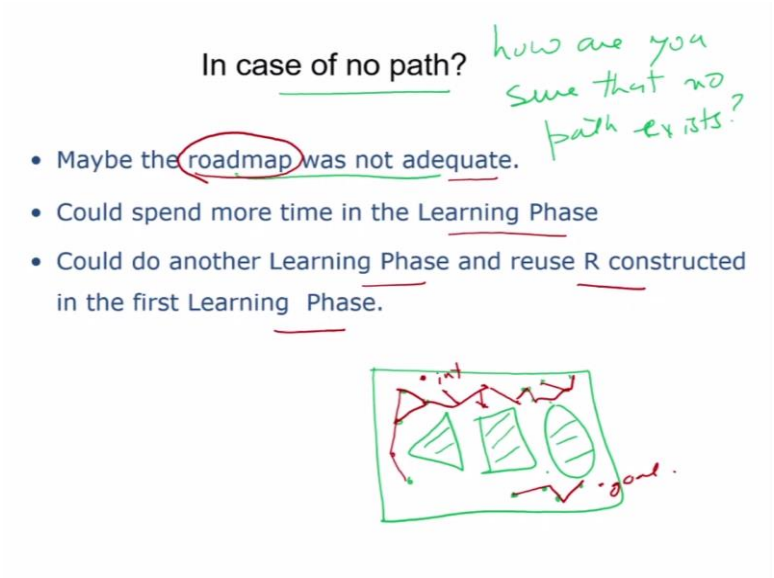
So, this is specifically the query phase where I am connecting the initial point and the goal point to the network of paths already generated and then what we are doing is we are finding the final path to take us from the initial point to the goal point.

(Refer Slide Time: 24:54)



So, as shown here we have a start point and a goal point and basically take the start connected to the nearest node take the goal connect to the nearest node and then try and find a path which will take me from there to there at connected path. Now path may exist, a path may not exist depending on how your sample and things like that. So, this is basically the idea of how the sampling based method work.

(Refer Slide Time: 25:22)



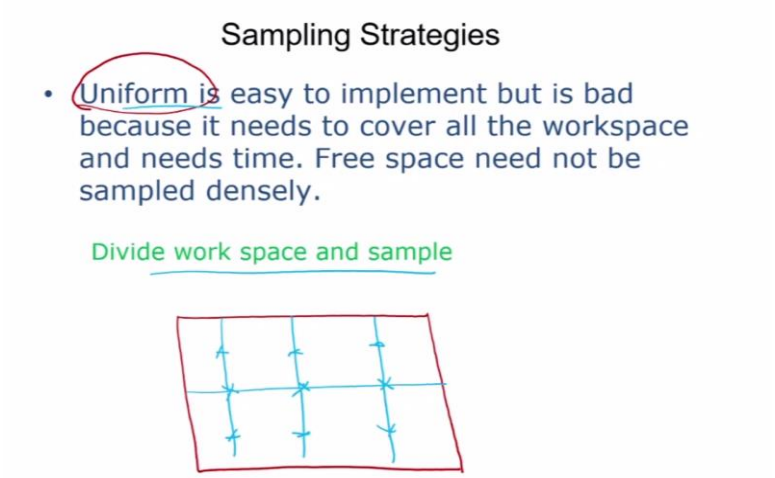
Now, obviously the question comes up that in this way in case of no path. Suppose we cannot find a path then what? Maybe the roadmap was not adequate now this is tricky because are you sure that there is no path then how are you sure that there is no path, but no path exist just because it could not find that path does not mean that there is no path. In this particular case, for example, here we have obstacles like this we have obstacles like this, obstacles like this.

And suppose we have a work space like this and we have obstacles here, here and here and I have sampled like this and I have sample this side, I have sample this side and have sample that side and what we have ended up with is set of graphs which are not connected. So, I have a set of graph like this which are not connected. Now I want to go from start point I want to go from this point my initial point to my goal point here.

Now you can see with our eyes because we have eyes that there is a path, but this roadmap is not being able to find the path only because the sampling was not good enough it is not cover everywhere so that does not mean that the path is not there. So, it basically means that may be the roadmap was not adequate, adequate means that did not cover the whole work space. So, now you could spend more time in the learning phase.

And you could do another learning phase and reuse our constructed in the first learning phase to sample more to the points. So, this is something that we could prove and then try again. So, you can see that one of the disadvantages of this method is that it can fail very simply because it depends on your sampling now.

(Refer Slide Time: 27:29)



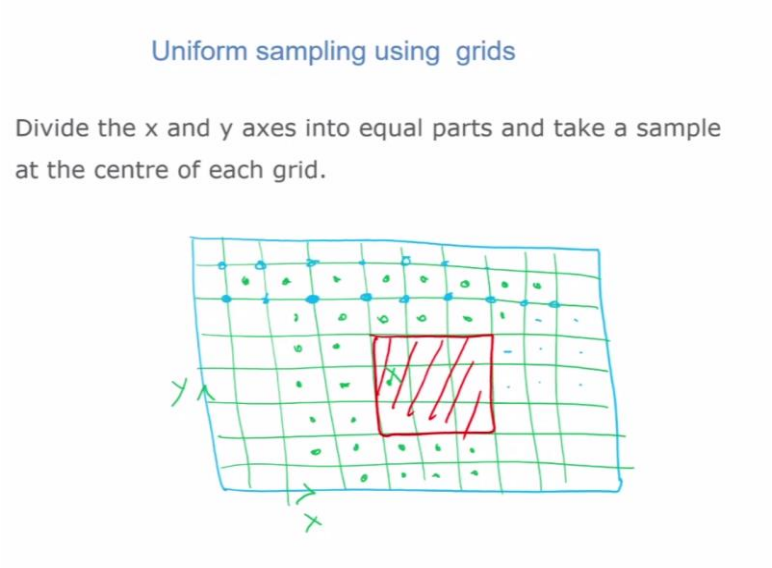
So, there are some sampling strategies that are used for so sampling strategies that are used is uniform sampling is easy to implement, but it is bad because it needs to cover all the work space and needs more time. A free space need not be sampled densely. So, divide work space and sample could be one way. So, uniform sampling means you are covering everything which may not be very good always because it is going to take more time we are covering the full work space so what was the point anyway.

And free space need not be sampled densely. So, for example, divide work space and sample uniform sampling what we can do is we have the work space like this we can divide the work space into two parts one part at this side and then I start sampling at and I divided into take a half of this one point then I have another point which is half of this is half here another half here.

Then I draw a line like this take the half of here, half of here, half of this line, half of this lines. So, basically what I am doing because I am dividing this equally and then I am sampling. So, what is happening is that I am getting a uniform kind of sampling as you can see here. So, basically I just divide the work space and sample into equal half this is something very similar to the cell decomposition methods.

But this is not a good idea because you are ultimately covering the whole work space which is actually not rquired.

(Refer Slide Time: 29:01)



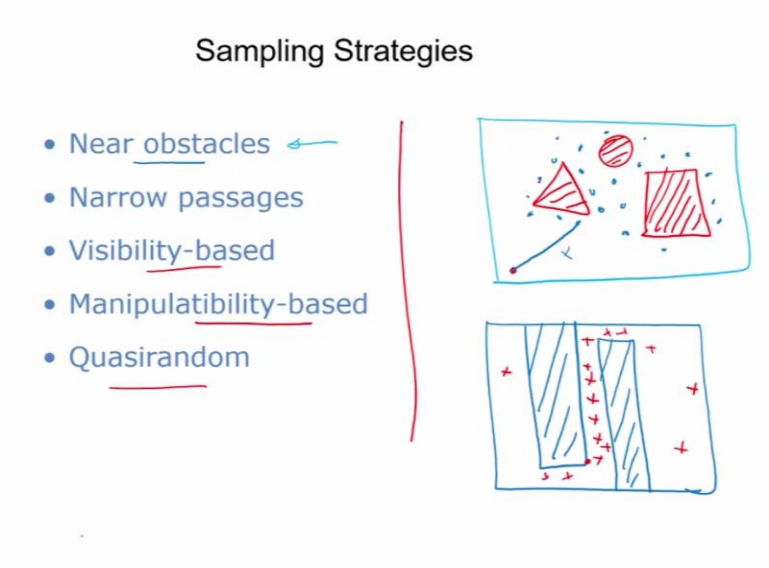
Now uniform sampling using grids so divide the x axes and the y axes into equal parts and take a sample at the centre of each grid, for example, I have the work space like this, this is my x axis and this is my y axes. So, what I can do is I can make into grids so we are familiar with grids like in the graph paper you have grids. So, basically I divide the work space into grids.

So, these are my grids and then what we can do is we can take the center of each of the grids. So, this is something like my uniform sampling each of those points suppose let us put an obstacle also becomes clearer suppose there is an obstacle here. So, these are obstacles so basically what we can see is that in the center point of these obstacles we cannot sampling points which are inside the obstacles you cannot sample that of course.

So, that is what we are seeing that now we have generated our sample of possible points using a grid now. Now the other possibility here is that you can take the intersection also I could have taken these points also then also this uniform sampling it is more or less the same. Here is the center of the grid or you take the intersection points of these straight lines then also we have covered the whole space using grids.

Now as we have seen that if there is an obstacle and we start doing uniform sampling then I have covered the whole work space and if the work space is very large then what was the point anyway because we are supposed to be doing a sampling based method not cell decomposition method.

(Refer Slide Time: 30:41)



So, there are some sampling strategies which are used which will tend to make your search faster or make the samples faster so that the whole process becomes faster. So, the first method is near obstacles no obviously that is clear. So, if you have work space like this where you have obstacles then you have obstacles like this then basically what you are doing is that is you are sampling here and this is free space.

So, there is no point in sampling very closely here because these are free space so what is the point in sampling there. So, a good idea would be to sample near the obstacles because you want to avoid the obstacles. So, I am going to sample near the obstacles such that and not in the free space so that is going to enhance my so this is near obstacles. So I am going to sample more near obstacles so that my network of paths are actually avoiding the obstacles and I do not care about the free space.

So, going from here I can go straight there I do not need to sample anywhere here. So, sampling near obstacles tends to help you in reducing the number of samples that are required and also helping you to avoid the obstacles. Narrow passages of course we have seen in the previous case that especially narrow passage is the one that causes lot of problem that if we have obstacle like this then we have obstacle like this.

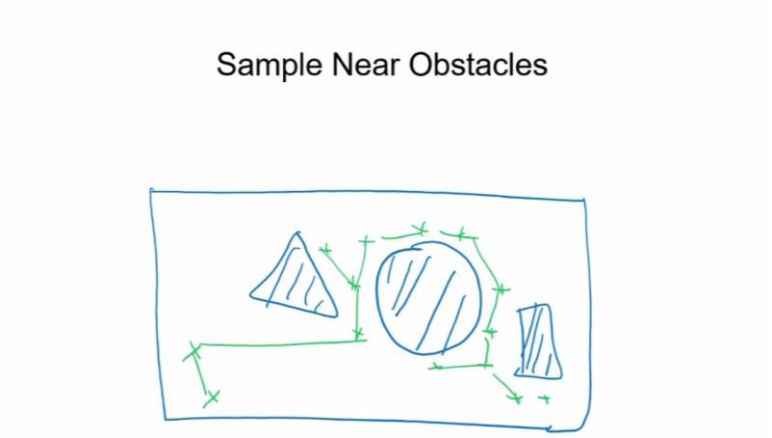
In that case what we can do is if there is a narrow passage here then let us make it narrower if there is a narrow passage here or there is a narrow passage here this is my narrow passage. So, sample more in narrow passages and less here so may be 1, 2, 3 samples here and narrow

passages the one that I need to sample more narrow passage here, narrow passage there. Visibility based if you remember the visibility graph method.

So, visibility based would be we are standing here which vertices can you see so based on that you sample, manipulation ability based we will come to that. So, manipulation ability is of motion so the direction in which you can move easily you sample such that you move more in that direction and that we will see that as we go along then quasirandom sampling instead of doing complete random sampling you can do quasirandom sampling in which it can be based in a particular direction.

For example more towards the obstacles or more towards goal. So, these are some of the sampling strategies that are used and let me now give you a few more examples as we go along.

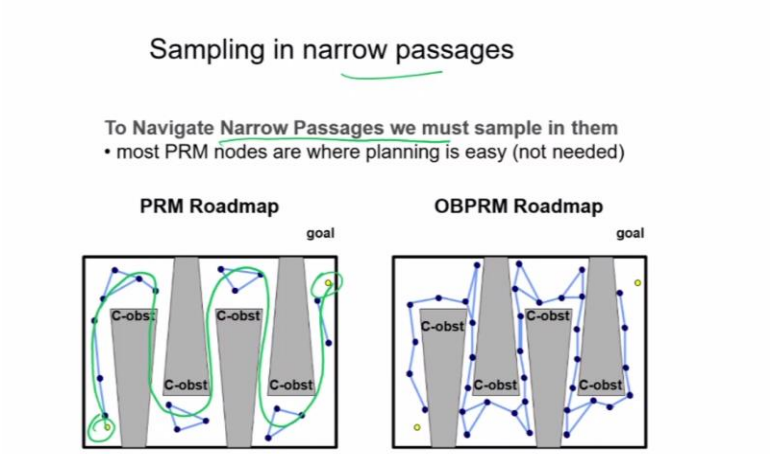
(Refer Slide Time: 33:31)



So, sampling near obstacles is very clear it says that if you have obstacles we can sample near obstacles which I just explained. So, there is an obstacle here, there is an obstacle here, there is an obstacle here. So, what we can do here is we will simply sample near obstacles and not in the free space that will give you a road map which is going to avoid the obstacles and will take you from one point to another point.

So, this is a road map which can take you from one point to another point by mainly sampling near the obstacles.

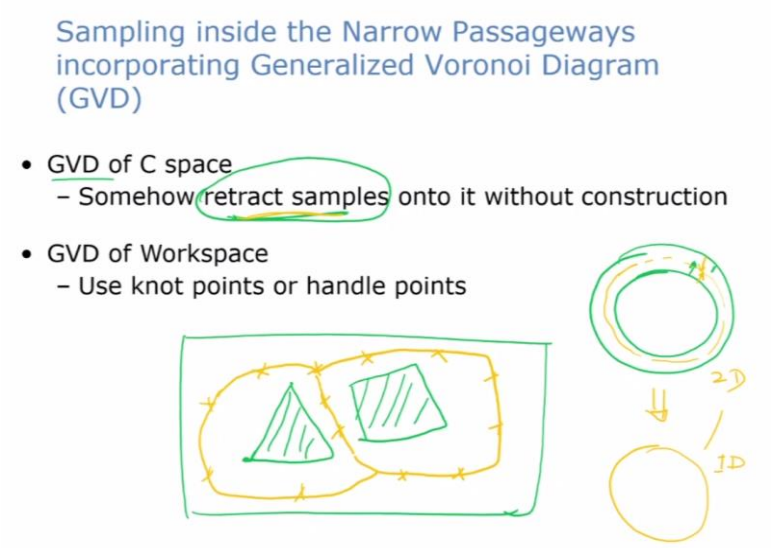
(Refer Slide Time: 34:10)



Then sampling in narrow passage is shown here. So, if you want to navigate inside narrow passages then obviously you have to sample inside them which means that in this particular case these are narrow passages. So, if do not sample in there what will happen you will not find the connected graph. So, although a path exist like in the left hand side from the start point to the end point we will not be able to find this because we have to go like this and we have not sampled inside the narrow passages.

So, sampling in narrow passages essentially to find the path that will take you from the initial point to the goal point by sampling inside narrow passages.

(Refer Slide Time: 34:49)



Now sampling inside narrow passages using generalized Voronoi diagram GVD planners, this is again by looking at let us see this is my obstacles there is one obstacle here, there is

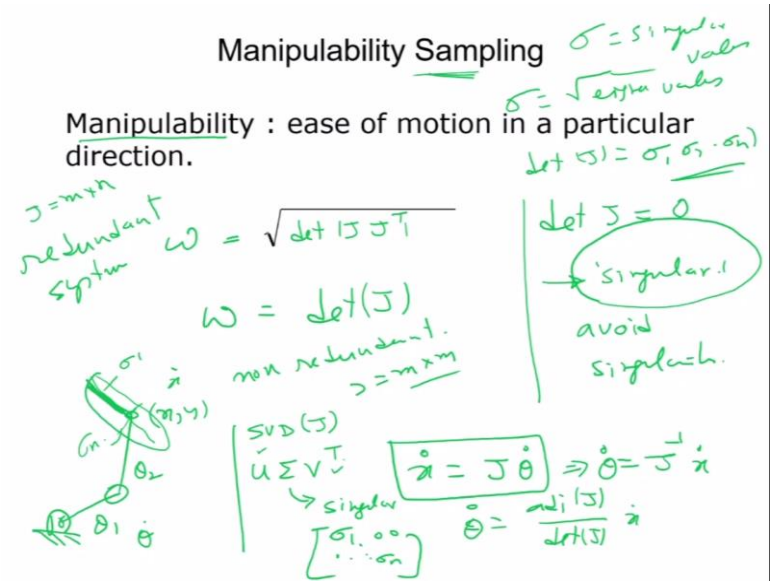
one obstacle here. So, using GVD suppose I come to this point. So, I can see this one I can see this one so I sample here now there is no vertex here so I do not sample here. So, essentially it was almost coming back to saying in that case sampling in narrow passages only by using the generalized Voronoi diagram.

The other is by using deformation retract. Now deformation retract is something which we have seen that can you find a path which is equidistance. So, the retract is what we saw that we have something like a doughnut and may shrink what will happen if this side shrinks and this side shrinks ultimately it will become a one line here with a circle. So, this will come and shrink here this will come and shrink here.

This surface will come to the circle this surface will also come to the circle. So, what we will get the deformation retract with a circle. So, it is like from 2D it has become 1D this we saw in case of one GVD planning. So, can we find a set of paths which are in between these obstacles like this we are at equidistance. There is something which we have seen at here. So, we can use a generalized Voronoi diagram this is something like the retract sample and sample in this positions.

So, now I know that there is a path from one point to another point and this is my network of paths so I did not have to sample everywhere here.

(Refer Slide Time: 36:24)



Now the ease of manipulation ability the manipulation ability is the ease of motion in a particular direction that basically means that we have seen that from kinematic we have seen

that $\dot{x}$ two link manipulator this is θ_1 this is θ_2 this is XY so $\dot{x} = J\ \dot{\theta}$ and we have seen that $\dot{\theta} = J^{-1}\dot{x}$ which means that $\dot{\theta} = \frac{adj|J|}{\det|J|}$.

Now wherever $\det|J| = 0$ it ends up in a singular position. So that means you should avoid singularities. Now singularities basically mean that your energy requirement become infinity and that is the point where you cannot go. So, for more details please refer to the earlier lectures and also for introduction to robotic course some of the textbook. So, very briefly speaking this relationship is a relationship between the end of vector velocity $\dot{x}$ here and the joint velocity $\dot{\theta}$ and the $\det|J| = 0$.

Then what will happen is your adjoin velocity will become infinity and to avoid that we say that there is a measure which is called the manipulation ability measure which is defined by $W = \sqrt{\det|J|\,J^T}$ for redundant manipulator so this is my W or W = $\det|J|$ so this is for redundant systems and this is for non redundant.

Why they are different because in case of redundant manipulator okay J is non square it is m into n so this is a non square matrix. So, basically if you multiply J X J transpose what will happen it will became a square matrix then you can take the determinant and find the square root whereas in the case of non redundant system basically J the square matrix m X m so it is a square matrix so there is no problem so it is determinant of J.

So, this is a measure which tells us which indicates the ease of motion in a particular direction. So, if you take the singular valued decomposition of J you can get SVD of J will give three matrixes use σ, v^T this is called my singular matrix which will be coming out like this singular values $\frac{\sigma_1}{\sigma_n}$ and this is basically the direction of an ellipse. This is the major axis and that is the minor axis of the ellipse it is my σ_1 and this is my σ_n.

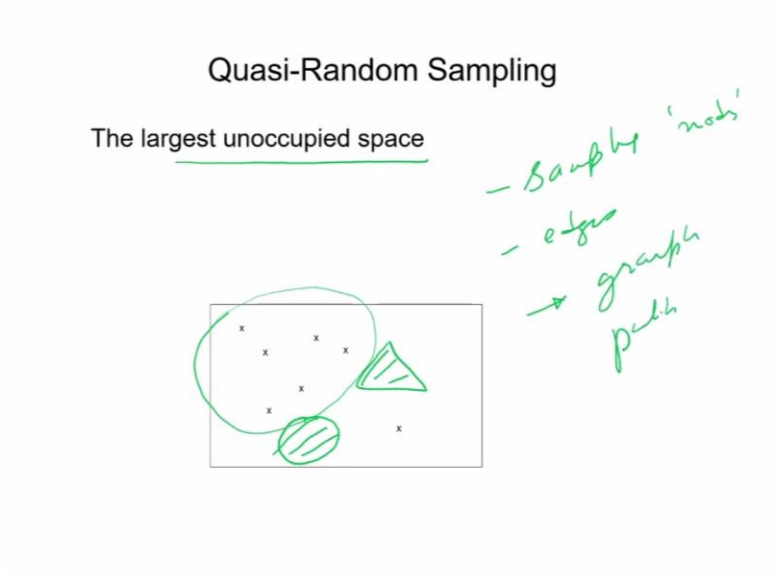
So, this σ is called the singular values of a matrix and $\sigma = \sqrt{eigenvalues}$ and the determinant of a matrix $\det|J| = \sigma_1\sigma_2\sigma_n$ and what does it basically indicate it gives the direction of an

ellipse and the direction of the axes is given by u and v^T which indicates which direction you can move easily.

So, in this direction it can move easily because σ_1 is much larger than σ_2. So, the sampling based manipulability sampling should be such that you should sample more in the direction in which the manipulator can move easily rather than trying to sample in a direction where it cannot move and also if it is going near a singular position there is no point in going near a singularity it is like an obstacle you cannot go there.

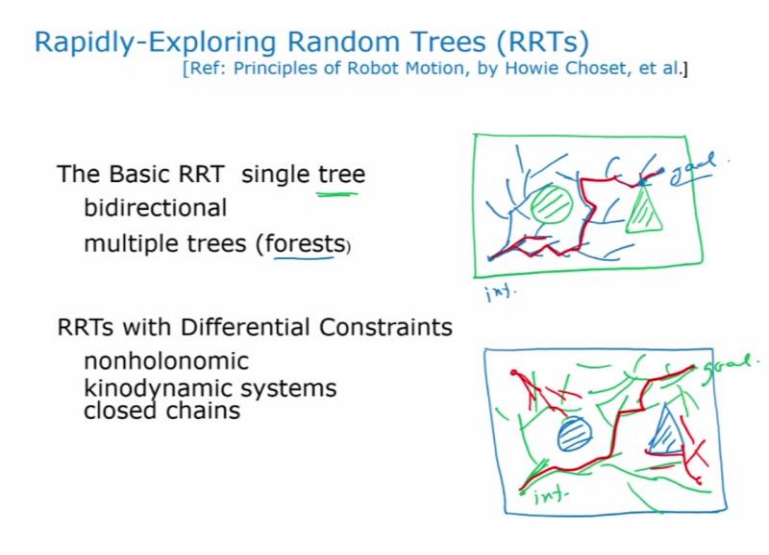
So, do not sample near singularities because the manipulability is very low there one way of trying to sample by using this manipulability measure is to sample near in directions in which manipulability is increased direction in which the manipulator can move very quickly. So, this is manipulability based sampling.

(Refer Slide Time: 40:38)



Quasi static sampling is basically we are sampling the largest unoccupied space. So, suppose I have work space like this and this is the unoccupied space so you do sampling there in a quasi random sampling method. Now, that is one way by which we did our sampling we generated so we did sampling then we generated our basically nodes then here we generated our edges and from the edges we made our graph and then we found a path that is how we did.

(Refer Slide Time: 41:18)



Now there can be other ways of doing this. One is what we call the rapidly exploring random tree or in short it is called RRT. Now this RRT is a very interesting way by which it operates is that it is like a tree as the name indicates. So, if we have work space suppose we have an obstacle here and obstacle there. So, in this particular case what we are doing is suppose I start this is my initial point so I start growing a tree it is like this is the trunk of the tree these are branches of the tree and the branches of the tree are increasing in that direction.

It is like we are growing a tree now branches cannot go obviously inside the obstacles so this is a tree I am growing and these are all branches of the tree. So, it is going from initial point to the goal point and then in order to find the connection what we can do is we can go along a particular branch and so I do not take this one I will take this like this, like this and then I see if I can go from here I can go to the curve.

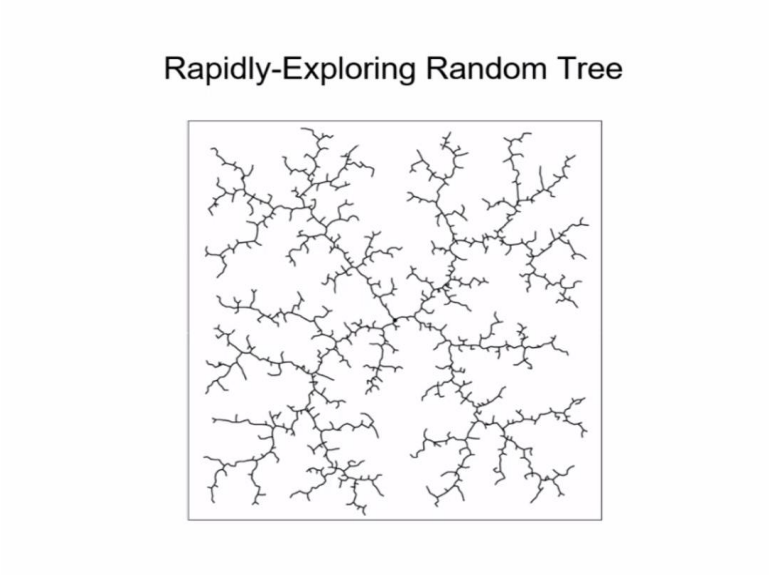
Wherever there is a path let us say there is a path there. Now this can be unidirectional so I have grown a single tree which we have gone from one direction to another direction then you can also have trees growing in multiple directions like in a forest so I have the obstacle here and we have a triangle here and we are starting from the position of the initial point here to the goal point there.

So, what we are basically doing is we are going from the initial point to the goal point growing two trees. So, this is a tree which is growing in this direction so this is a tree which is growing in this direction and there is another tree which is growing in that direction let us say

we have another tree which is growing from here two trees are growing by direction. So, now what is happening is somewhere they will come and meet.

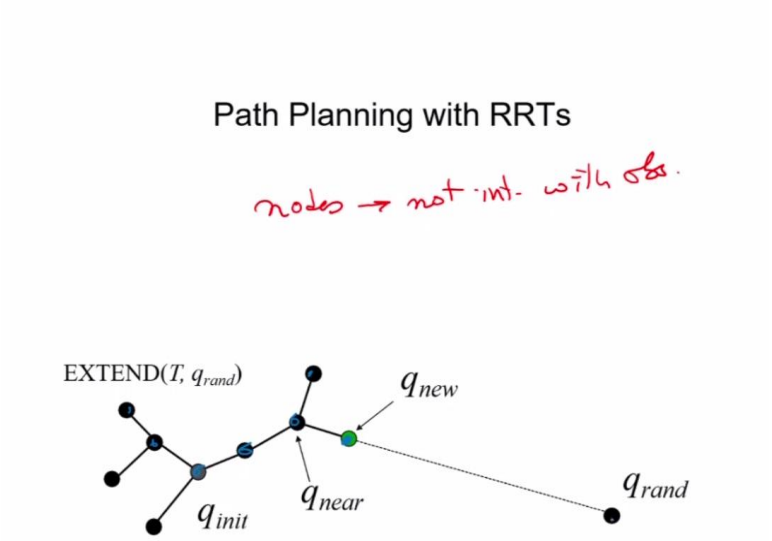
Now you can find the path and try and these are all the trees, branches of the trees. Now, basically you can find the connection like this, like this, like this, like this, like this. So, it is a bidirectional tree. Now you can have multiple trees growing in different directions one more tree from here and which can actually be a forest. So, this is basically called RRT or rapidly exploring random tree.

(Refer Slide Time: 43:50)



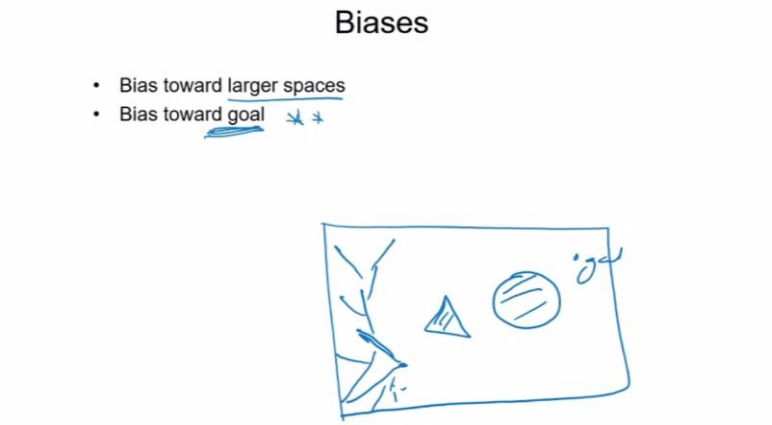
Basically what we do is in so this is a rapidly exploring random tree you can see here the trees is growing in different direction it is a start off from the center and it is growing in all directions. Now we can connect start point to this one and goal point to this one and then connect it.

(Refer Slide Time: 44:06)



Now, path planning with RRT suppose we are starting off again as in earlier case we need to have nodes not intersecting with obstacles. The nodes that are not intersecting with obstacles so we have started from here and then these are all nodes which are not intersecting with obstacles and we have come here this is a new one. Now what I need to do is I need to generate another node which is here let us say q_{random} and then see the connection between them.

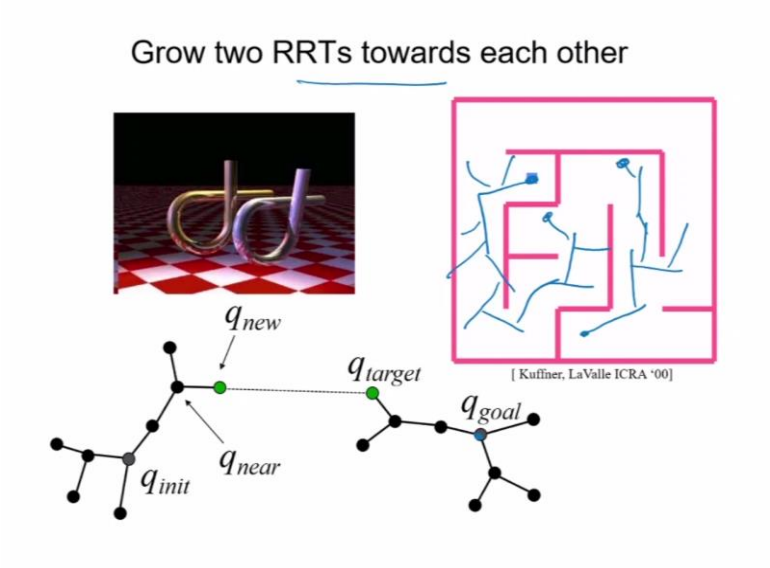
(Refer Slide Time: 44:42)



So, they can biased, for example, it can be biased towards large spaces, bias towards the goal. Bias towards the goal is a good idea the tree should grow towards the goal of course if I am growing a tree I have a tree and let us start from here there is an obstacle here and another obstacle there and my goal is here. When I start growing the tree in this direction the goal is in that direction so it should be biased towards the goal now it can be biased towards larger

spaces. So, biased towards goal can help the tree grow towards the goal and make your search faster.

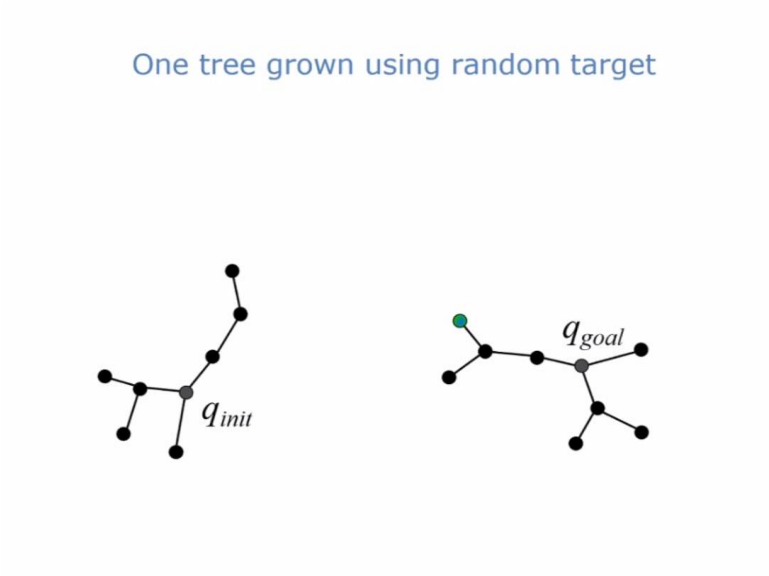
(Refer Slide Time: 45:17)



Now you can grow two RRTs towards each other. So, from the goal I start on and from the initial point I start another one. Now this is one tree this is another tree I want to connect them now and how do I connect these two trees. For example in this particular case we have initial point here goal point here I want to connect them. Now I want to grow up two trees one tree can probably come from here like this I am just giving an example.

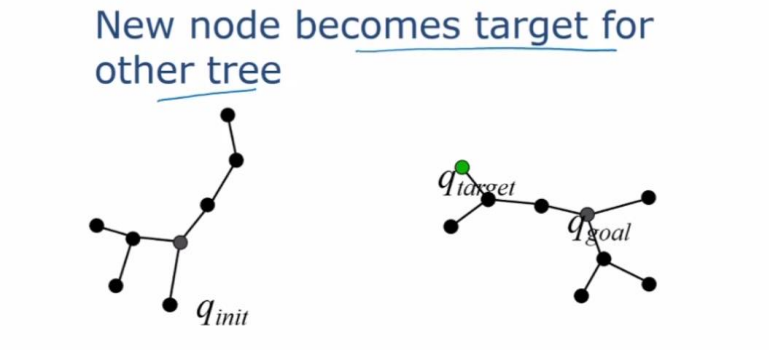
Another tree is coming from here and I need to connect the trees now. So, I need to connect this node and this node now and how do I connect these nodes that is something which we are looking at.

(Refer Slide Time: 45:56)



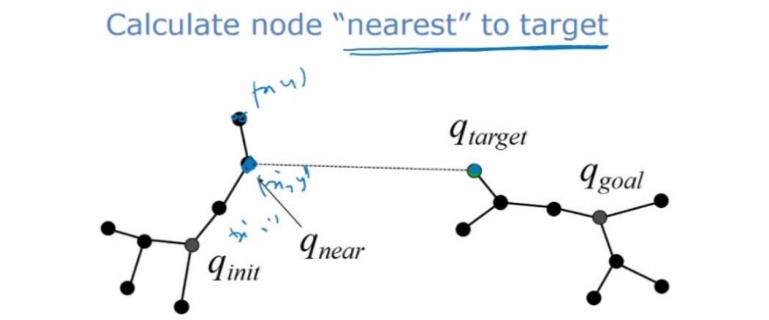
So, one tree grown using random target I want to see whether this can hit any point there it can hit any point of the initial tree.

(Refer Slide Time: 46:05)



So, what we do is a new node becomes target for the other tree. So, I generate another node which is going to become a target for the other tree.

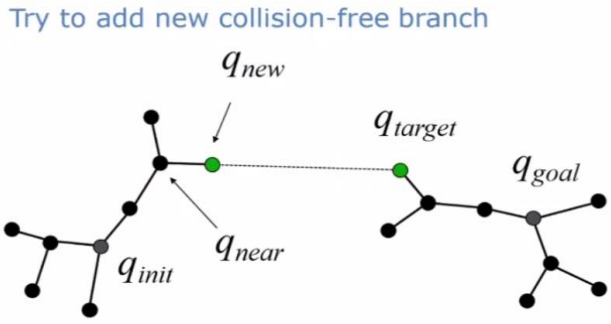
(Refer Slide Time: 46:16)



So, now how do I calculate? I can calculate the nearest node to the target. So, from here I want to make a connection so how do I get the nearest node? Now this again we have eyes we can see this is a computer program please remember these are coordinates and space $x'y'$. So, the computer program cannot see this directly. So, from here what it does is it has to find the target node to connect.

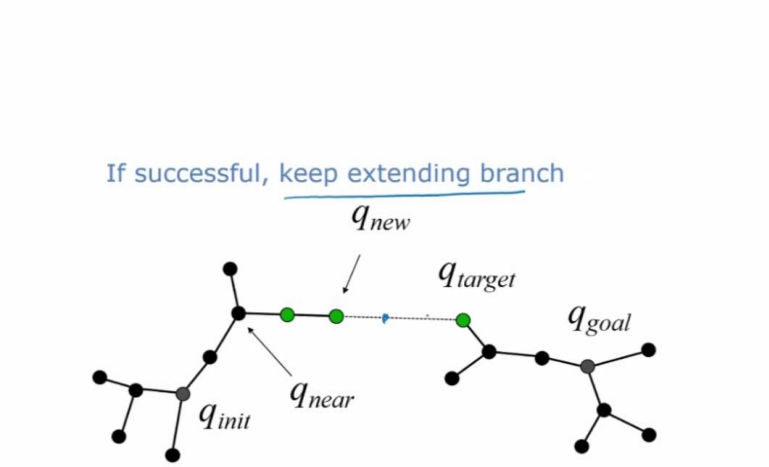
So, it can calculate a nearest node to the target so from the target this is my target so I find the nearest node this is my Euclidean distance so this is my distance which is nearest to that I am going to connect to this tree.

(Refer Slide Time: 46:58)



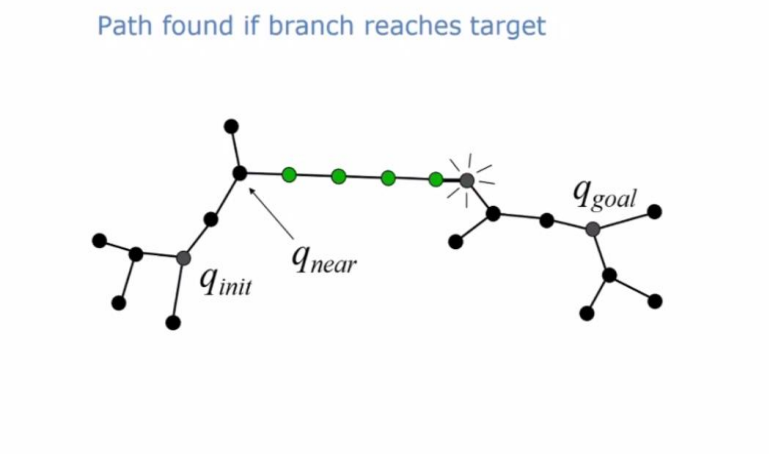
Now try to add new collision through branches so I have a q_{new} such that this the collection free branch which is connecting the q_{target} and q_{new} and q_{new} is connected to this one.

(Refer Slide Time: 47:14)



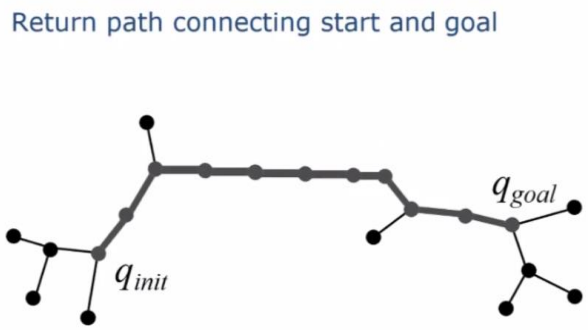
So, this is how and if this is successful keep extending the branch. So, what we are doing is we are making these new nodes from the target and putting this q_{new} such that ultimately it goes in q_{target}.

(Refer Slide Time: 47:29)



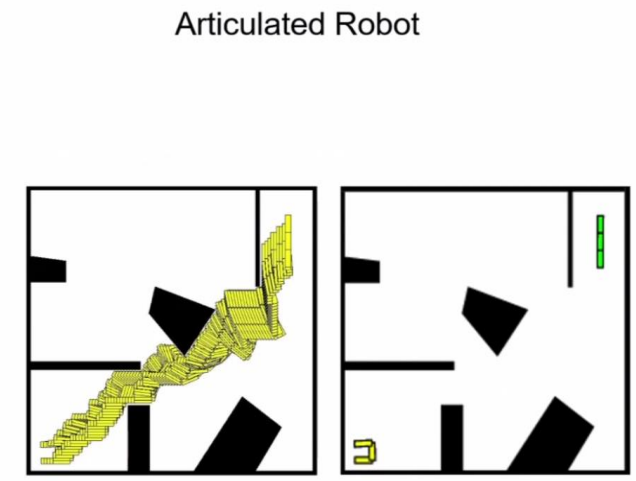
And then finally go and make the connection. So, if I just go back on this process, but I have a q_{target} the first thing I do is I need to this is my q_{target}. I need to identify which is the node nearest to it in terms of distance. So, I identify this is the shortest distance then what we do is from here along this direction I generate a node. So, I have generated a node here then on this line I generate another node here. So, in this way I am generating new nodes and then finally I am making this branch and connecting this with that.

(Refer Slide Time: 48:01)



So, now I have connected and made by trip. Now from the initial point to the goal point we are coming here and we are getting path.

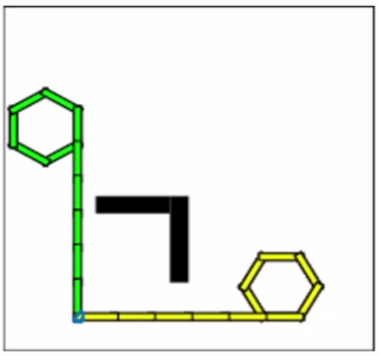
(Refer Slide Time: 48:12)



Now this is very useful in the case of articulated robot this is also useful mobile robot in path planning .

(Refer Slide Time: 48:21)

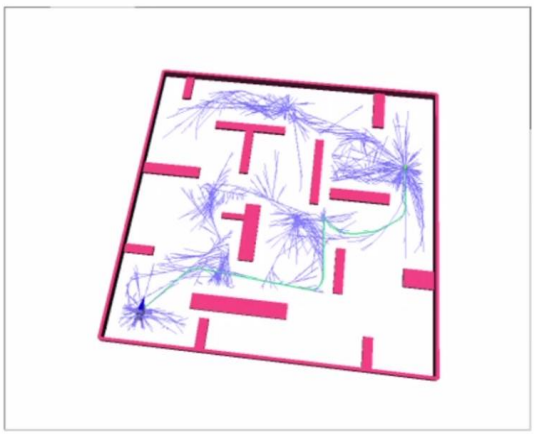
Highly Articulated Robot



There are lot of examples highly articulated robots large chain manipulators like this we have to go from one big configuration to another configuration. So, RRT is the method by which it can go very, very easily.

(Refer Slide Time: 48:31)

Hovercraft with 2 Thusters



These are examples we will come in terms of kinematic constraints little bit later. **(Video Starts: 48:37)** Let us have a look at the few videos in which for example here so this is the work space you can see here and this is an obstacle. The blue one is an obstacle here so this blue triangle here is an obstacle. I want to go from this point to this point. Again I am making the assumption that this is not having any kinematic constraint in terms of non holonomic constraint if this is a mobile robot it can move in any direction this is not constraint.

So, how do we start? I start making a tree from one side and we can see the trees increasing and it is increasing slowly and until it goes to the other side so this is the increase of the tree, increasing, increasing, increasing, increasing and then it is connecting the goal point. Now once it has connected the goal point I have gone from the initial point to the goal point. So, this is a unidirectional tree.

Now I can also have a bidirectional tree now this is sampling inside hollow passage. So, this is a hollow passage and we are sampling inside this hollow passage let me play the video from here let us say. Next is what we see is sampling inside hollow passages. So, this is the hollow passage which is shown and it is going finding the nodes and then going from one point to another point inside the hollow passage.

So, we can see it is generating the node and then going from one point to another point. Now this is another case where there are narrow passage and RRT is going in one direction from one side and is finally going and reaching the other side. So, you can see how the network of paths is increasing and that is reached now the robot is going from one point to the other point.

So, this is one way of finding a path by using the rapidly RRT method. There are few more examples. So, these are different configurations of the obstacles where we are going from one point to another point so it is exactly the same procedure. You can see it is increasing from left side and going up. There can be cases where it fails let us look at few such cases. Here it is an interesting case please have a look at this one.

So, here what is happening is that there is a very narrow passage and we have to go from the initial point to the goal point on the right and it is sampling, but it is not being able to find the narrow passage because narrow passage is very, very narrow there. Now this is another case in which we are having the same RRT which is going from one point to another point trying to go through the narrow passage and it says path not found.

Now this is an interesting case where instead of unidirectional we are using bidirectional. So, next one is bidirectional. So, the trees are going here from two directions one from the initial point and the other from the goal point and what is happening is that we can see that if it is

able to successfully sample in that narrow space then it is going to get the connection not otherwise.

So here also it is sampling, but then it is not getting a path because it has not managed to reach the corner of the samples. So, it is still a sampling. This is also showing you the importance of being able to sample in the narrow passages. Now let us look at this, it seems path not found, but you can very clearly see that there is a path..

In the next one what we are seeing is the next video is basically for a serial arm manipulator. Now for the serial arm it has to take the manipulator end effector and probe inside through there. So, what we will do just trying to find configurations which are possible which would take the end effector and put it through there using RRT. So, see as the video run as the video goes on we will see how it is finding.

On the right side of the program which is basically finding the sample points and connecting them by edges. So, you can see that the end effector is bending and then finally it is able to go through the other point that is hollow section there. So, now we will move forward and we will go through the narrow passage that we can see. So, it is found a way to go through put the end effector through that passage and also can come out now it is coming out.

Now this is another example of dynamic obstacle avoidance using RRT where the application of course as we come along we will see more application this is actually autonomous car, there is autonomous car you can see that the car is behind on the right side you see is the simulation. So, you can see the car coming out here that is the car. So, it has a lighter and a camera in front and the simulation is shown here on right side on here.

So, the moment it sees something in front it will stop it will find another path by shooting out another ray. So, there is a pedestrian who is approaching so we can see that immediately you could see so that was the person. So, it has changed his path exactly by using RRT and then comes back again. So, this is doing obstacle avoidance using RRT. So, this rays which are going out are like the branches of the tree that I was talking about.

So, this gives us an idea of how the system actually works in a real world situation. So, autonomous cars also use RRT. So, what we are seen is this autonomous vehicle actually

driving by itself and avoiding obstacles and this is an application of RRT. In today's class what we looked at is sampling based planning that is in which you do not have to search the whole work space you do not have to make the spaces.

And you can sample a part of the work space based on whether their near obstacles manipulation ability based sampling or sampling in narrow passages and then you can connect this path and find a path. So, this is a better way or faster way of finding a path to take it from initial point to a goal point especially in cases where you have very high degrees of freedom system. So, we will stop here today and we will continue in subsequent lectures. Thank you.