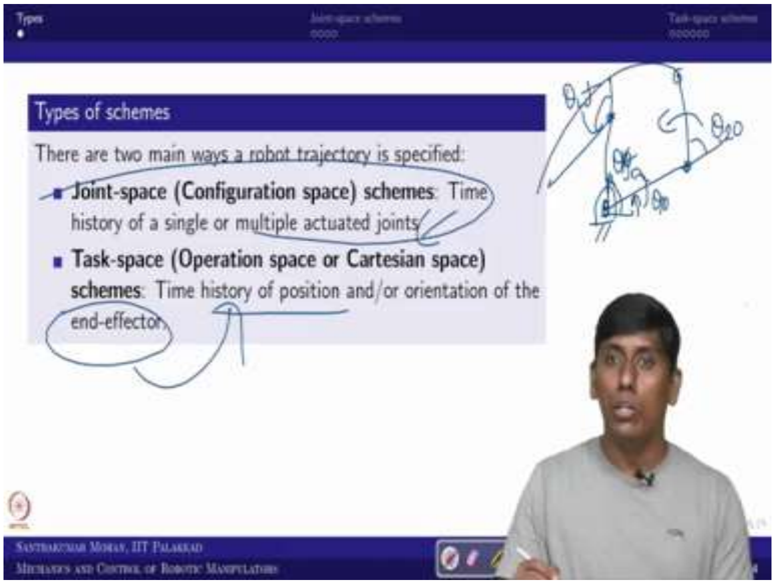
Mechanics and Control of Robotic Manipulators
Professor Santhakumar Mohan
Department of Mechanical Engineering
Indian Institute of Technology, Palakkad
Lecture No 31
Trajectory generation schemes for serial manipulators

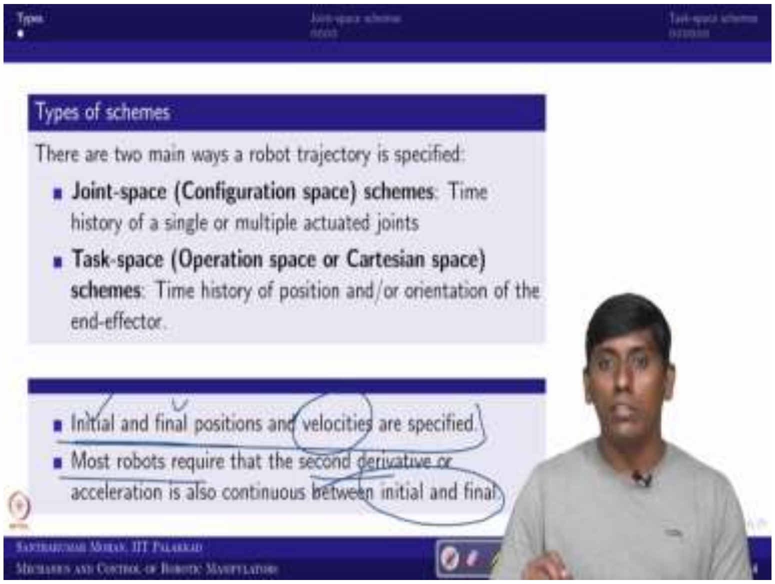
Welcome back to Mechanics and Control of Robotic Manipulator. Last two classes we have seen what is trajectory and how to generate a trajectory using a cubic polynomial or fifth order polynomial or cycloidal motion or straight away simple straight-line interpolation with parabolic blend. So, this particular lecture we are going to see along with a manipulator in the sense we are bringing some parameter of the manipulator.

(Refer Slide Time: 0:40)

So, for that we are trying to talk something called the types of what you call trajectory generation scheme for robotic manipulator in that you can see that there are two schemes that are available, one is joint space scheme, other one is task space scheme. So, this is what we are focusing in this particular lecture. We will see how that goes.

(Refer Slide Time: 0:57)





So, I already said there are two main ways a robot trajectory is specified we are specifically mentioning on manipulator. So, this all related to the manipulator case. So, in that sense we can call joint space or configuration space schemes. So, where the time trajectory or time history of a single or multiple actuated joints. So, whereas, the task space or operational space or some people call even Cartesian space schemes.

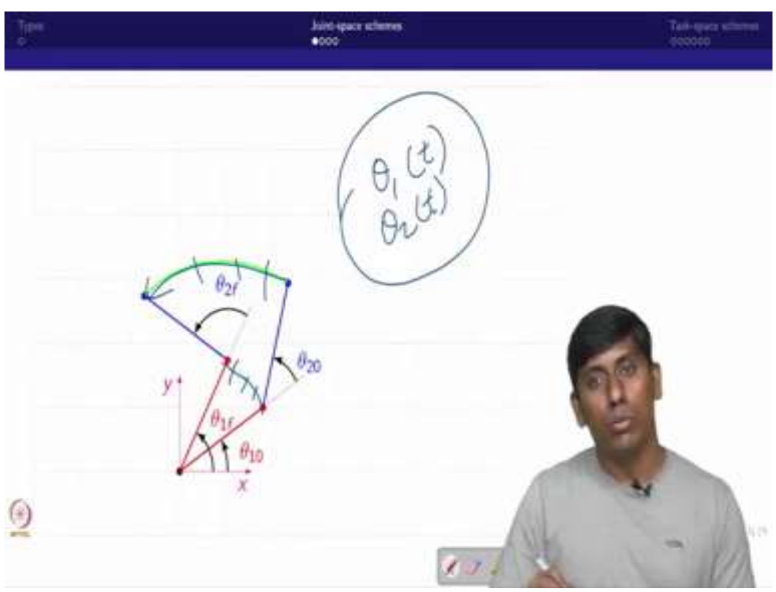
So, the time history of a position and or orientation of the end effector. So, in the sense, this is what we call end effector trajectory, this is we call joint space trajectory, or you can say joint position velocity and acceleration. So, in that sense, we will first talk about the joint space scheme even the joint space scheme can be done two ways. So, one is sequence in the sense series actuated joint.

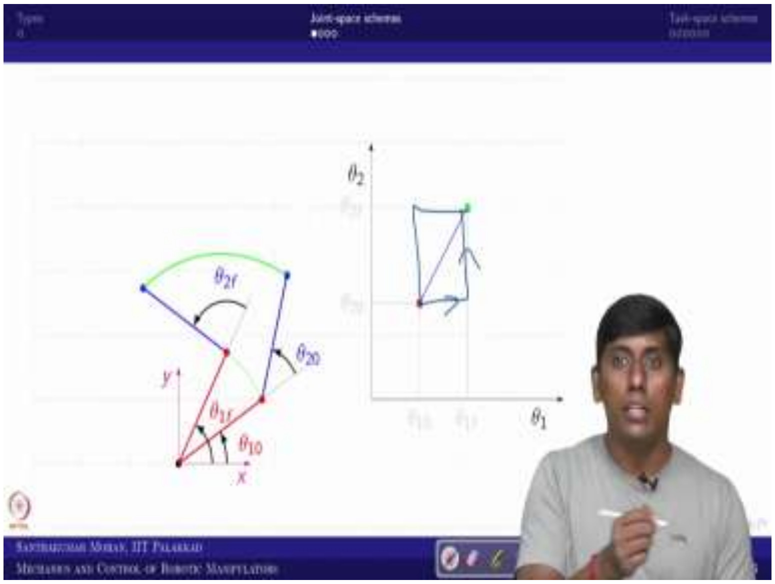
For example, now this is a simple serial 2R manipulator. So, first I rotate this then I will rotate this. So, in the sense I have another joint. So, now you can see like this is theta 2 f and this is theta 0 you can say 1 f and this is theta 1 0 and this is theta 2 0. So, now from here to here I can do first I rotate only first joint. So, I will keep its this then I will rotate the second joint, or I can synchronously rotate one to another.

So, we will see the first one before that we can just see that what are the things that are given to us. When we talk about manipulator trajectory generation, the initial final, initial and final positions and velocities are specified to us the acceleration is given or not according to the case if the acceleration given to us initial and final acceleration then we have to go for higher order polynomial.

But if it is not given only these are given then we can go for cubic polynomial. So, we already know this so sometime are most of the manipulator required what do you call second derivative in order to make the force finite or you can say force profile or you can say torque profile or the voltage which supposed to be given to the motor is required then people always go for the second derivative which is nothing but the acceleration is continuous which is given to us.

(Refer Slide Time: 3:29)





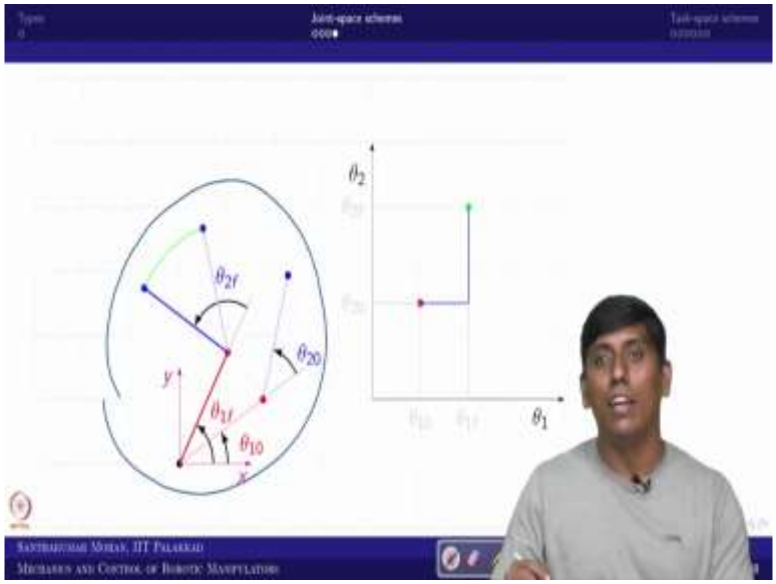
So, now in that sense so we will take one sample. So, we assume that this is the 0 0 location, and I am going to fix one of the 2R serial manipulate in this way. So, this is the initial joint position of link 1 and link 2 in the sense joint 2 this is joint 1 and this is the end effector. So, now this is given. So, now what additionally given the initial velocity and final velocity is supposed to be given that we assume in this case 0.

Similarly, we are trying to see what would be the final position and you can see velocity final velocities we assume 0 but this is the final position. So, now, what one can see so, how to reach from here to here and how to reach from here to here. So, I already said there is a two way. So, one way you can directly you can say synchronize so, where theta 1 of t and theta 2 of t can be synchronized in the sense both links 1 and 2 will follow the profile.

So, each segment and maintain this. For example, this made in 4 seconds. So, this equally it is making it enough four segment. So, like that it will go. So, that is what we are trying to see as the first scheme. So, in the sense that theta 1 and theta 2 I plot it where this the initial and this is the final. So, what we are trying to do? We are trying to go as a straight line. So, although we are going to follow a cubic polynomial, but we are trying to reach in a straight or you can say straight line interpolation.

So, this is what we call the first type of joint space scheme where it is synchronously moving. The other one it is go like this are goes like this. So, we will take this is the first one is easy where the first link or first joint rotate the second joint rotate like that.

(Refer Slide Time: 5:29)

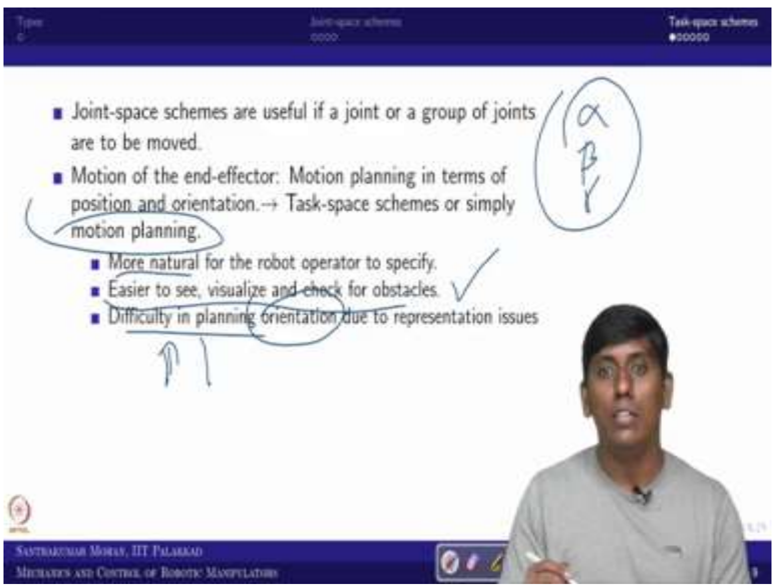


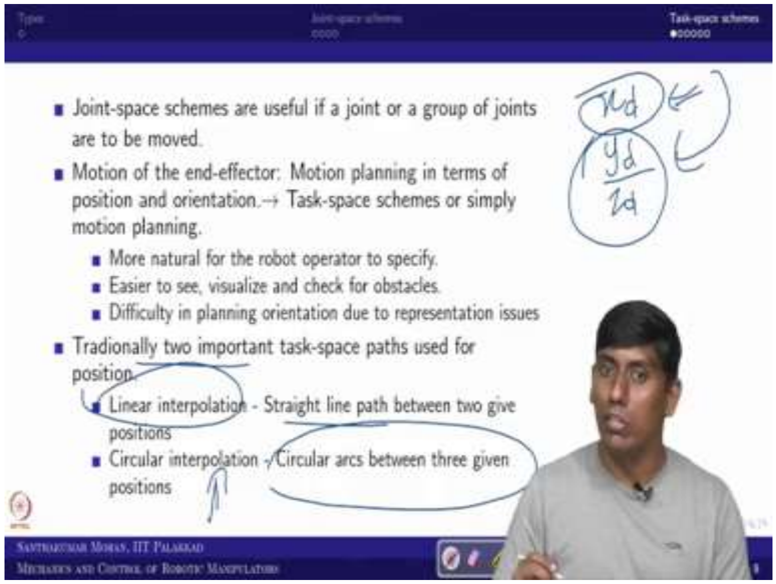
So, in the sense we take the same condition. So, these are the initial and final but what we are trying to do first we are rotate the joint 1. So, what we are trying to do first we rotate the joint 1. So, this one reminds so, just to rotate you can see so, then we will rotate this. So, that is what we do. So, in the sense that this is the first segment, and this is the second segment. So, this is what we call sequence joint space scheme the other one is synchronous joint space scheme.

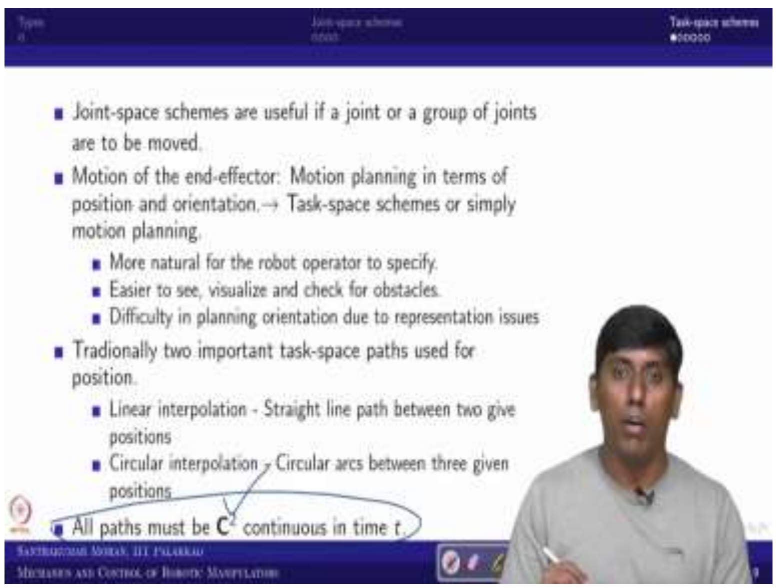
So, usually most of the robotic manipulator community use the; what you call synchronous one, the sequence one is very common for example you take robot operating system with the gazebo, or you can say move it and all. So, you can see like each joint would be a rotate then the next joint rotate like that it will be showing.

In the sense it will show the sequence is there any collision happening or not those things all will try to understand. So, this is what we call the joint space scheme. So, when we come to the task space scheme, which is little you can see peculiar kind of thing.

(Refer Slide Time: 6:41)







So, what we can see the joint space schemes are useful as long as you are using in a direct you call configuration space control. But you think about something like you have something like a scene tracking or probably a line following or curve following and all. So, you need to see how your end effector is following. In that sense so, you are bound to follow what you call the task space trajectory in the sense the end effector profile needs to follow for a given path.

So, now this trajectory you need to generate in smooth enough. So, for that we are coming back to the you can say end effector trajectory generation which is nothing, but we call motion of the end effector the motion planning in terms of position and orientation of the end effector. So, what that mean? So, we are trying to use task space scheme most of the people motion planning means nothing but task space you can say generation or trajectory generation.

This is what we people usually see. So, there are certain constraints that is what we are trying to address. But this is more natural because you can see realistic and try to see because the robot operator or the robotic task always defined with the end effector. So, then it is more natural for example, I want to pick this and place it here which is very natural to follow this.

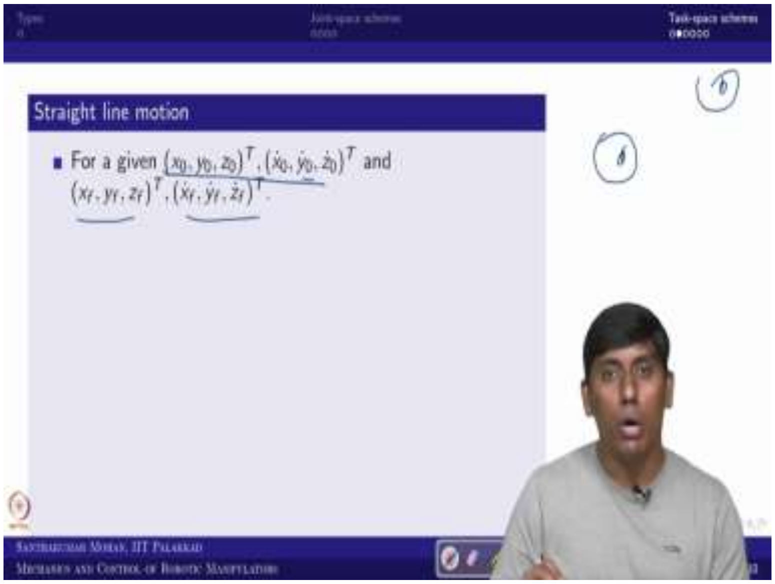
But if I do the same thing through the inverse kinematics and try to make a joint space scheme that is little you can say unrealizable one. And further you can see it is very easier to visualize, see, realize and even checking for obstacle everything is easy. If you talk about that is why we say this is more close to motion planning further what happened the difficulty in planning is exists which is we call limitation mainly on the orientation due to the representation issue.

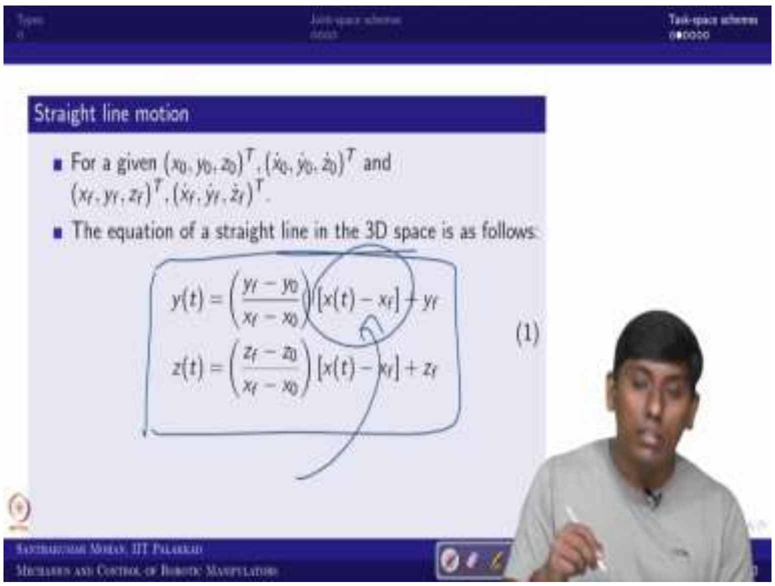
So, that representation you know the alpha, beta, gamma we call the roll pitch angle is not explicitly visible to us. So, in the sense the representation is complex in that sense the difficulty in orientational generation or trajectory is always difficult. So, further what you need to know traditionally there are two important task path we have to consider. So, which is universally follow, one is linear interpolation which I already shown.

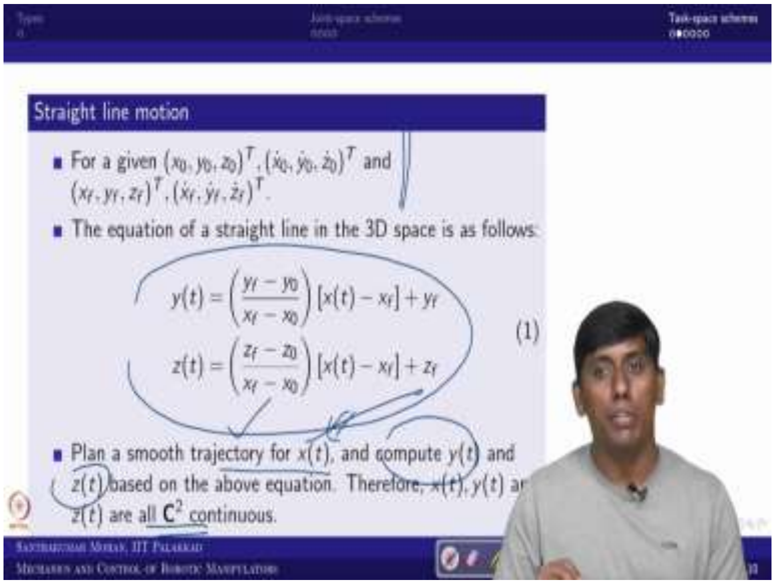
Which is nothing but a straight-line path even you want to go for a position tracking. So, you have x desired y desired and z desired needs to be generated, you can just generate only one and then you can generate based on that into two other that is what we are going to see in the next one. The circular arc is for given three you can say position we can make an arc, and do it which is not common and I am not addressing in this particular lecture.

If anybody interested, we can discuss in the live session. So, in that sense we are going to what you call one important constraint any path whether it is you call joint space trajectory or you can say task space trajectory it should be twice differentiable which is nothing but so C 2 Continuous. So, that is what we are trying to say in the sense the profile is smooth enough up to at least acceleration level. So, that is what we are trying to see.

(Refer Slide Time: 10:12)



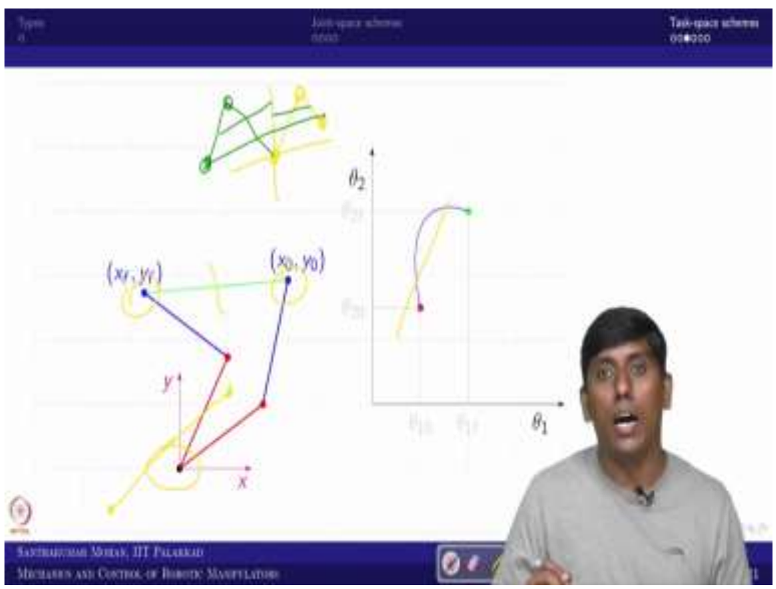




So, in that case so we assume that the initial position, initial velocity, initial sorry, final position, final velocity is given for a straight line this is the sense so, the spatial position and velocities are given to us. So, now we want to generate we assumed that accelerations are not consider. So, in the sense that straight line path for 3D space we can generate in a simple straight-line interpolation where the x of t is the only variable which you need to derive.

So, this x of t you can generate either cubic polynomial or fifth order polynomial all those things you can do it since here only initial and final velocity and initial and final positions are given. So, better we can do the cubic polynomial and then we can compute what you call y of t and you can say z of t. Further you can ensure that since you are following this is continuous, so, then obviously the y of t and z of t also continuous. So, that is what we see.

(Refer Slide Time: 11:21)



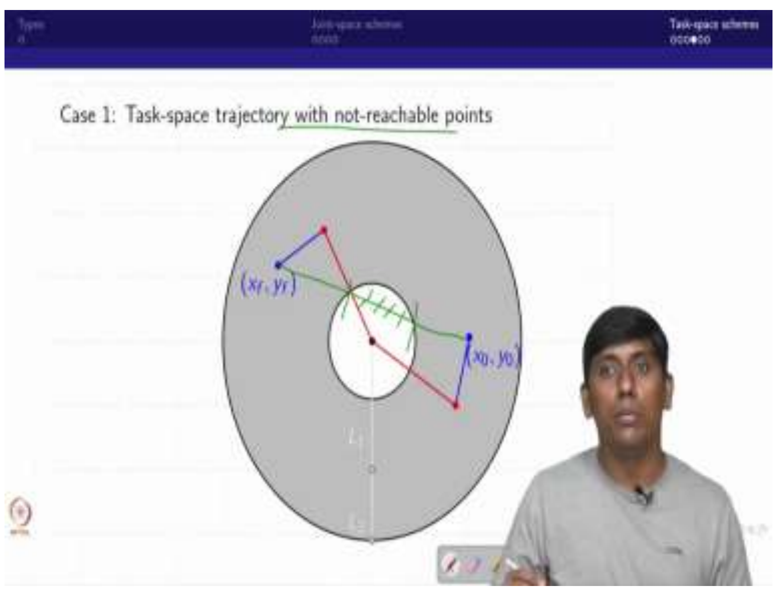
So, for understanding that we can take the same example which we have seen in the joint space trajectory. We can see that these are the two-given position. So, this is the initial position, and this is the final position the initial and final velocity we assumed to be 0. So, what we need to follow? So, you need to generate the trajectory which is on the straight line. So, this is what we are trying to do but if you generate this what would be the joint space trajectory look like?

So, we just wanted to see so, for that I am taking this is the corresponding so, theta 1 0 and this is corresponding theta 2 0 and this is theta 1 f and this is theta 2 f. So, now how look like earlier case it is a joint space trajectory which was I can show it in yellow this is a joint space trajectory generation. But the task is trajectory generation something like this. So, that is what you can see. So, now, this may end up with a few issues those issues we are seeing.

So, this can go outside what you call the workspace possibility. Because this point may be inside workspace, this is inside workspace, but the path can go outside of space this is one possibility. Now, other possibility is this path can for example, this is the initial position and is the final position I make a straight line. So, this zone may be near singular case in the sense that joint acceleration would be infinite.

Similarly, joint velocity would be infinite in the sense in finite joint torque is required to control that which is not preferred. Similarly, the other way around so, you have two situations can happen. So, for example, this is one position. So, this is one of the manipulator positions and the other position is just a mirror image. So, then if you are making it so, it is supposed to come to the other path so, it goes like this suddenly it has to come back it is very difficult to follow that way. So, in that sense the swapping may end up with you can say discontinuous velocity and acceleration these all can happen.

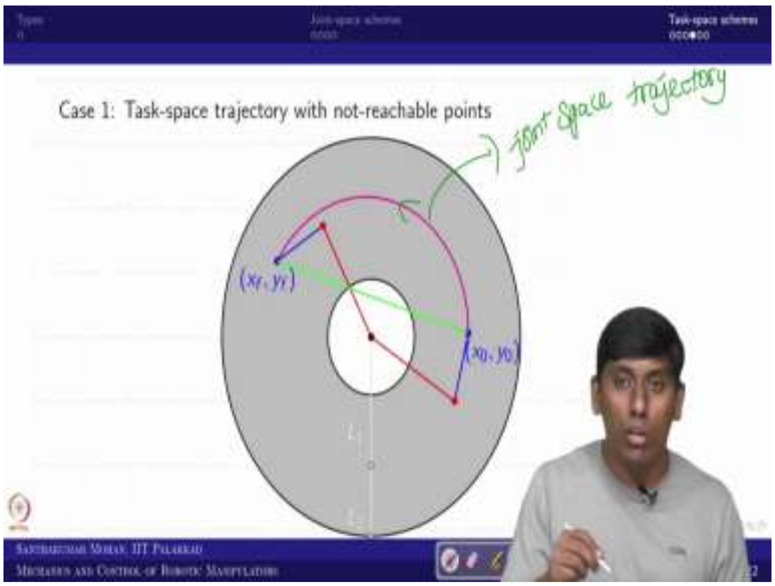
(Refer Slide Time: 13:52)



So, that is what we are trying to see for that we are just taking a small manipulator configuration where L 2 is smaller than L 1 we assume that both joint 1 and joint 2 can rotate 360-degree rotation. So, in the sense the workspace would be look like this. So, this is the workspace. So, now we take 2 points so, one point is here. This is the initial point, and this is the final point. So, now what you can see the path which is supposed to be generated in this way. Now, what you can see this segment is not trackable. Why? It is outside the workspace.

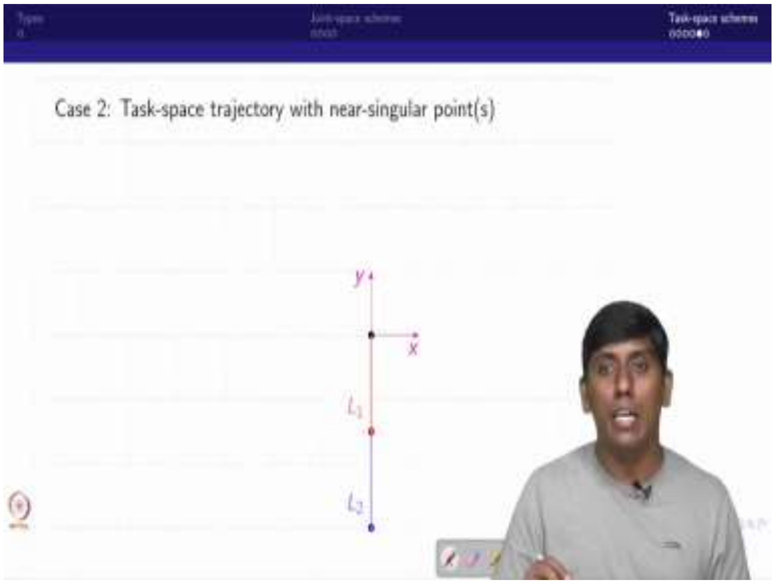
So, here it cannot maintain in this sense, it is not due to the joint limit restriction, it is due to the link or geometry restriction. So, not joint angle restriction it is just a physical geometry restriction. So, in this sense this is not possible. So, that is what we can say that the task space trajectory can end up with you can say not reachable points. This is one you can say case can happen.

(Refer Slide Time: 15:01)



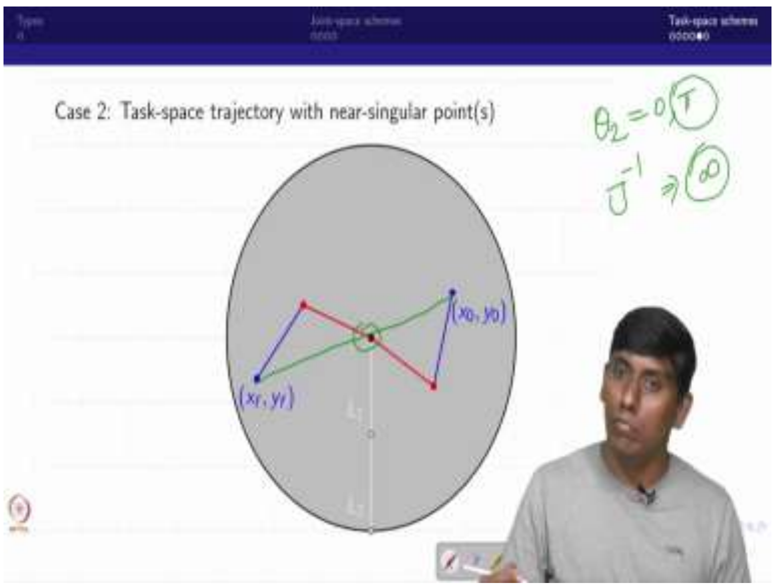
The second case what we are see even the better thing is if I generate this in a joint space trajectory still it is possible to track. So, this is a joint space trajectory. So, anyhow we will see this all-in simulation which will give more insight. But right now, we are trying to see what can be the cases. So, in that case this is possible if you do in joint space trajectory but what happened is not trackable in the task space.

(Refer Slide Time: 15:38)



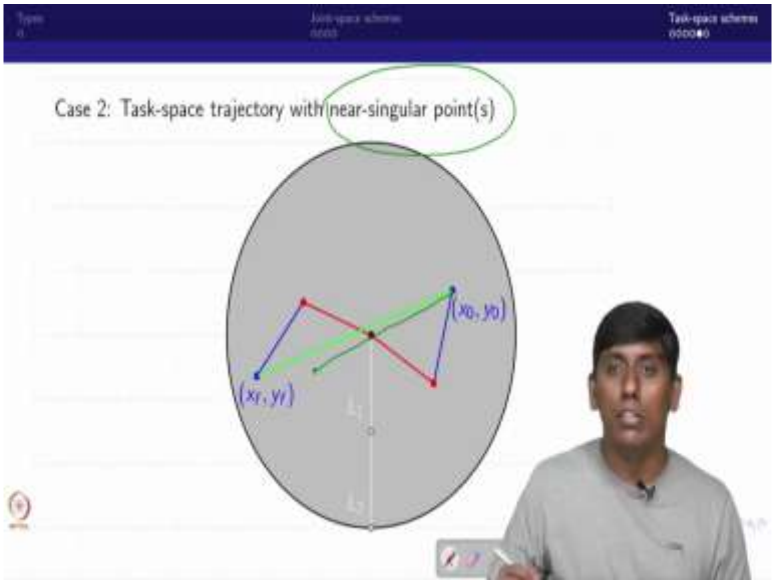
The same thing we can try to see. So, now L 1 and L 2 are same in the sense L 1 equal to L 2 now, what would be the workspace, workspace is a simple you can say filled circle.

(Refer Slide Time: 15:51)



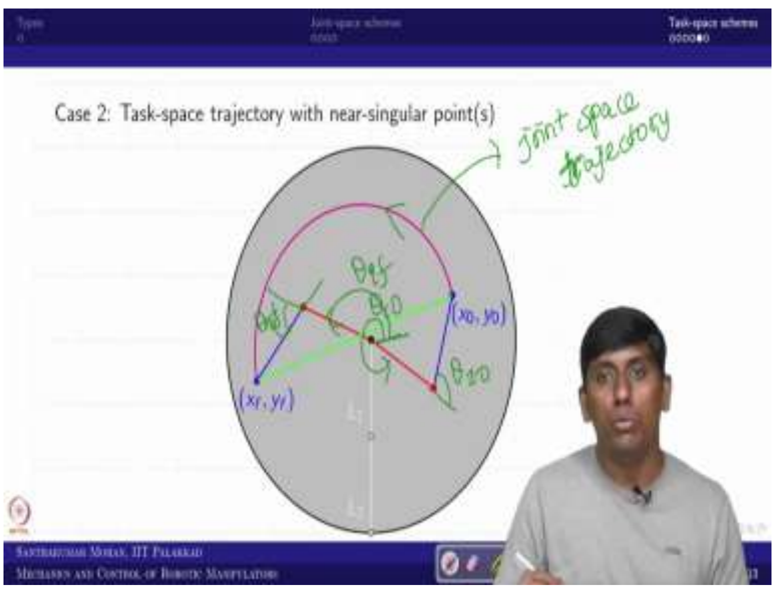
So, now we take the similar way there are two points I intentionally take this way. So, now if I can write you can see that it is coming near singular. So, what singular? So, where the theta 2 can be 0 or pi. So, in this case it can be go till phi or very close to phi in the sense the Jacobian inverse can go to an infinite goes to infinite. So, in the sense you are joint space acceleration can go infinite which is not possible.

(Refer Slide Time: 16:27)



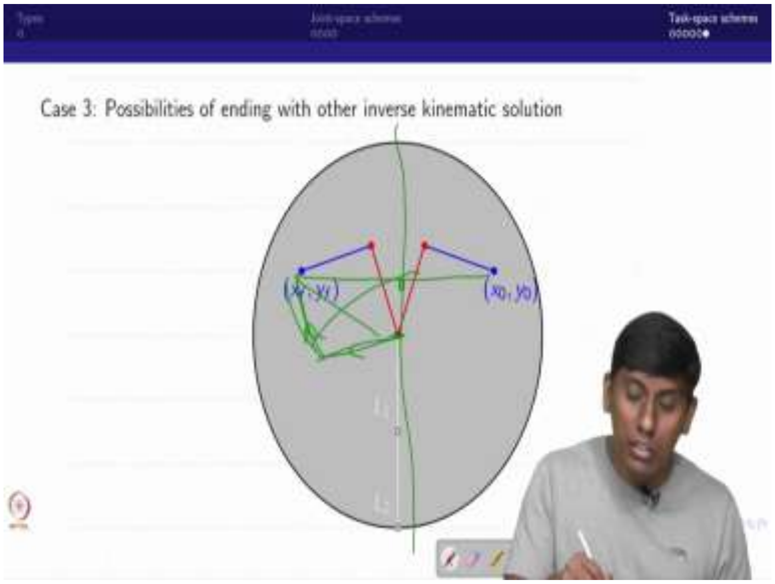
So, this is another case which is not exactly the singular point it is near singular point even some cases it can even pass through the singular point. For example, I take one point here. So, you can see like it is passed through the; you can say even the boundary.

(Refer Slide Time: 16:44)



So, in that sense this can be again overcome with the joint space trajectory. So, if you take joint space so, still this is so possible. So, this is possible from this point to this point. Only thing is So, you have this is theta1 0 and this is theta 2 0 and this is theta 1 f and this is theta 2 f that is what you would be knowing it.

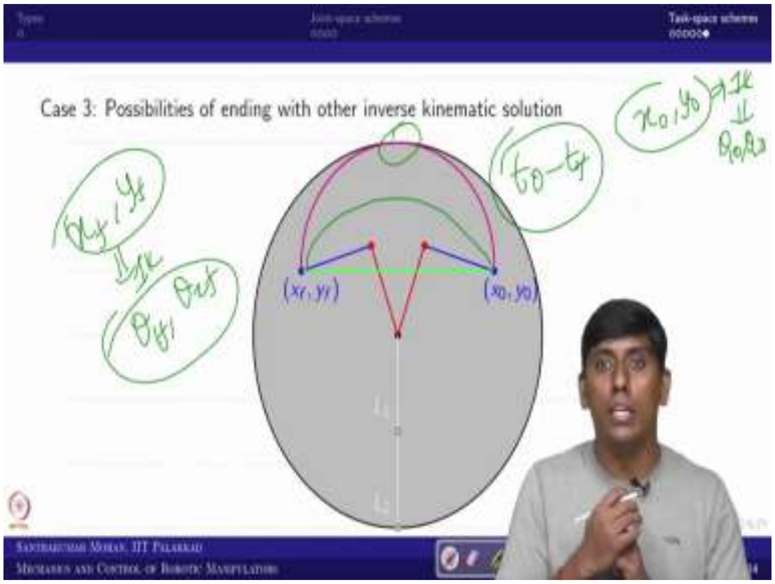
(Refer Slide Time: 17:18)



So, now we can see what would be the third case. So, possibility of ending with other inverse kinematics solution. So, very simple example this is what the given condition. So initial and this is the final you can see we just to make a line here it is look like a mirror image. So, in that sense, you can see like it is supposed to follow this straight line this instant need to change over for following the other path if it is simply going this as a joint space trajectory it is possible.

In the sense so this solution so this joint space solution is possible or if I give this configuration, it is still possible. But if I you this kind of configuration it is not possible that is what says that so it is ending with the other inverse kinematics solution then also it is not possible.

(Refer Slide Time: 18:13)



So, then people can think about even the joint space in this case even the joint space trajectory is very close to a singular point. But probably this example I took it that way but most of the cases this can be even still followed with joint space trajectory. So, how we can do the joint space trajectory. So, we can take this x naught and y naught and you use the inverse kinematics solution, and you can find theta 1 naught and theta 2 naught.

Similarly, you take x f and y f and take the inverse kinematics and you can find theta 1 f and theta 2 f and use t 0 to t f as the trajectory generation and then you can make the path. So, that way you can do it. So, that is better for making the task space trajectory. So far what we have seen. So, we have seen so many what you call trajectory generation schemes and even the type of you can say trajectory generation possible in manipulator.

One is joint space even in joint space there are two subclasses and in task space, we have seen even in the task space we have seen one specific interpolation method which is straight line interpolation which is universally followed that is why we have seen that and even if you follow that what would be the issues that issues, we have already seen. So, in that case, the next lecture what one can expect we will see first how individual trajectory can be generated that we will see in simulation.

Then we will see how that individual trajectory can be ported in joint space scheme or you call task space scheme. That is what we are you can say focusing in upcoming lectures. Until then see you, bye, see you then.