

IntroductionToRobotics-Lecture10

Instructor (Krasimir Kolarov): Good afternoon. My name is Krasimir Kolarov. I am going to be teaching the lecture today and also the co-author of the notes for the course. So if you have any complaints, direct it to me. If you have any praises, direct it to Oussama. I did my [inaudible] here at Stanford about 16 years ago. So I was in your shoes, and I've been kinda doing a few lectures as well as some of the classes completely since. I'm not working in the robotics area right now, but I'm staying pretty current in that.

So we're going to start as usual with a short video snippet. Do you wanna play the video?
[Video]

Suppose I need to deliver an emergency case of cold drinks to my friend Keith who lives about a half mile away, but I'm too busy to drive over. Fortunately, I have a 1990 model Nab Lab, a computer-controlled van equipped with television cameras to see the road, a scanning laser range finder that measures 3-D positions, computers to digitize and process the images and computer-controlled [inaudible]. I toss in the case of drinks and fire it up.

The Nab Lab built a map earlier by watching as I drove it around the neighborhood, including the locations of roads, shapes of intersections and the locations of 3-D objects. I add a few annotations to the map to tell the Nab Lab where to speed up, when to slow down and where to stop. I hit the run switch, step out of the Nab Lab and [inaudible].

The Nab Lab has several different ways of seeing roads. It needs hints from the map to know which roads to use [inaudible]. I told it to drive along the street using images from the color camera processed by a fast-simulated neuro-network [inaudible]. It digitizes images from a color camera and processes them to enhance the contrast between road and off-road. The enhanced images are fed to a simulated neuro-network, which has been trained by watching a human drive along similar roads.

Now this neuro-network directly outputs steering angles to the Nab Lab's steering wheel. When the Nab Lab approaches intersections, the cameras see only asphalt, and [inaudible] is unable to interpret the images. The map gives instructions to switch to landmark navigation. A laser range finder finds 3-D objects on the side of the road it has previously recorded in the map, and uses those objects as landmarks to update its position on the map.

Once the Nab Lab knows exactly where it is, it can drive fine using its inertial guidance system long enough to traverse or accurately turn through an intersection.

Leaving the intersection, the Nab Lab's map tells it to pay attention to its color cameras again and to increase its speed. [Inaudible] finds the road again and steers the Nab Lab towards its call.

Finally, the Nab Lab uses dead [inaudible] to predict when it should be approaching Keith's house, uses 3-D sensing to find his mailbox and comes to a stop. The drinks are still cold. [Crosstalk]

Instructor (Krasimir Kolarov): There should be a sound with the video. I can make [inaudible]. It's basically a navigation for a car. He's riding in his car several years ago, actually, well before the [inaudible] that make Stanford so famous in that area. So this is one of [inaudible].

That's – we can stop here. Thank you. So that has to do – the topic of the lecture today is trajectory generation, and it's quite relevant to the video that you saw because, in addition to the control functions – the sensor functions – the underground – the underlying mathematics is really planning for a path for an object among other objects, and that's basically trajectory planning.

So what we're going to be talking about today is really the basic mathematics, and that can be used at higher level planning concluding the run with the navigation video.

So we're going to design the project first. So we have a manipulator arm, and it's starting – we wanna move the manipulator arm from some initial position. [Inaudible] with the frame T sub A to some goal position, which will be the desired position: T sub C . And the manipulator has – is basically in a stationary frame, which is S in this case.

So we wanna plan a motion for the manipulator arm from T sub A to T sub C . In addition, to make things more interesting, we might have to go through some intermediate points, like for example, T sub B , and we have that because if we have an environment with obstacles and the manipulator is moving in that environment, you wanna make sure that you're avoiding the obstacles, in which case, you're introducing intermediate points for the manipulator to move for.

So this is the basic problem, and some terminology – we have puff points, the initial, the final point and the vital points that we'll be going through. And then, what we want to plan is the trajectory. The trajectory in the simplest case is a time history of the position, velocity and acceleration for each of the degrees of freedom. For the purposes of this lecture, and basically the class, will be planning each of the degrees of freedom independently, and then assume that the motion is realizable as a whole. Okay? Because once you put them all together, it starts getting very complicated.

So for each of the degrees of freedom, we'll be planning the trajectory.

What kind of constraints do we expect to see so there can be spatial constraints? Obviously, obstacles in the environment that we don't want to collide with, time constraints if the motion has to be done in a particular time frame for – especially if we have a industrial operation that we are trying to achieve, and everything is going on a [inaudible], and you have to do it in a particular time – and smoothness. You want the

manipulator to have a smooth motion because that uses much less energy, and it's easier to control.

So these are the type of constraints that will be into the problem.

Okay. So that's the setup for pretty much everything that we'll be discussing today. Initial point, goal point, intermediate points, constraints, and we're going to be charting the time history.

From a mathematical point of view, it's a very simple problem. Right? We are planning path.

We can look for the solution in several spaces – two main spaces, really. There is the joint space. For the manipulator – that's the native space for the manipulator. Right? So we want to go – in that case, it is easy to go through this intermediate point because we will know exactly what the joint configuration is going to be for the robot in this intermediate points. So at those points, in order to get the actual joint characteristic, we'll be solving the inverse kinematics at all the path points, and then we'll plan for a motion in that space.

So let's say we have the coordinates of a few of the points that we want to pass through. We solve the inverse kinematics for all these target points. We get the corresponding joint coordinates, and then we plan in joint space trajectory. Okay?

So that's pretty simple, and it's much less calculations. There is no problem with the singularities because the singularities occur in joint space. That's where the manipulator cannot move in a certain way because the physical structure is precluding it from doing that. Okay?

So in joint space – that's immediately obvious.

Less calculations. We are only doing the inverse kinematics at these target points. A problem – we cannot follow a straight line. Right? That's the simplest problem. Let's say we calculate the joint coordinates for the immediate in the initial point and the target point – forget about intermediate. So we have that. We converted it into joint space. We plan a path in joint space, but we have no idea whether that path in actual contingent space where the robot is moving is a straight line or what it is. It is what it is. Okay?

So we cannot force a particular trajectory very easily. Okay? If that's not – if that is not a problem for us – if we are okay – if it is not exactly straight line, but it approximates it – that's fine. So we have less calculations. But if we wanna follow a particular trajectory, doing it in joint space is very difficult.

That's much easier to do in contingent space. Right? We can actually track a shape because we are putting the points that we wanna go through in the actual contingent

space where the manipulator moves. If we give it a straight line to move on, it will move on straight line. Right?

So – and how do we do that? Basically, we plan in a contingent space using the coordinates, and then to find the orientation of the robot, we can use any of the mechanisms that you've learned so far: equivalent axis, [inaudible] angles – all these mechanisms to compute the corresponding angles for the joints. You can use those formulas. Right?

So we can track a shape here. It is much more expensive at one time because what happens? We have an initial point. We have a goal point. We track – we plan a trajectory. How do we make sure that the robot actually goes along that trajectory? Basically, at real time, we have to sample points along that trajectory often enough to force that kind of trajectory and then compute for that all the joint coordinates and make sure that we feed that to the actual robot to go through that. Right? So it's much more computationally intensive to force that particular trajectory.

Okay. And the other major problem is that we have a discontinuity problems here because if we are planning in contingent space, we have our nice straight line that we are following in contingent space. We convert to joint space – it might turn out that it is impossible to do that in joint space, and we'll see some examples right now of this kind of problems.

So both spaces have plusses and minuses. In reality, you usually use some sort of a hybrid approach to limit the computation, but those are to make sure that you're not colliding with obstacles along the way.

Any questions? If you have, just ask before I forget the answer.

So let's look at the planning difficulties. We have a robot. We have a starting point A. We have a target point B. It's a relatively simple to link robot. This is the reachable space – right – of the manipulator. When you stretch both links, you're traversing the outside circle. When you collapse one into the other, you're traversing the inside. So the gray shaded area is the reachable space for this robot.

So we have two points: initial A, go point B. They're both reachable for this manipulator. Right? They are in the reachable space. Now, if we plan a straight line in contingent space, you see that it goes through a space where you cannot reach. This point, C, is not reachable. So the intermediate point is not reachable. We wouldn't know that until we actually start doing this computations along the path and suddenly find out that we are running into a singularity. Okay?

So that's one type of problem.

Let's say they're all reachable. Okay? We have starting point A, go point, B. Everything along this path is reachable. Okay? The singularity is right there in the middle. Well, as

we approach that singular position, your joint velocities go to infinity, and obviously, you won't be able to follow this straight path. It will cause a deviation.

Again, we wouldn't be able to know that in advance if we plan in contingent space until we actually reach that point.

So here is another example in which both points are reachable. Everything along the path is reachable without infinite velocities. However, the two solutions are actually different – reachable in two different joint space areas. So we cannot go from point A to point B in a continuous motion along that path because point A is reachable from the left configuration. If you want, point B is reachable from the right configuration. They are not intersecting in the middle. Okay?

So this is the type of problems that we can encounter when we are planning in contingent space only.

So far, we kind of set up the problem and see what kind of difficulties there can be. Now let's put some formulas down on how do we actually plan?

We'll assume one generic verbal U , and – not me, you. So it's $X - U$ can be XYZ if you're doing the contingent coordinates. It can be $\alpha \beta \gamma$ if you're using the rational cosines. It can be t_1, t_2, t_6 – if you have a six degrees of freedom manipulator with joint angles. So we'll use that generic variable to denote any of those. And throughout the entire lecture here, we'll be using that – U – as the common variable. Just think about it that when you do the actual planning, you will substitute U for X , for Y , for Z , and then plan for all of those independently.

So – okay. We wanna go from one point to another point. Any space – one variable. What's the simplest way? Straight line. Right? You wanna go from here to there along the straight line. That's my simplest trajectory. The problem with the straight plan is that we have discontinuities in velocity at the path points. Right? Because a straight line only gives you, basically, two parameters, and you're not in control of how fast you go or acceleration. There isn't such –

So here is an example. You wanna go from point A to point D via point Bs and Cs. So A is the initial point. D is the target point. And B and C are intermediate points. All right?

So the simplest trajectory is we go from A to B, B to C, C to D along straight lines. And as we said, we'll see in the formulas in more detail, but basically, if we plan a straight line from A to B, we can't guarantee that the straight line from B to C will have the same velocity at point B as it was ending the velocity of the previous segment. Okay?

So it's going to be a discontinuous, jerky motion. If you are going – if you are starting and stopping in the intermediate one, and then you're starting from the intermediate, and then stopping in the next intermediate, that's fine. But usually, those intermediate points are introduced there so that we don't collide with obstacles, or we can achieve certain

tasks in the middle. They're not necessary to stop at them, and we probably don't want to stop at them because we are wasting energy.

So we wanna kinda go continuously from A to D, avoiding those obstacles on the side – going as close as possible to B and C. That's usually the goal.

So what do we do? We do straight lines with bends in those intermediate points. So we start – usually, again, we have time to accelerate on the robot. You just start from scratch. So we have a small bend, then we maintain a straight velocity for a while. Then we get a curve from B to maintain the continuous velocity. Then a straight line, then a curve, a straight line, and then we decelerate and stop gracefully at the end. Okay?

So you can think if you want about this vehicle that we sell – if it's planning a path the same kind of way.

So that's the next level of planning. And then, of course, another way to do it is instead of using straight lines, we can actually do a cubic polynomial. So the obvious power point here is – it almost looks like a straight line with bends, but think of it as a cubic polynomial. Right?

So you actually having a higher degree of freedom curve between each of the points along the path. Okay? That will be slightly more complicated from a formula point of view.

So everybody is following? It's pretty simple, but –

And then finally, if you have a lot of constraints that you want to satisfy along the way, you might want to use a higher order of polynomials, like quintics or septic or whatever because in this case – in the cubic polynomial case – we have a cubic polynomial, as we'll see in a moment, has four parameters. So you limit it on how many constraints you can satisfy. Say you're starting from certain position. You're ending at certain position. You're starting with velocity zero. You're ending with some velocity. That's about it.

If you wanna control acceleration, deceleration, things like that, you need higher degree polynomial because you need more coefficients to satisfy that motion. And we'll see that more in detail.

But of course, the higher degree of polynomials you use, the formulas get more complicated. Usually, we try to get away with the simplest things we can. Okay so far?

And that, again, is the planning for each of the degrees of freedom, each of the positions, each of the angles – for each of them, you can do that kind of planning independently.

So let's look at the actual formulas. If it is a single cubic polynomial, a general equation for that would be – of course, that's in as a function of time – would be A_0 plus A_1T plus A_2T^2 plus A_3T^3 . So as we said, we have four parameters here, which

can – we can use those parameters to satisfy certain conditions for the motion. Typically, what we'll have is we'll have as initial conditions some starting point and some ending point. Right? Those things will be given. Where do we start the motion? And what is the position for each of the intermediate points and the goal point where we want to go?

So the easiest two conditions. Right? So at time zero, U of 0 is U_0 , which basically immediately gives us the value for A sub 0. And then at time T sub F , which is the duration of this particular interval, we have some value U sub F . And that will give us one equation for the remaining three unknowns: A_1 , A_2 and A_3 . Okay? And then we can have more conditions. For example, for the velocity – this is a graph of the velocity of that function. Now, the velocity has only A_1 , A_2 and A_3 . That's just the derivative with respect to time. And again, as initial conditions – let's say that we want to start at velocity 0 – so start at rest, finish at rest. We'll probably not be finishing at rest in the intermediate points, but this is the simplest case. So U dot at zero is zero. U dot at TF is zero. So that immediately gives us a value for A_1 , which will be zero if U dot at zero is zero, and then another equation for A_2 and A_3 . So now, we have two equations for A_2 and A_3 : one that will come from U dot of Z sub F and one that will come from U of T sub F . All right?

So two linear equations, two unknowns – that's the beauty about working with polynomials is that everything is linear. Right? So we can find the solution, and as far as the acceleration is concerned, we are post. Basically, it's fixed. We can't control that. Whatever it is that it's – it will come from the formulas. Right? So that's why I was saying that if you want to control the acceleration, then you need a higher degree of polynomial to give you more parameters to work with. So basically, here is the solution. I'm not even gonna spend the time to derive it right now, but it's pretty simple – two linear equations with two unknowns: A_2 and A_3 . These are the values of A_2 and A_3 . A_1 we said is zero because the velocity at the beginning is zero, and A_0 is using zero because that's the position at time zero. So as you can see, the trajectory depends on the initial position, go position and the time that we want to reverse that segment. Okay? Pretty simple so far. Now, if we are using intermediate points, then what we want to do is that at the intermediate points in the middle, we don't want to stop. So we don't want the velocity there to be zero. Right? So for continuous motion with no stops, we need velocities at the intermediate points. So at time zero, the velocity will be some value – U sub 0 dot – and at time T will be some other value. Let's assume for a second that those are given. We'll see later how to deal with that. So those will be added to the initial conditions. We'll have the position in the beginning, the position at the end, the velocity at the beginning, the velocity at the end. Four conditions, four parameters – A_0 , A_1 , A_2 , A_3 – again, linear equation for – yes.

Student: why do you say acceleration as a constant – because if you differentiate the original equation twice, you still get the – [Crosstalk]

Instructor (Krasimir Kolarov): I'm sorry. It's not a constant [inaudible] show a line. Right. So – but it is fixed in terms of the value of the acceleration. It's fixed because we don't – it's a function of time, but the parameter we cannot control. We cannot – it would be a fixed line, so to say. Right.

Student:[Inaudible] six, eight, three times T.

Instructor (Krasimir Kolarov):I think I probably had it at a particular – oh, you’re right. Yeah. No. No, no, no, no, no. Hold on a sec. U dot – no, that’s the third derivative. Right? This is – yeah. The third derivative is six, eight, three. Here – this is the acceleration – U double dot. Right? It’s a straight line. What I meant is that the values A2 and A3 have already been computed using the conditions that we had before. Right? Because we had four conditions, four unknowns. We are computing it. So we really don’t have a control over that. So at every time, it will be fixed. We don’t have extra conditions that allow us to control the acceleration. So there is no control of the acceleration. It’s fixed in a sense that it comes out whatever it is going to be based on the other computation. If we want to control the acceleration – so have certain variables that we can introduce there – then we need a higher degree of polynomial to use for conditions for that. All right? So I’m sorry if I misspoke. I didn’t mean to be fixed in terms of a value. I also showed you that the curve is obviously not fixed like its line. But the numbers that define that line are fixed. We cannot compute them based on goal configurations. Like, I cannot say, “I want the acceleration at point T sub 0 to be a certain value.”

It will be basically the two times A sub 2, which will be determined from before. So I have no control over that.

Student:Why is acceleration going down? Why is the slope negative? [Crosstalk]

Instructor (Krasimir Kolarov):This particular curve is based on the conditions that were put in that particular curve. It doesn’t have to be that way. Right? It depends on what the numbers actually come with. That’s for that particular curve because we were starting in [inaudible] conditions.

Okay. So let’s see. Where – we were here. So we have different initial conditions. We have certain values. And now, obviously, the formulas are going to be different. A sub 0 is still U sub 0 because we start at – from T0, we have this U sub 0 is the initial point. A sub 1, now, is going to be U sub 0 dot because that’s the condition. And then for A2 and A3, we again have two equations with two unknowns. And they will be function – this time, not only of the positions and the time, but also of the target velocities. Okay?

So if we know the target positions – U sub 0, U sub F – the target velocities – U sub 0, U sub F dot – and the target duration for that segment, we can compute the trajectory using those conditions. Okay?

Now how to find those. Well obviously, if we know the actual velocity and angular velocity for the robot that we want to have for the actual coordinator, then we can use the inverse of the Jacobian to find the U dots in the contingent space. All right?

So if we say, “Okay. We want each of the links of the manipulator to have certain velocity,” – because when you have an industrial robot, they are all targeted for certain

velocities and certain angular velocities that they're the characteristic of the mechanical structure. You put those in the conditions. You can calculate some target velocities.

In other ways, you can have the system choose some reasonable velocities using heuristics. For example, if we have several links that we want to traverse in the trajectory – in addition of having continuous velocity, maybe we want to have some averages on each side of the motion so that you can have landed continuous motion with the least amount of energy dissipated.

So you can put this kind of heuristics – whatever is important for your type of motion that you're planning for – and have the system automatically compute the velocities.

And then finally, you can actually introduce additional constraints. So for example, we have U_1 is the velocity for the first – \dot{U}_1 is the velocity for the first segment. \dot{U}_2 is the velocity for the second segment. So we wanna make sure that at the Y point where the two segments meet, we have continuous velocity. So then $\dot{U}_1 - \dot{U}_2$ at T sub F will be zero. All right? Because it's the same velocity, and we probably want the same acceleration as well. So then here is the second derivative comes in as an additional condition and add that to the system. Okay?

Obviously, if we do that, then we have to take some other constraints out because we might have too many constraints. If we have four for each of the segments, then we don't have enough constraints to satisfy. We don't have enough parameters to satisfy all the constraints.

So these are the type of reasoning that you can use to compute those velocities. Typically, you will not be given the velocities. You'll just want to go from one point to that point to that point in some time frame – and the time might not be even given, as well. You just might want to use the time by heuristic – like how fast do we want to do? As fast as we can do it without actually going over the speed limit for that particular motion for the link.

Okay. So this is – so far, this is cubic [inaudible] interpolation. Right? Any questions? No?

So we'll move to linear now. So linear interpolation – straight line. Right? We have starting times T_0 , go time T sub F, position at the beginning U_0 , position at the end U_F , and that's it. There is no more conditions that we can satisfy in this case because we have two parameters: A_0 , A_1 . And we have two conditions, and that uniquely defines our motion. Right? That was the problem is that if we take an acceleration here, it would just be A_1 . Whatever is the value that comes in from those two conditions – and we cannot control it.

So we have discontinuous velocity.

Now we're going to introduce this parabolic blend that we were talking about. So we have the T_{0} is the starting time. The T_{F} is the ending time. And then we have this blend. And the blend – the first blend occurs at time T_{B} , and the next blend is at $T_{F} - T_{B}$. So we'll be assuming that the length of each of the blends are the same for simplicity. Why make our life more difficult?

So in this case, the equation for the parabolic blend itself is $U_{T} = \frac{1}{2} AT^2$. So we have one parameter, which is A . And we want to introduce some conditions for this parameter. So that will give us one more condition that we can satisfy, which in our case is velocity. We wanna make sure that the velocity's continuous throughout the motion.

So the velocity here is simply A times T . And if we put a condition for constant acceleration, for example, then that will give us that value A . Or, in that case, basically we'll have $U_{T} = \frac{1}{2} U'' T^2$, where U'' is the acceleration. Okay?

And that acceleration – we'll see in a moment how we can determine for continuous motion.

So following so far?

So another – so if we wanna have continuous velocity, then basically, we can calculate the time for the blend from a condition for a continuous velocity. So we want this function at U_{F} , U_{0} and U'' will give us the value for the blend for that particular region that achieves continuous velocity around. Okay?

If you wanna see the actual equations, they're in the book. They're a little bit more complicated, but it's basically a second degree of freedom polynomial.

Where T here is the duration of the entire motion, from T_{0} to T_{F} . So we basically have here the equation for the motion of using straight line with parabolic blends in a continuous fashion from time T_{0} to time T_{F} . Everything here on the right side is given, and it's function of things that are given: T_{0} , T_{F} , U_{0} , U_{F} – and we will see U'' – the acceleration – is not given right away, but we'll see how to compute it very easily.

Yes?

Student:[Inaudible] plus T_{B} ?

Instructor (Krasimir Kolarov):It's $T_{F} - T_{B}$ – it makes it – I'm sorry.

Student:[Inaudible]. [Crosstalk]

Instructor (Krasimir Kolarov):That's T_{0} . That's T_{B} . We wanna make sure that – oh. You're right. It should be. Interesting. We wanna make sure that the time of the blend

is the same. Right? Which is $T_{sub B}$, so that should really be $T_{sub 0}$ plus $T_{sub B}$ because you want that to be the location of the blend. Okay. Let me move over that. [Inaudible] right now, but I mean, it makes sense. The idea is that the blends have the same amount of time. Right? Okay.

I guess people are assuming that $T_{sub 0}$ is zero, and that's why it's $T_{sub B}$, but it doesn't have to be. So from that point of view, that's right. And I think that these formulas are computed with that in mind. Let me just check. Okay. We'll check on that and get back to you.

Okay. Any other questions? I'm glad you guys are paying attention. That's good.

So now, if we have several segments, things can get a little bit more hairy. Let's say we have the positions of the different points that we wanna go through. We have the slope of the different linear blends – of the different linear portions, which will basically give us the velocity. Then we have the time directions, which are, in this particular case, I to J is a segment. So a typical segment is from $T_{sub I}$ to $T_{sub J}$, and the duration of the entire segment will be T_{DIJ} . And then the duration of the next segment is $T_{sub DJK}$ etcetera. Then we have the duration of the actual blends will be denoted as $T_{sub K}$ for each of the blend. We have the slopes, and then the duration of the straight-line segments will be denoted with $T_{sub JK}$, which is the straight line between position J and position K. Okay?

So these are the parameters that we're introducing here. So then we have $T_{sub I}$ is the first blend. Then $T_{sub IJ}$ is the straight-line segment. $T_{sub J}$ is the next blend, etcetera for all of them. Okay?

The slopes we already denoted with $U_{dot IJ}$, JK , KL , LM . So what is given here – we'll come back to the picture. Actually, let's go back and look at that. What is given is the positions. $U_{sub I}$, $U_{sub J}$, $U_{sub K}$, $U_{sub L}$ – those are the points that we want to go through. Right? The initial of that final point and the intermediate points – that's one of the things that is given. Then the next thing is the desired time durations for the entire trajectory from one point to the other. So this whole thing – it is the only thing that is given. We're not going to be – we'll calculate all the others. We just want to have this blend, like, linear section with the blend for the entire portion, for the next portion, etcetera. So those $T_{desired IJ}$, $T_{desired JK}$, etcetera – those are the things that are given.

The T_{Is} , the T_{IJs} , etcetera will compute those. Okay?

Student: You don't actually go through the points – your desired positions –

Instructor (Krasimir Kolarov): Okay. Hold on to that. We'll address that very good point. Yeah.

So desired time duration, and then the other thing that is given or can be computed is the magnitude of the acceleration. So usually, there's a certain limit for the particular joint

that we are using – say, you can't go faster than that or with faster acceleration than that – and that would be your number.

So those here – this is not denoted here just because it's not the graphic term – but the magnitude of acceleration is given, as well. So using those, we want to compute the blend times – so how long each of the blends are, the straight segment times, the velocities, the signed accelerations because here, we just had the magnitude of the acceleration, but we don't know whether we are accelerating or deceleration, and that will depend on the motion. And that's basically it.

So those formulas are given in the book, in the notes, but we'll also look at them here briefly. One note is that the system usually calculates or uses default values for the acceleration based on the particular robot – based on the mechanical structure, how fast you want to drive it, what's the work space, etcetera. And also the system can calculate the desired time durations based on default velocities. All right? Pretty simple.

So here's the formulas. They will be different, clearly, for the first segment and the last segment than the intermediate ones because if you remember the picture – that's the [inaudible]. Okay. If you remember the picture, we're starting here with a full blend in the beginning, and we are ending with a full blend so that we can sort of accelerate and decelerate smoothly. And then in the middle, we have kind of half blends for each of the segments. So the formulas will be slightly different for the different ones. But it's all computed based on the conditions that we had. So for the first segment, we are given $U_{sub 1}$, $U_{sub 2}$, and then we're given the magnitude of the acceleration, or they system has computed that. So then, we can compute the actual acceleration based on the sign of the difference between the positions, whether it's accelerating or decelerating. So once we compute that, and then we know the duration of the entire blend part – not the blend part, but the entire segment, then we can compute $T_{sub 1}$ using that. Then we can compute the velocity for that segment – $U_{dot 1,2}$. And then we can compute the time for the linear part of the blend – of that part. Then moving on to the inside segment, again, we can find the velocities just simple position over time. Then we can find the signed acceleration the same way as for the first one. We can find the time for the linear blend by using the velocities that we found and the acceleration. And then finally, we can find the time for the straight-line segment by just subtracting from the whole time for the segment the times from the blends. And as you see here, the blends here are half-half on each side. And then when we get to the end, similarly to the first one, we find the actual acceleration – the signed acceleration. We find the time for the last blend. We find the velocity. And finally, we find the time for the last straight line segment. Okay? This is – it kind of looks hairy, but it's very simple formulas, very simple derivation. Basically second-degree equations for the times up there and similar linear equations for the velocities and the accelerations. So using those sets of formulas, we can go from the beginning to the end and compute the trajectory. And I don't know exactly what kind of homework you guys are getting, but you might actually have to do that for a project so that you can understand how it works. Okay? So, so far fine? Yeah? Yes.

Student: I have a question [inaudible].

Instructor (Krasimir Kolarov): Yes, I'm coming to it right now – about not going through the point, right? Okay. Here we go.

So what you actually see here is you're not going through the actual point. Right? You're going around them. Now remember that we introduce this [inaudible] point. The main reason, really, to have these [inaudible] points is when we're planning a motion for a robot with obstacles, we want to introduce this kind of intermediate points to make sure that we go around the obstacles. So it's really not that important that we go through the exact points unless we wanna force it, and we'll see right now that we can force it. But in principle, the [inaudible] points is just to make sure that we avoid certain spaces in the workspace. Okay? So it's not that important, but if we do want to go through them, then we have several mechanisms. We can introduce [inaudible] points. So here is the original [inaudible] point that we want to go to. We can introduce on both sides on a small distance to [inaudible] points and do the planning for that. And then the straight line will go through the [inaudible] point if we plan it the right way. Okay? If they're close enough. We can also double – okay. The other thing is we can use this efficiently high acceleration to actually force it through a particular point. Or if we want to stop them, we can simply repeat the [inaudible] point, and then we'll have – we'll make sure that we go that – through that in particular point. That will obviously affect the formulas. We wanna make sure that we don't have division by zeros, etcetera. But they are mechanisms. The bottom line is that these [inaudible] points are really there so that we have a general motion around the space that is avoiding obstacles. Okay. Now these were the two mechanisms. We can use cubic polynomials or straight lines with parabolic blends. As we said, if we want to satisfy more conditions, then we can use higher polynomials. For example, let's say we are given two positions, two velocities and two accelerations for that particular segment. Right? Now we have six conditions basically, right? So we can use a quintic. We have fifth degree polynomial, six parameters. Plug in those conditions there. Six equations, six unknowns – because some of them are relatively simple: $U_{sub\ 0}$, $U_{sub\ F}$ – will be taking parameters down. So it's not going to be that complicated if we use linear equation for that. Okay? The formulas are actually in the book if you're really curious to see what they look like. Now, another thing is we can use different functions. We were using polynomials because they result in a linear equation, so it's relatively simple to solve. If you want to – if you feel particularly challenged that particular day – you can use exponential functions, trigonometric functions – whatever you want to plan the trajectory in the space. Yeah?

Student: What book are you talking about? That's not in the notes, right?

Instructor (Krasimir Kolarov): Oh. Yeah. The Craig book – the recommended book. It might not be in the notes, you're right. Yeah, it's in Craig's book. Sorry. They're not in the notes. But they are available, and I think that 718 is actually – refers to Craig's book. I don't think you should worry too much. I don't think you'll be getting that on a test or anything like that. But –

Okay. So, so far – well, let's stop for a second and see. So far, we've basically looked at different mechanisms to plan past given conditions for the trajectory. Okay? And it's

been very theoretical, right? These are just math formulas of what you can do, which is the [inaudible] that will be good to know.

Now, what do you do when you're actually doing the planning for the robot? So run temp path generation – so basically, we need to feed something to the control system to tell the different joint positions, velocities, etcetera for the different joints of the robot. Right? So $tata$, here, stands for $tata_1$, $tata_2$, $tata_3$, $tata_6$ – depending on whether it's six degrees of freedom [inaudible] joints or [inaudible] – whichever. It's a generic terms. And those are the actual values that we feed to the robot.

So what do we do? The path generator computes the path at some update time. Right? At some update rate. So we saw the beginning point, intermediate points – we can compute all those values using the formulas that we saw. If we are planning in joint space directly – right – let's say we're using cubic splines. We can change the set of the coefficients at the end of each segment and feed that to the control system. Right?

So we start with a certain set of coefficients. We feed it to control the robot is moving. When we approach the other one, say, "Okay. Time this. Feed those numbers. Time that. Feed those numbers," etcetera, and we get that cubic splines.

If we're using linear with a parabolic blend, then we have to check at each date whether we are in the linear portion or in the blend portion because we have different formulas for the different – and depending on where we are, we feed those kind of values. Right?

We're saying here updates. Basically, a certain frequency that you are updating the control system. You compute the points, and you feed them. Right? Are you following so far? Okay?

So the cubic splines – we have those particular [inaudible] points. That's the linear parabolic – we have to figure out which part of the formulas we are. It's not a big deal. We have the formulas. It's simple computation.

The problem, of course, is that we're not following a particular trajectory. We're just moving kind of continuously into space. If we're doing the planning in contingent space, then we calculate the contingent position and orientation at each update point using the same formulas. Then, we have to calculate the joint space coordinates using either inverse Jacobian and derivatives or the find the equivalent frame representation, and then use the inverse kinematics functions to do $tata$, $tata \dot{}$ and $tata \ddot{}$. And you should know how to do that now from the kinematics that you've done so far. Right?

So this is how we can compute all that. On top of that, you have to remember that this is typically what we saw so far is just for one parameter, so we have to be careful to make sure that the motion is continuous if we're planning for all three of the parameters that the times are the same. So things get more complicated when you're trying to build a full system, but the underlying technology is what it is here. Okay?

So that's so much about the trajectory planning and different parameters and how we do that computation. Now, if you don't have any questions, what we can talk a little bit about obstacles.

So if we have – and there is a whole course on motion planning – at least there was when I was at school – John Claude Lapel was teaching it. I assume that he is still doing that. It's a very, very cool course. That's one of the – I did my thesis on that essay, so it's a lot of fun.

So in that case, basically, there is several considerations that we need to keep in mind. If we – let's say we have a six degrees of freedom puma arm. Right? So the question, then, is do you plan a path for the whole manipulator? When you're doing that, are you dealing with the global motion in the space or the locomotion where just the end factor is? So typically, you do some sort of a combination between a global and a locomotion planning. You're using global motion planning when you're moving from a relatively empty space, and you know you're not gonna be hitting obstacles. And then when you get to the place where it's more crowded with obstacles, then you moved to a lock or more precise motion planning toward the end factor only.

So that's one type of planning. Another one is configuration space approach, and I'll show you a few slides on that. In fact, let's switch to that. Let's see. What do I have? Okay. Let's switch to that. Okay.

So let's say we have an environment with obstacles here. Okay? And we have a point robot – a small, circular point robot that we wanna move around this environment. Okay? And that robot actually has certain – this circle has a radius. Okay? It's not just a point, but it's – it has a substance. So what we can do is we can plan in configuration space. Basically, we take the obstacles, and we grow them with the size of the point robot if you want. Okay? In which case, if you look at the [inaudible] up there, the idea is that if we plan a path for the red dot that doesn't collide with this grown obstacles, then we know for sure that the circle is not going to collide with the smaller target obstacles. Okay?

And so then we just have planning for path for one point in that space, which can be done many different ways geometrically. These grown obstacles are called C space – configuration space obstacles, and this is a C-space planning approach. And what we can do is we can put the grate around it and then plan the path of the point around this grate so that it doesn't collide with the grown obstacles. And then when we get back to the circular robot, it's not going to collide with the obstacles itself. Okay?

If we have a several degrees of freedom robot or we start adding orientation, then this configuration space obstacles can be not only planer, which is here, but then you get the three-dimensional obstacle because you're thinking about the orientation that you are approaching with. Or you can get many dimensional obstacles if you have many degrees of freedom.

So in its more generic form, that becomes planning for N-dimensional path in an N-dimensional space. Okay. And then you can start talking about kind of high math there. Let's go back.

Okay. So we have C-space, planning for a point robot. So you can put the graph representation of the free space, build a [inaudible] and know which part of the free space you are, and then path a plan – plan a path, and you can use things like the artificial potential methods to tell you, like, as you go closer to an obstacle, you can have a force that is repulsing you from the obstacle. And as you have to go, you have a force that is attracting you to the goal. And based on that, you can build an artificial potential fill – which is, by the way, what Osama did way back on his – it was very revolutionary work at that time – and then use those. And you get in all kind of interesting things of getting into [inaudible] minima, global minima, maximas, etcetera. So it's a fun thing.

And then to add on top of that, you can have multiple robots moving in the same environment. This is very much similar to the video that we saw in the beginning. Like, if you have a planning path for autonomous vehicle, if it's the only vehicle that is moving on the streets, then it will be the previous approach. If you have several vehicles, then you have certainly multiple robots, so you have to use those other robots as detractors, and then there will be a repulsing force from them. You can have moving obstacles. You can have moving robots. Things kind of get interesting. All right?

But that's not going to be covered here. If you are interested in that, check out the motion planning area course.

I think I'm about done unless you guys have any questions. I think the TA had some announcements to make. First, do you have any questions on the lecture?

Student: Is there any way that you can post that on the website?

Instructor (Krasimir Kolarov): The web?

Student: The slide show?

Instructor (Krasimir Kolarov): Well, yeah. I think that shouldn't be a problem. I'll talk to –

Student: [Inaudible].

Instructor (Krasimir Kolarov): But most of the – all this material should be in the notes, so – but we can certainly do so.

Student: I think most of it is in the notes, but if there's stuff that's not.

Instructor (Krasimir Kolarov): Okay. Let me give you the – it doesn't want to go. It's stuck. TA:

All right. A couple of announcements. First of all, the next homework is due on Monday at 5:00 p.m. So you can either turn it in to class on Monday or just put it in the box.

And the next thing is we're gonna pass out the exams and solutions right now. I'm gonna put them on this desk, and I think they're ordered in some way.

Student:Solutions will be right here. TA:

Solutions are right there. I'm gonna put – try to divide into three, so A through H, yours are gonna be right here. J through P – yeah, J through P –

Instructor (Krasimir Kolarov):So it is [inaudible] for you to actually get your exams back. If you stay until the end. TA:

To the end.

Student:Do you remember what the average was? 80? TA:

Um, I think it was.

[End of Audio]

Duration: 66 minutes