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 join 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 tatas – tata 1, tata 2, tata 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 discontinues 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 septics 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 A0 plus A1T plus A2T squared plus A3T to the third. 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 U0, 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: A1, A2 and A3. 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 A1, A2 and A3. 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 A1, which will be zero if U dot at zero is zero, and then another equation for A2 and A3. So now, we have two equations for A2 and A3: 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: A2 and A3. These are the values of A2 and A3. A1 we said is zero because the velocity at the beginning is zero, and A0 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' 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 – A0, A1, A2, A3 – 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 U1 is the velocity for the first – U1 dot is the velocity for the first segment. U2 dot 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 U dot 1 – U1 dot at T sub F will be U2 dot at 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 T0, go time T sub F, position at the beginning U sub 0, position at the end U sub F, and that's it. There is no more conditions that we can satisfy in this case because we have two parameters: A sub 0, A sub 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 sub 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 sub 0 is the starting time. The T sub F is the ending time. And then we have this blend. And the blend – the first blend occurs at time T sub B, and the next blend is at T sub F minus T sub 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 sub T is ½ AT squared. 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 sub T is ½ U double dot T squared, where U double dot 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 sub F, U sub 0 and U double dot 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 sub 0 to T sub 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 sub 0 to time T sub F. Everything here on the right side is given, and it's function of things that are given: T sub 0, T sub F, U sub 0, U sub F – and we will see U double dot – the acceleration – is not given right away, but we'll see how to compute it very easily.

Yes?

Student:[Inaudible] plus T sub B?

Instructor (Krasimir Kolarov):It's T sub F minus T sub B – it makes it – I'm sorry.

Student:[Inaudible]. [Crosstalk]

Instructor (Krasimir Kolarov):That's T0. That's T sub 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 TDIJ. 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 on 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 TIs, the TIJs, 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 that 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 sined 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, tata2, tata3, tata6 – 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 double dot. 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 tells 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

IntroductionToRobotics-Lecture11

Instructor (Oussama Khatib):All right. Let’s get started. So the video segment today is quite interesting. It was presented at the 2000 International Conference on Robotic and Automation, and I’m sure you’re going to like it. [Video]

A robotic reconnaissance and surveillance team, U.S.A.

A heterogeneous multi-robot system for surveillance and exploration tasks. At the first tier of this team is the Scout. Scout’s are small, mobile sensor platforms used in a cooperating group.

At the second tier is the Ranger. Rangers are larger robots used to transport, deploy, and coordinate the Scouts. Scouts are wholly original robots with cylindrical bodies 40 millimeters in diameter and 110 millimeters in length. The Scout carries a sensor payload used to relay environmental information to other robots. The most common Scout payload is a small video camera, but other payloads, such as microphones, are also used. Video data is broadcast to other systems via an analog RF transmitter. Scouts communicate with other robots using an RF data link.

One specialized Scout has a camera mounted in a custom-pan tilt unit allowing the robot to view its surroundings independently of the orientation of its body. The Scout has two modes of locomotion to allow it to navigate different kinds of terrain and obstacles.

The first mode uses its wheels, allowing it to drive over smooth surfaces. Here, the Scout demonstrates its ability to climb a 20-degree slope. The second mode of locomotion is the hop. The hop is accomplished by winching the Scout’s spring foot up around its body and then releasing it suddenly. Here, the Scout jumps over an obstacle.

Scouts are deployed by Rangers. The Ranger is a modified commercial all-terrain robot. The Ranger uses a launcher to deploy Scouts into the area in which they will operate.

A Ranger can carry and shoot up to 10 Scouts from its launcher. Ranger’s supervise the Scouts while working with other Ranger’s; Ranger’s report to a human group leader. The Scout’s are designed to withstand the impact of landing, and of being shot into and through obstacles, such as these simulated windows.

The Scout’s small size, it’s deployability through launching, and its multiple locomotion modes and sensor payloads, give it the ability to explore difficult-to-reach areas and report useful data.

Combining the Scouts with Rangers, which provide the ability to travel longer distances and to have greater computational resources, forms a useful reconnaissance and surveillance team.

Instructor (Oussama Khatib):Okay. What do you think? I guess we need a robot to do the – I mean to [inaudible] these devices, and so one more robot is still needed. Okay. So after completing the forward kinematics, after finishing the Jacobian, we are ready now for dynamics; are you ready? All right. So while dynamics, and then we will do the control and that's it, you have the basics. Kinematics, dynamics and control. Well, here is an example of robots that involves a lot of dynamics. Just imagine, like, moving the hand little bit, you can see all these coupling forces coming on the other hands, on the body. As you start moving, you have all these articulated body dynamics that are going to appear. And the dynamics of this system is quite complicated. In fact, if we go to this problem, we find that we need really to understand the dynamics of just one rigid body, and then combines these different dynamics together to understand the articulated multi-body system. So to do that, that is to find the dynamics of an articulated multi-body system; there are several formulations. In fact, there are many, many formulations. We will examine two of them. One is the Newton-Euler formulation. Have you heard about Newton? Yes. So what does it say to you, Newton-Euler, what does it tell you? So Newton Law is? You were saying?

Student:[Inaudible]

Instructor (Oussama Khatib):So mass acceleration equal to the force applied from a rigid body, right? So that is if you apply a force to a particle, it will accelerate along the same direction with an acceleration that is equal to the force divided by the mass.

Okay. So what about Euler? What does Euler do with dynamics here? You know, Euler angle, you know Euler parameters. Huh. So Euler was looking at angles, why? Angles measure what? Rotational motion. So linear motion – because force, acceleration of a mass, a particle, it's just going to be a linear dynamics. And Euler is dealing with the other side of dynamics, rotational motion. Now, if you have a particle, then it's really not rotational motion to talk about. So we go to the rigid body and we find that we need to address the problem of angular rotation, angular motion, and that is the formulation – the combination of Newton and Euler equations extended to the problem of multi-body.

So we will examine articulated multi-body dynamics, and we will find, similarly to the way, if you remember we found the Jacobian, by analyzing the static forces, propagation. You remember we break all the joints, remove the joints, and look at the stability of issue of the rigid-bodies. We are going to do the same thing with dynamics. Then we will examine another formulation – a formulation that captures the whole dynamics, linear and angular, in one equation, that is the Lagrange equation. And this formulation is relying on the energy that is the kinetic energy of the system. You know what is a kinetic energy? Most of you. What is the kinetic energy associated with a particle moving at a velocity v?

Let's see. One-half mv squared. Very good. And also, the potential energy. And that will lead us to a very interesting form that will give us the dynamic of articulated-body system in an explicit form. You remember how we did the explicit form for the Jacobian. We can find the Jacobian as a sum of contribution of the different velocities of the different lengths. Well, we're going to do the same. We're going to find the dynamics of the whole articulated-body system as a sum of the contribution to the mass properties, inertias, and masses. We establish something called the mass matrix associated with the dynamics. And we will see that from finding the energy – just finding the kinetic energy of the system of – or at least each of those lengths, adding them all together to find the total energy, we will be able to obtain the dynamic equation.

So this form is really important and we will examine this form, probably on Wednesday. But let me just to start, to give you an idea about what is happening when we look at the dynamics. This is a robot from France; it's called the MA Manipulator 23. It is a cable-driven robot, so all the motors are in the back and the cables are driving this traction.

So if we go and analyze the inertia view from just one axis, let's say, the first axis of rotation. So you have big inertia, smaller inertia, right? By putting the masses away from the axis, you are increasing the inertia perceived about this axis. So this inertia then depends on all the mass distribution, the lengths, the load, et cetera, that is associated with the manipulator.

If we go here, we have also changes in the inertia perceived, that it's independent of the previous lengths. So there is a structure to the way the inertia is affected by the motion of the structure and the configuration is going to change the value of that inertia that you are perceiving.

If you want, I can show you the equations. I don't know if you can see them, but here we go. So the first inertia perceived from this joint is, like, half the page. It's sine, cosine, and all of these things, and depending on – but, obviously, I mean, we can obtain these equations, it's not a problem. The problem is to understand what the structure of this equation is and how we can find those properties and how we can understand them. What – when we analyze later – when we later analyze the explicit form, you will see that essentially we're going to be able to – to gain like with the Jacobian – are going to be able to see the dynamics of a manipulator just by looking at the structure of the robot.

All right. Here is another robot, this is – this was analyzing my thesis. This was a robot that can carry 80 kilograms of its weight; heavy robot. It's a robot that has 6 degrees of freedom, and we are performing just some [inaudible] motion on the different joints. On joint, I believe, joints five or joint four, we have, basically no motion – just letting the joint to be controlled to its zero position. So what do we see? We see that on the top, the lower joints on the robot, during this [inaudible] motion of the other joints, there is little effect, I mean, you can see some errors, but the errors are sort of filtered somehow. On the lower joints, four and five, you see large errors. And that is reflecting the fact that if you have a heavy joint, there are disturbances, the inertia, the – this big inertia of the – of the robot is going to play the role of a filter. It's going to somehow reject the disturbances, and we will see little effect on those joints because they are quite heavy and the inertia is just taking. I mean, think about a truck moving fast and you hit it with, I don't know, a fly? It's not going to – to be affected. But for the fly, it is really terrible. So – so this is the fly and that up there is the truck. So if you want, here is the equation.

We're going to establish this equation. The equation is a vector, so gamma is the – the torques applied to the robot. Sometimes we call it tho or gamma. What is g? What would it be? Some force affected dependent on the configuration.

Student:[Inaudible]

Instructor (Oussama Khatib):The gravity, yeah. G, like gravity. Very good. And it's dependent on the configuration. For instance, if we take this joint, and if we put a weight, you're – you're going to feel a torque, right? If I take this joint to here, what is the torque due to the gravity?

Student:[Inaudible]

Instructor (Oussama Khatib):[Inaudible] Very good. So you get it. That is, so if we go up, basically the gravity is going to act on the structure and not on the truck.

Q double [inaudible] is the acceleration and m is mass acceleration, so m is? Okay. Q double [inaudible] is the vector of accelerations and m – hmm? Very good.

Student:[Inaudible]

Instructor (Oussama Khatib):Very good. It's a matrix. So matrix plus multiplied by a vector will give you a vector. So mq double [inaudible] is going to be the inertial forces generated at zero velocity by the motion by the acceleration of the joints. So it's playing the role of a mass, so if I had one degree of freedom, you will understand it. Mass acceleration. M is a mass.

But because this is a multi-body system, it's going to be a mass matrix. And the mass matrix – if we had a robot that is prismatic, so three prismatic joints, essentially along the diagonal you will see the element you're presenting, the masses, the total masses reflected here, here, and at the last joint. And it will be diagonal matrix.

But for articulated body with revolute joints, you're going to have off-diagonal terms that represent the coupling. That is the motion of one joint will accelerate the other joints. What about v? This is a vector. That is function of q and q duct. So – oh, I forget my – I had something to, well, maybe they took the [inaudible]. So what would be this thing that will depend on the velocities? So – use the microphone. If I – do this.

I'm wondering why it's standing like this. If I stop moving it will fall, [inaudible]. So there is a force pulling here, it's called – what?

Student:[Inaudible]

Instructor (Oussama Khatib):Centrifugal force, yes. Now, if you have multiple – multiple-body that are moving in addition to centrifugal, you will have Coriolis forces. That is the product of velocities that will be involved will cause both centrifugal and

Coriolis forces. So q as I said is your [inaudible] coordinates. M is called the mass matrix or the kinetic energy matrix because this mass matrix is associated with the kinetic energy of the system.

If you're not just a particle m, what is the kinetic energy? For a mass m, a particle of mass m, what is the kinetic energy associated with this mass when it's moving at a velocity v? Hmm?

Student:[Inaudible]

Instructor (Oussama Khatib):One-half – one-half mv squared. Yeah, that's correct. Good. Well, it turned out, it is the same for a mass matrix multi-body; it's going to be one-half – how do you do mv squared for a mass associated with articulated bodies and the mass is a matrix?

Student:[Inaudible]

Instructor (Oussama Khatib):So you get a v transpose and m and v, which makes it quadratic form. And basically your kinetic energy is just one-half q duct transpose mq duct. These are the centrifugal Coriolis forces, and we will see actually that these forces disappear if the velocity was zero. Or if the mass matrix was constant, that is the v term solely depends on the velocity – product of velocity, and on the fact that all the element involved are derivative – partial derivative, coming from partial derivative of the mass matrix. So if you have a mass matrix that is constant, then the derivative are zero, and v will be zero. Yes?

Student:[Inaudible]

Instructor (Oussama Khatib):No, no. Mass matrix or kinetic energy matrix. And the gravity forces and gamma is the derived forces. So gamma one will be acting along or about axis one. So you have q one, gamma one, q two, gamma two, right? So gamma could be a force if the joint is prismatic.

Now, again, in doing all the formulations, it's very simple. We saw this figure before. If you have a rigid body, you can study that the stability of the – you can state the static equilibrium of the rigid body under the forces applied. And you say the sum of the forces should be zero if the rigid body was at static equilibrium. And the moment computed about any point going to be equal to zero, as well.

Now, if this rigid body is moving, so here the rigid body is at static equilibrium; we do that analysis. But if the rigid body was moving, then the rigid body, the masses, and the inertias are going to generate an additional force. We saw this force here; the first force, mq double [inaudible] plus v, actually these are inertial forces. They are created by the fact that this is a rigid body with mass. So if we can compute the forces and the moment associated with this rigid body – yes?

Student: Where do we have our [inaudible]?

Instructor (Oussama Khatib):Well, they will come – they will come little later. We can create the – those spring and dumping into control to stabilize and control the robot, or the robot might have some dumping at issue of the joint because of friction. But we will come to that later.

So if we – if we consider those forces, then we can restate the static equilibrium by saying, as it is moving, we should have a static equilibrium that will equal – will be equal not to zero, but rather to f and to n, to those movement and forces applied.

So then we can come up with a relationship. So first of all, for the Newton equation, you – we saw the equation earlier. The Newton equation is describing the linear motion. The Euler equation is describing the angular motion. So mass acceleration equal force; moment equal inertia time acceleration – angular acceleration – plus this term that creates the centrifugal Coriolis forces. And by stating that equilibrium and then doing the projection on each axis, you'll remember in the static analysis we projected on the axis to see the forces acting on that axis. So this essentially eliminate inertial forces – internal forces acting on the structure. Then we will be able to find their equations. To do that we will project [inaudible] in the static case and find those component that is the torques applied at issue of the joint axis. So this is basically the Newton-Euler formulation.

Now, the Lagrange formulation doesn't go into the specific motion of each of the joints, doesn't require any elimination; it works out the problem differently from energy analysis point of view. So essentially what we're going to do there, we take the whole articulated-body system and we take the kinetic energy of each of the legs. So the kinetic energy of [inaudible] is, let's say it's ki – the total kinetic energy of the system is the sum of all the k [inaudible]. And we have also the potential energy.

Now, once we decided about our system of [inaudible] coordinates, in this case it will be q's, q one to q n. We can write the kinetic energy in this form. Earlier, we said the kinetic energy of an articulated-body system would be just this q duct transpose and q duct. And this means that k – this k over k is half the velocity vector of transpose multiplied by the matrix, plus multiplied by the vector. That gives you a scalar. Now, if you take the scalar and combine it with the potential energy, you will be able to immediately find those equations. Now, once you find those equations, actually you realize, "I know the equations actually directly from m." By computing the kinetic energy, you identify m, which will go here. So you know m q double [inaudible]. From the potential energy, you knew – you know the gravity. What is left is v, and we will see that. Once we know m we can find v.

So the whole equation of the dynamics can be obtained simply by taking the potential energy, taking its gradient and that gives you g, the gravity vector. And by taking m and computing m, how can we compute m? Well, we can say the kinetic energy should be – if – if we have a system [inaudible] coordinate, it's a quadratic form [inaudible] velocity. So from the top we can say the kinetic energy could be computed for each rigid body, and by doing the identity between the first expression and this expression, we can identify m. We will see that on Wednesday.

So this is a slide we saw before that is the idea of breaking the structure and analyzing each of them, and then eliminating the internal forces. Well, this is exactly what we're going to do again, now, but by adding the inertial forces. So in addition, here, we are at static equilibrium; the rigid body is not moving. If it is moving, there will be these forces and then we can say the forces fi are equal to fi plus fi plus one, from this relation. The sum of the forces should be equal to the linear acceleration, and the sum of movements should be equal to the angular inertial forces. And this is the algorithm, basically, that we will find – this algorithm will allow you to compute the dynamics.

So the Newton-Euler algorithm does the following: it propagates velocities, we know we did that for the Jacobian. You propagated velocities. You – as you propagate velocities, you compute your accelerations and from the accelerations, you can compute the inertial forces as you're propagating. And then it has a backpropagation, that is the projection on the axis by taking these forces and starting from the end and going back, you propagated your forces and when you reach the ground, basically, then you have all of your tools.

So we're going to start with the basics and the reason for that, I want you to understand a very important equation related to the rigid body, and this is the Euler equation. We need to understand what is the inertial for a rigid body. If we were working just with a single particle of mass m, the problem would be very simple. We don't need Euler equation. We are working with rigid body; when they move, this rigid body has a mass distribution. And we need to capture the mass distribution at some point, and that result into this inertial tensor that we are going to use to discard the rotational motion of a rigid body.

Now, this might be scary, but it's going to be really, really simple if you just pay attention. We will start from a particle. We take a rigid body and look at it as a collection of particles, and then we will look at the linear velocities of each of those particles and they move, and we group them together and we will find the inertial associated with the rigid body. Okay? Sounds good. All right. Let's try.

So as I said, for a particle m, if we apply a force there will be an acceleration that is equal to f divided by this mass. So the mass is resisting to accelerate. So if the mass was infinite, no motion, right? The lighter, the faster with the smaller forces. So this is the law I think everyone is familiar with. There is a velocity, actually, with this acceleration that is not aligned with the acceleration depending on how the trajectory is going. And we can think about this same equation in the following way. So I'm going to group the mass and the acceleration using the derivative of the velocity. So you can write the same equation in this form – the partial derivative of mv is equal to the force. What is mv, by the way? I'm sorry?

Student: Momentum.

Instructor (Oussama Khatib):Exactly. It is the momentum associated with this linear motion. It's the linear momentum – mv is the linear momentum of the particle. So this linear momentum is playing a really interesting role, you can see what is this role here? If we think about this linear momentum, the rate of change of the linear momentum is equal to? The applied force. Nice. So the rate of change of the linear momentum is equal to the force. We're going to show that actually Euler equation or what it does, it says the angular momentum, if we know the angular momentum; if we take the rate of change of the angular momentum is equal to the moment applied.

And with this symmetry, basically, then we will be able to compute this tensor, or this inertial matrix, associated with a rigid body. Okay. So the rate of change of the linear momentum is equal to the applied force. And we call the linear momentum phi, okay? Can you remember that? Phi is equal to mv.

See, dynamics is not that complicated. All right. Let's talk about the angular momentum. How can I make an angular momentum with a particle? So we have an – so we have this particle rotating and we have an inertial frame, and I'm going to compute the angular momentum. Well, basically, I need to – to cross [inaudible] this with a vector, right? To compute the movement of the force and then I can have the movement of the inertials. And so if we take the moment of this force on the right with respect to the origin, o, then we can take m – mv that is like a force, right? It's an inertial force, mv dot. You agree?

So we're going to take the moment of this equation with respect to o, so we need the vector that connect o to the particle, to the position of the particle. Right? Okay. Let's take the moment, you remember the moment is p cross f, on the right, or p cross mv dot. Okay. This is a very important step. If you understand this step, everything that we will talk about later will become just adding all of them together.

So I had a linear motion, I'm just looking at it from angular motion point of view and I'm just taking the moment of this equation with respect to a fixed point. All right. P cross f is a moment. We call it n. What is this? That's right; it's too complicated. All right.

Okay. So you said earlier, mv is the linear momentum. I'm going to take p cross mv. And if I take p cross mv, and I need to get the rate of change of that quantity, this is sort of the angular momentum and I'm computing it ahead of time just to show you what it's going to be. So if we do the computation it will be p cross mv dot, v cross mv, and v cross mv is equal to zero, so that gives you – because when you cross product a vector with itself, it gives you zero. So that gives you the rate of this quantity equal to the moment. And that is the angular momentum.

So the linear momentum is mv and the angular momentum is the vector locating m cross mv. Okay? We'll put it down, so now you have to remember mv, linear momentum; p cross mv, angular momentum. Correct?

Okay. Well, once we get a rigid body, all that we need is to do the sum. So this p will become pi's, mi's, and vi's, and we are going to add all these particles – a lot of them, many of them. So instead of doing this with just a sum, we will do an integral. And because we have objects in three-dimensional space, we will triple integral in dxy and z. And basically you get your inertia for the rigid body.

So let's call it first, this quantity p cross mv, we call it phi. It's the angular momentum. So I'm going to now think about this equation in the context of a rigid body. The angular momentum of a particle we know, we want to find the angular momentum of all the particles. And we assume that this rigid body is moving at some velocity and acceleration related to the instantaneous angular velocity and acceleration we know. Okay? So we need to locate each of the particles. So we have pi and the linear momentum is – I'm sorry, the linear velocity, vi of that particle, is going to be – we study that in the angular rotations, it will be omega cross pi, right?

So what is the angular momentum? The total angular momentum of this rigid body is going to be the sum of all the pi's locating this mi mass' moving at vi, velocity. So now we're going to take vi and bring them down here and we will have pi cross mi, omega cross pi, so we will have a more complicated expression. So we have omega cross pi. So this would be the total angular momentum of the rigid body. Right? Do you agree? If I can count them all, all the particles. It's difficult to count, but that's what's nice about the mathematical models that you can use. You can assume, or, let's assume that I can count them all.

Now, in this equation, do you see anything that is constant, that is not changing, that is independent of the rigid body? Louder. Louder.

Student:[Inaudible]

Instructor (Oussama Khatib):Omega. All these particles are moving at – rotating about that axis with omega. So omega is independent, so what are we looking for? If omega is independent [inaudible] I'm going to go to an integral, I need to get this omega out of the sum, right? We don't need it in the sum. So how do you do that?

Student:[Inaudible]

Instructor (Oussama Khatib):I'm sorry?

Student:[Inaudible] triple [inaudible]?

Instructor (Oussama Khatib):So, yeah. I mean, you basically – the omega is on the left side, you can, if you put minus, you can bring it to the other side. Omega cross p or minus p cross omega, right? You remember this?

And the other thing is the fact that mi representing that mass for that particle. If we assume that we have a homogenous object, then we can represent it – represent the mass by this more volume multiplied by the density, right? So using this and using this – well, I didn't do it yet, so we'll go from that sum to this integral with the same equation before tripping this. But this is what we need to get this out of this equation. It is independent of the equation.

So the phi is your total angular momentum. So let's write this in this following way. I'm going to write it minus p cross d and omega out and that means – and also substitute with the cross product operator. So p cross is p hat, and that leads to this form. In here what you see is minus p hat, multiplied by p hat, [inaudible] all of this is variable depending on the particle; omega is constant.

So phi is equal to this integral times omega. And this integral is essentially your inertia, the inertia of the rigid body. What we call the inertia tensor. So we can write it simply like this: phi is equal i omega.

So maybe I went too fast. You see this relation? Where i is this integral. So for linear motion, phi is equal mv for one particle, and we have Newton equation, which stays the rate of change of phi is equal to f. Okay? It's another writing of ma equal to f. No one remembers this one, the phi prime equal f, one. People are afraid of momentum.

Okay. Now with the angular momentum we're going to see the same thing. The rate of change of the angular momentum is equal to the applied moments. So phi is equal i omega and Euler equation is simply the rate of change of phi is equal to the applied moments. Couldn't be any simpler, right? Then you will be missing something.

Now phi prime is not as simple as the linear motion. When you take the derivative of phi, you get i omega that acts – this is like mass acceleration, but also because of omega, we have i omega in another product of velocities, and that produces centrifugal and Coriolis forces. So we have the two equations for a rigid body. So a rigid body has a linear motion; if I throw this like straight at you, it will – it's going to [inaudible] there will be some air resistance and you will feel some rotational motion. But this combination of linear motion and angular motion is captured by this for one rigid body. Well, I have to deal with multi-rigid bodies attached by these joints and when you put a joint, you are putting a constraint. You're throwing the thing is going to be pulled and pushed and that means we need to eliminate the internal forces in order to find the actual motion. So a very important thing that we establish with this relation is this i, and now we need to really, I mean, this is the thing that I need you to remember. You need to be able to compute i for each of the rigid body of your robot. So this is something that is absolutely needed. And in order to do that, you need to see little bit more into the structure of the inertial matrix or the inertial tensor. So we say the inertial tensor i is basically this integral over the volume of all the vectors locating all the points on the rigid body, and scaled by the density of masses on the rigid body. So if we take this quantity, p hat p hat, so this is the cross product operator, pretend that you can write it in this form. So p hat is what? You remember p hat, the cross product operator, it's a three by three matrix. So if you multiply a three by three matrix by itself, you're going to get another matrix, right? Now, this matrix could be written as p transposed p. P transpose p is what?

Student:

[Inaudible]

Instructor (Oussama Khatib):A scalar. So you are scaling – p transposed p is the square of the vector p. So you are scaling the identity matrix. So on the diagonal of the identity matrix you have the square of – the component of the vector p minus p, p transposed, which is a three by three matrix. So using this relation, we can rewrite the inertia tensor in this form. So let's take this computation. So p transposed p is x squared plus y square plus z squared. And p transpose p time the identity gives you this quantity. All right? Nothing complicated. The other one gives you this. If I have x, y, and z, I'm going to multiply p, p transpose, that will give me this matrix. And the result will be this matrix. So minus p hat p hat in that integral, is this. Okay? Essentially, this is locating the position x, y and z with this vector. And now when we do the multiplication it appears like this. All right. So we're almost there. We need to control – compute the integral of this and put the density and that – then we will have the inertia tensor. So the inertia tensor is – you see this equation here? Now I'm going to put it in that integral and I will find i. Ixx, Iyy and Izz, and Ixy, and Ixy, et cetera. Anything remarkable about this matrix you can tell me? Properties wise?

Student:

[Inaudible]

Instructor (Oussama Khatib):

Symmetric.

Student:

[Inaudible]

Instructor (Oussama Khatib):

What else?

Student:

[Inaudible]

Instructor (Oussama Khatib):Positive definite? That's if you're at zero. If you're at zero, you're in a black hole. Okay. So this ixx, yy, .zz's and the [inaudible] just we – I'm using the previous matrix to recompute this element, and this is your inertia tensor. Cool? You understand now the inertia tensor? Oh, what it is, this three by three matrix is going to each of the points, finding the distance, and then you're walking and integrating all of these from each of the component to find what is the waiting, the inertial properties about the axis xx – about the x-axis, y-axis, z-axis, and what are the coupling between different axes.

So the xx, yy, zz are called the moment of inertia. And the other one are called the product of inertias. All right. Here is an example. If we take a – if we take a rigid body that is nice, symmetric, like cylinder or [inaudible] or cube, basically, the property of symmetry when you do the integration, you're integrating between one side to the other and that leads to some nice properties because if you do this computation at the center of mass, you are going to be able to find the inertias and you will find, most of the time, zero product of inertias. And then if you need to do this computation at a different point, or what you need to do is now to look at your vector, so if you start from this vector, this point, you will be able to reach all these different points. Then if you start from a different point, to reach this point you can go like this and plus this. This one you already used to find the inertias about the center of mass, or what you need is to have this vector. That because you're going to add the same vector for all the points, it turned out that you have a very nice property, which is the property of parallel axis theorem that tells you the inertia about any point a, is the inertia about the center of mass plus the mass of the object multiplied by this quantity. That is the quantity that let us compute pc, this vector, in addition because all the masses are basically can be measured like, concentrated at this point. So this property is very nice because you can do simple computation at the center of mass, and then just do your transformation to move to a different point.

So there is an example of a cube where we do the computation at the center of mass, you have this example, and the answer is, you get Ixx at the center of mass, Iyy at the center of mass, and Izz at the center of mass, all equal to ma squared divided by six, for this cube. Now, if you want to do this computation at one of the edges, like at this point, or what you need to do is to add this quantity that represents the distance from that point to the center of mass, and scaled by the center of mass and you obtain this quantity.

So this is a very useful theorem if you are doing this computation. And what you notice, also, is that then you will have product of inertias. So the – at the center of mass, you have no product of inertia, you have a diagonal matrix. When you compute the inertias at a different point of the center of mass, you are going to have product of inertias.

All right. so now we have the inertia of a rigid body, then we can apply the equations of Newton-Euler, and we do this propagation and we will compute all the inertial forces as we propagate. And then we'll project back to compute the forces and we find the dynamic equations.

So as we saw, these are the two equations, the translational motion and the rotational motion. And what you need to do also is to compute accelerations; you remember here, we're saying you need to compute velocity, find the accelerations, and then you can find the inertial forces. So we can go over this and now, I'm not asking you to really follow the details of this computation, but essentially, you're going to start from the velocities. The recursive relation of velocities as we propagate, or maybe i plus one equal omega i plus the joint velocity of that revolute joint, and you have this relation. Now, if you take the derivative of this relation, you will find this derivative involving the acceleration of your revolute joint, you will also need to compute the linear accelerations. You start with the velocities, you take the derivatives, and you get your recursive relations, okay?

So we are forward-propagating to go to the last joint, and you have a lot of derivatives depending on the type, if d is revolute, prismatic or not. And we're not done because we have to find the velocities and acceleration at the center of mass, not only at the joint angles. So you need to do a small addition, the linear velocity at the center of mass, you need to multiply by this vector, omega I plus one. And that gives you the velocities, acceleration at the center of mass, and now you're ready.

So at the center of mass, you can write the forces – the linear and angular forces. So this is the inertial forces acting at the center of mass, and these are the linear forces acting at the center of mass, and this is the inertial tensor computed at the center of mass. Now, you take this moving, accelerating rigid body and you say that all the forces applied to this rigid body should be equal to the inertial forces and the acceleration of related to the movement through the Euler equation. And you write these two equations, and now you do the recursive relations with these expressions and you'll get your recursive relations so now you see it is fi as a function of i plus one, so you're backpropagating.

And as you backpropagate, you have to make sure, in order to find the torque, to project on the axis. So we did this, we do this, and now we have the recursive relation and then you project at each of the axis, your end, you compute from here, or f computed from here – and you have those relations, oh, my God. I'm lost. So you can see it's – I mean, it's really difficult. It is – but it is wonderful algorithm, in fact, to compute precisely your force as a function of the velocities, acceleration, inertias, and masses and everything. But you have no idea about what's going on, right? It's very difficult to see. You need to – okay – so – this is [inaudible] iterations from zero to five, and [inaudible] iteration with – don't forget this to eliminate your – okay?

Yeah. Well, I mean, it can – [inaudible] it can go to 25, but it works. Yeah, what about the gravity? We didn't talk about the gravity here. We forgot the gravity. Oops. So what do we do for the gravity? How do we account for the gravity in this algorithm? So, okay, you remember, I say the algorithm, you'll start from the base, you assume the base at zero velocity, zero acceleration, and you move out and you come back. Now, if you want to account for the gravity, what should we do?

Student:[Inaudible] accelerates at g.

Instructor (Oussama Khatib):Yeah. Just [inaudible]. Very good. So if you say I have a linear acceleration of equal to 1g from the beginning, you will be including the gravity. Good. [Inaudible] for a second. Now, all right. So good, we will skip this and to skip it, I will go to here. And we go to the Lagrange equations.

All right. So what we saw, the thing you have to remember and not forget, is how to compute your inertia. We – you need still to compute the inertia because when we go and compute the kinetic energy, that is needed in Lagrange formulation. You are going to compute the linear motion, kinetic energy, which is one-half mv square, and what about the angular motion? The kinetic energy associated with angular motion?

Omega –

Student:[Inaudible]

Instructor (Oussama Khatib):So it is one-half omega transpose i, omega. So we need this i anyway, you cannot escape. So you have to know how to compute i, correct.?

Okay. Lagrange equations, actually, I mean, we can skip that if you want. [Video]

An innovative –

Instructor (Oussama Khatib):No, no, no. What I meant is we really don't have to know all the details of the equations. But I really want you to understand Lagrange equations because they are going to be very useful for you when we get to control. And when we are going to control the robot, the robot has its own dynamics. And also we are going to apply to it external forces to control it. So these external forces are going to affect the dynamic of the robot in some way.

And we need to understand the Lagrange equation, not only to compute the dynamics. All that we need for the dynamics is the kinetic energy, and we know the answer from Lagrange equation or from Newton-Euler – from any formulation of the dynamic equations, we will have the same structure, mass q double duct plus v plus g equals torque. And to compute the mass, all that we need is to compute the kinetic energies. But we really need to understand what this – what is this structure of Lagrange equation, how under applied forces mechanical system is going to move.

So this has a very important rule, not only for the computing the dynamics, but really to understanding the control. Okay? How many of you have seen this equation before? Okay. Six. All right. So let's imagine this equation in the scalar case, so this equation would be just simply an equation where the torque is the torque applied to one revolute joint. So this is a scalar equation. What is l, those of you who have seen it before?

Student:[Inaudible]

Instructor (Oussama Khatib):So it is the Lagrangian and it is simply the kinetic energy minus the potential energy. And what is this q? So when we – when we look at – okay, maybe I think I have – I have k minus u, so l is k minus u. I don't know if I will put it in the – so most of the time when we are talking about natural gravity, the u is only function of where, what height you are at. So it's function of q. So we can rewrite this equation in this form, right? I'm separating the kinetic energy from the potential energy because the potential energy is independent of the time, and so – so you can write it in this way. And essentially, what you have in here, you have mq double duct, and plus v if you have multi-body, and here you have g, the gravity. So here, you just have the gravity vector. So in fact if we move the gravity vector to the other side, let's not worry about the gravity here in space or – the gravity vector is just the gradient of your potential energy. So essentially, it's saying that you're inertial forces are equal to the torque's minus the acting gravity. And if we think about it, this part of the inertial force, this is what it's going to give you. When – once we do the derivation that that left part will give you mass acceleration plus centrifugal Coriolis forces equal to these torques minus the gravity. So let's look at this little bit. As I said, the kinetic energy, if we have generalized velocities, if we say q is our generalized coordinates, q dot is our generalized velocities, we can write the kinetic energy as a quadratic form on the joints velocities. So k is q dot transpose n q dot. And if we take the derivative, so I'm going out to take this k from there, and differentiate with respect to q dot, okay? So partial derivative of k with respect to q dot, of this quantity, and this is q dot transpose m q dot. So what do you think the answer is? No, let's do it in the scalar case. So you are taking, let's say I'm taking mv square – one-half mv square and I'm taking the partial derivative with respect to v; what would be the answer?

Student:

[Inaudible]

Instructor (Oussama Khatib):Mv. Well, in the vector case, it will be mv as well. It will be mq dot. So nice. This is really nice. So now, take the second derivative of this, I mean, this derivative with respect to time. So if you take the derivative of mq dot with respect to time, you get mq double dot, plus m dot q dot. You see why m dot? Because m is function of q. Right? Function of q, so you get m dot q dot. Oh. So let's write this. We computed this, right? And it is – this first part is m q double dot plus m dot q dot. What about the other part?

You see k over there? Need to find the partial derivative of this k with respect to the q's. So in the kinetic energy, what is dependent on the q's? Can you point that to me? Okay. The kinetic energy is one-half q dot transpose, m of q, q dot. So what is dependent on q's?

Student:M.

Instructor (Oussama Khatib):M. All right. So I'm going to write it as a vector because this is – what does this mean? It's partial derivative with respect to q, one partial derivative with respect to q, two partial derivative with respect to q n, so it's like this. So I'm just writing exactly that quantity partial derivative with respect to q, one, two, q and –okay? You agree? So everyone who have never seen Lagrange equation before agrees also? Is this clear? So this means mass acceleration, m q, m dot q dot, so if q dot was zero, this disappears, right?

If m was constant, not configuration dependent, this would be zero and this would be zero. Right? Okay. This messy thing is your centrifugal Coriolos forces, it's product of inertias – if you think any element of this, it's a product – I'm sorry, product of velocities. You have always q dot, q dot, that is multiplied.

So this is what we call v, v is this vector minus half of this vector; and this is v. Okay. So this equation, the partial derivative of the kinetic energy leads to this equation. As I said, because we see the answer, you need to compute m, right? But m is there in the kinetic energy. And these things are function of m. So we really have just to compute m from the kinetic energy and we know the answer.

By the way, do you know why this, we already saw this example – what is the answer? Is, I'm mixed up. Okay. so this – I mean, I way to think about is it to form vector of v which is m square root of m q dot, and then your equation will be v transpose v for the kinetic energy, and then you can show that, et cetera. All right. If you want to see why it is equal to m q dot.

All right. So the equation of motion using Lagrange equations, are in this form, mass acceleration plus these two vectors, leads to this v vector that is function of q and q dot. And if q dot is equal to zero, v will be equal to zero. So this is the structure of our dynamic equation and what we know is inside the kinetic energy, there is this m. If I can't compute the kinetic energy some other ways, then I will find m. And what we're going to do, we're going to go to each of the rigid bodies and compute that kinetic energy for k1, k2, k3, to km, add them all together, and identify that expression to this expression and we can extract m.

And once we have m, using this relation between v and m dot q dot and the vector, we will be able to compute v. So this is what we will be doing on Wednesday.

[End of Audio]

Duration: 74 minutes

IntroductionToRobotics-Lecture12

Instructor (Oussama Khatib):All right. So today, we're going to the Alps. [Video Playing]

An innovative Space Rover with extended climbing abilities. Switzerland.

Autonomous mobile robots have become a key technology for unmanned planetary missions. To cope with the rough terrain encountered on most of the planets of interest, new locomotion concepts for rovers and micro-rovers have to be developed and investigated. In this video sequence, we present an innovative off-road rover able to passively overcome unstructured obstacles of up to two times its wheel diameter. Using a rhombus configuration, this rover has one wheel mounted on a fork in the front, one wheel in the rear, and two bogies on each side.

Here, we can the trajectory of the front wheel mounted on the fork. An instantaneous center of rotation, situated under the wheel axis, is helpful to get on obstacles. To ensure good adaptability of the bogey, it's necessary to set the pivot as low as possible, whilst simultaneously maintaining maximum ground clearance. This architecture provides a non-hyper static configuration, allowing the bogey to adapt passively to the terrain profile.

Motion in structured environments. For climbing stairs, it's necessary to have a good correlation between bogey size and step dimensions. We can see that the rover is able to climb regular stairs effortlessly.

Motion in an unstructured environment. Also made some outdoor tests on rocky terrain. As can be seen, a rover demonstrates excellent stability on rough terrain. It advances, despite a lateral or frontal inclination of 40 degrees and is able to overcome obstacles like rocks even with a single bogey.

In the next step, the robot will be equipped with adequate sensors for fully autonomous operation.

Instructor (Oussama Khatib):So it's quite interesting because when we think about wheeled robots, we think about mobile platforms, working in indoor environments and it's really difficult to imagine these machines going around and being able to, like, go over obstacles. So we usually try to use – what do we use in uneven terrain? I mean, what would be the solution, the other solution, alternative solution?

Student:[Inaudible]

Instructor (Oussama Khatib):Um hm.

Student:[Inaudible] legged?

Instructor (Oussama Khatib):Legged. So we all would think legged locomotion would be the solution. And, in fact, legged locomotion is very, I mean, adapted to those problems because now you can move each leg and go over obstacles. But in here, we can see a way that you, you – in fact, putting that compliance inside the structure and moving the structure as you are adapting to the environment. So this is really a quite nice solution. But there was also designs that combine legged locomotion together with wheeled locomotion.

So you have a hybrid solution, where you are using the wheels and pushing with the legs. So there are several ways of even going further, beyond just modifying the chassis itself, going to adding some propulsion by the legs. So in fact, this project was pursued further, and I'm not sure if we will have more videos on this one.

Okay. So let's go back to dynamics and I think, today, we will finish that portion of the lecture. I think I emphasize dynamics, I also emphasize the fact that dynamics is very, very closely related to control and we really need to understand those equations of motion in order to be able to control, well, the robot. So let's go back a little bit to what we saw on Monday about the Lagrange equations. We saw that we can describe the dynamic behavior of the robot that is its motion. It is as a function of the torques applied to the robot through this Lagrangian equation that involves the kinetic energy and the potential energy. So k is the kinetic energy and u is the potential energy. And we saw that because our potential energy is only function of the configuration, we can separate this equation and find the structure related to the inertial forces and the forces applied to the robot, that is the gravity and the torques. Which means that we can write the equation, put the inertial forces on one side and then analyze the dynamics, that is the inertial forces dynamics, and then see the effect of those forces applied to the system. That is, the external torques through the motors at each joint, we have a motor and the motor is applying a torque at the joint or a force at the prismatic joint and also the effect of the gravity that are coming. So if we analyze this equation, we saw that we can write it in this form, in the form of a mass, metrics multiplied by the acceleration at all the joints, plus some additional forces that are a function of the velocity, and function of the fact that the mass matrix is configuration-dependent. And those forces could be obtained if we express our kinetic energy in its quadratic form expression, that is one-half q-dot transpose m function of the configuration q-dot. This is a scalar and if we do the derivation of the scalar, we find, in fact, that we have this equation. That is, this part will bring those elements, the mass matrix, the acceleration and this part, which represent the centrifugal Coriolis forces. And we can see from this expression that when q-dot is equal to zero, this will disappear or, if the mass matrix was constant, this will also disappear. So we saw the proof of this, I think, that you remember. So in this form, the equation now can be written in term of the mass matrix, the derivative of the mass matrix and the velocities, and the gravity forces equal to the applied torques. Because the mass matrix is this quadratic form, now we can – if we are able to extract the kinetic energy from our analysis of the different motions of the links, we should be able to find the mass matrix directly from the kinetic energy. Then we put it there and because these solely function of the mass matrix and the velocities, we should be able to obtain v and that will give us the full equation, simply by computing the kinetic energy. Okay? So that's what we're going to do. We're going to do this analysis and find how we can find the mass matrix. And once we've found the mass matrix, we will find v. And then, g is a piece of cake. Okay. So how are we going to proceed? So here is an example of an arm and we are looking at one of the joints, one of the links, link i. So this is a rigid body and this rigid body has a mass distribution, has some inertia and has some mass. So what we're going to do; we're going to look at this kinetic energy associated with that specific link. And the idea is if we are able to describe and find this kinetic energy, we will be able then to go over all the different links and do the sum and find the total kinetic energy. And because we have this relation, we can put the two together and we can, then, identify the total kinetic energy from the individual kinetic energies. So for link i, the total kinetic energy is going to be the sum of the different kinetic energies. So link i will be here and when we walk through all the links, we are going to find different kinetic energies that we can sum and find the total kinetic energy. But then, we will say the kinetic energy of the system should be expressed as a quadratic form on the jaws velocities and the question is how can we then do this identification and extract m? So you see the algorithm? Just knowing this part and then identifying m from that sum. All right. So I'm sure half of you remember the kinetic energy and the other half does not remember. So how we define the kinetic energy? Those of you who remember? So you have a rigid body at rest and we move it to some velocity v. What is the kinetic energy? Yes?

Student: Do you want identity square?

Instructor (Oussama Khatib):Yeah, but what does that represent?

Student: Is it woven into too much of a rest to [inaudible]?

Instructor (Oussama Khatib):Yeah, I think I understood what you mean, but could you repeat it clearly? So if you want, then.

Student: So it's a work needed to move to [inaudible] from resting zero to – [Crosstalk]

Instructor (Oussama Khatib):From rest to its current state. Yes? Yes, the definition. So it's the work done by the external forces to bring the system to its state from rest. And that means we are going to take a look at this point mass and its final velocity v and that work will come to be one half m v square. Now this is for a point mass. But we are working with links and links are rigid objects, right? And they have rotational motion and – which means each of the particle is moving at different velocity, etc. And there is a quantity that allows us to evaluate that inertial forces generated by those particles, and this is the inertia. So the inertia of the rigid object is going to intervene now if we look at rotational motion.

So the kinetic energy associated with a rigid body whose inertia is i c c here represent the inertia computed with respect of the center of mass as we have everything is represented at the center of mass. We'll take this rigid object from rest, zero kinetic energy, when we reach omega, there will be some kinetic energy and this kinetic energy is going to be?

You.

Student: Could I ask you a question [inaudible] transposed [inaudible]

Instructor (Oussama Khatib):[Inaudible]

Student: Based on being transposed in the [inaudible].

Instructor (Oussama Khatib):Inertia? Oh, I get it. So if this was just one degree of freedom, this would be basically a scalar i, representing inertia about that axis omega square. But we are looking here at the spatial motion so i is a matrix and the omega is a vector, so we need to write it in this form. Okay? So if we take this rigid body and put to it, together v and omega, that it is undergoing both angular motion or rotational motion plus linear motion, then the two will combine and the total kinetic energy will be the sum of these two kinetic energies.

Okay? We're clear about – this is very important. Once you've understood the kinetic energy, you have all the dynamics. The rest is just math, basically. So you understand this, for one rigid body? Good. Then we're going to be ready. We just write this. All right. Now pay attention. From here, we're going to find the dynamic equations of a multi-body articulated system, directly by summing these k k is. If you understand this, you will see immediately that m is going to emerge this mass matrix for all the structure.

So the kinetic energy of one link is the sum, combining the kinetic energy associated with linear motion and the kinetic energy associated with angular motion. And let's do the sum. So the total kinetic energy is the sum and I'm going to write it.

Just to remind you: We have selected for this structure – we selected a set of [inaudible] q and that means that we have a set of generalized velocities q-dots. So these are the minimal number of parameters, or configuration parameters, needed to represent this configuration. Once q is defined, the configuration is locked, right? So now, we can say because we know these generalized coordinates and generalized velocities, we can say the kinetic energy is also this expression of half q-dot transpose and q-dot. Where m is this positive, definite, symmetric positive definite matrix that appears in this quadratic form. So we have these two different ways. The first one is here. We are saying we can compute individually the kinetic energies without worrying about the joints, the connections, the constraints, right? We are just going and looking at every link and we are evaluating its kinetic energy and saying the total kinetic energy is going to be the sum, without even thinking about what type of joints we have. Now we are saying if we write the expression as a function of the generalized velocities, we have this expression. And the two are equal; it's the same kinetic energy. Are you still following here? Good. And this is the key. Now we are going to identify this expression with the sum obtained by the individual links. And somehow, we are going to, like, work a little bit here to come up with this mass matrix. Okay. So now, help me. What is needed in here in order to extract m and find its expression?

Student: Jacobean?

Instructor (Oussama Khatib):So I heard Jacobean. Well, that's correct. [Crosstalk]

Student: It means, to translate the velocities to joint space.

Instructor (Oussama Khatib):So in the left-hand side, we have half and here we have half, right? In the left-hand side, we have q-dot transpose q-dot around the m. In here, we have v c i, the linear velocity at the center of mass of each of the links and the angle of velocity at each of those links. So what you are saying is we need somehow to express these v c is and omega is as a function of q-dot. And if we do that, then we can say – we can put it into an expression similar to this and extract them.

Is that a good way? You all agree? Should we do it? Why not? You have it in your hands, but let's do it. Okay. So to do this, we are going to use the Jacobean and what we need to do is, somehow, to come up with a Jacobean – this is not the same Jacobean we talked about before. That is, if we look at what we developed before, we developed a Jacobean at this point, you remember? At the last link, we had a Jacobean that allowed us to compute the linear and angular velocities at this point. And these were called what, those Jacobeans? We had two Jacobeans? Anyone remembers? Before the mid-term? J v, which is the Jacobean associated with linear motion and j omega. So easy, it's j omega. J omega, the Jacobean associated with angular motion, right? Okay. So but this was defined here at the end of factor, and this was the velocity – corresponding to the velocity generated by all the qs, all the q-dots, right? But now, I'm talking about the linear and angular velocities at this point. So it's not going to be this v c i is dependent on which velocities? Q-dot 1, q-dot 2 2, q-dot i minus 1 or i, we're not sure – well, you see, we are at the v c i, which is center of mass of the link. So q-dot i is just before. So it is dependent on up to q-dot i. Okay. All right. So I will define it later, but essentially, we need to come up with a Jacobean that will capture all these qs because I need a q-dot at the end. I don't want just to go from q-dot 1 to q-dot i, I need to be able to write this matrix so that I can multiply it by q-dot. So we will define j v i in a bit, but we need a matrix like this. You agree? What about omega? Well omega is going to be in the same form. We will have another matrix; we call it j omega i. This is different from j v and j omega. When we say j v and j omega, we mean the Jacobean associated with the end of factor. When we put the i, it is really related to the velocity of that specific one, okay? All right. Now let's go and plot this in this equation. So now you substitute v c i with j v i q-dot and the transpose of dot, so the q – what is the transpose, v c i transpose would be? Q.

Student:[Inaudible]

Instructor (Oussama Khatib):Q-dot transpose j v c i transpose. So when you do those, you're going to have the transpose of q-dot here, the j v i and the j v i coming from the v c i and the same thing for the omega. All right. We're almost there. You're going to see the mass matrix emerging. Someone can help with the next step? We're almost there. Actually some of you already see what is m, but let's – what do we need to do?

Student: Vector of the q 5?

Instructor (Oussama Khatib):So we notice this i; we have an i here; we have these i s but this, these q s are independent of the sum. So we can take them outside, right? And that's it. Do you see the Jacobean in the mass matrix? Do you see the mass matrix m? It is quite amazing. Your mass matrix is simply those Jacobean, transposed Jacobeans scaled by your mass property. That's it.

So all what this mass matrix is, is just to take the Jacobean associated with those specific points of the center of mass and see their impact on the velocity because they are capturing the effect of the velocity and you are scaling them by m i or i c i. So if your robot was one degree of freedom, so i will be one. Basically, this would be m 1 and j v 1, transpose j v 1 and that will be the inertia of that first link.

Now if you have multiple links, what you can see is, you can see the Jacobean matrix of all of these links are going to contribute so you can think about this as the sum of m m 1 m 2 m 3 m m. And you can see the impact of each of the links on the total mass matrix. So we will take an example a little later, but you can see how each of those links is going to affect the mass matrix. And as we propagate and move from one link to the next, we are capturing the inertial properties coming from the lower joints and moving down to the end. So that is your mass matrix and now, the rest is really computation, just getting this v vector from the partial derivatives of the mass matrix. That's it.

By the way, I still haven't defined what is j v i. It is – I said this is the Jacobean associated with that center of mass and I think we need to define it more carefully. So in order to be able to capture j v i at the center of mass, we – and express it as a function of all the qs. What we're going to do is, we're going to take this vector, locating the center of mass and taking its partial derivatives up to the point q i. This is column i, and every column after that will zero. So by definition, this matrix is the Jacobean matrix computed with this vector, p c i, and – up to this point, and then we are adding those zero columns. Okay?

So what about j omega i? What would be j omega i? Without looking at your notes. J omega used to be what?

Student: Epsilon [inaudible]

Instructor (Oussama Khatib):Epsilon i bar?

Student:Z [inaudible]

Instructor (Oussama Khatib):Z 1 epsilon 2 bar z 2? So we will do the same thing up to z i and we add zeros, columns. Okay? So now, you know this definition of j v i, j omega i, and now you can compute this? Okay. You already have the d h parameters for the Stanford [inaudible] on. Let's compute the mass matrix. You have ten minutes.

Student:[Inaudible]

Instructor (Oussama Khatib):No? Well, it will take more than ten minutes, I'm sure. So I will take an example of 2 degrees of freedom. In fact, before even going to the example, let's do something, something of just the analysis of this mass matrix we talked about.

So I'm not sure if you would really see what is in here but when you think about this mass matrix, as I said it is sort of like symmetric, positive definite. And it has a lot of properties and you can connect those properties to the structure of the robot. So first of all, let's see what m 1 1 represents. M 1 1. So I have a robot, I'm going to use my arm to illustrate it. You can help. So this is the first joint, rotation about this axis. And this is the second joint. Okay? So it's in the plane. So could you tell me what m 1 1 represents?

So let's take this manipulator and lock it. It is one link, the whole thing, we lock it, right? And I'm going to rotate about this axis. So I apply a torque; there is an acceleration. Basically, inertia from acceleration equal the torque, right? In that case. Just one degree of freedom, like a pendulum, the inertia of the pendulum. So that inertia of the pendulum is captured where, in this matrix? Can you see it? No? Not yet? It is m 1 1. M 1 1 is representing this inertia. Now this inertia is a function of what? Of the configuration. So if I change like this, I'm going to change the value of m 1 1. I move like this, it is lighter, heavier, okay? Okay, m 2 2, the same question. It represents what? Oh, come on; we did m 1 1, so m 2 2?

Student:[Inaudible] m 2 2 is locked?

Instructor (Oussama Khatib):So if we lock all the joints after m 2 2 – I mean after joint 2, m 2 2 would represent what?

Student: Inertia [inaudible]?

Instructor (Oussama Khatib):So it is really the inertia perceived at joint 2. So all these diagonal element are representing the inertia perceived – the effective inertia perceived at each of the joints. Okay? Let's go to m n n. Hey, wake up. Come on. M n n is this last link, okay? So this is representing the inertia perceived about this axis, right? It's function of what? So this is link n and it is not really my hand, it is just a constant link, so a rigid body. It cannot move. So m n n is representing that perceived inertia and its function of which variables?

Student:[Inaudible]

Instructor (Oussama Khatib):I'm sorry?

Student: Is it the last joint coordinates?

Instructor (Oussama Khatib):It is the last joint coordinates. Okay, everyone agrees with him? Okay, those who agree with him, please show your hands. One, two, you are in the minority. Democrats, you take over. So it's not correct. Actually, in this case, it's really the law of physics that are going to play and the law of physics says that if a moving one link about some axis, so it is the case of this link, everything is locked and the inertia is – could you tell me? If it's function of anything? Except the weight and the inertia of mass distribution on the link? So it's constant.

So m n n should be constant. It's not function of the q n. But thank you for just making the point to make sure that we emphasize this. M n n is not function of the lost link, lost joint, it is constant. Now the question to you: The previous joint, the previous joint here – well, let's take this one. So you can see, that previous joint, the inertia is going to depend on the joint there. So m n is constant. The previous joint will depend on the next one, right? And as we move, we see m 2 2 is function of what? M 2 2 is going to be function of all the joints that are following. So the robot, if we think about m 1 1, is m 1 1 function of joint 1? No, m 1 1 will not be dependent on the configuration of joint 1. Wherever m – q1 is, the inertia depends only on how we are displaying the structure. So the following joints. Everyone sees this? Good.

Student: So you're saying it's just constant.

Instructor (Oussama Khatib):No m 1 1 is function of what? I'm not saying – it's independent of the first joint.

Student:[Inaudible]

Instructor (Oussama Khatib):Because as I move this around – if I fix all the other joints, and I move about this, I'm not changing the inertia about this axis. But it's function of all the following joints. So m 1 1 is function of what? Q 2 to q m. M 2 2 is function of q 3 to q n. And the last one is constant. Okay. There are a lot of interesting properties about this matrix. What is the relationship between n 1 2 and m 2 1?

Student: The same [inaudible]

Instructor (Oussama Khatib):Identical, symmetric. If, you have a robot, one degree of freedom robot, m n n. And you model the mass matrix of that robot. And then you hook it to another structure. Well, when you find the total mass properties of that structure, the m n n is exactly the same that you computed for that robot. So if I take – if I take this structure, a robot weld joint 2 to joint n. With its mass properties, I'd find the mass matrix of that robot.

And now, I attach this robot to an additional joint. Look at what happens. In the new mass matrix, you will find the same block, the same matrix is completely inserted here and what you are adding now, you are adding the mass matrix, the mass properties perceived by this joint. But immediately, you are creating coupling. You are creating these couplings between the first joint and each of the joints of the structure you are adding.

So n 1 2, n 1 2 is representing the coupling between the acceleration of joint 2 on joint 1. M 2 1 is the opposite, joint 1 on joint 2. So think about this: If you multiply this matrix by q double dot, q double dot 1, it will be n 1 1 q double dot 1 plus n 1 2 q double dot 2. All of this is going equal to torque 1. So the first equation is the dynamics of the first joint and that dynamics of the first joint, if you try to imagine what that equation will be, it will be n 1 1 q double dot 1, plus all these coupling accelerations equal torque 1.

So when you have just lock joints, you have n 1 1 q double dot 1 more accelerations here. But as soon as you relax the lock and leave it away, as you stop moving, this is going to produce coupling forces. Do you see that? Okay. What does it mean that m is positive definite?

Student: It means that, no matter what q-dot is, the answer has to be greater than zero unless q-dot is identically zero.

Instructor (Oussama Khatib):Very good. So that is – physically, you cannot talk about an object with zero mass, right? So the object has to have a mass. And a mass is always positive, right? If you have a particle of, with some mass, it is always positive. Now when we go to articulated body systems, it's the same property, but it is in a matrix form. And this m has to be positive all the time, whatever. So if we think about the kinetic energy, it's one-half q-dot transpose m q-dot. Well the kinetic energy is always positive. Right? Or zero, if we are at rest. If we are at rest, the q-dots are zero.

So the quantity k one-half q-dot transpose m q-dot is going to be positive and zero only if q-dot equal to zero. Okay. Okay, to discuss this v vector, I'm going to simply the problem and we're going to analyze it with 2 degrees of freedom. I think that will make it a little easier but not completely easy. Because what we need to do is now a little bit of computation using Lagrange equations, going to those vectors we computed, m-dot q-dot minus this big vector. All of these computations, you do them once forever and then you know the structure and then that's it.

But I want you to understand where those equations are coming from, so we're going to analyze it on this. Okay? So this is a two degree of freedom manipulator and now, I'm writing these equations for 2 degrees of freedom. So you can see here – what is the first equation? Could someone read the first equation for me? You have two equations here, right? This is one vector equation that you could write in two equations. So could someone read the first equation? Yeah?

Student:M 1 1 q double dot 1. Is that 1 2 q double dot 2 plus v 1 plus g 1 is torque 1. .

Instructor (Oussama Khatib):So the first equation equals torque 1, the represent the dynamics of the first link. The second equation represent the dynamics of the second link. If we lock the first – the second link m 1 1 q double dot 1 plus v 1 plus g 1 equal torque 1. If the second one is moving, there is coupling coming from q double dot 2 on the first one and the opposite is on joint two. So what I'm going to do is to compute the v. Okay?

We're going to compute the v 1 and v 2 and we are going to do this by going to – by going to the equation of v.

This is a scary equation, but don't worry. It's all there. If you remember when we did the computation of the kinetic, the derivative of kinetic energy, we came up with v equals m-dot q-dot minus this vector. Remember that? Right? Everyone remembers? So basically, v is simply m-dot q-dot minus one-half. The first element is q-dot transpose m q 1 and m q 2. What does it mean, m q 1? Well, m q 1 means that this is the matrix, the mass matrix, all the element of that matrix are taken as derivative with respect to q 1, and the second one with respect to q 2.

So I'm rewriting this equation here in more explicit way; I'm saying it is m-dot. So what is m-dot? It is the derivative of the element of the matrix, right? You agree with this? And what is m q 1? M q 1 is the partial derivative of m 1 1 with respect to q 1, m 1 2 with respect to q 1, basically the matrix with respect to q 1. So I'm writing m 1 1. So it is partial derivative of 1 1 with respect to 1. Okay? And here, with respect to 2.

Okay. Do you see this notation? Basically, the notation is saying the element 1 1 is taking here, partial derivative with respect to 2 or to 1, depending on the variable, the last element. Okay? So we just rewrote v. Now for m, the time derivative of m 1 1, we will write it in this way. We write the time derivative of m 1 1 is the partial derivative with respect to 1, q-dot 1 and the partial derivative with respect to 2, q-dot 2. You agree? These are the only variables that are involved in the q. So I'm just expressing, expanding a little bit this equation, all right?

Okay. So this is just rewriting the equation and now we are going to – this is little bit more about what is 1 1 1. It is partial derivative of m 1 1 with respect to q 1, partial derivative of 2 2 with respect to q 2, etc. So let's rewrite this and if we go further and develop this computation and rewrite it, so this is how it comes. If we write the v vector, we can develop it as the sum of these two terms. So, I mean from the top there, you can see that the vector is involving velocities, a product of velocities, q-dot 1 multiplied by q-dot, the vector q-dot. So it will result into product of velocity q-dot 1 square, q-dot 2 square and q-dot 1 q-dot 2. Right? And we are grouping all of these in this form.

So this is the answer. I'm just doing it for saving time. You don't have to do it. So this is the answer. What is v is as a function of those partial derivatives and the velocities. Okay? Do you accept that? I mean, you can do it but, basically, you can – in fact, you might be surprised why I'm writing plus m 1 2 1 minus m 1 2 1. Writing there 1 1 plus 1 1 minus 1 1 , it is really interesting form. But what it turned out, is that, under this form, there is some pattern that is taking place. And if we look at this expression, this expression has a pattern that is repeated all the way. And when we go to n degrees of freedom, we find this pattern over again.

So it's sort of m i j k with permutation. First element plus minus and there is a permutation involving those three elements. So this expression is going to help us find – finding those matrices that are going to scale the velocities, the product of velocities. That is, if we think about 2 degrees of freedom, we have square of the first velocity, square of the second velocity, and product of the two. But if you go to, like 6 degrees of freedom, you have the square of all the velocities and many product of velocity. Q-dot 1 q-dot 2, q-dot 1 q-dot 3, to q 6 and you have all of these.

So we can always put them in terms of matrix multiplied by the velocities, square of the velocities and matrix multiplied by the product of velocities, and always involving those elements. Anyone knows what this represents? Well, Sir Christoffel discovered this pattern and we call them the Christoffel symbols, the BIJKs, that we can form from combining the permutation of the partial derivatives of m i j. So you start with a an element m i j and you take its partial derivative and then you form these symbols. A BIJK is one-half the element m i i j taken with respect to k and then i k taken with respect to j minus j k, taken with this vector i. Well, so this is the first element and using these symbols, we can simplify the writing of what we saw here, these two matrices, and write them in this form. So you have a matrix multiplied by the square of the velocities and another matrix multiplied, a column matrix in this case, multiplied by the product of velocities.

When we go further, this will generalize and this matrix is function of q; we call it c. This is the centrifugal force matrix because this matrix is multiplied by the square of the velocities and b is the coriolis matrix. And this matrix is multiplied by the product of velocity, and this generalized in this way. It's for n degrees of freedom. So the centrifugal force matrix is this matrix, but when multiplied by the square of the velocities, gives you the centrifugal forces. And the coriolis force matrix, when multiplied by the product of velocities, gives you the vector b q-dot q-dot we, symbolically we put it this way. This will give you the sum – the coriolis forces.

So the c matrix, centrifugal force matrix, has to be of dimension what? How many square velocities we have? We have q-dot. How many q-dots we have? n. So how many, how many squares? We have n square velocity, so this matrix will be an n by n matrix. Okay? This matrix v, how many q-dot, q-dot, how many product of velocities we have? Besides the square? Well it turned out we have n minus 1 times n divided by [inaudible] of, basically, a column. So for 6 degrees of freedom, this is what? 5 times 6 divided by 2, which is? 15. So if you go from 1 2 to 1 6, and then 2 3 to the end, you have a long vector of product of velocities.

How many rows we have here? The dimension of v is always going to be six, or n. So we will always have n rows. So you have more columns. How many columns? This is the 15 columns for 6 degrees of freedom. So this is a wide matrix multiplied by this long vector to produce your centrifuge – your coriolis forces. The coriolis forces are b multiplied by q-dot q-dot. And c also has n rows, but it is square matrix. Okay?

So we can compute v simply by finding the bs and these bs are simply function of the partial derivatives of the element of m. Once we computed m, we just do this differentiation and do the computation. Very simple. Well very simple if you're not doing it by hand. But if you're doing it by hand, for n degrees of freedom, it's complicated. But let's take an example in two minutes and you'll see it's not that difficult for few degrees of freedom and you get the sense of it. Okay? You get the idea here?

I mean, the main idea is to remember m gives you v and v is obtained by two matrices, c and b and these matrices are involving element but are the partial derivative of m. And that's why if n is constant, everything here is zero. So if the mass matrix was constant, there is no centrifugal coriolis forces. Okay, one more thing left is the gravity. I had mentioned the gravity and we need to deal with the gravity.

So how do we compute the gravity? So you have each of the links somewhere and you have the center of mass, right? And as you move up and down, you have different potential energy, right? Higher, better, so your height is very important. And you can compute the height, so – and then compute the potential energy of your specific link. And then you add them together. So the potential energy for, to compute h i, you have a vector – we already found this vector locating the center of mass; we have the height; we have the gravity pointing south. So we take the minus gravity vector; we multiply it by the p c i. With that product, we compute h and that gives you m i.

Now, what is the gravity forces? Well, it is just the gradient of that. You just take the partial derivative with respect to q and you find it. And what is the partial derivative gives you, usually? It gives you the columns of the Jacobean matrix. So essentially, your gravity is simply this minus multiplied by the j v is, times this vector m 1 m 2 to m m. Actually, a very simple way to think about it is, let's look at it this way: You have this manipulator; you have all these links; and you have the center of masses. If you are standing by this, it's almost like you are, at each of the link, you have a force pulling you down. And you are trying to compute the torque responding to that force. So what is the force pulling down? This is the mass of that particular joint multiplied by the gravity, right? Everywhere. Like this, right? You have weights.

So what is the torque responding to these? So let's start with the last one. F, what is the torque? Torque equal j transpose f. So j transpose, in this case, v n f, which is m n g. Right? Just add them all together. And how – and now you have your gravity. Okay? So now, we know the gravity; we know how to compute v. Let's take an example.

Okay, this is – do you see this robot? It's a little bit – I'm not sure. Can you see it? So this is a 2 degree of freedom robot and the first joint is?

Student:[Inaudible]

Instructor (Oussama Khatib):[Inaudible] lift. What happened to your voice? Come on. The second joint is?

Student:[Inaudible] prismatic joint.

Instructor (Oussama Khatib):Oh, yeah, lots better. All right, so we have [inaudible] prismatic joints. To simplify – to really – so what we are doing is, we are selecting – we are selecting this point and representing the end of factor, at this point, so this d 2 is measured from here to here, okay? I mean, we are not putting it in here, we are putting it at the center of mass. And we are looking at a first link that has a mass of m 1, a second link of m 2, and inertia tensor of i c 1 at the center of mass and i c 2. We are locating the vector p c 1 by this distance from the origin. And the origin, located here at the axis so x 1 y 1 and you have z 1. So could we find the dynamic equations of this robot? How to proceed? How are we going to do it? What is the mass matrix? Yes?

Student: The first one would be the first – well, starting from the n 1, let's say the mass of the second link times the distance from the [inaudible].

Instructor (Oussama Khatib):So I mean, we found the mass matrix is the sum from i equal to 1 to n, of m i j v i transpose j v i plus omega. So we can just take that expression and just write it. Okay? Write it out. You can start from the n, from – but basically, you need to compute this, okay? Do you agree with this? I mean, this is what we established so let's just try the equation. So what do we need to do in this equation, now?

What is missing? We need to compute these j v is and j omega is, right? How do we compute j v i? We need p c i, the vector locating the center of mass. Do you see that vector? Well, it is there. You have to be careful how you write it, so p c 1 is – do you agree? L 1 cosine 1 and L 1 sine 1, you agree? If anyone doesn't agree, please make sure. Sometimes, there are mistakes, so. There is no minuses? No, everyone agrees. Okay.

Okay, now once you have the p vector, you do the partial derivatives and you compute your Jacobean. All right? The first Jacobean, j v 1, we compute up to 1. So the first column and add a zero column. The second one, we compute up to column 2, well basically all of that. And now, we want to make – find expression m 1 j v 1 transpose j v 1. So you get these two expressions.

So the first element there is this 2 by 2 matrix. And the second element is that. So the mass matrix is equal to the sum of those four elements. The first element is this – we have 3 zeros and just this. So mass 1, the contribution of mass 1 to the total mass matrix appears here. What is this?

Student:[Inaudible]

Instructor (Oussama Khatib):If everything was zero, this is telling you that the center of mass of mass 1, multiplied by its distance to the axis, gives you some inertia. And this is the contribution of center of mass 1 to the mass matrix. Right? You understand that. And that makes sense. Mass to contribution is to joint 1 and its contribution is appearing by the distance square of that center of mass to the axis. Make sense? But mass 2 has another contribution on the second joint, okay? So you see how this is added. You start with one element, mass 1, and you see its contribution. Mass 1 will never appear anymore. Mass 2 is going to appear here and here. And in different robots, it will appear also in here, in the coupling. But as it happened because of the Jacobean, it's not going to appear anymore. The inertia. With z 1 and z 2, we are going to have the Jacobeans. Here are the two Jacobeans, and when we do the multiplication, we see the contribution of the inertia of link 1 appears only on joint 1. The contribution of the inertia on joint 2 appears only on joint 2because joint 2 is prismatic. So the total mass matrix is here. So the mass matrix, as it happened for this robot, is decoupled. There is no – of diagonal charts. You can see that m 2 is appearing here; m 2 is constant as we said. M 2 is representing the inertial properties viewed along axis 2. That is what? If you lock joint 1, you are moving a mass, you see? And that's all what you see? Right? Okay. Let's lock joint 2 and look at joint 1. When you repeat about joint 1, you're going to see the inertia of joint 1 – link 1 and link 2. You're going to see the distance of the center of mass by the scale by the mass of 1 and 2. Right? Okay. This is – yes?

Student: Well, why is it that, like the term m L square looks like an inertia –

Instructor (Oussama Khatib):It is the inertia of the center of mass when you are looking at the linear motion and then you have the angular motion bringing all the rest of the contribution of the inertial properties. You have both of them. And the mass n 1, which represents the mass of the link at the center of mass, brings in a – for a [inaudible] joint; brings a contribution to the inertial forces by the square. It has to be – I mean, to be a homogenous unit, you basically need distance square. So you have m 1 L 2 plus all the rest of the contribution of the inertia of the link because it is a rigid body and we computed its inertia separately. The next question is: What about how this is varying? As we move this m 2, as we extend the location of m 2, this is function of the distance. You see that? So it's varying.

Okay. We don't have much time, but I think we can do the centrifugal coriolis forces. So we need to compute the i j ks. All right, so I said it is simple and I didn't say why. It is really simple because there are a lot of things that just disappear. What about these m i j ks? What things you would remember from what we said? We said something about this, like, element. M n n is independent of any variable. M 2 2 is function of – m 2 2 is function of the configuration, so if you take the partial derivative of m 2 2, with respect to joint 1, you're going to have zero, joint 2 zero. It's function of 3 4 and the rest.

So many of those elements, like if I take i j with respect to i k, larger than i, than it's going to be zero. And that leads to properties that all the b i i is are zero. All the b i j is are zero for i greater than j and, in our robot, the only variable is just n 1 1. The only element that is changing. So with n 1 1, we have only to consider that it's derivative with respect to 2. That means that we have only m 1 1 2 that is non-zero. And m 1 1 2 is simply – the element was m 2 d d 2 square, so it will be 2 m 2 d 2. And when we write the matrices, that appears like this. So it appears in the b matrix and it appears in the c matrix.

So there are, indeed, centrifugal forces that will appear and the matrix v appears like this. Could you tell me where the centrifugal forces are appearing? Which joint is going to have, to see the effect of centrifugal forces? Joint 2because you can see here, you have faith that that 2, multiplied by this and this goes to v 2. You see that? Which joint will have coriolis forces? This joint?

Student:[Inaudible]

Instructor (Oussama Khatib):Louder. Joint?

Student:[Inaudible]

Instructor (Oussama Khatib):Can you hear that? This is a vector. You multiply this by the top; it goes to v 1. So v 1 has a centrifugal – joint 1 has a coriolis force and joint 2 has a centrifugal force. And it's very easy if you rotate this, you see that m 2 will start going away and that is joint 2. Centrifugal. And the first one is the product of velocity's coriolis. Okay, so the v vector is like this. The gravity vector is already there because we already know j v 1 and j v 2, just you computed. Be careful on your gravity vector. It is pointing down along the minus y direction. And this is your vector.

Student:[Inaudible]

Instructor (Oussama Khatib):So when we are in this configuration, the center of mass 1 and 2, at their distances, will produce a force projected with the cosine and on joint 2, we have only m 2 that is affecting the gravity. And finally, here is your model. Wow, on time. Okay, now we can use it, next time. See you on Monday.

[End of Audio]

Duration: 73 minutes

IntroductionToRobotics-Lecture13

Instructor (Oussama Khatib):Okay. Let's get started. So today's video is about dampening. This is from Vancouver Tech and it was presented as ISRR, International Symposium of Robotic Research in '93.

[Video playing]

At the University of Michigan Robotics Laboratory, we're interested in tasks involving dynamically dexterous interaction between robots and their environments. Computers currently play chess better that all but a few of the best human experts. But no machine has yet been built that can manipulate the physical pieces with anywhere near the skill and reliability of the youngest human chess novice. Our three degree of freedom direct drive robot is endowed with a juggling algorithm that transforms the positions and velocities of a falling ball into desired joint positions and velocities, which the robot is forced to track by use of a nonlinear inverse dynamics controller. Smooth position and velocity estimates are produced by a linear observer, which in turn, receives input from a real-time stereo vision system. The one juggle tasks requires the machine to bat a single ball into a stable periodic trajectory, passing through a user-specified apex. Adding a second ball with an independently specified apex point defines a two-juggle task. The juggling algorithm shown here employs an urgency measure to switch the machine's interest between the reference command corresponding to the two independent one juggles. It's worth emphasizing that there is no planning in the conventional sense taking place in this system. Rather, the robot's impact decisions are induced by its continuous motions in the effort to track a carefully distorted version of the positions and velocities of the two balls. Machine juggling skills, in and of themselves, seem unlikely to play a direct role in the social and economic impact of advanced robotics. However, we are convinced that the problems of controlling contexts, focusing visual attention, and coordinating in real time the constituent behaviors of such skills provides an invaluable laboratory for understanding what is hard about dynamical dexterity. Without a phase regulation control term, the balls quickly wander in phase and eventually fall in simultaneously. In contrast, with phrase regulation again enabled, nearly simultaneous falling balls are successfully separated. In this experiment, we failed to prevent a spatial collision. We hope in the future to better understand the nature of these and other dynamical obstacles, or order to control around them more effectively. Of course, there will always be situations from which the machine cannot recover.

Instructor (Oussama Khatib):Okay. So who's interested in juggling? Well, those who are interested in juggling could try it next quarter in Experimental Robotics. In fact, a lot of the projects in Experimental Robotics involve dynamic skills, throwing a ball into a basket, playing ping-pong, or whatever. So juggling is quite challenging, actually. Well, juggling requires control and here we are. So this is a little bit of a concept that we are going to see over the discussions on control. And the concept is instead of really thinking about the robot as a programmable machine where you need to find all the join motions corresponding to your task. So you want to move to some location and you want to be able to reach that location with some orientation of your vector. Well, basically, what you have to do is you have to solve this inverse kinematic problem to find the joint angles that would allow you to be in that configuration. I'm not sure if a human can do that. Humans usually are really poor at computation, so finding the inverse kinematics, finding all the joint angles that will put you in that final configuration is really difficult. So what do you think humans do?

Student:[Inaudible].

Instructor (Oussama Khatib):Feedback of what?

Student:[Inaudible].

Instructor (Oussama Khatib):So you sort of like think – try to reach for something. Try to reach for the chair in front of you. How do you do it? So you're looking at your hand and you look at the chair and you have this visual feedback. So it's sort of like your hand is attracted by a force pulling you toward that goal position you describe. And this is the concept you see here. It's sort of like potential energy, where the minimum of this potential energy is located at the goal position. And that is going to create a force pulling your hand toward the goal. Your hand is going to just move toward this goal without a priori imaging or knowing where your final configuration is going to be. The final configuration is going to emerge from your motion. We will come to this later. But this kind of idea is really what we call task oriented or operation space control. The idea of really doing the control, not through this inverse kinematic and programming the robot. Well, there is another method, and most robots today are controlled through inverse kinematic, that is, we control the joint motions. So you first decide where you're going to position your hand. So you need to find this configuration, which means all that you know is the position and orientation of the hand. You don't know yet this. So you need to do the inverse kinematic. You solve the inverse kinematic for six degrees of freedom. How about inverse kinematic for this? I'm not sure. But anyway, you might maybe use a mannequin and you just position it and you decide, well, this is a good configuration. And you start from here, and now you survey your joint angles to move to that final configuration. Well, it doesn't really work very well with humanoid robotics, and in fact, a lot of humanoid robotics today are suffering from this problem, the fact that we are still controlling robots using inverse kinematics. However, for [inaudible] robots, with few number of degrees of freedom or if you have a repeatable task, you're repeating the same motion over and over, basically, you've recorded the motion and what is left is how to track that motion. So today, we're going to discuss the basics of control. And we're going to really start slowly with just something like we saw on the video, just natural systems, like you're dropping a ball or you are looking at a pendulum moving. And you are trying to understand just the relationship between the potential energy applied to the system and the kinetic energy resulting from its motion. And then we will analyze how this behavior would allow us to create something like PD control, proportional derivative control. Well, in nature we don't have too much of I, integral action, but we will be able, also, to be able to add integral action if the error is large. And then we will apply this to controlling robots in joint space, so we can control this robot to follow a trajectory that is given in joint space. And then we will discuss how we can apply control techniques directly to the task in the way we do it human, that is, by directly applying a force, not through the inverse kinematic or the joints, but directly to controlling the end effector motion, velocity and acceleration. So at the end, we will see that motion control is not really the only thing we need to do when we have a robot. You really need to interact with the environment. And in order to interact with the environment, you need to control the contact forces. So if you are sliding over a surface, you are moving, and at the same time, you are applying a force of contact. And that is going to be a critical technique in order to interact with the world, affect objects, assemble, move, and cooperate with different robots. So a manipulator like this one can be controlled directly through its joint motion by simply imaging that you have some sort of springs at the joints, and if you put just springs, then this mechanism is going to oscillate if you disturb it. So what do we need to add? I'm just talking about passive mechanism. So we put the spring, and now it's going to hold itself at some configuration. And if you disturb it, it will oscillate. So what do we do to make it more like stable?

Student: Put some damper?

Instructor (Oussama Khatib):Put some damper. So if you place a spring and a damper at each of the joints, you will basically go to that resting point of all the springs and that will allow you to be at that configuration. Right? So to control the robot in joint space, just imagine that the resting point of the spring is changing. So little by little, you're here, then you're moving it there and there and there. And then you can control the joint motions. So this is typically the approach that we will have in joint space control, except the fact that we are not really dealing with the coupling, with the inertial forces created. And we will see little bit more about this. We will see why this could work. I mean, it's not obvious that it's going to work. With passive devices, if you put just spring and damper, the system is passive and is going to somehow rest at some configuration. There will be deflection due to – even at a steady state, like if you let it rest. What other forces will disturb the position? So the springs will go to their rest position but they will deviate a little bit because –

Student: Gravity.

Instructor (Oussama Khatib):Yeah, the gravity. So the gravity will create little disturbance. You need to compensate for the gravity, account for the gravity. You need also during motion account for these acceleration you are generating, that are scaled by the inertias and the masses so they produce coupling. As well as centrifugal coriolis forces. So the equation of dynamics is here. And now we need to account for that. But simply the concept of the control is just a spring, damper system, and the behavior is going to be very close to our mass spring damper, except the fact you have coupling. So if we start over and we want to control the robot directly with respect to the task. So we want to move this end effector to some location. What can you do? Still using some passive springs and dampers. I'm going to give you one big spring and a damper and you just need to place it somewhere.

Student: Perhaps try a GPS position and it could go there? I don't know.

Instructor (Oussama Khatib):The GPS position is good. It will give us where the robot is, and we know where the robot is going so we know the error between the two. But I'm asking what is the concept in terms of moving – I mean, implementing a controller that will work with the task, instead of working with the joints? So I don't want to use this. If I place all the springs at the joints, I need to know the joint displacement. I need inverse kinematics. If I want to control the hand –

Student:[Inaudible].

Instructor (Oussama Khatib):Yes. So, exactly. Just pull it. Right? Just put the spring there. Anyway, I gave you only one spring, so you would have to just place it somewhere. All right. Okay. What is going to happen here is you are going to pull the end effector to that location, to the resting position and it will do this. And everything will fall. You don't know where it's going to be, but it will fall. And we will see that the concept is as simple as this – well, in six dimensions x, y. So the spring is like six-dimensional spring. Okay. So this is basically the concept of task-oriented control. I mean, you can think about the spring as passive spring or some attractive potential energy that you are creating at the end effector with a gradient that is pushing you towards the goal. And that gradient is actually here, coming from the spring. The spring has a potential energy when it's disturbed, and when you go to rest, you reach the minimum of that energy. And essentially, you're applying the gradient of the potential energy. Okay. So in space, this is what is happening. As I said, we have an inverse kinematic problem, we have a task that is described in terms of x, y, z, the orientation of the end effector, alpha, beta, gamma, whatever representation you have. You need to compute the desired joint motions. And then you have those desired motion, one desired motion, two, etc. And you look where you are, so you measure from in quarters, q1, q2, qn, you form a small error between where you want to go and where you are, and then you are reducing this error by control, independent controllers, most of the time, sent to each of the joints. So you have several controllers at each of the joints taking the joint from some value of theta to another value of theta. The problem is you have this inverse kinematic over time in the system. Another approach that came about as early as '69, '70, '71, and there is a paper by Dan Whitney in '72 describing resolved motion rate control. So rate means we're looking at the derivative at the velocity. And the idea is to, instead of doing the inverse kinematic, using the forward kinematic and taking its inverse. The idea is to find a small displacement, delta theta or delta q, that corresponds to your desired displacement, delta x. So what do you think we have in our menu as models that could help there? So we would like to find a relationship between delta x, small displacement. So I'm at x, I would like to move a little bit, delta x. What would be the delta q? So what model should we use? And you don't say it. Someone else.

Student: Jacobian.

Instructor (Oussama Khatib):Yes, the Jacobian. Actually, the inverse of the Jacobian. So here is the Jacobian. It relates precisely delta x to delta theta. All right. If you take the inverse, you have to make sure you're not at a singularity, otherwise you have to do special treatment of the configuration. If you are outside of the singularity and if you have six degrees of freedom, regular case, rectangular matrix, then you can take the inverse. Otherwise, you have to resort to generalize inverses or pseudoinverses. So you compute delta theta. So for a small delta x, now you have your delta theta. Knowing your theta, knowing where you are, the next configuration you want to go to is what?

Student:[Inaudible].

Instructor (Oussama Khatib):Delta theta. So you start from your current position, x, you compute the error that is the x desired minus your current position, and then you compute your delta theta, and you add it to theta. So you keep controlling the robot to this theta plus, which is where you were plus the small displacement. And this is a vector. So here is the model. Now, the Jacobian inverse is inside your curval control loop. And you need to compute the forward kinematics, which is easier to compute, especially for you now, right. Very easy. So forward kinematics include the Jacobian. You just have to invert the Jacobian and for a small robot, you can get it almost in analytical form. So basically, you compute a delta q and you distribute it to all the joints. And you have controllers for each of the joints to move and form this error in delta q and you move. So now, you're continuously moving. Well, this has a lot of problems in terms of the conditioning of the Jacobian, the fact that the Jacobian has this strange thing about its metric, because the space where you are measure delta x involves linear motion and angular motion. So linear motion is measured in displacement, in centimeters or meters or inches. But it has also rotational motion, measured in degrees or radians. And it's all included in the Jacobian. So the metric of the Jacobian is not homogenous, and that creates problems. Also, you have the singularities, you have the redundancy, you have all of that, in addition to the fact that you have dynamics. So this works usually most of the time, but it works best if you use it to find the trajectory you want to execute. In the [inaudible] robots, often you want repeatable trajectory. And this doesn't repeat. You will drift. So if you do this simulation, you will be able to find a trajectory, so resolve the inverse kinematic this way and then come up with a trajectory that you can execute. Okay. Well, let's see how we're going to control the robot anyway. We get joint angles, we are following directly the trajectory. Whatever we do, we need to control the robot. We need to create a motor torque that is proportional somehow to the error, so we drive the joints to move toward the goal. So how does it work? By the way, how many of you have had some control classes? Okay, that's what I thought. So we'll start assuming you know nothing. Forget everything. Okay. So what is the simplest system we can consider? Well, I think a mass-spring system would be the simplest you can imagine. So you have a mass resting on a surface with zero friction, nothing sliding. And you have a spring. You pull it. What's going to happen? That you can imagine, I'm sure, everyone. What's going to happen? From rest, you pull it a little bit and let it go.

Student: Oscillate.

Instructor (Oussama Khatib):It will oscillate. So we are very interested in understanding this oscillation and how the oscillation is affected and by what it's affected. So this problem could be resolved and looked at through this same equation we used to find the dynamics. We look at this mass and we find its kinetic energy. And we look at the system. It has some potential energy. Where is the potential energy?

Student:[Inaudible]. Instructor:

I'm sorry?

Student:[Inaudible].

Instructor (Oussama Khatib):I cannot hear.

Student: The spring.

Instructor (Oussama Khatib):The spring, yeah. I'm sorry. I didn't hear you. The spring. Right. So when you're at rest, the potential energy is equal to zero. I mean, if you let it rest alone, you're not intervening, potential energy is equal to zero and the kinetic energy is equal to zero. The velocity is zero. The kinetic energy is ½ (mx)2. So if we disturb it and hold it, what happens to the kinetic energy? Still zero. Potential energy? Going to be positive. It will increase. Now, if we let go, the potential energy starts to decrease, and that energy is transferred to the kinetic energy. And then it keeps decreasing, keeps decreasing, we come to the minimum of the potential energy. We will have the maximum kinetic energy. And now the velocity starts to reduce, the kinetic energy reduces, and we start building potential energy. And essentially, this oscillation is a transfer between K and V. So this is K and we can write this equation. So if you write this equation using K that we saw here, you take the derivative with respect to x-dot. It gives you m x-dot. The derivative of K with respect to x is zero. So you get the derivative of that quantity, m x-double-dot equals F. And the potential energy of the spring is ½ (kx)2, so that gives you the gradient when you take the derivative with respect to x, you get minus kx. Okay. So the Lagrangian equation is a [inaudible] equation in this case, mass acceleration equals force, and the force is minus kx, it's a conservative force. So you are transferring energy between minus kx and kinetic energy, which is building velocity and acceleration. For some reason it's written twice. I don't know why it's written twice, but now I move this minus kx to the left-hand side of the equation and we have mass acceleration plus kx equal to zero. So there is no external forces. This kx is the gradient of potential energy, V. And this is the acceleration of your mass. Now, let's take a look at the response of the system. So I'm not sure if you can see it. You see this red potential energy over there? This is the potential energy of the spring. And let's imagine this green dot that is this point mass we are going to drop. So if we drop this point mass, it's going to fall and it will oscillate. There is no friction. It will keep oscillating forever. So this is time and we are looking at the frequency of crossing this axis. That is, we're going from one side of x, we're going to the negative side. And there is a frequency of crossing. So my question to you is, in relation to these two parameters, K and m, what is the affect of m on the frequency? So if your mass is heavy, heavier and heavier, what is going to happen to this frequency? And if you're k is smaller and smaller, what is going to happen to this frequency? So if k is very big, what will happen? If k is zero, what is going to happen? If k is zero, nothing will happen. If k is larger, the oscillation frequency. So frequency increases with k, and decreases with the mass. There is this quantity that we call the natural frequency of the system and this is the square root of k divided by m. Anyone knows why? Where is this omega coming from? Do you see it from this configuration somewhere?

Student:[Inaudible].

Instructor (Oussama Khatib):So basically – yeah?

Student:[Inaudible] omega and t.

Instructor (Oussama Khatib):Very good. You integrate the equation and analyze its response. If you don't trust this result, let's see how we can resolve and integrate this equation. So if we divide by m, we get the acceleration plus k divided by mx. Now, if you integrate this equation, you get the square root of this coefficient of x that will appear. And we usually will write this equation as omega2x. So the k divided by m is really the square of your natural frequency. And if you write this equation and do the integration of this equation, you get a semizoidal response, where omega appears as the frequency of [inaudible] over motion. So in fact, x that comes from the integration of this equation is some constant cosine omega t plus phi. What is phi? Depends on what? I heard initial conditions. Right. So from the initial conditions of position and velocity can determine c and phi, and this is your response. And you can see that this omega is strictly the square root of k divided by m. Well, if you understand this, we need just one more step and then you understand PD control. It's very simple. PD control, actually, is imitating the natural system to recreate a string. This k will become your stiffness. And m is the mass. k will become your proportional gain. And in a few minutes, we will see another k that involves the damping. That also comes into the equation, but not of conservative system but dissipative system, the system that dissipates energy because of friction. And then we will have the complete equation. So if we are looking at only conservative system without any damping, this is the response. Okay. So in fact, if we just add a little bit of friction underneath the mass as it's moving, there will be some dissipation of energy, and this dissipation would be a force opposing what? Opposing the motion, opposing the velocity. So it's sort of minus some coefficient times x-dot. And that friction, if we add it to the system, we have to add it on the right hand of the equation. So the Lagrangian equation is capturing the natural system on the left side. On the right side, we are putting a natural force, which is friction, but we cannot put it in this left side of the equation, because this force is not conservative, it is not a potential energy force. It cannot be integrated, so it appears on the right side of the equation. An external friction force applied by the environment on the object. And if we assume that this force is simply proportional to the velocity, it could be nonlinear. Friction can be nonlinear, it can have colon friction, it can have stiction. If you add this force to the previous equation that we add, it appears here. This is a second-order equation, general form of the equation. My system is not any more conservative, because if you oscillate now, you're going to lose energy, and little by little, you lose energy and you stop. So the mass acceleration plus b x-dot plus kx is the general form of a linear system of the second order. And if we take this system and analyze is, so we divide by the mass, do you see omega2 now? So we have omega2 and thank you very much, it's finished, so we can continue. Now what we are going to do with b divided by m? Well, when we integrate those equations, this term is going to appear in some form. So what we would like to do is to see how b is affecting this damping. So for instance, if you put a large b, very, very large b, and you start falling, you're falling, you're falling, well, if b is very large, are you going to cross? You will just reach that goal position without crossing. So you have sort of an over-dense system. If b is very small, you're going to oscillate, and eventually, you will lose the energy and converge towards the minimum of the energy. So here is an oscillatory damped system with higher values of b divided by m. We have an over damp system. And as we move from here to here, there is a special value at which we just go and reach the x axis, and this value is called the critically damped system. And remember this; we're going to use it a lot. Because we try to imitate this behavior when we control any of the system. We will try to make it critically damped system. So we need to know for which value b divided by m reaches this value, this state. And the value of b divided by m is simply 2 omegan. So when b divided by m is equal to 2 omegan, omega is square root of k divided by m. Well, then we have a critically damp system. And this comes just simply from the integration of the equation, this condition. So now, you can compute b. The critically damped b is equal to what? 2 omegan times m. So if you know your mass, if you know your k, then you can compute your b to be a gain for your control system. Here b is the natural damping of the environment, so the system is passive. So let's take this 2 omegan and try to make it explicit in that first equation. So I'm going to take this equation and I'm going to write is as a function of omega and as a function of this critically damped system. So to do that, we take b divided by m, which is the value that we have right now, and compare it to the value that will give me critically damped behavior. So b divided by m is compared to this critically damped b divided by m. So this is a sort of ratio. It's damping ratio. And I need to replace b divided by m by something that makes 2 omegan appear, so I need to divide by this and multiply by this. You agree? So on the left I have this ratio, and we call this the natural damping ratio. And we use this symbol to represent it. What do you call this symbol? Psi? So we use psi. Psi represents the natural damping ratio. It's b divided by m divided by 2 omegan. So for which value the natural damping ratio gives me critically damped system? Okay. When zetan is equal to one, it means that b is simply twice the square root of km. And for this value, I will able to have a critically damp system. Okay. So far so good. Not too confused? So we introduced two notions, the natural frequency, square root of k divided by m, and the natural damping ratio, b divided by two square root of km. And now we can analyze our system and write it in this form. So the acceleration, it was m. We divided by m and we can write the equation in this way, we can write it acceleration plus 2 zeta omega velocity, plus omega square root of velocity of x is equal to zero. Now, the time response of this requires us to integrate this equation. And if we integrate this equation, we will have this response. Because of the damping, the amplitude of the semizoidal is reduced as you move. There is a decrease, and this decreases is exponential. And this decrease depends on zeta and omega. You have the semizoidal motion, which is function of your natural frequency omega, but it's also function of your damping ratio. You can see when zeta is equal to one, this will become zero, because sine of zero. And if it's greater, then there is no cosine, because you will have only the exponential. So here is the response. So you have this exponential, and you have the frequency that is now function of omega and square root of one minus zeta2, that is the period is 2 pi omegan square root of one minus zetan2. So it's not omegan anymore. omegan was the natural frequency, but this thing that appears there is sort of a natural frequency that is affected by the damping. So we call it omega damped natural frequency. So it was omega the natural frequency, and now it is damped. All right. I think this is the last definition you need to remember. And with this, we can do almost everything, except the nominalities we have to deal with a little later. But omega, when you have damping, is really omegan the natural frequency that appears in your spring scaled by square root of one minus zeta2, which comes from your damping b divided by the mass and the spring or the gain of your system. All right. So these are characteristics of a second-order system, and what we need to do is to just inspire our control by this, and then we will be able to recreate that behavior simply by selecting what? If you start with the mass and now you want to create a system like this, what do you need to select?

Student:k? Instructor

Select the spring, which is k, the stiffness of the spring, and you need to select?

Student:b.

Instructor (Oussama Khatib):So by selecting b and k, you can create the second order of behavior on our mass. So if you have one joint with some inertia, to create a behavior like this, a closed loop behavior of second order with some natural frequency, some damping ratio, and some damped natural frequency, you should be able just to select b and k and find your system. So the control of a system is going to be almost the same. We are going to pick omega, we are going to pick zeta, which determine k and b. And then you will be able to control the closed loop.

Student: What happens when b is a function of the configuration? Does that affect how you handle that?

Instructor (Oussama Khatib):Right. Well, b could be actually the most general form of b is b is a function of x and x-dot. And even higher order. And then what you get is you do not get a linear system. You get a nonlinear system. So there will be additional disturbances on the system, and you need to do two things, either to model your friction and then try to compensate for that, and that's what we're going to do for centrifugal forces. That's what we're going to do for the fact that the mass is configuration dependent. But once you model it, you can integrate the model in your control and you can compensate for those nonlinearities. Then at the end, after compensation, you come to this form, a linearized form. So what we're going to do, actually, later, is to go and compensate for the gravity, compensate for centrifugal coriolis forces, compensate for nonlinearities, like friction, and then reach a level where we have simply a decoupled system with m or n masses that we can control using this. Now, compensating for friction is very dangerous. It's not easy. You cannot just go and do some estimate and compensate for the friction. You can compensate for the friction – we can try to compensate for the friction if you want. I'll show you how dangerous it is. Okay, here is our – what is the name of this robot? Let's move it. Well, it has some friction. This friction it has – you see I'm moving it and, well, it has friction because it has natural friction, but if I remove this friction, now I removed the friction. Look what's going to happen. I'm going to apply a small force. Are you all attached? All right. We saved it. So you can see if you compensated for the friction, you will have quickly oscillations and you will have instabilities. So let's say the robot is controlled now with these springs. You have 400 – these are the value of the springs. And you have some value for the b's, basically in here, 40. This is joint three. So if we change this to be 40, you see I'm pulling now. It is a little bit moving. You can see joint three. If I pull on joint two, it is stiffer. Joint three is responding. I mean, if we can use this game – four is good. No, that's too much. Let's make it four. Now it's easier to move. But you see the damping. It's still over-damped. When you move it, it is not responding, so let's make this very small. Now there is a little bit of motion if we make this zero. So now, if we start putting negative damping to compensate for the natural friction, we're going to go unstable. So this is small amount. How about minus nine? And let's put a little bit higher here. Okay. And obviously, you're going to go unstable. The real time is not real. That's why it is a little bit weird, but it is – so don't try this. No negative damping. Works. Positive damping is pretty good. Okay. So remember, this omega, is it bigger or smaller than omegan? So you get your omegan from square root of k divided by m, and now your omega after damping is smaller. Good. Okay. Don't look now at your notes, please. You have the answers, but think about this. What is your damp natural frequency? We just saw the expression so don't look at your notes. Try to compute it. So how do you compute your damp measure of frequency? You need to compute the undamped measure of frequency. The natural frequency is what? Eight divided by 2 square root is 2, your damping ration is what? b divided by square root of kilometers, km. So km is 16 square root is four, two is eight. I'm not going to do it. So this is your omega, zeta is – it's easy to remember b divided by 2km2. Remember that. And your omega is 1.6. So you reduce it from 2 to 1.6. So we have another video segment next time, but we will skip it now. We have a little bit more time, so I'm going to go over one degree of freedom that we are going to control exactly as we did with a passive system. So one degree of freedom robot, we're going to assume that the robot has just an inertia or maybe a mass, so it's sliding. Maybe mass is better. If we take a prismatic joint, so a prismatic joint is going to involve just the mass of the moving link. And we're going to move it with a force and we are going to create this force as a spring. And now we want to move to some location, so that would be the resting position of the spring. And then we can recreate exactly the same behavior on the robot. So here is the one link robot. It's just simply a mass. Probably this is the simplest robot you can imagine. It is a mass moving under this force of the motor. So the motor is going to apply a torque translated into a force, f. And you want to move it from its current position to x desired. Okay. Simple problem. So we have mass acceleration equals force. So now, your force is your motor force. And your task is to create a force that will let you move this prismatic joint from its current position to the goal position desired. So the same thing as a spring, what we're going to do is we are going to create this spring or potential energy, whose minimum is at x desired. So the potential energy that you're creating is positive everywhere except at xd is equal to zero, so it's zero at your desired position, and it means that you could have something like a quadratic potential energy with some gain, kp, which produces a gradient that is equal to what? What would be the gradient of this potential energy? f should be what? So you take partial derivative with respect to x. And that would be k times x minus x desired. So new system is simply mass acceleration plus kp(x) minus x desired. So here, we have the zero of the spring moving, changing with your desired position. That's the only difference. And that doesn't change anything. That just changes the zero. So instead of talking about stiffness, we are going to talk about the gain that you are using for your error in position, so we call it position gain. And immediately, we can go to the equation and analyze what happens if I apply this controller and whether this control is going to be stable or not. We can go and do this analysis on the equation on that system, and we analyze what happens when we apply a force that is the gradient of this potential energy. So our potential energy is this. We took the gradient of the potential energy and we applied it. Now going from this page to this page, it's interesting, because here I was looking at just one degree of freedom. But if we do here, we can show that whatever the number of degrees of freedom, if we apply a controller like this, essentially, so this could be six degrees of freedom, 20 degrees of freedom, whatever the number of the system. If your potential energy is in this form, and if you are applying a force that is the gradient of that potential energy, then what you can do is – you see this equation here? You're applying this force there. Well, you can move it to the left side and now you have your potential energy in this equation. So what is happened here is that your Lagrangian equation is showing you that you have a system on the left-hand side that involves only kinetic energy and potential energy equal to zero, zero external forces. So what do you expect in terms of the stability of the system? So if you have a mechanical system under potential energy and kinetic energy with no external forces. Because all the forces are conservative, are gradient. So what we can say about this system?

Student: Stable.

Instructor (Oussama Khatib):It is stable. So simply by selecting your controls, your motor controls, to be the gradient of potential energy, you guarantee that your system is going to be stable. So this is a very important result, because now we know that if we use this form of control, sort of proportional to the error, which is the derivative of our potential energy, then we are going to be stable. Now, stable is not sufficient, because it can be oscillatory stable. We know the response of this system is going to oscillate. So this force is not sufficient. What should we do? We should add some damping. We should add some versipity forces. So in here, I'm going to change this zero with some damping. Very good. We're almost there. So this force is not in the potential energy it cannot move to the left. It is going to be there. And I'm going to put an external versipity force. I'm going to put a damping force. But I need to know in which condition. What are the conditions on this force? What conditions are required in order to make the system asymptotically stable? What does it mean, asymptotically stable? You understand what it means? So this is the behavior of oscillation that are critically damped or over-damped, your system will converge toward the goal and reach that goal. This is what we are trying to achieve. And this versipity force that we saw before was doing something to the system. Someone here said something about that force. That force was doing something to the motion. So what is the condition on Fs? That is the question.

Student:[Inaudible].

Instructor (Oussama Khatib):It has to oppose the motion, right. So if you are moving in some direction, you should oppose that motion. So what would be the simplest way to do that? Fs should be if your motion is measured with a velocity x-dot.

Student:[Inaudible].

Instructor (Oussama Khatib):To the velocity, right. So if you are in higher dimension, basically this could be a force that is opposing your velocity. And what you need to do to make sure of is what? In order to oppose the velocity that the dot product between the two factor is negative. So if your force that product with the velocity is negative for any non-zero velocity, you are asymptotically stable. Very simple. So pick one force that satisfies this condition, the simplest one. You just said it. So you can pick minus kv x-dot with kv positive that would satisfy this condition. So if we apply the Fs equal to this linear damping, then essentially, we take this control, the conservative part of the force, and add to it the damping part. These two pieces represent the PD control, proportional derivative. So if you want to move to a goal position, all you need is a term that captures this error. This goal position could be far away, even, so it's a step response. You're stepping x to this goal position, and you'll have damping that is trying to reduce x-dot to zero, because here you can have x-dot minus x-dot desired. But because you are not tracking a trajectory, you are just going to a goal position, you want to stop at the goal position. So it's x-dot minus zero. This is the PD control. Now, how we design this control, how we pick kv, depends on the similar characteristics we studied earlier with passive system and natural system. So the kp is gong to be picked so that kp divided by m, the mass of the system, gives you the omega that you wish. So when you are controlling your robot, do you wish to have a small omega or a large omega? What does omega do to the response? Small omega. So if you want to move from here to here with small omega, you – takes long, long time. So usually, you want much faster response. If you want to move slowly not only the time response, but also the stiffness of your system, because your stiffness is depending on kp. Your disturbance rejection is depending on kp. We will analyze kp and see why we want higher kp's. We will analyze kp and see its limitations and we will see how we pick those kp's and kv's for given performance of omega and zeta on Wednesday.

[End of Audio]

Duration: 70 minutes

IntroductionToRobotics-Lecture14

Instructor (Oussama Khatib):Okay. Okay. Let's get started. So today's video segment is about tactile sensing. Now, I wonder what is difficult about building tactile sensors; anyone has an idea? So what is the problem with building a tactile sensor? Oh, you used to see the video first, okay. So, yeah.

Student: Do you need functions to be able to, I mean, do you need a perturbation to be able to see what you're touching sometimes?

Instructor (Oussama Khatib):Well, yeah, sometimes you, I mean, a human – tactile sensing is amazing. So you have the static information, so if you grab something, now the whole surface is in contact, and you can determine the shape, right? So what does it mean in term of, like, designing a tactile sensor, just if you think about the static case?

Student: It's soft, malleable.

Instructor (Oussama Khatib):Well, you need some softness in the thing you are putting. Then you need to take this whole information, what kind of resolution do you need, if you are touching to feel the edge? You need a lot of pixels, right? So how can you take this information and – first of all, how you determine that information; what kind of procedure do you – yes?

Student: Well, there's an element of pressure, like, how hard you're – the average – how are you touching on all these different things.

Instructor (Oussama Khatib):Okay. So you can imagine, maybe, a sort of resistive or capacitive sensor that will deflect a little bit and give you that information. How many of those you would need? You need, sort of, an array, right? So how large, like, let's say this is the end of factor. I'm trying to see if you did that problem – you're going to have a lot of information here, and you need to take it back, and you have a lot of wires; you have a matrix, and you're going to have a lot of, basically, information to transmit. So, the design of tactile sensors being this problem of how we can put enough sensors, and how we can extract this information and take it back. So these guys came up with an interesting idea; here it is. The light, please. [Video]:

A novel tactile sensor using optical phenomenon was developed. In the tactile sensors shown here, light is injected at the edge of an optical wave guide made of transparent material and covered by an elastic rubber cover. There is clearance between the cover and the wave guide. The injected light maintains total internal reflection at the surface of the wave guide and is enclosed within it. When an object makes contact with it, the rubber cover depresses and touches the wave guide. Scattered light arises at the point of contact due to the change of the reflection condition. Such tactile information can be converted into a visual image.

Using this principle, a prototype finger-shaped tactile sensor with a hemispherical surface was developed. A CCD camera is installed inside the wave guide to detect scattered light arising at the contact location on the sensor's surface. The image from the CCD camera is sent to the computer, and the location of the scattered light is determined by the image processing software. Using this information, the object's point of contact on the sensor's surface can be calculated.

To improve the size and the operational speed of the sensor, a miniaturized version was developed. The hemispherical wave guide with cover, the light source infrared LED's, a position-sensitive detector for converting the location of the optical input into an electric signal, and the amplifier circuit were integrated in the sensor body.

The scattered light arising at the point of contact is transmitted to the detector through a bundle of optical fibers. By processing the detector's electric signal by computer, it is possible to determine the contact location on the sensor's surface in 1.5 milliseconds. Through further miniaturization, a fingertip diameter of 20 millimeters has been achieved in the latest version of the tactile sensor. It is currently planned to install this tactile sensor in a robotic hand with the aim of improving its dexterity.

Instructor (Oussama Khatib):Okay. A cool idea, right? Because now you're taking this information, and taking it into a visual image, and transmitting the image, and, in fact, this was done a long time ago. I believe the emperor of Japan was visiting that laboratory, and he saw this, and he was quite impressed.

Before starting the lecture, just wanted to remind you that we are going to have two review sessions on Tuesday and Wednesday next week, and we will, again, sign up for two groups. I hope we will have a balance between those who are coming on Tuesday and Wednesday. We will do the signing up next Monday, so those who are not here today, be sure to come on Monday to sign up, all right?

Okay. Last lecture we discussed the controlled structure. We were talking still about one degree of freedom, and we are going to pursue that discussion with one degree of freedom. So we are looking at the dynamic model of a mass moving at some acceleration, X double dot, and controlled by a force, F. So the control of this robot is done through this proportional derivative controller involving minus KP, X desired and minus KV, X dot. So the KP is your position gain, and the KV is your velocity gain.

Now, if we take this blue controller and move it to the left, the closed loop behavior is going to be written as this second order equation, and in this equation, we can see that we have, sort of, mass, string, damper system whose rest position is at the desired XD position. So KV is your velocity gain, and KP is the position gain.

Now, if we rewrite this equation by dividing it by M, we are going to be able to see what closed loop frequency we have and what damping ratio we have, and every time, the lecture time, this finishes. So what is your closed loop frequency? KP is equal to 10, and the mass is equal to 1; what is the closed loop frequency?

Student: Square root of 10.

Instructor (Oussama Khatib):Square root of 10, and what is the damping ratio? A little bit more complicated, but we can rewrite this same equation in this form, 2 zeta omega and omega square where omega is your closed loop frequency, and where zeta is this coefficient, KV divided by 2 square root of KM, and omega is simply the closed loop frequency square root of KP divided by M.

So you remember this, but now the difference with before, before we had natural frequency, so we were talking about natural frequency and natural damping ratio. Now, this is your gain, and you are closing the loops, so this is your control gain; it's the closed loop damping ratio and the closed loop frequency, okay? So the only difference is instead of a natural system with spring and damper, now we are artificially creating a frequency through this closed loop, or we are creating this damping ratio through KV.

So, basically, this is what you are going to try to do, you are going to take your robot; you are going to find those gains, KP and KV, and try to control the robot with those gains. So, again, thinking about KP and KV, KV is affecting zeta, right? And KP is also affecting your omega. Now, when you are going to control your robot, what is the objective; what are you going to try to do? Let's think about it. You're trying to go somewhere, right, or you are trying to track a trajectory. So what do you want to achieve with those, I mean, here is your behavior; what would be good to achieve here? Yes.

Student: It could see in critical damping.

Instructor (Oussama Khatib):So we want to have a critically damped system most of the time, so we will reach those goal positions as quickly as possible without oscillation. So KV would be selected to achieve that value, and for that critically damped system, what is the value of zeta; anyone remembers? It was only two days ago. Zeta is equal to – for critically damped systems, zeta is equal to unity, 1. When zeta is equal to 1, that is when KV is equal to 2 square root of KPM, you have critically damped system.

So, basically, if you know your KP, if you already selected your KP, and if you want critically damped system, then immediately you can compute KV from M and KP, right, for that value, for zeta. So, basically, you are trying to set zeta. What about omega? So now, we need to set KP in order to compute zeta, and how do we set omega? Someone? No idea? So you have your robot, you go and you want to control, let's say, Joint 3. We can do it if you want. Where's my glasses? Here's the simulator. Oh, that doesn't have an F factor. Let's take this one.

So, here are your gains, and right now, if we ask the robot to – so, the robot is floating, and if we ask the robot to go its zero position, it's going to just move, and it's moving with a KP equal 400 and KV equal 40. These are the gain we set for the robot, but, in fact, this is controlled also with dynamics. So we will get to this a little later, but if we want to see the control without dynamics, we take this, probably, non-dynamic joint control, so this one.

So let's float it a little bit. Actually, I can exert a little force outside and see if it can move; it's really solid. Well, okay, won't move it too much. So let's reduce the gain here. So this springiness KP is 40. So see, now if I apply a force that is a deflection, right? And when I'm going to release, it's going to go there, oscillate a little bit, tiny bit, not too much. In fact, this has a lot of friction, natural friction. If we remove the friction and do the same thing, it will probably oscillate more – hm, not enough. Okay. Wow, still there is friction – nope. So let's put a little bit, minus how much? Minus two, this is -20; I think it will go unstable. Wow. So we see that your gain cannot be negative. It will – can you stop? Okay. We need some friction, otherwise it will not stop.

So, in fact, you can see there is a lot of coupling. I moved just one joint, and everything else is moving. Let's make this gain bigger. This is Joint 1, so if I pull on Joint 2, and the release – look at Joint 3; what is happening? So there is an inertial coupling coming from Joint 2 on Joint 3. Just by moving Joint 2, you are affecting Joint 3. You can see, again, Joint 2, release, and Joint 3 is moving. So in order to avoid that disturbance coming from the dynamic, what should we do with KP? Make it smaller or bigger? You're not sure. Should we try it?

So let's make it bigger; how big? 400? Okay, 400. Now we realize with 400, this is not damped enough because we need to compute this to make it a little bigger, so let's make it 20. Okay. So now, what do you expect; the disturbance will increase or will be reduced when I am going to release? More disturbance or less? Heath, less?

Student: Less.

Instructor (Oussama Khatib):Who agrees with less? Okay, and who disagrees with less? Everyone else, okay. So this is less? Yeah, it is less, actually. You're removing little faster, and you are still oscillating, and oscillation is because we don't have enough damping here. So if we increase the damping, it will oscillate less, and if we increase the gain – do you see what is happening now? It's going very quickly to its position.

So, in fact, the coupling – this is the degree – you look at the 90 degree between Joint Link 2 and Link 3. It is maintained, almost. In fact, if I increase Joint 2 as well, it will be hard to move it. So what is happening now with the response; do you see the response when we went to 1600? Faster or slower? Hm? Slower?

Student: No.

Instructor (Oussama Khatib):Faster. So the dynamic response of the closed loop is faster with higher gain. Well then, should we increase it, like, keep increasing? I don't know. We can try.

Student: But there's a limit at some point.

Instructor (Oussama Khatib):So what is the limit? So let's make it 3,000. Now, Joint 3 is locked; it's not moving anymore. Should we make it more? Okay. So what's going to happen? It's not moving anymore. Now, the problem – if this was a real robot, would 30,000 work? Why?

Student: Your motor's gonna saturate at some level in –

Instructor (Oussama Khatib):Well, suppose you have big motors. Yeah, saturation of the motors is one thing, but suppose you have really big motors; it's not a limitation.

Student: Wouldn't you have some sort of air drift?

Instructor (Oussama Khatib):Well, we'll discuss it a little later, but, essentially, what is going to happen is that – remember, inside the structure you have motors, you have transmissions, you have gears, and all of these are going to move, and they have flexibility in the structure. This flexibility makes it that you start to excite those mode of the flexible system, and as you start moving, the motors start to vibrate, and if you have flexibility in the structure, the structures start to vibrate, and when you hit those frequencies of vibration, the system will just go unstable.

So our KP, this KP that we want – oh, we closed it. Just one second, let's go back there. So this KP we have here, this KP cannot go too high. We want it as high as possible to increase what? What it does when KP is high? Disturbance reduction because errors are coming – dynamic coupling coming from other links will be rejected; it's stiffer. However, a KP cannot go too high because KP is deciding the closed loop frequency, and this closed frequency can go as high as those end-modeled flexibilities. Actually, we cannot even come close to them; we have to stay away from them. So omega cannot be too high, which means KP has a limit, but we want to achieve the highest KP.

So what is the relationship between KP, KV, and those performance? So from those two equations, we can write KP is M omega square, and KV is M to zeta omega, right? Just to rewriting these two equations and computing KV and KP. So when we are controlling a system, we are going to set what? We're going to set, really, the dynamics of the system, which means we need to set zeta and omega. So we set zeta and omega, and we can compute our KP and KV. Most of the time, zeta is equal to one. So KV is M to omega, and so all what is left is to set omega. So for 400, omega is equal to what? In the case of the robot in this simulation, we have 400 KP. So omega is equal to? Come on.

Student:[Off mic].

Instructor (Oussama Khatib):Square root –

Student:[Off mic].

Instructor (Oussama Khatib):Divided by – well, M is equal to 1, let's say, in that case. It's 20. It's 20 multiply – what is the frequency, the real frequency?

Student:[Off mic].

Instructor (Oussama Khatib):Omega divided by 2 Pi, right. So what is your frequency about – let's say divide by 6, 20 divided by 6. So it's very low, 3-4 hertz. In fact, if you're lucky, you can go, well, to 10 hertz. I mean, this would be great. So when we go to 1600, this is really nice, 40 divided by 6.

Well, in practice, you start with very low gains, and you start turning your gains up, up, up, up, and suddenly you are going to hit that, noise start to vibrate. So go down, but we will see some ways of doing this in a more precise way, but, again, what you are seeing here is KP and KV – now, if we think about two different links, one link that is heavy, and one link that is light. M equal 1 and M equal 100. Your gain KP is going to be – for the same frequency, is going to be much, much bigger for the bigger link. So that gain is scaled by the mass, and because it is scaled by the mass, we can think about the problem of setting the gains for the unit mass system.

You remember we said if I'm moving Joint 2, the inertia of Joint 2 is changing, big, small. So we need to be able to somehow account for the fact – so I set my frequency; I set omega and set zeta, and now I computed KP and KV, but M doubled. So I need to update my gains, right? If I want to move with the same closed loop frequency, I need, somehow, to update my gains, and that becomes nonlinear control. So we talk about the unit mass gains. So let's just imagine that your system, this mass was unit mass. Your gains will be simply omega square and 2 zeta omega, which is for one, this would be 2 omega. Very simple, just set omega and you get your KP and KV. Okay?

But we know the system is not going to be a unit mass. So for this M mass system, what are the gains? Gains from this KP prime and KV prime. What would be KP for M, a system with mass M using KP prime? KP will be M times KP prime, and KV just linear. So you take M, and you scale your gains, okay?

Well, what is the big deal about this; why I'm talking about? Well, the big deal is that M is going to change, so even for one changing mass you can make this nonlinear, and scale and track a constant frequency and constant damping ratio, but for a system with many degrees of freedom, we have a mass matrix, and we are going to use the same concept.

We are going to say I look at the unit mass system, and then I scale the unit mass system with the mass matrix, and everything will work exactly in the same way, and I will be compensating for the variation of the mass. This is the nonlinear dynamic of the coupling that we're going to introduce, and it is based on the idea that I design the unit mass system, and then I will scale the unit mass system with the mass matrix. Well, in this case, it is just a scalar, simple mass.

So this is what we call the control partitioning. If I have a system with a mass M, I basically – the composite in the mass and the unit mass system. So the blue is the unit mass system, and M is the scaling of the unit mass system. So I can now design a controller for the unit mass system with KP prime and KV prime, and then the KP and KV for the original system will be just scaled by that mass.

So here is my controller F. I'm going to write it as M time F prime where F prime is this quantity, a TD controller designed for unit mass. So we always denote this as primes of KV or KP. So when we say prime, we are talking about the unit mass system. The controller of the unit mass system F prime, and F is M times that F prime. That will make more sense when we go through the multi degree of freedom controller because M becomes the mass matrix, okay?

So, essentially, we have our initial system that is now controlled as a unit mass system scaled by the mass itself, and the behavior of the whole system is like this – well, the dynamic behavior, the dynamic response and the damping ratio are like this, but we have to be careful about other characteristics like the student's rejection, stiffness; they are not, and we will see that in a second. The dynamic behavior of the closed loop is like this.

So you design your controller for the unit mass and basically, if you scale with that mass, then you have the behavior of the unit mass, okay? So, in this case, what is omega for the system? It is simply the square root of KP prime, okay? And now we are going to introduce one more element. We talked about it Monday, and this is just a tiny nonlinearity. Let's add some friction.

So we started with the system without any nonlinearity, and now I'm just adding a little bit of friction, nonlinear friction, like some stiction on that joint. So the equation changed completely. That is, it's not nonlinear anymore. We cannot just treat it as a linear system, and we have to deal with a controller that is going to be nonlinear. So how can we deal with this? Come on, ideas.

So you have your joint, and it has a gear with, like, some friction that is – or even it has some gravity or whatever. Yes.

Student: So if you've got a certain type of friction, you can, like, if it's velocity, then you can put that into the motion equation –

Instructor (Oussama Khatib):Um, hm.

Student:- and change your V value, your KV.

Instructor (Oussama Khatib):KV, you mean.

Student: Yeah, yeah.

Instructor (Oussama Khatib):The KV. So if it is linear, yeah, I think you can, in fact, integrate it directly into KV, but if it is not nonlinear, like just the gravity. So what do we do – if we have the gravity, what do we do with the gravity? We model it. I know the model because I know the mass, the center of mass, all of these things. So if I can model it, I can somehow, like, anticipate what the gravity is going to be and try to compensate for it, very good. So we can compensate for the gravity.

Well, if we have a nonlinear term, what we will do is we put that compensation in the controller. So now the controller, it has the linear part which was F prime alpha F prime. Alpha F prime actually is mass F prime, and now we are going to add another term, beta, which will attempt to compensate for B. You do not know B exactly. You know, sort of, a model with some estimate of B. You don't know X exactly. You don't know X dot exactly. You have estimate of these, what we call the X hat, X dot hat, and B hat. Now, B has a structure. If it's the gravity, it's going to be, I don't know, ML cosign that angle, and you can estimate your mass, estimate your length, estimate the position and come up with an estimate of B, which would be B hat.

So, in that case, you can say alpha is simply the mass, an estimate of the mass, minus/plus one gram, probably you will find it, and your B hat is going to be an estimate of B given the state, your estimate of the state, and you'll probably have ten epsilons, little bit more of error. So we're assuming that we are going to have some errors, but by compensating for those nonlinearities, estimating the gravity and taking it out, later estimating centrifugal coriolis forces and trying to taking them out, we should be able to bring the closed loop system closer to a system that is a unit mass system because with this compensation, if everything was perfect, we compensated perfectly B, then basically beta will take out B. For each configuration, each velocity, beta is exactly compensating for B; it takes it out, and the system is linearized, right? Well, this will never happen in reality, but we will be very close.

So this is what we can write. We can say this is our system, and this is the controller. You understand this controller? This controller is a nonlinear controller, but it is attempting to render in the closed loop, your system, to become the coupled linear system. So here's the result. If B and B hat were identical – if B hat was compensating perfectly for B, and if the estimate of the mass matrix, later this mass was identical to M, then your system will behave this way. So what you designed for F prime will be part of the closed loop of the whole system. We're talking about 1 degree of freedom, but if we are – later we will see 20 degree of freedom, it would be the same, okay?

Well, here is how we can write this system. So our system was F with the output X, X dot, the state. Basically what we are doing is we are looking at the model of the system, and we are using X and X dot to estimate B, the nonlinearities in the system, and compensate for them. So F is going to have a component, which is B hat. In addition, our input control, which is F prime, is going to be scaled by an estimate of M, the mass of the system so that there is a virtual system here that would look like a unit mass system with an input F prime and this same output, and this big box, the red box, is like a system that is linear with unit mass, and that is the purpose of this design. Later, this will be centrifugal coriolis gravity forces, and this would be what – right, the mass matrix.

So, in fact, with many degrees of freedom, we will be able to do the same thing where this becomes the mass matrix, and here we will have V and G. You remember V? Centrifugal coriolis, and G, gravity, and you can add the friction as well, okay?

So, essentially, we are designing a nonlinear controller to compensate for centrifugal coriolis, gravity, and to decouple the system, to decouple the masses, the inertial forces, and to achieve a unit mass system behavior.

Okay. So let's see our design for F prime. F prime is in this structure, in the decoupled controlled structure, and if you have a desired position XD, what would be F prime? Just a goal position, so our goal position, we have X desired. F prime will be minus, minus something. Who remembers? I'm sure you remember. F prime is?

Student: Minus KV prime minus X dot minus KT prime times X minus XD.

Instructor (Oussama Khatib):You meant minus KP prime, X minus XD. So minus KV prime X dot minus KP prime, X minus XD, and the closed loop behavior would be very nice. So we linearized the system. All right. Well, most of the time you're not just going to a goal position. Most of the time you are tracking a trajectory, and on this trajectory you might have, like, you might have different accelerations at different point. You have different velocities, and whereas in this controller, we are just reaching through the goal position. KP prime is trying to reduce the error, and KV prime is trying to put just damping to bring the velocity to zero at the end point, but if you are tracking a trajectory, you have all of these desired things. You have desired position, function of time, desired velocity, and desired acceleration.

So we need to design a controller that is more suited for this. So what F prime would be? See, now we forget about the system because we know we can decouple it, make it linear. Let's think about the unit mass system, how you would design a unit mass system controller, and then you put it in that structure. So what is the objective if you have all these desired things? What should F prime be?

Okay. So you see on the top here is F prime. I have some desired acceleration. I have my acceleration, unit mass acceleration, equal to F prime, and I know my desired acceleration; it's X double dot desired. So if this was really a perfect system, and you are trying to track this acceleration desired, what F prime should be? I think the question is so simple that you cannot believe it. Come on, this is very simple, too simple. So my system is X double dot, and I know the desired acceleration, X double dot desired. What should F prime be? Come on.

Student: Minus the cost of minus X double dot, minus X, E double dot.

Instructor (Oussama Khatib):Yeah, I think you went too far. That is correct, but I'm just saying if the system was able to respond directly to F prime with no errors, nothing, and my system is X double dot, and have the desired acceleration X double dot desired. What I would do with F prime, just make F prime equal to?

Student:X double dot.

Instructor (Oussama Khatib):X double dot desired, right? Right? Okay. Okay, you see what we're talking about? You have your acceleration desired, so just put X double dot equal X double dot desired, and everything should just – you apply this force, and the system should follow X double dot desired, right?

Well, it won't. It will drift because there is really no feedback. You have your acceleration, and you are saying X double dot desired, this is my acceleration desired, and as soon as you start, the system will start accumulating errors, and it will drift. So what should we do? We should do the PD part, and that's why now we are going to add proportional control to the error, the position error. As you said, minus KP prime, X minus X desired.

What about the error in velocity? Because now I have X dot desired. What would be the term that I should use to follow X dot desired? So that would be minus KV – could you finish it? Minus KV –

Student:X dot minus X desired dot.

Instructor (Oussama Khatib):Exactly, from the error, X minus X dot desired, and I will – so here is the controller. So this time, if I have the full trajectory, I will form errors on the position, on the velocity, and I would feed forward the acceleration. So essentially, you are telling the system follow this desired acceleration. It's not going – there will be errors, and I'm tightening these errors. So the closed loop behavior of this is going to be controlling the error in acceleration, in velocity, and in position, if I have the full trajectory in time, and that will, basically, if I call X minus X desired the error, then I'm really controlling the error as a second order linear system, all right?

Okay. So now, we have to make sure that we can do this with the whole robot, and we have to make sure that this controller could work with those gains that we are trying to achieve, and we start analyzing the system. So let's imagine that I designed the system, the compensation, with the B hat – I'm sorry, they are not appearing as hats, but this is B hat and M hat, and I get everything over there, but then – now we are talking about the real system. So when we were running the simulation earlier we saw that a small external force will disturb the system. So there are a lot of forces coming from the errors in dynamics, errors in the gravity estimates, nonlinear forces coming from the gears and the friction that will affect this behavior, and as we start introducing disturbances in the system, we are going to see that these gains that we set are going to play a very important role in disturbance rejection.

So let's add a little bit of disturbance here. So if we add some disturbance, going to take a very simple type of disturbance like a bounded disturbance that we are adding from some, like, type of error in the gravity. Imagine that you have this link, and you have a little disturbance coming from the gravity. So what is the affect of this disturbance on the closed loop now?

So here is our controller with F prime scaling it by the mass estimate and B estimate, we are getting this closed loop. So this closed loop now is going to be equal, not to zero with the disturbance force, it's going to be equal to this disturbance force divided by M, right? By the way, in some textbooks, this is not divided by M; it's directly applied as if it was applied to the decoupled unit mass system.

So because the disturbance is an input of the system, you have to remember that we are dividing by M, so this is divided by M.

Okay. So let's see what is going to happen to your errors. I'm not sure how many of you know what we mean by a steady state error. What is a steady state error here? What do we mean by steady state error? Yes.

Student: It was [inaudible] equilibrium and velocity, what its position error's going to be.

Instructor (Oussama Khatib):So a steady state is like when the acceleration starts, the velocity starts, and you reach that position, but you are not reaching it exactly; there is a small error. So it's like a spring mass system if you apply a little bit of a preservation with an external force, it's not going to go to its rest position; it will be very close but not there, and that would be the steady state error.

So this is what happens when we have the velocity and acceleration errors equal to zero, and that means the last term, the error term KP prime is equal to F disturbance divided by M, and that means your error is going to be the disturbance force divided by M, KP prime. Again, in some textbooks, there is no M appearing there, and this is very important. KP prime is the gain, the position gain for the unit mass system. It renders your frequency constant over all the motion. When we vary M, KP prime, your frequency is square root of KP prime.

However, your stiffness, your position gain, is M times KP prime, and if M is changing, your stiffness is varying. So you do not have constant stiffness over the workspace as you vary M. So your error, your steady state error will vary depending on where you are and how big your M is, and you better get a large KP prime so to guarantee that you have minimum disturbance rejection. So I'm not sure if you are seeing it. KP is your stiffness, is your closed loop stiffness. KP prime is not; it's M, KP prime. Okay? So this is very important, remember this because we will use it when we go to N degrees of freedom. It is the same structure.

Okay. Well, here is the example I mentioned earlier. If we think about just a disturbance for this one degree of freedom, it's sort of a spring mass, and you are applying the disturbance force, and that means the steady state error, X minus X desired, will be given by this equation, which means you are going to rest, not at XD but at XD plus this force divided by the stiffness you have. So this is your delta X.

Okay. So the disturbance is going to produce an error delta X that is given by the disturbance divided by your stiffness, and this is your M times KP prime. Don't confuse KP prime for the unit mass system and the KP.

Okay. Now, how can you get rid of that error, the steady state error? Yeah, I know. I hid that slide, now you know. You don't know. All right, go ahead.

Student: Add an I [inaudible].

Instructor (Oussama Khatib):Do you think he didn't see the next slide? No, you didn't see it. All right. Okay. Why by adding I we can remove that error?

Student: Because over time, the I grows large, if you have a steady state error –

Instructor (Oussama Khatib):Um, hm.

Student:– and it'll correct for that.

Instructor (Oussama Khatib):Yeah, basically you detect the error, and as long as you have an error, you are adding, adding. So, essentially, the idea is now your F prime will include an additional term that magnifies this error with some gain, and keep magnifying and keep adding until you overcome that disturbance. Now, integral action is very good to reduce errors, but you have to be careful about the way you use it. Better to use it close to your goal position not all over, especially if you are moving fast and accelerating, winding and unwinding, that integral might create instabilities.

So the way you can analyze the disturbance is that now you have this equation, this closed loop, and if you take the derivative of this equation, basically, you see that now the steady state is going to be equal to zero. So if you build that integral, then you will take the error to zero. Actually, integral action is very nice when we go to force control. We will talk about it later. In motion control, you have to be very careful about its use.

There is another element of this when we look at the simulation of the puma we are looking from outside, but let's go a little bit inside and see what is happening. So you have the motor. You have a gear with some gear ratio that depends on the diameter of those two spheres R and little r. So what is the gear ratio here? Come on, we have a lot of mechanical engineers. Is it big or small? So what is a typical gear ratio for a robot? What do you think the gear ratio for Joint 2? When I move Joint 2 here, what is the gear ratio? Hm?

Student:[Off mic].

Instructor (Oussama Khatib):That's reasonable. Actually, the gear ratios for the puma vary between, like, 50 and 150. Some robots has 300 gear ratio, and they have multiple stages. All of the joints here, like, 1, 2, and 3 have two stages, and you get a lot of flexibilities, a lot of vibrations that appears and all of that. So it's actually very nice here because we have a very small gear ratio, but I'm not sure if you know what is the gear ratio; what is the gear ratio? About how big be the gear ratio here? 20, 40? No one knows?

Okay, the motor is going faster or the link is faster?

Student: The motor.

Instructor (Oussama Khatib):I'm sorry?

Student: Motor.

Instructor (Oussama Khatib):The motor, so the gear ratio is reducing the speed; the link is going slower. So how slower? One time, two – twice, threes? He said, like, two.

Student: It looks like two.

Instructor (Oussama Khatib):Yeah, it is two. I was worried. Okay. So the gear ratio is really R divided by the other r, the smallest r, okay? So that's your gear ratio, and the speeds, the link speed is smaller by that ratio than the speed of the motor, right? How about the torque? Well, as it's written, so we have a gear ratio of two here, the torque as the link is twice as big as the torque of the motor. So you take a motor, if the gear ratio is one, basically, you'll have very drive, and you need a big motor to produce a torque, right? But this big motor is heavy, and with the robot, putting a heavy motor here is a problem. So what do we do? We put higher and higher gear ratio reducing the size of the motor, and we can achieve the torque because we are using a high gear ratio.

Now, I wanted to show you this to illustrate another problem. It's not about just the speed reduction and torque increase; there is another effect that comes with the gear ratio. When we start moving, so we have this inertia. It's vying with this, right? Big inertia, small inertia as we move. Now, the question is what this motor is going to do to this inertia? The inertia of the link is here. There is another inertia coming from this rotation of the motor, and this inertia is going to be affected by the gear ratio, and the effective inertia perceived of this joint is going to be bigger than the link inertia.

So what is the effect of the inertia of the rotor of that motor there on the inertia of the link? So the inertia of the link alone is IL. The inertia of the motor is IM. What do you expect the real IL, the effective IL to be? So it's going to be equal to IL plus something, and this something is? Again, you?

Student: Nuh, uh.

Instructor (Oussama Khatib):Oh, him, okay, him.

Student: Two, twice.

Instructor (Oussama Khatib):Twice?

Student: Twice the inertia of the motor.

Instructor (Oussama Khatib):Twice the inertia of the motor, so you mean, in general, this is N times the inertia of the motor.

Student: No, it's –

Instructor (Oussama Khatib):That's what you mean.

Student: Inertia.

Instructor (Oussama Khatib):Yeah, so it is N the inertia of the motor, which is similar to the torque increase. The torque at the link is N times the torque of the motor. You're saying the inertia is the same, in the same proportion it is N times the torque, I mean, the inertia of the motor. Well, you're right, it is bigger. You're right, there is an increase by N, but it is not linear by N. Anyone can give me a better estimate? Now you know, but they are not looking. That's why no one is able to tell me.

So inertia of the motor is reflected at the link by – it's not a torque. It's not going to be reflected by N because the motor is moving much faster. The acceleration at the joint is moving also slower by that gear ratio. So the effective inertia reflected by the motor is? How many times IM? N square, much better. So you have a gear ratio of 100. It is big. The coefficient that you are going to carry is really big, and that makes the robot very dangerous, actually, because the effective inertia that you see at the joint is correlated with the impact force that you might reduce if you have a sudden collision, and if you are reflecting this small inertia by N square of the gear ratio, you're going to produce a large impact force.

So here is that nemesis. We can write the dynamic equation on the side of the motor, or we can write it on the side of the link, and when we write it on the side of the link, we see that it's IL plus N square or eight a square IM, and this is your effective inertia. Now, your effective inertia is always by the square of the gear ratio of the motor you are using. Well, the direct drive case is really ideal. We use it with SCARA-type robots where you do not have to carry the gravity, but with the robots that are articulated and that need to move in space, it's very difficult to build robots that have big motors and can carry the structure.

Now, variation of the effective inertia makes it that if you have variation of your IL – IL is going to change. IM is constant; it's the same motor. So the question is if IL is changing, what is the effect of that? Well, the effect of that is changing your time response because your KP, if you remember, is M times KP prime. So it affects, directly, your omega. So you might do an analysis and see if you are using constant gains, but the best way to select your gain is to go to the geometric average and look at your minimum – the minimum value of your IL and the maximum value of your IL, and take that midrange in geometric sense, and that gives you some estimate that you can use over all the range of motion.

Okay. Well, I think we are coming to a point where we can break. So let me remind you that next Tuesday and Wednesday we will have the review sessions, and on Monday we will be signing up for those review sessions. So please, those of you who are not here, please make sure you come and sign up. See you on Monday.

[End of Audio]

Duration: 74 minutes

IntroductionToRobotics-Lecture15

Instructor (Oussama Khatib):Okay. Okay. Let's get started. So today's video segment is about tactile sensing. Now, I wonder what is difficult about building tactile sensors; anyone has an idea? So what is the problem with building a tactile sensor? Oh, you used to see the video first, okay. So, yeah.

Student: Do you need functions to be able to, I mean, do you need a perturbation to be able to see what you're touching sometimes?

Instructor (Oussama Khatib):Well, yeah, sometimes you, I mean, a human – tactile sensing is amazing. So you have the static information, so if you grab something, now the whole surface is in contact, and you can determine the shape, right? So what does it mean in term of, like, designing a tactile sensor, just if you think about the static case?

Student: It's soft, malleable.

Instructor (Oussama Khatib):Well, you need some softness in the thing you are putting. Then you need to take this whole information, what kind of resolution do you need, if you are touching to feel the edge? You need a lot of pixels, right? So how can you take this information and – first of all, how you determine that information; what kind of procedure do you – yes?

Student: Well, there's an element of pressure, like, how hard you're – the average – how are you touching on all these different things.

Instructor (Oussama Khatib):Okay. So you can imagine, maybe, a sort of resistive or capacitive sensor that will deflect a little bit and give you that information. How many of those you would need? You need, sort of, an array, right? So how large, like, let's say this is the end of factor. I'm trying to see if you did that problem – you're going to have a lot of information here, and you need to take it back, and you have a lot of wires; you have a matrix, and you're going to have a lot of, basically, information to transmit. So, the design of tactile sensors being this problem of how we can put enough sensors, and how we can extract this information and take it back. So these guys came up with an interesting idea; here it is. The light, please. [Video]:

A novel tactile sensor using optical phenomenon was developed. In the tactile sensors shown here, light is injected at the edge of an optical wave guide made of transparent material and covered by an elastic rubber cover. There is clearance between the cover and the wave guide. The injected light maintains total internal reflection at the surface of the wave guide and is enclosed within it. When an object makes contact with it, the rubber cover depresses and touches the wave guide. Scattered light arises at the point of contact due to the change of the reflection condition. Such tactile information can be converted into a visual image.

Using this principle, a prototype finger-shaped tactile sensor with a hemispherical surface was developed. A CCD camera is installed inside the wave guide to detect scattered light arising at the contact location on the sensor's surface. The image from the CCD camera is sent to the computer, and the location of the scattered light is determined by the image processing software. Using this information, the object's point of contact on the sensor's surface can be calculated.

To improve the size and the operational speed of the sensor, a miniaturized version was developed. The hemispherical wave guide with cover, the light source infrared LED's, a position-sensitive detector for converting the location of the optical input into an electric signal, and the amplifier circuit were integrated in the sensor body.

The scattered light arising at the point of contact is transmitted to the detector through a bundle of optical fibers. By processing the detector's electric signal by computer, it is possible to determine the contact location on the sensor's surface in 1.5 milliseconds. Through further miniaturization, a fingertip diameter of 20 millimeters has been achieved in the latest version of the tactile sensor. It is currently planned to install this tactile sensor in a robotic hand with the aim of improving its dexterity.

Instructor (Oussama Khatib):Okay. A cool idea, right? Because now you're taking this information, and taking it into a visual image, and transmitting the image, and, in fact, this was done a long time ago. I believe the emperor of Japan was visiting that laboratory, and he saw this, and he was quite impressed.

Before starting the lecture, just wanted to remind you that we are going to have two review sessions on Tuesday and Wednesday next week, and we will, again, sign up for two groups. I hope we will have a balance between those who are coming on Tuesday and Wednesday. We will do the signing up next Monday, so those who are not here today, be sure to come on Monday to sign up, all right?

Okay. Last lecture we discussed the controlled structure. We were talking still about one degree of freedom, and we are going to pursue that discussion with one degree of freedom. So we are looking at the dynamic model of a mass moving at some acceleration, X double dot, and controlled by a force, F. So the control of this robot is done through this proportional derivative controller involving minus KP, X desired and minus KV, X dot. So the KP is your position gain, and the KV is your velocity gain.

Now, if we take this blue controller and move it to the left, the closed loop behavior is going to be written as this second order equation, and in this equation, we can see that we have, sort of, mass, string, damper system whose rest position is at the desired XD position. So KV is your velocity gain, and KP is the position gain.

Now, if we rewrite this equation by dividing it by M, we are going to be able to see what closed loop frequency we have and what damping ratio we have, and every time, the lecture time, this finishes. So what is your closed loop frequency? KP is equal to 10, and the mass is equal to 1; what is the closed loop frequency?

Student: Square root of 10.

Instructor (Oussama Khatib):Square root of 10, and what is the damping ratio? A little bit more complicated, but we can rewrite this same equation in this form, 2 zeta omega and omega square where omega is your closed loop frequency, and where zeta is this coefficient, KV divided by 2 square root of KM, and omega is simply the closed loop frequency square root of KP divided by M.

So you remember this, but now the difference with before, before we had natural frequency, so we were talking about natural frequency and natural damping ratio. Now, this is your gain, and you are closing the loops, so this is your control gain; it's the closed loop damping ratio and the closed loop frequency, okay? So the only difference is instead of a natural system with spring and damper, now we are artificially creating a frequency through this closed loop, or we are creating this damping ratio through KV.

So, basically, this is what you are going to try to do, you are going to take your robot; you are going to find those gains, KP and KV, and try to control the robot with those gains. So, again, thinking about KP and KV, KV is affecting zeta, right? And KP is also affecting your omega. Now, when you are going to control your robot, what is the objective; what are you going to try to do? Let's think about it. You're trying to go somewhere, right, or you are trying to track a trajectory. So what do you want to achieve with those, I mean, here is your behavior; what would be good to achieve here? Yes.

Student: It could see in critical damping.

Instructor (Oussama Khatib):So we want to have a critically damped system most of the time, so we will reach those goal positions as quickly as possible without oscillation. So KV would be selected to achieve that value, and for that critically damped system, what is the value of zeta; anyone remembers? It was only two days ago. Zeta is equal to – for critically damped systems, zeta is equal to unity, 1. When zeta is equal to 1, that is when KV is equal to 2 square root of KPM, you have critically damped system.

So, basically, if you know your KP, if you already selected your KP, and if you want critically damped system, then immediately you can compute KV from M and KP, right, for that value, for zeta. So, basically, you are trying to set zeta. What about omega? So now, we need to set KP in order to compute zeta, and how do we set omega? Someone? No idea? So you have your robot, you go and you want to control, let's say, Joint 3. We can do it if you want. Where's my glasses? Here's the simulator. Oh, that doesn't have an F factor. Let's take this one.

So, here are your gains, and right now, if we ask the robot to – so, the robot is floating, and if we ask the robot to go its zero position, it's going to just move, and it's moving with a KP equal 400 and KV equal 40. These are the gain we set for the robot, but, in fact, this is controlled also with dynamics. So we will get to this a little later, but if we want to see the control without dynamics, we take this, probably, non-dynamic joint control, so this one.

So let’s float it a little bit. Actually, I can exert a little force outside and see if it can move; it’s really solid. Well, okay, won’t move it too much. So let’s reduce the gain here. So this springiness KP is 40. So see, now if I apply a force that is a deflection, right? And when I’m going to release, it’s going to go there, oscillate a little bit, tiny bit, not too much. In fact, this has a lot of friction, natural friction. If we remove the friction and do the same thing, it will probably oscillate more – hm, not enough. Okay. Wow, still there is friction – nope. So let’s put a little bit, minus how much? Minus two, this is -20; I think it will go unstable. Wow. So we see that your gain cannot be negative. It will – can you stop? Okay. We need some friction, otherwise it will not stop.

So, in fact, you can see there is a lot of coupling. I moved just one joint, and everything else is moving. Let’s make this gain bigger. This is Joint 1, so if I pull on Joint 2, and the release – look at Joint 3; what is happening? So there is an inertial coupling coming from Joint 2 on Joint 3. Just by moving Joint 2, you are affecting Joint 3. You can see, again, Joint 2, release, and Joint 3 is moving. So in order to avoid that disturbance coming from the dynamic, what should we do with KP? Make it smaller or bigger? You’re not sure. Should we try it?

So let’s make it bigger; how big? 400? Okay, 400. Now we realize with 400, this is not damped enough because we need to compute this to make it a little bigger, so let’s make it 20. Okay. So now, what do you expect; the disturbance will increase or will be reduced when I am going to release? More disturbance or less? Heath, less?

Student: Less.

Instructor (Oussama Khatib):Who agrees with less? Okay, and who disagrees with less? Everyone else, okay. So this is less? Yeah, it is less, actually. You’re removing little faster, and you are still oscillating, and oscillation is because we don’t have enough damping here. So if we increase the damping, it will oscillate less, and if we increase the gain – do you see what is happening now? It’s going very quickly to its position.

So, in fact, the coupling – this is the degree – you look at the 90 degree between Joint Link 2 and Link 3. It is maintained, almost. In fact, if I increase Joint 2 as well, it will be hard to move it. So what is happening now with the response; do you see the response when we went to 1600? Faster or slower? Hm? Slower?

Student: No.

Instructor (Oussama Khatib):Faster. So the dynamic response of the closed loop is faster with higher gain. Well then, should we increase it, like, keep increasing? I don’t know. We can try.

Student: But there’s a limit at some point.

Instructor (Oussama Khatib):So what is the limit? So let’s make it 3,000. Now, Joint 3 is locked; it’s not moving anymore. Should we make it more? Okay. So what’s going to happen? It's not moving anymore. Now, the problem – if this was a real robot, would 30,000 work? Why?

Student: Your motor's gonna saturate at some level in –

Instructor (Oussama Khatib):Well, suppose you have big motors. Yeah, saturation of the motors is one thing, but suppose you have really big motors; it's not a limitation.

Student: Wouldn't you have some sort of air drift?

Instructor (Oussama Khatib):Well, we'll discuss it a little later, but, essentially, what is going to happen is that – remember, inside the structure you have motors, you have transmissions, you have gears, and all of these are going to move, and they have flexibility in the structure. This flexibility makes it that you start to excite those mode of the flexible system, and as you start moving, the motors start to vibrate, and if you have flexibility in the structure, the structures start to vibrate, and when you hit those frequencies of vibration, the system will just go unstable.

So our KP, this KP that we want – oh, we closed it. Just one second, let's go back there. So this KP we have here, this KP cannot go too high. We want it as high as possible to increase what? What it does when KP is high? Disturbance reduction because errors are coming – dynamic coupling coming from other links will be rejected; it's stiffer. However, a KP cannot go too high because KP is deciding the closed loop frequency, and this closed frequency can go as high as those end-modeled flexibilities. Actually, we cannot even come close to them; we have to stay away from them. So omega cannot be too high, which means KP has a limit, but we want to achieve the highest KP.

So what is the relationship between KP, KV, and those performance? So from those two equations, we can write KP is M omega square, and KV is M to zeta omega, right? Just to rewriting these two equations and computing KV and KP. So when we are controlling a system, we are going to set what? We're going to set, really, the dynamics of the system, which means we need to set zeta and omega. So we set zeta and omega, and we can compute our KP and KV. Most of the time, zeta is equal to one. So KV is M to omega, and so all what is left is to set omega. So for 400, omega is equal to what? In the case of the robot in this simulation, we have 400 KP. So omega is equal to? Come on.

Student:[Off mic].

Instructor (Oussama Khatib):Square root –

Student:[Off mic].

Instructor (Oussama Khatib):Divided by – well, M is equal to 1, let's say, in that case. It's 20. It's 20 multiply – what is the frequency, the real frequency?

Student:[Off mic].

Instructor (Oussama Khatib):Omega divided by 2 Pi, right. So what is your frequency about – let's say divide by 6, 20 divided by 6. So it's very low, 3-4 hertz. In fact, if you're lucky, you can go, well, to 10 hertz. I mean, this would be great. So when we go to 1600, this is really nice, 40 divided by 6.

Well, in practice, you start with very low gains, and you start turning your gains up, up, up, up, and suddenly you are going to hit that, noise start to vibrate. So go down, but we will see some ways of doing this in a more precise way, but, again, what you are seeing here is KP and KV – now, if we think about two different links, one link that is heavy, and one link that is light. M equal 1 and M equal 100. Your gain KP is going to be – for the same frequency, is going to be much, much bigger for the bigger link. So that gain is scaled by the mass, and because it is scaled by the mass, we can think about the problem of setting the gains for the unit mass system.

You remember we said if I'm moving Joint 2, the inertia of Joint 2 is changing, big, small. So we need to be able to somehow account for the fact – so I set my frequency; I set omega and set zeta, and now I computed KP and KV, but M doubled. So I need to update my gains, right? If I want to move with the same closed loop frequency, I need, somehow, to update my gains, and that becomes nonlinear control. So we talk about the unit mass gains. So let's just imagine that your system, this mass was unit mass. Your gains will be simply omega square and 2 zeta omega, which is for one, this would be 2 omega. Very simple, just set omega and you get your KP and KV. Okay?

But we know the system is not going to be a unit mass. So for this M mass system, what are the gains? Gains from this KP prime and KV prime. What would be KP for M, a system with mass M using KP prime? KP will be M times KP prime, and KV just linear. So you take M, and you scale your gains, okay?

Well, what is the big deal about this; why I'm talking about? Well, the big deal is that M is going to change, so even for one changing mass you can make this nonlinear, and scale and track a constant frequency and constant damping ratio, but for a system with many degrees of freedom, we have a mass matrix, and we are going to use the same concept.

We are going to say I look at the unit mass system, and then I scale the unit mass system with the mass matrix, and everything will work exactly in the same way, and I will be compensating for the variation of the mass. This is the nonlinear dynamic of the coupling that we're going to introduce, and it is based on the idea that I design the unit mass system, and then I will scale the unit mass system with the mass matrix. Well, in this case, it is just a scalar, simple mass.

So this is what we call the control partitioning. If I have a system with a mass M, I basically – the composite in the mass and the unit mass system. So the blue is the unit mass system, and M is the scaling of the unit mass system. So I can now design a controller for the unit mass system with KP prime and KV prime, and then the KP and KV for the original system will be just scaled by that mass.

So here is my controller F. I'm going to write it as M time F prime where F prime is this quantity, a TD controller designed for unit mass. So we always denote this as primes of KV or KP. So when we say prime, we are talking about the unit mass system. The controller of the unit mass system F prime, and F is M times that F prime. That will make more sense when we go through the multi degree of freedom controller because M becomes the mass matrix, okay?

So, essentially, we have our initial system that is now controlled as a unit mass system scaled by the mass itself, and the behavior of the whole system is like this – well, the dynamic behavior, the dynamic response and the damping ratio are like this, but we have to be careful about other characteristics like the student's rejection, stiffness; they are not, and we will see that in a second. The dynamic behavior of the closed loop is like this.

So you design your controller for the unit mass and basically, if you scale with that mass, then you have the behavior of the unit mass, okay? So, in this case, what is omega for the system? It is simply the square root of KP prime, okay? And now we are going to introduce one more element. We talked about it Monday, and this is just a tiny nonlinearity. Let's add some friction.

So we started with the system without any nonlinearity, and now I'm just adding a little bit of friction, nonlinear friction, like some stiction on that joint. So the equation changed completely. That is, it's not nonlinear anymore. We cannot just treat it as a linear system, and we have to deal with a controller that is going to be nonlinear. So how can we deal with this? Come on, ideas.

So you have your joint, and it has a gear with, like, some friction that is – or even it has some gravity or whatever. Yes.

Student: So if you've got a certain type of friction, you can, like, if it's velocity, then you can put that into the motion equation –

Instructor (Oussama Khatib):Um, hm.

Student:- and change your V value, your KV.

Instructor (Oussama Khatib):KV, you mean.

Student: Yeah, yeah.

Instructor (Oussama Khatib):The KV. So if it is linear, yeah, I think you can, in fact, integrate it directly into KV, but if it is not nonlinear, like just the gravity. So what do we do – if we have the gravity, what do we do with the gravity? We model it. I know the model because I know the mass, the center of mass, all of these things. So if I can model it, I can somehow, like, anticipate what the gravity is going to be and try to compensate for it, very good. So we can compensate for the gravity.

Well, if we have a nonlinear term, what we will do is we put that compensation in the controller. So now the controller, it has the linear part which was F prime alpha F prime. Alpha F prime actually is mass F prime, and now we are going to add another term, beta, which will attempt to compensate for B. You do not know B exactly. You know, sort of, a model with some estimate of B. You don't know X exactly. You don't know X dot exactly. You have estimate of these, what we call the X hat, X dot hat, and B hat. Now, B has a structure. If it's the gravity, it's going to be, I don't know, ML cosign that angle, and you can estimate your mass, estimate your length, estimate the position and come up with an estimate of B, which would be B hat.

So, in that case, you can say alpha is simply the mass, an estimate of the mass, minus/plus one gram, probably you will find it, and your B hat is going to be an estimate of B given the state, your estimate of the state, and you'll probably have ten epsilons, little bit more of error. So we're assuming that we are going to have some errors, but by compensating for those nonlinearities, estimating the gravity and taking it out, later estimating centrifugal coriolis forces and trying to taking them out, we should be able to bring the closed loop system closer to a system that is a unit mass system because with this compensation, if everything was perfect, we compensated perfectly B, then basically beta will take out B. For each configuration, each velocity, beta is exactly compensating for B; it takes it out, and the system is linearized, right? Well, this will never happen in reality, but we will be very close.

So this is what we can write. We can say this is our system, and this is the controller. You understand this controller? This controller is a nonlinear controller, but it is attempting to render in the closed loop, your system, to become the coupled linear system. So here's the result. If B and B hat were identical – if B hat was compensating perfectly for B, and if the estimate of the mass matrix, later this mass was identical to M, then your system will behave this way. So what you designed for F prime will be part of the closed loop of the whole system. We're talking about 1 degree of freedom, but if we are – later we will see 20 degree of freedom, it would be the same, okay?

Well, here is how we can write this system. So our system was F with the output X, X dot, the state. Basically what we are doing is we are looking at the model of the system, and we are using X and X dot to estimate B, the nonlinearities in the system, and compensate for them. So F is going to have a component, which is B hat. In addition, our input control, which is F prime, is going to be scaled by an estimate of M, the mass of the system so that there is a virtual system here that would look like a unit mass system with an input F prime and this same output, and this big box, the red box, is like a system that is linear with unit mass, and that is the purpose of this design. Later, this will be centrifugal coriolis gravity forces, and this would be what – right, the mass matrix.

So, in fact, with many degrees of freedom, we will be able to do the same thing where this becomes the mass matrix, and here we will have V and G. You remember V? Centrifugal coriolis, and G, gravity, and you can add the friction as well, okay?

So, essentially, we are designing a nonlinear controller to compensate for centrifugal coriolis, gravity, and to decouple the system, to decouple the masses, the inertial forces, and to achieve a unit mass system behavior.

Okay. So let's see our design for F prime. F prime is in this structure, in the decoupled controlled structure, and if you have a desired position XD, what would be F prime? Just a goal position, so our goal position, we have X desired. F prime will be minus, minus something. Who remembers? I'm sure you remember. F prime is?

Student: Minus KV prime minus X dot minus KT prime times X minus XD.

Instructor (Oussama Khatib):You meant minus KP prime, X minus XD. So minus KV prime X dot minus KP prime, X minus XD, and the closed loop behavior would be very nice. So we linearized the system. All right. Well, most of the time you're not just going to a goal position. Most of the time you are tracking a trajectory, and on this trajectory you might have, like, you might have different accelerations at different point. You have different velocities, and whereas in this controller, we are just reaching through the goal position. KP prime is trying to reduce the error, and KV prime is trying to put just damping to bring the velocity to zero at the end point, but if you are tracking a trajectory, you have all of these desired things. You have desired position, function of time, desired velocity, and desired acceleration.

So we need to design a controller that is more suited for this. So what F prime would be? See, now we forget about the system because we know we can decouple it, make it linear. Let's think about the unit mass system, how you would design a unit mass system controller, and then you put it in that structure. So what is the objective if you have all these desired things? What should F prime be?

Okay. So you see on the top here is F prime. I have some desired acceleration. I have my acceleration, unit mass acceleration, equal to F prime, and I know my desired acceleration; it's X double dot desired. So if this was really a perfect system, and you are trying to track this acceleration desired, what F prime should be? I think the question is so simple that you cannot believe it. Come on, this is very simple, too simple. So my system is X double dot, and I know the desired acceleration, X double dot desired. What should F prime be? Come on.

Student: Minus the cost of minus X double dot, minus X, E double dot.

Instructor (Oussama Khatib):Yeah, I think you went too far. That is correct, but I'm just saying if the system was able to respond directly to F prime with no errors, nothing, and my system is X double dot, and have the desired acceleration X double dot desired. What I would do with F prime, just make F prime equal to?

Student:X double dot.

Instructor (Oussama Khatib):X double dot desired, right? Right? Okay. Okay, you see what we're talking about? You have your acceleration desired, so just put X double dot equal X double dot desired, and everything should just – you apply this force, and the system should follow X double dot desired, right?

Well, it won't. It will drift because there is really no feedback. You have your acceleration, and you are saying X double dot desired, this is my acceleration desired, and as soon as you start, the system will start accumulating errors, and it will drift. So what should we do? We should do the PD part, and that's why now we are going to add proportional control to the error, the position error. As you said, minus KP prime, X minus X desired.

What about the error in velocity? Because now I have X dot desired. What would be the term that I should use to follow X dot desired? So that would be minus KV – could you finish it? Minus KV –

Student:X dot minus X desired dot.

Instructor (Oussama Khatib):Exactly, from the error, X minus X dot desired, and I will – so here is the controller. So this time, if I have the full trajectory, I will form errors on the position, on the velocity, and I would feed forward the acceleration. So essentially, you are telling the system follow this desired acceleration. It's not going – there will be errors, and I'm tightening these errors. So the closed loop behavior of this is going to be controlling the error in acceleration, in velocity, and in position, if I have the full trajectory in time, and that will, basically, if I call X minus X desired the error, then I'm really controlling the error as a second order linear system, all right?

Okay. So now, we have to make sure that we can do this with the whole robot, and we have to make sure that this controller could work with those gains that we are trying to achieve, and we start analyzing the system. So let's imagine that I designed the system, the compensation, with the B hat – I'm sorry, they are not appearing as hats, but this is B hat and M hat, and I get everything over there, but then – now we are talking about the real system. So when we were running the simulation earlier we saw that a small external force will disturb the system. So there are a lot of forces coming from the errors in dynamics, errors in the gravity estimates, nonlinear forces coming from the gears and the friction that will affect this behavior, and as we start introducing disturbances in the system, we are going to see that these gains that we set are going to play a very important role in disturbance rejection.

So let's add a little bit of disturbance here. So if we add some disturbance, going to take a very simple type of disturbance like a bounded disturbance that we are adding from some, like, type of error in the gravity. Imagine that you have this link, and you have a little disturbance coming from the gravity. So what is the affect of this disturbance on the closed loop now?

So here is our controller with F prime scaling it by the mass estimate and B estimate, we are getting this closed loop. So this closed loop now is going to be equal, not to zero with the disturbance force, it's going to be equal to this disturbance force divided by M, right? By the way, in some textbooks, this is not divided by M; it's directly applied as if it was applied to the decoupled unit mass system.

So because the disturbance is an input of the system, you have to remember that we are dividing by M, so this is divided by M.

Okay. So let's see what is going to happen to your errors. I'm not sure how many of you know what we mean by a steady state error. What is a steady state error here? What do we mean by steady state error? Yes.

Student: It was [inaudible] equilibrium and velocity, what its position error's going to be.

Instructor (Oussama Khatib):So a steady state is like when the acceleration starts, the velocity starts, and you reach that position, but you are not reaching it exactly; there is a small error. So it's like a spring mass system if you apply a little bit of a preservation with an external force, it's not going to go to its rest position; it will be very close but not there, and that would be the steady state error.

So this is what happens when we have the velocity and acceleration errors equal to zero, and that means the last term, the error term KP prime is equal to F disturbance divided by M, and that means your error is going to be the disturbance force divided by M, KP prime. Again, in some textbooks, there is no M appearing there, and this is very important. KP prime is the gain, the position gain for the unit mass system. It renders your frequency constant over all the motion. When we vary M, KP prime, your frequency is square root of KP prime.

However, your stiffness, your position gain, is M times KP prime, and if M is changing, your stiffness is varying. So you do not have constant stiffness over the workspace as you vary M. So your error, your steady state error will vary depending on where you are and how big your M is, and you better get a large KP prime so to guarantee that you have minimum disturbance rejection. So I'm not sure if you are seeing it. KP is your stiffness, is your closed loop stiffness. KP prime is not; it's M, KP prime. Okay? So this is very important, remember this because we will use it when we go to N degrees of freedom. It is the same structure.

Okay. Well, here is the example I mentioned earlier. If we think about just a disturbance for this one degree of freedom, it's sort of a spring mass, and you are applying the disturbance force, and that means the steady state error, X minus X desired, will be given by this equation, which means you are going to rest, not at XD but at XD plus this force divided by the stiffness you have. So this is your delta X.

Okay. So the disturbance is going to produce an error delta X that is given by the disturbance divided by your stiffness, and this is your M times KP prime. Don't confuse KP prime for the unit mass system and the KP.

Okay. Now, how can you get rid of that error, the steady state error? Yeah, I know. I hid that slide, now you know. You don't know. All right, go ahead.

Student: Add an I [inaudible].

Instructor (Oussama Khatib):Do you think he didn't see the next slide? No, you didn't see it. All right. Okay. Why by adding I we can remove that error?

Student: Because over time, the I grows large, if you have a steady state error –

Instructor (Oussama Khatib):Um, hm.

Student:– and it'll correct for that.

Instructor (Oussama Khatib):Yeah, basically you detect the error, and as long as you have an error, you are adding, adding. So, essentially, the idea is now your F prime will include an additional term that magnifies this error with some gain, and keep magnifying and keep adding until you overcome that disturbance. Now, integral action is very good to reduce errors, but you have to be careful about the way you use it. Better to use it close to your goal position not all over, especially if you are moving fast and accelerating, winding and unwinding, that integral might create instabilities.

So the way you can analyze the disturbance is that now you have this equation, this closed loop, and if you take the derivative of this equation, basically, you see that now the steady state is going to be equal to zero. So if you build that integral, then you will take the error to zero. Actually, integral action is very nice when we go to force control. We will talk about it later. In motion control, you have to be very careful about its use.

There is another element of this when we look at the simulation of the puma we are looking from outside, but let's go a little bit inside and see what is happening. So you have the motor. You have a gear with some gear ratio that depends on the diameter of those two spheres R and little r. So what is the gear ratio here? Come on, we have a lot of mechanical engineers. Is it big or small? So what is a typical gear ratio for a robot? What do you think the gear ratio for Joint 2? When I move Joint 2 here, what is the gear ratio? Hm?

Student:[Off mic].

Instructor (Oussama Khatib):That's reasonable. Actually, the gear ratios for the puma vary between, like, 50 and 150. Some robots has 300 gear ratio, and they have multiple stages. All of the joints here, like, 1, 2, and 3 have two stages, and you get a lot of flexibilities, a lot of vibrations that appears and all of that. So it's actually very nice here because we have a very small gear ratio, but I’m not sure if you know what is the gear ratio; what is the gear ratio? About how big be the gear ratio here? 20, 40? No one knows?

Okay, the motor is going faster or the link is faster?

Student: The motor.

Instructor (Oussama Khatib):I’m sorry?

Student: Motor.

Instructor (Oussama Khatib):The motor, so the gear ratio is reducing the speed; the link is going slower. So how slower? One time, two – twice, threes? He said, like, two.

Student: It looks like two.

Instructor (Oussama Khatib):Yeah, it is two. I was worried. Okay. So the gear ratio is really R divided by the other r, the smallest r, okay? So that’s your gear ratio, and the speeds, the link speed is smaller by that ratio than the speed of the motor, right? How about the torque? Well, as it’s written, so we have a gear ratio of two here, the torque as the link is twice as big as the torque of the motor. So you take a motor, if the gear ratio is one, basically, you’ll have very drive, and you need a big motor to produce a torque, right? But this big motor is heavy, and with the robot, putting a heavy motor here is a problem. So what do we do? We put higher and higher gear ratio reducing the size of the motor, and we can achieve the torque because we are using a high gear ratio.

Now, I wanted to show you this to illustrate another problem. It’s not about just the speed reduction and torque increase; there is another effect that comes with the gear ratio. When we start moving, so we have this inertia. It’s vying with this, right? Big inertia, small inertia as we move. Now, the question is what this motor is going to do to this inertia? The inertia of the link is here. There is another inertia coming from this rotation of the motor, and this inertia is going to be affected by the gear ratio, and the effective inertia perceived of this joint is going to be bigger than the link inertia.

So what is the effect of the inertia of the rotor of that motor there on the inertia of the link? So the inertia of the link alone is IL. The inertia of the motor is IM. What do you expect the real IL, the effective IL to be? So it’s going to be equal to IL plus something, and this something is? Again, you?

Student: Nuh, uh.

Instructor (Oussama Khatib):Oh, him, okay, him.

Student: Two, twice.

Instructor (Oussama Khatib):Twice?

Student: Twice the inertia of the motor.

Instructor (Oussama Khatib):Twice the inertia of the motor, so you mean, in general, this is N times the inertia of the motor.

Student: No, it's –

Instructor (Oussama Khatib):That's what you mean.

Student: Inertia.

Instructor (Oussama Khatib):Yeah, so it is N the inertia of the motor, which is similar to the torque increase. The torque at the link is N times the torque of the motor. You're saying the inertia is the same, in the same proportion it is N times the torque, I mean, the inertia of the motor. Well, you're right, it is bigger. You're right, there is an increase by N, but it is not linear by N. Anyone can give me a better estimate? Now you know, but they are not looking. That's why no one is able to tell me.

So inertia of the motor is reflected at the link by – it's not a torque. It's not going to be reflected by N because the motor is moving much faster. The acceleration at the joint is moving also slower by that gear ratio. So the effective inertia reflected by the motor is? How many times IM? N square, much better. So you have a gear ratio of 100. It is big. The coefficient that you are going to carry is really big, and that makes the robot very dangerous, actually, because the effective inertia that you see at the joint is correlated with the impact force that you might reduce if you have a sudden collision, and if you are reflecting this small inertia by N square of the gear ratio, you're going to produce a large impact force.

So here is that nemesis. We can write the dynamic equation on the side of the motor, or we can write it on the side of the link, and when we write it on the side of the link, we see that it's IL plus N square or eight a square IM, and this is your effective inertia. Now, your effective inertia is always by the square of the gear ratio of the motor you are using. Well, the direct drive case is really ideal. We use it with SCARA-type robots where you do not have to carry the gravity, but with the robots that are articulated and that need to move in space, it's very difficult to build robots that have big motors and can carry the structure.

Now, variation of the effective inertia makes it that if you have variation of your IL – IL is going to change. IM is constant; it's the same motor. So the question is if IL is changing, what is the effect of that? Well, the effect of that is changing your time response because your KP, if you remember, is M times KP prime. So it affects, directly, your omega. So you might do an analysis and see if you are using constant gains, but the best way to select your gain is to go to the geometric average and look at your minimum – the minimum value of your IL and the maximum value of your IL, and take that midrange in geometric sense, and that gives you some estimate that you can use over all the range of motion.

Okay. Well, I think we are coming to a point where we can break. So let me remind you that next Tuesday and Wednesday we will have the review sessions, and on Monday we will be signing up for those review sessions. So please, those of you who are not here, please make sure you come and sign up. See you on Monday.

[End of Audio]

Duration: 74 minutes

<html>
IntroductionToRobotics-Lecture16
</p><p>Instructor (Oussama Khatib):Okay. Well, we have a special today. This robot is going to show you a video. I'm not sure if you're familiar with those cassettes anymore. Do they exist still? So the [inaudible] here is going to start the VCR. Starting, and there is a video inside.

</p><p>So this video is about a very important aspect of robotics, which is compliant motion. You see, the sponge is pushing up, and you see no reflection on the sponge, right? That means there is no force applied. Here, we are coming to a surface that is unknown, and the robot is sliding over the surface. So it's making contact at different points, even if we remove the whole object. Now here is a wavy surface that is being followed just by saying press down and move to the right, cleaning a window without breaking it. It's very important.

</p><p>All of this cannot be done without force control and compliant motion control. So you put an error, and still you are doing the same task. Dealing with uncertainties requires you to be able to control the compliance. This is linear compliance. We can also do rotational compliance. So zero moment. When you push to the left or to the right, you will have a zero moment.

</p><p>If we could see the video. Here we are creating a strategy to do face-to-face assembly where the compact forces will drive the robot to move about a rotation center here. That will result in a robust strategy to do face-to-face without any planning of the trajectory.

</p><p>So from any contact point, you rotate, minimizing these moments of contact. This is another example of a peg in a hole where the forces of contact are driving the motion to rotate and to insert the peg into the hole. These were developed in the late '80s. Here's a very nice example. We're following this [inaudible] without any specification. Just the contact forces are driving this motion.

</p><p>So force control is very important, not only for compliant motion, but when you do cooperation between multiple robots, it is also the same requirements. You need to be able to control internal forces, otherwise you will break the offering. Here is an example of two pumas cooperating to manipulate a pipe. I believe we have three pumas. Actually, sorry, two pumas. We are moving a large object.

</p><p>You can imagine, now, you are doing internal force control and resulting force control to create the compliance, to produce that assembly. So you need to control internal forces, and there is a model called the Virtual Linkage Model that we developed to allow us to capture those variables that you have internally. For the first time, we were able to produce four-arm manipulation. You have three, and you have a driver.

</p><p>You can see resulting compliance motion, and also internal force control to maintain those contact points and to manipulate this object that we call augmented object. All of these issues, we will cover in advanced robotics, later.

</p><p>So I'm going to stop here. The tape is very long, but it gives you and idea about the issue of compliance. Let's just take a case of just one degree of freedom. You have an inner shaft, some displacement, and you have a force. We come to this all the time by decoupling the system. So if we think about the controller F-prime to be just proportional controller. So you are controlling X, Y and Z. You have a KP prime, KPY prime and KPZ prime.

</p><p>So what is the behavior? You push here, and you're going to feel some stiffness, right? This stiffness is going to come from your mass times AP prime. It's the overall stiffness. If I want to do compliance in the Z direction, I would like to push up the robot. I want the robot to move with the force I'm pushing with. What can we do?

</p><p>So if I have a very large KPZ, it's going to be very stiff, right? If I

reduce KP, what is going to happen? If I push, it will restore itself. I push, I reduce it, I reduce it. It will be a stiffness that is lower and lower, and you will basically move in that direction, right?

</p><p>If you make it zero, what's going to happen?

</p><p>Student:[Inaudible].

</p><p>Instructor (Oussama Khatib):Well, if it is zero, it is basically [inaudible]. That spring is cut. So you move it, it's going to move. From the control we developed for the position, because we have a control in X, Y and Z in task space of [inaudible] is stiff in this direction, is stiff in that direction. In this direction, if you change KP and make it zero, now it's free.

</p><p>Just by making KP equal to zero, you are going to have those relations – so the first one will be stiff, in the Y direction stiff, and here, you will just feel the Z-dot. So can we create compliance this way? Just compliance, now it is free?

</p><p>Actually, what we want to do is not only to create a compliance, we want to control the contact force. So if I'm controlling pushing the robot, I would like it to maintain zero force. So when it feels there is any force, it moves away. Which means we need feedback. We need a force sensor. This is really the requirement in order to interact with the environment.

</p><p>Most of the time, actually, you want to apply a specific force on the environment. That requires control of forces. Here we have only – we just remove the position control from one degree of freedom, one direction. So this compliance along the Z direction is the first step. We substitute this controller in this direction with a forced controller later, and then we are able to control forces in the Z direction, as we saw earlier. Position is in the other direction.

</p><p>So this is what we call both motion control together with force control. This KP prime has a very important role in determining the stiffness, but really, the cause of stiffness, you have multiplied KP prime by MHAT. So this is your cause of stiffness.

</p><p>If you think about it in terms of the spring aspect, you can see the KX is your stiffness. Now, what is the corresponding stiffness in joint space? You can compute it because this KX is displacing delta X, and delta X is J delta theta. So your K theta, the stiffness you have in joint space, is your KX with this transformation. So you can even evaluate your corresponding stiffness in joint space.

</p><p>Now, really dealing with the problem of force control, you need to be able to sense the force. The reason is, if you think about it, let's say I would like to control some F desired. If we set just the control force to the desired force, we will see that the robot is not even moving. The reason for that is the friction. So if you think about the friction, as long as your force design is within the breakaway friction, you're not going to move. Your force will not get the robot to move beyond that friction.

</p><p>Then you have, also, continuous column friction, and you have the viscous friction. So it is really difficult to do accurate force control. All the time, you are feeling those forces that are coming, so you cannot do it [inaudible]. So what you need to do, you need to measure your resulting force at the output. So if you apply one [inaudible] in this case, the output will be zero.

</p><p>You really need to be able to measure the actually F and foreman error between your desired [inaudible] and your actually force. Then you can make feedback control. Then again, you will be able to do it. So if you have a sensor, then your sensor is going to give you this sensed force, which is the displacement of the sensor.

</p><p>A sensor is essentially a stiffness that you're deforming. It's very stiff. The delta X is very time. So your sensor will give you the information, and now you can see that what you want is to achieve F equals F desired, which means that you can select a control of that [inaudible] forward that desired force, and deals with the error and the feedback through a controller that allows you to control this mass that is moving, but internal forces.

</p><p>Now, we can then use this relation between X and F and provide this equation in terms of force control. So we can take the second derivative of this in terms of forces. Then we will have this stiffness appearing in the equation. Then we close the loop. Once we close the loop, will are going to have the responses we saw on the video. That is, when we push, we will see not only the feet forward, but there will be an error coming from the sensor. This error will produce [inaudible] to move the robot in a way to reduce this error.

</p><p>So if you think about how we can do this, basically you have the relation between forces and displacement. You take the derivatives, and then you design a controller like this with a proportional term, derivative term, and then you are able to have a closed loop that minimizes this. The response now is in forced space for the robot.

</p><p>I'm not expecting you to understand all these details, but essentially, force control is going to be a very, very important aspect of robot control. We just dealt with the motion control. The question is now we combine the two, how we get the robot to apply a force in some direction while moving. If you are cleaning a surface, you need to be able to move, to control directions, and also to control the pressure. So you have to combine these two.

</p><p>The result is a sort of unification of the task control in terms of motion control and force control merged together. So there are directions of force control, and there are directions of motion control. These depend on the relationship between the objects you are assembling.

</p><p>So if you have a sphere, and if you want to put the sphere on a contact, you lost a degree of freedom. You cannot move in the Z direction. You can still slide on the XY plane. So how many degrees of freedom of motion do you have now? Imagine the ender factor is holding the sphere on the surface. How many degrees of freedom do you have? Two?

</p><p>Five. Yes, two in position, and then you still do rotations. You still have those rotations. So you lost one degree of freedom by this contact constraint. How many degrees of freedom do we lose here? Two. We lost the Z axis, and we lost the rotation about the Y axis. You cannot rotate about this axis.

</p><p>So you see, every time you have a different shape, you are going to lose different numbers of degrees of freedom. Here you lose three. So what does it mean? It means that we really need to evaluate those space where we can move, but at the same time, did we really lose those degrees of freedom?

</p><p>If I'm pushing here or controlling this movement, I still can do it. I can push with ten newtons, 20 newtons. So the degree of freedom just went to the space of forced control. In this contact, if you are talking about controlling that variable, the force, then you didn't lose the degree of freedom. It just went into the constraint space, and you can control that force now.

</p><p>We can describe the directions and the relationship between the two and separate the spaces. Here, it's very easy. You can say in X and Y, there is no rotation about axes X and Y. You cannot rotate this about X and Y. You can rotate it about Z.

</p><p>In the Z direction, you cannot move, but you can move in X and Y. So now the space is split into two parts. To split the space, we go to a description of this space, and we split it through the omega matrixes that allow us to project the

motion control in some subspace and the force control in another subspace.

</p><p>In this case, it's very simple what directions you can control forces. The Z direction. So essentially the motion control is in X and Y, and the Z direction is the compliment. If we call omega bar is the subspace for force control.

</p><p>So the result is a unified framework that we studied so far, that was this loop. If I have a motion, I can control it, and I can go to J transpose, and I can compensate for some difficult [inaudible] and do this loop. Now what we are doing, we're projecting these controls in their spaces, and we're doing unified control of motion and forces.

</p><p>Because we are controlling everything in the task space, we have F motion, F forces. We just add them together. We have a total force that we transform into a torque through Jacovian transpose. That is what is nice now. You unify all the characteristics.

</p><p>So you can see some lambdas and some other things. The lambdas is essentially the mass property because your mass matrix, MX. This is the mass properties that you are using to scale your controller to the [inaudible] system. This is then also in the force control loop.

</p><p>The result is really interesting because you can uncouple the two systems in their own subspaces. Then you can control them properly. So this is a little bit of the beginning of the introduction of advanced topics that we will see. I'm going to continue with this discussion, but you really have to deal with this very important problem. If you want your robot to interact with the world, you need absolutely to be able to control the contact forces. Contact forces doesn't mean only linear forces, but it also means you control moments.

</p><p>With moment control, you are going to be able to achieve a lot of affects. In any assembly, you're going to have some error in the relationship between the two. How do you deal with this? How do you deal with the fact that when you are pushing down, you're going to have a moment generated. You can use it to drive – I'm not going to drive it too far. It would spill the water, but you can use it to drive this rotation.

</p><p>The action force here with rotate if you select this point as a center, as your operational point. If you push from here, it's rotated. This will be fixed. You can fix it by controlling this point. Now you are rotating about the spot. We use this a lot in doing assemblies between objects because we are able to cope with the errors and uncertainties that we have. We know, to some extent, the relationships, but we don't know exactly where things are.

</p><p>So I'm going to continue with the discussion of advanced topics. This is part of a talk I gave recently. It starts with some really old robots. I don't know if you've seen – have you seen this robot, Leonardo? Okay. Tell me if you have seen this one.

</p><p>Can you believe this? You have a robot that can draw. 1773. Unbelievable. Actually, there were a whole family of robots that draw, that play music, that can write at that time. How can you make robots like this? How do you program this robot with the technology of the time?

</p><p>So basically, you're going to use the mechanical devices. If you want to see the computer that was used, here is the computer. So you have springs, and then you have to drive the motion. You can see all these sets of design trajectories on which you are moving the different parts. So the robot is going to write [Speaking foreign language].

</p><p>So robotics was around for a long time, right? Really, the question was just waiting. It was in our minds. We wanted to build those machines, and the thing was that we didn't have the right technology. The technology really came much later and

brought the first robots that were almost completely industrial robots that were used in this structured plant to perform repeated tasks in the automobile industry and others.

</p><p>Now, today, robotics has moved, as we saw at the beginning of the class, that robots are now used or the application of robots is conceived in many domains. Especially, we see a lot of robotics today in medical applications where robots are coming very close to the human, in fact, inside the body of the human.

</p><p>So this is bringing a lot of challenges. As we bring robots closer and closer to the human, this is bringing the real challenge of robotics. Robotics in the industrial setting requires really little from the robot. Once we understood the requirement, precision, repeatability and the performance and speed and robustness, we can engineer the machine to do that.

</p><p>Here the problem is we need much more intelligence in the robot to deal with many things that are not known in advance. You need to perceive the environment quickly, respond and react to everything that is happening, and you need to be also moving safely.

</p><p>So those challenges bring the perception and sensing issues that we need to deal with into an environment that is unstructured. This is a big challenge and also a good thing for researchers. Without this challenge, robotics would've become just automation. But having the challenge of an unstructured environment is bringing a lot of interesting issues to the research in sensing and perception, in planning and control. Why is planning and control hard here when you have those human-like robots? What is the problem?

</p><p>Student:[Inaudible].

</p><p>Instructor (Oussama Khatib):Well, in the planning, the world is changing all the time, and there's maybe something else you wanted to mention?

</p><p>Student:[Inaudible] solutions?

</p><p>Instructor (Oussama Khatib):Well, the number of degrees of freedom. You have a robot with not six or five anymore. You're talking about many degrees of freedom, and you need to resolve and respond in real time. You need, in those machines, to have – not just move to a position. You need almost human-like skills. So you are demanding much more.

</p><p>You need to deal with the human-robot interaction, which is going to involve both physical interaction, touching the human and working with the human, but also the communication, the interaction, the cognitive aspect of it. You need to build those robots, and you need to make sure that they are safe. You need to make sure that they are capable but safe.

</p><p>So a very important theme in this area is the interactivity of the robot and also the human-friendly design of the robot so that the robot can really be integrated and working with humans.

</p><p>So if we think about this problem and the challenges, you can see on the left – you can see a robot you don't want to be next to. It is a problem. On the right, you can see a robot – you might think, well, it's smaller. It might be safe. But actually, the danger in this robot is hidden inside. It's hidden inside, and we talked about it in class. You remember this. You remember this inner shot of the motor. This is going to be reflected.

</p><p>When you're going to have an impact force, you're going to really produce a large impact force because you are going to see the inertia of the rotor of the motor. So there was a lot of development in the area of making robots lighter, safer, and one of the most beautiful ones, you can see it on the top. This is the light arm from DLR, the NASA in Germany.

</p><p>They have been working – in fact, we had a lot of collaboration with them in designing torque sensors so that we close the loop at each of the joints. They developed a very nice mechatronic technology that resulted in this very beautiful robot. Now you can have compliance. You can control the robot to be compliant at all the joints.

</p><p>There is still a problem, the fact that the open-loop characteristics doesn't have time – makes it that you don't have time to close the loop. There are still open-loop characteristics that will be reflected during impact, which might make the robot dangerous.

</p><p>So perhaps one of the safest ways is to use elasticity with the actuator. Then when you have an impact, the elasticity takes the inertia. This is not going to give you the performance, and that is really the problem and the challenge. So if we think about the problem of collision, we see that essentially you have a robot moving at some speed. It has some stiffness. The environment has some stiffness, and when you have a collision, you are going to have an impact force.

</p><p>This typical problem was studied in the automobile industry, and they came up with a criteria call the head-injury criteria. The head-injury criteria measures the risk of injury, given the impact forces. In fact, if you think about it, in this plane, where we are looking at the fact of inertias and the stiffness, we can see that a Puma robot is sitting here at almost 90 percent risk of serious injury.

</p><p>The only way you can make it safe is by covering it with almost 20 centimeters of compliant material. So it will be really, really big.

</p><p>What is the problem? I know we have a lot of mechanical engineers, so they are going to help me now to solve this problem. How can we deal with this problem? Yeah?

</p><p>Student:[Inaudible]. Less reactive and more –

</p><p>Instructor (Oussama Khatib):Actually, the sensor idea is really, really good, but I think there is no way you can guarantee – if you have a robot, you can never guarantee passive safety that is not dependent on a computer or dependent on a controller or a sensor. Anything can break. How can I guarantee that whatever happens, this robot is safe?

</p><p>So I need to really go further. Not only to think about how I can improve the feedback or what we can do with the sensors, the skin and protection of the robot is good because that is most of the operation. But I also want to make sure that the if I have an impact for some reason, something broke in the robot, the controller, the computer, something happened. I need to guarantee that the impact force is below some acceptable amount.

</p><p>So in turns out that a large part of the design comes from the actuation. You can see it every time – you are trying to manipulate an object with this ender factor. How do you move this ender factor? You need lengths. Everything we were studying was the carrier of this ender factor, right? To actuate your lengths, you need motors. Because of the motors, your lengths become bigger and bigger because you put this motor here to carry this ender factor. Then you need to carry both with the length with another motor here, and you propagate the weight.

</p><p>So every time you design a robot, you're hitting this problem. How can I squeeze a lighter actuator here? So you put higher gear ratio, but you reflect larger inertia. But essentially the torque need here is very critical.

</p><p>So you come up with some specific, and you say, I need the 3.7 newton motor with continuous torque, a torque that you can sustain for a long time. A motor has a peak torque that could be reduced in a short time. If you keep applying it, you will burn your motor.

</p><p>So once you've designed your motor, and now you can lift the gravity of the link, ender factor and the load and all of that, what do you say? My robot has to accelerate, has to give me those performances, and now I will pick my motor to have those characteristics. So I will need strength to carry, and I need also the dynamic responses.

</p><p>Actually, this is what you need. You need performance. What you are doing here is you pick a motor, and then you say this motor has to be performance – as the acceleration I'm going to produce. This is really not necessary. If you think about the problem in the domain of control and the performance you need, the magnitude that you will need at high frequency is much reduced. So in fact, your requirement is you need large torque at low frequency and smaller torque at lower frequency.

</p><p>You can design your robot differently. We never think about it, but we can design the robot not with one motor but with multiple motors for each joint. Like humans. Humans have multiple muscles and different types of muscles actuating. So you can place a small motor. The structure will be light and very safe, and you put a big motor at the base. You remove the weight.

</p><p>But if you do this, this motor will be reflected at the impact force. I say this motor is only needed at low frequencies. So now, I can isolate the motor when I have an impact force. It will go through this elasticity.

</p><p>So this is the concept we development in DM2. It's called Distributed Micro Mini Actuation that really is interesting in the sense that you build a robot that has the capacity of a large robot but the safety and performance of a smaller one.

</p><p>This concept, wince, went much further, and seeing the performance you can achieve, it's amazing. You can take the Puma and build a robot like the Puma with ten times less effective inertia. You reduce your inertia from 35 to 3.5. That leads to a large reduction in the inertia properties that you see.

</p><p>I think you don't understand this. Let me show you what I mean. Where is my simulator? Has anyone seen the Puma simulator somewhere? Right here.

</p><p>Okay. So if we look at the Puma, we can display the inertia property of the Puma. You can see you have small inertia in this direction and large inertia in this direction. In fact, if I move the Puma, you can see that this inertia is changing. Here we have singularity. You remember the singularity here.

</p><p>If we look at it in this direction, we cannot move when it comes t overhead singularity. The inertia becomes very, very big. The inertia goes back to a reasonable value in here. So we can take this mass matrix and represent it as the inertial property and describe those properties on the robot.

</p><p>So let's go back. You understand those inertial properties. The green one is what you have now. You reduce your effective inertia by ten times and immediately, you improve your control performance. So this concept is quite complicated because it requires this elasticity. It requires the big motor. It requires transmissions to couple it.

</p><p>We came up with another idea, which is if we want an elastic actuator, why don't we go directly for muscle-like actuators, like in this nice concept? So we started building a robot like this, and the kids in this robot is that you use bones, muscles and air pressure. So you are bringing the energy through the air. Now you can lift the robot. You can produce the large magnitude of torque needed. Then you have a small motor on the joint.

</p><p>This motor will allow you to get the dynamics. So combining the two, you get hybrid actuation that essentially leads to a safe robot design. The problem with this is how can you manage all these tubes of air that are going to go through the

joints and control? Every joint will need two muscles.

</p><p>You realize you have a large amplifier, pressure regulators, and you cannot think, I'm going to take all these tubes. You will have a lot of them, and your arm will be very heavy.

</p><p>So the solution to this problem is what? Is we don't want the tubes, so we want to put this on the arm, big. So what do you do? Well, make is smaller. Now we've identified the right problem. Make it smaller. We can have a solution to this problem.

</p><p>Once a graduate student knows the problem, they solve it. In fact, it was a piece of cake making it smaller. No problem. So making it smaller, now you can distribute it. By distributing on the links, you essentially take one line of pressure running through the whole robot. At each of the joints, you are distributing the pressure, regulated to the value you need at that joint.

</p><p>So here is the arm they built. In fact, this arm was almost measured after my own arm. So it is really human-like arm with the constraints of a human and not too big. Here are the dimensions. I'm not sure about the torque. Here is the arm.

</p><p>You can hear the air pressure switching in control. So now you can do first control with this. You control the contact forces, and you produce motions with an arm that is lighter from the arm that we had before, 3.5. Here, we go to 1.5 kilograms, maximum. You can see on the red response, you have the macro robot, the muscle response is very slow. You get almost 0.5 hertz in the closed loop whereas with the macro mini, you can go to 35 hertz. Huge improvement.

</p><p>The other thing is by adding this tiny motor, you are improving the control, not only of the overall system but also by thinking about the macro differently and looking at what you are controlling, you can also do a lot of improvement. What we did, we added sensors to measure the tension. The actuation using muscles brings a lot of nonlinearities in the model. If you use a sensor, you can close a loop, and you can improve your control.

</p><p>So just the macro with a sensor goes to seven hertz. So this is really, really important always to use sensors where you can. With the group of [inaudible], we are working on the next prototype that will come probably in a few months. This prototype is bringing and integration of all these tubes directly inside the structure. This is going to be a cool arm.
</p><p>
</p><p>These are some of the built parts. You can see the overall design.

</p><p>So everything is integrated inside, and outside we have skin to protect in the first impact and to produce feedback. So you get tactile information about the contact information so you can control your robot better. Okay? So how many orders do we have today for this one?

</p><p>Please contact us. Yes?

</p><p>Student:[Inaudible] sensors are in the joint or the extremity or both/

</p><p>Instructor (Oussama Khatib):There are sensors inside the structure to measure tension on the muscles, but also we have sensors distributed on the skin outside. That will measure contact with the environment. So when you hit first, you reduce the impact force, and also, you measure where you are touching.

</p><p>Let's move to another aspect of this human-friendly goal and look at the planning and control and look at the human-robot interaction. So let me start with something that we basically developed in the early-'90s. I mentioned, and probably you saw this couple in our lab, Romeo and Juliet, the two platforms that we use to

develop and explore the area of human-friendly robotics. Here is the video.

</p><p>If we can have the projector please off?

[Music]

</p><p>Instructor (Oussama Khatib):So the idea is really not to think about the problem as moving the platform, stopping, manipulating, moving again, stopping, manipulating. It is about how you can combine mobility and manipulation in a way where you are controlling both. So if you are holding an object, and you have to move because of an obstacle, you should be able to uncouple the two.

</p><p>We call this the task, and this is the posture. We are able to control the whole robot to achieve a task and also to achieve the posture in an uncoupled way. With this, you can do a lot of things. You can clean the carpet.

[Music]

</p><p>Instructor (Oussama Khatib):Open the fridge. Thank you. Make a contact. Control the contact force. This is my shirt. I think it's the only shirt in the world that was ironed by the robot. I should add it was ironed one time.

</p><p>This is one of the most complex tasks you can imagine. You can imagine internal force is controlled, macromania actuation, cooperation between the two. All of this involves a lot of models and things we will be discussing in advanced robotics.

</p><p>This was accomplished and demonstrated at the opening of the Gates building. In fact, Bill Gates, when he came to the integration, he was quite amused with those robots, which you can, in fact, use to interact with a human. So my students used to dance with those robots.

</p><p>Here is another example of how you can interact with them by guiding the motion. So the human is giving the intelligent task of the guidance, and the robot is following. This is basically – Alan is dancing.

</p><p>By '97, when we accomplished this, I don't know if you remember, I mentioned by '97/'98, there was the Honda robot that brought the first walking stable machine, and it was a remarkable achievement. It dealt with the locomotion aspect, but it didn't really address the problem of manipulation.

</p><p>Since then, we had this challenge to bring together manipulation and mobility in the way that we did it for Romeo and Juliet, on a more complex robot that involves many degrees of freedom, a branching structure. We're not talking about contact with one ender factor. There are many different points you are controlling.

</p><p>You have multi-contact. You have many constraints. You have a lot of things taking place, and you need to deal with all of these simultaneously and in a coherent fashion. You cannot pull in one direction without accounting for what is happening in other directions.

</p><p>So the idea now of dealing with this problem through [inaudible] is crazy because you have a lot of degrees of freedom. You don't want to commit your final configuration before moving because there are constraints and joint limits and many things that happens. So an approach that we discussed in class earlier about task-oriented control makes a lot of sense. You pull toward the goal and the configuration emerges. You do not decide it ahead of time.

</p><p>This approach results into a very simple way of controlling the robot direction in the task space. So in the next two minutes, I'll show you how we do this here. Realize that there are a lot of details.

</p><p>Let's go to the arm. That's what we studied earlier. You have a force. You have a gradient of the force. How do you apply it? Torque equals?

</p><p>Student:[Inaudible].

</p><p>Instructor (Oussama Khatib):Transpose it. Okay. Very simple. That should work. We cannot just do it like this. We need to account for the mass properties. We need to establish the model between acceleration and forces, and by having a good estimate of this model, we can go and correct this. Now we uncouple the system, and we align the directions to follow the initial properties.

</p><p>Now, the question is if we go to this problem, how do you deal with all these points? How do you combine? How do you control this while you are controlling that? If you move one arm, everything else is going to move. How do you uncouple the motion of the right arm from the left arm while balancing and doing all these things?

</p><p>Student:[Inaudible].

</p><p>Instructor (Oussama Khatib):That is a good idea. Gyroscopes will give you a good sensing by integrating. You can find the orientation of the [inaudible]. You can use that information, but how can we deal with this problem of finding the dynamic equations of a system like this and controlling it?

</p><p>Yeah, we want to use energy, but first we need to get a model of the system for this task and that task and the head and the legs and so on. So we need to extend this operational space control or model, not only to find these mass properties, but I want to find all the mass properties and the coupling between them.
</p><p>
</p><p>It's not obvious, but it's very simple. So when you run into a problem like this, just sit down, relax, and just move back. Move back from the problem. Don't go too close to the problem. Move back. What do you see? You go to a higher-dimensional space. If you go to a higher-dimensional space, everything appears like one point.

</p><p>So if we take this problem, X1, X2, X3, and put them in a higher-dimensional space, that higher-dimensional space, they will be a point, a task, involving X1, X2, X3 all of these. We are back to the beginning.

</p><p>Now with this, you can find a Jacovian. Once you have a description, you find a Jacovian. You find your mass matrix, MX, [inaudible] is what I call it.

</p><p>On the diagonal, you have the mass properties at each of the points, and at the other diagonal, you have the coupling between them. Now you're back to this. Beautiful. You have the same model in the higher-dimensional space.

</p><p>Now you can use this model to control the system. It's very easy. We use the energy. That is, we use a force of task to control it with J transpose F. But now, suppose I'm controlling the legs and the hands. I have all this motion that is possible. This is the nonspace motion. This is what we will discuss with redundancy next quarter.

</p><p>In there, you have to guarantee that your control is not consistent. Do not interfere with the first control. To do that, you filter it with N, the null space of the Jacovian. The result is that you guarantee that there will be zero axil rotation coming from the control of the posture. That means when we move in the postural space, these points will be fixed. That is very, very important.

</p><p>Once you have this uncoupling, then you can control the system in a very effective way. So the control of the whole body is task field to control the task, posture field to control the posture and the two are uncoupled. So you can generate motions without any trajectory.

</p><p>The question is how can we find these energies. How can we find those criteria? If we are working with horses, maybe we should look at horses, but if we are working with human-like robots, we should look at humans. How do humans do it, and this really a good question.

</p><p>We already started to be inspire by the human, the way humans move with their hands, the way their move their body, and we need to see how all of this can be captured through simple mathematical models that we can use to reproduce the motion. So the starting point would be motion capture. Capture the motion, and then I'm not going to replay the motion directly because if I record the motion and replay it, then we cannot generalize. What we are after is not a specific motion. We are after understanding.

</p><p>So if you have a system you want to understand, what do you do? You shake it enough to find all the behaviors and to – and then you identify what the model of that system is. So you need a model of the system. It is really interesting. We were working with robots, and now we are working with a human model. The human models are essentially articulated by these systems. The mass matrix you computed is the same.

</p><p>You can use the same mass matrix. The actuation is different. So for the mass matrix, you can now take the rigid body part, the skeletal model, and now animate it. This is what you see. In fact, drop it in the gravity field. Potential.

</p><p>The idea is how can we go from the human to the robot with less degrees of freedom? We want to see the characteristics of the motion. To see the characteristics of the motion, you need to see the actuation models. You cannot do it without the actuation.

</p><p>So we model the muscles, and all of this was done with people who are working in biomechanics, the group of [inaudible], who provided us with all the data about muscles and skeletal system and the skeletal model. Once you have all of this, you can start doing the study and analysis, and you start to look at it and try to find out what's going on.

</p><p>So to analyze this, where do we start? We can say, let's start at something very simple. I'm going to give the human a task, which is to hold some object. The question is where the posture is going to be. Imagine you are pushing – it was very cold in the morning. Your car didn't start. You're going to push your car. How do you push the car? Do you push it this way, or do you push it like this? The answer is obvious. Actually, the child, at birth, they didn't know that. So little by little, you learn, and you discover something amazing about the human body, and then you start using it.

</p><p>What is amazing is in any mechanical structure like this, there is a mechanical advantage. When you discover the mechanical advantage, you start using it. So you do not just use the rigid body mechanical advantage. You are also using the way you're actuated and your muscles. The child, when he's moving first, will pull the muscles, and it hurts.

</p><p>You have this feedback, and little by little, you start to adjust to the motion and move your body correct. So humans, little by little, are learning, discovering ways to minimize the effort produced by the muscle. This is our speculation.

</p><p>We said there is muscular effort minimization, but it's not always like that. There are also a lot of other constraints. It's much easier to bring the spoon to the mouth than the – it's easier to bring the mouth to the spoon than the spoon to the mouth. The reason is mom says, this is not polite. You cannot do this. You have to do it this way.

</p><p>So there are a lot of social constraints, there are a lot of physical constraints on the system that you need to integrate. But essentially, you are minimizing the effort. If you say, I'm minimizing the effort, it's very important because now you bring the physiology in the model, and you start analyzing the physiology.

</p><p>So a robot produces a force by applying a [inaudible]. All of you now know this, right? A human produces that torque using another Jacovian, L transpose. This is the muscle attachment Jacovian, and these are the muscles. Now, if you think about it, you are really attuned to use this in a way to minimize your muscular [inaudible]. But every muscle has a different capacity, bigger muscles, smaller muscles, so you need to account for the capacity of the muscle.

</p><p>So here is our hypothesis. We think humans are adjusting their posture to minimize muscular, so there must be an energy. Okay? Now pay attention. We want to find this energy. So the muscle is M. Extension of the muscle, the tension, is M. If I'm minimizing the muscle effort, the energy's going to be M-squared. But there is the capacity, and what we speculated about is that this energy is weighted for different muscles with their capacity.

</p><p>So here is the inertia. Are you already sitting? Are you ready? Want to see it?

</p><p>If you look inside of this, this is what you see. This captures the following. It captures your task, captures the mechanical advantage of the skeletal system. It captures the muscle attachment and the capacity of your muscles. A muscle is weak. You compensate for it. If a muscle is missing, you're compensating for it.

</p><p>So in fact, it turns out that our student went to the motion capture lab, and we did the analysis. It's amazing. I see you sitting comfortably. Why don't you relax your back and move forward without touching anything with your back. Let's try it.

</p><p>We're going to see if it works also with you. So let's drink a cup of coffee. Don't drink it completely. Just go very close to drinking the coffee, and comfortably and like you're relaxed. Look around. Everyone is doing this not like this. It is this angle.

</p><p>This angle corresponds to the minimum of this energy. So it is not a coincidence. In fact, if you do this analysis and you increase the weight, this angle goes up. You can see it here, as you move and increase the weight from zero to 15 pounds, for instance, this is the speculated angle that you will find.

</p><p>You can check it, but it makes so much sense. You are discovering how to use this machine. You are able to use it for the specific task in its optimal way. It's not only Jacovian. That is, it's not only the skeletal model. It's also the muscle attachment and the way the muscle capacity is distributed.

</p><p>So with this, now, we can go and simulate and create. So we are doing – I'm not sure if you understand this, what I mean. You remember I showed you this? I think I showed you this, right? We saw Stan Bar. You saw this before, right?

</p><p>Quickly. We move to the goal, and you are able to track that motion, right? You remember you can fold the joint limits. So let's go back to this. So this is what you're doing. We are controlling one point, and the body is adjusting through our criteria in the null space. Now you can move it to robot. Not as a trajectory that you're copying, but as a criteria that you're applying to the robot.

</p><p>This is the environment that we developed that also contains neuromuscular models that can be used to analyze and look at the human. This is a recent work on a Tai Chi analysis, and the master here from Beijing is performing a motion that we can record, and then we can analyze the motion. You can start to see that all these

models can produce amazing things so that essentially you can analyze skills of a human. You can look at the behavior of a human and synthesize it.

</p><p>In fact, if you recall that motion, and if you play it back on the physical model of the robot, it will fall over because you need control. You have errors, and unless you control it correctly, it's going to fall. So what we are going to show here is half of the body is following the Tai Chi motion, and the other half is controlled. In fact, you can see that you can achieve those desired half while controlling the robot with other behaviors.

</p><p>So all of these are part of the development that is now being implemented on humanoid robotics like Ashimo. A key aspect of the implementation and task-oriented control is not only just motion and force control, but how do you deal with constraints? How do you deal with the fact that if you have a joint limit, and you're moving, and hit a [inaudible], what is going to happen?

</p><p>So this structure of control that I talked about produces a very useful way to apply constraints. Essentially, I mentioned about constraints in terms of attractive forces and repulsive forces. If you have a constraint, it has to take the highest priority. In the structure of task and posture control, we have priority. This task will not interfere with this.

</p><p>So if we know our constraint, and we know the Jacovian of that strength, then we can know the null space of that constraint. Then we can take this whole thing, put it in the null space of that constraint and then control the constraints.

</p><p>Now the robot is moving, and you can see two different postures because you have hip limits. Here you are stuck. You cannot reach because of joint limits. Here the body is going to move away to avoid self-collision. So the trajectory is going through the body, and the body will move away automatically by these repulsive forces.

</p><p>Obstacles? So it is very, very important to be able to create this interactive behavior that allows you to avoid collisions. At the same time, you need to think about the global path, not only about the local behavior.

</p><p>We have N degrees of freedom. The problem is exponentially in the number of degrees of freedom. You have to deal with a way of connecting the two. To do that, we developed a concept we call elastic planning, which is, essentially, connect all the plans that allows you to deform it in real time. By deforming this, we are able to change a feasible plan and adapt it to the real environment. This is an amazing thing because you are able to change a trajectory that requires hours of planning.

</p><p>If you didn't use this, for example, you would have the plan the whole trajectory with new obstacles and constraints. At the same time, you are reconstructing because when you are touching, tele-operating, interacting with surfaces, you need to reconstruct the surface to fit and control your robot. That means you need to model contact correctly. You need to deal with the contact models also in the control. This is a very, very interesting result that comes directly from control.

</p><p>You remember when we talked about collision, essentially, we are interested in the collision with the multi-body system. How can you resolve it? Well, people usually remove the joints, resolve the whole multi-body collision, and then they put the joint, eliminate constraints. But in there, we are going to be able to reduce the problem to this because we can use the mass matrix in this direction. We can use the effective inertia directly.

</p><p>The collision doesn't care about what humanoid robot you have. You just need to know what mass properties you have and the contact. So with this, we have a very effective algorithm that allows us to simulate and resolve the collision in real time. With the real-time resolution, you can go and use it in many different things. I believe I showed you this in the beginning of the class, but now you

understand what it means. We are able to find the properties and impact forces, and we are also able to deal with this problem of contact and collision.

</p><p>The problem is very difficult because when you are looking at a humanoid robot, you just push it and it will tip over. It is missing these six actuators. It's not connected to the ground. You spend your time balancing, and if you have any reaction force, it will tip it over. So the question is how can we move this body while maintaining, dealing with those constraints? How can you do it?

</p><p>Just say it. Say, I need to treat these as constraints. What are your constraints? [Inaudible] normals, take the Jacovian. Take the null space, and take this and put it in. Now you know how to do it, okay? Put it in the contact space. Then you are able to bring the two together, and now this motion will be consistent with the constraints, and you can control the forces and the constraints.

</p><p>If you remember in the beginning of the class, I would talk about omega and omega bar. This will become your omega bar, and this is your omega in the multi-contact space. So that was a really very important result to allow us to implement behavior with multi-contact and motions. Now distributing the effort between different surfaces that you are touching, moving, balancing, dealing with dynamic skills.

</p><p>Then you can start to build behavior over this that is now the robot is looking, watching. If there is an obstacle, it will move down. If there are constraints, you are able to deal with the constraints as it moves. It is automatically generating the right motions to avoid hurting those constraints.

</p><p>As we are building, we are moving up and up in the structure. So what we really have seen in the class is really the lower level in this system where we are looking at the execution of the motion controllers, the first controller. But on top of this, you can build behavior that makes use of those primitives, and you go up to levels where you require more abstraction of the sensor information, your perception. As you are building a solid foundation, you are going to be able to move higher and higher by integrating skills and learning from the human. Hopefully we will be able to get Ashimo to graduate.

</p><p>All right. Well, I guess I'm going to stop here. Tonight, we will see the rest of the group. Yesterday we had a very nice session. We will talk about the final. Those who I won't see tonight, I wish you good luck in the final. For those I will see, we will talk about it more in the evening. So 7:00. Okay. Stop here.
</p><p>
[End of Audio]
</p><p>
Duration: 72 minutes
</p><p>

</html>