

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 terms 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, \ddot{X} , and controlled by a force, F . So the control of this robot is done through this proportional derivative controller involving minus K_P , X desired and minus K_V , \dot{X} . So the K_P is your position gain, and the K_V 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, spring, damper system whose rest position is at the desired X_D position. So K_V is your velocity gain, and K_P 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? K_P 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 ω^2 where ω is your closed loop frequency, and where ζ is this coefficient, KV divided by $2 \sqrt{KM}$, and ω 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 ζ , right? And KP is also affecting your ω . 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 ζ ; anyone remembers? It was only two days ago. ζ is equal to – for critically damped systems, ζ is equal to unity, 1. When ζ is equal to 1, that is when KV is equal to $2 \sqrt{KPM}$, 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 ζ . So, basically, you are trying to set ζ . What about ω ? So now, we need to set KP in order to compute ζ , and how do we set ω ? 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 $M \omega^2$, and KV is $M \omega$, 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 $M \omega$, 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 times 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 ω 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 $F' \alpha F'$. $\alpha F'$ actually is mass F' , and now we are going to add another term, β , 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 \dot{X} exactly. You have estimate of these, what we call the \hat{X} , $\hat{\dot{X}}$, and \hat{B} . Now, B has a structure. If it's the gravity, it's going to be, I don't know, $M \cos \theta$ 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 \hat{B} .

So, in that case, you can say α is simply the mass, an estimate of the mass, minus/plus one gram, probably you will find it, and your \hat{B} 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 β will take out B . For each configuration, each velocity, β 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 \hat{B} were identical – if \hat{B} 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' 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 , \dot{X} , the state. Basically what we are doing is we are looking at the model of the system, and we are using X and \dot{X} to estimate B , the nonlinearities in the system, and compensate for them. So F is going to have a component, which is \hat{B} . In addition, our input control, which is F' , 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' 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 X_D , 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 K_V prime minus \dot{X} minus K_T prime times X minus X_D .

Instructor (Oussama Khatib): You meant minus K_P prime, X minus X_D . So minus K_V prime \dot{X} minus K_P prime, X minus X_D , 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. K_P prime is trying to reduce the error, and K_V 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 \ddot{X} 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 \ddot{X} , and I know the desired acceleration, \ddot{X} desired. What should F prime be? Come on.

Student: Minus the cost of minus \ddot{X} , 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 \ddot{X} , and have the desired acceleration \ddot{X} desired. What I would do with F prime, just make F prime equal to?

Student: \ddot{X} .

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

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

Student: \dot{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 δX .

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

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 I_L . The inertia of the motor is I_M . What do you expect the real I_L , the effective I_L to be? So it's going to be equal to I_L 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 I_M ? 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 I_L plus N square or eight a square I_M , 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 I_L – I_L is going to change. I_M is constant; it's the same motor. So the question is if I_L is changing, what is the effect of that? Well, the effect of that is changing your time response because your K_P , if you remember, is M times K_P prime. So it affects, directly, your ω . 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 I_L and the maximum value of your I_L , 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