MITOCW | Lecture 7 | MIT 6.832 Underactuated Robotics, Spring 2009

**RUSS TEDRAKE:** OK. Welcome back. So last time we talked about the cart-pole and the acrobat systems. Two of the model systems that are sort of the fundamentals of a lot of the underactuated control work in robotics. We really just talked about balancing at the top, showing that some of the linear optimal control things we've already done were consistent with balancing those underactuated systems at the top. And we said if we can, maybe today, design a controller to get as close to the top, then we could turn on something like a feedback controller based on the linear quadratic regulators to catch us.

So we were basically able to do linear control at the top. And today we're going to do the swing up, which is in some ways easier, and in some ways harder. It turns out very simple controllers will get the job done. But showing that they get the job done is quite another story. And really my goal for today is to give you just a little glimpse into what the world of nonlinear control looks like for the other guys, for the people who really do nonlinear control. All right. We're going to mostly hurtle over this stuff and get to the computational versions, but if you were going to be sort of a nonlinear robotics control guy, you'll see a little bit of that today and just get a sense. I hope.

And I'll finish the story with PFL, the partial feedback linearization, showing you the most general form and what it's good for in the swing up phase. OK. So non-linear control comes in a lot of varieties. I'm going to make this point a couple of times, but, roughly, one way to say what nonlinear control is that end up looking at your nonlinear equations of motion for a long time, and you try to find some tricks. So you can write down some non-linear controller, and then another trick just to prove that it works. OK. So one of the tricks that is pervasive in under actuated control, and other control, is to look at the energetics of the system. The energetics of-- the total energy of the system is obviously a very important quantity.

You can imagine with our acrobat, or our cart-pole, we have one motor to work with, multiple state variables, right. But the energy is just a single quantity. So you can imagine with one motor possibly regulating the energy of the system, even if we can't regulate the entire trajectory of the system. OK. So that's one of the reasons it's important. So let's start by just thinking about if we can regulate the energy of the system, what good is that going to do us. OK. And again let's start simple. Let's do the derivation on the simple pendulum, then we'll do the cart-pole and acrobat immediately after that.

So let's remember the phase portrait for the simple pendulum again. Let's say with no torque, no damping. We had a phase portrait-- I'm sure you remember-- which looked like this. And the way I had the coordinate system, we had a fixed point, unstable fixed point, fixed point. OK. If we started the system a little bit away from the fixed point, then we got these closed orbits, right. For the zero torque, zero damping case, we got these nice orbits. They were pretty circular, close to the origin. They got a little elongated as we went out. Look like an eyeball.

And then there was this very special orbit. I didn't dwell on it before, but we're going to dwell on it now for a second. A special orbit which ends up going right to that unstable, fixed point. OK. What defines these orbits? Energy, someone said, right. These are lines of constant energy in the simple pendulum, right. And there is a particular line of constant-- orbit of constant energy here, which, if you have that energy, this system we know is going to go around this way. There's no other choice. If we have that energy, right, then it's going to find its way up to that fixed point.

There's nothing else it can do. OK. So these orbits are called homoclinic orbits What's that? So a homoclinic orbit is something that goes from an unstable, fixed point to another unstable, fixed point. OK. Actually, homoclinic means it goes to the same unstable, fixed point. Heteroclinic means it goes to a different one, I guess. And in the simple pendulum with zero damping and zero actions. It's sort of easy to see that if we can design a trajectory which regulates the energy of the system, right, it's going to move. Let's say we set this as our desired energy for the system. If we can regulate the system's energy up to this, by applying a feedback law. That if we can get on that constant energy trajectory and stay there, then we're going to find ourselves getting to the unstable fixed point. OK. So that's an important observation.

Now, regulating the energy isn't actually enough to stay there. Well is it? I should get to the-- if I really regulate my energy perfectly, I should get to the unstable, fixed point. But if I'm just doing energy regulation, then if someone were to knock me here, I'd be OK. I just come right back. If I knock me here, I'd actually go all the way around to this one. So it's not actually, doesn't actually stabilize. Regulating the energy doesn't actually stabilize this point. But it could get me there. OK. So the first thing we're going to do today is show that if I regulate the energy of my system, on the simple pendulum, to put myself on the homoclinic orbit then I can get myself to the fixed point.

I turn on an LQR controller when I get there, and I've got a swing-up and balance controller. OK. All right. How do we regulate the energy of the simple pendulum? Let's just remember for the simple pendulum our equations of motion were m l squared theta dot plus m g l sin theta, according to the system I used, which was 0 down, equals my control input. And the energy is just 1/2 m l squared theta dot squared minus m g l cos theta, the total energy of the system. OK. So if I find myself in some theta theta dot configuration, what should I do to increase the energy the system?

**STUDENT:** Push in the direction of theta dot

**RUSS TEDRAKE:** Good. Yeah. Right. Whatever I want to do, whatever velocity I'm going at, if I push in the direction of theta dot, that's going to add kinetic energy to my system. OK. We can see that actually very explicitly, so what is the rate of change of e here? It's 1/2 m l squared theta dot theta double dot-- this half goes away-- minus m g l theta dot sin theta. I put my sin there again. And theta double dot is u minus m g l sin theta over m l squared. Can I do that? All right. So m l squared theta double dot is just u minus m g l sin theta plus m g l sin theta theta dot.

All right. So that works out nice. It's just u theta dot. So it's exactly what we said. If you want to increase the energy of the system, right, then you better make u the same sign as theta dot. You can make it exactly theta dot. You can make it a constant. You can make it-- just as long as it's the same sign, you're going to increase the energy of the system. And similarly you can decrease the energy of the system by pushing against theta dot. And that's obvious. Really, that's just either damping the system with-- you're adding positive damping to the system or negative damping to the system right. OK.

But seeing it out of these energy derivatives is the right way to see it if you want to do it for a more complicated system. OK. So good. So I can now imagine easily with my control torque regulating the energy the system, or increasing or decreasing it. So what do we want to do with it? Presumably we have some desired energy, e, which is the energy on that homoclinic orbit. Right. And what's that? What's the energy on the homoclinic orbit? M g l. Yeah. At the top it'll be [? cos of e ?]

The only problem with this parameterization is I have negative energy. I could have done a little bit more carefully by putting zero potential at the bottom, but right. This one-- still our desired energy is positive and it's at the top, m g l. That's the energy you'd have if theta equals pi and theta dot equals 0. OK. So what we care about is regulating the difference between our actual energy of our system and the desired energy the system. The desired energy is constant. So this is just e dot. e tilde dot is just e dot, which is just e theta dot.

So if I know to change e, I certainly know how to change e tilde. OK. So the cool thing here, let's choose u equals negative k theta dot e tilde, with k always greater than 0. The result then is e tilde is going to be negative k, which is positive, theta dot squared e tilde. Just a little manipulation. And that gives me that, as long as this thing is negative-- it's a linear equation in e-- then e tilde is going to go to 0. OK. And what is that? That's obvious. This is exactly the case of, I'm going to use u to add damping to my system. I'm going to add some negative damping. If e is greater than e desired, then this thing is positive, and the whole thing I'm adding, I'm going to be adding damping to the system to slow down.

And if e is less than e desired, then I'm going to put in sort of negative damping. I'm going to invert the sign of this and add energy to the system until e is up to e desired. And this is going to give me a nice first order response in e. So that should really do the trick. And it doesn't take much looking at this, the proof would be a little bit harder, but it doesn't take much looking at this. If I did have something like bounded torque, bounded control input-- let's say I saturated u at some control limits-- as long as my sign is correct, I could still add or remove damping, according to the sine of tilde, and have this result that e tilde is going to go to 0.

Yeah. OK. Even with bounded control inputs this is an OK. So what is the resulting phase portrait look like? OK so I have here exactly what we just said. e tilde is energy at the current state minus energy at the desired state. Let's just do negative k theta dot times e tilde as my control, with a pretty low k. And I remember what I called it here. That didn't stabilize it at the top. And so any small integration errors, or whatever is going to keep me from the top, is going to make me go around. But the phase portrait does exactly what we want it to do. This just started from two different initial conditions.

One, that's the same one I've been using for all the phase plots. One just above the origin there. It adds energy up, goes to the homoclinic orbit, just misses the fixed point, goes around again. And then the other one started over here with too much energy, and actually slowed down to the homoclinic orbit. OK. Simple. So for the simple pendulum, we could design a nonlinear controller with pretty simple reasoning about the energy. And we know it's going to get as close to the up right. Does it also work for more complicated systems? What would it do for the cart-pole or the acrobat. It turns out it's a pretty general idea.

OK. In fact, let's do the cart-pole, because the cart-pole actually has the pendulum dynamics embedded in it. It just happens to be coupled with the cart. OK. So if we, for instance, use our partial feedback linearization trick to decouple those dynamics, then we can really just keep thinking about pendula. OK. So here we go swing-up control-- non-linear swing-up control of the cart-pole system. So the dynamics were something I was willing to write on the board once and then set all parameters even gravity to one, and got some complaints but kept going. It turns out it would have been actually trivial for me to carry g around. I should have-- next time I'll do that next year.

So I don't even have to write that today because we know if we apply a direct-- sorry, the collocated PFL, then the result of my little tricks with the equations is I can command a force. The collocated means I'm going to use the action to linearize the state, with that action associated. So here I'm going to linearize the dynamics of the cart. And the resulting system was x double dot equals my new control input. I could set that to be whatever I'd like. And what did it leave us with? it left us with theta double dot doing something a little funny but not too bad. ubar is sort of my desired acceleration here.

OK so I've got some new, almost, pendulum dynamics here. And I've got obviously got this pendulum. And I've got this cart that I can do whatever the heck I want with. Because I've mastered the cart. So here's the idea. Let's make sure that the pendulum gets on the homoclinic orbit for the pendulum. That'll make sure it gets itself up to the top. And if we've done that, then we can just do a little bit of extra work to make sure that the car doesn't go crazy. OK. OK so you guys are actually going to-- on your problem set if you haven't already-- you're going to do the swing-up controller for the cart-pole.

So I'm going to keep with my all parameters happened to be one lingo here, which forces you to handle the terms when you go to do it for the problems set. It's not bad, but it makes you think about it I think. OK. In my all parameters equal 1 land, the energy of the pendulum still it's just 1/2 m l squared. So it's just 1/2 theta dot squared minus m g l cosine theta. Just minus cosine theta now. And e desired is going to be 1.

e tilde dot, that going to be theta dot theta double dot minus theta dot sine theta. theta double dot, though, is going to have a little bit different form than the simple pendulum. Because it had a few terms left over from the coupling with the cart. Right so if we stick those in we get theta dot negative u c minus s minus theta dot s

**STUDENT:** Is it plus theta dot s or minus theta dot s?

**RUSS TEDRAKE:** Good catch. Thank you. Thank you. I would have figured that out in a second when they things didn't cancel. Yeah so now this cancels leaving negative ubar cosine theta dot. OK. So we've got this slightly funky dynamics for the pendulum, now, because it's been augmented by the cart. But that's not too bad. So what should I do? What controller should I run to regulate this thing to the pendulum's homoclinic orbit? OK. So what do you want? You want ubar. It'll give us some constant. If I just make it, say, cosine theta dot. That'll make everything positive. There's a lot of ways to do it, but that's a pretty simple one.

Yeah and I'll keep that e tilde there so that I regulate to the proper energy, not just to zero energy. And that should do it. So now e tilde dot should be negative k something squared, c theta squared e tilde, which means e tilde it's going to go to 0. Which means my pendulum's is going to get its homoclinic orbit, and get to its unstable, fixed point. OK. What about the cart? What's the cart going to be doing during all this? Yeah it's going to be going back and forth. Well it depends, I mean, it could actually wind up or something, depending on the trajectory.

So for this system we actually have a simple-- if you know Lyapunov functions-- a simple Lyapunov function you construct from these energies. They would show that the pendulum gets to the top. If you want the pendulum and the cart to get to the origin, then we just get a little sloppy. And we say let's do, not u equals k theta e tilde c. We're missing a c. That's the first one. But let's add an extra term in minus k 2, the position of the cart, minus k 3, derivative of the cart. OK.

All right. So sort of a sad fact, that actually doing this is what makes it work. It's trivial that, sort of, that it makes it work. Experimentally, you'll see it works just fine. It actually breaks all the proofs. There's a proof for a set of parameters of k. So someone found like if k equals 2 and k 3 equals 3, then I can design Lyapunov function. It's not very satisfying. OK. But let's see how this works in practice. So you remember your cart-pole system now. So these controllers were described very nicely by Mark Spong in a paper in '96 so that's my spot '96 controller. Basically, you can see I just check some distance between the current state and some sort of hand designed distance space there.

I just see how close it is to the top. If it's close enough, I do LQR. If it's farther, I'm going to do this energy shaping control. The stuff that I commented out, I'm going to tell you about it later. But it turns out there's actually nice sort of analytical ways to get the basin of attraction using some semidefinite programming. We'll mention that later in the class. So you don't actually have to guess that metric, but to be fair, I guess that metric as something that is easy to guess. And that's good enough. OK. The LQR we did before, and the swing-up is just that simple. It's the pendulum energy. Not the total energy. It's just the energy of the pendulum on the system with a PD controller on the cart and the theta dot cosine theta e tilde dot controller on the pendulum.

I put a saturation in there so it doesn't do crazy torques. Let's see what we get. Now linear control is easy. OK. And the phase plots exactly what you'd expect. It looks like my gains a little high maybe. But you know it's regulated itself pretty quickly out to that high energy orbit. And then it just missed its chance in the first one, but by the second time the energy was close enough, turn on the balance controller, grabbed it. And you saw that it actually grabbed it off to the side and then slowly pulled itself back towards the center.

Different initial conditions. Yeah, so it works. It works well. It's easy right. OK. The reason I wanted to show you this, well first of all, it's cool. It's relevant. But I sort of want to give you a little glimpse into the world of nonlinear control. What tends to happen, if you're trying to come up with clever non-linear control solutions for systems, is you tend to have to examine carefully the equations of motion of your system. And sometimes you end up with these sort of bizarre looking feedback laws that work. Because they tend to cancel out the terms that you want to cancel out. So you can prove that something is always monotonically decreasing. Something like the energy is always monotonically decreasing or something like that. That's actually a pretty typical. That's a pretty representative case of a nonlinear control design.

And there's lots of cool stuff working in that realm. So typically to prove it more carefully we'd look-- we'd design a Lyapunov function. We're not going to use it right now in this class. But if you haven't seen Lyapunov functions it's a good thing to know about. Jean-Jacques Slotine teaches his nonlinear control course in mechanical engineering. He uses Lyapunov functions. But he also uses other metrics to prove stability. His favorite, I think, right now is contraction analysis. OK. It's worth listing those two because there's actually not a lot-- that's not really a very long list-- there's not actually a lot more different solution techniques that I can put up here. There's actually only sort of a handful of ways people have to prove stability in non-linear systems, continuous systems.

OK. So that's sort of approach one. Approach two, which we're going to do in this course, is let's not design specific controllers for specific systems. Let's turn it into an optimization problem and use our optimal control. OK. Good. So we know how to do the swing-up control of the cart-pole. It's pretty simple based on an energy argument. What would happen if I didn't use the PFL? I used the PFL here to make things analytically cleaner. But essentially what it's doing is the same thing as the simple pendulum. As it said, if the pendulum is moving this way, I want to add energy. Then just push it in that way. If it's derivative's this way, then push it in that way. There's a cosine which modifies things. Just because of the coupling really I think. Maybe the cosine's real. When you change signs you actually-- when you cross the origin you actually have to go the opposite way. Maybe the cosine's real actually.

But it's basically just doing the same thing we did-- had our intuition for in the pendulum, which is push the system the way it's already going. So it turns out you can do the same thing without PFL and it probably works. To be fair, I even spent a minute finding different parameters. I could've made a really compelling demo using the same parameters from PFL which worked horribly for the other system. But I actually found some good parameters and if I do that one, it still gets there. It's not going to be quite as pretty. That was actually probably the worst I've seen. Thank you. OK but it'll still work. The PFL, though, made the motion of that card pretty beautiful, if you care about that. And it made the math easy enough that we can really derive these things.

OK. So PFL really turns out to be an important tool in these systems. Yeah, I just said basically, let's put in this controller. Forget about the-- actually I did the whole controller, I think. I can look. And instead of putting it into ubar, which was my synthetic command through my PFL, let's just make that the force on the cart. Let's use this basic feedback law as the force directly on the cart. And forget about linearizing the cart dynamics. Yeah, because the intuition is right. It's just pushed that way to add energy. And if I add energy. I should get up near the top. And if it does work. And the PFL just makes it cleaner.

OK. What about the acrobat? I don't spend a lot of time because it's pretty similar. What would you do for the acrobat? If you want to get the acrobat to balance, just add energy. And get itself towards the top. It's funny. Zach, in our group, he took the class last year. He's building our cool acrobat now. He's the one you saw him hold up the poster. But for his first reaction, which I think he did on the real hardware, was just do a stabilizing controller at the bottom. And then flip the gains. So it just went kind of crazy. And I think that sort of worked, which is-- doesn't say a lot about how hard the problem is. But there's a more elegant way to do it. OK. How would you do it? Given this sort of line of thinking. We've got PFL at our disposal. We want to add energy. What would you do?

Yeah? No, but you're close. Well so collocated would get-- it controls the elbow, because the motors at the elbow. Yeah. So if you use collocated and you could still solve-- add energy into the first link so that it swings up like a pendulum. That's the same way we sort of add an extra term to keep the cart. Near zero we add an extra term to keep the second joint close to zero. That's exactly what it is. OK. So I'll just write it quickly because it's that simple. We're going to use collocated PFL to control theta two and energy shaping to drive theta 1 to up right.

So it turns out-- how do you add energy in the acrobat? if you want to add energy to the theta 1, whatever direction that one's moving, make the theta 2 move in that direction too. OK. So u bar, the command to your PFL controller should be q 1 dot e tilde, k greater than 0 It works out to be that simple. And we're going to, in general, we'll add another something to make sure that theta 2 doesn't deviate too far from where it went, from straight out. Interestingly, the energy that people use-- that Spong uses in his paper is not the energy sort of the simple pendulum created by link one. It's actually the total energy of the system and that still works.

**STUDENT:** You couldn't use the total energy of the cart-pole to try and invert that pendulum?

**RUSS TEDRAKE:** My guess is it would still work, but the proof is based on-- on not that. So I should say there is no proof, for the acrobat, that it will get to the top, that I know of. Spong's paper in '96, beautiful paper, says, we conjecture that you can show that this thing is semiglobal stable. The other thing I didn't say is that this thing actually doesn't work if you started at 0 0. Because if theta dot zero, you're not going to actually do anything. So it's not globally stable. It's semiglobally, so we need to knock a little bit. Have Zach pull with the string across through. But they conjecture that it's semiglobal, despite the fact that the energy-- that there might be multiple solutions.

The picture of the homoclinic orbit, I'll admit, is not as clean for the acrobat. But I think the intuition is that if you're doing some work to keep theta 2 near zero, then it's looking-- it's moving a lot like a pendulum, and you're thinking about the homoclinic orbit of that big pendulum. And it gets up there. I think these terms prevent it from sort of being in this lots of different configurations to get to the top. But presumably the proof of semiglobal, semiglobality, I guess, would be dependent on kl and k2 or k2 and k3. Good. People don't seem jumping out of your chair excited about that, but I hope you like it and get it and get it. It's an important class of derivations.

All right so in that trend. So we used PFL, now, to do-- to basically make energy shaping arguments simple. And to actually even simplify the analytics of those arguments. But actually PFL can do a lot of things. Disclaimer, PFL is bad. Don't use it if you care about system dynamics, but it'll do a lot of things for you. So let's just see how much it can do. I want to show you the slightly more general form, which is not, surprisingly, is not as common in the literature And just one general form, I guess, but this is using the notion of a task space.

OK. So last time, last time we talked about using partial feedback linearization to control all of your actuated joints perfectly, or to control all of your unsaturated joints perfectly. But naturally you'd like to see a form where you could control one unactuated joint, one actuated joint. Maybe you don't actually even care about controlling a particular actuators. Maybe what you care about is, let's say, controlling the end effector of your machine. So let's say I'm 120 degree of freedom robot with a few unactuate joints. I'm not actually bolted to the ground. Maybe my shoulder doesn't work today. And I want to control the endpoint of my arm right. We should be able to do that I think. And, intuitively, as long as my task space is coupled to my motors. I can have sort of 10 actuators on this robot and try to control the parts of joint over there.

As long as they're coupled with this inertial coupling idea. And I have sort of enough motors, as many motors as I have degrees of freedom of the thing I'm trying to control, you'd like to think that would work. So for the cart-pole, let's say, I'd like to think I could regulate-- I can't regulate the endpoint of the pole perfectly, because that would be x and y. It would be the two variables and I've only got one motor, but maybe I should be able to regulate the y position of the endpoint. On the acrobat, I should be able to regulate the y position of the end effector. Or, in little dog, the robot I showed you, we do regulate the center of mass of the robot using all the internal joints on the machine.

So let's see the slightly more general form. And then put it to use. OK. I'm still going to make the assumption that we have joined with zero torque and joints with torque. So I can use this form. I'm sure you can generalize it further. But excuse me. OK where I'm using q1 here to represent my passive joints. And let's say that there are m of them and q2 to represent my active joints, my actuator Let's say there are l of them. And let's say the q, m, n. So that's my equations of motion of the system I'm working on factored into block matrix form based on where the actuators are.

So let's try to control some subset of the actuated and unactuated joints. And in general we'll do that by defining some output function. Some function of just the full vector q, but we're not of q dot for now. And let's say y lives in p dimensional space. Just running out of letters here. OK. Let me tell you the answer first and then we'll make sure it works. So let's define j1 to be partial f partial q1, j2 to be partial f partial q2. And j is the full thing. So j1 is going to be what, p by m. And j2 is going to be p by l. And j is going to be p by n. My claim is if you command the actuated joints to do this.

OK. It's a little bit of a big pill to swallow, but I'm using this to represent the right Moore-Penrose pseudoinverse in general. OK. And there's nothing too egregious just yet. We've got h11 is invertible, we said. So that sort of looks OK. We're going to have to ask how invertible j bar is. It's got a j2 in there. j2 is not square. So we're going to have to think about that a little bit. That's why I have to use the pseudoinverse here. Good. So let's see if we believe that. Oh boy. This is subject to the rank of j bar equaling p.

And I don't want to write that on the next board. That's got to go there. OK. So let's just think about the dimensionality here. So j2 is p by l. So a number of things I'm trying to control by the number of things I have to control them. j1, p by m, this one's going to be m by m. This is going to be m by l. So this thing is going to work out to be p by l, as we expected. So I'm saying that I'm going to write j pseudo inverse because it's not square. But I only really want to run this controller if the rank of jbar is equal to p. It's not going to be greater than p ever. Because the matrix has only got p rows.

So the trivial condition is I need to have at least as many actuators as I have things I'm trying to control. If l is at least p, that's a necessary condition. I'm not going to get the if-- if I have less actuators than what I'm trying to control, this rank condition will always fail. If I have four actuators and two things I'm trying to control, but three of the actuators are on a different robot, then it's also going to fail. So it's not only the case that the counting game works. But if l is greater than p-- I'm sorry to be throwing around all these things-- if you have more actuators than things you're trying to control, then you should go ahead and use that pseudoinverse because you have choices. At any given time, if you want to produce a certain acceleration in your output space, whether it's the end effect or whatever. There are actually there's a manifold of solutions you can use to try to produce that same acceleration of the output.

So for those of you that have used sort of Jacobians and null spaces a lot in the previous robotics course, this is perfectly compatible then with sort of a null space decomposition. If you wanted to say I have a first priority task in this joyed space, and then a second priority task that works on the null space of that controller, that stuff all works in this framework. But I won't go into the details there. OK so j pseudoinverse could have a null space if you have more actuators than things you're trying to control. So we quickly make sure we think that works

So what do we do? We started with y is f of q. So we want to know how that thing's going to change. That's going to be j of q dot. And y double dot is going to be jdot qdot plus j, because remember j is partial f partial q right, plus jq double dot, which I'm going to break apart into j1 q1 double dot plus j2 q2 double dot. So I've got some contribution from the effects of my passive joint. Some from the active joints. We can use this first equation here to figure out what the accelerations of my passive joints are going to be with respect to the accelerations of my active joints.

The same thing we did before. So I can easily write down here that q1 double dot had better be negative h11 inverse, h12 q2 double dot plus phi1. If you stick that in, then you end up with y double dot is jdot qdot minus j1 h11 inverse h12 q2 double dot plus phi1. I would write this down if it's in your notes here, which happens to be jdot qdot plus jbar q2 double dot minus j1 h11 inverse phi1.

Therefore, If I can invert jbar then, where'd my control go. If you read it off, jbar inverse is going to cancel out that term. I'm going to subtract off the jdot qdot, and add back in the j1 h inverse phi1. And I'm going to be left with y double dot equals v. So that works. That's not too shocking. What's cool is that this task space PFL is sort of all, you need to know. if you're willing to remember this one, then the collocated and the non collocated come out for free. So if I were to choose and do the collocated, then that means my output function is this, which one, just q2 which means the partial with respect to q1 is 0. The partial with respect to q2 is just i. jdot is 0. jbar works out to be i if you look at It.

Long story short, everything drops out. You get exactly the collocated feedback that you wanted before. What's even cooler is the non-collocated does the exact same thing. The non-collocated, if you make y equal q1 and you put it through this task space PFL, not only do you get the same controller out that you would have gotten, but this rank condition on jbar for the collocated case, it's trivially true, which you'd expect. And for the non-collocated case it turns exactly into the strong inertial coupling condition of the other derivation. OK. So now we can do something interesting. I'm sorry to write so many symbols, but, because you're all falling asleep. Let's do something interesting with it. OK. So let's do the example I said. Let's take the cart-pole and make it go through some y trajectory, a sine wave or something like.

What would it take to make the cart-pole go like this for a long time? Speeding up and slowing down, but always actually moving in one direction. So my information is not going to be too pretty. I tried to come up with a really clever task space, five minutes before class, that would stay on the screen and regulate the output. I'm going to show you that leaves the screen quickly, but at least plot that it found the output. OK. So if I make y equal, this y, poor choice, is just the vertical position of the end point through the kinematics of the cart-pole.

Then, you could see it here. So why is just negative l cosine of the x2. I think this is my y desired is this y underscore. d Is my shorthand for desired. It's just some sine wave that's going to go between 1/4 of the length and 1/2 the length, or sorry 3/4 of the length. It would be unfortunate if I asked it to go somewhere. It couldn't go because things would blow up. My rank condition would stop being satisfied, but it's nice that it would at least alert me to that fact. OK. I put those exact equations I just showed you in a few lines of MATLAB. And I called it task space. There it goes, but after 10 seconds and it's quite far away from anything we're going to see. I can show you that a regulator that operate beautifully. OK

If you want to see a little bit more of it, I can zoom out. So it's really gone. OK. But again the output regulated beautifully. OK. So in retrospect the cart-pole might not have been the way to show you the power of partial feedback linearization, except for the fact that-- I mean if you really did care. Maybe you're a cart-pole bartender or something. You have to really keep some things steady. But, maybe, that's I guess it's still bad to drive away to infinity. Where you going? OK. So little dog is a good example, where we could nicely regulate the center of mass of the dog. And have it tip up and act like a simple pendulum, even though it had lots and lots of joints, and a passive joint in the middle.

OK so task space are actually a pretty fundamental concept in robotics. And they haven't seen them use that much in underactuated robotics, but they work just fine if you're going to use partial feedback linearization, which is bad. OK. Good. Well I was hoping to have an acrobat demo today, but Zach tells me were going to do it on Tuesday instead. And we're going to use it in the future throughout. So it's the same way I go back to the pendulum, we're going to go back and show how the better algorithms work on the acrobat and things like that. So you will see hardware again very soon. But not today.