

**RUSS TEDRAKE:**Welcome, back. Today we're going to break the mold of these one degree of freedom systems in a major way.

We're going to go to two degree of freedom systems. There are two sort of canonical underactuated systems. In fact, I'd say that the field of underactuated robotics spends most of its time thinking about a few canonical underactuated problems.

The two we're going to talk about today are the Acrobot and the cart-pole systems. The Acrobot is a two-link robotic arm. The only thing special about it-- it's operating in the vertical plane.

The only thing that makes it special is that somebody forgot to put a motor at the shoulder. So you've got a motor at the elbow and no motor at the shoulder. It's called the Acrobot because you can-- it's a little bit like an acrobat on a high bar that has to spin around and do tricks, even though they can't produce much torque with their wrist and they have to do it all with their waist.

The other one, the cart-pole system, you've probably seen it in an intro controls course. In our 6003 and our Signals and Systems course at the end of the year, they bring in a cart-pole and do a little demonstration. It's a cart with-- I made it a simple pendulum, since we thought a lot about the simple pendulums, on the cart.

You're allowed to push the cart sideways. But there's only a pin joint here holding the pendulum up. So you have to balance the pendulum with the cart by moving the cart.

Now, in our intro controls courses, they do linear control. At the top, they do some simple modeling, some pole placement in 6003. So they start it near the top. It stays near the top. They actually-- they put a wine glass or something on the top. They put a Christmas tree on the top. They do all these things, but they never start it from here because then the nonlinearity kicks in.

And that's why we need this course. We've got to-- that's a harder problem. So we're going to do the full Cart-Pole starting today. And we'll finish it on Thursday.

So the equations of motion of both of those are quite easy to derive. They take a half a page, and they're in the notes. Both of them are nicely described by the manipulator equations that I introduced in the first lecture. So let's work in the manipulator equation form.

So if you remember, I said that most of the systems we care about can be described by these equations. It happens that both of these systems are sort of trivially underactuated. They've got one actuator and one passive joint, so  $B$  turns out to be-- well, I've actually got it to be 0 and 1 for the Acrobot and 1, 0 for the cart-pole. But they're both sort of the standard form.

So the first thing I think that everybody does with these systems and that we have to discuss is, let's see if we can make it balance at the top. The task in both of these cases is to take the system from some arbitrary initial condition and get it up to the top, and balance. It turns out that, even though the systems are underactuated, you can do that. Just to show you the-- to help your intuition here, this is some Acrobot video from the web. They have a belt drive going down to their elbow motor and a big motor up here, like very big motor up here.

[VIDEO PLAYBACK]

And if you're willing to sort of pump energy up through the second link, then you can--

- Oh!

**RUSS TEDRAKE:**They were very excited. [LAUGHTER] Then you can add energy to the system, get it to swing around. And then the cool thing is, you can stabilize it at the top, which is actually pretty surprising. So this one took however many pumps, got itself the top, balanced, they said, oh!

So in this class, we're going to do better. So this is just a teaser. But here, if you were to take your optimal control tools and try to solve the same problem, the exact same system. Now it's in MATLAB. But we're going to do a lot better by thinking about sort of minimum time or even LQR solutions to get to the top. I think that's pretty elegant. So single pump, and then it's up. And of course, it will depend on your torque limits and all these things, but I think we can do very elegant control on these kind of problems.

OK, so I want to start by thinking about balancing at the top. So it's not obvious, if I've got a system with a motor at the elbow and no motor at the shoulder, that I could balance at the top. You'd think that, if I fall with my elbow motor, I've got nothing immediately to correct it. So how the heck do I stabilize that fixed point at the top?

Well, it turns out you can. And let's look at that. You've probably seen linearization before. It turns out, if you're linearizing things from the manipulated equations, it's pretty elegant. So I'm going to do that quickly here.

So our system that we're working with here is just an  $\dot{x}$ . It equals  $f$  of  $u$  in state-space form. But it happens because it's from the manipulator equation, that if we choose  $x$  to be  $q$ ,  $\dot{q}$ , like we always do, then  $f$  of  $x$ ,  $u$  has this sort of block matrix, a block vector form here, which is--  $\ddot{q}$  is now  $H^{-1}q$ ,  $Bu - Cq\dot{q}$  minus  $G$ . I just solve that for  $\ddot{q}$ .

We know that  $H$  is-- an inertial matrix, it's always uniformly positive definite. So for all  $q$ 's, it's positive definite. So I can take its inverse, and I get that solution for  $\ddot{q}$ . So the derivative of  $x$  is-- this is  $\dot{q}$ , sorry. It looks like a line, but it's a dot. And now we want to think about linearizing that system around a fixed point.

The way we're going to do that, of course, is taking a Taylor expansion. I'm going to take my  $\dot{x}$  and say it's approximately equal to  $f$  of  $x$  at the fixed point,  $u$  of  $x$  at the fixed point, plus partial  $f$ , partial  $x$ , with  $x$  evaluated at the fixed point,  $u$  evaluated at the fixed point, times  $x$  minus  $x^*$ , plus partial  $f$ , partial  $u$ , evaluated at the fixed point.

And it turns out for these equations, at a fixed point, that's really not a hard thing to compute. So can we do that quickly? So what's this term, first of all?

**AUDIENCE:** 0.

**RUSS TEDRAKE:**It's going to be 0, right? If we're at a fixed point, then the derivative at that fixed point better be 0. So this guy just disappears. Partial  $f$ , partial  $x$  we'll look at real quick. Partial  $f$ , partial  $u$  turns out to be not too hard.

Let's do a partial  $f$ , partial  $u$  first. It's even easier. So this is a vector, right? So I'm going to end up with, in general, a matrix of here, which contains the terms partial  $q$  dot, roughly, partial  $u$ , and partial  $q$  double dot, partial  $u$ . So what's partial  $q$  dot, partial  $u$ ?

**AUDIENCE:** 0.

**RUSS TEDRAKE:** 0, right? Partial q double dot, partial u, well, that turns out to be-- even though there's a lot of matrices flying around here with possibly nonlinear things inside, everything's linear in u. So that's actually pretty easy to write too. That turns out to be H inverse B. And this whole thing is going to be evaluated at our-- u doesn't matter in this case-- but evaluated at our fixed point.

OK, what about partial f, partial x? So now x is a bigger thing. So I'm going to have partial q dot, partial q here. This is sort of a block matrix form I'm using here-- partial q dot, partial q dot here, and then partial q double dot, partial q, partial q double dot, partial q dot. What's this?

**AUDIENCE:** 0.

**RUSS TEDRAKE:** 0. What's this one?

**AUDIENCE:** 1.

**RUSS TEDRAKE:** 1, or more generally, I, yeah. q double dot, partial q-- we need to use our chain rule. This is, something depends on q times something else that depends on q. So it's going to be partial H inverse, partial q, times Bu minus Cq dot minus G, plus H inverse, partial of that whole inside.

And H is potentially a little messy. H inverse is probably more messy. But it turns out this is going to be very simple, again, for us. So I claim that this term at the fixed point has also got to be 0. Do you buy that? Yep.

This thing has got to equal H, q double dot. H is positive definite. So for q double dot to be 0, this had better be 0. So that whole thing goes to 0. So this term-- or we don't have to do partial H inverse, partial q-- great.

And this one, again, is actually pretty simple. In the very first lecture, I did use B as potentially a function of q. But for these examples, it certainly isn't. It's just a constant. So this doesn't have any dependence on q. C does depend on q. G does depend on q. But at the fixed point, q dot had better be 0, if it's a fixed point. So the only term that actually survives out of this whole potentially scary thing is-- yeah, good-- is negative H inverse, partial G, partial q.

Take the derivative with respect to q dot. Again, this term is 0. So no matter-- H inverse, partial q dot doesn't matter. And then what in here depends on q dot? Well, C depends on q dot, both directly and internally. So we end up with partial-- so H inverse, partial C, partial q dot, times q dot, plus-- I'll do the whole thing as minus here-- plus C. But again, q dot is 0 at the fixed point, so it ends up-- this whole scary thing reduces to H inverse C.

**AUDIENCE:** So G is [INAUDIBLE] of [? theta? ?]

**RUSS TEDRAKE:** That's correct, yep. Those are our gravitational terms, yeah. OK, so this whole potentially scary thing works out to be 0, I, negative H inverse, partial G, partial q, negative H inverse, C. OK? There's a lot of beauty in the manipulator equations. It's a very nice middle ground between sort of any arbitrary nonlinear system, but it's got enough structure that you can play a lot of tricks like this. And often, things simplify. So I actually think it's a beautiful representation that we're lucky to have in robotics.

OK, now I've got this form. That's a linear system here, right? I've got-- if I just call this thing A and this thing B, then I've got  $x$  equals  $Ax$  plus  $Bu$  here. And if you prefer to really make it linear, then let's define  $\bar{x}$  to be  $x$  minus  $x^*$ ,  $\bar{u}$  to be  $u$  minus  $u^*$ , to put the origin at the fixed point. Oops. Now,  $\bar{x}$  dot is just  $x$  dot minus-- this thing's-- that's 0, so it's just  $\bar{x}$  dot. So I could equally write this as  $A\bar{x}$  plus  $B\bar{u}$ .

OK, so given the manipulator equations for the Acrobot, for the cart-pole, it's trivial to find a local approximation of those dynamics which is valid around the fixed point we're trying to stabilize. And if I have that, then I could play some of the linear control games that we started with. So the first thing to try, let's do LQR at the top.

It turns out LQR at the top just works really well. It's not too surprising, I guess. Actually, before we do LQR, let me take a minute and actually decide if we should-- if it's proper to do LQR. So there's a condition in LQR that had to be met in that derivation that I threw at you.

If the infinite horizon cost, that integral wasn't bounded, if your system didn't get to the origin, then the LQR cost would be infinite, and the LQR derivation would break. If your system-- if you cannot, with feedback, drive your system so that  $x$  is at the fixed point, then the LQR cost function will accumulate cost forever and blow up.

So really, the first thing, before we apply LQR, we better take a second to decide if we think the system can get to the origin with feedback. And that condition is called controllability. And controllability is a very powerful concept. And I want to make sure you understand the relationship between controllability and underactuation.

So the question is-- for this system, the question is, if I have my linear system and I started in some initial conditions that are non-0-- otherwise it's not so interesting-- if my initial conditions are non-0, can I design a feedback law, find some actions,  $u$ , that will drive my system to 0 in a finite time? So more generally, the definition of controllability says-- let's say,  $x$  at 0 equals some initial conditions to  $x$  at some final time to be some other initial conditions, given unbounded actions in a finite time.

So controllability is actually the thing we care about in life. For nonlinear systems, controllability is actually incredibly hard to evaluate. Most systems tend to be controllable. But it's a hard thing to evaluate for nonlinear systems. For linear systems, we have all the tools we could dream of to evaluate the controllability of the system.

So for linear controllability, it's sufficient to say,  $x$ ,  $t$  final equals 0. So a lot of times for linear systems, people just ask controllable-- say it's controllable if I can drive my system from any initial conditions to 0 in a finite time. And because everything's nice and linear, that's actually-- that's equivalent to the stronger definition for linear systems. In nonlinear systems, you have to evaluate every initial condition, every final condition, if you're not careful.

So what do you think? If the system is underactuated, then do you think it can be controllable? If I choose some place, some state for the Acrobot, and I choose a finite time, can you design a controller, potentially with really big actions, that gets me there in finite time?

**AUDIENCE:** No.

**RUSS TEDRAKE:** Tell me why. Tell me why you say no.

**AUDIENCE:** Well, it can [INAUDIBLE] you potentially [INAUDIBLE] stabilize for Acrobot. So regardless of the fact, which [INAUDIBLE] you can reach that [? theta. ?]

**RUSS TEDRAKE:** But there's nothing to say I can't leave that state and come back in some very finite, very small amount of time.

So you're saying, if I ask you to go here at some small amount of time, and I start right here, then I can't be there in some small amount of time. But what if I put in so much actuation that I go like this and I'm back there?

**AUDIENCE:** [INAUDIBLE] have to stay in the final state?

**RUSS TEDRAKE:** No, it does not have to stay in the final state.

**AUDIENCE:** So underactuated implies that the local dimensionality of the reachable state-space is lower than the full dimension, right? So if you just shrink your finite time to an arbitrarily small time, we won't be able to reach states outside that [INAUDIBLE].

**RUSS TEDRAKE:** I think that-- so this is exactly the point. So what he said is, he says, the underactuation is-- I'll use the word instantaneous. It's an instantaneous constraint on what you can do. At any instant in time, I can only produce accelerations in a certain direction. So how can I possibly, in some-- if I make my finite time small enough, how can I possibly get there? Well, finite time is actually different than 0 time. And the actions can potentially be huge.

But actually, most-- a lot of the underactuated systems are controllable. So we'll see it carefully in the linearization of this. But actually, controllability-- if you get one thing out of this lecture-- and I'm going to write at the end of the lecture again on the board-- there's a difference between controllability and underactuation. A lot of the underactuated systems are, in fact, controllable. And that's what makes-- gives us a few more things to talk about in the class.

So let's see if I can make that point to you. So how do we talk about controllability, in a linear system even? So what are the tools people use for controllability in a linear system? If you used them, call them out. Who's used controllability tools? Yeah.

**AUDIENCE:** There's the matrix C times B, C times A, times B, write it all out, then find the rank. Would that be true?

**RUSS TEDRAKE:** Good, there's a controllability matrix.

**AUDIENCE:** There's a controllability Gramian, I think.

**RUSS TEDRAKE:** Awesome, there's a controllability Gramian, which turns out to be almost exactly the-- what came out of our LQR derivation. Those are the two big ones. So there are controllability matrices, controllability Gramians. I'm going to say a minute about it.

But if you care about-- we can actually-- both of those are a little bit unintuitive, actually. And the proofs are-- the proof of the Gramian's not bad, but the proof of the-- the derivation of the controllability matrix is a little bit of black magic. So I decided instead to let's do a simpler case where we can actually understand it. But it'll be a less general result.

So let's look at the  $\dot{x} = Ax + Bu$  system. I'm going to make our derivation easier by making an assumption that the eigenvalues of  $A$  are all unique. If you're willing to make that assumption, we can see a lot of things. The more general derivations don't have that, but require black magic.

OK, so if you remember, our eigenvalue thing, an eigenvalue means that multiplying A times that is just the same as multiplying a scalar times n. If I compose all the-- sorry, these are eigenvectors. That's an eigenvalue. If I compose them all into a matrix form, I can see A times V, where V has all the eigenvectors as columns, is equivalent to-- well, let's write it like V times lambda, where lambda is a diagonal matrix, which has lambda 1, lambda 2, and so on.

The cool thing about-- the reason we assume that there's no repeated eigenvalues is that it implies that all of the-- eigenvalues are unique implies that all the eigenvectors are actually unique and that they span the space. And it implies that V inverse exists. As soon as you have repeated eigenvalues, you don't have that simplification. You have to do repeated roots and things.

But in that simplification, we can switch to modal coordinates. So let's-- if V is full rank, then I can just change coordinates from x through V inverse to some other coordinate system r. Why is that a good idea? That's a good idea, because then r dot is just going to be-- so V inverse times x dot, which is A times Vr. That's just substituting this into the x dot equals Ax plus Bu.

But what's V inverse AV? If you look at this, that's just our diagonal matrix. So on the eigen-- in the modal decomposition, the systems evolve without coupling. I can write-- component-wise, I could say,  $r_i$  dot, the i-th component is just  $\lambda_{ir}$ , plus some contributions from this guy.

So now I'm looking at my-- in these modal coordinates, I'm saying that the result of applying A-- this is the same thing I wrote before. This is the reason we can make-- in these phase plots, we can find the eigenvectors and just talk about the dynamics on those eigenvectors, same thing. It just says that, on the eigenvectors, the dynamics are just the eigenvalues. And then they've got this impulse, this input from the control actions.

So now let's think about what it would mean to be controllable in this sense. What kind of conditions would you want to say that the system is controllable? Yeah.

**AUDIENCE:** We can change  $r$  dot using  $u$ .

**RUSS TEDRAKE:** You can change  $r$  dot to do anything you want using  $u$ . You'd like to be able to say, make it act like the eigenvalues were arbitrarily fast, for instance, or arbitrarily unstable, if you chose to do something so silly.

So what is required to do that?

**AUDIENCE:** [INAUDIBLE] from the [? beta ?] values. There'd have to be some non-0's [INAUDIBLE].

**RUSS TEDRAKE:** Excellent. OK, so imagine you only have a single thing you're trying to control and lots of inputs. Then it should be sufficient that, if any one of those betas is non-0, that should be enough. OK, now let's say you have multiple things you're trying to control. You start having to worry about whether you can use  $u$  to control eigenvector 1 and eigenvector 2, both at the same time.

But it turns out, because we assumed we're in the case of distinct eigenvalues, if you think really hard-- and I will write a little bit more about this-- but it turns out it's still OK to just have one thing that can control you. Because things are converging at different rates, it's actually sufficient to be able to control-- if you can control each of them independently, then you can actually control them all.

So the condition turns out to be, for all  $i$ , there exists a  $j$ , such that  $\beta_{ij}$  is not equal to 0.  $\beta_{ij}$ , again, is my  $B$  matrix, but also permuted by this  $V$  inverse. So if I was looking at whether the system was underactuated, if  $B$  wasn't full rank, I'd be hosed. I'd be underactuated.

But this is actually a much less strict condition. I say, I only have to have one of my-- in my eigen modes, I have to be able to control each of them. So that's our first sort of glimpse at how you could imagine an underactuated system being completely controllable.

**AUDIENCE:** Is it the same as saying that, if you have [? any ?] dimension that you want to control, then  $\beta$  should have at least  $n$  different eigenvalues? [INAUDIBLE].

**RUSS TEDRAKE:** So excellent-- so this actually says-- for instance, in the limit, let's take the case where you have 100 degrees of freedom and one actuator. As long as  $B_{ij}$  is non-0 for all of those, then it says, with my one actuator, I could control-- I can drive that 100 degree of freedom system to the origin. That's a great question, a great example. So it does not have that same rank condition on  $B$ .

**AUDIENCE:** So as long as none of your eigenvectors are in the null space [INAUDIBLE].

**RUSS TEDRAKE:** Right. It's the-- I have to think about what  $V$  inverse is, but I think that's right. I think that's right. He said-- so you don't want  $B$  to be in the null space of the inverse. Right. Do I see another question over there? No?

OK, so 100 degree of freedom robot, one actuator, there's a chance that you can control it. Now it doesn't say anything about the trajectory you're going to take to get there. It might be really hard. But there's a chance you can control it.

There's lots of ways to see this. Let's leave it at that, if people are satisfied with that. Let's not-- I won't do my second derivation here. But let's get straight to the example.

So now, that says that, if I take my  $Ax + Bu$  representation of the Acrobot,  $B$  is going to be low rank. It's going to be-- so I used  $B$  actually twice already today. Did you notice that? Anybody catch me on that?

I used  $B$  in the manipulate equations, then I used  $B$  for the block form. And I swear I've lost sleep trying to figure out if I could put another letter in there, but I'm not happy with any other letters, because they both-- they almost mean the same thing. But I did slip that one past.

So this is  $B$  in the linear form. So it's not just 0, 1. It's  $H$  inverse times 0, 1. So it's going to have some 0's. Sorry, it doesn't-- so it's going to be a vector again. It actually doesn't have to have 0's. And in fact, it probably won't have 0's. I misspoke. This is going to be  $H_1, 1, H_1, 2$  in the Acrobot case,  $H$  inverse 1, 1.

The question is-- it's still low rank because it's a column. It's not a full matrix, so it's still underactuated. And can I drive that thing with LQR to the top? We just said that it's possible that these things are still controllable. The tests that you can use, the user's guide to controllability-- there's this funky test which says, if I take a matrix, a controllability matrix, which is the  $B$  matrix,  $AB$ -- these are just cascaded in columns here,  $A$  squared  $B$  up to the  $A$  to the  $n$ ,  $B$ , where  $n$  is the number of states.

**AUDIENCE:** Would it be  $n$  minus 1? [INAUDIBLE].

**RUSS TEDRAKE:**n minus 1-- thank you, yeah. Good, yeah. It turns out that if this matrix is still-- is full row rank, then the system is controllable. There's actually-- the derivation of that is in your notes that I'll post. It's not very hard, it's just a matter-- there's all these forms for e to the AT, the matrix exponential. And it involves one of the forms that kind of comes out of nowhere. And so I won't go through the derivation on the board.

But it turns out that it's sufficient to check the rank of this matrix. And that'll tell you if the system's controllable. And this is sort of primal enough and important enough that MATLAB has a function for it. So you can call-- I think it's ctrb, right? Is that what it is?

And then check the rank of it, and you could check if your systems are controllable. It turns out, if you linearize the dynamics of the Acrobot, you linearize the dynamics of the cart-pole, you pop in the A and B, you get a full rank matrix out. So they're both controllable, even though they're underactuated around the top.

**AUDIENCE:** Would these be in [INAUDIBLE]?

**RUSS TEDRAKE:**Yes. It's the linear form. Maybe I should call it B prime or something like that. But still, it feels unnatural to use either-- anything else in either case.

**AUDIENCE:** So these matrices are calculated for every single point, or--

**RUSS TEDRAKE:**So I just do it-- I evaluate these things once at the x star. So it's partial f, partial x, evaluated at x star.

**AUDIENCE:** So when you say something-- a system is controllable, because when we--

**RUSS TEDRAKE:**All I mean is that-- all I can show with the rank condition check in MATLAB is that the system linearized-- the linear system that approximates the Acrobot linearized around the top is controllable. Mm-hmm.

**AUDIENCE:** So controllability is defined [INAUDIBLE]?

**RUSS TEDRAKE:**No, controllability is a property of the system. The linearization, the linear system is only a relevant approximation of the Acrobot around a given state. And I'm saying that the linear system-- you can evaluate the linear system anywhere it's controllable. I can go from any initial condition. But it's only a relevant example-- it's only relevant to the Acrobot if it's close to the fixed point.

**AUDIENCE:** So that definition actually is for all the states.

**RUSS TEDRAKE:**That's right. So yes, if I can say this stronger thing for the nonlinear system, for the Acrobot, in order to say that, I would have to say, for any state, this is true. I can't say that. I went to a weaker form by looking just at the linear system. Mm-hmm. That's pretty big. Is that big enough?

**AUDIENCE:** Mm-hmm.

**RUSS TEDRAKE:**Yeah? OK. So let's take our A and B matrices that we got from doing the linearization around the fixed point. And if you like, if it helps, I can sort of show you how that goes. So very literally, I take my manipulator equations, I get an H, C, G, and B.

I take the gradient-- the partial G, partial q, I compute that. And then my A is negative inverse partial G, partial q. And then the-- C happens to be 0 for the Acrobot. I evaluated at the top, but that's not a general, so that's why I don't have the negative H inverse C there, and then my inverse H, B. So that gives me an A and a B.

And I can write my LQR controller to balance it at the top, by literally getting a and B from the linearized system, calling LQR. I chose-- the LQR syntax is A, B, q, r. So I chose q to be a diagonal matrix, 10, 10, 1, 1, just saying I penalize position errors 10 times more than velocity errors because it happens-- with units, you tend to do that. And r is 1.

And it's in a persistent loop just so I don't compute-- I don't call LQR every time I call my function. I only call it every time I start the system. And then my control law is u equals negative k, x minus x desired.

Now, you're going to do sort of the similar thing to what I'm doing for the Acrobot you're going to do for the cart-pole for your problem set. I will bet that half of you will forget to subtract out x desired at least once. I know I do all the time. So remember the minus x desired.

If I put that at the top, if I put initial conditions near the top, and I run it, look what happens. Oops. OK, I'm going to start it-- every time it flashes it's going to be new initial condition. It actually goes pretty far from the top and gets back. Wow, that's a good one. If I-- I'm just choosing Gaussian random variables at the top. So if we watch long enough, we might see one fail catastrophically.

So the LQR for the linear system will stabilize it from any point because the linear system is a bad approximation. For the nonlinear, if I started it way down at the bottom it's just going to go nuts. Wow, that's pretty good. Now, you might notice that my second link is pretty big compared to my first link. That helps. And you'll see why in a few minutes. But it's pretty good.

**AUDIENCE:** Does it have unbounded torque?

**RUSS TEDRAKE:** It does have unbounded torque. Whoa. [LAUGHTER] That's pretty good. It does have unbounded torque, yep. If I were to saturate the torque, it probably wouldn't do as well.

Now, the cool thing is-- I can stop that. Stop. These MATLAB timers don't like to stop. So I get these huge excursions from the upright in my balancing. But it's not actually because I started with crazy initial conditions. If you noticed, it wasn't like the plot was going, oh, and starting over here and then coming up. It was actually going like this.

So if you look at a time trajectory-- let's see if I got lucky. OK, almost all of them are like this. Are those lines dark enough to see? Yeah? The initial conditions are actually pretty small. But in order to stabilize the system, it actually goes way away from the fixed point and comes way back, big time. This is like-- this is the velocity of 18 radians per second or something. And then it finds its way back.

So for that reason, you might easily sort of, in LQR, say, OK, my linearization is good here, so any initial conditions here should work. But that's not actually true because your LQR might easily drive you further away before it comes back. If you were to-- if you're a linear controls guy, and you were to do-- this is a multi-input, multi-output system. But if you try to find the poles and 0's, that's what's this going to-- where are the 0's-- is this going to be?

**AUDIENCE:** [INAUDIBLE]

**RUSS TEDRAKE:** Yeah, there's three 0's in the right f plane for the Acrobot actually. It's a nonminimum phase system. The cart-pole has one 0 in the right f plane. It's not a general property of underactuated systems. But sometimes, in order to do this with less actuators than you might like, you have to do crazy things to get back to where you want to go. So I can get-- and I could-- if I tightened my time limit, the things I would do would be even crazier. But I could still do it.

It's actually-- just to say it, without really teaching it, but if you did care about sort of the basin of attraction of these systems, if you wanted to do as well as you possibly could with a linear controller on the nonlinear system, you probably wouldn't do what we just did. There's better tools from robust control, which would allow you to sort of design a linear controller but explicitly reason about how nonlinear the thing gets when you're away. And you can design a linear controller that has a bigger basin of attraction than if you just don't reason about the nonlinearities at all. So you can put a bound on how nonlinear things are and do a robust control synthesis, and get, in some sense, a better controller. You would have lower performance, potentially, but it would work better on the nonlinear system, bigger basin of attraction.

**AUDIENCE:** [INAUDIBLE]

**RUSS TEDRAKE:** Oh, my fixed point was  $\pi$  0, 0, 0. Or-- yeah. So those are all going to 0, that one's going to  $\pi$ , as they should. Yep. OK, excellent.

So LQR works, right? And it works-- this is actually sort of a problem, in my mind. Because that works and it works so well. And so many different systems, that's why people haven't thought about nonlinear control enough, in my mind. It's sort of unfortunate that that works so darn well, because sometimes that's all people do.

So let's think about if we're a little bit further from the fixed point, just to-- maybe I should even make the point-- in other words, you'll probably ask me if I don't do this, because you always test me on the limits of where my things work here. So what if I were to start it far away from the fixed point?

Let's be more dramatic. And let's just do it once. Oh, see, that's bad. When it takes that long, that means the integrator's choking because it can't simulate things right. So it's just complete nonsense if it's too far from the linearization point.

So what if we want to do control away from that fixed point? Then nothing I just said helps if I'm too far away from that fixed point. So what do we do? Well, you don't have to throw out linearization completely. We talked about, in the first lecture, how the underactuated systems are the systems that are not feedback linearizable. That's what distinguishes them.

They're not feedback linearizable. I can't just turn the nonlinear system into a linear system. But they are partial feedback linearizable. So if you want to stick to your guns and do feedback linearization, you can do half the work. So it actually is pretty elegant how that works out.

OK, to keep things fresh, let's do it on the cart-pole instead of the Acrobot. We've been talking about the Acrobot a lot. But I promise I wasn't going to do the derivation of the equations of motion, and I won't. But it turns out the result of the cart-pole is simple enough I can write it real quick.

The equations of motion for the cart-pole are  $mc$  plus  $mp$ . And these are in your notes. You don't have to write these down. I want you to see where the next line comes from.

OK, well, that's a reasonable thing we might get out of the Lagrange equations. I've got a force on my cart. I've got a 0 in the other equation. You can see how this could be easily separated into the manipulator equations. But since I'm going to be manipulating some of these things, let me just sort of arbitrarily set all the parameters to 1.

Let's just-- so-- it's easy to repeat these for the real equations. But let's just do this. And I'll get a  $2x$  double dot plus  $\theta$  double dot,  $c$ , which is cosine  $\theta$ --  $c$  is enough-- minus  $\theta$  dot squared  $s$  equals  $f$ . OK, and then  $x$  double dot  $c$ , plus  $\theta$  double dot, plus  $x$  equals 0. Now, those I can work with.

**AUDIENCE:** It says  $G$  [INAUDIBLE].

**RUSS TEDRAKE:** Yeah, that's just to save my handwriting. It's not to be physically accurate. It doesn't change the structure of the equations. If you like, you could set-- I bet I could come up with a parameterization which would keep  $G$  at around 10 and do OK, but--

**AUDIENCE:** [INAUDIBLE]

**RUSS TEDRAKE:** There you go, there you go. OK, so given these equations, can I make  $x$  double dot and  $\theta$  double dot do whatever I want? So no. We said that the feedback linearization trick required that  $B$  inverse, in general. And so we couldn't do that. But it turns out I can do something.

So what are these equations? This is a cart moving around with a pendulum on it. And I'm pushing the cart, and the pendulum's dangling away. So what would feedback linearization-- what would a partial feedback linearization mean?

Well, if I know the dynamics of the pendulum and I know the state of the pendulum, and I can control the force on my cart, then it's pretty reasonable to think that I could-- whatever force the pendulum was applying to push my cart around, I could just exactly cancel that out.

So I could turn my cart dynamics to sort of just do whatever I want and just cancel out the effect that that pendulum's adding to me. That seems reasonable. And that's called a collocated partial feedback linearization. I'm going to write PFL from now on.

It's collocated because the state I'm trying to linearize is the same one where the actuator is sitting. It's collocated. The state I'm linearizing is collocated with my actuators. So my goal is to make  $x$  double dot have the dynamics, whatever dynamics I choose--  $x$  double dot desired, let's say. So let's see if we can do this by manipulating the equations a little bit.

OK, so I can figure out how  $x$  double dot and  $\theta$  double dot are related with that second equation. So let's get rid of  $\theta$  double dot. I can see  $\theta$  double dot had better be negative  $x$  double dot  $c$  minus  $s$ . And if I insert that into the first equation, then I get  $2x$  double dot minus  $x$  double dot,  $c$  minus  $s$ . Oops.

Minus  $sc$  minus  $\theta$  dot squared  $s$  equals  $f$ . That means, if I apply the control law,  $2$  minus  $c$  squared,  $x$  double dot desired, minus  $sc$  minus  $\theta$  dot desired  $s$ , that implies that  $x$  double dot equals  $x$  double dot desired, like we wanted. And  $\theta$  double dot ends up doing something coupled.

But that's sort of the resulting dynamics. I didn't actually plan for that. That's just whatever I got out. Does that make sense? That's just me saying-- if you think about the controller here, that's just taking the terms that

Are going to be contributed to the dynamics by the pendulum and canceling them out by applying exactly the opposite forces. And the result is that my  $x$  moves exactly however I want it to do. So feedback linearization isn't dead if you have an underactuated system. But you can't feedback linearize the whole system. You can do this collocated feedback linearization.

Now, the cooler thing is, you can actually, often, also feedback linearize the passive joint. So it's pretty logical that I could move the cart in such a way that I can't allow the pendulum dynamics. It's a little less intuitive that I can make the pendulum dynamics do whatever I want with the cart. But you can, most of the time.

OK, so non-collocated means I'm going to use one of my actuators to control-- feedback linearize one of my passive joints. So now let's see if we can make  $\theta$  double dot do whenever we-- bend to our will. It turns out the manipulation is almost exactly the same.

Algebraically, it's not surprising that I can do either one. I've got my equations of motion here. I've got-- both  $x$  double dot and  $\theta$  double depend on my force. And they're coupled here. So sure, I can control either one of them. Physically, it's a little less intuitive. But algebraically, it's just as obvious.

OK, so let's just do the opposite one. So  $x$  double dot had better be  $\theta$  double dot, plus  $s$  over  $c$ , that whole thing negative, based on that second equation. And so I get  $2$  over  $c$ ,  $\theta$  double dot plus  $s$ , negative on there, plus  $\theta$  double dot  $c$ , minus  $\theta$  dot squared  $c$ -- oops,  $s$ -- sorry-- equals  $f$ .

Yeah, so sure, so if I apply  $f$  equals is the controller-- what's the best way to write this?  $c$  minus  $2$  over  $c$ ,  $\theta$  double dot, desired, minus  $\theta$  dot squared  $s$ . Is that right?

**AUDIENCE:** Minus 2 SOC from the first [INAUDIBLE].

**RUSS TEDRAKE:** Good, yes. Minus 2 tan theta. Good, thank you. Cool. So that's a much less intuitive result, I think. But it's a much more powerful one. It says, if I wanted to directly control the pendulum, I can. I have to give up something.  $x$  double dot, then, is going to end up being-- the resulting motion of the cart could be a little strange.

It's going to be whatever this  $\theta$  double dot desired plus  $s$  over  $c$  looks like. But who cares? If I'm just trying to keep the cart up, that's OK. In your 6003 type demos, they also worry about not running into the rails, which is important. But to first order, this is a good thing. What did I gloss over?

**AUDIENCE:** Theta goes to pi over 2.

**RUSS TEDRAKE:** Yeah, right? So I put a cosine on the bottom here without being careful about that. So what is that? What is that physically related to?

**AUDIENCE:** If your pendulum goes flat, then [INAUDIBLE].

**RUSS TEDRAKE:** Exactly. When my pendulum is directly sideways, then suddenly, nothing I do with the cart is going to control the accelerations of that pendulum. So instantaneously, you lose the ability to control that. If you're going to swing up from the bottom to the top, then you go through that. So who cares? So stop doing control for a second, and then you'll get back to a place where you can control it again.

So that sort of says everything, I think, that your intuition should relate about these things. And then for the Acrobot, it's sort of similarly surprising, but I can use my elbow torque to feedback linearize my shoulder torque-- same thing. So if I wanted to make it do whatever I want, really, then I can do that. I might have to spin like crazy or do something nuts. But I can make my passive joint do whatever I want. It's kind of cool.

In fact, just to-- I want to show the slightly more general derivation of that or form of that, but just to make the point. So one of the ways we've been playing with Little Dog-- this is our robotic dog. So Little Dog has actuators at all the internal joints. It's got-- if you're just looking at it from the side, all you care about, it's got one in the knee, it's got, actually, two in the hip-- but that doesn't matter here-- two in the other hip, one in the knee.

But the thing we're about to make it do is try to control the dynamics around the foot, which is just like the Acrobot. It doesn't-- the place where you might think you'd want it the most is where you don't have the actuator. So if you want to do something like this with your dog, then you've got a reason about the coupling, the inertial coupling, which is what we did here, that allows you to decide-- the slipping at the end is ugly and we didn't do that right.

But until the impact, we did a pretty good job, actually, of regulating the position of the dog, just using the controlled actuators and where our most essential variable was passive. It took a slightly more general form, which I'm going to show you on Thursday, to do that. But partial feedback linearization is sort of alive and well and useful in robotics.

Let me just-- I'll show you half of the-- or one of the slightly more general derivations of it, just because it's so easy with this algebra manipulation. So let's say I have manipulator equations of the form-- I'm just going to lump C, q dot, and G into a single term because they don't affect the derivation at all. And let's say that I've stacked-- I've reordered my equations of motion so that all of my passive joints are on top and all my actuated joints are on the bottom.

So these are all matrices. Let me call q1, the collection-- it's a vector of all the passive joints, and q2 all the active joints. Then if I break this up just a little bit, I can write the same equation as-- as that. I left these up the whole lecture thinking I was going to point to them all the time-- never did. I'm going to erase them now.

OK, so what is my collocated form? Collocated means I'm going to try to control q2. So let's solve for-- q1 is going to be H1, 1, inverse, H1, 2, q2 double dot plus phi 1. Am I allowed to do that, H1, 1, inverse? It takes a little bit of thinking, but it's actually OK. It's a positive definite matrix H. So it turns out the square diagonal entries are actually also have to be positive definite. So maybe take my word on that.

And then I can plug that in and see that, if I do tau-- let's see if I can just do it in the step here. Tau is going to have to be H2, 2, plus-- no, I missed a minus, didn't I? q1 is-- I missed a minus in here. Minus H1, 1, H1, 2-- I did that. I should have done it in two steps. It's H2, 1, H1, 1, inverse, H1, 2, q2 double dot desired, plus phi 2. Right?

In the non-collocated, we're going to have to solve for q2 double dot. That's going to be H1, 2, negative H1, 2, inverse, times H1, 1, q1 double dot, plus theta 1. I missed my theta 1 term in here somewhere. That should have been also in here. Yeah. OK, so the first step in the non-collocated, in general, is this H1, 2 inverse. Is that one OK?

**AUDIENCE:** [INAUDIBLE]

**RUSS TEDRAKE:** Not necessarily. I could have had a different number of actuated active and passive degrees of freedom. So let me write something that's a little bit better, the sort of pseudo inverse. And what matters-- the non-collocation, the partial feedback linearization is going to work if and only if that matrix is full row rank again.

So what does that mean? It means I can't use one active degree of freedom to control two or more passive degrees of freedom. I need to have at least as many actuators as degrees of freedom I'm going to try to control. That's reasonable. But it's a little bit more than that still. I actually have to have them inertially coupled in the right way.

So if I had two pendula on the table-- well, the table might have some dynamics. If I had two pendula bolted to completely independent bases, and I had an actuator on one and not an actuator on the other, that ain't gonna work. I can't make any math that's going to make that work.

So there has to be some inertial coupling between the two. So the rank condition on this is the condition of-- is sometimes called the condition of strong inertial coupling. The strong means that it's uniformly inertially coupled. It's just inertially coupled in some state. And if for all  $q$  this thing is full rank, then it's strong inertial coupling.

And there's even a more general form. So in general, you can-- and what we do in Little Dog is, we pick some combination of actuated and unactuated degrees of freedom. And actually, what we care about is a virtual degree of freedom, which is the center of mass. And I'll show you at the beginning of the next lecture the most general form of this.

But I can't put up these PFL equations without spending one minute at the end saying, PFL is still sort of bad, right? I don't really-- it works, and I use it because I want to control these robots. But I don't like feedback linearization. Feedback linearization is bad.

This is taking some beautiful nonlinear system that has beautiful equations and arbitrarily pumping in some potentially large amount of energy to squish those dynamics and bend them to your will. And that's good. That's the feedback way. But it's not the only way. So we do it when we have to. But it's better if you don't. Yeah.

**AUDIENCE:** Wouldn't you have [INAUDIBLE] errors in the [INAUDIBLE]?

**RUSS TEDRAKE:** It could be that, if you don't have the model perfect, that you could be doing bad things too. Any time you have large control gains going around, you're going to be sensitive. So don't always do this. But I do want you to know that you can do this if you so choose to take that path.

OK, cool, so on Thursday we will use these partial feedback linearizations and LQR together to make the Acrobot, the cart-pole swing up and balance. If all goes well and we don't have any License Manager issues, then we'll see it in class.