

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

RUSS TEDRAKE: I've been getting to play with this robot for a few years now-- three years of my life basically devoted to that robot. It was one of the most exciting, technically challenging, exhausting, stressful, but ultimately fulfilling things I've ever done. We got to basically take this robot, make it drive a car, get out of the car-- that was tough-- open the door, turn valves, pick up a drill, cut a hole out of the wall. Notice there's no safety harness. It's battery autonomous.

It has a walk over some rough terrain, climbed some stairs at the end. It had to do this in front of an audience. Basically, we got two tries. And if your robot breaks, it breaks, right? And there was a \$2 million prize at the end.

We wanted to do it not for the \$2 million prize, but for the technical challenge. And myself and a group of students, just like I said, absolutely devoted our lives to this. We spent all of our waking hours on this. We worked incredibly, incredibly hard.

So just to give you a little bit of context, DARPA, our national defense funding agency, has gotten excited about the idea of these grant challenges, which get people to work really, really hard. The self-driving cars were the first one. MIT had a very successful team in the Urban Challenge led by John.

And then it's unquestionably had transition impact into the world via Google, Uber, Apple, and John will tell you all about it. I think in 2012, DARPA was scratching their heads, saying, people haven't worked hard enough. And what's the new challenge going to be? And right around that time, there was a disaster that maybe helped focus their attention towards disaster response.

So ultimately, it was October 2012 that everything started with this kickoff for the DARPA Robotics Challenge. The official challenge was cast in the light of disaster response using the scenario of the nuclear disaster as a backdrop. But I think really their goal was to evaluate and advance the state of the art in mobile manipulation.

So if I'm the funding agency, what I think is that you see hardware coming out of industry that

is fantastic. So Boston Dynamics was building these walking robots and the like. This one is the one we've been playing, Atlas, built by Boston Dynamics, which is now Google.

AUDIENCE: Alphabet.

RUSS TEDRAKE: Alphabet, yeah. And then I think from the research labs, we've been seeing really sophisticated algorithms coming out but on relatively modest hardware. And I think it was time for a mash up, right? So they were very interesting in the way they set up the competition. It wasn't about making it a completely autonomous robot. There was there was a twist.

You could have a human operator, but they wanted to encourage autonomy. So what they did is they had a degraded network link between the human and the robot and some reward for going a little bit faster than the other guy. So the idea would be that if you had to stop and work over the degraded network link and control every joint of your robot, then you're going to be slower than the guy whose robot is making the decisions by itself.

That didn't play out as much as we expected, but that was the setup. That set up a spectrum where people could do full teleoperation, meaning joystick control of each of the joints if they wanted to. And maybe the goal is to have complete autonomy, and you can pick your place on the spectrum.

So MIT, possibly to a fault, aimed for the full autonomy side. The idea was, let's just get a few clicks of information from the human. Let the human solve the really, really hard problems that he could solve efficiently-- object recognition. Scene understanding-- we don't have to do that, but a few clicks from the human can communicate that. But let the robot do all of the dynamics and control and planning side of things. So those few clicks should see nearly autonomous algorithms for perception, planning, and control.

OK. So technically, I don't intend to go into too many details, but I would love to answer questions if you guys ask. And we can talk as much as we want about it. But the overarching theme to our approach when we're controlling, perceiving, everything is to formulate everything as an optimization problem.

So even the simplest example in robotics is the inverse kinematics problem where you're just trying to decide if I want to put my hand in some particular place. I have to figure out if I have a goal in the world coordinates. I have to figure out what the joint coordinates should be to make that happen.

So we have joint positions in some vector q , and we just say, I'd like to be as close as possible. I have some comfortable position for my robot. We formulate the problem as an optimization-- say, I'd like to be as close to comfortable as possible in some simple cost function. And then I'm going to start putting in constraints, like my hand is in the desired configuration.

But we have very advanced constraints. So especially for the balancing humanoid, we can say, for instance, that the center mass has to be inside the support polygon. We can say, we're about to manipulate something. So I'd like the thing I'm going to manipulate to be in the cone of visibility of my vision sensors.

I'd like my hand to approach. It doesn't matter where it approaches along the table, maybe, but the palm should be orthogonal to the table and should approach like this. And we put in more and more sophisticated collision avoidance type constraints and everything like this, and the optimization framework as is general and can accept those type of constraints. And then we can solve them extremely efficiently with highly optimized algorithms.

So for instance, that helped us with what I like to call the big robot little car problem. So we have a very big robot. It's a 400 pound, six foot something machine. And they asked us to drive a very little car to the point where the robot physically does not fit behind the steering wheel-- impossible. It just doesn't kinematically. Torso's too big, steering wheel's right there, no chance.

So you have to drive from the passenger seat. You have to put your foot over the console. You have to drive like this, and then our only option was to get out of the passenger side. So that was a hard problem kinematically, but we have this rich library of optimizations. We can drag it around. We can explore different kinematic configurations of the robot.

But we also use the same language of optimization and constraints, and then we put in the dynamics of the robot as another constraint. And we can start doing efficient dynamic motion planning with the same tools. So for instance, if we wanted Atlas to suddenly start jumping off cinder blocks or running, we did a lot of work in that regard to make our optimization algorithms efficient enough to scale to very complex motions that could be planned on the fly at interactive rates.

So one of the things you might be familiar with-- Honda ASIMO is one of the famous robots that walks around like this, and it's a beautiful machine. They are extremely good at real time

planning using limiting assumptions of keeping your center mass at a constant height and things like this. And one of the questions we asked is, can we take some of the insights that have worked so well on those robots and generalize them to more general dynamic tasks?

And one of the big ideas I want to try to communicate quickly is that even though our robot is extremely complicated, there's sort of a low dimensional problem sitting inside the big high dimensional problem. So if I start worrying about every joint angle in my hand while I'm thinking about walking, I'm dead, right? So actually, when you're thinking about walking, even doing gymnastics or something like this, I think the fundamental representation is the dynamics of your center of mass, your angular momentum, some bulk dynamics of your robot, and the contact forces you're exerting on the world, which are also constrained.

And in this sort of six dimensional-- 12 dimensional if you have velocities-- space with these relatively limited constraints, you can actually do very efficient planning and then map that in a second pass back to the full figure out what my pinky's going to do. So we do that. We spent a lot of time doing that, and we can now plan motions for complicated humanoids that were far beyond our ability to do it a few years ago. This was a major effort for us.

My kids and I were watching American Ninja Warrior at the time, so we did all the Ninja Warrior tasks. So there were some algorithmic ideas that were required for that. It was also just a software engineering exercise to build a dynamics engine that provided analytical gradients, exposed all the sparsity in the problem, and wrote custom solvers and things like that to make that work.

It's not just about humanoids. We spent a day after we got Atlas doing those things to show that we could make a quadruped run around using the same exact algorithms. It took literally less than a day to make all these examples work.

There's another level of optimization that's kicking around in here. So the humanoid, in some sense when it's moving around, is a fairly continuous dynamical system. There's punctuations when your foot hits the ground or something like this, so you think of that as sort of a smooth optimization problem.

There's also a discrete optimization problem sitting in there, too, even for walking. So if you think about it, the methods I just talked about-- we're really talking about, OK, I move like this. I would prefer to move something like this, but there's a continuum of solutions I could possibly take.

For walking, there's also this problem of just saying, am I going to move my right foot first or my left foot first? Am I going to step on cinder block one or cinder block two? There really is a discrete problem which gives a combinatorial problem if you have to make long-term decisions on that. And one of the things we've tried to do well is be very explicit about modeling the discrete aspects and the continuous aspects of the problem individually and using the right solvers that could think about both of those together.

So here's an example of how we do interactive footstep planning with the robot. If it's standing in front of some perceived cinder blocks, for instance, the human can quickly label discrete regions just by moving a mouse around. The regions that come out are actually fit by an algorithm. They look small, because they're trying to figure out if the center of the foot was inside that region, the whole foot would fit on that. And they're also thinking about balance constraints and other things like that.

But now we have discrete regions to possibly step in. We have a combinatorial problem and the smooth problem of moving my center of mass and the like, and we have very good new solvers to do that. And seeded inside that, I just want to communicate that there's all these little technical nuggets. We had to find a new way to make really fast approximations of big convex regions of free space.

So we have optimizations that just figured out-- the problem of finding the biggest polygon that fits inside all those obstacles is NP hard. We're not going to solve that. But it turns out finding a pretty good polygon can be done extremely fast now. And the particular way we did it scales to very high dimensions and complicated obstacles to the point where we could do it on raw sensor data, and that was an enabling technology for us.

So our robot now, when it's making plans-- so the one on the left is just walking towards the goal. The one on the right, we removed a cinder block. And normally, a robot would kind of get confused and stuck, because it's just thinking about this local plan, local plan, local plan. It wouldn't be able to stop and go completely in the other direction. But now, since we have this higher level combinatorial planning on top, we can make these big, long-term decision making tasks at interactive rates.

Also, the robot was too big to walk through a door, so we had to walk sideways through a door. And that was sort of a standing challenge. The guy who started the program putting footsteps down by hand said, whatever I do in footstep planning, I will never lay down

footsteps to walk through a door again. That was the challenge.

We did a lot of work on the balancing control for the robot, so it's a force controlled robot using hydraulic actuators everywhere. Again, I won't go into the details, but we thought a lot about the dynamics of the robot. How do you cast that as an efficient optimization that we can solve on the fly?

And we were solving an optimization at a kilohertz to balance the robot. So you put it all together. And as a basic competency, how well does our robot walk around and balance? Here's one of the examples at a normal speed from the challenge. So the robot just puts its footsteps down ahead. The operator is mostly just watching and giving high level directions. I want to go over here, and the robot's doing its own thing.

Now, all the other teams I know about were putting down the footsteps by hand on the obstacles. I don't know if someone else was doing it autonomously. We chose to do it autonomously. We were a little bit faster because of it, but I don't know if it was enabling. But very proud of our walking, even though it's still conservative. This is lousy compared to a human. Yeah?

AUDIENCE: So the obstacles are modeled by the robot's vision, or do you actually preset them?

RUSS TEDRAKE: So we knew they were going to be cinder blocks. We didn't know the orientation or positions of them, so we had a cinder block fitting algorithm that would run on the fly, snap things into place with the cameras-- actually, laser scanner. And then we walk up stairs.

Little things-- if you care about walking, the heels are hanging off the back. There's special algorithms in there to balance on partial foot contact and things like that. And that made the difference. We could go up there efficiently, robustly.

So I would say though, for conservative walking, it really works well. We could plant these things on the fly. And we also had this user interface that if the foot step planner ever did something stupid, the human could just drag a foot around, add a new constraint to the solver. It would continue to solve with a new constraint and adjust its solutions.

We could do more dynamic plans. We could have it run everything like that. We actually never tried this on the robot before the competition, because we were terrified of breaking the robot, and we couldn't accept the downtime. But now that the competition's over, this is exactly what

we're trying.

But the optimizations are slower and didn't always succeed. So in the real scenario, we were putting some more constraints on and doing much more conservative gaits. The balance control I'd say worked extremely well. So the hardest task was this getting out of the car task.

We worked like crazy. We didn't work on it until the end. I thought DARPA was going to scratch it, honestly. But in the last month, it became clear that we had to do it. And then we spent a lot of effort on it.

And we put the car in every possible situation. This was on cinder blocks. It's way high. It has to step down almost beyond its reachability in the leg. This thing was just super solid.

So Andres and Lucas were the main designers of this algorithm. I'd say it's superhuman in this regard, right? A human would not do that, of course, but standing on one foot while someone's jumping on the car like this-- it really works well. In fact, the hardest part of that for the algorithm was the fact that it's trying to find out where the ground is, and the camera's going like this. So that was the reason it had this long pause before it went down.

But there was one time that it didn't work well, and it's hard for me to watch this. But it turns out on the first-- you saw that little kick? This was horrible. I'll tell you exactly what happened, but I think it really exposed the limitation of the state of the art.

So what happened in that particular situation was the robot was almost autonomous in some ways, and we basically tried to have the human have to do almost nothing. And in the end, we got the humans checklist down to about five items, which was probably a mistake, because we screwed up on the checklist. So one of the five items was to-- we have one set of programs that are running when the robot's driving the car. And then all the human had to do was turn off the driving controller and turn on the balancing controller.

But it was exciting and the first day of the competition. And we turned on the balancing controller, forgot to turn off the driving controller. So the ankle was still trying to drive the car. Even that, the controller was robust enough. So I really think there's this fundamental thing that if you're close to your nominal plan, things were very robust.

But what happened is the ankle is still driving the car. I think we could balance with the ankle doing the wrong thing, except the ankle did the wrong thing just enough that the tailbone hit the seat of the car. That was no longer something we could handle, right?

So there was no contact sensor in the butt. That meant the dynamics model was very wrong. The state estimator got very confused. The foot came off the ground, and the state estimator had an assumption that the feet should be on the ground. That's how it knew where it was in the world. And basically, the controller was hosed, right?

And that was the only time we could have done that badly-- the vibrations and everything. I had emails from people of all walks of life telling me what they thought was wrong with the brain of the robot from shaking like that. But that was a bad thing.

So you know I think fundamentally, if we're thinking about plans-- and that's what we know how to do at scale for high dimensional systems is single solutions-- then we're close to the plan. Things are good. When we're far from the plan, we're not very good. And a change in the contact situation-- even if it's in a Cartesian space very close-- change in the contact situation is a big change to the plan.

There's lots of ways to address it. We're doing all of them now. It's all fundamentally about robustness. But ironically, the car was the only time we could have done that badly, right? So every other place, we worked out all these situations where, OK, the robot's walking, and then something bad happens and someone lances you or something.

We had recovery. And then even if it tried to take a step-- even if that failed, it would go into a gentle mode where it would protect its hands, because we were afraid of breaking the hands. It would fall very generally to the ground. All that was good.

We turned it off exactly once in the competition. We turned it off when we were in the car, because we can't take a step to recover when you're in the car and you're the same size of the car. And we didn't even want to protect our hands, because once we got our hand stuck on the steering wheel. So anyways, that was the only thing we could have sort of shaken ourselves silly and fallen.

And what happened? We fell down with our 400 pound robot. We broke the arm-- the right arm. Sadly, all of our practices ever were doing all the tasks right handed, but we got to show off a different form of robustness.

So actually, because we had so much autonomy in the system, we flipped a bit and said, let's use the left arm for everything. Which is more than just map the joint coordinates over here. It

meant you had to walk up to the door on the other side of the door. The implications back up quite a bit.

After having our arm just completely hosed, we were able to go through and do all the rest of the tasks except for the drill, which required two hands. We couldn't do that one. We had to pick up the drill and turn it on. So we ended the day in second place with a different display of robustness.

AUDIENCE: That's still pretty damn good.

RUSS TEDRAKE: We were happy, but not as happy as if we had not fallen. OK. So I think walking around, balancing-- we're pretty good, but there's a limitation. I really do think everybody has that limitation to some extent.

The manipulation capabilities of the robot were pretty limited, just because we didn't need to do it for the challenge. The manipulation requirements were minimal. You had to open doors. Picking up a drill was the most complicated thing.

We actually had a lot of really nice robotic hands to play with, but they all broke when you started really running them through these hard tests. So we ended up with these sort of lobster claw kind of grippers, because they didn't break. And they were robust, and they worked well. But it limited what we could do in manipulation.

Again, the planning worked very well. We could pick up a board and even plan to make sure that the board now didn't intersect with other boards in the world. And we have really good planning capabilities, and those worked at interactive rates-- the kinematic plans.

But the grasping was open loop, so there's really no feedback. So there's current sensing just to not overheat the hands. But basically, you do a lot of thinking to figure out how to get your hand near the board. And then you kind of close your eyes and go like this, and hope it lands in the hand. And most of the time, it does. Every once in awhile, it doesn't.

We experimented with every touch sensor we could get our hands on. That wasn't meant to be a pun. And we tried cameras and everything, but they were all just too fragile and difficult to use for the competition. We're doing a lot of work now doing optimization for grasping, but I'll skip over that for time.

So the other piece was, how does the human come into the perception side of the story? So

one of these tasks was moving debris out from in front of a door. This is what it looked like in the original version of the competition-- the trials. The robot would come up and throw these boards out of the way, and you see the human operators over there with her big console of displays.

This is what the laser in the robot's head sees. We have a spinning laser. We also have stereo vision. But the laser reconstruction of this gives you a mess of points. If you asked a vision algorithm-- some of you are vision experts I'm sure in the room. If you asked a vision algorithm to figure out what's going on in that mess of points, it's an extremely hard problem.

But we have a human in the loop. So the idea is that one or two clicks from a human can turn that from an intractable problem to a pretty simple problem. Just say, there's a two by four here. And then now a local search can do sort of RANSAC type local optimizations to find the best fit to a two by four to that local group of points, and that works well.

And so the robot didn't have to think about the messy point clouds when it's doing its planning. It could think about the simplified geometry from the CAD models. And most of the planning was just on the CAD models.

So this is what it looks like to drive the robot. So you click somewhere saying there's a valve, then the perception algorithm finds a valve. Then the robot starts going. It actually shows you a ghost of what it's about to do. And then if you're happy with it, and if all things are going well, you just watch. But if it looks like it's about to do something stupid, you can come in, stop, interact, change the plans, and let it do its thing.

It's kind of fun to watch the robot view of the world, right? So this is what the robot sees. It throws down its footsteps. It's deciding how to walk up to that valve.

Again, when the right arm was broken, this was one of our practice runs. The right arm was broken, it had a valve. We had to bit flip, and now it had to walk over to the other side of the valve. And there's a lot of things going on. A lot of pieces had to work well together to make all this work.

One of the questions that I'll get before you ask it. If you've written it down, OK, that's fine. Why were the robots so slow? Why were they standing still? A lot of people out there waiting for the human, maybe, but for us it wasn't. It wasn't the planning time. The planning algorithms were super fast.

Most of the time, we were waiting for sensor data. And that meant there was two things. There was waiting for the laser to spin completely around and also just being conservative-- wanting to get that laser data while the robot was stopped. And then there was getting the laser data back to the computer that had the fast planning algorithms in back. So if there was a network blackout, we had to wait a little bit, and that meant we were standing still.

But we've actually done a lot of work in lab to show that we don't have to stand still. This is now the robot walking with its laser blindfolded and using only stereo vision using one of the capabilities that came out of John's lab and others to do stereo fusion. The laser gives very accurate points, but it gives them slowly at a low rate. And you have to wait for it to spin around.

The camera is very dense, very high rate, but very noisy. And John and others have developed these new algorithms that can do real time filtering of that noisy data, and we demonstrated that they were good enough to do walking on. And so we put all the pieces together-- real time footstep planning, real time balancing, real time perception-- and we were able to show we can walk continuously. This will be the future.

So we had to do networking. We optimized network systems. We had to build servers, unit test logistics, politics. It was exhausting.

I think it was overall incredibly good experience-- a huge success. I think the robots can move faster with only small changes mostly on the perception side. The walking was sufficient. We can definitely do better. The manipulation was very basic. I think we need to do better there, but we didn't have to for those tasks. The robustness dominated everything.

So I'll just end and take questions, but I'll show this sort of fun, again, robot view. This is the robot's God's eye view of the world while it's doing all these tasks. You can sort of see what the robot labels with the geometry and what it's leading its points. And it's just kind of fun to have on in the background, and then I'll take any questions.

[APPLAUSE]