EPISODE 87

[INTRODUCTION]

[0:00:10.8] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

This week on the podcast, we're featuring a series of conversations from the AWS re:Iinvent Conference in Las Vegas. I had a great time at this event, getting caught up on the new machine learning and AI products and services announced by AWS and its partners.

If you missed the news coming out of re:Invent and want to know more about what one of the biggest AI platform providers is up to, make sure you check out Monday's show, TWiML Talk #83; a round table discussion I held with Dave McCrory and Lawrence Chung. We cover all of AWS's most important news, including the new SageMaker, DeepLends, recognition video, transcription, Alexa for business, Greengrass, ML inference and more.

This week, we're also running a special listener appreciation contest to celebrating hitting one million listens here on the podcast, and to thank you all for being so awesome. Tweet to us using the #twiml1mil to enter. Every entry gets a fly twiml1mil sticker, plus a chance to win a limited-run t-shirt commemorating the occasion.

We'll be digging into the magic TWiML swag bag in giving away some other mystery prices as well, so you definitely don't want to miss this. If you're not on Twitter, or you want more ways to enter visit twimlai.com/twiml1mil for the full rundown.

Before we dive in, I like to thank our good friends over at Intel Nervana for their sponsorship of this podcast and our re:Invent series. One of the big announcements at re:Invent this year was the release of Amazon DeepLens, a fully programmable deep learning enabled wireless video camera, designed to help developers learn and experiment with AI both in the cloud and at the edge.

DeepLens is powered by an Intel Atom x5 processor, which delivers up to 100 gigaflops of processing power to onboard applications. To learn more about DeepLens and the other interesting things Intel has been up to in the AI space, checkout intelnervana.com.

Today, we're joined by Aaron Ames, Professor of Mechanical and Civil Engineering at Caltech. Aaron joined me before his talk at the re:Invent Deep Learning Summit on iRobot Computer Vision and Autonomous Robotics.

While Aaron considers himself a hardware guy, we got into a great discussion centered around the intersection of robotics and machine learning. We cover a range of topics, including Boston Dynamics's back-flipping robot and how a system like that actually works; as well as the various humanoid robots his own lab has created, and more broadly his views on the role of end-to-end deep learning in robotics. I had a blast with this interview and I think you will too.

Now, on to the show.

[INTERVIEW]

[0:03:24.3] SC: All right, everyone. I am here in Las Vegas at Amazon re:Invent and I have a pleasure of being seated with Aaron Ames. Aron is a Professor of Mechanical and Civil Engineering at Caltech. He is going to be speaking here at re:Invent tomorrow as part of their deep learning summit. As you can tell from his department affiliation, he's not a deep learning guy. He is a robotics guy, a hardware, a self-professed hardware guy. Aaron, welcome to This Week in Machine Learning and AI.

[0:03:51.7] AA: Pleasure to be here. Thanks for having me on the interview. Yeah. I come from actually algorithms and mathematics is sort of my background and putting it on hardware. It's funny because what I do, I'm speaking of the Deep Learning Summit, I think partially to give this perspective of how learning and AI algorithms will play out on hardware platforms and what that connection will be.

I mean, historically in my research and in research like Boston Dynamics and all these cool things like the backflip video recently appeared –

[0:04:22.8] SC: That was incredible.

[0:04:23.4] AA: It was incredible, but guess how much learning is on those platforms.

[0:04:27.7] SC: Yeah, I imagine not much.

[0:04:28.8] AA: None. Zero. I mean, the core in doing these things is to take the dynamics of the system. There is physics driving it and then you develop algorithms using something called control theory to determine how to move the robot, how to make the actuators move in specific patterns so that you get these dynamic behaviors.

It's heavily tied with the physics. You have the physics, you make decisions based on the physics and it's all very deterministic and pre-programmed. AI on the other hand is a totally different animal. You start with data. It's all data-driven, data-centric. You take examples of things that work well, you label images and then you plug it into a deep neural network. There is other variants of learning as well that are a little more mathematical, and then the backend is that it learns or identifies these patterns.

That's really exciting, because its computer is doing things that we don't always expect they'll do. They can deal with highly unstructured and data-driven approaches. But it runs contrary to this whole hardware and theoretic approach that's often taken in robotics, because we need to know everything about the robot. It has to be very pre-programmed and pre-planned in a specific way.

The question of the deep learning, some that I'm addressing in my talk is what would this integration look like, or a view towards this integration of learning with hardware and robotics systems. Something that has yet to actually be done in a good way, because there is such different worlds. What's missing from the learning community? What's missing from the robotics community and how could they benefit each other?

[0:06:01.8] SC: Right. Let's dig into that. But before we do, I want to make sure that the audience gets to know you a little bit.

[0:06:06.6] AA: Absolutely.

[0:06:08.3] SC: How did you get interested in algorithms, math, hardware and the way you've put them altogether?

[0:06:13.6] AA: Right. Science fiction in short. I was driven by science fiction. When I was an undergrad, all I did was read Sci-Fi all the time.

[0:06:20.9] SC: Any favorites?

[0:06:22.8] AA: Asimov is the go-to classic, right? There is some other great ones, you know, Heinlein. The classic authors I think had a really unique perspective. Actually coming from – the authors has started in the 50s, 60s, there is such imagination to where we'd be, unconstrained by the problems that would later be confronted in robotics and in all these other things that I think it painted a picture of the world that was really enticing, of how robots can really work amongst us.

That's driven me for a long time is this internal fascination. I can't really explain it on my borderline, or maybe not even borderline obsessed with it. How do we make robots move like us and do things like us? That's driven me for a really long time. I wanted to delve into that. My background is actually highly theoretic. I didn't touch hardware until actually after my PhD. I studied walking, but from a theoretic perspective. I really wanted to understand the mathematical underpinnings of locomotion.

[0:07:20.7] SC: Did you start by looking at that from a robotics or hardware perspective, or like human walking?

[0:07:26.3] AA: I started it from a robotics perspective, but not so much hardware as the mathematics underlying dynamics and how do we understand that? How do we even model or formulate a mathematical model of walking and running and doing dynamic things on robots. I delved into that. I proved a lot of theorems on what this might look like. Well how do we quantify what this is? How do we characterize it? The basic science of dynamic robotic movement.

[0:07:58.2] SC: How far have we come in that? Do we have a strong analytical foundation in locomotion, or is it – do we bump up against an edge analytically and have to apply computation to?

[0:08:08.7] AA: That's an interesting question. Actually we have a really strong analytic foundation now for locomotion. It's been developed over the last, I mean I guess 20 years. Starting a little before I was a grad student, it was in its infancy and I started working with it.

Other people, great people at other universities have developed these frameworks; Jessy Grizzle at University of Michigan is one example. There is lots of people, but we've come a long way in our mathematical understanding of locomotion, what the models are, how do we quantify that behavior.

There is a lot of papers on this too. We can understand it mathematically, but what's interesting is we had those mathematical understanding a while ago now, but computationally realizing that math on hardware was a huge problem. That's where the blockade came is we can write a theorem and we could say, "If this thing exists, then we have walking. But how do you find the thing that exist?" That's a computational question. In the end, every mathematical thing you do has to be translated to algorithms on the robot.

How do you do that? Well, it turns out that recently there has been a huge surge in this computation area, huge breakthroughs, mainly due to the computation breakthroughs that have happened; the prevalence of cheap and vast computation.

It turns out there is a lot of analogies between making a robot move dynamically and learn. What I mean is in essence, it's a large optimization. That's what the math boils down to. How do you solve large-scale optimization problems efficiently? There has been some great results recently, some of which have come out of my lab, some of other labs, a bunch of people collaborating, where we can now solve these orders of magnitude faster than we could 10 years ago. I mean, it used to be take a day plus to generate one walking behavior for a humanoid robot over a day of computation, if it converged.

[0:10:03.7] SC: Is a walking behavior, what does that mean specifically?

[0:10:08.7] AA: The way you can think about a walking behavior is a periodic motion. That stable.

[0:10:14.7] SC: Okay. Given a set of parameters that define the hardware, or what the –

[0:10:19.9] AA: That's right. That's right.

[0:10:21.9] SC: - the joints, the angles, the lengths of the leg.

[0:10:23.6] AA: The masses, the inertias, all that stuff. What happens is –

[0:10:26.2] SC: At even lower level then.

[0:10:27.4] AA: Yeah. You pull all those things together to create a mathematical model, right? You get a differential equation if you'd like the technical term. It's actually a hybrid system too. Meaning, there is continuous dynamics. Think about just a bouncing ball, as almost the simplest example of a walking gate.

What I mean is a periodic motion. It falls through the air until it hits the ground, and then there is a discreet impact that pops it back up. Now imagine you have a little actuation with that bot, like a little spring and you can, you know. Then the goal of locomotion is create a stable periodic motion, so the ball bounces at the same height all the time.

That's a very low dimensional example. Now take a humanoid robot. You take all the physics that go into it, right? You have something like, you know, let's say 25 degrees of freedom. What that means is 25 joints that you can actuate, or other joints that you can actuate. The point is 25 things that can move.

Then you have to take that and you get some mathematical representation of it as a hybrid system, because there is this – the leg swinging forward, that's a continuous dynamic, just like when the ball is falling and the foot strikes the ground and you get these impacts. Then you have to create a periodic motion that coordinates all of those 25 degrees of freedom together in a synchronous way.

[0:11:43.1] SC: Did these 25 degrees of freedom translate into some series of hundreds of differential equations that you're trying to say?

[0:11:49.5] AA: Yeah. You actually get two times the number of degrees of freedom, because you usually were dealing with second order systems. So you end up with let's say 50 equations, 50 ordinary differential questions. Technically, an ordinary differential equation that's 50 dimension. You're dealing on some 50 dimensional space of evolution.

It's a very high-dimensional space, and that's for simple ones. It gets even higher. I mean, take a full humanoid with hands and everything. You could be dealing with a 100-dimensional system plus. This is all very nonlinear. More importantly, it's because you're generating these periodic motions, you have to really utilize these nonlinear dynamics. The nonlinear dynamics of the system to generate these behaviors.

[0:12:30.1] SC: What does that mean specifically?

[0:12:32.1] AA: What that means is you can't just – to provide an example, it's funny we get a lot of comments on YouTube for our videos and these are – I actually read them. Sometimes I enjoy reading them, although I never respond. I think that's the key is never respond in comment. But I like reading them.

A lot of questions revolve around, "Well, why not just take a human walking trajectory, right? Just record a human walking and pop it on the robot." Well, it's because the physics are different from the human and the robot. What I mean by the nonlinear dynamics is there is only certain trajectories that work for that system, that make sense, that are consistent with its dynamics. Those are the ones you have to find. You can't just put a human trajectory on. You'd have to modify it so it would be consistent with the dynamics of –

[0:13:17.1] SC: The robot would have to basically have human physical properties.

[0:13:19.7] AA: Yeah. Human actuation and everything else. It have to be perfectly human in some way and that's not going to happen and it shouldn't. It's like when we have planes that fly, they don't flap their wings. You'll have to exploit the – you want to be inspired by flight. You want to create life, but you want to do it in a way that's consistent with what we can build.

You take all these dynamics, just not to get too far in the weeds, and then you have to create these periodic motions, which again results in these optimization problems you have to solve. You could imagine what this might be in the sense of – even with the bouncing ball, you want to create a periodic trajectory. So start at some point and end at some point. Those are some constraints on the system, right? Another constraint has to satisfy the dynamics of the ball falling.

You put all those together and you crunch it into this big optimization. Like I mentioned, we've gone from maybe a day plus to solve this maybe to down to a couple minutes, even faster, sub-second, meaning almost real-time. Which means we can generate a gate and really fast. By a gate, I mean periodic motion. Then you could start putting all those together and create advanced behaviors. That's the paradigm for how you create walking gates, or any kind of behavior.

There is the backflip we mentioned earlier. How would you do that? Well, you can actually take the dynamics of that system, you can set that up as an optimization problem, where you go through this motion of flipping. Then you can crunch it into an optimization problem and generate those motions, and then pop that on the robot.

That sounds easy. It is a lot of difficulty in doing that. That's a non-trivial adventure to actually implement that, but that's the general trend. There is lots of ways that people do this. I mean, I don't want to go through all the background, but there is lots of ways people can generate these periodic motions they can have; reduced dimensional representations of the system that make it a little faster. There is a lot of tricks in all these stuff.

In the end, what you're fundamentally doing, the point of this discussion is to say this is the way the robotics community by in large approaches, the problem of generating behaviors on robots. They start with the physics. They set up some sort of optimization problem, computation problem which generates the – it gives you this period motion, which you then put on the rope.

[0:15:27.4] SC: When we see a video like the – people would be familiar with the Boston Dynamics and the backflip one that came out recently. In fact, you showed me a really interesting one on your YouTube of a robot called DURUS, just prior to the interview. When we see videos like that and let's take the back-flipping one, because it was –

[0:15:45.2] AA: Yeah. I think everyone has seen it. That's fine. I have to say it was very impressive. I mean, Boston Dynamics is always raising the bar for us academics. Just when you think, you mentioned DURUS. If you look at DURUS walking on my YouTube page and then you compare it with some of the walking that Boston Dynamics has, they're in the ballpark, right? They are. I mean, which is something I'm very proud of.

They had that stuff a couple years ago and now we have the mathematics understand it, and then they do the backflip. I'm like, "All right, so now we got to do the backflip, or some variance of that matches it." They raise the bar and they push us, which I think is great. It's a very impressive behavior, so there is no doubt about it.

[0:16:20.5] SC: When we look at that, are we seeing a robot basically performing a script and it can do just what we see starting from where it started, going through every point in space that it saw, or is there –

[0:16:33.9] AA: That's exactly right.

[0:16:33.9] SC: Is it some parametric thing, or there is some variability or –

[0:16:37.3] AA: It's a script.

[0:16:37.9] SC: It's a script.

[0:16:38.6] AA: You're exactly right. Here comes the crux of the problem and the exciting opportunities. Let's now talk about the backflip with Boston Dynamics. When you look at that video, I mean everyone is like, "Oh, Skynet is coming. The world of robots are taking over. Oh, my God." Again, we mentioned earlier, we don't know exactly what's on the robot. Just to be clear, they don't publicly release anything that's on there.

Although I know Boston Dynamics well enough to make a very educated guess. The guess is based on their past stuff too is exactly what you said. It's a pre-planned behavior. This robot has no knowledge of its environment in the sense that it's not observing where those blocks are and in real-time adjusting its behavior and learning how to do this behavior. They put those obstacles in the memory of the computer. They pre-planned those behaviors. They do a bunch of experiments until they get the right behavior.

They take a bunch of videos. What I loved about that backflip video is after the backflip, they showed failure cases. Which is important to show, because anytime you see a video of something working, there is a 1,000 or 10,000 cases where it didn't work at all. Finally you get everything tuned in just right, as that hardware is not quite as clean as the map I talked about. Then it goes. That's a pre-planned behavior that a robot has no awareness of what it's doing in a broader sense.

[0:17:56.7] SC: Even beyond the awareness and learning and ability to – the robot's ability to adjust, I think when – It's easy to envision something like that where – I guess, I'm thinking of it from a computer programmer's perspective. There is no function that says backflip start from X SQLs whatever. It is like a vector of points that it is just following.

[0:18:23.2] AA: Yeah. Effectively yeah. I mean, you can represent these behaviors as modules if you'd like, like backflip itself. Think about dynamic –

[0:18:31.7] SC: Backflip, like with how many – how much freedom? Backflip and you can give it a height and it would on the fly –

[0:18:38.5] AA: That's a good question. Exactly right.

[0:18:39.9] SC: I mean, we're not – That's far before we get to any conversation about awareness and learning.

[0:18:45.6] AA: You're absolutely right. I think this is a very astute question and comment is right now, I mean I don't know their exact capabilities, but typically it's backflip from a height of this high, maybe with some small variations. But if you change the terrain it was on, or change the box, or change any of those parameters, it wouldn't do it.

In fact, if you look at that video, you look at what it's landing on, it's somewhat a pad that looks like it has – it's very carefully constructed. It's not just a random floor. There was something special about what they were landing on, partially to absorb the shock of impact, I'm sure.

The point is as you said, yeah. Backflip is a canonical unit, but that is very constrained in the environments it can work on. You're right. This is something that's really important for people to understand I think from the learning perspective, is we're not like one step away from self-aware machines in this context. I mean, we can do pre-program behaviors in environments we completely understand and have characterized within a small window of perturbation.

We're getting really good at that. We couldn't do that 10 years ago, 20 years ago, we couldn't even do that. A robot couldn't do a backflip. Early it's not a humanoid, although Raibert has some great stuff from the 80s where he had two-legged, sort of pogo stick type robots that could do backflips back then. You should go check out some of the stuff from when he was a professor at MIT in his own scenes from the 80s.

[0:20:06.9] SC: What's the name again?

[0:20:08.1] AA: Marc Raibert.

[0:20:08.5] SC: Marc Raibert?

[0:20:09.1] AA: Yeah. He's the head of Boston Dynamics. It's his brain child, and he went and founded it from MIT. He actually started out at JPL, which is part of Caltech and then Carnegie Mellon, then MIT, then started Boston Dynamics and has been doing that forever.

If you look at this stuff in the 80s, it has the same characteristic. If you watched those videos, you see how the backflip came to be on Atlas, right? You really see the trend. But again, it's all these very structured thing. That leads to what do we do about unstructured environments? This is actually the core of what I'm going to talk about tomorrow is I want to set the stage with just like we had this conversation, I want to explain what it takes to make a robot walk.

Because I think when you understand that, or make a robot backflip, you realize how much machinery is there, and how first, you're not going to learn how to do that really. I mean, it's too high to mention of a problem for learning. You're not just going to plug in the –

[0:21:04.0] SC: Actuators and the –

[0:21:05.1] AA: Whatever into some deep neural network and expect the outcome to be a backflip. There is some structure in the system that has to be exploited. There is a place for control and dynamics, but there is a place for learning too. I think that's the thing is to understand the right context for it. Learning will not take over the world and learning will not solve all problems, but learning can handle unknown and unforeseen things. That's the role it can play in the context of like a backflip.

[0:21:33.0] SC: How do we go about getting to that? I mean, it seems like – it strikes me that there are lots of **[0:21:40.6]** ways of attacking that problem. I'm thinking that – the thought that comes to mind is one approach might be defining levels of abstraction, or primitives, or something like that and trying to figure those out and have the learning, the intelligence select those on a fly.

[0:21:59.3] AA: I think you're actually [inaudible 0:21:59.3]. I'm impressed. You need to come join the robotics walking community, because you nailed it. The current perspective, I even have some papers on this for about five years ago where we call it motion primitives and transitions. Basically what you do is you create primitives for all these different behaviors, because you're going to have to create the behaviors. Now that computation has gotten better, you could imagine – thin about a graph, where every node of the graph – every point is a behavior. Then you have transitions between those behaviors, which are admissible.

So you might have backflip followed walking, going up and down stairs. You might have a bunch of different primitives for different backflips for different heights and different terrain types, and you can build up this entire compendium. Then you could supervise how you pick an individual behavior with a learning output.

This has started to be done at – I mentioned Jessy Grizzle earlier. I work closely with him and have for a really long time, and he started to play with some of these ideas, putting learning on top of that sort of motion primitive and transition framework, where you decide what gate to do at any given time based on what the environment is doing.

A learning algorithm can do that really well, and that's a great place for learning. That's a great way of thinking about learning in the sense that you've taken the dynamics into account, you've taken the mathematical representation of dynamic motions to their extreme. You've utilized them all the way, and then you let learning do what learning does best, which is based on input data. The environment decide what to do on an output, but at a very high-level. If you look at the way the human brain works, this is very analogous to how the human brain and body work.

[0:23:41.5] SC: I'm not thinking about moving my legs and my [inaudible 0:23:43.4].

[0:23:43.5] AA: Exactly. Yeah, so we have motion primitive and we transition between them. More importantly, the architecture of the human body. Now I'm not an expert here, but I've certainly read a bit about it. It fits with this paradigm. Some people say, "We should just learn everything, because our brain learns everything." That's not actually true. Our brain is responsible for some of our motions, but in our spinal cord, we have a separate brain.

I mean, in essence. We have patterns that are generated. Those are your locomotion patterns, also your primitives. You can walk essentially with very little cognitive load. I mean, think about walking and texting on your phone. You don't have to think about it, right? That's the motion primitive acting. What happens when you get to a stair? You have to look up from your phone, you have a cognitive load. For a second, you have to think, "What am I going to do next?" You decide, you go in, you're back to you fold.

Think about anytime you could be doing something on your phone, that's where dynamics control and all those classic approaches would be used. Anytime you have to look up and see something, that's where machine learning will play a role. The real point here is it's not one area or the other. We really need to understand the intersection of these two domains. That's really the challenge problem of the next decade it might be, because there is a lot of people that do learning, a lot of people that do robotics. There is the beginning of connecting this up a little bit. But we need to make a concerted effort to really understand this connection point, because I think that's the key.

[0:25:06.2] SC: The using a learning system to plan a goal achievement across some set of primitives is one approach. We've already talked about the – we've already thrown out the window the idea of learning the motion primitives from the ground up, but is there a role in learning in making them more robust?

[0:25:27.4] AA: Yes, absolutely. Right. That's the second place where learning comes in. You think we'd have a script?

[0:25:32.8] SC: We had a script for this interview, because you're feeding right into my –

[0:25:37.0] AA: My talk and research. The second place, yes. Is we're learning place – we'll play a big role, is not at the level of planning, but at the level of unknown, unforeseen environments and influences.

The simplest example is walking on different terrain. When you're walking on flat hard ground, we have a perfect model that – remember that everything we talked about with generating models was built on the premise of having the model. Now if you walk on sand or dirt, it turns out there are no models of this. There is no simple models of these.

I mean, there is people they make their entire careers about modeling and simulating granular media, deforming and moving around. Sand and dirt crunching down, it's a really, really hard problem. There won't be some –

[0:26:20.3] SC: I mean, computationally it's difficult to just render it as a picture, let alone trying to figure out its physics and contact with other things.

[0:26:26.2] AA: Exactly. You could imagine days to generate physics simulations of sand moving. Now if takes a day to generate a physics simulation of a robot putting its foot in sand, you're probably not going to be real-time using those things. How do we bring it together?

We have some initial work with some colleagues I have at Georgia Tech on this idea, where we learn how to hop in granular terrain. We don't use neural nets. We use something called calcium processes, which are another way of learning, but it's not they're on that way. It's a variant that basically deals with some initial gas on the model and then you update that model as new data comes in.

We were able to not know what the train model was, but have a guess based on some physics. Then every time the robot would hop in the train, we'd take that data in and update the model of what the terrain forces look like. We iterated that through the optimization problem, so we'd update that every time we had a new hop, we'd take this new information, we generate a new model of the physics interaction with the world and then we'd run the optimization with that new model.

[0:27:30.6] SC: Now are we talking about something that's done in simulation as part of the –

[0:27:34.1] AA: We did this on hardware as well.

[0:27:34.9] SC: - planning process, or –

[0:27:36.5] AA: We did this on a hardware. We verified in simulation, but really this would have to be on the hardware, because you need the data. You need to sense what's happening with the environment.

[0:27:49.1] SC: Right, because you don't have the models to put in a simulator.

[0:27:52.2] AA: This is now a data-driven modification. You can mine the data with learning that model of the environment with the optimization framework that I discussed earlier in a feedback. We did that and we were able to actually – we wanted to hop – I mean, this is a very simple robot. This is not a walking robot, but it's like think about the bouncing ball again where we're going back to the basics. We wanted to make this thing hop at a specific height. We'd first try it without having a model of the granular train and it wouldn't do it.

[0:28:22.2] SC: I'm getting hung-up on the form factor here. Is this like a standalone robot that is – what does this thing look like?

[0:28:31.0] AA: All right. That's a good question.

[0:28:33.2] SC: Is it suspended in – is fixed in some way?

[0:28:37.1] AA: Yeah, understood. This is a very simple test bed meant to generate physics sub-terrain interaction. It was actually developed by a colleague of mine, Dan Goldman at Georgia Tech who is a physicist, and this was work with Patricio Velo as well at Georgia Tech who does machine learning stuff.

It's a very simple thing. It's a motor with a spring between basically the motor and the world. Basically, it can move a mass up and down to make this thing move on top of the spring. Then there is the spring and then there is like a foot between the granular terrain. This thing is only moving in one dimension.

[0:29:13.9] SC: It's a one-dimensional hopper.

[0:29:17.8] AA: Yeah. It's a one-dimensional hopper. That's right. What it does is it sits in a bed of poppy seeds. These poppy seeds are actually a really good model of different sand and dirt, but they don't get caught up in the actuator, because they're big enough that they don't get into all those things.

[0:29:30.7] SC: Details.

[0:29:31.4] AA: Exactly. What actually you do is there is these bed of poppy seeds, but you don't want to have it be changing every time you hop on it, because you can compact them, so it would be more hard terrain. It actually aerates the bed between every hop, so you move the poppy seeds, let them settle, hop, let them settle, hop. You do a successive experiments where you have the same kind of initial condition in the poppy seeds, just to make sure you have a consistent model you're learning.

That's the setup. It's a very isolated little box. Then you can use vision or anything else to identify what the forces are between the robot and the terrain. Then we ran this experiment. Again, this was a proof of concept that's probably one of the first examples of putting all these pieces together, and it shows you and this is an important point. It shows you where we're at in unifying learning and control and dynamics if we have to go back to a one-day hopper.

We published a paper on it, because it's new and interesting. As opposed to a humanoid robot, which is – You can imagine that we're just beginning this process and there is some other people working on this domain as well.

[0:30:37.5] SC: Skynet isn't as scary if you can just walk in the beach and just –

[0:30:40.9] AA: Exactly, yeah. That's why I advise to anybody, if a robot is chasing you trying to kill you, just run to a beach and you're going to be fine. Or go in some ice or something and it will fall over. You'll be safe.

[0:30:55.7] SC: That's right. We got those two pieces of learning, like what's next?

[0:31:02.0] AA: I think you did a good job exactly parsing the two forefronts that we need to push in terms of machine learning on robotics systems. This is the key point is there is a lot of work right now on learning as a standalone entity. Learn everything.

[0:31:22.1] SC: With this, I can interrupt as a point of reference for folks that are listening that want to hear a little bit more about that perspective, a good place to start would be the interview I did with Pieter Abbeel.

[0:31:33.2] AA: Yes. I know Pieter well. He does fantastic work. I really respect the work he does. But the idea there is to learn everything.

[0:31:45.3] SC: Right. We talked about that recently.

[0:31:46.1] AA: Yeah. I am very strong and adamant that that is not the right answer. Now people would disagree with me and that's okay. It's okay to disagree in academia. It shows that the problems aren't solved. Physical systems, robot arms, even he does manipulation stuff, have this wonderful structure we can exploit.

They might be high-dimensional, but they live on manifolds, which are low-dimensional surfaces in this high-dimensional space. Let's use physics and control to push the system to this low-dimensional space and then learn on that space. It will work orders of magnitude better, I promise. Well, I promise in that and I think I'm right with my opinion, which I may be wrong and that would be great.

At the point when a robot does a backflip with no knowledge of his physics or dynamics, a humanoid robot, then I'll be like, "Okay, I am ready to listen to the pure learning approach." But in the robotics community, the proof is in the pudding as they sometimes say. Or the proof is on the robot. The reality is we can do so much more with zero learning, model-based approaches on physical hardware than we can with learning.

Now that being said, I'm not trying to advocate that that's the end of the story. Like I've said to this whole interview, there is limitations to these. That's where learning will play a huge role. I think the forefront is don't learn everything. Don't fall into this trap of a big shiny black box that you put in what you want and it spits out the right answer.

If only because from a scientific perspective I find that very unsatisfying too, that's a separate point, but my opinion is don't fall into that paradigm, because you're restricting, or you're limiting yourself in your world. You instead unify; unify, unify, unify. That's my argument on the forefront of learning is –

[0:33:32.7] SC: What else does unify mean for you?

[0:33:33.7] AA: I mean, bring the learning, physics, dynamics, computation, control, mechanical design, actuators, all those pieces. What's beautiful about robots is they are a microcosm of the universe in some way. They're completely self-consistent systems that you have complete control and you create. You have computation, you have actuators, you have all these wonderful things.

Use that and understand those different pieces at a deep level, and then you'll really understand how to put them together. That's what I mean unify. Unify all computation, control, design and learning. Understand where they all fit together relative. Use the strengths of each to exploit them to their maximum, and that's where I think the really fun stuff is going to come in the next couple years in my opinion. I could be dead wrong in a year from now. If I am, I'm happy to admit it. But I think that's really where the future is in terms of learning in robotics.

[0:34:31.4] SC: Do you have any predictions in terms of what that really fun stuff is going to look like for us.

[0:34:35.7] AA: The forefront is the following in my opinion; it's getting robots out of the lab into the world. There's been a couple examples of that. We've taken some of our robots out of the lab in limited context. Boston Dynamics still has some of the best videos, where it's walking around in snow and stuff like that, right? That was purely reactive by the –

[0:34:52.0] SC: Primarily the four-legged ones, or the bipeds too?

[0:34:55.0] AA: They had a biped outside in the snow in one video. It was about a year or two years ago now. I can't remember, that's purely reactive. There is no learning there. It was walking in snow and on even terrain. Only through the robustness of the algorithms that are on there, right?

I think that's really the forefront is get out of structured environments, get out of the lab. Get on dynamic systems. I'm very wary of pure manipulation tasks, as they're deceptive in their simplicity. When a robot can't fall over, you can always correct and there is a robustness there that when you go to humanoid robots and dynamic robots, you don't have any more.

Get a dynamic robotic system, whatever that is; a four-legged robot, a two-legged robot, a hopping robot, whatever it happens to be out into the real world and make it do cool stuff. That's to me the way of really proving that you understand what's going on. Because that will take all these things we mentioned, especially in an autonomous context.

This is the second point. We actually just started the Center for Autonomy at Caltech called CAST. It's really aimed at doing these things, is how do we get robots into the wild and not prescript them all the way? That's really what I mean by getting in the wild is tell a robot go from A to B outside; a walking robot, or a humanoid robot. By A to B, it might be over a beach. It might be through some ice and snow. How do we do that? What's that? We actually frame these questions in the context of moonshot's forecast.

[0:36:30.9] SC: Just to give us a sense of –

[0:36:33.2] AA: Of how hard they are. By moonshot, really a lot of these are moonshots, meaning we could do it if we had massive amount of resources and people concerted. One example of a moonshot is have a robot walk the Pacific Crest Trail. This is a trail that goes from Mexico to Canada, and have it do it autonomously. What would it take to do that?

That exactly would require all the pieces we discussed today, plus many more that we don't even know, we don't yet. But that's the kind of thing we're thinking about is pushing these boundaries of what we can do with robotic systems, and in autonomous ways. Bringing autonomy in it and bringing them and understanding how that fits with both the mathematical representation of behaviors and learning, and where those each can play a role.

To me, that's the direction of push. It's challenging, but fun, but it's time for the real world is where we're at. For a long time, we couldn't even get robots to do stuff in our labs that was all that interesting. Now we're at the point where we can do some cool stuff in our lab. Leave the lab.

[0:37:33.4] SC: What I'm struggling with a little bit and trying to bridge our early conversation about these very scripted rigid tasks and even some of the stuff that the Boston Dynamics is walking in snow. It's like we're incorporating – we have to be incorporating in sensors. Is even the most primitive stuff, I'm thinking of that as just the actuators and not sensors, or is that not the way to think about it?

[0:37:57.8] AA: I mean, no. They all have sensors. The question is what are the sensors doing and what information are they taking in?

[0:38:03.5] SC: Yeah. How are they fit into whatever algorithms?

[0:38:05.5] AA: How are they fit into an end. Sensors are part of all of these. The question is what sensors and what are they sensing? From the backflip on, again I can't speak. I know they're hardware all the way in. But roughly speaking, you have encoders that look at the angles of all the joints, you have an IMU, Inertia Measurement Unit that tells you the global orientation of the robot, right?

[0:38:25.9] SC: Accelerometers and that kind of stuff.

[0:38:27.3] AA: Exactly. Yeah. Then there is sometimes some sort of sensing of the environment; has your foot touched down, and that's pretty essential one. When we walk with DURUS, I can tell you that. What we need are those three main components. We need to know when the foot is on the ground, so it sense the impact with the ground. We need to know what the angles are in all the joints, and we need to know – again, an IMU, an accelerometer. We need to know the global orientation of the robot relative to the above.

Those three pieces of information are all you really need to have a robot do a dynamic thing in a strained context. That means you know the environment, right? You don't need computer vision if you know how high the blocks are and where they're located relative to the robot, and you set the robot up in the same spot every time.

Obviously, when you start to do more unstructured things, you're going to have to bring in other sensors, cameras of course. That's one Pietro Perona is talking – another professor at Caltech is talking with me at the Machine Learning Summit and he does computer vision. We've started to say, "How can we integrate these two pieces together?"

We actually have a robot in our lab called Cassie, where he did his algorithms to parse where they can determine the pose of people. We want to use those pose information to have the robot do something. We want the vision of the robot, what the robot see that feed into its behaviors. We're just starting this track, but that gives you an idea. In this case, we need a vision system. We might need four sensors if you're going to observe the environment for the hopping behaviors we discussed. You're going to need to have a really nice notion of what the forces are on the system.

The more you want to do, the more sensors you need. This is again, going back in our conversation this is very consistent with the perspective of the human body. If you want to walk on flat ground with no obstacles, you can actually – you need very little sensors, right? You know when your foot strikes the ground and the rest is pretty – If you've never been on ice before and you take somebody on ice –

[0:40:19.9] SC: Look at the way they act.

[0:40:20.3] AA: - and doing a lot of sensing. They are doing a lot of sensing, they're doing a lot of computation, because they're learning. But then watch a couple of minutes on ice and people settle in and they're clearly – they've whittled down. They've taken all their sensor information and decided which sensor information is important and they're using that.

That's part of the problem too is how do you take all those information and would your environment then would allow the stuff that's actually relevant to what you're trying to do in the individual motion permit.

[0:40:42.9] SC: I think that's a great example, a great way of articulating it that really gets at what I was struggling with is the – when you think about the human on the ice, there is that bit of learning. If you take that to – I'm trying to reconcile that with Boston –

[0:41:00.6] AA: You have a Boston Dynamics.

[0:41:02.2] SC: The Boston Dynamics walking in the snow, you were saying that that robot is not learning. What is it doing with that sensor data that's allowing it to be more robust than what we talked about earlier?

[0:41:13.0] AA: Okay. You noticed it was walking in snow and small terrain differences, but not ice. The deciding factor here is friction. As long as it has sufficient friction when the foot touches down, you can do the same behavior you do on firm ground on non-firm ground as long as it's reasonably close, as long as the foot doesn't slip, or not too much. If it slips, you can catch yourself.

[0:41:38.8] SC: I may be confusing videos in my mind, but there is one where like it's running uphill.

[0:41:42.7] AA: It slips on ice. There is one word, even slips on icing and catches itself. But it slips on a small patch of ice and catches itself and then it's off the ice. This is I think what your question is and where we can separate is that there is a difference between reactive behavior that's robust enough to handle terrain differences with learning a new behavior itself.

That's the difference. If you're walking along and you slip on a little puddle – watch a person when they slip, they go [inaudible 0:42:14.7] and they catch themselves and then they keep walking. That's not a learned behavior, that's a reactive behavior. Or you missed a step. That's a great one when people don't know a step is coming and there is a step, and then you see them fall and you know this feeling too, right? You're already falling and catching yourself when you realize, "I just fell down a step."

Your body is doing stuff before you even realize what's happened. That's a great example. Next time you fall, please I mean –

[0:42:39.8] SC: I will remember this conversation.

[0:42:40.1] AA: Hopefully you don't fall on purpose, but after you catch yourself from falling think back and realize, you did all that stuff before you even thought about what you were doing. That's reactive. That's what Boston Dynamics does is their controllers are so robust and they're very impressively robust that they can react to all these different things and be robust to it.

[0:42:56.3] SC: It's not fair to say that it's simply actuating these motors to a series of pre-planned points, and that's how it's doing, walking or doing flips or something like that. It's more robust as more hierarchical.

[0:43:13.0] AA: There is more, yeah. So my description – same with our robots, my description of moving the robot through a series of pre-planned points, or trajectories we call it is a simplistic representation of what actually goes on the hardware. That's part of it. That's actually mathematically. That's what we call the feed forward term, or the nominal behavior.

Assuming everything is perfect, that's what it will do. But we do. We need to stabilize. We talked about stable periodic motions. We need to stabilize, we need to robustify that. For that, you add things that bring you back to that orbit if needed, and that's where this robustness and reactive behavior comes from.

There is a hierarchy. For Boston Dynamics, that hierarchy is based on foot placement typically – assuming based on the Raibert papers from the 80s and all these is that they – based on the orientation of the robot at a high-level, it will place its foot in different locations. It still has the nominal – listen to a Boston Dynamics video and you'll notice it's very time-based. That's because that's the nominal trajectories. You hear the consistent [inaudible 0:44:12.0].

[0:44:12.3] SC: I'm hearing that in my head. [Inaudible 0:44:13.3].

[0:44:14.2] AA: It doesn't change, right? All that's changing. That's the trajectory points. But then on top of that, there is a layer where it says, "Well, if I'm leaning too far to my left," I mean this is a simplification, "put my foot out here." React to that motion. There is that reactive layer as well. That's the robust – That's what you see acting when the big dog is on ice, when it's walking in snow. That's a reactive behavior that's all just a hierarchical algorithm, mathematical algorithm.

It's not learning. That's not what happens when you go on ice. In terms of ice, it's not a single perturbation to your behavior. It's an entirely new behavior you have to come up with. That's a new motion primitive. You have to learn that. By learn that, I don't necessarily mean machine learning that primitive. What I mean is you'd have to learn the fact that on ice, the friction model is different. Learn that friction model, put it back into the mathematical algorithms, modify the nominal behaviors and then make yourself robust.

That's where this feedback loop comes in. That's where learning will play a role, and that's what you do. I mean, if you look at a human, you go on ice and basically you're shuffling your feet around. You're learning the friction properties of ice. Once you have a pretty good model of those friction properties, you plug it into your nominal optimization method, which sits at your spinal cord. Then once you've got that down, you can walk fairly normally, because you've learned the thing you didn't know, and then you go back to doing the thing that you always do with a slight modification based on the different physical model of the world. That's the way we work, right?

It's funny. Human systems I think are great inspiration at every level, because it completely mirrors what we're finding on robotic systems inspirationally. Again, not in terms of we need to mimic what actually is happening in the human body, but the hierarchies where learning plays a role, where dynamics plays a role is really clear in the human body. I think it's great inspiration for robotic systems.

[0:46:02.6] SC: The same parallels are happening on the neural network side. Like we're taking inspiration from these things, bringing them in, try to evolve the way we think about the learning side.

[0:46:09.8] AA: Definitely inspiration is huge. But again, a word of caution is stay away from mimicry. Don't just try to create the exact same thing on a robot or in an AI. Well, there is X number of neurons in the human mind, so if we can hit that neuron, we'll have a smart robot. It's not the way – just like if you flap something it won't necessarily fly. But yes, look at the structures, really the structures are key and try to understand what they mean and then realize them on robotic systems.

[0:46:41.0] SC: Well, I really enjoyed this conversation. Any final words for folks or how can folks find you and learn more about your work?

[0:46:46.9] AA: The internet is available to find myself. My lab website is bipedalrobotics.com; just a simple name. You can find me on the Caltech website. Just Google Aaron Ames and there should be enough stuff. My students are on there too. They do amazing stuff. A lot of the videos you see are from my grad students.

If you're interested in robotics, please come to grad school somewhere. Feel free to ping us if you're really interested in Caltech. In general, studying these problems. This is where the fascinating point right now, and I think that's amazing.

My couple final closing statements are this is massively exciting, but be careful of the hype. Instead of just going for the hype, think about these things, think about where learning will play a role look to unification. Because that we can achieve these promises that are being made, but we have to be very smart about how we approach the problem. That's what makes it fun right now. These are not solved problems, and anybody that says they're solved I think doesn't know what they're talking about.

We're at this point where we're really trying to understand how learning and for example, robotic systems work together. It's an exciting time to be doing this, so I encourage everyone to really dig into it and see what they can learn.

[0:48:06.2] SC: Well, thanks Aaron.

[0:48:07.9] AA: Thanks so much.

[END OF INTERVIEW]

[0:48:12.5] SC: All right everyone, that's our show for today. Thanks so much for listening and for your continued feedback and support. For more information on Aaron or any of the topics covered in this episode, head on over to twimlai.com/talk/87. To follow along with the AWS re:Invent series, visit twimlai.com/reinvent.

To enter our twiml1mil contest, visit twimlai.com/twiml1mil. Of course, we'd be delighted to hear from you either via a comment on the show notes page, or via Twitter to @twimlai or @samcharrington.

Thanks once again to Intel Nervana for their sponsorship of this series. To learn more about their role in DeepLens and the other things they've been up to, visit intelnervana.com.

Of course, thanks once again to you for listening and catch you next time.

[END]