Interviewer: So if you could just start by telling us your name and where you were born and grew up and went to school.

Radhika Nagpal: Okay. So my name is Radhika Nagpal and I grew up in Amritsar in India, and I was actually born in the US. My father did his PhD at Georgia Tech. But when I was eight, our whole family moved back to India and I didn't come back to the US until undergraduate, which I did at MIT.

Interviewer: How did you decide to go to MIT?

Radhika Nagpal: <laughs>

Interviewer: What did you study there?

Radhika Nagpal: Well, at the time, Amritsar had a lot of turmoil, and there was a lot of terrorist activity and the city would sort of shut down. We used to have curfew at 7:00. So I think, as a high school student, my goal was to get as far away <laughs> from home as I could, and I applied to a bunch of schools in the US, figuring that was far and figuring that I could potentially go there. But I didn't know a whole lot about any of the schools and there wasn't the Internet. So we would mail and they would send us little brochures, and the brochures was basically all I knew, and my father knew a bunch about the universities. So I applied to five or six universities. And I decided to go to MIT for a couple of reasons, one of which was that, of course, MIT is really well known for being a great engineering school. But also, they had-- and this was 1989. They had a great program where they paid for your tuition if you couldn't pay for yourself. As long as you were admitted, they figured out how to cover you. So my parents were pretty poor at that time so it was a really big deal that they could do that and the other universities couldn't. And so I got to go to college <laughs> thanks to that program.

Interviewer: What did you major in?

Radhika Nagpal: I didn't know what I was going to major in, actually. So I went in thinking I would do electrical engineering. But mostly I knew that I didn't want to do biology and I didn't want to do mechanical engineering and I didn't know anything else. <laughs> But my first computer science class, six double "O" one, and that was it and then, I think six double "O" four, which is the computer architecture class. So programming and making computers and I was hooked.

Interviewer: How did you wind up getting interested in robotics?

Radhika Nagpal: That's a pretty long path. So I was initially interested in computer architecture and from that I got interested in networking and distributed systems and parallel computing. And I got a chance to work at Bell Labs and that was really fun. That's actually where I got interested in research. That was the really big eye opener for me, for what it meant to do research, how people had fun doing research. <laughs> And so I told them, I was, like, "I want this job." And they said, "Go get a PhD." But when I joined for a PhD, I sort of floundered a little bit. I wasn't sure what I wanted to do. A lot of topics that I thought I understood were really different from what I had predicted. And there was a new project that just started at that time, called amorphous computing. It was Jerry Sussman, Tom Knight, and Hal Abelson, and the idea was that we would learn from biology as a way to think about programming systems that had huge number of parts. If you think about a distributed system, like a network, they're saying, you know, cells are a network or ants are a network or physics is a network of molecules, and if you could compute with large numbers, you would think really different about computation. And it was just such a-- they published this white paper and it was so exciting. So a bunch of people flocked to that group <laughs> leaving other groups. And, actually, the area called synthetic biology grew out of that, as well. So it was like, not only would we learn from biology, we would program cells the way we program computers. That was kind of the idea. So it was a really, really exciting time, and I think that in the beginning, we thought, you know, "What would be a physical instantiation of this idea?" So putting sensor networks was one example. Modular robots that are kind of, like, made of different modules, another example. And then, people would talk about, like, smart dust and so it was a super exciting time. A lot of different fields were, sort of, pushing in this area of making really cheap individuals. But even then, I didn't want to do robotics, and one of the reasons was it was really hard and it was really expensive to own a robot. Computation was really hard. Cameras were really poor. So by the time I graduated, it was still super hard to do robotics because all of the processing and information and sensing was very different. But as I started as a faculty, maybe three, four years down, 3D printers started coming online. Just so many different ways you could actually build robots. You could put kits together, and the cost went down and then, I thought, "Oh, I don't have to write a grant, then." <laughs> Because I thought, "Oh, you know, if I wanted 10 robots, that would cost me a huge amount of money." And as soon as that cost went down, I got excited about owning robots, and so I was, maybe, my third or fourth year-- yeah, about, in faculty when I really started doing robotics. Not just talking about-- not just doing the theoretical side, but actually owning robots and building robots, and now we have a thousand robots in my lab. <laughs> So it was a very quick trajectory from nothing to a thousand.

Interviewer: So what year were you at Bell Labs? Who did you work with there?

Radhika Nagpal: I was at Bell Labs in 19-- <laughs> this might be tougher. You might have to verify this. '92? No, '93 and '94. So I first went there-- MIT had a program where you could partner with a company and you would do your master's with the company. So you would go and spend a whole semester there as part of your five-year program. And so I spent a semester there and then I spent a winter there, and then I deferred grad school and spent a year there. <laughs> Because it was a lot of fun. So I worked most closely with ~~Ray~~ Rae McLellan ~~[ph?]~~, who works in computer architecture. But I was actually right next to the Unix room. So I got to meet Brian Kernighan and Dennis Ritchie and Ken Thompson and Dave ~~Prizado [ph?]~~Presotto. And so it was this whole group in computer science that, especially at that point, C was the only language we used, and ~~________~~ Bjarne Stroustrup and C++ was sort of starting to happen, and so it was just this illustrious group and they would all go to lunch together. And I could go to lunch with them and they'd talk about their hobbies and they'd talk about their passions and then, we'd go and we'd work on ~~________~~ <inaudible> and it was great. Because Bell Labs, everybody-- there were not that many women and there were not that many students, and I got treated like royalty. <laughs> So it was really exciting to have all these super famous people, who still know me and still check up on me every so often to ask how I'm doing. So it was a really-- for me, a major experience to be at Bell Labs.

Interviewer: And for your PhD, what was your thesis topic and who was your advisor?

Radhika Nagpal: So my thesis topic was with Jerry Sussman and Hal Abelson, and my thesis topic was really about how you could take an idea of what you want a collective to do-- I mean, this is actually, maybe-- you know, it was my first example of being able to do this, and since then, a lot of my research is built around this idea that if you have a collective of individuals and they all have simple local rules, what you can't do is design the rules bottom up. Because you're just going to be stuck trying to see every variation of what goes on, and if you look what an individual is doing, it's not well connected with the global one. So is there a way to go inverse? Is it possible to write, like, a compiler where I say, "Well, what I want the group to achieve is this?" And then, the computer sort of figures out what it is that all of the individuals should do. But that program has to somehow deal with the fact that some people may not-- you know, some of your robots may not work or some of them might get lost. Or you don't know exactly if there were 1,000 or 1,200; right? So at that number, you don't want to be counting any more. So traditional planning thinks about maybe smaller groups that, as an individual, what can each of us do and how will we coordinate together? ~~Another~~ In our one, we're thinking you have a bunch of identical individuals and they're not able to coordinate well but you still want to achieve something. So you can up the number, but you can't make them more predictable. So in my thesis, I actually was thinking about a programmable material that might've been made out of many different actuators that would fold. And what's interesting is there are now folding robots. <laughs> You know, so for me, that was an idea and a way of thinking about a new sort of active material or active environments.

This was a lot of computation would be embedded in everything. That was kind of the idea, and if it was embedded in everything, then it's programming. It means you can't be rebooting individual bits of everything. It has to be sort of self-managing in some way. So I was thinking about active structures. But most of, I think, the real gist of what I did was show that you could take very complicated ideas globally and systematically compile them.

Interviewer: And who else did you work with or interact with while you were a graduate student?

Radhika Nagpal: So I think it was a pretty active time. So, obviously, there was all the synthetic biology stuff going on so Tom Knight was sort of one of the-- the iGEM started around that time, which is these huge competitions where people come and program cells. I was part of, like, the first set. We were maybe 20 of us and now, the competition has, like, 800, 900 people. So it's huge. It was a huge growth. But the other group is, of course, the AI Lab, and so I was always connected with people in robotics. So Holly Yanko-- actually, all of Rod Brooks' group. <laughs> Just because what I was interested in is very-- what I'm interested in now is very closely related to embodied intelligence, the idea that complexity that you see isn't necessarily arising from complex decisions and complex thinking at the level of the robot. It's arising from the interactions with the environment or the interactions with others and that that's what you want. You don't want to make a more and more complicated robot. You want to make a simpler and simpler robot. So his group had a lot. And in fact, I read, I remember, for one of my-- they have something called an area exam, where you're supposed to read papers, and I remember that Rod Brooks assigned me Lynn Parker's earlier papers and ~~Myah Materick's~~ ~~[ph?]~~Maja Mataric's early papers. And so that's sort of how I got to know them, and Cynthia ~~Brazil~~ Brezeal was a graduate student there. Holly Yanko was a graduate student there. Brian ~~Scazolatti~~ Scassellati ~~[ph?]~~ was a graduate student there. There's a huge number of people from that group that are now faculty in different places, in robotics, but also in other areas. But very much, like, the embodied intelligence kind of area was a big connection for me.

Interviewer: And when did you finish your PhD?

Radhika Nagpal: 2001.

Interviewer: And where did you go after that?

Radhika Nagpal: So at the time, 2001 was an interesting year. <laughs> So my entire game plan was to go back to Bell Labs. It was very clear to me. I'd come to get a PhD. I just hoped that after I did all this crazy stuff for my PhD, they would still hire me. But 2001, all of the research labs were tanking and so Xerox Park, which was my other favorite place, because they did modular robots, they were in trouble and DEC was in trouble and everything was in trouble. And Microsoft was just starting and Google didn't really-- I mean, Google existed. But the research lab didn't exist. So, suddenly, research labs were just not an option anymore, and so I also had my daughter when I was a grad student. So I had a little kid and I was finishing my PhD and it was like this is so complicated, like I just want~~ed to f~~ to defend ~~[ph?]~~. <laughs> And then, I'll think about what I'm going to do with my life. But at the same time, they started these lecture positions at MIT. Because a lot of faculty were on leave starting startups. So I decided that I'd be a lecturer for a couple of years, which basically was a term-limited thing, as a way of seeing if I liked being faculty. You know, it seemed like a good trial. I'd never really tried being a teacher before, and I really liked it, and so I did research and I also taught students, and it was like getting a little taste of what it would feel like to be faculty. So at the end of that, I applied for faculty positions, and I came to Harvard with one more detour. Actually, I have-- my career is defined by detours. <laughs> Wherever it's like, "This is the path," I take a little short deviation and come back, then a little short deviation. So my second short deviation after the lecturer position was to spend a year in a biology lab, and this was super fun. So this was maybe the second super influential thing for me was systems biology and synthetic biology was also rising at the same time, and they were really interested in collective behavior. And they were open enough to want to start conversations with computer scientists and mathematicians and physicists, and try to start bringing everyone together to think about important problems in biology. And so when I talked to them about active materials unfolding, many of the ideas had been taken from developmental biology, and so we connected and the new department chair there, Mark Kirschner ~~[ph?]~~ said, "Well, why don't you come spend a year here?" And he said, you know, "When you become faculty, you'll never have time again. But if you come for a year here, you'll influence everyone. We'll influence you. You can do some experiments." And so I actually spent a year where I-- and I say, "I tried." I tried very hard <laughs> to do actual experiments on the same organisms that I had read papers about, and it was an eye opening experience. I mean, it was such an incredible-- that group still is super connected. Because even though all of us now are much more senior and sort of established in our places, there's actually a lot of connection still between how we see groups working together, you know, how cells work together. So, actually, just this August, one of the people who was sort of a mentor for me then, we went and taught his group how to program kilobots. And so we had 20 biologists programming robots and I was, like, "This is just heaven." <laughs> But yeah, so I tried to do experiments and I learned a lot about how biologists think about this problem, and also, how hard it is to reverse engineer a real system that is robust, is doing all the things you care about but now, you have to sort of infer back what's going on. There's just limited tools you have to sort of ask that question. Whereas with robotics, we build it and then we still see things we didn't predict. But at least you can now start to work from first principles what happened, whereas that's much harder to do in biology. But that discovery process is very, very similar.

Interviewer: And you went into the computer science department?

Radhika Nagpal: The computer science department, yeah.

Interviewer: And were there other people doing robotics there at that time?

Radhika Nagpal: There were not. <laughs> So it was kind of-- so Harvard has a small group. But one of the defining features of the group, especially when I joined, was that there's a lot of interdisciplinary people. So there is David Parkes, works in economics in CS; Barbara Grosz~~s~~, who's been well known in AI for a long time, works on human computer interaction; ~~Stewart~~ Stuart Shieber ~~[ph?]~~ works on computational linguistics. So when I said, "Oh, I want to work on computation and biology," they were like, "Great." You know, and that was not the reaction I got from a lot of universities I went to. Whatever I did was weird. Whereas here, what I did seemed normal, in a sense. So early on, we started doing some robotics, and my first robots were actually built off of Mindstorms or even, literally, the older Mindstorms. So Lego robots were my first robots. And I think one advantage of not having a robotic~~s~~ist there ~~<inaudible>~~ was that I didn't have to be too embarrassed about it. There was-- actually, I take that back. Rob Howe was there, and he does hands. But he did also a lot of surgical robots. So there was a lot of distance then between what he did and what I did, and I didn't even consider myself a roboticist. I considered myself an AI. But over time, there's now a lot of people. So there's Rob Wood, who does insect scale robots. There's Conner Walsh, who does exoskeletons, and Rob Howe. So now, suddenly, we have this robotics group. Whereas before, when I joined, it was really sort of me alone trying to navigate my way. Of course, I wasn't that far from MIT so I could always go back and <laughs> get-- you know, I have a huge network to ask questions. So Holly Yanko and James ~~McLerkin~~ McLurkin ~~[ph?]~~ especially did a lot of swarm robotics at MIT at the time. So when I started my faculty position, when I taught swarm robotics, he would bring over his robots and teach a class. But in the process, he really taught me a lot about what is important in thinking about robots. What things bog you down, what things are important when you think of designing them. So in fact, many of the things that he wrote about and talked about influenced what we did with kilobots. Because he was the one who used to do swarm robotics and he had so many lessons that we learned from. So it's really these interactions. You sometimes have interactions because you're friends. <laughs> So James is friends with both me and my husband, and we just had things in common. So we would talk all the time and I was, like, "I will never have robots. You can just bring your robots." And now it's sort of the opposite. We all have lots of robots and it's great. Now, we talk about where ~~are~~ we are going with all our robots.

Interviewer: What was the first project you did with Mindstorms?

Radhika Nagpal: So interesting-- in one of the projects that I've been interested in since the beginning of my faculty position is self-assembly, but self-assembly where robots are building something. So inspired by how termites build mounds. So we did a lot of theoretical stuff on that and then, our first implementation was robots moving around tiles. And the idea was you needed only a few simple sensors and local ideas to do it. So we thought, "Well, we should be able to implement it with something as simple as the Lego Mindstorms robots. And it was sort of also a proof; right? If you say these are simple robots, well, how simple is simple? So the nice part about-- at that time, especially about using the Mindstorms was that, of course, you could change the body. So I found that a lot of things that interest me in robotics involved the body of the robot. And so you can't just go somewhere and say, "I want a robot that has this design and this arm and is positioned like this." You just get a robot, like a pioneer, or something. Something ________that can't move around and look. But I always wanted something that could move around and manipulate or climb, and those were always in the research category. Those were never things that were easy to buy. But with Legos, you could kind of build whatever you needed to build, and so we basically built the custom robot that we needed, and it was very complicated. <laughs> And it used, like, two computers, basically, or two of the Mindstorms bricks, and it has a lot of parts. But it had an arm that can move up and down and it had a gripper that could close and it had lots of touch sensors and vision sensors, and so we could actually implement the whole algorithms just in 2D. And I think that way, for me, 3D printers were, you know, an amazing enabler. So now, I have a lot more students in my lab who can imagine things and then they just happen. You know, we don't have to imagine things and then deal with the difficulty of trying to make that thing and the fact that it might be too heavy or it might be difficult to machine or it might take too much time. Or maybe the student doesn't have the physical skills. Even though they have the mental creativity, they don't have physical skills to do it. Those used to stop a lot of our projects before. So we would just stick with the Legos. And now, literally, students come and I'm, like, "Well, let's imagine that you wanted to make an army ant that was crawling on top of army ants and making a tower. How would you go about-- what would be the design of that robot? What would that robot need to know? What would it think? What sensors?" And, literally, we can compose those ideas so fast. So I feel like you're no longer constrained. And with 3D printers, you can also make many, which is another sort of fun thing. You can go through revisions. But at the end, if you have a good design, you don't have one robot. You can have 30 robots and that's a really big enabler, I think.

Interviewer: How did that research trajectory lead you to the kilobots?

Radhika Nagpal: <laughs> Well, there's like more and more and more;, right? So James McLerkin McLurkin always had, like, this beautiful swarm. But then, there's also a couple of groups in Europe that really have been in the forefront of this, and in particular, at EPFL, Floriano's Floreano's [ph?] group and Martinoli's [ph?] group and Dorego's Dorigo's [ph?] group in Brussels. So there's huge-- they've always been interested in swarm robotics. So I sort of came to that a little late. But many of them I met at conferences and I talked and they-- you know, sort of, these conversations where you run into somebody. Like, I ran into Martinoli, Alkura [ph?]Alcherio Martinoli, at an AI conference and we were sort of some of the few people doing robotics at that AI conference. So we ended up chatting and then, five hours later-- <laughs> we touched, like, so many topics. I had no idea that there was somebody who was interested in so many of the same topics I was interested in. But that relationship turned out to be really great. So they designed robots like the E-Puck, and his group was pushing numbers. So at that point they had, you know, swarms of about 200 robots, and that robot was one of the first robots I bought. So I bought robots that other people had designed, in order to use in my class or to use in my research. But you always sort of run into this thing where it's actually a great thing for classes, but for research, you always want to modify the robot in some desperate way. So we just sort of had lots of robots, and the TERMES robots are climbing robots. So those were-- we sort of started with those. But then, Mike Rubenstein joined my group, as a post doc, and his thesis was really closely related to things that I liked in my PhD thesis. So we had a lot in common and he thought about self-repair and developmental biology and self-assembly. And so I thought he was going to come to my group and we were going to work on future, sort of, theoretical ideas. And we did for a few months talk about various ones. But at the end of a couple of months <laughs> he came to me and he's, like, "You know, actually, what I want to do is build a thousand robots." And I was, like, "You've got to be kidding me?" <laughs> I was, like, "Have you seen the E-Pucks? We can't, like, handle 20 of them. You've got to be kidding me, with three TERMES." And I was, like, "Okay. Well, so what's going to make you succeed where everybody else failed? Because the problems are very real." But I think that just as James had influenced me, James and other people had influenced him. So he had the list of problems and we sort of started thinking about what were the key issues that stop you at a hundred. And I think he just turned the question around. He said, "You know, what if we start designing by assuming from the beginning that we'll make 1000? We won't say we'll make 10 and then we'll make 20 and then we'll scale up. We can only scale-- we can only go backwards. So we're going to start at 1000." So if you start at 1000, there's a whole bunch of things you're just not allowed to do. Like, you can't have manual labor in designing these robots. You can't have too much cost per robot. You need a lot of things to be made by pick and place machines. So once we started that trajectory, it was really more obvious how we were going to get there. But the two sort of pretty innovative ideas that Mike came up with were using vibration motors instead of regular wheels. And also, sort of, instead of using regular wireless or regular sort of IR into the upper <laughs> part of the atmosphere, using reflected IR, is just a lot more robust. So there were lots of problems I knew people had with robot-to-robot communication. So his technique basically avoided a lot of those problems. And the vibration motors were really fun. I mean, we'd seen a lot of the bristlebots and toothbrushes, and so we put these things on. But he was, like, "I think I've figured it out." And so we'd sit there and we'd bend the legs and everything, and it turned out that we must've spent six months or so on the locomotion strategy. Not realizing that we actually had no understanding <laughs> of what we were doing. So finally, after-- like, every time we would make a robot, it would locomote differently. So in one case, he bent the legs backwards and the robot went front. In the one case, he ________bends them back and the robot went backwards. We're like, "Okay. <laughs> This doesn't make any sense." So somebody from another group, ________Mahadevan's group, suggested that we put it under high-speed photography to see it. And then, we realized that the locomotion was really different from what we thought, and once we realized that, it turned out that the vibration motors have a bias. So what he had done is, in one robot, he had placed them one way and another in another way. But they look symmetric, so you don't really notice that you were doing that. And that's why all our robots were behaving differently. So sometimes, you think you know what you're doing, but especially with mechanical behavior and new mechanical behavior, often, you know, you really need to test out your intuition. Otherwise, it's so easy to be wrong. So I think a lot of what I've learned in the last five, six years, is just continually trying things. Because you try them and it gives you new intuition and new ideas. If you wait until you figure you have the whole problem solved, you may have just missed something that was really crucial to thinking about the problem. And I think that's maybe how a lot of roboticists feel, that the real world is much more complex than we want to give it credit for. So-- or than we want it to be. But sometimes, it can go the other way around that it can actually be easier. So the vibration motors actually turned out to be easier than we had predicted. You didn't need to bend the legs, at all. The legs were actually irrelevant. <laughs> So it turned out, you know, we didn't have to have any precision on how the legs were put in. So it actually-- our life became easier as a result of understanding what was going on.

Interviewer: So positioned as you are between, sort of, biology and computer science and robotics and the sort of longer trajectories of self-assembly, self-organizing systems, bionics, biologically ________, inspired, what does robotics really kind of bring to this equation? What have you learned from ____________?these sort of older histories?

Radhika Nagpal: Well, I think that-- I mean, there so many--I think that a lot of the areas that you mentioned, you know, we think of physical system--and I think that, in that sense, computer science and robotics go very well together. Robotics is that part of computer science that touches physical systems directly. But also brings in, like, mechanical engineering and electrical engineering. What are other ways of thinking about the same problem? So I find that, like, if you think about programmable materials, it's not really a robot. It's not a robot in any sense of the word that we might've thought robot meant. But it is an essential part of robotics because it's basically endowing physical things with behavior. And to me, I think one of the interesting things is that I now look at everything in biology as a robot; right? It has behavior. It has mechanics. It has physics. It has an interaction with the world and an interaction, and often the questions in biology are the same. How much comes from the body versus the brain? Or even if you were thinking of cell, how much comes from physics versus a cell's active decision to do something? So when I interact with _________ and we were teaching biologists to do robotics, what he was saying was he felt that robotics helps the biologist think algorithmically about cells. What is the algorithm that a cell runs? And I was just so-- my jaw <laughs> was on the floor. I'm, like, "He said the word algorithm." So I think it works both ways. You know, if you think about biological systems as running programs, that actually gives you a certain power in thinking about biology, as well. So I think robotics brings a huge flavor to all of these fields, and you can think of it also as AI brings because it's the same concept. But the fact that there's also a physical-- there has to be a physical part to it. It's not just what decision a cell makes. It's also about the forces a cell experiences. Or, you know, people will think about, "Well, maybe, there's external forces on something and that causes the cells to align" Or maybe the cells are actually measuring forces from nearby cells and aligning." That's a big difference between what kind of program is in the cell in the first case and the second case, and people want to know the difference. They want to know if they have to manipulate the genetic code of the cell or they need to manipulate just the environment and the cells will correctly heal something. So a lot of these questions are really important but they're kind of the duel dual of questions in robotics. So I think now that I've had a taste of building physical systems, it would be very hard to go back. You know, and I think that that's actually also the part that robotics brings is it's really fun and it's really tangible. And for me, anything, sensor networks, smart houses, all of these are sort of not so separate from robotics. As you may have already heard, there are a lot of people who work in a lot of cross areas with robotics. But I think it's just embedding computation into the physical world in some way is something that roboticists feel comfortable with.

Interviewer: And who are some graduate students or post docs that you trained and have gone to do work in robotics?

Radhika Nagpal: So mostly in robotics, I've had very recent students. So Michael Rubenstein is now a research scientist, as is Justin Orafel Werfel [ph?] and Kierstein Peterseon. But that's my most recent set. So most of my earlier students did AI, more AI, more theory, and some biology. So we do work with biologists, as well.

Interviewer: Unfortunately, we're running out of time...

Radhika Nagpal: Yeah.

Interviewer: But the wrap-up question is, what's your advice for young people who might be interested in a career in robotics?

Radhika Nagpal: That's a tough one. <laughs> Just do it. Just do it. You know, it's really fun, and I think a lot of times we spend too much time worrying about what we're doing. Especially, smart people spend too much time <laughs> worrying about what they're doing. And sometimes you need to just do it and find what you like. I didn't start by liking robotics. I started by liking something else. And I don't promise that robotics is the only thing I'm going to do with my life. I don't know. You know, every year, I'm like, "This is what I love the most." Five years down the road, it's something different. So if you like robotics now, do it now. Why worry about anything else? <laughs>

Interviewer: Thank you very much.

Radhika Nagpal: All right.

End of RadhikaNagpal.mp4