
UNIVERSITY OF PENNSYLVANIA

Robots That Crawl, Walk, and Slither

200 years ago a dishwasher would have been considered a robot (not to mention a miracle), but it's just a machine to us.

by Joel W. Burdick

Opposite: (from left) Howie Choset, Jim Ostrowski, Greg Chirikjian (PhD '92), and Joel Burdick pose with their robot snake.

What, exactly, *is* a robot? There are a lot of popular notions about what robots are, so let's begin with some definitions. The word *robot* comes from the Czech word *robota*, which connotes obligatory servitude. The word was first used in its current sense by Czech playwright Karel Capek, in his 1917 play *RUR*, which stood for Rossum's Universal Robots. Many of us working in robotics continue this usage implying servitude, and define a robot as a machine that, once programmed, is capable of independent action, and that can be reprogrammed (at least to some extent) to do different tasks. Robotics is just the latest in a long history of engineering's quest to improve mankind's reach and mobility (i.e., to move faster, farther, and higher) and to multiply human ability in order to improve our standard of living. In fact, much of robotics research was originally motivated by the goal of automating factories. That goal hasn't been reached, but it now appears that robots may have greater potential outside of the factory anyway.

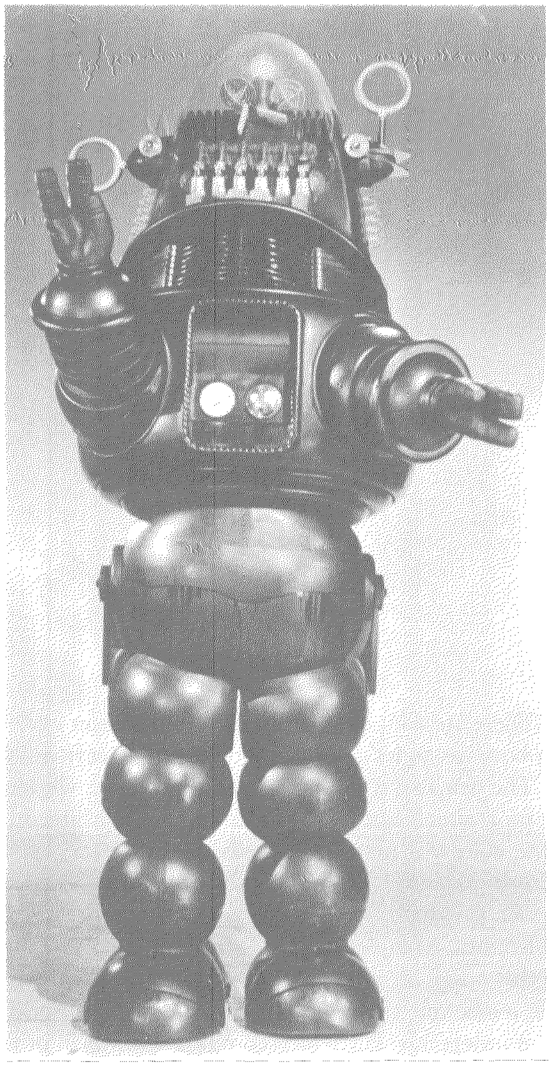
The word robot is really a metaphor for the next generation of automated machines. Once a certain level of automation becomes widespread, we no longer call it robotic. For example, 200 years ago a dishwasher would have been considered a robot (not to mention a miracle), but it's just a machine to us. Today's robots are essentially complicated electromechanical systems that are controlled by computers. This makes robotics research almost a branch of computer science. Hence, much of robotics research centers on developing computer algorithms—mathematical expressions or computer codes that take inputs from sensors, refer back to the assigned task, and output an action—that will enable robots to accomplish what we want them to do. Not only must the algorithms do complex things, but they must compute very quickly. A vision algorithm to detect falling boulders isn't much use if the robot is crushed before the computation finishes.

In addition to the quest for enhanced productivity, another grand theme of modern engineering is the drive to build increasingly large and complex systems in order to satisfy societal needs. Unfortunately, recent events such as the near-collapse of AT&T's phone system show that we're not very good at it yet. One reason robotics hasn't gotten as far as early enthusiasts had hoped is that we've underestimated the complexity of building "intelligent" robots. Most of the robots that people envision in science fiction are really of the complexity of the national phone system. Take Gog, one of my favorite science-fiction robots, from the 1954 movie of the same name. It could not only play tennis, it could also build nuclear bombs. And, unfortunately, a programming glitch caused it to almost destroy the world. A lot of engineers like me are interested in new approaches for designing and operating complex systems. Robotics provides us a convenient way to study such systems without using quite as much hardware as the phone company.

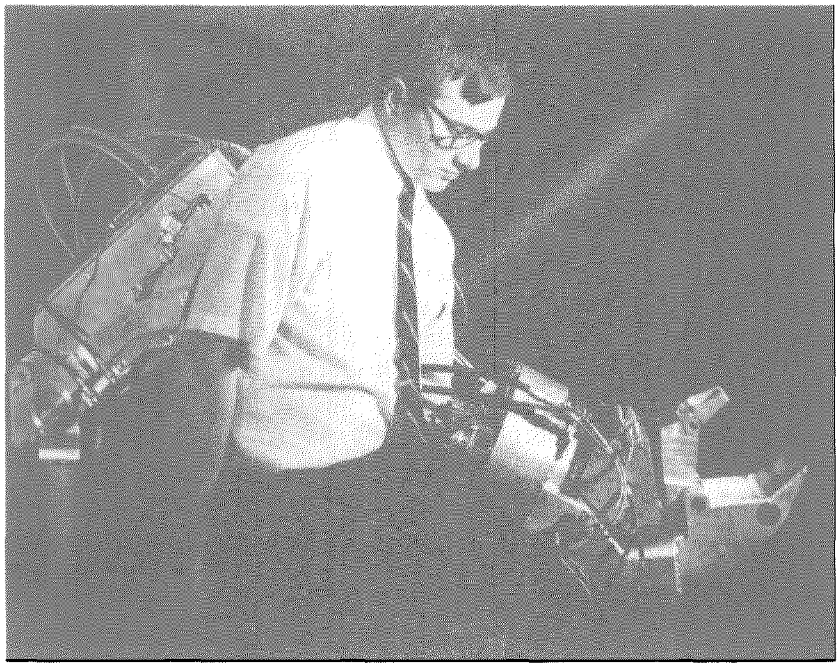
Mankind has been interested in building mechanical servants for a long time, but modern robotics dates from about 1960, when the first computer-controlled factory robot was installed. Its computer was quite crude by today's standards, but it *was* reprogrammable. By then, digi-

Right: Robbie the Robot (from the 1956 movie *Forbidden Planet*) is perhaps the archetypal science-fiction robot.

Below: The Hardiman project, developed by the General Electric Company for the Army and Navy in the late 1960s, never got much farther than this prototype arm. Its descendants live on in Hollywood, however, as in the "power loader" Sigourney Weaver wore in *Aliens* (opposite page).



Still from *Forbidden Planet* Courtesy of Turner Entertainment Co. and the Academy of Motion Picture Arts and Sciences. © 1956 Turner Entertainment Co. All rights reserved.

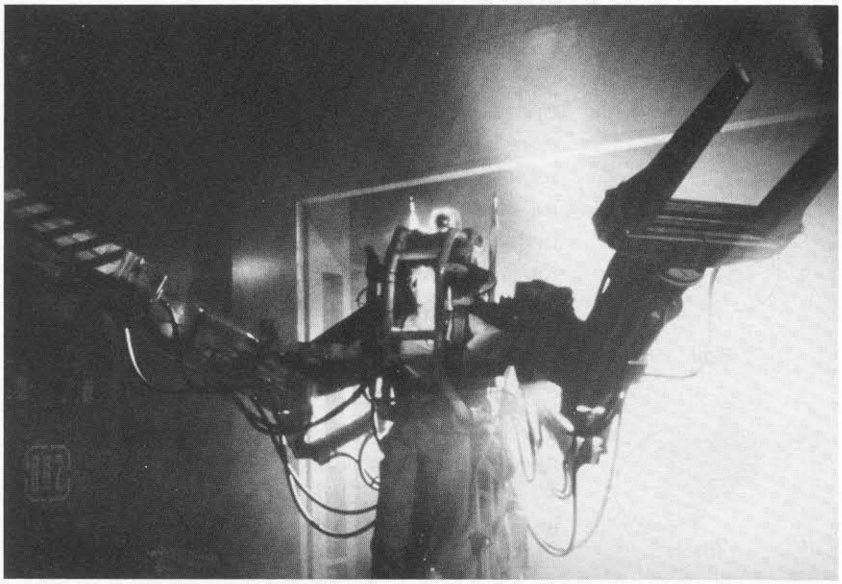


From 1960 to 1966, people were just experimenting with building different kinds of machines; there was no clear vision of what the significant problems of robotics are.

tal computers had been around long enough for people to begin thinking about smart machines for industrial automation. It made sense to remove people from assembly lines because such jobs are boring, dangerous, or costly. The year 1960 also saw the beginning of commercial nuclear power. The nuclear industry had a different vision for robotics. They wanted to build mechanical proxies so that humans wouldn't need to enter highly radioactive, dangerous, or inaccessible areas. Such robots weren't necessarily smart, as they would follow the motions of a human operator.

The early history of robotics is filled with ideas that either never panned out, or became evolutionary dead ends because of limitations in technology. One was General Electric's Hardiman, which was like those "power loaders" in *Aliens*—a machine you wore that increased your strength so you could lift heavy cargo. Hardiman was never completed, as the computers of the day weren't up to the task of controlling it, but there have been recent efforts at the University of Minnesota and the University of California at Berkeley to revive the concept. Some other dreams from the early days, such as multi-armed robots and robots with flexible torsos—things people never dreamed would be as hard to animate as they are—are also being revived, now that computer technology has become vastly more sophisticated. In general, from 1960 to 1966, people were just experimenting with building different kinds of machines; there was no clear vision of what the significant problems of robotics are.

The golden age of U.S. robotics research



Aliens © 1986 Twentieth Century Fox Film Corporation. All rights reserved.

There were some grand failures, such as the toilet manufacturer whose robots kept crushing the fixtures.

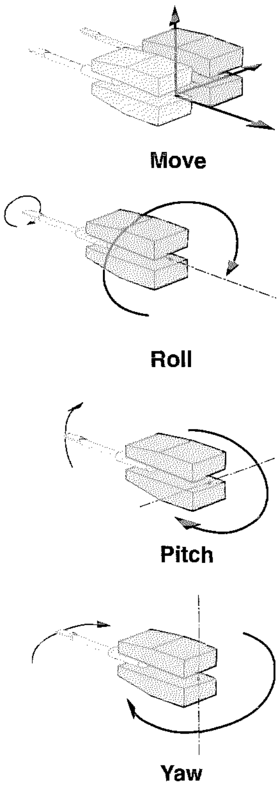
began around 1967–68. It was an exciting time of incredibly rapid advances—an explosion in robotic capabilities. Scientists and engineers were getting a clearer idea of the important technical problems, and developing the basic analytical and technological tools to solve them. Many students who earned their PhDs in robotics research immediately founded companies to transform their research into actual products, making the delay between laboratory research and industrial application very short.

The late 1970s and early 1980s saw an expansion in industrial robotics, and many corporations started experimenting with factory automation. Some applications, such as welding and spray painting in the automobile industry, proved successful. Others did not. There were some grand failures, such as the toilet manufacturer whose robots kept crushing the fixtures, and many inconclusive efforts that proved too costly to continue. There are several reasons why these efforts failed. In part, U.S. manufacturers didn't understand the capabilities and limitations of robots. The problem was often not in the robot, but in poor management techniques and poor manufacturing processes that couldn't be fixed by the magic bullet of robotics.

These sometimes naive and invariably expensive forays into automation were often driven by the overhype that predominated in the robotics industry during the late 1970s and early 1980s. In 1990, there was a great article in the *Los Angeles Times* entitled "Prediction Has Become Robotic," which pointed out that ever since the 1930s it's been predicted that ten years from now we're going to have robots in the home doing all sorts of things for us. Ten years go by and nothing happens. Then the hype begins again. In a typical example, the Quasar Company announced in 1978 that within two years it would be offering a $4,000 robot that would cook for you, mow your lawn, and vacuum your rug. We haven't seen it yet.

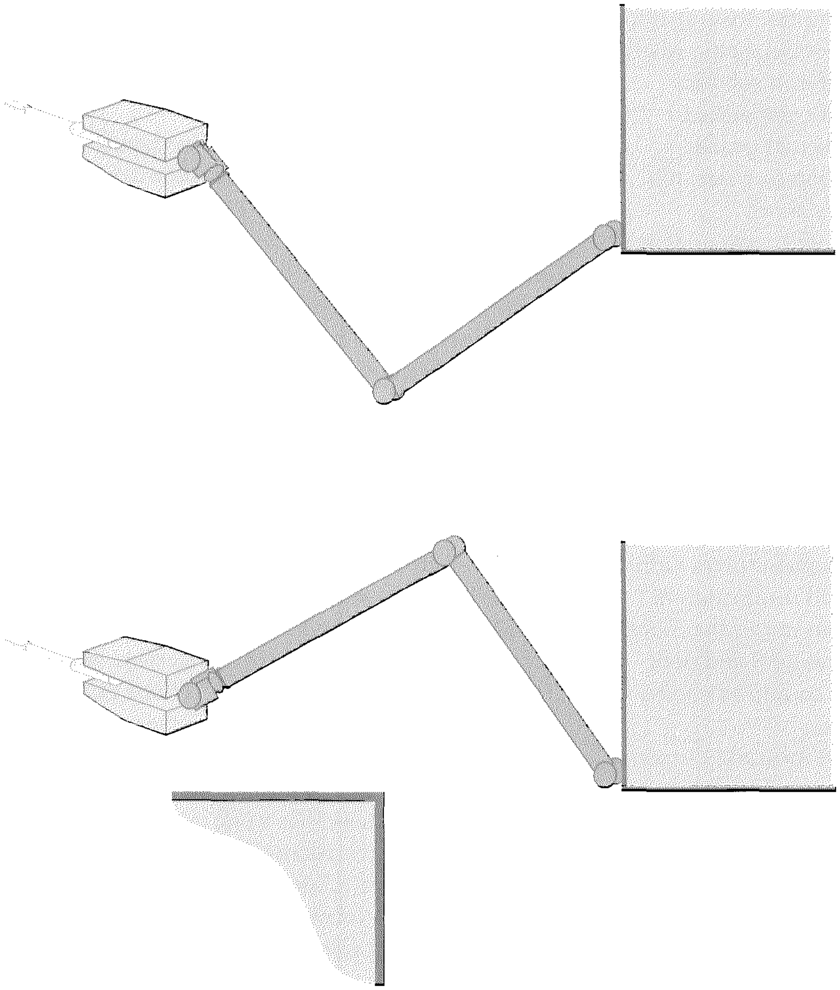
Because of the overhype and the dismal results of many experiments, American industry turned sour on robotics in the mid-1980s. The U.S. robotics industry started declining in 1987, and only recently has it begun to grow again. Back in 1979 people were predicting that the U.S. robotics industry would be a $4-billion-a-year industry by 1990. It's not, although Japan's is. Japan has about five times as many robots installed (about ten times as many per capita) as we do now.

The almost-exclusive focus on industrial automation had some negative effects on the rest of the field. One was that when the robotics industry didn't grow, funding for robotics research fell off. The other was that the preoccupation with industrial automation limited our vision of what robotics might do outside of the factory. Manufacturing is only about one-third of our gross domestic product. There are many other economically and scientifically important areas where robotics can have an impact: robots that explore other planets; robots that work in nuclear plants or toxic-waste sites; robots that work in the service industry; and robots that assist in medical procedures. The robotics community is now pursuing these applications. However, the activity level has been small until recently, because of



Right: A robot arm needs six degrees of freedom in order to position an object: one for each of the three dimensions of space and the three different kinds of rotation.

Below: Extra degrees of freedom allow the arm to avoid obstacles.



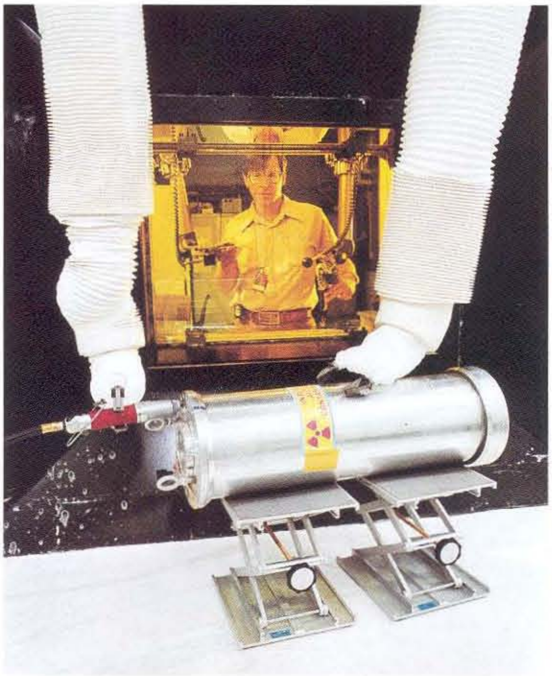
the earlier fixation on industrial automation.

As an academic, I can also claim that robotics research is useful in and of itself, beyond automation, medicine, and the other possible applications. Algorithms developed for robotics have recently found uses in other areas. For example, robotics algorithms developed at JPL are now being used by computational chemists at Caltech for molecular-dynamics simulations. And the next generation of air-traffic-control computers will be based on research into how robots could cooperate in a factory without hitting each other.

As for my own research, I study "dexterous multifunctional robots." By "dexterous" I mean nimble robots that can move in complicated ways. And by "multifunctional" I don't mean our friend Gog, who could fix a toaster in the morning, play tennis in the afternoon, and build a nuclear bomb at night. I mean that when we humans grab something we normally use our hands. But if need be, we can use other parts of our bodies—for example, slamming the refrigerator door shut with our hip while we have the mustard and mayonnaise jars in our hands, the baloney package in our teeth, the loaf of bread on one forearm, and the head of lettuce pressed between our other arm and our chest. While current robots may be mechanically capable of performing such maneuvers, they cannot figure out how to do it automatically, and thus require laborious, explicit programming by humans.

A lot of my research is related to a simple concept called redundancy. In Great Britain, if you're redundant you're unemployed, and in systems engineering redundancy means building backup systems. In robotics it takes on a different meaning. Most robot arms are built with six joints, because in order to grab some object, position it in three dimensions, and control its orientation in roll, pitch, and yaw, the arm needs to move in six directions—it needs six degrees of freedom. Adding extra joints to the robot makes it "kinematically redundant." Now, if both ends of the arm are stationary—because it's holding something in position, say—these extra degrees of freedom allow the rest of the arm to move internally. It can swing to avoid obstacles or to optimize mechanical properties, such as the force it's applying in some direction.

If adding a few degrees of freedom to a robot makes it more dexterous, what if we add a *lot*—hundreds, maybe thousands? Rather than looking like the typical robot arm, these things will look like snakes, and that's the slither part of my title. We call such robots "hyperredundant," a word coined by one of my first graduate students, Gregory Chirikjian (PhD '92), when he started



Above: Intelsat VI, stranded in a useless low orbit when its booster rocket failed, was plucked from space by (from left) Richard Hieb, Thomas Akers, and Pierre Thuot aboard the shuttle *Endeavour*. The satellite had to be retrieved by hand after unsuccessful attempts to grab it with a specially designed capture bar nearly sent it spinning out of reach. A robot tentacle that could wrap itself around the satellite to grip it might have had better luck. With Akers and Thuot keeping a firm grip on the satellite, Hieb was finally able to attach the capture bar so that the satellite could be stowed in the cargo bay while a new booster was attached.

Right: The nuclear-power industry uses robot arms, operated by remote control, to manipulate "hot" radioactive materials inside shielded rooms.

working on them just after I came to Caltech in 1988. These robots are not useful for industrial automation. But just as biology has evolved snakelike animals for certain niches, snakelike robots could be quite useful in some areas. In the nuclear industry, for example, they could crawl around inside reactor cores, which are very complicated spaces containing lots of obstacles. Or imagine that you want to retrieve a satellite in orbit. If you don't grab it properly, it can go spinning off into infinity. This almost happened on the Intelsat VI rescue mission in May 1992. But a tentacle could wrap around the satellite in a firm grip. And the Department of Energy has huge underground tanks—at Hanford, Washington, among other places—full of all kinds of nasty, unidentified toxins that have accumulated over the years. Robot snakes could go in and sample these materials and perhaps aid in their cleanup. More important, we are currently working on snakelike robots for medical applications.

Snakelike robots have been around for about 25 years, though little real progress has been made with them. The "father" of snake robots is Shigeo Hirosi, of the Tokyo Institute of Technology, who built the world's first serious working one in the mid-1970s. While impressive in their audacity—these robots were a great leap beyond what anyone else had done—his robots couldn't maneuver with any accuracy or speed. Hirosi is still at it, and his robots are very well engineered and work reasonably nicely, but most of the other researchers' prototypes are collecting dust. Hyperredundant robots have so many degrees of freedom that traditional algorithms for coordinating their motions are useless. The algorithms just take too long to compute. Some compute an entire gesture and then execute it, which means the robot sits and thinks for perhaps half an hour, then makes one rapid movement, and then goes catatonic again while it figures out its next move. The others calculate motion incrementally, so that the snake moves continually, but in glacial slow motion. Thus, no one has been able to figure out how to make robot snakes do anything useful. This daunting complexity has kept hyperredundant robots confined to the laboratory, and has discouraged most people from working in the robotic-snake field.

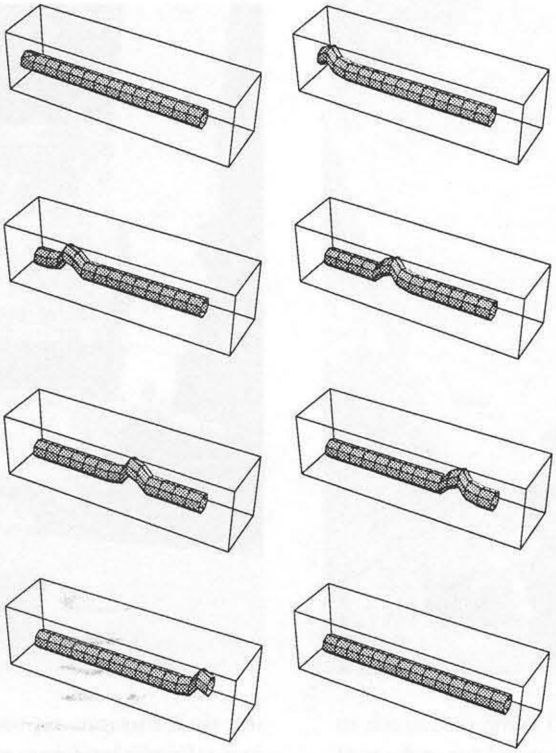
One of my long-term goals is to make these robots feasible—I believe they have tremendous promise in a number of areas that current robot technology can't approach. Our initial focus was to develop a set of building blocks. These building blocks, called primitives, are simple low-level operations, such as extending the snake to its full length in order for it to begin grasping something. Primitives are more complex operations than moving a single joint, but less complex than executing a complete task such as grappling a satellite. We had to develop the analytical tools needed to create the algorithms that would execute these primitives realistically.

For example, consider how such machines might get around. In Greg's thesis work, we developed a mathematical framework for precisely and efficiently controlling these complicated machines' locomotion. In nature, inchworms, earthworms, slugs, and snakes (which are analogous to our robots) all use different kinds of waves for locomotion. Inchworms use a form of movement analogous to a standing wave that resonates through the body. Earthworms move with an extensional wave that travels down the length of the body. Slugs use a vertical traveling wave, like when you snap a rope lying on the ground. And snakes use several motions. While stalking, they often use a "creeping gait" in which the abdominal muscles and scales move in a rhythmic wavelike pattern. Greg has found that there's an underlying set of equations that generates every kind of wave-like locomotion seen in nature, as well as some we haven't seen yet. Even the sidewinder, which moves forward by looping its body sideways, can be described exactly. But that's just half the battle. Once we know the mathematics of how the snake as a whole moves, we still have to determine the mathematics of how and when each muscle or actuator in the snake must move. This is the level of detail required to implement explicit algorithms to control robotic motion.

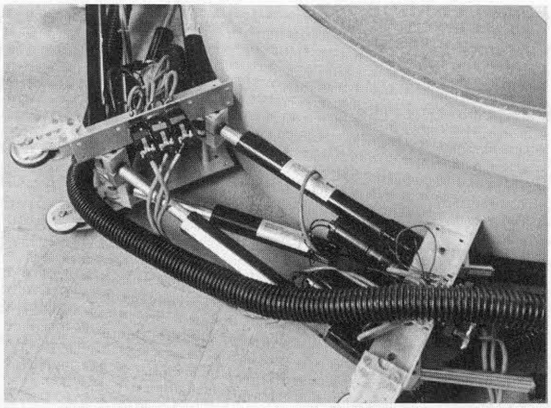
We also want our snake to operate in very tight corners, so Greg's been coming up with some very interesting obstacle-avoidance algorithms. Again, traditional algorithms, when applied to snakes, take so long they never compute. Greg's algorithms compute in real time—as fast as the machine can move—and can even handle moving obstacles.

In order to demonstrate our analyses, we've built a hyperredundant robot named Snakey. Snakey is properly called a "variable-geometry truss structure." Each of Snakey's segments is like the span of a bridge between two piers. The segment's trusses are actuators—pistons—that we can make longer and shorter so that Snakey as a whole will slither around. This design is quite different from what's found in nature. We've chosen it for its strength. Others have built snake robots with actual spinal columns, but those robots are too wimpy to do anything useful.

From basic functions like locomotion and obstacle avoidance, we can build more complex functions. Snakey can grasp and manipulate things like a tentacle would. Imagine you've grabbed a satellite, and you want to reorient it so you can work on a particular part of it. The obstacle-avoidance methodology curves the tentacle around the satellite to embrace it, and the locomotion algorithms "walk" along its surface. This reorients the satellite, turning it beneath the tentacle a little bit at a time. If we know the object's geometry, the algorithm automatically figures out how to move all 30 of Snakey's actuators. We can even break Snakey into two tentacles of five segments each, and have them

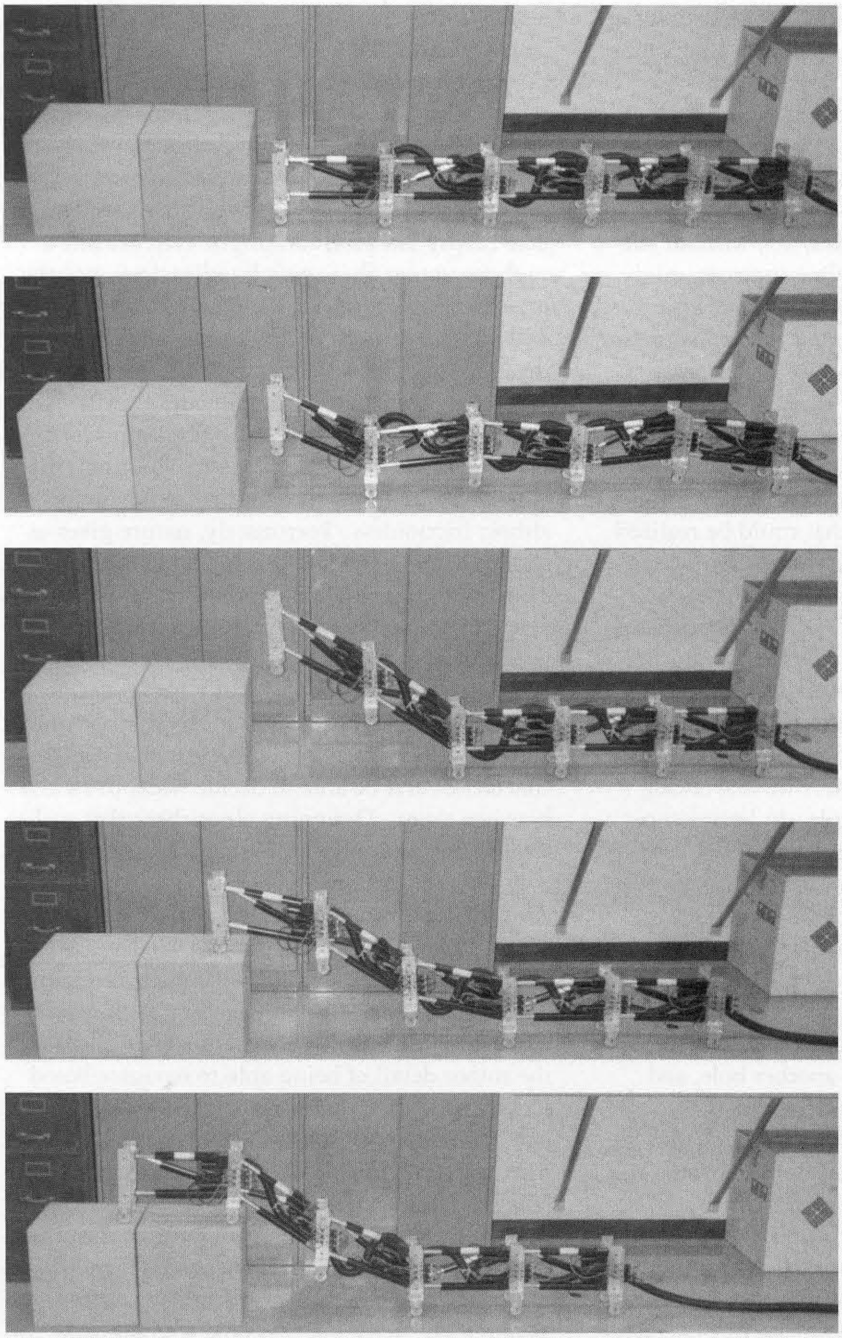


Top: The slug mode of traveling-wave locomotion.



Middle: Each of Snakey's ten segments consists of three actuators, or pistons—two parallel and one diagonal per segment—that, acting together, move the segment in two dimensions.

Bottom: (From left) I-Ming Chen, Brett Slatkin, Ostrowski, Choset, Chirikjian.



Snakey can avoid obstacles by going around or (in this case) over them. Only five segments were used in this demonstration in order to keep Snakey's length comfortably less than the width of the lab.

cooperate to manipulate an object.

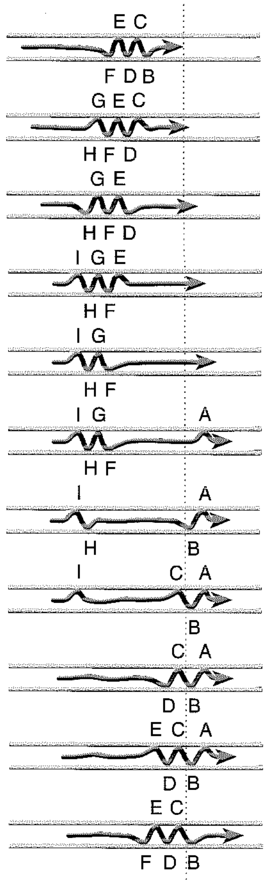
Snakey is so dexterous that it can curl up on itself. Traditional robots can't. It's also so big that it doesn't always all fit in our lab. It can actually stretch (unlike real snakes) from its minimum length of about 12 feet out to about 18 feet. Then we have to open the door and let it stick out in the hall, where it bumps against the far wall. It can do all the different kinds of snake locomotion. It can do the earthworm movement. And when we turn it on its side, so that it moves in the vertical plane, then we can do the inchworm standing-wave locomotion, using wheels that only turn in one direction so that the machine will move forward. Furthermore, we can make these complicated motions fairly precisely, so that Snakey goes where we want it to go.

Right now we have to tell Snakey where the obstacles are. It doesn't have enough on-board sensors to automatically detect objects and move based on that data only. This is one focus of our current research (with graduate student Howie Choset and visiting scientist Nobuaki Takanashi from NEC corporation, Japan). We hope to be able to demonstrate motion based strictly on sensor input in the near future.

We're now going beyond demonstrating algorithms to thinking about making these snakelike robots do practical things. We are currently helping JPL build a small snake-robot that will be carried around on the end of a larger traditional robot arm. This robot will peer around inside the complicated structure of the future space station to inspect for cracks, micrometeoroid damage, and so on. And, depending on up-coming budget constraints, we hope to start a project next year with JPL and the Kennedy Space Center to make a snakelike robot to service the space shuttle on the launch pad. In order to flip switches, remove dust covers, and rearrange thermal blankets in a fully loaded space shuttle cargo bay, humans must currently dangle from a gantrylike device called the Payload Ground Handling Mechanism (PGHM). Because each payload combination is different, NASA has to build special fixtures, at a cost of a million dollars per mission, to enable people to reach into the packed shuttle bay from the PGHM. A dexterous snakelike robot would be able to thread itself through the labyrinthine bay to do these simple tasks without special handholds or safety harnesses. Caltech will be developing all the algorithms for this robot.

But I'm most excited about applying these robots to medicine. Of the 21 million surgeries performed yearly in the U.S., surgeons have estimated that about 8 million could be done with minimally invasive techniques, i.e., without slitting somebody wide open. One common minimally invasive procedure is arthroscopic knee surgery. Presently, only about a million surgeries per year are performed this way. The average traditional surgery puts you in the hospital for about eight days. The equivalent minimally invasive surgery requires, on average, a four-day hospital stay. The potential savings in hospital costs and lost productivity that could be realized by the complete adoption of minimally invasive surgery amounts to about three-fourths of one percent of the GNP. There are two main ways in which the number of minimally invasive surgeries will increase. Training more surgeons to do them will put us at around three million. But getting to eight million will require a big leap in technology, and that's where we're working.

A snake in a confined space, such as a pipe, naturally adopts the "concertina" mode of locomotion. The letters mark parts of the snake's body that are held motionless, pressed against the pipe, while the rest of the body is thrust forward or drawn in to be held against the pipe in turn.

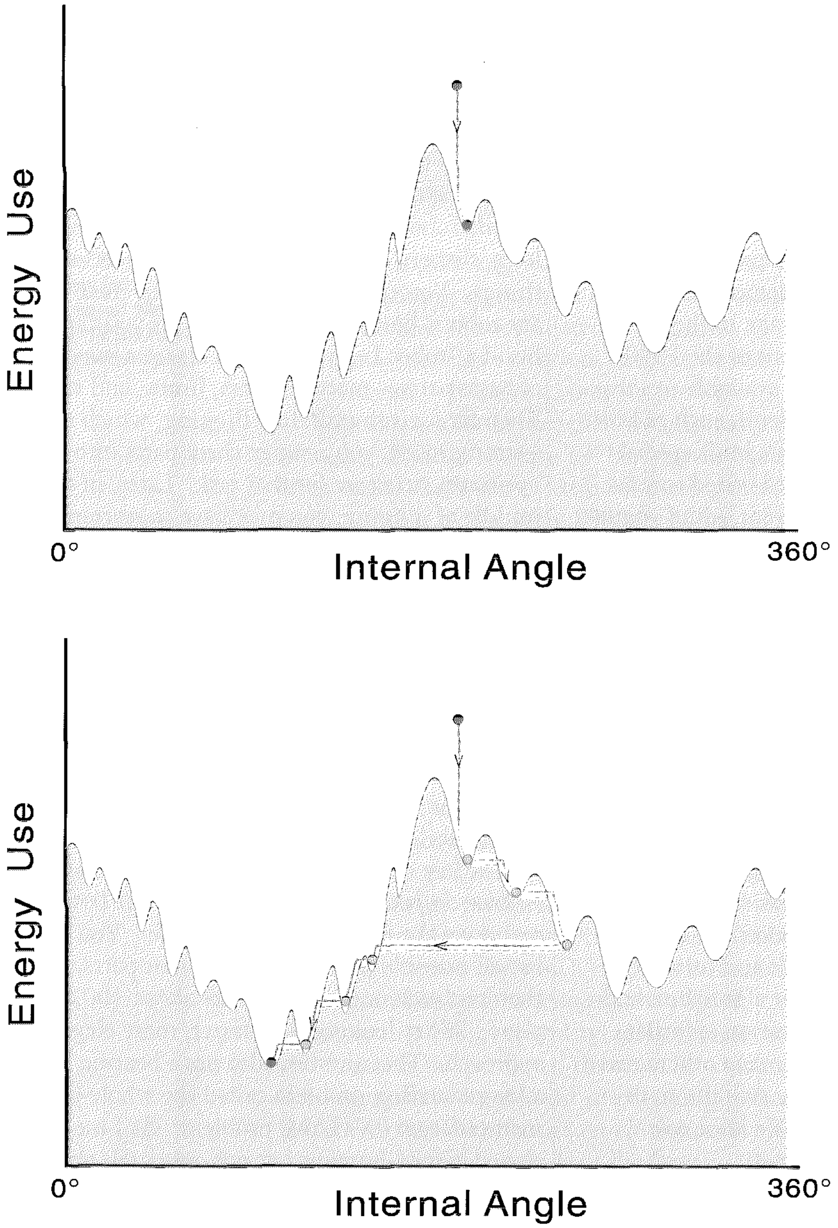


Let me give you an example. In laparoscopic gall-bladder removal, they poke a hole in your abdomen, stick in a hose, and fill you up with carbon dioxide until you look like a balloon. With the abdominal cavity inflated, the organs aren't all squashed against each other, so the doctors have more maneuvering room, and they're better able to see what they're doing. Then they insert a TV camera through another hole, and they come in with long, scissorlike tools through adjacent holes, snip your gall bladder off, and drag it out through one of the holes. Instead of a six-inch scar, and four weeks out of work, you have a bunch of eight-millimeter holes, and you are back to work in about four days. But there are lots of places in the human body that surgeons can't get to with current laparoscopic technology. In order to get to these deeper parts, we need long, thin, articulated, actively controlled devices—e.g., small hyperredundant robots.

I'm working with Dr. Warren Grundfest, of Cedars-Sinai Medical Center, and one of my graduate students, Brett Slatkin, to develop surgical robot technology. Right now we're working on a gastrointestinal robot—a robot "tapeworm" that crawls through your intestinal tract. Available endoscopes (which are long, semiflexible tubes containing a fiber-optic bundle for inspection and laser surgery) can't get to about 70–80 percent of your GI tract because both ends of it have sharp bends that are really hard (and very painful!) to get around. If you want to get beyond these bends, where the endoscope can't go, you have to start cutting. We're working on a tiny robot that can crawl through your intestine and negotiate those turns. It will not only reach the entire intestinal tract, but will also be less painful to the patient. It will have a TV camera to inspect the intestinal lining, possibly have a small arm to take biopsy samples, and might even act like a tugboat to tow fiber-optic bundles deep into the intestine for laser surgery. We think this device will ultimately eliminate about half a million invasive surgeries a year in the U.S. alone.

Our problem is that the human intestine is very hard to get a foothold in. It changes its diameter by a factor of four over very short lengths. It's elastic. It's squishy. And it's very slippery—almost frictionless. Fortunately, nature gives us some clues. There's a mode of snake locomotion, called the concertina mode, that snakes use in very constrained places. We've used our mathematical analyses to design algorithms to replicate that motion. But to successfully navigate the human intestine, the gastrointestinal robot must have mastered a number of gaits, including the concertina, and be able to decide when to switch between them. Designing algorithms that make that decision automatically is one incarnation of a very complex problem that I'll return to later. And there are other challenges to overcome before actual deployment of this robot. We have to shrink everything down to 8–10 millimeters in diameter, and pack in two TV cameras, a bunch of sensors, and a little robot arm. Then there's the minor detail of being able to navigate based strictly on sensory input rather than needing to have a detailed up-front model of the environment we're slithering through. These challenges will keep us busy for years to come.

I want to briefly summarize some of the other research efforts in my group that are related to the theme of dexterous and redundant robots. While there aren't many people who do snake ro-



A stylized graph of energy use versus internal angle for a robot arm. Top: A typical minimization algorithm drops a marble on the slope, where it rolls downhill and gets caught in the nearest valley. Bottom: TRUST's marble "tunnels" through the graph each time it gets stuck, until it reaches the lowest valley of all.

botics, there are a fair number who study legged locomotion—how robots can be made to walk. They'll tell you they're trying to develop machines that can maneuver over very rough terrain. That's a bunch of baloney. Like us, the people working in legged robotic locomotion do it for fun. The research in this field is divided into two areas, which are analogous to the conscious and unconscious motions that humans use to walk. There's an oscillatory, unconscious motion that you use when you're not thinking about walking—a natural rhythm that propels you forward and keeps you balanced. And there's a more irregular, planned motion, as when you're walking and suddenly realize you've forgotten your glass of milk, and you have to turn around and go back to get it. We're studying the rhythmic gaits. It's a real challenge to develop algorithms that mimic the smooth, stable, balanced stride that we use unconsciously.

Many algorithms have been proposed by other researchers, and many have been implemented in real walking robots. However, these efforts have often led to jerky or bouncy gaits. You wouldn't want to ride these machines, the way they lurch and sway. We've analyzed some of these algorithms to try to understand why they're so rough. As an example, there's a famous hopping algorithm by Mark Raibert (formerly of JPL, and now at MIT). When he tried to make his robots hop higher and higher, they didn't. They started limping. We found out why. For you physicists, it turns out to be a period-doubling bifurcation leading to chaotic motion. My grad student Jim Ostrowski is now trying to use our analytical tools to develop a more rigorous framework for designing running algorithms, like the one we've come up with for snakelike locomotion.

I now want to discuss a less glamorous concept, but one that's very important to me as a systems engineer. We often use a redundant robot's extra degrees of freedom to optimize some mechanical property—to maximize the force the robot exerts in some direction, or perhaps to minimize the energy the robot uses to move. It helps to think of this graphically. In the case of minimizing energy, the horizontal (x) axis represents the robot arm's internal angle, and the vertical (y) axis corresponds to the robot's energy use. We can then plot a curve that shows the energy use for a given internal arm angle. The lowest point on the curve corresponds to the internal arm configuration with lowest energy use. Typically, this curve has many little valleys, or local minima. We often want to find the global minimum—the lowest point anywhere—rather than getting stuck in one of the local minima. This global optimization problem has been extensively studied. Most algorithms make an initial guess about the best value of x, and then drop a marble, as it were, on the graph at that point and let the marble roll down the hill. The marble gets stuck in the nearest valley, which is almost always a local minimum. My grad student Bedri Cetin and I have collaborated with Jacob Barhen at JPL to develop a new algorithm that borrows an idea from quantum mechanics. When the marble gets stuck, it "tunnels" through the hillside to find a deeper valley, rolls down that hill, and continues to tunnel and roll, tunnel and roll until it finds the bottom. We call our algorithm TRUST, for Terminal Repeller Unconstrained Sub-energy Tunneling. It's substantially faster than well-known global optimization algorithms in standard benchmark tests.

More important, TRUST is analog, rather than digital, in nature. That is, the algorithm doesn't break the problem up into ones and zeros and solve it by discrete computation, but rather uses continuous mathematics. Consequently, we can build analog circuits to implement TRUST, and the answer emerges as a voltage in the circuit. This approach is reminiscent of the 1940s, when digital computers weren't available to solve complicated mathematical problems, such as how to lob an artillery shell. Engineers built special-purpose computers that were hard-wired to solve one particular problem. These went out of vogue in the 1960s as reprogrammable digital computers became powerful, because you didn't have to build a new computer for each new problem. But now we can easily and cheaply design and fabricate special-purpose analog computers on a chip, thanks to Carver Mead (BS '56, MS '57, PhD '60, Moore Professor of Computer Science), who has been pioneering this technology right here at Caltech. With proper algorithm design, these special-purpose chips can be much faster than digital computers, because the program is built right into the circuitry rather than encoded in a set of instructions that the computer has to read and then execute. Bedri has built and tested two of the three chips we need for our algorithm, and the third is under way. Optimization is important not only in robotics, but in many other areas of engineering, such as telephone switching, or the optimal design of bridges. We hope our algorithm will find widespread use.

Another robotic application of global optimization is grasping. Imagine your robot is trying to pick up a moon rock. The robot takes the rock's image with its TV camera, makes a mathematical representation of the rock's surface, and has to figure out where the best points are on that surface to grab it. Humans typically use an antipodal-point grasp, gripping the object at opposite points on its surface, like the north and south poles of the globe. Mathematically speaking, the object's surface normals—that is, lines drawn perpendicular to the surface—are antiparallel at the grasped points. This is one of the most stable grasps possible, and one that requires the least amount of friction, which is very important if you're grabbing an unknown object that might be slippery. So the problem is to find the antiparallel normals of a very complex shape automatically. By posing this as a global optimization problem and taking advantage of our algorithms, we can solve it quite rapidly.

And finally, my grad student I-Ming Chen is working on modular and self-configuring robotic systems. Suppose that you're a NASA engineer, and you want robots to build a base on the moon. You need strong robots to dig holes, nimble robots to put parts together, walking or otherwise mobile robots to transport things, and perhaps a long, slithery robot to peek into holes and inspect things. You can build and launch all these different robots, but that's expensive. Or you can build a "robot Lego set." You create a bunch of basic parts—motors, joints, limbs, and such. Then for a given task, like digging, which takes a strong robot, you arrange these parts into the optimum structure for that task. Later, in the middle of digging, you may have to rearrange the parts in a different way to do something else. This idea is potentially useful not only on the moon, but at any remote site where you can't afford to bring in a lot of hardware but you need to do a wide variety of things—such as in an underground toxic-waste tank, say. This is another wrinkle on the concept of redundancy, more like the engineering notion in which redundant subsystems improve the capability and reliability of the whole system.

The key question that I-Ming is working on is, how do you automatically figure out how to rearrange the parts for a specific task? You could list all possible combinations of your parts, and then test each one. That'll take about 100,000 years. We're looking for a faster, more elegant approach. This question also has a bearing on a long-standing problem called the whole-arm manipulation (WHAM) problem. As I mentioned before, humans can not only grab objects in their hand, but can use many other parts of their body for complex maneuvers and manipulations. We currently don't have a systematic way to enable robots to automatically plan such complex actions. Algorithmically, the WHAM problem has much in common with the modular

Even the Rose Parade is getting into the act. The General Motors float for 1992 featured a hyperredundant neck.

I will not predict the future of robotics. It's been predicted to death already and everybody has been wrong.

robot rearrangement problem. In both cases, the robot's planning algorithms must automatically determine how to coordinate the system's resources—the robot's various surfaces in the WHAM case—to solve a complex problem.

Robotics is a lot harder than we originally thought. Back in the late 1970s, Marvin Minsky, often called the father of artificial intelligence, predicted that by the early 1990s we'd have machines that were smarter than human beings and just as capable. He was clearly wrong. One of the reasons robotics is so hard is that truly intelligent, flexible robots are going to be complex systems beyond the capability of one person to create. The breakthrough will come when all of the building blocks developed by individual researchers, such as vision, tactile sensing, locomotion, manipulation, and machine intelligence, can be put into a unified framework—a Grand Unified Robotics Field Theory. We're working on some modest pieces of this unification—the WHAM problem is an aspect, as is the GI robot's deciding when to use which mode of locomotion.

Just as Minsky mispredicted the pace of advances in robotic capability, earlier researchers and futurists mispredicted the areas in which robotics may have its greatest impact. My personal opinion is that the robotics revolution, if it ever comes, will happen outside of the factory. Robots can be more than tools for increasing our productivity and standard of living. They may also play a role in solving some of our most vexing societal problems, such as the massive costs of medical care and toxic-waste cleanup. Developing robotic technology for these other applications may ultimately improve our ability to automate factories as well.

In conclusion, I will not predict the future of robotics. It's been predicted to death already and everybody has been wrong. Most of these predictions foundered because they assumed a lot of things would come together in a way that never materialized. Our research, however, is self-contained—we can move ahead regardless of what others do or don't do. But in the meantime, it seems that the more things change. . . . There was an item in the *New York Times* on August 13, announcing that the Matsushita Electric Industrial Company is developing a robot "that vacuums a room automatically and puts itself away." That's all it does, however. Maybe this one will actually work. □

Assistant Professor of Mechanical Engineering Joel W. Burdick came to Caltech in 1988 as the ink was still drying on his PhD in mechanical engineering from Stanford. Burdick earned his BS from Duke in 1981, and his MS from Stanford in 1982. Since his arrival, Burdick has won several awards reserved for young faculty of great promise, including the Office of Naval Research Young Investigator Award, the National Science Foundation Presidential Young Investigator Award, and the Richard P. Feynman–Hughes Fellowship. The latter recognizes faculty who combine innovative research with outstanding teaching. Burdick also won an ASCIT Excellence in Undergraduate Teaching Award in 1990. This article is adapted from a recent Watson Lecture.