Learning Navigation and Locomotion for Autonomous Ground Vehicles
Larry Jackel

How can a complex robotic vehicle get to where it needs to go, even if that means finding its way over big rocks, downed trees, or rubble? To do this, two major challenges must be solved: navigation and locomotion. That is, the vehicle needs to perceive its environment, and, it needs to send the right sequence of commands to its actuators so that it goes where it should. Today, I'll tell you about ideas we have that will give us the leap in performance that's needed so that robotic ground vehicles can both navigate and locomote through truly complex terrain.

First, a little background: The task of establishing the current state of the art in autonomous navigation was the goal of the DARPA PerceptOR program which has just concluded. In PerceptOR, we sponsored research for improving autonomous navigation, and, we measured the effectiveness of the developed technologies. Tests were conducted off-road, in places that were full of obstacles. The tests were very hard. Here's what we found: The vehicles did well in open environments, in woods, and even in tall grass. But when they had to traverse regions where there were thickets of large bushes or lots of fallen logs on the forest floor the vehicles often got into trouble.

In PerceptOR, a vehicle first senses its environment using either stereo cameras, or ladar. Both the stereo and ladar systems provide range data that then yields a 3-D view of the region in front of the vehicle. This 3-D view is then projected onto a 2-D cost map. Regions that are hard to cross, such as ditches or trees are colored pink. Safe regions are colored black. Using this map, a path planning algorithm decides where the vehicle should move next.

The vehicle can get into trouble when it fails to identify an obstacle, like a ditch, or a big fallen log. Another failure mode is when the perception system identifies objects that it thinks are lethal obstacles, but in reality are not serious obstacles at all --- shallow puddles or tall weeds. Researchers attempt to anticipate these circumstances by writing special software for the obstacles they think the vehicle will face. But there are so many different conditions that a vehicle can encounter, that relying on hand-tuning for every eventuality is not likely to get us the kind of performance we desire.

There is another basic limitation in these perception methods. The ladar and the stereo systems can only get range information for objects that are within about 100 feet of the vehicle. This means that the vehicles are effectively near-sighted. The result is that they can become trapped in cul-de-sacs. We can help them out of these kinds of jams by providing terrain data from some other source, such as detailed topo maps. But this fix is not always practical.

Thus, there are two basic problems in the current approach to autonomous navigation. Hand-tuning, and near-sighted range-finding.

Let me show you a system that doesn't suffer from these shortcomings. It's my dog Ben. He can run about 5 times faster than our robots. You can't catch him. Ben's eyes are only about 2 inches apart, so he can't get much distance information from stereo vision. Still, he doesn't get caught in the same kinds of cul-de-sacs that trip up our robots. I am not sure how Ben does his autonomous navigation, but I am absolutely sure he doesn't use ladar. Ben provides an existence proof that high performance navigation systems can be made just using visual cues.

Here's a photo that I took on a hike in Colorado. It's not hard for you to see the trail in this picture. You can do this with no stereo information. What you do have is an ability to understand images. To get a navigation system that really meets our needs I believe we need to break new ground in the task of image understanding. This doesn't mean that we need to label every object in a scene. It just means that we need to be able to analyze an image and find a path.

And, we must avoid hand-tuning and we must use our sensors to do more than range-finding.

Here's a possible approach to this challenge. Suppose we teleoperate a vehicle by looking at images transmitted from the cameras on the vehicle, and then send steering commands back to the vehicle. We can have the human drive the vehicle through all kinds of terrain. We keep a log of the video along with the associated human-generated steering commands. We can then use a machine learning method, called "learning from example" to teach a navigation system to emulate the human driver.

Computer scientists have used this kind of learning procedure before on many different kinds of pattern recognition problems, often with great success. While the vehicle navigation task requires a massive amount to be learned, the good news is that with advances in computing power, a learning problem this complex might now be tractable. In fact, in some recent experiments, a company called Net-Scale Technologies took a breadbox-size radio controlled monster truck and fitted it with a pair of small cameras. Using recorded videos and the associated commands from human teleoperators they trained a "convolutional neural net."

A convolutional neural net is not an ordinary multi-layer Perceptron. The network architecture is specifically designed to develop feature extractors for a task in machine vision. The architecture explicitly incorporates the 2-D nature of the task. Weights are grouped to form convolution kernels and the weights are **learned** from the training data.

After the neural net was trained, the vehicle was placed in a backyard in New Jersey and it had to find its way through an alley full of junk. In this video you see the images from the stereo cameras on the truck. Below these images, you can see a bar that represents the steering angle command that is sent to the steering actuator. The researcher who made this video says that the vehicle did a better job steering through these close quarters than he was able to do if he tried to control the vehicle himself.

We have not yet tackled the nasty terrain that the PerceptOR vehicles had to deal with. But achieving and then surpassing PerceptOR performance will be a goal of a new joint program in DARPA's IPTO and TTO. This new program will focus on learning navigation. Look for a new BAA on the IPTO and TTO websites later this month. In this new program learning algorithms will be developed on small, surrogate robotics vehicles and will then be ported to a vehicle we call "Spinner". (Show Spinner-short video)

Spinner was built by a team led by CMU in DARPA's Unmanned Ground Combat Vehicle Program that is just now concluding. This program's goal was to create robotic ground vehicles that have the physical ability to go over obstacles that would stop other vehicles in their weight class. Spinner has relatively simple locomotion control and as you can see in this video Spinner can readily deal with obstacles that would stop other vehicles. We think that leaned navigation, running on Spinner will give us a system with truly incredible autonomous mobility.

Of course there are many cases where we might need a more subtle approach, where we pick our way through obstacles instead of crunching them. As an example, take a look at Retiarius. Like Spinner, Retiarius was also built as part of DARPA's Unmanned Ground Combat Vehicle program.

Retiarius is about the size of a typical desk. It can get where it needs to go by careful use of its six wheels. As you can see, each wheel is at the end of an arm. Each arm can rotate through a full 360 degrees. Retiarius is like a giant six-legged bug. Retiarius is representative of a whole class of vehicles that require simultaneous, independent adjustment of numerous actuator controls. Other vehicles of this sort include slithering robotic snakes, and robotic walking machines. If we had proper control mechanisms we could have vehicles that squeeze through tunnels like ferrets or rats, or run through tight spaces on four legs like dogs or even on two legs like people.

Having so many controls gives these robots great potential but it makes them hard to operate. Here, Retiarius appears to be high on a ridge in the White Mountains of New Hampshire, climbing across boulders. This is the kind of behavior we need from an autonomous vehicle. But, I have to admit that I faked this image. I used Photoshop to paste an image of Retiarius onto one of my family vacation snapshots. I did it to show what we would like Retiarius to be able to do. Retiarius has the physical capability to climb like this, but the control system is too primitive to allow us to do sustained climbing at any acceptable speed.

As example of what Retiarius can do, here is a sequence of photos showing how a surrogate Retiarius with the same geometry as the real vehicle can climb over a concrete barrier while controlled by a human operator.

Thus, you can see that Retiarius has the physical ability to go over big obstacles. What it's lacking is the smarts. So we need to find a way for vehicles like Retiarius to locomote automatically. This is a really exciting problem, and it will likely take a new DARPA program to find solutions.

In nature, advanced animals learn to control their bodies. Our plan is to find ways to have robots **learn** control. Here's an illustration of what we want to do. We want to create a vehicle control function that takes as part of its input the current state of the vehicle including the position of all the arms, and the location and orientation of the chassis. As output, our vehicle control function has commands that tell each individual vehicle actuator what to do.

After each step, we evaluate its new position and configuration, and compare them to the desired position and then repeat the process, issuing a new set of commands. We keep doing this until we get to where we want to go. This kind of adaptive process works as long as each step usually moves us to our goal.

What’s inside the vehicle control function box? Basically the system consists of some sort of huge function approximater that takes all different combinations of the input state variables, including general information about where the vehicle should go, and generates the output commands. The approximater has lots of adjustable parameters, or weights that determine what the function actually does. Imagine a high degree polynomial in a high dimensional space. The weights are like the coefficients of the polynomial. Our task now reduces to tuning the weights so that the output commands are the right ones to move our vehicle in the direction we want to go.

To control complex vehicles like the ones we have been talking about we may need to set tens of thousands of weights. What method can we use to tune these weights? I think a promising method is “reinforcement learning.” Simply speaking, in reinforcement learning a weight is adjusted so that its value increased if it effective in moving the vehicle to the goal, and decreased if it is not effective.

Is there any reason to think that reinforcement learning can possibly work? Stanford University working with Lockheed Martin showed in simulation that a Retiarius like vehicle can use this method to learn how to cross a barrier. In this simulation the team placed the vehicle at the right of the barrier. They then used reinforcement learning to guide the vehicle guide the vehicle across the barrier. First we see results after some initial learning. The vehicle gets to the barrier, but then progress stops. With more learning we start to cross the barrier. Finally, with plenty of learning, we make our way across the barrier.

And finally, the results of learning how to cross the barrier at an oblique angle. I doubt that a person could hand program such a maneuver.

Let’s step back for minute and think about what we will have achieved if we can develop complex vehicles that have both learned navigation and learned locomotion. We will have robots that will be able to venture forth into new environments and find their way to a goal. They will be able to explore and then report back what they have found. These robots will adapt as they move along, becoming more capable with every move. They will learn to squeeze through narrow spaces and they will learn to move efficiently at speed.

Today, I have told you about challenging problems, that when solved, will enable autonomous robots to navigate and locomote through rough terrain. We have a very long way to go before we have mastered these challenges, but we think we see a path to their solution. We look for your participation to help make our dream a reality.