Chapter 16

Unconventional Vehicles

Solutions in this chapter:

- Creating Your Own SHRIMP
- Creating a Skier
- Creating Other Vehicles

Introduction

In the previous chapters, we discussed the two most common propulsion systems, wheels and legs, and looked into the many details regarding possible implementation schemes. Actually we didn't exhaust all the possibilities, so we are going to describe a few more robotic vehicles suited to very special tasks.

What mainly affects the mobility of a vehicle is the nature of the terrain that it has to move over. The scale of the robot, also, has a strong influence on the size of the obstacles it can overcome: A pebble two inches high is nothing for a wheel 20 inches in diameter, but it's insurmountable for a differential drive with wheels of only 3.5 inches (the largest contained in the MINDSTORMS kit). Scaling your robot up, however, is not always a practical option. In the specific case of LEGO, you're limited to the size of the available parts and their mechanical properties, and just like with real-life robots, they, too, often face constraints when it comes to weight and size.

The two robots of this chapter, a SHRIMP rover and a skier, are completely different in nature, but share the fact that they are designed for special surfaces: rough terrain for the first model, and snow for the second.

Creating Your Own SHRIMP

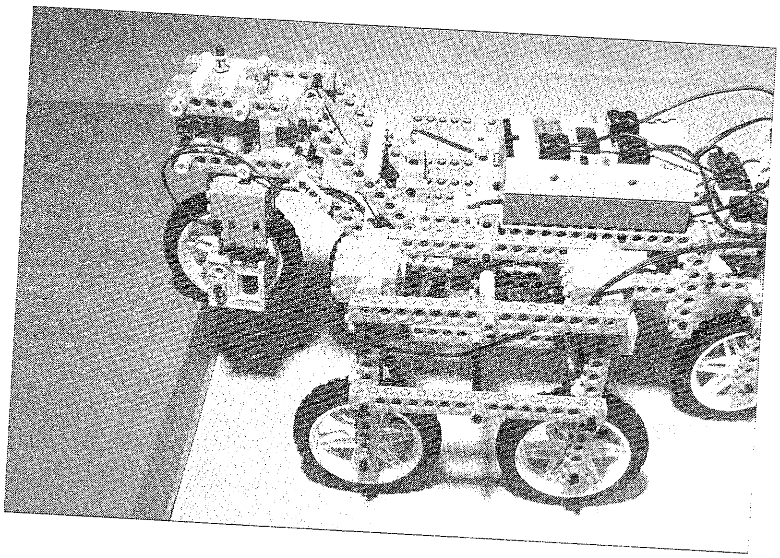
The original SHRIMP is a high-mobility wheeled rover designed by the Autonomous Systems Lab based in Lausanne, Switzerland. It features six wheels powered by independent motors: one front wheel mounted on an articulated fork; one rear wheel directly connected to the body; four wheels mounted on two lateral swinging bogies. (A *bogie* is a wheeled assembly that pivots one or more axles.) It performs amazingly well on many surfaces and against many kinds of obstacles. It's able to overcome obstacles as high as its wheels, even if they take the form of a stairway.

During summer 2000, we built our first LEGO version of SHRIMP, which had capabilities very similar to that of the original robot that inspired its design. The version described here is our first attempt at a turning SHRIMP; the first one, like the original, was not designed to turn.

Unfortunately, this project requires a lot of extra parts: seven motors, six gearboxes, ten universal joints, two polarity switches… not to mention a couple dozen of 1 x 16 beams. Later on in the chapter, we will give you some suggestions to help reduce the requirements, but the project remains rather demanding.

Let's start by looking at the SHRIMP in action to understand how it works. While the first wheel climbs the obstacle (Figure 16.1), the wheel assembly remains vertical, attached to the body with two parallel pairs of beams. This parallelogram geometry is the key to all of the SHRIMP abilities: The beams that connect the wheel assembly to the body convert the push from the other five wheels into vertical lift, while the front wheel itself follows the shape of the obstacle.

Figure 16.1 The SHRIMP Front Wheel Climbs a Step



When the first wheel is up, it's the bogies' turn to climb (Figure 16.2). They rely on the same principle: The bogie is a parallelogram attached to the body in the midpoints of its horizontal sides. When the bogie approaches the obstacle, those horizontal beams act like levers, with their fulcrums on the second wheel of the bogie. The load is applied in the midpoint of the levers, thus the first wheel has to lift only half of the weight applied to the bogie.

In Figure 16.3, the bogies are over the step, and pull up the rear wheel.

SHRIMP has an incredible ability to adapt to very complex terrain configurations. Some of its wheels may descend while others climb. Nevertheless, the body remains stable (Figure 16.4).

To turn properly, that is, with no skidding, the SHRIMP should rotate a minimum of four wheels. Do you remember the rule? If we extend imaginary lines through the axles of all the wheels, *they must meet at a single point*. This would be perfect, but very complex to build and control.

Figure 16.2 The First Wheels of the Bogies Climb the Step

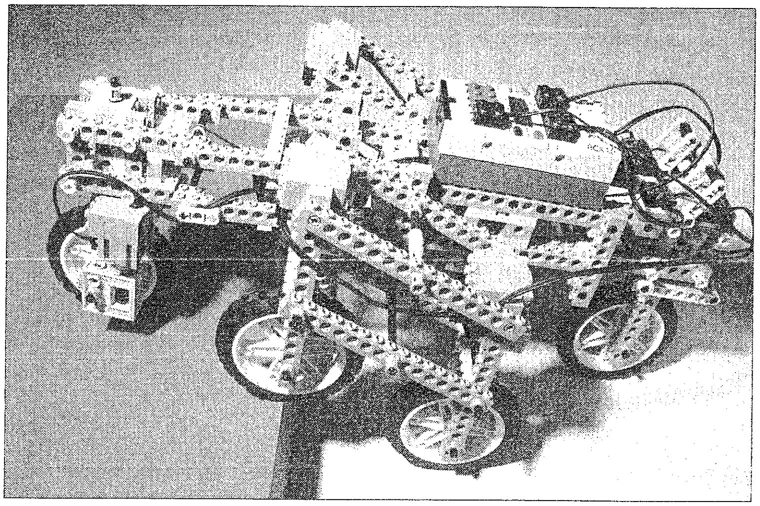


Figure 16.3 The Bogies Are Up and the Rear Wheel Climbs

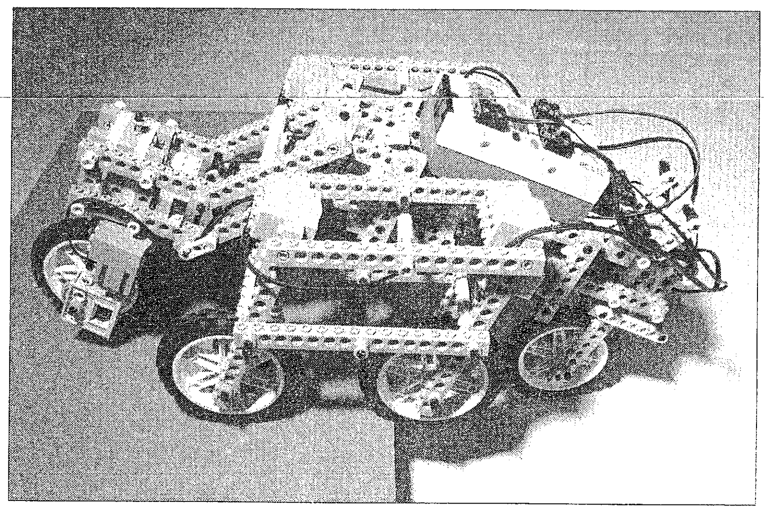
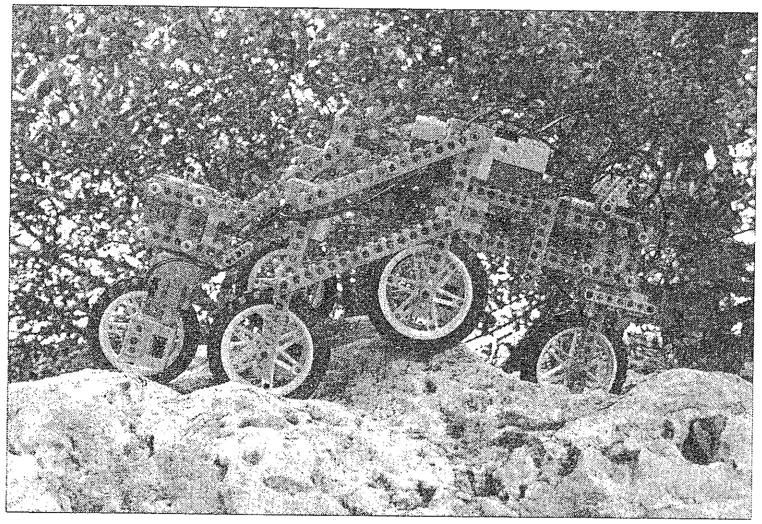


Figure 16.4 The SHRIMP Traversing a Rough Terrain



In our turning SHRIMP, we adopted a simplified scheme: the front and rear wheels turn, while the bogie wheels behave like a skid-steer. In other words, the inner ones stop while the outer ones continue to run. This is an approximate solution that introduces some slippage, but that in practice works very well on most terrains.

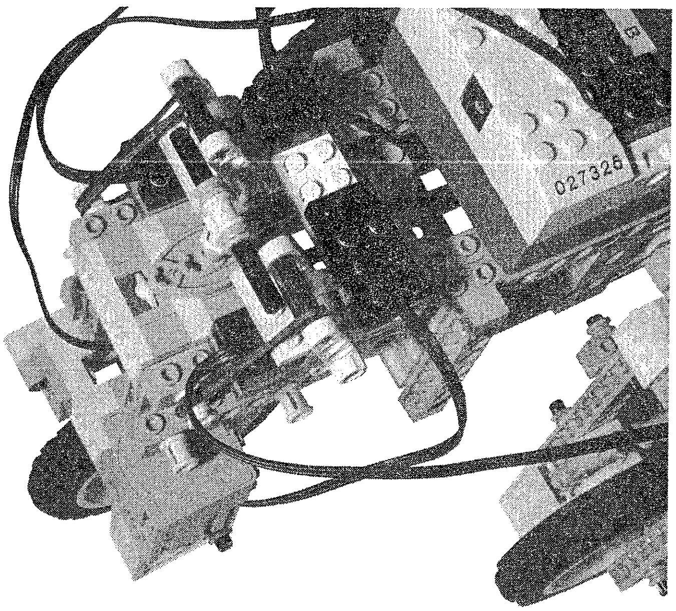
Though we reduced the complexity, there are still too many motors to control for a single RCX:

- The front and rear wheels form a group. Their motors need to always be powered when the robot is in motion.
- A motor turns the front and rear wheels.
- The wheels of the left bogie always run except when the robot turns left.
- The wheels of the right bogie always run except when the robot turns right.

We really wanted to avoid a second RCX, mostly so as not to add further weight to the robot. After long experimenting, and many useful tips from the LUGNET friends, we came out with a solution that saves not only the fourth motor port, but the third one also!

Our design requires two polarity switches, here used as simple on/off switches. The idea is that the steering system controls those switches too, and when the front and rear wheels turn, the inner bogie stops as a result. Figure 16.5 should help to explain the concept.

Figure 16.5 The SHRIMP Steering Control System



Two rubber bands keep the polarity switches gently pulled back in the on position. The pivoting axle of the rear wheel mounts a traverse axle, then, when turned, pushes the inner switch to the off position (the left switch controls the left bogie).

In the same picture, notice the touch sensor that detects the neutral position of the steering system, the only sensor used on this robot.

A single motor operates the entire steering system. It's placed in the bottom part of the body, and connects to the main steering axle through a pulley-belt-worm-24t geartrain (see Figure 16.6).

The main steer axle is a long joined axle that turns both the front and rear wheels. The rear steer assembly is rigidly attached to the body, while the front one forms one side of the parallelogram described previously. For this reason the steer axle requires two universal joints positioned precisely where the swinging beams connect to the body and to the steer assembly (see Figure 16.7).

16.5

Figure 16.6 Bottom View: The SHRIMP Steering Motor

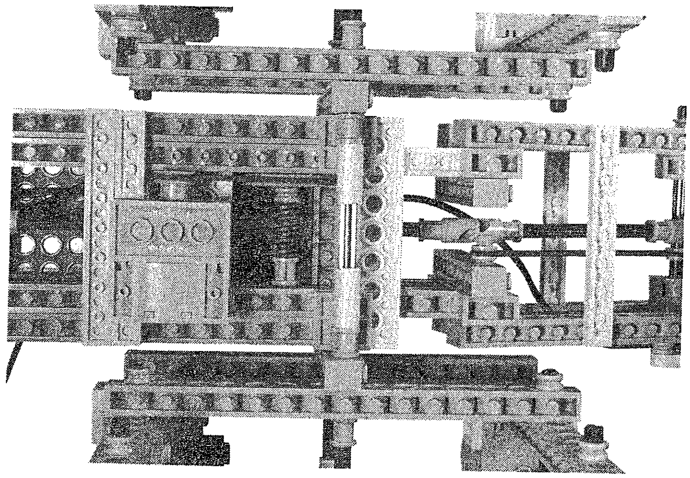
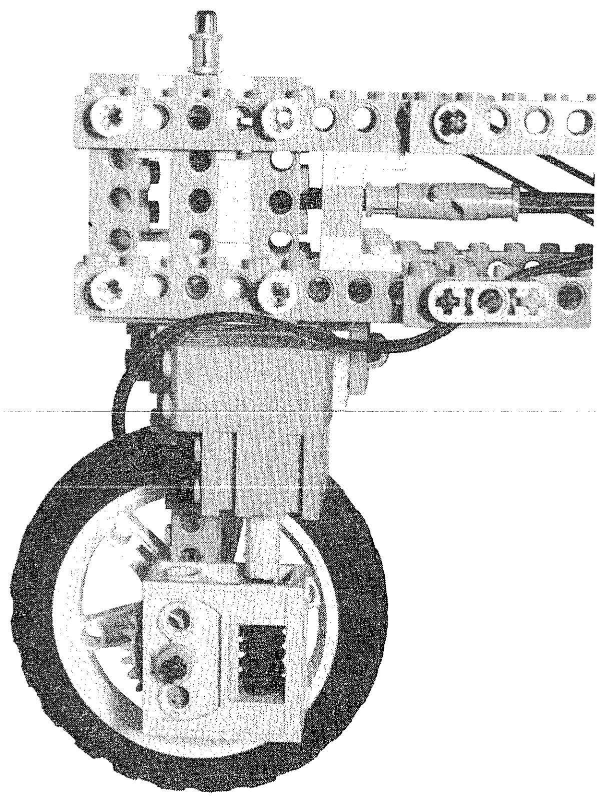
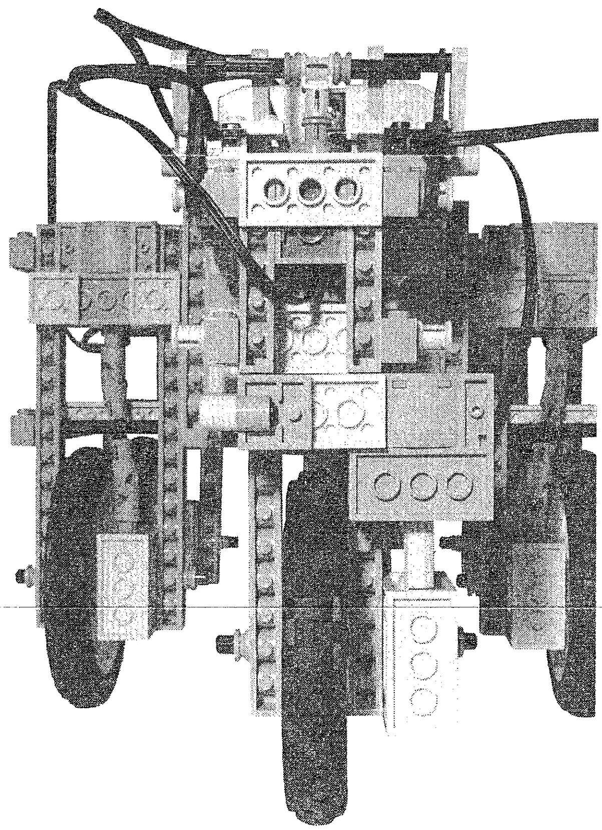


Figure 16.7 The SHRIMP Front Wheel, Side View



The long steer axle ends with bevel gear pairs on both sides, which transfer motion to the pivoting axles (see Figure 16.8).

Figure 16.8 The SHRIMP Rear Wheel, Rear View



The bogies mount four identical wheel groups, where a motor powers the wheel through a joined axle and a gearbox (see Figure 16.9).

In our first SHRIMP, the front and rear wheels were identical to the side ones, while in this version, the motor has been moved down to make the assembly more compact and leave the space above for the steering mechanism (Figure 16.10).

Figure 16.9 Close-Up of a SHRIMP Side Wheel

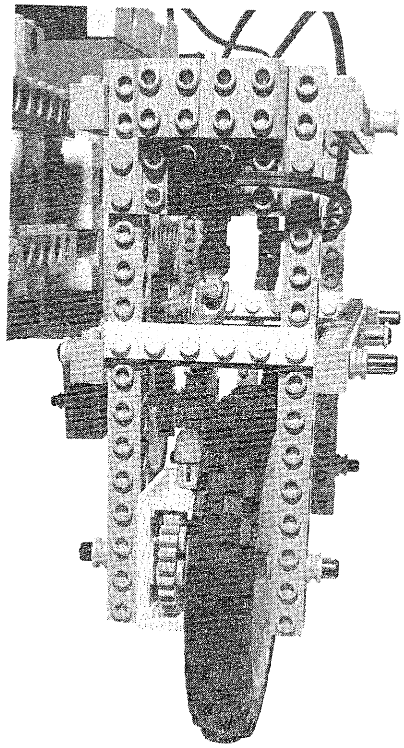
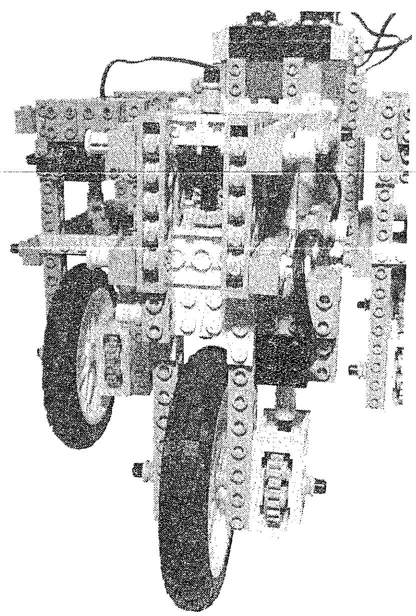
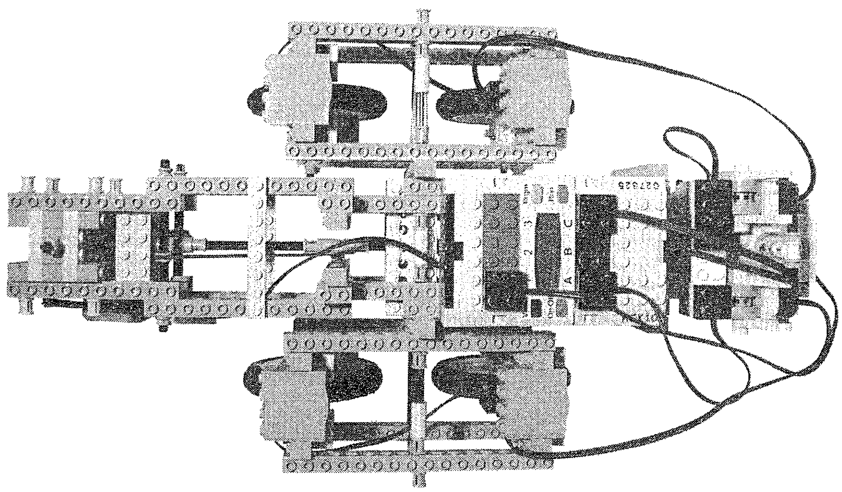


Figure 16.10 The SHRIMP Front Wheel



The RCX stays on the top, just behind the point where the bogies connect to the body (see Figure 16.11).

Figure 16.11 The SHRIMP Top View



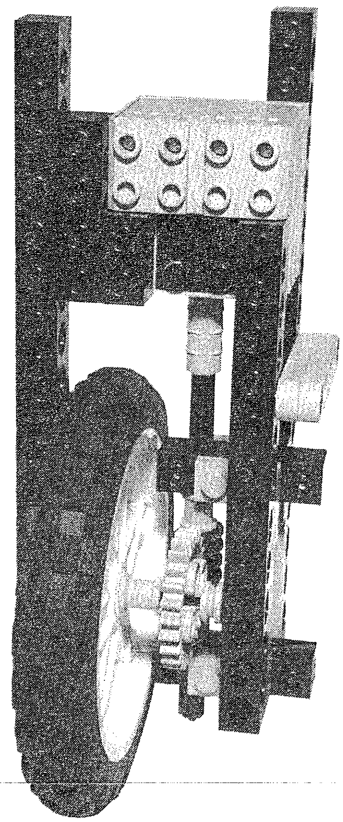
Building a SHRIMP

If you want to create your own SHRIMP, but don't have all the parts we used, the LEGO inventory offers many possible substitutes:

- **Gearboxes** A gearbox is a convenient way to match a worm gear to a 24t. But as you've seen in this book, there are many other assembly solutions, they're just a bit less compact.
- **Universal joints** Those that power the wheels are easily avoidable with a different construction. For example, our setup for the front wheel doesn't use them, and you can replicate this for all the wheels. In Figure 16.12, we show a wheel with no universal joints and no gearbox.
- **Polarity switches** You can use the free port of the RCX to control one bogie, and connect the other to the same port that drives the front and rear wheels. No polarity switches are needed for this configuration, but your SHRIMP would only turn in one direction.
- **Motors** A nonsteering SHRIMP also saves one motor. We didn't try, but we are sure it's possible to use a single motor instead of two in each bogie. In theory, with a lot of gearing, you can power your SHRIMP

with a single central motor: transport motion to the bogies through their supporting axles, and to the front and rear wheels with a system similar to the one we used for the steering setup. Such a SHRIMP will suffer from a lack of power and therefore climbing ability, due to the reduced number of motors and increased friction.

Figure 16.12 A Wheel Assembly



Creating a Skier

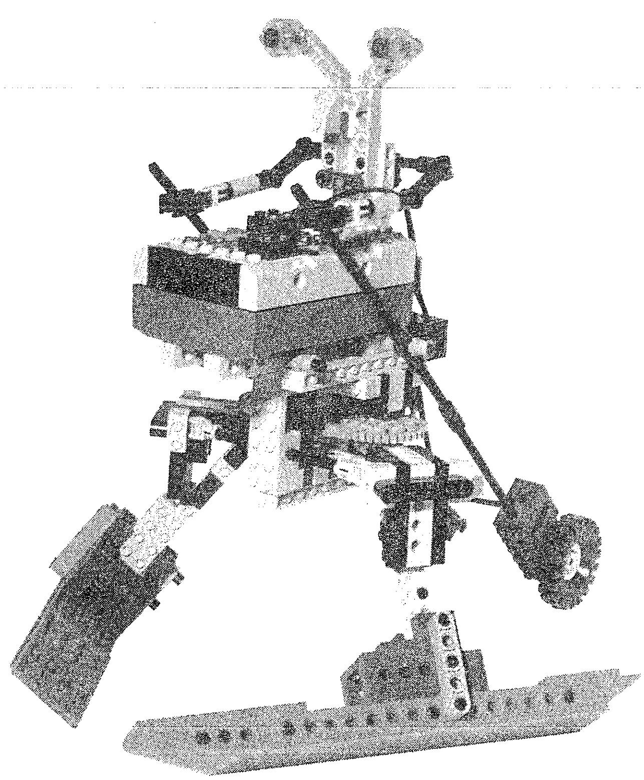
What we find most interesting in this project is not just the fact that this "skibot" robot can be used in the snow, but that without propulsion it descends snowy slopes like a true skier (well, almost!). It uses a technique known as *snowplowing*, due to the V-shape of the skis, often used by human skiers. In snowplowing of the human variety, the skier angles his toes inward in order to put the tips of his skis together and simultaneously dig the inside edges of the skis into the snow. To reduce speed, the skier pushes out the tails of his skis, increasing their angle to make a wider V; to increase speed, the skier draws the tails nearer, making the V narrower.

Our robotic skier is based on the same principle. It mounts onto a pair of skis, and while descending a slope, it varies the angle of the skis to increase or decrease the resistance and maintain a roughly constant speed. It uses only one motor and one sensor: The motor is on the back and operates the legs, making them more or less convergent, thus keeping the speed in the desired range; the sensor is a rotation, attached to a wheel at the end of the left ski pole, and serves to measure the speed.

Another interesting feature of this robot is that its geometry and the position of its center of gravity make it always point toward the direction of the maximum slope. There's no need to shift weight to control direction, this happens automatically because the motion along the longitudinal axis of the robot is the one that offers the lower resistance.

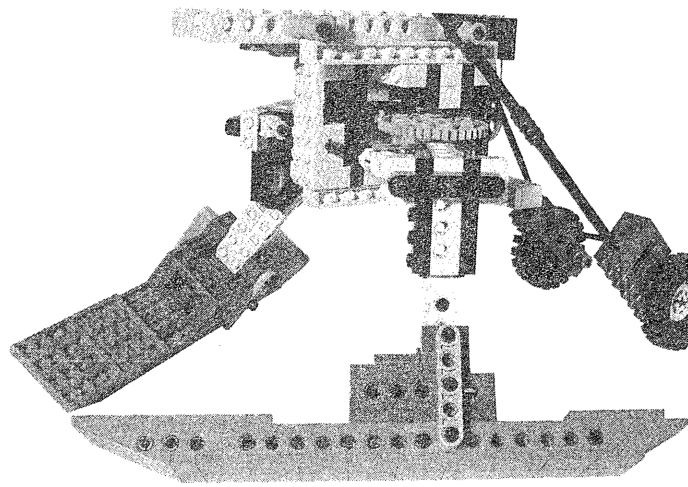
A general view of our skier can be seen in Figure 16.13.

Figure 16.13 The Skier



To build the skis you need some extra beams and, more important, many tiles (see Figure 16.14). We used 36 2 x 2 and two 1 x 4 tiles (available as spare parts at the LEGO Online Shop). If you are open to employing non-LEGO parts, you can build the skis from other materials, like strips of plastic available in hobby shops.

Figure 16.14 Side View of the Skier



The legs are not vertical, but rather are inclined outward. This is very important. For a human skier, it's what keeps the skis resting on their inner edges and producing the necessary resistance to gravity when in the convergent (snowplow) position (see Figure 16.15).

We achieved this effect by using some hinges and forming the legs from the diagonal of a perfect right triangle. If you don't have hinges, other possible solutions exist, like the one shown in Figure 16.16.

Each leg is rigidly attached to a 40t gear. The pictures don't show them, but we used the extra crossed holes of the gears to place pin-axle connectors into. The two gears meet a worm gear in the middle of the assembly, which receive motion from the motor and controls the convergence of the legs (Figure 16.17).

Looking at the bottom, you notice a longitudinal beam that locks the structures, and a transverse axle and beam that serve as boundaries to the movement of the legs (Figure 16.18). There are no limit switches. If the RCX tries to close or open the legs more than what's allowed, the belt will slip on the pulley.

Figure 16.15 Front View of the Skier

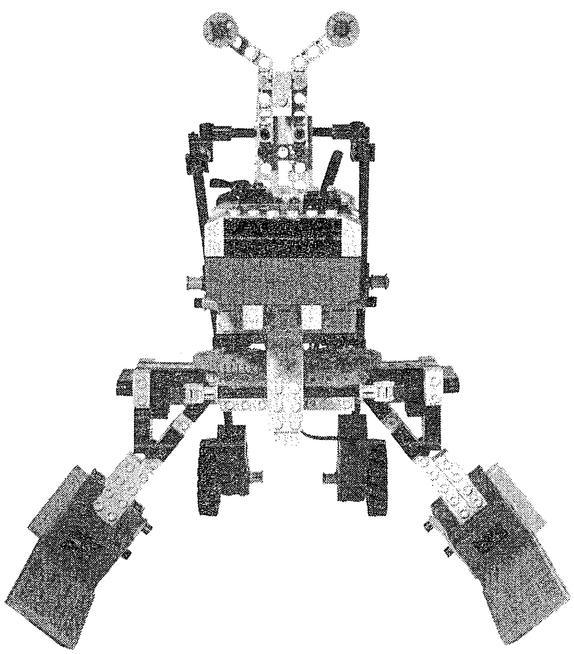


Figure 16.16 Alternative Leg Geometry

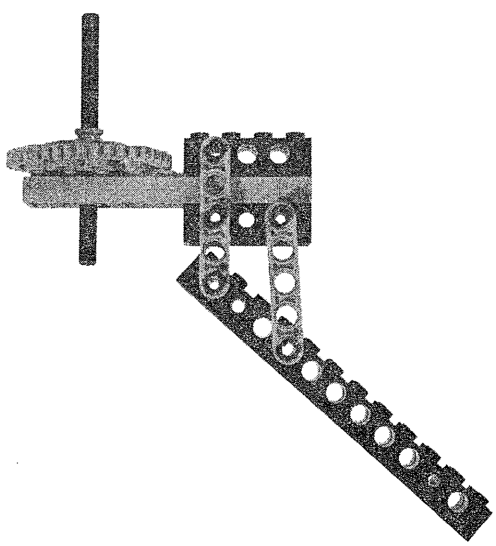


Figure 16.17 Rear View of the Skier

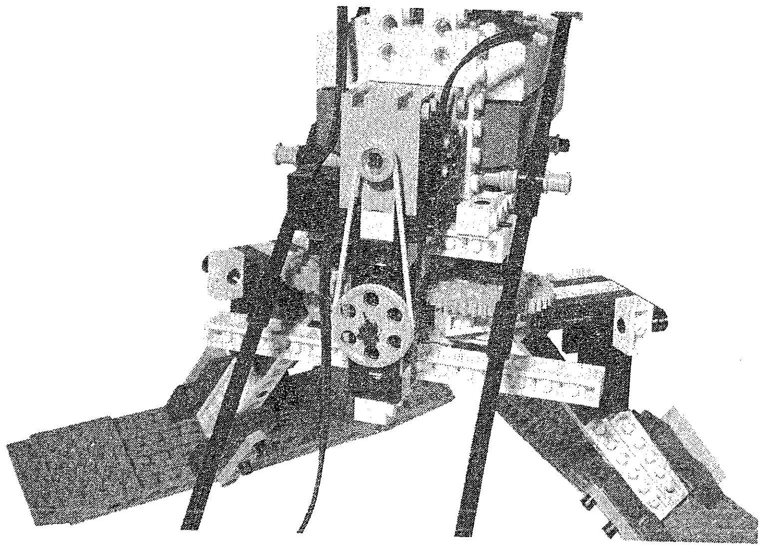
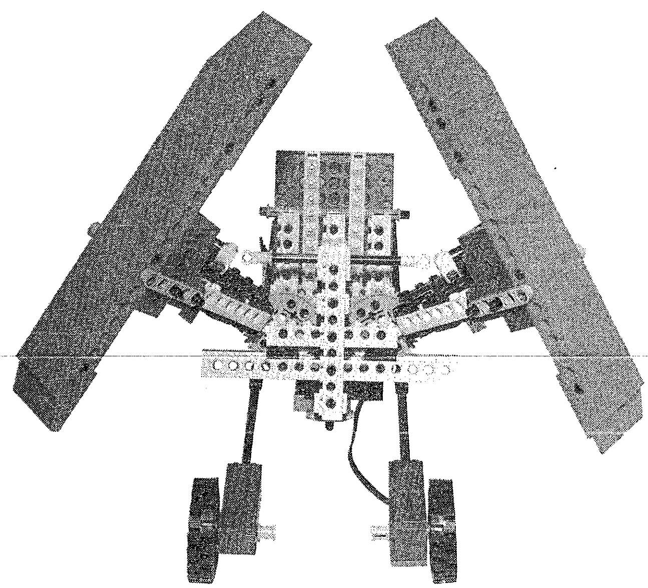
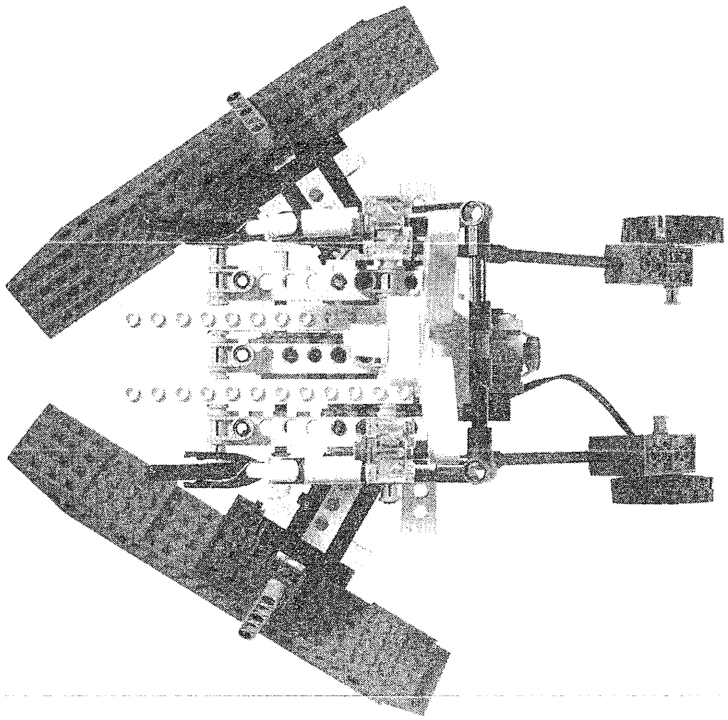


Figure 16.18 Bottom View of the Skier



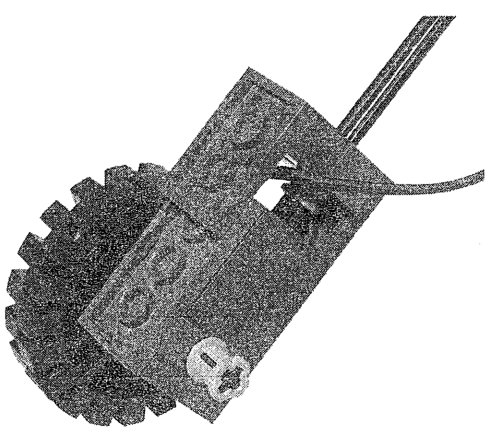
What's peculiar about this robot is that it has beams with all the possible orientations. The skis are studs down, the legs studs front, the body partly studs front, partly studs right, and partly studs up (see Figure 16.19).

Figure 16.19 Skier Top View (RCX Removed)



There's not much to say about the ski poles. The right one is just decorative, placed there for the sake of symmetry. The left one, meanwhile, incorporates a rotation sensor that's directly connected to a wheel (see Figure 16.20).

Figure 16.20 Detail of the Left Ski Pole



Programming this robot is so simple that the topic deserves only a few words. Inside the main program cycle, test the increment in the rotation sensor counts: If this falls in the range that represents the desired speed you chose, switch the motor off; if it's above the range, start the motor in the direction that closes the skis, and vice versa if it's below the range. This will speed up or slow down the skibot as needed.

If you test your skibot in the snow, try to find or create well-packed powder, like those normally found on ski runs. It's not able to ski on black diamond runs or in loose powder!

Inventing...

How Did We Test the Skibot without Snow?

Suppose you want to build this robot and test it, but currently there's no snow outside. This book was written during the hot Italian summer, so we faced that very question ourselves.

Well, we admit we had thought of going to the Alps to visit a glacier where some of the winter snow had survived. Unfortunately, we had to settle for a less expensive and time-consuming solution. We placed four large ice-packs, the kind used in portable camping refrigerators, on an inclined board. Then we covered them with frost taken from the freezer, gently pressing it down. It didn't make for a particularly long run, but it was enough to verify that our robot actually skied. A positive side effect of this experiment was that Mario's wife, coming back home, exclaimed: "You defrosted the freezer—bravo!"

What else could you improvise with to simulate a snowy run? Prepare an inclined plane (a table top would do the trick), and cover it with some fabric like a blanket, a sheet, a tablecloth or anything else you have handy. You have to adjust the slope depending on the kind of fabric you use, and the top will likely need to be steeper that the snowy slopes your robot can actually descend, because real snow produces much less friction.

Creating Other Vehicles

Here we present you with a list of suggestions for possible projects, their common denominator being that all of them are, at least in part, vehicles. They're meant just as starting points.

Elevator

We briefly discussed an elevator project in Chapter 4, in which we explained that a single touch sensor, placed in the elevator car, can control the positioning at an unlimited number of floors. We said also that a second touch sensor could serve the purpose of addressing the car to the proper level, using a simple system where the RCX counts the number of clicks on the sensor.

A variation on this theme is the car park elevator, where you emulate one of those automatic storing systems. It would be nice if your robotic parking could decode a sort of ticket, maybe using colors or shapes, so it can return the corresponding vehicle.

Train

The RCX is almost a natural extension to the LEGO 9v electric train system. They share the same voltage and the same connectors, and in fact many train fans currently use one or more RCXs to introduce automation in their layouts. This topic is so vast it would require a dedicated book, so we will provide only some basic tips.

There are two basic approaches to control a train with the RCX: a) the RCX is on the train; b) it supplies power to the tracks.

In case a), you put the RCX in the locomotive or in one car and connect an out port to the train motor. The train motor is also wired to the wheels, which normally draw current from the tracks, so it will happen that your RCX will supply power to the tracks, too. Nothing bad happens, but DON'T connect the train speed regulator to the tracks as well; you could damage your RCX.

If you are the kind of person who likes customizing things, you can open the train motor and interrupt the connection to the wheels, so your train will be totally independent from external sources. This way you can run many RCX-controlled trains on the same track. You can create some external references to read with sensors, so your train knows when to slow down or stop, or place a proximity sensor on the locomotive to avoid collisions.

In the second approach, b), you substitute the train speed regulator with the RCX and power the tracks from one out port. You can control three independent tracks or segments with a single RCX, and use the input ports, for example, to detect the arrival of the train at the station.

There are many other devices you can automate in your layout: switching points, level crossings, decouplers, semaphores, swing or draw bridges, and so on.

Cable Railway or Gondola

In a real cable railway, there are two pairs of cables: two supporting ones and two pulling ones. The supporting cables are more or less rigidly attached to the lower and upper stations, and work as railways for the cabs that have their pulleys running over them. The pulling cables transfer motion to the cabs: one cable goes from the first cab to the second across the upper station, while the second connects the two across the lower station.

You can place the motor either in the upper or the lower station. If you use the upper station, you can avoid the second pulling cable. Use touch sensors to control when the cab enters the station so you can stop the motor, possibly after a short slow down.

Boat

LEGO inventory includes different kinds of propellers, so one might wonder if it's possible to make a robotic boat. It is indeed possible, but it's not easy to provide the necessary flotation lift using only LEGO parts. The two solutions that come to our mind require uncommon parts, either single mould boats coming from the System product line, or a bunch of TECHNIC air tanks. The idea is to build a sort of catamaran, with both hulls made of two System boats or a row of air tanks.

A simple, cheap, and handy non-LEGO alternative for the hulls is to use common soft drink plastic bottles and attach them to long beams with rubber bands or duct tape.

The RCX will stay on the deck, together with the motor that drives the propeller and the other that controls direction. You can place bumpers on the front to make your robotic boat change direction when hitting an obstacle.

WARNING

The RCX, the motors and most other electronic components don't like water at all. While distilled water is a good insulator, common tap water, or water from the sea, lakes, or pools, conducts electricity extremely well and will damage your devices. Take every precaution not to soak or submerge them.

To minimize the risk of damages in case of an accidental bath, put the RCX into a small transparent plastic bag, with just the wires coming out from the opening, and seal the bag with a rubber band. Run your robot in a controlled environment with calm waters, like a pool with no people in it.

Sailing Tricycle

We are both fans of sailing, and in the wake of the great success of Luna Rossa, the Italian sailboat that won the Luis Vitton Cup 2000, we decided to build a robotic sailing tricycle or land yacht. We named it Duna Rossa (Red Dune) to mimic the original Luna Rossa (Red Moon).

Though building that tricycle has been a lot fun, we must admit that the performances were less than exciting. With a strong wind and a favorable slope... it moved!

See if you're able to do better: keep the structure as lightweight as possible, use a very large sail, and reinforce the mast with shrouds, forestays, and backstays (ropes).

The RCX controls two motors, one to steer the rudder and the other to operate the winch for the mainsail. You can detect the wind direction through a vane on the masthead connected to a rotation sensor. Monitor the position of the boom with a second rotation sensor to adjust it for the proper angle with the electric winch. Finally, you'll need a third sensor (a touch is enough) to control the position of the rudder.

You can program your robotic sailing tricycle for two basic behaviors: adjust the mainsail to keep with the desired course, or adjust the course to maintain a specific sailing point.

Summary

If there's a lesson you can draw from this chapter, it is that any problem in robotics requires a custom-tailored solution. After having long talked of standard mobility configurations, we described some situations where none of them applies.

You're not necessarily required to invent new solutions any time, more often you can just look around you, or on the net, and find something that helps you or points you in the right direction. SHRIMP was designed to move on bumpy terrains, but there are obviously other possible configurations. For example, the robotic vehicles designed for planetary exploration are a good source of inspiration, and there's a large quantity of documentation available in the public domain.

We must confess that our skier was born more for purposes of fun than to demonstrate some general principle. Nevertheless, this small and simple robot has its merits, helping us picture the wide range of applications robotics can be used for.