COS 116
The Computational Universe
Laboratory 3: Controlling the Robot

Last week you met the Scribbler robot and saw how to use pseudocode to control it. This week, you finally get to be in charge. By applying your knowledge of pseudocode, you'll explore the robot's capabilities and use it to investigate physical world phenomena.

If you get stuck at any point, feel free discuss with another student or a TA. However, you are not allowed to copy another student's program or answers.

You should hand in your lab report at the beginning of lecture on Tuesday, February 28. Include the following:

- **Printouts of the 9 numbered programs described below**
- **The 3 pictures that you draw with the Scribbler in Part I**
- **Responses to questions printed in bold (number them by Part/Experiment)**

You can print your programs directly from Scribbler Control Panel, or copy and paste them into a program like Microsoft Word. (Click "Copy as Text" from the Edit menu.)

Part I: Motion by Dead-reckoning

You can move the robot along a pre-determined path by specifying the motor speed and duration of each leg of the journey. This kind of navigation is called *dead reckoning*. The robot's basic movement commands all work this way (except that the Scribbler uses motor power to approximate speed.) One drawback of dead reckoning is that any error in the measurement of speed or direction accumulates over time, causing the robot's position to become less and less accurate.

In this section you'll use dead reckoning to make the Scribbler draw some simple figures. You'll also see how to apply basic geometry to plan the robot's course.

Before you begin, calibrate your robot's motors to ensure that it can drive straight. Your TA will provide instructions.

Experiment 1 — Draw a square

Instructions required: Forward, Spin Right, Pause, Do for *n* times

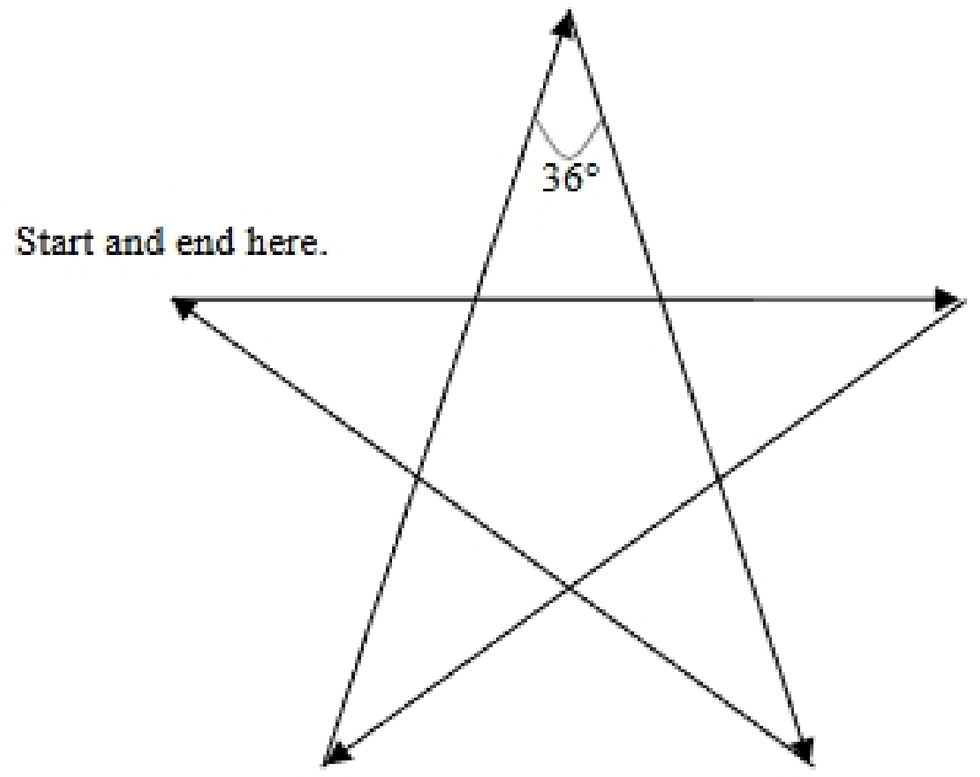
Program the Scribbler to draw a square with sides approximately 4 inches in length. Use trial and error to get the distances and angles right. Use a loop to avoid writing the same instructions four times.

Tips for accurate drawing: Tape the paper to your lab table so it doesn't slip. Always test the robot on the paper, since angles and distances may come out differently on other surfaces. Always insert a 0.2 second pause between motor commands to give the robot time to come to a stop.

Experiment 2 — Draw a star

Instructions required: Forward, Spin Right, Pause, Do for *n* times

Program the Scribbler to draw a five-point "pentagram" star, as show. Again, use a loop to avoid writing the same instructions multiple times.

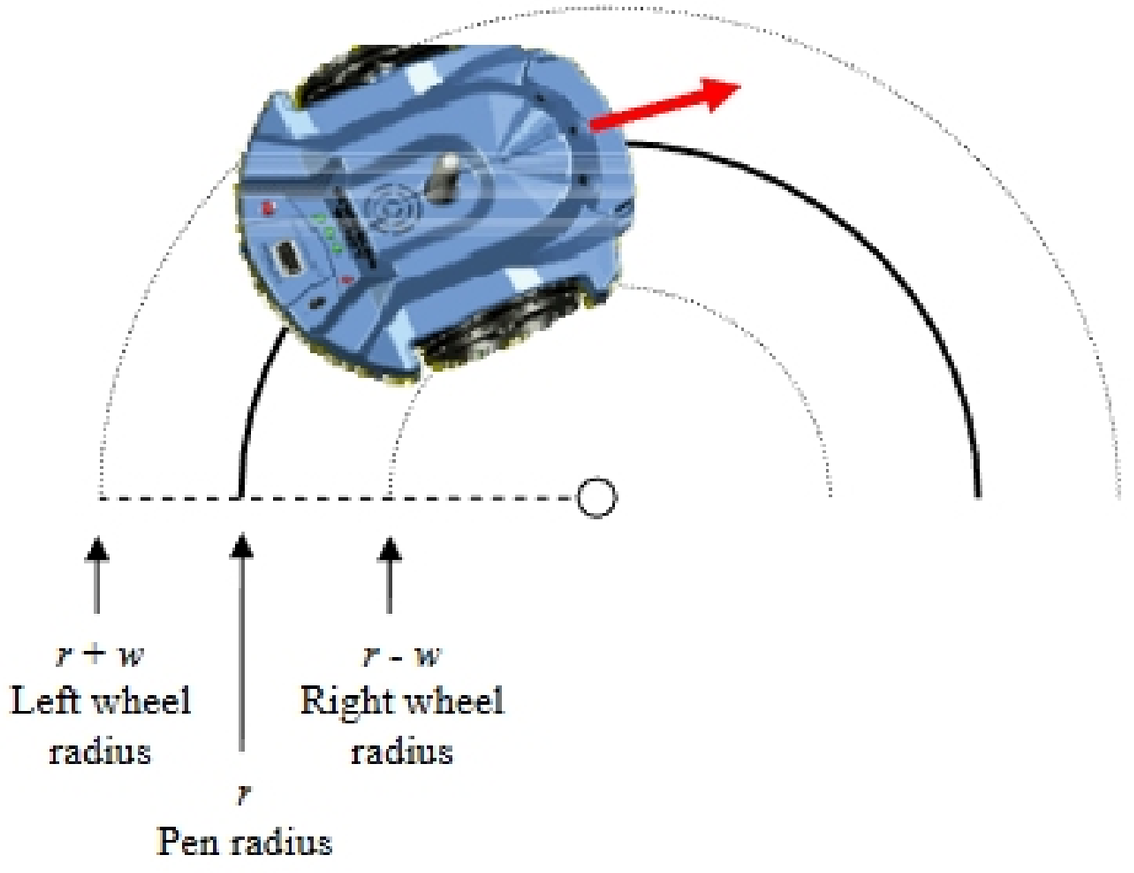


Each vertex has an interior angle of 36°. Use this fact and the time you ran the motors to make a 90° angle in Experiment 1 to estimate how long to run the motors for the angles of the star.

To draw more complicated pictures, you'll need to take advantage of the Scribbler's Advanced Motor Controls. These allow greater control over the speed of each wheel.

Click on the *Motor* command in Scribbler Control Panel, and check the box next to "Enable Advanced Motor Controls." Notice the additional sliders that become available. You can use these to set the speed of each motor from 100% (full power, forward) to -100% (full power, reverse). Try clicking each of the Basic Motor Control commands: Forward, Reverse, Left Turn, Right Turn, Left Spin, Right Spin. Note the Advanced Motor Control speed settings corresponding to each basic command.

One use of Advanced Motor Controls is to make the Scribbler turn along a path with a specific radius. Suppose you wanted to draw a circle with radius r, as shown at right. If the distance between the robot's wheel is $2w$, then in one full circle the left wheel will travel a distance of $2\pi(r + w)$, while the right wheel will travel a distance of $2\pi(r - w)$. Say it takes t seconds to draw the circle. Then: *left wheel speed* $= 2\pi(r + w) / t$, and *right wheel speed* $= 2\pi(r - w) / t$. This yields the proportion:



$$\frac{right\ speed}{left\ speed} = \frac{r-w}{r+w}$$

Measure *w* (half the distance between the wheels). Assume the left motor speed is 100% and solve the proportion to find the speed for the other motor that results in a curve with a 6 inch radius. What about a 3 inch radius? A 1 inch radius?

Experiment 3 — Draw a circle

Instructions required: Advanced motor controls

Program the Scribbler to draw a circle with a radius of approximately 5 inches. Use algebra to set the motor powers. Use trial and error to make sure the circle is complete without retracing too much of the line. Your program should use only a single instruction.

Part II: Motion with Sensor Feedback

Another way to direct a robot's motion is to use feedback from its sensors. In contrast to dead reckoning, sensor feedback allows the robot to limit the accumulation of positional error by continually updating its knowledge of its environment. In this section you'll use Scribbler's obstacle sensor and stall sensor to experiment with feedback-driven motion.

Experiment 4 — Avoiding obstacles with the obstacle sensor

Instructions required: Forward, Spin Right, Do forever, If

Program the Scribbler to move around the room while avoiding obstacles. Whenever an obstacle is detected by the obstacle sensor, change the robot's direction before proceeding forward.

Hint: Move the robot forward for a very short time, check the status of the obstacle sensor, and repeat.

As you saw last week, the obstacle sensor will not detect objects at very close range, and it cannot tell exactly how close an obstacle is. These limitations make the obstacle sensor ill-suited for navigating in confined spaces. The *stall sensor* should be used instead.

When the robot is applying power to its motors but they are prevented from turning because the robot it hitting an obstacle, we say the motors have *stalled*. The stall sensor detects this condition.

Up to this point, all the motor instructions you've used specify a duration for the movement. They switch the motor on, wait for some period of time, and then switch the motor off. The stall sensor can't be used with instructions like these, since the motors