Learning Roomba: Student’s Guide
Module 3 Controlling Movement

Drew Housten (dhousten@gmail.com)
1 Why is Movement Important?

Without moving, a robot would be rather boring. Consider a person that did not move at all. The person could sit and observe the world around him or her, but would not be able to act. Movement is necessary to get from one place to another. It is necessary to actively interact with the environment. It is necessary to perform many functions and achieve many tasks. So, one of the first steps to working with Robots is to make them move and understand how they can move.

2 Types of Movement

There are several different types of motion. This Module is only concerned with the ability for the robot to get from one place to another. The types of motion that can be used to achieve this goal depends on the Configuration. For this Module, we are discussing the types of motion associated with a differential drive robot, like the Roomba. To get from one place to another usually consists of a series of different types of motion. This section will discuss the types of motion conceptually, and then you will get to make the Roomba perform the actions through code you write.

2.1 Straight Motion

The first motion is to simply go forward and back without turning at all. This type of motion is called translational motion. To command a differential drive robot to perform this motion, both wheels need to rotate in the same direction at the same speed. If they both rotate forward, the robot will move forward. If they both rotate in the opposite direction, the robot will move backward. Figure 1 shows a forward motion command. The robot starts in the start position and ends in the end position after the command has been executed. Figure 2 shows a backward motion command. The command is actually a straight command with a negative distance associated with it. It will cause the robot to move in the opposite direction from the direction that it is facing.

![Figure 1: Forward Motion](image-url)
2.2 Point Turn

A point turn is a turn in place without translating at all. The type of motion is called rotational motion. A differential drive robot is able to turn in place by rotating the wheels in opposite directions at the same speed. The robot will turn one way if the left wheel rotates forward and will turn the other way if the left wheel rotates backward. Figure 3 shows the robot performing a right turn. The robot will start facing forward and end in the same spot turned to the right 45 degrees. Figure 4 shows the robot performing a left turn. The command is the same as the right turn except that the angle to turn is negative. The robot will start facing forward and end in the same spot turned to the left 45 degrees.

2.3 Swing Turn

A swing turn is a combination of translational motion and rotational motion. It causes the robot to turn while also moving forward (or backward). This motion is achieved by making the wheels rotate at different speeds. An example swing turn motion is shown in Figure 5. The specific path depends on how the speeds...
differ. For example, if the wheels are both moving forward and the left wheel is moving faster than the right, the robot will swing to the right as it moves forward.

Figure 5: Swing Turn Motion

3 How to control the Roomba

The Roomba is a differential drive robot, so it can perform the motions described above. To get the robot from one place to another, a series of motion commands can be provided to the robot’s motors. The motors will make the wheels spin at a specific speed and direction which will cause the robot to travel the expected motion.

Any path can be broken down into a series of motions. A path is a plan for the robot’s motion to get from one point to another. Consider Figure 6. This path can be broken down into a series of just forward and point turn motions. First, the robot must travel forward 3 units. Then it needs to turn to the right 90 degrees. That would get it through the first segment. The full list of commands is:

- Forward 3
- Turn 90
- Forward 4
- Turn -90
- Forward 4
- Turn -90
- Forward 3
- Turn -90
- Forward 6
- Turn -90
- Forward 3
- Turn 90
- Forward 1
- Turn 90
- Forward 2
Figure 6: A path consists of any number of motions
However, there is a limitation with real robots. The problem with real robots, like the Roomba, is that the robot does not always follow the commanded motion exactly. The Roomba knows how far the wheels have rotated, but that does not necessarily mean that the robot has moved the right distance. The wheels could slip, the ground could be uneven, or there could be obstacles in the way. These are just some of the reasons that the Roomba may not follow the exact path that you specify. We will talk about ways to overcome this problem in future Modules, but for now just be aware that the commanded path is not necessarily the same as the actual path traveled by the Roomba.

4 Movement Code Support

In Module 1 we wrote a program and ran it. The program created a Roomba object and made it move forward and then turn. Those were examples of how to make the Roomba move. Those commands and a few others are described here.

- **drive(distance):** This method will cause the Roomba to drive forward if the distance is positive and backward if the distance is negative. The distance traveled will be the distance argument in meters. This will allow the Roomba to perform straight motions by turning both wheels in the same direction and the same speed.

- **turn(angle):** This method will cause the Roomba to turn in place for the angle specified in degrees. The Roomba will turn to the right if the angle is positive and to the left if the angle is negative. This will allow the Roomba to perform point turns by turning the wheels in opposite directions at the same speed.

- **forwardForTime(speed, duration):** This is similar to the drive(distance) command except that the Roomba will move straight at a specified speed for a specified duration of time rather than a specified distance. This will also achieve straight motions.

- **turnForTime(speed, duration):** This is similar to the turn(angle) command except that the Roomba will turn in place at a specified speed for a specified duration of time rather than a specified angle of rotation. This will also achieve point turns.

- **forwardSpeed(speed):** This just sets the speed that the Roomba will move straight at. The Roomba will continue at this speed until a new command with speed 0 is sent. This can be combined with the following command to achieve swing turns.

- **turnSpeed(speed):** Similar to forwardSpeed, this will set the Roomba’s turn speed. The Roomba will continue at this speed until a new command with speed 0 is sent. This can be combined with the previous command to achieve swing turns.
We will primarily use the first two commands. This will restrict the Roomba’s motion to just straight motions and point turns, but they are simpler than swing turns and do not restrict where the Roomba can travel.

An example program using the drive(distance) and turn(angle) commands is provided in Figure 7 that would follow the path described in Figure 6.

```java
import roomba.roombanetwork.services.userservice.*;

public class MyRoombaProgram{

    public static void main(String [] args){
        UserService.setServerAddress("localhost");
        UserService.setName("Your Name");

        Roomba roomba = new Roomba();

        roomba.drive(3);
        roomba.turn(90);
        roomba.drive(4);
        roomba.turn(-90);
        roomba.drive(4);
        roomba.turn(-90);
        roomba.drive(3);
        roomba.turn(-90);
        roomba.drive(6);
        roomba.turn(-90);
        roomba.drive(3);
        roomba.turn(90);
        roomba.drive(1);
        roomba.turn(90);
        roomba.drive(2);

        UserService.disconnect();
        System.exit(1);
    }
}
```

Figure 7: Roomba motion code example
5 Exercises

Now is a chance to write your own programs with the Roomba. Three exercises are provided here to begin familiarizing yourself with the Roomba’s motion commands.

- First, write a program that will use each of the types of motion. Try to get an idea of how the motion works and how to make the Roomba do what you want it to.
- Write a program that will cause the Roomba to drive forward 1 meter, turn around, and drive back to the start position.
- Write a program that will cause the Roomba to drive forward 1 meter, turn left for 5 seconds, and then drive backwards for 0.5 meters.

6 Homework

Pseudocode is a written approach to having a program solve a problem. It does not have to be code that is runnable, but it should outline the basic steps necessary in the program to develop the solution. Write some pseudocode for a program that will make the Roomba drive in a square and return to the original location.