Scientific Experimentation and Evaluation Assignment: 1.1

Petr Lukin, Evgeniya Ovchinnikova

Lecture date: 04th October 2016

1 Construct the simple robot shown on the instruction sheet included in the LEGO construction kit.

We have assembled the robot in accordance to the instruction that is shown in Fig 1:
//TODO: take a photo of the assembled robot and attach here

2 Equip your robot with a pen, such that you can mark the position of the robot at its stop position.

As it is shown in Fig.1, the robot is equipped with two light sensors. NXT light sensors emit bright red light. We have narrowed the light spot, so it would be small enough to mark the position of the robot with a sufficient precision. The advantage of this method is that the robot won't be moved in the moment of position measurement, as it can easily be moved when a pen touches the surface.

3 Verify your robot is running LeJOS and program it to run for a certain distance.

We have installed and tested LeJOS. The test was performed with the following program for forward motion.

```
package LegoNXT;

import lejos.nxt.LightSensor;
import lejos.nxt.Motor;
import lejos.nxt.SensorPort;
import lejos.robotics.navigation.DifferentialPilot;
import lejos.robotics.navigation.MoveController;
import lejos.util.Delay;

public class goStraight {

        public static void main(String[] args) {
                double arcRad = 30;
                double angle = 90;
```

```
                LightSensor lightSens1 = new LightSensor(SensorPort.S1);
                LightSensor lightSens2 = new LightSensor(SensorPort.S2);
                DifferentialPilot dp = new DifferentialPilot
                (MoveController.WHEEL_SIZE_NXT1, 18, Motor.B, Motor.C,
                true);
                lightSens1.setHigh(100);
                lightSens2.setHigh(100);
                Delay.msDelay(5000);
                dp.setTravelSpeed(10);
                dp.travel(-Math.PI*2*arcRad*angle/360);
                dp.stop();
                Delay.msDelay(10000);
        }
}
```

4   What and how we are planning to do.

We are planning to observe the motion model of Lego NXT robot with a differential drive.
Based on these observations we will obtain a distribution of final poses and calculate
motion parameters.
The measuring system is following: the differential drive robot with two light sensors
with narrow window for light emitting so it would emit only a point of light and a large
sheet of paper with a coordinate system and two "garage" points which are determining
the initial position of the robot - the light points must be on these points. On the paper
sheet we can mark the robot's position using a pen that is not attached to the robot. To
measure the coordinates of the robot's positions we are going to use a ruler and to store
these results we are going to use a table in a LibreOffice Calc file.
The measuring method is the following: we are planning to estimate the position of
the robot based on positions of two points of light emitted by light sensors. The starting
position, the "garage", will also be marked as two points on the paper. The measurement
result will be obtained by conducting three series of experiments (around 30 times each):
forward motion, left arc motion, right arc motion. After each of those experiments we
measure the coordinates of the light points (measured value). For the forward motion
the program above will be used for the left and right motions the following programs will
be used:

```
package LegoNXT;

import lejos.nxt.Motor;
import lejos.nxt.SensorPort;
import lejos.robotics.navigation.DifferentialPilot;
import lejos.robotics.navigation.MoveController;
import lejos.util.Delay;
import lejos.nxt.LightSensor;

public class goLeft {

        public static void main(String[] args) {
```

```
                double arcRad = 30;
                double angle = 90;
                LightSensor lightSens1 = new LightSensor(SensorPort.S1);
                LightSensor lightSens2 = new LightSensor(SensorPort.S2);
                DifferentialPilot dp = new DifferentialPilot
                (MoveController.WHEEL_SIZE_NXT1, 18, Motor.B,
                Motor.C, true);
                lightSens1.setHigh(100);
                lightSens2.setHigh(100);
                Delay.msDelay(5000);
                dp.setTravelSpeed(10);
                dp.arc(-arcRad, angle);
                dp.stop();
                Delay.msDelay(10000);
        }
}

***

package LegoNXT;

import lejos.nxt.LightSensor;
import lejos.nxt.Motor;
import lejos.nxt.SensorPort;
import lejos.robotics.navigation.DifferentialPilot;
import lejos.robotics.navigation.MoveController;
import lejos.util.Delay;

public class goRight {

        public static void main(String[] args) {
                double arcRad = 30;
                double angle = 90;
                LightSensor lightSens1 = new LightSensor(SensorPort.S1);
                LightSensor lightSens2 = new LightSensor(SensorPort.S2);
                DifferentialPilot dp = new DifferentialPilot
                (MoveController.WHEEL_SIZE_NXT1, 18, Motor.B,
                 Motor.C, true);
                lightSens1.setHigh(100);
                lightSens2.setHigh(100);
                Delay.msDelay(5000);
                dp.setTravelSpeed(10);
                dp.arc(arcRad, -angle);
                dp.stop();
                Delay.msDelay(10000);
        }
}
```

5   Which difficulties we expect.

- First difficulty that we expect is slightly different wheel diameter and uneven work of the motors. NXT Lego robots usually have problems because of slightly unsynchronized motor work. In this case the forward motion will be not exactly straight and arcs will have slightly wrong parameters.

- Another possible difficulty in the measurement can be caused by incline of the surface, on which we put the sheet of paper, or the paper itself. In this case we could have the same problems as in previous case.

- Since robot is shaking while moving, light sensors can slightly change their positions. In this case on or the both points can be moved and we estimate the position of the robot incorrectly.

- Imperfection of the coordinate grid and a ruler that will cause a systematic error that we must take into account.

- Imperfection of a light point and a pen - both, the point, and a pen are not material points, so they have a certain diameter that will add a systematic error.