

Robotics Journal

M.Sc Part II Computer Science

Shubham Lad 502

October 13, 2023



R.J. College of Arts, Science & Commerce

Robotics

Seat number: 502

Contents

Practical No 1: A) Moving a robot left, right, and forward using gears. B) Moving a robot in a square motion	1
Practical No 2: Robot performing circular motion	4
Practical No 3: Moving a robot with motors	5
Practical No 4: A) Circular motion using arch function with gear. B) Circular motion using arch function without gear.	7
Practical No 5: A) Line-following robot. B) Line-following robot on a different image.	10
Practical No 6: A) Path finding robot. B) Path finding robot on a different image	13
Practical No 7: Obstacle resistance using touch sensor.	17
Practical No 8: Write a program to create a robot with ultrasonic sensor.	19
Practical No 9: Torch follower	21
Practical No 10: Shadow follower.	23

List of Figures

Practical No: 1

Aim:A) Moving a robot left, right, and forward using gears.

Code:

```
1 package robotics;
2 import ch.aplu.robotsim.Gear;
3 import ch.aplu.robotsim.NxtRobot;
4 /**
5  *
6  * @author
7  */
8
9 public class Robo_p1a {
10     /**
11      * @param args the command line arguments
12      */
13     Robo_p1a(){
14         NxtRobot robot=new NxtRobot();
15
16         Gear gear=new Gear();
17         robot.addPart(gear);
18
19         gear.setSpeed(10);
20
21         gear.forward(4000);
22         gear.left(5000);
23
24         gear.forward(5000);
25         gear.right(4800);
26
27         robot.exit();
28     }
29     public static void main(String[] args) {
30         // TODO code application logic here
31         Robo_p1a m=new Robo_p1a();
32     }
33
34 }
```

Output :

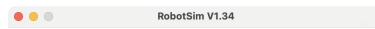


Aim:B) Moving a robot in a square motion.

Code:

```
1  package robotics;
2  import ch.aplu.robotsim.TurtleRobot;
3  /**
4   *
5   * @author mithunparab
6   */
7
8  public class Robo_p1b {
9      Robo_p1b()
10     {
11         TurtleRobot t = new TurtleRobot();
12         t.forward(100);
13         t.right(30);
14         t.forward(50);
15         t.left(90);
16         t.forward(70);
17     }
18     public static void main(String[] args){
19         new Robo_p1b();
20     }
21 }
```

Output :



Practical No: 2

Aim: Robot performing circular motion

Code:

```
1 package robotics;
2
3 /**
4  *
5  * @author
6  */
7 import ch.aplu.robotsim.Gear;
8 import ch.aplu.robotsim.NxtRobot;
9
10 public class Robo_p2 {
11
12     Robo_p2(){
13         NxtRobot robot=new NxtRobot();
14
15         Gear gear=new Gear();
16         robot.addPart(gear);
17         gear.setSpeed(150);
18
19         for (int i=0;i<100;i++){
20             gear.forward(150);
21             gear.right(30);
22         }
23         robot.exit();
24     }
25
26     public static void main(String[] args) {
27         // TODO code application logic here
28         Robo_p2 m=new Robo_p2();
29     }
30 }
```

Output :



Practical No: 3

Aim: Moving a robot with motors

Code:

```
1  package robotics;
2
3
4  import ch.aplu.robotsim.Motor;
5  import ch.aplu.robotsim.MotorPort;
6  import ch.aplu.robotsim.NxtRobot;
7  import ch.aplu.robotsim.Tools;
8  /**
9   *
10   * @author admin
11   */
12
13
14  //Aim: Move robot with the help of motors
15  public class Robo_p3 {
16      public Robo_p3(){
17          NxtRobot robot = new NxtRobot();
18          Motor motA = new Motor(MotorPort.A);
19          Motor motB = new Motor(MotorPort.B);
20          robot.addPart(motA);
21          robot.addPart(motB);
22          motA.forward();
23          motB.forward();
```

```

24     Tools.delay(2000);
25
26     motA.stop();
27     Tools.delay(1050);
28     motA.forward();
29     Tools.delay(2000);
30
31     motB.stop();
32     Tools.delay(1050);
33     motB.forward();
34     Tools.delay(2000);
35
36     robot.exit();
37 }
38
39 public static void main(String args[]){
40     Robo_p3 r = new Robo_p3();
41 }
42 }

```

Output :



Practical No: 4

Aim: A) Circular motion using arch function with gear.

Code:

```
1  package robotics;
2
3  import ch.aplu.robotsim.Gear;
4  import ch.aplu.robotsim.NxtRobot;
5  import ch.aplu.robotsim.Tools;
6  /**
7   *
8   * @author admin
9   */
10 public class Robo_p4a {
11
12     Robo_p4a(){
13         NxtRobot robot=new NxtRobot();
14         Gear gear = new Gear();
15         robot.addPart(gear);
16         gear.setSpeed(60);
17         gear.leftArc(0.2,7000);
18         gear.rightArc(0.2);
19         Tools.delay(5000);
20         robot.exit();
21     }
22
23     public static void main(String args[]){
24         Robo_p4a p4a=new Robo_p4a();
25     }
26 }
```

Output :

Aim: B) Circular motion using arch function without gear.

Code:



```
1 package robotics;
2
3 import ch.aplu.robotsim.*;
4 import ch.aplu.robotsim.NxtRobot;
5 /**
6  *
7  * @author admin
8  */
9 public class Robo_p4b {
10     Robo_p4b(){
11         NxtRobot robot=new NxtRobot();
12         Gear gear = new Gear();
13         robot.addPart(gear);
14         gear.forward(200);
15         gear.setSpeed(60);
16         gear.leftArc(0.2,7000);
17         gear.forward(200);
18         gear.leftArc(0.2,7000);
19         gear.forward(200);
20         gear.leftArc(0.2,7000);
21         gear.forward(200);
22         gear.leftArc(0.2,7000);
23         gear.forward(200);
24         robot.exit();
25     }
26     public static void main(String args[]){
27         Robo_p4b p4b=new Robo_p4b();
28         NxtContext.setStartPosition(250,200);
```

```
29     NxtContext.setStartDirection(90);  
30 }  
31 }
```

Output :



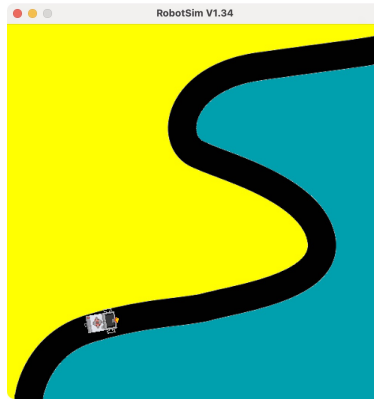
Practical No: 5

Aim: A) Line-following robot.

Code :

```
1 package robotics;
2
3 import ch.aplu.robotsim.*;
4
5 /**
6  *
7  * @author admin
8  */
9 public class Robo_p5a {
10     Robo_p5a(){
11         LegoRobot robot=new LegoRobot();
12         Gear gear = new Gear();
13         LightSensor ls = new LightSensor(SensorPort.S3);
14         robot.addPart(gear);
15         gear.setSpeed(20);
16         robot.addPart(ls);
17         while (true){
18             int v=ls.getValue();
19             if (v<100)//black
20                 gear.forward();
21             if (v>300 && v<750)//blue
22                 gear.leftArc(0.05);
23             if (v>800)//yellow
24                 gear.rightArc(0.05);
25         }
26     }
27     public static void main(String args[]){
28         Robo_p5a p5a = new Robo_p5a();
29     }
30
31     static{
32         RobotContext.setStartPosition(50, 490);
33         RobotContext.setStartDirection(-90);
34         RobotContext.useBackground("sprites/road.gif");
35     }
```

Output :



Aim: B) Line-following robot on a different image.

Code :

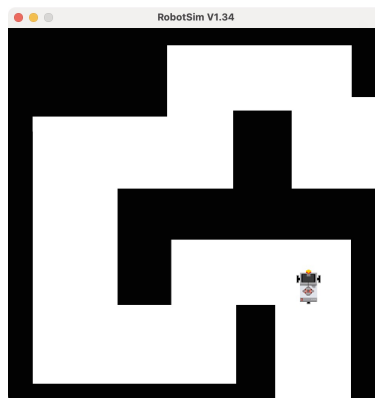
```
1 package robotics;
2
3 import ch.aplu.robotsim.Gear;
4 import ch.aplu.robotsim.LegoRobot;
5 import ch.aplu.robotsim.LightSensor;
6 import ch.aplu.robotsim.RobotContext;
7 import ch.aplu.robotsim.SensorPort;
8
9 /**
10  *
11  * @author admin
12  */
13 public class Robo_p5b {
14     Robo_p5b(){
15         LegoRobot robot=new LegoRobot();
16         Gear gear = new Gear();
17         LightSensor ls = new LightSensor(SensorPort.S3);
18         robot.addPart(gear);
19         gear.setSpeed(50);
```

```

20     robot.addPart(ls);
21     while (true){
22         int v=ls.getValue();
23         System.out.println(v);
24         gear.forward();
25         if (v>1000 || v<=7)
26             gear.leftArc(0.05);
27
28         // for (int i=0;i>10000;i++){
29             //gear.forward(150);
30             // gear.right(30);
31         // }
32         // gear.rightArc(0.05);
33     }
34 }
35 public static void main(String args[]){
36     Robo_p5b p5b = new Robo_p5b();
37 }
38 static{
39     RobotContext.setStartPosition(400, 450);
40     //RobotContext.setStartDirection(-90);
41     RobotContext.useBackground("sprites/bg2.gif");
42 }
43 }

```

Output :



Practical No: 6

Aim: A) Path finding robot.

Code :

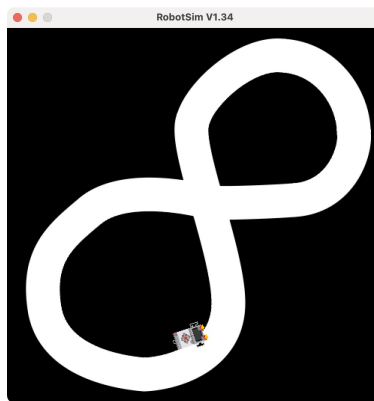
```
1 package robotics;
2
3 import ch.aplu.robotsim.*;
4 import ch.aplu.robotsim.LightSensor;
5 import ch.aplu.robotsim.NxtRobot;
6 import ch.aplu.robotsim.SensorPort;
7
8 /**
9  *
10  * @author admin
11  */
12 public class Robo_p6a {
13     Robo_p6a(){
14
15         NxtRobot robot=new NxtRobot();
16         Gear gear=new Gear();
17         LightSensor ls1=new LightSensor(SensorPort.S1);
18         LightSensor ls2=new LightSensor(SensorPort.S2);
19         robot.addPart(gear);
20         robot.addPart(ls1);
21         robot.addPart(ls2);
22         gear.forward();
23
24         while(true)
25         {
26             int rightValue = ls1.getValue();
27             int leftValue = ls2.getValue();
28             int d = rightValue-leftValue;
29
30             if (d>100)
31                 gear.rightArc(0.1);
32             if (d< -100)
33                 gear.leftArc(0.1);
34             if (d>-100 && d<100 && rightValue>500)
35                 gear.forward();
```

```

36     }
37 }
38
39 public static void main(String args[]){
40     Robo_p6a p6a=new Robo_p6a();
41 }
42 static{
43     NxtContext.setStartPosition(250,490);
44     NxtContext.setStartDirection(-90);
45     NxtContext.useBackground("sprites/path.gif");
46 }
47 }

```

Output :



Aim: B) Path finding robot on a different image

Code :

```

1 package robotics;
2
3 import ch.aplu.robotsim.Gear;
4 import ch.aplu.robotsim.LightSensor;
5 import ch.aplu.robotsim.NxtContext;
6 import ch.aplu.robotsim.NxtRobot;
7 import ch.aplu.robotsim.SensorPort;
8

```

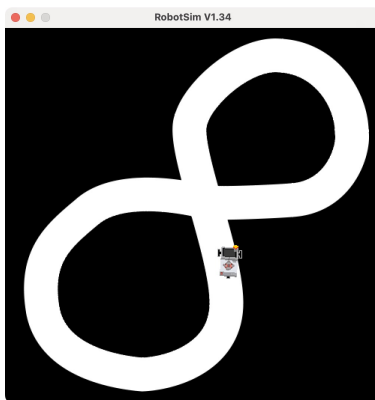


```

9  /**
10  *
11  * @author admin
12  */
13  public class Robo_p6b {
14
15      Robo_p6b(){
16          NxtRobot robot=new NxtRobot();
17          Gear gear=new Gear();
18          LightSensor ls1=new LightSensor(SensorPort.S1);
19          //LightSensor ls2=new LightSensor(SensorPort.S2);
20          robot.addPart(gear);
21          robot.addPart(ls1);
22          //robot.addPart(ls2);
23          gear.forward();
24
25          while(true)
26          {
27              int d = ls1.getValue();
28              //System.out.print(d+'\n');
29              //int leftValue = ls2.getValue();
30              //int d = rightValue-leftValue;
31
32              if (d>100)
33                  gear.rightArc(0.1);
34              if (d<100)
35                  //gear.forward();
36                  gear.leftArc(0.1);
37              if (d>100 && d<-100)
38                  gear.forward();
39          }
40      }
41
42      public static void main(String args[]){
43          Robo_p6b p6b=new Robo_p6b();
44      }
45      static{
46          NxtContext.setStartPosition(250,490);
47          NxtContext.setStartDirection(-90);
48          NxtContext.useBackground("sprites/path.gif");
49      }

```

Output :



Practical No: 7

Aim: Obstacle resistance using touch sensor.

Code :

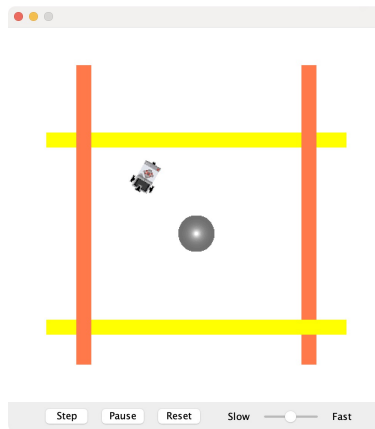
```
1 package robotics;
2 import ch.aplu.robotsim.*;
3
4 public class Robo_p7 {
5
6     static {
7         RobotContext.showNavigationBar();
8         RobotContext.useObstacle("sprites/bar2.gif", 250, 150); //up
9         RobotContext.useObstacle("sprites/bar1.gif", 400, 250); //right
10        RobotContext.useObstacle("sprites/torch.png", 250, 275); //center circle
11        RobotContext.useObstacle("sprites/bar2.gif", 250, 400); //down
12        RobotContext.useObstacle("sprites/bar1.gif", 100, 250); //left
13    }
14
15    public Robo_p7() {
16        LegoRobot robot = new LegoRobot();
17        Gear gear = new Gear();
18        TouchSensor ts = new TouchSensor(SensorPort.S3);
19        robot.addPart(gear);
20        robot.addPart(ts);
21        gear.setSpeed(30);
22        gear.forward();
23
24        while (true) {
25            if (ts.isPressed()) {
26                gear.backward(500);
27                gear.left(750);
28                gear.forward();
29            }
30        }
31    }
32
33    public static void main(String[] args) {
34        Robo_p7 mp = new Robo_p7();
35    }
```

36

37

```
}
```

Output :



Practical No: 8

Aim: Write a program to create a robot with ultrasonic sensor.

Code :

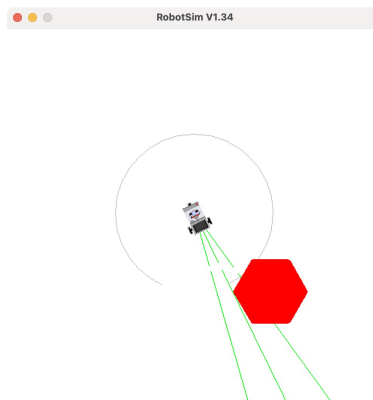
```
1  /*
2  Search fixed target using the ultrasonic Sensor
3  */
4  package robotics;
5
6  import ch.aplu.robotsim.*;
7  import java.awt.*;
8
9  public class Robo_p8 {
10     private final LegoRobot robot;
11     private final Gear gear;
12     private final UltrasonicSensor us;
13
14     public Robo_p8() {
15         robot = new TurtleRobot();
16         gear = new Gear();
17         robot.addPart(gear);
18         gear.setSpeed(10);
19         us = new UltrasonicSensor(SensorPort.S1);
20         robot.addPart(us);
21         us.setBeamAreaColor(Color.green); // May be commented out
22         us.setProximityCircleColor(Color.lightGray); // May be commented out
23         runRobotProgram();
24     }
25
26     private void runRobotProgram() {
27         searchTarget();
28         while (true) {
29             if (us.getDistance() < 10)
30                 gear.stop();
31         }
32     }
33
34     private void searchTarget() {
35         while (true) {
```

```

36         gear.right(50);
37         int distance = us.getDistance();
38         if (distance != -1) {
39             gear.right(1500);
40             gear.forward();
41             return;
42         }
43     }
44 }
45
46 public static void main(String[] args) {
47     Robo_p8 a = new Robo_p8();
48 }
49
50 // ----- Environment -----
51 static {
52     Point[] mesh = {
53         new Point(50, 0), new Point(25, 42), new Point(-25, 42),
54         new Point(-50, 0), new Point(-25, -42), new Point(25, -42)
55     };
56
57     RobotContext.useTarget("sprites/target_red.gif", mesh, 350, 350);
58 }
59 }

```

Output :



Practical No: 9

Aim: Torch follower.

Code :

```
1  package robotics;
2  import ch.aplu.robotsim.*;
3  /**
4   *
5   * @author
6   */
7
8  public class Robo_p9
9  {
10     public Robo_p9()
11     {
12         LegoRobot robot = new LegoRobot();
13         LightSensor lsFR = new LightSensor(SensorPort.S1, true);
14         LightSensor lsFL = new LightSensor(SensorPort.S2, true);
15         LightSensor lsRR = new LightSensor(SensorPort.S3, true);
16         LightSensor lsRL = new LightSensor(SensorPort.S4, true);
17
18         Gear gear = new Gear();
19         robot.addPart(gear);
20         robot.addPart(lsFR);
21         robot.addPart(lsFL);
22         robot.addPart(lsRL);
23         robot.addPart(lsRR);
24
25         gear.setSpeed(25);
26         gear.forward();
27         double s = 0.02;
28         while (!robot.isEscapeHit())
29         {
30             int vFR = lsFR.getValue();
31             int vFL = lsFL.getValue();
32             int vRR = lsRR.getValue();
33             int vRL = lsRL.getValue();
34             double d = 1.0 * (vFL - vFR) / (vFL + vFR);
35
```

```

36     if (vRL + vRR > vFL + vFR) // torch behind robot
37         gear.left();
38     else if (d > -s && d < s)
39         gear.forward();
40     else
41     {
42         if (d >= s)
43             gear.leftArc(0.05);
44         else
45             gear.rightArc(0.05);
46     }
47     Tools.delay(100);
48 }
49 robot.exit();
50 }
51
52 public static void main(String[] args)
53 {
54     new Robo_p9();
55 }
56
57 // ----- Environment -----
58 static
59 {
60     RobotContext.useTorch(1, 150, 250, 100);
61 }
62 }
63

```

Output :



Practical No: 10

Aim: Shadow follower.

Code :

```
1  package robotics;
2  import ch.aplu.robotsim.*;
3  import java.awt.Color;
4  /**
5   *
6   * @author
7   */
8
9  public class Robo_p10
10 {
11     public Robo_p10()
12     {
13         LegoRobot robot = new LegoRobot();
14         LightSensor ls = new LightSensor(SensorPort.S3, true);
15         robot.addPart(ls);
16         Gear gear = new Gear();
17         robot.addPart(gear);
18
19         gear.leftArc(0.5);
20         boolean isLightOn = false;
21         while (!robot.isEscapeHit())
22         {
23             int v = ls.getValue();
```

```

24     if (!isLightOn && v == 0)
25     {
26         isLightOn = true;
27         robot.playTone(2000, 100);
28     }
29     if (isLightOn && v > 0)
30     {
31         isLightOn = false;
32         robot.playTone(1000, 100);
33     }
34     Tools.delay(100);
35 }
36 robot.exit();
37 }
38
39 public static void main(String[] args)
40 {
41     new Robo_p10();
42 }
43
44 // ----- Environment -----
45 static
46 {
47     RobotContext.useTorch(1, 250, 250, 100);
48     RobotContext.useShadow(50, 150, 450, 200);
49     RobotContext.useShadow(100, 300, 350, 450);
50 }
51 }

```

Output :

