# Scientific Experimentation and Evaluation

Aaqib Parvez Mohammed
Chetan Sidnal
Mihir Patil
Sushma devaramani

June 5, 2018

# Contents

## List of Figures

# 1 Experiment 1: Manual Motion Observation

**Aim:** Manually measure the observable pose variation for three different constant velocity motions(Straight line, an arc to the left and an arc to the right) of LEGO NXT differential drive robot.

## 1.1 Setup

- The measurement system for the experiment includes a white sheet, a LEGO NXT differential drive robot, two pencils and a scale.
- The **device under test(DUT)** is the LEGO NXT differential drive robot.
- Clamp the sheet on a flat surface.
- Construct a robot equipped with two pencils in the front, such that the start and end positions can be marked. This constitutes the measurement facility.
- Install *Lejos* OS on the robot.
- Program the robot for it to run in three different constant velocity motions.

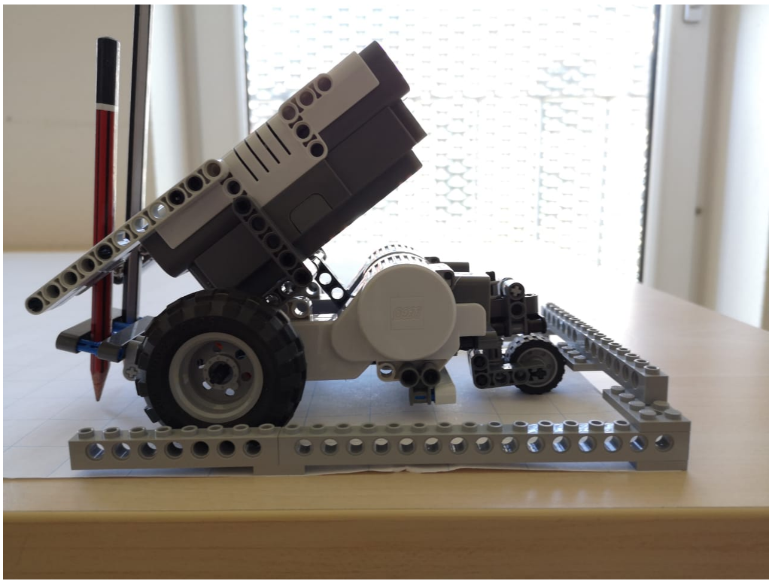

Figure 1: Side view

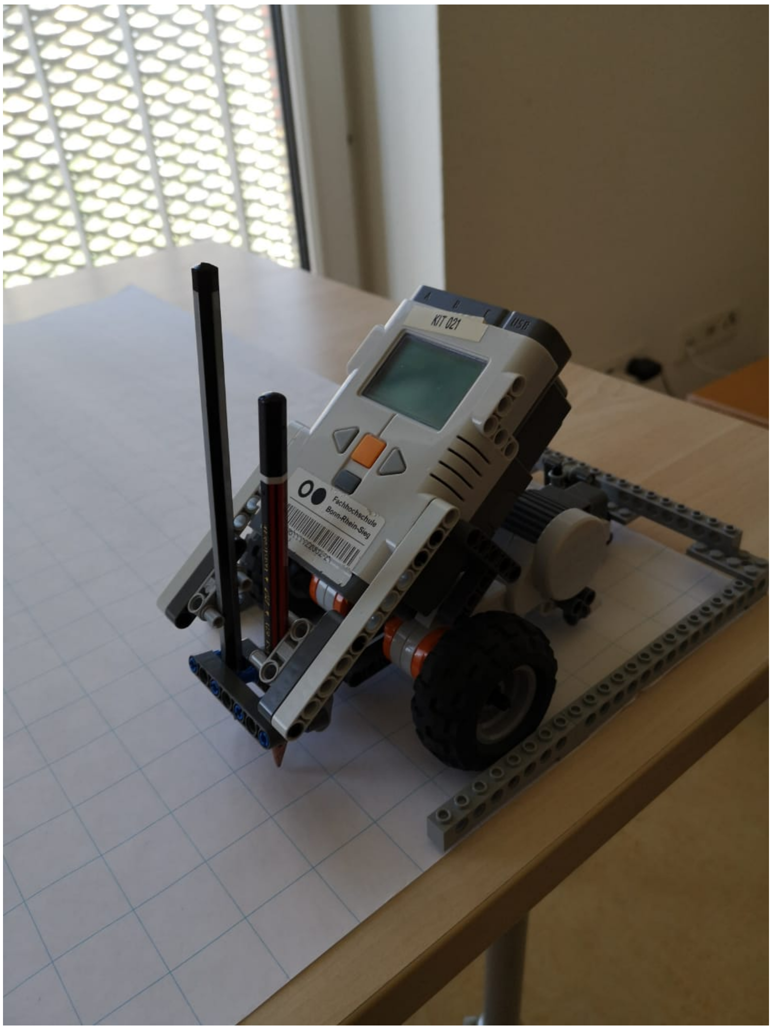

Figure 2: Isometric view

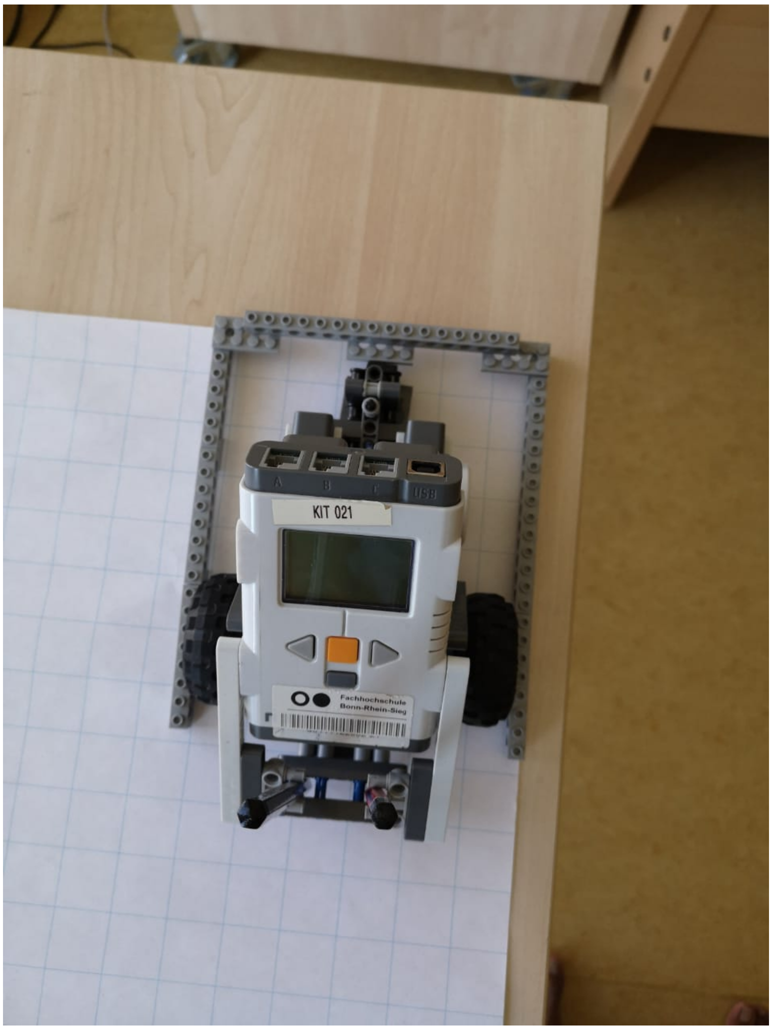

Figure 3: Top view

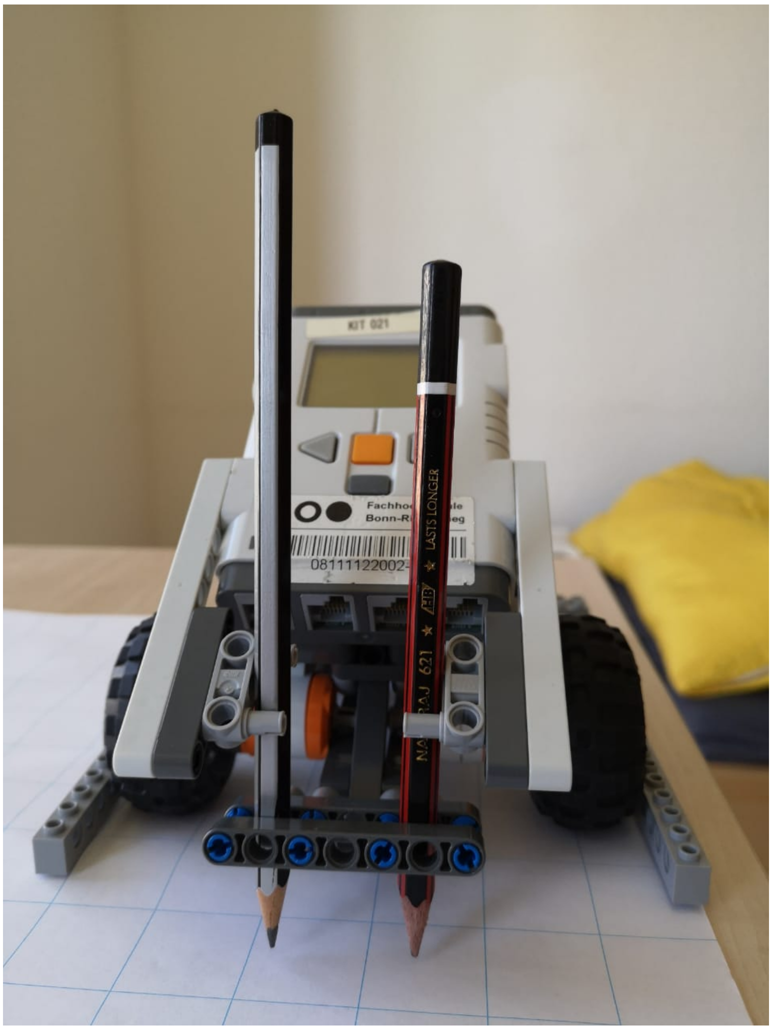

Figure 4: front view

## 1.2 Procedure

- The experimental setup is prepared.
- The starting position of the robot is marked with the help of the points drawn by the pencils.
- The center of the line joining the two points is recorded which will be the starting position of the robot.
- The robot is programmed to run in a straight line forward motion at a constant speed.
- Once the robot stops, the center of the line joining the last points drawn by the pencils will be the stop position.
- The **Measurand** i.e., the relative coordinates of the robot are measured by the difference in the coordinates of the starting position and the stopping position.
- **Measurement System:** Our measurement system consists of robot placed on a cardboard sheet. Two pencils are attached in the front to mark the position. As the robot is moved, the path is traced on the sheet. After reaching the end position, the final pose is marked. The variation in the path is observed and the error in pose is measured.
- The **Measurement result** i.e., the pose variation is obtained by finding the difference between the observed values and their mean.
- Repeat the above experiment with a program to run the robot with constant angular and translational velocities for a fixed time period, such that it describes an arc to the left.
- Repeat the above experiment with a program to run the robot with constant angular and translational velocities for a fixed time period, such that it describes an arc to the right.

## 1.3 Expected problems

- Parallax error.
- Positional errors.
- There might be slipping of wheels.
- The tip of the pencils might break during the motion of the robot.
- The pencils will have to be sharpened as the nibs might get blunt due to continuous drawing of lines which might result in measurement errors.

## 1.4 Expected performance

- The Rotation sensor measures the motor rotations with the accuracy of +/- one degree for one rotation(360 degree). Hence the accuracy in distance traveled for one rotation will be +/- 0.48mm.
- Reference: LEGO NXT: Features & Limitations.
- Assuming experiment distance = 704mm. Expected accuracy is ±2 and expected precision 704mm is±2mm.

### 1.4.1 Reasons

- The slippage of the wheels may cause poor accuracy and precision.
- The internal errors from the DUT may cause poor accuracy but the precision won't be affected significantly, since the DUT won't be changed while doing the measurement.

## 1.5 Execution of the experiment:

- The starting position of the robot was marked using two pencils which were placed on either side of the robot.
- The same points were used as the starting points while repeating the experiment.
- The position of the back wheel was also kept constant by drawing a line and using the same line as reference for future run of experiments.
- The position of the back wheel impacted the outcome of the experiment as a small deviation from the reference line would result in a considerable deviation of the outcome.
- The theta values were calculated by drawing a perpendicular to the line joining the two observed end points and calculating the angle using the formula $tan^{-1}(\frac{y}{x})$.
- Lego NXT 2.0 software was used to program the robot with the following parameters.
  No. of rotations = 4
  Power = 40
- The battery voltage before the experiment = 8.2V.
  The battery voltage after the experimmant = 4.2V.
- The observed data has been recorded in the ".csv" format.

- There was no pre-processing of data required as no outliers were detected.
- Graph has been plotted for all three motions.
  Note : All x,y measurements are in inches

### 1.5.1 Straight

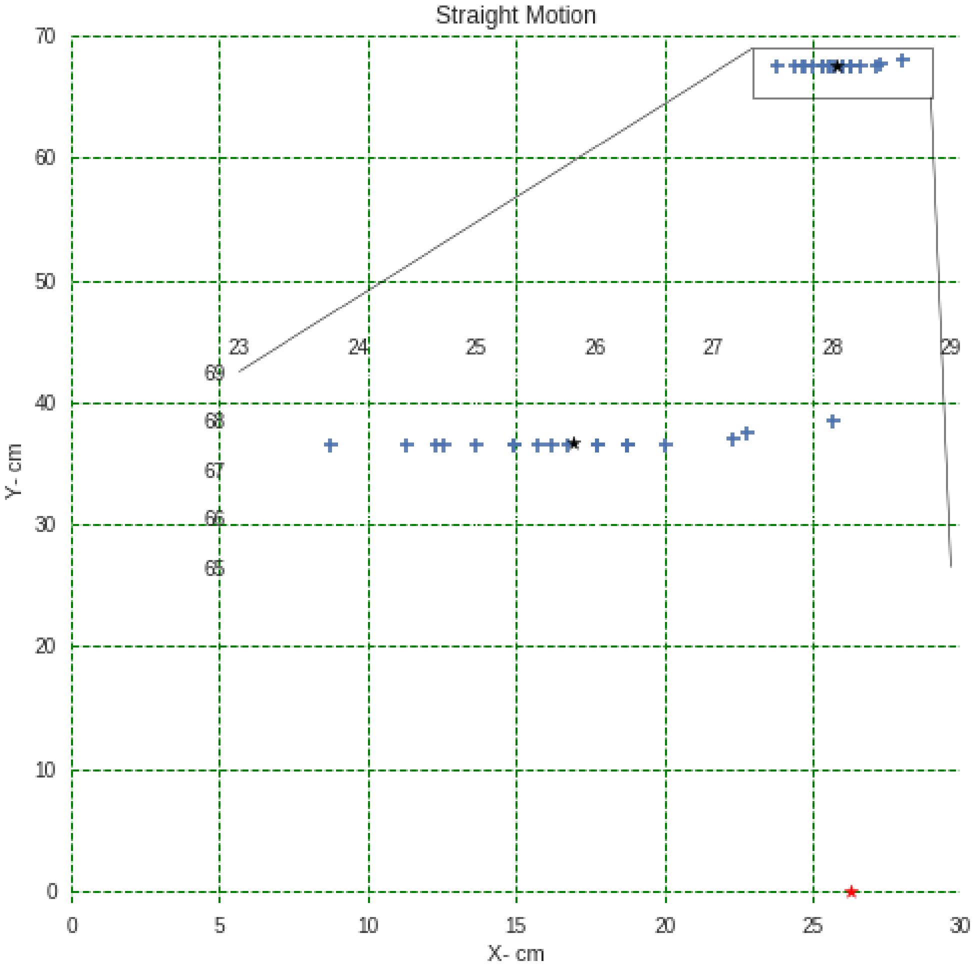

Figure 5: 20 iteration of moving straight

**Fitting the distribution for the obtained data:**
**(X, Y) points for straight line**

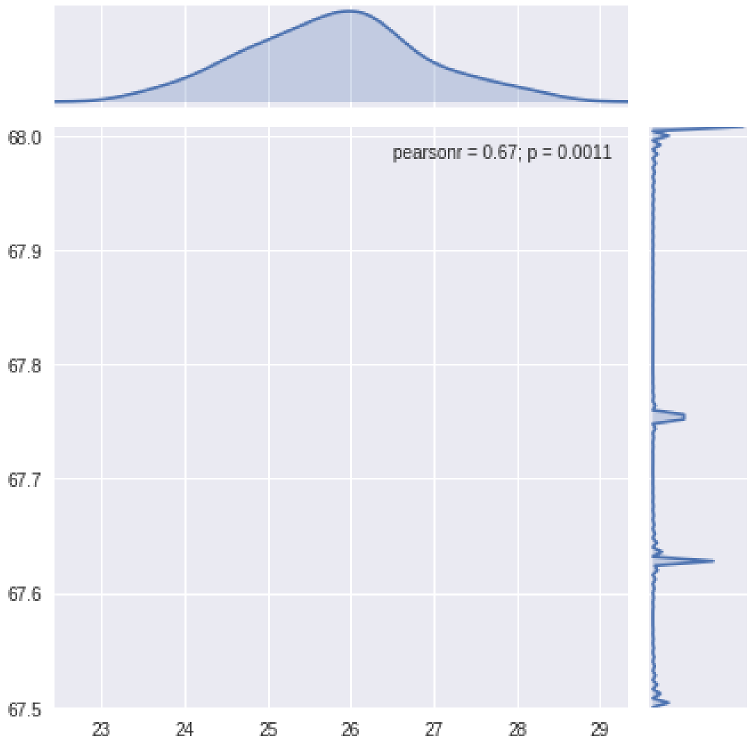

Figure 6: Density plot of (X,Y) for straight line motion

In the above graph, it can be observed that 'X' points fit a gaussian distribution, whereas 'Y' points does not fit any distribution, since they are mostly precise and accurate.

- $Range(x, y) = ([23.77 - 28.01], [67.50 - 68.01])$
- $Mean(x, y) = 25.81, 67.54$
- $\sigma(x, y) = 1.01, 0.12$
- $Accuracy(x, y) = x \pm 0.9, y \pm 0.1$

### 1.5.2 Right

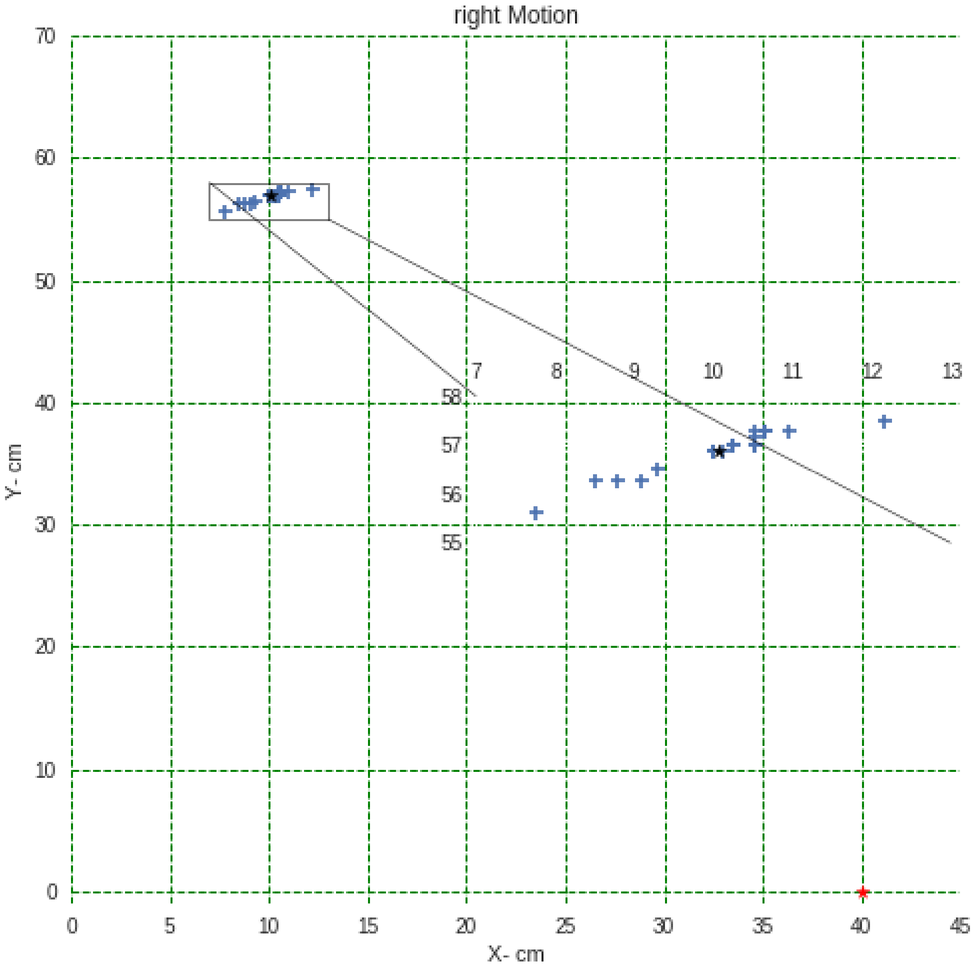

Figure 7: 20 observations of right arc motion

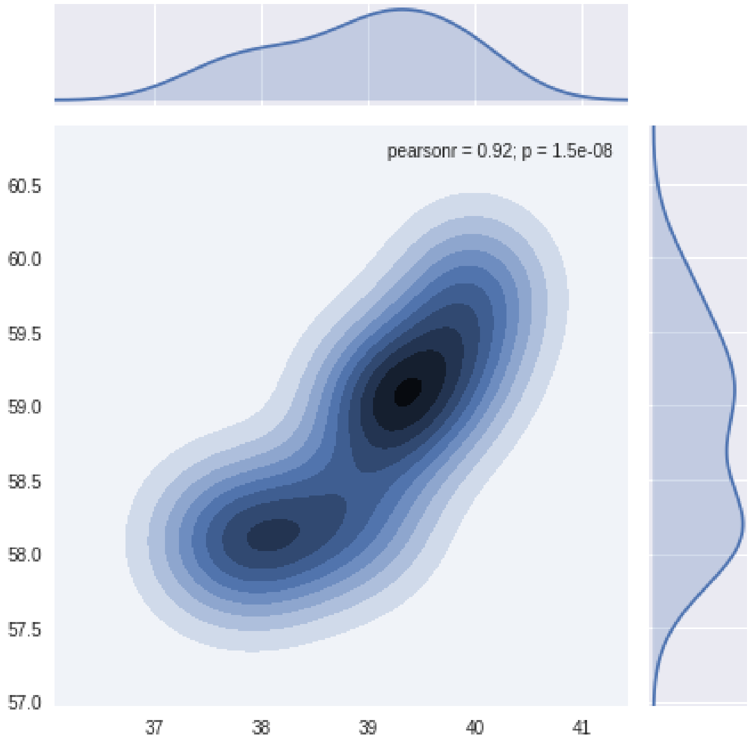

Figure 8: Density plot of (X, Y) points for right arc motion

In the above plot it can be observed that the distribution of X and Y positions for right arc motion is gaussian.

- $Range(x, y) = ([37.50 - 40.00], [58.01 - 59.78])$
- $Mean(x, y) = 38.95, 58.77$
- $Accuracy(x, y) = x \pm 0.5, y \pm 0.5$
- $\sigma(x, y) = 0.81, 0.61$
- $Mean(\theta) = 40.58degree$
- $Accuracy(\theta) = \theta \pm 5deg$
- $\sigma = 1.09deg$
- $range(\theta) = 38.83 - 43.33$

### 1.5.3 Left

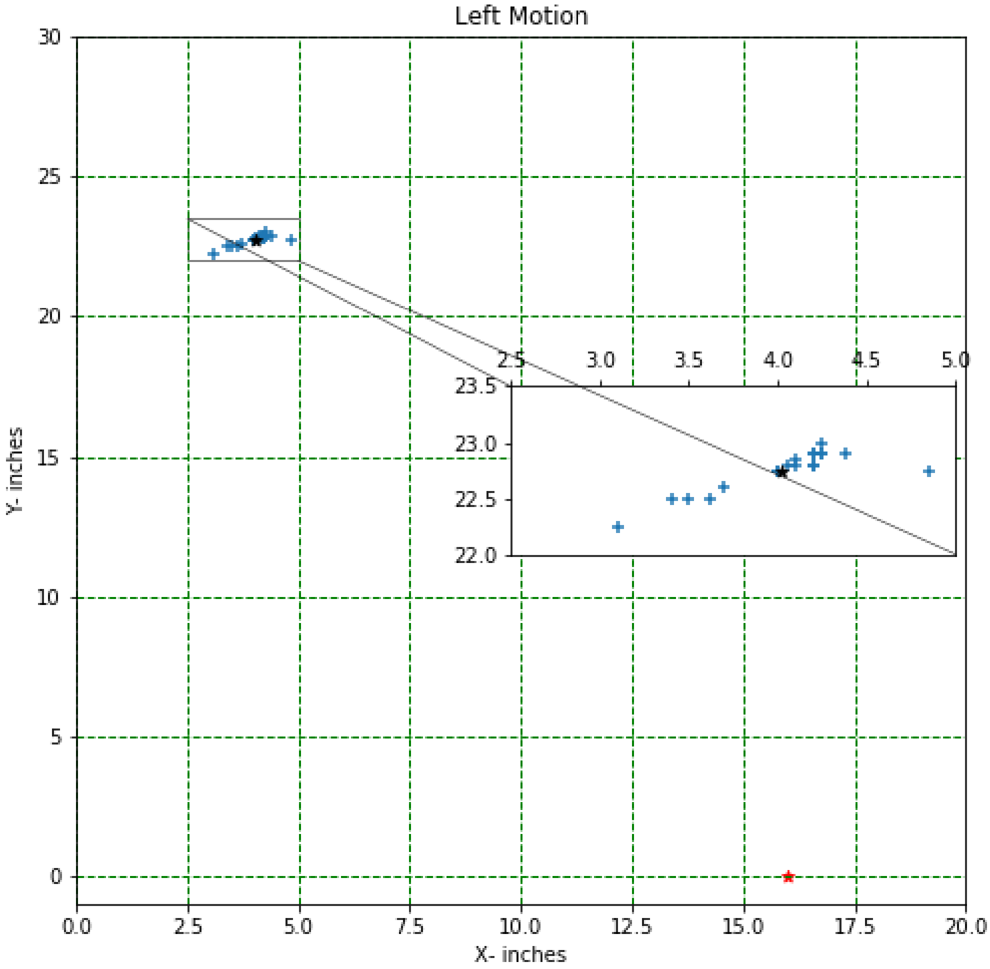

Figure 9: 20 iteration of moving left

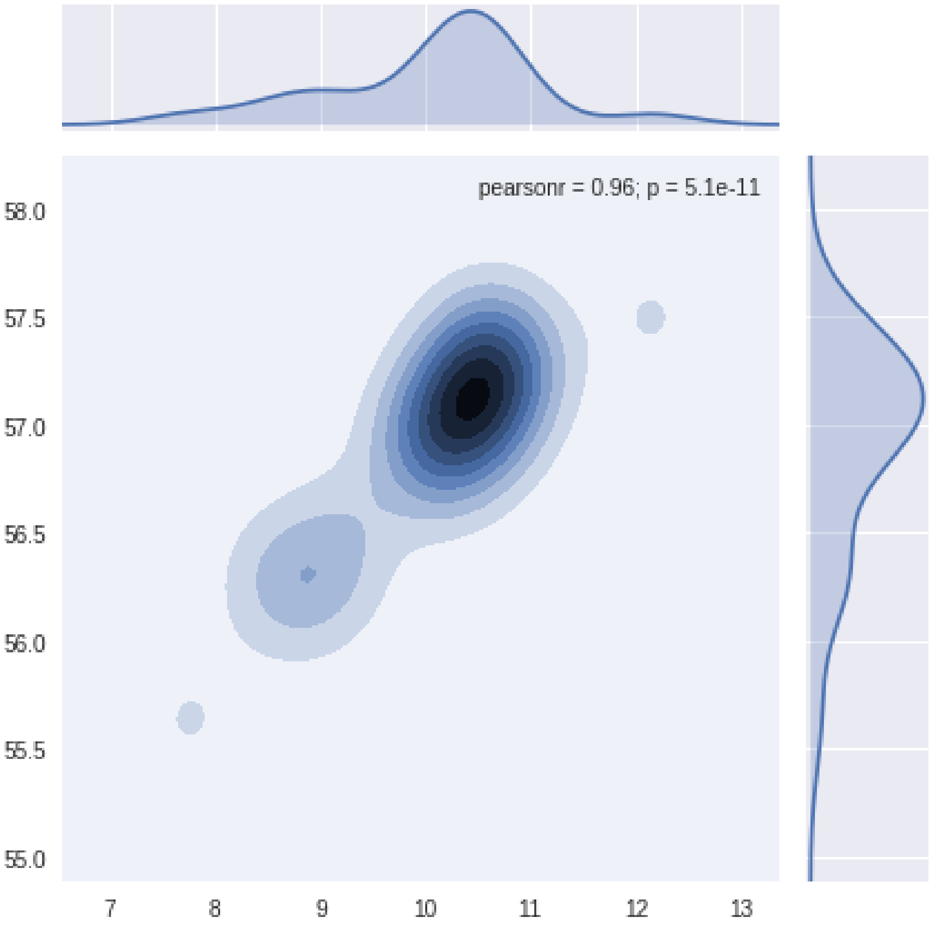

Figure 10: Density plot of (X, Y) for moving left

In the above figure it can be observed that the (X, Y) points obtained for left arc motion fit a gaussian distribution.

- $Range(x, y) = ([7.75 - 12.16], [55.63 - 57.50])$
- $Mean(x, y) = 10.05, 56.89$
- $Accuracy(x, y) = x \pm 0.5, y \pm 0.5$
- $\sigma(x, y) = 0.97, 0.47$
- $Mean(\theta) = 45.71 degree$
- $Accuracy(\theta) = \theta 7 \pm deg$
- $\sigma = 1.32 deg$
- $range(\theta) = 43.18 - 48.27$

**Boxplots**

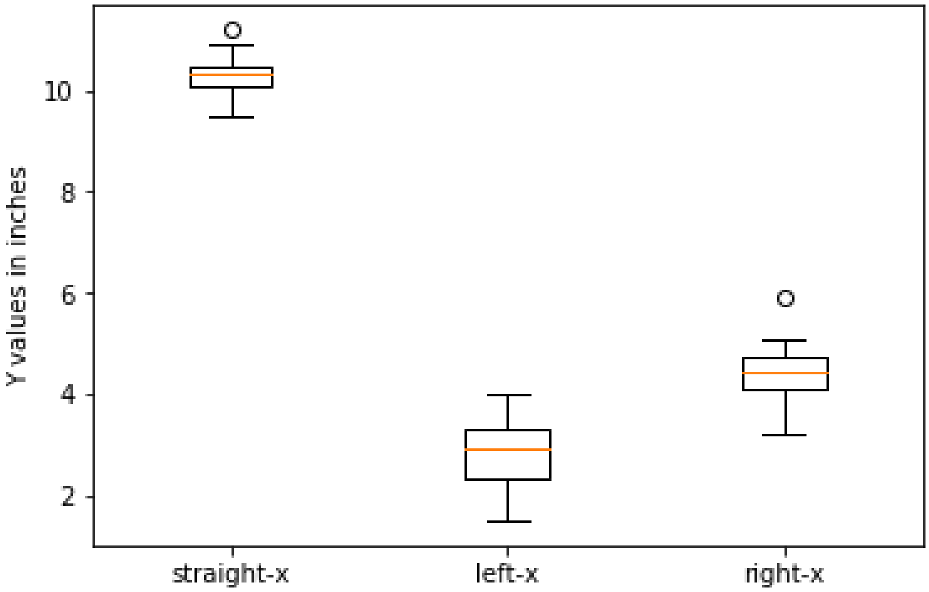

Figure 11: Box plot of X-values for straight line motion, left and right turn

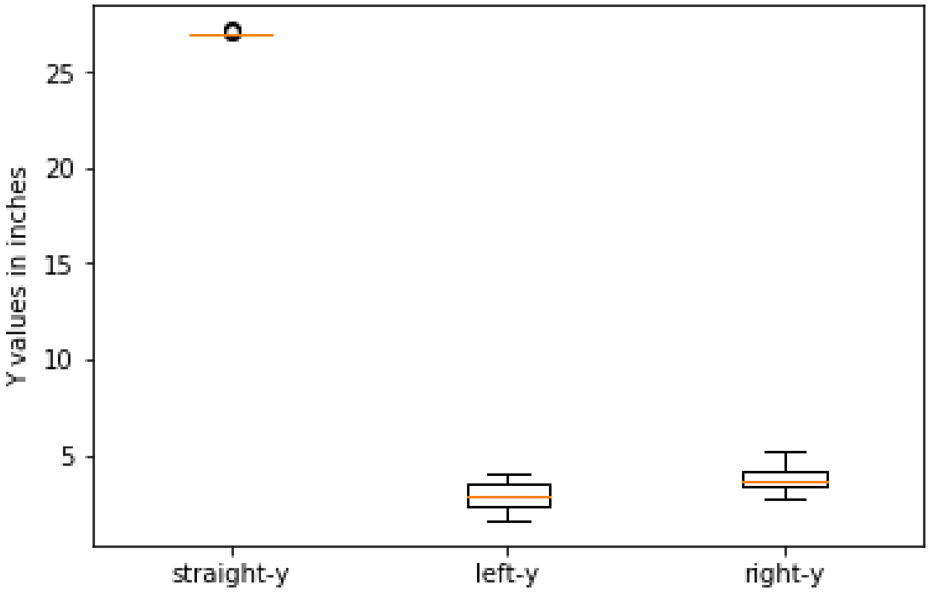

Figure 12: Box plot of Y-values for straight line motion, left and right turn

- 2.3 The chi-square test indicates that the observed data is a normal distribution.

```
In [9]: ks , p = stats.normaltest(straight_xy)
                 alpha = 1e-3
        # null hypothesis: x comes from normal distribution
                 if p.all() < alpha:
                         print("The␣null␣hypothesis␣can␣be␣rejected")
                 else:
                         print("The␣null␣hypothesis␣cannot␣be␣rejected

[Out]: The null hypothesis cannot be rejected

In [10]: ks , p = stats.normaltest(right_xy)
                 alpha = 1e-3
                 if p.all() < alpha:
                         print("The␣null␣hypothesis␣can␣be␣rejected")
                 else:
                         print("The␣null␣hypothesis␣cannot␣be␣rejected
[Out]: The null hypothesis cannot be rejected

In [11]: ks , p = stats.normaltest(left_xy)
                 alpha = 1e-3
                 if p.all() < alpha:
                         print("The␣null␣hypothesis␣can␣be␣rejected")
                 else:
                         print("The␣null␣hypothesis␣cannot␣be␣rejected
[Out]: The null hypothesis cannot be rejected
```

- Deliverable 2.2: The functions used for plotting are from the seaborn library and use a kernel density method to obtain a 2-D density plot of the x and y coordinates.

- Deliverable 2.5:

## 1.6 Straight

  - In the case of straight line motion the readings obtained for Y coordinates appear to be both precise and accurate, whereas the X coordinates appear to be precise but not very accurate.

## 1.7 Right

  - The readings obtained for Y coordinates appear to be both highly precise and accurate, whereas the X coordinates appear to be precise but not very accurate.

### 1.8 Left

- The readings obtained for X coordinates appear to have low precision and accuracy, whereas the Y coordinates appear to be precise with a low accuracy.

# 2 Appendix

## 2.1 Softwares used

- Python 2.7
- Jupyter Notebook

## 2.2 Packages used

- Matplotlib
- Numpy
- Scipy
- Seaborn
- Pylab

## 2.3 Tools used

- Onscreen ruler is used to mark scale on the sheet (metrics in inches)
- `https://www.piliapp.com/actual-size/inch-ruler/`

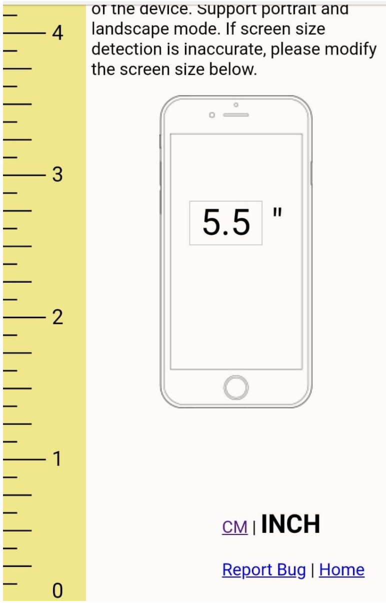
of the device. Support portrait and landscape mode. If screen size detection is inaccurate, please modify the screen size below.

Figure 13: Online ruler

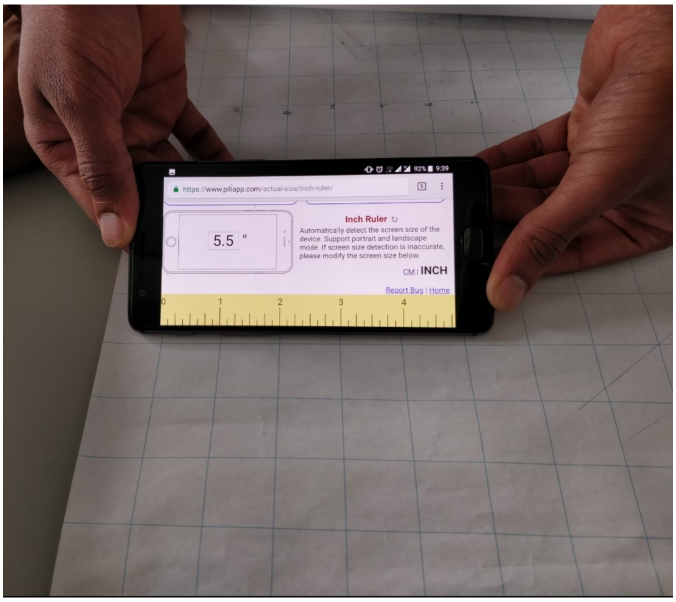
92% 9:39
https://www.piliapp.com/actual-size/inch-ruler/
Inch Ruler
Automatically detect the screen size of the device. Support portrait and landscape mode. If screen size detection is inaccurate, please modify the screen size below.
5.5 "
CM | INCH
Report Bug | Home
0
1
2
3
4

Figure 14: Marking scale on the sheet using onscreen ruler