Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Scientific Experimentation & Evaluation

Simon Deussen, Malika Navaratna, Shalaka Satheesh

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

April 2021

Contents

1

Task 1.1

1.1 Deliverables 1.1

Write a report detailing your envisaged experimental setup, expected problems and expected performance. In your report, use terminology from the lecture (i.e. measurement system, measurand, measured quantity, and so forth) to describe your experiment. Your report should cover:

1. The relevant aspects of the design of the robot, especially how you mark the stop position and how you ensure identical start positions.
2. An estimate of the expected precision of the to-be-observed data (i.e. the measurement process), including how you arrived at these estimates and why they are plausible.
3. An estimate of the propagated orientationâĂŹs uncertainty (including the upper and lower bounds) caused by the errors in the measurement process, using the method of Jacobian error propagation.

1.2 Design of Robot

- Our robot, 'Rosa' is designed based on the base design provided in the **Lego Mindstorms EV3**(evolution 3) kit (manual). Figure 1.1 shows our final design, with the measurement facility for the robot intact.
- The following are the materials used for building Rosa and hence represents the **measurement facility**:
  1. Lego EV3 kit
  2. Focusable Laser Module (2 units): It uses a working voltage of 3.0 - 5.0 V. It is powered from the USB port from the EV3 controller.
  3. Bread Board: for making connections between the laser module and the USB port
  4. Jumpers for powering the laser modules
- The robot is three-wheeled with a differential drive configuration. In this configuration (Figure 1.3, 1.4), two of the wheels placed in front of the robot on either sides, have independent actuators. The

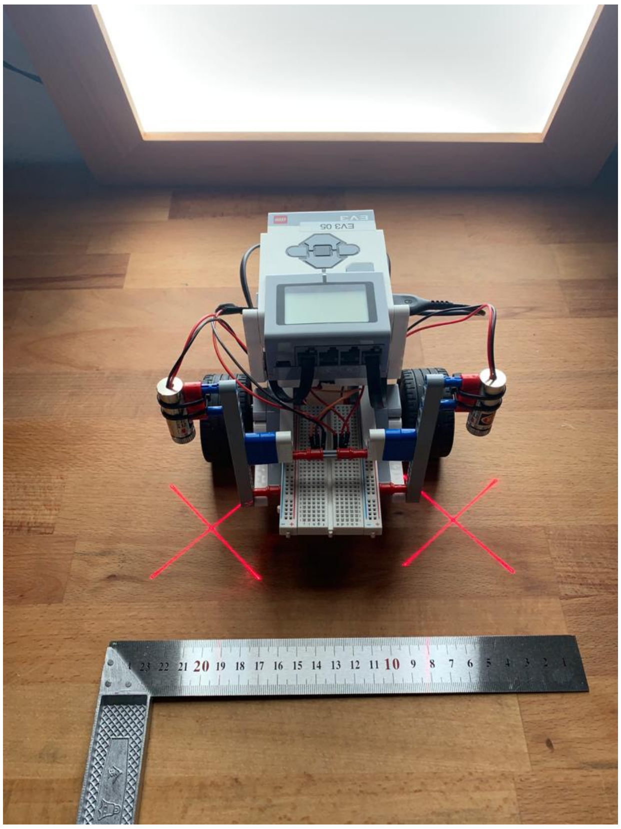


Figure 1.1: Rosa - Front View

last wheel in the back, is a castor wheel which is non-driven placed for increasing the stability of the bot.

- Commands for forward, right and left motions are made from the EV3 controller which is placed on top of the base of the bot.
- The measurements of the bot's motions are made on top a 96.8×68 cm sized grid sheet (grid size ≈ 2.5 cm).
- The measurement is done by using two focussable **laser modules** with a resolution of 1mm placed in the front of the robot. The lasers are mounted onto the lego blocks by using zip ties and tape.
- The two cross lasers provide precise marking of the pose of the robot. The aperture is 9.1 cm above ground, and 5.9 cm in front of the main axis. The distance between the two lasers is 14.8 cm.
- Since the lasers are working with 5V we are able to use the robots USB port for power supply, which makes any additional batteries unnecessary. We use a breadboard for making connections between the laser module and the USB port.

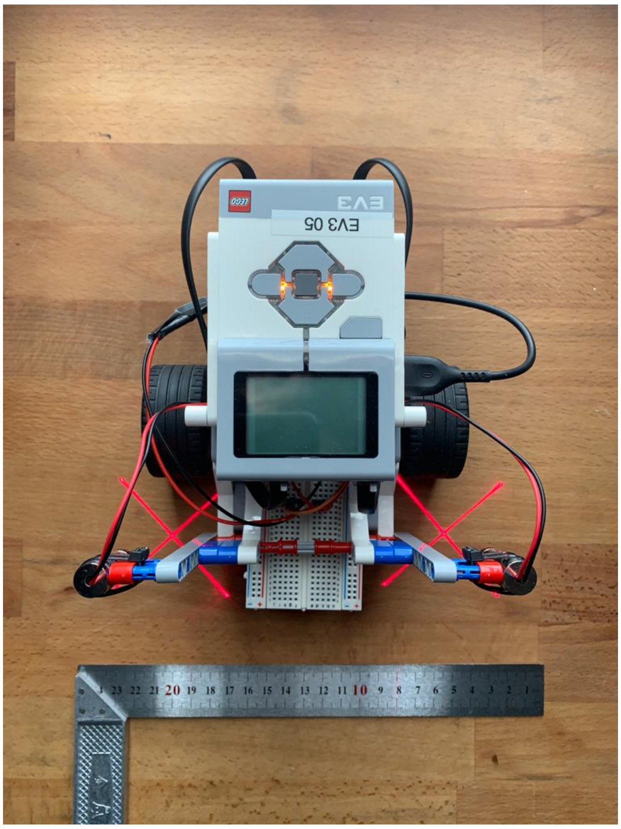


Figure 1.2: Rosa - Top View

- The **pose** of the robot, which is the **device under test (DUT)**, is described by the vector $[x, y, \theta]^T$.
- Here, the **measurand** is $[x, y]^T$ and it represents the **translation** of the robot which is measured in **millimeters**. The angle, θ represents the robot's **orientation** which is the **measurement result** obtained from the measured translation, in **degrees**.
- The Figure 1.6, 1.7 represents the distances between the two cross lasers and the center of the robot.
- Further explanations about the measurement facility is provided in section 1.3 Measurement Process
- **Ensuring identical start positions**:
  1. Securing the 96.8×68 cm sized grid sheet to a wooden plank with tape to prevent errors from changes in it's position.
  2. Marking the start points with the help of laser crosses of the resolution 1mm.
- **Marking stop positions**:

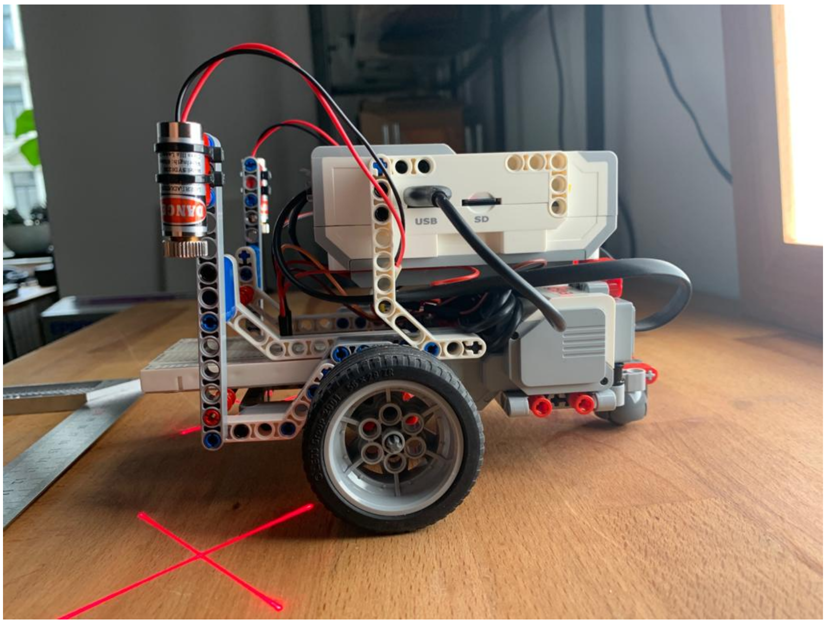


Figure 1.3: Rosa - Side View

1. The stop positions are marked with the help of the laser crosses, rulers and pencils on the grid. The mid point of the laser cross is marked with a pencil and the distance between the starting and end positions are measured using a ruler.

1.3 Measurement Process

- The cross lasers allow us to mark easily positions on the supplied grid paper.

1. The robot gets put onto the start position. For this we are using the lasers to position the robot exactly on its start position marks. For a faster positioning, we use a wooden plank on the back of the start position for guidance.
2. We start the robot's program and let it run.
3. Using a ruler we make small cross marks at both laser positions. Each cross marks get a number for later data survey.
4. The process (2 and 3) is repeated until enough data is gathered.
5. Now the cross marks needs to be measured. For this, we use a two step measurement process. We count the number of grids until the grid with the marked cross. Then, the mid point of the laser cross is marked with a pencil and the distance between the starting and end positions are measured using a ruler.
6. Each measurement is made by keeping in reference, the global coordinate system which is positioned at the centre of the axis connecting the wheel base.

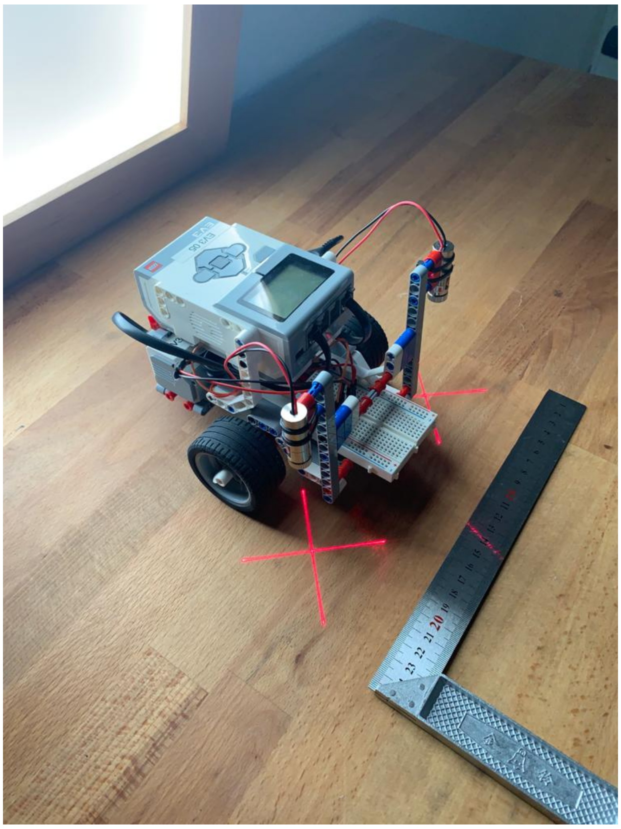


Figure 1.4: Rosa - Another Side View

- We define the coordinate system of the robot as following:

- The origin of the robot will lie on the mid point of the axial line connecting the wheels as seen on Figure 1.8

- The start position will always be set by aligning the left laser marker on a pre-marked position on the grid. The global coordinate system will also originate from this point, even when measuring the laser crosses for translation and orientation change.

- The global coordinate system originate at the initial position of the mid point of the axis connecting the robot's wheels(initial position of the robot's origin). Since the magnitude change of x and y (Δx & Δy) will be the approximately same for the two markers and the origin of the robot, to take the distance travelled by the robot origin we take the average change in x and y of the two markers. This distance will also be the x and y values with respect to the global coordinate system. The change in orientation will be found using the direction vector connecting the front and rear markers . See Figure 1.9 and equation 1.1 for details.

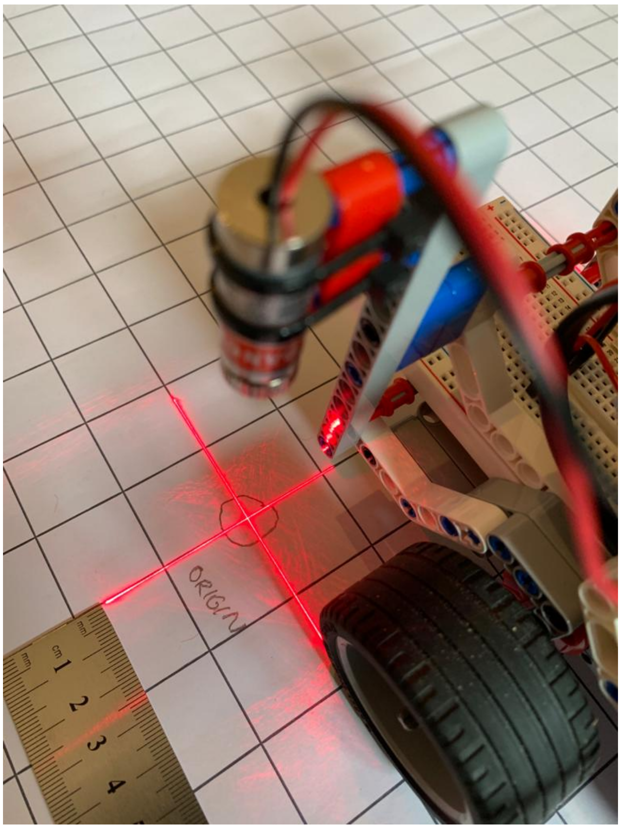
Facility.jpeg 

Figure 1.5: Rosa on the grid sheet

1.3.1 Estimate of Error

- The robot's distance measurements will be subjected to many errors. Some of them are listed below along with the rough estimate of their precision:

    1. Error when marking the laser cross due to the thickness of the laser crosses: $\pm 2mm$
    2. Parallax error when measuring the laser cross with the ruler used for our measurements: $\pm 1mm$
    3. Pressing the button pushes the robot from one position and can put extra weight on one side of the wheel: $\pm 4mm$
    4. Slippage: during multiple runs the laser markers can slightly change position: $\pm 1mm$
    5. If the grid is not held tightly there will be folds and the surface will not be flat: $\pm 1mm$
    6. Jerk: The sudden deceleration of the robot can overshoot the robot past its position specially if the momentum of the controller. This will not be same in every run.: $\pm 2mm$ (not a measurement error)

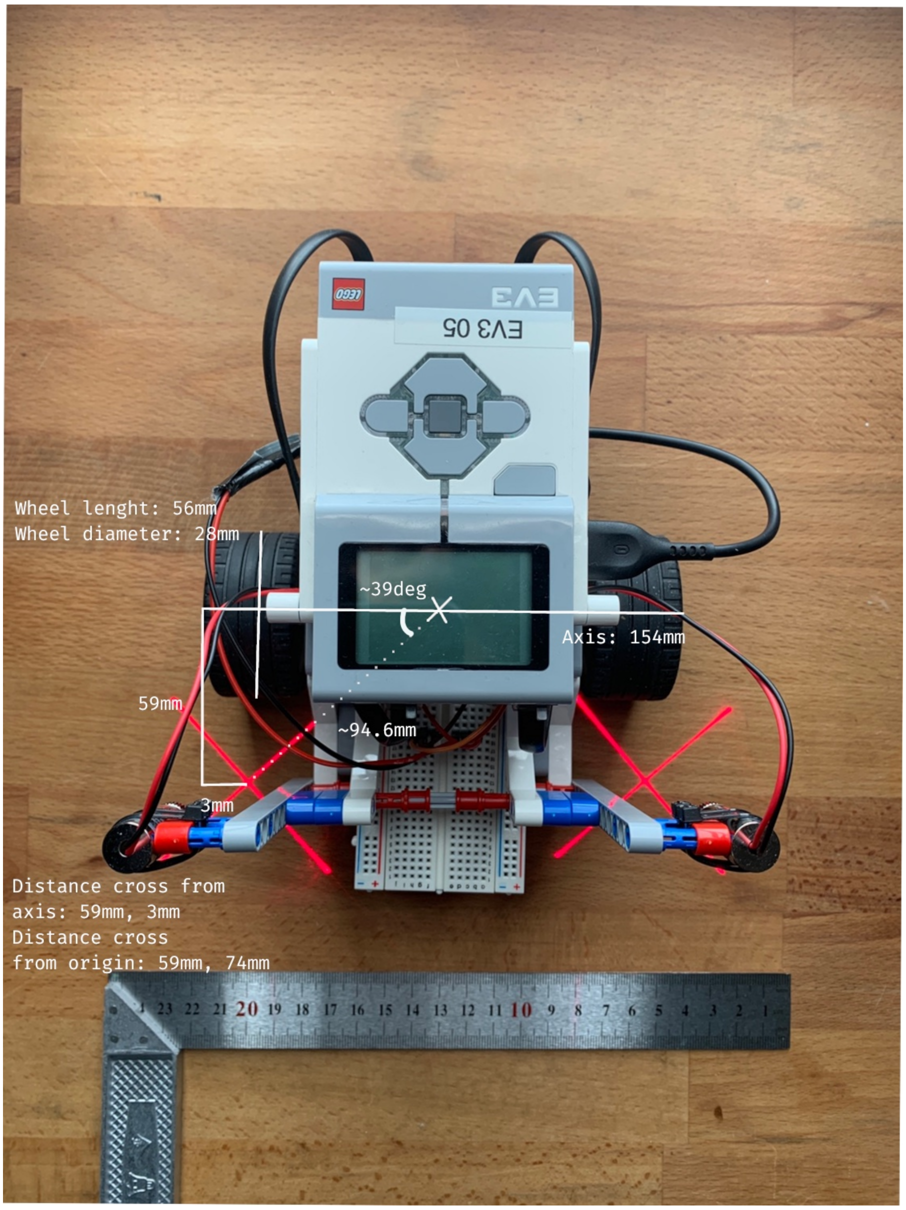


Figure 1.6: Rosa with the measurement facility

1.4 Error Propagation

- The errors mentioned above will propagate when making the distance measurements. This can be easily calculated using a Jacobian.

- Since there are 4 variables namely x_1, x_2, y_1, y_2 (refer Figure 1.9 for naming convention), the error associated to each of them contribute to the propagated error.

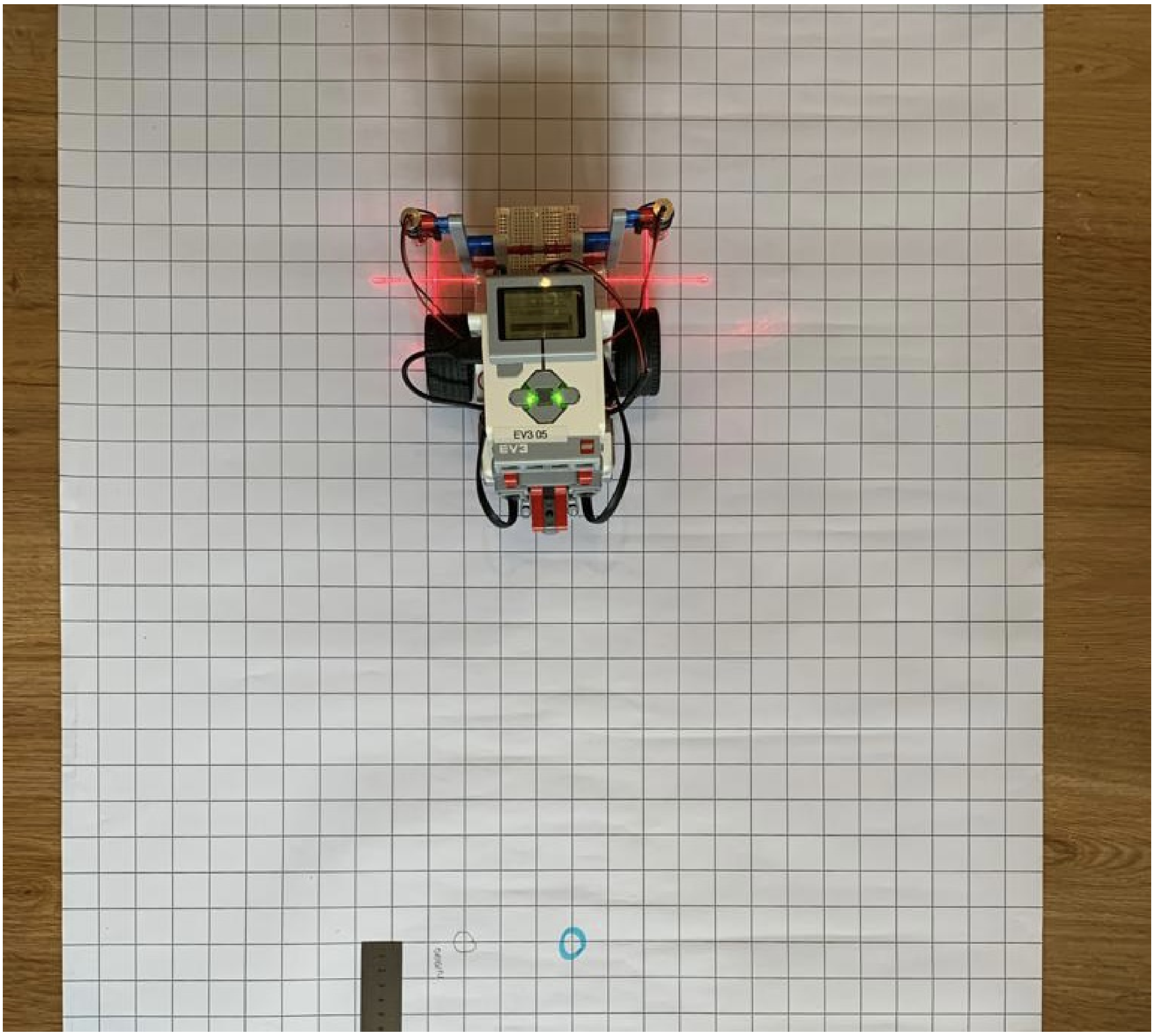


Figure 1.7: Rosa with the measurement facility 2

- The orientation is given by the equation 1.1. Since in our design the laser pointers are aligned along the x axis instead of the y axis the we calculate $\frac{\pi}{2} - \theta$ instead of directly finding θ therefore the orientation finding equation is slightly modified as below:

$$\frac{\pi}{2} - \theta = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{1.1}$$

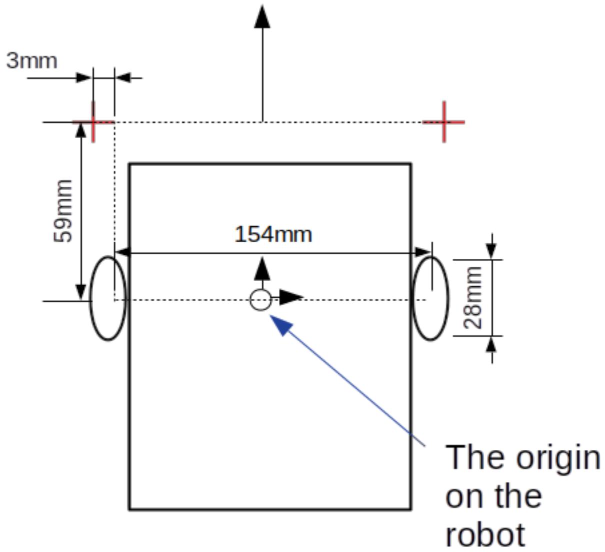


Figure 1.8: Dimensions of the robot

- If the variable matrix is defined as:

$$\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \tag{1.2}$$

- Then the Jacobian will be:

$$\begin{bmatrix} -\frac{y_1 - y_2}{(-x_1+x_2)^2+(-y_1+y_2)^2} & \frac{y_1 - y_2}{(-x_1+x_2)^2+(-y_1+y_2)^2} & -\frac{-x_1+x_2}{(-x_1+x_2)^2+(-y_1+y_2)^2} & \frac{-x_1+x_2}{(-x_1+x_2)^2+(-y_1+y_2)^2} \end{bmatrix}$$

- The upper bound and lower bound of errors were defined according to Figure 1.10 . The green arrows represent the vector in which direction the error should propagate for it to be the upper or lower bounds of the error.

- The python code given below is used to calculate the propagated error. In order to demonstrate the propagation of error, 5 runs of the straight, left and right motions were carried out.

- Note that for this example we used **only** the parallax error of measuring the laser cross with the ruler, with an estimated error of ± 1mm. Therefore, the upper bound the error values will be $[0.1, -0.1, -0.1, 0.1]$ (in centimeters) and the lower bound will be $[-0.1, 0.1, 0.1, -0.1]$ (in centimeters);

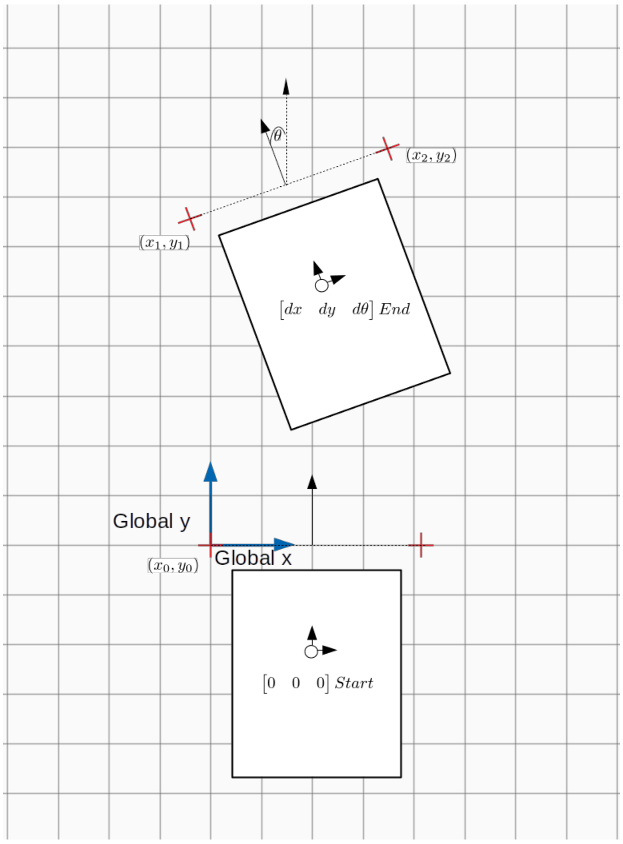


Figure 1.9: Robot orientation

where the variables are in the order of $[dx_1, dx_2, dy_1, dy_2]$.

- The figure 1.10 represent how we arrived at the order for the variables for the lower and upper bound error. We define the error vectors which will account for the direction and the magnitude of the error.

- Example:

1. For straight motion of measurements in order [x1,x2,y1,y2] [1.5 16.3 48.1 47.4]

   a. The error upper bound for straight motion is 0.7 °

   b. The error lower bound for straight motion is -0.7 °

2. For left motion of measurements in order [x1,x2,y1,y2] [-19.3 -11.2 24.6 36.8]

   a. The error upper bound for left motion is 1.1 °

   b. The error lower bound for left motion is -1.1 °

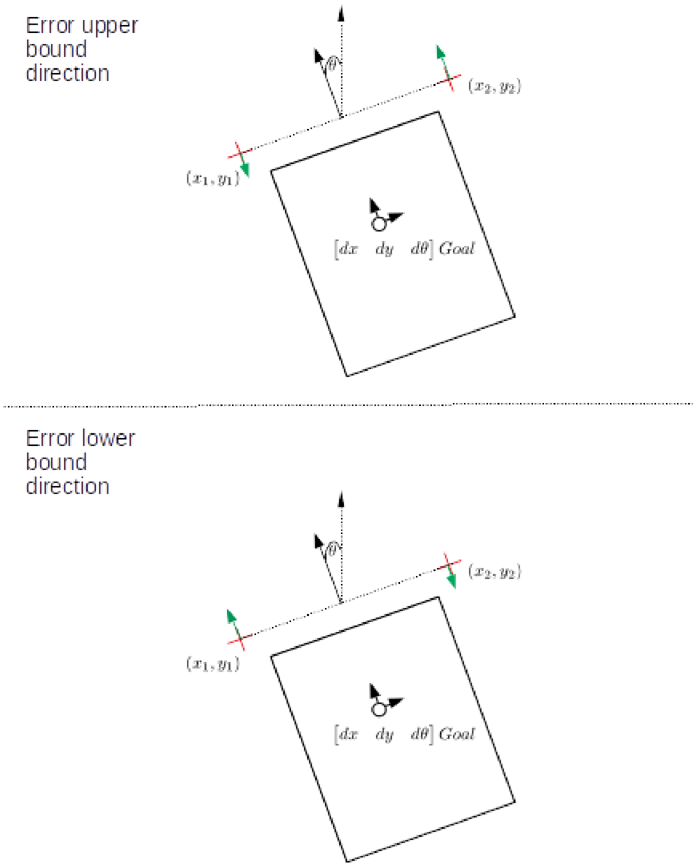


Figure 1.10: Error Bounds

3. For right motion of measurements in order [x1,x2,y1,y2] [27.4 35.3 37.1 24.5]

   a. The error upper bound for right motion is -0.3 °

   b. The error lower bound for right motion is 0.3 °

2

Task 1.2

2.1 Deliverables 1.2

Write a report detailing your execution of the experiment, including all observations made while running the robot. Especially detail and visualize the observed robot end poses and report statistical precision of the observed end poses (combined data). Summarize your findings, does the observed behaviour of the robot matches your expectations? Your report should cover:

1. The program and parameters used to drive the robot
2. Any observation made during the execution that may help one understand the outcome of the experiments (see also next week).
3. The observed data (spread-out of the manually measured end poses) as Excel or LibreOffice Calc files (stored in .csv file format and structured as presented in table 3 (section A.2)). For the data collected by the control script (using the encodersâĂŹ readings) you can take already generated files from the EV3 brick.
4. Visualization of the i) robotâĂŹs end poses (combined data from manual measurements), ii) complete robotâĂŹs paths (combined data from encoder measurements) as well as iii) visual documentation of all aspects of setup and execution you deem important. See figures 10, 11, 12 (section A.1) that show some examples on how the data can be visualized.
5. Three videos showing robotâĂŹs behaviour during the experiment (one for each type of motion)

2.2 Program & Parameters

2.3 Observations

2.3.1 Before the experiment

- Modifications were made to our initial measurement facility with the help of the pictures for the design of the measurement facility of Group 6. It consisted of 2 pencils attached with tape to hinges made of some lego blocks. After finishing the assembly several points were noted and thus some modifications were needed:

1. It was observed that tape is bad for attaching the pencils because it does not exert sufficient pressure for a straight marking. So instead of tape we used cable ties as seen in Figure 2.1, 2.2.

Figure 2.1: Attaching the pencils to Rosa

Figure 2.2: Attaching the pencils to Rosa

2. The front hinge was wobbly. During testing it freed itself from the original locking mechanism. This is why we needed to develop a guidance system, which holds the pen in place with an elastic band and guides the pen evenly to the ground during marking as seen in Figure 2.3.

Figure 2.3: Elastic bands for guidance

3. The design did not contain any help for resetting the robot to its start position. Since we did not need the lasers for marking positions any more, we attached them to the side of the robot's main corpus for easy resetting as seen in Figure 2.4, 2.5. Another thing we did, was to use a wooden plank (Figure 2.6) and some weights for creating a barrier to which we could roll the robot for a coarse positioning and the use the lasers for fast fine positioning.

Figure 2.4: Lasers for ease of resetting position

Figure 2.5: Lasers for ease of resetting position

Figure 2.6: Wooden plank for creating a barrier

4. The back hinge hold up fine, but it was a hassle to use the pen. So we attached some rails for easier one hand handling.

2.3.2 During the experiment

1. Marking the positions with this new apparatus was a bit slower than using the lasers. The 2 pens had to be put down carefully. One hand had to hold the robot in position while the other hand pressed down the pen. As seen in Figure 2.7, this created a small dot and then carefully a small cross was put on the paper next to the experiments index.

Figure 2.7: Making markings on the grid

2. With the old design we only needed to put the cross on the paper and did not need to touch the robot at all (besides during resetting the robot). This had saved some time and prevented accidentally moving the robot when using our old design.
3. For keeping track of the robot's logs easily, we modified the control script to name each run in the following pattern: [up — left-right][start_time][robot_path — both_motors].csv
4. After marking all 60 runs, we measured carefully the x and y position of each cross to a precision of 1 mm.

2.3.3 After the experiment

1. The handwritten measures are written down into .csv files for further analyses.

2.4 Data Collected

- Tables 2.1, 2.2, 2.3 show the data collected during the forward, right and left runs respectively.
- Carrying out the experiment in Task 1, we obtained a total of 9 set of measurements -
  - X axis coordinates (in cm) for the forward, left and right motions
  - Y axis coordinates (in cm) for the forward, left and right motions
  - Theta or the orientation (in degrees) for the forward, left and right motions
- Combining the measurements obtained during the experiment in Task 1 from all 6 groups, we got a total of 120 data points in each of the above mentioned sets.
- The visualisation do not include the data from group 2 as they posted their results too late.
- Of these, only group 1 had data consistent to our coordinate system. Therefore we perform the following transformations:
  - Group 2: Angles were measured from x-axis, with clockwise measurement positive. This was corrected by using: $-\theta + 90$ for the right and forward orientations. For the left orientation, this was corrected by using $\theta - 90$.
  - Group 3: Angles with clockwise measurement positive. This was corrected by using: $-\theta$.
  - Group 5: Angles with clockwise measurement positive. This was corrected by using: $-\theta$.
  - Group 6: Angles were measured from x-axis, with clockwise measurement positive. This was corrected by using: $\theta - 90$.

	X axis (cm)	Y axis(cm)	Orientation(deg)
1	0.6	45.4	-1.4
2	-1.3	45.9	1.7
3	-0.6	45.9	0.7
4	-0.6	46.6	0.7
5	0.4	45.3	-0.5
6	-0.2	44.5	1.9
7	0.4	45.5	1.2
8	0.2	44.7	0.9
9	-0.4	45.4	-0.2
10	0.1	46.2	-0.7
11	1.1	46.1	-1.2
12	1.2	46.8	0.7
13	0.8	45.5	-0.7
14	1.1	45.7	-1.2
15	-0.6	45.7	0.4
16	0.5	45.7	-0.2
17	1.4	46.8	-4.1
18	1.6	46.6	-2.2
19	-0.6	45.4	0.7
20	-0.4	46.9	1.2

Table 2.1: Measurements taken in the forward run

2.5 Visualisations

1. Looking at the Figures (2.8 2.9, 2.11, 2.15 & 2.13) which illustrate the manually measured final positions and the final positions from encoder logs, we can see that the curvature of the path found

	X axis (cm)	Y axis(cm)	Orientation(deg)
1	15.2	35.3	-57.4
2	13.3	34.3	-54.4
3	16.5	36.9	-60.8
4	14.2	35.5	-55.7
5	16.0	34.4	-51.2
6	15.0	35.9	-56.7
7	15.2	36.0	-58.7
8	15.3	35.3	-49.4
9	16.1	35.8	-58.9
10	15.4	35.9	-58.5
11	15.2	36.0	-59.8
12	15.0	35.6	-58.2
13	16.4	36.0	-58.7
14	16.7	36.1	-60.1
15	13.2	35.1	-54.5
16	14.1	34.9	-55.4
17	16.0	36.6	-59.6
18	16.0	35.9	-59.3
19	15.1	35.5	-58.0
20	15.5	36.5	-59.9

Table 2.2: Measurements taken in the right run

by encoder data is smaller than the manually measured data.

	X axis (cm)	Y axis(cm)	Orientation(deg)
1	-13.1	37.9	54.8
2	-12.3	35.9	53.3
3	-12.7	36.5	53.6
4	-12.3	36.7	53.5
5	-11.5	37.1	51.9
6	-12.6	37.0	53.3
7	-13.1	36.8	54.5
8	-11.7	36.7	51.9
9	-11.5	37.4	51.1
10	-12.1	36.4	52.9
11	-10.9	36.0	50.7
12	-12.3	37.2	52.8
13	-12.7	37.2	52.9
14	-11.8	37.5	52.1
15	-11.8	36.7	52.0
16	-12.4	38.1	53.8
17	-12.3	37.4	52.2
18	-12.8	37.8	53.5
19	-11.9	37.1	51.8
20	-12	36.8	51.8

Table 2.3: Measurements taken in the left run

2. A reason for this could be the (faulty) assumption of the mathematical formula used for the kinematic model. Specifically, the model assumes that the robot rotates and then moves in a straight line for

every time stamp it records. But in reality, the robot moves and rotates at the same time. If the robot is moving fast enough this will lead to a smaller curvature of (enocder logs') path.

Figure 2.8: Visualising the end poses from the manual measurements

Figure 2.9: Visualising the paths from Encoder Data

Figure 2.10: Visualising the end poses with manually measured data from all the groups

Figure 2.11: Manual Measurement v/s Encoder Data for Forward Run

Figure 2.12: Manual Measurement v/s Encoder Data for Forward Run - zooming in

Figure 2.13: Manual Measurement v/s Encoder Data for Right Run

Figure 2.14: Manual Measurement v/s Encoder Data for Right Run - zoomed in

Figure 2.15: Manual Measurement v/s Encoder Data for Left Run

Figure 2.16: Manual Measurement v/s Encoder Data for Left Run - zooming in

3

Task 2.0

3.1 Deliverables 2

Update your previous weekâĂŹs report and add a description of the test for a match with a and the computed motion and the actual and expected accuracy and precision. Include appropriate figures, diagrams, and images backing up any claims you make. The report must be self-contained and provide enough details to support any statement you make. If applicable, include a section on problems encountered. Your update should cover:

1. Any possible pre-processing of your data, like outlier detection and removal.
2. Fit of a Gaussian, either two individual ones in the x and y directions for each of the three cases or a proper two-dimensional distribution per case.
3. Check whether the data are actually distributed according to a Gaussian distribution.
4. List of used software, including source of any function you wrote for performing your analysis.
5. An answer to the following question: When analysing the data with respect to the executed motions, which characteristic of the data do you establish here: the accuracy, the precision, or both?
6. For presenting some of the statistical parameters that characterize the observed robot behaviour, in your report, as an example, you can use the structure defined by table 4 (section A.2).

3.2 Preprocessing of Data

- Carrying out the experiment in Task 1, we obtained a total of 9 set of measurements -
  - X axis coordinates (in cm) for the forward, left and right motions
  - Y axis coordinates (in cm) for the forward, left and right motions
  - Theta or the orientation (in degrees) for the forward, left and right motions
- Combining the measurements obtained during the experiment in Task 1 from all 6 groups, we got a total of 120 data points in each of the above mentioned sets.

Motion	Random Variable	Original Data Points	Outlier Count
Forward	X (cm)	120	6
	Y (cm)	120	3
	Orientation (degrees)	120	1
Left	X (cm)	120	0
	Y (cm)	120	1
	Orientation (degrees)	120	8
Right	X (cm)	120	5
	Y (cm)	120	2
	Orientation (degrees)	120	0

Table 3.1: Outlier Detection

- Of these, in order to use the data from all the groups, we performed some transformations mentioned in section 2.4.
- The first pre-processing step was to remove the outliers.
  - Chebyshev Theorem (Eq 3.1 and 3.2.) was used to remove the outliers. It states that "only a certain amount of data points in a probability distribution can be present from a particular distance from the mean of the distribution".

$$P(\mid X - \mu \mid \leq k\sigma) \geq (1 - \frac{1}{k^2}) \tag{3.1}$$

$$P(\mid X - \mu \mid \geq k\sigma) \leq \frac{1}{k^2} \tag{3.2}$$

  - The number of outliers detected in each set is represented in the Table 3.1. The code used for this is given below.

3.3 Fitting a Gaussian for each measurement

- The Figures **??**, **??**, **??**, **??**, **??**, **??**, **??**, **??** & **??** represent the fitting of a Gaussian to the data points after the removal of the outliers.
- The table 3.2 shows the various statistical measures computed for the manually measured data.
- Chi square test is performed in order to evaluate how well our data fits to the Gaussian distribution.
- The significance level taken for the test is 0.05. It is observed that majority of the data does not fit the Gaussian distribution.
- The accuracy is computed by comparing the mean value in each set to the true value (from the encoder logs). Further details regarding the accuracy and precision of the measurements are discussed

in section 3.6.

Motion	Random Variable	Mean	Variance (cm^2)	Accuracy (cm)	Chi Value	P-value	Null Hypothesis: Data Fits the Gaussian Distribution (Accept?)
	X (cm)	0.6	1.2	0.9	8.6	0.2	Suggest to accept
Forward	Y (cm)	46.1	0.9	0.5	5.6	0.5	Suggest to accept
	Orientation (degrees)	-1.0	2.0	0.9	8.0	0.2	Suggest to accept
	X (cm)	-16.1	2.8	0.5	5.6	0.2	Suggest to accept
Left	Y (cm)	35.6	0.9	0.9	10.2	0.1	Suggest to accept
	Orientation (degrees)	56.2	7.6	0.3	13.6	0.0002	No
	X (cm)	16.5	1.6	0.9	7.0	0.07	Suggest to accept
Right	Y (cm)	35.0	0.6	0.7	3.9	0.1	Suggest to accept
	Orientation (degrees)	-51.8	7.8	0.3	23.3	0.0001	No

Table 3.2: Statistical parameters for manual measurements

3.4 Uncertainity Ellipses after PCA

- After removing the outliers, PCA is used to further reduce noise in the data through dimensionality reduction.
- For this, data is projected into k dimensions by using an orthogonal linear transformation. Here, k is represented by the dimensions with the highest variance.
- The eigenvectors of the covariance matrix of the data is taken as the target dimensions and they are ranked according to their eigenvalues. Eigenvalues which have the highest values show the highest variance.
- The code snippet used for PCA is given below:
- Uncertainity ellipses are generally used to depict the pair-wise correlation that exists between any two given variables. If the correlation between the variables is zero, then the orientation of the error ellipse corresponds to that of the host coordinate system.
- Figures 3.4, 3.5 & 3.6 represent the uncertainity ellipses after PCA. After PCA, we see that the components are uncorrelated since we get nearly axis-aligned error ellipses.

3.5 List of software used

- All pre-processing and visualisation of of data was carried out using Python. The following libraries were used:

  1. numpy
  2. pandas
  3. scipy.stats
  4. matplotlib
  5. sklearn

3.6 Observations regarding manual measurements and encoder logs

During the process of visualising the evaluated data, we quickly discovered that the encoded data from the EV3 and the measured data did not fit together as good as we would have hoped. For each of the three movement cases we noticed mainly two modes in the data delivered by the encoder -

1. The encoder delivers data, which fits to the manually measured measurements.

OR

2. The encoder delivers data which is way off, the mean of the data points hover around 166 cm for both x and y axis.

We can compare standard error between the two different contributions:

1. Forward case: the standard error is much smaller along the x-axis (encoded: 4.9 cm, measured: 31 cm). Along the y-axis, the standard error is the same (around 28 cm for both)

2. Left case: The error along the x-axis is the same for both methods (24 cm), along the y-axis the error is smaller (encoded 17 cm, measured 25 cm)

3. Right case: Along the x-axis the encoded has an error of 38 cm and measured around 42 cm. Along the y-axis its 24 cm and 29 cm.

3.6.1 Final Thoughts

An answer to the following question: When analysing the data with respect to the executed motions, which characteristic of the data do you establish here: the accuracy, the precision, or both?

1. Since the measured data and encoded data are so far apart, we would rather look at how they are distributed by themselves than together. The raw data points are shown in Figures 3.7, 3.8 and 3.9

2. Hence it does not make much sense to look at the accuracy of measurements but rather at the precision - which takes into account the error associated with each measurement. For this, refer Figures 3.10, 3.11, 3.12 & 3.13 which represent the error ellipses for encoder data vs manually measured data.

3. Even after outlier removal from the encoder data, we see a mismatch between the two sets of measurements.

4. From the figures 3.10, 3.11, 3.12 & 3.13, it is clear that the measurements from the encoder data are more precise as they have smaller error ellipses. However, the standard deviation of the manually measured data is much higher resulting in larger ellipses and therefore we conclude that the manual measurements are less precise.

The data collection of the EV3 fails to provide data in meaningful ways. On the one hand, the data collection is unreliable, which means it randomly fails to deliver correlating to the true movements at all and on the other hand, even if the data is close to the true measurements, the distributions are completely different, see Figure 3.13.

[X-axis coordinates] [Y-axis coordinates] [Orientation]

Figure 3.1: Manually measured x, y and orientation in the forward direction

[X-axis coordinates] [Y-axis coordinates] [Orientation]

Figure 3.2: Manually measured x, y and orientation in the left direction

[X-axis coordinates] [Y-axis coordinates] [Orientation]

Figure 3.3: Manually measured x, y and orientation in the right direction

Figure 3.4: Forward Direction Measurements

Figure 3.5: Left Direction Measurements

Figure 3.6: Right Direction Measurements

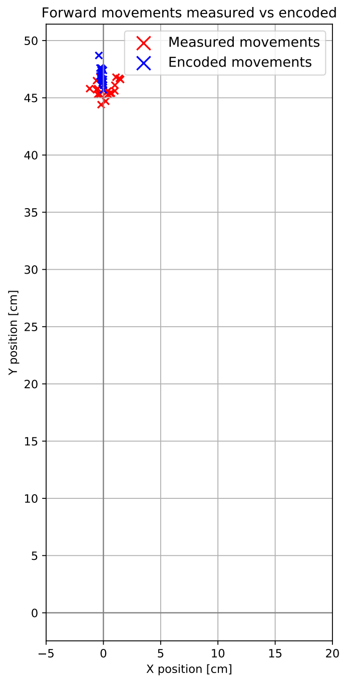


Figure 3.7: Visualising the manually measured forward poses and encoder logs

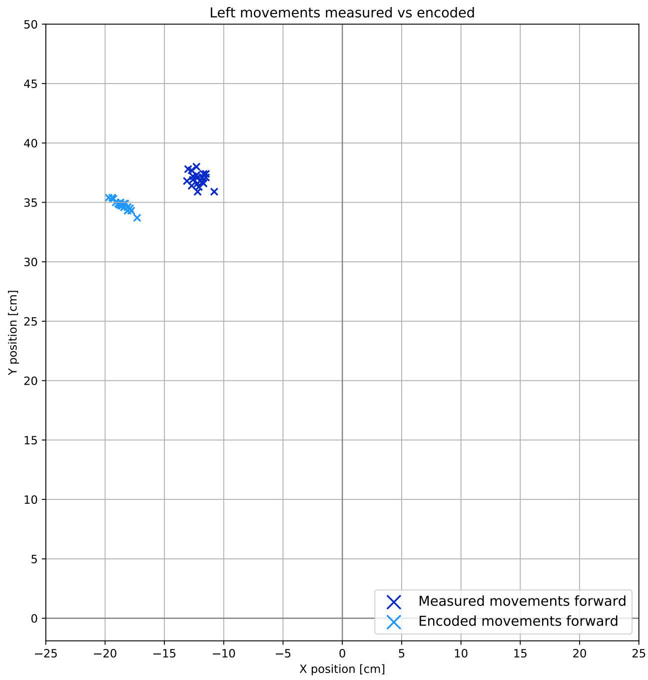


Figure 3.8: Visualising the manually measured left poses and encoder logs

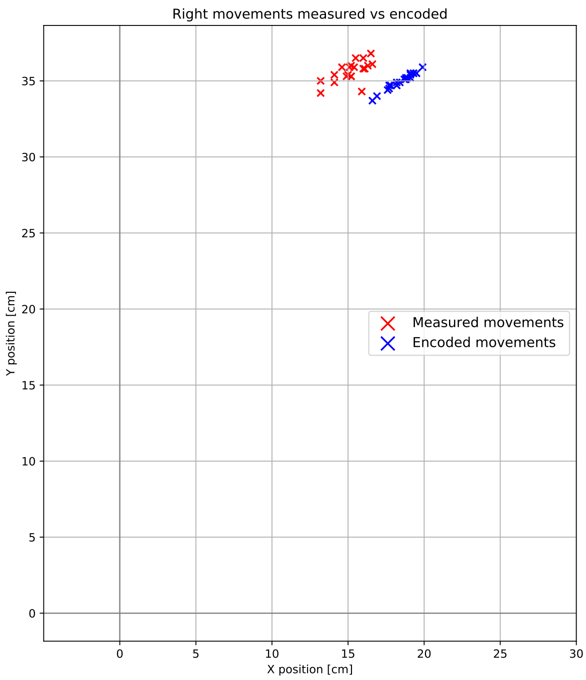


Figure 3.9: Visualising the manually measured right poses and encoder logs

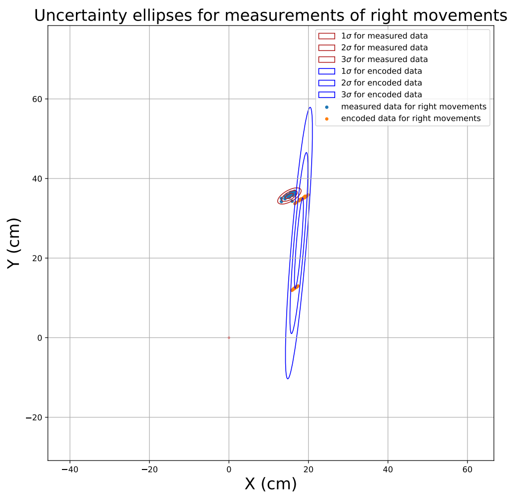


Figure 3.10: Overview of the encoder error

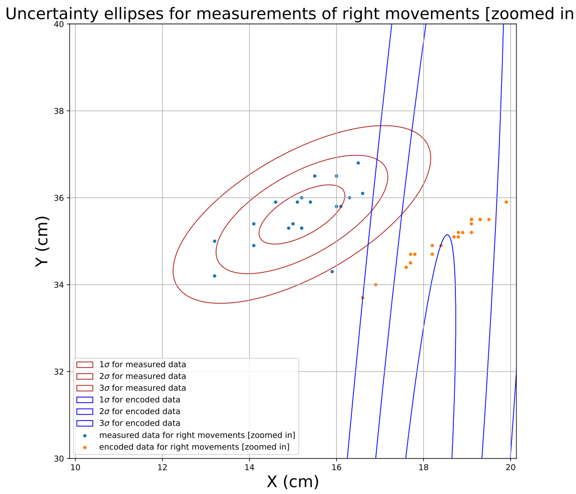


Figure 3.11: Zoomed in view of the encoder error

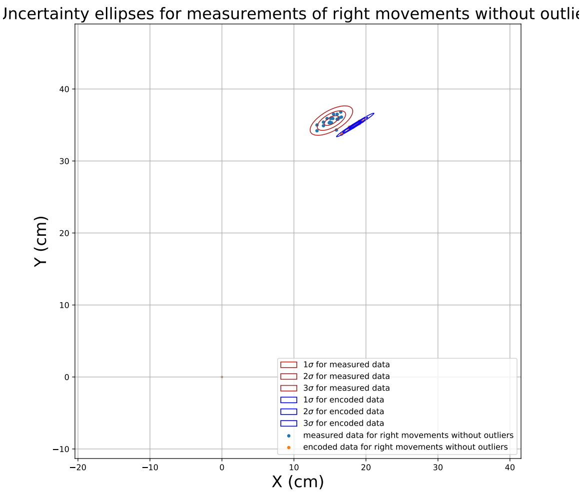


Figure 3.12: Overview of the encoder error without outliers

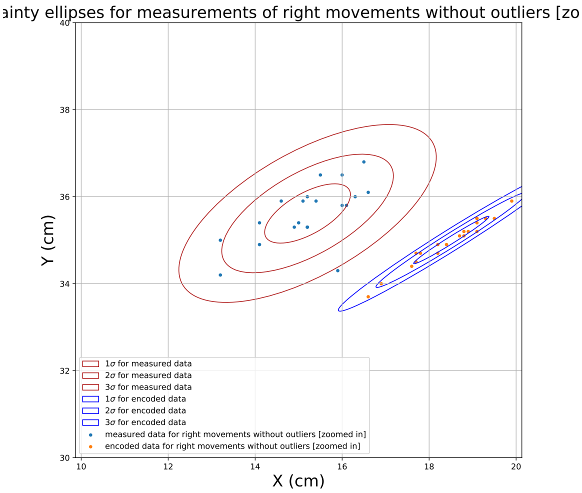


Figure 3.13: Zoomed in view of the encoder error without outliers

4

Task 4

4.1 Deliverables

Extend your report by describing the calibration process, including a theoretical part that describes the camera and lens errors measured and corrected by the calibration. The report should cover:

1. A description of the setup for calibration, including possible pitfalls
2. An estimation of the number of images and image positions required
3. A description of the intrinsic and extrinsic parameters (what do they mean?) calculated by the chosen calibration toolbox
4. Discuss possible problems or error sources that can disturb the calibration process. Include any observation you may have made while testing the proper functioning of the camera with your laptop
5. Describe the images poses used for calibration and report the found camera parameters including any error estimates (where applicable)

4.2 Description of set up

Camera calibration is carried out to estimate the intrinsic and extrinsic parameters of the camera and lens combination. Described below are the setup and the process carried out for calibration.

4.2.1 The set up

We were provided with a Basetech SC626 web camera that captures images with a 640x480 pixel for this experiment. We quickly had to realize, that the cameras own attachment was too loose and we could not fix it properly to any surfaces or tripods. Since we wanted do have it fixed in one position, we had to improvise.

Our solution was to disassemble the attachment by unscrewing it and used a countersunk m1.7 screw to fix it onto a short aluminium profile. This allowed us to use clamps for a solid attachment onto a shelve. See 4.1 for a overview and 4.2 for a zoomed in view.

Figure 4.1: Camera setup

Figure 4.2: Camera setup close up

Figure 4.3: Checkerboard pattern on the plank

Since the camera had really bad quality and was highly depended on the ambient lighting, we used parts of a full-spectrum daylight (5500k) led strip attached to another aluminium profile. This made sure that the lightning good enough for consistent pictures Figure (4.1).

As a next step, we printed the checkerboard pattern for calibration onto an A4 sheet. For stability reasons we used masking tape to fix it to a wooden plank. This allowed better handling and stopped the paper from wrinkling and other deformations, making sure the calibrations pattern lies on a single flat plane(making the assumption z=o on the surface of the pattern to be more accurate which is necessary for calibration) and will make the calibration results more accurate. see 4.3.

4.2.2 The calibration process

1. First a chessboard pattern was selected, downloaded and printed on a piece of A4 sheet paper.

2. The pattern was then pasted on top a wooden plank as show in the Figure 4.3.

3. Auto focus was disabled and the resolution was set to the highest possible in the camera(640x320 in the case of our camera).

4. Then, pictures were taken at various orientations and distances from the camera. The default image capture software on Ubuntu 18.04 called ”Cheese” was used to capture the images. Only the ones which are in focus, 26 of them were used for further processing. The references provided on the manual for practical experiments recommended minimum of 10 but we took 26 (as it was also stated in the experimentation manual)for better results and to see what kind of extreme views will make it

difficult for the algorithm to find corners. The pictures were taken so that the calibration pattern was rotated about all three of its axis.

5. Calibration target was detected and the parameters are computed. Details of which are given below.

4.3 Description of calibration parameters

- When a camera observes the 3D world it only sees the projection of the 3D world as 2D image on its sensor. The pinhole camera model given in Figure 4.4 is considered to be an good approximate for this model.

Figure 4.4: Pin Hole Camera Model [**?**]

- Using the Figure 4.4 as a model the mathematical description of the 3D world coordinates transformed to the 2D image coordinate system can be described as below

$$S\begin{bmatrix}U\\V\\1\end{bmatrix}=\begin{bmatrix}\alpha & \gamma & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

- The above model can be expressed in the compact form as given below

$$sm' = A[R|t]M'$$

- The following notations were used:
  - (X, Y, Z): Coordinates of a 3D point in the world coordinate space (measured in meters)
  - (u, v) : Coordinates of the projection on sensor in pixels (image coordinate system)
  - A : Matrix containing intrinsic parameters of camera
  - u_0, v_0 : Coordinates of principle point(usually mid point of image) in pixels. Also represented as c_x and c_y respectively
  - α, β : focal length along x and y in pixels respectively
  - γ : Skew angle between u and v axes (usually 0) in radians
  - s : scaling factor
  - m' : Augmented u,v matrix (image coordinate) in pixels
  - M' : Augmented X, Y, Z matrix(world coordinate) in meters
  - R : Rotation of image with respect to world coordinate in radians
  - t : Translation of image with respect to world coordinate in meters

- The most important point to remember is that there are three individual coordinate systems involved here.

  1. The world coordinate system represented by M' in meters
  2. The camera coordinate system found by $[R|t]M'$
  3. The image coordinate system m' found by $A[R|t]M'$

- The matrix M' contains $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ the world coordinates while matrix m' represent the $\begin{bmatrix} U & V & 1 \end{bmatrix}^T$ the image coordinates. The matrices R and T describe the transformation of world coordinate system to camera coordinate system. An multiplying the resulting matrix by A matrix will transform the world coordinates to image coordinate system

- The matrix A which represent the intrinsic parameters of the camera is unique for a given camera and lens combination setup given that the focal length does not change.

  1. The γ value which represent the skew angle between U and V axes is usually taken as 0 . (OpenCV library does this by default)
  2. The u_0 and v_0 are the principle points of the image plane a good estimate would be the center of the image frame. Given our camera has 640x480 a good estimate would be $u_0 = 320$ and $v_0 = 240$

- The value S is an arbitrary scaling factor for a given image

- Furthermore an image obtained from the camera will be distorted due to defects of the lens. These can be categorized mainly into two types

  – Tangential distortion: Tangential distortion occurs when the lens is not along its image axis (along the length with the imaging plane). This makes the image appear stretched longer or tilted distorting the perception of depth of the image. p1 and p2 are coefficients that represent the tangential distortion.

  – Radial Distortion: This is the most common type of distorting which make the image look like its bulged out or pulled in the middle . There are two types of distortions barrel(bulged out) and pincushion(pulled in the middle). This type of distortion make the straight line in real life look curved in the image. Lenses such as fish eye and other wide angle lenses have a significant radial distortion. While web cameras such as the one we use have much smaller barrel distortion $k_1 - k_6$ are radial distortion coefficients If $k_1 > 0$ barrel distortion $k_1 < 0$ pin cushion distortion

- The following equations can be used to correct for the distortion caused:
  Radial distortion corrected:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Tangential distortion corrected:

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy)]$$

$$r^2 = (\frac{x}{z})^2 + (\frac{y}{z})^2$$

- The camera calibration code follows the following steps
  1. Load the images of chessboard(also called checkerboard pattern)
  2. Define the 3D coordinates of world coordinate system starting from the surface of the board(therefore Z=0 on the board)
  3. Convert image to grayscale(make the contrast between corner colours easier to find)
  4. use findChessboardCorners() function to calculate the image coordinates(u and v) for each 3D point using the vertices of the chessboard Figure 4.5
  5. If corners were found use cornerSubPix() function to find subpixel level corner detection for more accurate results
  6. Use calibrateCamera() function which uses the Zhengyou Zhang algorithm to find the camera parameters
  7. If required use undistort() function to correct for camera distortions
- An important note is that OpenCV function calibrateCamera() return four matrices
  1. Camera Matrix: The intrinsic parameters of the camera as 3x3 Matrix
  2. Distortion coefficient : A vector with distortions parameters in the form $[k_1, k_2, p_1, p_2, k_3]$ An important note $k_4 - k_6$ are ignored in OpenCV and if the vector contains only four elements, it means that $k_3 = 0$
  3. Rotating : The rotation of each image w.r.t to the world coordinate system
  4. Translation : The translation of each image w.r.t to the world coordinate system
- The distortion coefficients do not depend on the scene viewed and intrinsic parameters will remain the same for a fixed focal length. This mean we can use these values throughout.

4.3.1 Problems during calibration: error sources

- Selection of the calibration target: The chessboard pattern was chosen since their corners are rather easier to detect and is usually invariant to lens distortion. Using square grids or circle hexagons can introduce errors due to lens distortion.

- The calibration target has to mounted on a flat surface. There will be a decrease in the accuracy of calibration if there is warping.
- Pictures taken from the same region can introduce a bias.
- Camera being on auto focus as it will cause the focal length to change. In our experiment we made sure the camera was not changing the focus
- Insufficient camera resolution. This means the accuracy of the coordinated of the corners found will less
- Since we held the calibration pattern by hand the image can move slightly while being captured which could result in some motion blur.
- The quantitative effect of camera shutter speed and lens aperture size on the accuracy of the parameters is difficult to find.
- If the borders of the calibration pattern are heavily occluded it is difficult for the algorithm to detect corners. This can be see from our test results.

4.4 Results and discussion

- The camera used for this experiment is an Basetech SC626 web cam that captures images with a 640x480 pixel resolution. It has a manually adjustable focus therefore for this experiment we left the focus constant.
- After running the code given in Calibration code the following results were obtained

Intrinsic Camera Matrix= $\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 788 & 0 & 320 \\ 0 & 824 & 220 \\ 0 & 0 & 1 \end{bmatrix}$

- One observation we can make is that from the algorithm we found $u_0 = 320$ and $v_0 = 220$ while we expected it to be $u_0 = 320$ and $v_0 = 240$ because the resolution of the camera is 640x480 pixels and the value is usually close to the mid point of the pixel values. Trusting the calibrated values will be more accurate this means the rest of the parameters should also withing acceptable levels.
- Distortion coefficients= $\begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} = \begin{bmatrix} 0.254 & -3.272 & -0.001 & 0.006 & 11.386 \end{bmatrix}$
- Having a positive $k_1 = 0.254$ we can see that there is a small barrel distortion(radial)
- An example of chessboard with its corners detected is depicted in Figure 4.5

Figure 4.5: Detected Corners

- While some extreme image orientations did not allow for corner detection. Looking at the images we can see that some of the images make it difficult to distinguish the distances between different places on the checker board. 3 out of the 19 images we used to calibrate were rejected as their corners could not be detected. Such as one given in Figure 4.6

Figure 4.6: Images corners could not be detected

- As we can see the parameters are really small and hence the distortion is small . The Figure 4.7 show an image before(left) and after(right) applying corrections for distortion

Figure 4.7: Corrected for distortion

- The best estimate for the error of these calculated parameter can be found using the re-projection error . The closer error is to o the better the results are. It is found using the L2 norm between the object points found using corner detection and the projected 2d points which were found by transformation of the 3D points to 2D projection using the rotation and translation vectors we find. The average error is found by taking the mean error.
- For our experiment the mean error is 0.3 pixels

4.4.1 Calibration code

5

Task 5

5.1 Deliverables

Run the complete experiment with a KUKA youBot arm, i.e. run all 180 experimental trials and perform any necessary preprocessing of the data. You should submit a written report covering your observations, including appropriate figures. Your report should include:

1. Any observations made during the execution that may help to understand the outcome of the experiments; for instance, are there any particular sources of error that may affect the results of the experiment?
2. A description of the pose filtering procedure you used and any observations that you may have made during the filtering process (e.g. on average, how many outliers are there per single experimental trial)
3. The saved preprocessed data as Excel or LibreOffice Calc (stored in .csv file format and structured as presented in table 3 (section A.2))
4. Combine (merge) your data with the data collected by all of your classmates
5. A visualization of the obtained final object poses (combined data). You are free to use any suitable visual representation for the data (using what you have already learned during the LEGO experiment). Hint: Given the different object-place combinations in the experiment, think about what we might want to illustrate with the visualization (e.g. the distribution of poses per object? the distribution of poses per motion direction? the distribution of all poses?)

5.2 Observations

5.2.1 Sources of error:

- Running the subscriber script before the arm has completed the placing task. The manual workflow was laborious and tedious. If interest persists, we are able to create a much better user experience.
- Shaking the table that KUKA youBot arm is place upon

- Leaving the object on the KUKA youBot arm grasped for too long this makes the heavy object slip down from original grasped position
- Not drawing the curtains while taking the readings (uneven light distribution)
- Object not placed properly within the container
- Object placed with random orientation in the container
- Placing the object in the container before the arm has reached the pre-grasp pose

5.3 Data Collection & Pose Filtering

- The placement experiment with KUKA youBot arm was carried out for 20 times in the straight, left and right directions using small, medium and large objects. Therefore each team collected a total of atleast a total of 180 readings.
- Compared to the data collected from the LEGO experiment, the data collected from this experiment was found to be more visually consistent and has fewer outliers. Tables 5.1, 5.2 & 5.3 represent the data of object poses collected by our group using the Largee, Medium and Small objects, for each direction. The coordinates, x & y were converted to centimeters from meters and the angles were converted from radians to degrees.
- The Figure 5.1 shows the camera and robot set up used for the experiment. After placing the object on the container the robot performs the motion on command. At the end of each motion the user run a python code to collect pose of the object and the

Figure 5.1: Test set up

- The world coordinate system was defined from the hand drawn coordinates as depicted in Figure 5.2

Figure 5.2: World Coordinate

- Given below in table 5.7, is a summary of the outliers found using the Chebyshev outlier detection method where we removed points outside two standard deviations. We used the data from all the groups.
- When performing outlier for removal for both end effector and object poses we removed the entire data point if either of position values(x or y coordinate) or orientation value was outside two standard deviations. The outlier removal was perfomed after taking the average of the 50 data readings per pose given by the camera system

	Straight			Right			Left		
Sl. No.	x (cm)	y (cm)	Orientation (deg)	x (cm)	y (cm)	Orientation (deg)	x (cm)	y (cm)	Orientation (deg)
1	16.40	-20.83	-92.97	-4.80	-34.55	-63.69	35.65	-27.36	-118.53
2	16.42	-20.91	-94.81	-4.80	-34.55	-63.72	35.71	-27.38	-118.77
3	16.27	-20.79	-92.8	-4.84	-34.15	-63.79	35.92	-27.44	-119.71
4	16.71	-21.06	-93.3	-4.91	-34.55	-63.69	35.5	-27.30	-118.54
5	16.41	-20.87	-92.88	-4.92	-34.35	-64.54	36.26	-27.68	-116.6
6	16.61	-21.17	-93.25	-4.95	-34.71	-62.36	35.50	-27.30	-118.54
7	16.66	-21.01	-92.10	-4.73	-34.37	-67.13	35.09	-27.13	-117.36
8	16.36	-20.84	-92.99	-4.85	-34.75	-61.17	35.6	-27.34	-118.70
9	16.42	-20.78	-94.78	36.14	-27.51	-117.96	36.05	-27.59	-116.48
10	16.45	-20.80	-94.69	-4.82	-34.28	-63.14	35.6	-27.34	-118.60
11	16.66	-21.00	-92.12	-4.88	-34.34	-64.14	35.66	-27.41	-115.79
12	16.59	-20.95	-93.87	-4.80	-34.57	-64.1	35.60	-27.35	-118.58
13	16.25	-20.81	-94.42	-4.89	-34.74	-62.77	35.53	-27.33	-118.33
14	16.51	-20.94	-94.59	-4.80	-34.55	-63.72	35.13	-27.14	-117.45
15	16.48	-20.96	-93.04	-4.84	-34.29	-62.78	35.54	-27.31	-118.65
16	16.64	-20.93	-93.77	-4.92	-34.55	-63.16	36.15	-27.63	-116.54
17	16.38	-20.84	-97.93	-4.77	-34.42	-63.93	35.41	-27.26	-118.16
18	16.42	-20.78	-94.78	-4.91	-34.56	-63.69	34.87	-27.08	-119.78
19	16.51	-20.96	-90.03	-4.79	-34.5	-63.96	35.51	-27.30	-118.54
20	16.42	-20.78	-94.78	-4.91	-34.29	-64.99	36.00	-27.55	-117.05

Table 5.1: Object pose data collected for the Large Object

	Straight			Right			Left		
Sl. No.	x (cm)	y (cm)	Orientation (deg)	x (cm)	y (cm)	Orientation (deg)	x (cm)	y (cm)	Orientation (deg)
1	16.30	-20.68	-96.57	-5.17	-34.29	-64.28	35.16	-27.31	-117.95
2	16.65	-20.43	-95.62	-5.16	-34.31	-64.39	35.88	-27.28	-115.11
3	16.66	-20.64	-97.1	-5.09	-34.13	-64.68	34.98	-27.11	-109.79
4	16.33	-20.55	-96.56	-5.20	-34.34	-61.43	35.38	-27.36	-109.64
5	16.39	-20.5	-94.66	-5.10	-34.08	-64.47	35.22	-27.30	-117.89
6	16.45	-20.66	-92.03	-5.10	-34.35	-61.34	35.22	-27.31	-117.92
7	16.61	-20.80	-93.60	-5.14	-34.09	-66.62	35.70	-27.30	-107.1
8	16.40	-20.61	-94.15	-5.10	-34.10	-64.48	35.81	-26.94	-118.95
9	16.33	-20.56	-92.83	-5.14	-34.39	-62.82	35.61	-26.97	-123.67
10	16.53	-20.43	-93.4	-5.22	-34.59	-60.79	35.13	-27.29	-117.97
11	16.30	-20.28	-101.92	-5.11	-34.15	-64.38	35.69	-27.03	-122.76
12	16.46	-21.09	-96.03	-5.15	-34.52	-61.48	35.40	-27.21	-117.78
13	16.14	-20.24	-93.65	-5.11	-34.23	-64.46	35.25	-27.29	-117.96
14	16.62	-21.07	-93.74	-5.14	-34.38	-66.50	35.64	-27.43	-116.44
15	16.35	-20.27	-96.50	-5.21	-34.64	-63.14	35.81	-27.31	-115.72
16	16.78	-20.70	-96.82	-5.17	-34.61	-64.46	35.24	-27.17	-117.57
17	16.14	-20.76	-93.42	-5.19	-34.56	-62.85	36.06	-27.53	-113.89
18	16.31	-20.74	-94.54	-5.15	-34.39	-64.38	34.86	-27.18	-116.65
19	16.58	-20.78	-93.45	-5.17	-34.27	-61.37	36.05	-27.52	-113.91
20	16.32	-20.50	-91.11	-5.13	-33.66	-61.66	35.40	-27.36	-117.85

Table 5.2: Object pose data collected for the Medium Object

	Straight			Right			Left		
Sl. No.	x (cm)	y (cm)	Orientation (deg)	x (cm)	y (cm)	Orientation (deg)	x (cm)	y (cm)	Orientation (deg)
1	16.16	-21.30	-93.49	-5.75	-34.61	-64.15	34.77	-27.50	-117.85
2	15.78	-21.14	-92.68	-5.68	-34.82	-64.08	34.85	-27.43	-117.39
3	15.62	-20.94	-93.23	-5.74	-34.63	-63.69	34.55	-27.44	-117.41
4	16.16	-21.3	-93.49	-5.74	-34.95	-60.28	34.83	-27.49	-118.01
5	15.78	-21.14	-92.68	-5.62	-34.68	-63.04	34.74	-27.53	-117.97
6	15.74	-20.99	-92.79	-5.71	-35.01	-63.85	36.04	-28.00	-117.97
7	15.54	-20.91	-93.55	-5.81	-34.89	-63.82	35.20	-27.57	-119.44
8	15.78	-21.14	-92.68	-5.73	-34.99	-60.13	34.59	-27.41	-117.49
9	16.16	-21.30	-93.49	-5.74	-34.93	-60.52	34.29	-27.30	-116.73
10	15.96	-21.22	-93.06	-5.74	-34.76	-62.33	35.56	-27.76	-116.72
11	15.73	-21.03	-94.16	-5.73	-34.76	-62.26	34.74	-27.53	-117.97
12	16.00	-21.23	-93.15	-5.76	-35.08	-58.21	34.74	-27.53	-117.97
13	16.16	-21.30	-93.49	-5.76	-35.08	-58.30	35.28	-27.66	-114.62
14	15.78	-21.14	-92.68	-5.74	-34.58	-63.87	36.04	-28.00	-117.97
15	15.78	-21.14	-92.68	-5.74	-34.59	-64.14	34.38	-27.35	-116.93
16	15.79	-21.14	-92.70	-5.66	-35.00	-62.4	34.98	-27.57	-118.4
17	15.75	-21.05	-94.25	-5.76	-34.67	-63.89	35.13	-27.64	-118.76
18	15.78	-21.06	-94.33	-5.73	-34.70	-62.50	35.53	-27.75	-117.51
19	15.83	-21.08	-94.37	-5.73	-35.02	-61.87	34.74	-27.53	-117.97
20	15.58	-20.93	-93.73	-5.62	-34.72	-62.57	35.00	-27.50	-117.73

Table 5.3: Object pose data collected for the Small Object

SI No	**Straight**			**Left**			**Right**		
	x(cm)	y(cm)	Orientation(deg)	x(cm)	y(cm)	Orientation(deg)	x(cm)	y(cm)	Orientation(deg)
1	17.31	-29.99	-5.29	33.48	-36.57	-28.15	-0.57	-41.24	20.34
2	17.31	-29.99	-5.29	33.48	-36.57	-28.15	-0.57	-41.20	20.59
3	17.41	-30.08	-4.59	33.45	-36.56	-28.56	-0.57	-41.20	20.59
4	17.35	-30.03	-5.10	33.48	-36.57	-28.15	-0.57	-41.54	19.30
5	17.31	-29.99	-5.29	33.47	-36.55	-28.21	-0.57	-41.21	20.57
6	17.32	-29.99	-5.25	33.48	-36.57	-28.16	-0.57	-41.26	20.27
7	17.31	-29.99	-5.29	33.48	-36.57	-28.15	-0.57	-41.22	20.55
8	17.31	-29.99	-5.29	33.48	-36.57	-28.15	-0.56	-41.24	20.54
9	17.31	-29.99	-5.29	33.48	-36.56	-28.19	-0.57	-41.21	20.57
10	17.31	-29.99	-5.29	33.48	-36.57	-28.15	-0.56	-41.25	20.52
11	17.31	-29.99	-5.29	33.41	-36.52	-28.31	-0.57	-41.21	20.57
12	17.34	-30.02	-5.14	33.48	-36.57	-28.15	-0.57	-41.26	20.22
13	17.31	-29.99	-5.29	33.44	-36.56	-28.56	-0.56	-41.24	20.54
14	17.31	-29.99	-5.29	33.48	-36.57	-28.15	-0.57	-41.31	20.06
15	17.31	-29.99	-5.29	33.46	-36.56	-28.41	-0.56	-41.30	20.45
16	17.58	-30.25	-3.46	33.48	-36.57	-28.15	-0.57	-41.21	20.57
17	17.31	-29.99	-5.29	33.48	-36.56	-28.21	-0.57	-41.25	20.35
18	17.52	-30.20	-4.08	33.46	-36.56	-28.41	-0.57	-41.39	19.33
19	17.31	-29.99	-5.29	33.46	-36.55	-28.29	-0.56	-41.24	20.49
20	17.31	-29.99	-5.29	33.44	-36.56	-28.60	-0.56	-41.27	20.50

Table 5.4: End effector pose data collected for the Large Object

SI No	Straight			Left			Right		
	x(cm)	y(cm)	Orientation(deg)	x(cm)	y(cm)	Orientation(deg)	x(cm)	y(cm)	Orientation(deg)
1	17.37	-30.02	-4.52	33.48	-36.57	-28.15	-0.56	-41.31	20.44
2	17.37	-30.02	-4.45	33.48	-36.57	-28.15	-0.57	-41.22	20.55
3	17.38	-30.02	-4.58	33.48	-36.57	-28.15	-0.56	-41.28	20.47
4	17.37	-30.02	-4.51	33.48	-36.57	-28.15	-0.57	-41.22	20.55
5	17.37	-30.02	-4.49	33.48	-36.57	-28.15	-0.56	-41.32	20.42
6	17.35	-30.01	-4.78	33.48	-36.56	-28.20	-0.56	-41.33	20.40
7	17.38	-30.03	-4.40	33.48	-36.57	-28.15	-0.56	-41.27	20.40
8	17.37	-30.03	-4.41	33.48	-36.57	-28.15	-0.57	-41.22	20.52
9	17.37	-30.03	-4.41	33.48	-36.57	-28.15	-0.56	-41.30	20.45
10	17.37	-30.02	-4.51	33.48	-36.56	-28.19	-0.56	-41.28	20.47
11	17.38	-30.03	-4.38	33.47	-36.56	-28.29	-0.56	-41.28	20.47
12	17.38	-30.03	-4.40	33.48	-36.57	-28.15	-0.57	-41.22	20.55
13	17.36	-30.02	-4.56	33.48	-36.57	-28.15	-0.56	-41.24	20.54
14	17.37	-30.03	-4.43	33.47	-36.56	-28.26	-0.53	-40.20	20.47
15	17.38	-30.03	-4.38	33.48	-36.57	-28.15	-0.56	-41.25	20.52
16	17.36	-30.02	-4.56	33.48	-36.57	-28.15	-0.56	-41.31	20.44
17	17.38	-30.03	-4.38	33.48	-36.57	-28.16	-0.56	-41.30	20.45
18	17.36	-30.02	-4.54	33.48	-36.57	-28.15	-0.56	-41.26	20.50
19	17.38	-30.03	-4.38	33.48	-36.57	-28.15	-0.56	-41.27	20.49
20	17.38	-30.03	-4.38	33.48	-36.57	-28.15	-0.56	-41.24	20.54

Table 5.5: End effector pose data collected for the Medium Object

	Straight			Left			Right		
	x(cm)	y(cm)	Orientation(deg)	x(cm)	y(cm)	Orientation(deg)	x(cm)	y(cm)	Orientation(deg)
1	17.64	-30.40	-3.31	32.61	-35.99	-29.67	-0.69	-41.21	20.55
2	17.54	-30.28	-4.15	33.31	-36.51	-28.83	-0.68	-41.42	20.72
3	17.49	-30.23	-4.51	33.55	-35.19	-28.74	-0.68	-41.42	20.72
4	17.60	-30.34	-3.66	33.41	-36.58	-28.71	-0.68	-41.26	20.59
5	17.36	-30.08	-5.56	32.56	-35.95	-29.73	-0.68	-41.26	20.59
6	17.62	-30.37	-3.52	33.39	-36.57	-28.73	-0.69	-41.21	20.55
7	17.65	-30.40	-3.22	32.61	-35.99	-29.67	-0.69	-41.21	20.55
8	17.64	-30.41	-3.00	32.56	-35.95	-29.73	-0.68	-41.39	20.69
9	17.60	-30.35	-3.62	33.41	-36.58	-28.71	-0.69	-41.21	20.55
10	17.33	-30.04	-5.82	33.41	-36.58	-28.71	-0.69	-41.21	20.55
11	17.64	-30.40	-3.31	33.41	-36.58	-28.71	-0.69	-41.21	20.55
12	17.64	-30.39	-3.36	33.41	-36.58	-28.71	-0.69	-41.21	20.55
13	17.48	-30.40	-4.25	32.57	-35.97	-29.71	-0.68	-41.22	20.56
14	17.55	-30.29	-4.04	33.31	-36.51	-28.83	-0.68	-41.27	20.59
15	17.65	-30.40	-3.17	33.41	-36.58	-28.71	-0.69	-41.21	20.55
16	17.49	-30.23	-4.51	33.41	-36.58	-28.71	-0.68	-41.42	20.72
17	17.42	-30.15	-5.09	33.41	-36.58	-28.71	-0.68	-41.42	20.71
18	17.64	-30.41	-3.07	33.39	-36.57	-28.73	-0.68	-41.33	20.65
19	17.45	-30.18	-4.83	32.73	-36.08	-29.52	-0.69	-41.21	20.55
20	17.60	-30.35	-3.62	32.56	-35.95	-29.73	-0.68	-41.32	20.64

Table 5.6: End effector pose data collected for the Small Object

Size	Motion	Original data points	Outliers
Large	Straight	123	8
	Left	123	7
	Right	126	3
Medium	Straight	123	11
	Left	123	11
	Right	124	12
Small	Straight	127	6
	Left	123	6
	Right	123	12

Table 5.7: Outliers detected in three motion types from combined group data

5.4 Visualisation

- The figures 5.3, 5.4, 5.7, 5.8, 5.5 & 5.6 represent the collective end poses of the Large, Medium and Small objects in the straight, right and left directions respectively.
- It was observed that the number of outliers is minimal. The most obvious outlier (Figure 5.7) was observed for the right motion.
- Visually, the measurements from the left motion seem the most precise.
- Visually the large object seems be more precise.(An quantitative evalution will be done in the next report). This could be accounted for the fact that its larger inertia will make it less susceptible to table shakes.

Figure 5.3: Straight motion before removal of outliers(Object Pose)

Figure 5.4: Straight motion after removal of outliers(Object Pose)

Figure 5.5: Left motion before removal of outliers(Object Pose)

Figure 5.6: Left motion after removal of outliers(Object Pose)

Figure 5.7: Right motion before removal of outliers(Object Pose)

Figure 5.8: Right motion after removal of outliers(Object Pose)

Figure 5.9: Straight motion object pose of all sizes

Figure 5.10: Left motion object pose of all sizes

Figure 5.11: Right motion object pose of all sizes

Figure 5.12: Straight Motion End effector poses and trajectory

Figure 5.13: Left Motion End effector poses and trajectory

Figure 5.14: Right Motion End effector poses and trajectory

6

Task 6

6.1 Deliverables

1. If necessary, rerun your experiment according to the feedback in class and describe how the new run improves on the previous run
2. Find suitable statistical techniques for analyzing the experimental results (using what youâĂŹve learned during the LEGO experiment) and answer the list of questions mentioned above.
3. For the case of statistical significance tests (effects of different object and pose variations), you can use an F-test to perform the statistical analysis (e.g. comparing pair-wise variance).
4. For presenting the statistical parameters that characterize the observed robot behaviour, in your report, as an example, you can use the structure defined by table 5 (section A.2)
5. Mention the list of used software and include the source of any functions you wrote for performing your analysis

6.2 Statistical analysis

- Data collected from all the groups were subjected to outlier removal as mentioned in section of chapter 5.
-

6.2.1 Accuracy and precision

6.3 Statistical Significance Tests

6.3.1 F-test

- Right tailed F-test is carried out between different measurements obtained from different sizes of objects for each direction.

Size	Direction	Random var.	Mean	Variance	Accuracy	Precision	Chi value	P value	Null hypothesis
large	left	x	35.532	0.196	0.990	4.854	26.678	0.001	Reject
large	left	y	-27.343	0.053	0.992	5.626	42.512	0.000	Reject
large	left	theta	-118.611	4.401	0.986	6.640	150.187	0.000	Reject
large	right	x	-4.903	0.004	0.891	5.442	8.955	0.176	Suggest to accept
large	right	y	-34.396	0.035	0.995	6.589	14.078	0.015	Reject
large	right	theta	-64.101	2.512	0.956	4.663	15.550	0.016	Reject
large	straight	x	16.421	0.025	0.992	5.017	6.799	0.340	Suggest to accept
large	straight	y	-20.883	0.009	0.986	6.852	23.240	0.001	Reject
large	straight	theta	-93.731	1.756	0.988	5.207	1.419	0.701	Suggest to accept
medium	left	x	35.724	0.166	0.990	4.689	7.906	0.162	Suggest to accept
medium	left	y	-27.157	0.031	0.995	5.184	3.806	0.578	Suggest to accept
medium	left	theta	-119.803	8.942	0.980	4.658	24.765	0.000	Reject
medium	right	x	-5.107	0.007	0.928	4.075	5.854	0.210	Suggest to accept
medium	right	y	-34.169	0.038	0.994	4.689	9.338	0.155	Suggest to accept
medium	right	theta	-64.494	1.349	0.962	4.786	3.536	0.472	Suggest to accept
medium	straight	x	16.537	0.038	0.988	4.439	2.114	0.833	Suggest to accept
medium	straight	y	-20.658	0.024	0.994	5.925	15.719	0.008	Reject
medium	straight	theta	-94.410	5.533	0.980	5.578	19.730	0.001	Reject
small	left	x	35.236	0.163	0.987	4.332	6.422	0.093	Suggest to accept
small	left	y	-27.489	0.028	0.989	4.521	16.845	0.005	Reject
small	left	theta	-117.703	0.890	0.987	5.449	35.687	0.000	Reject
small	right	x	-5.801	0.004	0.945	4.661	25.094	0.000	Reject
small	right	y	-34.562	0.030	0.992	4.597	7.294	0.295	Suggest to accept
small	right	theta	-63.572	4.222	0.948	3.849	49.805	0.000	Reject
small	straight	x	15.915	0.037	0.970	3.698	13.483	0.004	Reject
small	straight	y	-21.057	0.019	0.978	3.896	59.698	0.000	Reject
small	straight	theta	-93.257	0.590	0.993	4.725	6.119	0.190	Suggest to accept
all	straight	x	16.278	0.111	0.983	4.172	66.493	0.000	Reject
all	straight	y	-20.880	0.041	0.986	4.593	21.904	0.025	Reject
all	straight	theta	-93.638	2.584	0.988	7.316	119.031	0.000	Reject
all	right	x	-5.266	0.150	0.922	3.172	204.180	0.000	Reject
all	right	y	-34.370	0.071	0.994	5.889	48.107	0.000	Reject
all	right	theta	-63.938	3.367	0.953	6.284	175.331	0.000	Reject
all	left	x	35.532	0.196	0.990	4.854	26.678	0.001	Reject
all	left	y	-27.343	0.053	0.992	5.626	42.512	0.000	Reject
all	left	theta	-118.611	4.401	0.986	6.640	150.187	0.000	Reject

Sl. No.	Direction	Object Size	Object Size	Random Variable	F-Value	P-Value	Null Hypothesis
0	Straight	small	medium	x(in cm)	1.02	0.54	Suggest to accept
1	Straight	small	medium	y(in cm)	1.30	0.92	Suggest to accept
2	Straight	small	medium	Orientation (in deg)	9.37	1.00	Suggest to accept
3	Straight	small	large	x(in cm)	1.49	0.98	Suggest to accept
4	Straight	small	large	y(in cm)	2.06	1.00	Suggest to accept
5	Straight	small	large	Orientation (in deg)	2.97	1.00	Suggest to accept
6	Straight	medium	large	x(in cm)	1.51	0.99	Suggest to accept
7	Straight	medium	large	y(in cm)	2.68	1.00	Suggest to accept
8	Straight	medium	large	Orientation (in deg)	3.15	1.00	Suggest to accept
9	Right	small	medium	x(in cm)	2.04	1.00	Suggest to accept
10	Right	small	medium	y(in cm)	1.27	0.90	Suggest to accept
11	Right	small	medium	Orientation (in deg)	3.13	1.00	Suggest to accept
12	Right	small	large	x(in cm)	1.02	0.54	Suggest to accept
13	Right	small	large	y(in cm)	1.17	0.80	Suggest to accept
14	Right	small	large	Orientation (in deg)	1.68	1.00	Suggest to accept
15	Right	medium	large	x(in cm)	2.00	1.00	Suggest to accept
16	Right	medium	large	y(in cm)	1.09	0.67	Suggest to accept
17	Right	medium	large	Orientation (in deg)	1.86	1.00	Suggest to accept
18	Left	small	medium	x(in cm)	1.02	0.53	Suggest to accept
19	Left	small	medium	y(in cm)	1.11	0.72	Suggest to accept
20	Left	small	medium	Orientation (in deg)	10.05	1.00	Suggest to accept
21	Left	small	large	x(in cm)	1.21	0.90	Suggest to accept
22	Left	small	large	y(in cm)	1.87	1.00	Suggest to accept
23	Left	small	large	Orientation (in deg)	4.97	1.00	Suggest to accept
24	Left	medium	large	x(in cm)	1.19	0.88	Suggest to accept
25	Left	medium	large	y(in cm)	1.68	1.00	Suggest to accept
26	Left	medium	large	Orientation (in deg)	2.02	1.00	Suggest to accept

Table 6.1: Results of F-test

- The null hypothesis is that the variance of both the chosen variables are equal.
- In our experiment, we choose a significance level of 0.05. Thus, we suggest to accept the null hypothesis if the obtained p-value is greater than 0.05 and reject it otherwise.
- Table 6.1 shows the results of the test.
- The code used for this is given below: