

SEE REPORT WS2025 - Group 1

Fabian Huppertz, Shahzaib Waseem, Hari Krishna

Contents

I	Manual Motion Observation	1
I-A	Design of Experiment	1
I-B	Estimates of the expected Precision .	3
I-C	Jacobian Error Propagation	3
II	Robot Path Experiment	4
II-A	Experiment Setup	4
II-B	Experiment Observations and Hurdles	4
II-C	Raw Data from Measurements	4
II-D	Visualisation and Images	4
III	Statistical Analysis of the Exeriment Data	5
III-A	Chebishov and χ^2	5
III-B	Statistical Parameters	6
III-C	PCA and Ellipsoid Analysis	6
IV	Calibrating an Optical Tracking System	7
IV-A	Calibration Setup	7
IV-B	OpenCV Camera Parameters	8
IV-C	Experiment Observations	8
IV-D	Observations	8
	IV-D1 Camera Functionality Check	8
	IV-D2 Environmental Conditions .	8
	IV-D3 Image Capture Process .	8
	IV-D4 Observed Problems and Error Sources	8
	IV-D5 Reflections and Improvements	8
IV-E	Results	8
V	Measuring the Accuracy and Precision of KUKA youBot	9
V-A	Experiment Setup	9
V-B	Observations and Sources of Error . .	9
V-C	Results	9
VI	Measuring the Accuracy and Precision of a KUKA youBot Arm	10
VI-A	Statistical Analysis	10
VI-B	χ^2 Test for Goodness of Fit	10
VII	Chi-Squared Goodness-of-Fit Analysis	10

I. Manual Motion Observation

The goal of this experiment is to design a differential drive robot using LEGO EV3 and then measure its pose variation for three different constant velocity motions: an arc to the left, straight line ahead, and an arc to the right.

A. Design of Experiment

A differential drive robot is a mobile robot that moves using two independently driven wheels mounted on either side of its chassis. By varying the relative speed and direction of these wheels, the robot can move forward, backward, or rotate in place, making it highly maneuverable. This design is simple, cost-effective, and widely used in research and industry for tasks such as navigation, mapping, and control experiments.

The linear and angular velocities of a differential drive robot are given by Equations 1 and 2

$$v = \frac{r}{2} (\omega_r + \omega_l) \quad (1)$$

$$\omega = \frac{r}{L} (\omega_r - \omega_l), \quad (2)$$

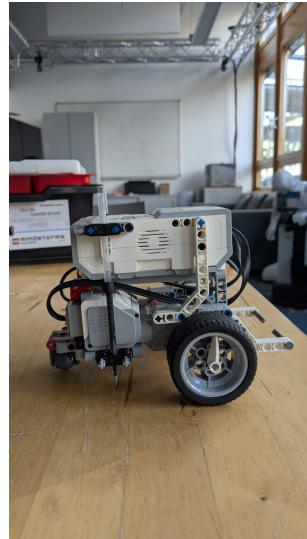
where v is the linear velocity, ω the angular velocity, r the wheel radius, L the distance between the wheels, and ω_r, ω_l the angular velocities of the right and left wheels, respectively.

Figure 1 and Figure 2 showcase various views of the robot that this team built. The construction and architecture of the robot is done in a way that the control unit comprises the middle part of the body. Since it is the most massive component of the robot, the control unit in the center, the center of gravity of the robot stays in approximately the middle of its body, which allows it to be stable. The two active wheels on both sides, along with the caster wheel, can be seen in Figure 2, specifically subfigure 2b. In order to take measurements, the main components used were a pen holder, a pen, and a so-called 'wall aligner'. Since the requirement of this assignment is to determine the end pose of the robot two points marked by the pens would be used to calculate the position of the robot as well as its orientation. The pen holder is a circular piece we found in the lego kit with a hole in the middle that became suitable for its designated purpose.

In order to ensure that the robot always starts from approximately the same pose in every trial of the experiment, the robot has a supposedly flat, long lego piece placed at the back of the robot. This wall aligner works by placing the robot in such a way that the wall aligner's flat surface completely pushes against it, eliminating a degree of freedom in the start position. Then the wheels will be placed between two prior marked lines.



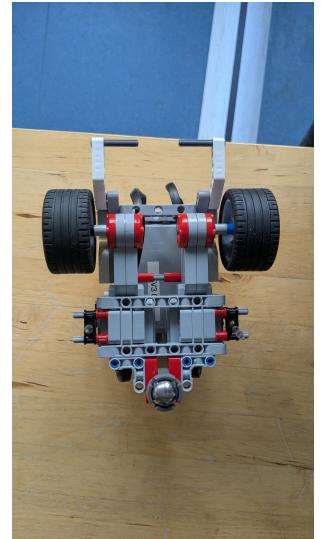
(a) Top view



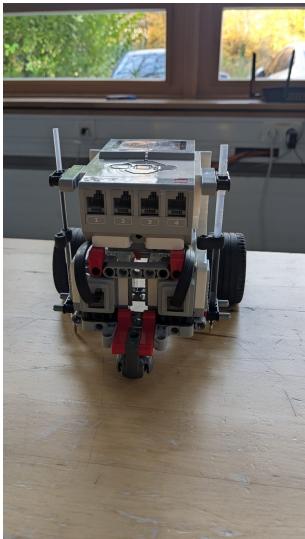
(b) Side view



(a) Second isometric view



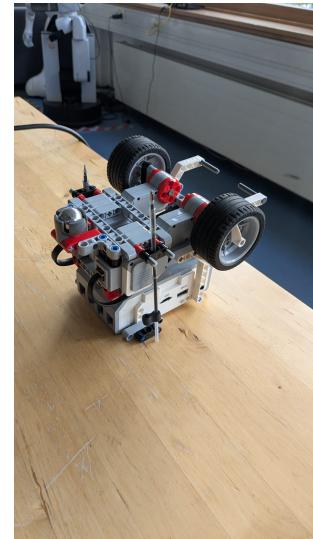
(b) Bottom view



(c) Front view



(d) Isometric view



(c) Bottom-isometric view

Fig. 1: Various orthographic and isometric views of the differential drive robot. The top, side, front, and isometric views (a to d) detail the chassis, wheel placement, and central control unit.

Fig. 2: Additional views of the robot focusing on the undercarriage. The second isometric, bottom, and bottom-isometric views (a to c) illustrate the passive caster wheel and pen placement mechanism.

B. Estimates of the expected Precision

Let \mathbf{F} be a vector function, defining the end pose of the robot, calculated from two measured positions on a 2D plane, \mathbf{L} and \mathbf{R} and the distance between the drive wheel axis and the pens midpoint, \mathbf{A} . The output components are defined such that \mathbf{F}_1 and \mathbf{F}_2 represent the position coordinates and \mathbf{F}_3 represents the orientation of the end pose. The input vectors and the functional relationship are defined in Equations 3 to 9.

$$\mathbf{L} = \begin{pmatrix} L_x \\ L_y \end{pmatrix} \quad (3)$$

$$\mathbf{R} = \begin{pmatrix} R_x \\ R_y \end{pmatrix} \quad (4)$$

$$\theta = \arctan \left(\frac{x_{2,1} - x_{1,1}}{x_{2,2} - x_{1,2}} \right) \quad (5)$$

$$\mathbf{A} = \begin{pmatrix} a \\ 0 \end{pmatrix} \quad (6)$$

$$\text{Rot}_{\mathbf{Z}}(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (7)$$

$$\mathbf{A}^R = \text{Rot}_{\mathbf{Z}}(\theta) \begin{pmatrix} a \\ 0 \end{pmatrix} = \begin{pmatrix} A_x^R \\ A_y^R \end{pmatrix} \quad (8)$$

$$\mathbf{F}(\mathbf{L}, \mathbf{R}) = \begin{pmatrix} \frac{1}{2}(L_x + R_x) + A_x^R \\ \frac{1}{2}(L_y + R_y) + A_y^R \\ \theta \end{pmatrix} \quad (9)$$

It is important to note that the angular component, \mathbf{F}_3 , exhibits a division by zero when $x_{1,2} = x_{2,2}$. In this specific case, the orientation is $\frac{\pi}{2}$ rad for $x_{1,1} > x_{2,1}$, or $-\frac{\pi}{2}$ rad for $x_{1,1} < x_{2,1}$.

As the proposed measurement mechanism utilizes two marked points, inherent errors exist in the measured coordinates. These uncertainties are characterized by the 5×5 input covariance matrix, \mathbf{C}_x , presented in Equation 10.

$$\mathbf{C}_x = \begin{pmatrix} \sigma_{L_x}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{L_y}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{R_x}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{R_y}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_a^2 \end{pmatrix} \quad (10)$$

The assumption that the uncertainty of each coordinate is uncorrelated simplifies \mathbf{C}_x to a diagonal matrix. The estimated standard deviation σ for each input coordinate, is provided in Table I.

These estimates are primarily informed by an analysis of the major and minor sources of error detailed below:

Parameter	Standard Deviation (σ)	Variance (σ^2)
L_x	2.0 mm	4.0 mm ²
L_y	2.0 mm	4.0 mm ²
R_x	2.0 mm	4.0 mm ²
R_y	2.0 mm	4.0 mm ²
a	1.0 mm	1.0 mm ²

TABLE I: Estimated Standard Deviation for Input Coordinates

- Major Sources of Error

 - Pen Wiggle and Holder Flex:

The non-rigid nature of the pen holding mechanism causes the pen tips to wobble or shift relative to the robot's chassis as the system undergoes dynamic movement, leading to inconsistent and random mark placement.

 - Marking Ambiguity:

The finite diameter of the pen tip (up to approximately 3 mm) creates an intrinsic marking error. The visual ambiguity in accurately estimating the true center of the resulting dot is a significant source of measurement uncertainty in the experiment.

- Minor Sources of Error

 - Parallax Error:

This is a systematic human error in measurement, arising when the observer's eye is not aligned perpendicularly with the measurement scale and the marked point. This leads to inconsistent readings when quantifying the marked coordinates.

- Negligible Sources of Error

 - Manufacturing Tolerances of Construction Components:

Component tolerances (e.g., LEGO bricks), typically in the range of $10 - 20 \mu\text{m}$.

C. Jacobian Error Propagation

The propagation of uncertainty in the measured input coordinates requires the calculation of the Jacobian matrix \mathbf{J} . The total input vector $\mathbf{x} = (L_x, L_y, R_x, R_y)^T$ is composed of all four scalar variables. Since \mathbf{F} is a vector-valued function with three outputs and four inputs, the resulting Jacobian matrix \mathbf{J} is a 3×5 matrix, presented in Equations 11. It is important to note, that the resulting Jacobian is too large to display here, and it can be seen instead in the code provided in the zip file, which includes this report.

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial F_1}{\partial L_x} & \frac{\partial F_1}{\partial L_y} & \frac{\partial F_1}{\partial R_x} & \frac{\partial F_1}{\partial R_y} & \frac{\partial F_1}{\partial A} \\ \frac{\partial F_2}{\partial L_x} & \frac{\partial F_2}{\partial L_y} & \frac{\partial F_2}{\partial R_x} & \frac{\partial F_2}{\partial R_y} & \frac{\partial F_2}{\partial A} \\ \frac{\partial F_3}{\partial L_x} & \frac{\partial F_3}{\partial L_y} & \frac{\partial F_3}{\partial R_x} & \frac{\partial F_3}{\partial R_y} & \frac{\partial F_3}{\partial A} \end{pmatrix} \quad (11)$$

The covariance matrix of the calculated end pose \mathbf{C}_F is then determined from the input covariance matrix \mathbf{C}_x

via the generalized error propagation formula, which can be seen in Equation 12

$$\mathbf{C}_F = \mathbf{J} \mathbf{C}_x \mathbf{J}^T, \quad (12)$$

where \mathbf{C}_x is the 5×5 covariance matrix of the input variables (L_x, L_y, R_x, R_y, A).

II. Robot Path Experiment

The goal of this experiment is to document the robot's experimental execution, including all program parameters and real-time observations. This report details and visualizes the robot's end-pose and path.

A. Experiment Setup

Figure 3 shows the final state of the sheet of paper used to collect the manual measurements. A video depicting the experiment, especially the robot moving forward, rightward, leftward is given with the following link: <https://drive.google.com/file/d/1O4k45PgLrehoumyVYWrPUiDslAiHdMky/view?usp=drivesdk>.

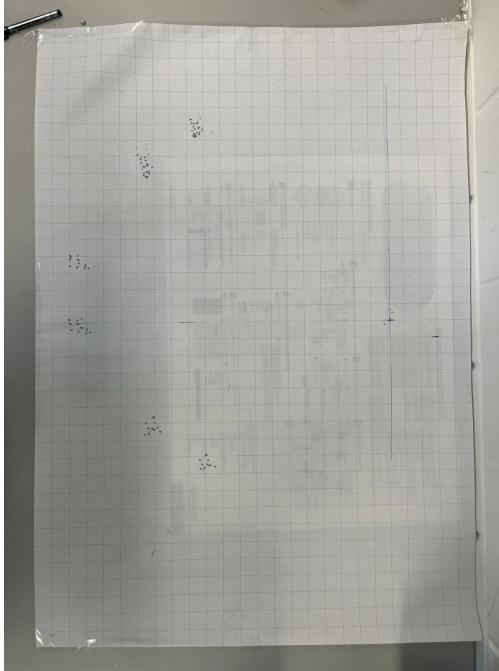


Fig. 3: Image of the sheet on which the markings were made.

B. Experiment Observations and Hurdles

During the experiment our team has made the following observations:

- The robot jerks at the beginning of a run, which leads to an unpredictable jump forward
- The used pens are not great for marking and needed to be repeatedly tapped on the paper to make a mark
- Due to the large number of points close together, that need to be measured, a system of crossing out past points needed to be developed

- Accidental run starts have happened, which leads to outliers in the encoder data, which were to complicated to remove during the experiment, and had to be left in

C. Raw Data from Measurements

Table II shows the raw data of the manual pen measurements from our group, where L_x is the X-coordinate of the left pen, L_y the Y-coordinate of the left pen, R_x the X-coordinate of the right pen, and R_y the Y-coordinate of the right pen at the robots end pose. The left pen of the robot, at it's start position, is the origin of the used coordinate frame, X is the forward direction from robots perspective, at it's start position, Y is to the right of X, and the angle Theta is measured from X in negative mathematical sense (clock wise). This raw data was then used to calculate the robots end pose. The path the robot took was taken from the encoder readings. All calculations where done using Python scripts, which are part of the zip file upload, that includes this report. For the calcualtions, A was measured to be $(6.9cm, 0)^T$.

Forward Movement				Rightward Movement				Leftward Movement			
L_x	L_y	R_x	R_y	L_x	L_y	R_x	R_y	L_x	L_y	R_x	R_y
46.0	0.8	46.0	10.3	36.8	22.1	29.4	28.2	28.9	-20.0	36.7	-14.5
48.3	0.0	48.1	9.6	37.6	25.3	29.9	30.7	28.4	-20.3	36.2	-12.3
47.1	0.4	48.2	10.0	36.0	22.5	28.5	28.4	28.6	-20.4	36.4	-14.9
47.4	-0.6	47.6	9.1	36.6	24.9	28.9	30.3	27.7	-22.6	35.9	-17.6
47.0	1.1	47.1	10.3	36.0	22.8	28.5	28.8	27.4	-21.3	35.5	-16.0
45.7	0.4	45.6	9.9	36.3	23.6	28.6	29.4	29.4	-23.2	37.5	-18.2
46.9	0.1	47.0	9.7	36.5	24.4	28.8	30.1	28.8	-20.7	36.5	-15.3
46.3	0.5	46.0	10.0	37.2	24.0	29.3	29.6	27.9	-20.3	35.7	-14.8
46.8	-0.1	46.9	9.5	36.8	22.5	29.2	28.5	28.9	-20.5	36.8	-15.0
47.8	-0.1	47.9	9.5	36.5	22.0	29.0	28.1	28.5	-20.0	36.2	-14.3
48.0	-0.1	47.9	9.6	36.4	22.7	28.7	28.6	28.8	-19.8	36.7	-14.5
47.4	-0.4	47.6	9.4	37.3	24.5	29.6	30.2	27.9	-20.0	35.8	-14.4
47.9	0.0	47.8	9.9	36.5	22.9	29.0	28.2	26.5	-21.6	34.5	-13.7
46.5	-1.4	46.9	8.4	36.4	23.5	28.3	31.3	28.6	-19.3	36.8	-9.3
45.3	-0.9	45.5	8.8	36.1	23.6	28.5	34.3	27.9	-19.8	35.6	-11.6
46.8	0.5	46.7	10.1	37.9	26.5	29.8	31.5	27.5	-21.2	35.8	-13.3
46.6	-0.2	46.6	9.5	35.6	23.2	28.1	29.2	27.5	-20.5	35.5	-14.9
45.5	-1.6	45.7	8.2	36.3	25.4	28.3	31.0	27.8	-19.9	35.6	-14.2
45.5	-1.2	45.5	8.5	37.0	23.2	29.6	29.0	27.7	-20.0	35.6	-14.3
47.5	-1.1	47.8	8.6	35.5	23.5	28.0	29.3	27.2	-21.4	35.5	-15.0
44.9	-1.4	45.1	8.2	36.8	23.5	29.3	29.4	28.2	-21.9	36.2	-16.5
48.0	0.5	47.7	10.0	37.2	25.0	29.2	30.2	27.4	-21.3	35.0	-16.0
47.8	-0.3	47.7	9.6	36.3	23.7	28.2	29.0	28.1	-21.6	36.1	-16.3
46.1	-0.8	46.5	9.0	36.8	23.4	29.1	29.2	28.5	-21.2	36.3	-15.8
47.5	-1.1	47.8	10.0	36.4	24.0	28.4	29.6	28.7	-21.1	36.7	-15.0

TABLE II: Manual pen measurements at the Robots end pose

D. Visualisation and Images

Figure 4 depicts the combined pen measured end poses of the robot of all four teams, where our team is Team 1. Figures 5 to 8 depict the robot end pose, calculated from the manual measurements, as well as the robots trajectory, calculated from the encoder data for each team.

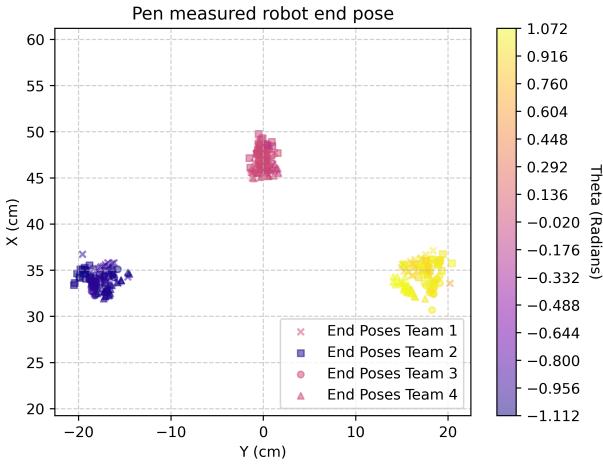


Fig. 4: Combined end pose plot of all teams pen measurements

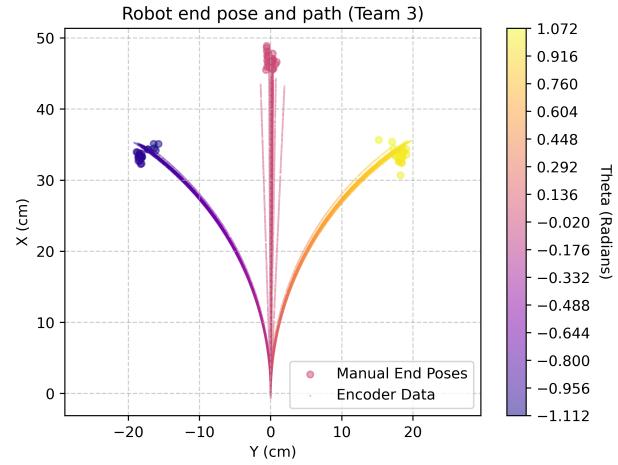


Fig. 7: Combined end pose and path plot of Team 3

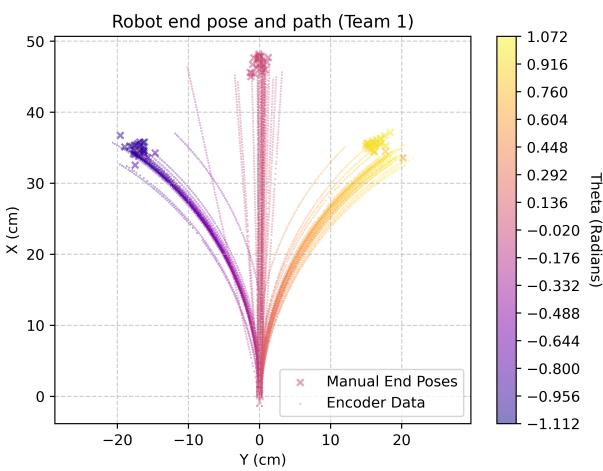


Fig. 5: Combined end pose and path plot of Team 1

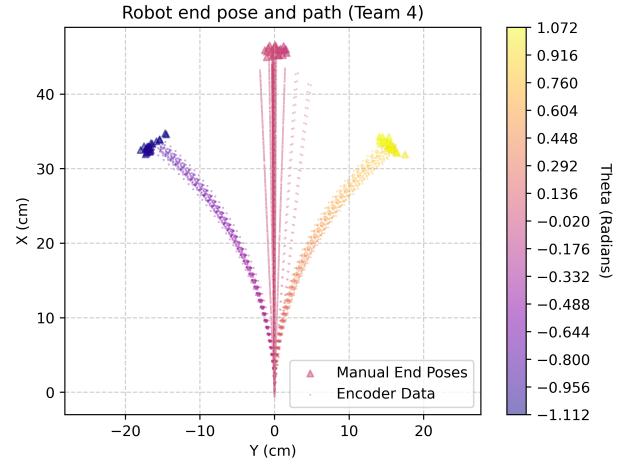


Fig. 8: Combined end pose and path plot of Team 4

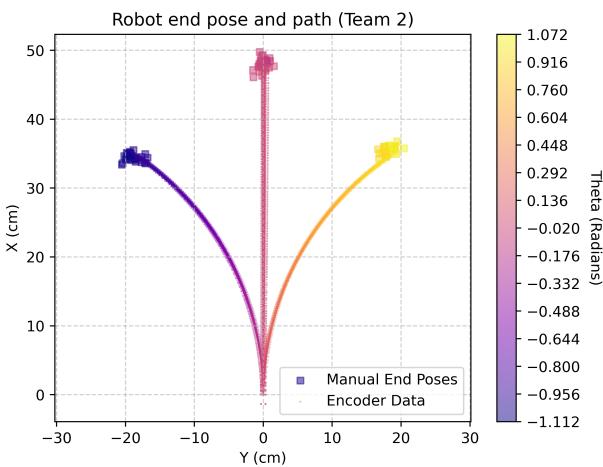


Fig. 6: Combined end pose and path plot of Team 2

III. Statistical Analysis of the Experiment Data

In this experiment the goal is to analyse the raw data using statistical tools. It is important to note that only precision can be analysed here, as there is no knowledge of the true value, only measurements. All data analysis was performed using the Python script Calculations.py and its dependencies, which is included in the code folder inside the delivered zip file.

A. Chebyshev and χ^2

When we use the Chebyshev function to remove the outliers with using the threshold as 2.0 standard deviations, we were able to remove 7 outliers in total out of the 75 manual readings that we measured. For reference we are attaching a screenshot of the results from the code we ran to find this in Figure 9

```
PS: C:\Users\harik\Downloads\Semester2\SEI_302025\codes & C:\Users\harik\appdata\local\Temp\Programs\Python311\python.exe c:/Users/harik/Downloads/Semester2/SEI_302025/code/Chebyshov.py

...Outlier Detection (Chebychev 2.0e) ...
Column 'X': 2 outliers found;
Column 'Y': 3 outliers found;
Column 'Theta': 1 outliers found;
Total points: 25, total outliers removed: 3, Filtered points: 22

...Outlier Detection (Chebychev 2.0e) ...
Column 'X': 2 outliers found;
Column 'Y': 3 outliers found;
Column 'Theta': 1 outliers found;
Total points: 25, total outliers removed: 2, Filtered points: 23

...Outlier Detection (Chebychev 2.0e) ...
Column 'X': 2 outliers found;
Column 'Y': 3 outliers found;
Column 'Theta': 1 outliers found;
Total points: 25, total outliers removed: 2, Filtered points: 23
```

Fig. 9: The results of the Chebyshov function removing the outliers from the manual data.

To further characterize the data's dispersion, statistical boundaries were calculated for the [Theta, X, Y] data for each travel direction. Confidence regions were generated for each data using two distinct methods: Chebyshev's inequality and the Chi-Squared (χ^2) distribution. Figures 10 to 12 depict these data with their corresponding boundaries.

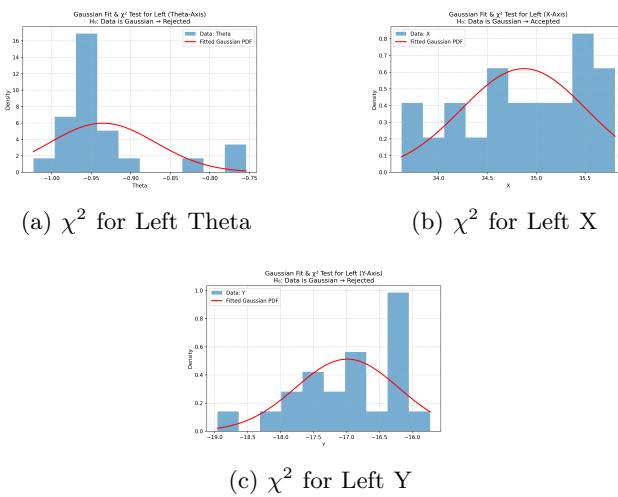


Fig. 10: The figures above (a to c) represent the Gaussian graph along the data of the motion of the robot going left with respective to Theta, X and Y.

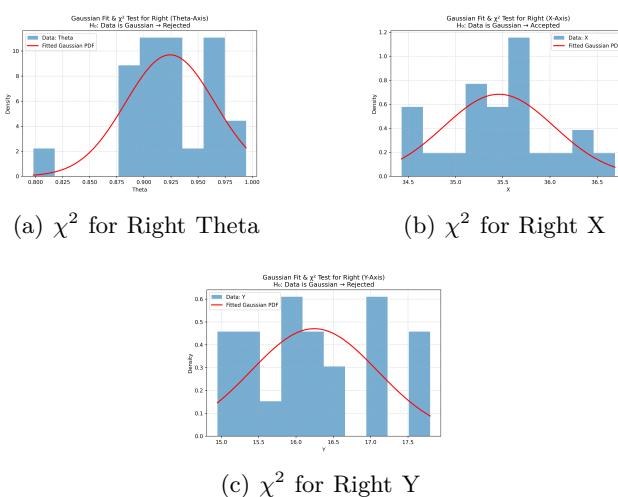


Fig. 11: The figures above (a to c) represent the Gaussian graph along the data of the motion of the robot going right with respective to Theta, X and Y.

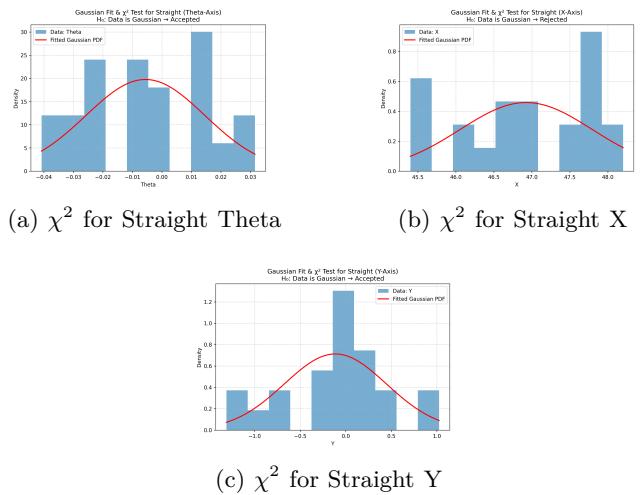


Fig. 12: The figures above (a to c) represent the Gaussian graph along the data of the motion of the robot going right with respective to Theta, X and Y.

B. Statistical Parameters

Table III contains statistical information for all motion directions and axes (X, Y, and Theta).

Axis	Mean	Std. Dev. (σ)	χ^2 Stat	p-value	GaussianFit
X	34.874 cm	0.642 cm	8.988	0.254	Accept
Y	-16.988 cm	0.779 cm	15.345	0.032	Reject
Theta	-0.935 rad	0.067 rad	41.871	5.5×10^{-7}	Reject

Table A: Left Direction

Axis	Mean	Std. Dev. (σ)	χ^2 Stat	p-value	GaussianFit
X	35.460 cm	0.584 cm	9.854	0.197	Accept
Y	16.245 cm	0.848 cm	17.325	0.015	Reject
Theta	0.924 rad	0.041 rad	17.553	0.014	Reject

Table B: Right Direction

Axis	Mean	Std. Dev. (σ)	χ^2 Stat	p-value	GaussianFit
X	46.925 cm	0.871 cm	26.944	3.4×10^{-4}	Reject
Y	-0.114 cm	0.560 cm	13.030	0.071	Accept
Theta	-0.006 rad	0.020 rad	14.031	0.051	Accept

Table C: Straight Direction

TABLE III: Summary of Statistical Analysis by Movement Direction

C. PCA and Ellipsoid Analysis

To further analyse the data, a principle component analysis (PCA) was done for the manual data for each direction the robot traveled. An ellipsoid for each direction was calculated based on the principle components (PCs) of their respective point cloud (the X, Y, Theta data points). Figures 13 to 15 depict these point clouds with their ellipsoids. In these figures, the principle components were set to a length of two standard deviations σ calculated from their corresponding principle component. As the PC

for Theta is very small numerically, it needed to be magnified, which lead to a skewed projection.

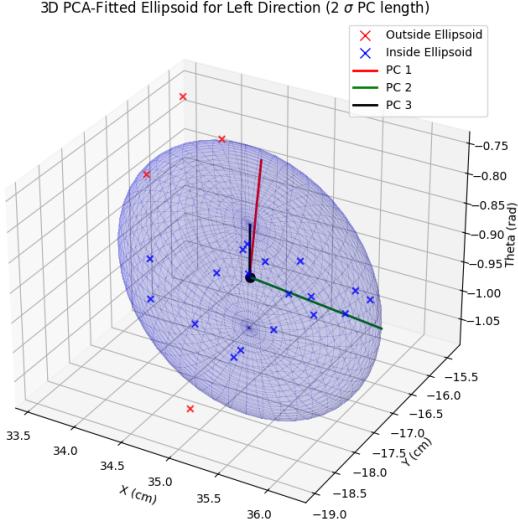


Fig. 13: 3D Ellipsoid based on PCA of manual data for the left direction

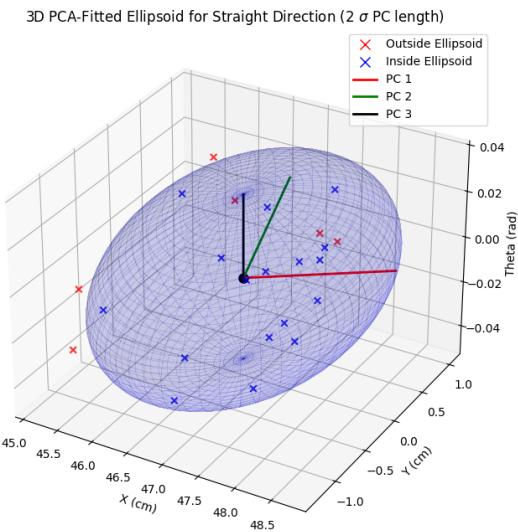


Fig. 14: 3D Ellipsoid based on PCA of manual data for the Straight direction

3D PCA-Fitted Ellipsoid for Right Direction (2 σ PC length)

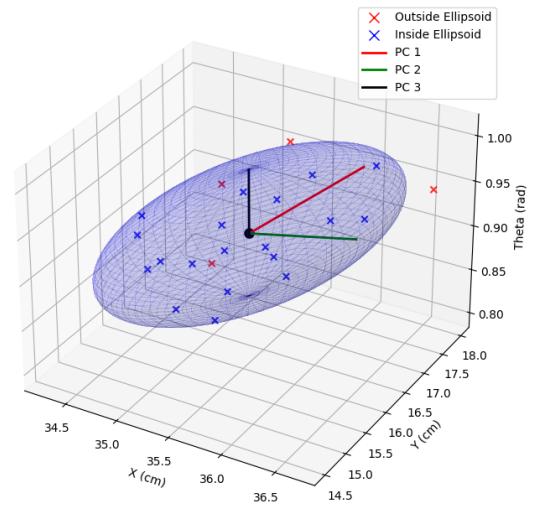


Fig. 15: 3D Ellipsoid based on PCA of manual data for the right direction

IV. Calibrating an Optical Tracking System

In this experiment, we want to determine the intrinsic and extrinsic parameters of the provided camera.

A. Calibration Setup

The setup for calibration involved mounting and securing the camera to the tripod provided in the lab. Once the tripod and camera were placed in a an area with significant illumination, the next step was to work on the settings of the camera. After disabling auto-focus and setting the camera resolution to maximum, as instructed for this experiment, the preparations were complete and the setup for the camera calibration was ready for experimentation. Possible pitfalls include environmental factors such as changes in luminance of the checkerboard, and human errors such as inconsistent and asymmetric changes in poses for the images taken, which could introduce different focal length and other parameter calculations. One of the most prominent example of these human errors would be changes in distance away from the camera. The total number of pictures taken for this experiment was 18. Considering the fact that changes in poses were applied after every other picture taken, this number was deemed to be enough.

B. OpenCV Camera Parameters

To un-distort an image, the toolbox OpenCV calculates the following intrinsic and extrinsic camera parameters:

- Intrinsic Parameters are properties of the camera and lens assembly:
 - Camera Matrix (**mtx**): A 3×3 matrix modeling the perspective projection. It transforms 3D coordinates from the camera frame to 2D pixel coordinates.
 - Distortion Coefficients (**dist**): A vector containing the lens distortions, k_1, k_2, p_1, p_2, k_3 , where k_i are the radial distortions, and p_i the tangential distortions.
- Extrinsic Parameters define the pose of the camera relative to the world coordinate system:
 - Rotation Vectors (**rvecs**): A list of vectors, representing the camera's orientation relative to the world frame for each image.
 - Translation Vectors (**tvecs**): A list of vectors, defining the camera's position relative to world coordinate system for each image.

C. Experiment Observations

D. Observations

1) Camera Functionality Check: During testing, there was no lag or focusing issues and all configurations adjustments were proper. Overall, the camera responded reliably for static image acquisition.

2) Environmental Conditions: The calibration experiment was conducted in (indoor) conditions under (natural lighting). The illumination uniformity was verified to minimize shadow and glare on the checkerboard surface(Captured the images by sitting opposite to the windows in the lab to minimize shadows). The camera was mounted on a laptop and images were captured without any shaking or movements.

3) Image Capture Process: The board was moved to different positions and angles to provide sufficient geometric diversity for accurate calibration. Certain frames were (discarded) due to (blur, board not in frame in total, poor corner detection, etc.). The process ensured that the checkerboard corners were well-detected in most images.

4) Observed Problems and Error Sources: Several factors that could influence calibration accuracy were identified:

- (Lighting irregularities or glare) affecting corner visibility.
- (Motion blur or unstable mounting device) causing corner mismatch.
- (Non-planar or bent checkerboard) introducing geometric distortion.
- (Incorrect checker square size input) affecting scaling accuracy.

These observations highlight the importance of stable setup, consistent lighting, and accurate checkerboard dimensions during calibration.

5) Reflections and Improvements: From the conducted calibration, it was observed that capturing images from diverse angles and maintaining uniform lighting significantly improved corner detection accuracy and reduced overall reprojection error. Future iterations could improve accuracy by (using more images, enhancing lighting uniformity, ensuring higher contrast patterns). Overall, the process provided valuable insight into the factors influencing intrinsic and extrinsic parameter estimation reliability.

E. Results

Figure 16 shows a sample of images in various poses, which were used to find the camera parameters using the OpenCV toolbox.

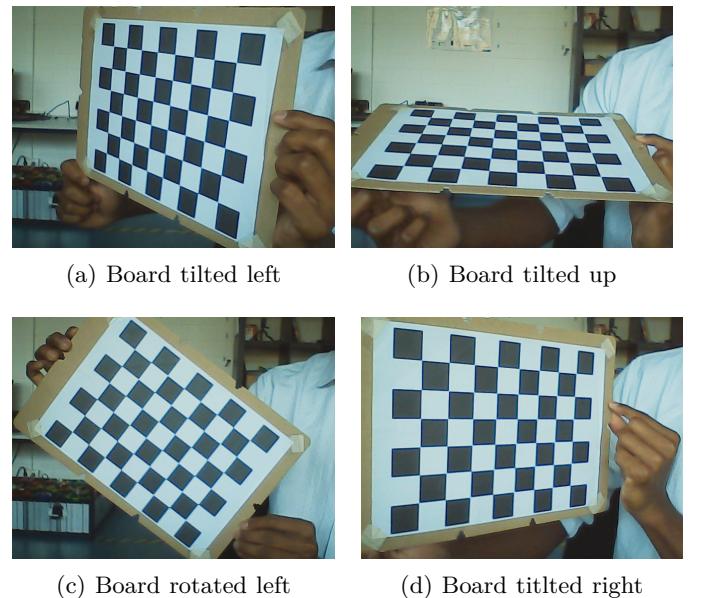


Fig. 16: A sample of images used to find the camera parameters

Figure 17 depicts the difference between an original image and its undistorted version using the OpenCV toolbox and the found camera parameters.

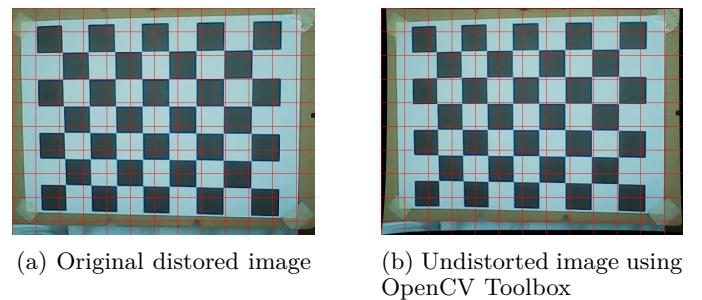


Fig. 17: Undistorting an image using OpenCV toolbox

The camera intrinsic and extrinsic parameters found are summarized in the following equations. In Equations 13 through 16, one can see the found parameters from the

calibration process, where **err** is the error estimate, and **r** and **t** are a sample from **rvecs** and **tvecs** respectively.

$$\mathbf{err} = 0.5 \quad (13)$$

$$\mathbf{mtx} = \begin{pmatrix} 825.647 & 0.000 & 334.645 \\ 0.000 & 829.003 & 220.793 \\ 0.000 & 0.000 & 1.000 \end{pmatrix} \quad (14)$$

$$\mathbf{dist} = \begin{pmatrix} 0.13401 \\ -0.30079 \\ -0.01151 \\ 0.00074 \\ -1.70837 \end{pmatrix} \quad (15)$$

$$\mathbf{r} = \begin{pmatrix} 0.05167 \\ 0.09337 \\ -0.00217 \end{pmatrix}, \mathbf{t} = \begin{pmatrix} -4.04096 \\ -2.22142 \\ 14.77100 \end{pmatrix} \quad (16)$$

V. Measuring the Accuracy and Precision of KUKA youBot

This report documents the execution of 225 experimental trials using the KUKA youBot arm to assess its accuracy and precision. We detail the experimental setup, observations regarding error sources, the data preprocessing and filtering pipeline, and a comparative visualization of internal encoder data versus external OptiTrack measurements.

A. Experiment Setup

The experiment was conducted using a KUKA youBot (ID 3) in lab C069. The task involved pick and place operations with three different objects moved to three target poses (Straight, Left, Right). The ground truth coordinates for the opti track positions and the end effector were defined in Tables IV and V respectively.

Pose	$x(m)$	$y(m)$	$\theta(rad)$
Pick	0.143	-0.351	-1.65
Straight	0.150	-0.212	-1.65
Left	0.356	-0.283	-2.08
Right	-0.064	-0.352	-1.13

TABLE IV: Ground truth object poses

Pose	$x(m)$	$y(m)$	$\theta(rad)$
Pick	0.152	-0.457	-1.65
Straight	0.167	-0.317	-1.62
Left	0.331	-0.380	-2.11
Right	-0.01	-0.433	-1.11

TABLE V: Ground truth end effector poses

Motion data was captured using an OptiTrack external vision system (Motive software) and internal ROS bagfiles.

B. Observations and Sources of Error

During the execution of the 225 trials (25 repetitions per configuration), several observations were made that are relevant to the data analysis:

- The OptiTrack system would be unable to detect the object, if one of the markers was blocked. We made sure that did not happen during trials.
- The rosbag creation was faulty sometimes, and these trials had to be redone.
- The larger the object, the longer it took to settle down and no longer oscillate.
- The data we have got doesn't match to the ground truth table given in the experiment setup section. This is probably due to the fact that the robot wasn't properly calibrated.

C. Results

Figures 18 to 20 depict the end points of the mass and the end effector and compares them to the given ground truth for each. These figures show that the experiment setup has a very large systematic error, leading to very low accuracy. The precision appears to be high except for the small mass. It is important to note that the orientation of the mass are not shown, due to misalignment of the object during placement back to the start position by a human. Also outlier were removed using Chebyshev twice, were everything out of two sigma was discarded as an outlier.

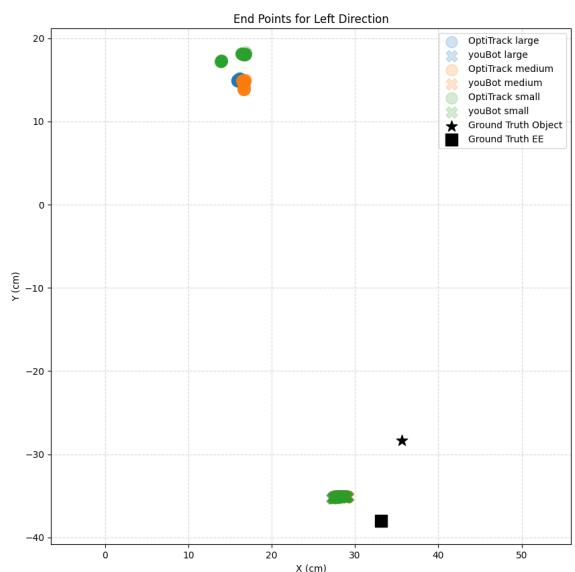


Fig. 18: Left direction end poses for each object size and their respective ground truth

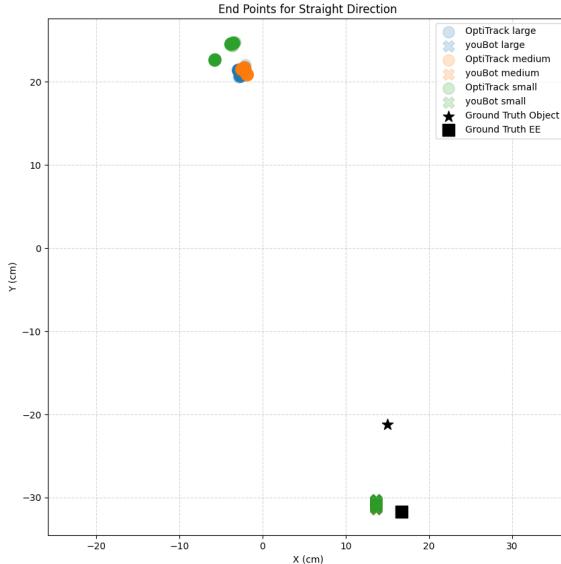


Fig. 19: Straight direction end poses for each object size and their respective ground truth

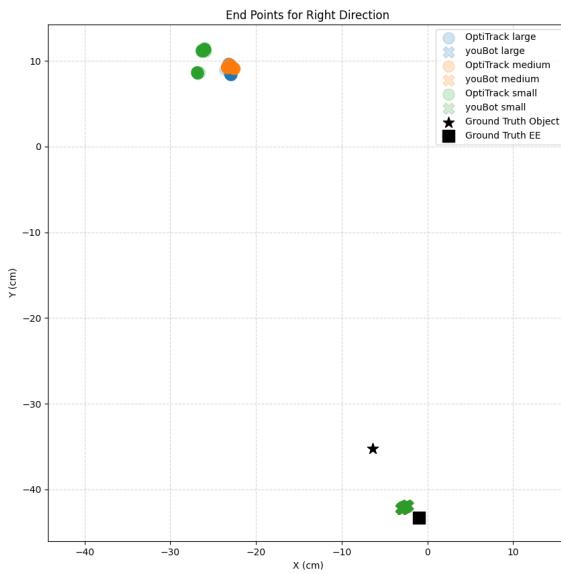


Fig. 20: Right direction end poses for each object size and their respective ground truth

VI. Measuring the Accuracy and Precision of a KUKA youBot Arm

This section assesses the experiments accuracy, precision, and determines whether the data is normal (gaussian) distributed.

A. Statistical Analysis

To determine the accuracy and precision of the youBot robot arm, the gathered data was further analyzed. To find the accuracy, the mean of the point clouds for each direction and object was compared with its respective ground truth. For precision, a Principal Component Analysis (PCA) was conducted (using only the X and Y coordinates of the data), and the largest standard deviation was used to describe the worst-case precision. The results of this analysis are depicted in Tables VI and VII. These tables show that, regarding accuracy, the larger the object, the higher the accuracy (though only marginally). For precision, the OptiTrack system shows that the small object has the worst precision, whereas for the youBot Arm, the large object has the worst precision. The medium object exhibits the highest precision for both systems.

Direction	Object	Accuracy	Precision
Left	Small	73.122	1.193
	Medium	69.719	0.324
	Large	69.666	0.410
Right	Small	49.978	1.193
	Medium	47.545	0.324
	Large	47.367	0.410
Straight	Small	52.143	1.193
	Medium	48.815	0.324
	Large	48.738	0.410

TABLE VI: Accuracy (Distance of Ground Truth) and Precision (Largest Standard Deviation) Data for the OptiTrack System

Direction	Object	Accuracy	Precision
Left	Small	76.769	1.489
	Medium	73.467	0.997
	Large	73.384	1.701
Right	Small	59.583	1.489
	Medium	57.069	0.997
	Large	56.896	1.701
Straight	Small	60.343	1.489
	Medium	57.153	0.997
	Large	57.049	1.701

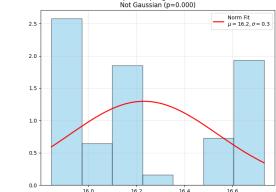
TABLE VII: Accuracy (Distance to Ground Truth) and Precision (Largest Standard Deviation) Data for the youBot Robot Arm

B. χ^2 Test for Goodness of Fit

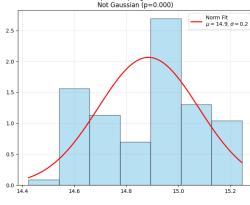
VII. Chi-Squared Goodness-of-Fit Analysis

The following figures present the results of the Chi-Squared (χ^2) goodness-of-fit test applied to the positional data for the robot's motion. These tests assess how well the observed data distributions (for X and Y coordinates) conform to a theoretical distribution (e.g., a Gaussian/Normal

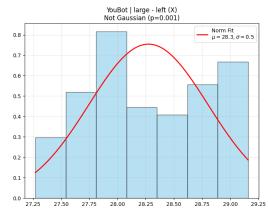
distribution for the errors/residuals). The measurements are analyzed separately for the OptiTrack and YouBot sensing systems, across different motion profiles and different masses.



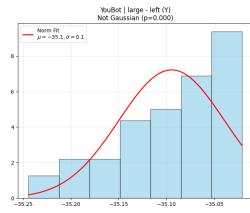
(a) χ^2 Goodness-of-Fit for OptiTrack Large Left X



(b) χ^2 Goodness-of-Fit for OptiTrack Large Left Y

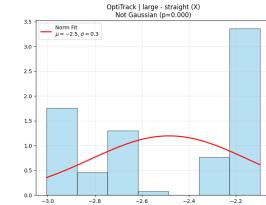


(c) χ^2 Goodness-of-Fit for YouBot Large Left X

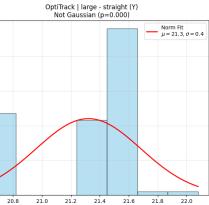


(d) χ^2 Goodness-of-Fit for YouBot Large Left Y

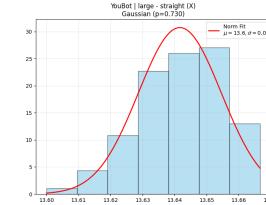
Fig. 21: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Left with the Large Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.



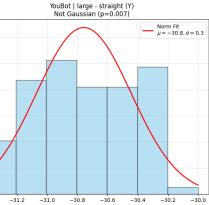
(a) χ^2 Goodness-of-Fit for OptiTrack Large Straight X



(b) χ^2 Goodness-of-Fit for OptiTrack Large Straight Y

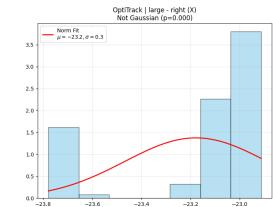


(c) χ^2 Goodness-of-Fit for YouBot Large Straight X

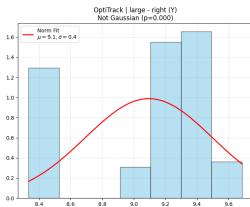


(d) χ^2 Goodness-of-Fit for YouBot Large Straight Y

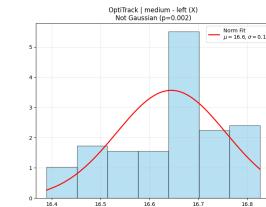
Fig. 23: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Straight with the Large Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.



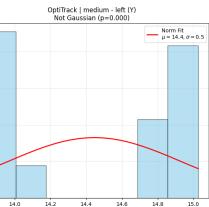
(a) χ^2 Goodness-of-Fit for OptiTrack Large Right X



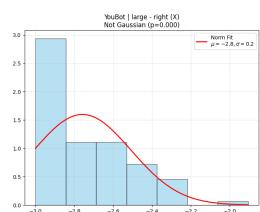
(b) χ^2 Goodness-of-Fit for OptiTrack Large Right Y



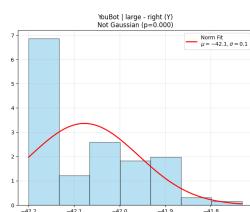
(a) χ^2 Goodness-of-Fit for OptiTrack Medium Left X



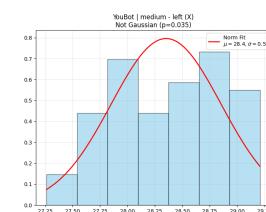
(b) χ^2 Goodness-of-Fit for OptiTrack Medium Left Y



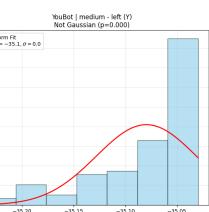
(c) χ^2 Goodness-of-Fit for YouBot Large Right X



(d) χ^2 Goodness-of-Fit for YouBot Large Right Y



(c) χ^2 Goodness-of-Fit for YouBot Medium Left X



(d) χ^2 Goodness-of-Fit for YouBot Medium Left Y

Fig. 22: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Right with the Large Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

Fig. 24: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Left with the Medium Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

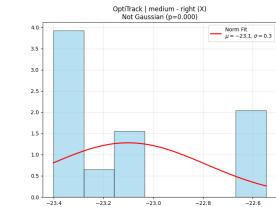
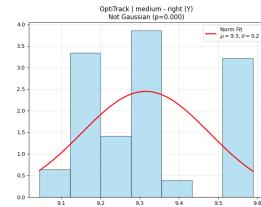
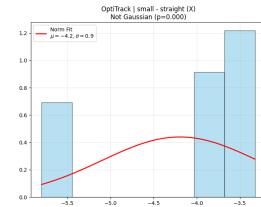
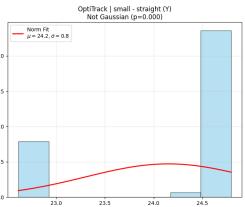
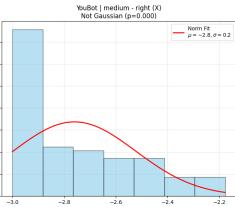
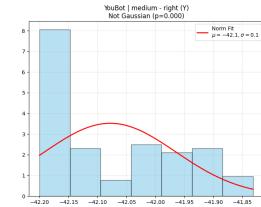
(a) χ^2 Goodness-of-Fit for OptiTrack Medium Right X(b) χ^2 Goodness-of-Fit for OptiTrack Medium Right Y(a) χ^2 Goodness-of-Fit for OptiTrack Small Straight X(b) χ^2 Goodness-of-Fit for OptiTrack Small Straight Y(c) χ^2 Goodness-of-Fit for YouBot Medium Right X(d) χ^2 Goodness-of-Fit for YouBot Medium Right Y

Fig. 25: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Right with the Medium Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

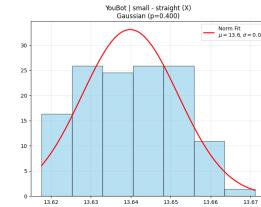
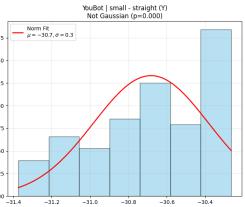
(c) χ^2 Goodness-of-Fit for YouBot Small Straight X(d) χ^2 Goodness-of-Fit for YouBot Small Straight Y

Fig. 27: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Straight with the Small Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

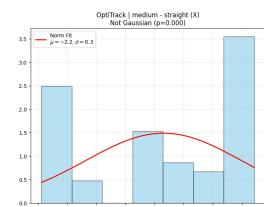
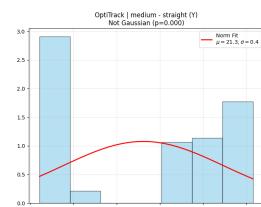
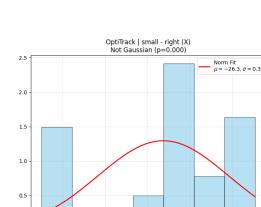
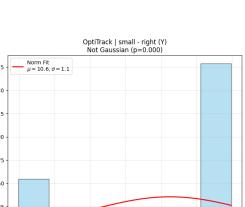
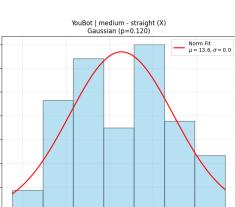
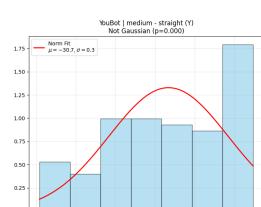
(a) χ^2 Goodness-of-Fit for OptiTrack Medium Straight X(b) χ^2 Goodness-of-Fit for OptiTrack Medium Straight Y(a) χ^2 Goodness-of-Fit for OptiTrack Small Right X(b) χ^2 Goodness-of-Fit for OptiTrack Small Right Y(c) χ^2 Goodness-of-Fit for YouBot Medium Straight X(d) χ^2 Goodness-of-Fit for YouBot Medium Straight Y

Fig. 26: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Straight with the Medium Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

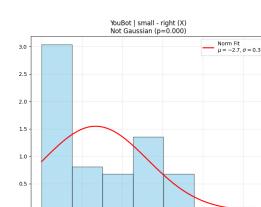
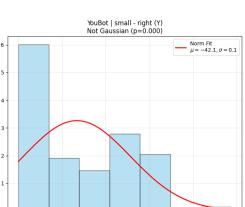
(c) χ^2 Goodness-of-Fit for YouBot Small Right X(d) χ^2 Goodness-of-Fit for YouBot Small Right Y

Fig. 28: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Right with the Small Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

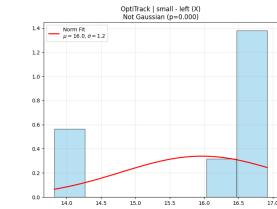
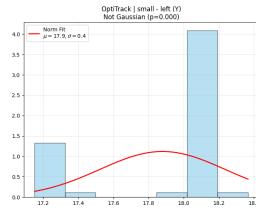
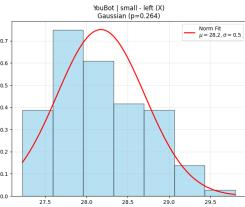
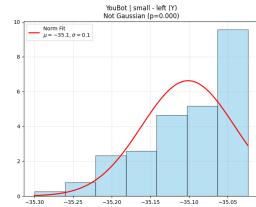
(a) χ^2 Goodness-of-Fit for OptiTrack Small Left X(b) χ^2 Goodness-of-Fit for OptiTrack Small Left Y(c) χ^2 Goodness-of-Fit for YouBot Small Left X(d) χ^2 Goodness-of-Fit for YouBot Small Left Y

Fig. 29: Chi-Squared Goodness-of-Fit results for the motion of the robot moving Left with the Small Mass, for the X and Y coordinates as measured by both OptiTrack and YouBot.

STATEMENT OF ORIGINALITY

We, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely our own work. The report was, in part, written with the help of the AI assistants Gemini and ChatGPT as described in the appendix. We are aware that content generated by AI systems is no substitute for careful scientific work, which is why all AI-generated content has been critically reviewed by us, and we take full responsibility for it.

08.12.2025

Date

Signature

Appendix

Usage of AI Assistants

In this work the AI Assistants, Gemini 2.5 Flash and ChatGPT 5, were used to help with Latex and Python code generation (e.g. turning multiple figures into one, plotting), as well as help with sentence structure, formulation and grammar.

Raw Data and Software

All of the raw data and code/software we used is in our repository at https://github.com/FHuppertz/SEE_WS2025.