HOCHSCHULE BONN-RHEIN-SIEG

Scientific Experimentation and Evaluation Manual for Practical Experiments

SUMMER SEMESTER 2021

In the SEE course, we cover practical experimentation and (statistical) data evaluation and data visualisation. This manual deals with practical experimentation, measurement and data analysis, using the experimental analysis of the characteristics of a differential drive as a running example.

1 Task 1: Manual Motion Observation

This task consists of two assignments. Your task is to construct a LEGO EV3 differential drive robot and manually measure the observable pose variation for three different constant velocity motions: an arc to the left, a straight line ahead, and an arc to the right.

1.1 Setup

Get a LEGO EV3 construction kit from your supervisors and a large grid-lined cardboard sheet (A0/A1 size). You will run the EV3 robot across the cardboard and record the end positions for multiple runs.

1. Construct a simple robot - you're free to choose the design.
2. Equip your robot and/or its environment with marking and measuring device/s (e.g. using pens, camera/s, lasers or any other means that you think are suitable), such that you can mark the position of the robot at its stop position.
3. Verify that your robot contains a control program necessary for running the pre-programmed distances. Use the provided program and do not change the predefined control parameters.
4. Run three experiments, each repeated **at least** 20 times (i.e. obtain **at least** 60 data points, better more):
   (a) set the robot to run with constant angular and translational velocities for a fixed time period, such that it describes an arc to the left
   (b) set the robot to run with constant angular velocity of zero degree per second and same translational velocity as in the previous experiment for the same fixed time period, such that it describes a straight line motion ahead

(c) set the robot to run with constant angular and translational velocities for a fixed time period, such that it describes an arc to the right, with the same velocities and times as before

For each experiment $(a - c)$ execute at least 20 runs and mark the robot's end pose after each run with a mark on the cardboard using the marking mechanism build onto the robot.

5. Determine the coordinates of each of the robot's end poses relative to its starting pose and tabulate the (x, y, θ) coordinates.

For step 5, you should use standard tabular structure (see table 3 in A.2) for each of the three different robot motions (extend number of rows if necessary). In order to keep consistency in data among groups, as a convention, you should choose a coordinate system such that: $i)$ the direction of Y-axis is governed by the robot's forward motion-direction at its start pose and $ii)$ the direction of X-axis should be perpendicular to that Y-axis (please follow right-hand rule for this coordinate system and see examples in appendix section A.1). This coordinate system should be fixed on the grid-lined sheet before the actual experiment and measurements start. Note: Steps 5 and 4 are best done in parallel.

1.2 Assignment 1.1: Design of Experiment

1. Read and analyze the following article: Sandve, Geir Kjetil, Anton Nekrutenko, James Taylor, and Eivind Hovig, "Ten simple rules for reproducible computational research", PLoS Computational Biology, vol. 9, p. e1003285, 2013.

2. Consider how to best record the end poses of the three times 20 runs of the robot.

3. Design and build a differential drive robot, including some means to obtain the robots end poses (measurement facility).

4. Describe in writing the robot's design, the measurement process and the expected problems. **Optional**: for making and documenting this design you can use some of the already existing CAD softwares, created specifically for the LEGO parts (e.g. official LEGO's design software *LDD – LEGO Digital Designer, STUDIO 2.0, etc.*, but there are other software that you can use as well for this purpose).

5. Give a rough (but justified by some arguments) estimation of the expected precision of your measurement process.

6. Propagate the estimated *error of your measurement process* using the *Jacobian* as an amplifier to compute its effect on the robot's orientation calculations. In other words, show us how the accuracy of orientation calculations is affected by the *error of your measurement process*. Additionally, include the upper and lower bounds of this propagated error. **Hints**: $i)$ if an analytical method does work out for you, instead, you can use a numerical method; $ii)$ use all different combinations of the error-values that can be induced (caused) by your measurement process.

Note: you are not yet requested to actually run the robot. This assignment deals with the experimental design.

1.3 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 orientation's uncertainty (including the upper and lower bounds) caused by the errors in the measurement process, using the method of *Jacobian error propagation*.

1.4 Assignment 1.2

1. Refine your experiment design according to feedback in class, if necessary
2. Run the requested experiment, by driving the robot (**at least**) 20 times on an arc to the right, (**at least**) 20 times straight ahead, and (**at least**) 20 times on an arc to the left with the same magnitude, but opposite direction of the commanded curvature to the right.
3. Measure and record the end poses of the robot, as well as any other observation that may prove useful for later analysis of the experiment. **Here, we are interested in seeing the poses that correspond to the center point of robot's axle**. Thus, if necessary (depending on the design of overall measurement facility), you should perform suitable transformations of all measured (raw) poses.
4. Make a video recording of robot's behaviour during the experiment
5. Combine (merge) your data with the data collected by all of your classmates.
6. Read and analyze the following article: Rougier, Nicolas P., Michael Droettboom, and Philip E. Bourne, "Ten Simple Rules for Better Figures", PLoS Computational Biology, vol. 10, p. e1003833, 2014.
7. Think of suitable methods to visualize the observed motion. Provide visualizations of the i) start robot pose, ii) the spread-out of the end robot poses (manual measurements - combined data) and iii) robot's paths (data generated by the control script from the encoder readings - combined data).

1.5 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 encoders' readings) you can take already generated files from the EV3 brick.
4. Visualization of the $i)$ robot's end poses (combined data from manual measurements), $ii)$ complete robot's 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 robot's behaviour during the experiment (one for each type of motion)

2 Task 2: Statistical Evaluation of the Previous Experiment

Last week, you ran three experiments to estimate your robot's motion behaviour when executing simple "v, ω" motions, i.e. motions characterised by an overlay of an angular velocity ω and a translational velocity v. This week, you will characterise the observed behaviour in terms of statistical parameters.

2.1 Task

No specific hardware setup required; you will use the data obtained by you and your colleagues in the previous assignment. Estimate the distribution governing the spread of the stop positions (for the combined data). Try to fit a multidimensional Gaussian for each of the three experiments and judge the fit: Does your data fit to a Gaussian? If not, which other distribution do you know that might be a good fit? Why? If you get a poor fit for a Gaussian, which effects may have caused this? Refer to your observations from last week.

Compare the estimated uncertainty of the measurement process itself with the statistical uncertainty of the measured stop positions. Which conclusions can be drawn from this comparison?

Note: If necessary, rerun the experiment from last week to obtain usable data. All of the outlined tasks should be performed with the combined data.

2.2 Assignment 2

1. Fix and, if necessary, rerun your experiment according to the feedback in class.
2. Compare the manually-measured end-poses and those estimated using the encoder data: do they fit? if not, describe possible causes for this.
3. Identify and remove outliers in your data using Chebyshev Theorem (threshold: 2σ).
4. Fit a Gaussian to your data and judge if the data is truly Gaussian or if it follows another distribution. Hint: Think about histograms and the Chi-Squared tests.
5. Perform PCA on the data and plot the resulting data in the new space along with their uncertainty ellipses
6. Compare the estimated uncertainty of the measurement process itself with the statistical uncertainty of the observed end-poses

2.3 Deliverables 2

Update your previous week's report and add a description of the test for a match with a Gaussian distribution. As well as a description of the observed differences between the observed 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 Task 3: Calibrating an Optical Tracking System

3.1 Setup

You will use the provided web cam which you can mount on a tripod or other fixture and a calibration tool to determine the intrinsic and extrinsic parameters of the provided cameras. The actual calculation can be performed using either i) the camera calibration toolbox from Matlab or ii) the OpenCV C++/Python library (recommended option).

3.2 Task

Get a camera from your supervisors and connect it to your computer such that you can obtain still images. Before you start calibration, make sure that the **auto focus** option is **disabled** and camera's **resolution** is set to the **highest possible**. Setup a camera calibration toolbox for either Matlab or OpenCV; read and understand its manual. Create a calibration tool (checkerboard/makers) as required by the chosen software library (this tool is **not** provided by your supervisors) and perform the necessary image captures to allow the toolbox to compute your camera's optical parameters. In particular:

1. Read the documentation about camera calibration that comes with the calibration toolbox.
2. For the Matlab toolbox look at [1] to get a brief description of calibration parameters. For a worked-out calibration example see [2]
3. For the OpenCV toolbox look at [3] to see a worked-out calibration example. For additional resources see [4] [5] [6]
4. Mount the camera on a tripod, build a checkerboard or markers that you can present to the camera from different perspectives, and run a calibration using an appropriate number of images to ensure reliable results.

3.3 Assignment 3.1

1. Read and understand the following paper (provided on LEA): Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1330-1334, 2000.
2. Based on your understanding of camera calibration, design a calibration setup and detail the actual calibration process. Test the provided camera on your laptop and ensure you can capture still images.

[1] http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html

[2] http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html

[3] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html

[4] https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[5] https://www.learnopencv.com/camera-calibration-using-opencv/

[6] https://medium.com/analytics-vidhya/camera-calibration-with-opencv-f324679c6eb7

3. Perform the camera calibration experiment. Document all relevant aspects needed to asses the quality of your obtained camera parameters.

3.4 Deliverables 3.1

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 Task 4: Measuring the Accuracy and Precision of a KUKA youBot Arm

This experiment is designed to estimate the *accuracy* and *precision* of a $5\ DoF$ robotic arm. The experiment involves a robot performing different *pick and place* tasks; in particular, our robot (a KUKA youBot arm) will be placing objects with three different masses and in three different poses (straight, left, right). The object **poses** $(x,\ y, \theta)$ are determined by tracking *ArUco* markers that are placed on top of the objects; in particular, an external vision system (a camera) is placed above the arm's workspace in order to determine the marker poses.

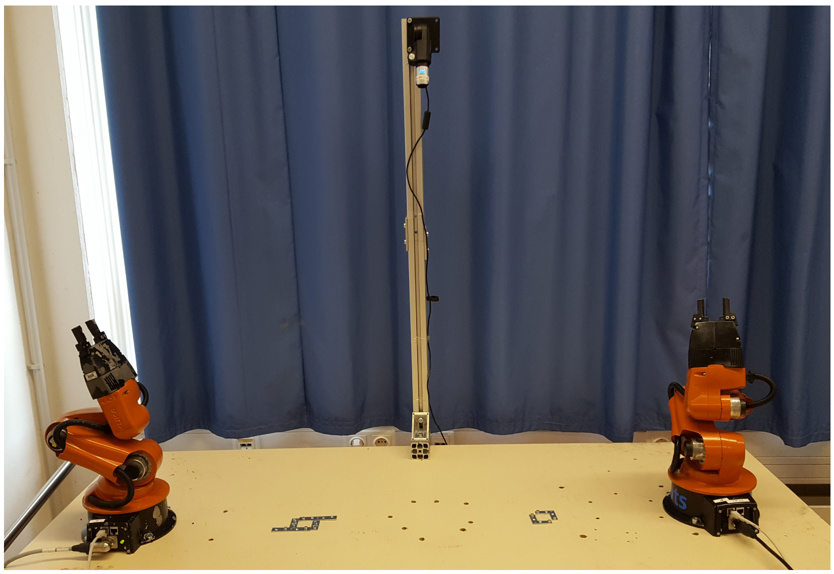

Figure 1: KUKA youBot arms used for performing pick and place tasks together with an external vision system (a Microsoft LifeCam)

Your task is to perform the experiment 20 times for each object-place combination, which results in 180 experimental trials in total. At the end, you need to perform a statistical analysis of the data in order to estimate the achieved accuracy and precision of the youBot arm motions.

4.1 Setup

Two KUKA youBot arms (IDs 1 and 3) are placed in our C022 lab and both of them are available for performing the experiment (that is, two groups can run the experiment at the same time). In particular, each group of students will use **one** youBot arm to conduct the experimental trials. Each arm is preprogramed to perform three different placing motions; the ground-truth initial and final poses for each of the three motions are given below and the values are expressed with respect to the world frame (its origin coincides with a point on the experiment-table):

Table 1: youBot arm 1

Pose	$x(m)$	$y(m)$	$\theta(rad)$
Pick	0.103	-0.142	-1.433
Straight	0.093	-0.013	-1.517
Left	0.260	-0.086	-1.829
Right	-0.105	-0.082	-0.962

Table 2: youBot arm 3

Pose	$x(m)$	$y(m)$	$\theta(rad)$
Pick	0.221	-0.366	-1.663
Straight	0.254	-0.249	-1.722
Left	0.524	-0.383	-2.040
Right	-0.055	-0.259	-1.019

A complete list of the hardware and software components that are provided for performing the experiment is given below:

Hardware

- A KUKA youBot arm placed on a table in the C022 lab
- A computer used for controlling the arm
- Objects with three different masses; these will be used by the robot for picking and placing
- Visual markers attached on the top of the objects
- An external camera placed above the robot's workspace and connected to an additional PC; you are not going to use this PC explicitly
- A fixed container on the table which ensures that the initial object position is kept constant throughout the experimental trials

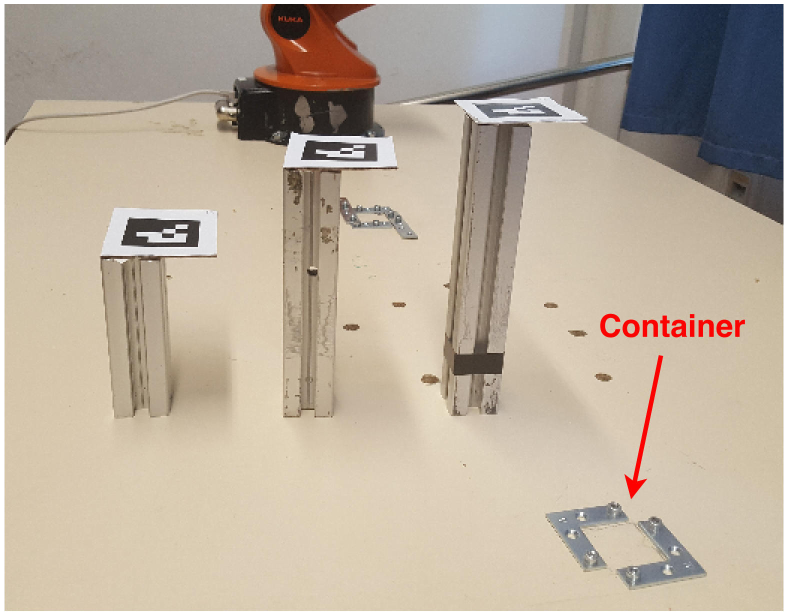

Figure 2: Objects with different masses and containers used for initial object placement

Software

- KUKA youBot drivers required for controlling the arm; these are set up on the provided computer
- Additional control script with pre-saved pick and place poses for the arm (also available on the provided computer); you will be using this script to command the robot to move in one of the three predefined placing poses
- A marker pose subscriber-script (this program is based on python2 and it is available for download on LEA); this script subscribes to the poses of the ArUco markers that are attached to the objects, which are published over ZMQ by a camera server

Guidelines For making sure that the experiment is performed correctly, several conditions must be satisfied throughout the experimentation process:

- An object should be placed in the container only **after** the arm has reached a pregrasp pose (see figures 3 and 4).

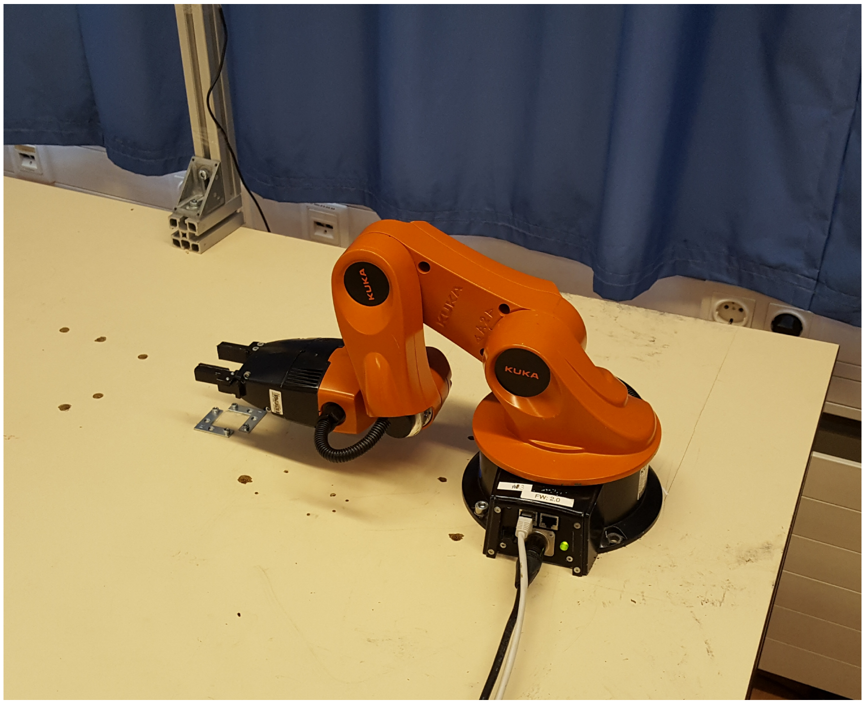

Figure 3: A KUKA youBot arm in its pregrasp pose

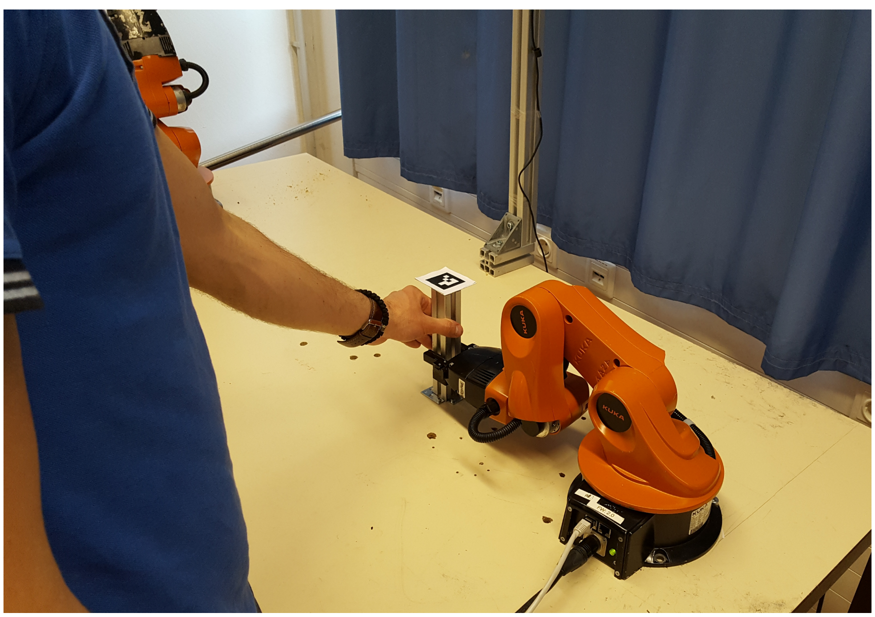

Figure 4: An object placed in the container after the arm has reached the pregrasp pose

- The chosen object must be placed in the container properly in order to ensure that the picking pose is always the same. More precisely, the object must not be touching any part of the upper container's surface, i.e. it must be placed **in between** the four borders of the container (see figure 5).

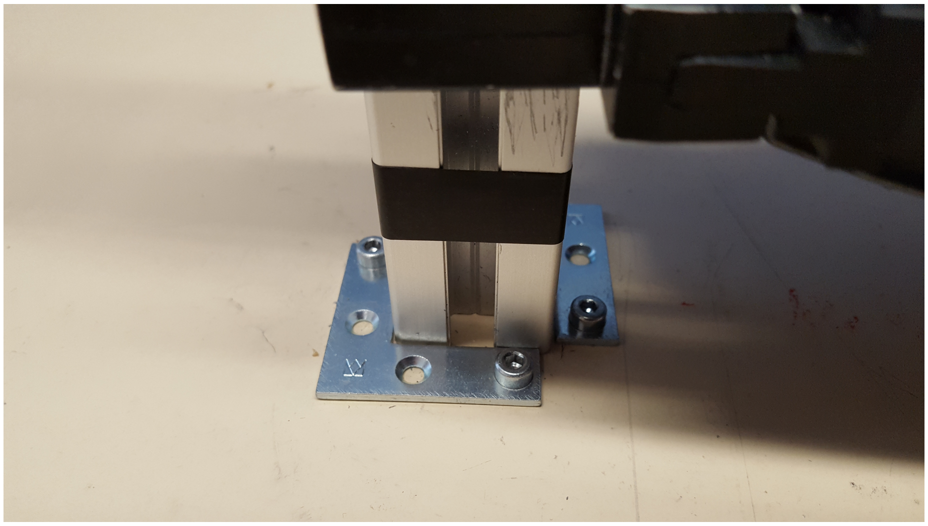

Figure 5: An object placed in the container; this ensures that the initial (grasping) position is constant

- Additionally, you must ensure that the object is placed in the container with a constant orientation. i.e. the marker attached to the object must always have the **same orientation** (in order for the ground-truth orientation measurements to make any sense). The correct orientations for the objects used by arms 1 and 3 are shown in figures 6 and 7 respectively.

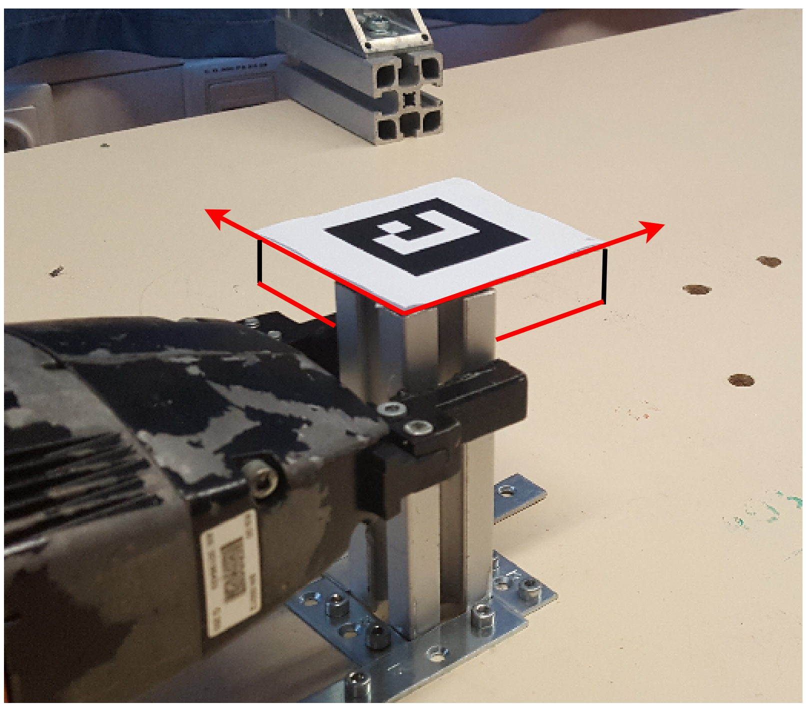

Figure 6: Correct orientation of an object used by youBot arm 1

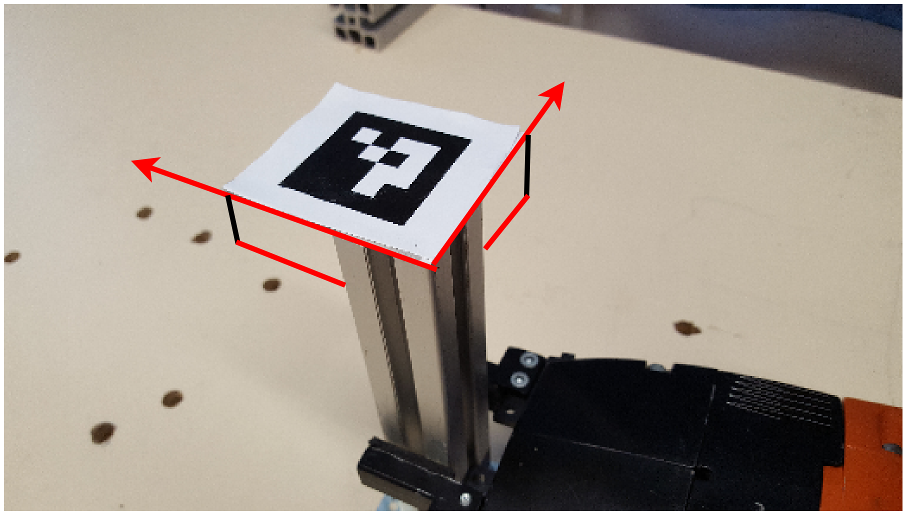

Figure 7: Correct orientation of an object used by youBot arm 3

- You must ensure that there is no object in the workspace of the arm while it is performing a motion since collisions with objects may damage the arm.
- In case of an **emergency situation**, the arm can be shut down by pressing the **small green button** that is located next to the arm's power cord (see figure 8). If such a situation occurs, the arm must be held while the button is being pressed; this is necessary to avoid the arm falling on the table and damaging itself. After the arm's motion is stopped, the arm should be set in its folded configuration (which is illustrated in figure 1).

Figure 8: The emergency (shutdown) button on a KUKA youBot arm. A green color of the button represents the running state of an arm, while a red color of the button would represent the state where all motors and controllers in the arm are shutdown.

- While performing the experiment, the light distribution in the lab should be **uniform**. In order to satisfy these lighting conditions, **close the lab curtains completely** and **turn on the lights** in the lab (as shown in figure 1). These conditions are important for producing good object pose estimates by the external vision system.
- Lastly, while recording the executed placing poses, you must ensure that the arms used by you and by your colleagues are **not moving**, i.e. you must wait until both arms are in a static pose before you run the script for gathering the estimated data. This is important for ensuring that the camera measurements are not corrupted by the **shaking table**.

4.2 Experiment Workflow

4.2.1 Step 1

The first part of the experiment involves running the software components that are required for controlling the arm and recording observable poses for all object-place combinations.

Before starting the experiment, you should make sure that the selected youBot arm is turned on. Additionally, you must connect the control computer to the corresponding youBot arm. Each given computer has a preconfigured static Ethernet connection with the arm (called *youBot_static_connection*). For using the control software components, it is necessary to open four terminals on the provided computer and run the following components:

```
roscore
roslaunch youbot_driver_ros_interface youbot_driver.launch
roslaunch youbot_moveit move_group.launch
roslaunch youBot_placing_experiment youBot_placing.launch
```

In order to familiarize yourself with the provided setup and the task in this experiment, you should see the example videos provided on LEA. The experiment should be performed 20 times for each object-pose combination. More specifically, the robot should be placing objects with three different masses and in three different positions (straight, left, right); this results in nine object-pose combinations, so the complete experiment involves 180 experimental trials.

As already described before, an external vision system is observing the poses of the objects that are placed by the arm. The generated observation data is published over the lab network using a ZMQ publisher that runs on a separate machine. For collecting the observation data, namely the values of the observed object poses, you need to download a ZMQ (python2-based) subscriber script (*subscriber.py*) from LEA. The computer on which the ZMQ script is downloaded and used is arbitrary, i.e. you can use your own laptop for this. Before running the subscriber, you should connect your computer to the lab WiFi network (the password is written on the router in the lab). For running the script, the following command should be executed in a terminal:

```
python <path_to_downloaded_script>/<script_name>.py <marker_ID>
```

Here, the *marker_ID* argument can take the values 0 or 1 depending on the youBot arm that is used in the experiment: youBot arm 1 should be using the objects with marker ID 1, while youBot arm 3 should be using the objects with marker ID 0. If these conditions are violated, the data received from the subscriber script will not correspond to the ground-truth values and the results of the experiment will not be fully usable. The marker IDs are written on the white surface of the corresponding markers.

For each experimental trial, you are advised to take multiple object pose readings in order to estimate the final pose of an object after placing. However, the subscriber script is already programmed to publish 50 camera readings, each time it is called from a terminal. This is because the camera pose estimates may have slight variations due to camera errors - even when an object is static - so by taking multiple measurements and filtering those, we are minimizing the effect of this error. The best way to obtain such data is to run the subscriber script only after the arm has completed the placing task, i.e. after the object has been placed and the arm has moved back (as shown in figure 9 and the LEA instruction video). These 50 camera-readings should be sufficient for each trial.

Figure 9: A KUKA youBot arm in its post-place position in from of a placed object

The filtering process of the pose estimates involves (i) averaging the camera readings provided by the script, so that a single experimental trial is represented by a single pose and (ii) detecting and removing any outliers from the noisy data. **However, you should also keep the raw pose measurements (all camera readings before filtering) because we want to use them when analyzing the data later.**

4.2.2 Deliverable 4.1

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?)

4.2.3 Step 2

After performing the experiment and gathering all the necessary data, you need to analyze the data and estimate the accuracy and precision of the used KUKA youBot arm. **All of the outlined tasks should be performed with the data that is combined from all of groups**. We are particularly interested in the following:

- Knowing the ground-truth values of the expected final object poses, what is the accuracy of the arm? What about its precision?
- Does the distribution of the final object poses follow a Gaussian distribution?
- How much the change in shape and mass of the grasped object (large-medium-small) affects the precision and accuracy of the robot? Is this effect statistically significant?
- How much the change of the placing pose (left-straight-right) affects the precision and accuracy of the robot? Is this effect statistically significant?
- Which of the above-mentioned variations (change in object vs change in placing-pose) affects the precision and accuracy of the robot more?
- If we are to use a single camera measurement for determining the final object pose instead of multiple filtered measurements (average of 50 camera readings), is the effect on the final pose estimates significant?

For analyzing the data, you should choose a statistical analysis technique that is most suitable for answering the above question (once again using what you have already learned during the LEGO experiment).

4.2.4 Deliverable 4.2

Update your previous week's report and include the following:

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 you've 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

A Appendix

A.1 Data visualization examples

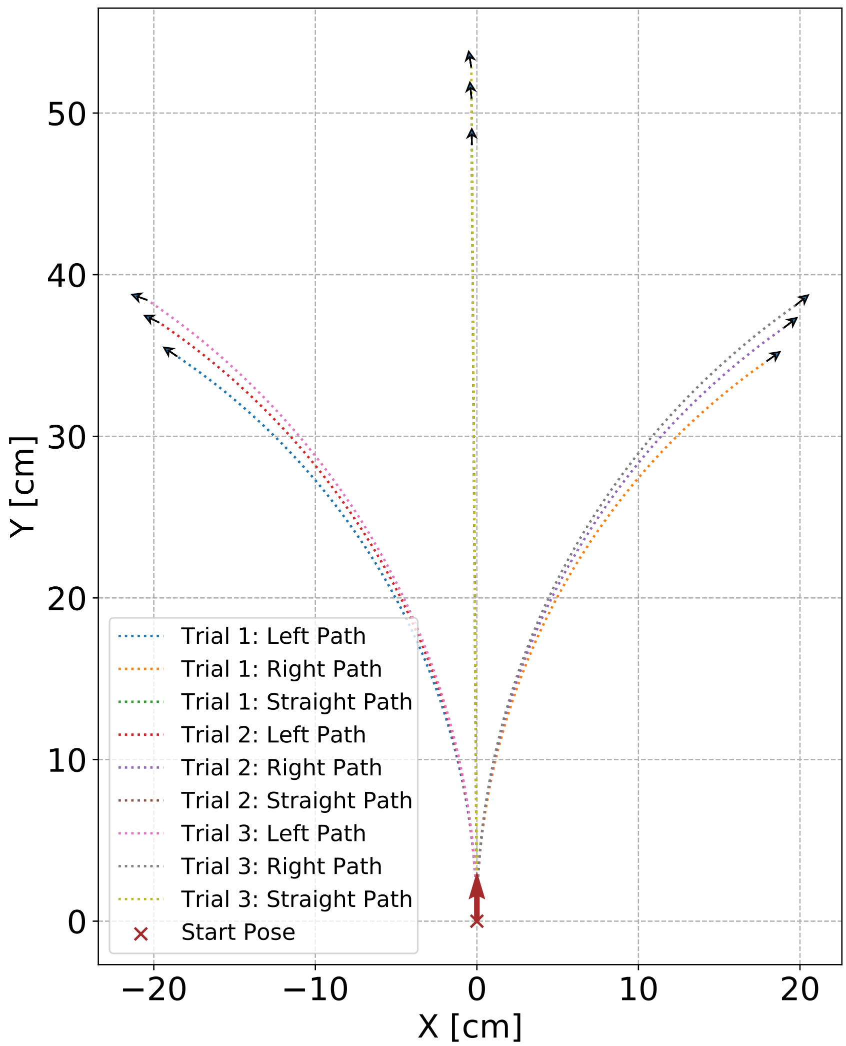

Figure 10: Example for visualizing EV3 motion paths: data generated by the robot's control script from the encoder readings.

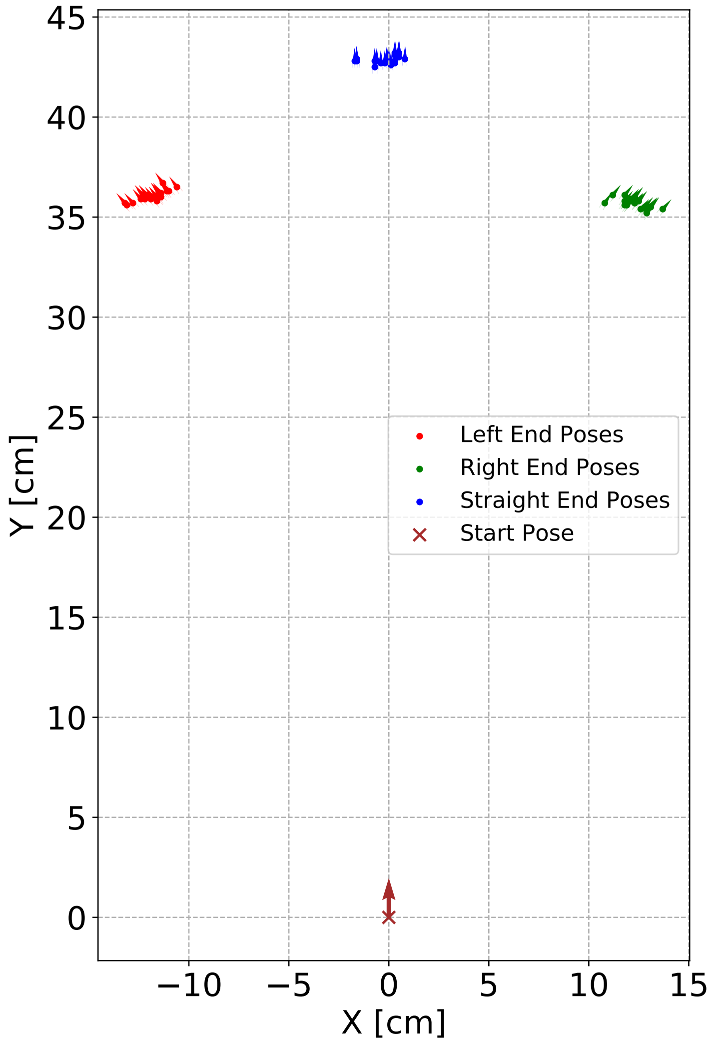

Figure 11: Example for visualizing EV3's end poses: data gathered from the process of manual measurements.

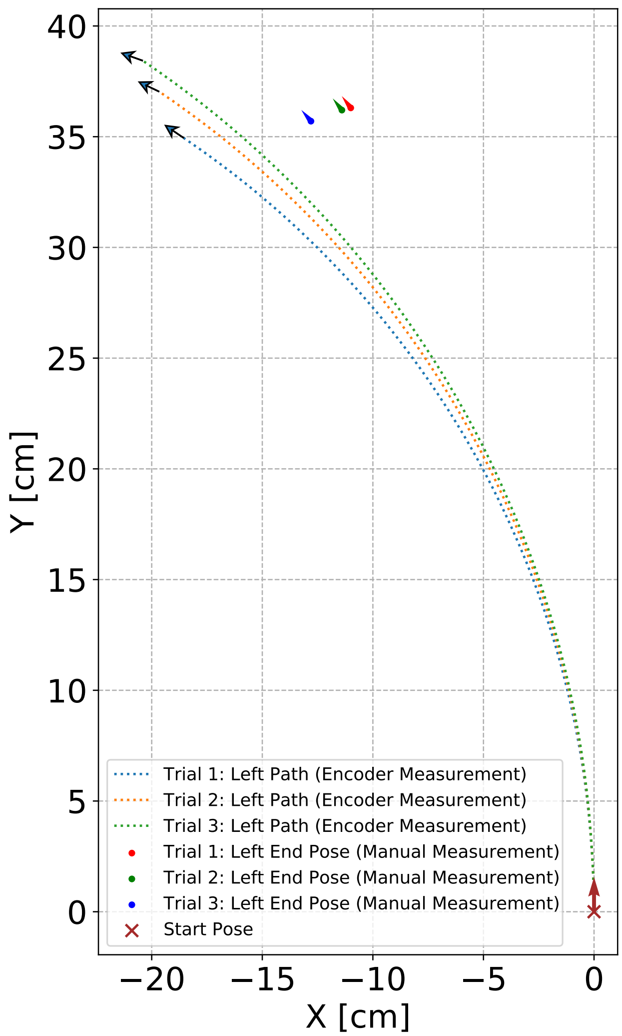

Figure 12: Example for visualizing only one type of EV3's motion, but in this case visualization covers both: i) data gathered from the process of manual measurements and ii) data generated by the robot's control script from the encoder readings.

A.2 Standard Tables

	X axis (cm)	Y axis (cm)	Orientation (deg)
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0
4	0.0	0.0	0.0
5	0.0	0.0	0.0
6	0.0	0.0	0.0
7	0.0	0.0	0.0
8	0.0	0.0	0.0
9	0.0	0.0	0.0
10	0.0	0.0	0.0
11	0.0	0.0	0.0
12	0.0	0.0	0.0
13	0.0	0.0	0.0
14	0.0	0.0	0.0
15	0.0	0.0	0.0
16	0.0	0.0	0.0
17	0.0	0.0	0.0
18	0.0	0.0	0.0
19	0.0	0.0	0.0
20	0.0	0.0	0.0

Table 3: Common table structure for storing measurements in the LEGO EV3 Robot and KUKA youBot experiments

Table 4: Common table structure for storing statistics in the LEGO EV3 Robot experiment

	Direction	Random Variable	Mean	Variance	Accuracy	Chi Value	P Value	Null Hypothesis
1	Left	X	0.0	0.0	0.0	0.0	0.0	Reject
2	Left	Y	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
3	Left	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
4	Right	X	0.0	0.0	0.0	0.0	0.0	Reject
5	Right	Y	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
6	Right	Theta	0.0	0.0	0.0	0.0	0.0	Reject
7	Straight	X	0.0	0.0	0.0	0.0	0.0	Reject
8	Straight	Y	0.0	0.0	0.0	0.0	0.0	Reject
9	Straight	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept

	Size	Direction	Rand. Var.	Mean	Variance	Accuracy	Chi Value	P Value	Null Hypothesis
1	Large	Left	X	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
2	Large	Left	Y	0.0	0.0	0.0	0.0	0.0	Reject
3	Large	Left	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
4	Large	Right	X	0.0	0.0	0.0	0.0	0.0	Reject
5	Large	Right	Y	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
6	Large	Right	Theta	0.0	0.0	0.0	0.0	0.0	Reject
7	Large	Straight	X	0.0	0.0	0.0	0.0	0.0	Reject
8	Large	Straight	Y	0.0	0.0	0.0	0.0	0.0	Reject
9	Large	Straight	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
10	Medium	Left	X	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
11	Medium	Left	Y	0.0	0.0	0.0	0.0	0.0	Reject
12	Medium	Left	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
13	Medium	Right	X	0.0	0.0	0.0	0.0	0.0	Reject
14	Medium	Right	Y	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
15	Medium	Right	Theta	0.0	0.0	0.0	0.0	0.0	Reject
16	Medium	Straight	X	0.0	0.0	0.0	0.0	0.0	Reject
17	Medium	Straight	Y	0.0	0.0	0.0	0.0	0.0	Reject
18	Medium	Straight	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
19	Small	Left	X	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
20	Small	Left	Y	0.0	0.0	0.0	0.0	0.0	Reject
21	Small	Left	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
22	Small	Right	X	0.0	0.0	0.0	0.0	0.0	Reject
23	Small	Right	Y	0.0	0.0	0.0	0.0	0.0	Suggest to Accept
24	Small	Right	Theta	0.0	0.0	0.0	0.0	0.0	Reject
25	Small	Straight	X	0.0	0.0	0.0	0.0	0.0	Reject
26	Small	Straight	Y	0.0	0.0	0.0	0.0	0.0	Reject
27	Small	Straight	Theta	0.0	0.0	0.0	0.0	0.0	Suggest to Accept

Table 5: Common table structure for storing statistics in the KUKA youBot experiment