HOCHSCHULE BONN-RHEIN-SIEG

Scientific Experimentation and Evaluation Quality Assurance Guidelines

SUMMER SEMESTER 2021

In the SEE course, students are assigned to perform tasks that include practical experimentation, measurements, (statistical) data analysis, and data visualizations. To support students in performing those tasks, this document aims to provide a set of suggestions on how certain technical aspects should be handled. More specifically, since on weekly basis students are also conducting assignment-peer-reviews for their colleagues, this document outlines some examples of common mistakes made during the semester work and guidelines on how those errors can be addressed/corrected. This manuscript is updated on a semester basis and any suggestion/feedback from students on how it can be further improved is welcomed. Each example that is outlined bellow will have a unique ID.

1 Tasks that repeat during the semester

1.1 Presentation of used program blocks and parameters in the report

During the semester, students are required to present programs and parameters that were used to run the robot and produce data-analysis results. Common mistakes made in performing this task are the following.

- **1.1.0** *Program/code blocks added in report as a screenshot from an IDE*: this is not a professional way of adding software blocks in a public document. Instead, either directly code snippets or pseudocode (derived based on those code blocks) should be added in the LaTeX environment that is used for generating the report. For example, if the programs are written in Python language you can use *pythonhighlight* LaTeX package for adding code snippets. However, if the pseudocode option is selected, it can have style of any language (e.g. Python, C, etc.).

- **1.1.1** *Program parameters listed in the report without associated units*: wherever applicable, a parameter should be accompanied with a unit (*kg, m, %*, etc.).

1.2 Data visualization/presentation (figures and tables) in the report

Multiple times during the semester, students are assigned to present $i)$ measurement data gathered during an experiment and $ii)$ results from analyzing those measurements, in form of figures and/or tables. Common mistakes made in performing this task are the following.

Figures formatted inadequately

- **1.2.0** *Missing axis labels and units.*
- **1.2.1** *Missing legends describing different elements in the figure.*
- **1.2.2** *Missing grid lines (grids) in the figure.*
- **1.2.3** *Missing appropriate caption under the figure.*
- **1.2.4** *Visualization of only single-group data included in the report:* since the students are instructed to combine (merge) data from all of the groups, it it necessary to additionally visualize the complete data set and clearly label (e.g. using different colors) data clusters of each group separately.
- **1.2.5** *Figures added as a screenshot from a IDE (e.g. Jupyter notebook):* a good way to generate figures for a public document is to instruct the program to export/save those plots in a PDF, SVG, or TIFF file formats and add those instead of screenshots.
- **1.2.6** *Figure's size is way too small for a good readability:* students do not have a limit on number of pages in their report and they can enlarge figures as far as document-margins allow.
- **1.2.7** *Axes in figures do not have equal scales:* if in a figure axes are described with the same units, those axes should have equal scales. See some examples on how to enable this scale in your program: Python and MATLAB. On each graph where equal-axes-scales are applicable, students should draw small circles, which should be visible behind the main data points/functions (by using some light color, e.g. gray and setting appropriate order), such that the axes-scales can be directly observed.
- **1.2.8** *Robot's orientations not presented or unclear in figures:* in places where it is requested to visualize full poses (e.g. measured end-poses of the LEGO EV3 robot or objects in the youBot experiment), it is necessary to clearly visualize orientations as well. Most common way to add ("draw") those is using arrows with appropriate sizes and colors.

See figure 1 for an example of a well formatted figure that outlines certain data points collected during an experiment.

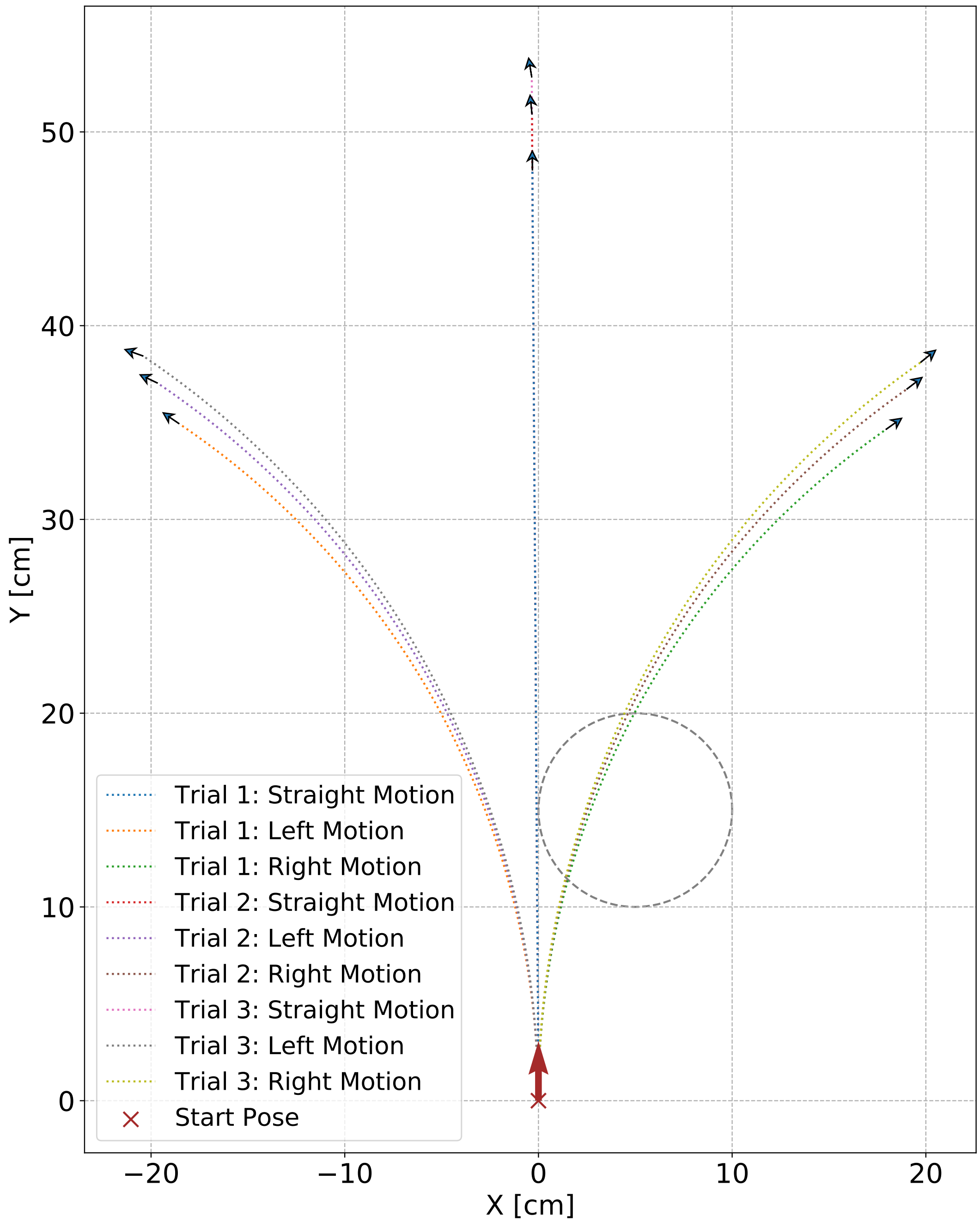

Figure 1: An example for visualizing motion paths of a LEGO EV3 robot: data generated by the robot's control script from the encoder readings.

Tables formatted inadequately

- **1.2.9** *Missing units where applicable.*
- **1.2.10** *Missing columns and rows:* if for presenting a certain type of data in the report, a template table is outlined in the SEE Manual document, students should follow this template when creating tables. More specifically, all of the columns/rows that are suggested there, should also exist in student's reports.
- **1.2.11** *Large tables split in half over two pages:* if a table is too large to fit in a portrait page, try placing it in a landscape environment. Or, if this is not sufficient, fit the data two or more separate tables.
- **1.2.12** *Inconsistent number of digits after the comma:* if certain columns/rows represent different variables but they share the same unit, all numbers in those fields should have the same number of digits after the comma.

See table 1 for an example of a well formatted table that present certain data points collected during the experiment process.

Table 1: Common table structure for storing measurements in the LEGO EV3 and KUKA youBot experiments.

	X axis (cm)	**Y axis (cm)**	**Orientation (deg)**
1	-19.32	24.01	89.2
2	-19.28	23.12	88.5
3	-19.36	24.32	89.8
4	-19.27	24.35	91.1
5	-19.19	25.65	88.5
6	-19.31	23.83	87.7
7	-19.41	23.15	85.3
8	-19.45	24.00	91.1
9	-19.19	25.67	90.7
10	-19.22	24.85	92.2
11	-19.32	25.83	87.2
12	-19.28	23.31	88.5
13	-19.32	25.27	86.9
14	-19.32	24.23	85.4
15	-19.41	24.79	90.3
16	-19.43	23.14	91.7
17	-19.35	25.16	85.8
18	-19.37	23.84	87.1
19	-19.00	24.36	89.0
20	-19.19	25.57	85.4
21	-19.48	24.13	86.3

1.3 Data analysis process

Multiple times during the semester, students are assigned to perform some data-analysis procedures, using common methods from statistics. Repeating mistakes made in performing this task are the following.

- **1.3.0** *Analysis of only single-group data presented in the report:* since the students are instructed to combine (merge) data from all of the groups, it it necessary to analyze the complete data. However, it is **not** necessary to separately perform analysis on only single-group data.
- **1.3.1** *Inadequate number of bins used in Histogram graphs and Chi-Squared tests:* students should explicitly define the number of bins in these analyses (e.g. by following the formula presented in one of the first lectures in the SEE class), i.e. they should not use the default bin-number specified by the used software libraries. Additionally, the number of bins used for the Chi-square test should be the same as number of bins for histogram analysis.
- **1.3.2** *Missing a description on the Chi-Squared test usage:* since the *Chi-Squared* test can be used for many different applications in statistics, students should in their report $i)$ include a brief description that explains how exactly was this method used/applied for their use case and $ii)$ summarize what do the results of this test tell us, i.e. mean for the gathered data.
- **1.3.3** *Incorrect usage of the Chi-Squared test functions in software libraries:* students should go through the Chi-Squared test documentation of the particular library and use associated methods appropriately (e.g. the method should compare the observed pose distribution to a Gaussian distribution).
- **1.3.4** *Incorrect orientation and position of uncertainty ellipses in the PCA analysis graphs:* the new space in which data is presented after the PCA analysis, should be spanned by two Eigenvectors, such that the larger Eigenvector defines the direction of the larger data spread. Smaller vector defines the orthogonal direction. Thus, the shape of ellipses should follow these directions. The center of those ellipses should always be at 0.0.

2 Tasks specific to certain assignments

2.1 Design of the LEGO-based differential drive and measurement system

The first task in this course requires a design and construction of a system that consists of $i)$ a LEGO EV3-based differential drive robot and $ii)$ a facility for measuring observable end-pose variations in three different constant-velocity robot motions. Common mistakes made in performing this task:

- **2.1.0** *Mechanical measuring devices designed to touch the cardboard sheet constantly during the robot's motion*: this design decision also introduces considerable disturbances in robot motions. Robot behaviour should be, as much as possible, independent of the measurement facility used in the experiments.

- **2.1.1** *Description of the "ensuring identical start positions" procedure missing in the report*: this procedure needs to be clearly described in report using figures and text.
- **2.1.2** *Images covering all sides of the robot and measuring system missing in the report.*
- **2.1.3** *Textual description of the measuring method and procedure missing in the report.*
- **2.1.4** *Description of the possible pose-transformations missing in the report*: since we are interested in seeing end-poses that correspond to the center point of robot's axle it is – in certain cases (depending on the design of overall measurement facility) – necessary to perform suitable transformations of all measured (raw) poses. If such transformations are used (applied) during the data acquisition, that process should be documented in the report.

2.2 Camera calibration

One of the tasks assigned to students during the semester is to calibrate a camera, i.e. use software tools and physical markers (e.g. a checkerboard) to determine its intrinsic and extrinsic parameters. Common mistakes made in performing this task are the following.

- **2.2.0** *Auto focus option is **not** disabled:* students can use any method that they find suitable to set this parameter, e.g. using a terminal-based Linux program "v4l2-ctl".
- **2.2.1** *Camera's resolution is **not** set to the highest possible:* students can use any method that they find suitable to set this parameter, e.g. using a terminal-based Linux program "v4l2-ctl".
- **2.2.2** *Insufficient number of images used for calibration:* for producing good calibration results it is usually necessary to have **at least** 25 images of the marker. However, this number can depend on the particular camera, software library and overall setup used for the experiment.
- **2.2.3** *Only a single configuration of the marker/calibration board used while taking the photos:* for producing good calibration results it is necessary to make photos with the marker being configured in different poses (making changes in orientations and not just in positions). A change in the orientation and position is good as long as the whole marker is clearly seen by the camera. Moreover, while changing those configurations, it is also necessary to – for only certain number of photos – cover edges of the camera's view, since major imperfections of the camera are located in those edges. See figure 2 for a few examples of well chosen marker configurations.
- **2.2.4** *Light distribution not being uniform during the image capturing:* for producing good calibration results it is always necessary to keep the light distribution uniform in the environment where the images are captured (in the room/lab where students choose to perform the experiment).
- **2.2.5** *Appropriate units missing in the report:* many software parameters, aspects of the marker and results of the calibration have associated units thus, it is necessary to correctly include those in the report.

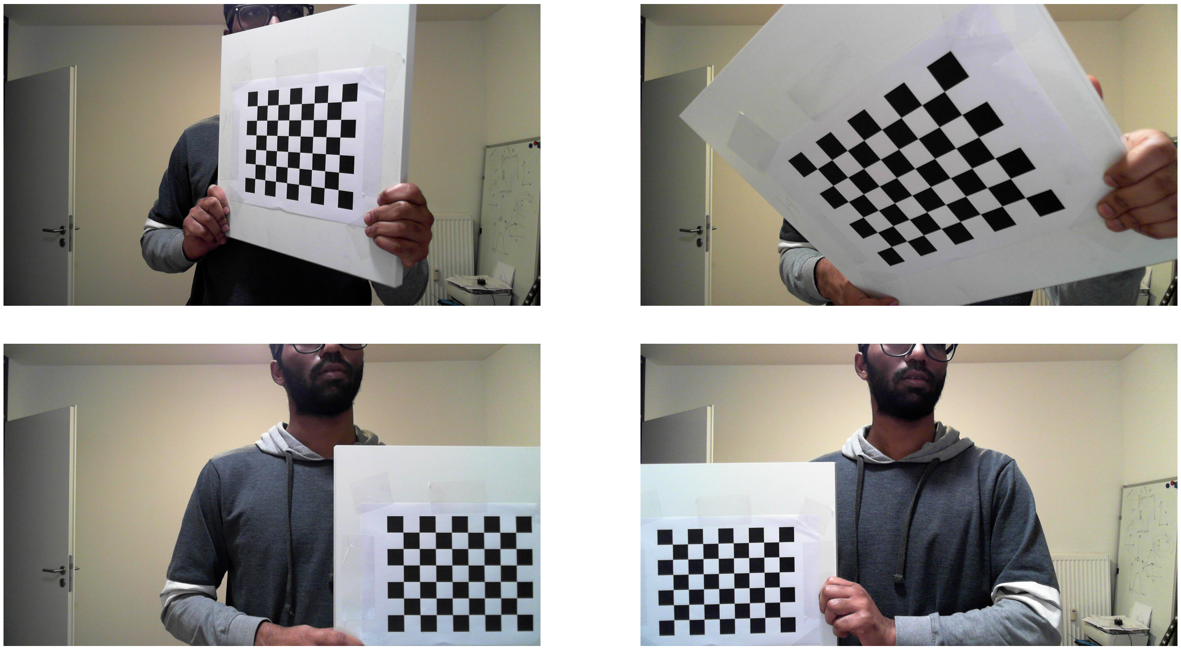

Figure 2: Examples of well chosen images for the camera calibration experiment.

- **2.2.6** *Appropriate description of the extrinsic parameters missing in the report:* it is insufficient to provide general description of these parameters – it is also necessary to outline their meaning w.r.t. student's particular physical and software setup. What is the semantics of computed position and orientation vectors? For instance, pose of which physical point/object do they represent (a point on checkerboard, or camera's center point, or something else in the environment), w.r.t. which frame of reference are they represented, etc.?

2.3 youBot experiment

The last task assigned to students during the semester is performing multiple experiments with a $5\ DoF$ KUKA youBot arm. This experiment is designed to estimate robot's *accuracy* and *precision* while it performs different *pick and place* tasks with three different masses. The object poses are determined by tracking markers that are placed on top of the objects with an external vision system (camera). Common mistakes made in performing this task are the following.

- **2.3.0** *Not being familiarized with the experiment procedure:* since the youBot experiment takes a significant amount of time to complete, it is very important for students to come in the lab prepared, so to avoid possible delays in completing this experiment. Thus, it is necessary that students carefully read and analyze experiment steps and details outlined in the SEE experiment-manual document.

- **2.3.1** *Not having well prepared "data acquisition and storage" strategy:* students should become familiar with the working principle of the subscriber-script that is used to collect a large amount of pose measurements from the server (as described in the SEE manual).

Additionally, it is necessary that students adjust/extend this script so that the program stores received data in a automatic time-efficient manner. See bellow an example of the **raw** JSON message that is received by the subscriber's script. Here, position values (X and Y axis respectively) are given in *[m]* units, while orientation value (Z axis) is given in *[rad]* unit. This example serves to help students in developing the above-mentioned subscriber-program extension.


```json
{
  "position": [
  0.16227603525502593,
  0.06682936707104165
  ],
  "orientation": [
  0.7743281838783742
  ],
  "marker_id": 0
}
```