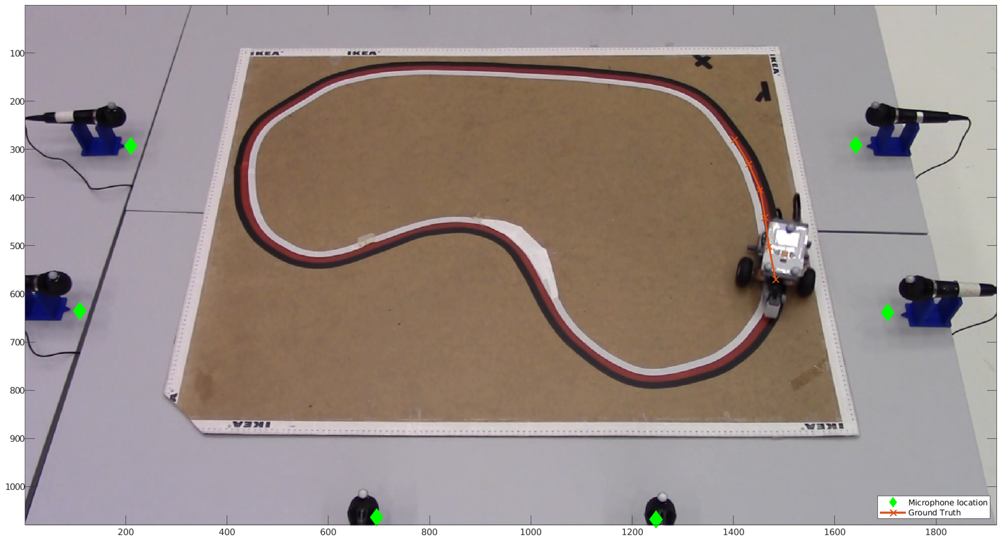
Sensor Fusion, TSRT14

Localization Using a Microphone Network

This version: 2022-05-12



REGLERTEKNIK AUTOMATIC CONTROL LINKÖPING

Name: ____________________

P-number: ____________________

Date: ____________________

Passed: ____________________

1 Introduction

This laboration is about localization using a network of sensors, a problem that is common in the sensor fusion world. A real world example of audio based localization is shooter localization. The sound of a gun being fired is quite distinctive and can be correlated with the recordings. Time of arrival at different microphones can be found, and the shooter can hence be located. Another example is localization of airplanes using radars.

You will use a network of microphones to localize a target that is emitting short sound pulses. To simplify things, the profile of the pulse that is being emitted is known. Hence, correlation can be performed with the recordings to find the *time of arrival* (TOA) of each pulse at each sensor. If the time of pulse emission is known, one can easily find the distance to the target using the (also assumed known) speed of sound. This gives a radial distance to the target, but not a bearing, and thus the target location could be anywhere on a circle (assuming a 2D world). Combining such circles for all the sensors, the intersection, which is where the target is located, can be found.

As in most real world cases, in this laboration, the time of emission is unknown. When all sensors are synchronized, one method of performing localization when time of emission is unknown, is to use *time difference of arrival* (TDOA). Knowing the TDOA, one can use the speed of sound to find the relative difference in distance between sensors and the target. Given all relative distances in the network, the location of the sound source can be found. For further information on TOA and TDOA, see Chapter 4 in [1].

2 Purpose of Laboration

During the laboration a LEGO® MINDSTORMS robot will autonomously follow a planar, closed loop track, while simultaneously repeatedly playing a sound pulse, see front cover. The purpose of the laboration is to track a sound-emitting robot by recording audio with eight microphones, which will be used to provide TDOA measurements.

3 Equipment

The equipment used is briefly explained below.

Sound Card

The sound card is a *Behringer UMC1820*, see Figure 1, which allows up to 18 channel recording and 20 channel playback simultaneously. The sound card has a software interface called *UMC Control Panel* which enables the user to configure some settings such as sample rate etc. For more information about the sound card, see `https://bit.ly/2uxamRp`.

(a) Front view **(b)** Rear view

Figure 1: *Behringer UMC1820* sound card.

Microphones

There are eight microphones used for recording. Make sure that the microphones are switched on before starting the recordings! There are eight microphone stands which are highly recommended to use.

Target

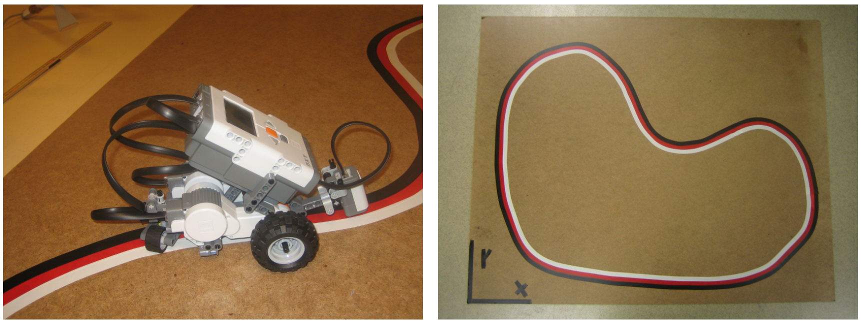
The target in this laboration is a LEGO® MINDSTORMS robot, see Figure 2a, which uses a light sensor to detect the different colors of the tapes that constitute the trajectory. The robot comes with pre-installed programs, see Section 6. For example, `SFdrive` makes the robot follow the taped lines while playing the sound signal described in Section 4.

Environment

The laboration environment, see Figure 2b, is a $0.991 \times 1.222\,\mathrm{m}^2$ wooden board with three colors of tape in a loop. The loop shape makes it possible to run the trajectory several times.

TDOA quality estimation

On the other side of the wooden board, a circle arc with radius $0.7\,\mathrm{m}$ is indicated. Place the microphones and the robot similar to Figure 3 when collecting data for TDOA variance and bias estimates. This orientation of the robot ensures that its speaker is facing the microphones.



(a) LEGO® MINDSTORMS robot. **(b)** Robot environment

Figure 2: Robot & robot environment.

4 Signal Processing

The robot emits a pulse train, where each pulse is an *orthogonal frequency-division multiplexing* (OFDM) signal generated to be insensitive to noise disturbances. The pulse $p(t)$ is generated from a finite number of discrete frequencies $P(f)$. An example of such a signal is shown in Figure 4. Since the emitted signal is known, it is possible to correlate the measured signals of the microphones with a copy of the pulse. This gives the cross-correlation and it will have quite distinct maxima. The maxima which will be the TOA on every channel. There will also be other maxima due to noise, echos etc. but these are hopefully less distinctive than the interesting peak. Since the robot is emitting a sequence of pulses, several maxima will be found. Even in situations with low *signal to noise ratio* (SNR), this method performs quite well.

In the laboration the robot will play a signal with pulse width $t = 0.1\,\mathrm{s}$ made of frequencies between $f = 800$–$1200\,\mathrm{Hz}$. The time between pulses is approximately $0.5\,\mathrm{s}$.

5 Provided Code

The following MATLAB functions are provided:

- `SFlabRun` — the main script for recording and preprocessing the data.
- `SFlabPlayAndRecord` — simultaneously plays sound on up to eight output channels and records from up to eight input channels.
- `SFlabPlotRecData` — plots the results from a recording session.

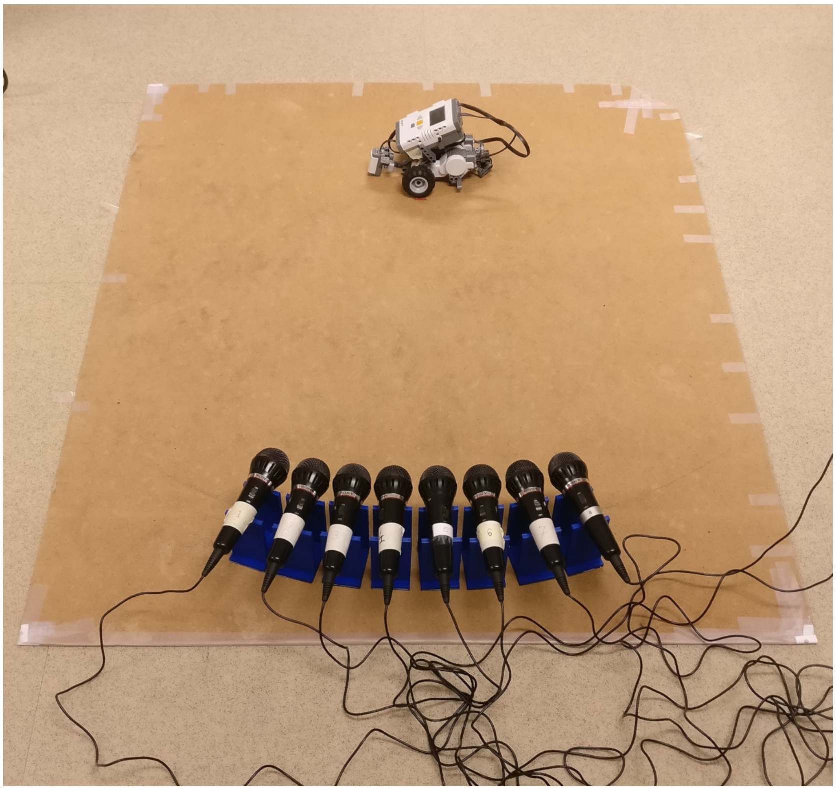


Figure 3: Suggested setup when estimating the quality of the TDOA measurements.

- `SFlabPlayRecData` — plays the recorded data from either all input channels, or from a user specified subset of the input channels.
- `SFlabFindPulseTimes` — finds the times at which pulses were detected in each input channel. These times can be used for TDOA localisation.
- `SFlabVisualizeResults` — plots the microphone positions and the trajectory estimates overlaid on the robot environment.
- `SFlabAnimateResults` — similar to `SFlabVisualizeResults`, but instead of simply plotting the given estimates, they are overlaid on top of images from a video of the collected data. Hence, you get an animation of how your estimates follows the robot around the track. **NB**: Only applicable to the pre-recorded datasets.

Type `help` followed by the function name in the MATLAB command prompt to see how to use the functions. A number of auxiliary functions are also provided, however, they are only used by the functions above and need not

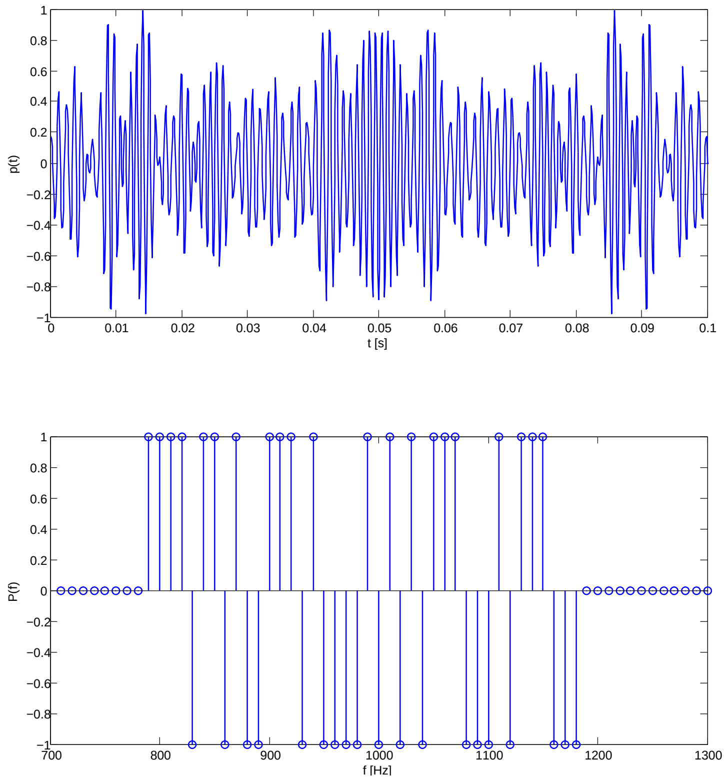


Figure 4: OFDM signal.

be called by the user. The MATLAB functions can be downloaded from the course homepage (`http://bit.ly/2uQfRL9`).

6 Robot Programs

The following *Not Exactly C* (NXC) functions are provided and stored in the LEGO® MINDSTORMS robot:

- `SFdrive` — makes the robot follow the trajectory while playing the sound.
- `SFpulse` — plays the sound described in Section 4, while the robot stands still.

The program are started the following way:

1. Press the orange button to start the robot's computer.
2. Select "My Files" and press the orange button.

3. Select “Software Files” and press the orange button.

4. Select the program of your choice and press the orange button.

To interrupt a program, press the gray rectangular button. Place the robot centered above the tape trajectory, in the clockwise direction, before the line following programs are started.

7 Preparations

- Read these lab instructions and make sure you have an overview of all the work needed to complete the lab, not only what is expected at the data collection session.
- Read Chapter 4 in [1].
- Plan at least two different microphone configurations. The two different configurations should correspond to fundamentally different situations.
- Download `SFlab1.zip` from the course homepage to your student account and unzip the files.

8 Tasks

The laboration is performed in groups of two students. Data acquisition is done in the lab, where all the necessary equipment is located. One 30 min time slot has been allocated for each group, which can be booked via the lab registration system. The remainder of the laboration is performed elsewhere using the collected data.

Note: The data collection session is just a small part of this laboration. You are expected to spend time with the tasks continuously throughout the course as you learn new techniques. Furthermore, this is an exploratory laboration; that is, part of the objective is that you yourself figure out exactly how to solve the posed problems. Hence, make sure to allocate time for the laboration work, and do not assume it can be finished the day before the deadline.

8.1 Data Collection

The first part of the lab is data collection. Perform the following tasks:

1. Use the suggested setup in Section 3 to obtain data to estimate the measurement precision.

2. Let the robot follow the trajectory, while playing the sound. Record data with at least two different microphone configurations. Do not forget to note the microphone positions. (Photos of the configuration are usually helpful later on.) The two different configurations should correspond to fundamentally different situations. Plan your configurations before the lab session. Record 45 s of data for each configuration (this corresponds to the robot driving the track approximately twice). Note the starting point of the robot, it can be used to initialize the localization.

3. Save all your recorded data using the `save` command in Matlab.

After a data recording it is important to verify that the recording was successful, here the commands `SFlabPlayRecData` and `SFlabPlotRecData` are useful. For each data set the command `SFlabFindPulseTimes` should be used to estimate the time of arrival of each pulse and each sensor. It is a good idea to run this command during the laboration to verify that all pulses are detected successfully.

8.2 Localization

When data has been recorded, the remainder of the lab should be performed elsewhere. The following tasks should be solved:

1. **Sensor calibration:** Describe the accuracy of the measurement (the estimated time of arrival) for each sensor in terms of the bias and standard deviation of its measurement noise. Use the matrix `tphat` from the calibration data to compute a vector of measurements errors `e` for each sensor (notice that this can be accomplished in different ways since the true time of arrival is unknown). Then compute histograms of the measurement errors and compare them to a normal distribution, using e.g. `histfit`. These histograms can be seen as empirical probability density functions of the measurement noise in each sensor. Also, compute the bias and the standard deviation of the measurement noise.

2. **Signal modeling:** Describe the different sensor models you will use in your localization algorithms. The models should be given both as equations
$$y = h(x) + e, \quad e \sim \mathcal{N}(0, R)$$
and as an m-file.

3. **Experiments:** Write a brief description of the microphone configurations, and the obtained data. The sensor locations and the target's initial position can, together with the m-file and noise distribution `pe` above, be used to construct a `sensormod` object. The configurations can then be illustrated with `sensormod.plot`.

4. **Configuration analysis:** Compare your two configurations using at least one of your sensor models in at least one of the following aspects:

   (a) Compute the *nonlinear least squares* (NLS) loss function $V(x) = \big(y - h(x)\big)^T R^{-1}\big(y - h(x)\big)$ on a grid over the table for a certain measurement y.

   (b) Compute the *Cramér-Rao lower bound* (CRLB) as a function of target position and represent this as a map over the table where each grid point gives a bound on the root mean square error $\text{RMSE} \geq \sqrt{\operatorname{tr}\left(I^{-1}(x)\right)}$.

   Use this map to motivate your preferred configuration.

5. **Localization:** Estimate the position of the target at each time instant. Compare at least two of the following localization algorithms:

   (a) NLS using a 3D grid search over $x = [p^T, r_0]^T$, where p corresponds to the position and r_0 to the unknown pulse emission time.

   (b) NLS using a gradient or Gauss-Newton search over $x = [p^T, r_0]^T$.

   (c) *Separable least squares* (SLS) using a *weighted least squares* (WLS) estimate of r_0 and a 2D grid search over p.

   (d) A TDOA approach, where the pairwise differences of detection times for the M sensors are used to eliminate r_0. Select one sensor as reference, construct the $M-1$ pairwise differences, and use as measurements.

6. **Tracking:** First, select two motion models. Then, compare the following filters for both models (that is, you will get four estimated trajectories):

   (a) Take the localization estimates above as the artificial measurements $y_k = \hat{p}_k + e_k$, $e_k \sim \mathcal{N}(0, P_k)$ at time instant k. Apply a *Kalman filter* (KF) (or *extended Kalman filter* (EKF) if your motion model is nonlinear), to the two different motion models in turn.

(b) Select one sensor as a reference. Make sure that this one gives good measurements. Use a TDOA measurement model with seven TOA differences. Apply a *point mass filter* (PMF), EKF, *unscented Kalman filter* (UKF), or *particle filter* (PF), to the two different motion models in turn.

7. **Sensitivity analysis:** Select one of the methods above and empirically evaluate how sensitive the result is with respect to the specified microphone locations (i.e. disturb the positions of the microphones). After all, these locations are measured by hand, and contain an uncertainty. What is the size of this uncertainty approximately, and will it affect the result?

9 Lab Report and Submission Instructions

The laboration is not examined at site, but using a written report. The report should be written according to the following guidelines

- Hard limit on 6 pages
- Use the provided LaTeX-template (`.sty` and `.tex`), which you can find on the course homepage
- Report should be able to stand on its own, i.e., it should be understandable **without** this lab memo. Further, a person that did not just perform the lab should be able to follow your presentation.
- Report should include
  - A description of how the data was gathered. This includes the microphone positions, and a motivation why the particular positions were chosen.
  - Descriptions of how you solved the tasks in Section 8.2. Include plots that illustrate your results.
  - Well supported conclusions.
  - Code in an appendix. **NB: The code is not included in the page limit.**

Lastly, keep in mind that your report will be evaluated using the criteria listed in Section 10.

9.1 Grading procedure

The submission, review and grading of the lab report consists of five stages:

First submission of the lab report. The first submission of the lab report will be used for peer review. Please write your group ID on this lab report. Do NOT write your names or personal identity numbers on the lab report. This report will not be graded by the assistant. The first submission will also be sent to **Urkund** automatically as you submit it.

Submission of the peer review. The length of the review report shall be 1–2 pages. Do NOT write any names on the peer review. Start the report by quoting your group ID as well as the group ID of the lab report you reviewed. For instructions on how to perform your peer review, please read Section 10.

Final submission of the lab report and of the review response. The lab report needs to be updated based on the peer review resulting in a final version of the report. A review rebuttal also needs to be written. In this review rebuttal, the comments of the reviewers need to be addressed and updates to the report need to be discussed. Please write your names and personal identity numbers on this version of the lab report!

Grading of the lab report, peer review and review rebuttal. The lab report, the peer review and the review rebuttal will be read by the assistant and given one of the following grades; pass, revisions needed, or fail.

Possibly submission of a revised version. If revisions are needed, a new version of the report must be handed in.

The deadlines for each of these stages can be found on the course webpage. If the updated version does not pass or the report is failed at first hand in, a new version will be read and graded in conjunction with the next course exam.

9.2 Submission

The submissions are handled through Lisam under “Submissions”. The procedure is as follows:

1. One group member submits the first draft and adds the other group members to that submission. Any students not in a submission group will not be able to get credits for the lab. Please name your group after the group ID provided by the course assistant in the beginning of the course.

2. Your submission will then be marked as "needs completion" and another group's report will be provided to you. Review the report and resubmit your review in the same interface.

3. Your submission will again be marked as "needs completion" and you will receive the peer review of your report. Make the necessary adjustments and resubmit your report in the same interface.

4. Should you require adjustments after the assistants' comments, the submission will yet again be marked as "needs completion". If so, review and submit a final report. When you have submitted your final report, the assistant will review the report and assign a pass or fail grade.

10 Peer Review

In the peer review, your task is to provide feedback to help the other students to improve their report, and also to allow you to learn from seeing how other people write their report. Shortly after handing in your lab report you will receive another group's lab report from the course assistant. Read this lab report and write a review report. Check the course homepage to find out when the review report is due. In the review report you should respond and discuss the following questions

1. Are the data sets presented clearly? Are the procedures to acquire data described in enough detail for the experiment to be repeated by someone else?

2. Is there a clear explanation of the solutions of the tasks in Section 8.2? Discuss each task 1–7 separately.

3. Are the conclusions well supported by the data, the experiment and the results? Do you agree with the conclusions? What would you like to add to the conclusions, based on the data and the results of the task?

4. What is a particular strength of this lab report? Discuss the content, not the format.

5. What suggestions can you provide to improve the overall quality of the report? Discuss clarity, readability and technical accuracy.

The questions shall be answered in the form of a discussion, present arguments for your point of view and propose alternative methods. Some more specific tips:

- Present useful criticism and make sure your comments are constructive.
- Use a positive tone and consider how you would feel if someone sent your review to you.
- Be clear and specific about things you think could be improved.
- Point out strengths as well as weaknesses.
- Use a courteous language.
- Avoid comments that might be read as insulting or inappropriately personal.

Most people ignore feedback that they find hostile, vague, or confusing. Try to keep your comments positive and specific: this will make them much more useful to your peers.

NB: The review that you write will also be evaluated by the lab assistants. It is a part of the examination of this lab.

References

[1] Fredrik Gustafsson. *Statistical Sensor Fusion.* Studentlitteratur, Lund, Sweden, 3. edition, 2018.