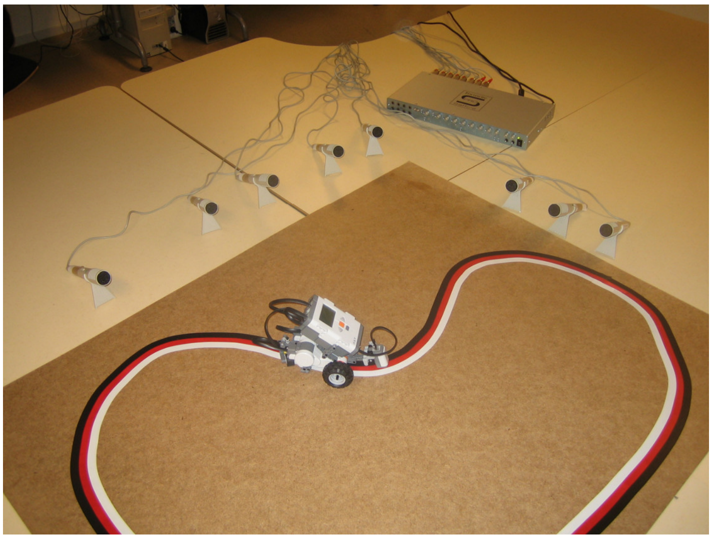
Sensor Fusion, TSRT14

Localization Using a Microphone Network

This version: March 19, 2019



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.

In this laboration 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 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.

In this laboration, as in most real world cases, 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.

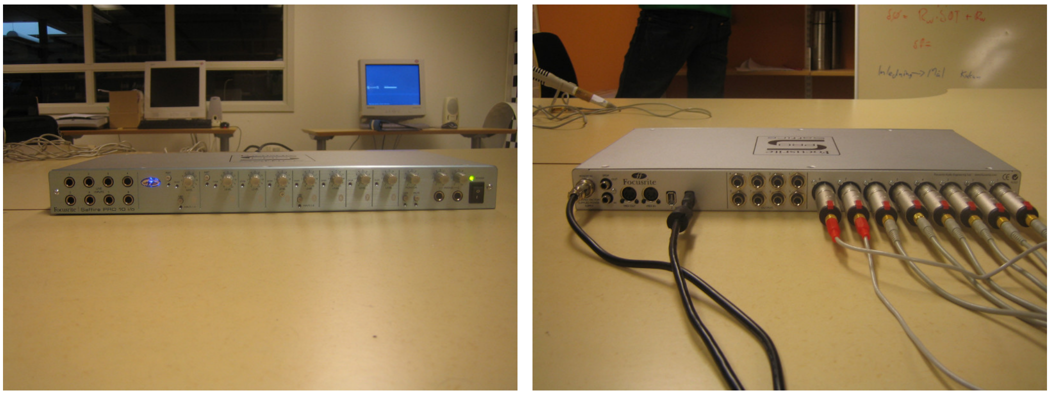
3 Equipment

The equipment used is briefly explained below.

Sound Card

The sound card is a *Saffire PRO 10 I/O*, see Figure 1, which allows up to eight channel recording and 10 channel playback simultaneously. Please refer to the *User Guide* available at the laboration site for more information about the hardware.

The sound card has a software interface called *SaffireControl PRO* (icon located on the desktop) which basically has the same functionality as the sound card itself, but also enables more settings. *SaffireControl PRO* shows if the sound card is connected correctly. You may have to restart the *SaffireControl PRO* if the sound card cannot be found.



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

Figure 1: *Saffire PRO 10 I/O* sound card.

Additional audio software

The software *Playrec* from http://www.playrec.co.uk is designed for simultaneous multi-channel recording and playback. It has been installed on the computers and makes it possible to use the *Saffire PRO 10 I/O* within MATLAB, see Section 5. There is no need to use *Playrec* directly at the laboration.

Microphones

There are eight microphones used for recording. Make sure that the microphones are switched on before starting the recordings!

Target

The target in this laboration is a LEGO® MINDSTORMS robot, see Figure 2, 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.

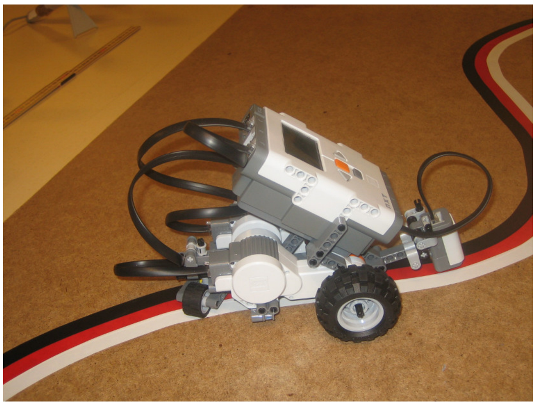


Figure 2: LEGO® MINDSTORMS robot.

Environment

The laboration environment, see Figure 3, 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.

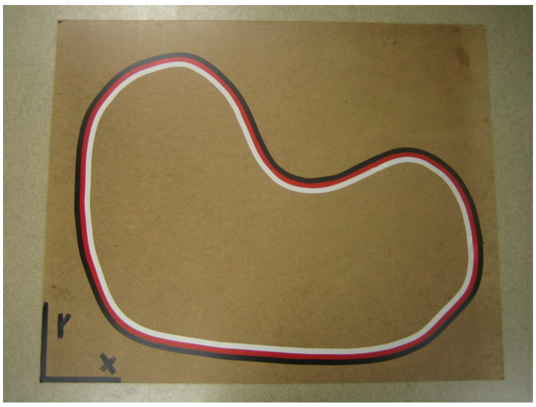


Figure 3: Robot environment.

TDOA quality estimation

On the other side of the wooden board, a circle arc with radius 0.7 m is indicated. Place the microphones and the robot similar to Figure 4 when collecting data for TDOA variance and bias estimates. This orientation of the robot ensures that its speaker is facing the microphones.

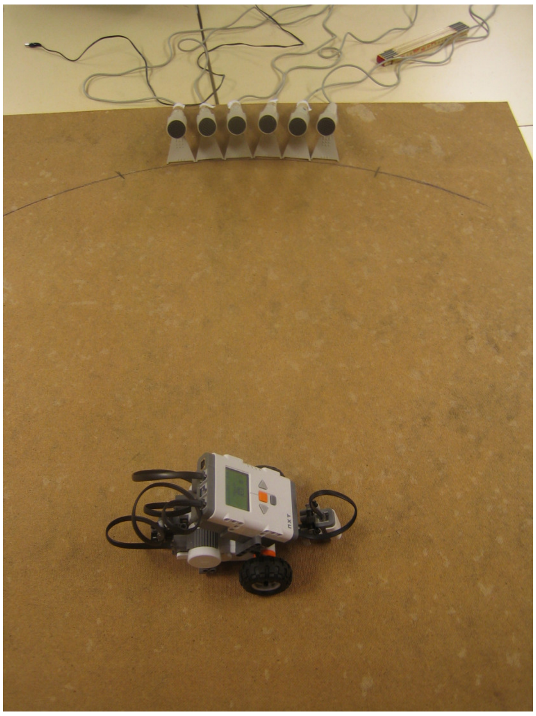


Figure 4: Suggested setup when estimating the quality of the TDOA measurements.

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 5. 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

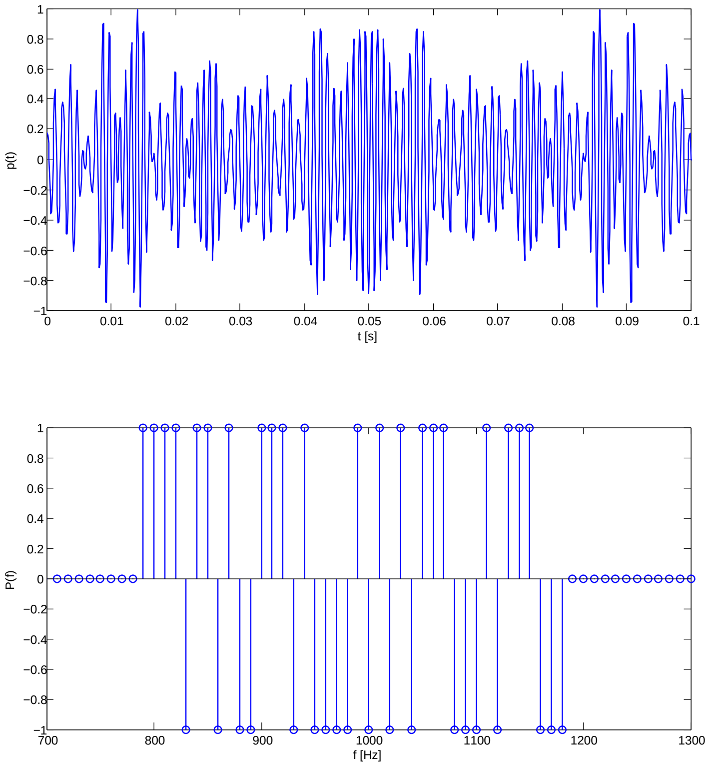


Figure 5: OFDM signal.

robot is emitting a sequence of pulses, several maxima will be found. Even in situations with a 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.
- `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.
- `SFlabCompEstimGroundTruth` — plots the microphone positions and the trajectory estimates overlaid on the robot environment.

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 be called by the user. The MATLAB functions can be downloaded from the course homepage.

6 Robot functions

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.
- `SFdriveSilent` — makes the robot follow the trajectory without playing the sound.

The functions are started like this:

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 function of your choice and press the orange button.

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

The robot runs on a battery. It is a good idea to charge the battery in-between experiments, since the line following capability degrades when the battery runs low.

7 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.

7.1 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.

7.2 Localisation

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). These measurement errors can be visualized in a normalized histogram

```
[N, l] = hist(e,20);
Wb=l(2)-l(1); % Bin width
Ny = length(e); % Nr of samples
bar(l, N/(Ny*Wb));
```

   which can be seen as an empirical probability density function of the measurement noise. Compare the histogram with a `pdfclass` distribution, for instance a normal distribution `pe = ndist(mean(e), var(e))`, where `mean(e)` is the bias and `sqrt(var(e))` is the standard deviation of the measurement noise. This distribution can be visualized using `plot(pe)`.

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 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 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 evaluate how sensitive the result is with respect to the specified microphone locations. 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?

8 Lab Report and submission instructions

The laboration is not examined at site, but using a written report. The report must include the following:

1. A description of how the data was gathered. This includes the microphone positions, and a motivation why the particular positions were chosen.

2. Descriptions of how you solved the tasks in Section 7.2. Include plots that illustrate your results.

3. Well supported conclusions.

4. Code in an appendix.

Make sure a person that did not just perform the lab can follow your presentation. Furthermore, keep in mind that your report will be evaluated using the criteria listed in Section 9.

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 author ID on this lab report. Do NOT write your names or person numbers on the lab report. This report will not be graded by the assistant. Please send this version of the report also to Urkund (the e-mail address can be found on the course webpage).

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 author ID as well as the author ID of the lab report you reviewed. For instructions on how to perform your peer review, please read Section 9.

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 response also needs to be written. In this review response, the comments of the reviewers need to be addressed and updates to the report need to be discussed. Please write your names and personnumbers on this version of the lab report!

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

Possibly submisssion 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.

To enable the peer review process, the submission of the lab reports, the reviews and review responses is done through a conference system. The link to the conference system can be found on the course webpage. In this system, each group needs to register both as an author and as a reviewer (only one author account and one reviewer account per group!). Details on which information needs to be filled in can be found in the conference system when signing up. Please make sure to use the same email address for the author account and for the reviewer account! After signing up, you receive an author ID and a reviewer ID. Please use the *author ID* on the first submission of the lab report and on the peer review. The *reviewer ID* does not need to be used anywhere!

9 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 7.2? 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.

10 Preparations

- 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.

References

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