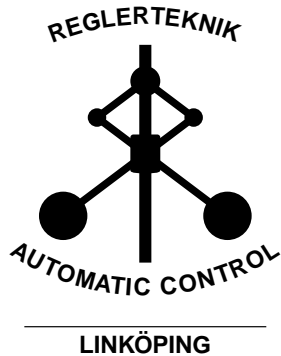
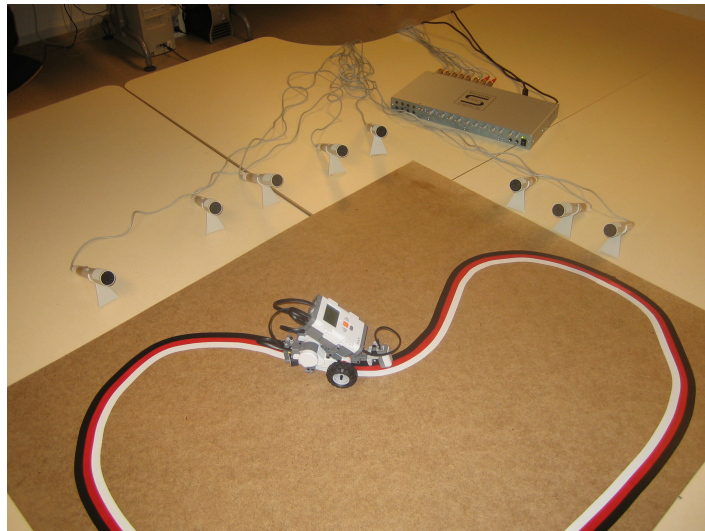


# Sensor Fusion, TSRT14

## Localisation Using a Microphone Network

This version: March 30, 2011



Name: \_\_\_\_\_

P-number: \_\_\_\_\_

Date: \_\_\_\_\_

Passed: \_\_\_\_\_

# 1 Introduction

This laboration is about localisation using a network of sensors, a problem that is common in the sensor fusion world. A real world example of audio based localisation is shooter localisation. 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 localisation of airplanes using radars.

In this laboration you will use a network of microphones to localise 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.

However in this laboration, as in most real world cases, the time of emission is not known. When all sensors are synchronised one method of performing localisation when time of emission is unknown, is using 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<sup>®</sup> 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 the robot by recording audio with eight microphones, and detecting the sound pulses in each recording. Finding the times at which the sound pulses were received at each microphone, TDOA localisation is to be performed.

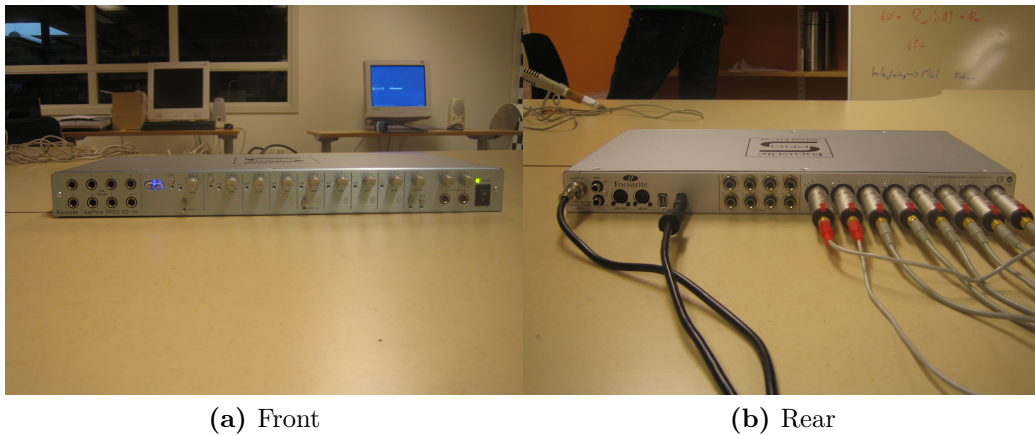
## 3 Equipment

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 10 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.



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

## Additional audio software

The software *Playrec* from <http://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. MATLAB code is provided, see Section 5, and there is no need to use *Playrec* directly at the laboration.

## Microphones

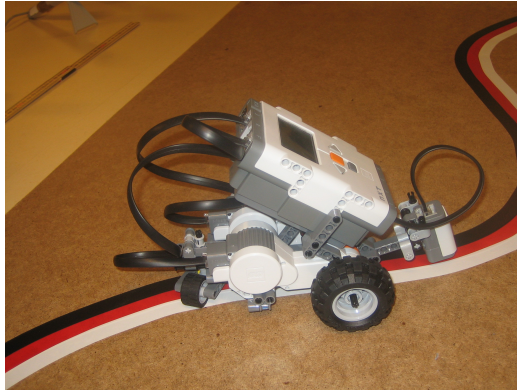
The microphones used for recording are eight *Labtec AM-22* dynamic microphones, see Figure 2. Make sure that the microphones are switched on when recording!



**Figure 2:** *Labtec AM-22* microphone.

## Target

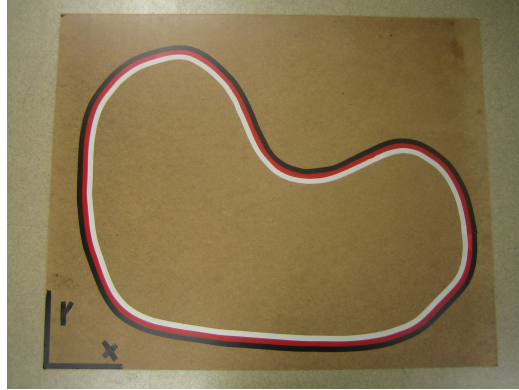
The target in this laboration is a LEGO<sup>®</sup> MINDSTORMS robot, see Figure 3, which uses a light sensor to detect the different colors of the tapes that constitute the trajectory. The robot comes with some programs installed, see Section 6. For example **SFdrive** makes the robot follow the taped lines while playing the sound signal described in Section 4.



**Figure 3:** LEGO<sup>®</sup> MINDSTORMS robot.

## Environment

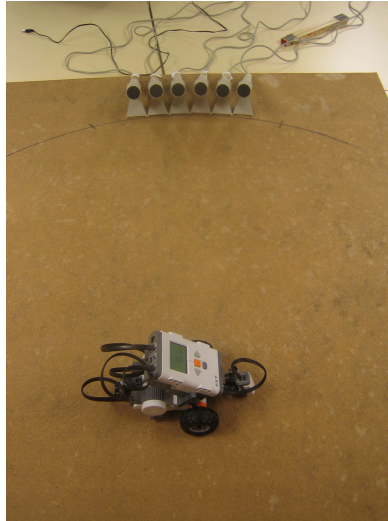
The laboration environment, see Figure 4, is a  $0.991 \times 1.222\text{m}^2$  wooden board with three colors of tape in a loop. The loop shape makes it possible to run the trajectory several times.



**Figure 4:** Robot environment.

### TDOA Quality Estimation

On the other side of the wooden board there is an arc of a circle with a radius of 0.7m from the center spot. Place the microphones and the robot similar to Figure 5 when collecting data for TDOA variance and bias estimates.

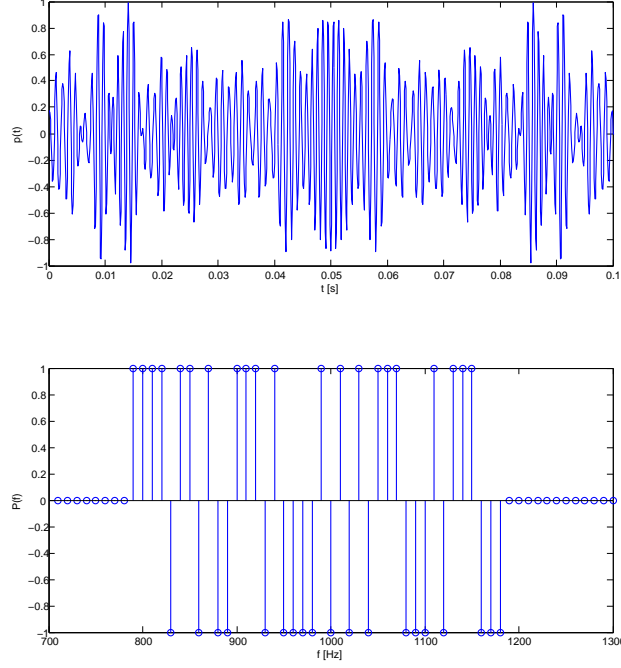


**Figure 5:** Suggested setup when estimating the quality of the TDOA measurements.

## 4 Signal Processing

The robot emits a pulse train, where each pulse is a OFDM signal generated to be insensitive to noise disturbances. The pulse  $p(t)$  is generated from a

finite number of discrete frequencies  $P(f)$  where the frequencies close by is set to zero. An example of such a signal is shown in Figure 6. Since the emitted



**Figure 6:** OFDM signal.

signal is known at the receiver the measured signal from each microphone can be correlated with a copy of the pulse. This is the autocorrelation 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 they will hopefully be much smaller. Since the 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  $t = 0.1s$  made of frequencies between  $f = 800 - 1200$  Hz. The time between pulses is approximately 0.5 seconds.

## 5 Provided code

The following MATLAB functions are provided:

- **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.
- **SFlabCalibrate** - can be used to obtain rough estimates of the precision with which localisation can be performed.
- **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 the interfaces of 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.

## 6 Robot functions

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

- **SFdrive** - makes the robot follow the trajectory while playing the sound.
- **SFpulse** - plays the sound described in Section 4.
- **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 stop a running 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 is worsened when the battery runs low.

## 7 Tasks

The laboration is performed in groups of two students. Data gathering is done in the lab, where all the necessary equipment is located. Time slots for data gathering can be booked via the course assistant. Each one hour time slot is shared by two student groups. The remainder of the laboration is performed elsewhere.

### 7.1 Data collection

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

1. Use `SFpulse` and `SFlabCalibrate`, see Section 3, to obtain estimates of the measurement precision.
2. Use `SFdriveSilent` and `SFlabPlayAndRecord` to estimate the measurement noise.
3. Record data with at least two different microphone configurations. The two different configurations should correspond to a good and a bad situation, and here you should think in advance how these configurations should be designed.

After a data recording it is important to see if the recording was successful, here the commands `SFlabPlayRecData` and `SFlabPlotRecData` are useful.

### 7.2 Localisation

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



1. **Sensor calibration:** Describe how accurate the sensors are in terms of bias and variance of the measurement noise for each microphone. What is the distribution of the noise (if  $\mathbf{e}$  is a vector of measurement errors, you can do `E=empdist(e); plot(E,estimate(ndist,E))` to compare your empirical data to the normal distribution)? The sensor measurement noise can be represented with a `pdfclass` distribution (for instance `pe=ndist(mean(e),var(e))`).
2. **Signal modeling:** Describe the different models of the sensor you will use in your localisation algorithms. The models should be given both as equations and as an m-file.
3. **Experiments:** A brief description of the microphone configurations, and the obtained data. The sensor locations and the target 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 in at least one of the following aspects:
  - (a) The NLS loss function  $V(x)$  on a grid over the table.
  - (b) CRLB for localisation. This can be represented as a map over the table where each grid point gives a bound on  $\text{RMSE} \geq \text{tr}(I^{-1}(x))$ .  
Use this map to motivate your preferred configuration.
5. **localisation:** Compare at least two of the following localisation algorithms:
  - (a) NLS using a 3D grid search over  $x$  and  $r_0$  (corresponding to the unknown pulse emission time).
  - (b) NLS using a gradient or Gauss-Newton search over  $x, r_0$ .
  - (c) SLS using a WLS estimate of  $r_0$  and a 2D grid search.
  - (d) A TDOA approach, where pairwise differences of detection times are used to eliminate  $r_0$ . All 28 pairs can be used, or one sensor can be a reference, and the seven pairwise differences are used only.
6. **Tracking:** First, select two motion models. Then, compare the following nonlinear filters for both models (that is, you will get four estimated trajectories):

- (a) Take the localisation estimates above as the artificial measurements  $y_k = \hat{x}_k + e_k$ ,  $e_k \in N(0, P_k)$  at time  $k$ . Apply the EKF using the two different motion models in turn.
  - (b) Select once sensor as a reference. Make sure that this one gives good measurements. Use a TDOA measurement model with seven TOA differences. Apply the PMF, EKF, UKF or PF.
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. **'Real-time' target tracking:** Your task here is to write your own tracking filter from scratch using the provided template. The template program takes care of waiting for a new measurement to arrive and reading the clock and computing the TOA measurements. Your motion model should allow for a varying sample interval. Otherwise, you are free to design your model and sensor model freely. Motivate your choices using the previous tasks. The template can be switched to work on data from file, or in real-time in the lab.

## 8 Report

The laboration is not examined at site, but through a written report. Check the course homepage to find out when the report is due. The report must include the following:

1. A description of how the data was gathered. This includes the microphone positions, and 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.

The report will be read by the assistant and given one of the following grades; pass, complement needed, or fail. If complement is needed, a new version of the report must be handed in within one week of its return to the student. If the complemented version is not passed 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 Preparations

Read chapter 4 in [1]. Do the case study from exercise two and three.

## References

- [1] F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, Lund, Sweden, first edition, 2010.