# Practical Assignment 1: Acoustic localisation

## Filtering and Identification (sc42025)

To work on the assignment, complete the exercises in this pdf by using the provided MATLAB live script template. Fill in your code and answers in the provided fields and follow further instructions that are written in the live script template. Before opening the template, replace the word 'template' in the file name with both your surnames. If you are stuck on an exercise, you can continue with the next exercise without needing the results of the previous. For questions, we have a question hour on December 9th.

When you are finished with the assignment, please copy all the functions that you wrote into the given templates into the space at the end of the Live Script. Check that the script still runs correctly. In addition, check that the display call of your test functions is not suppressed or commented out (it need to either display 'Correct' or 'Wrong'). Finally, please export the live script as a pdf (in the 'Live Editor tab', click on the arrow under 'Save' and then 'Export to pdf').

The hand-in for the practical assignments are in groups of 2. If you don't have a partner yet, you can start a thread on the Brightspace forum to find one. You don't need to register anywhere who you partnered up with. Just **put both your names and student numbers** on the report and hand it in through **only one of your Brightspace accounts**. We will make sure that the grade will be registered for both of you.

## Introduction

A small robot follows a path on a wooden board, surrounded by 8 microphones, as shown in Figure 1 (left). While moving, the robot emits sound pulses (little beeps) at regular intervals. We measure the time of arrival (TOA) of the pulses at each microphone. Assuming that the locations of the microphones are known and having only the data of the TOA measurements, your tasks is to compute 2D position estimates of the robot along the path. All positions are with respect to the coordinate system as shown in Figure 1 (right). You will use various algorithms that you learned during the course.

For a video showing the data collection, see
`https://www.youtube.com/watch?v=vT4HdnarF74&feature=youtu.be`.

## The given data

You are provided with three .mat-files containing the data you need to implement this assignment. The first .mat-file is called `calibration.mat`, and contains the time of arrival measurements from each microphone for a calibration sequence of measurements that was collected with the robot standing still with an equal distance to all microphones. The second and third .mat-files are called `experiments1.mat` and `experiments2.mat`, and they contains the TOA measurements from each microphone, position measurement, in case you don't manage to complete exercise 2, and the locations of each microphone on the board. The measurements represent the time when a sound pulse is measured by each microphone. Additionally, you are provided with the files `ground_truth1.mat` and `ground_truth2.mat`, which both contain the ground truth trajectories of the robot for each experiment. An overview of the data is given in Table 1.

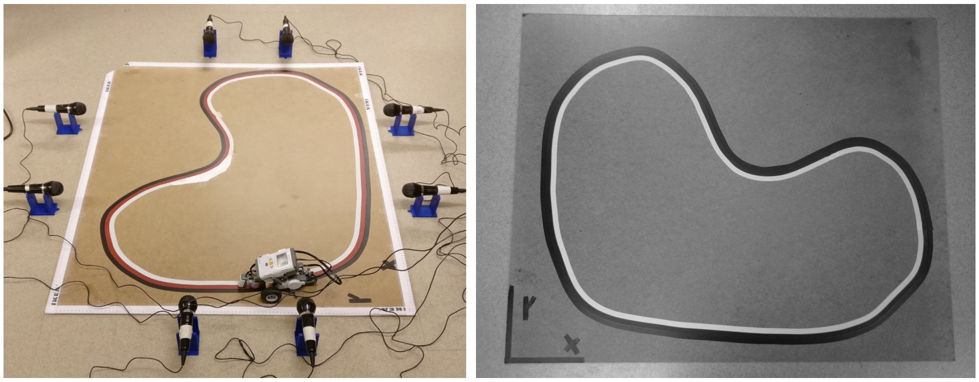

Figure 1: Left: A Lego Mindstorms robot autonomously follows a pre-defined circular path on a wooden board is of size $0.991 \times 1.222$ m$^2$ with a coordinate system defined by the indicated $x$- and $y$-axes. Eight microphones placed around this path can detect sound pulses emitted by the robot. Right: Birds eye view of the trajectory with given coordinate system.

Table 1: Given data

| File name | Variable(s) | Description |
|---|---|---|
| `calibration.mat` | `y_toa` | TOA measurements in seconds from 59 beeps measured with 8 microphones during calibration. |
| `experiment1.mat`/ `experiment2.mat` | `mic_locations` | Microphone locations in metres with respect to the given coordinate system, where the first column corresponds to the position along the $x$-axis and the second column to the position along the $y$-axis. |
| | `y_toa` | TOA measurements in seconds from 137 beeps measured with 8 microphones during the experiment. |
| | `y_pos` | Position measurements for exercise 3 that are only to be used if you cannot complete exercise 2. |
| `ground_truth1.mat`/ `ground_truth2.mat` | `ground_truth` | Ground truth trajectories of the robot for experiment 1 and 2. |

## The measurement model

We measure the TOA in seconds of the sound pulses emitted by the robot with $j = \{1, 2, \dots, 8\}$ microphones, as depicted in Figure 1 (left). The $j$th microphone detects the $k$th sound pulse at a measured TOA $y_{k,j}$. The TOA measurement model for the $j$th microphone at sound pulse $k$ is given by

$$y_{k,j} = \tau_k + \frac{1}{c}\|p_k - m_j\|_2 + b_j + e_{k,j}, \qquad e_{k,j} \sim \mathcal{N}(0, \sigma_j^2), \tag{1}$$

where $\tau_k$ is time of the sound emission and $\frac{1}{c}\|p_k - m_j\|_2$ is the time it takes for sound to travel the distance from the robot to the microphone. The parameter $c = 343\,\mathrm{m/s}$ is the speed of sound in open air, $p_k$ denotes the robot's position on the board when it emitted the $k$th pulse, $m_j$ denotes the position of the $j$th microphone, and $\|\cdot\|_2$ denotes the 2-norm. In addition, $b_j$ denotes measurement bias in seconds of the $j$th microphone, and $e_{j,k}$ denotes a measurement noise for the $j$th microphone. Note that the model (1) is nonlinear.

# Exercise 1: Calibration

In this assignment you will make use of the data set `calibration.mat`. For this data set, the microphones are placed at the same distance, denoted by $d$, from the stationary robot. Because the exact location of both the microphones and the robot are irrelevant for this assignment, this data set only contains the time of arrival (TOA) measurements $y_{k,i}$. In general, it is true that

$$\frac{1}{8}\sum_{i=1}^{8} y_{k,j} \approx \tau_k + \frac{d}{c}. \tag{2}$$

For this calibration, however, we assume that the average TOA at each time step is exactly the correct TOA, so the $\approx$ in (2) becomes an $=$.

a) Calculate the measurement errors of each microphone at each time step. Explain with an equation how you calculate it.

b) From the measurement errors, compute the measurement bias $b_j$ of each microphone. Explain with an equation how you calculate it.

c) From the measurement errors, compute the variance $\sigma_j^2$ of the measurement noise $e_{k,j}$ of each microphone. Explain with an equation how you calculate.

d) Visualise the measurement errors of all the microphones using histograms. Code is provided in the sample file to help you with the visualisation. Interpret your graph and reflect on why a calibration is necessary.

# Exercise 2: Nonlinear least squares

In this exercise you will estimate the position of the robot at each sound emission $k$ using nonlinear least squares. You will use the information from the measurement model (1). This implies solving at each $k$ a minimisation problem of the form

$$\min_{\theta} \|e\|^2_{R^{-1}} \qquad y_k = f(\theta_k) + e_k \qquad e_k \sim \mathcal{N}(0, R) \tag{3}$$

where $y_k$ is the TOA measurement vector for all microphones, $\theta_k$ is the variable we would like to estimate, $f$ is a nonlinear function of $\theta_k$ and $R$ denotes the covariance matrix of the noise.

a) Using model (1), define the nonlinear function $f(\theta_k)$ in (3) as explicitly depending on the entries of the state vector $\theta_k := [x_k \quad y_k \quad \tau_k]$. In addition, define the noise covariance matrix $R$.

At each iteration of the nonlinear least squares algorithm, the nonlinear least squares problem is linearised at the estimated state vector from the previous iteration $(i)$ using a first order Taylor series expansion like in Eqn. (30) of the lecture notes on Bayesian estimation.

b) Show that it is

$$\left.\frac{\mathrm{d}f(\theta_k)}{\mathrm{d}\theta_k}\right|_{\theta_k=\hat{\theta}_k^{(i)}} = \begin{bmatrix} \frac{\hat{x}_k^{(i)}-m_{x,1}}{c\|\hat{p}_k^{(i)}-m_1\|_2} & \frac{\hat{y}_k^{(i)}-m_{y,1}}{c\|\hat{p}_k^{(i)}-m_1\|_2} & 1 \\ \vdots & \vdots & \vdots \\ \frac{\hat{x}_k^{(i)}-m_{x,8}}{c\|\hat{p}_k^{(i)}-m_8\|_2} & \frac{\hat{y}_k^{(i)}-m_{y,8}}{c\|\hat{p}_k^{(i)}-m_8\|_2} & 1 \end{bmatrix}, \tag{4}$$

$$\text{with } \hat{p}_k^{(i)} = [\hat{x}_k^{(i)} \quad \hat{y}_k^{(i)}]^\top \text{ and } m_j = [m_{x,j} \;\; m_{y,j}]^\top. \tag{5}$$

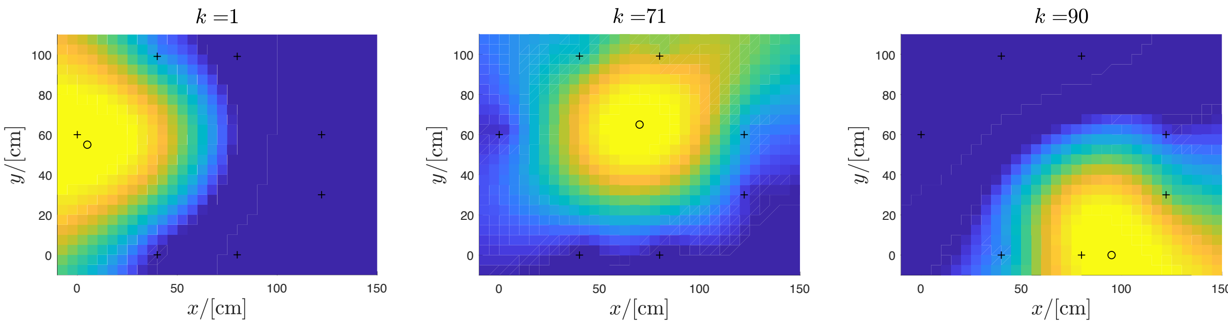


Figure 2: The value of the colour is proportional to the likelihood $\log(p(y_k \mid p_k))$ of the measurements at some select steps $k$ in the possible robot positions $p_k$ on the board. The closer the colour is to yellow, the higher is the likelihood, and the closer the colour is to blue, the lower is the likelihood. The circle indicates the position where the likelihood is maximized. The crosses indicate the microphone positions.

c) Implement the nonlinear least squares algorithm that computes the state estimate $\hat{\theta}_k$ at each time step.

d) Use the data in `experiment1.mat` to find the state estimate $\hat{\theta}_k$ for $k = 1, \ldots, 137$ with the nonlinear least squares algorithm. For this, calibrate the measurements by subtracting the estimated biases from 1b) from the TOA measurements. In case you were not able to compute the bias of the microphones, use a value of 0 for each microphone. In case you were not able to compute the variances of the microphone measurement noise, use $\sigma_j^2 = 10^{-10}$, for $j = 1, \ldots, 8$. Initialise the state as given in the live script.

e) Visualise the mean of the position estimates and their uncertainties, by un-commenting the lines of code for visualisation in the provided template. The function will plot the position estimates as dots, and their uncertainty will be indicated by an ellipsoid surrounding the estimate. Reflect on the quality of the estimate, their uncertainties and its relation to the location on the board. Elaborate on the information you get from the mean and covariance. Also refer to Figure 2, where the cost function is visualised across the board for certain time steps.

## Exercise 3: Kalman filtering using "position measurements"

In this exercise you will estimate the robot positions with a KF, given the position estimates from exercise 2 as "measurements", here called pseudo-measurements, and denoted by $y_k$. The uncertainty of each pseudo-measurement is then the "measurement uncertainty" in the Kalman filter. The measurement model for this case is therefore given as

$$y_k = p_k + v_k, \qquad v_k \sim \mathcal{N}(0, R_k), \tag{6}$$

with $p_k$ being the state and $v_k$ being a pseudo-measurement noise with noise covariance matrix $R_k$. Further, we assume a dynamic model where the current position is almost equal to the previous position, such that the position $p_k$ is given by the random walk model

$$p_{k+1} = p_k + w_k, \qquad w_k \sim \mathcal{N}(0, Q), \tag{7}$$

where $w_k$ is a process noise with covariance matrix $Q \in \mathbb{R}^{2\times 2}$.

a) Implement the Kalman filter to compute the position estimates $\hat{p}_k$ for $k = 1, \ldots, 137$.

b) Use your position estimates from exercise 2 as pseudo-measurements to compute position estimates with the KF. In case you did not manage to finish exercise 2, you can use the given pseudo-measurements y_pos. For initialization, use the given initial position and covariance matrix. Tune the filter by choosing values for $Q$ and $R_k$ and comment on your choice. We recommend that you select the matrix $R_k$ based on your findings from the previous assignment so that it reflects how uncertain each pseudo-measurement is.

c) Plot your estimates and their uncertainties in the trajectory using the provided function `plotresults`. Comment on how your result compares to the previous estimates, and why it is different/similar.

## Exercise 4: Extended Kalman filtering using time of arrival measurements

Computing position estimates from the TOA measurements and use them as pseudo-measurements is a pre-processing step in order to be able to use a Kalman filter that has a linear measurement model. In this assignment, you will not use the pseudo-measurements of the position, but the TOA measurements directly. Since here the measurement model is nonlinear, you will need an extended Kalman filter to estimate the state $\theta_k = [x_k \; y_k \; \tau_k]$.

The measurement model for the extended Kalman filter is given in (3) with $y_k$ being the TOA measurements corrected with the bias. The dynamic model for the position $x_k, y_k$ is given by (7). Augmenting the dynamic model with the third component of the state $\tau_k$, is is

$$\begin{bmatrix} p_{k+1} \\ \tau_{k+1} \end{bmatrix} = \begin{bmatrix} p_k \\ \tau_k + \Delta\tau \end{bmatrix} + w_k, \qquad w_k \sim \mathcal{N}(0, Q), \tag{8}$$

where $\Delta\tau$ is the duration between two pulses that is assumed to be constant and the third entry on the diagonal $Q \in \mathbb{R}^{3\times3}$ is the noise variance that is included in the model to account for small irregularities in the emitting of the robot's sound.

a) Implement the EKF to compute the state estimates $\hat{\theta}_k$.

b) Use the data in `experiment1.mat`. As in exercise 2, calibrate the measurements by subtracting the estimated biases from 1b) from the TOA measurements. In case you were not able to compute the bias of the microphones, use a value of 0 for each microphone. In case you were not able to compute the variances of the microphone measurement noise, use $\sigma_j^2 = 10^{-10}$, for $j = 1, ..., 8$. Compute the state estimates $\hat{\theta}_k$, for $k = 1, ..., 137$ with the EKF. Tune values for the matrices $R$ and $Q$ and comment on your choice. We recommend that you use information from the calibration in choosing these matrices. How do you choose them compared to the previous assignment?

c) Plot your position estimates and their uncertainty-ellipsoids in the trajectory using the provided function `plotresults`. Comment on how your result compares to the previous estimates, and why it is different/similar.

## Final note

If you want, you can run all the assignments you have implemented also on the data set `experiment2.mat`, this should give similar results to what you get from `experiments1.mat`. The benefit you get from doing this is that it can give you an indication if your implementation is correct (you should expect similar results with both data sets), but there is no need to report your results from this data set.

## Reminder

When you are finished with the assignment, please copy all the functions that you wrote into the given templates into the space at the end of the Live Script. Check that the script still runs correctly. In addition, check that the display call of your test functions is not suppressed or commented out (it need to either display 'Correct' or 'Wrong').
Finally, please export the live script as a pdf (in the 'Live Editor tab', click on the arrow under 'Save' and then 'Export to pdf').