Filtering and Identification Practical Assignment 1

Acoustic localisation

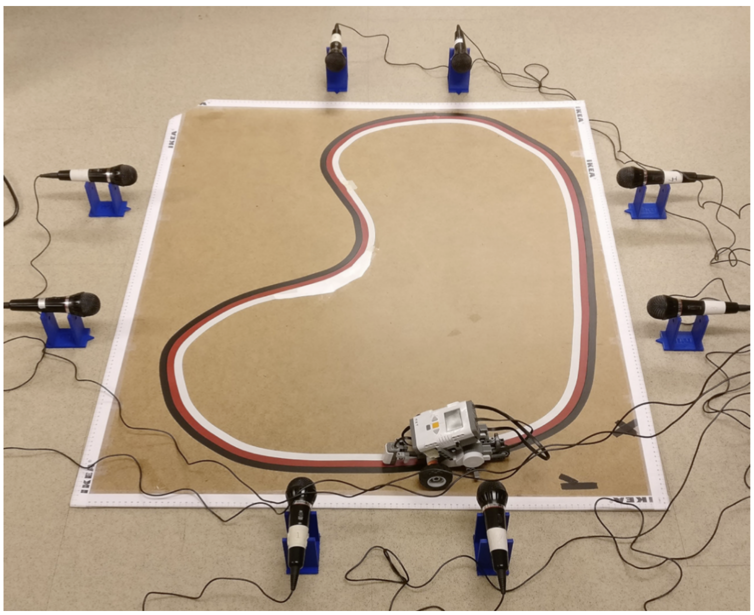

Figure 1: A Lego Mindstorms robot autonomously follows a pre-defined circular path. Eight microphones placed around this path can detect sound pulses emitted by the robot.

Introduction

In this lab you will localise a mobile robot on a wooden board using a network of microphones as shown in Figure 1. The wooden board is of size 0.991×1.222 m^2 with a coordinate system defined by the x- and y-axes indicated in Figure 2. All position vectors used in this assignment are in 2 dimensions, and defined with respect to this coordinate system. T The robot follows the path indicated in red in Figure 1 and makes two laps. Your task is to localise this robot along the path using various algorithms that you learned during the course. For a video of the data collection, see `https://www.youtube.com/watch?v=vT4HdnarF74&feature=youtu.be`.

Through the assignment, we will use boldface to denote vectors, and non-capital italics to denote scalars.

The lego robot emits sound pulses (little beeps) at regular intervals. The time of the pulse emission is an unknown variable, but you can assume that it is an outcome of the following stochastic process

$$\tau_{k+1} = \tau_k + \Delta\tau + e_{\tau,k}, \qquad e_{\tau,k} \sim \mathcal{N}(0, \sigma_\tau^2), \tag{1}$$

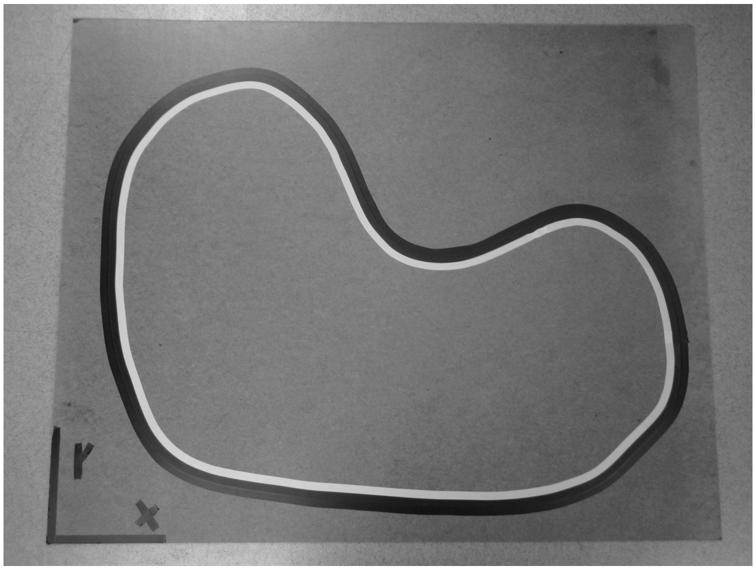

Figure 2: Birds eye view of the trajectory with axes directions indicated.

where τ_k is the k'th sound pulse, $\Delta\tau$ is the constant interval between each sound emission, and $e_{\tau,k}$ is a noise that is included in the model to express the fact that the robot system is not able to give out a sound pulse at perfectly regular intervals, and sometimes will beep a bit earlier or later than the scheduled time.

We will measure the time of arrival of this sound pulse using 7 of the 8 depicted microphones. We are not using all 8 microphones in Figure 1, because one of the microphones gave faulty readings, so only data from 7 microphones are included in the given data set. We will give each microphone a number $m = \{1, 2, \ldots, 7\}$. The time each pulse arrives at the robot is given by the time of emission τ_k plus the time it takes for sound to travel the distance from the robot to the microphone $\frac{1}{c}\|\mathbf{p}_{\mathrm{r},k} - \mathbf{p}_m\|_2$, where the parameter $c = 343\,\mathrm{m/s}$ is the speed of sound in open air, $\mathbf{p}_{\mathrm{r},k}$ denotes the robot's position on the board when it emitted the (k)'th pulse, and $\mathbf{p}_m$ denotes the position of microphone m. The microphones will detect the (k)'th sound pulse at a measured time of arrival $y_{m,k}$. The measurement model for the time of arrival at each microphone is given by

$$y_{m,k} = \tau_k + \frac{1}{c}\|\mathbf{p}_{\mathrm{r},k} - \mathbf{p}_m\|_2 + b_m + \epsilon_{m,k}, \qquad \epsilon_{m,k} \sim \mathcal{N}(0, \sigma_m^2), \tag{2}$$

where b_m denotes each microphone's measurement bias in seconds, $\|\cdot\|_2$ denotes the 2-norm, and $\epsilon_{m,k}$ denotes a measurement noise for the measurement at time step k. The subscript r,k denotes the position of the robot (r) at time step k. Note that the model (2) is nonlinear.

You are provided with two .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 .mat-file is called `experiments.mat`, and it contains the time of arrival measurements from each microphone and the locations of each microphone on the board. An overview of the data in each of the .mat-files is given in Tables 1 and 2. The measurements represent the time when a sound pulse is measured by each microphone. You can find these files in the attached folder. In addition, you get a template to fill in your scripts and functions, as well as a function to plot your results.

In this lab, we expect you to complete the following assignments and discuss your solutions in a report, which has your code in an appendix. Note that the report should motivate all your design choices, e.g. choice of model and of tuning parameters. Furthermore, you are not allowed to use any Matlab toolboxes, i.e. we

Table 1: Variables in the data set `calibration.mat`.

Variable	Description
`y`	Time of arrival measurements in seconds for N measurements from 7 microphones during calibration.

Table 2: Variables in the data set `experiment.mat`.

Variable	Description
`y`	Time of arrival measurements in seconds for N measurements from 7 microphones during the experiment.
`mic_locations`	Microphone locations in metres, where the first column corresponds to the position along the x-axis and the second column to the position along the y-axis.

expect you to implement nonlinear least squares, Kalman filtering and extended Kalman filtering yourself.

Assignment 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 $d = \|\mathbf{p}_\mathrm{r} - \mathbf{p}_m\|_2 \, \forall m$, 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`. In general, it is true that

$$\frac{1}{7}\sum_{m=1}^{7} y_{m,k} \approx \tau_k + \frac{d}{c}. \tag{3}$$

For this calibration, however, we assume that the average TOA at each time step is exactly the correct TOA, so the $\approx$ in (3) becomes an $=$.

a) Calculate the measurement errors of each microphone at each time step.

b) From the measurement errors, estimate the measurement bias b_m of each microphone.

c) From the measurement errors, estimate the variance σ_m^2 of the measurement noise ϵ_m of each microphone.

d) Visualise the measurement errors of one of the microphones using a histogram. The following code snippet can be useful for making a histogram, which can be seen as an empirical probability density function of the measurement noise.

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

Assignment 2: Nonlinear least squares

In this assignment you will make use of the data set experiment.mat and estimate the position of the robot for each pulse k using nonlinear least squares. You will use the information from the measurement model (2). This implies solving each time step a minimisation problem of the form

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\epsilon}\|_{\Sigma} \qquad \mathbf{y} = f(\boldsymbol{\theta}) + \boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Sigma) \tag{4}$$

where $\boldsymbol{\theta}$ is the variable we would like to estimate, f is a nonlinear function of $\boldsymbol{\theta}$ and Σ denotes the covariance matrix of the noise.

2.1 Implement the NLS algorithm

To implement the NLS algorithm, follow the following steps.

a) Subtract the estimated biases from the TOA measurements to achieve calibrated measurements. Then, for each time step, rewrite the measurement model for all microphones so that minimising the measurement noise corresponds to minimising a weighted least squares problem on the form described in (4). Define the state vector $\boldsymbol{\theta}$, the nonlinear function f, and the noise covariance Σ. Use the results from Assignment 1. Implement f, using the template f.m.

b) Show that the gradient of the nonlinear function is

$$\left.\frac{\mathrm{d}f(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \begin{bmatrix} \frac{\hat{x}_{\mathrm{r},k}-x_1}{c\|\hat{\mathbf{p}}_{\mathrm{r},k}-\mathbf{p}_1\|} & \frac{\hat{y}_{\mathrm{r},k}-y_1}{c\|\hat{\mathbf{p}}_{\mathrm{r},k}-\mathbf{p}_1\|} & 1 \\ \vdots & \vdots & \vdots \\ \frac{\hat{x}_{\mathrm{r},k}-x_7}{c\|\hat{\mathbf{p}}_{\mathrm{r},k}-\mathbf{p}_7\|} & \frac{\hat{y}_{\mathrm{r},k}-y_7}{c\|\hat{\mathbf{p}}_{\mathrm{r},k}-\mathbf{p}_7\|} & 1 \end{bmatrix}, \tag{5}$$

$$\text{with } \hat{\mathbf{p}}_{\mathrm{r},k} = [x_{\mathrm{r},k} \quad y_{\mathrm{r},k}]^{\top} \text{ and } \mathbf{p}_m = [x_m \;\; y_m]^{\top}, \;\; m = \{1, \ldots, 7\}. \tag{6}$$

and implement it, using the template Jacobian.m.

c) Complete the template nls, using your functions from a) and b). Compute both the estimated mean and the covariance.

2.2 Estimate positions for each k

Now use the function from Assignment 2.1 to compute the estimate for each pulse k. For this, follow this pseudo code and use a maximum number of iterations of maxiter = 100 and the initial estimate th_hat0 = [10 60 0].

```
for k = 1:length(y)
    [th_hat(k,:),diagP(k,:)] = nls(yk,biases,stds,th_hat0,maxiter,mic_locations);
end
```

2.3 Visualise and interpret

Visualise the mean of the position estimates and their uncertainties, using the function plotresults. The function will plot the position estimates as dots, and their uncertainty will be indicated by an ellipsoid surrounding the estimate. Reflect on your solution. Elaborate on the information you get from the mean and covariance. If you find it helpful, you can refer to Figure 3, where the cost function is visualised across the board for certain time steps.

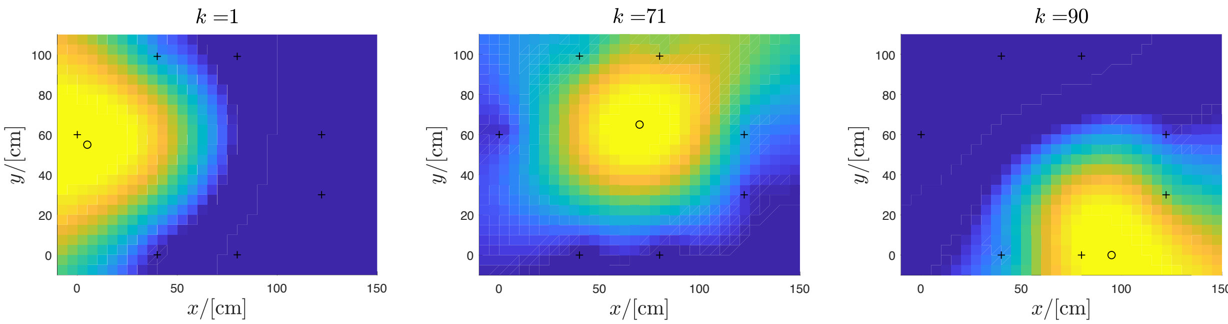

Figure 3: The value of the colour is proportional to the likelihood $\log(p(\mathbf{y}_{1:7,k} \mid \mathbf{p}_\mathrm{r}))$ of the measurements at some select steps k in the possible robot positions $\mathbf{p}_\mathrm{r}$ on the board. The closer the colour is to yellow, the higher is the likelihood, 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.

Assignment 3: Kalman filtering using "position measurements"

It is possible to use the position estimates from Assignment 2 as "measurements" $\tilde{\mathbf{p}}_k$ in a Kalman filter. Note that the uncertainty of that estimate is now the "measurement uncertainty" in the Kalman filter. The measurement model for this case is therefore given as

$$\tilde{\mathbf{p}}_k = \mathbf{p}_{\mathrm{r},k} + \mathbf{v}_{\mathrm{r},k}, \qquad \mathbf{v}_{\mathrm{r},k} \sim \mathcal{N}(\mathbf{0}, R_k), \tag{7}$$

with $\tilde{\mathbf{p}}_k$ being a pseudo-measurement, and $\mathbf{v}_{\mathrm{r},k}$ being a pseudo-measurement noise.

Implement the Kalman filter and compute the position estimates. As a dynamic model you can assume that the current position is almost equal to the previous position. More precisely, assume that the position $\mathbf{p}_{\mathrm{r},k}$ is given by the random walk model

$$\mathbf{p}_{\mathrm{r},k+1} = \mathbf{p}_{\mathrm{r},k} + \mathbf{w}_{\mathrm{r},k}, \qquad \mathbf{w}_{\mathrm{r},k} \sim \mathcal{N}(\mathbf{0}, Q), \tag{8}$$

where $\mathbf{w}_{\mathrm{r},k}$ is a process noise.

Notice that you will have to choose values yourself for Q and R_k when you are implementing this filter - write down what values you choose for Q and R_k and why you chose them this way. 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.

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.

Assignment 4: Extended Kalman filtering using time of arrival measurements

Instead of pre-processing the measurements, for this assignment you have to use them directly in a filter. Since here the measurement model is nonlinear, you will need an extended Kalman filter to estimate the state $[(\mathbf{p}_{\mathrm{r},k})^\top, \tau_k]$.

The measurement model for the extended Kalman filter is given by the TOA measurement model in equation (2). The dynamic model in this assignment is given by (8) and (1).

You will again have to choose your own values for the matrices R and Q. We recommend that you use information from the calibration in choosing these matrices. How do you choose them compared to the previous assignment?

Plot your estimates and their uncertainty-ellipsoids in the trajectory. Comment on how your result compares to the previous estimates, and why it is different/similar.