3 Optimization-based estimation

The aim of this exercise is to get acquainted with the procedure for optimization-based parameter estimation and optimal sensor placement. To this end, self-localization problems of a mobile robot in different two-dimensional enclosures will be studied.

This script is not intended to be self-contained. It is recommended to study at least chapter 3 of the corresponding lecture notes for the VU Optimization-Based Control Methods [3.1].

If you have any questions or suggestions regarding the exercise please contact

- Vojtech Mlynar <mlynar@acin.tuwien.ac.at> or

3.1 Robot self localization in a convex enclosure

Consider a mobile (differential drive) robot in a quadratic enclosure as shown in Figure 3.1. To perform a meaningful task, the robot needs to be aware of its position $\mathbf{p} = \begin{bmatrix} x_\mathrm{R} & y_\mathrm{R} \end{bmatrix}^\mathrm{T}$ with respect to the global coordinate frame. To this end, N beacons are placed at the positions $\mathbf{b}_n = \begin{bmatrix} x_n & y_n \end{bmatrix}^\mathrm{T}$, $n = 1, \dots, N$, inside the enclosure. These beacon positions are known by the robot. Each beacon sends out a signal, which allows the robot to determine its distance

$$\rho_n = \|\mathbf{p} - \mathbf{b}_n\|_2 + v_n = \sqrt{(x_\mathrm{R} - x_n)^2 + (y_\mathrm{R} - y_n)^2} + v_n, \tag{3.1}$$

$n = 1, \dots, N$, to the respective beacon. The uncertainty in the distance determination is modeled by the measurement noise v_n with the variance q_n. With N beacons available, it is customary to stack (3.1) to obtain the nonlinear functional relation

$$\boldsymbol{\rho} = \mathbf{f}(\mathbf{p}, \mathbf{b}_1, \dots, \mathbf{b}_N) + \mathbf{v} \tag{3.2}$$

with

$$\mathbf{f}(\mathbf{p}, \mathbf{b}_1, \dots, \mathbf{b}_N) = \begin{bmatrix} \sqrt{(x_\mathrm{R} - x_1)^2 + (y_\mathrm{R} - y_1)^2} \\ \sqrt{(x_\mathrm{R} - x_2)^2 + (y_\mathrm{R} - y_2)^2} \\ \dots \\ \sqrt{(x_\mathrm{R} - x_N)^2 + (y_\mathrm{R} - y_N)^2} \end{bmatrix} \tag{3.3}$$

$\boldsymbol{\rho} = \begin{bmatrix} \rho_1, \rho_2, \dots, \rho_N \end{bmatrix}^\mathrm{T}$ and $\mathbf{v} = \begin{bmatrix} v_1, v_2, \dots, v_N \end{bmatrix}^\mathrm{T}$. Subsequently, the aim is to use (3.2) to calculate an estimate $\hat{\mathbf{p}}$ of the actual robot position $\mathbf{p}$.

> *Exercise* 3.1 (Prepare at home). Design and test an optimization-based estimator for the robot position $\mathbf{p}$. Proceed as follows:
>
> 1. Create a function `getDistances` in MATLAB which models $\mathbf{f}$ in (3.2), i. e., which

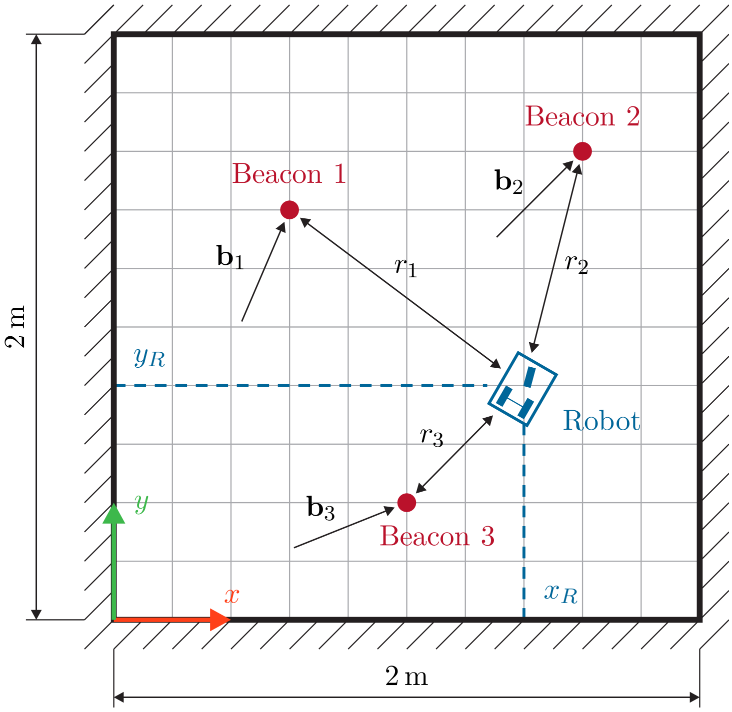

Figure 3.1: A mobile robot in a quadratic enclosure with three beacons for localization.

calculates the noise free distance measurements ρ_n for a known robot location $\mathbf{p}$ and known beacon locations $\mathbf{b}_n$, $n = 1, \dots, N$.

2. The robot is following the trajectory

$$x_\mathrm{R}(t) = (1\mathrm{m}) - (0.6\mathrm{m})\cos(2\pi t) \tag{3.4a}$$
$$y_\mathrm{R}(t) = (1\mathrm{m}) - (0.6\mathrm{m})\sin(2\pi t), \tag{3.4b}$$

$t \in [0, 1]$. Use the function `getDistances` to set up the optimization problem

$$\hat{\mathbf{p}} = \arg\min_{\tilde{\mathbf{p}}} \quad \|\boldsymbol{\rho} - \mathbf{f}(\tilde{\mathbf{p}}, \mathbf{b}_1, \dots, \mathbf{b}_N)\|^2_{\mathbf{Q}^{-1}} \tag{3.5a}$$
$$\text{s.t.} \quad 0 \le \tilde{x}_\mathrm{R} \le 2\mathrm{m} \tag{3.5b}$$
$$0 \le \tilde{y}_\mathrm{R} \le 2\mathrm{m} \tag{3.5c}$$

with $\mathbf{Q}$ as noise covariance matrix, for calculating an estimate $\hat{\mathbf{p}}$ of the robot position $\mathbf{p}$ from noisy measurement data in $\boldsymbol{\rho}$. Use the MATLAB routine fmincon to solve (3.5).

3. Consider the following three sets of beacon locations

$$\mathcal{B}_1 = \left\{ \mathbf{b}_1 = \begin{bmatrix} 0.5\text{m} \\ 0.5\text{m} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 1\text{m} \\ 1.5\text{m} \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 1.5\text{m} \\ 0.5\text{m} \end{bmatrix} \right\} \tag{3.6a}$$

$$\mathcal{B}_2 = \left\{ \mathbf{b}_1 = \begin{bmatrix} 0.5\text{m} \\ 0.5\text{m} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 0.5\text{m} \\ 1\text{m} \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 0.5\text{m} \\ 1.5\text{m} \end{bmatrix} \right\} \tag{3.6b}$$

$$\mathcal{B}_3 = \left\{ \mathbf{b}_1 = \begin{bmatrix} 0.5\text{m} \\ 0.5\text{m} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 0.5\text{m} \\ 1.5\text{m} \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 1.5\text{m} \\ 0.5\text{m} \end{bmatrix}, \mathbf{b}_4 = \begin{bmatrix} 1.5\text{m} \\ 1.5\text{m} \end{bmatrix} \right\} \tag{3.6c}$$

and assume equal noise covariances for q_n for each beacon. Compare the estimated and the true trajectory of the robot for the given sets of beacon locations and the different noise variances $q_n = 1 \cdot 10^{-6}\,\text{m}^2$, $q_n = 5 \cdot 10^{-4}\,\text{m}^2$, and $q_n = 1 \cdot 10^{-3}\,\text{m}^2$. To this end, discretize the robot trajectory with 100 points. Use the estimate of the previous point in the trajectory as initial guess in the optimization for the current point.

In particular for a low number of beacons, the estimation accuracy of the localization algorithm heavily depends on the specific beacon placement. To improve the estimation accuracy of the localization algorithm, the number and positions of the beacons should be optimized.

Exercise 3.2 (Prepare at home). Analyze and optimize the properties of the estimation strategy developed in Exercise 3.1 based on the sensitivity

$$\mathbf{S}(\mathbf{p}, \mathbf{b}_1, \dots, \mathbf{b}_N) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right|_{\mathbf{p}, \mathbf{b}_1, \dots, \mathbf{b}_N}. \tag{3.7}$$

Proceed as follows:

1. Extend the function `getDistances` created in task 1 of Exercise 3.1 to also calculate and return the sensitivity matrix $\mathbf{S}$ according to (3.7) for a known robot location $\mathbf{p}$ and known beacon locations $\mathbf{b}_n$, $n = 1, \dots, N$.

2. Visualize the A- and D-optimality criteria related to $\mathbf{S}$ for all possible robot positions $0 \leq x_\text{R} \leq 2\text{m}$, $0 \leq y_\text{R} \leq 2\text{m}$ and the beacon sets in (3.6). Use a grid spacing of 4cm and the MATLAB routine imagesc or surf.

3. Visualize the (approximate) spatial direction of the largest variance of the position estimator developed in Exercise 3.1 for all possible robot positions $0 \leq x_\text{R} \leq 2\text{m}$, $0 \leq y_\text{R} \leq 2\text{m}$ and the beacon sets in (3.6). Use the MATLAB routines eig and quiver.

4. Calculate the optimal positions $\mathbf{b}_n^*$ for $N = 3$ and $N = 4$ beacons by solving

the optimization problem

$$\mathbf{b}_1^*, \ldots, \mathbf{b}_N^* = \arg \min_{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N} \int_0^{2\mathrm{m}} \int_0^{2\mathrm{m}} r(\mathbf{R}(\mathbf{p}))\, \mathrm{d}x_{\mathrm{R}}\, \mathrm{d}y_{\mathrm{R}} \tag{3.8a}$$

$$\text{s.t.} \quad 0 \le \tilde{x}_n \le 2\mathrm{m}\ , \quad \forall n = 1, \ldots, N \tag{3.8b}$$

$$0 \le \tilde{y}_n \le 2\mathrm{m}\ , \forall \quad n = 1, \ldots, N \tag{3.8c}$$

with r according to the D-optimality measure and

$$\mathbf{R}(\mathbf{p}) = \mathbf{S}^{\mathrm{T}}(\mathbf{p}, \tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N)\mathbf{Q}^{-1}\mathbf{S}(\mathbf{p}_{ij}, \tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N) \tag{3.9}$$

or

$$\mathbf{R}(\mathbf{p}) = \tilde{\mathbf{S}}^{\mathrm{T}}(\mathbf{p}, \tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N)\tilde{\mathbf{S}}(\mathbf{p}_{ij}, \tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N), \tag{3.10}$$

where $\mathbf{Q}$ denotes again the noise covariance matrix and $\tilde{\mathbf{S}}$ is the column normalized version of $\mathbf{S}$, see [3.1]. Consider the beacon sets in (3.6) as initial conditions in the optimization. Find a suitable discretization of (3.8a) and use the MATLAB routine fmincon to solve (3.8).

> **Remark:** The number of sampling points in the discretization of (3.8a) heavily influences the computation time of the optimizer. Keep the number of points reasonably low to facilitate a faster computation time. Similarly, choose adequate tolerances for fmincon.

5. Compare the estimated and the true trajectory of the robot for the obtained optimal beacon locations and the different noise variances $q_n = 1 \cdot 10^{-6}\,\mathrm{m}^2$, $q_n = 5 \cdot 10^{-4}\,\mathrm{m}^2$, and $q_n = 1 \cdot 10^{-3}\,\mathrm{m}^2$.

3.2 Robot self localization in a non-convex enclosure

For a convex shaped enclosure as in Figure 3.1, it is always guaranteed that all beacons are visible to the robot. For a non-convex enclosure as in Figure 3.2, the set of possible robot locations $\mathcal{P}$ is non-convex and visibility of all beacons is generally not guaranteed. Essentially, this means that only those distance measurements ρ_n can be included in (3.2) where the respective beacon is actually visible. Apart from the number of beacons which are necessary for self localization of the robot, the non-convex shape of the enclosure also influences the optimal positioning of beacons. Throughout the subsequent exercises, these difficulties will be treated in more detail.

Exercise 3.3 (Exercise during the lab). Design and test an optimization-based estimator for the robot position $\mathbf{p}$ in the non-convex enclosure shown in Figure 3.2. Proceed as follows:

1. In addition to the function `getDistances` from Exercise 3.1 and 3.2 create a function `getObstruction` in MATLAB to model the visibility of each beacon from

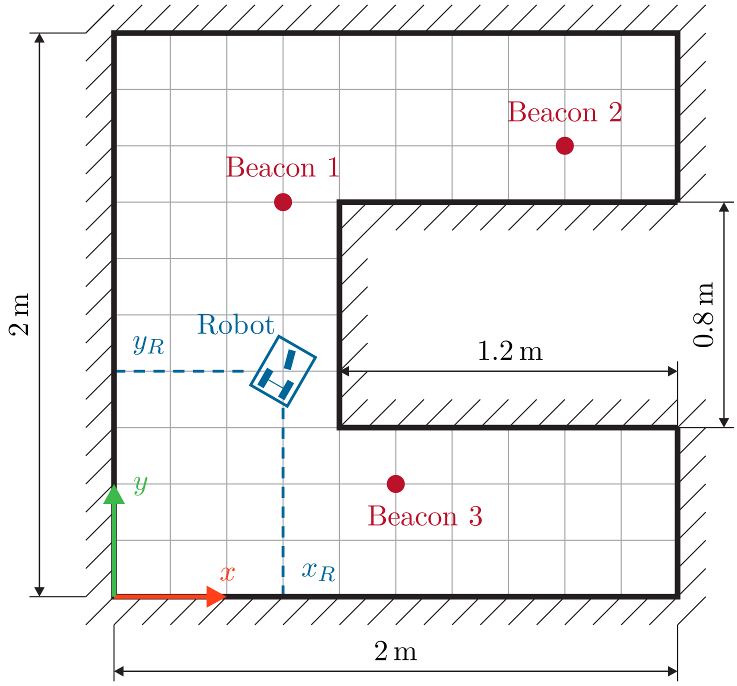

Figure 3.2: A mobile robot in a non-convex enclosure with possible hidden beacons.

the robots position $\mathbf{p} \in \mathcal{P}$ based on the known beacon locations $\mathbf{b}_n \in \mathcal{P}$, $n = 1, \dots, N$. To determine the visibility of each beacon within `getObstruction`, calculate the line of sight between the robot and each beacon and accumulate the length ℓ_n of the line of sight which is outside the enclosure. This integral values (sums) ℓ_n should constitute the return values of `getObstruction`.

2. The individual beacons can be considered visible, if the respective return value of `getObstruction` is zero. Based on this information, delete all rows in (3.2) which do not correspond to visible beacons.

3. The robot is performing the trajectory

$$x_{\mathrm{R}}(t) = (1.6\mathrm{m}) - (1.2\mathrm{m}) \sqrt[5]{\sin^2(\pi t)}\mathrm{sgn}(\sin(\pi t)) \tag{3.11a}$$

$$y_{\mathrm{R}}(t) = (1\mathrm{m}) - (0.7\mathrm{m}) \sqrt[5]{\cos^2(\pi t)}\mathrm{sgn}(\cos(\pi t)), \tag{3.11b}$$

$t \in [0, 1]$. Use the functions `getDistances` and `getObstruction` to set up the optimization problem

$$\hat{\mathbf{p}} = \arg\min_{\tilde{\mathbf{p}}} \quad \|\boldsymbol{\rho}_{\mathrm{red}} - \mathbf{f}_{\mathrm{red}}(\tilde{\mathbf{p}}, \mathbf{b}_1, \dots, \mathbf{b}_N)\|^2_{\mathbf{Q}^{-1}_{\mathrm{red}}} \tag{3.12a}$$

$$\text{s.t.} \quad \tilde{\mathbf{p}} \in \mathcal{P} \tag{3.12b}$$

for calculating an estimate $\hat{\mathbf{p}}$ of the robots position $\mathbf{p}$. Here, $\boldsymbol{\rho}_{\text{red}}$ and $\mathbf{f}_{\text{red}}$ indicate the rows of $\boldsymbol{\rho}$ and $\mathbf{f}$ in (3.2) which corresponding to visible beacons and $\mathbf{Q}_{\text{red}}$ is the respective covariance matrix. Use the MATLAB routine fmincon to solve (3.12).

4. Consider the following three sets of beacon locations:

$$\mathcal{B}_1 = \left\{ \mathbf{b}_1 = \begin{bmatrix} 0.2\text{m} \\ 0.2\text{m} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 0.6\text{m} \\ 0.2\text{m} \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 0.2\text{m} \\ 0.6\text{m} \end{bmatrix}, \right.$$
$$\left. \mathbf{b}_4 = \begin{bmatrix} 0.2\text{m} \\ 1.4\text{m} \end{bmatrix}, \mathbf{b}_5 = \begin{bmatrix} 0.2\text{m} \\ 1.8\text{m} \end{bmatrix}, \mathbf{b}_6 = \begin{bmatrix} 0.6\text{m} \\ 1.8\text{m} \end{bmatrix} \right\} \tag{3.13a}$$
$$\mathcal{B}_2 = \left\{ \mathbf{b}_1 = \begin{bmatrix} 0.2\text{m} \\ 0.2\text{m} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 1.8\text{m} \\ 0.2\text{m} \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 0.2\text{m} \\ 0.6\text{m} \end{bmatrix}, \right.$$
$$\left. \mathbf{b}_4 = \begin{bmatrix} 0.2\text{m} \\ 1.4\text{m} \end{bmatrix}, \mathbf{b}_5 = \begin{bmatrix} 0.2\text{m} \\ 1.8\text{m} \end{bmatrix}, \mathbf{b}_6 = \begin{bmatrix} 1.8\text{m} \\ 1.8\text{m} \end{bmatrix} \right\}. \tag{3.13b}$$

Compare the estimated and the true trajectory of the robot for these beacon locations and the different noise variances $q_n = 1 \cdot 10^{-6}\,\text{m}^2$, $q_n = 5 \cdot 10^{-4}\,\text{m}^2$, and $q_n = 1 \cdot 10^{-3}\,\text{m}^2$.

To ensure an appropriate localization accuracy for every possible robot position $\mathbf{p} \in \mathcal{P}$, the actual beacon placement is significantly more critical for the non-convex enclosure in Figure 3.2 than it is for the simple convex enclosure in Figure 3.1. In this regard, it is natural to ask for the optimal beacon placement in this case.

Exercise 3.4 (Exercise during the lab). Analyze and optimize the properties of the estimation strategy developed in Exercise 3.3. To improve the speed of convergence when solving the subsequent optimization problem, i. e., to avoid problems with vanishing gradients, the analysis is based on the sensitivity for all (including hidden) beacons

$$\mathbf{S}(\mathbf{p}, \mathbf{b}_1, \ldots, \mathbf{b}_N) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right|_{\mathbf{p}, \mathbf{b}_1, \ldots, \mathbf{b}_N}. \tag{3.14}$$

The visibility of the individual beacons is taken into account by using the return values ℓ_n of the function `getObstruction` to modify the actual covariances q_n of the measurement noise v_n according to

$$\bar{q}_n = q_n + \lambda \ell_n, \tag{3.15}$$

where $\lambda \approx 100\text{m}$ is an additional tuning parameter of the optimization procedure. Subsequently, proceed as follows:

1. Visualize the A- and D-optimality criteria related to $\mathbf{S}$ for all allowed robot positions $\mathbf{p} \in \mathcal{P}$ and the beacon sets in (3.6). Use a grid spacing of 4cm and

the MATLAB routine imagesc or surf.

2. Visualize the (approximate) spatial direction of the largest variance of the position estimator developed in Exercise 3.3 for all possible robot positions $\mathbf{p} \in \mathcal{P}$ and the beacon sets in (3.13). Use the MATLAB routine quiver.

3. Calculate the optimal positions $\mathbf{b}_n^*$ for $N = 6$ beacons by solving the optimization problem

$$\mathbf{b}_1^*, \ldots, \mathbf{b}_N^* = \arg \min_{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N} \quad \int_{\mathcal{P}} r(\mathbf{R}(\mathbf{p}))\, \mathrm{d}\mathbf{p} \tag{3.16a}$$

$$\text{s.t.} \qquad \mathbf{b}_n \in \mathcal{P} \;, \quad \forall n = 0, \ldots, N \tag{3.16b}$$

with r according to the D-optimality measure and

$$\mathbf{R}(\mathbf{p}) = \mathbf{S}^{\mathrm{T}}(\mathbf{p}_{ij}, \tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N)\bar{\mathbf{Q}}^{-1}\mathbf{S}(\mathbf{p}, \tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_N), \tag{3.17}$$

where $\bar{\mathbf{Q}}$ denotes the modified noise covariance matrix built from (3.15). Consider the beacon sets in (3.13) as initial conditions in the optimization. Find a suitable discretization of (3.16a) and use the MATLAB routine fmincon to solve (3.16).

> **Remark:** The number of sampling points in the discretization of (3.16a) heavily influences the computation time of the optimizer. Keep the number of points reasonably low to facilitate a faster computation time. Similarly, choose adequate tolerances for fmincon.

4. Compare the estimated and the true trajectory of the robot for the obtained optimal beacon locations and the different noise variances $q_n = 1 \cdot 10^{-6}\,\mathrm{m}^2$, $q_n = 5 \cdot 10^{-4}\,\mathrm{m}^2$, and $q_n = 1 \cdot 10^{-3}\,\mathrm{m}^2$. Assume equal noise covariances for each beacon.

3.3 References

[3.1] A. Steinboeck, *Skriptum zur VU Optimierungsbasierte Regelungsmethoden (SS 2023)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2023. [Online]. Available: https://www.acin.tuwien.ac.at/master/optimierungs-basierte-regelungsmethoden/