Norwegian University of Science and Technology
Department of Mathematical Sciences

TMA4180
Optimisation I
Spring 2020

Exercise set "Project II"

1 Robots revisited

The second project in the course on optimisation will again be concerned with planar robots, but now with problems related to motion planning. Before introducing these problems, we will recall the notation introduced in the first project.

We consider again a planar robot consisting of n rigid segments of lengths ℓ_i, $1 \leq i \leq n$, that are connected by revolute joints. The hand or tool is attached, by means of a freely orientable joint, to the last segment, while the first segment is attached to the origin of the plane $\mathbb{R}^2$, again by means of a revolute joint. A sketch of such a robot with $n = 3$ segments is given in Figure 1.

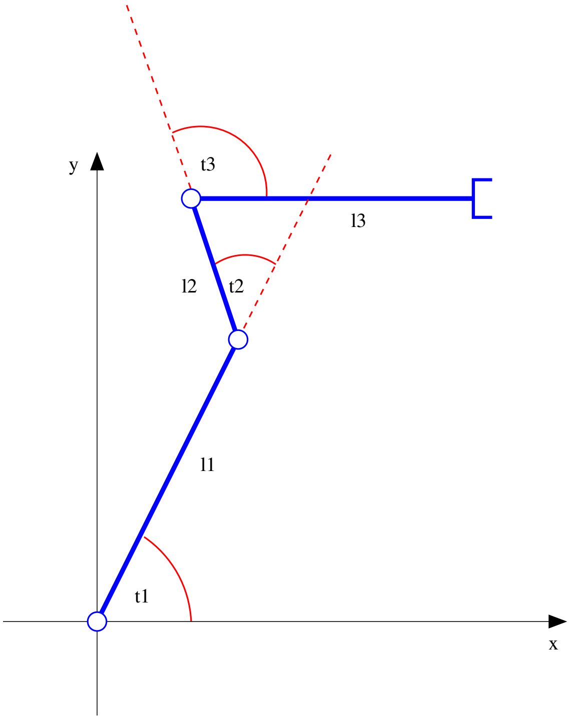

Figure 1: Sketch of the type of planar robots we consider in this project. The segments of the robot are of (fixed) lengths ℓ_i, $i = 1, 2, 3$. The joint angle ϑ_1 denotes the angle between the first segment and the x-axis. The subsequent joint angles ϑ_i denote the angle between the i^{th} and the $(i-1)^{\text{st}}$ segment. Note that, in this particular configuration, the angle ϑ_3 is negative, whereas the other two angles are positive.

The set of all points that are reachable by the hand is called the *configuration space* of the robot, and will be denoted by $\mathcal{C}$. The set of all possible joint parameters is denoted by $\mathcal{J}$ and called the *joint space* of the robot. In the situation we consider here, we can describe the robot completely by the set of the angles ϑ_i, $i = 2, \ldots, n$, between the i^{th} and $(i-1)^{\text{st}}$ segment of the robot, as well as the angle ϑ_1 between the x-axis and the first segment (again, see Figure 1). In that way, we can view the joint space $\mathcal{J}$ as a subset of $\mathbb{R}^n$.

In case the joints may move freely without any restrictions, the joint space is equal to the whole $\mathbb{R}^n$. As has been shown in the first project, the configuration space $\mathcal{C}$ in such a situation is either a disc of radius $\sum_i \ell_i$ or an annulus in case one of the segment lengths is larger than the sum of the lengths of all other segments. In this case, the outer radius of this annulus will be again equal to $\sum_i \ell_i$, but the inner radius will be equal to $\ell_j - \sum_{i\neq j} \ell_i$, where j is such that ℓ_j is maximal. Note, however, that the configuration space will usually be much more complicated and effectively impossible to describe in case the joints have to satisfy additional restrictions.

Moreover, we have seen that the configuration of the robot given the joint parameters is described by the function $F\colon \mathcal{J} \to \mathbb{R}^2$,

$$F(\vartheta) = \sum_{i=1}^{n} \ell_i \begin{pmatrix} \cos \sum_{j=1}^{i} \vartheta_j \\ \sin \sum_{j=1}^{i} \vartheta_j \end{pmatrix},$$

where $\vartheta = (\vartheta_1, \ldots, \vartheta_n)$ is the vector of joint angles. Additionally, the position of the k^{th} joint is given by

$$\sum_{i=1}^{k} \ell_i \begin{pmatrix} \cos \sum_{j=1}^{i} \vartheta_j \\ \sin \sum_{j=1}^{i} \vartheta_j \end{pmatrix}.$$

2 Motion planning

In the first part of the project, we have studied the problem of inverse kinematics, where one is given a point $p \in \mathbb{R}^2$ and wants to find joint parameters $\vartheta \in \mathcal{J}$ that either generate this configuration or come as close as possible. In this project, in contrast, we want to plan a path for a robot that visits a number of given points $p^{(1)}$,..., $p^{(s)} \in \mathcal{C}$ and then returns to $p^{(1)}$. One simple possibility for finding such a path would consist in solving the inverse kinematic problem for each of these points, that is, to find for each $i = 1, \ldots, s$ some $\vartheta^{(j)} \in \mathcal{J}$ satisfying $F(\vartheta^{(j)}) = p^{(j)}$, and then to rotate the joints of the robot with constant speed first from the angles $\vartheta^{(1)} \in \mathcal{J}$ to $\vartheta^{(2)}$, then on to $\vartheta^{(3)}$, $\vartheta^{(4)}$ and so on, and at the end back to $\vartheta^{(1)}$. However, the paths one obtains in this manner will essentially be random, and one cannot expect that they will be reasonable in any way.

As an alternative, one can try to choose the angles $\vartheta^{(j)}$ in such a way that the distances between consecutive angles on the path are minimal. In order words, try to find a path through the given points in such a way that the joints of the robot rotate as little as possible.

One possibility for doing so is the following: Given an s-tuple of joint parameters $\varTheta := (\vartheta^{(1)}, \dots, \vartheta^{(s)}) \in \mathcal{J}^s$, we define

$$E(\varTheta) := \frac{1}{2}\Big[\|\vartheta^{(2)} - \vartheta^{(1)}\|^2 + \cdots + \|\vartheta^{(s)} - \vartheta^{(s-1)}\|^2 + \|\vartheta^{(1)} - \vartheta^{(s)}\|^2\Big]$$

or, equivalently,

$$E(\varTheta) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{s-1}(\vartheta_i^{(j+1)} - \vartheta_i^{(j)})^2 + \frac{1}{2}\sum_{i=1}^{n}(\vartheta_i^{(1)} - \vartheta_i^{(s)})^2. \tag{1}$$

That is, the function E measures the sum of the squares of the distances between consecutive angles. Then we define the optimal path for the robot by interpolating between the joint angles solving the constrained optimisation problem

$$\min_{\varTheta \in \mathcal{J}^s} E(\varTheta) \qquad \text{subject to} \qquad F(\vartheta^{(j)}) = p^{(j)} \text{ for } j = 1, \dots, s. \tag{2}$$

3 Tasks

The main goal of this project is the numerical solution of the problem of planning a path of a robot through a set of given points. More specifically, you should solve the problem (2) with two different types of feasible regions $\mathcal{J}$:

1. Where the joints may move freely, and
2. Where each joint is limited to a given range of angles.

3.1 Solution of (2) for $\mathcal{J} = \mathbb{R}^n$

Consider first the situation where one wants to solve the problem (2) without any constraints on the joint angles, that is, with $\mathcal{J} = \mathbb{R}^n$.

$\boxed{1}$ Show that the optimisation problem (2) has a solution provided that all the points $p^{(j)}$ are reachable by the robot (that is, $p^{(j)} \in \mathcal{C}$ for all $j = 1, \dots, s$).

$\boxed{2}$ Discuss the properties of the optimisation problem (2).

Interesting properties that should be discussed are the coercivity of the optimisation problem and the consequences for the solution, but also the existence of KKT points (or: whether the LICQ holds). If possible, try to address the question of the uniqueness of the solution and also the existence of local minimisers.[1]

[1] For the discussion of uniqueness and existence of local minima, you can of course also rely on your numerical results.

$\boxed{3}$ Implement a numerical method that, given segment lengths ℓ_i, $1 \le i \le n$, and points $p^{(j)} \in \mathbb{R}^2$, $1 \le j \le s$, either computes a solution of the optimisation problem (2) or determines that one of the points is not an element of $\mathcal{C}$, in which case no solution exists.

As a simple test example, you may, for instance, use the following setup:

- The robot is given by: $\ell_1 = 3$, $\ell_2 = 2$, $\ell_3 = 2$.
- The points on the robot's path are: $p^{(1)} = (5, 0)$, $p^{(2)} = (4, 2)$, $p^{(3)} = (6, 0.5)$, $p^{(4)} = (4, -2)$, $p^{(5)} = (5, -1)$.

However, do not restrict yourself only to this example, but rather test your code in a variety of different settings.

3.2 Angular constraints

Now assume that there are some additional constraints on the movement of the joints. More precisely, we assume that all the joints have to satisfy the constraint

$$-c \le \vartheta_i^{(j)} \le c \tag{3}$$

for some $c > 0$, usually with $0 < c < \pi$. Then one has to solve the optimisation problem

$$\min_{\Theta \in \mathcal{J}^s} E(\Theta) \qquad \text{subject to} \qquad \begin{cases} F(\vartheta^{(j)}) = p^{(j)} & \text{for } j = 1, \dots, s, \\ -c \le \vartheta_i^{(j)} \le c & \text{for } j = 1, \dots, s, \text{ and } i = 1, \dots, n, \end{cases} \tag{4}$$

involving both equality and inequality constraints. Note, however, that the inequality constraints are of a very simple form compared to the equality constraints.

$\boxed{4}$ Discuss the properties of the optimisation problem (4). To which extent does the introduction of the additional inequality constraints (3) complicate or simplify the problem?

$\boxed{5}$ Implement a numerical method that, given segment lengths ℓ_i, $1 \le i \le n$, points $p^{(j)} \in \mathbb{R}^2$, $1 \le j \le s$, and a parameter $c > 0$ either computes a solution of the optimisation problem (4) or determines that one of the points is not an element of $\mathcal{C}$, in which case no solution exists.

As a simple test example, you may use the same setup as in Section 3.1 with the additional parameter $c = \pi/2$. In addition, use the same settings for the robot, but the points:

- $p^{(1)} = (-1, 5)$, $p^{(2)} = (-3, 3)$, $p^{(3)} = (-3, -4)$, $p^{(4)} = (0, 5)$, $p^{(5)} = (3, 2)$.

Because of the additional constraint (3), the configuration space $\mathcal{C}$ of the robot cannot be characterised as easily as in the unconstrained case. Therefore it is necessary to make sure that your algorithm correctly handles points that lie outside of $\mathcal{C}$. Your report should, of course, contain a description of the method you use for handling these situations.