Norwegian University of Science and Technology
Department of Mathematical Sciences

TMA4180
Optimization I
Spring 2016

Exercise set "project"

Instructions

- The deadline for submitting the project is Friday, April 15, at 16:00.
- You may work on the project either alone or in pairs. You are free to discuss the project in larger groups, but when you write the report or do the implementations or numerical tests, you should do so in the small groups.
- Write you candidate number(s) for the course Optimization I on the top of the report; **do not use names, do not use your student number**.
- Submit your reports electronically by e-mail to Torbjørn. The report should be submitted as a pdf-file; also attach your code in a zip-file (or a similar archive).
- The reports should be no longer than 10 pages; 5–6 pages (in LaTeX or equivalent) are perfect.
- You can write your project in English or Norwegian.
- When you refer to books or other sources of information, always provide citations.
- In your report, provide concise answers to the different questions. Additionally, provide sufficient details about the implementation (algorithm used, line-search method and parameters, starting values, parameter updates,...). You may include code snippets of important parts of the algorithms in the report, but do not provide listings of the full code. Also, report on the algorithmic performance and try to relate the actual performance with what is predicted by theory.
- You are more than welcome to experiment with different algorithms and also more complicated problem settings. Correct and relevant extra work may—to some degree—compensate for deficiencies in other tasks.
- The project counts for 30% of the final grade. If you don't hand it in, you may still take the exam, but you cannot achieve more than 70% in the course (which corresponds to a C).

1 Robots

We consider 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 (see Figure 1).

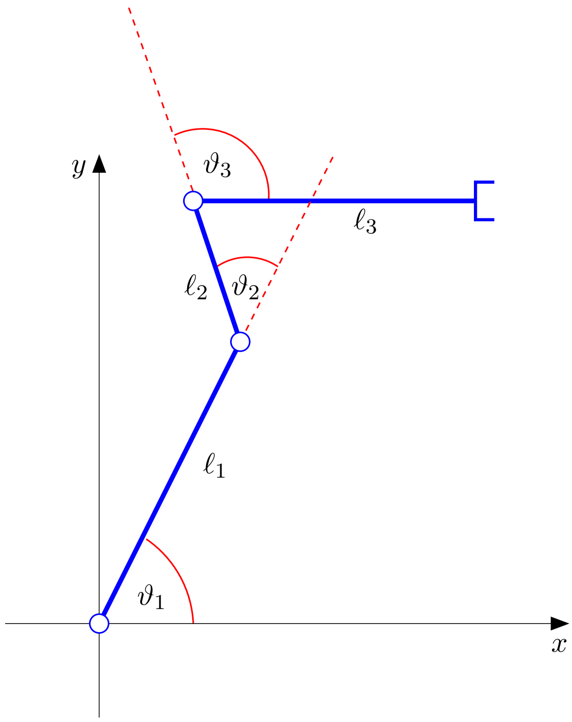

Figure 1: Sketch of the type of planar robots we consider in this project. The segments of the robot are of (fixed) lengths ℓ_i, $i = 1, 2, 3$. The joint angle ϑ_1 denotes the angle between the first segment and the x-axis. The subsequent joint angles ϑ_i denote the angle between the i^{th} and the $(i-1)^{\text{st}}$ 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 $\mathcal{C}$. Assuming that there are no restrictions for the joint angles (that is, each joint can perform rotations by 360°), it is easy to see that the configuration space of the robots we consider in this projects is always either an annulus or a disc centered at the origin. In case the lengths ℓ_i of the segments of the robot satisfy the compatibility condition $\ell_i \leq \sum_{j \neq i} \ell_j$ for all i, we actually have that the configuration space is a disc of radius $\sum_i \ell_i$ centered at the origin.

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 identify $\mathcal{J}$ with $\mathbb{R}^n$.

2 Forward and Inverse Kinematics

The *forward kinematic problem* in robotics is concerned with determining the configuration of the robot from the joint parameters. That is, one wants to find a function $F\colon \mathcal{J} \to \mathcal{C}$ mapping the joint parameters to the position of the end-effector. For the type of robots and the parametrisation of the joints we consider here, it is easy to see that F takes the form

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

where $\vartheta = (\vartheta_1, \ldots, \vartheta_n)$ is the vector of joint angles.

1 Implement a function that, given segment lengths ℓ_i, $1 \le i \le n$, calculates and plots the position of the robot's joints in the plane.

In contrast, in the *inverse kinematic problem* a target configuration $p \in \mathcal{C} \subset \mathbb{R}^2$ is given, and one wants to find joint parameters $\vartheta \in \mathcal{J}$ that generate this configuration. In other words, one wants to solve the equation $F(\vartheta) = p$ for ϑ. Note that this equation is usually ill-posed in the sense that it does not have a unique solution (a complete rotation of a joint of the robot, which can be obtained by adding 2π to the corresponding angle, does not change the configuration).

Also, for more complex robots also involving some constraints for the different joints, it can be difficult to decide whether a given point p can actually be reached by the robot. Thus it makes sense to replace the solution of the equation $F(\vartheta) = p$ by the solution of the optimisation problem

$$\min_{\vartheta \in \mathcal{J}} d(\vartheta) \qquad \text{with} \qquad d(\vartheta) := \frac{1}{2}\|F(\vartheta) - p\|^2. \tag{1}$$

2 Compute the gradient of the function $d\colon \mathcal{J} \to \mathbb{R}$ defined in (1).

3 Show that the function d is non-convex.

4 Implement a numerical method that, given segment lengths ℓ_i, $1 \le i \le n$, and a target configuration $p \in \mathbb{R}^2$, computes a solution of the optimisation problem (1).

Test your method at least for the following settings:

- $n = 3$, $\ell = (3, 2, 2)$, $p = (3, 2)$.
- $n = 4$, $\ell = (3, 2, 1, 1)$, $p = (3, 2)$.
- $n = 4$, $\ell = (3, 2, 1, 1)$, $p = (0, 0)$.

Implement any method you like for the solution of this unconstrained optimisation problem, but provide some justification for your choice in your report.

3 Optimal Control—Part I

Now we want to study the problem of moving the robot between different configurations. More precisely, we want to move the robot in unit time with constant speed φ from one configuration $p \in \mathcal{C}$ described by the joint parameters $\vartheta \in \mathcal{J}$ to some target configuration q. In other words, we want to find some $\varphi \in \mathcal{J}$ such that $F(\vartheta + \varphi) = q$.

In order to obtain a reasonable solution for this problem, we now assign a cost to the movement speed φ, and try to minimise the cost over all admissible movement speeds. The simplest possibility is the use of the squared Euclidean norm of φ as cost functional. Then we arrive at the constrained optimisation problem

$$\min_{\varphi \in \mathcal{J}} \frac{1}{2}\|\varphi\|^2 \qquad \text{subject to } F(\vartheta + \varphi) = q. \tag{2}$$

For the numerical solution of (2), we can use the *quadratic penalty method* (see Nocedal & Wright, Chapter 17.1; in particular Algorithm 17.1). Here the idea is to replace the problem (2) by an unconstrained problem of the form

$$\min_{\varphi \in \mathcal{J}} Q(\varphi;\mu) \qquad \text{with} \qquad Q(\varphi;\mu) = \frac{1}{2}\|\varphi\|^2 + \frac{\mu}{2}\|F(\vartheta + \varphi) - q\|^2,$$

depending on some parameter $\mu > 0$. As μ tends to infinity, the minimisers of $Q(\cdot;\mu)$ converge to solutions of (2).

5 Implement a numerical method based on the quadratic penalty method that, given segment lengths ℓ_i, $1 \le i \le n$, initial angles $\vartheta \in \mathcal{J}$, and a target configuration $q \in \mathcal{C}$ computes an approximate solution of the optimisation problem (2).

Test your method at least for the following settings:

- $n = 3$, $\ell = (3, 2, 2)$, $\vartheta = (\pi/2, \pi/2, \pi/2)$, $q = (2, 1)$.
- $n = 4$, $\ell = (3, 2, 1, 1)$, $\vartheta = (\pi/2, \pi/2, \pi/2, \pi/2)$, $q = (0, 0)$.
- $n = 3$, $\ell = (3, 2, 2)$, $\vartheta = (0, 0, 0)$, $q = (-5, 0)$.

4 Optimal Control—Part II

As a next step, we drop the assumption that the movement of the robot needs to be uniform. Instead, we divide the time interval $[0, 1]$ into s subintervals $(\tau_{i-1}, \tau_i) = \big((i-1)/s, i/s\big)$, $i = 1, \ldots, s$, and only require the movement speed to be constant on each of these intervals. That is, we want to find for each interval (τ_{i-1}, τ_i) a movement speed $\varphi^{(i)} \in \mathcal{J}$ of the robot such that the robot arrives at time $\tau = 1$ at the target configuration q and the total movement speed is minimal. Since the final joint parameters are given by $\vartheta + \frac{1}{s}\sum_{i=1}^{s} \varphi^{(i)}$, this leads to the constrained optimisation problem

$$\min_{\substack{\varphi^{(i)} \in \mathcal{J}\\ i=1,\ldots,s}} \frac{1}{2}\sum_{i=1}^{s} \frac{1}{s}\|\varphi^{(i)}\|^2 \qquad \text{subject to } F\Big(\vartheta + \frac{1}{s}\sum_{i=1}^{s} \varphi^{(i)}\Big) = q. \tag{3}$$

6 Show that the problems (2) and (3) are equivalent in the following sense: We have that $(\varphi^{(i)})_{i=1}^s \in \mathcal{J}^s$ is a solution of (3), if and only if $\varphi^{(i)} = \varphi^{(j)}$ for all i, j, and each $\varphi^{(i)}$ is a solution of (2).

The next numerical task is an implementation of the optimisation problem (3), again using a quadratic penalty method. As a preparation, it will be necessary to compute the gradient of the penalty term.

7 Compute the gradient of the function

$$D\colon \mathcal{J}^n \to \mathbb{R}, \qquad D\big((\varphi^{(i)})_{i=1}^s\big) = \frac{1}{2}\Big\| F\Big(\vartheta + \frac{1}{s}\sum_{i=1}^s \varphi^{(i)}\Big) - q\Big\|^2.$$

8 Implement a numerical method based on the quadratic penalty method for the solution of (3) and verify numerically the theoretical result of Problem 6.

5 Constraints of the Joint Space

Now we add additional constraints to the joint space $\mathcal{J}$ in the form that we restrict the possible movement of the different joints: We assume that each joint angle ψ_j, $j = 1, \ldots, n$, at each time has to satisfy the constraints

$$-\frac{3\pi}{4} \le \psi_j \le \frac{3\pi}{4}.$$

Within the setting of the optimal control problem discussed in Section 4, this leads to the additional constraints

$$-\frac{3\pi}{4} \le \vartheta_j + \frac{1}{s}\sum_{i=1}^k \varphi_j^{(i)} \le \frac{3\pi}{4}, \qquad k = 1, \ldots, s,\ j = 1, \ldots, n. \tag{4}$$

Define now the *logarithmic barrier functional*

$$Lb\big((\varphi^{(i)})_{i=1}^s\big) := \sum_{k=1}^s \sum_{j=1}^n \Big[\log\Big(\frac{3\pi}{4} + \vartheta_j + \frac{1}{s}\sum_{i=1}^k \varphi_j^{(i)}\Big) + \log\Big(\frac{3\pi}{4} - \vartheta_j - \frac{1}{s}\sum_{i=1}^k \varphi_j^{(i)}\Big)\Big].$$

It is easy to see that this functional is finite, if and only if all the constraints in (4) are strictly satisfied. Moreover, the value of this functional tends to $-\infty$, as the joint angles approach the boundary of the feasible domain.

Consider now the unconstrained optimisation problem

$$\min_{\substack{\varphi^{(i)} \in \mathcal{J} \\ i=1,\ldots,s}} Q\big((\varphi^{(i)})_{i=1}^s; \mu, \beta\big) := \frac{1}{2}\sum_{i=1}^s \frac{1}{s}\|\varphi^{(i)}\|^2 + \mu D\big((\varphi^{(i)})_{i=1}^s\big) - \beta Lb\big((\varphi^{(i)})_{i=1}^s\big). \tag{5}$$

Then the solutions of this problem are, with large parameter $\mu > 0$ and small parameter $\beta > 0$, reasonable approximations of the solution of the problem (3) with additional joint constraints given by (4).

9 Implement a numerical method for the solution of the unconstrained problem (5). Pay attention to the fact that the step length in a line search or trust region method has always to be chosen in such a way that the logarithmic barrier function stays finite.

Test your method at least on the following examples:

- $n = 3$, $\ell = (3, 2, 2)$, $\vartheta = (0, \pi/2, 0)$, $q = (3, -4)$, $s = 20$.
- $n = 3$, $\ell = (3, 2, 2)$, $\vartheta = (\pi/2, \pi/2, \pi/2)$, $q = (2, 1)$, $s = 20$.
- $n = 4$, $\ell = (3, 2, 1, 1)$, $\vartheta = (\pi/2, \pi/2, \pi/2, \pi/2)$, $q = (0, -1)$, $s = 40$.