UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR ELECTRICAL
ENGINEERING IN MEDICINE

RO4001 – Model Predictive Control

Exercise Sheet 3

Fall semester 2020/21 Nov. 17, 2019

The exercises may be solved individually or in small groups. You are *not* allowed to use a calculator or computer unless this is explicitly stated.

Exercise 1 (convex sets)

a) Which of the following sets are convex? Give a proof or a counter example.

i) For a positive definite matrix $M \succ 0$, the set

$$\left\{ x \in \mathbb{R}^2 \,\middle|\, x^{\mathrm{T}} M^{-1} x \leq 1 \right\} .$$

ii) A *wedge*, i.e., for some $a_1, a_2 \in \mathbb{R}^n$ and $\beta_1, \beta_2 \in \mathbb{R}$ the set:

$$\left\{ x \in \mathbb{R}^n \,\middle|\, a_1^{\mathrm{T}} x \leq \beta_1 \,,\;\; a_2^{\mathrm{T}} x \leq \beta_2 \right\} .$$

iii) The following transformation of a convex set $C \subseteq \mathbb{R}^n$, defined via $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$:

$$\{ Ax + b \in \mathbb{R}^m \mid x \in C \} .$$

iv) The set of points closer to one set $S \subseteq \mathbb{R}^n$ than another $T \subseteq \mathbb{R}^n$, i.e., if $\mathrm{d}(x, y)$ denotes the Euclidean distance between two points $x, y \in \mathbb{R}^n$ and the distance between a point x and a set S is defined as $\mathrm{d}(x, S) \triangleq \inf_{y \in S} \mathrm{d}(x, y)$:

$$\{ x \in \mathbb{R}^n \mid \mathrm{d}(x, S) \leq \mathrm{d}(x, T) \} .$$

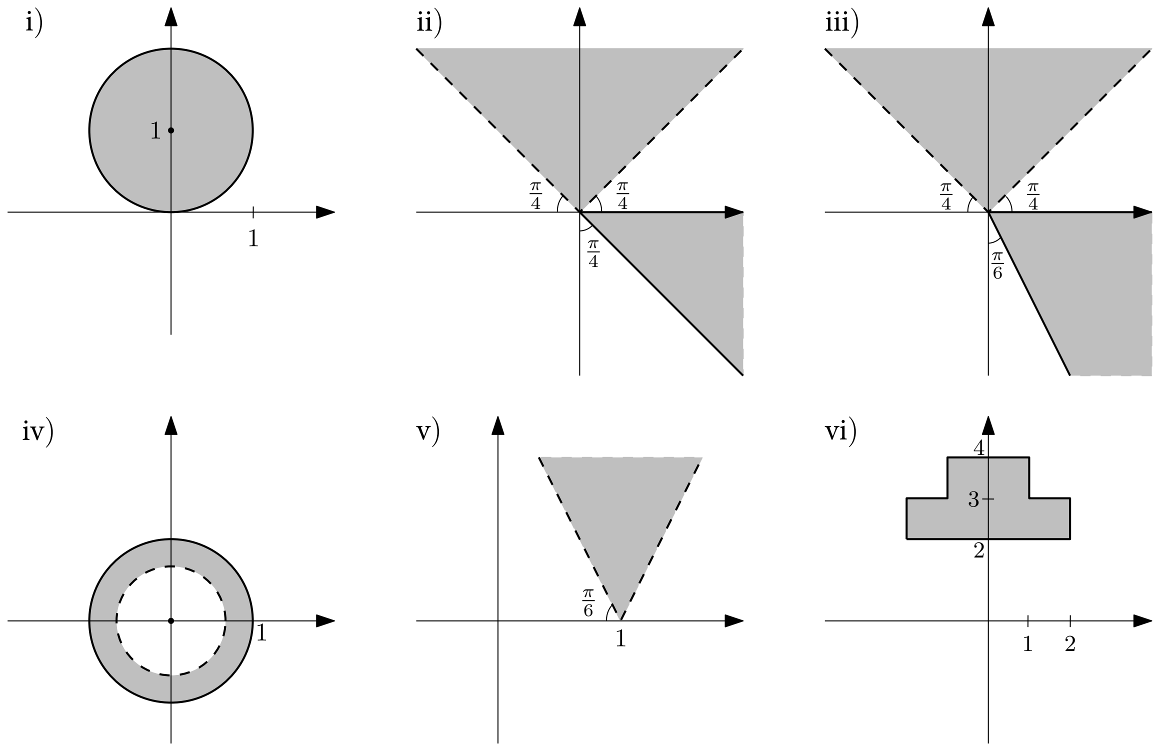
v) For any function $f : S \to \mathbb{R}$, where $S \subset \mathbb{R}^n$ is any (not necessarily convex) set, the set:

$$\{ x \in \mathbb{R}^n \mid \|x - y\| \leq f(y) \;\text{ for all }\; y \in S \} .$$

vi) If $x = [x_1 \;\; x_2 \;\; \cdots \;\; x_n]^{\mathrm{T}}$, the set:

$$\left\{ x \in \mathbb{R}^n \,\middle|\, \left| \sum_{i=1}^{n} x_i \cos(it) \right| \leq 1 \;\text{ for }\; |t| \leq \frac{\pi}{3} \right\} .$$

b) Characterize the following sets according to whether they are open, closed, convex and/or cones, and if they are cones whether the are pointed and/or proper.



Exercise 2 (convex functions)

For each of the following functions, determine whether they are (strictly) convex, (strictly) concave, or none of the four.

a) $f : \mathbb{R} \to \mathbb{R}$ defined as $f(x) = e^x - 1$.

b) $f : \mathbb{R} \setminus \{0\} \to \mathbb{R}$ defined as $f(x) = \frac{1}{x}$.

c) $f : \mathbb{R}_{++} \to \mathbb{R}$ defined as $f(x) = \frac{1}{x}$.

d) $f : \mathbb{R}^2_{++} \to \mathbb{R}$ defined as $f(x_1, x_2) = x_1 x_2$.

e) $f : \mathbb{R}^2_{++} \to \mathbb{R}$ defined as $f(x_1, x_2) = \frac{1}{x_1 x_2}$.

f) $f : \mathbb{R}^n \to \mathbb{R}$ defined as $f(x) = \|Ax + b\|_2$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

g) $f : \mathbb{R} \setminus \{0\} \to \mathbb{R}$ defined as $f(x) = \frac{1}{|x|^\alpha}$, where $\alpha \geq 0$.

Exercise 3 (linear inequalities for a tetrahedron)

Derive a mathematical expression of the form $Ax \leq b$ for the tetrahedron depicted in Figure 1.

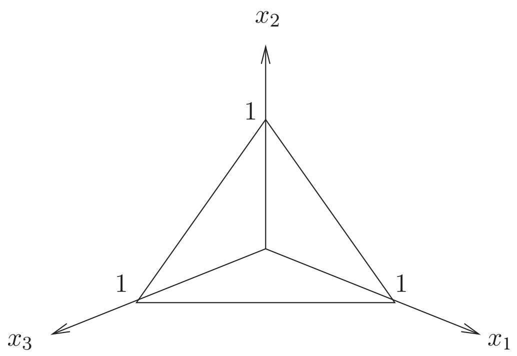


Figure 1: A tetrahedron

Exercise 4 (nonlinear programming)

Note: Use MATLAB (command `fmincon`) to solve this exercise.

Consider the following nonlinear program (NLP):

$$\min_{x \in \mathbb{R}^2} \; x_2 \tag{1a}$$
$$\text{s.t. } x_1^2 + 4x_2^2 \leq 4 \tag{1b}$$
$$x_1 \geq -2 \tag{1c}$$
$$x_1 = 1 \tag{1d}$$

a) How many inequality and equality constraints does problem (1) have?

b) Sketch the feasible set of problem (1). What is the optimal solution?

c) Bring the problem into the NLP standard form:

$$\min_{x \in \mathbb{R}^2} \; f(x) \tag{2a}$$
$$\text{s.t. } g(x) \leq 0 \tag{2b}$$
$$h(x) = 0 \tag{2c}$$

by defining the dimension n and the functions f, g, h along with their dimensions appropriately. Can the problem be cast as a standard LP or QP? Is the problem convex?

d) Start MATLAB and use `help fmincon` to learn about the syntax of MATLAB's NLP solver `fmincon`. Then formulate three MATLAB functions for f, g, h of problem (1) and choose an initial guess for x. Solve the problem using `fmincon`. Check that the output corresponds to what you expected.

Exercise 5 (Chebyshev center of a polyhedron)

Note: Use MATLAB (command `linprog`) to solve this exercise.

The goal of this exercise is to compute the largest Euclidean ball that fits into a given polyhedron

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\} \quad ,$$

where $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^p$ are known. The Euclidean ball is represented as

$$\mathcal{B} = \{x_c + v \in \mathbb{R}^n \mid \|v\|_2 \leq r\} \quad ,$$

where $x_c \in \mathbb{R}^n$ is the (unknown) center of the ball and $r \in \mathbb{R}_+$ is its (unknown) radius. Note that the ball's center is called the *Chebyshev center* of the polyhedron. It is the point that is the farthest away (namely by r) from the boundary of $\mathcal{P}$.

a) It shall be shown that computing the Euclidean ball can be formulated as a convex optimization problem.

  i) Consider only one of the linear constraints that defines $\mathcal{P}$, which is denoted $a_i^\mathrm{T} x \leq b_i$ for $i \in \{1, 2, \ldots, p\}$. Show that the constraint for the Euclidean ball to be contained in this halfspace of $\mathbb{R}^n$ can be written as

$$a_i^\mathrm{T} x_c + \|a_i\|_2 r \leq b_i \quad , \tag{3}$$

  which is a linear constraint in the decision variables x_c and r. (*Hint:* You may use the fact that $\sup\left\{a_i^\mathrm{T} v \mid \|v\|_2 \leq r\right\} = r\|a_i\|$.)

  ii) Show that the problem of computing the Euclidean ball can be formulated as an optimization problem. To which class does it belong?

b) Now write a MATLAB script that computes the Euclidean ball for an arbitrary polytope. Your script shall

  i) Take a deterministic polytope $\mathcal{P}$ or generate a random polytope $\mathcal{P}$ via the inputs A and b,

  ii) Use the command `linprog` (use `help linprog` to learn about the syntax) to solve the resulting optimization problem,

  iii) Be verified by computing the Chebyshev center for the polyhedron in $\mathbb{R}^2$ given by

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 0 \\ 1 & -3 \\ -1 & -1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix} , \qquad b = \begin{bmatrix} 6 \\ 2 \\ 9 \\ 5 \\ 3 \\ 3 \end{bmatrix} .$$

  (*Solution*: $x_c^\star = \begin{bmatrix} -0.543 & -0.591 \end{bmatrix}^\mathrm{T}$, $r^\star = 2.457$)

Exercise 6 (least squares)

Note: Use MATLAB (CVX toolbox) to solve this exercise.

This exercise is an introduction to *Disciplined Convex Programing* via CVX, which is a very powerful and versatile toolbox for solving convex optimization problems in MATLAB.

a) Go to the CVX web site (`http://cvxr.com`) and download the current version of CVX. Install CVX by executing the following steps:

  – Copy the files into an arbitrary folder on your computer.

  – In MATLAB, navigate into this folder and run the command `cvx_setup`.

b) Download the CVX manual to get an idea about what you can do with this software `http://cvxr.com/cvx/doc/CVX.pdf`

c) To get started with CVX, we will solve a simple least-squares problem:

$$\min_{a_0,a_1,a_2,a_3,a_4,a_5} \sum_{i=1}^{N} \left\| y_i - \left(a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + a_4 x_i^4 + a_5 x_i^5\right) \right\|_2 \quad . \tag{4}$$

Here $(x_i, y_i) \in \mathbb{R}$ are $N = 100$ given data points and $a_0, a_1, a_2, a_3, a_4, a_5 \in \mathbb{R}$ are the coefficients of a second order polynomial.

i) Open a new script in MATLAB. Generate the data points x_i as follows:

$$x_i \sim \mathcal{U}\left([-2,2]\right) \qquad \forall\ i = 1,2,\ldots,N \ ,$$

where $\mathcal{U}\left([-2,2]\right)$ denotes the uniform distribution on the interval $[-2,2]$. Generate the true polynomial coefficients randomly as

$$\bar{a}_j \sim \mathcal{N}(0,1) \qquad \forall\ j = 0,1,\ldots,5 \ ,$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ. Finally, compute the data points as

$$y_i = \sum_{j=0}^{5} \bar{a}_j x_i^j + \varepsilon_i \ , \qquad \text{where } \ \varepsilon_i \sim \mathcal{N}(0,100) \ , \qquad \forall\ i = 1,2,\ldots,N \ .$$

ii) The CVX code for solving the least squares optimization problem is given below:

```
cvx_begin
    variable x_c(6)
    minimize( norm(A*x_c-b,2) );
cvx_end
```

Set up the matrices `A` and `b` and solve for the estimated polynomial coefficients `x_c`$\in \mathbb{R}^6$. Compare the CVX solution with the MATLAB least squares solution that is obtained with the backslash operator:

```
x_ls=A\b;
```

d) One big advantage of an optimization problem is its general formulation that allows the addition of arbitrary constraints. They can be used, for example, to implement a priori knowledge about the coefficients or to impose restrictions on the desired solution. As an experiment, we will now

- restrict the random values of $\bar{a}_4, \bar{a}_5$ to the interval $[-0.5, +0.5]$, and
- impose corresponding constraints on the computed coefficients `x_d`$\in \mathbb{R}^6$, by modifying the CVX problem as follows:

```
cvx_begin
    variable x_d(6)
    minimize( norm(A*x_d-b,2) );
    subject to
        (list of constraints)
cvx_end
```

i) Insert the appropriate list of constraints and solve for `x_d`$\in \mathbb{R}^6$.

ii) Create a figure in MATLAB that shows the data points along with the curves corresponding to the least squares solution `x_c`$\in \mathbb{R}^6$ and the constrained solution `x_d`$\in \mathbb{R}^6$. What can you conclude? (*Hint:* You can use the following line specifications in MATLAB to adjust the layout, `plot(X,Y,'k',`*specs*`)`, where *specs* may include options such as `'LineWidth'2`, `'LineStyle','none'`, `'MarkerEdgeColor','k'`, `'MarkerFaceColor','k'`, `'MarkerSize',8`.)

e) Now the cost function (4) shall be modified to include an additional Tikhonov regularization term:

$$\min_{a_0,a_1,a_2,a_3,a_4,a_5} \sum_{i=1}^{N} \left\| y_i - \left(a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + a_4 x_i^4 + a_5 x_i^5\right) \right\|_2 + \gamma \left\| a \right\|_1 \quad , \tag{5}$$

where $a \triangleq [a_0 \;\; a_1 \;\; a_2 \;\; a_3 \;\; a_4 \;\; a_5]^{\mathrm{T}}$ and γ is a weighting parameter (you can start with $\gamma = 1$). This approach is used to promote sparsity of the solution. (It is also known as a *Lasso regression*.)

i) Adjust the CVX formulation accordingly and solve for the solution `x_e`$\in \mathbb{R}^6$. Compare `x_e` with `x_c` and `x_d` – what do you observe?

ii) Add the curve corresponding to `x_e` to the MATLAB figure that also shows the data points and the other curves. What do you conclude when comparing the three solutions?