Exam — November 2020 – Grading template
Optimization in Systems and Control (SC42055/6)

Important: Please recall the following instructions from the exam procedure:

- Note that – just as in previous years – **correct results without proper and correct motivation will not receive any marks.**

For an example on how proper and correct motivations look like, please consult the worked solutions for Sample Exams 1 and 2 and for the exams of October 2013 and October 2014

Additional scoring guidelines

S0: correct result without proper and correct motivation: 0

S1: wrong α or β: 0 for corresponding (sub)question

S2: small computation error that does affect result: -0.5

S3: partially incomplete motivation for convexity or simplification: -50%

S4: missing, wrong, or not properly motivated $N\times$ in Question 1: 0 for (c) and -1 for (d)

S5: multi-start listed when it is not needed: -0.5

S6: ∇f as row vector: -0.5

S7: if in Question 1 the answer for (c) and/or (d) is formally correct, but an error is made in (b) that affects the result for (c) and/or (d): 0 for (c) and/or (d)

S8: redundant function in stopping criterion: 0

S9: introduction of redundant variables that are not needed at all: -0.5

On the next pages **short** answers are given with scores marked in red. To earn the indicated score the corresponding answer has to be given completely, including the information inside the brackets; else the score is 0.

QUESTION 1: Optimization methods I ($8 \times 7.5 = 60$ points)

- **P1**

1 (a) get rid of 3rd root (as it is an increasing function) and of the constant 6
elimination not possible

3 (b) convex objective (as it is a sum of even powers and a square – note that $x_3 x_4$ can be absorbed: $x_3^2 - x_3 x_4 + x_4^2 = (x_3 - 0.5x_4)^2 + 0.75x_4^2$ or $0.5x_3^2 + 0.5(x_3 - x_4)^2 + 0.5x_4^2$): 1.5
nonconvex constraint (as we have a non-affine equality constraint): 1.5

1 (c) NCC: nonconvex constrained optimization problem

2 (d) multi-start: 0.5
M8: Lagrange method + Davidon-Fletcher-Powell quasi-Newton algorithm: 1.5

0.5 (e) Lagrange conditions with ε

- **P2**

1 (a) note that the 1st constraint makes argument strictly positive, so the function $(\cdot)^{-5}$ is decreasing: 0.5
get rid of $(\cdot)^{-5}$ and remove the constant -7, which yields **min** $5x_1 + 4x_2 - 2x_3 - 8x_4$: 0.5
for simplifications of the constraints, see (b)

3 (b) linear objective function: 1
1st constraint linear
2nd constraint can be transformed into 2^4=16 linear constraints: $\pm x_1 \pm x_2 \pm x_3 \pm x_4 \leqslant 121$: 1.5
3rd constraint can be transformed into 2 linear constraints: $-196 \leqslant 7x_1 - \ldots \leqslant 196$: 0.5
4th constraint is linear

1 (c) MILP: mixed-integer linear programming problem
if the objective function was not identified or not proven to be linear in (b), then S7 applies and the scores for (c) and (d) will be 0

2 (d) M11: Branch-and-bound method for mixed-integer linear programming

0.5 (e) optimum found once entire tree is explored (note that solution set is bounded, so tree is finite)

- **P3**

1 (a) no simplification possible (except for substituting $y(k)$ in the objective function to get an expression in θ only): 0.5
for simplifications of the constraints, see (b): 0.5

3 (b) note that $y(k)$ and thus also $z(k,\theta)$ depends linearly on θ, which means that the objective function is convex quadratic: 2
constraint can be transformed into $2^7 = 128$ linear constraints: $\pm\theta_1 \pm \theta_2 \pm \ldots \leqslant 10 + \alpha$: 1

1 (c) QP: convex quadratic programming problem

2 (d) M3: Ellipsoid algorithm: 1
as there is no dedicated QP algorithm: 1

0.5 (e) $\|x_k - x^*\| \leqslant \varepsilon_x$ or $|f(x_k) - f(x^*)| \leqslant \varepsilon_f$ **and** $g(x_k) \leqslant \varepsilon_g$

- **P4**

  1 (a) if α is odd, note that the absolute values make the argument nonnegative, so the function $(\cdot)^{3+\alpha}$ is increasing; if α is even, then the function $(\cdot)^{3+\alpha}$ is increasing $\rightarrow$ get rid of $(\cdot)^{3+\alpha}$ and drop the constant 7: 0.5
  this yields $\mathbf{min}_{x\in\mathbb{R}^5}\ 13|x_1|+2|x_2|+8|x_3|+6|x_4|+3|x_5|$
  introduce dummy variables α_i with $\alpha_i \geqslant |x_i|$ to obtain **min** $13\alpha_1+2\alpha_2+8\alpha_3+6\alpha_4+3\alpha_5$
  for simplifications of the constraints, see (b): 0.5

  3 (b) objective function in α_i is linear: 0.5
  1st constraint can be recast as **2** linear constraints: $-16 \leq 5+7x_1+\ldots \leqslant 16$: 0.5
  2nd constraint can be written as **4** linear constraints: $-8+3x_5 \leqslant 9$, $4+7x_1+\ldots \leqslant 9$, $4x_3+3x_4 \leqslant 9$, $2x_1+x_5-9 \leqslant 9$: 0.5
  3rd constraint is nonconvex *but* can be written as the **union** of 10 linear constraints: $x_1 \geqslant 2+\beta$ OR $x_1 \leqslant -2-\beta$ OR $x_2 \geqslant 2+\beta$ OR $x_2 \leqslant -2-\beta$ OR ... OR $x_5 \leqslant -2-\beta$: 1
  Each constraint $\alpha_i \geqslant |x_i|$ can be written as 2 linear constraints: $\alpha_i > x_i$ and $\alpha_i \geqslant -x_i$: 0.5

  1 (c) $10 \times$ LP (see also Footnote 1): linear programming problem

  2 (d) $10 \times$ M1 (see also Footnote 1): Simplex algorithm for linear programming

  0.5 (e) simplex algorithm will always find a global optimum in a finite number of iterations

- **P5**

  1 (a) no simplification possible for the objective function: 0.5
  for simplifications of the constraints, see (b): 0.5

  3 (b) The first term of the objective function is convex (as $(\cdot)^4-1$ is convex (even power of an affine argument) and as $\exp(\cdot)$ is convex and increasing). For the second term there are two cases:

  - If β is odd, the second term becomes $-\log_3(\cdot)$ with a linear argument; so then the term is convex.
  - If β is even, the second term becomes $\log_3(\cdot)$; so then the term is not convex.

  So the objective function is convex if β is odd and nonconvex if β is even: 1.5
  1st constraint can be rewritten as 3 linear constraints: $7x_1+9x_2 \geqslant 1$, $8x_3+x_4 \geqslant 1$, $x_2+\ldots \geqslant 1$: 0.5
  2nd constraint becomes $x_1^2+\ldots \leqslant 27$ if α is odd; this is a convex constraint as even powers of x_i are convex. 2nd constraint becomes $-27 \leqslant x_1^2+\ldots$ if α is even; as this constraint always holds for any $x \in \mathbb{R}^4$ this is actually also a convex constraint: 1

  1 (c) For β odd: CP: convex optimization problem
  For β even: NCC: nonconvex constrained optimization problem

  2 (d) For β odd: M3: Ellipsoid algorithm: 2
  For β even: multi-start: 0.5 + M10 : Barrier function approach + steepest descent method: 1.5
  If β is even and α is even, then multi-start : 0.5 + M2: Gradient projection method with variable step size line minimization is a valid alternative: 1.5

  0.5 (e) For β odd: $\|x_k-x^*\| \leqslant \varepsilon_x$ or $|f(x_k)-f(x^*)| \leqslant \varepsilon_f$ **and** $g(x_k) \leqslant \varepsilon_g$
  For β even: KKT conditions with ε (list them!) **OR** (for M10) $\|\nabla f_{\text{barrier+}}(x_k)\| \leqslant \varepsilon$ (where $f_{\text{barrier+}}$ is the sum of the simplified objective function and the barrier function)

- **P6**

1 (a) no simplification possible for the objective function (except for dropping the term 7β: 0.5 for simplifications of the constraints, see (b): 0.5

3 (b) As term $-\alpha x_1 x_2$ cannot be absorbed the objective function is nonconvex: 0.5
It is easy to verify that the product of cosh functions cannot be concave as it is the product of U-shaped functions. So this product can either be convex or nonconcave+nonconvex. However, even if the product would be convex[1], the left-hand constraint $\cosh(3) \leqslant \cosh(2x_1)\cosh(4x_2)\ldots$ would be nonconvex. So the first constraint is nonconvex: 1
2nd constraint can be written as the **union** of 2 linear constraints: $x_1 + x_2 + 2x_3 + x_4 \geqslant 4$ OR $4x_1 - x_2 + 6x_3 - 5x_4 \geqslant 4$: 1
3rd constraint is convex as the left-hand side is a sum of even powers: 0.5

1 (c) NCC: nonconvex constrained optimization problem

2 (d) multi-start: 0.5 + M10 : Barrier function approach + steepest descent method: 1.5

0.5 (e) KKT conditions with ε (list them!) **OR** $\|\nabla f_{\text{barrier+}}(x_k)\| \leqslant \varepsilon$ (where $f_{\text{barrier+}}$ is the sum of the simplified objective function and the barrier function)

- **P7**

1 (a) As $\sinh(\cdot)$ is an increasing function we can maximize its argument; next, the constant 1 can be dropped: 0.5
as $-\cosh$ is a decreasing function for **nonnegative** arguments (note that the term x_3x_4 can be absorbed), we can minimize the argument. So we finally get $\min_{x\in\mathbb{R}^4} 2x_1^2 + 4x_2^2 + x_3^2 + (x_3 - 4x_4)^2$: 0.5
for simplifications of the constraints, see (b)

3 (b) the objective function can be written as a sum of squares (see (a)), so it is a convex quadratic function: 1.5
1st constraint is linear
since an even power is a U-shaped function, the second constraint can be transformed into the **union** of 2 linear constraints: $7x_1 + \ldots \leqslant -\sqrt[2\beta]{\alpha}$ OR $7x_1 + \ldots \geqslant \sqrt[2\beta]{\alpha}$: 1.5

1 (c) $2\times$ QP: convex quadratic programming problem

2 (d) $2\times$ M3: Ellipsoid algorithm: 1
as there is no dedicated QP algorithm: 1

0.5 (e) $\|x_k - x^*\| \leqslant \varepsilon_x$ or $|f(x_k) - f(x^*)| \leqslant \varepsilon_f$ **and** $g(x_k) \leqslant \varepsilon_g$

[1] Actually, the production function is convex as can be shown by expanding the following argument: The function $g(x,y) = \cosh(x)\cosh(y)$ is convex as if can be written as the sum of exponential functions with linear arguments; indeed, $g(x,y) = \frac{e^x + e^{-x}}{2}\frac{e^y + e^{-y}}{2} = 0.25(e^x e^y + e^{-x}e^y + e^x e^{-y} + e^{-x}e^{-y}) = 0.25(e^{x+y} + e^{-x+y} + e^{x-y} + e^{-x-y})$.

- **P8**

1 (a) no simplification possible

3 (b) due to the presence of cos and sin the objective function is nonconvex: 3
the constraints are linear

1 (c) NCC: nonconvex constrained optimization problem

2 (d) multi-start: 0.5 + M2: Gradient projection method with variable step size line minimization **OR** M10: Barrier function approach + steepest descent method: 0.5
It is important to indicate that although the numerical evaluation of the gradient is tedious in this case (due to the need for numerical integration and the large number of variables), there are no gradient-free methods available: 1

0.5 (e) KKT with ε (list them!) for M2 **OR** $\|\nabla f_{\text{barrier+}}(x_k)\| \leqslant \varepsilon$ (where $f_{\text{barrier+}}$ is the sum of the simplified objective function and the barrier function)

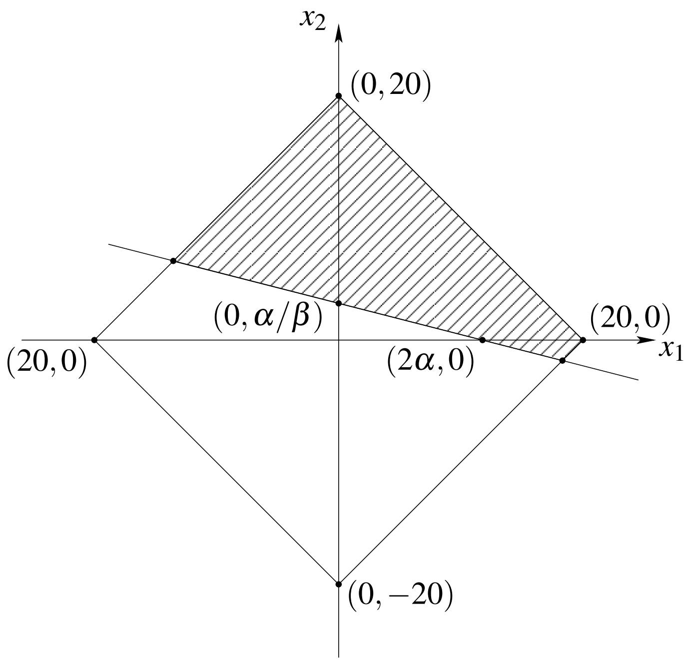
QUESTION 2: Optimization methods II $(10 + 12 = 22$ points)

- **Question 2.a**

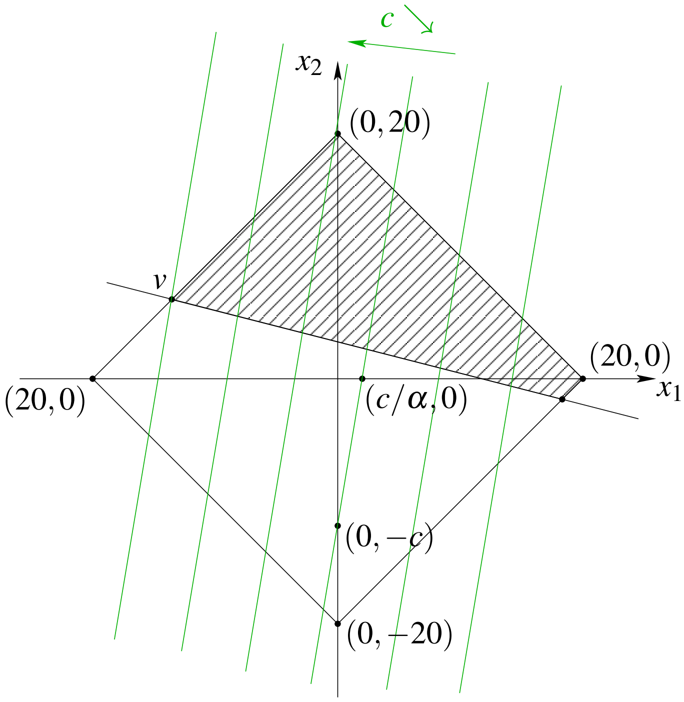
2 (a) Note that the question is about the **original, unsimplified problem!**

As $3^{(\cdot)}$ is a convex function with a linear argument, the objective function is convex: 1
The first constraint is convex as a norm function is convex in its argument: 0.5
The second constraint is linear and thus convex
Hence, the given problem is convex: 0.5

7 (b) First we simplify the optimization problem by making use of the fact that $3^{(\cdot)}$ is an increasing function. Moreover, the constraint $\|x\| \leqslant 20$ can be written as 4 linear constraints $\pm x_1 \pm x_2 \leqslant 20$ corresponding to a diamond-shaped set (see below). The last constraint is linear. So we actually have to solve a linear programming problem: 2
We solve it by using the graphical solution method. The feasible set is indicated below by the hashed area where the line defined by $0.5x_1 + \beta x_2 = \alpha$ crosses the x_1 and x_2 axes in the points $(2\alpha, 0)$ and $(0, \alpha/\beta)$ respectively (note that 2α and $\frac{\alpha}{\beta}$ are always between 0 and 20, so for all combinations of α and β the shape of the feasible set is similar): 2



Now we consider lines of constant cost (the term $-\beta$ can be dropped): $\alpha x_1 - x_2 = c$. This line crosses the x_1 and x_2 axes in the points $(c/\alpha, 0)$ and $(0, -c)$ respectively. In the following plot some of these lines are plotted. Also here it holds that for all values of α we get a similar shape (the only difference is the slope α of the lines, but these slopes are always larger than or equal than 1 (i.e., the angle of the constant-cost lines is always larger than or equal to 45°).



So we conclude that the cost is minimized in the vertex point v if $\alpha > 1$ and that the cost is minimal in any point on the line between v and (0.20) if $\alpha = 1$: 3

Note: the coordinates of the point v are $\left(\dfrac{\alpha - 20\beta}{\beta + 0.5}, \dfrac{\alpha + 10}{\beta + 0.5}\right)$.

1 (c) The optimum is indeed a global optimum as the problem is linear and thus convex and as for a convex problem a local optimum is also global.
Alternatively, you may argue that for a linear programming problem the global optimum can always be found in a vertex and that we have determined the vertex for which the cost function was minimal.

- **Question 2.b**

2 (a) First of all we transfer the maximization problem into a minimization problem:
$\min_{(x,y)\in\mathbb{R}^2} -\alpha-(-1)^\alpha(x^2+2y^2-xy-\beta y)$
If α is odd, we get $-\alpha+x^2+2y^2-xy-\beta y=(x-0.5y)^2+1.75y^2-\beta y-\alpha$ as objective function, which is the sum of two squares and an affine function. So the objective function and thus also the problem are convex.
if α is even, we get $-\alpha-x^2-2y^2+xy+\beta y=-(x-0.5y)^2-1.75y^2+\beta y-\alpha$ as objective function, which is concave (due to the two negative squares). So then the problem is nonconvex.

10 (b) First we compute the gradient of the objective function[2]: 2

$$\nabla f(x,y)=\begin{cases}\begin{bmatrix}2x-y\\4y-x-\beta\end{bmatrix} & \text{if } \alpha \text{ is odd}\\ \begin{bmatrix}-2x+y\\-4y+x+\beta\end{bmatrix} & \text{if } \alpha \text{ is even}\end{cases}$$

In the steepest descent approach the search direction is the **negative** gradient: 2
So we evaluate the negative gradient in the point $(\alpha,0)$: 1

$$-\nabla f(\alpha,0)=\begin{cases}\begin{bmatrix}-2\alpha\\\alpha+\beta\end{bmatrix} & \text{if } \alpha \text{ is odd}\\ \begin{bmatrix}2\alpha\\-\alpha-\beta\end{bmatrix} & \text{if } \alpha \text{ is even}\end{cases}$$

Then we determine the line on which the search will be performed as $(x,y)=(\alpha,0)-s\nabla f(\alpha,0)$ where s is the step size: 1

$$\begin{cases}\begin{cases}x=\alpha-2\alpha s\\y=(\alpha+\beta)s\end{cases} & \text{if } \alpha \text{ is odd}\\ \begin{cases}x=\alpha+2\alpha s\\y=-(\alpha+\beta)s\end{cases} & \text{if } \alpha \text{ is even}\end{cases}$$

Next we fill out x and y in the objective function, and take the derivative w.r.t. s and put it equal to 0: 1
The resulting s value (if **positive**) yields the optimal step size $s_1=s^*$: 1
Using this value x_1 and y_1 can be computed: 2

[2] If a sign error is made here, marks can still be scored for: use of negative gradient [2], line search equation [1], and the determination of s^* [1], unless s^* is negative.