TTK4135 Optimization and Control
Final Exam - Spring 2010

Department of Engineering Cybernetics

1 Nelder-Mead method (30 %)

a) A derivative-free optimization algorithm uses the function value at a set of sample points to determine a new iterate, as opposed to using derivatives. This can for instance be useful if the optimization problem contains nondifferentiable functions, or if calculating derivatives is impractical. Some DFO algorithms calculate approximated derivatives, but this can not be regarded as a general-purpose strategy.

b) The centroid of the $n = 2$ best points x^2 and x^3 is

$$\overline{x} = \frac{1}{2}\begin{bmatrix} 0.5 + 0.7 \\ 0.8 + 0.5 \end{bmatrix} = \begin{bmatrix} 0.60 \\ 0.65 \end{bmatrix} \tag{1.1}$$

The reflection point is given by

$$g(-1) = \overline{x} - 1(x^1 - \overline{x}) = \begin{bmatrix} 0.50 \\ 0.50 \end{bmatrix} \tag{1.2}$$

The points $x^1,\ x^2,\ x^3$ are ordered so that $f(x^1) \leq f(x^2) \leq f(x^3)$. That is, from worst to best. Note that the algorithm in the textbook uses x^1 as the best point and x^{n+1} as the worst point.

c) If $f(x^{refl}) = 35$, the reflection point is better than the current best. The Nelder-Mead algorithm then computes the expansion point

$$g(-2) = \overline{x} - 2(x^1 - \overline{x}) = \begin{bmatrix} 0.40 \\ 0.35 \end{bmatrix} \tag{1.3}$$

If $f(g(-2)) > f(x^{refl})$, x^1 would be replaced by $g(-2)$. Else, x^1 would be replaced by $x^{refl} = g(-1)$

If $f(x^{refl}) = 25$, the reflection point is neither best nor worst. The Nelder-Mead algorithm then replaces the worst point x_1 with the reflection point, i.e.,

$$x^1 = x^{refl} = \begin{bmatrix} 0.50 \\ 0.50 \end{bmatrix} \tag{1.4}$$

If $f(x^{refl}) = 10$, the reflection point is is worse than x^2. Further, as $f(x^{refl}) = 10 \leq f(x^1) = 15$, the Nelder-Mead algorithm performs inside contraction:

$$f_{1/2} = f\left(g\left(t = \frac{1}{2}\right)\right), \quad g\left(\frac{1}{2}\right) = \overline{x} + \frac{1}{2}(x^1 - \overline{x}) = \begin{bmatrix} 0.650 \\ 0.725 \end{bmatrix} \tag{1.5}$$

If $f\left(g\left(t = \frac{1}{2}\right)\right) > f(x^1)$, x^1 would be replaced by $g\left(t = \frac{1}{2}\right)$.

d) Assuming that $f\left(g\left(t=\frac{1}{2}\right)\right)=18$, we have that

$$f\left(g\left(t=\frac{1}{2}\right)\right)>f(x^1) \tag{1.6}$$

meaning that the "innside" contraction was an improvement. Hence, the new triangle is defined by $g\left(t=\frac{1}{2}\right), x^2, x^3$ with $f\left(g\left(t=\frac{1}{2}\right)\right)<f(x^2)<f(x^3)$.

e) Given a large nonlinear optimization problem, reasons for choosing the Nelder-Mead algorithm over SQP can be that the derivatives can be difficult or expensive to compute, as well as that each iteration will be cheaper and one may find the optimum faster. Also, the Nelder-Mead algorithm requires less storage, and constraint handling is not supported (this can be dealt with in the implementation). Reasons for choosing SQP can be that the Nelder-Mead algorithm may fail to converge (stagnation), and that derivative-free methods tend to be less effective for large problems.

2 Optimality conditions and nonlinear problems (40 %)

a) In a general QP problem, $f(x)=\frac{1}{2}x^\top Gx+x^\top c$, and $c_i(x)=a_i^\top x-b_i,\ i\in\mathcal{E}\cup\mathcal{I}$. Here, G is a symmetric $n\times n$ matrix and c, x and $\{a_i\},\ i\in\mathcal{E}\cup\mathcal{I}$ are vectors in $\mathbb{R}^n$.

An example of a convex QP problem is

$$\begin{aligned} \min_x \quad & x_1^2+x_2^2+2x_1 \\ \text{s.t.} \quad & x_1+x_2-3=0 \\ & x1\geq 0 \\ & 2-x_2\geq 0 \end{aligned} \tag{2.1}$$

An example of a nonconvex QP problem is

$$\begin{aligned} \min_x \quad & -x_1^2+x_2^2+2x_1 \\ \text{s.t.} \quad & x_1+x_2-3=0 \\ & x1\geq 0 \\ & 2-x_2\geq 0 \end{aligned} \tag{2.2}$$

b) For (1) to be a convex problem, f has to be a convex function, and the feasible region has to be convex as well. The function f is a convex function if its domain S is a convex set and if for any two points x and y in S, the following property is satisfied:

$$f(\alpha x+(1-\alpha)y)\leq\alpha f(x)+(1-\alpha)f(y),\quad\forall\alpha\in[0,1] \tag{2.3}$$

The feasible region is convex if all equality constraint functions are linear, and all the inequality constraint functions are concave. A function c_i is concave if $-c_i$ is convex.

c) We are to show that when f is convex, any local minimizer is a global minimizer of f. This can be proved by contradiction. Let x^* be a local, but not global, minimizer of f. Hence, there is a feasible point z such that $f(z)<f(x^*)$. Consider the line segment

$$x=\lambda z+(1-\lambda)x^*,\quad\lambda\in(0,\ 1] \tag{2.4}$$

that joins z and x^*. As f is convex,

$$f(x)=f(\lambda z+(1-\lambda)x^*)\leq\lambda f(z)+(1-\lambda)f(x^*)<f(x^*) \tag{2.5}$$

Since any neighborhood $\mathcal{N}$ of x^* will contain a piece of the line segment (2.4), there has to be points $x \in \mathcal{N}$ where (2.5) is satisfied. This contradicts x^* being a local, but not global, minimizer of f. Hence, x^* must be a global minimizer.

d) Consider the optimization problem

$$\min\ x_1 + \sqrt{3}x_2 \quad \text{s.t.} \quad x_1^2 + x_2^2 - 1 = 0 \tag{2.6}$$

The Lagrangean function for the problem is

$$\mathcal{L}(x, \lambda) = x_1 + \sqrt{3}x_2 - \lambda_1(x_1^2 + x_2^2 - 1) \tag{2.7}$$

The KKT conditions for the problem are

$$\nabla_x \mathcal{L}(x, \lambda) = \begin{bmatrix} 1 - 2\lambda_1 x_1 \\ \sqrt{3} - 2\lambda_1 x_2 \end{bmatrix} = 0 \tag{2.8a}$$

$$c_1(x) = x_1^2 + x_2^2 - 1 = 0 \tag{2.8b}$$

$$\lambda_1 c_1(x) = \lambda_1(x_1^2 + x_2^2 - 1) = 0 \tag{2.8c}$$

Second-order sufficient conditions for the problem are

$$w^\top \nabla_{xx}\mathcal{L}(x, \lambda) w = w^\top \begin{bmatrix} -2\lambda_1 & 0 \\ 0 & -2\lambda_1 \end{bmatrix} w > 0, \quad \text{for all } w \in F_2(\lambda),\ w \neq 0 \tag{2.9}$$

For $x = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^\top$, equation (2.8a) gives

$$\nabla_x \mathcal{L}(x, \lambda) = \begin{bmatrix} 1 - \lambda_1 \\ \sqrt{3} - \sqrt{3}\lambda_1 \end{bmatrix} \tag{2.10}$$

which is zero for $\lambda_1 = 1$. Since

$$c_1(x) = \left(\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2 - 1 = 0 \tag{2.11}$$

equations (2.8b) and (2.8c) are also satisfied. Hence, the KKT conditions (2.8) are satisfied for $x = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^\top$ with $\lambda_1 = 1$. The second-order sufficient condition (2.9) with $\lambda = 1$,

$$w^\top \nabla_{xx}\mathcal{L}(x, \lambda) w = w^\top \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} w = -w_1^2 - w_2^2 \not> 0, \quad \forall w \neq 0 \tag{2.12}$$

is not satisfied.

For $x = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)^\top$, equation (2.8a) gives

$$\nabla_x \mathcal{L}(x, \lambda) = \begin{bmatrix} 1 + \lambda_1 \\ \sqrt{3} + \sqrt{3}\lambda_1 \end{bmatrix} \tag{2.13}$$

which is zero for $\lambda_1 = -1$. Since

$$c_1(x) = \left(-\frac{1}{2}\right)^2 + \left(-\frac{\sqrt{3}}{2}\right)^2 - 1 = 0 \tag{2.14}$$

equations (2.8b) and (2.8c) are also satisfied. Hence, the KKT conditions (2.8) are satisfied for $x = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)^\top$ with $\lambda_1 = -1$. The second-order sufficient condition (2.9) with $\lambda_1 = -1$,

$$w^\top \nabla_{xx}\mathcal{L}(x,\lambda)w = w^\top \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} w = w_1^2 + w_2^2 > 0, \quad \forall w \neq 0 \tag{2.15}$$

is satisfied. Hence, $x = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)^\top$ is (at least) a local minimizer.

e) A suitable merit function ϕ_1 for problem (2.6) is

$$\phi_1(x;\mu) = f(x) + \mu_k \|c(x)\|_1 = x_1 + \sqrt{3}x_2 + \mu_k \|x_1^2 + x_2^2 - 1\|_1 \tag{2.16}$$

The purpose of the penalty parameter μ_k is to penalize constraint violation, and its value determines the weight that we assign to constraint satisfaction relative to minimization of the objective. There are several strategies for determining suitable values for μ_k so that the merit function does not impede the impede the progress of an optimization algorithm.

In SQP, the direction p_k is a descent direction for the merit function ϕ_1 provided that $p_k \neq 0$, $\nabla^2_{xx}\mathcal{L}$ is positive definite and $\mu_k > \|\lambda_{k+1}\|_\infty$. This means that provided a good strategy for updating μ_k, the merit function will generally decrease from one iteration to the next. However, one way of avoiding the Maratos effect involves allowing the merit function to increase on certain iterations (a nonmonotone approach).

The objective function f does not necessarily decrease from one iteration to the next. In particular, when starting infeasible, it is not uncommon that f increases when the algorithm enters (or is close to) the feasible region.

f) In general, the sequence will probably not approach the feasible set immediately, but most likely take a step towards the feasible region that gives a large decrease in the objective. Also, as the SQP method linearizes the constraints of the original problem, it is not very likely that the step calculated from the first SQP iteration will lead to a feasible point.

3 Optimal control and MPC (30 %)

a) The given problem is an infinite horizon discrete LQR problem. The expression for K_i and the discrete algebraic Riccati equation are given in the appendix. As our time horizon is infinite, we need a constant gain matrix and require a stationary solution to the Riccati equation, i.e.,

$$R_{i+1} = R_i \;\forall i \tag{3.1}$$

This gives the algebraic Riccati equation

$$R = Q + A^\top R\left(I + BP^{-1}B^\top R\right)^{-1} A \tag{3.2}$$

with

$$R = R^\top \geq 0 \tag{3.3}$$

as there is no explicit weight at the end of the horizon in the objective. The gain matrix is found from

$$K = -P^{-1}B^\top R\left(I + BP^{-1}B^\top R\right)^{-1} A \tag{3.4}$$

Two conditions for stability (by which we mean asymptotic stability), are (1) the pair (A, B) stabilizable and (2) the pair (A, D) detectable, with $Q = D^\top D$.

A stable, discrete, closed-loop system

$$
\begin{aligned}
x_{i+1} &= Ax_i + Bu_i, \quad u_i = Kx_i, \ 0 \le i \le \infty \\
&= (A + BK)\, x_i
\end{aligned}
\tag{3.5}
$$

will have $|\lambda_i(A+BK)| < 1$ for all $1 \le i \le n_x$. That is, the absolute value of all eigenvalues is less than one.

b) A suitable optimization problem is

$$
\begin{aligned}
\min_{z} \quad & f_0(z) = \sum_{k=0}^{N-1} \left((x_k - x_k^{ref})^\top Q_k (x_k - x_k^{ref}) + u_k^\top P_k u_k \right) + (x_N - x_N^{ref})^\top S (x_N - x_N^{ref}) \\
\text{s.t.} \quad & x_{k+1} = g(x_k, u_k), \quad 0 \le k \le N-1 \\
& 0 \le u_k \le 1, \quad 0 \le k \le N-1 \\
& x_0 = \text{given}
\end{aligned}
\tag{3.6}
$$

where $Q_k = Q_k^\top = \text{diag}(0, 0, q_{3,k}),\ q_{3,k} > 0,\ P_k = P_k^\top > 0,\ S = S^\top > 0$, and $z^\top = [u_0, \ldots, u_{N-1}, x_0^\top, \ldots, x_N^\top]$. The system dimensions are given by $x_k \in \mathbb{R}^3$ and $u_k \in \mathbb{R}^1$.

c) To penalize change in the manipulated variable from one time-step to the next, the term

$$
(u_{k+1} - u_k)^\top P_k' (u_{k+1} - u_k) \tag{3.7}
$$

can be added to the objective function above, with $P_k' = P_k'^\top > 0$. It also common to include a constraint on the form

$$
\Delta U_{min} \le u_{k+1} - u_k \le \Delta U_{max} \tag{3.8}
$$

d) The general idea behind MPC is to solve an optimization problem (a QP for linear MPC) at each iteration k using the latest measurement y_k. (For linear MPC, the QP is similar to the one in solved in LQR, but bounds on inputs and outputs are also included as inequality constraints.) The solution to this optimization problem includes an optimal input sequence, but only the first element of this input sequence is used. The next input comes from the optimal input sequence obtained from the optimization problem solved at the next iteration.

Some reasons for the industrial success of MPC are (1) inherent multivariable control, (2) handling of constraints, both on inputs and states, (3) the possibility of operation closer to constraints, usually leading to a more profitable process, and (4) understandable and intuitive theory.

e) This can be achieved through adding the constraints

$$
\underline{x}_3 - \epsilon_k \le x_{3,k} \le \underline{x}_3 + \epsilon_k \tag{3.9a}
$$

$$
\epsilon_k \ge 0 \tag{3.9b}
$$

and then add the following term to the objective function

$$
\rho \epsilon_k \tag{3.10}
$$

with $\rho > 0$, and let ϵ_k be an optimization variable. This is an exact penalty method, provided that ρ is large enough. It is also possible to use a quadratic formulation in the objective, but this does not lead to an exact penalty method.