TTK4135 – Optimalization and Control

Sondre Bø Kongsgård
sondrebk@stud.ntnu.no

Contents

1 Definitions

Convex programming Used to describe a special case of the general constrained optimization problem in which

- The objective function is convex
- The equality constraint functions $c_i(\cdot),\ i \in \mathcal{E}$ are linear, and
- The inequality constraint functions$c_i(\cdot),\ i \in \mathcal{I}$ are concave.

Convex function 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:

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

Active set The active set $\mathcal{A}(x)$ at any feasible x consists of the equality constraint indices from $\mathcal{E}$ together with the indices of the inequality constraints i for which $c_i(x) = 0$; that is

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} | c_i(x) = 0\} \quad (2)$$

LP:

$$\mathcal{A}(x) = \{i \in \mathcal{E} | a_i^T x = b_i\} \cup \{i \in \mathcal{I} | x_i = 0\} \quad (3)$$

QP:

$$\mathcal{A}(x) = \{i \in \mathcal{E} \cup \mathcal{I} | a_i^T x = b_i\} \quad (4)$$

Basic feasible point A vector x is a basic feasible point if it is feasible and if there exists a subset $\mathcal{B}$ of the index set $\{1, 2, \ldots, n\}$ such that

- $\mathcal{B}$ contains exactly m indices
- $i \notin \mathcal{B} \quad \Rightarrow \quad x_i = 0$ (that is, the bound $x_i \geq 0$ can be inactive only if $i \in \mathcal{B}$)
- The $m \times m$ matrix $\mathcal{B}$ defined by

  $$B = [A_i]_{i \in \mathcal{B}} \quad (5)$$

  is nonsingular, where A_i is the ith column of A

Degeneracy A basis $\mathcal{B}$ is said to be degenerate if $x_i = 0$ for some $i \in \mathcal{B}$, where x is the basic feasible solution corresponding to $\mathcal{B}$. A linear program is said to be degenerate if it has at least one degenerate basis.

LICQ (Linear Independence Constraint Qualification) Given the point x and the active set $\mathcal{A}(x)$, we say that the LICQ holds if the set of active constraint gradients $\nabla c_i(x),\ i \in \mathcal{A}(x)$ is linearly independent.

Strict complementarity Given a local solution x^* and a vector λ^* satisfying the KKT conditions, we say that the strict complementarity condition holds if excactly one of λ_i^* and $c_i(x^*)$ is zero for each index $i \in \mathcal{I}$. In other words, we have that $\lambda_i^* > 0$ for each $i \in \mathcal{I} \cap \mathcal{A}(x^*)$.

Search directions Several approaches to line search directions can be used:

Method	Formula
Steepest descent	$p_k = -\nabla f_k$
Newton direction	$p_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k$
Quasi-Newton direction	$p_k = -B_k^{-1} \nabla f_k$
	For updating B_k
SR1(Symmetric-rank-one)	
BFGS	

The Wolfe conditions Are used in line-search methods to decide if the decrease in the objective function is sufficient. The first one (*sufficient decrease condition*):

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^\top p_k \quad (6)$$

There is also a second Wolfe condition (the *curvature condition*: $\nabla f(x_k + \alpha_k p_k)^\top p_k \geq c_2 \nabla f_k^\top p_k$), but this condition can be dispensed if backtracking is used (find an acceptable step length by reducing α_k a finite number of trials – it will eventually be small enough for eq. (6) to hold).

Merit functions SQP methods often use merit functions to decide whether a trail step should be accepted. In line search methods, the merit function controls the size of the step.

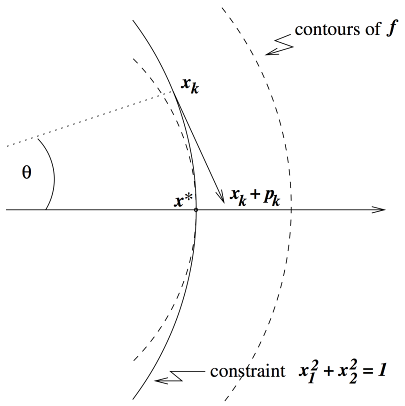
$$l_1: \quad \phi_1(x;\mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)| + \mu \sum_{i \in \mathcal{I}} [c_i(x)]^- \quad (7)$$

Notation: $[z]^- = \max\{0, -z\}$

The positive scalar μ is the penalty parameter, which determines the weight that we assigned to constraint satisfaction relative to minimization of the objective.

Exact merit function: A merit function $\phi(x;\mu)$ is exact if there is a positive scalar μ^* such that for any $\mu > \mu^*$, any local solution of the nonlinear programming problem is a local minimizer of $\phi(x;\mu)$.

Maratos effect The phenomenon where a merit function prevents rapid convergence because steps that make good progress toward a solution are rejected.



Figur 1: Maratos Effect

Model predictive control (MPC) A form of control in which the current control action is obtained by solving, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant.

- Advantages
  - Inherent multivariable control
  - Constraint handling, both on inputs and states
  - The possibility of operation closer to constraints, usually leading to a more profitable process
  - Understandable and intuitive theory
- Challenges
  - Update model
  - Tuned basic controllers
  - Calibrated sensors

Full-space VS Reduced-space formulation

- In full-space optimization, we include all inputs and all states in our objective function. The number of optimization variables is #steps · (#states + #inputs).
- In reduced-space optimization, we remove the states from the objective function by replacing them using the model ($x_{t+1} = A_t x_t + B_t u_t$). The number of optimization variables is #steps · #inputs.

	Pros	**Cons**
Full-space	Often sparsity in matrices	Many variables
Reduced-space	Less variables	Normally dense matrices

Stabilizability A system (A, B) is stabilizable if all the uncontrollable modes are asymptotically stable.

Controllability A system (A, B) is controllable if for any initial state x_0 and any final state x_N, there exists a finite number of inputs $u_0, \dots, u_{N-1}$ to transfer x_0 to x_N.

Detectability A system (A, D) is detectable if all the unobservable modes are asymptotically stable.

Observability A system (A, D) is observable if the state x_N can be determined from the system model, its inputs and outputs for a finite number of steps.

Linear quadratic Gaussian control (LQG) The combination of an LQ controller and a Kalman filter.

Dual problem The dual objective function $q : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$q(\lambda) \stackrel{\text{def}}{=} \operatorname*{Inf}_x \mathcal{L}(x, \lambda) \tag{8}$$

The dual problem is defined as follows:

$$\max_{\lambda \in \mathbb{R}^n} q(\lambda) \quad \text{subject to } \lambda \geq 0 \tag{9}$$

2 Background Material

Matrix factorizations

Cholesky	**LU**	**QR**	**Symmetric indefinite**
$A = LL^\top$	$PA = LU$	$A^T P = QR$	$PAP^\top = LBL^\top$
	$Ax = b$ $L\underbrace{(Ux)}_{y} = b$ Triangular forward substitution $Ly = b$ for y Triangular backward substitution $Ux = y$ for x		
$A \in \mathbb{R}^{nxn}$ $A = A^T$ (Symmetric) $A \succ 0$	$A \in \mathbb{R}^{nxn}$	$A \in \mathbb{R}^{mxn}$ (only real A)	$A = A^T$ A can be indefinite

Matrix calculus

Derivative

$$\begin{aligned} \nabla(c^\top \mathbf{x}) &= c \\ \nabla(\mathbf{x}^\top c) &= c \end{aligned} \tag{10}$$

$$\nabla\left(\frac{1}{2}\mathbf{x}^\top G\mathbf{x}\right) = \frac{1}{2}G\mathbf{x} + \frac{1}{2}G^\top \mathbf{x} \tag{11}$$

Gradient

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial \mathbf{x}}^\top\right] = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{12}$$

Hessian

$$\nabla_{xx} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1^2} & \frac{\partial f}{\partial x_1 \partial x_2} & \cdots \\ \frac{\partial f}{\partial x_2 \partial x_1} & \frac{\partial f}{\partial x_2^2} & \\ \vdots & & \ddots \end{bmatrix} \tag{13}$$

Jacobian

$$Jf(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \\ \vdots & & \ddots \end{bmatrix} \tag{14}$$

Elements of Analysis

Lipschitz continuous

$$||f(x_1) - f(x_0)|| \leq L||x_1 - x_0||, \quad \forall\, x_0, x_1 \in \mathcal{N} \tag{15}$$

Mean value theorem

$$f(x + p) = f(x) + \nabla f(x + \alpha p)^\top p \tag{16}$$

3 Tips & Tricks

LP Remember to read the task closely, and see whether it asks for the given resources to be *fully utilized*. If not, then introduce slack variables in your constraints.

Prediction horizon Prediction horizon Generally, the prediction horizon N should be chosen such that

$$\text{dominant dynamics} < N < \text{control interval} \tag{17}$$

Slack variables We normally have tighter bounds on the state variables than reality since some of the constraints are hard and must always be satisfied. Therefore, the state constraints (A.9d) may be violated for all the time. In this case a feasible point may not exist and a control input may not be available.

Fix: (soften the constraints by using slack variables)

$$\min_{z \in \mathbb{R}^n} f(z) = \cdots + p^\top \epsilon + \frac{1}{2}\epsilon^\top S\epsilon \tag{18}$$

$$\begin{gathered} x^{\text{low}} < x_t < x^{\text{high}} \\ \Downarrow \\ x^{\text{low}} - \epsilon_t < x_t < x^{\text{high}} + \epsilon_t \end{gathered} \tag{19}$$

Linearization of a constraint Use the following formula to linearize a constraint (eg. for use in a QP problem):

$$\nabla c_i(x_i)^\top p + c_1(x_i) \tag{20}$$

Limit on control input The constraint $-\Delta u^{\text{high}} \leq \Delta u_t \leq \Delta u^{\text{high}}$ is a limit on change of control input and/or limit wear and tear of control input.

Rosenbrock function A nonconvex function on the form

$$\begin{aligned} f(x_1, x_2) &= (a - x_1)^2 + b(x_2 - x_1^2)^2 \\ a, b &\geq 0 \end{aligned} \tag{21}$$

State feedback infinite LQ control Let system (A, B) be stabilizable and (A, D) detectable. D is defined by $Q = D^\top D$. Then the closed loop system given by the optimal solution is asymptotically stable.

What to do if the Hessian is positive semi-definite?
(When using the SQP algorithm) The KKT matrix will be singular. Three options to account for this:

- Using Quasi-Newton method
- Adding a multiple of the identity matrix
- Modified Cholesky factorization

4 Algorithms

Active set methods Maintain estimates of the active and inactive index sets that are updated at each step of the algorithm. Examples: the simplex method and active-set method.

Simplex

One Step of Simplex()

```
Given $\mathcal{B}, \mathcal{N}, x_B = B^{-1}b \geq, x_N = 0$;
Solve $B^\top \lambda = C_B for \lambda$,
Compute $s_N = c_N - N^\top \lambda$; (* pricing *)
if $s_N \geq 0$
        stop; (* optimal point found *)
Select $q \in \mathcal{N}$ with $s_q < 0$ as the entering index;
Solve $Bd = A_q$ for d;
if $d \leq 0$
        stop; (* problem is unbounded *)
Calculate $x_q^+ = \min_{i|d_i>0}(x_B)_i/d_i$, and use $p$ to
    denote the minimizing $i$;
Update $x_B^+ = x_B - dx_q^+, x_N^+ = (0, \ldots, 0, x_q^+, 0, \ldots, 0)^\top$;
Change $\mathcal{B}$ by adding $q$ and removing the basic
    variable corresponding to column $p$ of $B$
```

Nelder-Mead method A popular derivative-free optimization (DFO) method.

The centroid of the best n points is denoted by

$$\bar{x} = \sum_{i=1}^{n} x_i \tag{22}$$

The reflection point is given by

$$\bar{x}(t) = \bar{x} + t(x_{n+1} - \bar{x}) \tag{23}$$

One Step of Neldear-Mead Simplex()

```
Compute the reflection point $\bar{x}(-1)$
    and evaluate $f_{-1} = f(\bar{x}(-1))$;
if $f(x_1) < f_{-1} < f(x_n)$
        (* reflected point is neither best
        nor worst in the new simplex *)
        replace $x_{n+1}$ by $\bar{x}(-1)$ and go to next iteration;
Ah, see p. 238 - 239 for the rest of the algorithm
```

MPC-based controllers MPC merges feedback control with dynamic optimization.

Name	Optimization problem solved	Note
MPC	Non-specified (any)	
Linear MPC	QP with linear equality constraints and inequalities on x, u and Δu	
LQ	Convex QP with only linear equality constraints	*Finite-horizon:* $u_t = K_t x_t$ *Infinite-horizon:* $u_t = K x_t$
LQGC	Same as LQ	Combined with a Kalman filter
NMPC	Non-linear optimization problems	

State feedback MPC procedure()

```
1 for $T = 0, 1, 2, \ldots$ do
2         Get the current state $x_t$
3         Solve a dynamic optimization problem on
          the prediction horizon from $t$ to $t + N$
          with $x_t$ as the initial condition
4         Apply the first control move $u_t$ from
          the solution above
```

For an output feedback MPC procedure, line 2 in the algorithm above changes to:

2 Compute an estimate of the current state $\hat{x}_t$ based on the measured data up until time t

Methods for optimizing dynamic systems Two important classes:

- *Quasi dynamic optimization*: Optimize a dynamic system by repetitive optimization on a static model.
  - Advantages: Smaller problems and less complex formulations.
  - Disadvantages: It cannot handle a system with significant dynamics.
- *Dynamic optimization*: Optimize on a dynamic model. In this case the solution will be a function of time, i.e. all decision variables will be functions of time.

5 Theorem

When f is convex, any local minimizer is a global minimizer of f 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

$$x = \lambda z + (1-\lambda)x^*,\ \lambda \in (0, 1] \tag{24}$$

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

$$f(x) = f(\lambda z + (1-\lambda)x^*) \leq \lambda f(z) + (1-\lambda)f(x^*) < f(x^*) \tag{25}$$

Since any neighborhood $\mathcal{N}$ of x^* will contain a piece of the line segment eq. (24), there has to be points $x \in \mathcal{N}$ where eq. (25) is satisfied. This contradicts x^* being a local, but not global, minimizer of f. Hence, x^* must be a global minimizer.