OptCtrl Cheat Sh33t

1 Basics

Table of gradients and derivatives

$$\begin{aligned}
\nabla_x \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{A}^T \\
\nabla_x \boldsymbol{x}^T\boldsymbol{A} &= \boldsymbol{A} \\
\nabla_x \boldsymbol{x}^T\boldsymbol{x} &= 2\boldsymbol{x} \\
\nabla_x \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^T\boldsymbol{x} \\
\nabla_x \boldsymbol{b}^T\boldsymbol{x} &= \boldsymbol{b} \\
\nabla_x (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) &= \boldsymbol{A}^T \\
\nabla_x (\boldsymbol{x} + \boldsymbol{y})^T\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{y}) &= (\boldsymbol{A} + \boldsymbol{A}^T)(\boldsymbol{x} + \boldsymbol{y})
\end{aligned}$$

Formulations

`Linear program`

$\min \boldsymbol{c}^T\boldsymbol{x}$, s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$

`Quadratic program`

$\min \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$, s.t. $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{E}\boldsymbol{x} = \boldsymbol{d}$

2 Static unconstrained optimization

`First order necessary optimality condition`

$(\nabla f)(\boldsymbol{x}^*) = 0$

`Second order necessary optimality condition`

$(\nabla^2 f)(\boldsymbol{x}^*) \geq 0$

`Second order sufficient optimality condition`

If $(\nabla f)(\boldsymbol{x}^*) = 0$ and $(\nabla^2 f)(\boldsymbol{x}^*) > 0$, then $\boldsymbol{x}^*$ is a strict local minimizer.

`Theorem 2.4`

If $f(\boldsymbol{x})$ is convex, any local minimizer $\boldsymbol{x}^*$ is a global minimizer. If $f(\boldsymbol{x}) \in C^1(X_{ad})$, then any stationary point is a global minimizer.

`Convexity`

If $f(\cdot)$ is convex any local miminizer is a global minimizer.
If $f(\cdot)$ is convex any strict local minim. is a global unique.
If $f(\cdot)$ is strictly convex any local minim. is a global unique.

2.1 Numerical minimization algorithms

2.1.1 Schematic line search method

1. Input start value $\boldsymbol{x}_0$ and stopping criteria ϵ.

2. Initialize $k = 0$.

3. repeat

   (a) Search direction $\boldsymbol{s}_k$

   (b) Step length α_k

   (c) $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k$

   (d) $k = k + 1$

4. until $||f(\boldsymbol{x_{k+1}})|| \leq \epsilon$

2.2 Determination of step length

2.2.1 Armijo conditions

Usually $\epsilon_0 \leq 0.01$, $\epsilon_1 > 1$:

$$\text{upper bound: } f(\boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k) \leq f(\boldsymbol{x}_k) + \epsilon_0 \alpha_k \boldsymbol{s}_k^T (\nabla f)(\boldsymbol{x}_k)$$
$$\text{lower bound: } f(\boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k) \geq f(\boldsymbol{x}_k) + \epsilon_0 \epsilon_1 \alpha_k \boldsymbol{s}_k^T (\nabla f)(\boldsymbol{x}_k)$$

1. If upper bound satisfied, increase α_k by ϵ_1 until upper bound violated.

2. If lower bound violated, decrease α_k by ϵ_1 until lower bound satisfied.

3. Assign α_k as step length

2.2.2 Wolfe conditions

$\epsilon_1 \in (0, 1)$, $\epsilon_2 \in (e_1, 1)$, typically $\epsilon_2 = 0.9$ (Newton) or $\epsilon_2 = 0.1$ (Conj. grad.). (Strong Wolfe):

$$\text{upper bound: } f(\boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k) \leq f(\boldsymbol{x}_k) + \epsilon_1 \alpha_k \boldsymbol{s}_k^T (\nabla f)(\boldsymbol{x}_k)$$
$$\text{lower bound: } |\boldsymbol{s}_k^T (\nabla f)(\boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k)| \geq \leq \epsilon_2 |\boldsymbol{s}_k^T (\nabla f)(\boldsymbol{x}_k)|$$

2.2.3 Backtracking

\+ Newton
\- quasi-Newton, conj. Grad.

$\alpha_k^0 > 0$ ($\alpha_k^0 = 1$ Newton), $\rho \in (0, 1)$, $\epsilon_0 \in (0, 1)$
Repeat:
$\alpha_k \leftarrow \rho \alpha_k$
until $f(\boldsymbol{x_k} + \alpha_k s_k) \leq f(\boldsymbol{x_k}) + \epsilon_0 \alpha_0 s_k^T (\nabla f)(\boldsymbol{x_k})$

2.2.4 Others

Nested intervals
\+ easily implementable, numerically robust
\- larger number of iteration steps

Quadratic interpolation
\+ very efficient

2.3 Numerical minimization methods

2.3.1 Line search methods

`Steepest descent`
$\boldsymbol{s}_k = -(\nabla f)(\boldsymbol{x}_k)$

+ Simple / low computation (no explicit evaluation of Hessian needed)
+ Good convergence for starting values not close to local minimizer
- Slow convergence depending on conditioning (linear convergence)

`Conjugated gradient`
+ well understood convergence for linear and quadratic probs.
- not for nonlinear

$\boldsymbol{s}_k = -(\nabla f)(\boldsymbol{x}_k) + \beta_k \boldsymbol{s}_{k-1}, k \geq 1$
$\boldsymbol{s}_0 = -(\nabla f)(\boldsymbol{x}_0)$
Fletcher-Reeves:
$\beta_k^{FR} = \frac{(\nabla f)^T(\boldsymbol{x}_k)(\nabla f)(\boldsymbol{x}_k)}{(\nabla f)^T(\boldsymbol{x}_{k-1})(\nabla f)(\boldsymbol{x}_{k-1})}$

`Newton method`
$\boldsymbol{s}_k = -(\nabla^2 f)^{-1}(\boldsymbol{x}_k)(\nabla f)(\boldsymbol{x}_k)$

+ quadratic convergence if Hessian is positive definite
- loss of pos. def. of the Hessian if $\boldsymbol{x}$ not sufficiently close to $\boldsymbol{x}^*$
- requires evaluation and inversion of Hessian

`Quasi Newton method`
$\boldsymbol{s}_k = -B_k^{-1}(\nabla f)(\boldsymbol{x}_k)$
$\boldsymbol{p}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \qquad \boldsymbol{q}_k = (\nabla f)(\boldsymbol{x}_{k+1}) - (\nabla f)(\boldsymbol{x}_k)$
$H_{k+1} = H_k + \frac{(p_k - H_k q_k)(p_k - H_k q_k)^T}{q_k^T(p_k - H_k q_k)}$

H_{k+1} is inverting B_{k+1}.

BFGS with:
$H_{k+1} = \left(E - \frac{p_k q_k^T}{q_k^T p_k}\right) H \left(E - \frac{q_k p_k^T}{q_k^T p_k}\right) + \frac{p_k p_k^T}{q_k^T p_k}$

+ less computationally expensive than Newton
- more iterations

2.3.2 Trust region methods

Define region around current search iterate to trust the quadratic model being an adequate approximation of the objective function.

$m(\boldsymbol{s}_k) = f(x_k) + s_k^T(\nabla f)(x_k) + \frac{1}{2} s_k^T B_k s_k \approx f(\boldsymbol{x}_k + \boldsymbol{s}_k)$

Solution s_k^* is minimizer of $m(s_k)$ in radius Δ_k. Step length and search direction are calculated simultaneously. Depending on the ratio of actual reduction and predicted reduction the region is shrunk if ratio is $< 0 \ll 1$. For ratios ≈ 1 region may be expanded.

2.3.3 Direct search methods

Derivative free methods, so no explicit knowledge of gradient or Hessian of $f(\cdot)$ is needed to find minimum.

`Simplex algorithm of Nelder and Mead`
Build a simplex, which is a convex hull, spanned by $n+1$ points. Take worst vertex (highest value) and replace with new point which lies on the line spanned by the highest value through the centroid $\bar{x}$ of the simplex.
$x_k^{ref} = \bar{x}_k + \alpha(\bar{x}_k - x_{k,max})$

3 Static constrained optimization

3.1 Equality constrained

`Regular point`
$\boldsymbol{x}^*$ satisfying $\boldsymbol{g}(\boldsymbol{x}^*) = 0$ is a regular point, if $\text{rank}(\nabla \boldsymbol{g})(\boldsymbol{x}^*) = \text{rank}[(\nabla g_1), \cdots, (\nabla g_p)](\boldsymbol{x}^*) = p$

`Tangent space`
$\mathcal{T}_{x^*}\mathcal{M} = \{\boldsymbol{d} \in R^n : (\nabla g_j)^T(\boldsymbol{x}^*)\boldsymbol{d} = 0, j = 1, ..., p\}$

`Lemma 3.1`
All $\boldsymbol{d}$ fulfilling $(\nabla \boldsymbol{g})^T(\boldsymbol{x}^*)\boldsymbol{d} = 0$ must also satisfy $(\nabla \boldsymbol{f})^T(\boldsymbol{x}^*)\boldsymbol{d} = 0$

`First order necessary optimality condition`
$(\nabla f)(\boldsymbol{x}^*) + (\nabla g)(\boldsymbol{x}^*)\boldsymbol{\lambda}^* = \boldsymbol{0}$ or
$\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{0}$ and $\nabla_{\boldsymbol{\lambda}} l(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{0}$

`Langrangian`
$l(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = f(\boldsymbol{x}^*) + (\boldsymbol{\lambda}^*)^T \boldsymbol{g}(\boldsymbol{x})$

`Second order necessary optimality condition`
$\boldsymbol{d}^T(\nabla^2 l)(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)\boldsymbol{d} \geq 0$

`Second order sufficient optimality condition`
$\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{0}$, $\nabla_{\boldsymbol{\lambda}} l(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0$ and $\boldsymbol{d}^T(\nabla^2 l)(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)\boldsymbol{d} > 0$

`Sensitivity (eq.constr.) - UNFINISHED(??)`
$\frac{\partial}{\partial \boldsymbol{c}} f(\boldsymbol{x}(\boldsymbol{c}))|_{\boldsymbol{c}=0} = -(\boldsymbol{\lambda}^*)^T$

3.2 Inequality constrainted

`Regular point (LICQ)`
$\boldsymbol{x}^*$ satisfying $\boldsymbol{g}(\boldsymbol{x}^*) = 0$ and $\boldsymbol{h}(\boldsymbol{x}^*) \leq 0$ is a regular point, if the gradient vectors $(\nabla \boldsymbol{g})$ and $(\nabla \boldsymbol{h})$ (only active ieq.) are linearly independent.

`KKT first order necessary optimality condition`
$(\nabla f)(\boldsymbol{x}^*) + (\nabla g)(\boldsymbol{x}^*)\boldsymbol{\lambda}^* + (\nabla h)(\boldsymbol{x}^*)\boldsymbol{\mu}^* = \boldsymbol{0}$
$\boldsymbol{h}^T(\boldsymbol{x}^*)\boldsymbol{\mu}^* = 0$
$\boldsymbol{\mu}^* \geq \boldsymbol{0}$

`Complementary slackness condition`

$h_l(\boldsymbol{x}^*) < 0 \rightarrow \mu_l^* = 0$ (inactive)
$h_l(\boldsymbol{x}^*) = 0 \rightarrow \mu_l^* > 0$ (active)

`KKT second order necessary optimality condition`

$\boldsymbol{d}^T(\nabla^2 l)(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d}(>) \geq 0$ (sufficient) with

$\boldsymbol{d} \in \mathcal{T}_{x^*}\mathcal{M} = \{\boldsymbol{d} \in R^n : (\nabla g_j)^T(\boldsymbol{x}^*)\boldsymbol{d} = 0,\ j = 1, \cdots, p,\ (\nabla h_l)^T(\boldsymbol{x}^*)\boldsymbol{d} = 0, \forall l \in \mathcal{A}_{ieq}(\boldsymbol{x})\}$

and the Lagrangian

$l(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = f(\boldsymbol{x}^*) + (\boldsymbol{\lambda}^*)^T\boldsymbol{g}(\boldsymbol{x}) + (\boldsymbol{\mu}^*)^T\boldsymbol{h}(\boldsymbol{x}^*)$

`Theorem 3.9`

The static constrained opt. problem is convex if $f(\boldsymbol{x})$ is convex on the admissible set and $\boldsymbol{g}(\boldsymbol{x})$ is linear and $\boldsymbol{h}(\boldsymbol{x})$ is convex.

`Sensitivity (ieq.constr.) - UNFINISHED(??)`

$\frac{\partial}{\partial \boldsymbol{c}} f(\boldsymbol{x}(\boldsymbol{c}, \boldsymbol{d}))|_{\boldsymbol{c}=0, \boldsymbol{d}=0} = -(\boldsymbol{\lambda}^*)^T$ and
$\frac{\partial}{\partial \boldsymbol{d}} f(\boldsymbol{x}(\boldsymbol{c}, \boldsymbol{d}))|_{\boldsymbol{c}=0, \boldsymbol{d}=0} = -(\boldsymbol{\mu}^*)^T$

4 Numerical optimization algorithms

4.1 Primal methods

Primal methods work on the orignal problem by searching a solution in the feasible space, i.e. active constraints. + Each point is feasible - no problem with premature termination
+ Convergence to at least a local minimum can be guaranteed.
+ Do not rely on e.g. convexity
- Require initilization phase to determine feasible starting point
- Computational issues may arise due to restriction on feasible space
- Problems with inequality constraints may let the algorithm fail

4.1.1 Active set methods

1. Determine working set from feasible starting point, solve opt. problem over the working surface
2. Add newly encountered ieq.constr. to the working set, but do not drop.
3. Determine Lagrange multipliers if minimum in $f(\boldsymbol{x})$ is found. If they are nonnegative, the solution is (locally) optimal.
4. Drop inequality constraints with negative Lagrange multipliers and restart with the new working set.

+ Often very effective for static constrained optimization (rare zigzagging)
- Each iterate has to be exact to the intermediate optimization problem to ensure the signs of Lagrange multipliers are correct

4.1.2 Gradient projection methods

Move along the projected $-(\nabla f)(\boldsymbol{x}_k)$ to a certain point on the tangent plane and then perpendicular to it, to reach a feasible point on the manifold $\mathcal{M}$.

- Complex implementation - Recomputation of projection matrix in each step
- Previously inactive inequality constraints may be violated when moving in direction of projected gradient (use interpolation - trial and error or relax constraints)

4.2 Penalty and barrier methods

Directly work in n-dimensional space of decision variables. + easy to implement but still ensure (slow) convergence.
+ Lagrange multipliers can be recovered

4.2.1 Penalty methods

Approximating constrained optimization problem by unconstrained optimization problem:

$\min f(\boldsymbol{x}) + cP(\boldsymbol{x})$, $c = const. > 0$, $P(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$ and $P(\boldsymbol{x}) = 0$ only if $\boldsymbol{x} \in X_{ad}$.

+ Superlinear convergence to feasible point...
- ... which may be far from optimal
- Ill-cond. or round-off errors may yield slow convergence or premature termination

4.2.2 Barrier methods (interior point)

Constrained opt. problem is transformed into (sequence of) unconstrained problem to create barriers which prevent the iterates to leave the X_{ad}.

+ Competetive algorithm
- Finding feasible starting point satisfying ieq. constraints may be difficult (nonlinear problems)
- Ill-conditioning or round-off errors (near X_{ad} boundary) of c_k may arise

4.3 Sequential quadratic programming

Solving a sequence of quadratic programs in Newton manner. From Newton method it is known that:
$(\nabla F)(x_k)r_k = -F(x_k)$, where $x_{k+1} = x_k + r_k$

This can be viewed as: $\min_{p_k} f(x_k) + (\nabla f)^T(x_k)p_k + \frac{1}{2}p_k^T L(x_k, \lambda_k)p_k$
s.t $(\nabla g)^T(x_k)p_k + g(x_k) = 0$
So $p_k = 0$ yields x^*, λ^* that satisfies KKT conditions of the minmization problem.
Active set methods can be used to solve SQP. Convergence can be ensured if start is close to the local minimizer or else the Hessian of the Langrangian can become indefinite, but however this can be overcome by using a (modified) BFGS method or global SQP by introducing step length $alpha_k$ to ensure convergence from remote points.

5 Dynamic optimization

5.1 Unconstrained

`Cost functional in Bolzano form`

$J(\boldsymbol{u}) = \underbrace{\varphi(t_e, \boldsymbol{x}(t_e))}_{\text{terminal constraint}} + \int_{t_0}^{t_e} l(t, \boldsymbol{x}(t), \boldsymbol{u}(t))dt$

`Dynamic system`

$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}),\ t \geq t_0,\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in R^n$

`Gateaux derivative (holds for any norm on X)`

$\delta J(\boldsymbol{x}, \boldsymbol{\xi}) = \lim_{\eta \to 0} \frac{J(\boldsymbol{x}+\eta\boldsymbol{\xi}) - J(\boldsymbol{x})}{\eta} = \frac{\partial}{\partial \eta} J(\boldsymbol{x} + \eta\boldsymbol{\xi})|_{\eta=0}$

`First order necessary optimality condition`

$\delta J(\boldsymbol{x}^*, \boldsymbol{\xi}) = 0$

`Euler-Lagrange equations`

Given a functional in Lagrange form and $\boldsymbol{x}^*(t)$ is a local minimizer, then $\boldsymbol{x}^*(t)$ fulfills for all $t \in [t_0, t_e]$:

$\frac{\partial}{\partial t}(\nabla_{\dot{\boldsymbol{x}}} l)(t, \boldsymbol{x}^*(t), \dot{\boldsymbol{x}}^*(t)) - (\nabla_{\boldsymbol{x}} l)(t, \boldsymbol{x}^*(t), \dot{\boldsymbol{x}}^*(t)) = \boldsymbol{0}$

(i) If $l = l(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))$ (independent of t) then

$H(\boldsymbol{x}, \dot{\boldsymbol{x}}) = (\nabla_{\dot{x}} l)(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} - l(\boldsymbol{x}, \dot{\boldsymbol{x}}) = c$

is an invariant of the E-L eq. and remains const. along $\boldsymbol{x}^*(t)$.

(ii) If $l = l(t, \dot{\boldsymbol{x}}(t))$ (independent of $\boldsymbol{x}(t)$) then

$\frac{\partial}{\partial t}(\nabla_{\dot{\boldsymbol{x}}} l)^T(t, \dot{\boldsymbol{x}}) = \boldsymbol{0}$

is an invariant of the E-L eq.

`Legendre condition`

$\boldsymbol{x}^*(t)$ is a local minimizer if for all $t \in [t_0, t_e]$:

$(\nabla^2_{\dot{\boldsymbol{x}}} l)(t, \boldsymbol{x}^*(t), \dot{\boldsymbol{x}}^*(t)) \geq 0$

`Euler-Lagrange eq. (free end-point) - Transvers. cond.`

$[(\nabla_{\dot{\boldsymbol{x}}} l)(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) + (\nabla_{\boldsymbol{x}} \varphi)(t, \boldsymbol{x}(t))]_{t=t_e^*, \boldsymbol{x}=\boldsymbol{x}^*} = \boldsymbol{0}$

$[l(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) - (\dot{\boldsymbol{x}})^T(\nabla_{\dot{\boldsymbol{x}}} l)(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) + \frac{d}{dt}\varphi(t, \boldsymbol{x}(t))]_{t=t_e^*, \boldsymbol{x}=\boldsymbol{x}^*} = 0$

5.2 Constrained

5.2.1 Equality constrained

`Existence of Lagrange multipliers (single eq. constr.)`

$\delta J(\boldsymbol{x}^*, \boldsymbol{\xi}) + \lambda \delta G(\boldsymbol{x}^*, \boldsymbol{\xi}) = 0, \quad \forall \boldsymbol{\xi} \in X$

`Existence of Lagrange multipliers (multiple eq. constr.)`

$\delta J(\boldsymbol{x}^*, \boldsymbol{\xi}) + [\delta G_1(\boldsymbol{x}^*, \boldsymbol{\xi}) \cdots \delta G_p(\boldsymbol{x}^*, \boldsymbol{\xi})]\boldsymbol{\lambda} = 0, \quad \forall \boldsymbol{\xi} \in X$

5.2.2 Inequality constrained

`Existence of Lagrange multipliers (multiple ieq. constr.)`

$$\delta J(\boldsymbol{x}^*, \boldsymbol{\xi}) + [\delta G_1(\boldsymbol{x}^*, \boldsymbol{\xi}) \cdots \delta G_p(\boldsymbol{x}^*, \boldsymbol{\xi})]\boldsymbol{\mu} = 0, \quad \forall \boldsymbol{\xi} \in X$$
$$(G_j(\boldsymbol{x}^*) - s_j)\mu_j = 0$$
$$\mu_j \geq 0$$

`Complementary slackness condition`
For constraints $G_j(\boldsymbol{x}^*) = s_j$,
(i) $(G_j(\boldsymbol{x}^*) - s_j) < 0$ implies $\mu_j = 0$
(ii) $(G_j(\boldsymbol{x}^*) - s_j) = 0$ implies $\mu_j > 0$

5.2.3 Isoperimetrically constrained

6 Optimal control

`Hamiltonian function`
$H(t, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = l(t, \boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$

6.1 Unconstrained

6.1.1 Fixed-time, free-endpoint

Two-point-BVP since initial state in $\boldsymbol{x}^*(t)$ and final state in the adjoint state $\boldsymbol{\lambda}^*(t)$ are given.

$\dot{\boldsymbol{x}}^* = (\nabla_\lambda H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*), \;\; \boldsymbol{x}^*(t_0) = \boldsymbol{x}_0$
$\dot{\boldsymbol{\lambda}}^* = -(\nabla_x H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*), \;\; \boldsymbol{\lambda}^*(t_e) = \boldsymbol{0}$
$\boldsymbol{0} = (\nabla_u H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*)$

$\boldsymbol{u}^*(t)$ must be a stationary point of the Hamiltonian, e.g. $H(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*) = const.$ for the triple to be a local minimizer of J.

Legendre condition:
$(\nabla_u^2 H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*) \geq 0$ (positive semi-definite)

With terminal cost:
$\boldsymbol{\lambda}^*(t_e) = (\nabla_x \varphi)(t_e, \boldsymbol{x}^*(t_e))$

6.1.2 Free-time, fixed-endpoint

Minimization problem in Bolzano form, with p equality constraints $G_k(t_e, \boldsymbol{u}) = \psi_k(t_e, \boldsymbol{x}(t_e)) = 0$:

$\dot{\boldsymbol{x}}^* = (\nabla_\lambda H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*), \;\; \boldsymbol{x}^*(t_0) = \boldsymbol{x}_0$
$\dot{\boldsymbol{\lambda}}^* = -(\nabla_x H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*), \;\; \boldsymbol{\lambda}^*(t_e^*) = (\nabla_{x_e} \phi)(t_e^*, \boldsymbol{x}^*(t_e^*), \boldsymbol{\mu}^*)$
$\boldsymbol{0} = (\nabla_u H)(t, \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*)$

with
$\phi(t_e, \boldsymbol{x}(t_e), \boldsymbol{\mu}) = \varphi(t_e, \boldsymbol{x}(t_e)) + \boldsymbol{\mu}^T \psi(t_e, \boldsymbol{x}(t_e))$

and transversality conditions $\psi(t_e^*, \boldsymbol{x}^*(t_e^*)) = [\psi_1(t_e^*, \boldsymbol{x}^*(t_e^*)) \cdots \psi_p(t_e^*, \boldsymbol{x}^*(t_e^*))]^T = \boldsymbol{0}$
$\frac{\partial}{\partial t_e} \phi(t_e^*, \boldsymbol{x}^*(t_e^*), \boldsymbol{\mu}^*) + H(t_e^*, \boldsymbol{x}^*(t_e^*), \boldsymbol{u}^*(t_e^*), \boldsymbol{\lambda}^*(t_e^*)) = 0$

with inequality constraints $G_k(t_e, \boldsymbol{u}) = \psi_k(t_e, \boldsymbol{x}(t_e)) \leq 0$ then the first transversality condition becomes:

$\psi_k(t_e, \boldsymbol{x}(t_e)) \leq 0$
$\boldsymbol{\mu}^* \geq 0$
$\boldsymbol{\psi}^T(t_e, \boldsymbol{x}(t_e)) \boldsymbol{\mu}^* = 0$

6.1.3 Reachability condition

Regularity condition can be interpreted as a reachability condition. If it does not hold, it may not be possible to find a $\boldsymbol{u}^*(t)$ to transfer the state $\boldsymbol{x}(t)$ from initial to end state in finite time.

6.2 Input constrained

`Hamiltonian`
$H(x,u,\bar{\lambda}) = \bar{\lambda}\bar{f}(x,u) = [\bar{\lambda}_1 \cdots \bar{\lambda}_n]f(x,u) + \bar{\lambda}_{n+1}l(x,u)$

6.2.1 Pontryagin maximum principle (autonomous systems)

$\dot{\bar{\boldsymbol{x}}}^* = (\nabla_{\bar{\lambda}}H)(\boldsymbol{x}^*,\boldsymbol{u}^*,\bar{\boldsymbol{\lambda}}^*) = \begin{bmatrix}\boldsymbol{f}((\boldsymbol{x}^*,\boldsymbol{u}^*)\\ l((\boldsymbol{x}^*,\boldsymbol{u}^*)\end{bmatrix}$

with $\bar{\boldsymbol{x}}^*(t_0) = \begin{bmatrix}\boldsymbol{x}_0\\ 0\end{bmatrix}$, $\boldsymbol{x}^*(t_e) = \boldsymbol{x}_e$

$\dot{\bar{\boldsymbol{\lambda}}}^* = -(\nabla_{\bar{x}}H)(\boldsymbol{x}^*,\boldsymbol{u}^*,\bar{\boldsymbol{\lambda}}^*) = \begin{bmatrix}-(\nabla_{\bar{x}}H)(\boldsymbol{x}^*,\boldsymbol{u}^*,\bar{\boldsymbol{\lambda}}^*)\\ 0\end{bmatrix}$

$\bar{\lambda}^*_{n+1} = const. \geq 0$ and $H(\boldsymbol{x}^*,\boldsymbol{u}^*,\bar{\boldsymbol{\lambda}}^*) = const. \geq 0$

For free endtime t_e - transversality condition:
$H(\boldsymbol{x}^*(t_e^*),\boldsymbol{u}^*(t_e^*),\bar{\boldsymbol{\lambda}}^*(t_e^*)) = 0$

with target set condition → transversality condition:
$(\lambda^*(t_e^*))^T\boldsymbol{d} = 0,\ \forall \boldsymbol{d} \in \mathcal{T}_{x^*(t_e^*)}X_{ta}$

Normal case: If $\boldsymbol{u}^*$ dependent of $l(\cdot)$, then set $\bar{\lambda}^*_{n+1}(t) = 1$
Abnormal case: If $\boldsymbol{u}^*$ independent of $l(\cdot)$, then $\bar{\lambda}^*_{n+1}(t) = 0$

6.2.2 Pontryagin maximum principle (non-autonomous systems)

Same as autonomous systems, except that $\boldsymbol{f}$ and l depend on time and
$\frac{\partial}{\partial t}H(t_e^*,\boldsymbol{x}^*,\boldsymbol{u}^*,\bar{\boldsymbol{\lambda}}^*) = (\bar{\boldsymbol{\lambda}}^*)^T(\frac{\partial}{\partial t}\boldsymbol{f})(t_e^*,\boldsymbol{x}^*,\boldsymbol{u}^*)$

6.3 Nonlinear affine input systems

$\dot{\boldsymbol{x}} = \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{j=1}^m \boldsymbol{f}_j(\boldsymbol{x})u_j,\ t > t_0,\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0,\ u_j \in [u_j^-, u_j^+]$

6.3.1 Cost functionals minimizing consumption

Due to affine input structure the problem can be split into independent problems.

$J(u) = \int_{t_0}^{t_e}(l_0(x(t)) + \frac{1}{2}\sum_{j=1}^m r_j|u_j(t)|^2)dt$
$q_j(x,\lambda) = \lambda^T f_j(x)$

Optimal control is defined as - singular case if $q_j(x,\lambda) = \pm r_j$:

$$u_j^* = \begin{cases} u_j^- & q_j(x,\lambda) > r_j \\ 0 & q_j(x,\lambda) \in (-r_j, r_j) \\ u_j^+ & q_j(x,\lambda) < -r_j \end{cases}$$

6.3.2 Cost functionals addressing energy optimality

Squared dependence of input in cost functional (*). Can also be split into m independent minimization problems.

Optimal control is defined as, where $u_j^0 = -\frac{q_j(\cdot)}{r_j}$.

$$u_j^* = \begin{cases} u_j^- & u_j^0 \leq u_j^- \\ u_j^0 & u_j^0 \in (u_j^-, u_j^+) \\ u_j^+ & u_j^0 \geq u_j^+ \end{cases}$$

6.3.3 Cost functionals addressing time optimality

Optimal control is either u_j^- $(q_j(\cdot) > 0)$ or u_j^+ $(q_j(\cdot) < 0)$ where $q_j(\cdot)$ directly results from the minimization of the Hamiltonian (bang-bang control) - singular case if $q_j(\cdot) = 0$
$J(u) = \int_{t_0}^{t_e} 1dt = t_e - t_0$

6.3.4 Singular case

$\zeta^*(t) = (\nabla_u H)(t^*, x^*, u^*, \bar{\lambda}^*)$
Determine k by $\frac{\partial}{\partial u}\left[\frac{d^k}{dt^k}\zeta^*(t)\right] \neq 0$ and this also provides the optimal control u^*.
Then $(-1)^{\frac{k}{2}}\frac{\partial}{\partial u}\left[\frac{d^k}{dt^k}\zeta^*(t)\right] \geq 0$ has to be fulfilled for a minimum along the arc.

7 Numerical solution of optimal control problems

7.1 Indirect methods

Directly approach the 2-point BVP and solve:
$\dot{\boldsymbol{x}} = (\nabla_\lambda H)(t, \boldsymbol{x}, \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}),\ \ \boldsymbol{x}^*(t_0) = \boldsymbol{x}_0,\ \psi(t_e, \boldsymbol{x}(t_e)) = \boldsymbol{0}$
$\dot{\boldsymbol{\lambda}}^* = -(\nabla_x H)(t, \boldsymbol{x}, \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}),\ \ \boldsymbol{\lambda}(t_e) = (\nabla_{x_e}\phi)(t_e, \boldsymbol{x}(t_e), \boldsymbol{\mu}),$
$\dot{\boldsymbol{\mu}} = \boldsymbol{0}$

+ prove insight into structure of optimal solution
+ determine highly accurate or even exact solution
- utilize adjoint variables for sensitivity analysis and controller design

7.1.1 Discretization methods

Discretizing $[t_0, t_e]$ into $N+1$ steps to approximate solution at discretization points using trapezoidal rule. Numerical solution by finding zeros of nonlinear algebraic system using e.g. Newtons method.

+ numerically robust solution due to simultaneous consideration of differential equations and boundary conditions
+ special matrix structure can be exploited for numerical solution (reuse previous results $\boldsymbol{F}(\cdot)$)
∘ convergence relies on the initial guess of adjoint variables
∘ number of discretization steps influences accuracy and computational burden

7.1.2 Shooting method

Replace BVP by IVP by guessing initial states $\boldsymbol{\lambda}(t_0) = \boldsymbol{\lambda}_0$ and $\boldsymbol{\mu}$. Solve residual terms at boundary $t = t_e$ with Newton method to get iterative solution.

+ Low implementation effort
∘ Relies on proper choice of initial state $\boldsymbol{\lambda}_0$
- canonical equations tend to be weakly stable, thus numerical issues may arise due to integration in large intervals. Use multiple shooting instead.

7.1.3 Collocation methods

Use linear combination of linear independent basis functions ((Legendre) polynomials) which fulfill the boundary conditions at $t = t_0$ and $t = t_e$. It is imposed that differential equations and boundary conditions are satisfied pointwise at $K + 1$ distinct collocation points, to then determine the coefficients for the basis functions.

7.1.4 Extension to free endtime

Use time scaling so that $\frac{d}{dt} = \frac{1}{\nu}\frac{d}{d\tau}$ in order to transform E-L equations (or canonical equations) to a fixed time interval $\tau \in [0, 1]$. Determination of optimal endtime t_e^* reduces to find constant scaling factor $\nu > 0$.

7.2 Direct methods

Discretize infinite-dimensional opt. prob. to obtain a finite-dimensional static optimization problem. Assuming Bolzano form with fixed endtime and equality and inequality constraints. The time interval $[t_0, t_e]$ is discretized into $N + 1$ stages and control inputs are parametrized in each subinterval with order of approximation by functions that are piecewise constant (sample and hold), piecewise linear (linear interpolation), etc. In practice: Lagrange polynomials.

+ avoid determination of canonical equations
+ simpler incorporation of state and path constraints
+ compute Lagrange multiplier in post-processing step
+ improved convergence behaviour
+ allow solving of optimal control problems for systems with ODE, differential-algebraic equations and PDE
- only suboptimal solution

7.2.1 Direct sequential methods

Problem reduces to static optimization problem.

+ Use SQP for solving nonlinear optimization problem.
∘ Accuracy depends on the used numerical solver and is independent of the time grid.
- Problems may arise with unstable systems or differential equations with no solution for certain decision variables.

7.2.2 Direct simultaneous methods

Discretized in the control and state variables.

- Differential equations are fulfilled at converged solution $\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{u}}^*$ only.

- Inequality constraints are satisfied at discretization points t^j only.
- Number of time intervals influences approximation of optimal control and accuracy of approxmiation of the solution to the differential equations.