

Model Predictive Control and Reinforcement Learning

– Lecture 2: Nonlinear Optimization –

Joschka Boedecker and Moritz Diehl

University of Freiburg

October 4, 2023

universität freiburg

Outline of the lecture



- 1 Basic definitions
- 2 Some classifications of optimization problems
- 3 Optimality conditions
- 4 Nonlinear programming algorithms
- 5 Optimal control problems as nonlinear programs



The slides of this talk were prepared with help by Armin Nurkanović.

What is an optimization problem?



Minimize (or maximize) an objective function $F(w)$ depending on decision variables w subject to equality and/or inequality constraints

What is an optimization problem?

Minimize (or maximize) an objective function $F(w)$ depending on decision variables w subject to equality and/or inequality constraints

An optimization problem

$$\min_w F(w) \quad (1a)$$

$$\text{s.t. } G(w) = 0 \quad (1b)$$

$$H(w) \geq 0 \quad (1c)$$

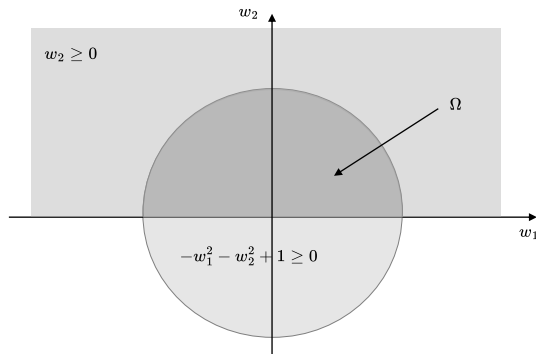
Terminology

- ▶ w - decision variable
- ▶ F : objective/cost function
- ▶ G, H : equality and inequality constraint functions

- ▶ Optimization is a powerful tool used in all quantitative sciences
- ▶ Only in few special cases a closed form solution exist
- ▶ Use an iterative algorithm to find solution
- ▶ The optimization problem may be parametric, and all functions depend on a fixed parameter p

Definition

The feasible set of the optimization problem (1) is defined as $\Omega = \{w \in \mathbb{R}^n \mid G(w) = 0, H(w) \geq 0\}$. A point $w \in \Omega$ is called a feasible point.



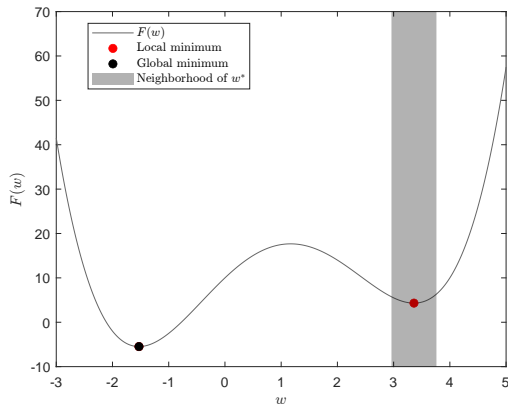
The feasible set is the intersection of the two grey areas (halfspace and circle)

Basic definitions: local and global minimizer

Definition

- ▶ A point $w^* \in \Omega$ is called a **local minimizer** of the NLP (1) if there exists an open ball $\mathcal{B}_\epsilon(w^*)$ with $\epsilon > 0$, such that for all $w \in \mathcal{B}_\epsilon(w^*) \cap \Omega$ it holds that $F(w) \geq F(w^*)$.
- ▶ A point $w^* \in \Omega$ is called a **global minimizer** of the NLP (1) if for all $w \in \Omega$ it holds that $F(w) \geq F(w^*)$.

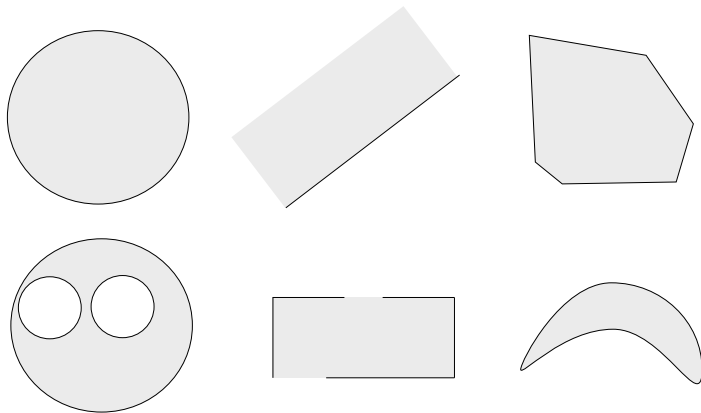
The value $F(w^*)$ at a local/global minimizer w^* is called local/global minimum.



$$F(w) = \frac{1}{2}w^4 - 2w^3 - 3w^2 + 12w + 10$$

Convex sets

A key concept in optimization is convexity



A set Ω is said to be convex if for any w_1, w_2 and any $\theta \in [0, 1]$ it holds $\theta w_1 + (1 - \theta)w_2 \in \Omega$

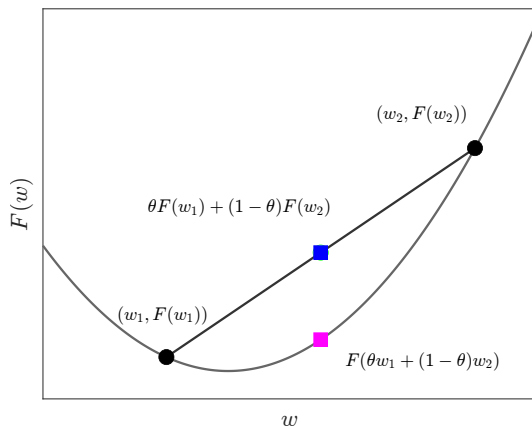
- ▶ A function F is convex if for every $w_1, w_2 \in \mathbb{R}^n$ and $\theta \in [0, 1]$ it holds that

$$F(\theta w_1 + (1 - \theta)w_2) \leq \theta F(w_1) + (1 - \theta)F(w_2)$$

- ▶ F is concave if and only if $-F$ is convex
- ▶ F is convex if and only if the epigraph

$$\text{epi}F = \{(w, t) \in \mathbb{R}^{n_w+1} \mid F(w) \leq t\}$$

is a convex set





A convex optimization problem

$$\begin{array}{ll}\min_w & F(w) \\ \text{s.t.} & G(w) = 0 \\ & H(w) \geq 0\end{array}$$

An optimization problem is convex if the objective function F is convex and the feasible set Ω is convex.

- ▶ Every locally optimal solution is global
- ▶ First order conditions are necessary and sufficient (we come back to this)
- ▶ *"...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."* R. T. Rockafellar, SIAM Review, 1993

Outline of the lecture



- 1 Basic definitions
- 2 Some classifications of optimization problems
- 3 Optimality conditions
- 4 Nonlinear programming algorithms
- 5 Optimal control problems as nonlinear programs



Optimization problems can be:

- ▶ unconstrained ($\Omega = \mathbb{R}^n$) or constrained ($\Omega \subset \mathbb{R}^n$)
- ▶ convex or nonconvex
- ▶ linear or nonlinear
- ▶ differentiable or nonsmooth
- ▶ continuous or (mixed-)integer
- ▶ finite or infinite dimensional



Optimization problems can be:

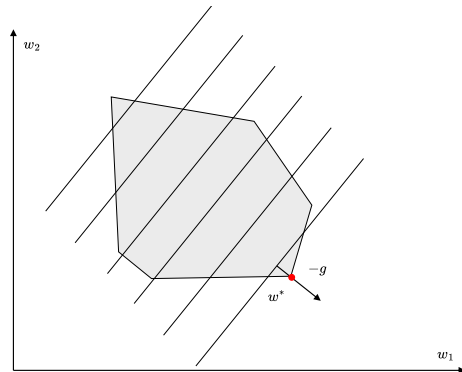
- ▶ unconstrained ($\Omega = \mathbb{R}^n$) or constrained ($\Omega \subset \mathbb{R}^n$)
- ▶ convex or nonconvex
- ▶ linear or nonlinear
- ▶ differentiable or nonsmooth
- ▶ continuous or (mixed-)integer
- ▶ finite or infinite dimensional

*"... the main fact, which should be known to any person dealing with optimization models, is that in general, optimization problems are unsolvable."
Yurii Nesterov, Lectures on Convex Optimization, 2018.*

Class 1: Linear Programming (LP)

Linear program

$$\begin{aligned} \min_w \quad & g^\top w \\ \text{s.t.} \quad & Aw - b = 0 \\ & Cw - d \geq 0 \end{aligned}$$



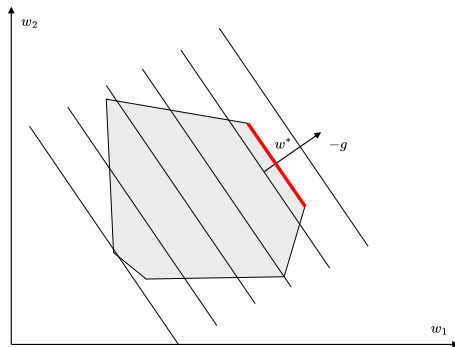
- ▶ convex optimization problem
- ▶ 1947: simplex method by Dantzig, 1984: polynomial time interior-point method by Karmarkar
- ▶ if not unbounded, the solution is always at edge or vertex of the feasible set
- ▶ today very mature and reliable

Class 1: Linear Programming (LP)



Linear program

$$\begin{aligned} \min_w \quad & g^\top w \\ \text{s.t.} \quad & Aw - b = 0 \\ & Cw - d \geq 0 \end{aligned}$$

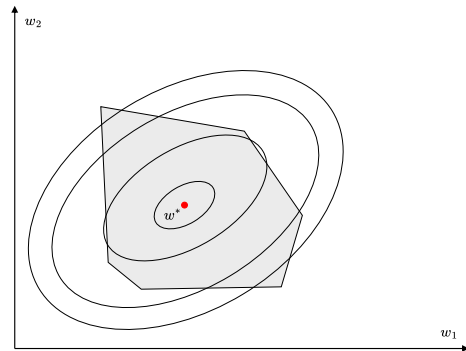


- ▶ convex optimization problem
- ▶ 1947: simplex method by Dantzig, 1984: polynomial time interior-point method by Karmarkar
- ▶ if not unbounded, the solution is always at edge or vertex of the feasible set
- ▶ today very mature and reliable

Class 2: Quadratic Programming (QP)

Quadratic program

$$\begin{aligned} \min_w \quad & \frac{1}{2} w^\top Q w + g^\top w \\ \text{s.t.} \quad & A w - b = 0 \\ & C w - d \geq 0 \end{aligned}$$



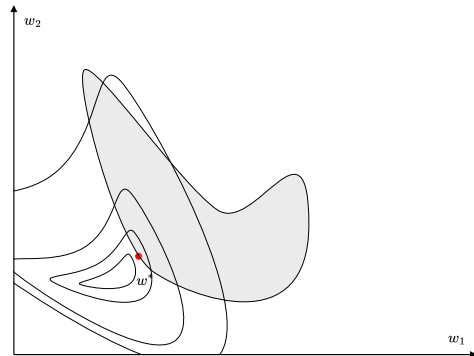
- ▶ depending on Q , can be convex and nonconvex
- ▶ solved online in linear model predictive control
- ▶ many good solvers: Gurobi, OSQP, HPIPM, qpOASES, OOQP, ...
- ▶ subproblems in nonlinear optimization

Class 3: Nonlinear Program (NLP)



Nonlinear programming problem

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$



- ▶ can be convex and nonconvex
- ▶ solved with iterative Newton-type algorithms
- ▶ solved in nonlinear model predictive control

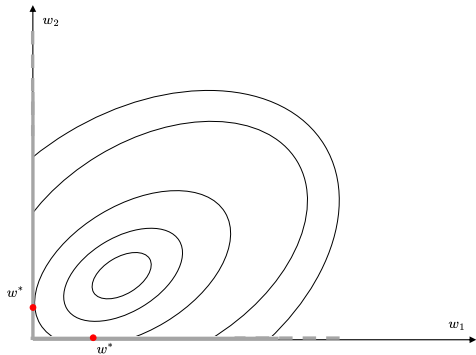
Class 4: Mathematical programs with Complementarity Constraints (MPCC)



MPCC

$$\begin{aligned} \min_{w_0, w_1, w_2} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \\ & 0 \leq w_1 \perp w_2 \geq 0 \end{aligned}$$

$$w = [w_0^\top, w_1^\top, w_2^\top]^\top$$



- ▶ Special case of nonlinear programs treated extensively in this course
- ▶ Standard constraint qualifications fail to hold
- ▶ Very powerful modeling concept
- ▶ Requires specialized theory and algorithms (Lectures by C. Kirches)

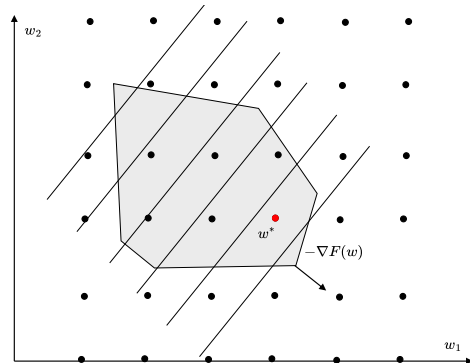
Class 5: Mixed-integer programming

Mixed-integer nonlinear program (MINLP)

$$\begin{aligned} \min_{w_0 \in \mathbb{R}^p, w_1 \in \mathbb{Z}^q} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$w = [w_0^\top, w_1^\top]^\top, n = p + q$$

- ▶ Combinatorial problem, feasible set is finite
- ▶ Branch and bound, brunch and cut methods
- ▶ Requires solution of many relaxed continuous convex or nonconvex problems
- ▶ Optimization problems treated in this course can always be reformulate into MINLPs (but not very efficient)





Continuous-time Optimal Control Problem

$$\min_{x(\cdot), u(\cdot)} \int_0^T L_c(x(t), u(t)) dt + E(x(T))$$

$$\text{s.t. } x(0) = \bar{x}_0$$

$$\dot{x}(t) = f_c(x(t), u(t))$$

$$0 \geq h(x(t), u(t)), \quad t \in [0, T]$$

$$0 \geq r(x(T))$$

- ▶ Infinite dimensional problem, can be convex or nonconvex
- ▶ Dynamic constraint can be replaced by $\dot{x}(t) = f_c(x(t), u(t))$:
 - ▶ DAE
 - ▶ PDE
 - ▶ stochastic ODE/PDE
 - ▶ **Nonsmooth ODE** - this course
- ▶ All or some components of $u_i(t)$ may take values in \mathbb{Z} (mixed-integer OCP)

Direct optimal control methods solve Nonlinear Programs (NLP)



Continuous time OCP

$$\begin{aligned} \min_{x(\cdot), u(\cdot)} \quad & \int_0^T L_c(x(t), u(t)) dt + E(x(T)) \\ \text{s.t.} \quad & x(0) = \bar{x}_0 \\ & \dot{x}(t) = f_c(x(t), u(t)) \\ & 0 \geq h(x(t), u(t)), \quad t \in [0, T] \\ & 0 \geq r(x(T)) \end{aligned}$$

Direct methods like direct collocation, multiple shooting first discretize, then optimize.

Direct optimal control methods solve Nonlinear Programs (NLP)



Continuous time OCP

$$\begin{aligned} \min_{x(\cdot), u(\cdot)} \quad & \int_0^T L_c(x(t), u(t)) dt + E(x(T)) \\ \text{s.t.} \quad & x(0) = \bar{x}_0 \\ & \dot{x}(t) = f_c(x(t), u(t)) \\ & 0 \geq h(x(t), u(t)), \quad t \in [0, T] \\ & 0 \geq r(x(T)) \end{aligned}$$

Direct methods like direct collocation, multiple shooting first discretize, then optimize.

Discrete time OCP (an NLP)

$$\begin{aligned} \min_{x, u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.



Discrete time OCP (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear MPC solves Nonlinear Programs (NLP)

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear MPC solves Nonlinear Programs (NLP)

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$



- 1 Basic definitions
- 2 Some classifications of optimization problems
- 3 Optimality conditions
- 4 Nonlinear programming algorithms
- 5 Optimal control problems as nonlinear programs

Algebraic characterization of **unconstrained** local optima

Consider the unconstrained problem: $\min_{w \in \mathbb{R}^n} F(w)$

First-Order **Necessary** Condition of Optimality (FONC)

$$w^* \text{ local optimum} \Rightarrow \nabla F(w^*) = 0, w^* \text{ stationary point}$$

Second-Order **Necessary** Condition of Optimality (SONC)

$$w^* \text{ local optimum} \Rightarrow \nabla^2 F(w^*) \succeq 0$$

Algebraic characterization of **unconstrained** local optima

Consider the unconstrained problem: $\min_{w \in \mathbb{R}^n} F(w)$

First-Order **Necessary** Condition of Optimality (FONC)

$$w^* \text{ local optimum} \Rightarrow \nabla F(w^*) = 0, \quad w^* \text{ stationary point}$$

Second-Order **Necessary** Condition of Optimality (SONC)

$$w^* \text{ local optimum} \Rightarrow \nabla^2 F(w^*) \succeq 0$$

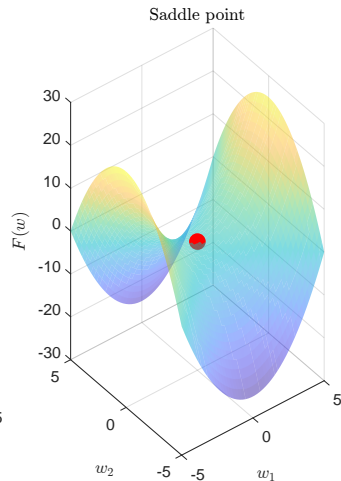
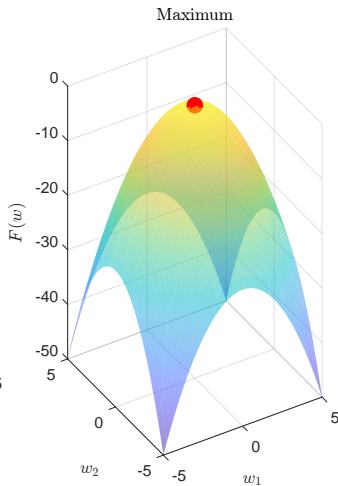
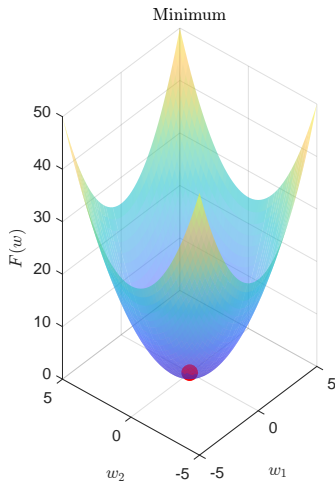
Second-Order **Sufficient** Conditions of Optimality (SOSC)

$$\nabla F(w^*) = 0 \text{ and } \nabla^2 F(w^*) \succ 0 \Rightarrow x^* \text{ strict local minimum}$$

$$\nabla F(w^*) = 0 \text{ and } \nabla^2 F(w^*) \prec 0 \Rightarrow x^* \text{ strict local maximum}$$

No conclusion can be drawn in the case $\nabla^2 F(w^*)$ is indefinite!

Type of stationary points

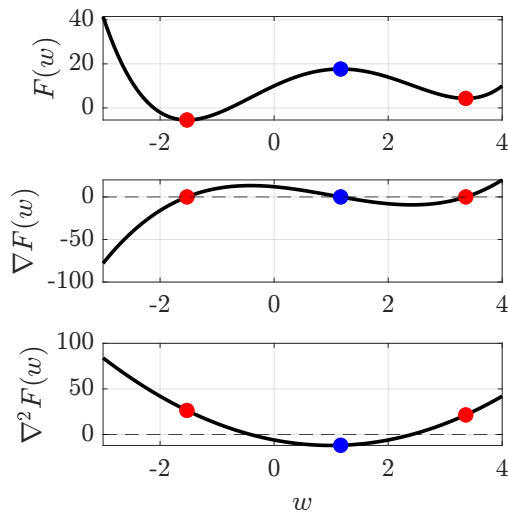


A stationary point can be a minimum, maximum or a saddle point

Optimality conditions - unconstrained

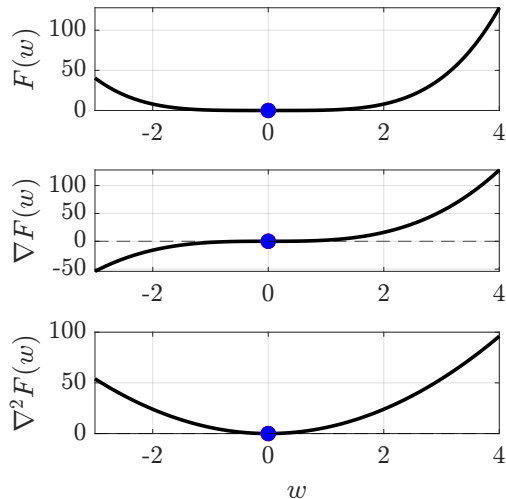


- Necessary conditions: find a candidate point (or to exclude points)
- Sufficient conditions: verify optimality of a candidate point



Optimality conditions - unconstrained

- Necessary conditions: find a candidate point (or to exclude points)
- Sufficient conditions: verify optimality of a candidate point
- A minimizer must satisfy SONC, but does not have to satisfy SOSC





Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$ is the **Lagrangian**



Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$ is the **Lagrangian**

Definition (LICQ)

A point w satisfies Linear Independence Constraint Qualification **LICQ** if and only if $\nabla G(w)$ is full column rank



Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} & F(w) \\ \text{s.t.} & G(w) = 0 \end{aligned}$$

$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$ is the **Lagrangian**

Definition (LICQ)

A point w satisfies Linear Independence Constraint Qualification **LICQ** if and only if $\nabla G(w)$ is full column rank

First-order Necessary Conditions

Let F, G in \mathcal{C}^1 . If w^* is a (local) **minimizer**, and w^* satisfies **LICQ**, then there is a **unique vector** λ such that:

$$\nabla_w \mathcal{L}(w^*, \lambda^*) = \nabla F(w^*) - \nabla G(w^*) \lambda = 0$$

Dual feasibility

$$\nabla_\lambda \mathcal{L}(w^*, \lambda^*) = G(w^*) = 0$$

Primal feasibility



Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$



Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

Definition (LICQ)

A point w satisfies LICQ if and only if

$$[\nabla G(w), \quad \nabla H_{\mathcal{A}}(w)]$$

is full column rank

Active set $\mathcal{A} = \{i \mid H_i(w) = 0\}$

The KKT conditions

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

Definition (LICQ)

A point w satisfies LICQ if and only if

$$[\nabla G(w), \quad \nabla H_{\mathcal{A}}(w)]$$

is full column rank

Active set $\mathcal{A} = \{i \mid H_i(w) = 0\}$

Theorem (KKT conditions)

Let F, G, H be \mathcal{C}^1 . If w^* is a (local) minimizer and satisfies LICQ, then there are unique vectors λ^* and μ^* such that (w^*, λ^*, μ^*) satisfies:

$$\nabla_w \mathcal{L}(w^*, \mu^*, \lambda^*) = 0, \quad \mu^* \geq 0,$$

$$G(w^*) = 0, \quad H(w^*) \geq 0$$

$$\mu_i^* H_i(w^*) = 0, \quad \forall i$$

Dual feasibility

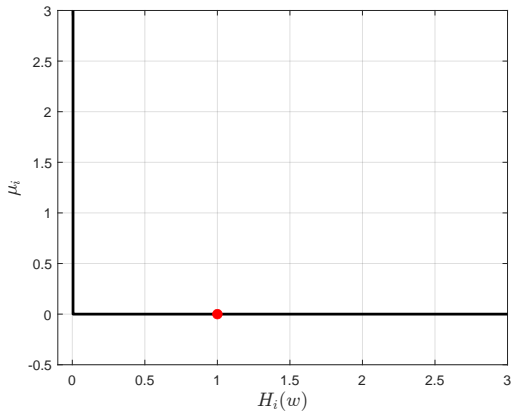
Primal feasibility

Complementary slackness

The complementary slackness condition

Active constraints:

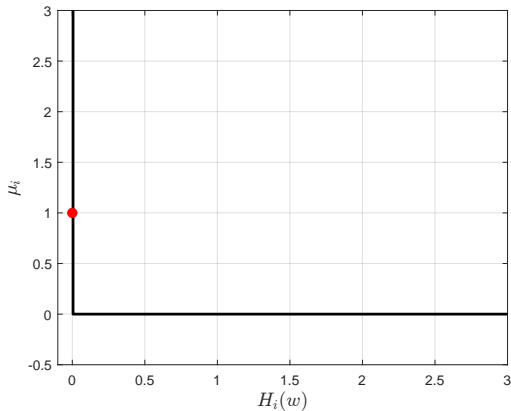
- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and H_i is inactive



The complementary slackness condition

Active constraints:

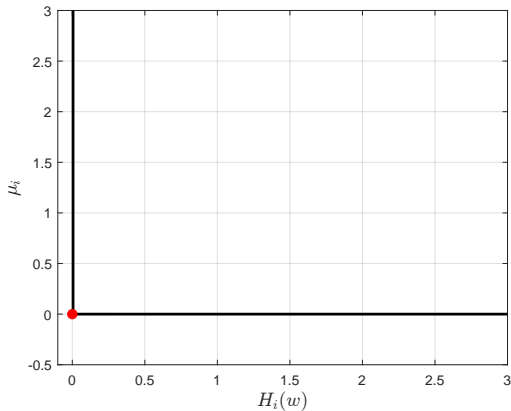
- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and H_i is inactive
- ▶ $\mu_i^* > 0$ and $H_i(w) = 0$ then $H_i(w)$ is strictly active



The complementary slackness condition

Active constraints:

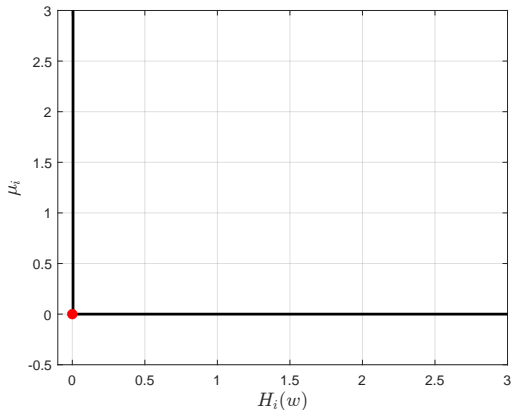
- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and H_i is inactive
- ▶ $\mu_i^* > 0$ and $H_i(w) = 0$ then $H_i(w)$ is strictly active
- ▶ $\mu_i^* = 0$ and $H_i(w) = 0$ then $H_i(w)$ is weakly active



The complementary slackness condition

Active constraints:

- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and H_i is **inactive**
- ▶ $\mu_i^* > 0$ and $H_i(w) = 0$ then $H_i(w)$ is **strictly active**
- ▶ $\mu_i^* = 0$ and $H_i(w) = 0$ then $H_i(w)$ is **weakly active**
- ▶ We define the **active set** \mathbb{A}^* as the set of indices i of the active constraints

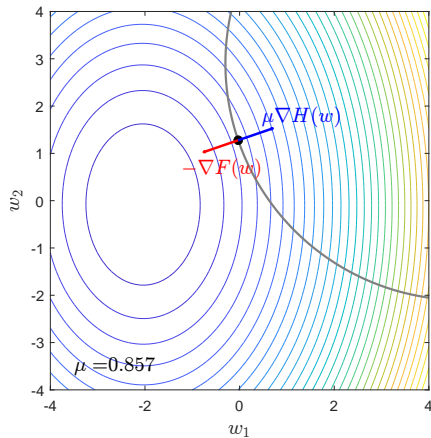


Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence



$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$



Some intuitions on the KKT conditions

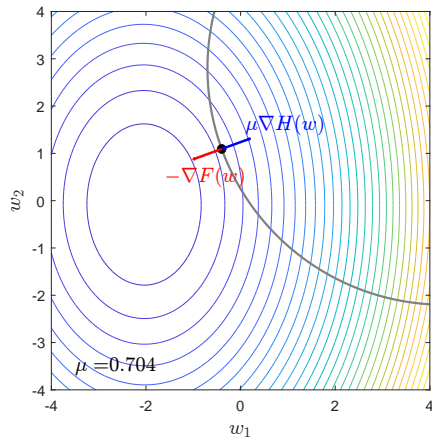
Ball rolling down a valley blocked by a fence



$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

► $-\nabla F$ is the gravity



Some intuitions on the KKT conditions

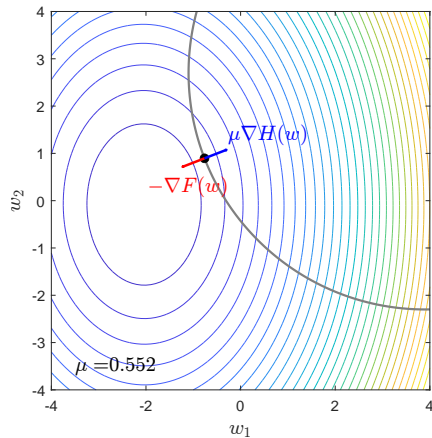
Ball rolling down a valley blocked by a fence



$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

► $-\nabla F$ is the gravity



Some intuitions on the KKT conditions

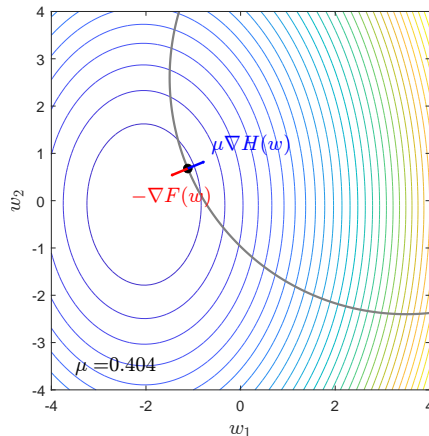
Ball rolling down a valley blocked by a fence



$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball



Some intuitions on the KKT conditions

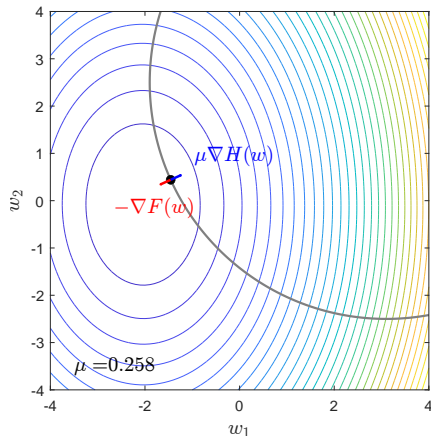
Ball rolling down a valley blocked by a fence



$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.



Some intuitions on the KKT conditions

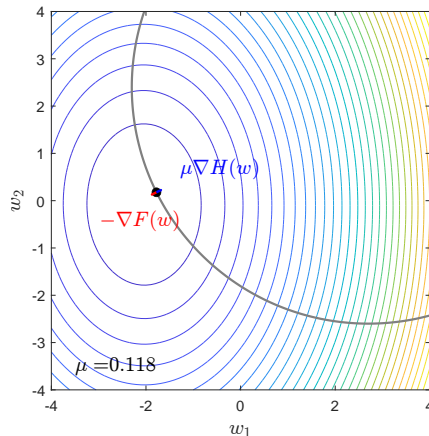
Ball rolling down a valley blocked by a fence



$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.



Some intuitions on the KKT conditions

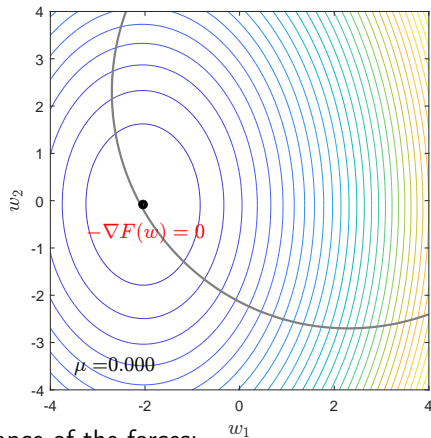
Ball rolling down a valley blocked by a fence



$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.
- ▶ Weakly active constraint:
 $H(w) = 0$, $\mu = 0$ the ball touches the fence but no force is needed



Balance of the forces:

$$\nabla \mathcal{L}(w, \mu) = \nabla F(w) - \mu \nabla H(w) = 0$$

Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence



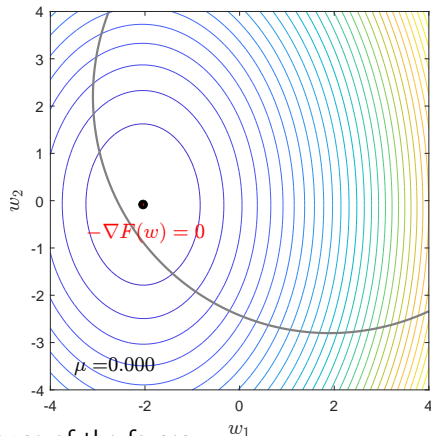
$$\min_{w \in \mathbb{R}^n} F(w)$$

$$\text{s.t. } H(w) \geq 0$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.
- ▶ Weakly active constraint:
 $H(w) = 0$, $\mu = 0$ the ball touches the fence but no force is needed
- ▶ Inactive constraint $H(w) > 0$, $\mu = 0$

$$H(w) > 0, \quad \mu = 0$$

- ▶ Complementary slackness $\mu H = 0$ describes a contact problem



Balance of the forces:

$$\nabla \mathcal{L}(w, \mu) = \nabla F(w) - \mu \nabla H(w) = 0$$



Optimality conditions for NLP with equality and/or inequality constraints:

- **First-Order Necessary Conditions:** Under LICQ and differentiability, a **local optimum** of the NLP satisfies the **KKT conditions**.



Optimality conditions for NLP with equality and/or inequality constraints:

- ▶ **First-Order Necessary Conditions:** Under LICQ and differentiability, a **local optimum** of the NLP satisfies the **KKT conditions**.
- ▶ **Second-Order (Necessary or) Sufficient Conditions** require **positive-(semi)-definiteness** of the Hessian in so called **critical directions** (feasible and non-ascent directions)



Optimality conditions for NLP with equality and/or inequality constraints:

- ▶ **First-Order Necessary Conditions:** Under LICQ and differentiability, a **local optimum** of the NLP satisfies the **KKT conditions**.
- ▶ **Second-Order (Necessary or) Sufficient Conditions** require **positive-(semi)-definiteness** of the Hessian in so called **critical directions** (feasible and non-ascent directions)

Nonconvex problem \Rightarrow minimum is not necessarily global.

But some nonconvex problems have a **unique minimum**



- 1 Basic definitions
- 2 Some classifications of optimization problems
- 3 Optimality conditions
- 4 Nonlinear programming algorithms
- 5 Optimal control problems as nonlinear programs

Newton's method

To solve a nonlinear system, solve a sequence of linear systems



Linearization of F at linearization point \bar{w}

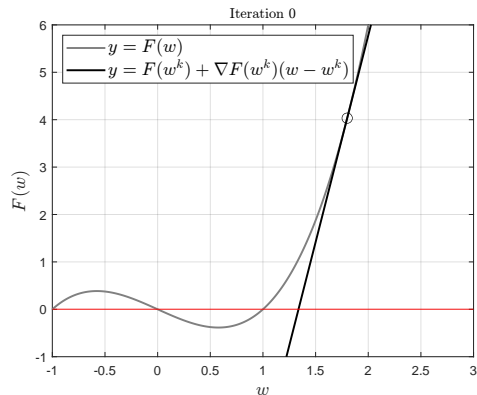
equals

First order Taylor series at \bar{w}

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \frac{\partial F}{\partial w}(\bar{w})(w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



Newton's method

To solve a nonlinear system, solve a sequence of linear systems



Linearization of F at linearization point \bar{w}

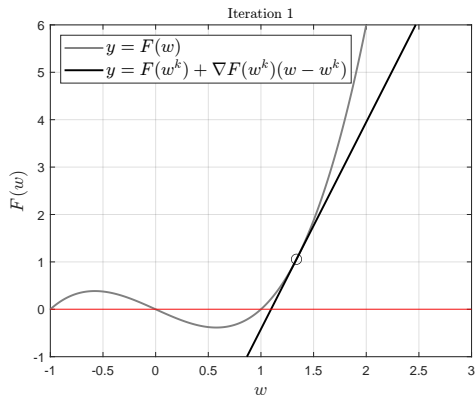
equals

First order Taylor series at \bar{w}

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



Newton's method

To solve a nonlinear system, solve a sequence of linear systems



Linearization of F at linearization point \bar{w}

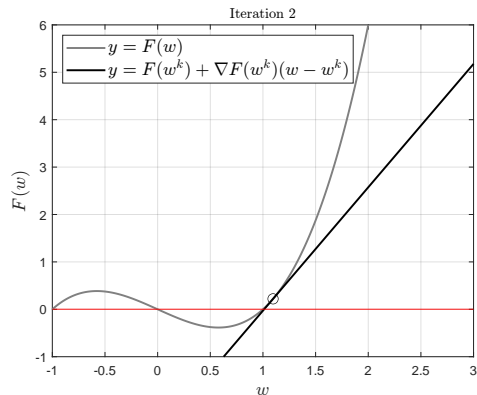
equals

First order Taylor series at \bar{w}

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



Newton's method

To solve a nonlinear system, solve a sequence of linear systems



Linearization of F at linearization point \bar{w}

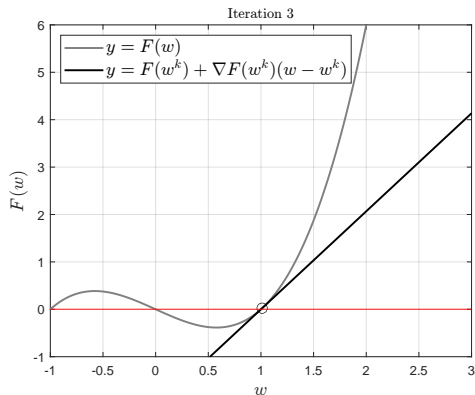
equals

First order Taylor series at \bar{w}

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



General Nonlinear Program (NLP)

In direct methods, we have to solve the discretized optimal control problem, which is a Nonlinear Program (NLP)

General Nonlinear Program (NLP)

$$\min_w F(w) \quad \text{s.t.} \quad \begin{cases} G(w) = 0 \\ H(w) \geq 0 \end{cases}$$

We first treat the case without inequalities

NLP only with equality constraints

$$\min_w F(w) \quad \text{s.t.} \quad G(w) = 0$$



Lagrange function

$$\mathcal{L}(w, \lambda) = F(w) - \lambda^T G(w)$$

Then for an optimal solution w^* exist multipliers λ^* such that

Nonlinear root-finding problem

$$\begin{aligned}\nabla_w \mathcal{L}(w^*, \lambda^*) &= 0 \\ G(w^*) &= 0\end{aligned}$$



How to solve nonlinear equations

$$\begin{aligned}\nabla_w \mathcal{L}(w^*, \lambda^*) &= 0 \\ G(w^*) &= 0 \quad ?\end{aligned}$$

Linearize!

$$\begin{aligned}\nabla_w \mathcal{L}(w^k, \lambda^k) + \nabla_w^2 \mathcal{L}(w^k, \lambda^k) \Delta w - \nabla_w G(w^k) \Delta \lambda &= 0 \\ G(w^k) + \nabla_w G(w^k)^T \Delta w &= 0\end{aligned}$$

This is equivalent, due to $\nabla \mathcal{L}(w^k, \lambda^k) = \nabla F(w^k) - \nabla G(w^k) \lambda^k$ with the shorthand $\lambda^+ = \lambda^k + \Delta \lambda$ to

$$\begin{aligned}\nabla_w F(w^k) + \nabla_w^2 \mathcal{L}(w^k, \lambda^k) \Delta w - \nabla_w G(w^k) \lambda^+ &= 0 \\ G(w^k) + \nabla_w G(w^k)^T \Delta w &= 0\end{aligned}$$

Newton Step = Quadratic Program

Conditions

$$\begin{array}{rcl} \nabla_w F(w^k) & + \nabla_w^2 \mathcal{L}(w^k, \lambda^k) \Delta w & - \nabla_w G(w^k) \lambda^+ = 0 \\ G(w^k) & + \nabla_w G(w^k)^T \Delta w & = 0 \end{array}$$

are optimality conditions of a quadratic program (QP), namely:

Quadratic program

$$\begin{array}{ll} \min_{\Delta w} & \nabla F(w^k)^T \Delta w + \frac{1}{2} \Delta w^T A^k \Delta w \\ \text{s.t.} & G(w^k) + \nabla G(w^k)^T \Delta w = 0, \end{array}$$

with

$$A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k)$$

Newton's method

The full step Newton's Method iterates by solving in each iteration the Quadratic Program

$$\begin{aligned} \min_{\Delta w} \quad & \nabla F(w^k)^T \Delta w + \frac{1}{2} \Delta w^T A^k \Delta w \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^T \Delta w = 0, \end{aligned}$$

with $A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k)$. This obtains as solution the step Δw^k and the new multiplier $\lambda_{\text{QP}}^+ = \lambda^k + \Delta \lambda^k$

New iterate

$$\begin{aligned} w^{k+1} &= w^k + \Delta w^k \\ \lambda^{k+1} &= \lambda^k + \Delta \lambda^k = \lambda_{\text{QP}}^+ \end{aligned}$$

This Newton's method is also called “Sequential Quadratic Programming (SQP) for equality constrained optimization” (with “exact Hessian” and “full steps”)



Regard again NLP with both, equalities and inequalities:

NLP with equality and inequality constraints

$$\min_w F(w) \quad \text{s.t.} \quad \begin{cases} G(w) = 0 \\ H(w) \geq 0 \end{cases}$$

Lagrangian function for NLP with equality and inequality constraints

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^T G(w) - \mu^T H(w)$$



Theorem (Karush-Kuhn-Tucker (KKT) conditions)

Let F, G, H be \mathcal{C}^2 . If w^* is a (local) minimizer and satisfies LICQ, then there are unique vectors λ^* and μ^* such that (w^*, λ^*, μ^*) satisfies:

$$\nabla_w \mathcal{L}(w^*, \mu^*, \lambda^*) = 0$$

$$G(w^*) = 0$$

$$H(w^*) \geq 0$$

$$\mu^* \geq 0$$

$$H(w^*)^\top \mu^* = 0$$

- ▶ These contain nonsmooth conditions (the last three) which are called *complementarity conditions*
- ▶ This system cannot be solved by Newton's Method. But still with SQP...

Sequential Quadratic Programming (SQP)

By Linearizing all functions within the KKT Conditions, and setting $\lambda^+ = \lambda^k + \Delta\lambda$ and $\mu^+ = \mu^k + \Delta\mu$, we obtain the KKT conditions of a Quadratic Program (QP) (we omit these conditions).

QP with inequality constraints

$$\begin{aligned} \min_{\Delta w} \quad & \nabla F(w^k)^T \Delta w + \frac{1}{2} \Delta w^T A^k \Delta w \\ \text{s.t.} \quad & \begin{cases} G(w^k) + \nabla G(w^k)^T \Delta w & = & 0 \\ H(w^k) + \nabla H(w^k)^T \Delta w & \geq & 0 \end{cases} \end{aligned}$$

with

$$A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k)$$

and its solution delivers

$$\Delta w^k, \quad \lambda_{\text{QP}}^+, \quad \mu_{\text{QP}}^+$$

Constrained Gauss-Newton Method

In special case of least squares objectives

Least squares objective function

$$F(w) = \frac{1}{2} \|R(w)\|_2^2$$

can approximate Hessian $\nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k)$ by much cheaper

$$A^k = \nabla R(w) \nabla R(w)^T.$$

Need no multipliers to compute A^k ! QP= linear least squares:

Gauss-Newton QP

$$\begin{aligned} \min_{\Delta w} \quad & \frac{1}{2} \|R(w^k) + \nabla R(w^k)^T \Delta w\|_2^2 \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^T \Delta w = 0 \\ & H(w^k) + \nabla H(w^k)^T \Delta w \geq 0 \end{aligned}$$

Convergence: linear (better if $\|R(w^*)\|$ small)

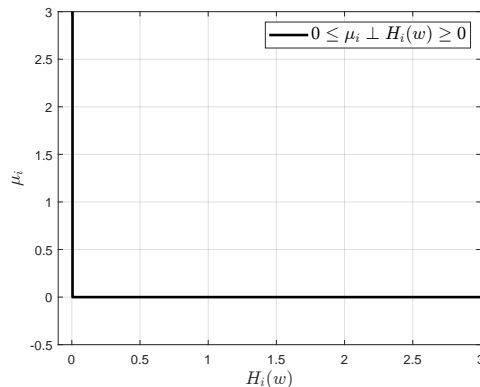
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

KKT conditions

$$\begin{aligned} \nabla F(w) - \nabla H(w)^\top \mu &= 0 \\ 0 \leq \mu \perp H(w) &\geq 0 \end{aligned}$$

Main difficulty: inequality conditions introduce nonsmoothness in the KKT conditions



The barrier problem



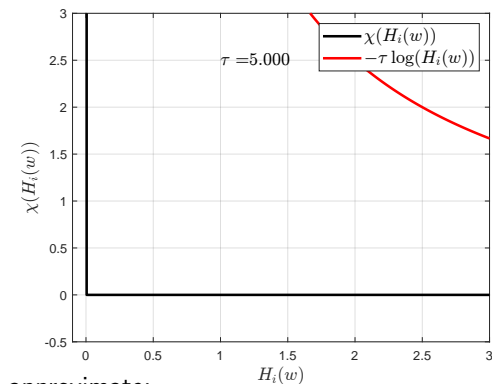
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

Main idea: put inequality constraint into objective



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem



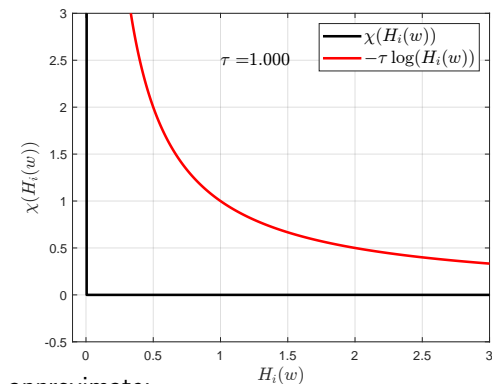
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

Main idea: put inequality constraint into objective



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem



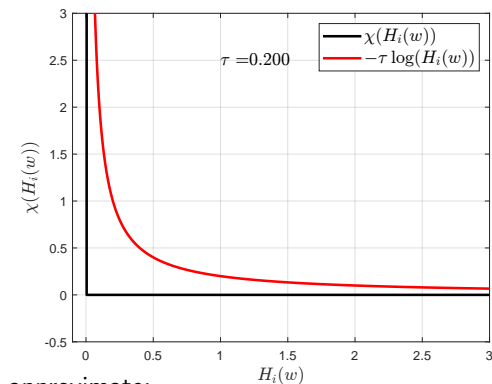
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

Main idea: put inequality constraint into objective



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem



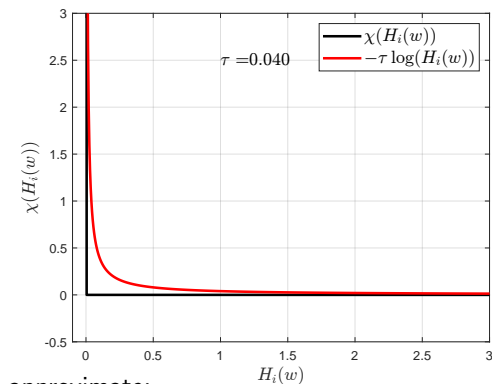
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

Main idea: put inequality constraint into objective



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem



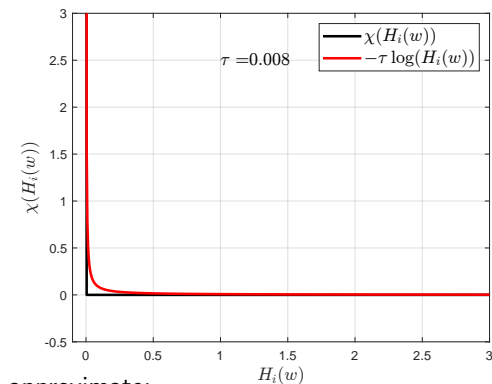
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

Main idea: put inequality constraint into objective



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem



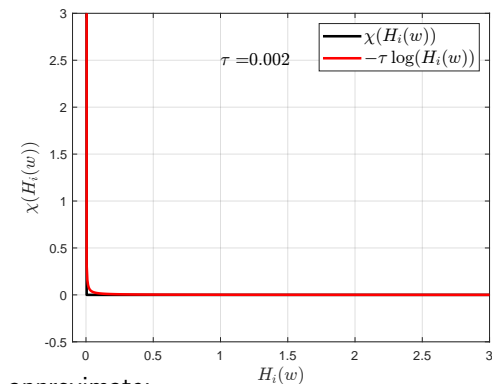
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

Main idea: put inequality constraint into objective



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

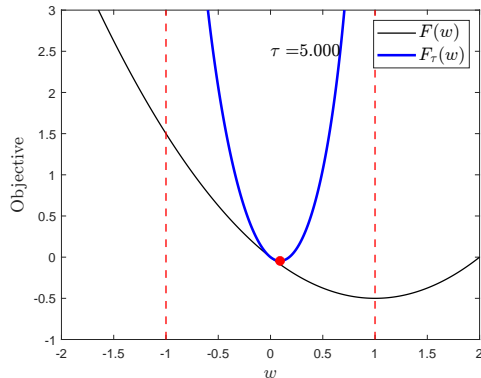
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



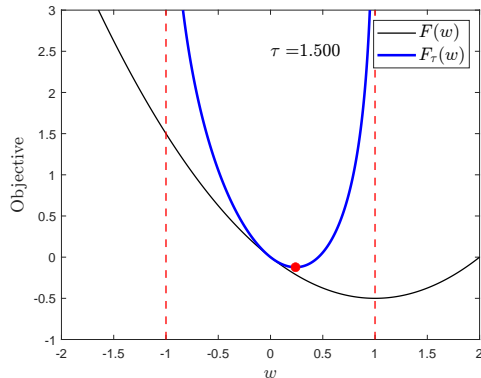
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



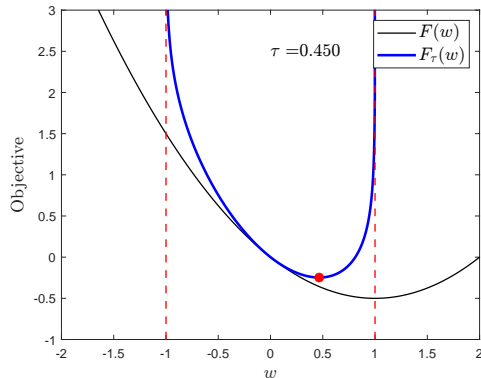
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



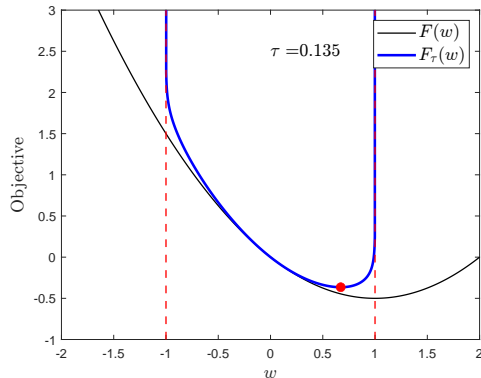
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



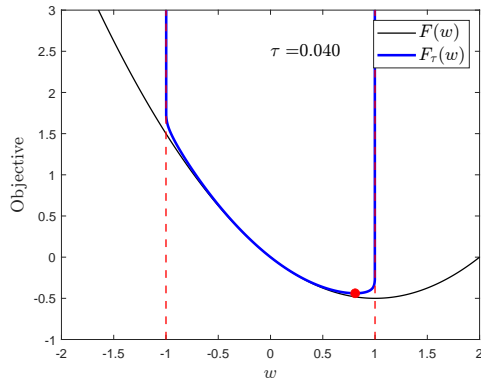
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



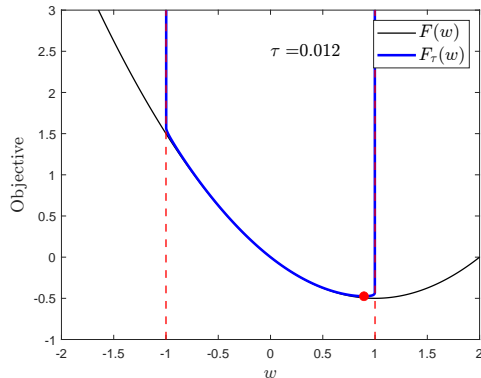
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



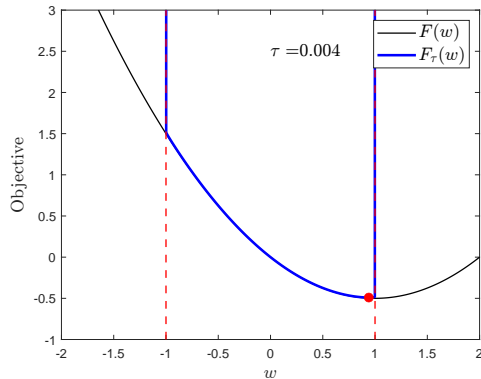
An example of the barrier problem

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w \quad 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

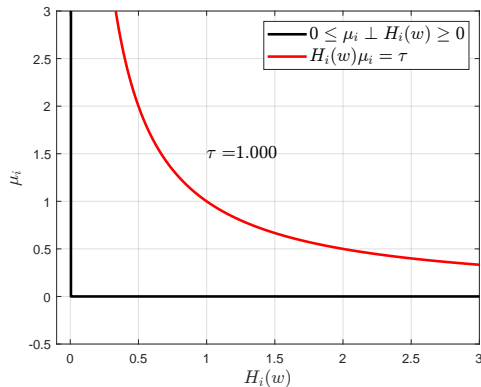
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w)^\top \mu = 0$$

$$H_i(w) \mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

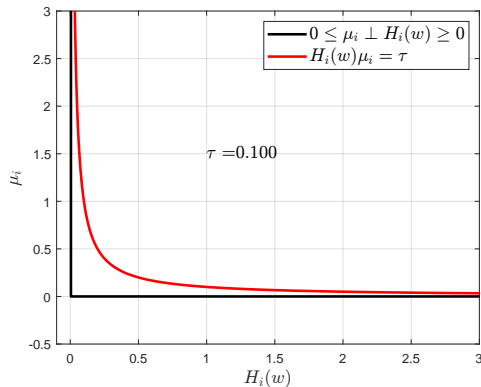
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w)^\top \mu = 0$$

$$H_i(w) \mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

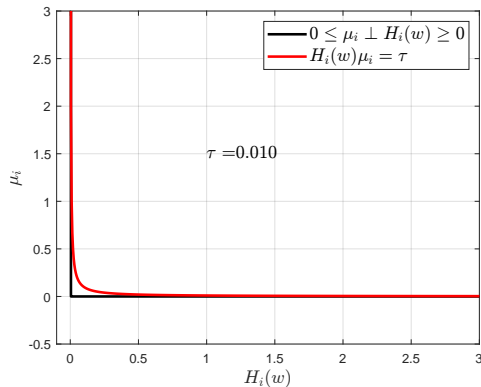
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w)^\top \mu = 0$$

$$H_i(w) \mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Barrier problem

$$\min_w F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

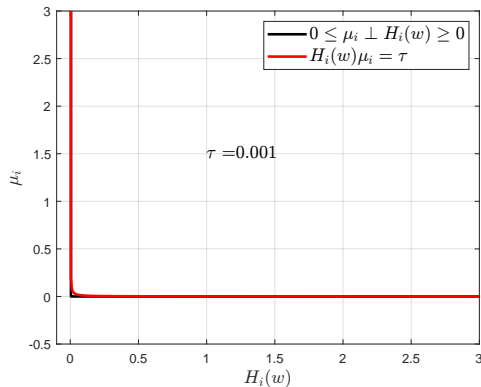
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w)^\top \mu = 0$$

$$H_i(w) \mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Primal-dual interior point method

Nonlinear programming problem

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

Smoothed KKT conditions

$$R_\tau(w, s, \lambda, \mu) = \begin{bmatrix} \nabla_w \mathcal{L}(w, \lambda, \mu) \\ G(w) \\ H(w) - s \\ \text{diag}(s)\mu - \tau e \end{bmatrix} = 0$$

$$(s, \mu > 0)$$

$$e = (1, \dots, 1)$$

Solve approximately with Newton's method for fixed τ

$$R_\tau(w, s, \lambda, \mu) + \nabla R_\tau(w, s, \lambda, \mu) \Delta z = 0$$

with $z = (w, s, \lambda, \mu)$

Line-search

Find $\alpha \in (0, 1)$

$$\begin{aligned} w^{k+1} &= w^k + \alpha \Delta w \\ s^{k+1} &= s^k + \alpha \Delta s \\ \lambda^{k+1} &= \lambda^k + \alpha \Delta \lambda \\ \mu^{k+1} &= \mu^k + \alpha \Delta \mu \end{aligned}$$

such that $s^{k+1} > 0, \mu^{k+1} > 0$

and reduce $\tau \dots$



- ▶ Newton type optimization solves the necessary optimality conditions
- ▶ Newton's method linearizes the nonlinear system in each iteration
- ▶ for constraints, need Lagrangian function, and KKT conditions
- ▶ for equalities KKT conditions are smooth, can apply Newton's method
- ▶ for inequalities KKT conditions are non-smooth, can apply Sequential Quadratic Programming (SQP)
- ▶ QPs with inequalities can be solved with interior point methods
- ▶ Also NLPs with inequalities can be solved with interior point methods (e.g. by the IPOPT solver)



- 1 Basic definitions
- 2 Some classifications of optimization problems
- 3 Optimality conditions
- 4 Nonlinear programming algorithms
- 5 Optimal control problems as nonlinear programs

Nonlinear MPC solves Nonlinear Programs (NLP)

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} c(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear MPC solves Nonlinear Programs (NLP)

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} c(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

Nonlinear MPC solves Nonlinear Programs (NLP) in (s,a) Notation



Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{s,a} \quad & \sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\ \text{s.t.} \quad & s_0 = \bar{x}_0 \\ & s_{k+1} = f(s_k, a_k) \\ & 0 \geq h(s_k, a_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(s_N) \end{aligned}$$

Variables $s = (s_0, \dots, s_N)$ and $a = (a_0, \dots, a_{N-1})$ can be summarized in vector $x = (s, a) \in \mathbb{R}^n$.

Nonlinear MPC solves Nonlinear Programs (NLP) in (s,a) Notation



Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{s,a} \quad & \sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\ \text{s.t.} \quad & s_0 = \bar{x}_0 \\ & s_{k+1} = f(s_k, a_k) \\ & 0 \geq h(s_k, a_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(s_N) \end{aligned}$$

Variables $s = (s_0, \dots, s_N)$ and $a = (a_0, \dots, a_{N-1})$ can be summarized in vector $x = (s, a) \in \mathbb{R}^n$.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & F(x) \\ \text{s.t.} \quad & G(x) = 0 \\ & H(x) \geq 0 \end{aligned}$$

Discrete Time Optimal Control Problem



$$\begin{aligned} \min_{s, a} \quad & \sum_{i=0}^{N-1} C(s_i, a_i) + E(s_N) =: F(x) \quad (\text{OC-NLP}) \\ \text{s.t.} \quad & s_{i+1} = f(s_i, a_i), \quad i=0, \dots, N-1 \\ & s_0 = \bar{s}_0 \\ & 0 \leq h(s_i, a_i) \\ & 0 \leq r(s_N) \end{aligned}$$

Definition of Functions F, G, H

IDEA: SOLVE (OC-NLP) UNMODIFIED
BY AN NLP-SOLVER
LIKE SQP, NONLIN. IP, ...

$$x = \begin{pmatrix} s_0 \\ a_0 \\ s_1 \\ a_1 \\ \vdots \\ s_{N-1} \\ a_{N-1} \\ s_N \end{pmatrix}$$

$$G(x) = \begin{bmatrix} s_0 - \bar{s}_0 \\ s_1 - f(s_0, a_0) \\ s_2 - f(s_1, a_1) \\ \vdots \\ s_N - f(s_{N-1}, a_{N-1}) \end{bmatrix}$$

$$\begin{array}{ll} \min & F(x) \\ x & \\ \text{s.t.} & G(x) = 0 \\ & H(x) \geq 0 \end{array}$$

$$G: \mathbb{R}^{(N+1)n_s + Nn_a}$$

$$\mathbb{R}^{(N+1)n_s}$$

$$H(x) = \begin{bmatrix} h(s_0, a_0) \\ \vdots \\ h(s_N) \end{bmatrix}$$



- ▶ Moritz Diehl, Sébastien Gros. "Numerical optimal control (Draft)," Lecture notes, 2019.
- ▶ Jorge Nocedal, Stephen J. Wright, Numerical optimization. New York, NY: Springer New York, 2006.