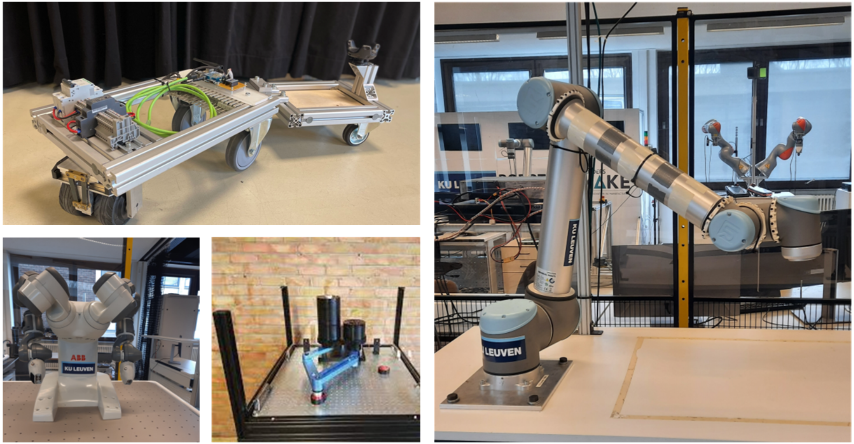
Introduction to nonlinear programming, optimal control, and model predictive control

David Kiessling
KU Leuven and Flanders Make @ KU Leuven
27/02/2025

0 Outline

1 Outline

1 We would like to control mechatronic systems



Photos by courtesy of KU Leuven and Flanders Make

1 How can we control the systems?

- Classical control: PID, flatness based control, extremum seeking etc..
- We would like to control nonlinear systems
- We have constraints that need to be satisfied, e.g., actuator constraints, obstacles, etc...
- We would like to perform optimal actions, e.g., minimal time, minimal energy, etc.

1 Continuous time optimal control problems (OCP)

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_{\mathrm{c}}(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \qquad\quad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}
$$

1 Continuous time optimal control problems (OCP)

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_{\mathrm{c}}(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \qquad\quad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}
$$

- decision variables $x(\cdot)$, $u(\cdot)$ in infinite dimensional function space
- infinitely many constraints for $t \in [0,T]$
- smooth ordinary differential equations (ODE)

$$\dot{x}(t) = f_{\mathrm{c}}(x(t),u(t))$$

1 Continuous time optimal control problems (OCP)

$$\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_\mathrm{c}(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \qquad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}$$

- decision variables $x(\cdot)$, $u(\cdot)$ in infinite dimensional function space
- infinitely many constraints for $t \in [0,T]$
- smooth ordinary differential equations (ODE)

$$\dot{x}(t) = f_\mathrm{c}(x(t),u(t))$$

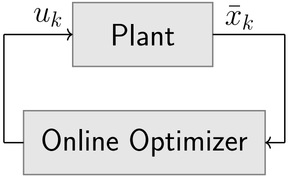
- more generally, dynamic model can be based on
  - differential algebraic equations (DAE)
  - partial differential equations (PDE)
  - stochastic ODE
- all or some components of $u(t)$ may take integer values (mixed-integer OCP)

1 Closing the control loop: Model Predictive Control

- Often the model does not fully describe the real system: ”model-plant-mismatch”
- There are external disturbances

1 Closing the control loop: Model Predictive Control

- Often the model does not fully describe the real system: "model-plant-mismatch"
- There are external disturbances



Main challenges:

- Recursive Feasibility
- Coping with computational complexity

2 Outline

2 Direct Method: “First discretize, then optimize”

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_\mathrm{c}(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \qquad\quad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}
$$

2 Direct Method: "First discretize, then optimize"

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_\mathrm{c}(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \qquad\quad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}
$$

- discretize the time:
  $[0,T] \to \{0 = t_0, t_1, \ldots, t_N = T\}$
- discretize the control, e.g., piecewise constant/linear, B-splines

2 Direct Method: "First discretize, then optimize"

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_{\mathrm{c}}(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}
$$

- discretize the time: $[0,T] \to \{0 = t_0, t_1, \ldots, t_N = T\}$
- discretize the control, e.g., piecewise constant/linear, B-splines

- discretize ODE with numerical integration method, e.g., Runge-Kutta method RK4
- choose an OCP discretization method, e.g., (direct) single shooting, multiple shooting, direct collocation
- approximate integral in objective function with a numerical integration scheme, evaluate terminal cost at grid point
- evaluate constraints at grid points

2 Comparison Continuous and Discrete Time

Continuous time OCP

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_{t=0}^{T} \ell(x(t),u(t))\,\mathrm{d}t + M(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0, \\
& \dot{x}(t) = f_c(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(t),u(t)), \quad t \in [0,T], \\
& 0 \geq c(x(T)).
\end{aligned}
$$

Discrete time multiple shooting OCP

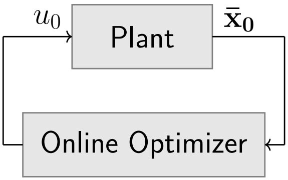
$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k,u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{x}_0, \\
& x_{k+1} = f(x_k,u_k), \quad k = 0,\dots,N-1, \\
& 0 \geq c_k(x_k,u_k) \quad k = 0,\dots,N-1, \\
& 0 \geq c_N(x_N)
\end{aligned}
$$

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

2 Using the OCP within the MPC

Discrete time OCP

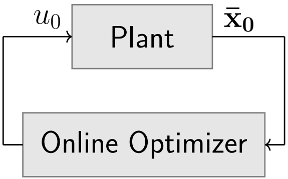
$$\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k,u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{\mathbf{x}}_\mathbf{0}, \\
& x_{k+1} = f(x_k,u_k), \quad k=0,\dots,N-1, \\
& 0 \geq c_k(x_k,u_k) \quad k=0,\dots,N-1, \\
& 0 \geq c_N(x_N)
\end{aligned}$$



2 Using the OCP within the MPC

Discrete time OCP

$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k, u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{\mathbf{x}}_0, \\
& x_{k+1} = f(x_k, u_k), \quad k = 0, \dots, N-1, \\
& 0 \geq c_k(x_k, u_k) \quad k = 0, \dots, N-1, \\
& 0 \geq c_N(x_N)
\end{aligned}
$$

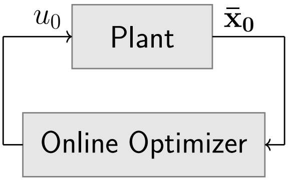


- Embed state (measurement/estimate) into OCP and solve it

2 Using the OCP within the MPC

Discrete time OCP

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k,u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{\mathbf{x}}_{\mathbf{0}}, \\
& x_{k+1} = f(x_k,u_k), \quad k = 0,\ldots,N-1, \\
& 0 \geq c_k(x_k,u_k) \quad k = 0,\ldots,N-1, \\
& 0 \geq c_N(x_N)
\end{aligned}
$$

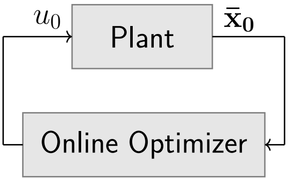


- Embed state (measurement/estimate) into OCP and solve it
- OCP solution "predicts the future"

2 Using the OCP within the MPC

Discrete time OCP

$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k,u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{\mathbf{x}}_\mathbf{0}, \\
& x_{k+1} = f(x_k,u_k), \quad k=0,\dots,N-1, \\
& 0 \geq c_k(x_k,u_k) \quad k=0,\dots,N-1, \\
& 0 \geq c_N(x_N)
\end{aligned}
$$

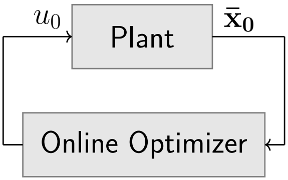


- Embed state (measurement/estimate) into OCP and solve it
- OCP solution "predicts the future"
- Apply first control u_0 to system

2 Using the OCP within the MPC

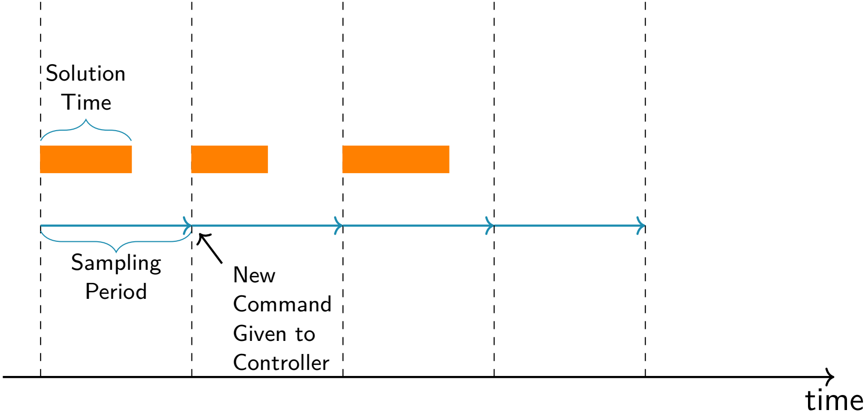
Discrete time OCP

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k,u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{\mathbf{x}}_0, \\
& x_{k+1} = f(x_k,u_k), \quad k = 0,\ldots,N-1, \\
& 0 \geq c_k(x_k,u_k) \quad k = 0,\ldots,N-1, \\
& 0 \geq c_N(x_N)
\end{aligned}
$$

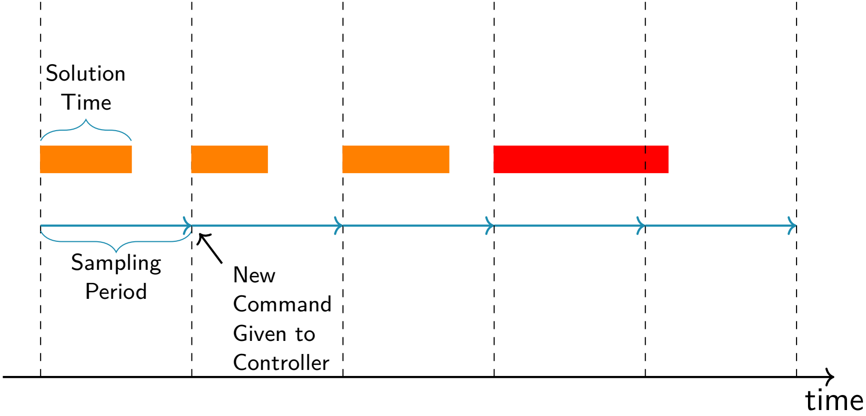


- Embed state (measurement/estimate) into OCP and solve it
- OCP solution "predicts the future"
- Apply first control u_0 to system
- Shift the time horizon by one time step and repeat the procedure

2 MPC Framework



2 MPC Framework



How can we solve the discrete-time OCPs?

3 Outline

1. Motivation
2. From Continuous to Discrete Time Optimal Control
3. Introduction to Nonlinear Programming
4. Solution Methods for Nonlinear Programs
5. Software for Solving Optimal Control Problems
6. Key Takeaways

3 Nonlinear Program

Summarizing the variables in $w = (x, u) \in \mathbb{R}^n$ and summarizing all constraints in functions yields:

Discrete time OCP

$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l_k(x_k, u_k) + M(x_N) \\
\text{s.t.} \quad & x_0 = \bar{x}_0, \\
& x_{k+1} = f(x_k, u_k), \quad k = 0, \dots, N-1, \\
& 0 \geq c_k(x_k, u_k) \quad k = 0, \dots, N-1, \\
& 0 \geq c_N(x_N).
\end{aligned}
$$

General Nonlinear Program (NLP)

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

3 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

3 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

$$\min_{w \in \mathbb{R}^n} \; F(w) \tag{1a}$$

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

$$H(w) \geq 0 \tag{1c}$$

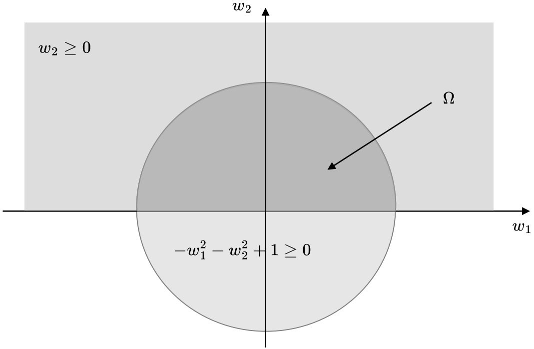
Terminology

- $w \in \mathbb{R}^n$ - decision variable
- $F : \mathbb{R}^n \to \mathbb{R}$ - objective
- $G : \mathbb{R}^n \to \mathbb{R}^{n_G}$ - equality constraints
- $H : \mathbb{R}^n \to \mathbb{R}^{n_H}$ - inequality constraints

- If F, G, H are nonlinear and smooth, we speak of a *nonlinear programming problem* (NLP).
- Only in few special cases a closed form solution exists.
- Use an iterative algorithm to find an approximate solution.
- Problem may be parametric, and some (or all) functions depend on a fixed parameter $p \in \mathbb{R}^p$, e.g. model predictive control.

3 Basic definitions: the feasible set

The feasible set of the optimization problem (2) is defined as
$\Omega = \{w \in \mathbb{R}^n \mid G(w) = 0, H(w) \geq 0\}$. A point $w \in \Omega$ is is called a feasible point.



In the example, the feasible set is the intersection of the two grey areas (halfspace and circle).

3 Basic definitions: local and global minimizer

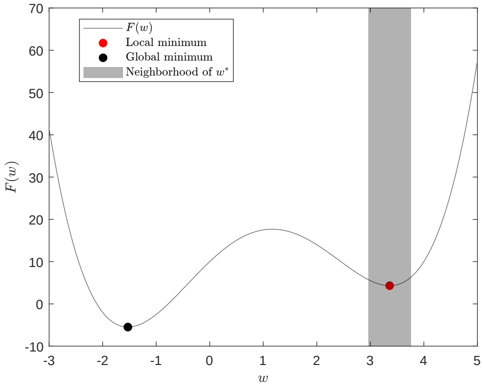
Definition (Local minimizer)

A point $w^* \in \Omega$ is called a **local minimizer** of the optimization problem (2) 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 $F(w) \geq F(w^*)$.

Definition (Global minimizer)

A point $w^* \in \Omega$ is called a **global minimizer** of (2) 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**.



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

3 The Karush-Kuhn-Tucker (KKT) conditions

NLP:

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

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

Assumptions:

- F, G, H continuously differentiable
- w^* is a (local) minimizer and a constraint qualification is satisfied

3 The Karush-Kuhn-Tucker (KKT) conditions

NLP:

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

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

Assumptions:

- F, G, H continuously differentiable
- w^* is a (local) minimizer and a constraint qualification is satisfied

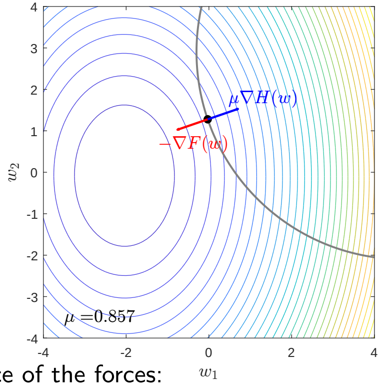
then there are unique vectors λ^* and μ^* such that (w^*,λ^*,μ^*) satisfies:

$\nabla_w\mathcal{L}(w^*,\mu^*,\lambda^*)=0,\quad \mu^*\geq 0,$	**dual feasibility**
$G(w^*)=0,\quad H(w^*)\geq 0$	**primal feasibility**
$\mu_i^* H_i(w^*)=0,\quad \forall i$	**complementary slackness**

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t.} \; H(w) \geq 0$$



Balance of the forces:

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

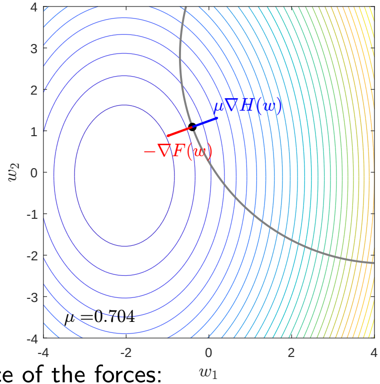
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t. } H(w) \geq 0$$

- $-\nabla F$ is the gravity



Balance of the forces:

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

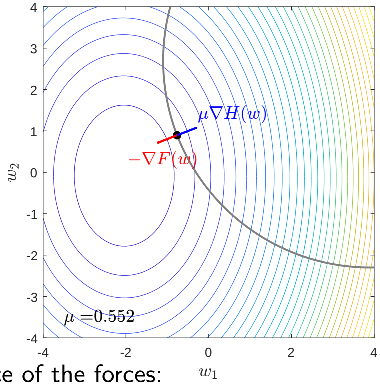
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t.} \; H(w) \geq 0$$

- $-\nabla F$ is the gravity



Balance of the forces:

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

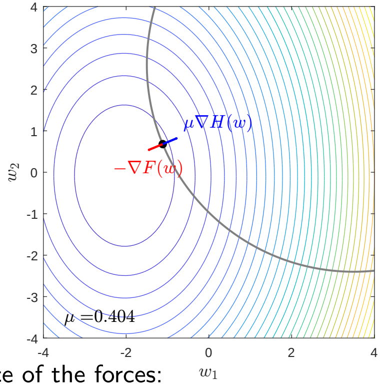
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t. } H(w) \geq 0$$

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



Balance of the forces:

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

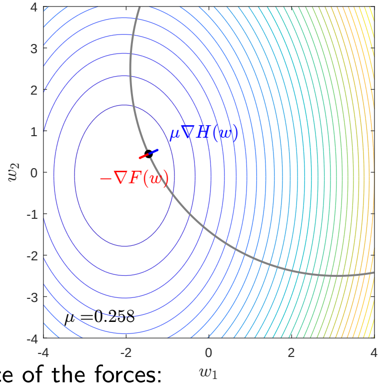
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t. } H(w) \geq 0$$

- $-\nabla F$ 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.



Balance of the forces:

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

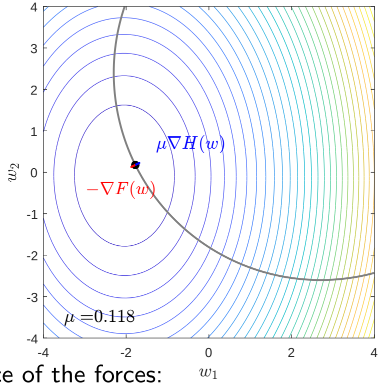
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t. } H(w) \geq 0$$

- $-\nabla F$ 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.
- active constraint: $H\,(w) = 0, \;\; \mu > 0$



Balance of the forces:

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

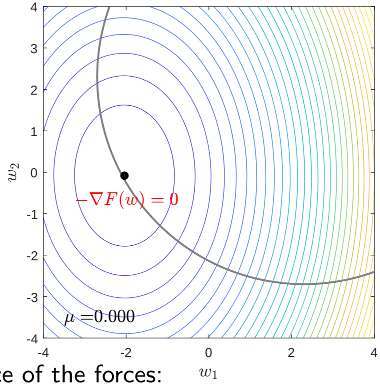
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w\in\mathbb{R}^n} \; F(w)$$
$$\text{s.t.} \; H(w) \geq 0$$

- $-\nabla F$ 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.
- active constraint: $H\,(w) = 0, \;\; \mu > 0$
- weakly active constraint: $H\,(w) = 0, \;\; \mu = 0$ the ball touches the fence but no force is needed



Balance of the forces:

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

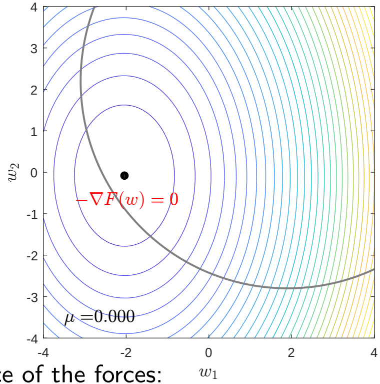
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

3 Some intuitions on the KKT conditions

Ball rolling down a valley blocked by a fence - test problem with two variables and one inequality constraint

$$\min_{w \in \mathbb{R}^n} \; F(w)$$
$$\text{s.t.}\; H(w) \geq 0$$

- $-\nabla F$ 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.
- active constraint: $H(w) = 0, \;\; \mu > 0$
- weakly active constraint: $H(w) = 0, \;\; \mu = 0$ the ball touches the fence but no force is needed
- inactive constraint: $H(w) > 0, \;\; \mu = 0$



Balance of the forces:

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

Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

4 Outline

4 Newton's method

Find $F(w) = 0$

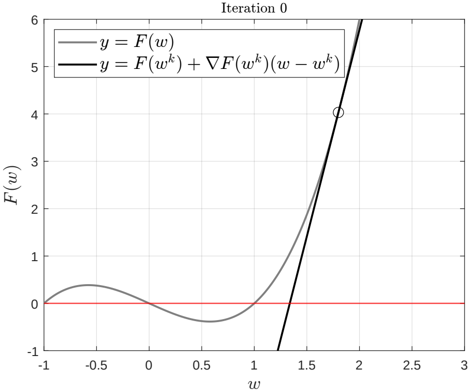
Linearization of F at linearization point $\bar{w}$

equals

First-order Taylor series at $\bar{w}$

equals

$$\boxed{\frac{\partial F}{\partial w} F_{\mathrm{L}}(w; \bar{w}) := F(\bar{w}) + \frac{\partial F}{\partial w}(\bar{w}) \quad (w - \bar{w}) \frac{\partial F}{\partial w}}$$



4 Newton's method

Find $F(w) = 0$

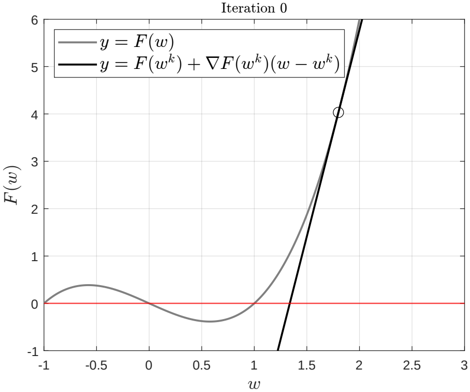
Linearization of F at linearization point $\bar{w}$

equals

First-order Taylor series at $\bar{w}$

equals

$$\boxed{\frac{\partial F}{\partial w} F_{\mathrm{L}}(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^{\top} (w - \bar{w}) \frac{\partial F}{\partial w}}$$



4 Newton's method

Find $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

equals

First-order Taylor series at $\bar{w}$

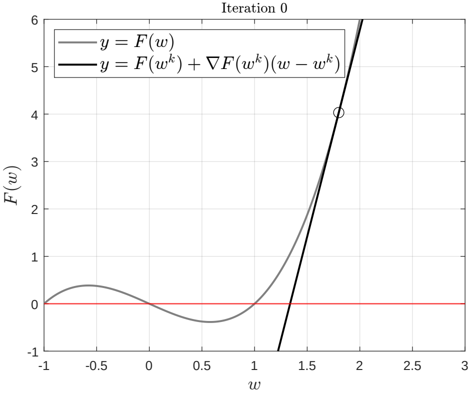
equals

$$\boxed{\frac{\partial F}{\partial w} F_{\mathrm{L}}(w;\bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w}) \frac{\partial F}{\partial w}}$$

Newton's methods, solve sequence of:

$$F(w^k) + \nabla F(w^k)^\top \Delta w = 0,$$

update $w^{k+1} = w^k + \Delta w$.
(for continuously differentiable $F : \mathbb{R}^n \to \mathbb{R}^n$)



4 Newton's method

Find $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

equals

First-order Taylor series at $\bar{w}$

equals

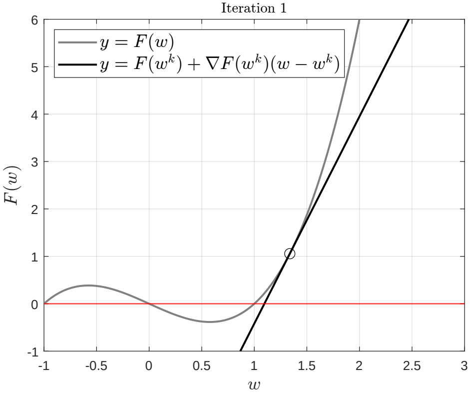
$$\boxed{\frac{\partial F}{\partial w} F_{\mathrm{L}}(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w}) \frac{\partial F}{\partial w}}$$

Newton's methods, solve sequence of:

$$F(w^k) + \nabla F(w^k)^\top \Delta w = 0,$$

update $w^{k+1} = w^k + \Delta w$.

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



4 Newton's method

Find $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

equals

First-order Taylor series at $\bar{w}$

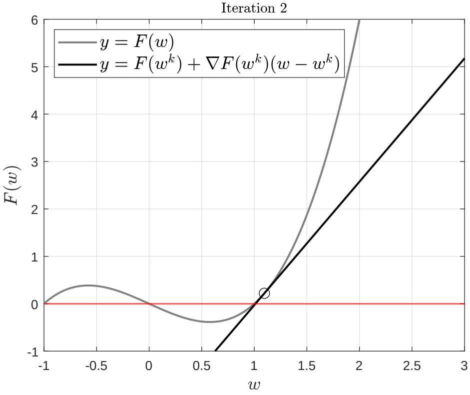
equals

$$\boxed{\frac{\partial F}{\partial w} F_{\mathrm{L}}(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w}) \frac{\partial F}{\partial w}}$$

Newton's methods, solve sequence of:

$$F(w^k) + \nabla F(w^k)^\top \Delta w = 0,$$

update $w^{k+1} = w^k + \Delta w$.
(for continuously differentiable $F : \mathbb{R}^n \to \mathbb{R}^n$)



4 Newton's method

Find $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

equals

First-order Taylor series at $\bar{w}$

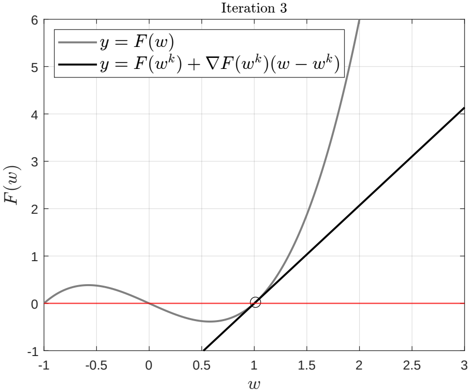
equals

$$\boxed{\frac{\partial F}{\partial w} F_{\mathrm{L}}(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w}) \frac{\partial F}{\partial w}}$$

Newton's methods, solve sequence of:

$$F(w^k) + \nabla F(w^k)^\top \Delta w = 0,$$

update $w^{k+1} = w^k + \Delta w$.
(for continuously differentiable $F : \mathbb{R}^n \to \mathbb{R}^n$)



4 Optimality conditions with inequalities

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:

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

- Last three *complementarity conditions* make the KKT conditions nonsmooth
- This system cannot be solved by plain Newton's method.

4 Methods for Solving NLPs

Sequential Quadratic Programming

- Keep the inequalities of the problem
- Locally approximate the NLP with quadratic optimization problems (QPs)
- Solve sequence of QPs (in case without inequalities equivalent to Newton's method)

Interior-Point Methods

- Smooth the complementarity conditions

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

- discard the inequality constraints
- perform Newton's method on the smoothed system
- drive τ to 0

4 Additional Features Necessary for Convergence

- SQP or IP methods provide a search direction
- This is not sufficient for Convergence
- Globalization is necessary: E.g., Line search in combination with filter and feasibility restoration
- (And often many heuristics)
- State-of-the-art general purpose interior-point solver: `IPOPT`
- State-of-the-art general purpose SQP solver: `Uno`

4 Pseudocode of an optimization algorithm

Algorithm 1: General Optimization Algorithm

Define initial guess (w_0, λ_0, μ_0);
for $k = 0, 1, 2, \ldots$ **do** `// main optimization loop`
    Evaluate functions and derivatives;
    if *algorithm converged* **then**
        stop;
    Solve subproblem (QP or primal-dual linear system);
    if *subproblem cannot be solved* **then**
        Use fallback strategy, e.g., feasibility restoration;
    for $l = 0, 1, 2, \ldots$ **do** `// globalization loop`
        Calculate trial iterate **if** *trial iterate is acceptable (to, e.g., filter)* **then**
            stop;
        Reduce step size;
return optimal solution

5 Outline

5 Structure-Exploiting OCP Solvers

- acados:
  - SQP method with line search and funnel for globalization
  - additional structure-exploiting methods available
  - https://github.com/acados/acados

5 Structure-Exploiting OCP Solvers

- `acados`:
  - SQP method with line search and funnel for globalization
  - additional structure-exploiting methods available
  - https://github.com/acados/acados

- `FATROP`:
  - inspired by `IPOPT`
  - Interior-point method with line search, filter and feasibility restoration
  - https://github.com/meco-group/fatrop

5 Structure-Exploiting OCP Solvers

- `acados`:
  - SQP method with line search and funnel for globalization
  - additional structure-exploiting methods available
  - https://github.com/acados/acados

- `FATROP`:
  - inspired by `IPOPT`
  - Interior-point method with line search, filter and feasibility restoration
  - https://github.com/meco-group/fatrop

Both solvers solve multiple shooting discretized problems, they rely on high-performance linear algebra package `BLASFEO`, and both solvers are interfaced in `IMPACT`

6 Outline

6 Key Takeaways

6 Key Takeaways

- OCPs are given in continuous time

6 Key Takeaways

- OCPs are given in continuous time
- Discretize the system ODE with a suitable integrator, explicit RK4

6 Key Takeaways

- OCPs are given in continuous time
- Discretize the system ODE with a suitable integrator, explicit RK4
- Discretize the OCP with suitable method, e.g., direct multiple shooting

6 Key Takeaways

- OCPs are given in continuous time
- Discretize the system ODE with a suitable integrator, explicit RK4
- Discretize the OCP with suitable method, e.g., direct multiple shooting
- Use a suitable solver to solve the discrete-time OCPs

6 Key Takeaways

- OCPs are given in continuous time
- Discretize the system ODE with a suitable integrator, explicit RK4
- Discretize the OCP with suitable method, e.g., direct multiple shooting
- Use a suitable solver to solve the discrete-time OCPs
- MPC requires to solve OCPs ”in the loop”

6 Key Takeaways

- OCPs are given in continuous time
- Discretize the system ODE with a suitable integrator, explicit RK4
- Discretize the OCP with suitable method, e.g., direct multiple shooting
- Use a suitable solver to solve the discrete-time OCPs
- MPC requires to solve OCPs "in the loop"

IMPACT allows easy modelling, built-in discretization techniques, interfaces tool state-of-the-art solver AND

6 Key Takeaways

- OCPs are given in continuous time
- Discretize the system ODE with a suitable integrator, explicit RK4
- Discretize the OCP with suitable method, e.g., direct multiple shooting
- Use a suitable solver to solve the discrete-time OCPs
- MPC requires to solve OCPs "in the loop"

IMPACT allows easy modelling, built-in discretization techniques, interfaces tool state-of-the-art solver AND easy-deployment

Thank you very much for your attention!

This work was supported by the Flanders Make SBO project DIRAC: Deterministic and Inexpensive Realizations of Advanced Control. The slides were prepared with help from Prof. Jan Swevers and Dr. Wilm Decré. Some material of the slides was provided by Prof. Moritz Diehl, and Dr. Armin Nurkanovíc.