Efficient algorithms and differentiable solutions for optimal control

Jonathan Frey

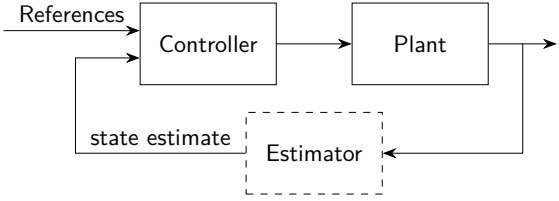
Systems Control and Optimization Laboratory (syscop)
Department of Microsystems Engineering (IMTEK)
University Freiburg

PhD defense
October 15, 2026

universität freiburg

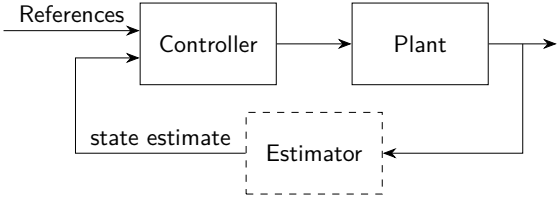
Motivation

State-feedback control loop



Motivation

State-feedback control loop



Optimal control problem (OCP)

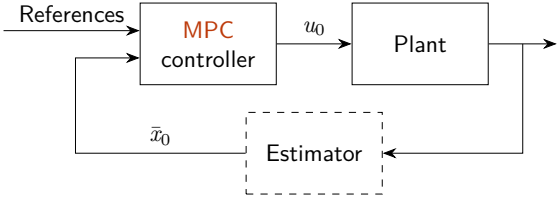
$$\begin{aligned} \underset{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, u_i) + M(x_N) \\ \text{subject to} \quad & \bar{x}_0 = x(0), \\ & x_{i+1} = \phi_i(x_i, u_i), \quad i = 0, \dots, N-1, \\ & 0 \geq g_i(x_i, u_i), \quad i = 0, \dots, N-1 \end{aligned}$$

Optimization problem with:

- States x_i and controls u_i on horizon N
- Desired behavior: cost L_i, M
- System knowledge: dynamics model ϕ_i
- Requirements: constraints g_i

Motivation

State-feedback control loop



Model predictive control (MPC)

- Solve OCP (approximately)
- Apply optimal solution $u_0(\bar{x}_0)$
- Get new state-estimate $\bar{x}_0$

Needs:

- Fast, embedded OCP solvers

Optimal control problem (OCP)

$$\begin{aligned} \underset{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, u_i) + M(x_N) \\ \text{subject to} \quad & \bar{x}_0 = x(0), \\ & x_{i+1} = \phi_i(x_i, u_i), \quad i = 0, \dots, N-1, \\ & 0 \geq g_i(x_i, u_i), \quad i = 0, \dots, N-1 \end{aligned}$$

Optimization problem with:

- States x_i and controls u_i on horizon N
- Desired behavior: cost L_i, M
- System knowledge: dynamics model ϕ_i
- Requirements: constraints g_i

acados – fast embedded solvers for nonlinear optimal control



- 🕰️ History: 2016 – now
- 🔓 Open-source, permissive license
- 🔌 Embeddable: Minimal dependencies
- ⚡ Fast:
  - Written in `C`
  - High-performance linear algebra via `BLASFEO`
  - Exploits OCP structure

acados – fast embedded solvers for nonlinear optimal control

acados

- History: 2016 – now
- Open-source
- Embeddable
- Fast

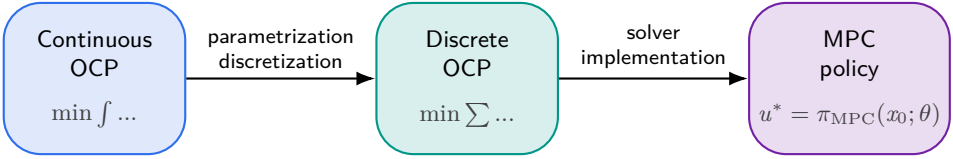
Application references

Romero-2022, Saviolo-2024, Salzmann-2023, Reiter-2025, Schimpe-2022, Reiter-2023, Reiter-2024, Dong-2024, Koehler-2021, Loew-2020, Gordon-2022, Carlos-2020, Carlos-2020, Zanelli-2021, Cecchin-2023, Rathod-2021, Gao-2024, Subash-2024, Weiss-2025, Winkler-2021, Homburger-2025, Homburger-2026, Koehler-2025, Reiter-2025, Prochazka-2026 Ge-2026, Prochazka-2026, Wuest-2024, Taouil-2026

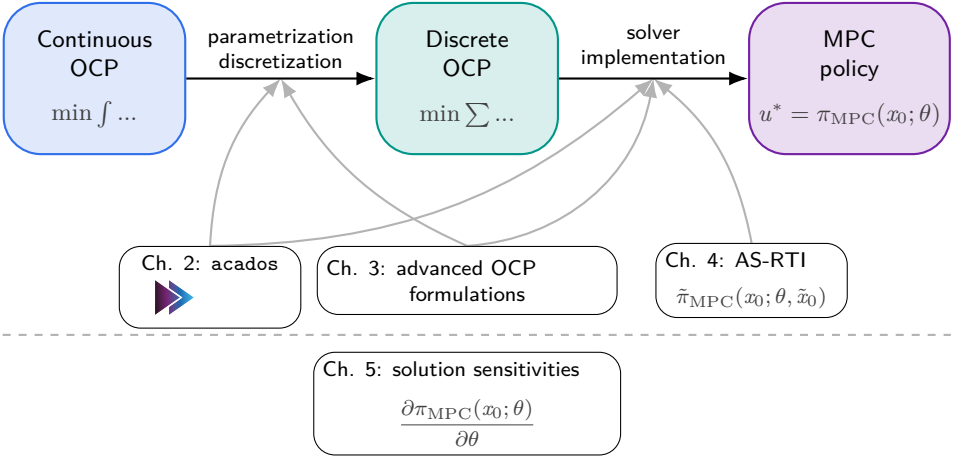
- Enabled & enhanced real-world MPC in various fields, academia & industry



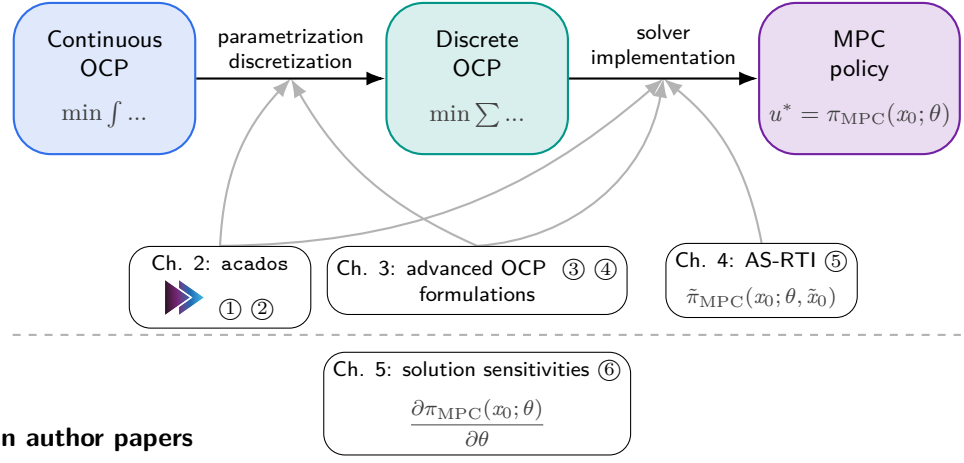
Thesis Overview



Thesis Overview



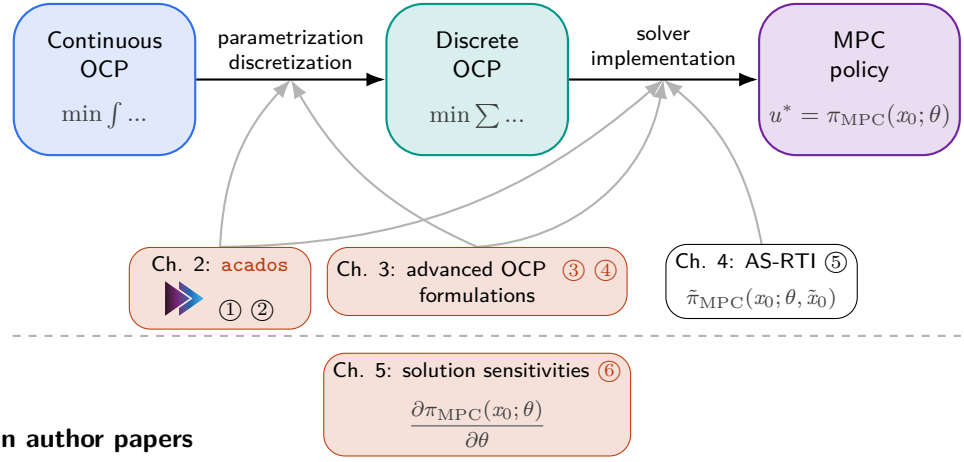
Thesis Overview



Main author papers

① MPC'21 ② ECC'23 ③ EJC'24 ④ OCAM'25 ⑤ LCSS'24
⑥ Neurips'26 ⑦ IFAC'20 ⑧ ECC'24 ⑨ IFAC'26

Thesis Overview



Main author papers

① MPC'21 ② ECC'23 ③ EJC'24 ④ OCAM'25 ⑤ LCSS'24
⑥ Neurips'26 ⑦ IFAC'20 ⑧ ECC'24 ⑨ IFAC'26

This talk.

OCP discretization

$$
\begin{aligned}
&\underset{x(\cdot),v(\cdot)}{\text{minimize}} && \int_0^\infty \ell(x(t), v(t))\,\mathrm{d}t \\
&\text{subject to} && \bar{x}_0 = x(0), \\
& && 0 = f^{\text{impl}}(t, \dot{x}(t), x(t), v(t)), && t \in [0, \infty), \\
& && 0 \geq g(x(t), v(t)), && t \in [0, \infty),
\end{aligned}
$$

OCP discretization

$$
\begin{aligned}
\underset{x(\cdot),v(\cdot)}{\text{minimize}} \quad & \int_0^T \ell(x(t), v(t))\,\mathrm{d}t + M(x(T)) \\
\text{subject to} \quad & \bar{x}_0 = x(0), \\
& 0 = f^{\text{impl}}(t, \dot{x}(t), x(t), v(t)), \qquad t \in [0, T), \\
& 0 \geq g(x(t), v(t)), \qquad t \in [0, T), \\
& 0 \geq g_{\text{e}}(x(T)),
\end{aligned}
$$

- Select finite time horizon T, terminal cost M and constraints g_{e}

OCP discretization

$$
\begin{aligned}
\underset{\substack{x(\cdot),\\ u_0,\dots,u_{N-1}}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \ell(x(t), u_i)\,\mathrm{d}t + M(x(T)) \\
\text{subject to} \quad & \bar{x}_0 = x(0), \\
& 0 = f^{\text{impl}}(t, \dot{x}(t), x(t), u_i), \qquad && t \in [t_i, t_{i+1}), \\
& 0 \geq g(x(t), u_i), && t \in [t_i, t_{i+1}), \\
& 0 \geq g_{\text{e}}(x(T))
\end{aligned}
$$

- Select finite time horizon T, terminal cost M and constraints g_{e}
- Time grid $0 = t_0 < \ldots < t_N = T$
- Control parametrization via $u_i = v(t)$ for $t \in [t_i, t_{i+1})$

OCP discretization

$$
\begin{aligned}
\underset{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, u_i) + M(x_N) \\
\text{subject to} \quad & \bar{x}_0 = x_0, \\
& x_{i+1} = \phi_i(x_i, u_i), && i = 0, \ldots, N-1, \\
& 0 \geq g_i(x_i, u_i), && i = 0, \ldots, N-1, \\
& 0 \geq g_{\mathrm{e}}(x_N)
\end{aligned}
$$

- Select finite time horizon T, terminal cost M and constraints g_{e}
- Time grid $0 = t_0 < \ldots < t_N = T$
- Control parametrization via $u_i = v(t)$ for $t \in [t_i, t_{i+1})$
- Discrete state trajectory x_i with dynamics ϕ_i – usually high-order integrator
- Cost L_i
- Constraints g_i

OCP discretization

$$
\begin{aligned}
\underset{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, u_i) + M(x_N) \\
\text{subject to} \quad & \bar{x}_0 = x_0, \\
& x_{i+1} = \phi_i(x_i, u_i), && i = 0, \ldots, N-1, \\
& 0 \geq g_i(x_i, u_i), && i = 0, \ldots, N-1, \\
& 0 \geq g_{\mathrm{e}}(x_N)
\end{aligned}
$$

- Select finite time horizon T, terminal cost M and constraints g_{e}
- Time grid $0 = t_0 < \ldots < t_N = T$
- Control parametrization via $u_i = v(t)$ for $t \in [t_i, t_{i+1})$
- Discrete state trajectory x_i with dynamics ϕ_i – usually high-order integrator
- Cost L_i – usually first-order shooting node discretization
- Constraints g_i – usually enforced at shooting nodes only

Cost discretization

$$L_i(x_i, u_i) \approx \int_{t_i}^{t_{i+1}} \ell(x(t), u_i) \mathrm{d}t.$$

Options

1. Explicit Euler shooting node (SN) cost discretization

$$L_i^{\mathrm{SN}}(\cdot) = \Delta t_i \ell(x_i, u_i),$$

Cost discretization

$$L_i(x_i, u_i) \approx \int_{t_i}^{t_{i+1}} \ell(x(t), u_i) \mathrm{d}t.$$

Options

1. Explicit Euler shooting node (SN) cost discretization

$$L_i^{\mathrm{SN}}(\cdot) = \Delta t_i \ell(x_i, u_i),$$

2. High-order implicit Runge-Kutta (RK) – like dynamics ϕ_i
   - ✅ more accurate – especially relevant for long intervals
   - ❌ additional computational cost

Cost discretization

$$L_i(x_i, u_i) \approx \int_{t_i}^{t_{i+1}} \ell(x(t), u_i) \mathrm{d}t.$$

Options

1. Explicit Euler shooting node (SN) cost discretization

$$L_i^{\mathrm{SN}}(\cdot) = \Delta t_i \ell(x_i, u_i),$$

2. High-order implicit Runge-Kutta (RK) – like dynamics ϕ_i
   - ✅ more accurate – especially relevant for long intervals
   - ❌ additional computational cost

Gauss–Newton Runge–Kutta (GNRK) integration (Frey, Baumgärtner, & Diehl, 2024)

- Cost with outer convex-structure
- Gauss-Newton-type Hessian approximation

Cost discretization

$$L_i(x_i, u_i) \approx \int_{t_i}^{t_{i+1}} \ell(x(t), u_i) \mathrm{d}t.$$

Options

1. Explicit Euler shooting node (SN) cost discretization

$$L_i^{\mathrm{SN}}(\cdot) = \Delta t_i \ell(x_i, u_i),$$

2. High-order implicit Runge-Kutta (RK) – like dynamics ϕ_i
   - ✅ more accurate – especially relevant for long intervals
   - ❌ additional computational cost

Gauss–Newton Runge–Kutta (GNRK) integration (Frey, Baumgärtner, & Diehl, 2024)

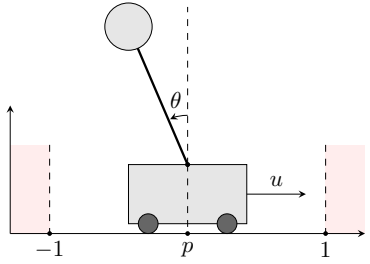
- Cost with outer convex-structure
- Gauss-Newton-type Hessian approximation

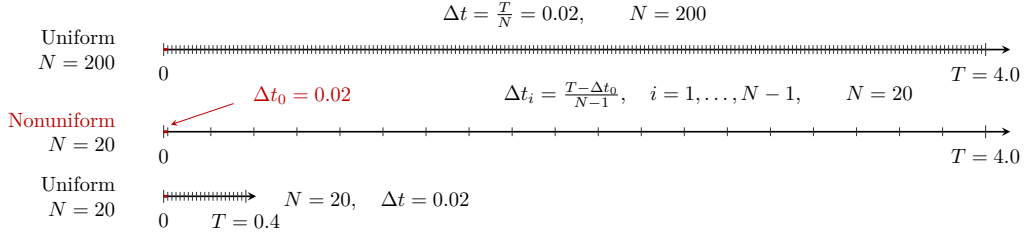
✅ Positive definite, optimization friendly Hessian
✅ Small computational cost
✅ Handle constraints reformulated with smooth, convex penalties

Gauss–Newton Runge–Kutta integration

Benchmark

- Cartpole $x = [p, \theta, s, \omega]^\top$, $u \in [-40, 40]$
- Cost $\ell_{\text{pend}}(x, u) = x^\top Q x + u^\top R u + \rho(x)$
- L_2 constraint penalty ρ for bounds $p \in [-1, 1]$
- Closed-loop MPC with $\Delta t_{\text{plant}} = 0.02\text{s}$
- Various acados controllers

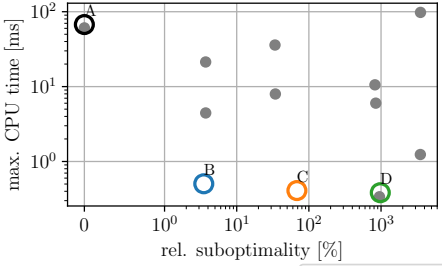


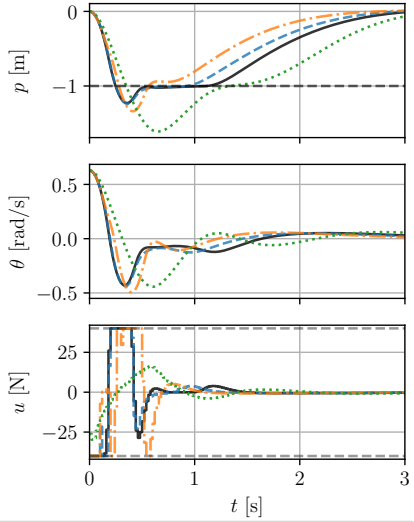


Gauss–Newton Runge–Kutta integration

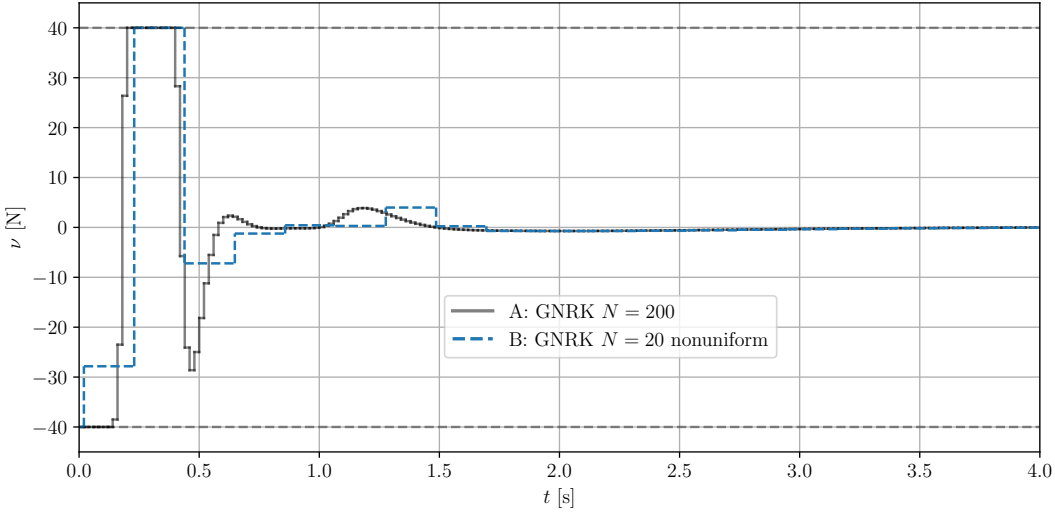
Observations & Takeaways

- $N = 20$: B, C, D: similar runtimes
  - ⚡ GNRK little extra computations
- D: Uniform grid: suboptimal, large violations
  - 💡 nonuniform grids can improve performance
- B: GNRK beats C: baseline
  - 🎯 accuracy gain



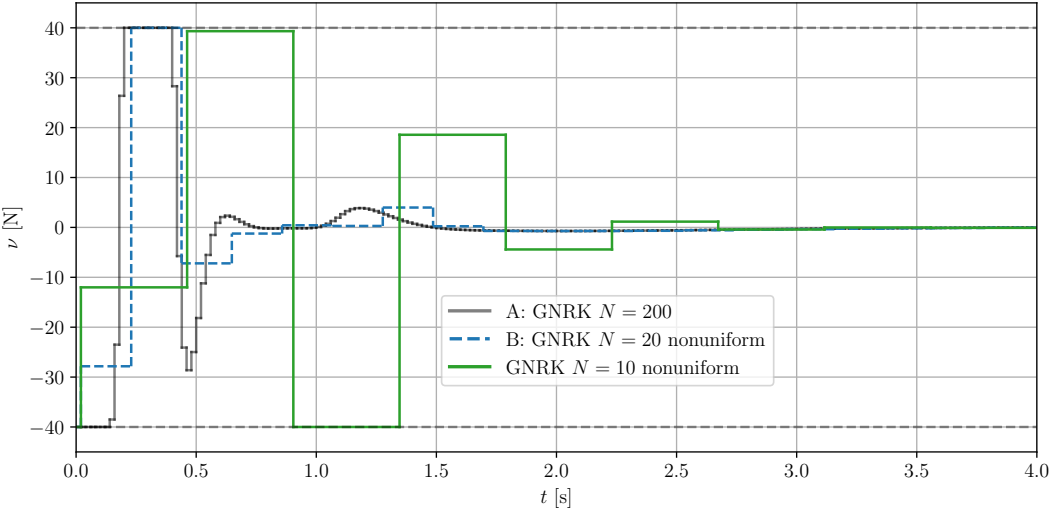


Open-loop approximations – the MPC "plan"



Similar approximation quality with less shooting interval possible?

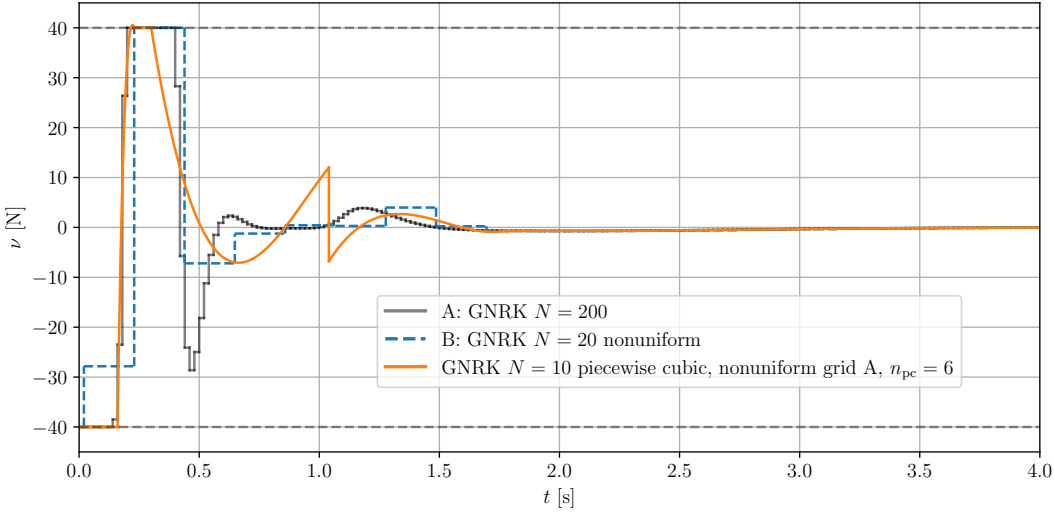
Open-loop approximations – the MPC "plan"



⚠️ On large intervals: dynamic control behavior cannot be represented with constant controls

🎯 Despite accurate cost integration.

Open-loop approximations – the MPC "plan"



⚠ On large intervals: dynamic control behavior cannot be represented with constant controls

💡 Piecewise polynomial control parametrizations enable better approximations.

Multi-phase OCP formulations (Frey et al., 2025)

$$
\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\eta}}{\text{minimize}} \quad & \sum_{k=1}^{M} \sum_{j=0}^{N_k-1} L_{k,j}(x_{k,j}, u_{k,j}, \Delta t_{k,j}) + E_k(x_{k,N_k}, \eta_k) \\
\text{subject to} \quad & x_{1,0} = \bar{x}_0, \\
& x_{k+1,0} = \Gamma_k(x_{k,N_k}, \eta_k), \quad k = 1, \dots M-1, \\
& x_{k,j+1} = \phi_{k,j}(x_{k,j}, u_{k,j}), \ j = 0, \dots, N_k - 1, k = 1, \dots M, \\
& 0 \geq g_{k,j}(x_{k,j}, u_{k,j}), \ j = 0, \dots, N_k - 1, k = 1, \dots M, \\
& 0 \geq g_{\mathrm{e}}(x_{M,N_M}),
\end{aligned}
$$

- M phases with "standard" OCPs with horizon lengths N_k
- Transition functions Γ_k, discrete decision variables η_k

Multi-phase OCP formulations (Frey et al., 2025)

$$
\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\eta}}{\text{minimize}} \quad & \sum_{k=1}^{M} \sum_{j=0}^{N_k-1} L_{k,j}(x_{k,j}, u_{k,j}, \Delta t_{k,j}) + E_k(x_{k,N_k}, \eta_k) \\
\text{subject to} \quad & x_{1,0} = \bar{x}_0, \\
& x_{k+1,0} = \Gamma_k(x_{k,N_k}, \eta_k), \quad k = 1, \dots M-1, \\
& x_{k,j+1} = \phi_{k,j}(x_{k,j}, u_{k,j}), \; j = 0, \dots, N_k - 1, k = 1, \dots M, \\
& 0 \geq g_{k,j}(x_{k,j}, u_{k,j}), \; j = 0, \dots, N_k - 1, k = 1, \dots M, \\
& 0 \geq g_{\mathrm{e}}(x_{M,N_M}),
\end{aligned}
$$

- M phases with "standard" OCPs with horizon lengths N_k
- Transition functions Γ_k, discrete decision variables η_k
- Structurally different phases k
  - State and control representations & dimension $x_{k,j} \in \mathbb{R}^{n_{\mathrm{x},k}}$, $u_{k,j} \in \mathbb{R}^{n_{\mathrm{u},k}}$
  - Functions $L_{k,j}$, E_k, $\phi_{k,j}$, $g_{k,j}$

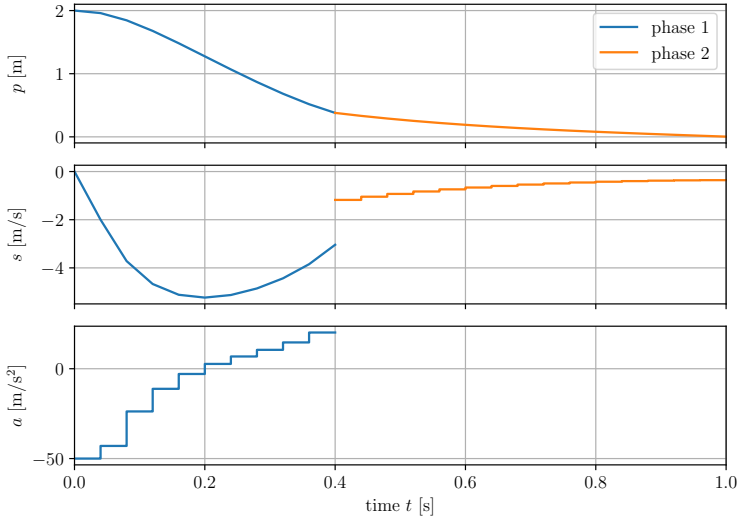
Multi-phase OCP formulations (Frey et al., 2025)

$$
\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\eta}}{\text{minimize}} \quad & \sum_{k=1}^{M} \sum_{j=0}^{N_k-1} L_{k,j}(x_{k,j}, u_{k,j}, \Delta t_{k,j}) + E_k(x_{k,N_k}, \eta_k) \\
\text{subject to} \quad & x_{1,0} = \bar{x}_0, \\
& x_{k+1,0} = \Gamma_k(x_{k,N_k}, \eta_k), \quad k = 1, \dots M-1, \\
& x_{k,j+1} = \phi_{k,j}(x_{k,j}, u_{k,j}), \; j = 0, \dots, N_k - 1, k = 1, \dots M, \\
& 0 \geq g_{k,j}(x_{k,j}, u_{k,j}), \; j = 0, \dots, N_k - 1, k = 1, \dots M, \\
& 0 \geq g_{\mathrm{e}}(x_{M,N_M}),
\end{aligned}
$$

- M phases with "standard" OCPs with horizon lengths N_k
- Transition functions Γ_k, discrete decision variables η_k
- Structurally different phases k
  - State and control representations & dimension $x_{k,j} \in \mathbb{R}^{n_{\mathrm{x},k}}$, $u_{k,j} \in \mathbb{R}^{n_{\mathrm{u},k}}$
  - Functions $L_{k,j}, E_k, \phi_{k,j}, g_{k,j}$
- Same structure: Discrete dynamics $\phi_{k,j}$ and transitions Γ_k
- Structure-exploiting algorithms of acados are applicable

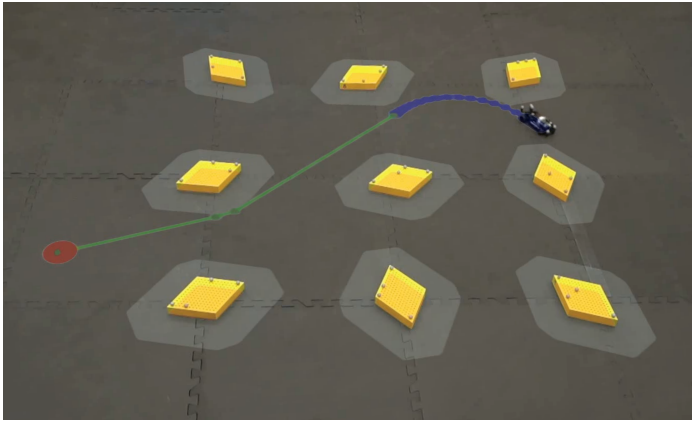
Multi-phase application examples

- Models of different fidelity for different parts of the horizon



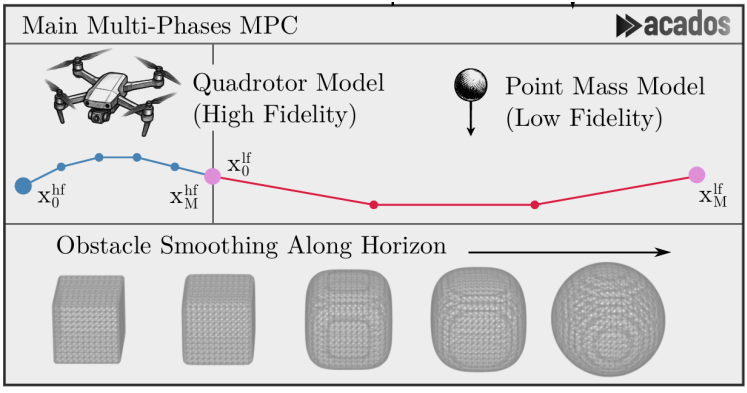
Multi-phase application examples

- Models of different fidelity for different parts of the horizon
  - Combining trajectory optimization and path planning.
    For mobile robots (Köhler et al., 2025) and drones (Reiter et al., 2025)



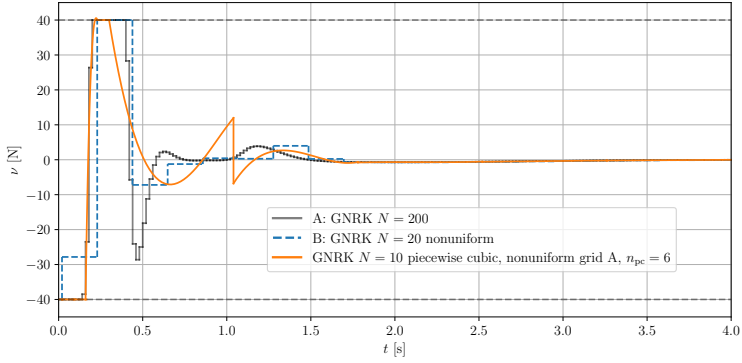
Multi-phase application examples

- Models of different fidelity for different parts of the horizon
  - Combining trajectory optimization and path planning.
    For mobile robots (Köhler et al., 2025) and drones (Reiter et al., 2025)



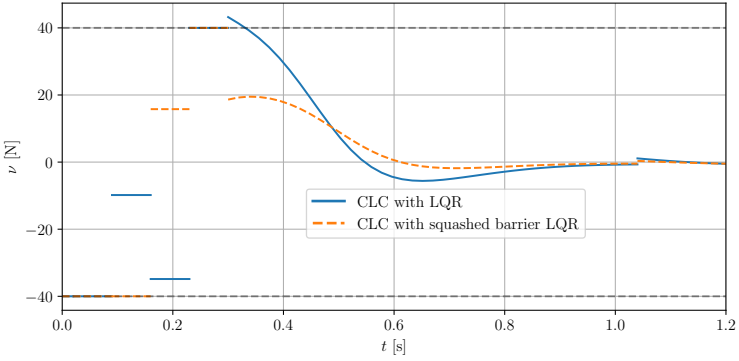
Multi-phase application examples

- Models of different fidelity for different parts of the horizon
- Piecewise polynomial control parametrization of varying degree



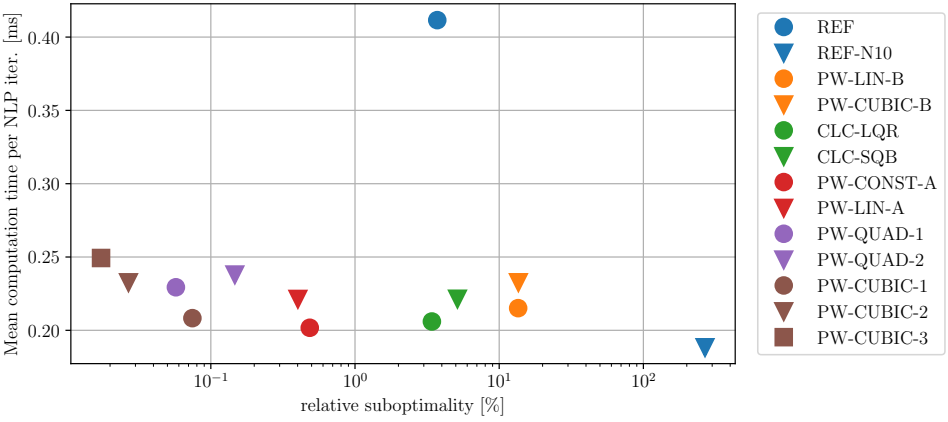
Multi-phase application examples

- Models of different fidelity for different parts of the horizon
- Piecewise polynomial control parametrization of varying degree
- Closed-loop costing (CLC): Second horizon with policy $u = \kappa(x)$.

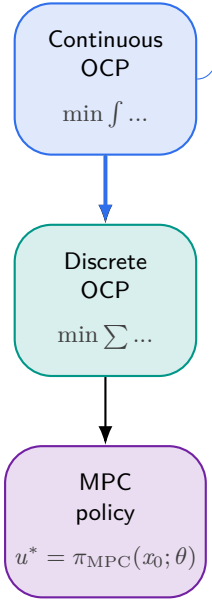


Revisiting the pendulum benchmark

Pareto-optimal MOCP-based controllers



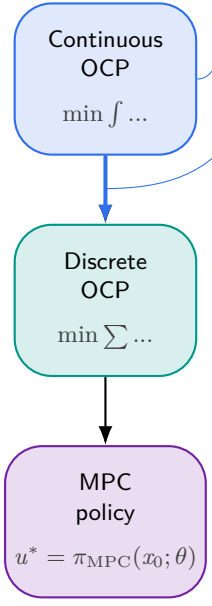
From formulation to solver with acados



Definition in high-level language

- Python / MATLAB / Octave
- CasADi for nonlinear expressions
- Problem classes: Single- or multi-phase OCP, simulation, estimation

From formulation to solver with acados

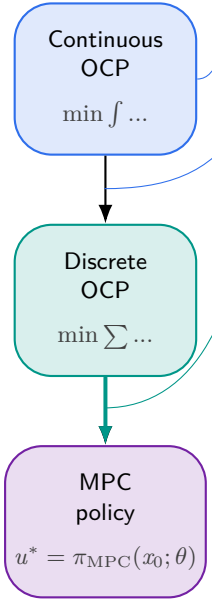


Definition in high-level language

Integration of model dynamics

- Integrators $f^{\mathrm{impl}}(\dot{x}, x, u) = 0$ ➡️ $x_+ = \phi(x, u)$
  - implicit & explicit Runge-Kutta (IRK, ERK)
  - structure-exploiting GNSF
  - standalone
  - efficient sensitivity propagation

From formulation to solver with acados



Definition in high-level language

Integration of model dynamics

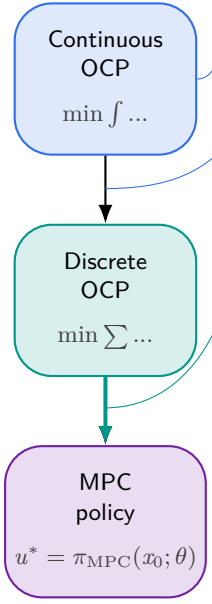
$f^{\mathrm{impl}}(\dot{x}, x, u) = 0 \quad \Rightarrow \quad x_+ = \phi(x, u)$

Solver specification / design

NLP solvers:

- Sequential quadratic programming (SQP)
- Differential dynamic programming (DDP)
- Byrd-Omojokun SQP
- (Advanced-step) real-time iteration (AS-RTI)

From formulation to solver with acados



Definition in high-level language

Integration of model dynamics

$f^{\mathrm{impl}}(\dot{x}, x, u) = 0 \quad \Rightarrow \quad x_+ = \phi(x, u)$

Solver specification / design

- NLP solvers: SQP, DDP, (AS)-RTI
- Leveraging submodules:
  - Linearization
    - Cost, constraint, dynamics modules
    - Exact Hessian or Gauss-Newton-type

From formulation to solver with acados

Continuous OCP

$\min \int \ldots$

Discrete OCP

$\min \sum \ldots$

MPC policy

$u^* = \pi_{\mathrm{MPC}}(x_0; \theta)$

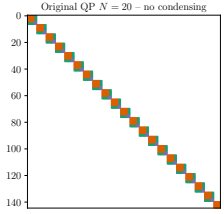
Definition in high-level language

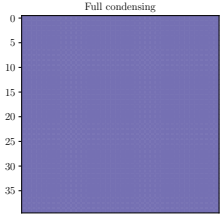
Integration of model dynamics

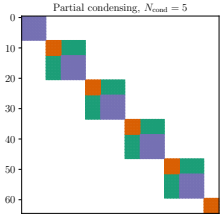
$f^{\mathrm{impl}}(\dot{x}, x, u) = 0 \quad \Rightarrow \quad x_+ = \phi(x, u)$

Solver specification / design

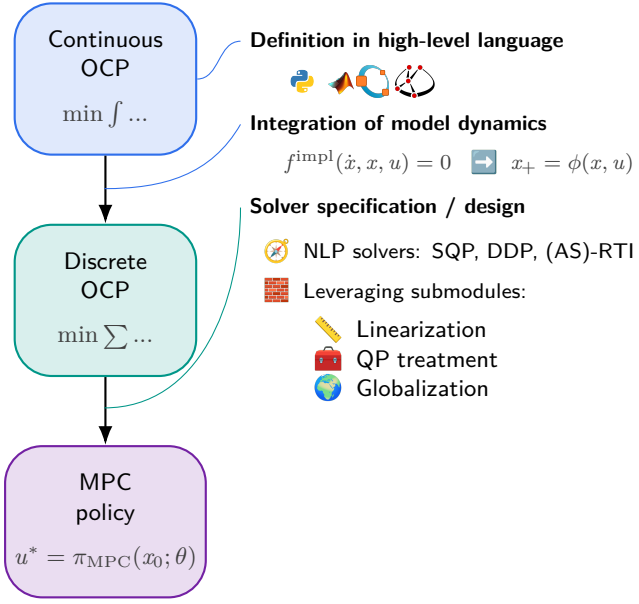
- NLP solvers: SQP, DDP, (AS)-RTI
- Leveraging submodules:
  - Linearization
  - QP treatment
    - Structured formulation via `HPIPM`
    - Full and partial condensing via `HPIPM`
    - Scaling, regularization
    - QP solvers: `HPIPM`, `DAQP`, `qpOASES`, `OSQP`, `Clarabel`



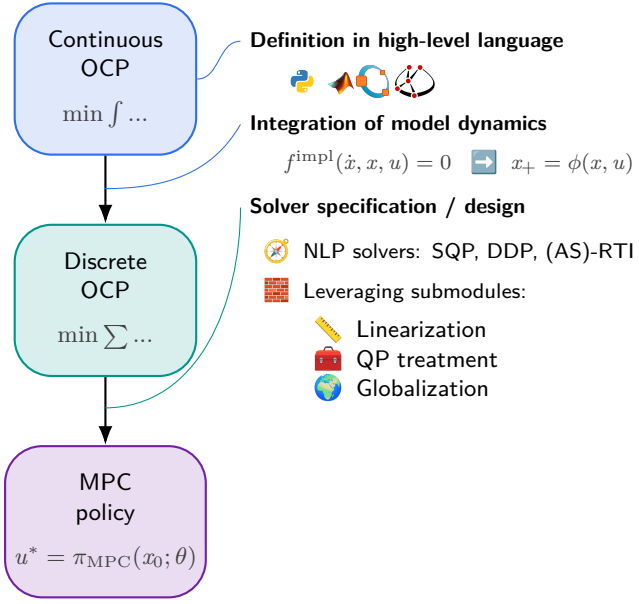




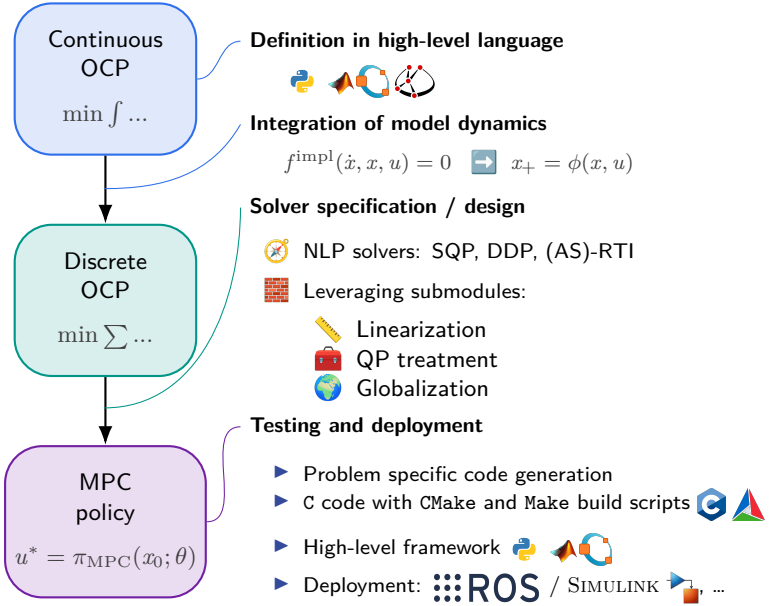
From formulation to solver with acados



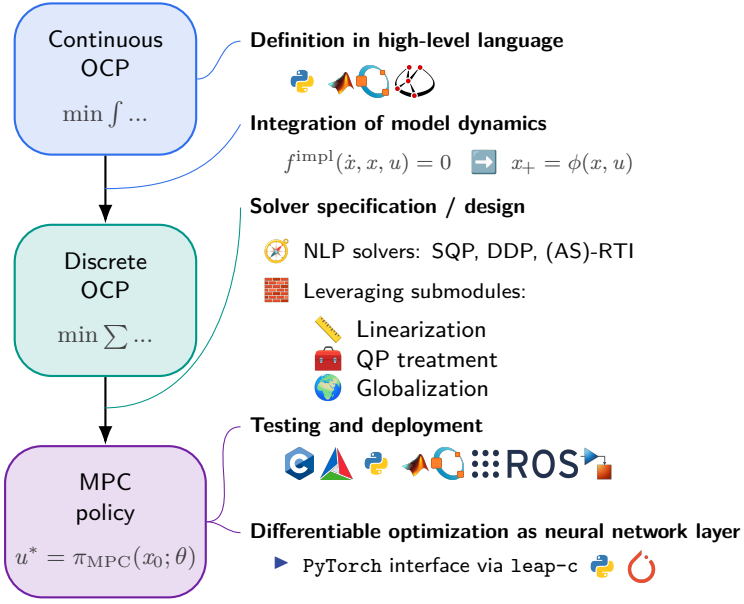
From formulation to solver with acados



From formulation to solver with acados

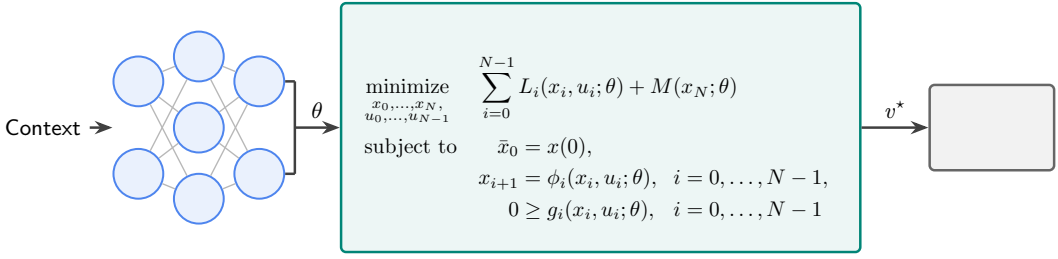


From formulation to solver with acados



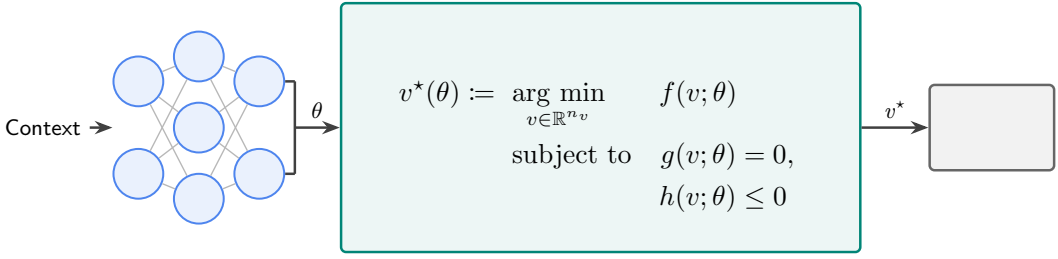
Differentiable nonlinear MPC – Intro

Motivation: Optimization solver as differentiable layer in neural networks



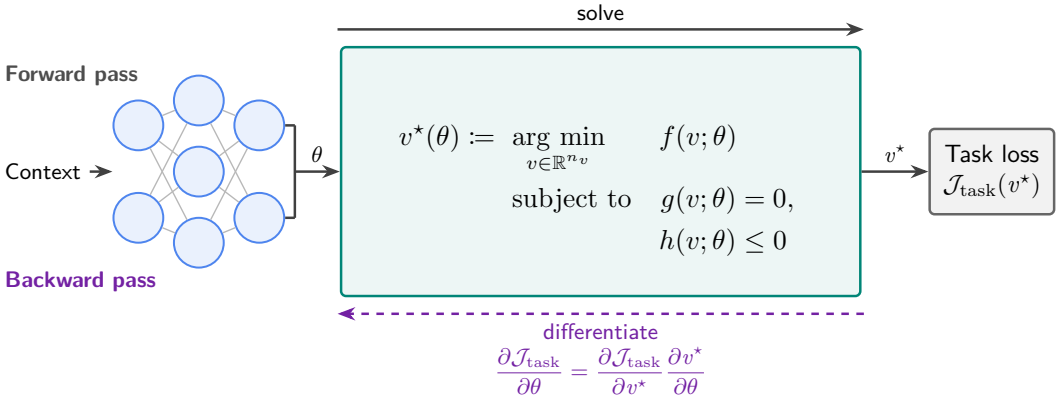
Differentiable nonlinear MPC – Intro

Motivation: Optimization solver as differentiable layer in neural networks



Differentiable nonlinear MPC – Intro

Motivation: Optimization solver as differentiable layer in neural networks



Simple dense NLP example

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & (x-\theta^2)^2 \\ \text{subject to} \quad & -1 \leq x \leq 1, \end{aligned}$$

Nondifferentiable solution map

$$x^\star(\theta) = \begin{cases} \theta^2, \text{ if } \theta \in [-1,1] \\ 1, \text{ otherwise} \end{cases}$$

Derivative

$$\partial_\theta x^\star(\theta) = \begin{cases} 2\cdot\theta, \text{ if } \theta \in (-1,1) \\ 0, \text{ if } |\theta| > 1 \\ \text{not defined, for } \theta \in -1, 1 \end{cases}$$

Simple dense NLP example

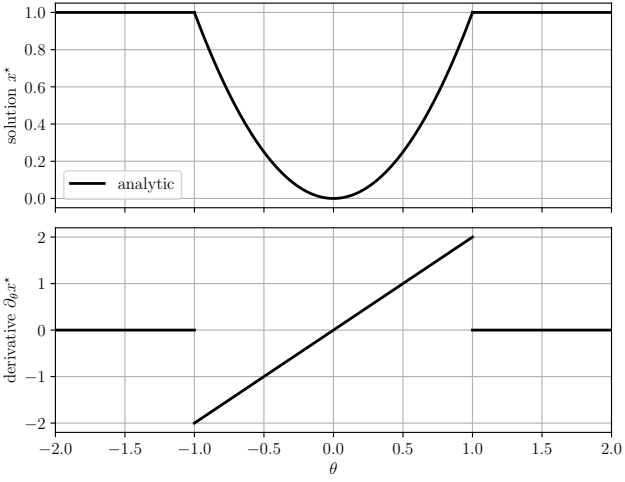
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & (x - \theta^2)^2 \\ \text{subject to} \quad & -1 \leq x \leq 1, \end{aligned}$$

Nondifferentiable solution map

$$x^\star(\theta) = \begin{cases} \theta^2, \text{ if } \theta \in [-1, 1] \\ 1, \text{ otherwise} \end{cases}$$

Derivative

$$\partial_\theta x^\star(\theta) = \begin{cases} 2 \cdot \theta, \text{ if } \theta \in (-1, 1) \\ 0, \text{ if } |\theta| > 1 \\ \text{not defined, for } \theta \in -1, 1 \end{cases}$$



Barrier formulations & SBQP

Exact NLP

$$
\begin{aligned}
v^\star(\theta) := \quad & \underset{z \in \mathbb{R}^{n_z}}{\arg\min} && f(z;\theta) \\
& \text{subject to} && g(v;\theta) = 0, \\
& && h(v;\theta) \leq 0
\end{aligned}
$$

Barrier formulations & SBQP

Exact NLP

$$
\begin{aligned}
v^\star(\theta) := \quad & \underset{z \in \mathbb{R}^{n_z}}{\arg\min} && f(z;\theta) \\
& \text{subject to} && g(v;\theta) = 0, \\
& && h(v;\theta) \leq 0
\end{aligned}
$$

Barrier NLP

$$
\begin{aligned}
& \underset{v \in \mathbb{R}^{n_v}}{\text{minimize}} && f(v;\theta) - \tau \sum_{i=1}^{n_h} \log(-h_i(v;\tau)) \\
& \text{subject to} && g(v;\theta) = 0
\end{aligned}
$$

- Interior-point methods (IPM) solve barrier NLP for $\tau \to 0$.
- IPM with $\tau \to \tau_{\min} > 0$ solves barrier NLP.

Simple dense NLP example

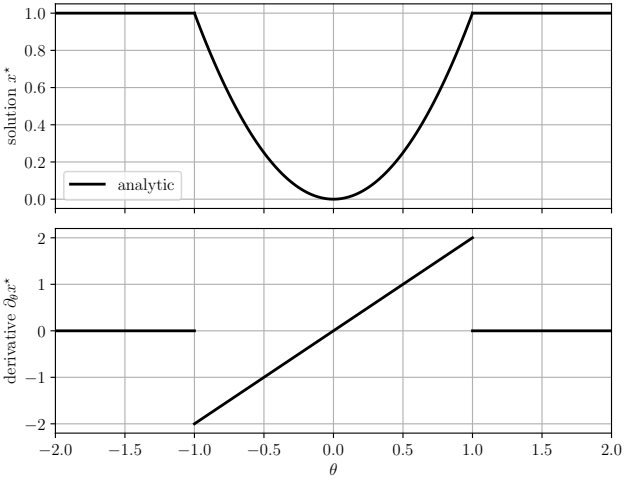
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & (x - \theta^2)^2 \\ \text{subject to} \quad & -1 \leq x \leq 1, \end{aligned}$$

Nondifferentiable solution map

$$x^\star(\theta) = \begin{cases} \theta^2, \text{ if } \theta \in [-1, 1] \\ 1, \text{ otherwise} \end{cases}$$

Derivative

$$\partial_\theta x^\star(\theta) = \begin{cases} 2 \cdot \theta, \text{ if } \theta \in (-1, 1) \\ 0, \text{ if } |\theta| > 1 \\ \text{not defined, for } \theta \in -1, 1 \end{cases}$$



Simple dense NLP example

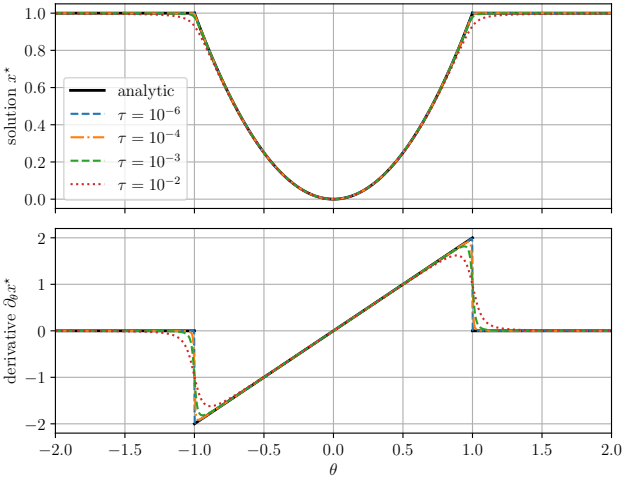
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & (x-\theta^2)^2 \\ \text{subject to} \quad & -1 \le x \le 1, \end{aligned}$$

Nondifferentiable solution map

$$x^\star(\theta) = \begin{cases} \theta^2, \text{ if } \theta \in [-1,1] \\ 1, \text{ otherwise} \end{cases}$$

Derivative

$$\partial_\theta x^\star(\theta) = \begin{cases} 2\cdot\theta, \text{ if } \theta \in (-1,1) \\ 0, \text{ if } |\theta| > 1 \\ \text{not defined, for } \theta \in -1,1 \end{cases}$$



💡 Barrier NLP solution map differentiable across active-set changes.

Barrier formulations & SBQP

Exact NLP

$$
\begin{aligned}
v^\star(\theta) := \quad & \underset{z \in \mathbb{R}^{n_z}}{\arg\min} && f(z;\theta) \\
& \text{subject to} && g(v;\theta) = 0, \\
& && h(v;\theta) \leq 0
\end{aligned}
$$

Barrier NLP

$$
\begin{aligned}
& \underset{v \in \mathbb{R}^{n_v}}{\text{minimize}} && f(v;\theta) - \tau \sum_{i=1}^{n_h} \log(-h_i(v;\tau)) \\
& \text{subject to} && g(v;\theta) = 0
\end{aligned}
$$

- Interior-point methods (IPM) solve barrier NLP for $\tau \to 0$.
- IPM with $\tau \to \tau_{\min} > 0$ solves barrier NLP.

Sequential barrier quadratic programming (SBQP)

$$
\begin{aligned}
& \underset{\Delta v}{\text{minimize}} && q_k^\top \Delta v + \frac{1}{2} \Delta v^\top Q_k \Delta v - \tau_{\min} \sum_{i=1}^{n_h} \log(s_i) \\
& \text{subject to} && g_k + G_k \Delta v = 0, \\
& && h_k + H_k \Delta v - s = 0,
\end{aligned}
$$

- Hessian approximation $Q_k \approx \nabla^2_{vv} \mathcal{L}(v_k, \lambda_k, \mu_k; \theta)$
- Update $v_{k+1} = v_k + \Delta v^k_{\text{BQP}}$
  $\lambda_{k+1} = \lambda^k_{\text{BQP}}, \mu_{k+1} = \mu^k_{\text{BQP}}$

Barrier formulations & SBQP

Exact NLP

$$
\begin{aligned}
v^\star(\theta) := \underset{z \in \mathbb{R}^{n_z}}{\arg\min} \quad & f(z;\theta) \\
\text{subject to} \quad & g(v;\theta) = 0, \\
& h(v;\theta) \leq 0
\end{aligned}
$$

Barrier NLP

$$
\begin{aligned}
\underset{v \in \mathbb{R}^{n_v}}{\text{minimize}} \quad & f(v;\theta) - \tau \sum_{i=1}^{n_h} \log(-h_i(v;\tau)) \\
\text{subject to} \quad & g(v;\theta) = 0
\end{aligned}
$$

- Interior-point methods (IPM) solve barrier NLP for $\tau \to 0$.
- IPM with $\tau \to \tau_{\min} > 0$ solves barrier NLP.

Sequential barrier quadratic programming (SBQP)

$$
\begin{aligned}
\underset{\Delta v}{\text{minimize}} \quad & q_k^\top \Delta v + \frac{1}{2} \Delta v^\top Q_k \Delta v - \tau_{\min} \sum_{i=1}^{n_h} \log(s_i) \\
\text{subject to} \quad & g_k + G_k \Delta v = 0, \\
& h_k + H_k \Delta v - s = 0,
\end{aligned}
$$

- Hessian approximation $Q_k \approx \nabla^2_{vv} \mathcal{L}(v_k, \lambda_k, \mu_k; \theta)$
- Update $v_{k+1} = v_k + \Delta v^k_{\text{BQP}}$
  $\lambda_{k+1} = \lambda^k_{\text{BQP}}, \mu_{k+1} = \mu^k_{\text{BQP}}$

In case of convergence

🎯 Barrier NLP solution

📉 Linear convergence via SCP

Solution sensitivities via SBQP

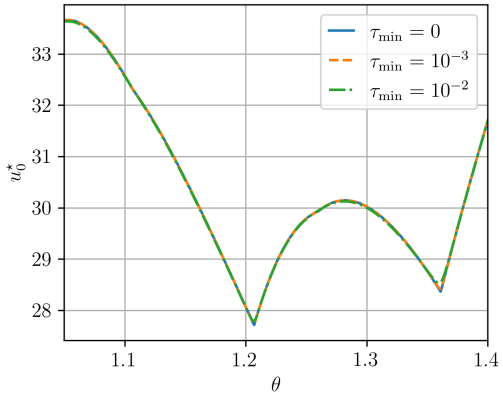


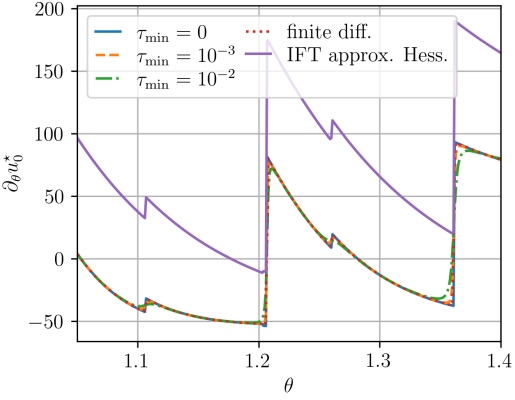
- 📜 Barrier QP sensitivities match barrier NLP ones
  - ⚠️ If exact Hessian of Lagrangian $Q_k = \nabla^2_{vv}\mathcal{L}(v_k, \lambda_k, \mu_k; \theta)$ is used.

Solution sensitivities via SBQP

- 📜 Barrier QP sensitivities match barrier NLP ones
  - ⚠️ If exact Hessian of Lagrangian $Q_k = \nabla^2_{vv}\mathcal{L}(v_k, \lambda_k, \mu_k; \theta)$ is used.
- 🤝 Two-solver approach
  - ➡️ Benefits of Hessian approximations in solve
  - ⬅️ Exact Hessian for sensitivity computation

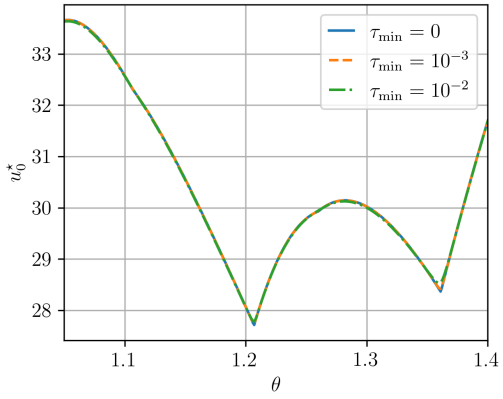
Highly-parametric optimal control example

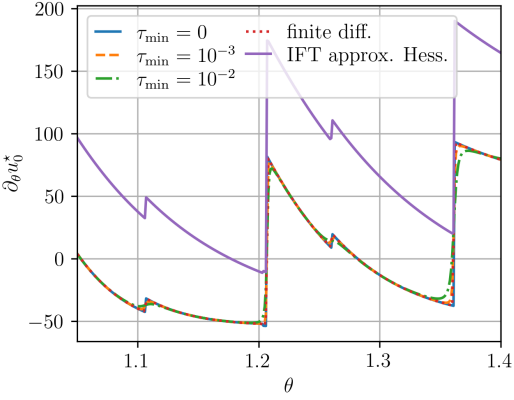




- Pendulum on cart inspired
- θ in cost, dynamics, constraints
- θ mass of cart

Highly-parametric optimal control example





- Pendulum on cart inspired
- θ in cost, dynamics, constraints
- θ mass of cart

Wrong results!

⚠ Gauss-Newton Hessian approx. in IFT

Solution sensitivities via SBQP

- 📜 Barrier QP sensitivities match barrier NLP ones
  - ⚠️ If exact Hessian of Lagrangian $Q_k = \nabla^2_{vv}\mathcal{L}(v_k, \lambda_k, \mu_k; \theta)$ is used.
- 🤝 Two-solver approach
  - ➡️ Benefits of Hessian approximations in solve
  - ⬅️ Exact Hessian for sensitivity computation

- 📚 Barrier QP sensitivities via implicit function theorem

$$\frac{\partial w^{\text{sol}}}{\partial \theta} = -\mathcal{M}_\star^{-1} \frac{\partial r_{\text{IPM},\star}}{\partial \theta},$$

- ⏩ Same matrix inverse as in IPM: $\mathcal{M}^{-1}$
  - ➡️ OCP-structure exploitation via Riccati

Solution sensitivities via SBQP

- 📜 Barrier QP sensitivities match barrier NLP ones
  - ⚠️ If exact Hessian of Lagrangian $Q_k = \nabla^2_{vv}\mathcal{L}(v_k, \lambda_k, \mu_k; \theta)$ is used.
- 🤝 Two-solver approach
  - ➡️ Benefits of Hessian approximations in solve
  - ⬅️ Exact Hessian for sensitivity computation

- 📚 Barrier QP sensitivities via implicit function theorem

$$\frac{\partial w^{\text{sol}}}{\partial \theta} = -\mathcal{M}_\star^{-1} \frac{\partial r_{\text{IPM},\star}}{\partial \theta},$$

- Same matrix inverse as in IPM: $\mathcal{M}^{-1}$
  - ➡️ OCP-structure exploitation via Riccati
- Efficient adjoint sensitivity computation

$$\underbrace{-\frac{\partial r_{\text{IPM},\star}}{\partial \theta}^{\top} \underbrace{(\mathcal{M}_\star^{-\top} \frac{\partial \mathcal{J}_{\text{task}}}{\partial v^\star})}_{\texttt{HPIPM Riccati}}}_{\text{adjoint AD via } \texttt{CasADi}}$$

Benchmark: bounded LQR problems

Timings in [ms]; Solving 128 instances with $N=20,\ n_x=8,\ n_u=4,\ n_\theta=248$.
In parentheses are multiples of the `acados` runtime.

	Nominal solution			Solution + adjoint sens.		
$u_{\max}$	`acados`	`mpc.pytorch`	`cvxpygen`	`acados`	`mpc.pytorch`	`cvxpygen`
10^4	**8.5**	78 ($\times 9.2$)	262 ($\times 31$)	**34.5**	125 ($\times 3.6$)	658 ($\times 19$)
1.0	**17.6**	21024 ($\times 1200$)	6402 ($\times 360$)	**42.0**	21899 ($\times 520$)	6845 ($\times 160$)

🚀 proposed implementation ≥ 3 times faster over state-of-the-art implementations on CPU.

Conclusion



Explored potential of advanced OCP formulation: GNRK and MOCP

Conclusion


Explored potential of advanced OCP formulation: GNRK and MOCP


Combination with state-of-the-art real-time algorithms (AS-RTI)

Conclusion



- 📈 Explored potential of advanced OCP formulation: GNRK and MOCP
- 🤝 Combination with state-of-the-art real-time algorithms (AS-RTI)
- ⬅️ SBQP enables differentiable constrained NMPC

Conclusion

- 📈 Explored potential of advanced OCP formulation: GNRK and MOCP
- 🤝 Combination with state-of-the-art real-time algorithms (AS-RTI)
- ⬅️ SBQP enables differentiable constrained NMPC
- 🔀 Synergies across contributions
- ⏩ All advances in acados, available for real-world application

Conclusion

- 📈 Explored potential of advanced OCP formulation: GNRK and MOCP
- 🤝 Combination with state-of-the-art real-time algorithms (AS-RTI)
- ⬅️ SBQP enables differentiable constrained NMPC
- 🔀 Synergies across contributions
- ⏩ All advances in acados, available for real-world application

- 🙏 Thanks to all collaborators and supporters
- 🙏 Thanks for your attention!

References I

Adabag, E., Greiff, M., Subosits, J., & Lew, T. (2025). *Differentiable model predictive control on the GPU*. Retrieved from `https://arxiv.org/abs/2510.06179`

Amos, B., Jimenez, I., Sacks, J., Boots, B., & Kolter, J. Z. (2018). Differentiable MPC for end-to-end planning and control. *Advances in neural information processing systems*, *31*.

Andersson, J. A. E., & Rawlings, J. B. (2018). Sensitivity analysis for nonlinear programming in CasADi. In *Proceedings of the ifac conference on nonlinear model predictive control (nmpc)*.

Arnström, D., Bemporad, A., & Axehill, D. (2022). A dual active-set solver for embedded quadratic programming using recursive LDL^T updates. *IEEE Transactions on Automatic Control*. doi: 10.1109/TAC.2022.3176430

Baumgärtner, K., Wang, Y., Zanelli, A., & Diehl, M. (2022). Fast nonlinear model predictive control using barrier formulations and squashing with a generalized Gauss-Newton Hessian. In *Proceedings of the ieee conference on decision and control (cdc)* (pp. 558–563).

Baumgärtner, K., Zanelli, A., & Diehl, M. (2023). Stability analysis of nonlinear model predictive control with progressive tightening of stage costs and constraints. *IEEE Control Systems Letters*.

References II

Bock, H. G., Diehl, M., Kostina, E. A., & Schlöder, J. P. (2007). Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-time and online pde-constrained optimization* (pp. 3–22). SIAM. doi: 10.1137/1.9780898718935.ch1

Carlos, B. B., Sartor, T., Zanelli, A., Diehl, M., & Oriolo, G. (2020). Least Conservative Linearized Constraint Formulation for Real-Rime Motion Generation. In *Proc. 21st ifac world congress* (pp. 9519–9525).

Carlos, B. B., Sartor, T., Zanelli, A., Frison, G., Burgard, W., Diehl, M., & Oriolo, G. (2020). An efficient real-time nmpc for quadrotor position control under communication time-delay. In *2020 16th international conference on control, automation, robotics and vision (icarcv)* (p. 982-989). doi: 10.1109/ICARCV50220.2020.9305513

Cecchin, L., Frey, J., Gering, S., Manderla, M., Trachte, A., & Diehl, M. (2023, October). Nonlinear Model Predictive Control for Efficient Control of Variable Speed Variable Displacement Pumps. In *Ifac-papersonline* (Vol. 56, pp. 331–336). South Lake Tahoe, US: IFAC. Retrieved 2024-05-23, from `https://www.sciencedirect.com/science/article/pii/S2405896323023790` (Number: 3) doi: 10.1016/j.ifacol.2023.12.046

References III

Dong, S., Ghezzi, A., Harzer, J., Frey, J., Gao, B., Chen, H., & Diehl, M. (2024). Real-time nmpc with convex–concave constraints and application to eco-driving. *IEEE Transactions on Control Systems Technology*, *33*(2), 807–814. doi: 10.1109/TCST.2024.3494993

Frey, J., Baumgärtner, K., & Diehl, M. (2024). Gauss-Newton Runge-Kutta integration for efficient discretization of optimal control problems with long horizons and least-squares costs. *European Journal of Control*, 101038.

Frey, J., Baumgärtner, K., Frison, G., & Diehl, M. (2025). Multi-phase optimal control problems for efficient nonlinear model predictive control with acados. *Optimal Control Applications and Methods*, *46*(2), 827-845. doi: https://doi.org/10.1002/oca.3234

Frey, J., Baumgärtner, K., Frison, G., Reinhardt, D., Hoffmann, J., Fichtner, L., ... Diehl, M. (2026). Differentiable nonlinear model predictive control. *Advances in neural information processing systems (accepted for publication)*.

Frey, J., Cairano, S. D., & Quirynen, R. (2020). Active-Set based Inexact Interior Point QP Solver for Model Predictive Control. In *Proceedings of the ifac world congress.*

Frey, J., De Schutter, J., & Diehl, M. (2023). Fast integrators with sensitivity propagation for use in CasADi. In *Proceedings of the european control conference (ecc).*

References IV

Frey, J., Gao, Y., Messerer, F., Lahr, A., Zeilinger, M. N., & Diehl, M. (2024). Efficient zero-order robust optimization for real-time model predictive control with acados. In *Proceedings of the european control conference (ecc)*.

Frey, J., Kiessling, D., Baumgärtner, K., & Diehl, M. (2026). Anderson acceleration for linearly converging SQP-type methods. In *Proceedings of the ifac world congress (accepted)*. Retrieved from `https://arxiv.org/abs/2604.14803`

Frey, J., Nurkanović, A., & Diehl, M. (2024). Advanced-step real-time iterations with four levels – new error bounds and fast implementation in acados. *IEEE Control Systems Letters*. doi: 10.1109/LCSYS.2024.3412007

Gao, Y., Messerer, F., van Duijkeren, N., Houska, B., & Diehl, M. (2024). Real-time-feasible collision-free motion planning for ellipsoidal objects. *Proceedings of the IEEE Conference on Decision and Control (CDC)*.

Ge, J., Miao, X., su, J., Li, S., & Chen, W.-H. (2026, 10). Op-follower: Onboard vision-based obstacle-aware predictive person following for quadruped robots via ibvs-nmpc control.

Gordon, D. C., Norouzi, A., Winkler, A., McNally, J., Nuss, E., Abel, D., ... Koch, C. R. (2022). End-to-end deep neural network based nonlinear model predictive control: experimental implementation on diesel engine emission control. *Energies*, *15*(24), 9335.

References V

Homburger, H. (2026). *Optimal control for efficient vessel operation: From theory to real-world applications* (Doctoral dissertation). doi: 10.6094/UNIFR/284230

Homburger, H., Frey, J., Wirtensohn, S., Diehl, M., & Reuter, J. (2025). Millisecond NMPC for swing-up and stabilization of the Furuta pendulum in real world. *IEEE Transactions on Control Systems Technology*, 1-7. doi: 10.1109/TCST.2025.3616367

Köhler, J., Kötting, P., Soloperto, R., Allgöwer, F., & Müller, M. A. (2021). A robust adaptive model predictive control framework for nonlinear uncertain systems. *International Journal of Robust and Nonlinear Control*, *31*(18), 8725–8749.

Köhler, J., Zhang, D., Soloperto, R., Carron, A., & Zeilinger, M. (2025). *An MPC framework for efficient navigation of mobile robots in cluttered environments*. Retrieved from `https://arxiv.org/abs/2509.15917`

Löw, S., Obradovic, D., & Bottasso, C. (2020, 07). Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction. doi: 10.13140/RG.2.2.29851.28966

Nurkanović, A., Zanelli, A., Albrecht, S., & Diehl, M. (2019). The Advanced Step Real Time Iteration for NMPC. In *Proceedings of the ieee conference on decision and control (cdc).*

References VI

Nurkanović, A., Zanelli, A., Frison, G., Albrecht, S., & Diehl, M. (2020). Contraction properties of the advanced step real-time iteration. In *Proceedings of the ifac world congress* (Vol. 51).

Pineda, L., Fan, T., Monge, M., Venkataraman, S., Sodhi, P., Chen, R. T., ... others (2022). Theseus: A library for differentiable nonlinear optimization. *Advances in Neural Information Processing Systems*, *35*, 3801–3818.

Pirnay, H., López-Negrete, R., & Biegler, L. T. (2012, 01). Optimal sensitivity based on IPOPT. *Mathematical Programming Computation*, *4*(4), 307–331. Retrieved from `https://doi.org/10.1007/s12532-012-0043-2` doi: 10.1007/s12532-012-0043-2

Procházka, O., Pauko, J., Pěnička, R., & Saska, M. (2026). *Nmpp: Nonlinear model predictive planning for agile uav flight in cluttered environments.* Retrieved from `https://arxiv.org/abs/2610.08695`

Rathod, N., Bratta, A., Focchi, M., Zanon, M., Villarreal, O., Semini, C., & Bemporad, A. (2021). Model predictive control with environment adaptation for legged locomotion. *IEEE Access*, *9*, 145710–145727.

Rawlings, J. B., Mayne, D. Q., & Diehl, M. M. (2017). *Model predictive control: Theory, computation, and design* (2nd ed.). Nob Hill.

References VII

Reiter, R., Baumgärtner, K., Quirynen, R., & Diehl, M. (2024). Progressive smoothing for motion planning in real-time NMPC. *Proceedings of the European Control Conference (ECC)*.

Reiter, R., Nurkanović, A., Frey, J., & Diehl, M. (2023). Frenet-cartesian model representations for automotive obstacle avoidance within nonlinear mpc. *European Journal of Control*, *74*, 100847. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0947358023000766` (2023 European Control Conference Special Issue) doi: https://doi.org/10.1016/j.ejcon.2023.100847

Reiter, R., Qin, C., Bauersfeld, L., & Scaramuzza, D. (2025). *Unifying quadrotor motion planning and control by chaining different fidelity models.* Retrieved from `https://arxiv.org/abs/2512.12427`

Romero, A., Penicka, R., & Scaramuzza, D. (2022). Time-optimal online replanning for agile quadrotor flight. *IEEE Robotics and Automation Letters*, *7*(3), 7730–7737.

Salzmann, T., Kaufmann, E., Arrizabalaga, J., Pavone, M., Scaramuzza, D., & Ryll, M. (2023). Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms. *IEEE Robotics and Automation Letters*, *8*(4), 2397–2404.

References VIII

Saviolo, A., Frey, J., Rathod, A., Diehl, M., & Loianno, G. (2024). Active learning of discrete-time dynamics for uncertainty-aware model predictive control. *IEEE Transactions on Robotics*, *40*, 1273-1291. doi: 10.1109/TRO.2023.3339543

Schaller, M., Banjac, G., Diamond, S., Agrawal, A., Stellato, B., & Boyd, S. (2022). Embedded code generation with cvxpy. *IEEE Control Systems Letters*, *6*, 2653–2658.

Schaller, M., & Boyd, S. (2025). Code generation for solving and differentiating through convex optimization problems. *arXiv preprint arXiv:2504.14099*. Retrieved from `https://arxiv.org/abs/2504.14099`

Schimpe, A., Majstorovic, D., & Diermeyer, F. (2022). Steering action-aware adaptive cruise control for teleoperated driving. In *2022 ieee international conference on systems, man, and cybernetics (smc)* (pp. 988–993).

Subash, A., Kloeser, D., Frey, J., Reiter, R., Diehl, M., & Bohlmann, K. (2024). Model predictive control for frenet-cartesian trajectory tracking of a tricycle kinematic automated guided vehicle. In *2024 ieee/rsj international conference on intelligent robots and systems (iros)* (p. 11948-11953). doi: 10.1109/IROS58592.2024.10802822

References IX

Taouil, I., Ciebelski, M., Omar, S., Zhao, H., Dai, A., Johnson, A. M., & Khadiv, M. (2026). *Motiondisco: Motion discovery for extreme humanoid loco-manipulation*. Retrieved from `https://arxiv.org/abs/2606.06139`

Tran-Dinh, Q., Savorgnan, C., & Diehl, M. (2012). Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization. *SIAM J. Optimization*, *22*(4), 1258–1284.

Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., van Duijkeren, N., Zanelli, A., ... Diehl, M. (2021, 09). acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 147–183. doi: 10.1007/s12532-021-00208-8

Weiss, R., Frey, J., Diehl, M., & Reuter, J. (2025). Efficient QP solution in nonlinear MPC for an industrial heating process described by a 2-D partial differential equation. In *Proceedings of the european control conference (ecc)*. doi: 10.23919/ECC65951.2025.11186880

Winkler, A., Frey, J., Fahrbach, T., Frison, G., Scheer, R., Diehl, M., & Andert, J. (2021). Embedded real-time nonlinear model predictive control for the thermal torque derating of an electric vehicle. In *Proceedings of the ifac world congress* (Vol. 54, pp. 359–364).