Efficient Zero-Order Robust Optimization for Real-Time Model Predictive Control with acados

Jonathan Frey, Yunfan Gao, Florian Messerer, Amon Lahr, Melanie Zeilinger, Moritz Diehl

Systems Control and Optimization Laboratory (syscop)

ECC 2024
June 28, 2024

universität freiburg

Overview

- Optimal Control Problem (OCP) formulations with uncertainties
  - 🌫️ systematic treatment of uncertainty
  - 🦺 Robust formulations
  - 🎲 Stochastic formulations

Overview

- Optimal Control Problem (OCP) formulations with uncertainties
  - 🌫️ systematic treatment of uncertainty
  - 🦺 Robust formulations
  - 🎲 Stochastic formulations
  - 😵‍💫 computationally expensive
- Zero Order Robust Optimization (zoRO)
  - separates propagation of uncertainties from nominal dynamics
  - + computationally cheaper
  - + converges to suboptimal but feasible point

Overview

- Optimal Control Problem (OCP) formulations with uncertainties
  - systematic treatment of uncertainty
  - Robust formulations
  - Stochastic formulations
  - computationally expensive
- Zero Order Robust Optimization (zoRO)
  - separates propagation of uncertainties from nominal dynamics
  - \+ computationally cheaper
  - \+ converges to suboptimal but feasible point
- This paper
  - Combine Real-Time Iteration (RTI) & zoRO
  - New implementation acados
    - efficient
    - user-friendly
    - `Python` interface

Robust & stochastic optimal control

$$\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}\\ P_0,\ldots,P_N}} \quad \sum_{k=0}^{N-1} l(u_k, x_k) + M(x_N) \tag{1a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \tag{1b}$$

$$P_0 = \bar{P}_0, \tag{1c}$$

$$x_{k+1} = \psi_k(x_k, u_k, 0), \qquad k = 0, \ldots, N-1, \tag{1d}$$

$$P_{k+1} = \Phi_k(x_k, u_k, P_k), \qquad k = 0, \ldots, N-1, \tag{1e}$$

$$0 \geq h_k(x_k, u_k) + \beta_k(x_k, u_k, P_k), \; k = 0, \ldots, N-1, \tag{1f}$$

$$0 \geq h_N(x_N) + \beta_N(x_N, P_N) \tag{1g}$$

- Nominal trajectory $(x_0, u_0, \ldots, u_{N-1}, x_N)$
- Discrete dynamics $\psi_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$
  - depend on uncertain variable $w_k \in \mathbb{R}^{n_w}$
- Uncertainty dynamics Φ_k
- Matrices $P_0, \ldots, P_N \in \mathbb{R}^{n_x \times n_x}$

Robust & stochastic optimal control

$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}\\ P_0,\dots,P_N}} \quad & \sum_{k=0}^{N-1} l(u_k, x_k) + M(x_N) && \text{(1a)}\\
\text{s.t.} \quad & x_0 = \bar{x}_0, && \text{(1b)}\\
& P_0 = \bar{P}_0, && \text{(1c)}\\
& x_{k+1} = \psi_k(x_k, u_k, 0), \quad k = 0, \dots, N-1, && \text{(1d)}\\
& P_{k+1} = \Phi_k(x_k, u_k, P_k), \quad k = 0, \dots, N-1, && \text{(1e)}\\
& 0 \geq h_k(x_k, u_k) + \beta_k(x_k, u_k, P_k), \quad k = 0, \dots, N-1, && \text{(1f)}\\
& 0 \geq h_N(x_N) + \beta_N(x_N, P_N) && \text{(1g)}
\end{aligned}
$$

- Nominal trajectory $(x_0, u_0, \dots, u_{N-1}, x_N)$
- Discrete dynamics $\psi_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$
  - depend on uncertain variable $w_k \in \mathbb{R}^{n_w}$
- Uncertainty dynamics Φ_k
- Matrices $P_0, \dots, P_N \in \mathbb{R}^{n_x \times n_x}$

 w_k independently normal distributed with zero mean and covariance W_k, e.g. $w_k \sim \mathcal{N}(0, W_k)$.

 x_k, P_k: mean, variance of a stochastic state χ_k, usually approximated as $\chi_k \sim \mathcal{N}(x_k, P_k)$,

Robust & stochastic optimal control

$$\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}\\ P_0,\dots,P_N}} \quad \sum_{k=0}^{N-1} l(u_k, x_k) + M(x_N) \tag{1a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \tag{1b}$$

$$P_0 = \bar{P}_0, \tag{1c}$$

$$x_{k+1} = \psi_k(x_k, u_k, 0), \qquad k = 0, \dots, N-1, \tag{1d}$$

$$P_{k+1} = \Phi_k(x_k, u_k, P_k), \qquad k = 0, \dots, N-1, \tag{1e}$$

$$0 \geq h_k(x_k, u_k) + \beta_k(x_k, u_k, P_k), \; k = 0, \dots, N-1, \tag{1f}$$

$$0 \geq h_N(x_N) + \beta_N(x_N, P_N) \tag{1g}$$

- Robust setting: $(w_0, \dots, w_{N-1}) \in \mathcal{E}\left(0, \operatorname{blkdiag}(W_0, \dots, W_{N-1})\right)$
- Ellipsoid: $\mathcal{E}(q, Q) :=$
  $\{\, x \in \mathbb{R}^{n_q} \mid (x-q)^\top Q^{-1}(x-q) \leq 1 \,\}$
  with center q, positive definite $Q \in \mathbb{R}^{n_q \times n_q}$

- w_k independently normal distributed with zero mean and covariance W_k, e.g. $w_k \sim \mathcal{N}(0, W_k)$.
- x_k, P_k: mean, variance of a stochastic state χ_k, usually approximated as $\chi_k \sim \mathcal{N}(x_k, P_k)$,

Uncertainty propagation & backoff

- The uncertainty propagation

$$\Phi_k(P_k, x_k, u_k) = (A_k + B_k K_k) P_k (A_k + B_k K_k)^\top + G_k W_k G_k^\top \tag{2}$$

  with $A_k := \frac{\partial \psi_k}{\partial x}(x_k, u_k, 0)$, and $B_k := \frac{\partial \psi_k}{\partial u}(x_k, u_k, 0)$
  $G_k = \frac{\partial \psi_k}{\partial w}(x_k, u_k, 0)$ constant for additive noise
- Precomputed linear feedback gains $K_k \in \mathbb{R}^{n_u \times n_x}$ can reduce conservatism (Mayne et al., 2011)

Uncertainty propagation & backoff

- The uncertainty propagation

$$\Phi_k(P_k, x_k, u_k) = (A_k + B_k K_k) P_k (A_k + B_k K_k)^\top + G_k W_k G_k^\top \tag{2}$$

with $A_k := \frac{\partial \psi_k}{\partial x}(x_k, u_k, 0)$, and $B_k := \frac{\partial \psi_k}{\partial u}(x_k, u_k, 0)$
$G_k = \frac{\partial \psi_k}{\partial w}(x_k, u_k, 0)$ constant for additive noise

- Precomputed linear feedback gains $K_k \in \mathbb{R}^{n_u \times n_x}$ can reduce conservatism (Mayne et al., 2011)
- Backoff term for a constraint component $h_{k,i}(\cdot)$

$$\beta_{k,i}(x_k, u_k, P_k) := \gamma \sqrt{\nabla h_{k,i}(x_k, u_k)^\top \begin{bmatrix} \mathbb{1}_{n_x} \\ K \end{bmatrix} P_k \begin{bmatrix} \mathbb{1}_{n_x} \\ K \end{bmatrix}^\top \nabla h_{k,i}(x_k, u_k)} \tag{3}$$

Uncertainty propagation & backoff

- The uncertainty propagation

$$\Phi_k(P_k, x_k, u_k) = (A_k + B_k K_k) P_k (A_k + B_k K_k)^\top + G_k W_k G_k^\top \tag{2}$$

  with $A_k := \frac{\partial \psi_k}{\partial x}(x_k, u_k, 0)$, and $B_k := \frac{\partial \psi_k}{\partial u}(x_k, u_k, 0)$
  $G_k = \frac{\partial \psi_k}{\partial w}(x_k, u_k, 0)$ constant for additive noise
- Precomputed linear feedback gains $K_k \in \mathbb{R}^{n_u \times n_x}$ can reduce conservatism (Mayne et al., 2011)
- Backoff term for a constraint component $h_{k,i}(\cdot)$

$$\beta_{k,i}(x_k, u_k, P_k) := \gamma \sqrt{\nabla h_{k,i}(x_k, u_k)^\top \begin{bmatrix} \mathbb{1}_{n_x} \\ K \end{bmatrix} P_k \begin{bmatrix} \mathbb{1}_{n_x} \\ K \end{bmatrix}^\top \nabla h_{k,i}(x_k, u_k)} \tag{3}$$

🦺 γ equals 1

🎲 $\gamma \sim$ probability of constraint satisfaction

- x_k, P_k normal distribution: inverse normal cumulative density function
- x_k, P_k first moments of general distribution: Chebyshev's inequality

Zero Order Robust Optimization (zoRO) algorithm

- (Feng et al., 2020), (Zanelli et al., 2021)
- Eliminate uncertainty matrices from the OCP (1)
- Solve reduced subproblems in nominal variables

$$\min_{\substack{x_0,\ldots,x_N,\\u_0,\ldots,u_{N-1}}} \quad \sum_{k=0}^{N-1} l(u_k, x_k) + M(x_N) \tag{4a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \tag{4b}$$

$$x_{k+1} = \psi_k(x_k, u_k, 0), \qquad k = 0, \ldots, N-1, \tag{4c}$$

$$0 \geq h_k(x_k, u_k) + \hat{\beta}_k, k = 0, \ldots, N-1, \tag{4d}$$

$$0 \geq h_N(x_N) + \hat{\beta}_N \tag{4e}$$

- values $\hat{\beta}_k$ approximate the backoff terms $\beta_k(x_k, u_k, P_k)$.

Zero Order Robust Optimization (zoRO) algorithm

- (Feng et al., 2020), (Zanelli et al., 2021)
- Eliminate uncertainty matrices from the OCP (1)
- Solve reduced subproblems in nominal variables

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} l(u_k, x_k) + M(x_N) && \text{(4a)}\\
\text{s.t.} \quad & x_0 = \bar{x}_0, && \text{(4b)}\\
& x_{k+1} = \psi_k(x_k, u_k, 0), \quad k = 0, \ldots, N-1, && \text{(4c)}\\
& 0 \geq h_k(x_k, u_k) + \hat{\beta}_k, k = 0, \ldots, N-1, && \text{(4d)}\\
& 0 \geq h_N(x_N) + \hat{\beta}_N && \text{(4e)}
\end{aligned}
$$

- values $\hat{\beta}_k$ approximate the backoff terms $\beta_k(x_k, u_k, P_k)$.
- zoRO alternates two steps:
  1. approximate backoff terms using current guess trajectory, i.e. uncertainty propagation (2), backoff computation (3)
  2. solve subproblem (4) approximately

zoRO: Convergence properties

- Interpret the zoRO algorithm as inexact Newton-type method
- Tailored Jacobian approximation: neglecting sensitivities of uncertainty matrices wrt. nominal trajectory
- 🌟 zoRO returns a suboptimal yet feasible solution at convergence (Zanelli et al., 2021)

zoRO: Convergence properties

- Interpret the zoRO algorithm as inexact Newton-type method
- Tailored Jacobian approximation: neglecting sensitivities of uncertainty matrices wrt. nominal trajectory
- 🌟 zoRO returns a suboptimal yet feasible solution at convergence (Zanelli et al., 2021)
- Assumption of strong regularity (Zanelli et al., 2021)
  - 🌟 Approximation error scales with uncertainty magnitude
  - 🌟 zoRO recovers optimal solution in the limit for vanishing uncertainty
  - 🌟 For sufficiently small uncertainties & if the unmodified SQP algorithm converges, zoRO converges linearly.
    - extension to state- and input-dependent uncertainties $W(y)$ in (Lahr et al., 2023)

zoRO: Convergence properties

- Interpret the zoRO algorithm as inexact Newton-type method
- Tailored Jacobian approximation: neglecting sensitivities of uncertainty matrices wrt. nominal trajectory
- 🌟 zoRO returns a suboptimal yet feasible solution at convergence (Zanelli et al., 2021)
- Assumption of strong regularity (Zanelli et al., 2021)
  - 🌟 Approximation error scales with uncertainty magnitude
  - 🌟 zoRO recovers optimal solution in the limit for vanishing uncertainty
  - 🌟 For sufficiently small uncertainties & if the unmodified SQP algorithm converges, zoRO converges linearly.
    - extension to state- and input-dependent uncertainties $W(y)$ in (Lahr et al., 2023)
- Adjoint correction (Feng et al., 2020; Bock et al., 2007; Wirsching et al., 2006)
  - Compensate for the suboptimality
  - Additional computational cost: compute neglected sensitivities in the direction of Lagrange multipliers with the backwards-mode AD
  - not implemented in acados

acados zoRO SQP implementation

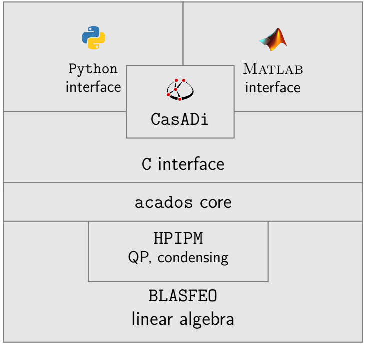
acados – very briefly

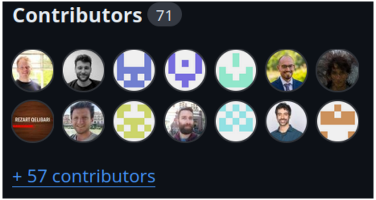
- Fast embedded solvers for
  - Initial value problems for ODEs and DAEs
  - OCP structured NLP
- Exploit OCP specific block-sparse structure
- Principles
  - efficiency
  - flexibility
  - modularity
  - portability
- Modeling nonlinearities with `CasADi`
- High-level interface: `Python`, MATLAB, `Octave`
- Generate problem specific `C` solver from templates

⭐ `https://github.com/acados/acados` ⭐

🤝 `https://discourse.acados.org/`

📚 `https://docs.acados.org/`





zoRO with Sequential Quadratic Programming (SQP) & RTI

- Solve nominal problem (4) with Sequential Quadratic Programming (SQP)
- Alternate SQP iteration (step 2) and backoff-update (step 1).
- Take linearizations A_k, B_k, $\nabla h_{k,i}(x_k, u_k)$ from solver memory

zoRO with Sequential Quadratic Programming (SQP) & RTI

- Solve nominal problem (4) with Sequential Quadratic Programming (SQP)
- Alternate SQP iteration (step 2) and backoff-update (step 1).
- Take linearizations A_k, B_k, $\nabla h_{k,i}(x_k, u_k)$ from solver memory

⏩ Real-time iteration (RTI) (Diehl et al., 2005)

- performs 1 SQP iteration
- preparation phase: linearizations
- feedback phase:
  - perform computations that need $\bar{x}_0$
  - genalize: computations that need bound values

zoRO with Sequential Quadratic Programming (SQP) & RTI

- Solve nominal problem (4) with Sequential Quadratic Programming (SQP)
- Alternate SQP iteration (step 2) and backoff-update (step 1).
- Take linearizations A_k, B_k, $\nabla h_{k,i}(x_k, u_k)$ from solver memory
- ⏩ Real-time iteration (RTI) (Diehl et al., 2005)
  - performs 1 SQP iteration
  - preparation phase: linearizations
  - feedback phase:
    - perform computations that need $\bar{x}_0$
    - genalize: computations that need bound values
- Insert backoffs by adapting bounds
  - ⏩ part of preparation phase, after nominal preparation

zoRO with Sequential Quadratic Programming (SQP) & RTI

- Solve nominal problem (4) with Sequential Quadratic Programming (SQP)
- Alternate SQP iteration (step 2) and backoff-update (step 1).
- Take linearizations A_k, B_k, $\nabla h_{k,i}(x_k, u_k)$ from solver memory
- ⏩ Real-time iteration (RTI) (Diehl et al., 2005)
  - performs 1 SQP iteration
  - preparation phase: linearizations
  - feedback phase:
    - perform computations that need $\bar{x}_0$
    - genalize: computations that need bound values
- Insert backoffs by adapting bounds
  - ⏩ part of preparation phase, after nominal preparation
- Related (Hewing et al., 2020)
  - Gaussian process-based MPC
  - use previous MPC solution for backoff computation

Efficient implementation with acados template interface

- Previous zoRO implementations
  - tailored to specific problems
  - 🐌 bottleneck: backoff computations in `Python`

Efficient implementation with acados template interface

- Previous zoRO implementations
  - tailored to specific problems
  - 🐌 bottleneck: backoff computations in `Python`
- `custom_update()`
  - `C` function that can be called between `solve()` calls
  - allows various alternating algorithms
- 🛋️ convenient, flexible formulation via `ZoroDescription` to specify
  - γ, K, P_0, indices of constraints to be tightened $\mathcal{I}_{k,\mathrm{tight}}$

Efficient implementation with acados template interface

- Previous zoRO implementations
  - tailored to specific problems
  - 🐌 bottleneck: backoff computations in `Python`
- `custom_update()`
  - `C` function that can be called between `solve()` calls
  - allows various alternating algorithms
- 🛋️ convenient, flexible formulation via `ZoroDescription` to specify
  - γ, K, P_0, indices of constraints to be tightened $\mathcal{I}_{k,\mathrm{tight}}$
- ⚡ generate efficient zoRO update function from template
  - solver interactions in `C`
  - backoff computations in `C` with `BLASFEO`
  - no online memory allocation
- ⚙️ Possible interactions:
  - Update P_0, W_k
  - Output P_k

Hanging chain benchmark problem [1]

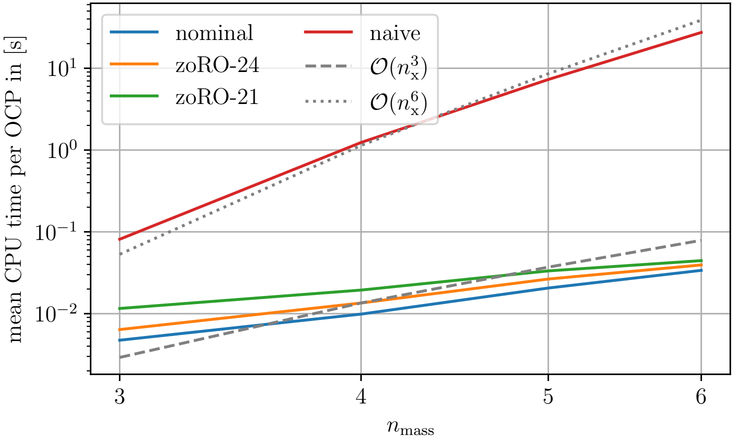


Figure: Mean computation time for solving one OCP for different number of masses n_mass.

[1] https://github.com/FreyJo/zoro-NMPC-2021

Differential drive with obstacles: OCP

- Considered in (Gao et al., 2023)
- State $x = (p_x, p_y, \theta, v, \omega)^\top$
- Control: forward acceleration a and angular acceleration α
- ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = (v\cos(\theta), v\sin(\theta), \omega, a, \alpha)^\top$
- Collision avoidance constraints $\left\|(p_x, p_y)^\top - (q_{x,i}, q_{y,i})^\top\right\|_2 \geq r + r_i^{\mathrm{obs}}$
- Index set of constraints to be tightened $\mathcal{I}_{k,\mathrm{tight}}$
  - Tighten upper bounds on v, not lower: $v \geq 0$
  - Terminal constraint: tight bounds on velocities: not tightened
- Solution with `acados`: QP solver `DAQP` (Arnstrom et al., 2022), full condensing
- Closed loop simulation:
  - additive noise after RK4 simulation
  - Process noise is sampled from multivariate normal distribution with zero mean and covariance $W = \mathrm{diag}\left(2\cdot 10^{-6}, 2\cdot 10^{-6}, 4\cdot 10^{-6}, 1.5\cdot 10^{-3}, 7\cdot 10^{-3}\right)$.
- $\bar{P}_0 = W$, $\gamma = 3.0$

Differential drive with obstacles: Closed-loop trajectory

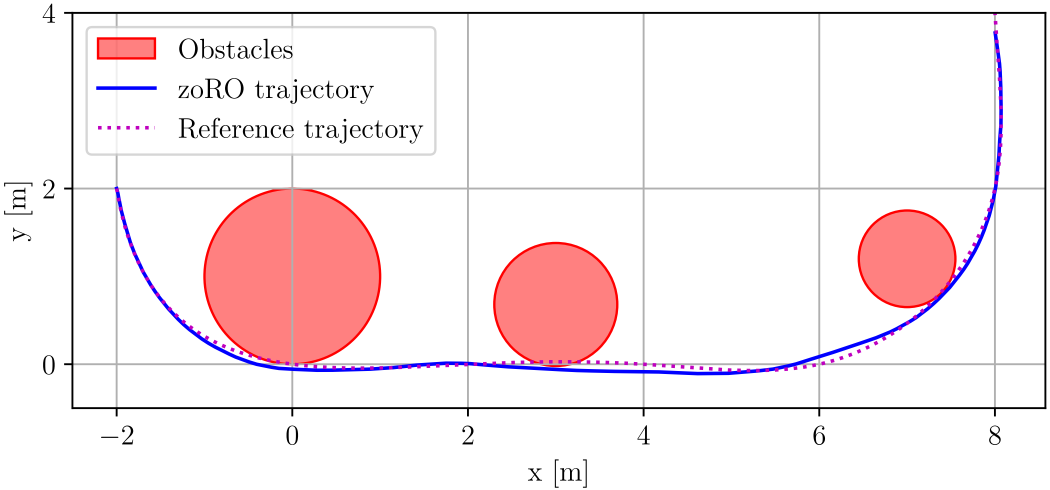


Figure: Closed-loop trajectory differential drive robot.

Differential drive with obstacles: Open-loop trajectory

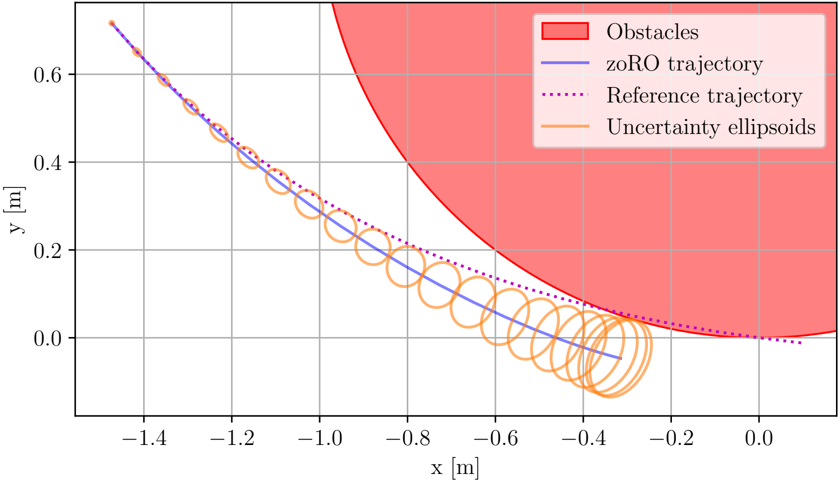


Figure: Open-loop trajectory differential drive robot.

Differential drive with obstacles: Computation times

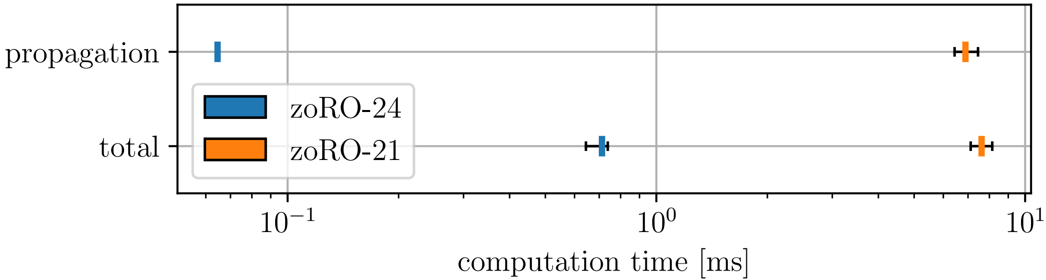


Figure: Computation times of zoRO variants. The whiskers indicate the minimum and maximum value.

Differential drive with obstacles: Computation time details

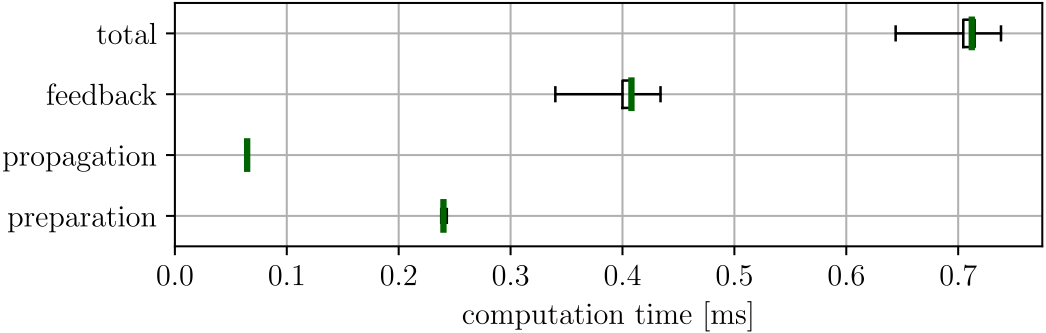


Figure: Computation times for the proposed zoRO implementation zoRO-24 on the differential drive robot. The whiskers indicate the minimum and maximum value.

Final slide

Summary

- Concise description of zoRO
  - robust 🦺, stochastic 🎲
  - combination with real-time iteration ⏩
- New acados zoRO implementation
  - ⚡ Fast
  - 🤸 Flexible
  - 🐍 Python frontend
- Two examples ⛓ 🤖

Final slide

Summary

- Concise description of zoRO
  - robust 🦺, stochastic 🎲
  - combination with real-time iteration ⏩
- New acados zoRO implementation
  - ⚡ Fast
  - 🤸 Flexible
  - 🐍 Python frontend
- Two examples ⛓️ 🤖

Future work

- Adjoint corrections (Feng et al., 2020)
- Optimize over linear feedback matrices (Messerer & Diehl, 2021)

Final slide

Summary

- Concise description of zoRO
  - robust 🦺, stochastic 🎲
  - combination with real-time iteration ⏩
- New acados zoRO implementation
  - ⚡ Fast
  - 🤸 Flexible
  - 🐍 Python frontend
- Two examples ⛓️ 🤖

Future work

- Adjoint corrections (Feng et al., 2020)
- Optimize over linear feedback matrices (Messerer & Diehl, 2021)

We hope this work enables more real-world robust MPC applications 🏗️ 🚆 ⛵

Thanks for your attention! Feedback is appreciated! 🙏

Arnstrom, 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

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

Diehl, M., Bock, H. G., & Schlöder, J. P. (2005). Real-time iterations for nonlinear optimal feedback control. In *Proceedings of the ieee conference on decision and control and european control conference (cdc-ecc)* (pp. 5871–5876). doi: 10.1109/CDC.2005.1583100

Feng, X., Cairano, S. D., & Quirynen, R. (2020). Inexact Adjoint-based SQP Algorithm for Real-Time Stochastic nonlinear MPC. In *Proceedings of the ifac world congress.*

Gao, Y., Messerer, F., Frey, J., van Duijkeren, N., & Diehl, M. (2023). Collision-free motion planning for mobile robots by zero-order robust optimization-based mpc. In *Proceedings of the european control conference (ecc).*

References II

Hewing, L., Kabzan, J., & Zeilinger, M. N. (2020). Cautious model predicitive control using gaussian process regression. *IEEE Transaction on Control Systems Technology*, *28*(6), 2736–2743.

Lahr, A., Zanelli, A., Carron, A., & Zeilinger, M. N. (2023). Zero-order optimization for Gaussian process-based model predictive control. *European Journal of Control*, 100862.

Mayne, D., Kerrigan, E., van Wyk, E. J., & Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, *21*, 1341-1353.

Messerer, F., & Diehl, M. (2021). An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback. In *Proceedings of the ieee conference on decision and control (cdc).*

Wirsching, L., Bock, H. G., & Diehl, M. (2006). Fast NMPC of a chain of masses connected by springs. In *Proceedings of the ieee international conference on control applications, munich* (pp. 591–596). doi: 10.1109/CACSD-CCA-ISIC.2006.4776712

References III

Zanelli, A., Frey, J., Messerer, F., & Diehl, M. (2021). Zero-order robust nonlinear model predictive control with ellipsoidal uncertainty sets. *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*. doi: https://doi.org/10.1016/j.ifacol.2021.08.523