This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3412007

# Advanced-Step Real-Time Iterations with Four Levels – New Error Bounds and Fast Implementation in acados

Jonathan Frey[1,2], Armin Nurkanović[1], Moritz Diehl[1,2]

***Abstract*— The Real-Time Iteration (RTI) is an online nonlinear model predictive control algorithm that performs a single Sequential Quadratic Programming (SQP) per sampling time. The algorithm is split into a preparation and a feedback phase, where the latter one performs as little computations as possible solving a single prepared quadratic program. To further improve the accuracy of this method, the Advanced-Step RTI (AS-RTI) performs additional Multi-Level Iterations (MLI) in the preparation phase, such as inexact or zero-order SQP iterations on a problem with a predicted state estimate. This paper extends and streamlines the existing local convergence analysis of AS-RTI, such as analyzing MLI of level A and B for the first time, and significantly simplifying the proofs for levels C and D. Moreover, this paper provides an efficient open-source implementation in acados, making it widely accessible to practitioners.**

***Index Terms*— Model predictive control, numerical algorithms, software**

## I. Introduction

NONLINEAR Model Predictive Control (NMPC) requires at every sampling instant an approximate online solution of a discrete-time optimal control problem (OCP) of the form

$$\min_{\substack{s_0,\dots,s_N,\\u_0,\dots,u_{N-1}}} \quad \sum_{i=0}^{N-1} L_i(s_i,u_i) + E(s_N) \tag{1a}$$

$$\text{s.t.} \quad s_0 = x, \tag{1b}$$

$$s_{i+1} = \phi_i(s_i,u_i),\ k=0,\dots,N-1, \tag{1c}$$

$$0 \leq h_i(s_i,u_i),\ k=0,\dots,N-1, \tag{1d}$$

$$0 \leq h_N(s_N). \tag{1e}$$

Its optimization variables are the states $s_i \in \mathbb{R}^{n_x}$ at $\tau^i$, $k = 0,\dots,N$ and the control inputs $u_i \in \mathbb{R}^{n_u}$ acting on shooting intervals $[\tau^i,\tau^{i+1}]$, $k = 0,\dots,N-1$. The values $s_i$ and $s_{i+1}$ are coupled by the discrete-time dynamics $\phi_i$, which represent the evolution of the real system over a shooting interval. The cost is given by the path cost terms $L_i$ and the terminal cost term $E$. The path constraints are given by $h_i$ and the terminal constraint by $h_N$. In NMPC, once the initial state $x \in \mathbb{R}^{n_x}$ is known, the parametric nonlinear program (NLP) in (1) is solved, and the first control input $u_0$ is fed back to the plant.

[1]Department of Microsystems Engineering (IMTEK), University Freiburg, Germany, jonathan.frey@imtek.uni-freiburg.de

[2]Department of Mathematics, University Freiburg, Germany

This research was supported by DFG via Research Unit FOR 2401, project 424107692 and 525018088, by BMWK via 03EI4057A and 03EN3054B, and by the EU via ELO-X 953348.

Due to the availability of real-time algorithms and efficient open-source software implementations, NMPC is increasingly used in industrial applications. Real-time algorithms minimize the feedback delay by reducing the online computational load. The computations are typically divided into two phases: 1) a preparation phase, in which all computations that can be done without knowing $x$ are performed; and 2) the feedback phase, in which the new control input $u_0$ is computed once $x$ is known.

The Real-Time Iteration (RTI) [1] performs a single Sequential Quadratic Programming (SQP) iteration per sampling time. In the preparation phase, all function and derivative evaluations necessary to construct a Quadratic Program (QP) are performed. In the feedback phase, only a single QP is solved. The Multi-level Iteration (MLI) was introduced in [2] and extended in [3], [4]. It is an SQP-based method that offers several variants, also called levels, which only partially update the QP data to reduce the computation time. The levels are sorted by the amount of QP data that is updated, where level A only updates the initial state and level D updates all QP data. The advanced-step controller (ASC) [5] solves in the preparation phase an advanced problem to convergence, i.e., the OCP (1) with a predicted state $x_{\text{pred}}$. In the feedback phase, depending on how active-set changes are handled, it solves a linear system, linear program, or QP [6]. The AS-RTI method [7], [8] combines the RTI and ASC approaches. In the preparation phase, it approximately solves an advanced problem with some MLI variant, and in the feedback phase, like RTI, it solves a single QP. This method adds some flexibility by allowing one to trade off computational complexity for numerical errors. It has good theoretical properties [7], but has only been used in prototypical simulation experiments so far [3], [9].

The open-source acados software package implements efficient algorithms for embedded optimal control, with a focus on SQP-type algorithms that thoroughly exploit the block structure of optimal control problems, such as (1). It is written in C and relies on the high-performance linear algebra package BLASFEO [10]. The high-level interfaces to MATLAB and Python, its flexible problem formulation, and the variety of solver options have made it an attractive option for real-world NMPC applications.

*Contributions.* This paper extends the previously existing analysis of AS-RTI [7], [8], such as analyzing Level B iterations for the first time, which are computationally cheap and converge to suboptimal but feasible solutions of the original problem [2]. It streamlines and significantly simplifies the proofs for the other two levels, C, D, compared to the previous AS-RTI papers [7], [8]. Moreover, it presents an efficient implementation of the AS-RTI method in `acados` [11]. Implementation details and the various algorithmic options are discussed. We extensively test different variants of the implemented algorithm on a benchmark example and demonstrate in a Pareto plot how one can trade-off computational complexity for optimality.

*Outline.* Section II recalls the AS-RTI method, Section III derives novel error bounds, and Section IV discusses implementation details. Section V provides extensive numerical experiments and Section VI summarizes the paper.

## II. The Advanced-Step Real-Time Iteration

This section presents AS-RTI and algorithmic ingredients.

### A. Sequential Quadratic Programming

The parametric NLP (1) can be written more compactly as

$$\min_{w\in\mathbb{R}^{n_w}} \quad f(w) \tag{2a}$$
$$\text{s.t.} \quad 0 = g(w) + Mx, \tag{2b}$$
$$0 \leq h(w), \tag{2c}$$

where the parameter $x$ enters the equality constraint linearly and $M$ is an embedding matrix of appropriate size. Parametric NLPs can always be brought into this form, by introducing auxiliary variables and a linear equality constraint [1]. The functions $f:\mathbb{R}^{n_w}\to\mathbb{R}$, $g:\mathbb{R}^{n_w}\to\mathbb{R}^{n_g}$, $h:\mathbb{R}^{n_w}\to\mathbb{R}^{n_h}$ are assumed to be twice continuously differentiable.

The Langrangian function of (2) is $\mathcal{L}(z) = f(w) - \lambda^\top(g(w)+Mx) - \mu^\top h(w)$ with $z=(w,\lambda,\mu)$ and Lagrange multipliers $\lambda\in\mathbb{R}^{n_g}, \mu\in\mathbb{R}^{n_h}_{\geq 0}$ corresponding to (2b) and (2c), respectively. We denote the primal-dual solution of (2) by $\bar{z}(x) = (\bar{w}(x),\bar{\lambda}(x),\bar{\mu}(x))$, which is under suitable assumptions locally unique [12].

In the following, we consider a sequence of parameters $\{x^k\}_{k\geq 0}$, i.e., the state at each sampling time $t^k$, and perform a fixed number of iterations for a fixed parameter $x^k$. We use the following notation. The solution $\bar{z}(x^k)$ for a fixed $x^k$ is often abbreviated to $\bar{z}^k$. Furthermore, by performing $j$ iterations for a fixed $x^k$, we compute an approximation $z^{k,j}\approx\bar{z}^k$. If only a single iteration is performed, as in RTI, index $j$ is omitted and we write $z^k = z^{k,1}$.

For a fixed parameter $x^k$, a local minimizer of NLP (2) can be computed by an SQP-type method [12]. Given a solution guess $z^{k,0} = (w^{k,0},\lambda^{k,0},\mu^{k,0})$ sufficiently close to a local minimizer, a sequence of QPs is solved

$$\min_{\Delta w} \quad (a^{k,j})^\top \Delta w + \tfrac{1}{2}\Delta w^\top A^{k,j}\Delta w \tag{3a}$$
$$\text{s.t.} \quad g^{k,j} + Mx^k + G^{k,j}\Delta w = 0, \tag{3b}$$
$$h^{k,j} + H^{k,j}\Delta w \geq 0. \tag{3c}$$

The symmetric positive definite matrix $A^{k,j}$ is an approximation of the Lagrange Hessian $\nabla^2_{ww}\mathcal{L}(z^{k,j})$. The vector $a^{k,j} = \nabla_w f(w^{k,j})$ denotes the objective gradient, $g^{k,j} = g(w^{k,j})$, $h^{k,j} = h(w^{k,j})$ the constraint residuals, and $G^{k,j} = \nabla_w g(w^{k,j})^\top$, $H^{k,j} = \nabla_w h(w^{k,j})^\top$ are the Jacobians of the constraints. The primal-dual solution of (3) is denoted $(\Delta w^{k,j}, \lambda^{k,j}_{\mathrm{QP}}, \mu^{k,j}_{\mathrm{QP}})$, and a full SQP step updates the iterates by setting $w^{k,j+1} = w^{k,j}+\Delta w^{k,j}$, $\lambda^{k,j+1} = \lambda^{k}_{\mathrm{QP}}$, and $\mu^{k,j+1} = \mu^{k,j+1}_{\mathrm{QP}}$. Every fixed parameter $x^k$ results in a different sequence of iterates and corresponding QP data, indexed by $j$, hence the use of both indices.

### B. Real-time NMPC algorithms

The QP that computes the feedback for the new parameter $x^{k+1}$ at $t^{k+1}$ is prepared in the preparation phase during $[t^k, t^{k+1}]$. In the RTI, the QP is constructed at the previous output $z^k = z^{k,1}$, computed at time $t^k$. In AS-RTI, the point $z^k$ is further refined by computing an approximate solution to an advanced problem - an NLP (2) with a predicted parameter $x^{k+1}_{\mathrm{pred}}$. The improved linearization point $z^k$ is now denoted by $z^k_{\mathrm{lin}}$. The way the approximation is computed provides a lot of flexibility in algorithm choice and allows for a whole family of different algorithms. Finally, in the feedback phase, QP (3) evaluated at $z^k_{\mathrm{lin}} \approx \bar{z}(x^{k+1}_{\mathrm{pred}})$ and $x^{k+1}$ is solved, resulting in the new output $z^{k+1} = z^{k+1,1}\approx\bar{z}(x^{k+1})$. The main steps of the AS-RTI are:

- (S1) At time $t=t^k$: Predict the initial state $x^{k+1}_{\mathrm{pred}}$ at $t^{k+1}$
- (S2) At $t\in\left[t^k, t^{k+1}\right)$: Starting from the last output $z^k$, iterate on (2) with $x = x^{k+1}_{\mathrm{pred}}$ with some MLI variant, see Sec. II-C, to obtain $z^k_{\mathrm{lin}}$ – ("*the inner iterations*").
- (S3) At $t\in\left[t^k, t^{k+1}\right)$: Construct QP (3) on the linearization point $z^k_{\mathrm{lin}}$.
- (S4) At time $t^{k+1}$, solve (3) with $x = x^{k+1}$.

Note that in the RTI scheme, step (S2) simplifies to setting $z^k_{\mathrm{lin}} = z^k$. In the ASC, $z^k_{\mathrm{lin}}$ is a local minimizer of (2) with $x = x^{k+1}_{\mathrm{pred}}$. Then, in (S4), a linear system or QP is solved to obtain $z^{k+1}$ [5], [6]. If an advanced problem with a perfect prediction $x^{k+1} = x^{k+1}_{\mathrm{pred}}$ is solved to local optimality, there is no numerical error in the feedback, i.e., $z^{k+1} = \bar{z}(x^{k+1})$. We denote AS-RTI with level X iteration as *AS-RTI-X*.

### C. Multi-Level Iterations

In step (S2), we use some Multi-Level Iteration (MLI) variant to compute $z^k_{\mathrm{lin}}$. All MLI levels start with a reference point $\hat{z}^k = (\hat{w}^k,\hat{\lambda}^k,\hat{\mu}^k)$, which can be e.g., the linearization point for the previous feedback phase in the AS-RTI context. The different MLI levels recompute different values of (3) and use evaluations at $\hat{z}^k$ for the others.

*Level D iterations.* Level D iterations are essentially full SQP iterations as described in Section II-A. All functions and derivative derivations are evaluated exactly.

*Level C iterations.* In level C iterations, all matrices in QP (3) are fixed: $\hat{A}^k\approx\nabla^2_{ww}\mathcal{L}(\hat{z}^k)$, $\hat{G}^k = \nabla_w g(\hat{w}^k)^\top$, $\hat{H}^k = \nabla_w g(\hat{w}^k)^\top$. Only the vectors, i.e., $g^{k,j}$, $h^{k,j}$ and $a^{k,j}$, are updated. Note that the SQP subproblem (3) only uses the objective gradient instead of the Lagrangian for $a^{k,j}$, since primal variables are updated in a delta and the duals in an absolute fashion as in [12, Sec. 18.1]. To modify (3) to take the latest multipliers into account in the Lagrange gradient but fix the linearization of the constraints, $a^{k,j}$ needs to be updated

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3412007

for $j \geq 0$ as $a^{k,j} = \nabla \mathcal{L}(w^{k,j}, \lambda^{k,j}, \mu^{k,j}) + \lambda^{k,j} \hat{G}^k + \mu^{k,j} \hat{H}^k$. For a $\hat{z}^k$ close enough to a solution, these iterations converge linearly to a local optimum of (2) [2].

*Level B iterations.* In level B iterations, also called zero-order iterations [13], only functions and no derivatives are evaluated to set up QP (3). More specifically, only $g^{k,j}, h^{k,j}$ are obtained by evaluations. The objective gradient is approximated by $a^{k,j} = \nabla_w f(\hat{w}^k) + \hat{A}(w_{\mathrm{B}}^{k,j} - \hat{w}^k)$. Here, $w_{\mathrm{B}}^{k,j+1}$ are the iterates computed by $w_{\mathrm{B}}^{k,j+1} = w_{\mathrm{B}}^{k,j} + \Delta w^{k,j}, w_{\mathrm{B}}^{k,0} = \hat{w}^k$, $\lambda_{\mathrm{B}}^{k,j+1} = \lambda_{\mathrm{QP}}^{k,j}$ and $\mu_{\mathrm{B}}^{k,j+1} = \mu_{\mathrm{QP}}^{k,j}$, where $(\Delta w^{k,j}, \lambda_{\mathrm{QP}}^{k,j}, \mu_{\mathrm{QP}}^{k,j})$ is the solution of QP (3) with the partially updated data as just described. We call the generated sequence $\{z_{\mathrm{B}}^{k,j}\}_{j \geq 0}$ for a fixed $k$ the *level B iterates*.

**Proposition 1.** (Adapted from [2, Theorem 1.4]) If for a fixed parameter $x^k$, the level B iterates $\{z_{\mathrm{B}}^{k,j}\}$ converge to a limit point $\bar{z}_{\mathrm{B}}(x^k) = (\bar{w}_{\mathrm{B}}(x^k), \bar{\lambda}_{\mathrm{B}}(x^k), \bar{\mu}_{\mathrm{B}}(x^k))$, short $\bar{z}_{\mathrm{B}}^k$, then $\bar{z}_{\mathrm{B}}^k$ is a primal-dual solution of the parametric NLP

$$\min_{w} \quad f(w) + w^\top \beta^k \tag{4a}$$

$$\text{s.t.} \quad 0 = g(w) + M x^k, \tag{4b}$$

$$0 \leq h(w), \tag{4c}$$

$$\text{with } \beta^k = \nabla f(\hat{w}^k) + \hat{A}^k(\bar{w}_{\mathrm{B}}^k - \hat{w}^k) - \nabla f(\bar{w}_{\mathrm{B}}^k) + (\nabla g(\bar{w}_{\mathrm{B}}^k) - (\hat{G}^k)^\top)\bar{\lambda}_{\mathrm{B}}^k + (\nabla h(\bar{w}_{\mathrm{B}}^k) - (\hat{H}^k)^\top)\bar{\mu}_{\mathrm{B}}^k.$$

Therefore, the level B iterations converge to a solution of an NLP that is parametric in $x^k$ and $\beta^k$. The local minimizer of (4) is feasible for (2) but not optimal, as the objective is altered by $w^\top \beta^k$.

*Level A iterations.* Compared to level B iterations, a level A iteration does not evaluate the constraint residuals and only updates them with respect to the parameter, i.e., $g^k = g(\hat{w}^k) + x^k, h^k = h(\hat{w}^k)$. The new solution approximation is obtained by a single QP solve, with $w_{\mathrm{A}}^k = \hat{w}^k + \Delta w$. It is important to note that level A iterations only generate a new value $w_{\mathrm{A}}^k$ for every new parameter value $x^k$, instead of a sequence as in the other levels. The reason is that the QP is a piecewise linear approximation of the solution map $\bar{z}(x)$, which is evaluated at given parameters $x^k$.

## III. Improved error estimates for the AS-RTI

This section recalls some results for the convergence of predictor-corrector methods and derives novel error bounds for different variations of the AS-RTI method.

### A. Error bounds for the feedback phase

For ease of exposition, let us consider only equality-constrained problems, or assume a fixed active set. Later, in Remark 10, we comment on what needs to be changed to generalize the result to inequality-constrained problems. Regard the parametric optimization problem

$$\min_{w} f(w) \quad \text{s.t.} \quad 0 = g(w) + Mx. \tag{5}$$

The Karush–Kuhn–Tucker (KKT) conditions of (5) can be stated as the nonlinear root-finding problem

$$F(z, x) = \begin{bmatrix} \nabla f(w) - \nabla g(w)\lambda \\ g(w) + Mx \end{bmatrix} = 0, \tag{6}$$

with $z = (w, \lambda) \in \mathbb{R}^{n_z}$, $n_z = n_w + n_g$ and the function $F : \mathbb{R}^{n_z} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ which is at least once continuously differentiable. Its zeros are denoted by $\bar{z}(x) = \bar{z}$.

We make a regularity assumption on the solutions of the parametric NLP (5). The Linear Independence Constraint Qualification (LICQ) is said to hold at a point $w$ if the vectors $\nabla g_i(w), i = 1, \ldots, n_g$ are linearly independent. The Second order Sufficient Conditions (SOSC) is said to hold at a KKT point $\bar{z}^k$ if $Z^\top \nabla^2_{ww} \mathcal{L}(\bar{z}^k) Z \succ 0$, where $Z \in \mathbb{R}^{n_w \times (n_w - n_g)}$ is a basis for the null space of $\nabla g(\bar{z}^k)^\top$.

**Assumption 2.** (LICQ, SOSC) For all parameters $x \in X \subseteq \mathbb{R}^{n_x}$, all local minimizers $\bar{z}(x)$ of (2) satisfy the LICQ and SOSC conditions.

The Jacobian of $F(z, x)$, which due to linearity in $x$ only depends on $z$, is denoted by $J(z) := \frac{\partial (F(z,x))}{\partial z}$. Recall that $z^{k+1} \approx \bar{z}(x^{k+1})$ and $z^k \approx \bar{z}(x^k)$. A predictor-corrector step for the parametric root-finding problem (6) can be written

$$z^{k+1} = z^k - J(z^k)^{-1} F(z^k, x^{k+1}). \tag{7}$$

Note the dependence on the next parameter $x^{k+1}$. If a fixed $x^k$ is used instead, equation (7) reduces to an exact Newton-step, a corrector step. If $z^k = \bar{z}(x^k)$, due to the linearity of $F(z, x)$ in $x$, equation (7) reduces to a predictor step: $z^{k+1} = z^k + \frac{\partial \bar{z}(x^k)}{\partial x}(x^{k+1} - x^k)$, see [14, Sec. 1.2] for details.

We proceed by restating some results that we use to study the AS-RTI error, starting with the convergence of Newton's method, cf. [15, Sec. 8.3.3.]. Suppose that the parameter $x^k$ is fixed, then (7) is simply a standard exact Newton-step $z^{k,j+1} = z^{k,j} - J(z^{k,j})^{-1} F(z^{k,j}, x^k)$. One may also use a Jacobian approximation $M^{k,j} \approx J(z^{k,j})$.

**Assumption 3.** ($\omega$ and $\kappa$ conditions) There exist $\omega < \infty$ and $\kappa < 1$ such that, for any fixed $x^k$, given iterate $z^{k,j}$ and Jacobian approximation $M^{k,j}$, the following holds:

(a) $\|(M^{k,j})^{-1}(J(z^{k,j}) - J(z))\| \leq \omega \|z^{k,j} - z\|, \ \forall z,$
(b) $\|(M^{k,j})^{-1}(J(z^{k,j}) - M^{k,j})\| \leq \kappa.$

Here, (a) is a rescaled Lipschitz condition on the Jacobian, and (b) measures the Jacobian approximation error. For the exact Jacobians $(M^{k,j} = J(z^{k,j}))$ it holds that $\kappa = 0$.

**Assumption 4.** (Initialization) A starting point $z^{k,0}$ of a sequence generated by a Newton-type method satisfies

$$\|\bar{z}^k - z^{k,0}\| < r_z = \tfrac{2(1-\kappa)}{\omega}. \tag{8}$$

Assumption 4 tells how close to a solution one must initialize so that Newton's method converges with full steps. Both Ass. 3 and 4 are standard for local Newton-type convergence analysis to state the following theorem [1], [2], [13], [14].

**Theorem 5.** (Newton-type convergence, cf. [15, Thm. 8.7]) Regard the continuously differentiable function $F$ in (6) with a fixed $x^k$, and a solution $\bar{z}(x^k)$ (short $\bar{z}^k$) with $F(\bar{z}^k, x^k) = 0$. If Ass. 3 holds, the sequence $\{z^{k,j}\}$ generated by $z^{k,j+1} = z^{k,j} - (M^{k,j})^{-1} F(z^{k,j}, x^k)$ satisfies the inequality

$$\|\bar{z}^k - z^{k,j+1}\| \leq \left(\kappa + \tfrac{\omega}{2}\|\bar{z}^k - z^{k,j}\|\right) \|\bar{z}^k - z^{k,j}\|. \tag{9}$$

Moreover, if Assumption 4 holds, then the sequence $\{z^{k,j}\}$ converges to $\bar{z}^k$ as $j \to \infty$.

Define $\alpha^k := \kappa + \frac{\omega}{2}\|\bar{z}^k - z^{k,0}\|$, with $\alpha^k < 1$ due to Ass. 4. Applying (9) recursively, results in the useful inequality

$$\|\bar{z}^k - z^{k,j+1}\| \leq (\alpha^k)^{j+1}\|\bar{z} - z^{k,0}\|. \quad (10)$$

Next, we look at the solution map $\bar{z}(x)$ for different parameters $x$. Assumption 2 implies that $J(\bar{z}(x))$ is invertible for all $x \in X$, cf. [12, Lem. 16.1]. Thus, applying the implicit function theorem [12, Thm. A.1] to (6) entails that $\bar{z}(x)$ is locally unique and Lipschitz continuous

$$\|\bar{z}(x^{k+1}) - \bar{z}(x^k)\| \leq \sigma\|x^{k+1} - x^k\|. \quad (11)$$

By applying a single predictor-corrector step (7) for the parameters $x^{k+1}$ and $x^k_{\text{lin}}$, the corresponding solution map approximations $z^{k+1} \approx \bar{z}(x^{k+1})$ and linearization point $z^k_{\text{lin}} \approx \bar{z}(x^k_{\text{lin}})$ can be related by combining (9) and (11) as follows.

**Theorem 6.** (Adapted from [14, Th. 3.5], [13, Lem. 3.1.5]) Regard the continuously differentiable function $F$ in (6), and solution $\bar{z}(x)$ with $F(\bar{z}(x), x) = 0$. Let Assumptions 2 and 3 hold. Then, the iterates $z^{k+1}$ and $\bar{z}^k_{\text{lin}} = z^k$ generated by (7) for the parameters $x^{k+1}$ and $x^k_{\text{lin}} = x^k$ satisfy

$$\begin{aligned}\|\bar{z}^{k+1} - z^{k+1}\| &\leq (\kappa\sigma + \tfrac{\omega\sigma^2}{2}\|x^{k+1} - x^k_{\text{lin}}\|)\|x^{k+1} - x^k_{\text{lin}}\| \\ &+ (\kappa + \omega\sigma\|x^{k+1} - x^k_{\text{lin}}\| + \tfrac{\omega}{2}\|\bar{z}^k_{\text{lin}} - z^k_{\text{lin}}\|)\|\bar{z}^k_{\text{lin}} - z^k_{\text{lin}}\|. \quad (12)\end{aligned}$$

Note that index $j$ is omitted since a single Newton-type step is computed for every parameter. Furthermore, if for some fixed $x^0$ the point $z^0 \approx \bar{z}^0$ satisfies Ass. 4, there exists $r_x$ (depending on $\sigma, \kappa$ and $\omega$) such that for $\|x^{k+1} - x^k\| < r_x$, the sequence $\{z^k\}$ generated by (7) remains bounded with $\|z^k - \bar{z}(x^k)\| < r_z$, cf. [14, Corollary 3.6.].

### B. Error bounds for the preparation phase

Given a linearization point $z^k_{\text{lin}}$ and $x^k_{\text{lin}}$, in the feedback phase, i.e., step (S4) of AS-RTI, a QP is solved to obtain an approximation $z^{k+1} \approx \bar{z}(x^{k+1})$. In the absence of inequalities, instead of a QP, the linear system (7) is solved, and we have the error bound in (12). We see from (12) that the accuracy of the new output $z^{k+1}$ improves with a smaller difference between the parameters via $\|x^{k+1} - x^k_{\text{lin}}\|$ and higher accuracy of the linearization point via $\|\bar{z}^k_{\text{lin}} - z^k_{\text{lin}}\|$. The goal of AS-RTI is to reduce $\|\bar{z}^k_{\text{lin}} - z^k_{\text{lin}}\|$ by solving an advanced problem with $x^k_{\text{lin}} = x^{k+1}_{\text{pred}} \approx x^{k+1}$ performing iterations with some MLI variant, which results in tighter bounds for $\|\bar{z}^{k+1} - z^{k+1}\|$ in (12). We proceed by quantifying the error for each variant.

*Level D iterations.* Here, we use Newton-type steps in (7) with $M^{k,j}_{\text{D}} \approx J(z^{k,j})$ and assume that Ass. 3 holds for some $\kappa_{\text{D}}$ and $\omega_{\text{D}}$. Starting with the previous output $z^k$ and predicted parameter $x^{k+1}_{\text{pred}}$, and assuming that $\|z^k - \bar{z}(x^{k+1}_{\text{pred}})\| < \frac{2(1-\kappa_{\text{D}})}{\omega_{\text{D}}}$, a modification of Ass. 4, we carry out $N_{\text{D}}$ Newton-type iterations to obtain $z^k_{\text{lin}}$. Theorem 5 and (10) yield

$$\|z^k_{\text{lin}} - \bar{z}(x^{k+1}_{\text{pred}})\| \leq (\alpha^k_{\text{D}})^{N_{\text{D}}}\|z^k - \bar{z}(x^{k+1}_{\text{pred}})\|. \quad (13)$$

with $\alpha^k_{\text{D}} = \kappa_{\text{D}} + \frac{\omega_{\text{D}}}{2}\|z^k - \bar{z}(x^{k+1}_{\text{pred}})\| < 1$. In the limiting case where $x^{k+1}_{\text{pred}} = x^{k+1}$ and $j \to \infty$, we see that in the right-hand side of (12) becomes zero, i.e. $z^{k+1} = \bar{z}^{k+1}$.

*Level C iterations.* In level C, we essentially proceed as in level D, except that we use a constant Jacobian approximation $M^{k,j} = M^k_{\text{C}} = \begin{bmatrix} \hat{A}^k & (\hat{G}^k)^\top \\ \hat{G}^k & 0 \end{bmatrix}$ computed at the reference point $\hat{z}^k = z^{k-1}$. We assume that that Ass. 3 holds for some $\kappa_{\text{C}}$ and $\omega_{\text{C}}$. Similar to level D, starting with the previous output, $z^k$ and $x^{k+1}_{\text{pred}}$, and assuming that $\|z^k - \bar{z}(x^{k+1}_{\text{pred}})\| < r_{z,\text{C}} = \frac{2(1-\kappa_{\text{C}})}{\omega_{\text{C}}}$, a modification of Ass. 4, we perform $N_{\text{C}}$ Newton-type iterations to obtain $z^k_{\text{lin}}$. Applying Thm 5 and (10) yields

$$\|z^k_{\text{lin}} - \bar{z}(x^{k+1}_{\text{pred}})\| \leq (\alpha^k_{\text{C}})^{N_{\text{C}}}\|z^k - \bar{z}(x^{k+1}_{\text{pred}})\|. \quad (14)$$

with $\alpha^k_{\text{C}} = \kappa_{\text{C}} + \frac{\omega_{\text{C}}}{2}\|z^k - \bar{z}(x^{k+1}_{\text{pred}})\| < 1$. Since $M^k_{\text{C}}$ is fixed, it is a less accurate but computationally cheaper Jacobian approximation and usually $\kappa_{\text{D}} \ll \kappa_{\text{C}}$ holds. For similar $\omega_{\text{D}}$ and $\omega_{\text{C}}$, it follows that $r_{z,\text{C}} < r_{z,\text{D}}$. In other words, the previous output $z^k$ must be closer to the solution $\bar{z}(x^{k+1}_{\text{pred}})$ for level C than for D to achieve contraction of the iterates.

*Level B iterations.* Following Proposition 1, Level B iterations for NLPs without inequality constraints converge to a solution of

$$\min_{w} f(w) + w^\top \beta^k \quad \text{s.t.} \quad 0 = g(w) + Mx^k, \quad (15)$$

with $\beta^k = \nabla f(\hat{w}^k) + \hat{A}^k(\bar{w}^k_{\text{B}} - \hat{w}^k) - \nabla f(\bar{w}^k_{\text{B}}) + (\nabla g(\bar{w}_{\text{B}})^k - (\hat{G}^k)^\top)\bar{\lambda}^k_{\text{B}}$. The KKT conditions of (15) read

$$F_{\text{B}}(z, \beta^k, x^k) = F(z, x^k) + [(\beta^k)^\top, 0]^\top. \quad (16)$$

A sequence of iterates is generated via $z^{k,j+1} = z^{k,j} - (M^k_{\text{B}})^{-1}F_{\text{B}}(z^j, \beta, x^{k+1}_{\text{pred}})$, where the parameters $\beta^k$ and $x^{k+1}_{\text{pred}}$ are fixed, and $M^k_{\text{B}} = M^k_{\text{C}}$. We have the following estimate.

**Proposition 7.** Assume that LICQ and SOSC hold for (15) at all $\bar{z}_B(x, \beta)$, for $x \in X$ and $\beta \in \mathbb{R}^{n_w}$. Suppose that Ass. 3 holds for $z^{k,j+1} = z^{k,j} - (M^k_{\text{B}})^{-1}F_{\text{B}}(z^j, \beta^k, x^{k+1}_{\text{pred}})$ with constants $\kappa_{\text{B}}$ and $\omega_B$, and that $\|z^k - \bar{z}^k_{\text{B}}\| < \frac{2(1-\kappa_{\text{B}})}{\omega_{\text{B}}}$. Then the sequence of iterates $\{z^j_{\text{B}}\}_{j=1,\ldots,N_{\text{B}}}$ fulfills

$$\|\bar{z}(x^{k+1}_{\text{pred}}) - z^k_{\text{lin}}\| \leq \sigma_{\text{B}}\|\beta^k\| + (\alpha^k_{\text{B}})^{N_{\text{B}}}\|\bar{z}_{\text{B}}(x^{k+1}_{\text{pred}}, \beta^k) - z^k\|,$$

where $\sigma_B$ is the Lipschitz constant of $\bar{z}_{\text{B}}(x, \beta)$ and $\alpha^k_{\text{B}} = \kappa_{\text{B}} + \frac{\omega_B}{2}\|z^k - \bar{z}_B(x^{k+1}_{\text{pred}}, \beta^k)\| < 1$.

*Proof.* Adding and subtracting $\bar{z}_{\text{B}}(x, \beta^k)$ in the left term of the next equation, and using the triangle inequality we get

$$\|\bar{z}(x) - z^k_{\text{lin}}\| \leq \|\bar{z}(x) - \bar{z}_{\text{B}}(x, \beta^k)\| + \|\bar{z}_{\text{B}}(x, \beta^k) - z^k_{\text{lin}}\|.$$

For the first term on the right, we note that $\bar{z}_{\text{B}}(x, 0) = \bar{z}(x)$. Since LICQ, SOSC hold for (15) by applying the implicit function theorem to (16), we have that $\|\bar{z}(x) - \bar{z}_{\text{B}}(x, \beta)\| \leq \sigma_{\text{B}}\|\beta\|$. For the second term on the left, we apply (10) with $N_{\text{B}}$ iterations, and obtain that $\|\bar{z}_{\text{B}}(x, \beta) - z^k_{\text{lin}}\| \leq (\alpha_{\text{B}})^{N_{\text{B}}}\|\bar{z}_{\text{B}}(x^{k+1}_{\text{pred}}, \beta) - z^k\|$. By using these two terms and $x = x^{k+1}_{\text{pred}}$ the result of this proposition is obtained. □

Even if level B iterations are fully converged $j \to \infty$, the error $\|\bar{z}(x^{k+1}_{\text{pred}}) - z^k_{\text{lin}}\| \leq \sigma_{\text{B}}\|\beta\|$ remains, as they converge to feasible but suboptimal points of (2), cf. Prop. 1. Analogously to C and D, the previous output $z^k$ has to be close enough to $\bar{z}_B(x^{k+1}_{\text{pred}})$ such that the iterates contract.

*Level A iterations.* Applying Thm 6 with $x^k_{\text{lin}} = x^{k+1}_{\text{pred}}$ yields an error bound for the feedback phase of AS-RTI-A.

**Proposition 8.** Let the assumptions of Theorem 6 hold. When $z^k_{\text{lin}}$ is obtained via a Level A iteration $z^k_{\text{lin}} = z^{k-1} - (M^{k-1})^{-1}F(z^{k-1}, x^{k+1}_{\text{pred}})$, the feedback error satisfies

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3412007

$$\|\bar{z}^{k+1} - z^{k+1}\| \leq (\kappa\sigma + \tfrac{\omega\sigma^2}{2}\|x^{k+1} - x^{k+1}_{\text{pred}}\|)\|x^{k+1} - x^{k+1}_{\text{pred}}\| + (\kappa + \omega\sigma\|x^{k+1} - x^{k+1}_{\text{pred}}\| + \tfrac{\omega}{2}\|\bar{z}^{k}_{\text{lin}} - z^{k}_{\text{lin}}\|)\|\bar{z}^{k}_{\text{lin}} - z^{k}_{\text{lin}}\|. \quad (17)$$

Compared to RTI, where $x^k_{\text{lin}} = x^k$ and $z^k_{\text{lin}} = z^k$, AS-RTI-A can significantly reduce the error if $x^{k+1}_{\text{pred}} \approx x^{k+1}$ and if the predictor provides a good approximation $z^k_{\text{lin}} \approx \bar{z}(x^{k+1}_{\text{pred}}) = \bar{z}^k_{\text{lin}}$.

**Remark 9.** The same linearization at $z^{k-1}$ is reused to compute both an approximation for $x^{k+1}_{\text{pred}}$ and $x^k$. It is important to start from $z^{k-1}$ for computing $z^k_{\text{lin}}$ instead of $z^k$ to avoid taking the same corrector step twice, cf. (7).

**Remark 10** (Extension to inequality constraints)**.** All results are derived from two inequalities: convergence of Newton's method in Theorem 5, and Lipschitz continuity of $\bar{z}(x)$ in (11). The KKT conditions of inequality constrained NLP (2) can be written as a generalized equation solved by the Newton-Josephy method, which is equivalent to SQP. Using Robinson's strong regularity, which is implied by strong SOSC and LICQ, Theorem 5 can be generalized via [14, Thm. 3.5] and Eq. (11) via [14, Lem. 3.3].

## IV. Implementation in acados

Next, we discuss some practical aspects for an efficient implementation of AS-RTI within the acados software, which has been developed as part of this work.

*a) Condensing and QP solution with two phases:* The acados software offers a variety of QP solvers. HPIPM offers efficient methods to transform OCP-structured QPs into dense ones or ones with a shorter horizon by full and partial condensing [16]. Since in the preparation phase all matrices of the QP are readily available, most of the condensing operations can be performed in that phase. Moreover, it is common to eliminate the initial state variable from the QP. An efficient split of operations is realized by implementing functions that assume that only matrices of QP (3) are known and a second one that completes the computations once the vector quantities are known. This split functionality is utilized when implementing the level A, B, and C iterations.

*b) Advancing:* There are two main strategies to set up the advanced problem, (S1). 1) Simulate with $\phi_0(\cdot)$ internally at the current SQP iterate. 2) Simulate the system externally. Both strategies work in general for nonuniform discretization grids in (1), which have been shown to be superior with respect to uniform ones in [17]. Option 1) assumes that $\phi_0$ models the systems evolution over the sampling time. In contrast, 2) does not require this assumption and allows using a higher fidelity model than used in the OCP.

## V. Numerical experiments

This section compares different MPC controllers in an open-source numerical simulation study [18] using acados v0.3.2 [19] via Python on a Laptop with an Intel i5-8365U CPU, 16 GB of RAM running Ubuntu 22.04.

*a) Inverted pendulum on cart test problem:* The differential state of the model is $x = [p, \theta, s, \omega]^\top$ with cart position $p$, cart velocity $s$, angle of the pendulum $\theta$ and angular velocity $\omega$. The control input $u$ is a force acting on the cart in the horizontal plane. The ODE, describing the system dynamics can be found e.g. in [11]. In our OCP formulation, $u$ is constrained to be in $[-40, 40]$. The goal is to stabilize the system in the unstable upright position driving all states to zero. We formulate the linear least squares cost $l(x, u) = x^\top Qx + u^\top Ru$ with cost weights are $Q = \text{diag}(100, 10^3, 0.01, 0.01)$, $R = 0.2$. The terminal cost term is set to $E(x) = x^\top Px$, where $P$ is obtained as a solution of the discrete algebraic Riccati equation with cost and dynamics linearized at the steady state.

TABLE I

Timings, relative suboptimality, stationarity residual and constraint violation for different controllers.

| algorithm | max. time prepare [ms] | max. time feedback [ms] | relative subopt. [%] | mean $10^3\|g\|$ | mean $\|\nabla_w \mathcal{L}\|$ |
|---|---|---|---|---|---|
| SQP-100 | 0.00 | 5.518 | 0.05 | 0.00 | 0.00 |
| SQP-2 | 0.00 | 0.264 | 0.25 | 12.99 | 7.75 |
| AS-RTI-D-2 | 0.35 | 0.020 | 0.04 | 0.74 | 1.28 |
| AS-RTI-D-1 | 0.23 | 0.020 | 0.25 | 8.26 | 5.38 |
| AS-RTI-C-2 | 0.32 | 0.020 | 0.04 | 5.34 | 2.20 |
| AS-RTI-C-1 | 0.23 | 0.021 | 0.24 | 10.70 | 5.85 |
| AS-RTI-B-2 | 0.28 | 0.021 | 0.57 | 1.28 | 8.20 |
| AS-RTI-B-1 | 0.19 | 0.021 | 0.54 | 8.77 | 7.98 |
| AS-RTI-A | 0.13 | 0.019 | 0.54 | 12.41 | 6.63 |
| RTI | 0.11 | 0.022 | 3.55 | 14.06 | 8.26 |

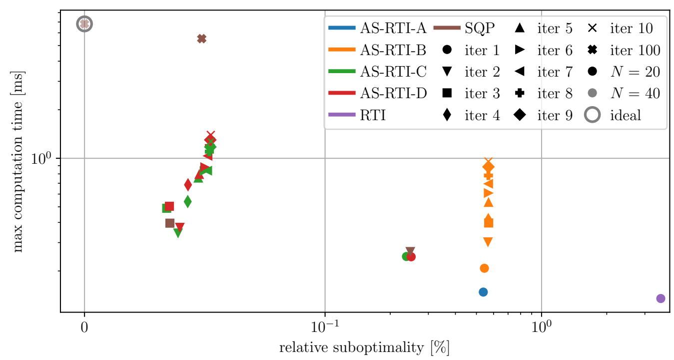


Fig. 1. Pareto plot: timings vs. relative suboptimality.

*b) Scenario:* The system is simulated for four seconds at a sampling time of $\Delta t = 0.05$s. We simulate 20 different scenarios, in each of which, the system starts at an upward position with a random initial value for $p$ and is disturbed at two time instances, at 0s and 2s by overwriting the control action with random value in $[-100, 100]$. The OCP is formulated with a time horizon of 2s divided into $N = 20$ shooting intervals, the first is of length $\Delta t$ and the remaining uniformly split the rest of the time horizon. The dynamics are discretized using one step of an implicit Radau IIA method of order three with three Newton iterations on each shooting interval, respectively of order seven with 20 Newton iterations for the simulation step of the plant.

*c) Controllers:* We apply a variety of controllers and report their performance in Table I. All controllers use the full condensing functionality from HPIPM [16] and the active-set solver DAQP [20]. The solvers labeled *SQP $n$* apply $n$ SQP iterations. Different variants of AS-RTI controllers are labeled *AS-RTI-X-$n$* performing $n$ level X iterations on the advanced problem in each preparation phase. Additionally, we compare with a plain *RTI* controller.

*d) Evaluation:* The Pareto plot in Fig. 1 compares the controller variants in terms of maximum computation time and relative suboptimality. The latter is evaluated by computing the closed-loop cost and comparing with a controller that uses

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3412007

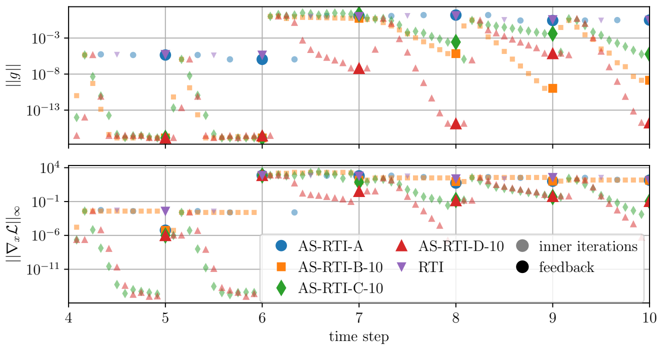


Fig. 2. Primal and dual residuals of inner iterations and the applied feedback steps for RTI and different AS-RTI variants.

a finer uniform discretization grid with $N = 40$ and fully converged SQP, which is marked as *ideal* in Fig. 1. In Table I, we additionally report the maximum timings for preparation and feedback phase over all simulations. Additionally, the mean values of the constraint violation and the Lagrange gradient over all simulation steps are listed. The shooting gaps show satisfaction of nonlinear constraints, quantified as $\|g\|$. The example only contains linear inequality constraints, which are always satisfied. Thus, it allows one to compare closed-loop suboptimality and constraint violation simultaneously. Figure 2 shows how primal and dual infeasibility evolve over all AS-RTI iterations over a few time steps after applying a large disturbance to the plant.

*e) Discussion:* Firstly, we can see from Table I that the timings of the feedback step for all AS-RTI variations are consistent and a multiple lower compared to their preparation phase, which is enabled by the split condensing described in Sec. IV. In our closed-loop simulation, the delay from the computation of the feedback phase is neglected, which would impact all real-time controllers similarly, but would drastically degrade the performance of the SQP algorithms included for reference. The classic RTI algorithm has a rather high suboptimality and is the fastest real-time algorithm.

AS-RTI-A, with a single precondensed QP solve, is able to greatly improve on the classic RTI algorithm with much better performance and a marginally increased computational load, cf. Fig. 1. In Figure 2, one can observe that level B iterations converge to a feasible linearization point. This iterate is associated with a fixed suboptimality which is visible in the second subplot and consistent Proposition 7.

In contrast to the level B iterations, the level C iterations converge to a feasible locally optimal point. The same holds for level D iterations. Since iterations of level D are more accurate than level C, it follows that $\kappa_{\mathrm{D}} < \kappa_{\mathrm{C}}$, which implies a faster error reduction in (13) compared to (14). This can be observed in Figure 2 which shows faster convergence for D, i.e., the steeper slope in both primal and dual infeasibility.

Overall, we see in Figure 2 that the feedback iterates of all AS-RTI variants are more accurate than the one of plain RTI. This is due to the tighter bound in (12), more precisely, the reduced error in the linearization point $\|\bar{z}^k_{\mathrm{lin}} - z^k_{\mathrm{lin}}\|$, and the smaller parameter difference $\|x^{k+1} - x^{k+1}_{\mathrm{pred}}\|$ for AS-RTI compared to $\|x^{k+1} - x^k\|$ for plain RTI.

## VI. Conclusion

This paper streamlines and extends the existing analysis of AS-RTI with Multi-Level Iterations (MLI) of all for levels. It is shown that if the current solution is sufficiently close to the next solution, the numerical error can be reduced by a few computationally cheap MLI iterations. Furthermore, an efficient implementation in the open-source package `acados` is presented making the method widely available for real-world applications on embedded hardware. Numerical examples confirm the theory and demonstrate how to assemble efficient variants of the AS-RTI method. In particular, AS-RTI-A can significantly improve control performance over standard RTI at only 20% additional cost per sampling time, by computing a single additional QP solution to the advanced problem.

## References


[1] M. Diehl, "Real-time optimization for large scale nonlinear processes," Ph.D. dissertation, Univ. of Heidelberg, 2001.
[2] H. G. Bock, M. Diehl, E. A. Kostina, and J. P. Schlöder, "Constrained optimal feedback control of systems governed by large differential algebraic equations," in *Real-Time and Online PDE-Constrained Optimization*. SIAM, 2007, pp. 3–22.
[3] A. Nurkanović, S. Albrecht, and M. Diehl, "Multi-level Iterations for Economic Nonlinear Model Predictive Control," *Recent Advances Model Predictive Control: Theory, Algorithms, Appl.*, 2021.
[4] L. Wirsching, "Multi-level iteration schemes with adaptive level choice for nonlinear model predictive control," Ph.D. dissertation, Univ. of Heidelberg, 2018.
[5] V. M. Zavala and L. T. Biegler, "The advanced step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, 2009.
[6] J. Jäschke, X. Yang, and L. T. Biegler, "Fast economic model predictive control based on nlp-sensitivities," *J. Process Control*, vol. 24, no. 8, pp. 1260–1272, 2014.
[7] A. Nurkanović, A. Zanelli, S. Albrecht, and M. Diehl, "The Advanced Step Real Time Iteration for NMPC," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2019.
[8] A. Nurkanović, A. Zanelli, G. Frison, S. Albrecht, and M. Diehl, "Contraction properties of the advanced step real-time iteration," in *Proc. the IFAC World Congr.*, vol. 51, no. 1, 2020.
[9] A. Nurkanović, A. Mešanović, A. Zanelli, J. Frey, G. Frison, S. Albrecht, and M. Diehl, "Real-time nonlinear model predictive control for microgrid operation," in *Proc. Amer. Control Conf. (ACC)*, 2020.
[10] G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, and M. Diehl, "BLASFEO: Basic linear algebra subroutines for embedded optimization," *ACM Trans. Math. Softw.*, vol. 44, no. 4, pp. 42:1–42:30, 2018.
[11] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, Oct 2021.
[12] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Ser. Operations Research and Financial Eng., 2006.
[13] A. Zanelli, "Inexact methods for nonlinear model predictive control: stability, applications, and software," Ph.D. dissertation, Univ. of Freiburg, 2021.
[14] Q. Tran-Dinh, C. Savorgnan, and M. Diehl, "Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization," *SIAM J. Optimization*, vol. 22, no. 4, 2012.
[15] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill, 2017.
[16] G. Frison, D. Kouzoupis, J. B. Jørgensen, and M. Diehl, "An efficient implementation of partial condensing for nonlinear model predictive control," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2016.
[17] J. Frey, K. Baumgärtner, and M. Diehl, "Gauss-Newton Runge-Kutta integration for efficient discretization of optimal control problems with long horizons and least-squares costs," in *accepted for ECC 2024*. [Online]. Available: https://arxiv.org/abs/2310.00618
[18] J. Frey, "AS-RTI code," https://doi.org/10.5281/zenodo.11065524.
[19] "acados releases," https://www.doi.org/10.5281/zenodo.7371687.
[20] D. Arnstrom, A. Bemporad, and D. Axehill, "A dual active-set solver for embedded quadratic programming using recursive LDL$^T$ updates," *IEEE Trans. Automatic Control*, 2022.