The Advanced Step Real Time Iteration for NMPC

Armin Nurkanović, Andrea Zanelli, Sebastian Albrecht, Moritz Diehl

Abstract— **This paper introduces an extension to the well-known Real Time Iteration (RTI) for Nonlinear Model Predictive Control (NMPC). We combine algorithmic ideas of the RTI, Advanced Step Controller and Multi-Level Iteration (MLI) framework and get thereby a family of new algorithms that allow one to trade control performance for computational efficiency in a flexible way. The main idea is to improve the linearization point for a new iteration by making cheap iterations with a new initial parameter prediction. We derive a general contraction estimate for the new algorithm and show that this approach yields closer tracking of the optimal solution manifold and results in better control performance. The efficacy of our approach is shown on a nontrivial numerical example.**

I. Introduction

Nonlinear Model Predictive Control (NMPC) is becoming more and more a standard tool in academia and industry [21]. It enables one to incorporate nonlinear dynamics and constraints directly into an Optimal Control Problem (OCP). When using NMPC to control a system, one has to solve online a series of parametric OCPs with different initial states. In each of these OCPs the latest information about the system state is incorporated. Solving optimization problems online is in general a computationally intensive task. However, recent progress both in software [23] and numerical algorithms [15], [17], [19] made it possible to achieve computation times in the range of milli- and micro-second timescales for various kinds of applications.

Since feedback delays can largely degrade control performance, in many online algorithms the computations are divided into an expensive and long *preparation phase*, where calculations can be performed without the knowledge of the current measurement, and a short *feedback phase* [9]. In the feedback phase just a few calculations are performed to take into account the new measurement, such that the feedback delay can be reduced. Examples of such algorithms are the C/GMRES algorithm [18], the Sequential Quadratic Programming (SQP)-based Real Time Iteration [8], the Multi Level Iteration (MLI) [4] and the interior-point-based Advanced Step Controller (ASC) by Zavala and Biegler [30].

In online optimization it is usually easy to achieve recursive feasibility between two subsequent optimization problems with nominal system dynamics. However, ensuring recursive optimality is a more challenging task [21]. Since in online optimization the feedback has typically to be delivered at a fixed rate, in practice there is usually enough time to perform few additional calculations in the preparation phase, which should be used to ensure also recursive optimality in the nominal case.

A. Contributions and Outline

This paper presents a new variant of the Real Time Iteration, which we denote as the *Advanced Step Real Time Iteration* (AS-RTI). The main idea is to do inexact Newton steps or only predictor steps on an advanced problem to improve the linearization point for the next Real Time Iteration. An advanced problem is an OCP with a predicted initial value, based on the last control input and the last actual measurement. Instead of doing simple *post iterations*, i.e. further iterations with the fixed old state estimate, we do a few cheap iterations as in the MLI [4], but on the advanced problem as in the Advanced Step Controller [30]. This allows us to combine the benefits of both approaches and thereby one can assemble a large number of different NMPC schemes. The benefits of this simple idea are formalized in Theorem 6 and further explained in the discussion thereafter.

In Section II, we introduce the NMPC problem and algorithm class we aim to tackle. This is followed by a brief presentation of the RTI, MLI and the Advanced Step Controller in Section III, which serve as building blocks for our new algorithm. In Section IV, we present the AS-RTI and variants of it and describe the algorithmic features in detail, followed by Section V, where we prove a general contraction estimate for the new family of algorithms. In Section VI, we show the efficacy of our approach on a nontrivial numerical example where we show better performance compared to the standard RTI using the same sampling time.

II. Nonlinear Model Predictive Control and Continuation Methods

In this paper, we will consider the following discrete time OCP, which can e.g be obtained with the Direct Multiple Shooting parameterization [5] of a continuous time OCP:

This research was supported by the German Federal Ministry of Education and Research (BMBF) via the funded Kopernikus project: SynErgie (03SFK3U0) and by the German Federal Ministry for Economic Affairs and Energy (BMWi) via eco4wind (0324125B) and DyConPV (0324166B), and by DFG via Research Unit FOR 2401.

Armin Nurkanović is with Siemens Corporate Technology, 81739 Munich, Germany and the Department of Microsystems Engineering (IMTEK) University Freiburg, 79110 Freiburg, Germany and Sebastian Albrecht is with Siemens Corporate Technology, 81739 Munich, Germany {armin.nurkanovic, sebastian.albrecht}@siemens.com

Andrea Zanelli is with the Department of Microsystems Engineering (IMTEK) and Moritz Diehl is with the Department of Microsystems Engineering (IMTEK) and Department of Mathematics, University Freiburg, 79110 Freiburg, Germany {andrea.zanelli, moritz.diehl}@imtek.uni-freiburg.de

$$
\begin{aligned}
\min_{\substack{u_0,\ldots,u_{N-1}\\ x_0,..x_N}} \quad & \sum_{i=0}^{N-1} l_i(x_i,u_i) + l_N(x_N) && \text{(1a)}\\
\text{s.t.} \quad & x_0 - \xi = 0, && \text{(1b)}\\
& x_{i+1} - f(x_i,u_i) = 0,\ i = 0,\ldots,N-1, && \text{(1c)}\\
& h(x_i,u_i) \geq 0, \qquad\quad i = 0,\ldots,N-1, && \text{(1d)}\\
& r(x_N) \geq 0. && \text{(1e)}
\end{aligned}
$$

Here, N is the horizon length, the optimization variables are $x_i \in \mathbb{R}^{n_x}$, $u_i \in \mathbb{R}^{n_u}$, and the objective function contributions $l_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ and $l_N : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$. The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ is the state transition map which is usually obtained by numerical integration. The two functions $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_h}$ and $r : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_r}$ define the inequality constraints. The vector ξ denotes the initial value for the state vector. We assume all functions to be at least two times continuously differentiable.

When using NMPC, we solve the OCP (1) at every sampling time with a new ξ and use the optimal solution $\bar{u}_0(\xi)$ to control the process. This Nonlinear Program (NLP) can be written in the following compact form:

$$
\begin{aligned}
\min_{w} \quad & \phi(w) && \text{(2a)}\\
\text{s.t.} \quad & b(w) + \Lambda\xi = 0, && \text{(2b)}\\
& c(w) \geq 0, && \text{(2c)}
\end{aligned}
$$

where $\Lambda = [-I, 0, \dots]^T$ is a suitable matrix that embeds the parameter ξ linearly and w collects all state and control variables:

$$
w = [x_0^{\mathrm{T}}, u_0^{\mathrm{T}}, x_1^{\mathrm{T}}, u_1^{\mathrm{T}}, \ldots, x_{N-1}^{\mathrm{T}}, u_{N-1}^{\mathrm{T}}, x_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n_w},
$$

with $n_w = (N+1)\cdot n_x + N \cdot n_u$. The function $b(w)$ collects the equality constraints (1b) and (1c), while $\phi(w)$ represents the cost function, and $c(w)$ collects the inequality constraints (1d) and (1e). The Lagrangian of the NLP (2) reads as

$$
\mathcal{L}(w,\lambda,\mu) = \phi(w) - \lambda^{\mathrm{T}} b(w) - \lambda^{\mathrm{T}}\Lambda\xi - \mu^{\mathrm{T}} c(w), \qquad (3)
$$

where $\lambda \in \mathbb{R}^{(N+1)\cdot n_x}$ and $\mu \in \mathbb{R}^{N \cdot n_h + n_r}$ denote the vectors containing the Lagrange multipliers.

A. Newton-type Optimization

The NLP (2) can be solved to local optimality with standard NLP algorithms [16]. Although other approaches exist [16], in this paper, we will focus on SQP. Assuming we start with a primal-dual guess (w^0, λ^0, μ^0) close enough to the solution, a full SQP step is performed as

$$
w^{k+1} = w^k + \Delta w^k,\ \lambda^{k+1} = \lambda^k_{\mathrm{QP}},\ \mu^{k+1} = \mu^k_{\mathrm{QP}}, \qquad (4)
$$

where $(\Delta w^k, \lambda^k_{\mathrm{QP}}, \mu^k_{\mathrm{QP}})$ corresponds to the primal-dual solution of the Quadratic Program (QP)

$$
\begin{aligned}
\min_{\Delta w} \quad & \frac{1}{2}\Delta w^{\mathrm{T}} A^k \Delta w + a^{k\mathrm{T}}\Delta w && \text{(5a)}\\
\text{s.t.} \quad & B^k \Delta w + b(w^k) + \Lambda\xi = 0, && \text{(5b)}\\
& C^k \Delta w + c(w^k) \geq 0, && \text{(5c)}
\end{aligned}
$$

where $A^k \in \mathbb{R}^{n_w} \times \mathbb{R}^{n_w}$ is a symmetric matrix representing the exact Hessian of the Lagrangian (3) or an approximation of it at the current iterate (w^k, λ^k, μ^k), $a^k = \nabla_w \phi(w^k)$ is the gradient of the cost function and B^k and C^k are the Jacobians of the constraints $b(\cdot)$ and $c(\cdot)$ at the current iterate w^k.

B. Predictor-Corrector Path-Following Methods

We will consider the parametric NLP (2) and for notational convenience and simplicity of exposition, assume that we do not have inequality constraints. The Karush-Kuhn-Tucker (KKT) [16] conditions of this problem can be written in compact form as

$$
F(z,\xi) = \hat{F}(z) + C\xi = 0, \qquad (6)
$$

where $z := [w^{\mathrm{T}},\ \lambda^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n_z}$ collects its primal-dual variables, $\hat{F} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z}$ collects the KKT conditions for the equality constrained NLP and $C \in \mathbb{R}^{n_z \times n_z}$ is an appropriate matrix that embeds the parameter ξ. We assume $F(z,\xi)$ for a given ξ to be at least once continuously differentiable. A solution for a given parameter ξ will be denoted as $\bar{z}(\xi)$. A full exact Newton step for this problem reads as [16]

$$
z^{k+1} = z^k - \left[\frac{\partial F}{\partial z}(z^k,\xi)\right]^{-1}(\hat{F}(z^k) + C\xi). \qquad (7)
$$

Following the presentation in [22], if the parameter ξ enters F linearly, which can be always achieved via an intermediate variable [8], one step of the path-following predictor-corrector method reads as

$$
\begin{aligned}
z^{k+1} =& z^k - \left[\frac{\partial F}{\partial z}(z^k,\xi^k)\right]^{-1}(\hat{F}(z^k) + C\xi^k)\\
&+ \left[\frac{\partial F}{\partial z}(z^k,\xi^k)\right]^{-1} C(\xi^{k+1} - \xi^k) \qquad\qquad (8)\\
=& z^k - \left[\frac{\partial F}{\partial z}(z^k,\xi^k)\right]^{-1}(\hat{F}(z^k) + C\xi^{k+1}),
\end{aligned}
$$

where $z^{k+1} \approx \bar{z}(\xi^{k+1})$ is now an approximate solution for the new parameter value ξ^{k+1}, given $z^k \approx \bar{z}(\xi^k)$. This corresponds to a standard Newton step with the new parameter ξ^{k+1}. Note that if we keep the parameter fixed, i.e. $\xi^{k+1} = \xi^k$, the last equation reduces to a standard full Newton step (7), often referred to as *corrector step*, and if $z^k = \bar{z}(\xi^k)$ holds, then equation (8) reduces to

$$
z^{k+1} = z^k - \left[\frac{\partial F}{\partial z}(z^k,\xi^k)\right]^{-1} C(\xi^{k+1} - \xi^k), \qquad (9)
$$

which is denoted as a *predictor step*, i.e. a first-order approximation of $\bar{z}(\xi^{k+1})$ where the sensitivities can be obtained via the implicit function theorem. Therefore, equation (8) has both predictive and corrective capabilities. In practice, instead of the exact Jacobian $J(z^k) := \frac{\partial F}{\partial z}(z^k,\xi^k)$ and its factorization, one often uses a Jacobian approximation $M_k \approx J(z^k)$ and thereby we get a Newton-type step

$$
z^{k+1} = z^k - M_k^{-1}(\hat{F}(z^k) + C\xi^{k+1}). \qquad (10)
$$

If we consider the inequality constraints in (2), the map $\bar{z}(\xi)$ is not smooth, but the solution manifold has smooth parts where the active set does not change and nondifferentiable points whenever the active set changes [14]. Furthermore, a *generalized tangential predictor* can be obtained by solving the QP (5). Such a predictor is a piecewise affine one, i.e. the QP can ”jump” over active set changes, for more details see [6], [22].

III. Algorithmic Ingredients

In this section we review the three algorithmic approaches that are the basis for the new algorithm presented in the subsequent section.

A. Standard Real Time Iteration

An algorithm based on the predictor-corrector method in the SQP framework is the Real Time Iteration (RTI) introduced in [8]. The RTI does not distinguish between OCPs with different parameters ξ and iterates while the problem changes. Only one full SQP iteration is done per sampling time, so that the algorithm never iterates to convergence for a fixed value of ξ, which ensures that it always works with the most recent state estimate and does not loose time by working on outdated information. Stability and convergence for a stable active set have been proven in [7], [10].

In general, the value of the current state ξ will not be equal to the optimization variable x_0, but, since (1b) is linear in x_0, the constraint is satisfied after the first full Newton step. The idea to linearly embed the initial value into the NLP is known as *Initial Value Embedding* (IVE) [6]. Moreover, in the RTI framework, each SQP iteration is divided into a long *preparation phase* and a short *feedback phase*. This reordering of the computations does not create any additional overhead per iteration. In those two phases the following calculations are performed.

Preparation Phase: Functions and derivatives are evaluated at the available linearization point $z^k = (w^k, \lambda^k, \mu^k)$. Sometimes the linearization point for a new iterate is adapted, e.g. with a shifting strategy [9]. Since the new measurement ξ enters the OCP linearly, the Hessian of the Lagrangian A^k, the gradient of the cost function a^k and the Jacobians of the constraints B^k and C^k do not depend on ξ, hence they can be evaluated before a new measurement is available.

Feedback Phase: When the current state of the system ξ is available, the possibly condensed QP (5) is solved and the new control input $u_0^{k+1} = u_0^k + \Delta u_0^k$ can be passed to the process. Thereby, the feedback delay is reduced to solving a single QP. A single RTI is summarized in Algorithm 1.

Algorithm 1: Single Real Time Iteration

```
Input: Last iterate $z^k$
Output: New iterate $z^{k+1}$
Preparation Phase:
    Postprocess and possibly adapt the last iterate by shifting
    Evaluate all functions and derivatives at $z^k$
    Possibly condense the QP (5)
Feedback Phase:
    Embed current state estimate $\xi$ into the QP (5)
    Solve QP, compute next iterate $z^{k+1}$ via (4) and send first control $u_0^{k+1}$ to the process
```

B. Multi-Level Iteration

In [4], an extension of the RTI, the Multi-Level Iteration has been introduced. The main idea is to update the matrices and vectors of the QP (5) at different time scales since different calculations have different computational loads. The modes or levels of the MLI can be run in parallel and the levels can exchange information in various ways, see e.g. [25]. The feedback rate is determined by the fastest level. We will briefly explain the version presented in [4], further extensions can be found in [17], [25]. For the lowest level, denoted as level A, we assume that a reference QP (5) with fixed $\widehat{A}, \widehat{a}, \widehat{B}, \widehat{b}, \widehat{C}$ and $\widehat{c}$ is given. The iterations start from a reference solution $(\widehat{w}, \widehat{\lambda}, \widehat{\mu})$, where the reference QP is provided by higher levels of the MLI. The goal of the lowest level of the MLI is to provide feedback as fast as possible and to take at least the active set changes into account. All matrices and vectors of the reference QP are held unchanged, only a new initial value ξ is embedded and the feedback $\widehat{u}_0 + \Delta u_0^k$ is sent to the process. In a predictor-corrector setting this level has just a predictor part and is equivalent to linear Model Predictive Control (MPC) [6]. In combination with higher levels of the MLI scheme, the reference QP changes and level A can be interpreted as an adaptive linear MPC [4], see also [13]. Level B changes the vectors of the QP (5) $\widehat{b}$, $\widehat{c}$, i.e. new function evaluations are performed, and $\widehat{a}$ is updated in an approximate fashion, see [4]. These iterations converge to a suboptimal, but feasible solution of the original NLP (2). Local stability properties of such an approximate policy are proven in [26] and [27]. Moreover, due to the fact that the linearization point is kept unchanged, an efficient algebraic elimination that can speed up integration, condensing and QP solution can be used [27]. Level C iterations are based on an adjoint SQP algorithm [4], where in contrast to level B, a^k is calculated as

$$a^k = \nabla_w \mathcal{L}(w^k, \lambda^k, \mu^k) + \widehat{B}^{\mathrm{T}} \lambda^k + \widehat{C}^{\mathrm{T}} \mu^k. \qquad (11)$$

Here, the Jacobians of the constraints do not need to be evaluated, $\nabla_w \mathcal{L}(w^k, \lambda^k, \mu^k)$ can be computed efficiently with the reverse mode of automatic differentiation with a cost approximately five times higher than for the evaluation of (3) [12]. Level C iterations can be shown to converge to optimal solutions of the original problem [4]. Finally, the level D iterations are the standard RTI. Various extension of level C and D iterations and other extensions to MLI are discussed in [17].

C. The Advanced Step Controller

When using the RTI or MLI approach, one might need to sacrifice optimality to achieve fast feedback. In order to avoid the possible convergence issues of a predictor-corrector algorithm performing just one iteration and to avoid suboptimal solutions, in the *Advanced Step Controller* [30] a more conservative approach is taken. For each new measurement ξ the NLP (2) is solved to convergence using an Interior-Point algorithm. Obviously, this is computationally more expensive, but yields an accurate locally optimal solution. In order to take the feedback delay into account, this online algorithm solves an *advanced* problem in the preparation phase with a predicted state $\tilde{\xi}$ as initial value. Furthermore, the solution is not applied directly to the process, but an additional linear system solve based on the last Newton iteration's matrix factorization is performed in the feedback phase to get a *tangential predictor* to correct for the mismatch between the predicted $\tilde{\xi}$ and actual measurement ξ. While such a tangential predictor can not "jump" over active set changes solving a linear system is often cheaper than solving a QP. Furthermore, the scheme relies on having an accurate model to predict the state at the next sampling time.

IV. The Advanced Step Real Time Iteration

One could also apply the same strategy to an RTI and get what we call an *Advanced Step Real Time Iteration* (AS-RTI). Thereby, an advanced problem would be solved in the preparation phase with an SQP method, and afterwards, for the delay compensation, an extra QP is solved to get a generalized tangential predictor to compensate the mismatch between the predicted and actual measurement. Obviously, this approach is computationally more expensive than the standard RTI, but it alleviates the possible convergence issues and suboptimality of solutions and delivers a predictor that can "jump" over active set changes. However, solving the advanced problem to convergence increases the computational burden of the preparation phase significantly.

Instead of doing so, we propose to use some of the levels of the MLI to iterate on the advanced problem to get an improved guess $\tilde{z}^{k+1}$. The aim of the new algorithm is to combine the benefits of both paradigms explained above. The MLI scheme contains all the ingredients one needs to refine a solution while still keeping the computational burden low.

The limiting cases would be: (a) doing just one level A iteration in the preparation phase, and (b) the full convergent SQP as explained above. In between there is a wide family of possible algorithms as we can now assemble the preparation phase in a different fashion. This approach is summarized in Algorithm 2.

In practice, we will usually extend the preparation phase with a few adjoint SQP iterations, since they are computationally cheap, they yield optimality improvement and no new matrix factorizations are needed while solving the QP [4]. It is reasonable to assume that, if $\tilde{\xi}^{k+1}$ is close to ξ^{k+1}, we might have fewer active set changes between the corresponding parametric NLPs. Hence, with AS-RTI we determine the active set of the advanced problem, so QP solvers that can be warm-started as qpOASES [11] or first order methods as OSQP [2] will have a better initial guess and the feedback delay is reduced.

Algorithm 2: Single Advanced Step Real Time Iteration

```
Algorithm 2a: Preparation Phase
Input: $z^k$, $\xi^k$, $u_0^k$, all data for the QP at iteration $k$
Output: New iterate guess $\tilde{z}^{k+1}$
1 Predict $\tilde{\xi}^{k+1}$ with $\tilde{\xi}^{k+1} = f(\xi^k, u_0^k)$.
2 Predict (and possibly refine/correct) optimal
  solution $\tilde{z}^{k+1}$ for predicted state $\tilde{\xi}^{k+1}$ by iterating
  with some mode of the MLI on an advanced
  problem corresponding to $\tilde{\xi}^{k+1}$
3 Evaluate all functions and derivatives at $\tilde{z}^{k+1}$
  needed for the QP (5)
4 Possibly condense the QP
Feedback Phase:
Algorithm 2b: Feedback Phase
Input: Solution guess $\tilde{z}^{k+1}$, $\xi^{k+1}$
Output: New iterate $z^{k+1}$, $u_0^{k+1}$, all data for the QP
        solved at iteration $k+1$
5 Embed current state estimate $\xi^{k+1}$ into the QP (5)
6 Solve QP, compute next iterate $z^{k+1}$ via (4) and
  send first control $u_0^{k+1}$ to the process
```

In the following, we will briefly analyze the simplest case of Algorithm 2, namely performing just another QP solve (level A iteration) with respect to the standard RTI. With the next lemma we investigate this simplest variant of the AS-RTI and show that, for a perfect prediction of the parameter, the RTI and this AS-RTI variant have the same linearization points, but still the AS-RTI achieves better tracking of the solution manifold. Of course, a perfect prediction will never be available in practice, but we want to show that the reordering of calculations brings us closer to the solution manifold. To simplify the analysis, we assume a stable active set.

Lemma 1: Assuming a single level A iteration at line 2 of Algorithm 2a and a perfect prediction of the parameter ξ, Algorithms 1 and 2 have the same linearization points.

Proof. A perfect prediction means $\tilde{\xi}^k = \xi^k$, for all $k \geq 0$. Let $\tilde{z}^k$ be the linearization point at iteration k, then the output of the feedback phase of the AS-RTI obtained via a predictor-corrector iteration (8) reads as

$$\begin{aligned} z^k =& \tilde{z}^k - \Big[\frac{\partial F}{\partial z}(\tilde{z}^k, \xi^k)\Big]^{-1}(\hat{F}(\tilde{z}^k) + C\xi^k) \\ &- \Big[\frac{\partial F}{\partial z}(\tilde{z}^k, \xi^k)\Big]^{-1} C(\tilde{\xi}^k - \xi^k) \\ =& \tilde{z}^k - \Big[\frac{\partial F}{\partial z}(\tilde{z}^k, \xi^k)\Big]^{-1}(\hat{F}(\tilde{z}^k) + C\xi^k), \end{aligned} \tag{12}$$

i.e. we have just the corrector step since the prediction is perfect. At the next iteration $(k+1)$, in the preparation phase of Algorithm 2 we have the linearization point prediction (line 2) using (9)

$$\begin{aligned} \tilde{z}^{k+1} &= z^k - \Big[\frac{\partial F}{\partial z}(\tilde{z}^k, \xi^k)\Big]^{-1} C(\tilde{\xi}^{k+1} - \xi^k) \\ &\overset{(12)}{=} \tilde{z}^k - \Big[\frac{\partial F}{\partial z}(\tilde{z}^k, \xi^k)\Big]^{-1}(\hat{F}(\tilde{z}^k) + C(\xi^k + \tilde{\xi}^{k+1} - \xi^k)) \\ &= \tilde{z}^k - \Big[\frac{\partial F}{\partial z}(\tilde{z}^k, \xi^k)\Big]^{-1}(\hat{F}(\tilde{z}^k) + C\xi^{k+1}), \end{aligned}$$

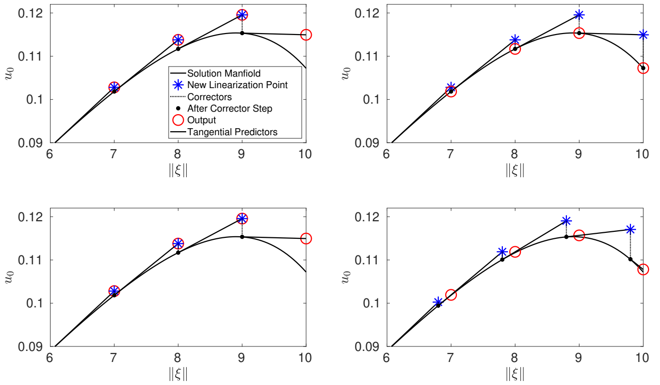

Fig. 1. Illustration of the results of Lemma 1. On the top left one can see the iterates, linearization points, outputs and tangential predictors of the RTI scheme, and on the top right of the AS-RTI with an extra QP solve in the preparation phase. One can see that, in the case of a perfect prediction, both schemes "visit" the same linearization points, but have different outputs for the corresponding parameters ξ. Furthermore, the AS-RTI is closer to the solution manifold. Note that in this case we have only a prediction of the optimal solution $\tilde{z}^{k+1}$ at line 2 of Algorithm 2a, since just one QP solve is performed. Bottom right shows the AS-RTI without a perfect prediction. In this case the the RTI scheme (bottom left) does not "visit" the same linearization points as the AS-RTI (bottom right). One can see that the new linearization point is on the same tangent as the output of the previous iteration.

which is a standard RTI. Using induction, this holds for for all $k \geq 0$. ■

The "advanced step" nature of the algorithm is also illustrated in Figure 1. Note that in this variant of the AS-RTI two QPs are solved per sampling time.

V. Contraction Estimate

In this section we want to estimate the contraction between two subsequent iterations of a more general AS-RTI. First, we restate some standard assumptions and a theorem from [28], [22], without proof, which we will use to derive the contraction estimate for the AS-RTI.

Assumption 1: (ω and κ conditions) There exist $\omega < \infty$ and $\kappa < 1$ such that, for any given iterate z^k and iteration matrix M_k, the following holds:

1) $\|M_k^{-1}(J(z^k) - J(z))\| \leq \omega\|z^k - z\|, \ \forall z,$
2) $\|M_k^{-1}(J(z^k) - M_k)\| \leq \kappa.$

Let κ and ω be the infimum of all possible constants for which the above inequalities hold.

Remark 1: Note that, if we use exact Jacobians ($M_k = J(z^k)$), then $\kappa = 0$.

Assumption 2: (Lipschitz continuity) There exists a constant σ such that, for every solution $\bar{z}^k$ and $\bar{z}^{k+1}$ associated with the parameters ξ^k and ξ^{k+1}, respectively, the following holds:

$$\|\bar{z}^{k+1} - \bar{z}^k\| \leq \sigma\|\xi^{k+1} - \xi^k\|.$$

Theorem 2: [28] Regard a nonlinear differentiable function $F : \mathbb{R}^n \to \mathbb{R}^n$. Let Assumptions 1 and 2 hold. Then, for the sequence of iterates $(z^k)_{k\geq 0}$ generated via the Newton-type predictor-corrector update (10), the following inequality holds:

$$\|e^{k+1}\| \leq \Big(\kappa + \omega\sigma\|\xi^{k+1} - \xi^k\| + \frac{\omega}{2}\|e^k\|\Big)\|e^k\| + \Big(\kappa\sigma + \frac{\omega\sigma^2}{2}\|\xi^{k+1} - \xi^k\|\Big)\|\xi^{k+1} - \xi^k\|, \tag{13}$$

where $e^{k+1} := z^{k+1} - \bar{z}^{k+1}$ and $e^k := z^k - \bar{z}^k$.

Assumption 3: (Initialization) For all algorithms in this paper we will assume the following: at an initial point z^0 and a solution $\bar{z}^0$ the following inequality holds

$$\|z^0 - \bar{z}^0\| \leq r_z < \bar{r}_z := 2(1-\kappa)/\omega. \tag{14}$$

We define the following constants: $c_1 := \omega\sigma$, $c_2 := \frac{\omega}{2}$, $c_3 := \kappa\sigma$, $c_4 := \frac{\omega\sigma^2}{2}$.

Corollary 3: Let Assumptions 1, 2 and 3 hold. Then, for any $k \geq 0$, for the iterates generated via (10), the following holds:

$$\|z^{k+1} - \bar{z}^{k+1}\| \leq r_z \tag{15}$$

provided that $\|\xi^{k+1} - \xi^k\| \leq r_\xi$, where

$$r_\xi := \begin{cases} \frac{\sqrt{(c_3 + r_z c_1)^2 + 4c_4(1-\kappa-c_2 r_z)r_z} - (c_3 + r_z c_1)}{2c_4} & \text{if } c_4 > 0, \\ \frac{(1-\kappa-c_2 r_z)r_z}{c_3 + r_z c_1} & \text{if } c_4 = 0. \end{cases} \tag{16}$$

Proof. The proof is omitted here for brevity, but follows similar lines as the proof of Corollary 3.6 in [22]. ■

From now on, the second index denotes what we call an *inner iteration*, i.e. the Newton-type iterations (10) where we keep the parameter ξ^k fixed (e.g. at line 2 of Algorithm 2a if we refine the predicted solution). For instance, a Newton-type step reads as $z^{k,j+1} = z^{k,j} - M_{k,j}^{-1}(\hat{F}(z^{k,j}) + C\xi^k)$. For notational convenience let $e^{k+1,j} := z^{k+1,j} - \bar{z}^{k+1}$. Fixing the parameter ξ, the well-known Newton-type contraction estimate proven e.g. in [3] holds.

Corollary 4: Let Assumptions 1 and 3 hold and let the parameter ξ be fixed. Then, the Newton-type iterations (10) converge to $\bar{z}^k$ and the following contraction estimate holds:

$$\|e^{k,j+1}\| \leq \Big(\kappa + \frac{\omega}{2}\|e^{k,j}\|\Big)\|e^{k,j}\|. \tag{17}$$

Proof. Fixing ξ^k in (10), (17) follows from (13). ■

Denote $\alpha_{k,j} = \kappa + \frac{\omega}{2}\|e^{k,j}\|$ ($\alpha_{k,0} < 1$ due to (14)), where $j \geq 0$ is the iteration index for a fixed parameter. Similarly, we define $\tilde{e}^{k,j} := \tilde{z}^{k,j} - \bar{z}(\tilde{\xi}^k)$ and $\tilde{\alpha}_{k,j} = \kappa + \frac{\omega}{2}\|\tilde{e}^{k,j}\|$.

Lemma 5: Let Assumptions 1 and 3 hold. For the Newton-type iterations (10) with a fixed parameter ξ^k, the following inequality holds:

$$\|e^{k,j}\| \leq (\alpha_{k,0})^j\|e^{k,0}\|, \tag{18}$$

where $j \geq 0$ is the number of iterations.

Proof. Starting from (17), at the first iteration, we have that

$$\alpha_{k,1} = \kappa + \frac{\omega}{2}\|e^{k,1}\| \overset{(17)}{\leq} \kappa + \frac{\omega}{2}\underbrace{\alpha_{k,0}}_{<1}\|e^{k,0}\| < \underbrace{\kappa + \frac{\omega}{2}\|e^{k,0}\|}_{=\alpha_{k,0}}.$$

Applying this recursively for $j \geq 1$ and using inequality (17) we obtain inequality (18). ■

Assumption 4: (Predicted parameter) In all iterations of Algorithm 2 we make a parameter prediction $\tilde{\xi}^{k+1}$ such that, the following holds

$$\|\xi^{k+1} - \tilde{\xi}^{k+1}\| \leq r_{\xi}. \tag{19}$$

The assumption above ensures that we make a reasonably good parameter prediction, such that the result from Corollary 3 can be used. The following theorem gives the general contraction estimate for two subsequent iterations of Algorithm 2 with Newton-type iterations at line 2 of Algorithm 2a.

Theorem 6: Let Assumptions 1, 2, 3 and 4 hold. Furthermore, assume that we make $j \geq 0$ inner Newton-type iterations for a parameter prediction $\tilde{\xi}^{k+1}$ in the preparation phase of AS-RTI. Then, for the sequence of iterates $(z^k)_{k \geq 0}$, the following holds:

$$\begin{aligned} \|e^{k+1}\| \leq (\tilde{\alpha}_{k+1,0})^j \left[\beta_k \|e^k\| + \gamma_k \|\tilde{\xi}^{k+1} - \xi^k\|\right] \\ + \delta_k \|\xi^{k+1} - \tilde{\xi}^{k+1}\|, \end{aligned} \tag{20}$$

where we have defined the positive constants $\hat{\beta}_k$, β_k, γ_k, δ_k, respectively, as

$$\hat{\beta}_k := \kappa + c_1 \|\xi^{k+1} - \tilde{\xi}^{k+1}\| + c_2 (\tilde{\alpha}_{k+1,0})^j \|\tilde{e}^{k+1,0}\|, \tag{21}$$

$$\beta_k := \hat{\beta}_k \big(\kappa + c_1 \|\tilde{\xi}^{k+1} - \xi^k\| + c_2 \|e^k\|\big), \tag{22}$$

$$\gamma_k := \hat{\beta}_k \big(c_3 + c_4 \|\tilde{\xi}^{k+1} - \xi^k\|\big), \tag{23}$$

and

$$\delta_k := c_3 + c_4 \|\xi^{k+1} - \tilde{\xi}^{k+1}\|. \tag{24}$$

Proof. Under the assumptions of the theorem, the contraction from Theorem 2 holds between any two iterates. In Algorithm 2, we start from z^k and, from the first predictor-corrector iteration and for the prediction $\tilde{\xi}^{k+1}$, we have, due to (13):

$$\begin{aligned} \|\tilde{e}^{k+1,0}\| \leq \big(\kappa + c_1 \|\tilde{\xi}^{k+1} - \xi^k\| + c_2 \|e^k\|\big) \|e^k\| \\ + \big(c_3 + c_4 \|\tilde{\xi}^{k+1} - \xi^k\|\big) \|\tilde{\xi}^{k+1} - \xi^k\|, \end{aligned} \tag{25}$$

Now, due to (18) in Lemma 5, for all inner iterations, i.e. for $j \geq 0$, we have the following relation:

$$\|\tilde{e}^{k+1,j}\| \leq (\tilde{\alpha}_{k+1,0})^j \|\tilde{e}^{k+1,0}\|, \tag{26}$$

and, due to our assumption that (19) holds and Corollary 3, we have $\tilde{\alpha}_{k,j} < 1, \forall j \geq 0, \forall k \geq 0$. In the feedback phase of Algorithm 2, we have one more predictor-corrector step, now for the true parameter ξ^{k+1}. Hence, due to (13), the following holds:

$$\begin{aligned} \|e^{k+1}\| \leq \big(\kappa + c_1 \|\xi^{k+1} - \tilde{\xi}^{k+1}\| + c_2 \|\tilde{e}^{k+1,j}\|\big) \|\tilde{e}^{k+1,j}\| \\ + \underbrace{\big(c_3 + c_4 \|\xi^{k+1} - \tilde{\xi}^{k+1}\|\big)}_{\overset{(24)}{=} \delta_k} \|\xi^{k+1} - \tilde{\xi}^{k+1}\|. \end{aligned}$$

Plugging the estimate for $\|\tilde{e}^{k+1,j}\|$ from (26) into the last equation one gets

$$\begin{aligned} \|e^{k+1}\| \leq \underbrace{\big(\kappa + c_1 \|\xi^{k+1} - \tilde{\xi}^{k+1}\| + c_2 (\tilde{\alpha}_{k+1,0})^j \|\tilde{e}^{k+1,0}\|\big)}_{\overset{(21)}{=} \hat{\beta}_k} \\ \cdot (\tilde{\alpha}_{k+1,0})^j \|\tilde{e}^{k+1,0}\| + \delta_k \|\xi^{k+1} - \tilde{\xi}^{k+1}\|. \end{aligned}$$

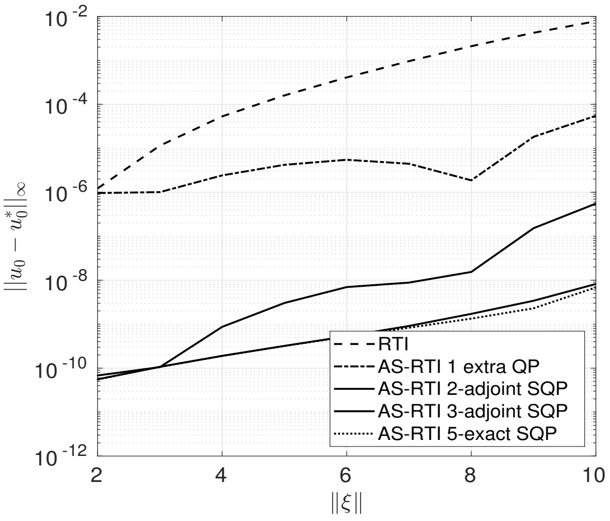

Fig. 2. Comparison of errors of the RTI and AS-RTI on a simple parametric problem for a simple pendulum model, see Exercise 8.10 in [21].

Now, if we replace $\|\tilde{e}^{k+1,0}\|$ with its estimate (25), we obtain

$$\begin{aligned} \|e^{k+1}\| \leq (\tilde{\alpha}_{k+1,0})^j \hat{\beta}_k \cdot \big[\big(\kappa + c_1 \|\tilde{\xi}^{k+1} - \xi^k\| + c_2 \|e^k\|\big) \\ \cdot \|e^k\| + \big(c_3 + c_4 \|\tilde{\xi}^{k+1} - \xi^k\|\big) \|\tilde{\xi}^{k+1} - \xi^k\|\big] \\ + \delta_k \|\xi^{k+1} - \tilde{\xi}^{k+1}\|. \end{aligned} \tag{27}$$

Due to the definitions of β_k in (22) and γ_k in (23) the contraction estimate (20) follows from (27). ■

We see from the results of the last theorem that the distance of the new iterate z^{k+1} to the solution manifold depends on: (a) the quality of our parameter prediction, (b) how close we were to the solution in the previous iterate, (c) the number of inner iterations in the preparation phase and (d) the distance between two subsequent parameters. Furthermore, the quality of the derivatives has an effect on the inner iterations since with smaller κ the term $\alpha_{k,0}^j$ will shrink faster. For instance, with more inner iterations the first two terms in (20) will become very small, and depending on the quality of our parameter prediction and Jacobian approximation, the third term might be small as well, which means that we can track the optimal solution manifold closely. In the limiting case with perfect predictions and $j \to \infty$, we have ideal NMPC. This is also illustrated in Figure 2. The figure shows the errors of different AS-RTI schemes applied to a simple parametric example from [21].

VI. Numerical Experiments

As a benchmark, we consider the swing-up of an inverted pendulum, considered e.g. in [28], which is described by the differential equations

$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ \frac{-ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + F}{M + m - m(\cos(\theta))^2} \\ \omega \\ \frac{-ml\cos(\theta)\sin(\theta)\omega^2 + F\cos(\theta) + (M+m)g\sin(\theta)}{l(M + m - m(\cos(\theta))^2)} \end{bmatrix}.$$

We consider an OCP of the form (1). The discretized dynamics $f(\cdot)$ is obtained by applying multiple shooting and the explicit Runge-Kutta scheme of order four with fixed stepsize and one integration step. A prediction horizon $T = 2$ [s] is used and the trajectories are discretized using $N = 60$ multiple shooting intervals. Simple bounds are imposed on the control input:

$$-10 \ \le u_i \le 10, \ \ i = 0, \dots, N-1.$$

The cost function for $i = 0, \dots, N-1$ reads as

$$l_i(x_i, u_i) = \begin{bmatrix} x_i - x_{\text{ref}} \\ u_i - u_{\text{ref}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x_i - x_{\text{ref}} \\ u_i - u_{\text{ref}} \end{bmatrix}$$

and the terminal cost as

$$l_N(x_N) = (x_N - x_{\text{ref}})^{\mathrm{T}} Q (x_N - x_{\text{ref}}),$$

with $Q = \operatorname{diag}([1, \ 0.01, \ 0.002, \ 0.002])$ and $R = 0.001$. Finally, the following time-varying reference is used:

$$x_{\text{ref}}(t) = \begin{cases} [0, \ \pi, \ 0, \ 0, \ 0]^{\mathrm{T}}, \ t < 1s, \\ [0, \ 0, \ 0, \ 0, \ 0]^{\mathrm{T}}, \ t \ge 1s. \end{cases}$$

We compare our algorithm with ideal NMPC which we obtain via a fully converged solution at every sampling instant using IPOPT [24] and its interface in CasADi [1] in MATLAB. The RTI and AS-RTI are implemented in MATLAB using the ACADO standalone integrators [20] and qpOASES [11] using its `acados` interface [23]. A Gauss-Newton Hessian approximation is used.

Furthermore, we apply a disturbance to the system between time $t_1 = 6.0$ [s] and $t_2 = 6.2$ [s] by replacing the optimal input u_0 with the perturbed input $u_0 + 8$. Figure 3 depicts the RTI and AS-RTI compared to ideal NMPC. In this scenario, with the AS-RTI, we make in the preparation phase 3 adjoint SQP (level C) inner iterations. Comparing the figures one can see that with little additional computational effort in the preparation phase of the AS-RTI we get closer to ideal NMPC, which confirms the results of Theorem 6 and good performance of our algorithm.

VII. Conclusions and Outlook

In this paper, we have presented a new family of algorithms for NMPC combining the ideas of the Real-Time Iteration [8], Multi-Level Iteration [4] and the Advanced Step Controller [30]. We proved in Theorem 6 contraction of the new algorithms under standard assumptions. A nontrivial numerical example confirms our theoretical results and it shows that, with few and cheap additional iterations, we can get significantly closer to ideal NMPC behavior.

We proved our results under the assumption that the active set does not change. However, for a more general analysis, it would be reasonable to extend the results using the techniques from [22], [29] and generalized equations to generalize the results of Theorem 6 in the presence of active set changes. Next, it would be interesting to investigate theoretically and numerically the influence of suboptimality iterations, such as level B, which improve feasibility, but not optimality, on the overall performance of Algorithm 2.

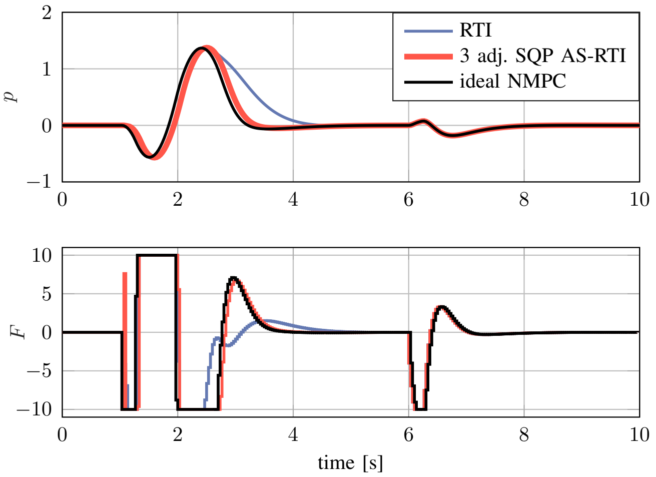

Fig. 3. Comparison of the closed-loop trajectories of the inverted pendulum example for controls generated with the RTI, AS-RTI and ideal NMPC.

References

[1] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

[2] G. Banjac, B. Stellato, N. Moehle, P. Goulart, A. Bemporad, and S. Boyd. Embedded code generation using the OSQP solver. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2017.

[3] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften*. Universität Bonn, Bonn, 1987.

[4] 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*, pages 3–22. SIAM, 2007.

[5] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the IFAC World Congress*, pages 242–247. Pergamon Press, 1984.

[6] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, University of Heidelberg, 2001.

[7] M. Diehl, H. G. Bock, and J. P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

[8] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.

[9] M. Diehl, H. J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In L. Magni, M. D. Raimondo, and F. Allgöwer, editors, *Nonlinear model predictive control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 391–417. Springer, 2009.

[10] M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, and J. P. Schlöder. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Proc.-Control Theory Appl.*, 152(3):296–308, 2005.

[11] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[12] A. Griewank and A. Walther. *Evaluating Derivatives*. SIAM, 2nd edition, 2008.

[13] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl. From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*, 2016.

[14] J. Guddat, F. Guerra Vasquez, and H.T. Jongen. *Parametric Optimization: Singularities, Pathfollowing and Jumps*. Teubner, Stuttgart, 1990.

[15] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl. Recent advances in quadratic programming algorithms for nonlinear model predictive control. *Vietnam Journal of Mathematics*, 46(4):863–882, 2018.

[16] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2006.

[17] A. Nurkanović, S. Albrecht, and M. Diehl. Multi Level Iterations for Economic Nonlinear Model Predictive Control. In T. Faulwasser, M. A. Müller, and K. Worthmann, editors, -. Springer, 2019.

[18] T. Ohtsuka. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.

[19] R. Quirynen. *Numerical Simulation Methods for Embedded Optimization*. PhD thesis, KU Leuven and University of Freiburg, 2017.

[20] R. Quirynen, S. Gros, and M. Diehl. Fast auto generated ACADO integrators and application to MHE with multi-rate measurements. In *Proceedings of the European Control Conference (ECC)*, pages 3077–3082, 2013.

[21] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill, 2nd edition edition, 2017.

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

[23] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl. Towards a modular software package for embedded optimization. In *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, 2018.

[24] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[25] L. Wirsching. *Multi-Level Iteration Schemes with Adaptive Level Choice for Nonlinear Model Predictive Control*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2018.

[26] A. Zanelli, R. Quirynen, and M. Diehl. An efficient inexact NMPC scheme with stability and feasibility guarantees. In *Proceedings of 10th IFAC Symposium on Nonlinear Control Systems*, 2016.

[27] A. Zanelli, R. Quirynen, and M. Diehl. Efficient zero-order NMPC with feasibility and stability guarantees. In *Proceedings of the European Control Conference (ECC)*, 2019.

[28] A. Zanelli, R. Quirynen, J. Jerez, and M. Diehl. A homotopy-based nonlinear interior-point method for NMPC. In *Proceedings of the IFAC World Congress*, 2017.

[29] A. Zanelli, Q. Tran-Dinh, and M. Diehl. Contraction estimates for abstract real-time algorithms for nmpc. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Nice, France, Dec 2019. (accepted).

[30] V. M. Zavala and L. T. Biegler. The advanced step NMPC controller: Optimality, stability and robustness. *Automatica*, 45:86–93, 2009.