Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Contraction Properties of the Advanced Step Real-Time Iteration for NMPC ⋆

Armin Nurkanović [*,**] **Andrea Zanelli** [**] **Sebastian Albrecht** [*]
Gianluca Frison [**] **Moritz Diehl** [**]

[*] *Siemens Corporate Technology, Munich, Germany (e-mail:armin.nurkanovic,sebastian.albrecht@siemens.com)*
[**] *University of Freiburg, Freiburg, Germany (e-mail:andrea.zanelli, gianluca.frison,moritz.diehl@imtek.uni-freiburg.de)*

Abstract: In this paper we study the contraction properties of the recently introduced Advanced Step Real-Time Iteration (AS-RTI) under active-set changes. Compared to the well-known Real-Time Iteration, in order to improve optimality, in the AS-RTI some extra calculations on a problem with a predicted state are carried out. This enables us to trade off controller performance for computational load in a simple manner. Under standard assumptions, we prove the contraction of the iterates and boundedness of the numerical error. To achieve these goals we use generalized equations, Robinson's strong regularity and recently presented results for abstract real-time algorithms. Finally, we present a numerical benchmark, where the performance of different variants of the AS-RTI is demonstrated on an industrial case study.

Keywords: nonlinear model predictive control, numerical methods, nonlinear optimization

1. INTRODUCTION

Nonlinear Model Predictive Control (NMPC) is an advanced optimization-based control strategy which is becoming increasingly widespread in industry and academia. One of the main advantages of NMPC over classical control strategies is that nonlinearities and constraints can be handled directly. In NMPC one has to repeatedly (approximately) solve in real-time a discrete-time parametric Optimal Control Problem (OCP):

$$
\begin{aligned}
P_{\text{OCP}}(x): \quad & \min_{w} \quad \sum_{i=0}^{N-1} \ell_i(s_i, u_i) + \ell_N(s_N) \\
& \text{s.t.} \quad s_0 - x = 0, \\
& \qquad s_{i+1} - \psi(s_i, u_i) = 0, \qquad \forall i \in \mathcal{I}, \\
& \qquad \pi(s_i, u_i) \leq 0, \qquad \forall i \in \mathcal{I}, \\
& \qquad \pi_N(s_N) \leq 0.
\end{aligned} \tag{1}
$$

Here, $N \in \mathbb{N}$ is the horizon length and we define the index set $\mathcal{I} \coloneqq \{0, \dots, N-1\}$. We define the vector $w \coloneqq (w_0, w_1, \dots w_N)$ with $w_i \coloneqq (s_i, u_i)$, $\forall i \in \mathcal{I}$ and $w_N \coloneqq s_N$, with $s_i \in \mathbb{R}^{n_s}$, $u_i \in \mathbb{R}^{n_u}$ being the predicted states and inputs of the controlled system. The objective terms $\ell_i : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and $\ell_N : \mathbb{R}^{n_s} \to \mathbb{R}$ are the stage and terminal costs, respectively. The function $\psi : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_s}$ describes the discrete-time system dynamics. The functions $\pi : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$ and $\pi_N : \mathbb{R}^{n_s} \to \mathbb{R}^{n_r}$ define the stage and terminal constraints, respectively and the parameter x represents the initial state of the system. We assume that all functions are twice continuously differentiable.

In NMPC, at every sampling instant a new state estimate x^k corresponding to time t^k is received. Then, after (approximately) solving the OCP, the first control input $\bar{u}_0(x^k)$ (the optimal solution is an implicit function of the parameter value) is passed to the system and held constant for the sampling time T_s. We denote the optimal solution of the parametric OCP as $\bar{w}(x)$. In general, it is not possible to solve the OCP (1) immediately. In order to reduce the computation times many NMPC algorithms seek approximate solutions. The aim of these algorithms is to closely track $\bar{w}(x)$ with a high sampling rate in real-time (see e.g. (Diehl et al., 2009)). Hence, the goal is to develop discrete-time schemes which provide computationally cheap approximations of $\bar{w}(x^k)$ (Zavala and Anitescu, 2010; Tran-Dinh et al., 2012; Zanelli et al., 2019). The possibility to track $\bar{w}(x)$ with a lower sampling time enables one to use larger and more detailed models, longer prediction horizons and include more decision variables, which is desirable in many industrial applications.

The key for success of real-time NMPC algorithms is that the OCP does not have to be solved to convergence, but to keep the numerical error bounded over time (Zavala and Anitescu, 2010; Zanelli et al., 2019). Examples of real-time NMPC algorithms are the Advanced Step Controller (ASC) (Zavala and Biegler, 2009) and the C/GMRES algorithm (Ohtsuka, 2004). Among others, Sequential Quadratic Programming (SQP) based algorithms are the Real-Time Iteration (RTI) scheme (Diehl, 2001), the Multi-Level Iteration (MLI) (Bock et al., 2007) and the recently propose Advanced Step Real-Time Iteration (AS-RTI) (Nurkanović et al., 2019b). Many other variants of the RTI and MLI schemes can be found in the literature,

⋆ 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.

for an overview see e.g. (Nurkanović et al., 2019a). An augmented Lagrangian tracking scheme is presented in (Zavala and Anitescu, 2010) and general predictor-corrector algorithms for sampled-data NMPC are described in e.g. (Anitescu and Zavala, 2017). Error estimates for continuation methods for generalized equations are studied in (Dontchev et al., 2013) and general Q-linearly convergent real-time algorithms are studied in (Zanelli et al., 2019).

1.1 Contributions and Outline

In this paper we provide a more general theoretical analysis for the recently introduced AS-RTI scheme (Nurkanović et al., 2019b). Moreover, we confirm our theoretical findings on a numerical benchmark. The main contributions of this paper are: (1) we prove contraction of the iterates of the AS-RTI under active-set changes if the computations are carried out with Q-linearly convergent algorithms, (2) we provide sufficient conditions for boundedness of the numerical error for the AS-RTI, (3) we present a numerical benchmark in which the AS-RTI is used to control a diesel generator.

In Section 2 we describe the AS-RTI scheme. This is followed by a brief presentation of Robinson's strong regularity and contraction results for general Q-linearly convergent real-time algorithms in Section 3. Using these tools we prove our main theoretical results in Section 4. In Section 5 we provide an industrial use-case: controlling the frequency of a diesel generator and the voltage at the load bus. There we show the benefits of few additional computations in the AS-RTI compared to the RTI.

2. THE ADVANCED STEP REAL-TIME ITERATION

This section describes the AS-RTI scheme introduced in (Nurkanović et al., 2019b). The AS-RTI combines ideas from the RTI, MLI and ASC. We highlight the benefits and drawbacks of these schemes and how the AS-RTI mitigates some of the issues.

2.1 Sequential Quadratic Programming

We consider the following parametric Nonlinear Program (NLP):

$$P(x): \quad \begin{aligned} \min_{w} \quad & f(w) \\ \text{s.t.} \quad & g(w) + \hat{M}x = 0, \\ & w \in \Omega, \end{aligned} \tag{2}$$

with $w \in \mathbb{R}^n$ and where $\Omega \subseteq \mathbb{R}^n$ is a nonempty, closed and convex set. The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^{n_g}$ are twice continuously differentiable functions. The matrix $\hat{M} = [-I, 0, \dots]^\top$ embeds linearly the parameter $x \in X$, where X is the set of all possible parameter values. The set Ω is usually described by a set of inequalities, i.e. $\Omega = \{w \mid h(w) \leq 0\}$, with $h : \mathbb{R}^{n_w} \to \mathbb{R}^{n_h}$ being also a twice continuously differentiable function. Convexity of the set Ω in (2) can always be achieved by introducing slack variables. Problem $P_{\text{OCP}}(x)$ in (1) can be cast in this form by properly defining the functions $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$.

For a fixed x, the problem $P(x)$ in (2), can be solved to local optimality with standard NLP algorithms, see (Nocedal and Wright, 2006). Let $\lambda \in \mathbb{R}^{n_g}$ and $\mu \in \mathbb{R}^{n_h}$ be the Lagrange multipliers corresponding to the equality and inequality constraints, respectively. In SQP, we assume to start with a primal-dual guess $z^0 \coloneqq (w^0, \lambda^0, \mu^0)$ close enough to the solution and a full SQP step is computed as:

$$w^{k+1} = w^k + \Delta w^k, \;\; \lambda^{k+1} = \lambda^k_{\text{QP}}, \;\; \mu^{k+1} = \mu^k_{\text{QP}}. \tag{3}$$

Here, $(\Delta w^k, \lambda^k_{\text{QP}}, \mu^k_{\text{QP}})$ is the primal-dual solution of the Quadratic Program (QP):

$$\min_{\Delta w} \quad \frac{1}{2}\Delta w^\top A^k \Delta w + a^{k\top}\Delta w \tag{4a}$$

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

$$H^k \Delta w + h(w^k) \leq 0, \tag{4c}$$

where $A^k \in \mathbb{R}^{n_w} \times \mathbb{R}^{n_w}$ is a symmetric positive definite matrix being the exact Hessian of the Lagrangian (5) or an approximation of it at the current iterate $z^k = (w^k, \lambda^k, \mu^k)$. The Lagrangian of the NLP (2), with an explicit definition of the set Ω, reads as

$$\mathcal{L}(w, \lambda, \mu) = f(w) + \lambda^\top g(w) + \lambda^\top \hat{M}x + \mu^\top h(w). \tag{5}$$

The vector $a^k = \nabla_w f(w^k)$ is the gradient of the cost function and G^k and H^k are the Jacobians of the constraints $g(\cdot)$ and $h(\cdot)$ at the current iterate w^k, respectively.

2.2 The Advanced Step Real-Time Iteration

The RTI and MLI schemes perform a single (inexact) SQP iteration per sampling time which might lead to convergence issues and larger numerical errors. The ASC solves an *advanced problem* $P_{\text{OCP}}(\tilde{x}^{k+1})$ to local optimality, with a predicted state $\tilde{x}^{k+1}$ at the time we need the next feedback. This might be computationally too expensive for a given sampling rate. Since in practice the sampling time is often fixed due to sensor hardware limits, one can perform a few more computations than a single (inexact) SQP step as in the RTI or MLI, but still not solve the problem to convergence as the ASC. The recently proposed AS-RTI (Nurkanović et al., 2019b) bridges this gap and combines the benefits of the RTI, MLI and ASC and enables one to trade off computational load for controller performance in a flexible way. It was successfully used in simulation experiments for wind turbine control (Nurkanović et al., 2019a) and microgrid operation (Nurkanović et al., 2020).

In general, feedback delay can severely degrade the controller's performance. Therefore, in the AS-RTI as in many other real-time NMPC algorithms, the computations are divided into a *preparation phase* where all computations can be performed without the knowledge of x^{k+1} and a *feedback phase*, after the parameter value x^{k+1} is available (Diehl et al., 2009). In the AS-RTI few additional computations on an *advanced problem* $P_{\text{OCP}}(\tilde{x}^{k+1})$ are carried out in the preparation phase to overcome the possible convergence issues of performing a single (inexact) SQP iteration per sampling time. Depending on the available time, one can perform extra calculations ranging from a single QP (level A of the MLI) or a few iterations of some level of the MLI up to a fully converged SQP solution in the limit. For more details about the different levels of the MLI see (Bock et al., 2007; Nurkanović et al., 2019a). In the feedback phase, just as in the RTI and MLI schemes,

Algorithm 1: The Advanced Step Real-Time Iteration Scheme

Algorithm 1a: Preparation Phase

Input: z^k, x^k, u_0^k, all data for the QP at iteration k
Output: New iterate guess $\tilde{z}^{k+1}$

1 Get prediction $\tilde{x}^{k+1}$, e.g. with $\tilde{x}^{k+1} = \psi(x^k, u_0^k)$.
2 Get optimal solution prediction $\tilde{z}^{k+1}$ by iterating with some mode of the MLI on the *advanced problem* $P_{\text{OCP}}(\tilde{x}^{k+1})$ (*inner iterations*)
3 Evaluate all functions and derivatives at $\tilde{z}^{k+1}$ needed to construct the QP (4)
4 Possibly condense the QP (4)

Algorithm 1b: Feedback Phase

Input: Solution guess $\tilde{z}^{k+1}$, new state estimate x^{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 x^{k+1} into the QP (4)
6 Solve QP, compute next iterate z^{k+1} via (3) and send first control u_0^{k+1} to the controlled dynamical system

a single QP is solved. This accounts also for the mismatch between the predicted $\tilde{x}^{k+1}$ and actual state x^{k+1} and provides a *generalized tangential predictor* (Diehl et al., 2009). This is a piece-wise affine approximation of the solution manifold $\bar{z}(x^{k+1})$, see (Diehl, 2001). We call the last QP solve *outer iteration* and the calculations in the preparation phase regarding $P_{\text{OCP}}(\tilde{x}^{k+1})$ *inner iterations*. Note that in the ASC, in the feedback phase an additional linear system based on the last Newton iteration's matrix factorization is solved which provides an *affine tangential predictor* (Diehl et al., 2009). A linear system solve in the feedback phase of the ASC is usually cheaper than a QP solve. The affine tangential predictor can not "jump" over active-set changes (Diehl et al., 2009). However, in (Jäschke et al., 2014) an extension for the ASC is introduced to handle active-set changes and non-unique multiplier values via a path-following algorithm. In the AS-RTI, we do not solve $P_{\text{OCP}}(\tilde{x}^{k+1})$ to convergence, but adapt the computational load according to the given sampling time T_s.

In contrast to the AS-RTI, the RTI and MLI have cheaper preparation phases, which enable higher sampling rates. However, compared to them, loosely speaking the new linearization point is closer to the point on the solution manifold $\bar{z}(x^{k+1})$ which we aim to track at time t^{k+1}. This is latter formalized in Section 4. Additionally, we can also use the AS-RTI approach in a parallel MLI setting to refine the linearization point and improve optimality (Nurkanović et al., 2019a). Depending on which calculations we choose to do in the preparation phase of the AS-RTI one can assemble a wide number of different schemes. A single AS-RTI iteration is summarized in Algorithm 1.

To get a new guess for the parameter for the advanced problem, one can simply use the discrete-time system model to predict $\tilde{x}^{k+1} = \psi(x^k, u_0^k)$ while using the last available parameter and control input. Since the evaluation of this function might be expensive we can apply a *shift strategy*, i.e. reuse the solution from a previous problem $P_{\text{OCP}}(x^k)$ and set $\tilde{x}^{k+1} = s_1$. The goal is to ensure a reasonably good prediction, which is close to the true value x^{k+1}.

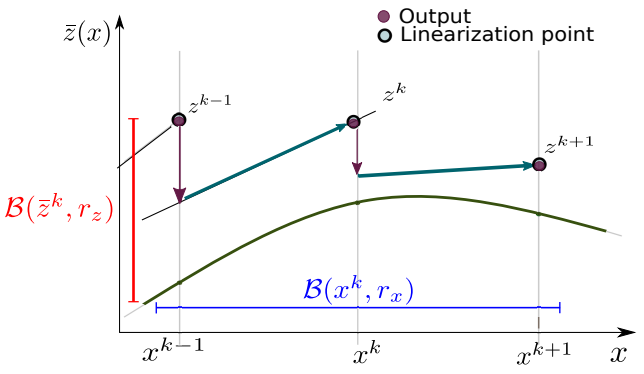

Fig. 1. Tangential predictors and solution manifold tracking with the Real-Time Iteration. The last output is the new linearization point.

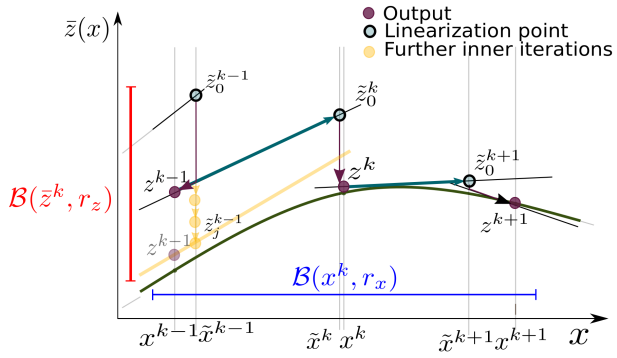

Fig. 2. Tangential predictors and solution manifold tracking with the Advanced Step Real-Time Iteration. The linearization point is due to the inner iterations refined.

Interestingly, the RTI and AS-RTI with an extra QP solve and a perfect prediction ($\tilde{x}^{k+1} = x^{k+1}$) have the same linearization points, but different outputs $u^0(x^{k+1})$, cf. *Lemma* 1 in (Nurkanović et al., 2019b).

Figures 1 and 2 illustrate the linearization points, outputs and tangential predictors of the RTI scheme and the AS-RTI with an extra QP solve in the preparation phase, respectively. In the AS-RTI case we use the tangential predictor from the previous iteration and get $z(\tilde{x}^{k+1})$ as the new linearization point (observe that $\tilde{z}_0^k$ is on the same tangent as the previous output z^{k-1} in Figure 2). When implementing this, one has to take care of not adding the same corrector twice. The QP solution $(\Delta w^k, \lambda_{\text{QP}}^k, \mu_{\text{QP}}^k)$ has both predictor (moving along the tangent) and corrector (getting closer to the manifold) properties, see (Diehl et al., 2009). Now in the AS-RTI solving an extra QP (4), with $\tilde{x}^{k+1}$ one gets $(\Delta\tilde{w}^{k+1}, \tilde{\lambda}_{\text{QP}}^{k+1}, \tilde{\mu}_{\text{QP}}^{k+1})$. This should be added to $(w^{k-1}, \lambda^{k-1}, \mu^{k-1})$ and not to (w^k, λ^k, μ^k), otherwise we add the same corrector twice. Doing further iterations with optimality improving levels of the MLI (see (Bock et al., 2007)) will bring us closer to $\bar{z}(x^{k+1})$. This ultimately leads to smaller numerical errors as discussed in Section 4.

3. CONTRACTION ESTIMATE FOR ABSTRACT REAL-TIME ALGORITHMS

In this section we recall the results from (Zanelli et al., 2019) which we use to analyze the contraction properties of the AS-RTI when using Q-linearly convergent algorithms. We consider again the parametric NLP $P(x)$ in (2). Moreover, due to the considered formulation, the results are more general than for the special case of $P_{\text{OCP}}(x)$ in (1).

Following (Tran-Dinh et al., 2012; Zanelli et al., 2019) one can write the first order necessary optimality conditions of the NLP (2), i.e. the KKT conditions, as:

$$\nabla f(w) + \nabla g(w)\lambda + \mathcal{N}_\Omega(w) \ni 0, \tag{6a}$$

$$g(w) + \hat{M}x = 0. \tag{6b}$$

The set-valued map $\mathcal{N}_\Omega(w) : \mathbb{R}^{n_w} \rightrightarrows \mathbb{R}^{n_w}$ is the normal cone to Ω at w, which is defined as:

$$\mathcal{N}_\Omega(w) \coloneqq \begin{cases} \{v \in \mathbb{R}^n \mid v^\top(w-u) \geq 0,\ \forall u \in \Omega\}, & \text{if } w \in \Omega, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Now, introducing $z \coloneqq (w, \lambda)$, $K = \Omega \times \mathbb{R}^{n_g}$,

$$F(z) \coloneqq \begin{pmatrix} \nabla f(y) + \nabla g(y)\lambda \\ g(y) \end{pmatrix} \quad \text{and} \quad M = \begin{bmatrix} 0 \\ \hat{M} \end{bmatrix}, \tag{7}$$

equations (6) can be rewritten as a Generalized Equation (GE)

$$F(z) + Mx + \mathcal{N}_K(z) \ni 0. \tag{8}$$

We define its solution mapping as

$$\bar{Z}(x) \coloneqq \{z \,|\, F(z) + Mx + \mathcal{N}_K(z) \ni 0\},$$

which is the set of KKT points of $P(x)$ for a given x.

Definition 1. (Strong Regularity (Robinson, 1980)). Let $\bar{z}(x) \in \bar{Z}(x)$ and let $F'(z)$ be the Jacobian of $F(z)$. We say that the GE (8) is strongly regular at $\bar{z}(x)$, if there exist neighborhoods $\mathcal{B}(0, \bar{r}_\delta)$ and $\mathcal{B}(0, \bar{r}_z)$ such that the linearized GE

$$F(\bar{z}(x)) + F'(\bar{z}(x))\Delta z + Mx + \mathcal{N}_K(\bar{z}(x) + \Delta z) \ni \delta, \tag{9}$$

with unknown variable Δz has a unique solution in $\mathcal{B}(0, \bar{r}_z)$ and its solution is Lipschitz continuous in $\mathcal{B}(0, \bar{r}_\delta)$ with a Lipschitz constant σ:

$$\|\Delta\bar{z}(\delta') - \Delta\bar{z}(\delta)\| \leq \sigma\|\delta' - \delta\|, \quad \forall \delta', \delta \in \mathcal{B}(0, \bar{r}_\delta).$$

In the context of optimization, a point $\bar{z}(x)$ is strongly regular if it satisfies the Linear Independence Constraint Qualification (LICQ) and the strong second order sufficient condition (SSOSC), cf. Proposition 1.28 in (Izmailov and Solodov, 2014). More details about this concept can be found in the seminal paper of Robinson (1980) and its application in real-time optimization can be found in e.g. (Tran-Dinh et al., 2012; Zavala and Anitescu, 2010; Anitescu and Zavala, 2017; Zanelli et al., 2019; Dontchev et al., 2013).

We make now the following regularity assumption.

Assumption 2. The set $\bar{Z}(x)$ is nonempty and the corresponding GE (8) is strongly regular at $\bar{z}(x)$ for all $x \in X$.

The following lemma provides conditions under which the solution manifold $\bar{z}(x)$ is Lipschitz continuous.

Lemma 3. (Lemma 3.3, (Tran-Dinh et al., 2012)). Let Assumption 2 hold. Then, for any x in X, there exist neighborhoods $\mathcal{B}(x, \tilde{r}_x)$ of x and $\mathcal{B}(\bar{z}(x), \tilde{r}_z)$ of $\bar{z}(x)$, such that the GE (8) has a unique solution in $\mathcal{B}(\bar{z}(x), \tilde{r}_z)$ for any $x' \in \mathcal{B}(x, \tilde{r}_x)$. Moreover, there exists a positive constant $\gamma \geq 0$ such that the following holds:

$$\|\bar{z}(x') - \bar{z}(x)\| \leq \gamma\|x' - x\|,\ \forall x', x \in \mathcal{B}(x, \tilde{r}_x). \tag{10}$$

Following (Zanelli et al., 2019) we can analyze real-time methods for the parametric NLP (2) independently of the concrete numerical scheme used to solve $P(x)$ associated with a fixed parameter value x. For this we require the use of algorithms that calculate an element of $\bar{Z}(x)$ for some x that have at least local Q-linear convergence.

We denote $\bar{z}^k \coloneqq \bar{z}(x^k)$, and the associated error as $e^k \coloneqq z^k - \bar{z}^k$. When we solve repeatedly (8) for some fixed x^k we will equip the iterations with a second index z_j^k, $j \geq 0$, which counts the number of iterations for a fixed x^k. Regarding AS-RTI, this corresponds to the *inner iterations* and the corresponding error reads as $e_j^k \coloneqq z_j^k - \bar{z}^k$. If we use a predicted parameter $\tilde{x}^{k+1}$, we denote the corresponding iterate as $\tilde{z}_j^{k+1}$ and the error as $\tilde{e}_j^{k+1} \coloneqq \tilde{z}_j^{k+1} - \bar{z}(\tilde{x}^{k+1})$, $j \geq 0$. Properties of a Q-linearly convergent algorithm are summarized in the following assumption.

Assumption 4. (Q-linear Convergence). There exists a radius $\hat{r}_z$ such that, for any given $\bar{z}(x^k) \in \bar{Z}(x^k)$, any $x^k \in X$, and any z_j^k in $\mathcal{B}(\bar{z}(x^k), \hat{r}_z)$, the algorithm used to solve (8) can produce z_{j+1}^k such that

$$\|e_{j+1}^k\| \leq \left(\kappa + \frac{\omega}{2}\|e_j^k\|\right)\|e_j^k\|, \tag{11}$$

for some positive constants $0 \leq \kappa < 1$ and $0 \leq \omega < \infty$.

From now on, we denote $\alpha_j^k \coloneqq (\kappa + \frac{\omega}{2}\|e_j^k\|)$, and analogously for the predicted parameter we adapt the notation $\tilde{\alpha}_j^k \coloneqq (\kappa + \frac{\omega}{2}\|\tilde{e}_j^k\|)$. This assumption covers many standard algorithms for solving (8), or equivalently, to solve the NLP (2). Examples are the generalized Newton-type methods (Tran-Dinh et al., 2012; Izmailov et al., 2013; Zanelli et al., 2019). This abstraction covers also the SQP method from Section 2 (Izmailov et al., 2013). Other examples of Q-linearly convergent algorithms are Sequential Convex Programming (Tran-Dinh et al., 2012) and ADMM (Zanelli et al., 2019).

The next lemma provides a bound on the numerical error due to performing a single iteration of a Q-linearly convergent algorithm for solving (8) for each new parameter x^k.

Lemma 5. (Lemma 1, (Zanelli et al., 2019)). Suppose Assumptions 2 and 4 hold. Then there exist some constants $r_z > 0$ and $r_x > 0$, such that, for any z^k in $\mathcal{B}(\bar{z}^k, r_z)$, and any x^{k+1} in $\mathcal{B}(x^k, r_x)$, the following inequality holds

$$\begin{aligned} \|e^{k+1}\| \leq \kappa\|e^k\| + c_1\|e^k\|\|x^{k+1} - x^k\| + c_2\|e^k\|^2 \\ + c_3\|x^{k+1} - x^k\| + c_4\|x^{k+1} - x^k\|^2, \end{aligned} \tag{12}$$

with $c_1 \coloneqq \omega\gamma$, $c_2 \coloneqq \frac{\omega}{2}$, $c_3 \coloneqq \kappa\gamma$, $c_4 \coloneqq \frac{\omega\gamma^2}{2}$.

Now we have all the tools to state sufficient conditions for a bounded numerical error for tracking the optimal solution manifold $\bar{z}(x)$.

Theorem 6. (Theorem 1, (Zanelli et al., 2019)). Suppose that Assumptions 2 and 4 hold. There exists a positive

constant $0 < r_x^s < r_x$ (same as in *Lemma* 5), such that, if $\|x^{k+1} - x^k\| \leq r_x^s$ for all $k \geq 0$ and $\|e^0\| \leq r_z$, then

$$\|e^{k+1}\| \leq r_z, \ \forall k \geq 0, \tag{13}$$

where

$$r_x := \begin{cases} \frac{\sqrt{(c_3+r_z c_1)^2 - 4c_4(\kappa-1-c_2 r_z)r_z} - (c_3+r_z c_1)}{2c_4} & \text{if } c_4 > 0, \\ \frac{(1-\kappa-c_2 r_z)r_z}{c_3+r_z c_1} & \text{if } c_4 = 0. \end{cases} \tag{14}$$

Similar results on the boundedness of the numerical error for truncated Newton methods, i.e. where the linearization of the GE (8) is solved approximately can be found in (Zavala and Anitescu, 2010).

4. CONTRACTION PROPERTIES

In this section we present the main theoretical results of this paper. We derive sufficient conditions for the contraction of the iterates of the AS-RTI scheme if the inner and outer iterations are carried out with Q-linearly convergent algorithms. Furthermore, we provide sufficient conditions for the boundedness of numerical error. In (Nurkanović et al., 2019b) we proved similar results for Newton-type methods under the simplifying assumption of a fixed active-set. Using the tools from the previous section, it turns out that the proofs follow similar reasoning as in (Nurkanović et al., 2019b). These results hold for general parametric NLPs as $P(x)$ in (2).

First, we will make an assumption regarding the initialization of a Q-linearly convergent algorithm. This will ensure that the numerical error e_j^k for a fixed x^k gets smaller if we perform further iterations of the algorithm. This is needed in the proof of the next theorem.

Assumption 7. (Initialization) Suppose that the following condition holds at an initial point z^0 and a solution $\bar{z}^0$:

$$\|z^0 - \bar{z}^0\| \leq \hat{r}_z < \hat{r}_z^s := 2(1-\kappa)/\omega. \tag{15}$$

Furthermore, we assume to always have a reasonably good parameter prediction, so that we can use the results of Theorem 6.

Assumption 8. (Predicted Parameter) In all iterations of Algorithm 1 the parameter predictions $\tilde{x}^{k+1}$ satisfy:

$$\|x^{k+1} - \tilde{x}^{k+1}\| \leq r_x^s \text{ and } \|\tilde{x}^{k+1} - x^k\| \leq r_x^s, \quad \forall k \geq 0, \tag{16}$$

where the positive constant r_x^s is the same as in Theorem 6.

When we use different Q-linearly convergent algorithms for inner and outer iterations, we equip the corresponding constants from Assumption 4 or Lemma 5 with a superscript *in* and *out*, respectively. For example, in Assumption 4 we distinguish between κ^{in} and κ^{out} for inner and outer iterations, respectively. With the next theorem we provide a general contraction estimate for an iteration of the AS-RTI scheme described in Section 2.

Theorem 9. Suppose that Assumptions 2, 4, 7 and 8 hold. Moreover, assume that we make $j \geq 0$ inner iterations on an advanced problem $P(\tilde{x}^{k+1})$ in the preparation phase of AS-RTI and that $r_z = \min(r_z^{\text{out}}, r_z^{\text{in}})$. Then, for the sequence of errors $\{e^k\}$, the following inequality holds:

$$\begin{aligned} \|e^{k+1}\| \leq (\tilde{\alpha}_0^{k+1})^j \left[\nu^k \|e^k\| + \zeta^k \|\tilde{x}^{k+1} - x^k\|\right] \\ + \eta^k \|x^{k+1} - \tilde{x}^{k+1}\|, \end{aligned} \tag{17}$$

where we have defined the positive constants $\hat{\nu}^k$, ν^k, ζ^k, η^k, respectively, as:

$$\hat{\nu}^k := \kappa^{\text{out}} + c_1^{\text{out}} \|x^{k+1} - \tilde{x}^{k+1}\| + c_2^{\text{out}} (\tilde{\alpha}_0^{k+1})^j \|\tilde{e}_0^{k+1}\|, \tag{18}$$

$$\nu^k := \hat{\nu}^k \big(\kappa^{\text{in}} + c_1^{\text{in}} \|\tilde{x}^{k+1} - x^k\| + c_2^{\text{in}} \|e^k\|\big), \tag{19}$$

$$\zeta^k := \hat{\nu}^k \big(c_3^{\text{in}} + c_4^{\text{in}} \|\tilde{x}^{k+1} - x^k\|\big), \tag{20}$$

and

$$\eta^k := c_3^{\text{out}} + c_4^{\text{out}} \|x^{k+1} - \tilde{x}^{k+1}\|. \tag{21}$$

Proof: To use the same r_z for inner and outer iterations we set $r_z = \min\{r_z^{\text{out}}, r_z^{\text{in}}\}$. For the first inner iteration, due to Assumptions 2 and 4 there exist some constants $r_z > 0$ and $r_x > 0$, such that, for any z^k in $\mathcal{B}(\bar{z}^k, r_z)$. Moreover, due to Assumption 8, it holds that $\tilde{x}^{k+1}$ in $\mathcal{B}(x^k, r_x)$. Using Lemma 5, it holds that

$$\begin{aligned} \|\tilde{e}_0^{k+1}\| \leq \big(\kappa^{\text{in}} + c_1^{\text{in}} \|\tilde{x}^{k+1} - x^k\| + c_2^{\text{in}} \|e^k\|\big) \|e^k\| \\ + \big(c_3^{\text{in}} + c_4^{\text{in}} \|\tilde{x}^{k+1} - x^k\|\big) \|\tilde{x}^{k+1} - x^k\|. \end{aligned} \tag{22}$$

Due to Assumption 8 and Theorem 6, it holds that $\|\tilde{e}_0^{k+1}\| \leq r_z \leq \hat{r}_z$. Assuming we made $j \geq 0$ iterations, due to Assumptions 4 and 7 the following holds

$$\begin{aligned} \tilde{\alpha}_1^{k+1} = \kappa^{\text{in}} + c_1^{\text{in}} \|\tilde{e}_1^{k+1}\| &\overset{(11)}{\leq} \kappa^{\text{in}} + c_1^{\text{in}} \underbrace{\tilde{\alpha}_0^{k+1}}_{<1} \|\tilde{e}_0^{k+1}\| \\ &< \kappa^{\text{in}} + c_1^{\text{in}} \|\tilde{e}_0^{k+1}\| = \tilde{\alpha}_0^{k+1}. \end{aligned}$$

Moreover, applying this inductively we have

$$\|\tilde{e}_j^{k+1}\| \leq (\tilde{\alpha}_0^{k+1})^j \|\tilde{e}_0^{k+1}\| < r_z. \tag{23}$$

Furthermore, due to the last inequality it holds that $\tilde{z}_j^{k+1} \in \mathcal{B}(\bar{z}(\tilde{x}^{k+1}), r_z)$ and x^{k+1} in $\mathcal{B}(\tilde{x}^{k+1}, r_x)$ (holds due to Assumption 8). Therefore, we can use Lemma 5 for the *outer iteration* which yields

$$\begin{aligned} \|e^{k+1}\| \leq \big(\kappa^{\text{out}} + c_1^{\text{out}} \|x^{k+1} - \tilde{x}^{k+1}\| + c_2^{\text{out}} \|\tilde{e}_j^{k+1}\|\big) \|\tilde{e}_j^{k+1}\| \\ + \underbrace{\big(c_3^{\text{out}} + c_4^{\text{out}} \|x^{k+1} - \tilde{x}^{k+1}\|\big)}_{\overset{(21)}{=} \eta^k} \|x^{k+1} - \tilde{x}^{k+1}\|. \end{aligned}$$

Using the estimate for $\|\tilde{e}_j^{k+1}\|$ from (23), from the last equation we get

$$\begin{aligned} \|e^{k+1}\| \leq \underbrace{\big(\kappa^{\text{out}} + c_1^{\text{out}} \|x^{k+1} - \tilde{x}^{k+1}\| + c_2^{\text{out}} (\tilde{\alpha}_0^{k+1})^j \|\tilde{e}_0^{k+1}\|\big)}_{\overset{(18)}{=} \hat{\nu}^k} \\ \cdot (\tilde{\alpha}_0^{k+1})^j \|\tilde{e}_0^{k+1}\| + \eta^k \|x^{k+1} - \tilde{x}^{k+1}\|. \end{aligned}$$

Now, if we replace $\|\tilde{e}_0^{k+1}\|$ with its upper bound (22), we obtain

$$\begin{aligned} \|e^{k+1}\| \leq (\tilde{\alpha}_0^{k+1})^j \hat{\nu}^k \big[\big(\kappa^{\text{in}} + c_1^{\text{in}} \|\tilde{x}^{k+1} - x^k\| + c_2^{\text{in}} \|e^k\|\big) \|e^k\| \\ + \big(c_3^{\text{in}} + c_4^{\text{in}} \|\tilde{x}^{k+1} - x^k\|\big) \|\tilde{x}^{k+1} - x^k\|\big] \\ + \eta^k \|x^{k+1} - \tilde{x}^{k+1}\|. \end{aligned} \tag{24}$$

Using the definitions of ν^k in (19) and ζ^k in (20) the inequality (17) follows from the last inequality. This completes the proof. □

Similar to *Theorem* 6, we give sufficient conditions for the boundedness of the numerical error of the AS-RTI.

Lemma 10. Suppose that Assumptions 2, 4, 7 and 8 hold. Moreover, assume that we perform $j \geq 0$ inner iterations on an advanced problem $P(\tilde{x}^{k+1})$ in the preparation phase of AS-RTI. Then there exists a positive constant $0 < r_x^s < r_x$ (same as in *Theorem* 6), such that, if $\|x^{k+1} - x^k\| \leq r_x^s$ for all $k \geq 0$ and $\|e^0\| \leq r_z$, then

$$\|e^{k+1}\| \leq r_z, \ \forall k \geq 0, \tag{25}$$

where r_x^s is given by (14) and $r_z = \min(r_z^{\text{out}}, r_z^{\text{in}})$.

Proof: Taking $r_z = \min(r_z^{\text{out}}, r_z^{\text{in}})$ we can use the same r_z for both inner and outer iterations. Since the assumptions of Theorem 6 are satisfied, for the first inner iteration corresponding to $P(x^0)$ and $P(\tilde{x}^1)$ we conclude that $\|\tilde{e}_0^1\| \leq r_z \leq \hat{r}_z$ (cf. also proof of Lemma 1 in Zanelli et al. (2019)). Using this inequality and Assumptions 4 and 7, we conclude, that for a fixed parameter (further inner iterations) $\tilde{\alpha}_j^1 < 1$ for all $j \geq 0$ (the errors shrinks), i.e. it holds that $\|\tilde{e}_j^1\| < r_z$, $\forall j \geq 0$. For the outer iteration corresponding to $P(\tilde{x}^1)$ and $P(x^1)$, since $\|\tilde{e}_j^1\| < r_z$ and $\|x^1 - \tilde{x}^1\| \leq r_x^s$, by applying Theorem 6 we obtain that $\|e^1\| \leq r_z$. Applying this argument inductively, we conclude that (25) holds $\forall k \geq 0$. □

Following (Nurkanović et al., 2019b), from the results of Theorem 9 we can make several observations regarding a new iterate z^{k+1}: (1) having a better parameter guess, i.e. having smaller $\|x^{k+1} - \tilde{x}^{k+1}\|$, decreases the distance to $\bar{z}^{k+1}$, (2) being closer to the solution in the previous iterate, i.e. smaller $\|e^k\|$, also improves the solution, (3) increasing the number of inner iterations j further decreases the error e^{k+1}, (4) the distance between the two parameters also $\|x^{k+1} - x^k\|$ affects the numerical error, (5) the value of the constant κ^{in} for the inner iterations affects the solution since for smaller κ^{in} the term $(\tilde{\alpha}_0^{k+1})^j$ shrinks faster. Having several inner iterations will make the first term on the r.h.s of (17) become very small thus the error is also small. Furthermore, having $j \to \infty$ in the limit and $\tilde{x}^{k+1} = x^{k+1}$ we obtain ideal NMPC in the nominal case. Obviously, in the context of NMPC the number of inner iterations we can make depends on the available time determined by the sampling time T_s. On the other hand, if T_s is fixed and larger, the distance between two subsequent parameter grows ($\|x^{k+1} - x^k\| = O(T_s)$). This also implies the growth of the numerical error. We can compensate this with further inner iterations. A natural extension is to perform more corrector steps in the feedback phase for a fixed new parameter and reduce the numerical error, cf. (Dontchev et al., 2013). However, in the context of NMPC, such a control input becomes outdated and, due to the feedback delay, the controller performance can be degraded.

5. NUMERICAL EXAMPLE

As a numerical benchmark we consider a diesel generator (DG) which is connected with a power line to a time-varying load. A similar example, with an additional photovoltaic source was considered in (Scholz et al., 2019). A typical DG consists of a synchronous generator (SG) with governor (GOV) and an automatic voltage regulator (AVR), as depicted in Fig. 3, for more details see (Kundur, 1993). In this paper we consider an SG model with 5 differential and 11 algebraic states (Kundur, 1993). The goal of the GOV is to control the power generation P_1 of the DG by controlling the diesel engine. The input of the GOV is the generator frequency ω, as well as the reference power P_{ref}. We use a standard IEEE DEGOV1 model, which consists of 8 differential and 2 algebraic states. The AVR controls the terminal voltage of the generator through the field winding voltage E_{fd} from the exciter. The inputs to the AVR are the reference voltage V_{ref} and the DG voltage V_1. For the AVR we use the standard IEEE AC5A model, which consists of 5 differential and one algebraic state. Due to page limits we omit a more detailed description of the models, for details on DG and general microgrid modeling we refer to (Kundur, 1993; Nurkanović et al., 2020).

The DG has a nominal power of $S_N = 325$ kVA and the control variables are $u(t) = (P_{\text{ref}}(t), V_{\text{ref}}(t))$. The admittance of the power line is $Y_{12} = 137.93 - 344.83i\ \Omega^{-1}$ and it connects the DG at node 1 with the load at node 2. The load is modeled as time varying parameter. The DG, the power line and load, where the connection is modeled via power-flow, cf. (Kundur, 1993), result in a Differential Algebraic Equation (DAE) of index-1 with 17 differential states $x(t)$ and 18 algebraic states $z(t)$. Our goal is to regulate the voltage at the load V_2 and the electric frequency ω of the DG at 1 per unit (p.u.). We express this with the objective

$$L(x(t), z(t), u(t)) = \rho(\|\omega(t) - 1\|^2 + \|V_2(t) - 1\|^2), \tag{26}$$

where ρ is a penalty factor which we fix to 10^2. Moreover, we require the voltages at the DG (V_1) and load (V_2), and the frequency ω to be in specific ranges:

$$0.9 \text{ p.u.} \leq V_i \leq 1.1 \text{ p.u.}, \quad i = 1, 2, \tag{27a}$$

$$0.95 \text{ p.u.}, \leq \omega \leq 1.05 \text{ p.u.}. \tag{27b}$$

Additionally, the production of active and reactive power by the DGs is limited by its nominal power

$$P_1^2 + Q_1^2 \leq S_N^2. \tag{28}$$

Together with the objective (26), the discretized DAE model of the DG, and the constraints (27) and (28) evaluated at the discretization grid we obtain an OCP of the form $P_{\text{OCP}}(x)$ in (1). To discretize the continuous-time dynamics we use direct multiple shooting (Bock and Plitt, 1984) with the Gauss-Legendre Implicit Runge-Kutta scheme of order four with a fixed step-size $h = T/N$. For the NMPC prediction horizon we chose $T = 10$ s.We perform the numerical benchmark with two different discretization grids, where the trajectories are discretized using $N = 50$ and $N = 40$ multiple shooting nodes, which results in a sampling time of $T_s = 200$ ms and $T_s = 250$ ms, respectively. We use a time-varying load profile at node 2 with $P_2 = 300$ kW and $Q_2 = 100$ kVAr with a scheduled load increase at $t = 4$ s to $P_2 = 305$ kW and $Q_2 = 100$ kVAr which is also in the NMPC prediction. At $t = 1$ s an unforeseen load drop to $P_2 = 30$ kW and $Q_2 = 10$ kVAr occurs until the scheduled load increases. After noticing the load drop, the prediction is adapted to the new value after one sampling time.

We implement the AS-RTI in `acados` through its MATLAB interface (Verschueren et al., 2019). We use `HPIPM` (Frison and Diehl, 2020), an interior-point based QP solver for the SQP subproblems. In all experiments we use a Gauss-Newton (GN) Hessian approximation. In the simulation we compare the following different schemes: 1)

AS-RTI with k inner GN-SQP iterations denoted as AS-RTI-k and, 2) the RTI. We observed that the solution does not improve with further inner iterations even when solving the advanced problem to convergence. The simulation results for the two schemes with $T_s = 250$ms are depicted in Fig. 4. Both schemes are able to stabilize the system and bring ω and V_2 to 1. Observe that the overshoot at $t = 1$ s remains the same for all schemes, since the load drop is not predicted, and the NMPC controller can react only after noticing it, i.e. after the time T_s has passed. The load changes both unpredicted ($t = 1$s) and predicted ($t = 4$ s) for more than 90% compared to the initial load value. Compared to the AS-RTI, the voltage oscillations with the RTI last longer after the predicted load change. The CPU times of all schemes in this experiments are provided in Table 1. We compare the schemes by the resulting running cost, defined as $J(T_{\text{sim}}) := \int_0^{T_{\text{sim}}} L(x(\tau), z(\tau), u^*_{\text{NMPC}}(\tau))\ \mathrm{d}\tau$, where $u^*_{\text{NMPC}}(t)$ is the resulting NMPC closed loop input fed back to the system over the simulation time T_{sim}. All considered schemes are real-time feasible, however with the AS-RTI with few additional computations we improve the running cost $J(\cdot)$ and the improvement is larger with a larger sampling time. With more inner iterations the computational load in the preparation phase is increasing, however in the feedback phase we still solve only a single QP and thus the feedback delay stays small. All simulations are run on a HP Z-book equipped with an Intel i7-6820HQ CPU with 2.70 GHz and 16 GB RAM under Windows 10.

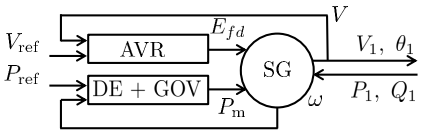

Fig. 3. Outline of a DG model. It consists of a synchronous generator (SG), automatic voltage regulator and exciter (AVR), and of a diesel engine (DE) and governor model (GOV).

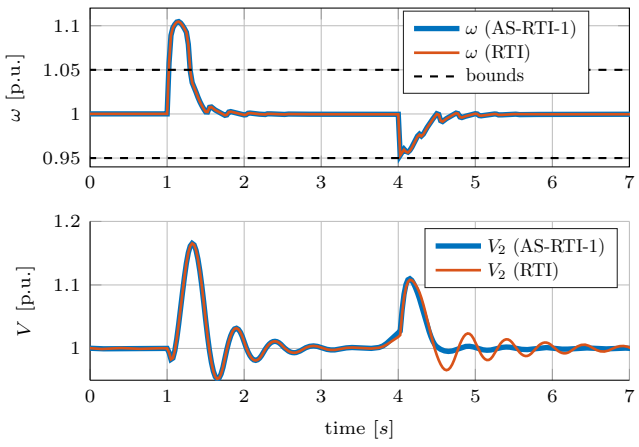

Fig. 4. Frequency of the DG (top) and voltage at the load (bottom) for two different schemes with $T_s = 250$ms: 1) AS-RTI with one inner iterations (blue), 2) RTI (red).

Algorithm	N	J	Prep. Ph.			Feedback Ph.		
			max	min	mean	max	min	mean
RTI	50	8.04	16.01	11.00	12.19	4.00	1.96	2.50
AS-RTI-1	50	7.69	40.97	24.03	26.85	4.00	1.97	2.53
AS-RTI-2	50	7.69	58.99	37.00	41.42	3.02	2.00	2.61
AS-RTI-3	50	7.69	67.03	50.98	55.18	4.00	1.97	2.55
RTI	40	12.28	12.03	8.03	9.94	2.96	0.99	1.76
AS-RTI-1	40	11.76	23.04	19.00	20.30	2.04	0.97	1.70
AS-RTI-2	40	11.77	41.01	29.99	32.44	3.23	0.98	1.85
AS-RTI-3	40	11.77	54.02	38.99	43.20	2.03	0.93	1.68

Table 1. CPU times of different NMPC schemes in milliseconds.

6. CONCLUSION

In this paper we derived a contraction estimate for the AS-RTI under active-set changes. Our result holds for general algorithms with at least Q-linear convergence, including the setting where different algorithms for inner and outer iterations are used. This abstraction covers a broad spectrum of methods used in practice. Furthermore, we provided sufficient conditions for boundedness of the numerical error of the AS-RTI. The efficacy of the AS-RTI is demonstrated on a complex numerical benchmark showing that a few additional computations can improve the controller performance confirming our theoretical findings.

REFERENCES

Anitescu, M. and Zavala, V.M. (2017). MPC as a DVI: Implications on Sampling Rates and Accuracy. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 1933–1938. IEEE.

Bock, H.G., Diehl, M., Kostina, E.A., and Schlöder, J.P. (2007). Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-Time and Online PDE-Constrained Optimization*, 3–22. SIAM. doi:10.1137/1.9780898718935.ch1.

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

Diehl, M. (2001). *Real-Time Optimization for Large Scale Nonlinear Processes*. Ph.D. thesis, University of Heidelberg. URL `http://archiv.ub.uni-heidelberg.de/volltextserver/1659/`.

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

Dontchev, A.L., Krastanov, M., Rockafellar, R.T., and Veliov, V.M. (2013). An Euler-Newton Continuation Method for Tracking Solution Trajectories of Parametric Variational Inequalities. *SIAM Journal on Control and Optimization*, 51(3), 1823–1840.

Frison, G. and Diehl, M. (2020). HPIPM: a high-performance quadratic programming framework for model predictive control. *arXiv preprint arXiv:2003.02547*.

Izmailov, A.F., Kurennoy, A.S., and Solodov, M.V. (2013). The Josephy-Newton Method for Semismooth General-

ized Equations and Semismooth SQP for Optimization. *Set-Valued and Variational Analysis*, 21(1), 17–45.

Izmailov, A.F. and Solodov, M.V. (2014). *Newton-type methods for optimization and variational problems*. Springer.

Jäschke, J., Yang, X., and Biegler, L.T. (2014). Fast economic model predictive control based on NLP-sensitivities. *Journal of Process Control*, 24(8), 1260–1272.

Kundur, P. (1993). *Power System Stability and Control*. McGraw-Hill.

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

Nurkanović, A., Albrecht, S., and Diehl, M. (2019a). Multi Level Iterations for Economic Nonlinear Model Predictive Control. In T. Faulwasser, M.A. Müller, and K. Worthmann (eds.), -. Springer.

Nurkanović, A., Mešanović, A., Zanelli, A., Frey, J., Frison, G., Albrecht, S., and Diehl, M. (2020). Real-time nonlinear model predictive control for microgrid operation. In *Proceedings of the American Control Conference (ACC)*. Denver, USA. (submitted).

Nurkanović, A., Zanelli, A., Albrecht, S., and Diehl, M. (2019b). The Advanced Step Real Time Iteration for NMPC. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*. Nice, France. (accepted).

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

Robinson, S.M. (1980). Strongly Regular Generalized Equations. *Mathematics of Operations Research, Vol. 5, No. 1 (Feb., 1980), pp. 43-62*, 5, 43–62.

Scholz, R., Nurkanović, A., Mešanović, A., Gutekunst, J., Potscka, A., Bock, H.G., and Kostina, E. (2019). Model-based Optimal Feedback Control for Microgrids with Multi-Level Iterations. *Submitted to Operations Reserach 2019*.

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

Verschueren, R., Frison, G., Kouzoupis, D., van Duijkeren, N., Zanelli, A., Novoselnik, B., Frey, J., Albin, T., Quirynen, R., and Diehl, M. (2019). acados: a modular open-source framework for fast embedded optimal control. *arXiv preprint*. URL `https://arxiv.org/abs/1910.13753`.

Zanelli, A., Tran-Dinh, Q., and Diehl, M. (2019). Contraction Estimates for Abstract Real-Time Algorithms for NMPC. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*. Nice, France. (accepted).

Zavala, V. and Anitescu, M. (2010). Real-Time Nonlinear Optimization as a Generalized Equation. *SIAM J. Control Optim.*, 48(8), 5444–5467.

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