An Adaptive Partial Sensitivity Updating Scheme for Fast Nonlinear Model Predictive Control

Yutao Chen, Mattia Bruschetta, Davide Cuccato, and Alessandro Beghi

Abstract—In recent years, efficient optimization algorithms for Nonlinear Model Predictive Control (NMPC) have been proposed, that significantly reduce the on-line computational time. In particular, direct multiple shooting and Sequential Quadratic Programming (SQP) are used to efficiently solve Nonlinear Programming (NLP) problems arising from continuous-time NMPC applications. One of the computationally demanding steps for on-line optimization is the computation of sensitivities of the nonlinear dynamics at every sampling instant, especially for systems of large dimensions, strong stiffness, and when using long prediction horizons. In this paper, within the algorithmic framework of the Real-Time Iteration (RTI) scheme based on multiple shooting, an inexact sensitivity updating scheme is proposed, that performs a partial update of the Jacobian of the constraints in the NLP. Such update is triggered by using a Curvature-like Measure of Nonlinearity (CMoN), so that only sensitivities exhibiting highly nonlinear behaviour are updated, thus adapting to system operating conditions and possibly reducing the computational burden. An advanced tuning strategy for the updating scheme is provided to automatically determine the number of sensitivities being updated, with a guaranteed bounded error on the Quadratic Programming (QP) solution. Numerical and control performance of the scheme is evaluated by means of two simulation examples performed on a dedicated implementation. Local convergence analysis is also presented and a tunable convergence rate is proven, when applied to the SQP method.

Index Terms—nonlinear model predictive control, RTI, partial sensitivity update optimization algorithms

I. INTRODUCTION

Nonlinear Model Predictive Control (NMPC) has been studied and applied intensively in the last decades. In NMPC, a nonlinear Optimal Control Problem (OCP) has to be solved on-line at every sampling instant. The OCP can be converted to a finite dimensional Nonlinear Programming (NLP) problem by direct methods, such as direct multiple shooting [1] and direct collocation [2]. The NLP problem can then be solved by a number of optimization algorithms, e.g., Interior Point Methods (IPM) [2] and Sequential Quadratic Programming (SQP) [3]. Fast NMPC algorithms based on direct methods have been proposed to speed up on-line optimization, see [4]–[6].

Efficient SQP algorithms based on direct multiple shooting for systems governed by Differential Algebraic Equations (DAE) have been thoroughly studied (see e.g. [7]). One of the computationally demanding steps of SQP methods when applied to NMPC is the computation of sensitivities at each sampling instant, i.e. the Hessian of the Lagrangian and the Jacobian of the constraints. There are several methods for computing such sensitivities, e.g. finite difference [8], complex-step differentiation [9], and automatic differentiation [10].

Particularly, the Jacobian of constraints contains sensitivities of integration operators that parameterize continuous-time dynamics. Although efficient implementations of numerical integration with sensitivity generation are available [11], [12], sensitivity computation of this type still largely contributes to the overall on-line computational burden, especially for systems that are highly stiff or governed by implicit differential equations and DAEs.

In this paper, on one most promising SQP-based NMPC algorithms that is the Real-Time Iteration (RTI) [13], in which only one SQP iteration is performed at each sampling instant, is taken as the reference approach. The underlying idea is to initialize the new NLP by using information from the previous one, including states, controls, and multipliers making the closed-loop trajectory converging as system dynamics evolve, i.e. “on the fly” [14].

In the RTI framework with multiple shooting parameterization, a number of tailored approaches are available that employ suitable inexact sensitivities. In Multi-Level RTI (ML-RTI) [15], sensitivities are updated every \( m > 1 \) sampling instants. Hence, sensitivities are updated at a slower rate than other QP components. However, the choice of \( m \) is not intuitive and generally application dependent, thus requiring a long and complex tuning procedure. In ADJoint sensitivity RTI (ADJ-RTI) sensitivities computed off-line are used [16]–[19], and an adjoint sensitivity on-line computation is performed to identify the correct active-set and to ensure local convergence. Although computational cost is considerably reduced, thanks to the reduced number of sensitivity computations and condensing operations [18], this approach is effective only for systems exhibiting mild nonlinearities. Recently, partial sensitivity updating schemes called CMoN-RTI and DOPUS, that are tailored for multiple-shooting based NMPC, have been proposed [20]–[22]. In such schemes, the multi-stage feature of NLP problems arising in NMPC applications and the iterative nature of the solver are exploited. As a result, sensitivities are partially updated between two consecutive sampling instants. A so-called Curvature-like Measure of Nonlinearity (CMoN) or norm-criterion is used in a monitoring strategy to decide which and how many sensitivities should be updated. However, these monitoring strategies rely on heuristics and are strongly dependent on the application at hand.
In this paper, the partial sensitivity scheme CMoN-RTI of [21] is extended and improved. In particular, three main features are provided, namely:

- a solution accuracy control strategy;
- a practical tuning procedure;
- convergence analysis.

From parametric optimization theory, the accuracy of the QP solution is related to parameters in the monitoring strategy. An advanced tuning strategy for CMoN-RTI is here developed that provides an automatic way to select which and how many sensitivities should be updated, while guaranteeing the QP solution a bounded Distance to Optimum (DtO) (i.e. the distance between the solution of the inexact sensitivity QP and that of the exact sensitivity QP). The tuning parameter is therefore the DtO tolerance, which has an important physical meaning.

The proposed scheme can significantly reduce the computational load when the system nonlinear dynamics are excited only on a small part of the prediction horizon (e.g., when regulating a system around its steady state or tracking a reference with look ahead). Moreover, since the additional computational time required by CMoN-RTI with respect to RTI is almost negligible, CMoN-RTI is a sensible alternative in all the scenarios where RTI is effective, as it usually yields an improvement in the average computational performance, hence saving computational power, and possibly an increase of the control frequency. In the worst case, CMoN-RTI degrades to RTI. A practical implementation of the scheme is given and its effectiveness is demonstrated by closed-loop simulations on two classical examples. An application of CMoN-RTI applied to the SQP framework with multiple iterations is also introduced. A tunable local convergence rate is proven.

The paper is organized as follows. In Section II RTI and some inexact sensitivity schemes are briefly introduced to define the algorithmic framework. In Section III the CMoN-RTI scheme is presented in detail. Section IV is devoted to the derivation of the advanced tuning strategy and to practical implementation aspects. In Section V closed-loop simulation results using CMoN-RTI are shown. The CMoN-SQP is described in Section VI, and its convergence properties are discussed and demonstrated by a numerical example.

II. ALGORITHMIC FRAMEWORK

In this section, the standard RTI scheme [13] is introduced as the algorithmic framework of the paper. ML-RTI [15] and ADJ-RTI schemes [17] are here presented as two variants of RTI, with inexact sensitivity updating strategies.

A. Real-Time Iteration Scheme

In NMPC, a NLP problem can be formulated by applying direct multiple shooting [11] to an OCP over the prediction horizon $T = [t_0, t_f]$, which is divided into $N$ shooting intervals $[t_0, t_1, \ldots, t_N]$, as follows

\[
\begin{align*}
\min_{s_k, u_k} & \sum_{k=0}^{N-1} h_k(s_k, u_k) + h_N(s_N) \\
\text{s.t.} & \quad 0 = x_0 - \hat{x}_0, \\
& \quad 0 = x_{k+1} - \phi_k(x_k, u_k), \quad k = 0, 1, \ldots, N - 1, \\
& \quad 0 \geq r(x_k, u_k), \quad k = 0, 1, \ldots, N - 1, \\
& \quad 0 \geq l(s_N),
\end{align*}
\]

where $\hat{x}_0$ is the measurement of the current state. System states $x_k \in \mathbb{R}^{n_x}$ are defined at the discrete time point $t_k$ for $k = 0, \ldots, N$ and the control inputs $u_k \in \mathbb{R}^{n_u}$ are piece-wise constant. Here, (1d) is the inequality constraint where $r(x_k, u_k)$ is a numerical integration operator that solves the following initial value problem (IVP)\(^{\text{I}}\) and return the solution at $t_{k+1}$.

$$0 = f(\dot{x}, x(t), u(t), t), \quad x(0) = x_k.$$ 

The NLP problem (1) depends on the state and control initialization $w$ and the state measurement $\hat{x}_0$, where $w = (w_0^T, w_1^T, \ldots, w_{N-1}^T, x_N^T)\top$ and $w_k = (x_k^T, u_k^T)\top$ for $k = 0, \ldots, N - 1$. By embedding (1b) into (1c), the NLP problem can be written in a compact form as

\[
\begin{align*}
\min_{w} & \quad A(w) \\
\text{s.t.} & \quad B(w) = 0, \\
& \quad C(w) \leq 0.
\end{align*}
\]

In RTI, problem (2) is solved by a tailored SQP method, where only one SQP iteration is performed at each sampling instant. At sampling instant $i$, the QP subproblem initialized at $w^i$ is defined as

\[
\begin{align*}
\min_{\Delta w} & \quad \frac{1}{2} \Delta w^\top H^i \Delta w + \nabla A^i \Delta w \\
\text{s.t.} & \quad b^i = 0, \\
& \quad c^i \leq 0,
\end{align*}
\]

where $\Delta w = w - w^i$ and $\nabla$ is the gradient or Jacobian operator over $w$ if no subscript is provided. The equality and inequality constraints are given by

$$b^i = B(w^i) + \nabla B(w^i) \Delta w,$$

$$c^i = C(w^i) + \nabla C(w^i) \Delta w.$$

$H^i$ is the Hessian of the Lagrangian of (2), which is defined by $L(w, \lambda, \mu) := A(w) + \lambda^\top B(w) + \mu^\top C(w)$, where $\lambda, \mu$ are Lagrangian multipliers associated with equality and inequality constraints, respectively. For most QP problems arising from NMPC, the Gauss-Newton Hessian approximation provides a sufficiently accurate Hessian with reduced computational burden [13]. Being independent of Lagrangian multipliers, the Gauss-Newton Hessian is adopted in this paper. Given the multi-stage nature of problem (1), matrices $H^i$ and $\nabla C(w^i)$

\(^{\text{I}}\)For simplicity we consider Ordinary Differential Equations (ODEs) only but the extension to DAEs can be easily derived.
are block diagonal. In particular, the Jacobian matrix $\nabla B(w^i)$ has the following form:

$$\nabla B(w^i) = \begin{bmatrix} I_a \\ -\nabla \phi_0^T \\ -I_a \\ \vdots \\ -\nabla \phi_{N-1}^T \\ -I_a \\ \end{bmatrix}, \tag{4}$$

where $\nabla \phi^k = \nabla \phi_k(x_k^i, u^i_k)$, $k = 0, \ldots, N - 1$ and $I_a$ is an identity matrix of size $a$. The solution $(\Delta w^{QP}, \lambda^{QP}, \mu^{QP})$ of (3) is used to update the solution of (2) by a single, full Newton step as

$$w^{i+1} = w^i + \Delta w^{QP}, \tag{5}$$
$$\lambda^{i+1} = \lambda^{QP}, \mu^{i+1} = \mu^{QP}.$$ 

Since only one QP problem is solved at each sampling instant, the RTI scheme is a special case of the linear, time-varying MPC strategy \[14\].

B. Inexact Sensitivity Schemes

When formulating problem (3), the Jacobian matrix $\nabla B(w)$ is computed at each sampling instant to obtain the current linearization of system dynamics \[17\]. Such computation involves sensitivity propagation of the numerical integration operator $\phi_k$ in (13) w.r.t. the initialization $w_k$ for each shooting interval, which can be computationally expensive for systems that are highly stiff or governed by implicit differential equations and DAEs. To avoid the repeated sensitivity computations, in ML-RTI, firstly proposed in \[15\], the computation of $\nabla B(w)$ is performed at a slower rate than other components, e.g. $\nabla A$ and $B(w)$. Therefore, problem (3) with currently available but inexact sensitivities is solved at a faster rate \[23\], \[24\]. To account for the inexact Jacobian, a so-called optimality improvement step is employed by solving a slightly modified QP problem as

$$\begin{align*}
\min_{\Delta w} & \frac{1}{2} \Delta w^T H^i \Delta w + \nabla \tilde{A}^i \Delta w \\
\text{s.t.} & \tilde{b}^i = 0, \\
& c^i \leq 0,
\end{align*} \tag{6a}$$

where $\tilde{b}^i = B(w^i) + \nabla \tilde{B} \Delta w$ and $\nabla \tilde{B}$ is the Jacobian from the previous sample. The QP gradient is modified as

$$\nabla \tilde{A}^i = \nabla A^i + (\nabla B(w^i) - \nabla \tilde{B})^T \lambda^i,$$

where $\nabla \tilde{A}$ can be efficiently computed by applying adjoint sensitivity propagation schemes for $\nabla B^T(\tilde{w})\lambda$ \[17\], \[18\], which are much cheaper than the computation of the full Jacobian matrix $\nabla B(w)$. The multi-level framework with inexact sensitivities can be summarized in Algorithm 1.

**Algorithm 1** Multi-Level inexact sensitivity RTI scheme

1: Initialize (2) at $(w^0, \lambda^0, \mu^0)$. Choose a sensitivity update interval $m \in \mathbb{N}^+$.  
2: for $i = 0, 1, \ldots, m$ do  
3: Compute $H^i, \nabla \tilde{A}^i, B^i, C^i, \nabla C^i$.  
4: if $i \mod m = 0$ then  
5: Update the sensitivity $\nabla B^i$.  
6: Set $\nabla B \leftarrow \nabla B^i$.  
7: end if  
8: Solve (6) and obtain $(\Delta w^{QP}, \lambda^{QP}, \mu^{QP})$.  
9: Update the solution of the NLP problem by (5).  
10: end for

The ADJ-RTI scheme is a special variant of Algorithm 1 with $m = \infty$, in which the Jacobian matrix is computed only once off-line at the pre-defined initialization trajectory $w^0$ \[17\]. When applied in the SQP framework with multiple SQP iterations, such an inexact sensitivity scheme is proved to be convergent to the local minimum of the exact sensitivity NLP problem \[2\], \[15\]. The feasibility and stability of the adjoint scheme in SQP framework, without the optimality improvement, is analyzed in \[19\]. However, there remain some open issues when applying Algorithm 1 in RTI:

- It is not trivial to choose an appropriate sensitivity update interval $m$, or a pre-defined trajectory $w^0$, such that the inexact Jacobian $\nabla \tilde{B}$ is a good approximation of the exact one for every sampling instant $i > 0$.
- The sensitivities over the prediction horizon, either updated or not, are treated as a whole. Therefore, the structure of the Jacobian matrix is not exploited.

In the following Sections, a new sensitivity updating scheme is introduced that aims at overcoming the limitations described above.

III. CMoN-RTI

Several attempts to exploit the structure of the QP (3) are present in the literature. A Mixed-Level scheme has been proposed in \[25\], where only the first $N_c$ blocks in (3) are updated. However, choosing $N_c$ from heuristics may not be adequate for controlling highly nonlinear systems. Partial updating schemes where a fixed number of Jacobian blocks are updated have been independently proposed in \[20\] and \[22\] by using either CMoN or a so-called “norm criterion”. In \[21\], an inexact scheme has been proposed, where a varying number of sensitivities are updated, namely, only the most “nonlinear” ones. In this paper, the CMoN-RTI scheme is extended by introducing adjoint CMoN on dual variables and analyzing the QP problem by using parametric optimization theory. In particular, an advanced tuning strategy is developed, that automatically provides the number of Jacobian blocks to be updated while granting that the DoI remains below a user-defined tolerance. For the sake of clarity, in the following the CMoN-RTI scheme proposed in \[21\] is summarized.

A. Curvature-like Measure of Nonlinearity

Studies on Measures of the Nonlinearity (MoNs) of nonlinear dynamic systems can be traced back to the 1980s. The three main classes of MoNs are:

---

\[15\] All QP components in (3) can be evaluated at different rates but the constraints Jacobian only is here considered. Please refer to \[15\] for the complete ML-RTI scheme.
1) the distance between a nonlinear system and its best linear approximation [26];
2) the gap metric between two linear systems obtained by linearizing a nonlinear system around two different operating conditions [27];
3) the curvature MoN (CMoN) at a point in the parameter space along a given direction. [28], [29].

Global and off-line metrics are developed in [26], [27]. CMoN is a local metric originally introduced to measure the non-linearity in an estimation setting [28], [30], [31] and then progressively update the reference starting from the end of the prediction horizon [32]. This approach has the beneficial impact that the initial part of the predicted trajectory is not affected by the reference change. As a result, given a sufficiently high sampling frequency, sensitivity update is necessary only in the final part of the prediction horizon, whereas information from the past can be effectively used elsewhere.

IV. AN ADVANCED TUNING STRATEGY

In [11], thresholds $\eta_{pri}$ and $\eta_{dual}$ regulate the trade-off between the accuracy of the Jacobian approximation and the computational cost, by determining the updating subset with the largest CMoN values. An intuitive way of choosing the

The term $\Delta \lambda_{k+1}^{-1} \nabla \phi_k^i$ can be computed by efficient adjoint sensitivity schemes. As will be shown in Sec. IV, it together with (10) play important roles in controlling the accuracy of QP solutions.

At each sampling instant, the nonlinearity of a dynamic system over the entire prediction horizon can be estimated by applying (9) and (10) to each shooting interval.

B. Updating Logic

Due to the multiple shooting discretization, each block $\nabla \phi_k^i$ in (4), and the corresponding CMoN $\kappa_k^i$, uniquely depend on the initialization $u_k$. Hence, evaluation of CMoN, integration, and sensitivity generation can be performed independently at each shooting interval. Thus, the set of sensitivity blocks $\{\nabla \phi_k^i\}$ can be divided into two parts:

1) an updating subset where the sensitivity blocks are updated; and
2) the remaining subset where the sensitivity blocks are kept unchanged.

If the first subset is much smaller than the second one, a significant reduction of computational cost for sensitivity evaluations can be achieved. To this end, CMoN can be used to determine such an updating subset. Intuitively, when $\kappa_k^i$ is sufficiently small, the sensitivity $\nabla \phi_k^i$ is close enough to $\nabla \phi_k^{-1}$, hence sensitivity update is not necessary for the current sampling instant. The block $k$ of the Jacobian matrix $\nabla B$ is updated according to the following strategy. Set the values of thresholds $\eta_{pri}$ and $\eta_{dual}$, where the subscript $pri$ denotes the primal variable and $dual$ the dual variable. Then,

$$\nabla \phi_k^i = \begin{cases} \nabla \phi_k^i - 1, & \text{if } \kappa_k^i \leq \eta_{pri}, \& \tilde{\kappa}_k^i \leq \eta_{dual}, \\ \text{eval}(\nabla \phi_k^i), & \text{otherwise} \end{cases}$$

The proposed strategy is effective in both of the following cases:

1) Regulation: Given a sufficiently long prediction horizon, the system nonlinearity is typically excited in a small part of the predicted trajectory, that is, far from the steady state. As the system is approaching its steady state, less and less sensitivities are expected to be updated.

2) Reference tracking: Assuming that future reference is known in advance, a widely used choice is to progressively update the reference starting from the end of the prediction horizon [14]. This approach has the beneficial impact that the initial part of the predicted trajectory is not affected by the reference change. As a result, given a sufficiently high sampling frequency, sensitivity update is necessary only in the final part of the prediction horizon, whereas information from the past can be effectively used elsewhere.

In the paper, all vector and matrix norms are Euclidean.
A relation that reflects inaccuracy of the sensitivities Jacobian in (6) affects the accuracy of both primal and dual solutions. A perturbation parameter. Three Lemmas describing the stability of the QP solution w.r.t. to such parameter are then given.

\[ M(0) = \begin{bmatrix}
- \nabla_{\Delta w}L_{QP}(0) & \cdots & \cdots & \cdots & \cdots & \cdots \\
- \Delta \mu_{\text{pr}}L_{\Delta w} & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
- \nabla_{\Delta w}b_{n,0} & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} , \quad (12a) \]

\[ N(0) = \begin{bmatrix}
- \nabla_{\Delta w}L_{QP} & \cdots & \cdots & \cdots & \cdots & \cdots \\
- \Delta \mu_{\text{pr}} & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
- \nabla_{\Delta w}b_{n,0} & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} \top , \quad (12b) \]

\[ \mathcal{L}_{QP}(0) = \frac{1}{2} \Delta w \top H \Delta w + \nabla \mathcal{L} \Delta w + b \top (0) \Delta \lambda + c \top \Delta \mu . \quad (12c) \]

\[ s.t. b(p) = 0, \quad (13b) \]

\[ c \leq 0, \quad (13c) \]

where \( \nabla \mathcal{L} \) is the gradient of the Lagrangian of (6). \( b(p) = B(w) + (\nabla B + P) \Delta w \), \( P := \nabla B - \nabla B \) is the Jacobian approximation error, and \( \nabla B \) is the inexact Jacobian with partially updated blocks. The perturbation vector \( \mathbf{p} = \text{vec}(P) \in \mathbb{R}^{n_p} \) is the vectorization of \( P \) after eliminating zero elements.

According to Definition 1 the exact Jacobian QP problem (6) is referred to as \( \mathcal{Q}(0) \). Due to multiple shooting discretization, \( P \) has the following banded block structure

\[ P = \begin{bmatrix}
O_{n_x} & P_0 & O_{n_x} \\
P_1 & O_{n_x} & \cdots \\
& \ddots & \ddots \\
& & \ddots & P_{N-1} & O_{n_x} \\
\end{bmatrix} , \]

where \( O_{a} \) is a zero matrix of dimension \( a \) and \( P_b \in \mathbb{R}^{n_x \times (n_x + n_w)} \) is the \( k \)-th block of the Jacobian approximation, and in general is a dense matrix.

**Definition 2.** Define \( \Delta \mathbf{y}(p) = (\Delta \mathbf{w}^\top(p), \Delta \mu^\top(p), \Delta \lambda^\top(p))^\top \) the solution of (13), where \( \Delta \mathbf{w}(p), \Delta \mu(p), \Delta \lambda(p) \) are the increments of optimization variables, multipliers for inequality and equality constraints, respectively.

Observe that QP (13) has a modified objective gradient with respect to (6). However, it can be easily proved that these two formulations are equivalent (14). The additional computational cost can be neglected since both formulation (13) and (6) contain adjoint sensitivities in their objective. We adopt (13) as the standard form hereafter.

The following Lemma shows that the distance between the primal solutions of \( \mathcal{Q}(0) \) and \( \mathcal{Q}(p) \) is bounded, and the bound is of the same order of the Jacobian approximation error.

**Lemma 1.** (12). Let \( \Delta \mathbf{w}(0) \) and \( \Delta \mathbf{w}(p) \) minimize \( \mathcal{Q}(0) \) and \( \mathcal{Q}(p) \) over corresponding feasible sets, respectively. Then there exists constants \( c \) and \( c^* > 0 \) such that

\[ \| \Delta \mathbf{w}(p) - \Delta \mathbf{w}(0) \| \leq c \epsilon \text{ whenever } c \leq c^* \text{ and } \epsilon = \| \nabla G - \nabla G \| = ||P||. \]

The following Lemma shows that the solution \( \Delta \mathbf{y}(p) \) is a unique minimizer of (13). Moreover, the active set is locally stable.
Lemma 2. [36] Under the assumption on differentiability, second-order sufficient conditions, constraints linear independence and the strict complementary slackness condition, there exists a unique solution $\Delta y(p)$, which is continuously differentiable w.r.t. $p$ for $p \in$ a neighborhood of 0. Moreover, the set of active inequality constraints is unchanged, strict complementary slackness holds, and the active constraint gradients are linearly independent at $\Delta w(p)$.

Finally, the following Lemma provides a linearly approximated relationship between the exact and inexact solutions.

Lemma 3. [36] A first order approximation of $\Delta y(p)$ in a neighborhood of $p = 0$ is given by

$$\Delta y(p) = \Delta y(0) + M^{-1}(0)p + O(\|p\|^2)$$

where $M, N$ are given in (12), and $b_k$ and $c_k$ are the $k$-th row of $b(p)$ and $c$, respectively.

Remark 1. Lemma [3] is a sufficient but not necessary condition for the results it holds. It is either not a necessary condition for Lemma 2. Modern studies based on perturbation theory show that the solution manifold $\Delta y(p)$ is nonsmooth but continuous. Therefore, $\Delta y(p)$ can be close enough to $\Delta y(0)$ even in the presence of active-set changes. The reader is referred to [37], [38] and references therein for more details.

B. First Order Error Analysis

In the neighborhood of $p = 0$, (3) can be rewritten as

$$\Delta y(0) = \Delta y(p) - M^{-1}(p)N(p)p$$

As shown in Appendix A, it holds that

$$N(p)p = \begin{bmatrix} P^T \Delta \lambda(p) \\ Q \end{bmatrix}$$

and, by pre-multiplying (14) by $M(p)$, it follows that

$$M(p)\Delta y(0) = M(p)\Delta y(p) + N(p)p.$$ Therefore, the DtO at the sampling instant $i$ satisfies

$$\|e_i\|^2 := \|\Delta y(0_i) - \Delta y(p_i)\|^2$$

$$\leq \|M^{-1}(p_i)\|^2 (\|P^{\alpha} \Delta \lambda(p_i)\|^2 + \|P^i \Delta w(p_i)\|^2).$$

Note that, given a finite dimensional and non-singular real matrix $M(p)$, its Euclidean norm

$$\rho_i := \|M^{-1}(p_i)\|$$

is bounded. Hence, the DtO $\|e_i\|$ is bounded only if $\|P^{\alpha} \Delta \lambda(p_i)\|$ and $\|P^i \Delta w(p_i)\|$ are bounded. The two bounds are referred as the dual bound and primal bound respectively, and are discussed in the following.

1) Primal Bound: By using the primal threshold $\eta_{pri}$ in the updating logic (11), one obtains

$$\|P^i q^{i-1}\| \leq 2\eta_{pri} \|V^{i-1}\|,$$

where $q^{i-1} = [q_0^{i-1} \ldots q_N^{i-1}]^T$ and $V^{i-1}$ is a vector of directional sensitivities given by

$$V^{i-1} = [(\nabla \phi_0^{i-1} q_0^{i-1})^T \ldots (\nabla \phi_N^{i-1} q_N^{i-1})^T]^T.$$ Derivation details are presented in Appendix B. Moreover, there exists a $\alpha > 0 \in \mathbb{R}$ such that

$$\|P^i \Delta w(p_i)\| = \alpha \|P^i q^{i-1}\|$$

Hence, a bound in the direction of the primal variable is as follows

$$\|P^i \Delta w(p_i)\|^2 \leq 4\alpha^2 \eta_{pri}^2 \|V^{i-1}\|^2.$$ 2) Dual Bound: Similarly, for adjoint CMoN it holds that

$$\|\Delta \lambda^T(p^{i-1})P_i\| \leq \eta_{dual} \|V_{dual}^i\|.$$ where $V_{dual}^i = [\lambda_1^{i-1} \nabla \phi_0^{i-1} \ldots \lambda_N^{i-1} \nabla \phi_N^{i-1}]^T$.

There exists a $\beta > 0 \in \mathbb{R}$ such that

$$\|P^i \Delta \lambda^T(p^{i-1})P_i\| \leq \beta \|\Delta \lambda^T(p^{i-1})P_i\|.$$ Hence, a bound in the direction of the dual variable is obtained as follows

$$\|P^i \Delta \lambda^T(p^{i-1})P_i\|^2 \leq \beta^2 \eta_{dual}^2 \|V_{dual}^i\|^2.$$ 

C. Thresholds Estimation

Given a DtO tolerance $\epsilon_i$ at the sampling instant $i$, let

$$\beta \eta_{dual}^2 \|V_{dual}^i\|^2 \leq (1 - c_1)\epsilon_i^2 / \rho_i^2,$$

$$4\alpha^2 \eta_{pri}^2 \|V^{i-1}\|^2 \leq c_1 \epsilon_i^2 / \rho_i^2$$

where $0 < c_1 < 1$ is a tuning parameter that trades off impact of the primal and dual bounds on the DtO. By substituting (20) into (15), one obtains $\|e_i\|^2 \leq \|\epsilon_i\|^2$. Therefore, the primal and dual thresholds satisfy the following inequalities:

$$0 \leq \eta_{pri} \leq \frac{\sqrt{c_1} \epsilon_i}{2\alpha \rho_i \|V^{i-1}\|} := \mathcal{U}_1$$

$$0 \leq \eta_{dual} \leq \frac{\sqrt{1 - c_1} \epsilon_i}{\beta \rho_i \|V^{i-1}\|} := \mathcal{U}_2.$$

Theorem 1. $\mathcal{U}_1, \mathcal{U}_2 : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ are piecewise discontinuous functions of $(\eta_{pri}, \eta_{dual})$ and their ranges are finite sets.

The proof of Theorem 1 is given in Appendix C. Given Theorem 1, a formal solution to find the maximal $(\eta_{pri}, \eta_{dual})$ is then to solve the following problem

$$\max_{\eta_{pri}, \eta_{dual}} \eta_{pri}, \eta_{dual}$$

s.t. $\eta_{pri} - \mathcal{U}_1(\eta_{pri}, \eta_{dual}) \leq 0,$

$$\eta_{dual} - \mathcal{U}_2(\eta_{pri}, \eta_{dual}) \leq 0.$$ The solution of problem (22) provides the maximal values of the thresholds, corresponding to the minimum number of sensitivity updates while guaranteeing a bounded DtO. Note that for a given $\epsilon_i \geq 0$, there always exists at least one feasible solution to (22), i.e. $(\eta_{pri}, \eta_{dual}) = 0$, that makes CMoN-RTI coincide with the standard RTI scheme.
\[ \gamma^i = \sigma(M^{-1}(p^i)) + 1, \]

where \( \sigma(M^{-1}(p^i)) \) is the standard deviation operation and \( \sigma(M^{-1}(p^i)) \) is the set of singular values of \( M^{-1}(p^i) \). To make on-line computation feasible, \( \gamma^0 \) can be used, as it can be computed off-line. Effectiveness of this choice is discussed in Section V.

Finally, the approximated thresholds estimates are given by

\[
\begin{align*}
\eta^i_{\text{pri}} &= \frac{\gamma^0}{\sqrt{\eta^i_{\text{pri}}}} + \epsilon^i_{\text{pri}} \\
\eta^i_{\text{dual}} &= \frac{\gamma^0}{\sqrt{\eta^i_{\text{dual}}}} + \epsilon^i_{\text{dual}}.
\end{align*}
\]

A summary of the practical implementation of CMoN-RTI is given in Algorithm 2.

**Algorithm 2 A practical implementation of CMoN-RTI**

1. Choose an initial point \((w^0, \lambda^0, \mu^0)\)
2. Choose \(0 < c_1 < 1\)
3. Compute \(\rho^0\) by (16)
4. Compute \(\gamma^0\) by (24)
5. Set \(q_k \leftarrow 0, \phi_k \leftarrow 0, \nabla \phi_{k-1} \leftarrow 0, w_{k-1} \leftarrow 0\) for all \(k\)
6. for \(i = 0, 1, \ldots\) do
7. Compute \(\nabla L^i, H^i, B^i, C^i, \nabla C^i\)
8. for \(k = 0, 1, \ldots, N - 1\) do
9. Perform integration and obtain \(\phi_k\)
10. Choose the DtO tolerance \(\epsilon^i\) by (27)
11. Compute \(\kappa^i_k, \tilde{\kappa}^i_k\) by (9) and (10)
12. Update \(\nabla \phi_k\) by (11)
13. end for
14. Solve QP (6) and obtain \((\Delta w^{QP}, \Delta \lambda^{QP}, \Delta \mu^{QP})\)
15. Update the initialization by \(w^{i+1} = w^i + \Delta w^{QP}, \lambda^{i+1} = \lambda^i + \Delta \lambda^{QP}, \mu^{i+1} = \mu^i + \Delta \mu^{QP}\)
16. Compute \((\eta^i_{\text{pri}}, \epsilon^i_{\text{dual}})\) by (25)
17. end for

**V. NMPC SIMULATION CASE STUDY**

In this section, Algorithm 2 is applied to two examples, namely, the control of an inverted pendulum and of a chain of masses. Numerical integration and sensitivity generation are performed by a 4th order explicit Runge-Kutta integrator with 4 steps per shooting interval, provided by the CasADi toolbox [39] using automatic differentiation. The QP problem is solved by using HPIPM, a structure-exploiting interior point solver based on hardware tailored linear algebra libraries [40]. Algorithmic parameters are chosen as described in Section IV.D for all examples. The computing environment is Ubuntu 16.04 on a PC with Intel core i7-4790 running at 3.60GHz, and the implementation is coded in plain C with -O2 compilation optimization flag.

**A. Inverted Pendulum**

An inverted pendulum is mounted on top of a cart and can roll up to 360 degrees. The dynamic model is given by

\[
\ddot{\theta} = \frac{-m_1 l \sin(\theta) \dot{\theta}^2 + m_1 g \cos(\theta) \sin(\theta) + F}{m_2 + m_1 - m_1 \cos(\theta)^2} - \frac{1}{l(m_2 + m_1 - m_1 \cos(\theta)^2)}(F \cos(\theta) - m_1 l \cos(\theta) \sin(\theta) \dot{\theta}^2 + (m_2 + m_1) g \sin(\theta)),
\]

\[
\dot{\theta} = \frac{-m_1 l \sin(\theta) \dot{\theta}^2 + m_1 g \cos(\theta) \sin(\theta) + F}{m_2 + m_1 - m_1 \cos(\theta)^2} - \frac{1}{l(m_2 + m_1 - m_1 \cos(\theta)^2)}(F \cos(\theta) - m_1 l \cos(\theta) \sin(\theta) \dot{\theta}^2 + (m_2 + m_1) g \sin(\theta)),
\]

\[
\dot{\theta} = \frac{-m_1 l \sin(\theta) \dot{\theta}^2 + m_1 g \cos(\theta) \sin(\theta) + F}{m_2 + m_1 - m_1 \cos(\theta)^2} - \frac{1}{l(m_2 + m_1 - m_1 \cos(\theta)^2)}(F \cos(\theta) - m_1 l \cos(\theta) \sin(\theta) \dot{\theta}^2 + (m_2 + m_1) g \sin(\theta)),
\]

where \( l \) is the length of the pendulum, \( m_1 \) is the mass of the pendulum, \( m_2 \) is the mass of the cart, \( g \) is the acceleration due to gravity, and \( \theta \) is the angle of the pendulum from the vertical. The control input \( F \) is used to keep the pendulum upright. The problem is formulated as a NMPC problem with\( l = 1 \text{m}, m_1 = 1 \text{kg}, m_2 = 1.5 \text{kg}, g = 9.8 \text{m/s}^2\), initial state \( \theta(0) = \frac{\pi}{2} \text{ rad}, \dot{\theta}(0) = 0 \text{ rad/s} \), and horizon \( N = 100 \).

The control objective is to keep the pendulum upright for as long as possible. The penalty function is defined as

\[
J = \sum_{k=0}^{N-1} \left( \ddot{\theta}^2 + \dot{\theta}^2 \right) + \sum_{k=1}^{N} \left( \ddot{\theta}^2 \right) + \sum_{k=0}^{N-1} \left( \ddot{\theta}^2 \right) + \sum_{k=0}^{N-1} \left( \ddot{\theta}^2 \right)
\]

subject to

\[
\dot{\theta}(0) = 0, \quad \ddot{\theta}(0) = 0, \quad \theta(N) = 0, \quad \ddot{\theta}(N) = 0.
\]
where \( p, \theta \) are the cart position and swinging angle, respectively, and \( F \) is the control force acting on the cart. The model and values of parameters \( m_1, m_2, l, g \) are taken from [41]. For this example, a time-varying reference is given to the inverted pendulum to track different horizontal displacements and swing angles. A perfect initialization is chosen by optimally solving the OCP for \( t = 0 \) off-line. A short (\( N = 40 \)) and a long (\( N = 120 \)) prediction horizon are applied with a control interval \( T_s = 0.05 \) s. The tolerance on DtO in CMoN-RTI follows the rule given by

\[
\epsilon^i = \epsilon^{abs} \sqrt{n} + \epsilon^{rel} \| \Delta y^i \|,
\]

where \((\epsilon^{abs}, \epsilon^{rel})\) are the absolute and relative tolerances, \( n \) is the number of optimization variables, and \( y = (x, \lambda, \mu) \) is the optimal triple. Such choice ensures that the DtO tolerance scales with the size of the problem and the scale of the variable values [42]. For this problem, we set

\[
\epsilon^{abs} = 10^{-1}, \epsilon^{rel} = 10^{-1}.
\]

In Figure 2, the percentage of exactly computed sensitivities per sampling instant is given. The percentage starts from 0% when \( N = 40 \) since there is no reference change within the prediction horizon in the first 3 \( s \). CMoN-RTI is able to adapt to reference changes, as can be seen from the peaks at around \( t = 3, 8, 13, 18 \) s.

In Figure 1, the closed-loop state and control trajectories generated by the standard RTI and CMoN-RTI with the two prediction horizons are shown. The control performance of CMoN-RTI is indistinguishable from that of the standard RTI scheme, which demonstrates that CMoN-RTI is able to maintain the closed-loop performance as the standard RTI while using much less sensitivity computations.

In Figure 3, the percentage of exactly computed sensitivities per sampling instant to compute the DtO. In both cases, the DtO is lower than the tolerance.

To examine the effectiveness of using \( \rho^0 \) and \( \gamma^0 \) in (25), the relative difference is defined as

\[
r := \frac{\gamma^0 \rho - \gamma^0 \rho^0}{\rho^0}.
\]

For the inverted pendulum example, it is observed that the maximum value of \( r \) is 24\%, i.e. a sufficiently small difference.
B. Chain of Masses with Nonlinear Springs

A chain of masses is a system with \( n \) masses connected by springs on a chain \([16]\). The dynamic model is given by

\[
\begin{align*}
\dot{p}_i(t) &= v_i(t), \quad i = 1, \ldots, n-1, \\
\dot{v}_i(t) &= \frac{1}{m}(F_{i+1}(t) - F_i(t) - g), \\
\dot{p}_n(t) &= u(t),
\end{align*}
\]

where \( p_i(t) \in \mathbb{R}^3 \) and \( v_i(t) \in \mathbb{R}^3 \) are the positions and velocities of the \( i \)-th mass, respectively, and

\[
F_i(t) = D(x_i(t) - x_{i-1}(t))(1 - \frac{L}{\|x_i(t) - x_{i-1}(t)\|_2}) + F_{NL},
\]

is the spring force from mass \( i \) to \( i + 1 \) and \( F_{NL} \) is its nonlinear component. The velocities of the free mass \( \dot{p}_n(t) \) are controlled by \( u(t) \). As demonstrated in \([16], [18]\), ADJ-RTI is able to stabilize the chain of masses if \( F_{NL} = 0 \), i.e. the chain is connected by linear springs. In this paper, nonlinear springs \([43]\) are considered with

\[
F_{NL} = D_3(x_i(t) - x_{i-1}(t))\left(\frac{\|x_i(t) - x_{i-1}(t)\|_2 - L}{\|x_i(t) - x_{i-1}(t)\|_2}\right)^3.
\]

A total of 50 simulations are performed while using the standard, ML-, ADJ- and CMoN-RTI, with randomly assigned initial positions and velocities of the masses, see e.g. Fig. 4 for the positions and control trajectories generated in one of the simulations. For ML-RTI, the entire constraint Jacobian matrix is updated every \( m = 2 \) sampling instants; For ADJ-RTI, the Jacobian matrix is computed off-line at the steady state trajectory; For CMoN-RTI, the DoT tolerance is chosen as in \([28]\). To ensure that an accurate representation of the system is always used in the controller, at least 10\% sensitivities are updated at each sampling instant. These sensitivities are those having the largest values of CMoN, hence exhibiting the most significant nonlinearities \([20]\).

Control performance, numerical robustness, and efficiency of CMoN-RTI are evaluated and compared with standard RTI, ML-RTI, and ADJ-RTI. Firstly we collect statistics of the stabilizing time \( t_{st} \), defined as

\[
t_{st} = \arg \min_t t \quad \text{s.t.} \|u(t)\|_\infty < 0.1, \forall t \geq t,
\]

from the 50 simulations. In Fig. 5 the statistics of the stabilizing time \( t_{st} \) for the positions and control trajectories generated in one of the simulations. For ML-RTI, the entire constraint Jacobian matrix is updated every \( m = 2 \) sampling instants; For ADJ-RTI, the Jacobian matrix is computed off-line at the steady state trajectory; For CMoN-RTI, the DoT tolerance is chosen as in \([28]\). To ensure that an accurate representation of the system is always used in the controller, at least 10\% sensitivities are updated at each sampling instant. These sensitivities are those having the largest values of CMoN, hence exhibiting the most significant nonlinearities \([20]\).
The average KKT value at every sampling instant of the successfully stabilized cases among 50 simulations using RTI, ML- and CMoN-RTI. The KKT value is computed as the norm of the Lagrangian of the NLP as an indicator of optimality.

Fig. 6. The average KKT value at every sampling instant of the successfully stabilized cases among 50 simulations using RTI, ML- and CMoN-RTI. The KKT value is computed as the norm of the Lagrangian of the NLP as an indicator of optimality.

Table I: The average and maximal computational time per sampling instant in milliseconds [ms] for CMoN-RTI and the standard RTI scheme for the chain of masses with prediction length \(N = 40, 80, 160\). Sens. stands for sensitivity evaluation time and \(QP\) is the QP solving time.

| \(N\) | CMoN-RTI | RTI | CMoN-RTI | RTI |
|------|----------|-----|----------|-----|
|      | Average  | Maximal | Speedup factor |
|      | Total | Sens. | QP. | Total | Sens. | QP. | Total | Sens. | QP. |         |
| 40   | 17.9   | 7.5   | 8.8   | 23.5   | 14.1   | 7.9   | 37.1   | 19.0   | 16.1   | 40.6   | 24.2   | 14.7   | 9.4%    |
| 80   | 29.7   | 9.6   | 17.8  | 43.5   | 24.4   | 16.2  | 61.4   | 25.5   | 34.4   | 70.1   | 37.4   | 31.3   | 14.2%   |
| 160  | 66.8   | 13.8  | 46.3  | 93.8   | 45.4   | 43.3  | 134.7  | 30.8   | 98.7   | 144.8  | 52.4   | 94.0   | 7.5%    |

Fig. 7. The average percentage of exactly updated sensitivities at every sampling instant of the stabilized cases among 50 simulations for chain of masses using CMoN-RTI. At least 10% of sensitivities are updated at each sampling instant.

The mean and interquartile range (IQR) of \(t_{st}\) of CMoN-RTI is very close to those of the standard RTI. This means that CMoN-RTI has a similar control performance to the standard RTI in most of the situations. On the other hand, ML-RTI has a similar stabilizing time to RTI when \(N\) is short, whereas \(t_{st}\) grows significantly as \(N\) becomes larger. ADJ-RTI, initialized at the steady state trajectory, is not able to provide acceptable control performance, especially when \(N\) is large.

The control performance is also evaluated by assessing the optimality of each controller. In Fig. 6 the average Karush-Kuhn-Tucker (KKT) value, i.e. the norm of the gradient of the Lagrangian of the NLP as an indicator of optimality. It can be observed that
- ML-RTI KKT values exhibit strongly oscillatory behavior since the Jacobian update is performed every \(m = 2\) sampling instants only.
- As the system converges “on the fly”, the KKT of CMoN-RTI decreases smoothly as that of the standard RTI.

Table II: The number of simulations (among 50) where each controller cannot stabilize the chain.

| \(N\) | ML-RTI | ADJ-RTI | CMoN-RTI |
|------|--------|---------|----------|
| 40   | 4      | 7       | 0        |
| 80   | 5      | 42      | 0        |
| 160  | 7      | 50      | 0        |

As for numerical robustness, the number of simulations where each controller fails to stabilize the system within 50s is reported in Table II. Given that the initial condition of each simulation is randomly assigned, the numerical robustness or the sensitivity w.r.t. initialization of each controller can be assessed. CMoN-RTI is able to stabilize the chain within 50s in all situations, although the maximal stabilizing time is larger than that of the standard RTI (see Fig. 5). ML-RTI has a few failed cases if \(N = 40\) and this number increases as the prediction horizon grows. Not surprisingly, ADJ-RTI exhibits the poorest robustness properties as it heavily depends on the quality of the off-line Jacobian matrix.

To evaluate computational efficiency, the average percentage of exactly updated sensitivities using CMoN-RTI at every sampling instant is reported in Figure 7. In the first 20s, the KKT values of CMoN-RTI and RTI are almost identical, however, the number of updated sensitivities is at most 80% and it reduces to 60% when \(N\) becomes larger. After \(t = 30s\),
when the system is close to its steady state, updating only 10\% of the blocks allows to still maintain small KKT values.

Table I shows the average and maximal computational time of CMoN-RTI and the RTI scheme per sampling instant.

For this example, the speedup factor is computed by using the maximal computational time. With different prediction horizons, the computational time for evaluating sensitivities varies from about 60\% to 36\% of the full RTI step, and the one for the QP varies from about 36\% to 65\%. As a result, the speed up factor is strongly related to the distribution of computational time among the critical steps of the full RTI. In the specific example, the speedup factor is always positive with a maximum value of 14.2\% when \( N = 80 \). Also, observe that the computational performance obtained in the examples is related to the use of the simple explicit Runge-Kutta integrator.

For systems that require the use of more complex integrators, whose sensitivities are more computational expensive, CMoN-RTI is expected to achieve a greater speedup factor.

VI. CONVERGENCE ANALYSIS

Algorithm 2 is a partial sensitivity updating scheme in the framework of RTI between two consecutive sampling instants. It can also be straightforwardly extended to the SQP framework, where a sequence of QP problems is solved until convergence is achieved. The resulting algorithm, denoted hereafter as CMoN-SQP, partially updates sensitivities between two consecutive SQP iterations. In the SQP scenario, the Two-side-Rank-One (TR1) updating SQP algorithm has been proposed in [41] for equality constrained problems. Similar to the famous Symmetric-Rank-One (SR1) updating scheme [8], the TR1 scheme requires Hessian and Jacobian updates to satisfy both direct and adjoint secant conditions. This method is extended to linearly inequality constrained problems in [34] and its local convergence is proved.

Differently from the TR1 scheme, which adopts a rank one Jacobian matrix update, CMoN-SQP achieves a block update by exploiting the structure of the problem. In addition, the primal and dual bounds are satisfied, instead of enforcing secant conditions. In the following, local convergence of CMoN-SQP is proved and it is shown that the convergence rate is tunable via the choice of the DiO tolerance.

A. Local Convergence of CMoN-SQP

Consider the parametric QP problem (13). Solving problem (13) in a SQP algorithm is equivalent to solving the following nonlinear system:

\[
F(y) = 0, y := \begin{bmatrix} w \\ \lambda \end{bmatrix}, F(y) = \begin{bmatrix} R^T \nabla C_\lambda(w, \lambda) \\ B(w) \\ C_a(w) \end{bmatrix},
\]

where \( \lambda \) denotes the multiplier for both equality and active inequality constraints, \( C_a \) contains the active constraints, and \( R \) is a matrix with orthonormal column vectors, such that \( \nabla C_a R = 0 \) [34]. The Jacobian matrix of the nonlinear system is

\[
\nabla F(y^i) = \frac{\partial F}{\partial y}(y^i) = \begin{bmatrix} R_i^T H_i \\ R_i^T \nabla B^T(w^i) \\ \nabla C_a(w^i) \end{bmatrix},
\]

where \( H^i \) is an approximation of the exact Hessian, i.e. the Gauss-Newton approximation which is independent of the multiplier \( \lambda \). Let \( J_i \) be an approximation of the exact Jacobian

\[
J_i = \begin{bmatrix} R_i^T H_i & R_i^T \nabla B^T(w^i) \\ \nabla C_a(w^i) \end{bmatrix}.
\]

The following theorem indicates that the proposed scheme is convergent in the neighborhood of \( p = 0 \).

Theorem 3. Let \( F : \mathbb{V} \rightarrow \mathbb{R}^n, \mathbb{V} \subset \mathbb{R}^n \) be continuously differentiate. Consider the two sequences

\[
\{y_s\} : y_t^{i+1} = y_s + \Delta y_s^i,
\]

\[
\{y_p\} : y_t^{i+1} = y_p + \Delta y_p^i,
\]

where

\[
\Delta y_s^i = -\nabla F^{-1}(y_t^i) F(y_t^i),
\]

\[
\Delta y_p^i = -J^{-1}(y_t^i) F(y_t^i)
\]

Assume that

1) the Jacobian matrix is invertible, uniformly bounded, and has uniformly bounded inverses,

2) there exists a \( \kappa_0 < 1 \) such that \( \|\Delta y_s^{i+1}\| \leq \kappa_0 \|\Delta y_s^i\| \)

for all \( i > m_1, m_1 \in \mathbb{N} \). Hence, starting from \( y_0 \in \mathbb{V} \), the sequence \( \{y_s\} \) converges to a local optimizer \( y_s^* \).

3) \( J(y_t^i) \) is generated by Algorithm 2.

Then,

1) there always exists a set of scalars \( \{i \in \mathbb{N} | e^i \geq 0\} \)

such that the distance between the sequences \( \{y_p\} \) and \( \{y_s\} \) is sufficiently small at each iteration,

2) there always exists a set of scalars \( \{i \in \mathbb{N} | e^i \geq 0\} \)

and a \( \kappa_2 \) satisfying \( \kappa_0 \leq \kappa_2 < 1 \) such that \( \|\Delta y_p^{i+1}\| \leq \kappa_2 \|\Delta y_p^i\| \)

for all \( i > m_2, m_2 \in \mathbb{N} \), and the sequence \( \{y_p\} \) converges to \( y_p^* = y_s^* \) starting from \( y_0^* \).

Proof: Let the locally exact solution initialized at \( y_p^i \) be

\[
\Delta y_p^i = -\nabla F^{-1}(y_t^i) F(y_t^i).
\]

Assume that at iteration \( i \), the DiO is satisfied as

\[
\|\Delta y_p^i - \Delta y_0^i\| = \|e^i\| \leq \epsilon^i.
\]

Let \( \|d_t^i\| = \|y_p^i - y_s^i\| \) be the distance between the two sequences at the current iteration. Observe that

\[
\nabla F(y_t^i) = \nabla F(y_s^i) + d_t^i \nabla^2 F(y_s^i) + O(\|d_t^i\|^2),
\]

\[
F(y_p^i) = F(y_s^i) + \nabla F(y_s^i) d_y^i + O(\|d_y^i\|^2).
\]

Assume that \( \|d_t^i\| \) is sufficiently small and \( O(\|d_t^i\|^2) \) can be neglected, then by combining (30) and (31), it follows that

\[
\Delta y_t^i - \Delta y_s^i = -\nabla F^{-1}(y_s^i)(d_t^i \nabla^2 F(y_s^i) \Delta y_0^i - d_y^i).
\]

As a result,

\[
\|\Delta y_t^i - \Delta y_s^i\| \leq g^i \|d_t^i\| ,
\]

where \( g^i \) is a constant and \( \|d_t^i\| \) is the distance between the two sequences at the current iteration.
where \( g' = ||\nabla F^{-1}(y'_p)(\Delta y_p^T \nabla^2 F(y'_p)) + \mathcal{I}|| \). The distance between the two solutions at the current iteration is

\[
\|\Delta y^i_p - \Delta y^i_s\| \leq \|\Delta y^i_p - \Delta y^i_0\| + \|\Delta y^i_0 - \Delta y^i_s\| \\
\leq \varepsilon^i + g'^i \|\Delta y^i_p\|,
\]

and the distance between the two sequences at the next iteration is

\[
\|d^i_{y,p}\| = \|y^i_{p} - y^i_s\| \\
\leq \|d^i_{y,p}\| + \|\Delta y^i_p - \Delta y^i_s\| \\
\leq \varepsilon^i + (1 + g^i)\|\Delta y^i_p\|.
\]

Since Algorithm 2 always starts from \(\|d^0_{y,p}\| = 0\), \(\|d^i_{y,p}\|, \forall i > 0\) is a linear combination of \((\varepsilon^0, \varepsilon^1, \ldots, \varepsilon^i)\). Therefore, it is always possible to choose a set of scalars \(\{i \in \mathbb{N}^+ | \varepsilon^i \geq 0\}\), such that \(\|d^i_{y,p}\| \approx 0\). Equivalently, the sequence \(\{y_p\}\) can be sufficiently close to \(\{y_s\}\) at every iteration.

Consider now the convergence properties of \(\{y_p\}\). By assumption 2 it follows that

\[
\|y^i_{p} - y^i_s\| \leq \|d^i_{y,p}\| + \kappa_0\|\Delta y^i_s\| \\
\leq \|d^i_{y,p}\| + \kappa_0\|d^i_{y,p}\| + \kappa_0\|\Delta y^i_s\| \\
= \kappa_1 + \kappa_0\|\Delta y^i_s\|
\]

where \(\kappa_1 = \|d^i_{y,p}\| + \kappa_0\|d^i_{y,p}\|\). Since \(\kappa_0 < 1\) and \(\|d^i_{y,p}\|, \|d^i_{y,s}\|\) can be arbitrarily small, there exists a \(\kappa_2\) satisfying \(\kappa_0 \leq \kappa_2 < 1\) such that

\[
\|y^i_{p} - y^i_s\| \leq \kappa_1 + \kappa_0\|\Delta y^i_s\| \leq \kappa_2\|\Delta y^i_s\|.
\]

Therefore, the sequence \(\{y_p\}\) is convergent and its convergence rate is at most identical to that of \(\{y_s\}\). As proved in [15], [24], when \(\{y_p\}\) does converge, it converges to the exact limit \(y^*\) of the sequence \(\{y_s\}\).

Theorem 3 shows that the Jacobian approximation error can be controlled by using user-defined DIO tolerances, hence the convergence can be satisfied by using appropriate tuning configurations. The convergence rate is also shown to be tunable, which increases the flexibility of the proposed algorithm. If \(\varepsilon^i = 0, \forall i \geq 0\), CMoN-SQP becomes the standard SQP algorithm with the same convergence rate.

### B. Numerical Examples

As an example, the CMoN-SQP scheme is applied to the inverted pendulum [26]. The control objective is to invert the pendulum from bottom to top. CMoN-SQP is used to solve the OCP in open-loop at time \(t = 0\) with \(N = 40\). Since only local convergence is of interest, the initialization of the OCP is in a neighborhood of the optimal solution and a full Newton-step is adopted at each iteration.

Figure 8 shows the convergence behavior of two different DIO choices of CMoN-SQP. The left y-axis reports the KKT value, that indicates the optimality of the solution. The right y-axis reports the percentage of sensitivities being updated at each iteration. To show how the choice of DIO affect the convergence rate, the following DIO tolerances are used:

\[
s_1 : (e^{abs} = 10^{-2}, e^{rel} = 10^{-2}) \\
s_2 : (e^{abs} = 10^{-1}, e^{rel} = 10^{-1})
\]

A more aggressive setting \((s_2)\) leads to less sensitivity evaluations but slower convergence rate. Figure 9 shows the convergence behavior of three choices of \(c_1\) for [25]. The convergence rate is not sensitive to the values of \(c_1\). In practice, to solve a structured NLP problem by using CMoN-SQP, one would achieve a satisfactory trade-off between the cost of sensitivities and the number of iterations by properly tuning the DIO tolerance.

### VII. Conclusion

In this paper, the partial sensitivity updating scheme CMoN-RTI of [21] is extended by proposing an advanced tuning strategy with solution accuracy control and convergence analysis. In CMoN-RTI, sensitivities are updated based on CMoN of the dynamic system over the prediction horizon. The CMoN works as a metric to evaluate the quality of sensitivity approximation,
and an updating logic is designed with the use of primal and dual thresholds. By using parametric optimization theory, an advanced strategy for tuning such thresholds is proposed. Such strategy automatically determines the number of updated sensitivities and guarantees the Dto of QP solutions under a user-defined tolerance.

Closed-loop simulations show that the CMoN-RTI scheme exhibits good control performance when applied to highly nonlinear control problems. The application to an inverted pendulum shows that CMoN-RTI can adapt to reference changes while satisfying the Dto tolerance. The results from a chain of masses with nonlinear springs demonstrate the superior control performance, numerical robustness, and efficiency of CMoN-RTI.

The proposed scheme has also been extended to full SQP algorithms, denoted as CMoN-SQP and its local convergence properties, numerical robustness, and efficiency of CMoN. Future studies may focus on possible extensions of CMoN-RTI. While ADJ-RTI and Mixed-Level schemes can benefit from condensing steps with significantly reduced computational efforts, CMoN-RTI usually requires to perform a full condensing step at every sampling instant. Further improvements can be achieved by adopting partial condensing methods.

APPENDIX A

In this Appendix, the computation of \( M, N \) in (12a) and (12b) is detailed. For elements in \( M \), it holds

\[
\nabla^2 \mathcal{L}_{QP} = H, \\
\nabla \phi_k = \nabla C_k, k = 1, \ldots, n_t, \\
\n\nabla b_j(p) = \nabla B_j + \hat{P}_j, j = 1, \ldots, n_E,
\]

where \( \hat{P}_j \) is the \( j \)th row of \( P \). For elements in \( N \), it holds

\[
\nabla^2 \Delta w \mathcal{L} = \begin{bmatrix}
O_1 & 0 & \cdots & 0 \\
0 & \ddots & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & O_{Nn_x+1}
\end{bmatrix}, \\
\n\nabla \Delta w_j = \begin{bmatrix}
\Delta w_{j1} \\
\vdots \\
\Delta w_{jNn_x+1}
\end{bmatrix}, \quad j = 0, \ldots, n_f,
\]

where \( \Delta w_j \) is the \( j \)th column of \( \Delta w \).

APPENDIX B

In this Appendix, the expression for \( V_{pri}^{i-1} \) in (17) is derived. From the updating logic (11), it can be easily obtained that

\[
\|P_k q_k^{i-1}\| = \left\| \begin{bmatrix} O & P_0 \\ P_1 & \ddots \\ \vdots & \ddots & \ddots \\ P_{N-1} & 0 & O \end{bmatrix} \begin{bmatrix} q_{k0}^{i-1} \\ \vdots \\ q_{kN}^{i-1} \end{bmatrix} \right\| = \sum_{k=0}^{N-1} \|P_k q_k^{i-1}\|^2
\]

For the full Jacobian matrix, it holds

\[
\|P^i q^{i-1}\| = \left\| \begin{bmatrix} O & P_0 \\ P_1 & \ddots \\ \vdots & \ddots & \ddots \\ P_{N-1} & 0 & O \end{bmatrix} \begin{bmatrix} q_{01}^{i-1} \\ \vdots \\ q_{N1}^{i-1} \end{bmatrix} \right\| = \sum_{k=0}^{N-1} \|P_k q_k^{i-1}\|^2
\]

A similar derivation can be conducted for \( V_{pri}^{i-1} \). The details are hence omitted.

APPENDIX C

A proof of Theorem 1 is given in this Appendix.

**Proof:** According to (11), given the primal and dual thresholds, \( \nabla \phi_j \) are updated for

\[
j \in \{ j | \kappa_j^i > \eta_{pri}, \bar{\kappa}_j^i > \eta_{dual}; j = 0, 1, \ldots, N - 1 \}.
\]

It follows that sensitivities with larger CMoN values always get updated first. As a consequence, there are finite number of combinations (actually \( N + 1 \)) of possible updated sensitivities, resulting in a finite number of possible \( P^i \) matrices. Therefore, \( \rho^i, \alpha^i, \beta^i \) which are defined in (16), (18) and (19), are functions of \( P^i \) and have at most \( N + 1 \) possible values. Hence, the range of \( U_1, U_2 \) are finite.

In addition, for some \( k \neq j \), \( \forall (\eta_{pri}^i, \eta_{dual}^i) \) satisfying

\[
\kappa_k^i < \eta_{pri}^i < \kappa_j^i, \\
\bar{\kappa}_k^i < \eta_{dual}^i < \bar{\kappa}_j^i,
\]

there exist \( (\epsilon_{pri}, \epsilon_{dual}) \neq 0 \in \mathbb{R} \) such that

\[
k_k^i < \eta_{pri}^i + \epsilon_{pri} < \kappa_j^i, \\
\bar{k}_k^i < \eta_{dual}^i + \epsilon_{dual} < \bar{k}_j^i.
\]

Hence, \( P^i \) remains constant under the perturbation of \( (\eta_{pri}, \epsilon_{dual}) \). The values of \( U_1, U_2 \) are constant for any \( (\eta_{pri}^i, \eta_{dual}^i) \) satisfying (32). Discontinuity exists at \( (\eta_{pri}^i, \eta_{dual}^i) = (\kappa_j^i, \bar{\kappa}_j^i) \), when the matrix \( P^i \) has different number of nonzero blocks resulting in different values of \( U_1, U_2 \).

\[\Box\]
