

Efficient Zero-Order NMPC with Feasibility and Stability Guarantees

Andrea Zanelli¹ and Rien Quirynen² and Moritz Diehl¹

Abstract—This paper discusses systems theoretic and computational aspects of a feasible, but suboptimal, nonlinear model predictive control scheme based on fixed sensitivities of the functions representing the constraints and cost of the underlying nonlinear programs. In particular, it will be shown how, by freezing the sensitivities computed at the desired steady state of the system, an efficient, structure-exploiting scheme is obtained that can considerably speed up the computations required for both construction and solution of the quadratic subproblems. Moreover, the local stability properties of the converged solution are analysed using results on pseudoexpansions of generalized equations present in the literature. The effectiveness of the proposed scheme is demonstrated on a non-trivial benchmark where large speedups can be achieved.

I. INTRODUCTION

Nonlinear model predictive control (NMPC) is an advanced optimization-based control technique that requires one to solve a series of neighbouring nonlinear programs (NLPs) [16]. Due to the computational burden associated with the solution of such problems, it has been historically primarily used in the chemical and process industry. In fact, in these fields, the typically slow dynamics of the systems to be controlled allow for long sampling times within which it is possible to carry out the required expensive operations.

During the last decade, considerable progress has been made related to algorithms and software implementations for NMPC that has significantly reduced its computational footprint [6]. For example, applications in the fields of automotive [2] and aerospace [19], [12], where sampling times are typically in the order of milli- and microseconds, have been targeted successfully.

Among others, a way of speeding up the computations is by using approximate formulations and inexact algorithms at the price of slower convergence or suboptimality of the obtained feedback policy. The schemes proposed in [5], [7] and [14] use a limited number of iterations in order to obtain an approximate solution. In [3], a variant of the multi-level real-time iteration scheme is introduced that uses fixed sensitivities in order to compute a feasible, but suboptimal solution. In [20], the local stability properties of the converged solution are analyzed in a simplified setting in which the active-set is assumed to be fixed.

This research was supported by the EU via H2020-ITN-AWESCO (642 682), by the Federal Ministry for Economic Affairs and Energy (BMW) via eco4wind (0324125B) and DyConPV (0324166B), and by DFG via Research Unit FOR 2401.

¹ Andrea Zanelli and Moritz Diehl are with the University of Freiburg, Freiburg, Germany andrea.zanelli@imtek.de, moritz.diehl@imtek.de

² Rien Quirynen is with Mitsubishi Electric Research Laboratories quirynen@merl.com

A. Contributions and outline

In this paper, we further develop the results obtained in [20] by extending stability results and proposing an efficient implementation of the scheme. In particular, by formulating the first-order optimality conditions of the NLP as a generalized equation, a stability result based on a local approximation of the Lyapunov function associated with the ideal setup is derived. To this end, the so-called *pseudoexpansion* [4] of the suboptimal solution is used in order to describe its asymptotic behavior in a neighborhood of the steady state. Although the idea is similar to the one used in [20], the main difference lies in the fact that the analysis carried out based on the framework of generalized equations allows one to make statements that are valid even across active-set changes, where an analysis based on Taylor series expansions would fail.

A second and complementary contribution of this work is an efficient structure-exploiting scheme that can significantly reduce the computation times. The scheme is based on a two-level algebraic elimination procedure that, starting from the large and sparse direct collocation formulation, allows one to solve lower dimensional quadratic programs (QPs). First, the exact lifted integrators proposed in [15] which, in this context, can be seen as a scheme to eliminate the internal variables associated with implicit integrators, are used to bring the linear systems into multiple-shooting form. Secondly, a condensing procedure is used to eliminate the state variables, such that a dense QP with fewer variables needs to be solved. Due to the fact that the sensitivities are kept constant, most computations needed for both elimination procedures can be carried out offline. These computations include factorizations needed to solve the collocation equations and the computation of the dense Hessian and its factorization. The only computations left to be carried out online consist in the evaluation of the (explicit) nonlinear functions describing cost and constraints of the NLP, condensing and expansions of right-hand sides and solution of the dense QP. For the latter, efficient rank one updates can be exploited to speed up the required computations [8].

The paper is structured as follows. In Section II, background material and preliminaries are presented. In Section III, the main theoretical result on the stability properties of the scheme are derived. Section IV describes the implementation details and, in Section V, numerical results based on an efficient C code implementation of the algorithm are presented.

II. BACKGROUND AND PRELIMINARIES

A. Problem formulation and standard stability results

Consider the following NMPC formulation:

$$\begin{aligned} \min_{\substack{x_0, \dots, x_N \\ u_0, \dots, u_{N-1}}} & \sum_{i=0}^{N-1} l(x_i, u_i) + m(x_N) \\ \text{s.t.} & \quad x_0 - \bar{x}_0 = 0, \\ & \quad \psi(x_i, u_i) - x_{i+1} = 0, \quad i = 0, \dots, N-1, \\ & \quad \pi(x_i, u_i) \leq 0, \quad i = 0, \dots, N-1, \\ & \quad \pi_N(x_N) \leq 0, \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ represent states and controls. The functions $l(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$, and $m(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ denote the stage and terminal costs respectively. The stage and terminal constraint functions are denoted by $\pi(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_\pi}$ and $\pi_N(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_{\pi_N}}$, while $\psi(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ describes the dynamics of the system. Finally, \bar{x}_0 represents the initial state of the system.

In the following, we will report standard stability results from [16] for the sake of completeness and clarity of the derivations carried out in Section III.

To this end, let $\mathbb{Z} := \{(x, u) \mid \pi(x, u) \leq 0\}$, $\mathbb{U}(x) := \{u \in \mathbb{R}^{n_u} \mid (x, u) \in \mathbb{Z}\}$, $\mathbb{X} := \{x \in \mathbb{R}^{n_x} \mid \mathbb{U}(x) \neq \emptyset\}$ and $\mathbb{X}_f := \{x \in \mathbb{X} \mid \pi_N(x) \leq 0\}$. Moreover, let $\bar{\mathbb{X}} \subseteq \mathbb{X}$ denote the set containing all the \bar{x}_0 for which (1) has a solution. We require the following assumptions to hold:

Assumption 1 (Continuity of system and cost): Assume that the origin is a steady state with $\psi(0, 0) = 0$, $l(0, 0) = 0$ and $m(0) = 0$. Moreover, assume that $l(\cdot)$, $m(\cdot)$, $\psi(\cdot)$, $\pi(\cdot)$ and $\pi_N(\cdot)$ are continuous.

Assumption 2 (Properties of constraint sets): The set \mathbb{Z} is closed and the set $\mathbb{U}(x)$ is compact and uniformly bounded in \mathbb{X} . The set $\mathbb{X}_f \subseteq \mathbb{X}$ is compact and each set contains the origin.

Assumption 3 (Basic stability assumption): $m(\cdot)$, \mathbb{X}_f and $l(\cdot)$ satisfy the following properties:

- 1) For all $x \in \mathbb{X}_f$, there exists a $u^\dagger(x)$ (such that $(x, u^\dagger) \in \mathbb{Z}$) satisfying

$$\begin{aligned} \psi(x, u^\dagger) &\in \mathbb{X}_f, \\ m(\psi(x, u^\dagger)) - m(x) &\leq -l(x, u^\dagger). \end{aligned}$$

- 2) There exist \mathcal{K}_∞ functions α_l and α_m such that

$$\begin{aligned} l(x, u) &\geq \alpha_l(\|x\|), \quad \forall x \in \bar{\mathbb{X}}, \forall u \text{ such that } (x, u) \in \mathbb{Z}, \\ m(x) &\leq \alpha_m(\|x\|), \quad \forall x \in \mathbb{X}_f. \end{aligned}$$

Assumption 4 (Weak controllability): There exists a \mathcal{K}_∞ function $\alpha(\cdot)$ such that, for the optimal cost associated with (1)

$$V^*(\bar{x}_0) := \sum_{i=0}^{N-1} l(x_i^*, u_i^*) + m(x_N^*), \quad (2)$$

the following holds:

$$V^*(x) \leq \alpha(\|x\|), \quad \forall x \in \bar{\mathbb{X}}. \quad (3)$$

Under the requirement that Assumptions 1, 2, 3 and 4 hold, it can be shown that $V^*(\bar{x}_0)$ is a Lyapunov function for

the closed-loop system obtained by controlling the system with the optimal feedback law $u_0^*(\bar{x}_0)$ in a receding horizon fashion [16]:

Theorem 1 ([16], Theorem 2.19): Suppose Assumptions 1, 2, 3 and 4 are satisfied. Then

- 1) There exist \mathcal{K}_∞ functions α_1 and α_2 such that

$$\begin{aligned} \alpha_1(\|x\|) &\leq V^*(x) \leq \alpha_2(\|x\|) \\ V^*(\psi(x, u_0^*(x))) - V^*(x) &\leq -\alpha_1(\|x\|) \end{aligned} \quad (4)$$

for all $x \in \bar{\mathbb{X}}$.

- 2) The origin of the closed-loop system is asymptotically stable in $\bar{\mathbb{X}}$.

We will make the additional assumption that $\alpha_l(\cdot)$ is a quadratic function:

Assumption 5: There exists $\beta > 0$ such that

$$l(x, u) \geq \beta \|x\|^2, \quad \forall x \in \bar{\mathbb{X}}, \forall u \text{ such that } (x, u) \in \mathbb{Z}. \quad (5)$$

III. STABILITY OF ZERO-ORDER NMPC

In order to simplify the following derivations, the case where $l(\cdot)$ and $m(\cdot)$ are quadratic functions will be considered and the NMPC problem (1) will be rewritten in compact form:

$$\begin{aligned} \min_y & \quad \frac{1}{2} y^T D y \\ \text{s.t.} & \quad g(y) + C \bar{x}_0 = 0, \\ & \quad h(y) \leq 0, \end{aligned} \quad (6)$$

where $y \in \mathbb{R}^{n_y}$ and the functions $g(\cdot) : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_g}$ and $h(\cdot) : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_h}$ are in addition assumed to be twice continuously differentiable.

We are interested in computing a solution of (6) for a given value of the parameter \bar{x}_0 by solving a series of quadratic programs (QPs) of the form

$$\begin{aligned} \min_{\Delta y} & \quad a_k^T \Delta y + \frac{1}{2} \Delta y^T D \Delta y \\ \text{s.t.} & \quad g(y^{[k]}) + G \Delta y + C \bar{x}_0 = 0, \\ & \quad h(y^{[k]}) + H \Delta y \leq 0, \end{aligned} \quad (7)$$

with $\Delta y := y - y^{[k]}$, where $y^{[k]}$ denotes the current value of the primal variables at iteration k . In the following, the stability properties of an inexact scheme will be analyzed. In particular, the algorithm uses only ‘‘zero-order’’ information, in the sense that it does not require the online evaluation of first- and second-order derivatives. Namely, the following approximations will be used:

$$\begin{aligned} a_k &:= D y^{[k]}, \quad G := \frac{\partial g}{\partial y}(0), \\ H &:= \frac{\partial h}{\partial y}(0), \quad D := \nabla_y^2 L(0, 0, 0, 0), \end{aligned} \quad (8)$$

where

$$L(y, \lambda, \mu, \bar{x}_0) := \frac{1}{2} y^T D y + \lambda^T (g(y) + C \bar{x}_0) + \mu^T h(y) \quad (9)$$

is the Lagrangian of (6) and D is assumed to be positive-definite. It is assumed that for $\bar{x}_0 = 0$ the solution to (6) is $y = 0$, $\lambda = 0$ and $\mu = 0$.

If convergence is achieved, i.e. $\tilde{y} := y^{[k+1]} = y^{[k]}$, $\tilde{\lambda} := \lambda^{[k+1]} = \lambda^{[k]}$ and $\tilde{\mu} := \mu^{[k+1]} = \mu^{[k]}$, from the first-order optimality conditions of the QPs, we obtain

$$\begin{aligned} D\tilde{y} + G^T\tilde{\lambda} + H^T\tilde{\mu} &= 0, \\ g(\tilde{y}) + C\bar{x}_0 &= 0, \\ h(\tilde{y}) &\leq 0, \\ h_i(\tilde{y})\tilde{\mu}_i &= 0, \quad i = 0, \dots, n_h - 1, \end{aligned} \quad (10)$$

which can be interpreted as the first-order optimality conditions of the nonlinear program [3]

$$\begin{aligned} \min_y \quad & \frac{1}{2}y^T Dy + \xi(\bar{x}_0)^T y \\ \text{s.t.} \quad & g(y) + C\bar{x}_0 = 0, \\ & h(y) \leq 0, \end{aligned} \quad (11)$$

with

$$\begin{aligned} \xi(\bar{x}_0) &:= (G^T - \nabla g(\tilde{y}(\bar{x}_0)))\tilde{\lambda}(\bar{x}_0) + \\ & (H^T - \nabla h(\tilde{y}(\bar{x}_0)))\tilde{\mu}(\bar{x}_0). \end{aligned}$$

The local convergence of the algorithm has been analyzed in [3] based on an adaptation of standard arguments for Newton-type algorithms.

Remark: although we will focus in the following on the algorithm with fixed sensitivities described above, in a practical implementation it is possible in principle to update them in an asynchronous fashion as proposed in [3] in order to cope, for example, with changes in the reference.

A. Solution pseudoexpansion

In order to quantify the magnitude of the deviation of the solution \tilde{y} to the approximate NLP (11) from the optimal solution y^* to the original NLP in (6), it is possible to use an approximation of the solutions of problems (6) and (11) [4]. To this end, we will rewrite the optimality system of (6) as a generalized equation.

Definition 2: Let the generalized equation

$$F(z, \bar{x}_0) \in N(z), \quad (12)$$

represent the optimality system of (6), where

$$F := \begin{pmatrix} \nabla_y L(y, \nu, \bar{x}_0) \\ \bar{g}(y) + \bar{C}\bar{x}_0 \end{pmatrix} \quad (13)$$

with $z := (y, \nu)$, $\nu := (\lambda, \mu)$, $\bar{C} := [C^T \ 0]^T$, $\bar{g}(y) := (g(y), h(y))$ and $N := \{0\} \times N_K^{-1}$, where

$$N_K^{-1}(\nu) := \begin{cases} v \in K : \langle \nu, v \rangle = 0, & \text{if } \nu \in K^- \\ \emptyset, & \text{otherwise} \end{cases} \quad (14)$$

and $K^- := \mathbb{R}^{n_g} \times \mathbb{R}_+^{n_h}$ is used to denote the polar cone of the set $K := \{0\} \times \mathbb{R}_+^{n_h}$.

It is possible to approximate the solution $z(\bar{x}_0)$ by $z^0 + z^1(\bar{x}_0)$, which we will call a *pseudoexpansion* [4], where z^0 denotes the unperturbed solution $z(0)$ and $z^1(\bar{x}_0)$ represents the solution to the generalized equation (with unknown ζ)

$$F(z^0, 0) + \nabla_z F(z^0, 0)^T \zeta + \nabla_{\bar{x}_0} F(z^0, 0)^T \bar{x}_0 \in N(z^0 + \zeta). \quad (15)$$

Definition 3: Let, for $\bar{x}_0 = 0$, z^0 be a solution of the generalized equation (12). We say that the strong stability conditions hold (that z^0 is strongly stable) if there exist $\epsilon > 0$ and $M > 0$ such that for all $v \in B_V(0, \epsilon)$ the linearized generalized equation

$$F(z^0, 0) + \nabla_z F(z^0, 0)^T \zeta + v \in N(z^0 + \zeta) \quad (16)$$

has in $B_Z(0, M)$ a unique solution $\hat{z}(v)$ and

$$\hat{z}(\cdot) : B_V(0, \epsilon) \rightarrow B_Z(0, M) \quad (17)$$

is Lipschitz continuous, where

$$B_X(x, r) = \{x' \in X : \|x - x'\| < r\} \quad (18)$$

is used to denote the open ball of radius $r > 0$ centered at x .

We will assume in the following that strong stability conditions hold for z^0 .

Assumption 6: The solution $z(\bar{x}_0)$ of (12) is strongly stable for all \bar{x}_0 in \mathcal{P} , with $0 \in \mathcal{P}$.

In particular, the following theorem [4, Theorem 5.1] will be useful to derive results on the suboptimality of the solution \tilde{y} .

Theorem 4: Suppose that, for $\bar{x}_0 = 0$, z^0 is a solution of the generalized equation (12) and that strong stability holds. Then, for all \bar{x}_0 in a neighborhood of 0, the mappings $z(\bar{x}_0)$ and $z^1(\bar{x}_0)$ are well defined in the vicinity of z^0 and in $B_Z(0, M)$, respectively. In addition, $z(\bar{x}_0)$ is Lipschitz continuous:

$$z^1(\bar{x}_0) = O(\|\bar{x}_0\|) \quad (19)$$

and the following holds:

$$z(\bar{x}_0) = z^0 + z^1(\bar{x}_0) + o(\|\bar{x}_0\|). \quad (20)$$

Assumption 7: Let the generalized equation

$$\tilde{F}(z, \bar{x}_0) \in N(z), \quad (21)$$

represent the optimality system associated with (11) and let $\tilde{z}(\bar{x}_0)$ denote a solution of (21). Assume that, for all \bar{x}_0 in \mathcal{P} , $\tilde{z}(\bar{x}_0)$ is strongly stable.

Proposition 5: Regard problem (11) and let Assumption 7 hold. Then, the following holds:

$$\xi(\bar{x}_0) = O(\|\bar{x}_0\|^2). \quad (22)$$

Proof: Due to Assumption 7, and given that $\tilde{z}^0 = 0$, $\tilde{\nu} = O(\|\bar{x}_0\|)$ and $\tilde{y} = O(\|\bar{x}_0\|)$. Moreover, since $G^T - \nabla g(\tilde{y}(\bar{x}_0)) = O(\|\bar{x}_0\|)$ and $H^T - \nabla h(\tilde{y}(\bar{x}_0)) = O(\|\bar{x}_0\|)$, then $\xi(\bar{x}_0) = O(\|\bar{x}_0\|^2)$. ■

Consider now the following parametrization of problem (11):

$$\begin{aligned} \min_y \quad & \frac{1}{2}y^T Dy + \xi^T y \\ \text{s.t.} \quad & g(y) + C\bar{x}_0 = 0, \\ & h(y) \leq 0, \end{aligned} \quad (23)$$

where ξ is regarded as a parameter.

Proposition 6: Let Assumption 7 hold. Then, for every \bar{x}_0 in \mathcal{P} , the following holds:

$$z(\bar{x}_0) - \tilde{z}(\xi, \bar{x}_0) = O(\|\xi\|). \quad (24)$$

Proof: The result is a direct consequence of the facts that $\tilde{z}(0, \bar{x}_0) = z(\bar{x}_0)$ and that, due to Theorem 4, $\tilde{z}^1(\xi, \bar{x}_0) = O(\|\xi\|)$ for all \bar{x}_0 in \mathcal{P} . ■

Lemma 7: Let Assumptions 6 and 7 hold. Then

$$z(\bar{x}_0) - \tilde{z}(\bar{x}_0) = O(\|\bar{x}_0\|^2). \quad (25)$$

Proof: The result is a direct consequence of Propositions 5 and 6. ■

Lemma 8: Let Assumptions 6 and 7 hold. Then, the following holds:

$$\frac{1}{2}y^*(\bar{x}_0)^T D y^*(\bar{x}_0) - \frac{1}{2}\tilde{y}(\bar{x}_0)^T D \tilde{y}(\bar{x}_0) = O(\|\bar{x}_0\|^4). \quad (26)$$

Proof: In the following regard $\tilde{y} = \tilde{y}(\bar{x}_0)$, $y^* = y^*(\bar{x}_0)$ and $\xi = \xi(\bar{x}_0)$. Given that \tilde{y} is a minimizer for (11), it holds that

$$\frac{1}{2}\tilde{y}^T D \tilde{y} + \xi^T \tilde{y} \leq \frac{1}{2}y^{*T} D y^* + \xi^T y^*,$$

for any feasible y in $B_Y(\tilde{y}, \sigma)$ for some $\sigma > 0$ and, in particular, for $\|\bar{x}_0\|$ sufficiently small, we can write

$$\frac{1}{2}\tilde{y}^T D \tilde{y} - \frac{1}{2}y^{*T} D y^* \leq \xi^T (y^* - \tilde{y}).$$

Moreover, due to Proposition 5 and Lemma 7, we have that $\xi(\bar{x}_0) = O(\|\bar{x}_0\|^2)$ and $y^*(\bar{x}_0) - \tilde{y}(\bar{x}_0) = O(\|\bar{x}_0\|^2)$, which implies that

$$\frac{1}{2}\tilde{y}^T D \tilde{y} - \frac{1}{2}y^{*T} D y^* = O(\|\bar{x}_0\|^4). \quad (27)$$

B. Lyapunov function for the Zero-Order scheme

The results stated in Lemma 8 can be used to establish local stability guarantees for the Zero-Order NMPC scheme under analysis.

Theorem 9: Let all the above Assumptions hold. Then, the origin is a locally exponentially stable equilibrium for the closed-loop system obtained by applying the suboptimal policy $\tilde{u}_0(\bar{x}_0)$.

Proof: Let $\tilde{V}(\bar{x}_0) := \frac{1}{2}\tilde{y}(\bar{x}_0)^T D \tilde{y}(\bar{x}_0)$. Due to Assumption 3, we can write the following:

$$\begin{aligned} \tilde{V}(\bar{x}_0) &\geq \beta\|\bar{x}_0\|^2 + \hat{V}_{N-1}(\psi(\bar{x}_0, \tilde{u}(\bar{x}_0))) \\ &\geq \beta\|\bar{x}_0\|^2 + \hat{V}_{N-1}(\psi(\bar{x}_0, \tilde{u}(\bar{x}_0))) \\ &\quad - m(\tilde{x}_N) + m(\psi(\tilde{x}_N, u^\dagger(\tilde{x}_N))) + l(\tilde{x}_N, u^\dagger(\tilde{x}_N)) \\ &\geq \beta\|\bar{x}_0\|^2 + V^*(\psi(\bar{x}_0, \tilde{u}(\bar{x}_0))), \end{aligned}$$

where

$$\hat{V}_{N-1}(\psi(\bar{x}_0, \tilde{u}_0)) := \sum_{i=1}^{N-1} l(\tilde{x}_i, \tilde{u}_i) + m(\tilde{x}_N). \quad (28)$$

Finally, applying (26), we have that

$$\tilde{V}(\psi(\bar{x}_0, \tilde{u}(\bar{x}_0))) - \tilde{V}(\bar{x}_0) \leq -\beta\|\bar{x}_0\|^2 + O(\|\bar{x}_0\|^4),$$

which shows that a positive definite function $\tilde{\alpha}$ must exist such that

$$\tilde{V}(\psi(\bar{x}_0, \tilde{u}(\bar{x}_0))) - \tilde{V}(\bar{x}_0) \leq -\tilde{\alpha}(\|\bar{x}_0\|) \quad (29)$$

holds in a neighborhood of the origin. ■

IV. IMPLEMENTATION DETAILS

In this section, the implementation details of the proposed algorithm are presented. In particular, the iterations (7) will be specialized to the structure obtained by using the direct collocation discretization scheme and an efficient elimination strategy based on lifted integrators [15] and the condensing routines proposed in [9] will be described. The main underlying idea is that, due to the fact that the sensitivities are being frozen according to (8), several computations that need in general to be carried out online can be performed offline.

Remark: although in some cases it might be more computationally efficient to use explicit integrators, the two-level elimination strategy will be described for the case where implicit integrators are used to discretize the continuous-time dynamics. This is done on purpose, since the proposed algorithm can significantly speed up the computations associated with the solution of implicit collocation equations as well.

A. Two-level algebraic elimination

Consider the following discrete-time optimal control problem in direct collocation form:

$$\begin{aligned} \min_{x, u, v} \quad & \sum_{i=0}^{N-1} l(x_i, u_i) + m(x_N) \\ \text{s.t.} \quad & x_0 - \bar{x}_0 = 0, \\ & \phi(w_i, v_i) = 0, \quad i = 0, \dots, N-1, \\ & x_i + C v_i - x_{i+1} = 0, \quad i = 0, \dots, N-1, \\ & \pi(x_i, u_i) \leq 0, \quad i = 0, \dots, N-1, \\ & \pi_N(x_N) \leq 0, \end{aligned} \quad (30)$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $v \in \mathbb{R}^{n_v}$ are the states, controls and collocation variables, respectively, and $w_i := [x_i^T, u_i^T]^T$. The equation $\phi(w_i, v_i) = 0$ represents the collocation equations

$$\phi(w_i, v_i) := \begin{bmatrix} \psi_c(v_i^1, x_i + T_{\text{int}} \sum_{s=1}^q a_{1,s} v_i^s, u_i) \\ \vdots \\ \psi_c(v_i^q, x_i + T_{\text{int}} \sum_{s=1}^q a_{q,s} v_i^s, u_i) \end{bmatrix} \quad (31)$$

associated with stage i , where q denotes the number of collocation nodes and the scalars $a_{i,j}$ with $i, j = 1, \dots, q$ are the coefficients of the collocation method. The integration step size is represented by T_{int} and C in (30) is a constant matrix that depends on T_{int} and the collocation nodes. The function

Algorithm 1 Level-1 Elimination: Lifted Integrators

input: current iterates $w^{[k]}, v^{[k]}$
output: updated iterates $w^{[k+1]}, v^{[k+1]}$

- 1: L1-Condensing procedure
 - 2: **for** $i = 0, \dots, N - 1$ **do**
 - 3: $\Delta \tilde{v}_i \leftarrow M^{-1} \phi(w_i^{[k]}, v_i^{[k]}),$
 - 4: $\bar{c}_i \leftarrow x_i^{[k]} + C v_i^{[k]} - x_{i+1}^{[k]} + C \Delta \tilde{v}_i$
 - 5: **end for**
 - 6: QP solution (Algorithm 2)
 - 7: $w^{[k+1]} \leftarrow w^{[k]} + \Delta w$
 - 8: L1-Expansion procedure
 - 9: **for** $i = 0, \dots, N - 1$ **do**
 - 10: $v_i^{[k+1]} \leftarrow v_i^{[k]} + \Delta \tilde{v}_i + V \Delta w_i$
 - 11: **end for**
-

ψ_c in (31) characterizes the fully implicit continuous-time dynamics

$$0 = \psi_c(\dot{x}(t), x(t), \bar{u}). \quad (32)$$

Remark: as described in [15], the presented algorithm can be easily extended to the case where a differential-algebraic equation describes the dynamics, and to the case where more than one intermediate integration step is carried out per shooting node. For the sake of brevity, we will restrict ourselves to the slightly less general formulation (30)-(31). The proposed algorithm solves, at every iteration, QP subproblems of the form

$$\begin{aligned} \min_{\substack{\Delta x_0, \dots, \Delta x_N \\ \Delta u_0, \dots, \Delta u_{N-1}}} & \sum_{i=0}^{N-1} f_w(\Delta w_i) + f_{x_N}(\Delta x_N) \\ \text{s.t.} & \quad x_0 - \bar{x}_0 = 0, \\ & \quad g_i(\Delta w_i, \Delta x_{i+1}) = 0, \quad i = 0, \dots, N-1, \\ & \quad h_i(\Delta w_i) \leq 0, \quad i = 0, \dots, N-1, \\ & \quad h_N(\Delta x_N) \leq 0, \end{aligned} \quad (33)$$

where

$$\begin{aligned} f_w(\Delta w_i) &:= \frac{1}{2} \Delta w_i^T Q \Delta w_i + q_i^T \Delta w_i, \\ f_{x_N}(\Delta x_N) &:= \frac{1}{2} \Delta x_N^T Q_N \Delta x_N + q_N^T \Delta x_N \end{aligned} \quad (34)$$

and

$$g_i(\Delta w_i, \Delta x_{i+1}) := A \Delta x_i + B \Delta u_i - \Delta x_{i+1} + \bar{c}_i.$$

Moreover

$$\begin{aligned} h_i(\Delta w_i) &:= H \Delta w_i + \bar{h}_i, \\ h_N(\Delta x_N) &:= H_N \Delta x_N + \bar{h}_N. \end{aligned} \quad (35)$$

The matrices and vectors in the QP subproblem (33) are all fixed and precomputed offline apart from \bar{c}_i , \bar{h}_i and \bar{h}_N , which need to be updated online. In particular, they are defined as evaluations of first- and second-order derivatives at the steady state (primal and dual) solution of the following quantities:

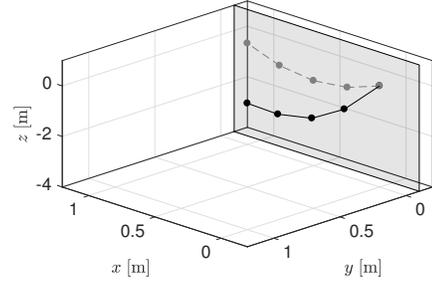
$$\begin{aligned} Q &:= \nabla_w^2 l, & Q_N &:= \nabla_x^2 m, \\ q_i &:= \nabla_w l + Q w_i^{[k]}, & q_N &:= \nabla_x m + Q_N x_N^{[k]}, \end{aligned} \quad (36)$$

Algorithm 2 Level-2 Elimination: States Condensing

input: current iterates $w^{[k]}$ and QP data \bar{c}_i, \bar{h}_i for $i = 0, \dots, N - 1$ and \bar{h}_N
output: primal step Δw

- 1: L2-Condensing procedure
 - 2: update condensed QP ([11] - Algorithms 6, 8, 14)
 - 3: Condensed QP solution
 - 4: compute Δu
 - 5: L2-Expansion procedure
 - 6: compute Δx ([11] - Equation 9.1)
-

Fig. 1: Nonlinear hanging chain benchmark for $n_m = 5$ masses [13]. The dashed sketch describes the equilibrium at which the fixed quantities used by Algorithm 1 and 2 are computed.



and

$$A := I + C V_x, \quad B := C V_u, \quad (37)$$

where

$$V_x := -\frac{\partial \phi^{-1}}{\partial v} \frac{\partial \phi}{\partial x}, \quad V_u := -\frac{\partial \phi^{-1}}{\partial v} \frac{\partial \phi}{\partial u}, \quad V := [V_x, V_u].$$

Moreover, the matrix M^{-1} in Algorithm 1 is defined as

$$M^{-1} := -\frac{\partial \phi^{-1}}{\partial v}. \quad (38)$$

Finally,

$$H := \frac{\partial \pi}{\partial w}, \quad H_N := \frac{\partial \pi_N}{\partial x}. \quad (39)$$

While the terms \bar{c}_i , $i = 0, \dots, N - 1$ are being updated using Algorithm 1, the terms \bar{h}_i , $i = 0, \dots, N - 1$ and \bar{h}_N are computed by evaluating the constraint functions at the current iterates:

$$\bar{h}_i := \pi(x_i^{[k]}, u_i^{[k]}), \quad i = 0, \dots, N - 1, \quad \bar{h}_N := \pi_N(x_N^{[k]}).$$

After problem (33) is formed through steps 1 – 5 of Algorithm 1, a (states) condensing routine is used to update a condensed QP whose solution delivers the Newton step in the input variables Δu as described in Algorithm 2. Notice that, since the QP matrices in (33) are constant throughout the iterations, the step 5 of Algorithm 2, only involves the update of gradients and right-hand sides resulting in a tailored condensing routine that is significantly cheaper than the standard implementation.

N	10	10	10	20	20	20	30	30	30
n_m	5	6	7	5	6	7	5	6	7
n_s	2	4	6	2	4	6	2	4	6
RTI	1.20	4.65	19.96	2.54	9.97	40.41	4.80	16.04	61.41
0-RTI	0.43	0.44	0.94	0.81	1.32	2.34	1.84	2.65	4.23
speedup	2.79	10.56	21.37	3.13	7.55	17.27	2.61	6.05	14.52

TABLE I: Closed-loop worst-case computation time, in milliseconds, for the standard (RTI) and Zero-Order (0-RTI) real-time iteration schemes for different prediction horizons N , number of masses n_m and number of stages for the collocation integrators n_s . In all simulations the system is steered to steady state. Using the 0-RTI scheme, a maximum increase of less than 0.1% in the closed-loop cost is incurred with respect to the standard RTI.

V. NUMERICAL RESULTS

The Zero-Order scheme described in Algorithms 1 and 2 has been implemented in C within the framework for nonlinear embedded optimization `acados` [17] using the high-performance linear algebra library `BLASFEO` [10]. The QP solver `qpOASES` [8] is used in order to exploit hot-starting of its active-set strategy and avoid the necessity of factorizing the condensed Hessian at every iteration. The efficient condensing routines implemented in `HPIPM` [1] are used in order to carry out steps 1 and 5 in Algorithm 2. In the following, a numerical case-study based on a scalable example is presented where it is shown how considerable speedups can be achieved. All benchmarks are run on a Dell XPS13-9360 equipped with an Intel i7-7560U with maximum and minimum frequency set to the nominal value of 2.40 GHz.

Remark: although the theoretical results derived in Section III are related to the converged solution associated with problem (11), in this section a real-time implementation of the algorithm will be considered which requires the solution of a single QP per sampling time as usually done in practice when using RTIs [5]. An extension of the systems theoretic results is subject of undergoing research.

A. Nonlinear hanging chain - Timings

The system as presented in [18] and [8] consists in a hanging chain of masses connected by springs described by a differential equation with $n_x = 6(n_m - 2) + 3$ states, where n_m represents the number of masses in the chain. The chain is controlled by adjusting the velocities of the mass at one of its ends resulting in $n_u = 3$ controls, while the opposite end is fixed. Figure 1 shows a sketch of the system under consideration.

Following the notation in (30), a tracking formulation with

$$l = \frac{1}{2}(x - x_{ss})^T Q(x - x_{ss}) + \frac{1}{2}(u - u_{ss})^T R(u - u_{ss}) \quad (40)$$

and

$$m = \frac{1}{2}(x - x_{ss})^T Q_N(x - x_{ss}) \quad (41)$$

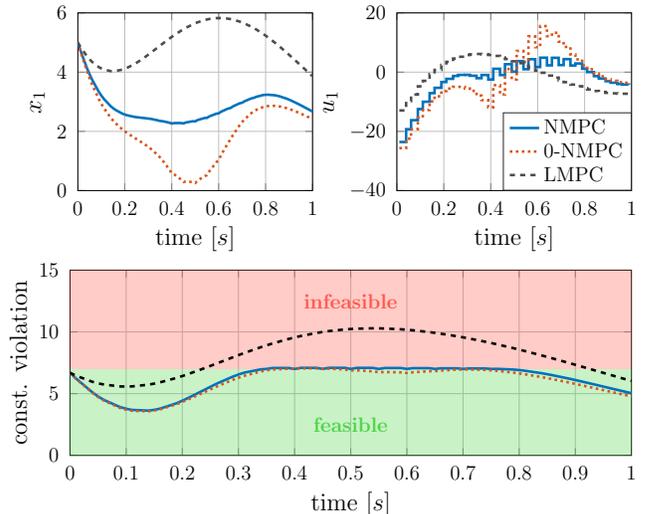


Fig. 2: Open-loop trajectories obtained with the original (solid), Zero-Order (dotted) and linear-quadratic (dashed) formulations. An additional constraint is added to the problem used in [18] that requires the position of the actuated mass to be within a ball. Although the Zero-Order scheme is clearly suboptimal, the obtained trajectories satisfy the nonlinear constraint (unlike with the linear-quadratic formulation).

will be used, where $Q = Q_N = 100 \cdot I_{n_x}$ and $R = I_{n_u}$. Double-sided box constraints are imposed on the inputs $u_{\min} \leq u \leq u_{\max}$ and single-sided constraints for the states $x_{\min} \leq x$ are included that represent the wall on the side of the chain. Implicit lifted collocation integrators [15] of type Gauss-Legendre with order $2n_s$ are used to discretize the dynamics of the system.

Table I shows the worst-case CPU time in milliseconds obtained in a closed-loop simulation using the standard (RTI) and the proposed (0-RTI) real-time iteration schemes. Especially for large number of masses n_m and number of stages of the collocation integrators n_s , a large speedup can be achieved with respect to the standard scheme. For these benchmarks, the maximum increase in the closed-loop cost with respect to the standard RTI is below 0.1%.

B. Nonlinear hanging chain - Control performance

In order to show the benefit in terms of control performance of the proposed scheme, a slight adaptation of the original formulation used in [18] will be taken into account. In particular, a convex quadratic constraint that requires the position of the actuated mass to be within a ball of a fixed radius $\bar{\rho}$ centered around \bar{p} is introduced:

$$\|p - \bar{p}\|_2^2 - \bar{\rho}^2 \leq 0, \quad (42)$$

where p represents the position of the actuated mass.

Figure 2 shows the open-loop trajectories obtained by solving the exact NLP, the approximate Zero-Order and the linear-quadratic formulations obtained by using the required fixed quantities computed at the steady-state. Although the Zero-Order trajectories are clearly suboptimal, the advantage over

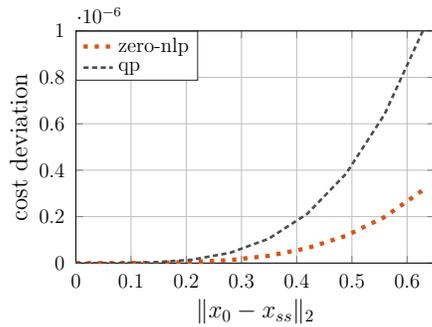


Fig. 3: Asymptotic suboptimality of Zero-Order and linear-quadratic formulations in open-loop as a function of the deviation of x_0 from the steady-state. The Zero-Order scheme provides a superior approximation of the optimal cost.

the linear-quadratic formulation is evident due to the fact that the additional nonlinear constraint (42) can be satisfied when using the proposed approach.

Finally, in order to illustrate the superior asymptotic approximation of the optimal cost, Figure 3 shows the deviation of the open-loop costs obtained with the Zero-Order and linear-quadratic formulations.

VI. CONCLUSIONS AND OUTLOOK

In this paper, an efficient NMPC scheme that uses frozen sensitivities is presented. Both systems theoretic and computational aspects are discussed and an efficient implementation in C is provided that can achieve speedups of more than one order of magnitude with respect to state-of-the-art algorithms and implementations.

The stabilizing properties of the converged solution are analyzed and a local Lyapunov function is constructed using results from the field of generalized equations.

An extension of the stability proof to the real-time variant of the scheme is subject of undergoing research.

REFERENCES

- [1] HPIPM, High-performance interior-point qp solvers. <https://github.com/giaf/hpipm>, 2017.
- [2] T. Albin, D. Ritter, N. Liberda, R. Quirynen, and M. Diehl. In-vehicle realization of nonlinear MPC for gasoline two-stage turbocharging airpath control. *IEEE Transactions on Control Systems Technology*, pages 1–13, 2017.
- [3] H. G. Bock, M. Diehl, E. A. Kostina, and J. P. Schlöder. Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-Time and Online PDE-Constrained Optimization*, pages 3–22. SIAM, 2007.
- [4] J. Frederic Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40:228–264, 1998.
- [5] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*, volume 920 of *Fortschritt-Berichte VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik*. VDI Verlag, Düsseldorf, 2002. PhD Thesis.
- [6] M. Diehl, H. J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In L. Magni, M.D. Raimondo, and F. Allgöwer, editors, *Nonlinear model predictive control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 391–417. Springer, 2009.
- [7] Christian Feller and Christian Ebenbauer. A stabilizing iteration scheme for model predictive control based on relaxed barrier functions. *arXiv preprint at arXiv:1603.04605*, April 2016.

- [8] H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.
- [9] G. Frison and J. B. Jørgensen. A fast condensing method for solution of linear-quadratic control problems. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pages 7715–7720, 2013.
- [10] G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, and M. Diehl. BLAS-FEO: Basic linear algebra subroutines for embedded optimization. *ACM Transactions on Mathematical Software (TOMS)*, 44(4):42:1–42:30, 2018.
- [11] Gianluca Frison. *Algorithms and Methods for High-Performance Model Predictive Control*. PhD thesis, Technical University of Denmark (DTU), 2015.
- [12] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart. Fast nonlinear model predictive control for multicopter attitude tracking on so(3). In *IEEE Conference on Control Applications*, Sidney, Australia, Sept 2015.
- [13] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl. Recent advances in quadratic programming algorithms for nonlinear model predictive control. *Vietnam Journal of Mathematics*, 46(4):863–882, 2018.
- [14] T. Ohtsuka. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.
- [15] Rien Quirynen, Sébastien Gros, Boris Houska, and Moritz Diehl. Lifted collocation integrators for direct optimal control in ACADO toolkit. *Mathematical Programming Computation*, 9(4):527–571, 2017.
- [16] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill, 2nd edition edition, 2017.
- [17] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Niels van Duijkeren, Andrea Zanelli, Rien Quirynen, and Moritz Diehl. Towards a modular software package for embedded optimization. In *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, 2018.
- [18] L. Wirsching, H. G. Bock, and M. Diehl. Fast NMPC of a chain of masses connected by springs. In *Proceedings of the IEEE International Conference on Control Applications, Munich*, pages 591–596, 2006.
- [19] A. Zanelli, G. Horn, G. Frison, and Moritz Diehl. Nonlinear model predictive control of a human-sized quadrotor. In *Proceedings of the European Control Conference (ECC)*, pages 1542–1547, 2018.
- [20] A. Zanelli, R. Quirynen, and M. Diehl. An efficient inexact NMPC scheme with stability and feasibility guarantees. In *Proceedings of 10th IFAC Symposium on Nonlinear Control Systems*, Monterey, CA, USA, August 2016.