Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-18 (2024) 335–340

Time-certified Input-constrained NMPC via Koopman Operator ⋆

Liang Wu * **Krystian Ganko** * **Ricahrd D. Braatz** *

* *Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: {liangwu,kkganko,braatz}@mit.edu).*

Abstract: Determining solving-time certificates of nonlinear model predictive control (NMPC) implementations is a pressing requirement when deploying NMPC in production environments. Such a certificate guarantees that the NMPC controller returns a solution before the next sampling time. However, NMPC formulations produce nonlinear programs (NLPs) for which it is very difficult to derive their solving-time certificates. Our previous work, Wu and Braatz (2023), challenged this limitation with a proposed input-constrained MPC algorithm having exact iteration complexity but was restricted to linear MPC formulations. This work extends the algorithm to solve input-constrained NMPC problems, by using the Koopman operator and a condensing MPC technique. We illustrate the algorithm performance on a high-dimensional, nonlinear partial differential equation (PDE) control case study, in which we theoretically and numerically certify the solving time to be less than the sampling time.

Keywords: Nonlinear model predictive control, Koopman operator, Extended dynamic mode decomposition, feasible path-following interior-point method, iteration complexity.

1. INTRODUCTION

Model predictive control (MPC) is a model-based optimal control technique widely applied in a range of applications, including in manufacturing processes, energy systems, and robotics, see Qin and Badgwell (2003). At each sampling time, MPC solves an on-line optimization which is formulated with a dynamical prediction model and user-specified constraints and objectives.

Linear MPC is formulated using a linear process model, which leads to solving a quadratic program (QP). Nonlinear MPC (NMPC) instead adopts a nonlinear model, which results in a nonlinear program (NLP) that has a higher computational burden than the corresponding QP. Nearly all industrial systems are better described by nonlinear models, but deploying more computationally expensive NMPC under real-time process requirements for fast time-scale applications, such as robotics, is more challenging than deploying linear MPC.

A key requirement for deploying MPC in production environments is the execution speed (or, throughput), which is measured by both (*i*) the average execution time (to free the processor to execute other tasks), and (*ii*) the worst-case execution time which needs to be less than the sampling time. Most studies concentrate on developing algorithms with fast average execution time, see Ferreau et al. (2014); Stellato et al. (2020); Wu and Bemporad (2023b,a). The emphasis, however, should fall more on the certification of worst-case execution time, which is evident from the common assumption that the solution of the current MPC optimization task must be prepared before the arrival of the next sampling time, e.g., (Zavala and Biegler, 2009, Assumption 5).

The worst-case execution time is computed from the worst-case number of floating-point operations ("FLOP") using the approximate relation

$$\text{execution time} = \frac{\text{total \# FLOP in NLP solve}}{\text{average \# FLOP per second}},$$

in which the denominator roughly depends on the embedded processor technology. [1] Determining the worst-case number of floating point operations subsequently requires ascertaining the worst-case number of iterations in the iterative optimization algorithm used by the MPC scheme. Certifying the number of iterations of an iterative optimization algorithm is particularly challenging for the on-line NMPC optimization. In most iterative optimization algorithms, the number of iterations depends on the data of the optimization such as the Hessian matrix and the gradient vector, and the data of the MPC optimization depends on the feedback states at each sampling time. More specifically for NMPC formulations, techniques that simplify the NLPs by reformulation as QPs (e.g., via successive online linearization or real-time iteration, see Gros et al. (2020)) cause the Hessian matrix to also become time-varying.

The remainder of this section summarizes past research in obtaining worst-case iteration numbers in linear MPC, and the novel extensions of this work to NMPC schemes. In Giselsson (2012); Richter et al. (2011); Bemporad and

⋆ This work was supported by the U.S. Food and Drug Administration under the FDA BAA-22-00123 program, Award Number 75F40122C00200. KG was also supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0022158.

[1] The number of flops used in each MPC calculation is sufficiently large that using an average in the denominator is highly accurate.

10.1016/j.ifacol.2024.09.052

Patrinos (2012), accelerated gradient methods were used to solve the dual problem of linear MPC. The certification procedure on the worst-case number of iterations was also provided to determine the worst-case execution time. In these first-order methods, the iteration bound depends on the distance between the initial point and the optimal solution, which is unknown in advance and needs to be estimated. In practice, these methods are too conservative, i.e., their derived worst-case iteration bound is typically about two orders of magnitude larger than the actual number of iterations, see Richter et al. (2011).

In Cimini and Bemporad (2017); Arnström and Axehill (2019); Cimini and Bemporad (2019), active-set methods were used to solve QPs that arise from linear MPC, and the certification procedure of the iteration bound was described. These works rely on a technique that combines explicit MPC (i.e., off-line generation of a lookup table for the feedback control law) and implicit MPC (i.e., on-line optimization to solve for the feedback control law) to certify the iteration bound. Explicit MPC (Bemporad et al. (2002); Alessio and Bemporad (2009)) directly provides the certificate of the execution time as the worst-case search time in the lookup table. However, explicit MPC is practically limited to small and medium-sized problems where lookup table sizes are manageable. Another interesting work from Okawa and Nonaka (2021) also illustrated how to derive the iteration bound. The input-constrained MPC was first formulated as a linear complementarity problem (LCP) and then a *modified N-step vector* was found via linear programming to ensure that the LCP is solved in N iterations.

Unfortunately, the complexity of the procedures to derive certificates of worst-case execution time for these linear MPC methods hinders their extension to NMPC problems. Our previous work, Wu and Braatz (2023), first proposed an input-constrained linear MPC algorithm with the exact number of iterations,

$$\mathcal{N} = \left\lceil \frac{\log(\frac{2n}{\epsilon})}{-2\log(\frac{\sqrt{2n}}{\sqrt{2n}+\sqrt{2}-1})} \right\rceil + 1.$$

where $\lceil c \rceil$ maps c to the least integer greater than or equal to c. This result is independent of the optimization problem data and dependent only on the number of variables n and the stopping criterion ϵ, making it suitable for parametric MPC problems. In this work, we extend the result to input-constrained nonlinear MPC problems by using the Koopman operator, which identifies a linear model by "lifting" the state-space dimension of nonlinear dynamical systems. Our result is the first to produce a time-certified algorithm for input-constrained NMPC problems.

2. PROBLEM FORMULATIONS

Consider the input-constrained NMPC formulation

$$\begin{aligned} \text{(NMPC)} \quad \min \quad & J(\hat{x}_t) = l_N(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{s.t.} \quad & x_0 = \hat{x}_t \\ & x_{k+1} = f(x_k, u_k),\ k \in \mathbb{Z}_{0,N-1}, \\ & u_{\min} \le u_k \le u_{\max},\ k \in \mathbb{Z}_{0,N-1}, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}, u_k \in \mathbb{R}^{n_u}$ denote the states and the control input, respectively, at time instance k and $\hat{x}_t$ denotes the feedback states at the current sampling time t. The nonlinear function $f(\cdot) : \mathbb{R}^{n_x} \times \in \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ defines the dynamic model of the plant. The control inputs are constrained in $[u_{\min}, u_{\max}]$, which are from the physical limits of the actuators. This article uses the standard formulation in which $l_N = \frac{1}{2}\|x_N - x_r\|^2_{W_N}$ and $l(x_k, u_k) = \frac{1}{2}\|u_k - u_r\|^2_{W_u} + \frac{1}{2}\|x_k - u_r\|^2_{W_x}$, where x_r, u_r are the targeted tracking references for the states and the control inputs, and W_N, W_u, W_x denote the weight matrices for the terminal states, the control input, and the non-terminal states, respectively.

The formulation (NMPC) (1) is an NLP that must be solved at every sampling time. There exist many efficient algorithms for solving (NMPC), but they lack the certificate of worst-case solving time—which, by the arguments above, is required for deploying NMPC in production environments. Here we first employ the Koopman operator to obtain a lifted high-dimensional linear predictor for the nonlinear dynamical system via data-driven models. Then, by condensing the Koopman-transformed MPC problem (i.e., eliminating the lifted high-dimensional states), the resulting problem becomes a box-constrained QP depending only on the control inputs.

2.1 Koopman and Extended Dynamic Mode Decomposition

Koopman (1931) and Koopman and Neumann (1932) proposed an alternative perspective grounded in operator theory to represent the uncontrolled discrete-time nonlinear dynamical system $x_{k+1} = f(x_k)$. Koopman demonstrated the existence of an infinite-dimensional linear operator $\mathcal{K}$, which advances the evolution of an infinite-dimensional Hilbert space of measurement functions $\psi(x)$ described as

$$\mathcal{K}\psi(x_k) \triangleq \psi(f(x_k)). \tag{2}$$

Since Koopman operator theory was described first for autonomous dynamical systems, numerous schemes (see Williams et al. (2016); Proctor et al. (2018); Korda and Mezić (2018)) have been proposed to extend the application of the Koopman operator to controlled systems of the form

$$x_{k+1} = f(x_k, u_k), \tag{3}$$

where $u_k \in \mathbb{R}^{n_u}$ denotes the control input of the system at time step k. To generalize the Koopman operator to (3), we adopt the scheme from Korda and Mezić (2018) which introduced an extended state vector as

$$\mathcal{X} = \begin{bmatrix} x \\ \mathbf{u} \end{bmatrix},$$

where $\mathbf{u} \triangleq \{u_i\}_{i=0}^{\infty} \in l(\mathcal{U})$ and $u_i \in \mathcal{U}$ represents the control input sequence and $l(\mathcal{U})$ denotes the space of all control input sequences $\mathbf{u}$. The dynamics of the extended state $\mathcal{X}$ are described as

$$f_{\mathcal{X}}(\mathcal{X}) = \begin{bmatrix} f(x, \mathbf{u}(0)) \\ \boldsymbol{S}\mathbf{u} \end{bmatrix},$$

where $\mathbf{u}(i)$ denotes the ith element of $\mathbf{u}$ and $\boldsymbol{S}$ represents the left shift operator, $(\boldsymbol{S}\mathbf{u})(i) \triangleq \mathbf{u}(i+1)$. Then the Koopman operator associated with the dynamics of the extended state can be defined on the set of extended observables $\phi(\mathcal{X})$ as

$$\mathcal{K}\phi(\mathcal{X}) \triangleq \phi(f_{\mathcal{X}}(\mathcal{X})).$$

The infinite-dimensional Koopman operator must be truncated in practice, and several finite-dimensional approximations have been proposed (see, e.g., Williams et al. (2015, 2016); Korda and Mezić (2018)) which employ a data-driven Extended Dynamic Mode Decomposition (EDMD) algorithm. In EDMD specifically, the set of extended observables is designed as the "lifted" mapping

$$\phi(x, \mathbf{u}) = \begin{bmatrix} \psi(x) \\ \mathbf{u}(0) \end{bmatrix}, \tag{4}$$

where $\psi(x) \triangleq \left[\psi_1(x), \cdots, \psi_{n_\psi}(x)\right]^\top$, n_ψ is the designed number of observables (with $n_\psi \gg n_x$), and $\mathbf{u}(0)$ denotes the first component of the sequence $\mathbf{u}$.

The EDMD approach expands the nonlinear observables $\phi(x, \mathbf{u})$ in a basis function set, e.g., Radial Basis Functions used in Korda and Mezić (2018), instead of directly solving for them via optimization. Only the Koopman operator is learned via an optimization procedure. In particular, the approximate Koopman operator identification problem is reduced to a least-squares problem, which assumes that the sampled data $\{(x_j, \mathbf{u}_j), (x_j^+, \mathbf{u}_j^+)\}\ \forall j = 1, \cdots, N_d$ are collected with the update mapping

$$\begin{bmatrix} x_j^+ \\ \mathbf{u}_j^+ \end{bmatrix} = \begin{bmatrix} f(x_j, \mathbf{u}_j(0)) \\ \boldsymbol{S}\mathbf{u}_j \end{bmatrix},$$

where the superscript + denotes the value at the next time step. Then an approximation of the Koopman operator, $\mathcal{A}$, is obtained by solving

$$J(\mathcal{A}) = \min_{\mathcal{A}} \sum_{j=1}^{N_d} \|\phi(x_j^+, \mathbf{u}_j^+) - \mathcal{A}\phi(x_j, \mathbf{u}_j)\|^2. \tag{5}$$

Since there is no need to predict the future control input sequence, the last n_u rows of $\mathcal{A}$ can be discarded. Additionally, let $\bar{\mathcal{A}}$ be the remaining part of $\mathcal{A}$ after discarding the part associated with the future control input. Then $\bar{\mathcal{A}}$ can be decomposed into $A \in \mathbb{R}^{n_\psi \times n_\psi}$ and $B \in \mathbb{R}^{n_\psi \times n_u}$ as

$$\bar{\mathcal{A}} = [A, B]$$

so that the problem (5) can be reduced to

$$J(A, B) = \min_{A,B} \sum_{j=1}^{N_d} \|\psi(x_j^+) - A\psi(x_j) - B\mathbf{u}_j(0)\|^2. \tag{6}$$

We finally obtain the identified linear predictor model in the "lifted" space as

$$\psi_{k+1} = A\psi_k + Bu_k, \tag{7}$$

where $\psi_k \triangleq \psi(x_k) \in \mathbb{R}^{n_\psi}$ denotes the lifted state space. Additionally, the output matrix C is obtained as the best projection of x onto the span of ψ in a least-squares sense, i.e., as the solution to

$$J(C) = \min_{C} \sum_{j=1}^{N_d} \|x_j - C\psi(x_j)\|^2. \tag{8}$$

At the end, a linear model for y can be formulated using

$$y_k = C\psi_k.$$

Remark 1. As Korda and Mezić (2018) claims, if the designed lifted mapping $\psi(x)$ contains the state x after the re-ordering $\psi(x) = [x^\top, \hat{\psi}(x)]^\top$, then the solution to (8) is $C = [I, 0]$.

2.2 Transforming NMPC to condensed MPC

After obtaining the approximate lifted predictor, NMPC can now be transformed into the MPC problem:

$$\begin{aligned} \min \quad & J(\hat{x}_t) = \tfrac{1}{2}\|C\psi_N - x_r\|^2_{W_N} \\ & \quad + \tfrac{1}{2}\sum_{k=0}^{N-1} \|u_k - u_r\|^2_{W_u} + \|C\psi_k - x_r\|^2_{W_x} \\ \text{s.t.} \quad & \psi_0 = \psi(\hat{x}_t), \\ & \psi_{k+1} = A\psi_k + Bu_k,\ k \in \mathbb{Z}_{0,N-1}, \\ & -e \le u_k \le e,\ k \in \mathbb{Z}_{0,N-1}, \end{aligned} \tag{9}$$

where the control inputs are assumed to have been scaled into the unit box constraints $[-e, e]$.

The main drawback of the Koopman operator is that the extremely high-dimensional lifted state space vector may increase the computational cost. This potential concern can be avoided by using the condensed MPC problem formulation. Define $z \triangleq \mathrm{col}(u_0, \cdots, u_{N-1}) \in \mathbb{R}^n$, where $n = N \times n_u$, $\bar{Q} \triangleq \mathrm{diag}(C^\top W_x C, \cdots, C^\top W_x C, C^\top W_N C)$, $\bar{R} \triangleq \mathrm{diag}(W_u, \cdots, W_u)$,

$$S \triangleq \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix},\ H \triangleq \bar{R} + S^\top \bar{Q} S.$$

These matrices are calculated off-line, so their computation cost is not included in the on-line computational cost. Then (9) is equivalently constructed as

$$z^* = \arg\min_z J(\hat{x}_t) = \tfrac{1}{2} z^\top H z + z^\top h \tag{10a}$$

$$\text{s.t. } -e \le z \le e, \tag{10b}$$

where

$$h \triangleq S^\top \bar{Q} g - \begin{bmatrix} W_u u_r \\ \vdots \\ W_u u_r \\ W_u u_r \end{bmatrix},\ g \triangleq \begin{bmatrix} A \\ \vdots \\ A^{N-2} \\ A^{N-1} \end{bmatrix} \psi_0 - \begin{bmatrix} C^\top W_x x_r \\ \vdots \\ C^\top W_x x_r \\ C^\top W_N x_r \end{bmatrix} \tag{11}$$

needs to be computed on-line since $\psi_0 = \psi(\hat{x}_t)$ and x_r, u_r are time-varying.

3. TIME-CERTIFIED IPM ALGORITHM

We solve the Box-QP (10) by adopting the path-following full-Newton Interior Point Method (IPM) with the exact number of iterations from our previous work Wu and Braatz (2023). Its Karush–Kuhn–Tucker (KKT) conditions are

$$Hz + h + \gamma - \theta = 0, \tag{12a}$$
$$z + \alpha - e = 0, \tag{12b}$$
$$z - \omega + e = 0, \tag{12c}$$
$$\gamma\alpha = 0, \tag{12d}$$
$$\theta\omega = 0, \tag{12e}$$
$$(\gamma, \theta, \alpha, \omega) \ge 0. \tag{12f}$$

The path-following IPM introduces a positive parameter τ to replace (12d) and (12e) by

$$\gamma\alpha = \tau^2 e, \tag{13a}$$
$$\theta\omega = \tau^2 e. \tag{13b}$$

It is well known that, as τ approaches 0, the path $(z_\tau, \gamma_\tau, \theta_\tau, \alpha_\tau, \omega_\tau)$ approaches a solution of (12).

The feasible path-following IPM algorithm has the best theoretical iteration complexity of $O(\sqrt{n})$. In addition, our algorithm is based on feasible IPM wherein all iterates lie in the strictly feasible set

$$\mathcal{F}^0 \triangleq \{(z,\gamma,\theta,\alpha,\omega)|\ (12a)\text{–}(12c)\ \text{satisfied}, (\gamma,\theta,\alpha,\omega) > 0\}.$$

3.1 Strictly feasible initial point

Our previous Wu and Braatz (2023) proposed a novel cost-free initialization strategy to find a strictly feasible initial point that also satisfies the specific conditions. First, an obvious strictly feasible initial point is

$$z^0 = 0, \quad \gamma^0 = \|h\|_\infty - \tfrac{1}{2}h, \quad \theta^0 = \|h\|_\infty + \tfrac{1}{2}h,$$

$$\alpha^0 = e, \quad \omega^0 = e,$$

where $\|h\|_\infty = \max\{|h_1|, |h_2|, \cdots, |h_n|\}$. It is straightforward to see that the above initial point strictly lies in $\mathcal{F}^0$.

Remark 2. (Initialization strategy). If $h = 0$, the optimal solution of problem (10) is $z^* = 0$; in the case of $h \neq 0$, we first scale the objective (10a) (which does not change the optimal solution) as

$$\min_z \tfrac{1}{2} z^\top \left(\frac{2\lambda}{\|h\|_\infty} H\right) z + z^\top \left(\frac{2\lambda}{\|h\|_\infty} h\right).$$

With the definitions $\tilde{H} = \frac{H}{\|h\|_\infty}$ and $\tilde{h} = \frac{h}{\|h\|_\infty}$, $\|\tilde{h}\|_\infty = 1$ and (12a) can be replaced by

$$2\lambda \tilde{H} z + 2\lambda \tilde{h} + \gamma - \theta = 0,$$

and the initial points

$$z^0 = 0,\ \gamma^0 = 1 - \lambda\tilde{h},\ \theta^0 = 1 + \lambda\tilde{h},\ \alpha^0 = e,\ \omega^0 = e, \quad (14)$$

can be adopted, where

$$\lambda = \frac{1}{\sqrt{n+1}}.$$

It is straightforward to verify that (14) lies in $\mathcal{F}^0$. The reason to use the scale factor $\frac{2\lambda}{\|h\|_\infty}$ is to make the initial point satisfy the neighborhood requirements, e.g., see (Wu and Braatz, 2023, Lemma 4).

3.2 Newton direction

Denote $v = \text{col}(\gamma, \theta) \in \mathbb{R}^{2n}$ and $s = \text{col}(\alpha, \omega) \in \mathbb{R}^{2n}$. Then replace (13a) and (13b) by $vs = \tau^2 e$ to obtain the new complementary condition

$$\sqrt{vs} = \sqrt{\tau^2 e}. \quad (15)$$

From *Remark 2*, $(z, v, s) \in \mathcal{F}^0$ and a direction $(\Delta z, \Delta v, \Delta s)$ can be obtained by solving the system of linear equations

$$2\lambda\tilde{H}\Delta z + \Omega \Delta v = 0, \quad (16a)$$

$$\Omega^\top \Delta z + \Delta s = 0, \quad (16b)$$

$$\sqrt{\frac{s}{v}}\Delta v + \sqrt{\frac{v}{s}}\Delta s = 2(\tau e - \sqrt{vs}), \quad (16c)$$

where $\Omega = [I, -I] \in \mathbb{R}^{n \times 2n}$. Letting

$$\Delta\gamma = \frac{\gamma}{\alpha}\Delta z + 2\left(\sqrt{\frac{\gamma}{\alpha}}\tau e - \gamma\right), \quad (17a)$$

$$\Delta\theta = -\frac{\theta}{\omega}\Delta z + 2\left(\sqrt{\frac{\theta}{\omega}}\tau e - \theta\right), \quad (17b)$$

$$\Delta\alpha = -\Delta z, \quad (17c)$$

$$\Delta\omega = \Delta z \quad (17d)$$

reduces (16) into a more compact system of linear equations,

$$\left(2\lambda\tilde{H} + \text{diag}\left(\frac{\gamma}{\alpha}\right) + \text{diag}\left(\frac{\theta}{\omega}\right)\right)\Delta z = 2\left(\sqrt{\frac{\theta}{\omega}}\tau e - \sqrt{\frac{\gamma}{\alpha}}\tau e + \gamma - \theta\right) \quad (18)$$

3.3 Iteration complexity and algorithm implementation

Let's denote $\beta \triangleq \sqrt{vs}$ and define the proximity measure

$$\xi(\beta, \tau) = \frac{\|\tau e - \beta\|}{\tau}. \quad (19)$$

Lemma 1. (See Wu and Braatz (2023)). Let $\xi := \xi(\beta, \tau) < 1$. Then the full Newton step is strictly feasible, i.e., $v_+ > 0$ and $s_+ > 0$.

Lemma 2. (See Wu and Braatz (2023)). After a full Newton step, let $v_+ = v + \Delta v$ and $s_+ = s + \Delta s$, then the duality gap is

$$v_+^T s_+ \leq (2n)\tau^2.$$

Lemma 3. (See Wu and Braatz (2023)). Suppose that $\xi = \xi(\beta, \tau) < 1$ and $\tau_+ = (1-\eta)\tau$ where $0 < \eta < 1$. Then

$$\xi_+ = \xi(\beta_+, \tau_+) \leq \frac{\xi^2}{1 + \sqrt{1-\xi^2}} + \frac{\eta\sqrt{2n}}{1-\eta}.$$

Furthermore, if $\xi \leq \frac{1}{\sqrt{2}}$ and $\eta = \frac{\sqrt{2}-1}{\sqrt{2n}+\sqrt{2}-1}$, then $\xi_+ \leq \frac{1}{\sqrt{2}}$.

Lemma 4. (See Wu and Braatz (2023)). The value $\xi(\beta, \tau)$ before the first iteration is denoted as $\xi^0 = \xi(\beta^0, (1-\eta)\tau^0)$. If $(1-\eta)\tau^0 = 1$ and $\lambda = \frac{1}{\sqrt{n+1}}$, then $\xi^0 \leq \frac{1}{\sqrt{2}}$ and $\xi(\beta, w) \leq \frac{1}{\sqrt{2}}$ are always satisfied.

Lemma 5. (See Wu and Braatz (2023)). Let $\eta = \frac{\sqrt{2}-1}{\sqrt{2n}+\sqrt{2}-1}$ and $\tau^0 = \frac{1}{1-\eta}$, Algorithm 1 exactly requires

$$\mathcal{N} = \left\lceil \frac{\log(\frac{2n}{\epsilon})}{-2\log(\frac{\sqrt{2n}}{\sqrt{2n}+\sqrt{2}-1})} \right\rceil + 1 \quad (20)$$

iterations, the resulting vectors being $v^\top s \leq \epsilon$.

Theorem 1. Let m_{lifting} denote the number of FLOP required by the lifting mapping. Then Algorithm 1 requires at most $m_{\text{lifting}} + (2Nn_\psi^2 + \frac{N(N+1)n_u n_\psi}{2} + Nn_x n_\psi + Nn_u^2 + 2n) + n + 5n + 3 + \mathcal{N}\big(1 + \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n + 2n^2 + 10n + 5n\big)$ FLOP.

Proof 1. In Algorithm 1: Step 1 takes m_{lifting} and $(2Nn_\psi^2 + \frac{N(N+1)n_u n_\psi}{2} + Nn_x n_\psi + Nn_u^2 + 2n)$ FLOP, which can be achieved by an efficient implementation of (11); Step 2 takes n FLOP to find the infinity norm of h; Step 3 takes $5n+3$ FLOP to assign the values for 5 vectors and 3 scalars. Each iteration of Step 4 takes in total $\big(1 + \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n + 2n^2 + 10n + 5n\big)$ FLOP.

Remark 3. By Theorem 1, the lifted high-dimensional states brought by the Koopman operator only slightly increase the on-line computation cost.

Algorithm 1 A time-certified IPM algorithm for input-constrained Koopman MPC (9)

Input: the current feedback states $\hat{x}_t$, the state reference signal x_r, the lifting mapping ψ, and the stopping tolerance ϵ; the required exact number of iterations $\mathcal{N} = \left\lceil \frac{\log(\frac{2n}{\epsilon})}{-2\log(\frac{\sqrt{2n}}{\sqrt{2n}+\sqrt{2}-1})} \right\rceil + 1$.

1. $\psi_0 \leftarrow \psi(\hat{x}_t)$ and calculate h from (11);
2. **if** $\|h\|_\infty = 0$, **return** $z^* = 0$; **otherwise**,
3. $(z, \gamma, \theta, \phi, \psi)$ are initialized from (14) where $\lambda \leftarrow \frac{1}{\sqrt{n+1}}$, $\eta \leftarrow \frac{\sqrt{2}-1}{\sqrt{2n}+\sqrt{2}-1}$ and $\tau \leftarrow \frac{1}{1-\eta}$;
4. **for** $k = 1, 2, \cdots, \mathcal{N}$ **do**
   4.1. $\tau \leftarrow (1-\eta)\tau$;
   4.2. solve (18) for Δz by using the Cholesky decomposition method with one forward substitution and one backward substitution;
   4.3. calculate $(\Delta\gamma, \Delta\theta, \Delta\alpha, \Delta\omega)$ from (17);
   4.4. $z \leftarrow z+\Delta z$, $\gamma \leftarrow \gamma+\Delta\gamma$, $\theta \leftarrow \theta+\Delta\theta$, $\alpha \leftarrow \alpha+\Delta\alpha$, $\omega \leftarrow \omega + \Delta\omega$;
5. **end**

4. NONLINEAR PDE CONTROL CASE STUDY

This section illustrates the effectiveness of our time-certified algorithm for a nonlinear PDE control example. The PDE plant under consideration is the nonlinear Korteweg-de Vries (KdV) equation that models the propagation of acoustic waves in plasma or shallow water waves (see Miura (1976)) as

$$\frac{\partial y(t,x)}{\partial t} + y(t,x)\frac{\partial y(t,x)}{\partial x} + \frac{\partial^3 y(t,x)}{\partial x^3} = u(t,x) \qquad (21)$$

where $x \in [-\pi, \pi]$ is the spatial variable. We consider the control input u to be $u(t,x) = \sum_{i=1}^{4} u_i(t)v_i(x)$, in which the four coefficients $\{u_i(t)\}$ are subject to the constraint $[-1, 1]$ and are computed by the model predictive controller, and $v_i(x)$ are predetermined spatial profiles given as $v_i(x) = e^{-25(x-m_i)^2}$, with $m_1 = -\pi/2$, $m_2 = -\pi/6$, $m_3 = \pi/6$, and $m_4 = \pi/2$.

The control objective is for the spatial profile $y(t,x)$ to track the given reference signal. In our closed-loop simulation, we discretize the x-axis of the nonlinear KdV equation at $N = 128$ nodes, and adopt a spectral method involving the Fourier transform and split stepping to solve the nonlinear KdV equation, e.g., see Meylan (2012). The sampling time is chosen as $\Delta t = 0.01$ s for data generation and the model predictive controller. The setting for our closed-loop simulation includes:

i) Data generation: The data are collected from 1000 simulation trajectories with 200 samples. At each simulation, the initial condition of the spatial profile is a random combination of four given spatial profiles, i.e., $y_1^0(0,x) = e^{-(x-\pi/2)^2}$, $y_2^0(0,x) = -\sin(x/2)^2$, $y_3^0(0,x) = e^{-(x+\pi/2)^2}$, $y_4^0(0,x) = \cos(x/2)^2$. The four control inputs $u_i(t)$ are distributed uniformly in $[-1, 1]$.

ii) Koopman predictor: We choose the lift function ψ consisting of the origin states (128 spatial nodes), the constant 1, the elementwise product of the origin states with one element shift, and the elementwise square of the origin states, which leads to the lifted state dimension $N_{lift} = 3 \times 128 + 1 = 385$. Then the lifted linear predictor with $A \in \mathbb{R}^{385\times385}$ and $B \in \mathbb{R}^{385\times4}$ is obtained from the Moore-Penrose pseudoinverse of the lifting data matrix, and its output matrix is $C = [I_{128}, 0] \in \mathbb{R}^{128\times385}$.

iii) MPC settings: Set the prediction horizon $N = 10$, the state cost matrix $W_x = W_N = I_{128}$, and the control inputs matrix $W_u = 0.01I_4$, and the control inputs are subject to $[-1, 1]$. The state references $x_r \in \mathbb{R}^{128}$ are piecewise constant taking the values $[0.5, 0.25, 0, 0.75] \times \text{ones}(128, 1)$ for a 50 s simulation time, with the control input reference $u_r = 0$.

Before the closed-loop simulation, we can calculate the worst-case total floating operations required at each sampling time. The dimension of the resulting Box-QP problem (10) is $n = 4 \times N = 40$, and we adopt the stopping criteria $\epsilon = 1 \times 10^{-6}$, so the required number of iterations is

$$\mathcal{N} = \left\lceil \frac{\log(\frac{2\times40}{1e-6})}{-2\log(\frac{\sqrt{2\times40}}{\sqrt{2\times40}+\sqrt{2}-1})} \right\rceil + 1 = 202.$$

Further, the FLOP for the lifting mapping is $m_{\text{lifting}} = 2N = 256$, and, by Theorem 1, Algorithm 1 exactly require $256 + (2\times10\times(385)^2 + 5\times11\times4\times385 + 10\times128\times385 + 10\times4\times385 + 2\times40) + 40 + 200 + 3 + 202\times(1 + 1/3(40)^3 + 1/2(40)^2 + 40/6 + 2(40)^2 + 600) = 0.0088 \times 10^9$ FLOP, which approximately leads to the execution time 0.0066 s on a machine with 1 GFLOP/s computing power (a trivial requirement for most processors today). Thus, we can get a certificate that the execution time will be less than the adopted sampling time $\Delta t = 0.01$ s.

In context, we ran our closed-loop simulation on a modern MacBook Pro with 2.7 GHz 4-core Intel Core i7 processors and 16GB RAM. Algorithm 1 is executed in MATLAB2023a via a C-mex interface. The number of iterations was exactly 202, and the maximum execution time was about 0.0075 s less than $\Delta t = 0.01$s. The closed-loop simulation results are plotted in Fig. 1, which shows that the MPC algorithm provides quick and accurate tracking of the spatial profile $y(t,x)$ to the given reference profile. The control inputs do not violate $[-1, 1]$, and also track the control input reference $u_r = 0$ well.

5. CONCLUSION

This article proposes a time-certified algorithm for input-constrained NMPC problems, in which Koopman operator is used to identify a lifted high-dimensional linear predictor model. The resulting small Box-QP is formulated by eliminating the lifted high-dimensional states, and finally, our previous time-certified algorithm (see Wu and Braatz (2023)) is applied to solve the Box-QP. Future work includes (i) improving the computation efficiency further while preserving the time-certified feature; and (ii)

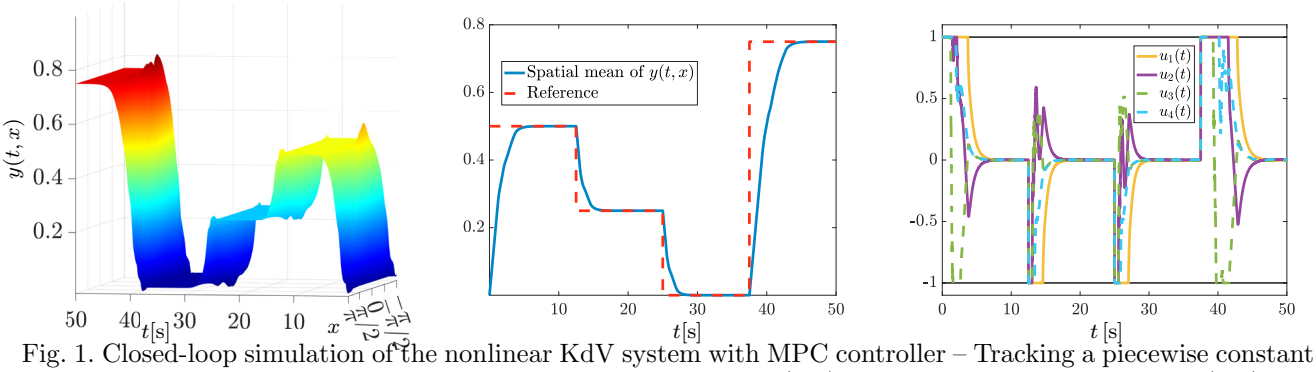

Fig. 1. Closed-loop simulation of the nonlinear KdV system with MPC controller – Tracking a piecewise constant spatial profile reference. Left: time evolution of the spatial profile $y(t,x)$. Middle: spatial mean of the $y(t,x)$. Right: the four control inputs.

extension to general NMPC problems while preserving the time-certified feature.

REFERENCES

Alessio, A. and Bemporad, A. (2009). A survey on explicit model predictive control. In L. Magni, D.M. Raimondo, and F. Allgöwer (eds.), *Nonlinear Model Predictive Control: Towards New Challenging Applications*, 345–369. Springer, Berlin–Heidelberg.

Arnström, D. and Axehill, D. (2019). Exact complexity certification of a standard primal active-set method for quadratic programming. In *Proceedings of the 58th IEEE Conference on Decision and Control*, 4317–4324.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Bemporad, A. and Patrinos, P. (2012). Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. *IFAC Proceedings Volumes*, 45(17), 14–20.

Cimini, G. and Bemporad, A. (2017). Exact complexity certification of active-set methods for quadratic programming. *IEEE Transactions on Automatic Control*, 62(12), 6094–6109.

Cimini, G. and Bemporad, A. (2019). Complexity and convergence certification of a block principal pivoting method for box-constrained quadratic programs. *Automatica*, 100, 29–37.

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

Giselsson, P. (2012). Execution time certification for gradient-based optimization in model predictive control. In *Proceedings of the 51st IEEE Conference on Decision and Control*, 3165–3170.

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2020). From linear to nonlinear MPC: Bridging the gap via the real-time iteration. *International Journal of Control*, 93(1), 62–80.

Koopman, B. (1931). Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5), 315–318.

Koopman, B. and Neumann, J. (1932). Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3), 255–263.

Korda, M. and Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93, 149–160.

Meylan, M. (2012). Numerical solution of the KdV. URL `wikiwaves.org/Numerical_Solution_of_the_KdV`.

Miura, R.M. (1976). The Korteweg–deVries equation: A survey of results. *SIAM Review*, 18(3), 412–459.

Okawa, I. and Nonaka, K. (2021). Linear complementarity model predictive control with limited iterations for box-constrained problems. *Automatica*, 125, 109429.

Proctor, J., Brunton, S., and Kutz, J. (2018). Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1), 909–930.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Richter, S., Jones, C.N., and Morari, M. (2011). Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, 57(6), 1391–1403.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4), 637–672.

Williams, M., Hemati, M., Dawson, S., Kevrekidis, I., and Rowley, C. (2016). Extending data-driven Koopman analysis to actuated systems. *IFAC-PapersOnLine*, 49(18), 704–709.

Williams, M., Kevrekidis, I., and Rowley, C. (2015). A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25, 1307–1346.

Wu, L. and Bemporad, A. (2023a). A construction-free coordinate-descent augmented-Lagrangian method for embedded linear MPC based on ARX models. *IFAC-PapersOnLine*, 56(2), 9423–9428.

Wu, L. and Bemporad, A. (2023b). A Simple and Fast Coordinate-Descent Aaugmented-Lagrangian Solver for Model Predictive Control. *IEEE Transactions on Automatic Control*, 68(11), 6860–6866. doi: 10.1109/TAC.2023.3241238.

Wu, L. and Braatz, R.D. (2023). A direct optimization algorithm for input-constrained MPC. *arXiv preprint arXiv:2306.15079*.

Zavala, V.M. and Biegler, L.T. (2009). The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1), 86–93.