Solving Nonlinear MPC Problems in the Koopman Lifted Space: The Case Study of Mobile Robot Navigation in Cluttered Environments

Lorenzo Calogero, Andrea Usai, and Alessandro Rizzo

I. Introduction

The Koopman operator framework allows to transform nonlinear dynamical systems into equivalent linear ones within a higher-dimensional state space. Its application can be extended to nonlinear optimal control problems, enabling their efficient solution in the Koopman lifted space [1].

Here, we present a comprehensive analytical framework to lift general Nonlinear Model Predictive Control (NMPC) problems in the Koopman space, converting them into equivalent quadratic programs (QPs) – referred to as Koopman NMPC (K-NMPC) – that can be solved with superior computational performance [2].

Moreover, we advance analytical Koopman operator methods by proposing an algorithmic procedure to generate an invariant basis of Koopman observables to lift both the nonlinear prediction model and the nonlinear state constraints of NMPC; additionally, we present a general method to arbitrarily reduce the dimensionality of the Koopman lifted space, lowering the K-NMPC complexity and handling the infinite-dimensional case.

Our K-NMPC approach is validated through hardware-in-the-loop experiments on the case study of mobile robot navigation in cluttered environments, showcasing its solid performance and a ten-fold reduction in computation times.

II. Koopman Operator Theory

Considering a continous-time input-affine nonlinear dynamical system, evolving in the state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$,

$$\dot{x}(t) = f_0(x(t)) + \textstyle\sum_{j=1}^{n_u} g_j(x(t))u_j(t), \quad x(0) = x_0, \tag{1}$$

the Koopman operator allows to transform it into an equivalent bilinear system (called Koopman lifted system), evolving in a higher-dimensional state space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$, $n_z \gg n_x$, i.e.,

$$\dot{z}(t) = Az(t) + \textstyle\sum_{j=1}^{n_u} B_j z(t)u_j(t), \quad z(0) = \phi(x_0), \tag{2}$$

where $\mathcal{Z}$ is called lifted state space, with lifted state z. Systems (1) and (2) are put in relation through a suitable basis of functions ϕ, called observables, i.e.,

$$\Phi = \{\phi_i(x)\}_{i=1}^{N_o} \subset \mathscr{F}, \quad \phi(x) = [\phi_i(x)]_{i=1}^{N_o}, \tag{3}$$

where $\phi_i : \mathcal{X} \to \mathbb{R}$, $N_o = n_z$, and $\mathscr{F} \subseteq \mathcal{C}^1$ is a Banach space of continuously differentiable functions [3]–[5].

Let $x(t) = \varphi^t(x_0, \boldsymbol{u})$, with $\boldsymbol{u} = [u(t)]_{t\in[0,+\infty)}$, be the flow of system (1). The Koopman operator $\mathcal{K}^{t,\boldsymbol{u}} : \mathscr{F} \to \mathscr{F}$ is defined as

$$\mathcal{K}^{t,\boldsymbol{u}}\phi(\cdot) = \phi \circ \varphi^t(\cdot, \boldsymbol{u}), \quad \forall\, \phi \in \mathscr{F}, \tag{4}$$

which can be interpreted as the evolution of the observables ϕ along the trajectories $x(t)$ of system (1), yielding the lifted trajectories $z(t) = \phi(x(t))$. The dynamics of the lifted trajectories is given by

$$\dot{z} = \dot{\phi}(x) = \tfrac{\partial}{\partial x}\phi(x)\,\dot{x} = \nabla_x\phi(x)\,f(x,u) = L_f\phi(x)$$

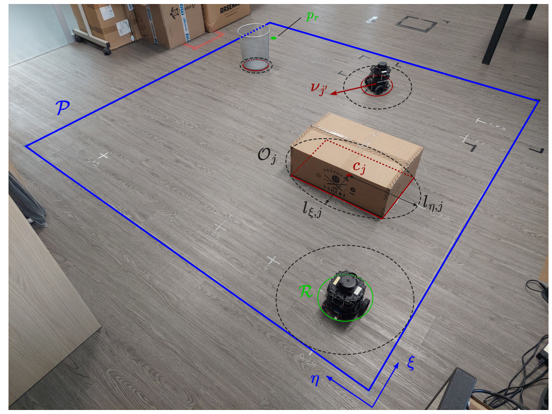

Fig. 1. Experimental setup.

$$= L_{f_0}\phi(x) + \textstyle\sum_{j=1}^{n_u} L_{g_j}\phi(x)u_j, \tag{5}$$

where $L_\star = \star\cdot\nabla$ is the Lie derivative. Eq. (5) can be directly related to the Koopman lifted system (2) under the following Assumption:

Assumption 1. *For system* (1) *and the basis* Φ (3)*, it holds that*

$$L_{f_0}\phi(x) \in \mathrm{span}(\Phi), \quad L_{g_j}\phi(x) \in \mathrm{span}(\Phi) \tag{6}$$

with $j = 1, \dots, n_u$ *and* $\forall\, \phi \in \Phi$.

Then, the following Theorem establishes the direct relation between systems (1) and (2):

Theorem 1 ([1, Theorem 1])**.** *Let system* (1) *and the basis of observables* $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3) *be given. Let* $x(t)$ *be the solution of system* (1) *and* $z(t)$ *the solution of the Koopman lifted system* (2)*. If Assumption 1 is satisfied, then the two solutions are equivalent through the map* $z(t) = \phi(x(t))$, $x(t) = \phi^{-1}(z(t))$, $\forall\, t \geq 0$.

Remark 1. *For Assumption 1 to hold, an infinite-dimensional basis* Φ *may be needed, yielding an infinite-dimensional Koopman lifted system* (2)*, i.e.,* $|\Phi| = N_o = n_z \in \mathbb{Z}_{>0} \cup \{+\infty\}$.

A. Generating the Basis of Observables

A key challenge in the Koopman operator framework is to analytically derive the basis Φ (3) for system (1) satisfying Assumption 1. Such a task can be performed by relying on the novel procedure proposed in [1, Algorithm 1], which constructs Φ in an iterative way, starting from an initial hand-picked set Φ_{in} and by progressively evaluating the lifted dynamics through Eq. (5).

B. Dimensionality Reduction of the Koopman Lifted System

As noted in Remark 1, an infinite-dimensional basis Φ (3) may be needed to exactly lift system (1).

To reduce the Koopman lifted system (2) to a finite dimension of choice, we set the maximum number of observables $\overline{N}_o < N_o$ to be generated by [1, Algorithm 1]; then, the lifted equations associated with the lastly generated observables will contain a residual nonlinear term $f_{\text{res}}(z, u)$, which can be relaxed through its first-order Taylor expansion around $(\overline{z}, \overline{u}) \in \mathcal{Z} \times \mathcal{U}$, yielding a bilinear parameter-varying reduced lifted system [1], i.e.,

$$\dot{z} = A(\overline{z}, \overline{u})z + B(\overline{z}, \overline{u})u + \textstyle\sum_{j=1}^{n_u} B_j z u_j + b(\overline{z}, \overline{u}). \tag{7}$$

Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy (e-mails: {lorenzo.calogero, andrea.usai, alessandro.rizzo}@polito.it).

L. Calogero was supported by the European Union NextGenerationEU (NGEU)–Piano Nazionale di Ripresa e Resilienza (PNRR) Project and the Italian Ministry of University and Research (MUR) under Grant 352/2022. A. Usai and A. Rizzo were supported by the Sustainable Mobility National Research Center (MOST) and the European Union NGEU–PNRR Project (Mission 4, Component 2, Investment 1.4–D.D. 1033 17/06/2022) under Grant CN00000023.

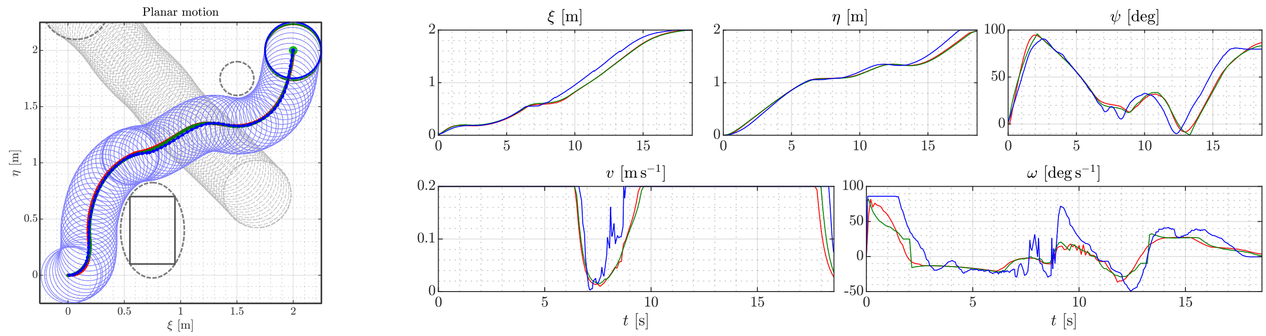

Fig. 2. Planar motion and closed-loop trajectories of the ego robot within the cluttered environment: NMPC ——; K-NMPC (simulation) ——; K-NMPC (real) ——.

III. Koopman NMPC: Lifting Nonlinear MPC into the Linear Koopman Space

In the following, we extend the Koopman operator framework to optimization problems. Specifically, we transform a generic NMPC problem in the state x – comprising nonlinear prediction model and nonlinear state constraints – into an equivalent quadratic program (QP) in the lifted state z (K-NMPC) [1].

The NMPC problem is formulated as follows, for each $k \geq 0$:

$$\min_{\hat{x},\hat{u}} \; J(\hat{x},\hat{u}) \tag{8a}$$

$$\text{s.t.} \; \hat{x}_0 = x_k, \quad \hat{x}_{i+1} = f^d(\hat{x}_i,\hat{u}_i), \tag{8b}$$

$$\hat{u}_i \in \mathcal{U}, \quad \hat{x}_i \in \mathcal{X}, \quad h_i^d(\hat{x}_i, c_k, \nu_k) \leq 0, \tag{8c}$$

$$i = 0,\ldots,N_p-1,$$

$$J(\hat{x},\hat{u}) = \textstyle\sum_{i=0}^{N_p-1} \|\hat{x}_i - x_{r,k+i}\|_Q^2 + \|\hat{u}_i\|_R^2 + \textstyle\sum_{i=1}^{N_p-1} \|\hat{x}_i - \hat{x}_{i-1}\|_{Q_\Delta}^2 + \|\hat{u}_i - \hat{u}_{i-1}\|_{R_\Delta}^2. \tag{8d}$$

In the following, we shall focus on the practical case study of mobile robot navigation in cluttered environments (Fig. 1).

A. Nonlinear Prediction Model: Mobile Robot

As prediction model of plant (1) (i.e., the ego mobile robot), we employ the kinematic unicycle model, i.e.,

$$\dot{x}(t) = f(x(t),u(t)) = [v(t)\cos\psi(t),\; v(t)\sin\psi(t),\; \omega(t)]^\top, \tag{9}$$

where $x = [\xi,\eta,\psi]^\top$ and $u = [v,\omega]^\top$, being (ξ,η) the robot planar position, ψ the heading angle, v the linear velocity, and ω the angular velocity. Eq. (8b) is the discrete-time version of model (9).

B. Nonlinear State Constraints and Observables Generation: Cluttered Environment

Nonlinear state constraints are given by the obstacles – either stationary or moving – that the ego robot has to avoid. We assume that each j-th obstacle can be fully enclosed in a safety ellipsoid, yielding the following constraints:

$$1 - \frac{(\xi - c_{\xi,j}(t) - \nu_{\xi,j}(t)\tau)^2}{(l_{\xi,j}+\alpha_j r)^2} - \frac{(\eta - c_{\eta,j}(t) - \nu_{\eta,j}(t)\tau)^2}{(l_{\eta,j}+\alpha_j r)^2} \leq 0$$

$$\Rightarrow h_{j,\tau}(x, c_j(t), \nu_j(t)) \leq 0, \quad j = 1,\ldots,N_{\text{obst}}, \tag{10}$$

where N_{obst} is the number of obstacles; $c_j(t) = [c_{\xi,j}(t), c_{\eta,j}(t)]^\top$ and $(l_{\xi,j}, l_{\eta,j})$ are the j-th ellipsoid center and semi-axes, respectively; $\nu_j(t) = \dot{c}_j(t)$; r is the radius of the ego robot; α_j is a safety margin. Eq. (8c) is the discrete-time version of constraints (10).

From Eqs. (9) and (10), we can define the initial set of observables $\Phi_{\text{in}} = \{\xi,\eta,\psi,\xi^2,\eta^2\}$, which includes the system states and the nonlinear terms ($[\xi^2,\eta^2]^\top = \sigma(x)$) introduced by Eq. (10).

The complete basis Φ is generated from Φ_{in} through [1, Algorithm 1], obtaining a finite-dimensional basis of 14 observables,

$$\Phi = \Phi_{\text{in}} \cup \{\mathrm{c}_\psi, \mathrm{s}_\psi, \xi\mathrm{c}_\psi, \eta\mathrm{s}_\psi, \mathrm{c}_\psi^2, \xi\mathrm{s}_\psi, \mathrm{s}_\psi^2, \eta\mathrm{c}_\psi, \mathrm{c}_\psi\mathrm{s}_\psi\}, \tag{11}$$

where $\mathrm{c}_\psi = \cos\psi$ and $\mathrm{s}_\psi = \sin\psi$. Thus, system (9) with constraints (10) admits an exact finite-dimensional Koopman lifted system, with dimension $n_z = |\Phi| = 14$, and lifted state $z = [z_x^\top, z_\sigma^\top, \cdots]^\top = \phi(x) = [x^\top, \sigma(x)^\top, \cdots]^\top$.

C. Lifted System Reduction and Linearization

The Koopman lifted system arising from Eqs. (9)-(11) can be further reduced in dimension (so to have a less complex K-NMPC problem) as in Sec. II-B. Also, its bilinearity (7) is linearized around $(\overline{z},\overline{u}) = (z_k, \hat{u}^*_{1|k-1})$ to obtain a discrete-time linear parameter-varying prediction model, i.e., $z_{k+1} = A_d(\overline{z},\overline{u})z_k + B_d(\overline{z},\overline{u})u_k + b_d(\overline{z},\overline{u}) = A_{d,k}z_k + B_{d,k}u_k + b_{d,k}$.

D. Koopman NMPC Formulation

The resulting K-NMPC optimal control problem is given by

$$\min_{\hat{z},\hat{u}} \; J(\hat{z}_x,\hat{u}) \tag{12a}$$

$$\text{s.t.} \; \hat{z}_0 = \phi(x_k), \quad \hat{z}_{i+1} = A_{d,k}\hat{z}_i + B_{d,k}\hat{u}_i + b_{d,k}, \tag{12b}$$

$$\hat{u}_i \in \mathcal{U}, \quad \hat{z}_{x,i} \in \mathcal{X}, \quad C_i(c_k,\nu_k)\hat{z}_{\sigma,i} \leq d_i(c_k,\nu_k), \tag{12c}$$

$$i = 0,1,\ldots,N_p-1,$$

where the prediction model (12b) and constraints (12c) are linear thanks to the Koopman lifting. The obtained QP-MPC (12) is equivalent to NMPC (8) (albeit with the relaxations introduced in Sec. II-B and III-C) through the map $z = \phi(x)$.

IV. Experimental Results

Our K-NMPC approach is validated through hardware-in-the-loop experiments on a real differential wheeled mobile robot, tasked to reach a target position within a cluttered environment, populated by stationary and moving obstacles (Fig. 1).

Results are reported in Fig. 2, showing that K-NMPC manages to successfully attain the control task, effectively avoiding all the obstacles, with remarkably similar trajectories to NMPC. K-NMPC execution time is within $[1.54, 3.57]$ ms (average: 2.26 ms), while NMPC achieves $[20.03, 55.29]$ ms (average: 27.92 ms); thus, K-NMPC outperforms NMPC by over an order of magnitude.

References

[1] L. Calogero, M. Boggio, C. Novara, and A. Rizzo, "A General Analytical Framework for Fast Solving Nonlinear MPC Problems in the Linear Koopman Space," in *Proc. Eur. Control Conf.*, 2025, pp. 743–748.

[2] L. Calogero, M. Pagone, and A. Rizzo, "Enhanced Quadratic Programming via Pseudo-Transient Continuation: An Application to Model Predictive Control," *IEEE Control Syst. Lett.*, vol. 8, pp. 1661–1666, June 2024.

[3] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control," *PLoS One*, vol. 11, no. 2, 2016.

[4] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman Theory to Allow for Inputs and Control," *SIAM J. Appl. Dyn. Syst.*, vol. 17, no. 1, pp. 909–930, 2018.

[5] A. Mauroy, I. Mezić, and Y. Susuki, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications.* Cham, Switzerland: Springer, 2020.