

Immersion-based model predictive control of constrained nonlinear systems: Polyflow approximation

Zheming Wang and Raphaël M. Jungers

Abstract—In the framework of Model Predictive Control (MPC), the control input is typically computed by solving optimization problems repeatedly online. For general nonlinear systems, the online optimization problems are non-convex and computationally expensive or even intractable. In this paper, we propose to circumvent this issue by computing a high-dimensional linear embedding of discrete-time nonlinear systems. The computation relies on an algebraic condition related to the immersibility property of nonlinear systems and can be implemented offline. With the high-dimensional linear model, we then define and solve a convex online MPC problem. We also provide an interpretation of our approach under the Koopman operator framework.

I. INTRODUCTION

Model Predictive Control (MPC) is a powerful control technology for general constrained systems due to its ability to handle hard constraints on control and states. The control action is determined online repeatedly by optimizing the future evolution of the trajectories of the system with the given cost function and constraints, see [1]. For general nonlinear systems, the online optimization problem to be solved is often non-convex and computationally expensive. To tackle this issue, many nonlinear MPC algorithms have been proposed (we refer the reader to the survey papers [2], [3]). In this paper, we intend to alleviate the online computation complexity by using linear embedding of nonlinear systems. More precisely, we use the state immersion method [4]–[6] to obtain a linear model with a higher dimension and convert the original non-convex MPC problem into a convex problem with this linear model.

While the immersion method can be equivalent to feedback linearization in the special cases where the immersion is a state diffeomorphism, they are in general different as an immersion does not necessarily preserve the dimension of the system. Hence, the immersion method has more freedom to find linear representations of nonlinear systems. It has been used in stabilization and observer design for nonlinear systems [7], [8]. Immersion is intrinsically related with the so-called “polynomial vector flows” or *polyflows* for continuous-time systems, see, e.g., [9]–[11]. Recently, a new immersion technique has been proposed in [12] for continuous-time

autonomous systems using *polyflows*. This technique is then extended to discrete-time autonomous systems in [13], which presents an immersibility algebraic condition that can be computed in a data-driven fashion. In this paper, we derive a similar immersibility algebraic condition for control systems, which again allows to compute an immersion efficiently.

Let us mention that linear embedding of nonlinear systems has already been used for analysis and control under the Koopman operator framework, see [14]–[16] and the references therein. The linear model is typically obtained by computing a truncation of the Koopman operator using a finite number of observables or basis functions. While our proposed *polyflow* approximation can be viewed as a Koopman-based lifting approach with a special set of basis functions, it is derived from an immersibility algebraic condition and is equipped with an algorithmic way of determining the basis functions. See Section IV for more developments on the link between this work and Koopman-based approaches.

The rest of the paper is organized as follows. The next section reviews some preliminary results on immersion and gives the problem statement. Section III presents an immersibility algebraic condition for control systems and discusses optimal control using linear embedding. In Section IV, we present a computational procedure for determining a linear model. Numerical experiments are provided in Section V.

Notation. The non-negative integer set is indicated by \mathbb{Z}^+ . I_n is the $n \times n$ identity matrix and $\mathbf{0}_{n \times m}$ is the $n \times m$ matrix of all zeros (subscript omitted when the dimension is clear). \mathbb{B}_n is the unit closed ball in \mathbb{R}^n . $\|x\|_p$ denotes the ℓ_p -norm of x ($\|x\| = \|x\|_2$ by default). Given a map T , let $T(X)$ denote $\{T(x) : x \in X\}$ and $T^{-1}(Y)$ denote the preimage of the set Y under the map T , i.e., $T^{-1}(Y) := \{x : T(x) \in Y\}$ (T is not necessarily invertible).

II. PRELIMINARIES AND PROBLEM STATEMENT

We consider the following discrete-time nonlinear system

$$x(t+1) = f(x(t), u(t)), \quad t \in \mathbb{Z}^+ \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input, and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is some continuous function. Let \mathbf{u} denote a control sequence $\{u(0), u(1), \dots\}$ and $\phi(t; x, \mathbf{u})$ denote the solution of (1) with a initial state x at $t = 0$ and a control sequence \mathbf{u} .

A. Immersion

This paper is concerned with linear embedding of the nonlinear system (1). More precisely, we aim to immerse

The authors are with the ICTEAM Institute, UCLouvain, Louvain-la-Neuve, 1348, Belgium.

Raphaël M. Jungers is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 864017 - L2C. Raphaël M. Jungers is also supported by the Walloon Region and the Innoviris Foundation.

Email addresses: zheming.wang@uclouvain.be (Zheming Wang), raphael.jungers@uclouvain.be (Raphaël M. Jungers)

System (1) into a linear system ($\tilde{n} > n$):

$$\Sigma(A, B, C) : \quad \begin{aligned} \tilde{x}(t+1) &= A\tilde{x}(t) + Bu(t), \\ \tilde{y}(t) &= C\tilde{x}(t), t \in \mathbb{Z}^+ \end{aligned} \quad (2)$$

where $\tilde{x}(t) \in \mathbb{R}^{\tilde{n}}$, $\tilde{y}(t) \in \mathbb{R}^n$, $A \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, and $C \in \mathbb{R}^n \times \tilde{n}$. Similarly, let $\tilde{\phi}(t; \tilde{x}, \mathbf{u})$ denote the solution of (2) with an initial state \tilde{x} (at time $t=0$) and a control sequence \mathbf{u} .

Let us first recall the notion of immersibility, see, e.g., [4]–[6].

Definition 1: System (1) is immersible into $\Sigma(A, B, C)$ given in (2) if there exists a map $T : \mathbb{R}^n \rightarrow \mathbb{R}^{\tilde{n}}$ such that, for any initial state $x \in \mathbb{R}^n$ and control sequence \mathbf{u} , $\phi(t; x, \mathbf{u}) = C\tilde{\phi}(t; T(x), \mathbf{u})$ for all $t \in \mathbb{Z}^+$, where $\phi(t; x, \mathbf{u})$ is the solution of (1) and $\tilde{\phi}(t; T(x), \mathbf{u})$ is the solution of (2).

From this definition, it can be verified that System (1) is immersible into $\Sigma(A, B, C)$ if there exist a map $T : \mathbb{R}^n \rightarrow \mathbb{R}^{\tilde{n}}$ and matrices $A \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $B \in \mathbb{R}^{\tilde{n} \times m}$ and $C \in \mathbb{R}^n \times \tilde{n}$ such that, $\forall(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$,

$$T(f(x, u)) = AT(x) + Bu, \quad x = CT(x). \quad (3)$$

A necessary and sufficient condition of immersibility is given in [6]. To state this immersibility condition, we bring in the notation in [6]. Let

$$\hat{f}^{\ell+1}(x, u) = f(\hat{f}^\ell(x, u), 0), \ell \geq 1. \quad (4)$$

with $\hat{f}^1(x, u) = f(x, u)$ and $\hat{f}^0(x, u) = x$. Define the space of \mathbb{R} -valued functions

$$\Phi = \text{span}_{\mathbb{R}}\{x_1, \dots, x_n, \hat{f}_1^1(x, 0), \dots, \hat{f}_n^1(x, 0), \dots, \hat{f}_1^k(x, 0), \dots, \hat{f}_n^k(x, 0), \dots\}, \quad (5)$$

where $\hat{f}_i^k(x, 0)$ is the i^{th} component of $\hat{f}_n^k(x, 0)$ for $i = 1, 2, \dots, n$. Then, we have the following characterization of immersible systems:

Theorem 1 ([6]): System (1) is immersible into a linear system in form of (2) if and only if: i) Φ , defined as in (5), is finite-dimensional; 2) $D_{u^T} \hat{f}^\ell(x, u)$ is constant (independent of x and u) for all $\ell \geq 1$, where $\hat{f}^\ell(x, u)$ is defined as in (4).

B. Problem statement

The goal of this paper is to design controllers using the linear embedding in (2). We consider constrained nonlinear systems with state and input constraints as follows:

$$x(t) \in X, \quad u(t) \in U, \quad t \in \mathbb{Z}^+ \quad (6)$$

where $X \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^m$ are the given state and input constraint sets. The control objective is to steer the state to the origin while fulfilling the constraints in (6). This can be accomplished under the MPC framework. Given the current state $x(t)$ at time t , we define the MPC problem as follow:

$$\min_{\mathbf{u}_N} \sum_{i=0}^{N-1} l(x_i, u_i) + l_f(x_N) \quad (7a)$$

$$\text{s.t.} \quad x_{i+1} = f(x_i, u_i), x_i \in X, u_i \in U, \quad (7b)$$

$$x_0 = x(t), x_N \in X_f, i = 0, 1, \dots, N-1, \quad (7c)$$

where N is the horizon length, $\mathbf{u}_N := \{u_0, u_1, \dots, u_{N-1}\}$, $l(x, u)$ is the stage cost, $l_f(x)$ is the terminal cost, and $X_f \subseteq X$ is some appropriate terminal set. Once the problem above is solved, the first element of the control sequence is applied to the system and this process is repeated. Under some condition, recursive feasibility and stability is guaranteed, see, e.g., [1]. While there are well-known nonlinear MPC techniques, see [2], [3], solving the nonlinear MPC problem online is computationally expensive in general. In this paper, we propose to use a lifted linear model (which is computed offline) in the online MPC problem to reduce computational burden, motivated by the immersibility property. Eventually, we migrate the complexity of the online computation of nonlinear optimization problems to the offline computation of linear embedding.

III. OPTIMAL CONTROL VIA IMMERSION

In this section, we discuss optimal control of nonlinear systems when a linear immersion is available.

A. Discrete-time polyflows

We first show a nilpotency property of immersible discrete-time nonlinear systems. For continuous-time autonomous systems, when a system is immersible to a linear system, it enjoys a Lie derivative nilpotency property and the solution of the system is called a *polyflow*, see [10]. Based on this nilpotency property, a linearization technique called *polyflow* approximation has been proposed in [12] and it is proved that this approximation exhibits better convergence properties than Taylor approximation. While *polyflows* are defined for continuous-time systems, the initial definition of [12] can be generalized to discrete-time systems, see [13]. In this paper, we further generalize the concept of *polyflows* for control nonlinear systems in the form of (1).

Similar to the Lie derivative nilpotency property for continuous-time, we also derive a nilpotency condition for immersibility in discrete-time, which allows to use powerful linear optimal control techniques [17] for nonlinear systems. Before we present the nilpotency condition, the following lemma is needed.

Lemma 1: Consider System (1), let $\hat{f}^\ell(x, u)$ be defined as in (4) for all $\ell \in \mathbb{Z}^+$. Then, it holds that $\hat{f}^k(f(x, u), 0) = \hat{f}^{k+1}(x, u)$ for all $k \in \mathbb{Z}^+$.

Proof: The proof goes by induction. Suppose $\hat{f}^k(f(x, u), 0) = \hat{f}^{k+1}(x, u)$ holds for some integer $k \geq 0$. Then, $\hat{f}^{k+1}(f(x, u), 0) = f(\hat{f}^k(f(x, u), 0), 0) = f(\hat{f}^{k+1}(x, u), 0) = \hat{f}^{k+2}(x, u)$, from the definition in (4). It is obvious that $\hat{f}^0(f(x, u), 0) = f(x, u) = \hat{f}^1(x, u)$. \square

Inspired the Lie derivative nilpotency property in [10], we modify Theorem 1 and obtain a nilpotency condition for immersibility of discrete-time nonlinear systems.

Theorem 2: Consider System (1), let $\hat{f}^\ell(x, u)$ be defined as in (4) for all $\ell \in \mathbb{Z}^+$. System (1) is immersible into a linear system in form of (2) if and only if there exist a finite

$k \in \mathbb{Z}^+$, $\{\alpha_\ell \in \mathbb{R}^{n \times n}\}_{\ell=0}^k$ and $\{b_\ell \in \mathbb{R}^{n \times m}\}_{\ell=0}^k$ such that

$$\hat{f}^{k+1}(x, 0) = \sum_{\ell=0}^k \alpha_\ell \hat{f}^\ell(x, 0), \quad (8)$$

$$\hat{f}^\ell(x, u) = \hat{f}^\ell(x, 0) + b_\ell u, \ell = 1, 2, \dots, k+1. \quad (9)$$

Proof: (Necessity) Suppose System (1) is immersible into a linear system in the form of (2). From Theorem 1, Φ is finite-dimensional and there exists a $k \in \mathbb{Z}^+$ such that $\{x_1, \dots, x_n, \hat{f}_1^1(x, 0), \dots, \hat{f}_1^k(x, 0), \dots, \hat{f}_n^k(x, 0)\}$ spans Φ . Hence, there exist $\{\alpha_\ell \in \mathbb{R}^{n \times n}\}_{\ell=0}^k$ such that $\hat{f}^{k+1}(x, 0) = \sum_{\ell=0}^k \alpha_\ell \hat{f}^\ell(x, 0)$. Again, from Theorem 1, $D_{u^T} \hat{f}^\ell(x, u)$ is constant for all $\ell \geq 1$. Let $b^\ell = D_{u^T} \hat{f}^\ell(x, u)$ for $\ell \geq 1$. Thus, $\hat{f}^\ell(x, u)$ can be written as $\hat{f}^\ell(x, u) = \hat{f}^\ell(x, 0) + b^\ell u$ for all $\ell \geq 1$.

(Sufficiency) Suppose (8) – (9) hold. From Definition 1, to show that System (1) is immersible into a linear system, we only need to show that there exist a map $T: \mathbb{R}^n \rightarrow \mathbb{R}^{\tilde{n}}$ and matrices $A \in \mathbb{R}^{\tilde{n} \times \tilde{n}}, B \in \mathbb{R}^{\tilde{n} \times m}$ and $C \in \mathbb{R}^{n \times \tilde{n}}$ such that (3) hold. Let

$$T(x) := \begin{pmatrix} x \\ \hat{f}^1(x, 0) \\ \vdots \\ \hat{f}^k(x, 0) \end{pmatrix}. \quad (10)$$

From (8), we have

$$\begin{aligned} & \begin{pmatrix} \hat{f}^1(x, 0) \\ \hat{f}^2(x, 0) \\ \vdots \\ \hat{f}^{k+1}(x, 0) \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{0} & I_n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & I_n \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{k-1} & \alpha_k \end{pmatrix} \begin{pmatrix} x \\ \hat{f}^1(x, 0) \\ \vdots \\ \hat{f}^k(x, 0) \end{pmatrix} \\ &:= AT(x). \end{aligned} \quad (11)$$

With this and Lemma 1, we can get

$$\begin{aligned} T(f(x, u)) &= \begin{pmatrix} f(x, u) \\ \hat{f}^1(f(x, u), 0) \\ \vdots \\ \hat{f}^k(f(x, u), 0) \end{pmatrix} = \begin{pmatrix} \hat{f}^1(x, u) \\ \hat{f}^2(x, u) \\ \vdots \\ \hat{f}^{k+1}(x, u) \end{pmatrix} \\ &\stackrel{(9)}{=} \begin{pmatrix} \hat{f}^1(x, 0) + b^1 u \\ \hat{f}^2(x, 0) + b^2 u \\ \vdots \\ \hat{f}^{k+1}(x, 0) + b^{k+1} u \end{pmatrix} \\ &= AT(x) + \begin{pmatrix} b^1 \\ b^2 \\ \vdots \\ b^{k+1} \end{pmatrix} u \\ &:= AT(x) + Bu \end{aligned} \quad (12)$$

Therefore, (3) is satisfied with $C = [I_n \ 0_{n \times kn}]$. \square

The nilpotency condition above is in a similar form of the Lie derivative nilpotency property for *polyflows* in [10]. Hence, analogously, we say that the solution of System (1) satisfying (8) – (9) is a *polyflow* when the condition in Theorem 2 is satisfied.

To show the equivalence to Theorem 1, we derive the following corollary.

Corollary 1: Consider System (1), let $\hat{f}^\ell(x, u)$ be defined as in (4) for all $\ell \in \mathbb{Z}^+$. If there exist a finite $k \in \mathbb{Z}^+$, $\{\alpha_\ell \in \mathbb{R}^{n \times n}\}_{\ell=0}^k$ and $\{b_\ell \in \mathbb{R}^{n \times m}\}_{\ell=0}^k$ such that (8) – (9) hold, then, $D_{u^T} \hat{f}^\ell(x, u)$ is constant for all $\ell \geq 1$.

Proof: With (8) – (9), we only need to show that $D_{u^T} \hat{f}^{k+2}(x, u)$ is constant. From Lemma 1, we have $\hat{f}^{k+2}(x, u) = \hat{f}^{k+1}(f(x, u), 0) = \sum_{i=0}^k \alpha_i \hat{f}^i(f(x, u), 0) = \sum_{\ell=0}^k \alpha_\ell \hat{f}^{\ell+1}(x, u) = \sum_{\ell=0}^k \alpha_\ell (\hat{f}^{\ell+1}(x, 0) + b_{\ell+1} u)$. Hence, $D_{u^T} \hat{f}^{k+2}(x, u)$ is constant. \square

While Theorem 2 is a modification of Theorem 1 (originated from [6]), the reformulation in (8)-(9) facilitates the computation of linear embedding approximations, as shown in the next section.

A special class of nonlinear systems The immersibility property only hold for very special classes of systems. In a recent paper [13], we have shown a special class of discrete-time autonomous nonlinear systems with guaranteed immersibility. With linear control input, such systems become:

$$\begin{aligned} x^1(t+1) &= A^1 x^1(t) + \varphi(x^2(t)) + B^1 u(t), \\ x^2(t+1) &= A^2 x^2(t), t \in \mathbb{Z}^+ \end{aligned} \quad (13)$$

where $x^1 \in \mathbb{R}^{n_1}, x^2 \in \mathbb{R}^{n_2}, n_1 + n_2 = n, A^1 \in \mathbb{R}^{n_1 \times n_1}, A^2 \in \mathbb{R}^{n_2 \times n_2}, B^1 \in \mathbb{R}^{n_1 \times m}$ and $\varphi: \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_1}$ is some polynomial function.

Corollary 2: System (13) is immersible to a linear system in form of (2).

Proof: From Theorem 2 in [13], there exist a finite $k \in \mathbb{Z}^+$ and $\{\alpha_\ell \in \mathbb{R}^{n \times n}\}_{\ell=0}^k$ such that (8) is satisfied. The condition (9) can be easily verified. \square

B. Linear quadratic regulator with immersion

Now, we consider optimal control of System (1) with no constraint by the use of immersion. Suppose System (1) is immersible into $\Sigma(A, B, C)$ under the map $T(x)$, we formulate the following regulation problem of System (1) with the initial state $x(0)$

$$\min_u \sum_{t=0}^{\infty} \|x(t)\|_Q^2 + \|u(t)\|_R^2 \quad (14a)$$

$$\text{s.t. } x(t+1) = f(x(t), u(t)), t \in \mathbb{Z}^+ \quad (14b)$$

where $Q \succeq 0$ and $R \succ 0$ are user-defined parameters. With the condition (3), this optimal control problem can be equivalently written as the linear-quadratic regulator (LQR) for $\Sigma(A, B, C)$:

$$\min_u \sum_{t=0}^{\infty} \|\tilde{x}(t)\|_{C^T Q C}^2 + \|u(t)\|_R^2 \quad (15a)$$

$$\text{s.t. } \tilde{x}(t+1) = A\tilde{x}(t) + Bu(t), \tilde{x}(0) = T(x(0)). \quad (15b)$$

Hence, when (A, B) is stabilizable and (A, C) is observable, the optimal feedback control is

$$u(t) = KT(x(t)), t \in \mathbb{Z}^+, \quad (16)$$

where K is obtained by solving the discrete-time algebraic Riccati equation [17]

$$P = A^T P A - (A^T P B)(R + B^T P B)^{-1} (B^T P A) + C^T Q C, \quad K = -(R + B^T P B)^{-1} B^T P A. \quad (17)$$

C. Immersion-based MPC

In the presence of constraints in (6), we can design MPC for System (1) using $\Sigma(A, B, C)$ and $T(x)$ in a similar way. We consider the following stage and terminal costs in (7)

$$l(x, u) = \|x_i\|_Q^2 + \|u_i\|_R^2, l_f(x) = \|T(x)\|_P^2. \quad (18)$$

Let

$$\tilde{X}_f := \{x \in \mathbb{R}^{\tilde{n}} : C(A + BK)^k x \in X, K(A + BK)^k x \in U, k \geq 0\}. \quad (19)$$

The terminal constraint set in (7) is chosen to be the preimage of \tilde{X}_f under the map $T(x)$:

$$X_f = T^{-1}(\tilde{X}_f). \quad (20)$$

The set \tilde{X}_f is the maximal invariant set of the system $\tilde{x}^+ = (A + BK)\tilde{x}$ and can be efficiently computed using the classic algorithm in [18] when X and U are polytopes. There also exist algorithms for handling non-convex constraints, see [19]–[21]. As shown in [22], from the immersibility property, $T^{-1}(\tilde{X}_f)$ is the maximal invariant set of the closed system $x^+ = f(x, KT(x))$ with the control law $u = KT(x)$. With these definitions, the MPC problem is cast as follows:

$$\min_{\mathbf{u}_N} \sum_{i=0}^{N-1} (\|x_i\|_Q^2 + \|u_i\|_R^2) + \|T(x_N)\|_P^2 \quad (21a)$$

$$\text{s.t. } x_{i+1} = f(x_i, u_i), x_i \in X, u_i \in U, \quad (21b)$$

$$x_0 = x(t), x_N \in T^{-1}(\tilde{X}_f), i = 0, \dots, N-1. \quad (21c)$$

With (3), Problem (21) can be equivalently written as,

$$\min_{\mathbf{u}_N} \sum_{i=0}^{N-1} (\|\tilde{x}_i\|_{C^T Q C}^2 + \|u_i\|_R^2) + \|\tilde{x}_N\|_P^2 \quad (22a)$$

$$\text{s.t. } \tilde{x}_{i+1} = A\tilde{x}_i + B u_i, C\tilde{x}_i \in X, u_i \in U, \quad (22b)$$

$$\tilde{x}_0 = T(x(t)), \tilde{x}_N \in \tilde{X}_f, i = 0, \dots, N-1. \quad (22c)$$

Let the problem above be denoted as $\tilde{\mathbb{P}}_{\Sigma(A, B, C)}(x)$. Under the assumption that X and U are convex, this problem is convex and can be efficiently solved. In particular, when X and U are convex polytopes, it becomes a quadratic optimization problem. The feasible domain of $\tilde{\mathbb{P}}_{\Sigma(A, B, C)}(x)$ is given by

$$D_{\Sigma(A, B, C)} := \{x \in X : \exists \mathbf{u}_N \text{ s.t. (22b) - (22c)}\}. \quad (23)$$

IV. LINEAR EMBEDDING COMPUTATION

This section presents the detailed procedure for computing linear embedding $\Sigma(A, B, C)$. We also show the connections between the proposed immersion-based approach and Koopman-based optimal control [14], [15].

A. Polyflow approximation

For general nonlinear systems, the immersibility property does not hold and we can only obtain approximate immersion or linear embedding. Similar to the *polyflow* approximation method in [12] for continuous-time autonomous systems, we develop a linearization technique for discrete-time control systems by using the nilpotency condition in Theorem 2. Given the similarity in the form of the linearized system, we also call it *polyflow* approximation.

Definition 2: Given a $k \in \mathbb{Z}^+$ and $\{\alpha_\ell \in \mathbb{R}^{n \times n}\}_{\ell=0}^k$, let

$$A = \begin{pmatrix} \mathbf{0} & I_n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & I_n \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{k-1} & \alpha_k \end{pmatrix}, \quad B = \begin{pmatrix} D_{u^T} \hat{f}^1(0, 0) \\ D_{u^T} \hat{f}^2(0, 0) \\ \vdots \\ D_{u^T} \hat{f}^{k+1}(0, 0) \end{pmatrix}, C = (I \quad 0_{n \times kn}). \quad (24)$$

We call $\Sigma(A, B, C)$ where matrices (A, B, C) are in the form of (24) a k^{th} *polyflow* approximation of System (1).

To determine a *polyflow* approximation of order k , we need to find $\{\alpha_\ell \in \mathbb{R}^{n \times n}\}_{\ell=0}^k$ such that $\|\hat{f}^{k+1}(x, 0) - \sum_{\ell=0}^k \alpha_\ell \hat{f}^\ell(x, 0)\|$ is minimized for any $x \in X$. To do so, we generate a set of M data points and solve the following problem for numerical *polyflow* approximation

$$\min_{\alpha_k} \sum_{i=1}^M \|\hat{f}^{k+1}(x_i, 0) - \alpha_k \hat{F}^k(x_i)\|_2^2, \quad (25)$$

where $\alpha_k := \{\alpha_0, \alpha_1, \dots, \alpha_k\}$ and

$$\hat{F}^k(x) = \begin{pmatrix} x \\ \hat{f}^1(x, 0) \\ \vdots \\ \hat{f}^k(x, 0) \end{pmatrix}, k \in \mathbb{Z}^+. \quad (26)$$

B. EDMD and polyflow basis

Besides *polyflow* approximation above, the nilpotency condition in Theorem 2 also allows us to compute linear embedding in a fully data-driven fashion using the extended dynamical mode decomposition (EDMD) algorithm [23], like Koopman-based optimal control [14], [15]. We first need the following proposition.

Proposition 1: System (1) is immersible into $\Sigma(A, B, C)$, if and only if there exist a map $T : \mathbb{R}^n \rightarrow \mathbb{R}^{\tilde{n}}$ and matrices $A \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $B \in \mathbb{R}^{\tilde{n} \times m}$ and $C \in \mathbb{R}^{n \times \tilde{n}}$ such that (3) holds $\forall (x, u) \in \mathbb{R}^n \times \mathbb{R}^m$.

Although this result can be easily derived from Theorem 2, it can be proved independently. An elementary proof is given in [24].

Motivated by Proposition 1, to compute a linear approximation, we first select the map $T(x)$. Then, we generate a set of M snapshot data points $\{(x_i, u_i, x_i^+) : i = 1, 2, \dots, M\}$ (where $x_i^+ = f(x_i, u_i)$) from $X \times U$ and formulate the following problem, motivated from (3),

$$\min_{A, B} \sum_{i=1}^M \|T(x_i^+) - AT(x_i) - Bu_i\|_2^2. \quad (27)$$

Note that C is usually known from the selection of $T(x)$. We arrive at the same problem as in [14], [15], but from a different path. This shows the link between the immersibility property and the Koopman operator, as already mentioned in [13] for discrete-time autonomous systems.

In particular, when the dynamics $f(x, u)$ is known, we propose to use a special set of basis functions. We call the functions $\hat{F}^k(x)$ defined in (26) the *polyflow* basis of degree k for any $k \in \mathbb{Z}^+$. The *polyflow* basis is also called delay coordinates for autonomous systems under the Koopman operator framework [25], [26]. To determine an appropriate basis, we solve Problem (25) and increase k until the approximation error is sufficiently small. In case that there is redundancy in $\hat{F}^k(x)$, i.e., some component is a linear combination of others, we remove the redundant components and let the lifting map $T(x)$ be $V\hat{F}^k(x)$, where $V \in \mathbb{R}^{\tilde{n} \times n(k+1)}$ is some proper matrix and \tilde{n} is the number of linearly independent components of $\hat{F}^k(x)$. The issue of redundancy occurs when one component of $f(x, 0)$ is linear in x . In a data-driven fashion, we can also remove redundancy using singular value decomposition (SVD). When the system is equipped with a well-designed nominal controller $\kappa_{nominal}(x)$, we can also choose the basis from the closed-loop system $x^+ = f(x, \kappa_{nominal}(x))$.

Remark 1: While the link between the immersibility property and the Koopman operator is mentioned above, we emphasize that our immersion-based approach has two key differences compared with Koopman-based approaches [14], [15]. First, the underlying problem in (27) stems from the immersibility condition in Theorem 2 while Koopman-based approaches essentially solve a model reduction problem of the infinite-dimensional Koopman operator. Second, we provide a way to compute endogenously a basis of functions motivated by the nilpotency immersibility condition. On the contrary, the Koopman-based approaches in [14], [15] fix the basis a priori, which incurs additional conservativeness.

V. EXPERIMENT: OPTIMAL PEST CONTROL

Consider the pest control problem in [27]:

$$\begin{aligned} v(t+1) &= v(t) + cv(t)(1 - v(t)/\kappa) - rv(t)p(t), \\ p(t+1) &= dp(t) + v(t)p(t) - a(t)p(t), \end{aligned}$$

where v is the valuable population, p is the pest population, r is a population interaction constant which measures the efficiency of the pest population, c is the intrinsic growth

rate of the valuable population, κ is the carrying capacity, d is the intrinsic growth rate of the pest population, and a is the control action to decrease the growth of the pest population. The parameters are chosen to be: $r = 0.5, c = 0.2, \kappa = 2, d = 0.2$. The objective is to steer the valuable population to $v = 1$ with the pest population and the control being $p = 0.2$ and $u = 0.2$. The constraints are: $0.5 \leq v \leq 1.5, 0 \leq p \leq 1$, and $0 \leq u \leq 0.4$. Let $x_1 = v - 1, x_2 = p - 0.2, u = a - 0.2$. Hence, $X = \{(x_1, x_2) : \|x_1\| \leq 0.5, -0.2 \leq x_2 \leq 0.8\}$ and $U = \{u : \|u\| \leq 0.2\}$. Let $N = 10, Q = I$ and $R = 0.1$.

First, we make comparison to Jacobian linearization about the origin $(x, u) = (0, 0)$. Let $\bar{A} = \frac{\partial f(x, u)}{\partial x}|_{(0,0)}$ and $\bar{B} = \frac{\partial f(x, u)}{\partial u}|_{(0,0)}$. We then define $\tilde{\mathbb{P}}_{\Sigma(\bar{A}, \bar{B}, I)}$ as in (22) with the feasible domain $\mathcal{D}_{\Sigma(\bar{A}, \bar{B}, I)}$. For the *polyflow* approximation, we randomly sample 10^5 points in X , solve Problem (25) with $k = 5$, and obtain a 5^{th} *polyflow* approximation $\Sigma(A, B, C)$ with the feasible domain $\mathcal{D}_{\Sigma(A, B, C)}$. The simulation results are given in Figure 1. While recursive feasibility of the MPC problem is not guaranteed in both methods because of mismatch between the linearized model and the actual system, this figure shows that the *polyflow* approximation outperforms Jacobian linearization as the *polyflow* approximation is able to generate feasible trajectories from initial states that are not contained in $\mathcal{D}_{\Sigma(\bar{A}, \bar{B}, I)}$.

We also make comparison with Koopman-based approaches that use monomials [16] and radial basis functions. We use monomials with the maximal degree being 6, given by

$$\{x_1^{s_1} x_2^{s_2} : 1 \leq s_1 + s_2 \leq 6, s_1, s_2 \in \mathbb{Z}^+\}.$$

The number of monomials is $8!/(6!2!) - 1 = 27$. The radial basis functions are taken from [15] in the form of

$$g(x) = \|x - x_0\|^2 \log(\|x - x_0\|),$$

where x_0 is randomly selected with the uniform distribution on X . For a fair comparison, we generate 25 radial basis functions and include the state itself in the basis. Hence, the dimension of the lifted state-space is also 27. The matrices A, B are obtained from Problem (27). We then solve the MPC problems starting from the same initial state. To measure their performance quantitatively, we compute the LQ cost for 100 steps:

$$\sum_{t=0}^{100} (\|x(t)\|_Q^2 + \|u(t)\|_R^2).$$

The trajectories starting from $x(0) = [0.1488 \ -0.1319]^T$ are shown in Figure 2 with their corresponding LQ costs. As we can see from this figure, both the *polyflow* approximation and the EDMD with *polyflow* basis outperform the other two EDMD approaches. The LQ cost of the EDMD with monomials is not computed because it loses feasibility at $t = 3$. The performance of the EDMD with radial basis functions relies on the randomly generated points x_0 . Here, we select one realization where the MPC problem does not lose feasibility. Note that the dimension of the lifted system is 12 for both the *polyflow* approximation and the EDMD with *polyflow* basis.

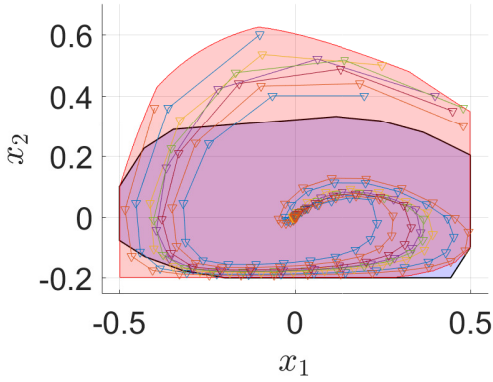


Fig. 1: Polyflow approximation versus Jacobian linearization of pest control dynamics: the blue area refers to $\mathcal{D}_{\Sigma(\bar{A}, \bar{B}, I)}$ and the red area refers to $\mathcal{D}_{\Sigma(A, B, C)}$.

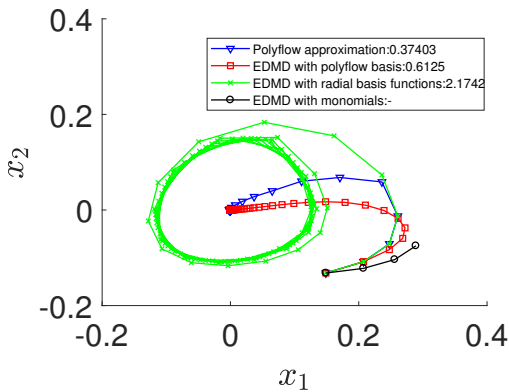


Fig. 2: Comparison with radial and monomial basis functions on pest control.

VI. CONCLUSIONS

In this paper, we present an approach to convert the non-convex MPC problem of nonlinear systems into a convex problem via linear embedding. The computation of linear embedding is implemented offline via *polyflow* approximation, based on a nilpotency algebraic condition of the immersibility property for nonlinear control systems. From the derivation of this algebraic condition, we also show the link between the immersibility property and the Koopman operator. With this link, the proposed *polyflow* approximation can be considered as a Koopman-based lifting approach with a special basis, which is computed endogenously. Finally, we evaluate our immersion-based MPC approach on the pest control problem by comparison with locally linearized MPC and other Koopman-based control approaches using monomial and radial basis functions.

REFERENCES

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[2] M. Cannon. Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, 28(2):229–237, 2004.

[3] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[4] S. Monaco and D. Normand-Cyrot. The immersion under feedback of a multidimensional discrete-time non-linear system into a linear system. *International Journal of Control*, 38(1):245–261, 1983.

[5] J. Levine and R. Marino. Nonlinear system immersion, observers and finite-dimensional filters. *Systems & Control Letters*, 7(2):133–142, 1986.

[6] H.G. Lee and S. I. Marcus. Immersion and immersion by nonsingular feedback of a discrete-time nonlinear system into a linear system. *IEEE transactions on Automatic Control*, 33(5):479–483, 1988.

[7] A. J. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 3(1):47–52, 1983.

[8] A. Astolfi and R. Ortega. Immersion and invariance: A new tool for stabilization and adaptive control of nonlinear systems. *IEEE Transactions on Automatic control*, 48(4):590–606, 2003.

[9] H. Bass and G. Meisters. Polynomial flows in the plane. *Advances in Mathematics*, 55(2):173–208, 1985.

[10] A. van den Essen. Locally finite and locally nilpotent derivations with applications to polynomial flows, morphisms and \mathcal{G}_a -actions. ii. *Proceedings of the American Mathematical Society*, 121(3):667–678, 1994.

[11] A. van den Essen. Locally nilpotent derivations and their applications, iii. *Journal of Pure and Applied Algebra*, 98(1):15–23, 1995.

[12] R. M. Jungers and P. Tabuada. Non-local linearization of nonlinear differential equations via polyflows. In *Proceedings of the American Control Conference*, 2019.

[13] Z. Wang and R. M. Jungers. A data-driven immersion technique for linearization of discrete-time nonlinear systems. *IFAC World Congress*, 2020.

[14] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of koopman eigenfunctions for control. *arXiv preprint arXiv:1707.01146*, 2017.

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

[16] A. Mauroy and J. Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65(6):2550–2565, 2020.

[17] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[18] E.G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36:1008–1020, 1991.

[19] N. Athanasopoulos and R. M. Jungers. Computing the domain of attraction of switching systems subject to non-convex constraints. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 41–50. ACM, 2016.

[20] Z. Wang, R. M. Jungers, and C. J. Ong. Computation of the maximal invariant set of linear systems with quasi-smooth nonlinear constraints. In *Proceedings of the European Control Conference*, pages 3803–3808, 2019.

[21] Z. Wang, R. M. Jungers, and C. J. Ong. Computation of the maximal invariant set of discrete-time linear systems subject to a class of non-convex constraints. *Automatica*, 125:109463, 2021.

[22] Z. Wang, R. M. Jungers, and C. J. Ong. Computing invariant sets of discrete-time nonlinear systems via state immersion. *IFAC World Congress*, 2020.

[23] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[24] Zheming Wang and Raphaël M Jungers. Immersion-based model predictive control of constrained nonlinear systems: Polyflow approximation. *arXiv preprint arXiv:2011.13255*, 2020.

[25] I. Susuki, Y. and Mezić. A prony approximation of koopman mode decomposition. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 7022–7027, 2015.

[26] H. Arbabi and I. Mezić. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.

[27] W. Ding, R. Hendon, B. Cathey, E. Lancaster, and R. Germick. Discrete time optimal control applied to pest control problems. *Involve, a Journal of Mathematics*, 7(4):479–489, 2014.