2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Controlled Invariant Sets for Polynomial Systems Defined by Non-polynomial Equations

Tomoyuki Iori

Abstract— **This study proposes a method for designing a feedback controller that makes a prescribed set invariant for a given polynomial system. Although the system is restricted to polynomial systems, the class of invariant sets is not limited to algebraic sets; it is the zero sets of nonlinear functions satisfying a specific type of partial differential equations (PDE) with coefficients in polynomials. Based on the algebraic relations between a nonlinear function and its derivatives derived from the PDE, a constructive sufficient condition for the existence of the desired controllers is provided. Using symbolic computation, the controllers can be computed exactly. Numerical examples are provided to illustrate the effectiveness of the proposed method.**

I. Introduction

A subset of the state space is said to be *invariant* for a dynamical system if the trajectory of the system remains in the subset as long as it starts from the subset and is defined. Invariant sets for dynamical systems are crucial in systems and control theory [1], [2]. For example, it has been used to ensure recursive feasibility in model predictive control [3], [4], certify stability in terms of Lyapunov functions [1], [2], and provide safety guarantees through barrier certificates and control barrier functions [5], [6]. On the one hand, various estimation and approximation techniques for invariant sets have been studied to determine the limitations of a given control system under certain constraints [7]–[9]. On the other hand, a problem of designing a controller that makes a prescribed set invariant for a given dynamical system can be considered. This type of problem has been studied using various techniques, such as barrier certificates [5], template polynomials in constraint satisfaction problems [10], and symbolic computations based on commutative algebra and algebraic geometry [11], [12], [16].

In [11], a sufficient condition was provided for the existence of a polynomial feedback controller that renders a prescribed algebraic set invariant for the corresponding closed-loop system. The construction of the controller is performed symbolically by using the theory of Gröbner bases [13]. Therefore, this method does not suffer from numerical computational errors. This construction method and the condition for existence were extended to the case of output feedback control in [14]. In similar problem settings, controllers consisting of rational functions have been considered [15]. The class of invariant sets was extended to semi-algebraic sets [12], [16]. In [16], it was shown that the controllers can include continuously differentiable functions. However, in all of the above results, the class of invariant sets is limited to the sets described by polynomials, that is, algebraic or semi-algebraic sets.

The aim of this paper is to push the boundary of the class of invariant sets to a more general one for polynomial systems. Specifically, zero sets of nonlinear functions satisfying a certain type of partial differential equation (PDE) with polynomial coefficients are considered. By imposing algebraic relations between the function defining the invariant set and its derivatives, we can derive a constructive sufficient condition for the existence of the desired controllers. Note that algebraic relations of a function and its derivatives have also been used, for example, in model structure simplification via immersion [18] and state estimation of nonlinear systems [19].

A similar approach was considered in [17], where the system and invariant set were defined by the compositions of polynomial functions and a continuously differentiable function that may be non-polynomial. The Jacobian of the non-polynomial function is also assumed to consist of compositions of polynomials and the non-polynomial function itself, which is satisfied by, for example, cosine and sine functions. This assumption allows us to apply the results developed in [11] to images of the non-polynomial function. Consequently, invariant sets can be constructed in the preimage of the non-polynomial function, which leads to defining more general invariant sets than algebraic sets.

The target system and its invariant set must be defined using a common non-polynomial function in [17]. In this paper, although the target system is assumed to be polynomial one, the invariant set can be defined using non-polynomial functions that are not included in the target system, which highlights the difference between the proposed and existing methods. It is worth mentioning that many non-polynomial dynamical systems can be converted to polynomial systems via immersion [18], and the proposed method can also be applied to the converted systems.

The remainder of this paper is organized as follows: Section II formulates the problem setting and introduces a class of nonlinear functions that define the invariant sets by imposing an assumption described by a certain type of PDE. Section III briefly reviews the invariance property for autonomous systems and reformulates the sufficient condition for the existence of the desired controllers in a form compatible with the assumption made in the previous section. Section IV presents the controller synthesis problem, a

This work was partly supported by JSPS KAKENHI Grant Number 22K17855.

T. Iori is with the Department of Information and Physical Sciences, Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565–0871, Japan `t-iori@ist.osaka-u.ac.jp`

sufficient condition for the existence of the desired controller, and an algorithm for constructing it. Section V presents numerical examples to illustrate the effectiveness of the proposed method and the differences between the proposed framework and existing method [17]. Finally, Section VI concludes the study and discusses future work.

Notations: For the field of real numbers $\mathbf{R}$ and a vector of indeterminates $y = [y_1 \ \cdots \ y_n]^\top$, $\mathbf{R}[y]$ denotes the ring of polynomials in $y_1, \dots, y_n$ over $\mathbf{R}$. The set of $m \times l$ matrices with components in $\mathbf{R}[y]$ is denoted by $\mathbf{R}[y]^{m\times l}$. When $l = 1$, $\mathbf{R}[y]^{m\times 1}$ is abbreviated by $\mathbf{R}[y]^m$. The ring of all continuously differentiable functions in y, which is endowed with usual addition and multiplication, is denoted by $C_1(y)$. For a subset $V \in \mathbf{R}^n$, the set of functions that vanish at all $y \in V$ is denoted by $\mathcal{I}_{C_1}(V) := \{f \in C_1(y) \mid f(y) = 0 \text{ for all } y \in V\}$. For a given elements $v_1, \dots, v_l \in R^n$ with a subring $R \subset C_1(y)$, $\langle v_1, v_2, \dots, v_l\rangle$ denotes the R-module generated by $v_1, \dots, v_l$, that is, $\langle v_1, v_2, \dots, v_l\rangle_R := \{v \in R^n \mid c_1 v_1 + \cdots + c_l v_l, c_j \in R \ (j = 1, \dots, l)\}$. The subset $\{c \in R^l \mid c_1 v_1 + \cdots c_l v_l = 0\}$ is called the *syzygy* of $v_1, \dots, v_l$ and denoted by $\mathrm{Syz}^R(v_1, \dots, v_l)$. $\nabla_y := [\partial_{y_1} \ \cdots \ \partial_{y_n}]^\top$ denotes the column vector of the differential operators, where $\partial_{y_i} = \partial/\partial y_i$. The symbols ∇_y and ∂_{y_i} are abbreviated as ∇ and ∂_i, respectively, if y is clearly specified according to the context. For $f(y) \in C_1(y)^n$, the Lie derivative $\mathcal{L}_f p$ of $p \in C_1(y)$ with respect to f is defined as $\mathcal{L}_f p := f^\top \nabla p$.

II. Problem Setting

Consider the following control-affine nonlinear system:

$$\dot{x} = f(x) + G(x)u, \tag{1}$$

where $x(t) \in \mathbf{R}^n$ and $u(t) \in \mathbf{R}^m$ are the state and control input, respectively, at time $t \in \mathbf{R}_+$, and $f(x) \in \mathbf{R}[x]^n$ and $G(x) = [g_1(x) \ g_2(x) \ \cdots \ g_m(x)] \in \mathbf{R}[x]^{n\times m}$ are polynomial vector and matrix, respectively, in the components of x.

In this study, the following problem is considered: For a given nonlinear function $p(x)$, find a feedback controller $u = \alpha(x)$ that renders the zero set $\{x \in \mathbf{R}^n \mid p(x) = 0\}$ invariant. The class of functions to which $p(x)$ belongs is stated by the following assumption.

Assumption 1: Let $\mathcal{B} := [1 \ \nabla^\top]^T$ and $l > 1$ be an integer. For each $k = 1, \dots, l$, there exist $a_k(x) = [a_{k1} \ a_{k2} \ \cdots \ a_{kn}]^\top \in \mathbf{R}[x]^n$ and $b_k(x) \in \mathbf{R}[x]$ such that $p(x)$ satisfies the following PDE:

$$[b_k \ a_k^\top]\mathcal{B}p(x) = b_k(x)p(x) + a_k^\top(x)\nabla p(x) = 0. \tag{2}$$

Evidently, polynomials and rational functions satisfy Assumption 1. Moreover, the exponentials and logarithms of polynomials and rational functions satisfy Assumption 1, as shown in the following examples.

Example 1: For a rational function $r(x) = n_r(x)/d_r(x)$ ($n_r, d_r \in \mathbf{R}[x]$), $p(x) = \exp(r(x))$ satisfies Assumption 1 because $\partial_i \exp(r(x)) = \exp(r(x))\partial_i r(x)$ yields

$$d_r^2 \partial_i p = (d_r \partial_i n_r - n_r \partial_i d_r)p.$$

Example 2: For $r(x)$ defined in the previous example, $p(x) = \log(r(x))$ also satisfies Assumption 1 because $\partial_i \log(r(x)) = \partial_i r(x)/r(x)$ implies that

$$(d_r \partial_j n_r - n_r \partial_j d_r)\partial_i p - (d_r \partial_i n_r - n_r \partial_i d_r)\partial_j p = 0.$$

Note that logarithms do not belong to the class of functions considered in [17] because the derivative of a logarithm is the reciprocal of its argument, which cannot be expressed as a polynomial of the logarithm itself.

Example 3: Trigonometric functions $\cos(x)$ and $\sin(x)$ for $x \in \mathbf{R}$ cannot be treated in this framework because their derivatives $-\sin(x)$ and $\cos(x)$ do not admit any algebraic relations of the form (2). However, the situation in the multi-dimensional case is different. For instance, $\sin(a^\top x)$ for $x \in \mathbf{R}^n$ and a constant vector $a \in \mathbf{R}^n$, $a_j \partial_i \sin(a^\top x) - a_i \partial_j \sin(a^\top x) = 0$ holds for each pair $i, j \in \{1, \dots, n\}$ with $i \neq j$.

III. Invariance in Autonomous Case

Before discussing controller synthesis, the invariance result for an autonomous system:

$$\dot{x} = h(x) \tag{3}$$

with $h(x) \in C_1(x)^n$ is revisited. The initial state $x_0 \in \mathbf{R}^n$ is arbitrary, and $I(x_0) \subset \mathbf{R}$ denotes the maximal existence interval of the solution $\varphi_{x_0}(t)$ to (3) starting from $\varphi_{x_0}(0) = x_0$. For $p(x) \in C_1(x)$, its zero set $V := \{x \in \mathbf{R}^n \mid p(x) = 0\}$ is said to be *invariant for* $h(x)$ if $x_0 \in V$ implies $\varphi_{x_0}(t) \in V$ for all $t \in I(x_0)$.

The following is a standard result for the controlled invariance, which is included for completeness:

Lemma 1: For $h(x)$ and V as defined above, the following statements hold:

(i) If there exists $\lambda(x) \in C_1(x)$ such that

$$\mathcal{L}_h p(x) = \lambda(x)p(x), \tag{4}$$

then, V is invariant for h.

(ii) Suppose $\mathcal{L}_h p(x) \in C_1(x)$. If V is invariant for h, then

$$\mathcal{L}_h p(x) \in \mathcal{I}_{C_1}(V).$$

Proof: (i) For the solution $\varphi_{x_0}(t)$ to (3), the time derivative of $p(\varphi_{x_0}(t))$ is given by

$$\begin{aligned}\frac{d}{dt}p(\varphi_{x_0}(t)) &= \mathcal{L}_h p(\varphi_{x_0}(t)) \\ &= \lambda(\varphi_{x_0}(t))p(\varphi_{x_0}(t)),\end{aligned}$$

where the last equality is obtained from (4). It suffices to show that $p(\varphi_{x_0}(t)) = 0$ for all $t \in I(x_0)$ if $x_0 \in V$. Indeed, if $x_0 \in V$, then

$$\frac{d}{dt}p(x_0) = \mathcal{L}_h p(x_0) = \lambda(x_0)p(x_0) = 0$$

holds, implying that $p(\varphi_{x_0}(t)) = p(x_0) = 0$ for all $t \in I(x_0)$ because of the uniqueness of the solution.

(ii) From the assumption, $p(\varphi_{x_0}(t)) = 0$ for all $t \in I(x_0)$ if $x_0 \in V$. Hence, the time derivative of $p(\varphi_{x_0}(t))$ is also zero for all $t \in I(x_0)$; that is,

$$\mathcal{L}_h p(\varphi_{x_0}(t)) = 0.$$

In particular, at $t = 0 \in I(x_0)$, we obtain $\mathcal{L}_h p(x_0) = 0$ for any $x_0 \in V$. Thus, $\mathcal{L}_h p(x)|_{x\in V} = 0$, which implies $\mathcal{L}_h p(x) \in \mathcal{I}_{C_1}(V)$. ∎

The left-hand side of (4) includes $\nabla p(x)$ whereas the right-hand side includes $p(x)$. Hence, if an algebraic relation exists between $\nabla p(x)$ and $p(x)$, (4) can be reduced to an algebraic condition by isolating the non-polynomial components in $p(x)$. This consideration leads to another formulation of (4) as follows:

Corollary 1: For $\mathcal{B}$ defined in Assumption 1, if there exists $\lambda(x) \in C_1(x)$ such that

$$[\lambda \; h^\top]\mathcal{B} \circ p(x) = 0, \tag{5}$$

V is invariant for h.

Proof: The condition (5) is equal to $\mathcal{L}_h p(x) + \lambda(x)p(x) = 0$, which is equivalent to (4) with the replacement of λ with $-\lambda$. ∎

IV. Controller Synthesis for Invariant Set Defined by Non-Polynomial Functions

Let us now return to the controller design problem. The problem of determining a feedback controller $u = \alpha(x)$ that renders the zero set $\{x \in \mathbf{R}^n \mid p(x) = 0\}$ invariant can be reformulated as follows: By using $\mathcal{B}$ in Assumption 1, (4) can be rewritten as

$$\begin{aligned} 0 &= \mathcal{L}_{f+Gu} p - \lambda p \\ &= \begin{bmatrix} 0 \\ f \end{bmatrix}^\top \mathcal{B}p + \sum_{j=1}^{m} u_j \begin{bmatrix} 0 \\ g_j \end{bmatrix}^\top \mathcal{B}p - \lambda \begin{bmatrix} 1 \\ 0_n \end{bmatrix}^\top \mathcal{B}p \\ &= \left(\begin{bmatrix} 0 \\ f \end{bmatrix} + \sum_{j=1}^{m} u_j \begin{bmatrix} 0 \\ g_j \end{bmatrix} - \lambda \begin{bmatrix} 1 \\ 0_n \end{bmatrix} \right)^\top \mathcal{B}p, \end{aligned} \tag{6}$$

where $0_n \in \mathbf{R}^n$ denotes the zero vector. Hence, V is invariant for $f + Gu$ with feedback $u = \alpha(x)$ if there exist $\alpha_j \in C_1(x)$ and $\lambda \in C_1(x)$ such that

$$v_0 + \sum_{i=1}^{m} u_i v_i - \lambda v_{m+1} = 0_{n+1}, \tag{7}$$

where

$$\begin{aligned} v_0 &:= [0 \; f^\top]^\top, \quad v_{m+1} := [1 \; 0_n^\top]^\top \\ v_j &:= [0 \; g_j^\top]^\top \quad (j = 1, \ldots, m). \end{aligned} \tag{8}$$

However, (7) implies that $\lambda = 0$ because the first components of v_i are zero, except for v_{m+1}. This results in $v_0 + u_1 v_1 + \cdots + u_m v_m = 0_{m+1}$, which corresponds to the trivial system $\dot{x} = f + Gu = 0_n$. To obtain more nontrivial feedback controllers, the algebraic property of $p(x)$ imposed by Assumption 1 is utilized.

As $p(x)$ satisfies Assumption 1, there exist $a_k \in \mathbf{R}[x]^n$ and $b_k \in \mathbf{R}[x]$ such that $[b_k \; a_k^\top]\mathcal{B}p = 0$ holds for each $k = 1, \ldots, l$. This indicates that the addition of any linear combination of vectors

$$v_{m+1+k} := [b_k \; a_k^\top]^\top \quad (k = 1, \ldots, l) \tag{9}$$

to the left-hand side of (7) does not change the right-hand side. Consequently, the following theorem is obtained:

Theorem 1: For $v_j \in \mathbf{R}[x]^{n+1}$ $(j = 0, \ldots, m+l+1)$ defined so far, if there exist $\alpha_1, \alpha_2, \ldots, \alpha_{m+l+1} \in C_1(x)$ such that

$$v_0 + \sum_{j=1}^{m+l+1} \alpha_j v_j = 0_{n+1} \tag{10}$$

holds, then V is invariant for $f+Gu$ with feedback $u_j = \alpha_j(x)$ for $j = 1, \ldots, m$.

Proof: As $v_{m+k+1} = [b_k \; a_k^\top]^\top$, by multiplying $(\mathcal{B}p)^\top$ to both sides of (10), the terms corresponding to $j \in \{m+2, m+3, \ldots, m+l+1\}$ are nullified. Hence, we have

$$(\mathcal{B}p)^\top v_0 + \sum_{j=1}^{m+1} \alpha_j (\mathcal{B}p)^\top v_j = 0,$$

which is equivalent to (6) with $u_j = \alpha_j(x)$ and $\lambda(x) = \alpha_{m+1}(x)$. ∎

All solutions $w = [\alpha_1 \; \alpha_2 \; \cdots \; \alpha_{m+l+1}] \in \mathbf{R}[x]^{m+l}$ to (10) are expressed as follows:

$$w = w_0 + \langle w_1, \ldots, w_s \rangle_{\mathbf{R}[x]},$$

and $w_0, w_1, \ldots, w_s \in \mathbf{R}[x]^{m+l}$ can be computed symbolically using some computer algebra system such as `Risa/Asir` [22], `SINGULAR` [20], `Macaulay2` [21], or `Maple` (see, e.g., [23]–[25]). Moreover, as shown in [16], all solutions $w \in C_1(x)^{m+l}$ can be expressed as

$$w = w_0 + \langle w_1, \ldots, w_s \rangle_{C_1(x)}.$$

Finally, the design of a feedback controller $u = \alpha(x)$ that renders the zero set $\{x \in \mathbf{R}^n \mid p(x) = 0\}$ invariant is summarized in Algorithm 1.

Note that the information of $p(x)$ is encoded in the polynomials $a_k(x)$ and $b_k(x)$ and is thus considered through those polynomials. To the best of the author's knowledge, no algorithm exists for determining a_k and b_k for a given $p(x)$; hence, they must be handcrafted.

V. Numerical Example

Consider system (1) with

$$f(x) = \begin{bmatrix} x_2 \\ (1 - x_1^2)x_2 - x_1 \\ (x_1 - x_2)x_3 \end{bmatrix}, \quad G(x) = \begin{bmatrix} 0 & 0 \\ x_2 & 0 \\ 0 & x_3 \end{bmatrix}, \tag{11}$$

Algorithm 1 Feedback controller synthesis that achieves invariance of V for $f + Gu$

Input: Polynomial vectors $f, g_1, g_2, \ldots, g_m, a_1, a_2, \ldots, a_l \in \mathbf{R}^n[x]$ and polynomials $b_1, b_2, \ldots, b_l \in \mathbf{R}[x]$

Output: Feedback controller $u = \alpha(x)$ that makes $V = \{x \in \mathbf{R}^n \mid p(x) = 0\}$ invariant

1: Define $v_0, v_1, \ldots, v_{m+l+1} \in \mathbf{R}[x]^{n+1}$ as in (8) and (9)
2: Find a polynomial vector $w_0 \in \mathbf{R}[x]^{m+l}$ that satisfy (10)
3: Find a set of generators $w_1, \ldots, w_s \in \mathbf{R}[x]^{m+l}$ of the syzygy module $\mathrm{Syz}^{\mathbf{R}[x]}(v_1, \ldots, v_{m+l+1})$
4: Select the coefficients $\beta_i \in C_1(x)$ $(i = 1, \ldots, s)$ and define $\alpha = w_0^{[1:m]} + \sum_{i=1}^{s} \beta_i w_i^{[1:m]}$, where $w_i^{[1:m]}$ denotes the vector consisting of the first m components of w_i, that is, $[w_{i1} \; w_{i2} \; \cdots \; w_{im}]^\top \in \mathbf{R}[x]^m$

where $x(t) \in \mathbf{R}^3$ and $u(t) \in \mathbf{R}^2$. For this system, invariant sets defined by two nonlinear functions are considered. Both of them are non-polynomial, and one of them does not belong to the class considered in the existing literature [17]. Note that all the symbolic computations for the numerical examples were performed using `Risa/Asir` and took less than one second on a laptop PC (Intel Core i7-1255U, 1.70GHz; RAM: 16.0GB; Windows 11 64bit).

A. Invariant set defined by a non-polynomial function

Consider the nonlinear function

$$p(x) = \exp\left(-\frac{x_1^2}{10} + \frac{x_2^2}{10}\right) - x_3, \tag{12}$$

whose derivatives are given as follows:

$$\begin{aligned} \partial_1 p &= -\frac{x_1}{5} \exp\left(-\frac{x_1^2}{10} + \frac{x_2^2}{10}\right), \\ \partial_2 p &= \frac{x_2}{5} \exp\left(-\frac{x_1^2}{10} + \frac{x_2^2}{10}\right), \\ \partial_3 p &= -1. \end{aligned} \tag{13}$$

From (12) and (13), the following relations can be obtained:

$$\begin{aligned} \partial_1 p - \frac{x_1 x_3}{125}\partial_3 p + \frac{x_1}{125} p &= \frac{1}{125}[x_1 \;\, 1 \;\, 0 \;\, x_1 x_3]\mathcal{B}p = 0, \\ \partial_2 p + \frac{x_2 x_3}{125}\partial_3 p - \frac{x_2}{125} p &= \frac{1}{125}[x_2 \;\, 0 \;\, 1 \;\, x_2 x_3]\mathcal{B}p = 0, \end{aligned} \tag{14}$$

where $\mathcal{B} = [1 \;\, \partial_1 \;\, \partial_2 \;\, \partial_3]^\top$. For the vectors v_j $(j = 0, \ldots, 5)$ obtained from (11) and (14), the basis of solutions w_0 and w_1 are computed using `Risa/Asir` as

$$w_0 = \frac{1}{5}\begin{bmatrix} 0 \\ (x_1^2 - 1)x_2^2 + (2x_1 - 5)x_2 + 5x_1 \\ (x_1^2 - 1)x_2^2 - 2x_1 x_2 \\ -5x_1(x_2 + 1) + 5x_2 \\ 5x_2 \end{bmatrix}, \quad w_1 = \begin{bmatrix} -5 \\ -x_2^2 \\ x_2^2 \\ 5x_2 \\ 0 \end{bmatrix}.$$

Hence, each controller expressed as

$$\alpha(x) = \frac{1}{5}\begin{bmatrix} 0 \\ (x_1^2 - 1)x_2^2 + (2x_1 - 5)x_2 + 5x_1 \end{bmatrix} + \beta(x)\begin{bmatrix} -5 \\ -x_2^2 \end{bmatrix} \tag{15}$$

with $\beta \in C_1(x)$ makes $V = \{x \in \mathbf{R}^3 \mid p(x) = 0\}$ invariant.

To illustrate the effectiveness of the controller (15), numerical simulations were conducted for the uncontrolled system and two closed-loop systems with $\beta(x) = 0$ and $\beta(x) = \sin(10x_1)/3$. The initial state is set to $x_0 = [1 \;\, 1 \;\, 1]^\top$, which satisfies $p(x_0) = 0$; thus $x_0 \in V$, and the time interval is set to $t \in [0, 10]$. Fig. 1 shows the state trajectories obtained by the uncontrolled system and both the controllers. Fig. 2 shows the input sequences generated by the controllers. Unlike the uncontrolled trajectory, which immediately jumps out of V, the controlled trajectories remain within V. This is also confirmed by Figs. 3 and 4, which show the time series of each axis $\{\varphi_{x_0}(t)\}_i$ $(i = 1, 2, 3)$ and $p(\varphi_{x_0}(t))$. The difference between the trajectories with $\beta = 0$ and $\beta = \sin(10x_1)/3$ shows the effect of the choice of $\beta(x)$, which indicates that this degree of freedom may be utilized for other purposes, such as stabilization or energy minimization, while maintaining the invariance of V.

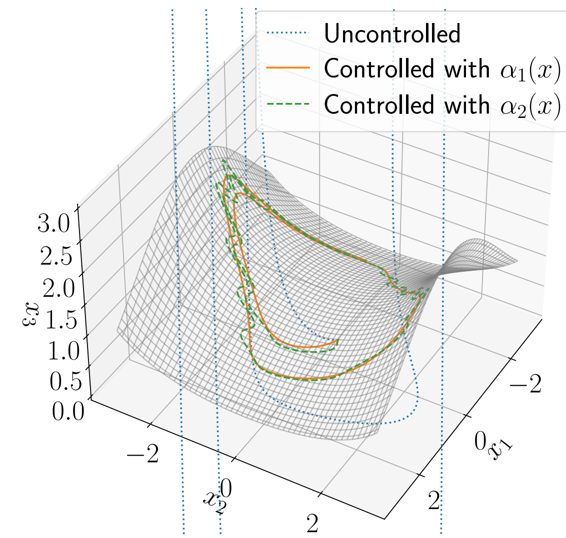

Fig. 1: Comparison of state trajectories. Dotted curve is uncontrolled trajectory, solid curve is controlled trajectory with $\beta = 0$, and dashed curve is controlled trajectory with $\beta = \sin(10x_1)/3$. Gray meshed surface depicts V.

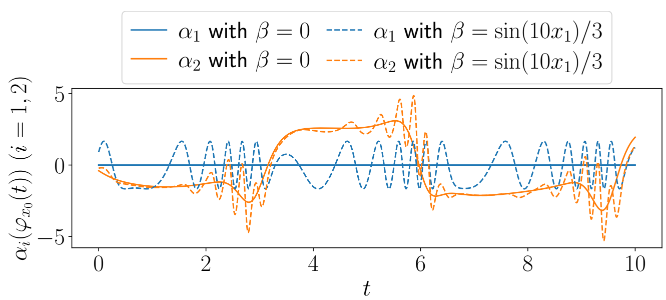

Fig. 2: Input sequences generated by feedback controller $\alpha(x) = [\alpha_1(x) \;\, \alpha_2(x)]^\top$ for case V-A.

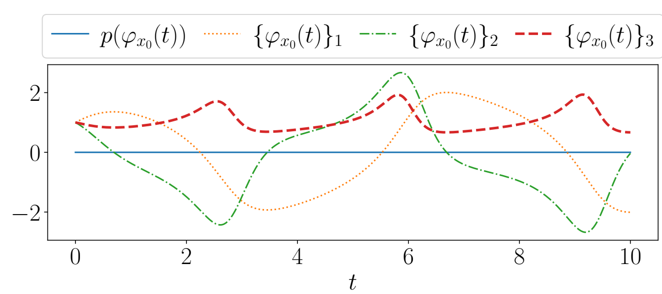

Fig. 3: Time series of $p(x)$ and each axis of $\varphi_{x_0}(t)$ for trajectory with $\beta = 0$. Solid curve shows $p(\varphi_{x_0}(t))$. Dotted, dash-dotted, and dashed curves show x_1-, x_2-, and x_3-components, respectively.

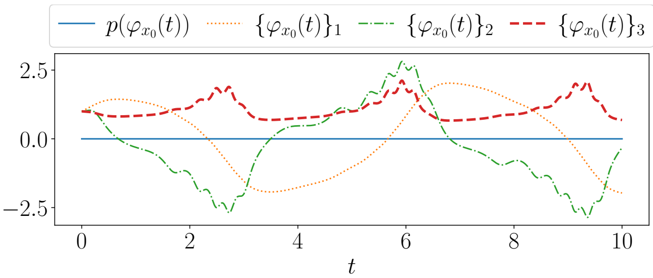

Fig. 4: Time series of $p(x)$ and each axis of $\varphi_{x_0}(t)$ for trajectory with $\beta = \sin(10x_1)/3$. Solid curve shows $p(\varphi_{x_0}(t))$. Dotted, dash-dotted, and dashed curves show x_1-, x_2-, and x_3-components, respectively.

B. Invariant set not considered in literature [17]

Consider another nonlinear function

$$p(x) = \log(x_1^2 + x_2^2 + 1) - \frac{x_3^2}{4} - 1,$$

which includes a logarithm and does not belong to the class of functions considered in [17] (see also Example 2). By inspecting $p(x)$ and its derivatives, the algebraic relations among them can be determined as follows:

$$(x_1^2 + x_2^2 + 1)x_3\partial_1 p + 4x_1\partial_3 p = 0,$$
$$(x_1^2 + x_2^2 + 1)x_3\partial_2 p + 4x_2\partial_3 p = 0.$$

These relations, combined with (11), lead to the following family of controllers:

$$\alpha(x) = \begin{bmatrix} x_1^2 - 1 \\ x_2 - x_1 \end{bmatrix} + \beta(x)\begin{bmatrix} -(x_1^2 + x_2^2 + 1)x_3^2 \\ -4x_2^2 \end{bmatrix}.$$

Note that the closed-loop system with $u = \alpha(x)$ satisfies $\mathcal{L}_{f+G\alpha}p = 0$ for any $\beta(x) \in C_1(x)$, which is a more restrictive condition than (4). This is because, for logarithms, b_k in Assumption 1 is always zero, which implies $\alpha_{m+1} = \lambda = 0$ in (10). Therefore, condition (10) is more restrictive than the general condition (4) when logarithms are included in $p(x)$ although it can be addressed by the proposed method.

Numerical simulations are also conducted for this case. The initial state is set to $x_0 = [2\ 1\ \sqrt{\log(6) - 1}]^\top$ such that $p(x_0) = 0$ and the time interval is set to $t \in [0, 10]$. Two closed-loop systems with $\beta(x) = 0$ and $\beta(x) = 1/(x_3^2 + 100)$ are considered. Fig. 5 shows the invariant set and the state trajectories obtained by both the controllers, while Fig. 6 shows the input sequences. Moreover, Figs. 7 and 8 show the time series for both the controllers. The trajectory with $\beta = 0$ depicts a circle with a constant x_3-coordinate, which is also observed in Fig. 5. This is because when $\beta = 0$, the closed-loop system is reduced to a linear system: $\dot{x}_1 = x_2, \dot{x}_2 = -x_1, \dot{x}_3 = 0$.

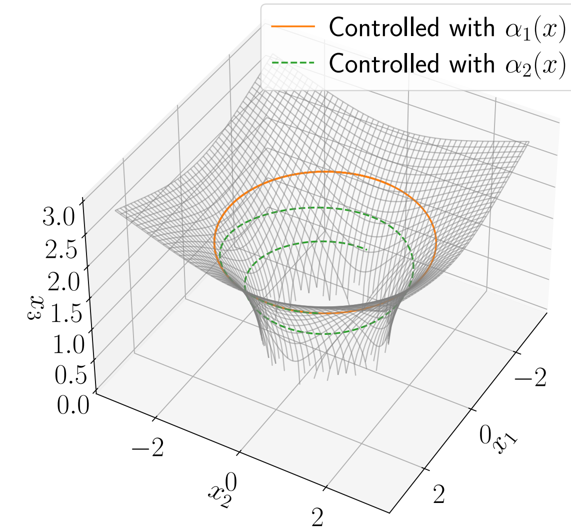

Fig. 5: Comparison of state trajectories. Solid curve is controlled trajectory with $\beta = 0$, and dashed curve is controlled trajectory with $\beta = 1/(x_3^2+100)$. Gray meshed surface depicts V.

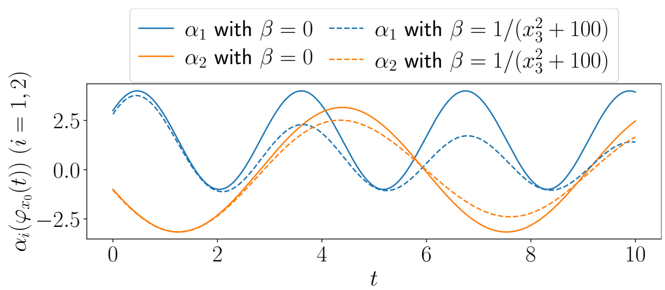

Fig. 6: Input sequences generated by $\alpha(x) = [\alpha_1(x)\ \alpha_2(x)]^\top$ for case V-B.

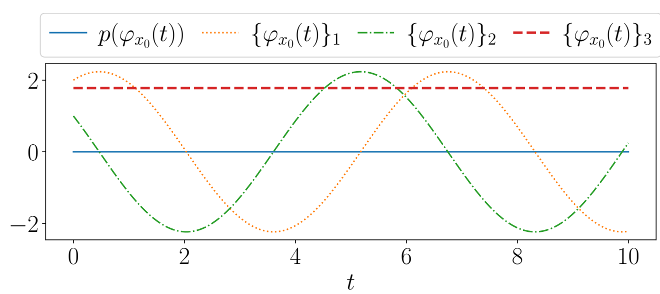

Fig. 7: Time series of $p(x)$ and each axis of $\varphi_{x_0}(t)$ for trajectory with $\beta = 0$. Solid curve shows $p(\varphi_{x_0}(t))$. Dotted, dash-dotted, and dashed curves show x_1-, x_2-, and x_3-components, respectively.

VI. Conclusion

This study proposes a controller synthesis method for polynomial systems that provides a set of controllers to achieve the invariance of a prescribed set. Compared with existing methods, the proposed method can handle the invariant sets defined with a wider class of nonlinear functions, which are defined by certain algebraic relations between the function and its derivatives. The construction of controllers is performed algorithmically using symbolic computations, except

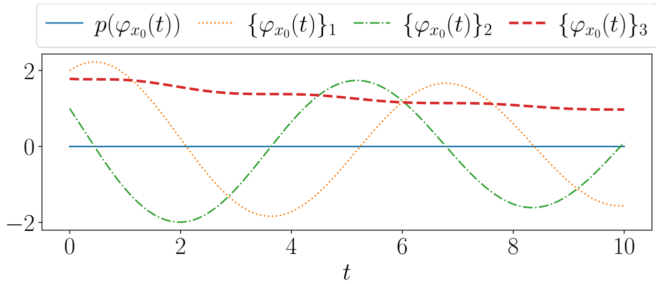

Fig. 8: Time series of $p(x)$ and each axis of $\varphi_{x_0}(t)$ for trajectory with $\beta = 1/(x_3^2 + 100)$. Solid curve shows $p(\varphi_{x_0}(t))$. Dotted, dash-dotted, and dashed curves show x_1-, x_2-, and x_3-components, respectively.

for finding algebraic relations. For future work, invariant sets defined by multiple nonlinear equations and inequalities would be considered.

References

[1] H. K. Khalil, *Nonlinear Systems*, 3rd ed., Prentice-Hall, 2002.

[2] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, 2nd ed., Birkhäuser, 2015.

[3] E. C. Kerrigan andJ. M. Maciejowski, "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control," in *Proceedings of IEEE Conference on Decision and Control (CDC2000)*, pp. 4951–4956, 2000.

[4] R. Gondhalekar, J. Imura, and K. Kashima, "Controlled invariant feasibility — A general approach to enforcing strong feasibility in MPC applied to move-blocking," *Automatica*, vol. 45, no. 12, pp. 2869–2875, 2009.

[5] A. Rantzer and S. Prajna, "On analysis and synthesis of safe control laws," in *Proceedings of the 42nd allerton conference on communication, control, and computing*, pp. 1468–1476, 2004.

[6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[7] M. Korda, D. Henrion, and C. N. Jones, "Convex computation of the maximum controlled invariant set for polynomial control systems," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2944–2969, 2014.

[8] M. Korda, "Computing controlled invariant sets from data using convex optimization," *SIAM Journal on Control and Optimization*, vol. 59, no. 5, pp. 2871–2899, 2020.

[9] Y. Gao, K. H. Johansson, and L. Xie, "Computing probabilistic controlled invariant set," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3138–3151, 2021.

[10] R. Dimitrova and R. Majumdar, "Deductive control synthesis for alternating-time logics," in *Proceedings of the 14th international conference on embedded software*, pp. 14:1–14:10, 2014.

[11] E. Zera and S. Walcher, "Controlled invariant hypersurfaces of polynomial control systems," *Qualitative Theory of Dynamical Systems*, vol. 11, no. 1, pp. 145–158, 2012.

[12] M. Harms, C. Schilli, and E. Zerz, "Polynomial control systems: invariant sets given by algebraic equations/inequations," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 677–680, 2017.

[13] D. A. Cox, J. B. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*, 4th ed., Springer-Verlag, 2015.

[14] T. Yuno and T. Ohtsuka, "Lie derivative inclusion with polynomial output feedback," *Transactions of the Institute of Systems, Control and Information Engineers*, vol. 28, no. 1, pp. 22–31, 2015.

[15] C. Schilli, E. Zerz, and V. Levandovskyy, "Controlled and conditioned invariance for polynomial and rational feedback systems," in *Advances in Delays and Dynamics: Algebraic and Symbolic Computation Methods in Dynamical Systems*, vol. 9, Springer, 2020.

[16] T. Yuno, E. Zerz, and T. Ohtsuka, "Invariance of a class of semialgebraic sets for polynomial systems with dynamic compensators," *Automatica*, vol. 122, no. 109243, 2020.

[17] M. Harms, C. Schilli, and E. Zerz, "Invariant sets for a class of nonlinear control systems tractable by symbolic computation," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3899–3903, 2023.

[18] T. Ohtsuka, "Model structure simplification of nonlinear systems via immersion," *IEEE Transactions on Automatic Control*, vo.. 50, no. 5, pp. 607–618, 2005.

[19] T. Iori and T. Ohtsuka, "Nonlinear Bayesian filtering via holonomic gradient method with quasi moment generating function," *Asian J. Control*, vol. 25, no. 4, pp. 2655–1670, 2023.

[20] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, "SINGULAR 4-2-0 —A computer algebra system for polynomial computations," 2019.

[21] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry," 2020.

[22] M. Noro, N. Takayama, H. Nakayama, K. Nishiyama, and K. Ohara, "Risa/Asir: A computer algebra system," 2020.

[23] G.-M. Greuel and G. Pfister, *A singular introduction to commutative algebra*, 2nd ed., Springer-Verlag, 2002.

[24] Z. Lin, L. Xu, and N. K. Bose, "A tutorial on Gröbner bases with applications in signals and systems," *IEEE Transactions on Circuits and Systems I*, vol. 55, no. 1, pp. 445–461, 2008.

[25] D. A. Cox, J. B. Little, and D. O'Shea, *Using algebraic geometry*, 2nd ed., Springer-Verlag, 2005.