Nonlinear Dyn (2021) 105:3139–3148
https://doi.org/10.1007/s11071-021-06768-7

ORIGINAL PAPER

Iterative stability analysis for general polynomial control systems

Bo Xiao · Hak-Keung Lam · Zhixiong Zhong

Received: 27 January 2021 / Accepted: 22 July 2021 / Published online: 25 August 2021

Abstract The main challenge of the stability analysis for general polynomial control systems is that non-convex terms exist in the stability conditions, which hinders solving the stability conditions numerically. Most approaches in the literature impose constraints on the Lyapunov function candidates or the non-convex related terms to circumvent this problem. Motivated by this difficulty, in this paper, we confront the non-convex problem directly and present an iterative stability analysis to address the long-standing problem in general polynomial control systems. Different from the existing methods, no constraints are imposed on the polynomial Lyapunov function candidates. Therefore, the limitations on the Lyapunov function candidate and non-convex terms are eliminated from the proposed analysis, which makes the proposed method more general than the state-of-the-art. In the proposed approach, the stability for the general polynomial model is analyzed and the original non-convex stability conditions are developed. To solve the non-convex stability conditions through the sum-of-squares programming, the iterative stability analysis is presented. The feasible solutions are verified by the original non-convex stability conditions to guarantee the asymptotic stability of the general polynomial system. The detailed simulation example is provided to verify the effectiveness of the proposed approach. The simulation results show that the proposed approach is more capable to find feasible solutions for the general polynomial control systems when compared with the existing ones.

Keywords General polynomial control system · Non-convex problem · Iterative stability analysis · Sum-of-squares (SOS)

1 Introduction

Polynomial control systems have been successfully applied to fulfilling different control objectives due to its rigorous mathematical framework and the ability to deal with nonlinearity. Based on the technique of sum-of-squares (SOS), there are promising research outcomes for polynomial control systems and SOS. For example, in [1], fault-tolerant control for nonlinear systems was conducted in the polynomial control systems. In [2], the fault-tolerant control was extended to polynomial fuzzy control systems. Converse SOS was discussed in [3] and the existence of a global polynomial Lyapunov function was discussed in [4]. In [5], the output-feedback sampled-data polynomial controller for nonlinear systems was designed. In [6], the input-delay for the sampled-data H_∞ control of

B. Xiao (✉)
Imperial College London, London, UK
e-mail: b.xiao@imperial.ac.uk

H.-K. Lam (✉)
King's College London, London, UK
e-mail: hak-keung.lam@kcl.ac.uk

Z. Zhong (✉)
Minjiang University, Fuzhou, China
e-mail: zhixiongzhong2012@126.com

polynomial systems was reported. The research on control design of polynomial systems with input saturation was reported in [7]. The research works focus on the domains of attraction for polynomial nonlinear systems were reported in [8,9] Also the approaches in [10–18] adopted polynomial terms in the fuzzy-model-based (FMB) control. It is worth mentioning that the SOS-based techniques also have the potential to be applied to solve the dynamic problems in the heat transfer investigation [19–21].

Polynomial control systems can also be regarded as a polynomial extension of linear control systems, in which the polynomial terms can be processed. When the order of all the polynomial terms in the polynomial control systems is reduced to 0, the polynomial control systems become linear control systems, suggesting its generalizability. For many reported control systems, such as the linear control systems and Takagi-Sugeno (T–S) FMB control systems, the techniques of linear matrix inequalities (LMIs) can be used to solve the stability conditions efficiently. For example, in [22–24], the H_∞ problem was investigated through LMIs. In [25–27], the LMIs techniques were adopted to deal with the time-delay control issues. Also, LMIs in T-S FMB control can be found in [28–32].

Despite the success of LMIs in control applications, LMIs can no longer be used for polynomial control systems due to the polynomial terms in the model and controller, which cannot be handled by the LMI solver. In contrast, the stability conditions of polynomial control systems can be represented by SOS and solved efficiently by a third party MATLAB® toolbox SOSTOOLS [33]. Adopting the polynomial control systems can have a range of applications. However, due to the polynomial Lyapunov function candidate, the stability conditions are not convex in most cases for general polynomial control systems.

It should be pointed out that in most of the literature regarding polynomial control systems, there are some constraints that need to be imposed on the polynomial Lyapunov function candidate. In [10–14], the polynomial Lyapunov function candidates are constrained to be dependent on only part of the state variables according to nonzeros rows in the input matrix $B(x(t))$. This constraint is considered to be strong since it often makes the polynomial Lyapunov function candidate in the form of constant Lyapunov function candidate in cases that $B(x(t))$ does not have all-zero rows. Therefore, this constraint introduces conservativeness into the stability analysis, forgoing the advantages of polynomial Lyapunov function candidates.

To make the stability analysis of general polynomial control systems possible, in [1,2], the polynomial Lyapunov function candidate can be dependent on all the state variables, and thus the general polynomial control systems can be analyzed. However, there is a bound/index required for the non-convex terms in the general polynomial control system, which is another constraint imposed on the stability analysis.

To remove the constraints on the general control systems, in [15], a two-step stability analysis was provided to solve the general polynomial control systems without any other constraints on the stability conditions. The results are more general, and thus more general forms of the polynomial Lyapunov function candidate can be adopted in the stability analysis and control synthesis. However, in the two-step approach, when the stability analysis fails to find a feasible solution in the first step or the second step, the solving process is terminated and no feasible solution can be found.

Motivated by this specific difficulty in general polynomial control systems, the non-convex problem is investigated in this paper. The purpose of this paper is to propose an iterative stability analysis for general polynomial control systems. In the proposed approach, the non-convex terms in the stability conditions are firstly omitted to make the stability conditions convex, then a predefined convex term will be added manually into the stability conditions to make the convex stability conditions easier to find a feasible initial solution. In order to keep the impact of the manually added term as small as possible, the stability conditions are rewritten as an optimization problem. By solving the optimization problem, the value of the manually added term will be minimized. Once the initial solutions are obtained, the non-convex terms are approximated by the initial solutions, which makes the non-convex stability conditions into convex and can be further solved by SOSTOOLS. If the newly obtained solutions satisfy the original non-convex stability conditions, the newly obtained solutions are qualified as feasible solutions for the general polynomial control system. If not, the non-convex terms are approximated by the newly obtained initial solutions to obtain the most updated solutions iteratively. The novelty and contribution of the paper are summarized as follows:

(1) The approximation of non-convex terms is presented to render the stability conditions into convex form for the general polynomial control systems.
(2) An iteration approximation approach is proposed that the intermediate solutions dependent on the solutions in both the previous and current iterations. The solutions are verified by the original non-convex stability conditions to guarantee the asymptotic stability of the general polynomial system.
(3) In the proposed stability analysis for general polynomial control systems, no constraint is imposed on the polynomial Lyapunov function candidate. Therefore, the polynomial Lyanpunov function candidate can depend on any system state variables.
(4) Compared with the works reported in [1,2,10–14], there is no constraints imposed on the polynomial Lyapunov functions in the proposed approach; Compared with the works reported in [15], the proposed approach is more general and has more potential to find feasible solutions.

This paper is organized as follows. In Sect. 2, the preliminaries of general polynomial control systems and polynomial Lyapunov function candidate will be introduced. In Sect. 3, the algorithm for iterative stability will be presented and discussed in detail. The simulation example and the comparison with the state-of-the-art are provided in Sect. 4. A conclusion is drawn in Sect. 5.

Notation: The following notations are employed throughout the paper, the monomial vector $x(t) = [x_1(t), \ldots, x_n(t)]^T$, the superscript T stands for matrix transposition, the superscript -1 in the expression $X(x(t))^{-1}$ stands for matrix inverse, the expressions of $P(x(t)) > 0$ represents the corresponding polynomial $p(x(t), \upsilon) = \upsilon^T P(x(t))\upsilon$ can be decomposed into SOS, where υ is a nonzero arbitrary vector with proper dimensions.

2 Preliminaries

Consider the following polynomial control system:

$$\begin{cases} \dot{x}(t) = A(x(t))x(t) + B(x(t))u(t) \\ u(t) = G(x(t))x(t), \end{cases} \tag{1}$$

where $A(x(t)) \in \mathbb{R}^{n\times n}$ and $B(x(t)) \in \mathbb{R}^{n\times m}$ are the system and input polynomial matrices. $G(x(t)) \in \mathbb{R}^{m\times n}$ is the polynomial feedback gain matrix and $u(t) \in \mathbb{R}^m$ is the control input.

The dynamics of closed-loop polynomial control system can be written as:

$$\dot{x}(t) = A(x(t))x(t) + B(x(t))G(x(t))x(t)$$

or

$$\dot{x}(t) = (A(x(t)) + B(x(t))G(x(t)))x(t).$$

To derive the stability conditions, it firstly needs to define the polynomial Lyapunov function candidate $V(t)$ as:

$$V(t) = x(t)^T X(x(t))^{-1} x(t), \tag{2}$$

where $X(x(t)) = X(x(t))^T > 0$.

The differential of $V(t)$ should be guaranteed to always be negative to ensure the asymptotic stability of the polynomial control system:

$$\begin{aligned}\dot{V}(t) &= \dot{x}(t)^T X(x(t))^{-1} x(t) + x(t)^T X(x(t))^{-1} \dot{x}(t) \\ &\quad + x(t)^T \dot{X}(x(t))^{-1} x(t).\end{aligned} \tag{3}$$

The expression $\dot{V}(t)$ can be further deducted using the dynamics of the closed-loop control system in (1) as follows:

$$\begin{aligned}\dot{V}(t) &= x(t)^T (A(x(t)) \\ &\quad + B(x(t))G(x(t)))^T X(x(t))^{-1} x(t) \\ &\quad + x(t)^T X(x(t))^{-1} (A(x(t)) \\ &\quad + B(x(t))G(x(t)))x(t) + x(t)^T \\ &\quad \left(\sum_{c=1}^{n} \frac{\partial X(x(t))^{-1}}{\partial x_c(t)} \frac{dx_c(t)}{dt} \right) x(t) < 0,\end{aligned} \tag{4}$$

where $\frac{dx_c(t)}{dt} = A^c(x(t))x(t) + B^c(x(t))G(x(t))x(t)$, $A^c(x(t)) \in \mathbb{R}^n$ and $B^c(x(t)) \in \mathbb{R}^m$ are the c-th rows of the matrices $A(x(t))$ and $B(x(t))$, respectively. To deal with the term $\sum_{c=1}^{n} \frac{\partial X(x(t))^{-1}}{\partial x_c(t)} \frac{dx_c(t)}{dt}$ in (4), let us introduce the following lemma [10] here.

Lemma 1

$$\frac{\partial X(x(t))^{-1}}{\partial x_c(t)}$$

$$= -X(x(t))^{-1}\frac{\partial X(x(t))}{\partial x_c(t)}X(x(t))^{-1}, \forall c = 1, \ldots, n. \tag{5}$$

Proof of Lemma 1 Given that

$$\frac{\partial I}{\partial x_c(t)} = 0, \tag{6}$$

replacing I by $X(x(t))^{-1}X(x(t))$, then (6) can be rewritten as

$$\frac{\partial X(x(t))^{-1}X(x(t))}{\partial x_c(t)} = 0. \tag{7}$$

□

Then it follows that:

$$X(x(t))^{-1}\frac{\partial X(x(t))}{\partial x_c(t)} + \frac{\partial X(x(t))^{-1}}{\partial x_c(t)}X(x(t)) = 0. \tag{8}$$

In addition, to make the stability analysis numerically feasible, adopting $z(t) = X(x(t))^{-1}x(t)$ to do variable replacement, we can further define:

$$\dot{V}(t) = z(t)^T Q(x(t))z(t) \tag{9}$$

where $Q(x(t)) = A(x(t))X(x(t)) + X(x(t)) A(x(t))^T + B(x(t))N(x(t)) + N(x(t))^T B(x(t))^T - \sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t))+B^c(x(t))N(x(t))X(x(t))^{-1}) x(t)$. The variable $N(x(t)) = G(x(t))X(x(t))$ is introduced, in which $X(x(t))$ and $N(x(t))$ are the decision variables. The polynomial feedback gains can be calculated through the relationship $G(x(t)) = N(x(t))X(x(t))^{-1}$ after solving the values of $X(x(t))$ and $N(x(t))$.

By guaranteeing $Q(x(t)) < 0$ (or $-Q(x(t))$ as SOS matrix), the closed-loop control system can be guaranteed to be stable. However, it can be found that this stability condition is not convex due to the non-convex terms in $Q(x(t))$, which makes SOSTOOLS cannot be applied to solve the stability conditions.

Remark 1 When compared with other non-general condition in [10–14], the existing conditions therein can be considered as the sub-set of the proposed approach since there are constraints imposed on the Lyapunov function candidate. To make it possible for SOSTOOLS to solve the stability conditions in a numerical way, the non-convex (nonlinear) term $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t))+B^c(x(t))N(x(t))X(x(t))^{-1}) x(t)$ needs special attention. In the following section, an iterative stability analysis is introduced to deal with this non-convex issue.

3 Iterative stability analysis

To process the non-convex term $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c (x(t))+B^c(x(t))N(x(t))X(x(t))^{-1})x(t)$ through SOS programming, an iterative stability analysis approach is presented in this paper.

From the definition of $N(x(t))$ and $X(x(t))^{-1}$, it can be shown that the following equation is always hold:

$$G(x(t)) = N(x(t))X(x(t))^{-1}. \tag{10}$$

It can be found that $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t)) + B^c(x(t))N(x(t))X(x(t))^{-1})x(t)$ can be rewritten as $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t)) + B^c(x(t))G(x(t)))x(t)$.

Remark 2 It should be pointed out that using the variable substitution $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t)) + B^c(x(t)) G(x(t)))x(t)$ will not make the stability condition convex. Since the variable $G(x(t))$ is not independent with the other two decision variables $N(x(t))$ and $X(x(t))$, therefore, all the three variables cannot all be the decision variables at the same time. In the following analysis, we will use initial solutions of $N(x(t))$ and $X(x(t))$ to approximate $G(x(t))$, which makes the iterative stability conditions convex.

The stability conditions can be rewritten as:

$$\begin{aligned} Q(x(t)) = {} & A(x(t))X(x(t)) + X(x(t))A(x(t))^T \\ & + B(x(t))N(x(t)) + N(x(t))^T B(x(t))^T \\ & - \sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t)) \\ & + B^c(x(t))G(x(t)))x(t) < 0, \end{aligned} \tag{11}$$

$$X(x(t)) > 0. \tag{12}$$

To apply iterative stability analysis, let us start with $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(A^c(x(t)) + B^c(x(t))G(x(t)))x(t) = 0$ as the 1-st iterative stability condition and solve the SOS-based iterative stability condition:

minimize α subject to:

$$A(x(t))X(x(t)) + X(x(t))A(x(t))^T + B(x(t))N(x(t)) + N(x(t))^T B(x(t))^T - \alpha M(x(t)) < 0, \quad (13)$$

$$X(x(t)) > 0, \quad (14)$$

where $M(x(t)) \in \mathbb{R}^{n\times n}$ is a predefined polynomial matrix and $M(x(t)) > 0$. $\alpha M(x(t))$ is a manually added term to make the condition easier to be feasible. In the corresponding SOS conditions, $X(x(t))$ and $N(x(t))$ are the decision variables to be solved by SOSTOOLS.

It can be seen that (13) and (14) are convex conditions, and thus it can be solved by SOS solver. After (13) and (14) are solved, $X^{(1)}(x(t))$, $N^{(1)}(x(t))$ are used as the initial solutions for the 1-st iteration as the superscript (1) indicates. From the initial solutions $X^{(1)}(x(t))$, $N^{(1)}(x(t))$, $G^{(1)}(x(t))$ can be calculated as $G^{(1)}(x(t)) = N^{(1)}(x(t))X^{(1)}(x(t))^{-1}$.

It should be noted that the term $X^{(1)}(x(t))^{-1}$ will still make the stability conditions non-convex. From the definition of the inverse matrix, we have:

$$X^{(1)}(x(t))^{-1} = \frac{\operatorname{adj}(X^{(1)}(x(t)))}{\det(X^{(1)}(x(t)))},$$

where $\operatorname{adj}(X^{(1)}(x(t)))$ is the adjoint matrix of $X^{(1)}(x(t))$ and $\det(X^{(1)}(x(t)))$ is the determinant of $X^{(1)}(x(t))$.

Adopting $\operatorname{adj}(X^{(1)}(x(t)))$ and $\det(X^{(1)}(x(t)))$, we have:

$$G^{(1)}(x(t)) = N^{(1)}(x(t))X^{(1)}(x(t))^{-1} = N^{(1)}(x(t))\frac{\operatorname{adj}(X^{(1)}(x(t)))}{\det(X^{(1)}(x(t)))}. \quad (15)$$

For any $X(x(t)) > 0$, it is always true that $\det(X(x(t))) > 0$. Therefore, the iterative stability conditions can be defined as following:

$$\begin{aligned} Q^{(1)}(x(t)) &= \det(X^{(1)}(x(t)))\big(A(x(t))X^{(1)}(x(t)) \\ &\quad + X^{(1)}(x(t))A(x(t))^T \\ &\quad + B(x(t))N^{(1)}(x(t)) + N^{(1)}(x(t))^T B(x(t))^T\big) \\ &\quad - \sum_{c=1}^{n} \frac{\partial X^{(1)}(x(t))}{\partial x_c}(\det(X^{(1)}(x(t)))A^c(x(t)) \\ &\quad + B^c(x(t))N^{(1)}(x(t))\operatorname{adj}(X^{(1)}(x(t))))x(t) < 0, \end{aligned} \quad (16)$$

$$X^{(1)}(x(t)) > 0. \quad (17)$$

The following stability conditions can be verified:

$$Q^{(1)}(x(t)) < 0, \quad (18)$$

$$X^{(1)}(x(t)) > 0. \quad (19)$$

If the stability conditions in (18) and (19) are valid, the polynomial control system is guaranteed to be stable, $X(x(t))$, $N(x(t))$ and $G(x(t))$ are feasible solutions. Otherwise, substitute $G^{(1)}(x(t))$ into (11), and solve (11) again with $G(x(t)) = G^{(1)}(x(t))$ as *a prior*:

$$\begin{aligned} &\det(X^{(1)}(x(t)))(A(x(t))X(x(t)) + X(x(t))A(x(t))^T \\ &\quad + B(x(t))N(x(t)) + N(x(t))^T B(x(t))^T) \\ &\quad - \sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(\det(X^{(1)}(x(t)))A^c(x(t)) \\ &\quad + B^c(x(t))N^{(1)}(x(t))\operatorname{adj}(X^{(1)}(x(t))))x(t) < 0, \end{aligned} \quad (20)$$

$$X(x(t)) > 0, \quad (21)$$

$$X(x(t)) - X^{(1)}(x(t)) - \beta_1^{(1)} I < 0, \quad (22)$$

$$X^{(1)}(x(t)) - X(x(t)) - \beta_2^{(1)} I < 0, \quad (23)$$

$$N(x(t)) - N^{(1)}(x(t)) - \beta_3^{(1)} I < 0, \quad (24)$$

$$N^{(1)}(x(t)) - N(x(t)) - \beta_4^{(1)} I < 0 \quad (25)$$

where $\beta_1^{(1)}, \beta_2^{(1)}, \ldots, \beta_4^{(1)}$ are predefined sufficiently small constants for the 1-st iteration. By solving the SOS-based stability condition in (20) to (25), we can obtain the solution for the 2-nd iteration: $X^{(2)}(x(t))$ and $N^{(2)}(x(t))$. Also, $G^{(2)}(x(t)) = N^{(2)}(x(t))X^{(2)}(x(t))^{-1}$. In the stability conditions, since the non-convex terms are approximated by the initial solutions, the feasible solutions are dependent on both the current and the initial solutions. Therefore, the parameters $\beta_1^{(1)}, \beta_2^{(1)}, \ldots, \beta_4^{(1)}$ are used to relate the current and initial solutions in a numerical way.

Remark 3 It is worth mentioning that in the above stability conditions, the non-convex terms are approximated by $\sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(\det(X^{(1)}(x(t)))A^c(x(t)) + B^c(x(t))N^{(1)}(x(t))\operatorname{adj}(X^{(1)}(x(t)))x(t)$, which are the solutions from the previous iteration and do not need to be solved again in the current iteration. Therefore, the stability conditions are convex for SOS programming.

Let's check whether the iterative stability conditions are SOS:

$$Q^{(2)}(x(t)) < 0, \tag{26}$$

$$X^{(2)}(x(t)) > 0, \tag{27}$$

if not, repeat the iteration using solutions $X^{(2)}(x(t))$ and $N^{(2)}(x(t))$ as *a prior*.

In a more general form, for the k-th iteration, the iterative stability conditions can be expressed as follows:

$$\begin{aligned}
&\det(X^{(k)}(x(t)))(A(x(t))X(x(t)) + X(x(t))A(x(t))^T \\
&\quad + B(x(t))N(x(t)) + N(x(t))^T B(x(t))^T) \\
&\quad - \sum_{c=1}^{n} \frac{\partial X(x(t))}{\partial x_c}(\det(X^{(k)}(x(t)))A^c(x(t)) \\
&\quad + B^c(x(t))N^{(k)}(x(t))\operatorname{adj}(X^{(k)}(x(t))))x(t) < 0,
\end{aligned} \tag{28}$$

$$X(x(t)) > 0, \tag{29}$$

$$X(x(t)) - X^{(k)}(x(t)) - \beta_1^{(k)} I < 0, \tag{30}$$

$$X^{(k)}(x(t)) - X(x(t)) - \beta_2^{(k)} I < 0, \tag{31}$$

$$N(x(t)) - N^{(k)}(x(t)) - \beta_3^{(k)} I < 0, \tag{32}$$

$$N^{(k)}(x(t)) - N(x(t)) - \beta_4^{(k)} I < 0 \tag{33}$$

where $\beta_1^{(k)}, \beta_2^{(k)}, \ldots, \beta_4^{(k)}$ are predefined sufficiently small constants to be determined for k-th iteration. In the iterative stability analysis, since the non-convex terms are approximated by the previous solutions, the feasible solutions are dependent on both the current solutions and the previous solutions. Therefore, the parameters $\beta_1^{(k)}, \beta_2^{(k)}, \ldots, \beta_4^{(k)}$ are used to relate the solutions in two consecutive iterations in a numerical way.

By solving the SOS-based stability conditions in (28) to (33), we can obtain the $k+1$-th iterative solution: $X^{(k+1)}(x(t))$ and $N^{(k+1)}(x(t))$. Also, $G^{(k+1)}(x(t)) = N^{(k+1)}(x(t))X^{(k+1)}(x(t))^{-1}$.

Until

$$Q^{(k+1)}(x(t)) < 0, \tag{34}$$

$$X^{(k+1)}(x(t)) > 0. \tag{35}$$

Then $X^{(k+1)}(x(t))$, $N^{(k+1)}(x(t))$ and $G^{(k+1)}(x(t))$ are feasible solutions for the general polynomial control system.

The iterative stability analysis of the general polynomial control system is summarized in Algorithm. 1.

Algorithm 1 Algorithm for iterative stability analysis

```
Choose the order of the polynomial Lyapunov function candidate $V(t)$ and the order of the decision variable $N(x(t))$.
Set the maximal number of iterations $K$.
Solve $A(x(t))X(x(t))+X(x(t))A(x(t))^T+B(x(t))N(x(t))+N(x(t))^T B(x(t))^T - \alpha M(x(t)) < 0$ and $X(x(t)) > 0$ to minimize $\alpha$.
Save the initial solution $N^{(1)}(x(t))$ and $X^{(1)}(x(t))$.
Approximate the non-convex term using prepared solutions: $G^{(1)}(x(t)) = N^{(1)}(x(t))X^{(1)}(x(t))^{-1}$.
for 1 : $K$-th iteration do
    Set the values for $\beta_1^{(k)}, \beta_1^{(k)}, \ldots, \beta_4^{(k)}$.
    Solve the stability conditions from (28) to (33).
    Save the solutions $N^{(k+1)}(x(t))$ and $X^{(k+1)}(x(t))$.
    if $Q^{(k+1)}(x(t)) < 0$ & $X^{(k+1)}(x(t)) > 0$ is valid then
        Terminate and return $X^{(k+1)}(x(t))$, $N^{(k+1)}(x(t))$ and $G^{(k+1)}(x(t))$ as feasible solutions.
    else Store $X^{(k+1)}(x(t))$ and $N^{(k+1)}(x(t))$ as the initial solutions for the non-convex term in the next iteration.
    end if
end for
No feasible solution can be found.
```

4 Simulation example

In this section, a numerical example is provided to verify the effectiveness of the iterative stability analysis. Let us consider a polynomial control system, where $x(t) = [x_1(t), x_2(t)]^T$:

$$\begin{aligned}
&A(x(t)) \\
&\quad = \begin{bmatrix} -1 + x_1(t) + \frac{3}{4}x_1(t)^2 - \frac{3}{2}x_2(t)^2 & \frac{1}{4} - x_1(t)^2 \\ x_1(t) & x_2(t) \end{bmatrix},
\end{aligned}$$

$$B(x(t)) = \begin{bmatrix} 1 & 0 \\ x_1(t) & 2 \end{bmatrix}.$$

In the simulation, $X(x(t))$ is chosen as a polynomial matrix with 2-nd order of $x_1(t)$ and $x_2(t)$, $N(x(t))$ is chosen as a polynomial matrix with 2-nd order of $x_1(t)$. $M(x(t))$ is chosen as the identity matrix. $\beta_1^{(k)} = \beta_2^{(k)} = 0.2$ and $\beta_3^{(k)} = \beta_4^{(k)} = 0.05$ for every iteration.

Firstly, the stability conditions in (13) and (14) are adopted to find the initial solutions. Then the solutions

$X^{(1)}(x(t))$, $N^{(1)}(x(t))$ can be obtained. Using the initial solution to approximate the non-convex term and re-solve the solutions for the convex stability conditions, after the 1-st iteration feasible solution can be found.

For the $N(x(t))$ matrix, we rewrite $N(x(t))$ as:

$$N(x(t)) = \begin{bmatrix} N_{11}(x(t)) & N_{12}(x(t)) \\ N_{21}(x(t)) & N_{22}(x(t)) \end{bmatrix}$$

where

$$\begin{aligned} N_{11}(x(t)) &= -3.746 \times 10^{-6} x_1(t)^2 \\ &\quad - 0.0001958 x_1(t) - 0.5418, \\ N_{12}(x(t)) &= -1.112 \times 10^{-6} x_1(t)^2 \\ &\quad + 0.003416 x_1(t) + 0.07179, \\ N_{21}(x(t)) &= 0.03998 x_1(t)^2 + 0.09521 x_1(t) + 0.282, \\ N_{22}(x(t)) &= -0.278 x_1(t)^2 - 0.02907 x_1(t) - 1.009. \end{aligned}$$

For the $X(x(t))$ matrix, we define

$$X(x(t)) = \begin{bmatrix} X_{11}(x(t)) & X_{12}(x(t)) \\ X_{21}(x(t)) & X_{22}(x(t)) \end{bmatrix}$$

where

$$\begin{aligned} X_{11}(x(t)) &= 5.266 \times 10^{-9} x_1(t)^2 + 6.099 \times 10^{-7} x_1(t) \\ &\quad + 1.299 \times 10^{-8} x_2(t)^2 + 0.1238, \\ X_{12}(x(t)) &= 2.785 \times 10^{-9} x_1(t)^2 - 3.664 \times 10^{-7} x_1(t) \\ &\quad + 8.491 \times 10^{-10} x_2(t)^2 + 0.1132, \\ X_{21}(x(t)) &= 2.785 \times 10^{-9} x_1(t)^2 - 3.664 \times 10^{-7} x_1(t) \\ &\quad + 8.491 \times 10^{-10} x_2(t)^2 + 0.1132, \\ X_{22}(x(t)) &= 3.536 \times 10^{-9} x_1(t)^2 - 7.849 \times 10^{-7} x_1(t) \\ &\quad + 1.918 \times 10^{-9} x_2(t)^2 + 0.155. \end{aligned}$$

In order to have a better understanding of the time-response of states $x_1(t)$ and $x_2(t)$, the time response of $x_1(t)$ and $x_2(t)$ can be viewed in Figs. 1 and 2. In the figures, it can be observed that the polynomial control system can be stabilized swiftly from 4 different initial states: the bold blue curve represents the state response with initial state $x(0) = [-1.5, 3]^T$; The dashed bold red curve represents the state response with initial state $x(0) = [1.5, 3]^T$; The green curve represents the state response with initial state $x(0) = [3, -1]^T$; The dashed black curve represents the state response with initial state $x(0) = [-3, -1]^T$.

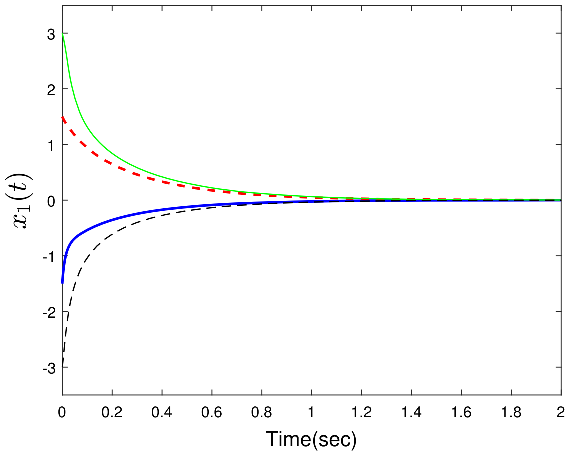

Fig. 1 The time-response simulation for $x_1(t)$ from different initial states

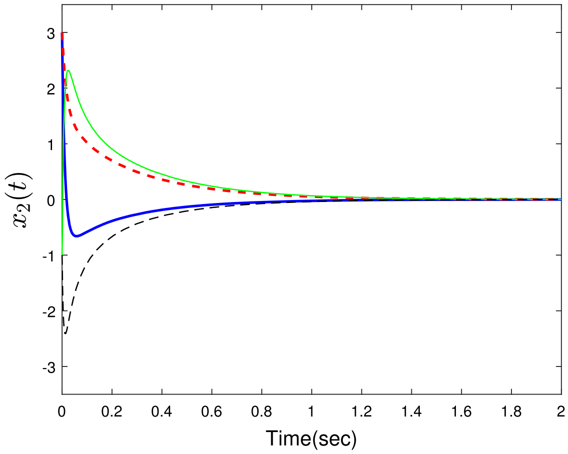

Fig. 2 The time-response simulation for $x_2(t)$ from different initial states

In addition, the 2-dimensional control input during the control process can be viewed in Figs. 3 and 4. In these figures, it can be seen that the two control inputs are within reasonable ranges. As the same in the time-response simulations: the bold blue curve represents the control input with initial state $x(0) = [-1.5, 3]^T$; The dashed bold red curve represents the control input with initial state $x(0) = [1.5, 3]^T$; The green curve represents the control input with initial state $x(0) = [3, -1]^T$; The dashed black curve represents the control input with initial state $x(0) = [-3, -1]^T$.

The vector field and phase plot of the polynomial control system with 12 initial start states can be viewed in Fig. 5. In the figure, the black circles represent the ini-

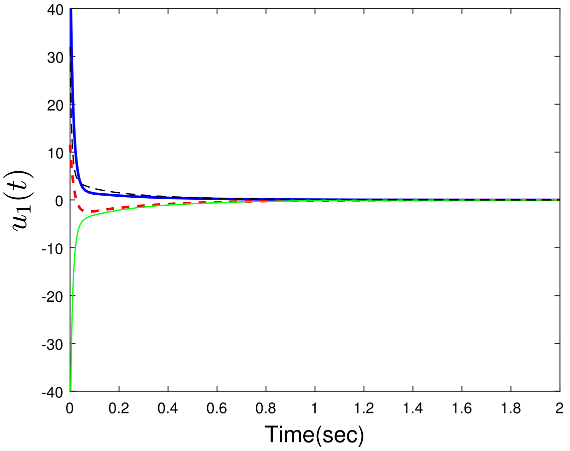

Fig. 3 The control input $u_1(t)$ from 4 different initial states

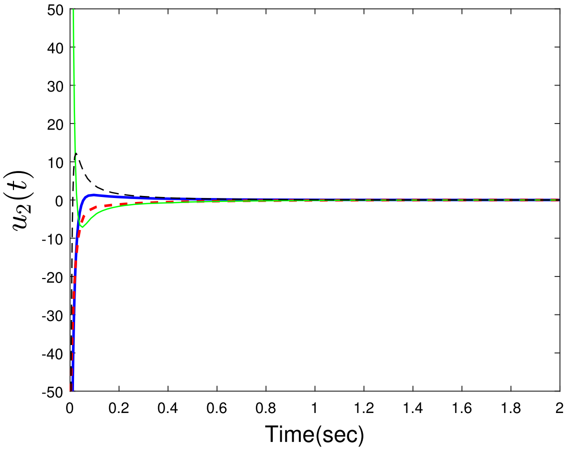

Fig. 4 The control input $u_2(t)$ from 4 different initial states

tial state for every trajectory. The blue arrows represent the vector fields in terms of size and direction. From Fig. 5, it can be seen that all the phase flows (red trajectories) start from different initial states (black circles) follow the vector field (blue arrows) into the stabilization state smoothly.

To compare our method with the two-step approach reported in [15], detailed simulations have been done using the two-step approach with the same polynomial model and all the other parameters, such as the same degrees of $X(x(t))$, $N(x(t))$, etc.

From the simulation results, it is found that the two-step approach is also able to find a feasible solution $N(x(t))$ as:

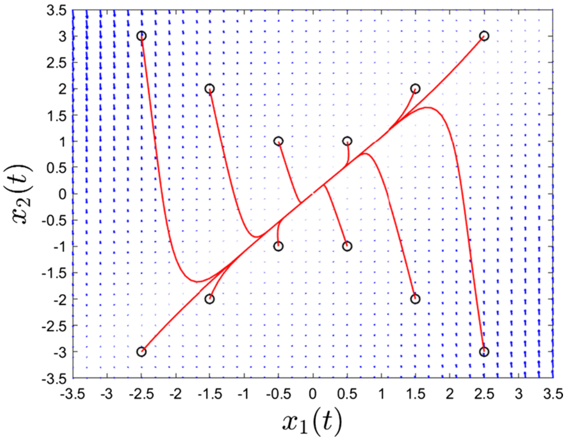

Fig. 5 Phase plot of the polynomial control system

$$\begin{aligned}
N_{11}(x(t)) &= -3.746\times10^{-6}x_1(t)^2\\
&\quad -0.0001975x_1(t)-0.5249,\\
N_{12}(x(t)) &= -1.112\times10^{-6}x_1(t)^2\\
&\quad -0.003686x_1(t)+0.07221,\\
N_{21}(x(t)) &= 0.002368x_1(t)^2+0.2854x_1(t)+0.1736,\\
N_{22}(x(t)) &= -0.278x_1(t)^2-0.02918x_1(t)-1.003.
\end{aligned}$$

and $X(x(t))$ as:

$$\begin{aligned}
X_{11}(x(t)) &= 3.877\times10^{-9}x_1(t)^2-9.467\times10^{-9}x_1(t)\\
&\quad +1.36\times10^{-8}x_2(t)^2+0.0002032,\\
X_{12}(x(t)) &= 3.351\times10^{-9}x_1(t)^2-2.408\times10^{-9}x_1(t)\\
&\quad +9.92\times10^{-10}x_2(t)^2+0.0002164,\\
X_{21}(x(t)) &= 3.351\times10^{-9}x_1(t)^2-2.408\times10^{-9}x_1(t)\\
&\quad +9.92\times10^{-10}x_2(t)^2+0.0002164,\\
X_{22}(x(t)) &= 2.259\times10^{-9}x_1(t)^2-3.979\times10^{-9}x_1(t)\\
&\quad +3.048\times10^{-9}x_2(t)^2+0.004621.
\end{aligned}$$

However, when the order of $N(x(t))$ is set to 0, the two-step approach cannot find a feasible solution while the iterative stability analysis is still able to find a feasible solution at the 2-nd iteration. The feasible solutions are:

$$N(x(t)) = \begin{bmatrix} -0.05362 & -0.006776\\ 0.1625 & -1.285 \end{bmatrix}$$

and $X(x(t))$ as

$$
\begin{aligned}
X_{11}(x(t)) &= 1.204 \times 10^{-8} x_1(t)^2 + 3.592 \times 10^{-7} x_1(t) \\
&\quad + 4.43 \times 10^{-8} x_2(t)^2 + 0.07741, \\
X_{12}(x(t)) &= 8.896 \times 10^{-9} x_1(t)^2 + 1.581 \times 10^{-7} x_1(t) \\
&\quad + 4.474 \times 10^{-9} x_2(t)^2 + 0.08018, \\
X_{21}(x(t)) &= 8.896 \times 10^{-9} x_1(t)^2 + 1.581 \times 10^{-7} x_1(t) \\
&\quad + 4.474 \times 10^{-9} x_2(t)^2 + 0.08018, \\
X_{22}(x(t)) &= 7.972 \times 10^{-9} x_1(t)^2 + 1.051 \times 10^{-7} x_1(t) \\
&\quad + 2.68 \times 10^{-9} x_2(t)^2 + 0.09714.
\end{aligned}
$$

Remark 4 From the comparison with the two-step approach reported in [15], it can be found that the proposed approach has more potential to find feasible solutions for the general polynomial control systems. In addition, when the manually convex term added in the first step of the two-step approach to form the optimization problem, the two-step approach can be considered as a special case in the iteration approach with only 1 iteration.

Remark 5 Having demonstrated the merits of the iterative approach, the shortcoming of the proposed method is that it demands more computational resources when compared with the existing methods. In this paper, the simulations are conducted based on Matlab® 2018a, and the computer is equipped with Intel Core i7-7700K along with 16GB memory. The compared two-step approach took 9.60s, the proposed method took 20.46s. From the simulation time, it can be found that it takes longer for the iterative method to find feasible solutions.

5 Conclusion

In conclusion, to confront the non-convex problem directly, an iterative stability analysis is presented for general polynomial control systems. Unlike most of the methods reported in the literature, there is no constraint imposed on the forms of the polynomial Lyapunov function candidates or the non-convex terms. Furthermore, the relationship between the solutions of successive iterations is utilized to obtain the feasible solutions, which are verified by the original non-convex conditions. In addition, the comparison between the proposed approach and the current state-of-the-art has been conducted. From the comparison results, it demonstrates that the proposed method has more potential to find feasible solutions than the reported methods, which shows the effectiveness of the method in this paper.

Acknowledgements This work was supported by Imperial College London and King's College London.

Data Availability Statement Data sharing is not applicable to this article since no associated data.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ma, H.-J., Yang, G.-H.: Fault-tolerant control synthesis for a class of nonlinear systems: sum of squares optimization approach. Int. J. Robust Nonlinear Control **19**(5), 591–610 (2009)
2. Ye, D., Diao, N.-N., Zhao, X.-G.: Fault-tolerant controller design for general polynomial-fuzzy-model-based systems. IEEE Trans. Fuzzy Syst. **26**(2), 1046–1051 (2017)
3. Peet, M.M., Papachristodoulou, A.: A converse sum of squares Lyapunov result with a degree bound. IEEE Trans. Autom. Control **57**(9), 2281–2293 (2012)
4. Ahmadi, A.A., Parrilo, P.A.: Sum of squares certificates for stability of planar, homogeneous, and switched systems. IEEE Trans. Autom. Control **62**(10), 5269–5274 (2017)
5. Lam, H.K.: Output-feedback sampled-data polynomial controller for nonlinear systems. Automatica **47**(11), 2457–2461 (2011)
6. Kim, H.S., Park, J.B., Joo, Y.H.: Input-delay approach to sampled-data H_∞ control of polynomial systems based on a sum-of-square analysis. IET Control Theory Appl. **11**(9), 1474–1484 (2017)
7. Vafamand, N., Mardani, M.-M., Khayatian, A., Shasadeghi, M.: Non-iterative sos-based approach for guaranteed cost control design of polynomial systems with input saturation. IET Control Theory Appl. **11**(16), 2724–2730 (2017)

8. Topcu, U., Packard, A., Seiler, P.: Local stability analysis using simulations and sum-of-squares programming. Automatica **44**(10), 2669–2675 (2008)
9. Zarei, M., Kalhor, A., Brake, D.: Arc length based maximal lyapunov functions and domains of attraction estimation for polynomial nonlinear systems. Automatica **90**, 164–171 (2018)
10. Tanaka, K., Yoshida, H., Ohtake, H., Wang, H.O.: A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems. IEEE Trans. Fuzzy Syst. **17**(4), 911–922 (2009)
11. Tanaka, K., Ohtake, H., Wang, H.O.: Guaranteed cost control of polynomial fuzzy systems via a sum of squares approach. IEEE Trans. Syst. Man Cybern. Part B Cybern. **39**(2), 561–567 (2009)
12. Xiao, B., Lam, H.K., Li, H.: Stabilization of interval type-2 polynomial-fuzzy-model-based control systems. IEEE Trans. Fuzzy Syst. **25**(1), 205–217 (2017)
13. Xiao, B., Lam, H.K., Yu, Y., Li, Y.: Sampled-data output-feedback tracking control for interval type-2 polynomial fuzzy systems. IEEE Trans. Fuzzy Syst. **28**(3), 424–433 (2020)
14. Lam, H.K., Tsai, S.-H.: Stability analysis of polynomial-fuzzy-model-based control systems with mismatched premise membership functions. IEEE Trans. Fuzzy Syst. **22**(1), 223–229 (2014)
15. Lam, H.K., Wu, L., Lam, J.: Two-step stability analysis for general polynomial-fuzzy-model-based control systems. IEEE Trans. Fuzzy Syst. **23**(3), 511–524 (2015)
16. Lam, H.K.: A review on stability analysis of continuous-time fuzzy-model-based control systems: From membership-function-independent to membership-function-dependent analysis. Eng. Appl. Artif. Intell. **67**, 390–408 (2018)
17. Xiao, B., Lam, H.K., Zhong, Z., Wen, S.: Membership-function-dependent stabilization of event-triggered interval type-2 polynomial fuzzy-model-based networked control systems. IEEE Trans. Fuzzy Syst. (2019). https://doi.org/10.1109/TFUZZ.2019.2957256
18. Xiao, B., Lam, H.-K., Zhou, H., Gao, J.: Analysis and design of interval type-2 polynomial-fuzzy-model-based networked tracking control systems. IEEE Trans. Fuzzy Syst. (2020). https://doi.org/10.1109/TFUZZ.2020.3006587
19. Khan, N.S., Kumam, P., Thounthong, P.: Magnetic field promoted irreversible process of water based nanocomposites with heat and mass transfer flow. Sci. Rep. **11**(1), 1–25 (2021)
20. Khan, N.S., Ali, L., Ali, R., Kumam, P., Thounthong, P.: A novel algorithm for the computation of systems containing different types of integral and integro-differential equations. Heat Transfer **50**(4), 3065–3078 (2021)
21. Khan, N.S., Kumam, P., Thounthong, P.: Computational approach to dynamic systems through similarity measure and homotopy analysis method for renewable energy. Crystals **10**(12), 1086 (2020)
22. Gahinet, P., Apkarian, P.: A linear matrix inequality approach to H_∞ control. Int. J. Robust Nonlinear Control **4**(4), 421–448 (1994)
23. Chilali, M., Gahinet, P.: H_∞ design with pole placement constraints: an LMI approach. IEEE Trans. Autom. Control **41**(3), 358–367 (1996)
24. Masubuchi, I., Kamitane, Y., Ohara, A., Suda, N.: H_∞ control for descriptor systems: a matrix inequalities approach. Automatica **33**(4), 669–673 (1997)
25. Fridman, E., Shaked, U.: A descriptor system approach to H_∞ control of linear time-delay systems. IEEE Trans. Autom. Control **47**(2), 253–270 (2002)
26. Lin, C., Wang, Q.-G., Lee, T.H.: A less conservative robust stability test for linear uncertain time-delay systems. IEEE Trans. Autom. Control **51**(1), 87–91 (2006)
27. Gao, H., Chen, T.: New results on stability of discrete-time systems with time-varying state delay. IEEE Trans. Autom. Control **52**(2), 328–334 (2007)
28. Guan, X., Chen, C.: Delay-dependent guaranteed cost control for T-S fuzzy systems with time delays. IEEE Trans. Fuzzy Syst. **12**(2), 236–249 (2004)
29. Zhou, S., Li, T.: Robust stabilization for delayed discrete-time fuzzy systems via basis-dependent Lyapunov–Krasovskii function. Fuzzy Sets Syst. **151**(1), 139–153 (2005)
30. Wu, H.-N.: Delay-dependent stability analysis and stabilization for discrete-time fuzzy systems with state delay: a fuzzy Lyapunov–Krasovskii functional approach. IEEE Trans. Syst. Man Cybern. Part B Cybern. **36**(4), 954–962 (2006)
31. Wu, L., Su, X., Shi, P., Qiu, J.: Model approximation for discrete-time state-delay systems in the T–S fuzzy framework. IEEE Trans. Fuzzy Syst. **19**(2), 366–378 (2011)
32. Wu, L., Su, X., Shi, P., Qiu, J.: A new approach to stability analysis and stabilization of discrete-time T-S fuzzy time-varying delay systems. IEEE Trans. Syst. Man Cybern. Part B Cybern. **41**(1), 273–286 (2011)
33. Prajna, S., Papachristodoulou, A., Parrilo, P.A.: "SOSTOOLS: sum of squares optimization toolbox for Matlab. User's guide," Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, **91125**, (2004)