Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Control synthesis for polynomial nonlinear systems and application in attitude control*

Chang-fei TONG†, Hui ZHANG†‡, You-xian SUN
(*Institute of Industrial Process Control, Zhejiang University, Hangzhou 310027, China*)
†E-mail: cftong@iipc.zju.edu.cn; zhanghui@iipc.zju.edu.cn
Received Sept. 11, 2007; revision accepted Mar. 10, 2008

Abstract: A method for positive polynomial validation based on polynomial decomposition is proposed to deal with control synthesis problems. Detailed algorithms for decomposition are given which mainly consider how to convert coefficients of a polynomial to a matrix with free variables. Then, the positivity of a polynomial is checked by the decomposed matrix with semidefinite programming solvers. A nonlinear control law is presented for single input polynomial systems based on the Lyapunov stability theorem. The control synthesis method is advanced to multi-input systems further. An application in attitude control is finally presented. The proposed control law achieves effective performance as illustrated by the numerical example.

Key words: Nonlinear control, Attitude control, Polynomial systems
doi:10.1631/jzus.A071486 **Document code:** A **CLC number:** TP273

INTRODUCTION

In recent years, considerable attention has been devoted to the study of polynomial nonlinear systems. Significant progress has been made in the stability analysis of those systems by sum of squares (SOS) decomposition approach (Parrilo, 2000; Papachristodoulou and Prajna, 2005; Fisher and Bhattacharya, 2007). Stability analysis with this methodology is mainly based on the Lyapunov stability theorem. Lyapunov functions are constructed by SOSTOOLS (Parrilo, 2000) which converts the problems into semidefinite programs (SDPs) with the SDP problems being further solved by SeDuMi (Sturm, 1999). Furthermore, Chesi *et al*. considered the stability of uncertain polynomial systems (Chesi *et al.*, 2005; Chesi, 2007).

Though stability analysis is solved effectively in SOS approach, control synthesis for nonlinear systems still remains a stubborn problem since the nonlinear components of variables in the SOS terms cannot be solved directly by SOSTOOLS. To solve the synthesis problem, an iterative algorithm was proposed in (Jarvis-Wloszek, 2003) to obtain a stabilized controller. However, the controller is not globally optimal; furthermore, the iterative algorithm may fail to get a solution in some cases. Given the difficulties of control synthesis based on Lyapunov stability theorem, it is most striking to find that the new convergence criterion presented in (Rantzer, 2001) based on the so-called density function ρ has much better convexity properties. Then, Prajna *et al.*(2004) exploited this criterion to solve the control synthesis problems, and Ataei-Esfahani and Wang (2007) applied it to the control design of a hypersonic aircraft. Unfortunately, the convergence criterion via the density function does not involve any information about the convergence rate, so the controller designed by this scheme may perform slow convergence in some instances. Moreover, the criterion by Rantzer (2001)

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 60674028 and 60736021), the Hi-Tech Research and Development Program (863) of China (Nos. 2006AA04Z184 and 2007AA041406), the Key Technologies R&D Program of Zhejiang Province, China (No. 2006C11066), and the Joint Funds of NSFC-Guangdong Province of China (No. U0735003)

is "almost" condition, which means that there may exist unstable zero measure sets. Since lines in 2D plane and surfaces in 3D or higher dimensional space are zero measure sets, there may exist potentially unstable sets in practical applications of controlled systems designed using the density function, which are not suitable for systems with high safety requirements, e.g., flight control systems.

In this paper, we propose algorithms for polynomial decomposition which can effectively check positive polynomials. Then, a new method for control synthesis of polynomial nonlinear systems is proposed. The main idea of the method is that, by searching a Lyapunov function $V(\boldsymbol{x})$ and inspecting the derivative of $V(\boldsymbol{x})$ along the system trajectory, a nonlinear controller is set up to cancel the positive part of $\dot{V}(\boldsymbol{x})$. Furthermore, negativity of $\dot{V}(\boldsymbol{x})$ is checked when $u(\boldsymbol{x})=0$ for $\boldsymbol{x}\in\mathbb{R}_0^n$ so as to guarantee the global stability of the closed loop system. Our work solved by PENBMI (http://www.penopt.com/) obtains better results than the iterative algorithm does and advances the performance in the sense of reliability requirements compared to the density function scheme.

The notations throughout this paper are:

$\Re_n$: the set of all polynomials in n variables; $\Re_n^r$: the set of all polynomial vectors in n variables with r dimensions; $\Re_{n,d}^r$: the set of all polynomial vectors in n variables with r dimensions and d the maximum degree of the elements in the vector; $\mathbb{N}$: the nonnegative integer set; $\mathbb{R}_0^n$: the n-dimensional real vector set except the zero point $\mathbf{0}$; Σ_n: the set of SOS polynomials in n variables, defined as

$$\Sigma_n:=\left\{p\in\Re_n \mid p=\sum_{i=1}^{k} f_i^2,\ f_i\in\Re_n,\ i=1,2,\ldots,k\right\}.$$

Obviously, if $p\in\Sigma_n$, then $p(\boldsymbol{x})\geq 0$, $\forall\boldsymbol{x}\in\mathbb{R}^n$.

$$c(n,r)=\binom{r}{n}=\begin{cases} n!/[r!(n-r)!], & r>0,\\ 1, & r=0,\\ 0, & r<0.\end{cases}$$

POLYNOMIAL DECOMPOSITION AND POSITIVE POLYNOMIAL VALIDATION

In our work, positive polynomial validation is solved by polynomial decomposition, which transmits the coefficients of a polynomial to matrix inequalities. We set up the decomposition framework by giving several novel definitions which will facilitate the expression of our decomposition algorithms.

Definition 1 A monomial $m(\boldsymbol{x})$ in n variables is a function defined as $m(\boldsymbol{x})=\prod_{i=1}^{n} x_i^{r_i}$ for $r_i\in\mathbb{N}$, and the degree of a monomial is defined as $\deg(m):=\sum_{i=1}^{n} r_i=r$.

Definition 2 A polynomial $f(\boldsymbol{x})\in\Re_n$ is a finite linear combination of monomials,

$$f(\boldsymbol{x})=\sum_r c_r m_r(\boldsymbol{x}),\quad c_r\in\mathbb{R}.$$

The degree of $f(\boldsymbol{x})$ is denoted by

$$\deg(f):=\max_r \deg(m_r).$$

Definition 3 $\boldsymbol{x}^{\{r\}}$ is called the homogeneous monomial base of degree r for $\boldsymbol{x}=[x_1\ x_2\ \cdots\ x_n]^{\mathrm{T}}\in\mathbb{R}^n$. $\boldsymbol{x}^{\{r\}}$ can be generated as

$$\boldsymbol{x}^{\{r\}}=\begin{bmatrix} x_1^r[E_2(\boldsymbol{x})]^{\{0\}}\\ x_1^{r-1}[E_2(\boldsymbol{x})]^{\{1\}}\\ x_1^{r-2}[E_2(\boldsymbol{x})]^{\{2\}}\\ \vdots\\ x_1[E_2(\boldsymbol{x})]^{\{r-1\}}\\ x_1^0[E_2(\boldsymbol{x})]^{\{r\}}\end{bmatrix},\ \text{with}\ \boldsymbol{x}^{\{0\}}=1,\ E_2(\boldsymbol{x})=\begin{bmatrix} x_2\\ x_3\\ \vdots\\ x_n\end{bmatrix},$$

$E_2(\boldsymbol{x})$ is a shift function which deletes the first element of $\boldsymbol{x}$. The dimension of $\boldsymbol{x}^{\{r\}}$ is calculated by

$$\dim(\boldsymbol{x}^{\{r\}})=c(n+r-1,r)=\frac{(n+r-1)!}{r!(n-1)!}.\qquad(1)$$

Definition 4 $\boldsymbol{x}^{|r|}$ is called the full monomial base of degree r for $\boldsymbol{x}\in\mathbb{R}^n$. $\boldsymbol{x}^{|r|}$ is generated from the homogeneous monomial bases as

$$\boldsymbol{x}^{|r|}=\left[(\boldsymbol{x}^{\{0\}})^{\mathrm{T}}\quad(\boldsymbol{x}^{\{1\}})^{\mathrm{T}}\quad\cdots\quad(\boldsymbol{x}^{\{r\}})^{\mathrm{T}}\right]^{\mathrm{T}}.$$

The dimension of $\boldsymbol{x}^{|r|}$ is calculated by

$$\dim(\boldsymbol{x}^{|r|})=c(n+r,r)=\frac{(n+r)!}{r!n!}. \tag{2}$$

Definition 5 $R(m(\boldsymbol{x}))\in\mathbb{N}^{1\times n}$ is called the exponent mapping of monomial $m(\boldsymbol{x})$ for $\boldsymbol{x}\in\mathbb{R}^n$. If $m(\boldsymbol{x})=\prod_{i=1}^{n}x_i^{r_i}$, then $R(m)=[r_1\ r_2\ \dots\ r_n]$. We denote $R(m(\boldsymbol{x}))$ by $\boldsymbol{R}$ throughout this paper unless otherwise stated.

Definition 6 $H(\boldsymbol{R},d):\mathbb{N}^{1\times n}\times\mathbb{N}\rightarrow\mathbb{N}^{k\times n}$ is called the d-degree homogeneous decomposition array of monomial $m(\boldsymbol{x})$ for $\boldsymbol{x}\in\mathbb{R}^n$ and $\boldsymbol{R}=R(m(\boldsymbol{x}))$, if $\mathrm{sum}(H_i(\boldsymbol{R},d))=d$ and every element in $H_i(\boldsymbol{R},d)$ is not greater than the corresponding element in $\boldsymbol{R}$. Where $H_i(\boldsymbol{R},d)$ is the ith row of $H(\boldsymbol{R},d)$ and $H_i(\boldsymbol{R},d)\neq H_j(\boldsymbol{R},d)$ for $i\neq j$, and k is the maximum number of such $H_i(\boldsymbol{R},d)$'s.

Definition 7 $D(\boldsymbol{R},r):\mathbb{N}^{1\times n}\times\mathbb{N}\rightarrow\mathbb{N}^{m\times n}$ is called the r-degree full base decomposition array of monomial $m(\boldsymbol{x})$ with respect to $\boldsymbol{R}$ in the polynomial $f(\boldsymbol{x})\in\Re_{n,2r}$, if $\bar{D}_i(\boldsymbol{R},r)\triangleq\boldsymbol{R}-D_i(\boldsymbol{R},r)\in\mathbb{N}^{1\times n}$ is not empty and the first nonzero element of $D_i(\boldsymbol{R},r)-\bar{D}_i(\boldsymbol{R},r)$ is nonnegative. Where $D_i(\boldsymbol{R},r)$ is the ith row of $D(\boldsymbol{R},r)$ and $D_i(\boldsymbol{R},r)\neq D_j(\boldsymbol{R},r)$ for $i\neq j$, and m is the maximum number of such $D_i(\boldsymbol{R},r)$'s.

Definition 8 $L_{\mathrm{o}}(\boldsymbol{R}):\mathbb{N}^{1\times n}\rightarrow\mathbb{N}$ is the index mapping of monomial $m(\boldsymbol{x})$ with respect to $\boldsymbol{R}$ in the full monomial base. $L_{\mathrm{o}}(\boldsymbol{R})$ is calculated by

$$\begin{aligned}L_{\mathrm{o}}(\boldsymbol{R})=&\sum_{i=1}^{n-1}c(n-i+s-s_i-1,\ s-s_i-1)\\&+c(n+s-1,\ s-1)+1,\end{aligned} \tag{3}$$

where $s=\sum_{i=1}^{n}r_i$, $s_i=\sum_{j=1}^{i}r_j$. So $L_{\mathrm{o}}(\boldsymbol{R})$ calculates the index of $m(\boldsymbol{x})$ in the full monomial base $\boldsymbol{x}^{|r|}$.

Positive polynomial validation

Any polynomial $f(\boldsymbol{x})\in\Re_{n,d}$ can be written in linear form with full base $\boldsymbol{x}^{|d|}$ as

$$f(\boldsymbol{x})=\boldsymbol{C}\boldsymbol{x}^{|d|},\quad \boldsymbol{C}\in\mathbb{R}^{1\times c(n+d,d)}. \tag{4}$$

If $f(\boldsymbol{x})\in\Re_{n,d}$ is in even degree, i.e., $d=2r$, then it can be rewritten in quadratic form as

$$f(\boldsymbol{x})=(\boldsymbol{x}^{|r|})^{\mathrm{T}}\boldsymbol{P}_f\boldsymbol{x}^{|r|}, \tag{5}$$

where $\boldsymbol{P}_f\in\mathbb{R}^{c(n+r,r)\times c(n+r,r)}$ is a symmetric matrix. However, $\boldsymbol{P}_f$ is not unique since many monomials can be constructed by multiplication of different monomials. For example, $x_1^2x_2^2$ can be constructed by four types of multiplication in $\boldsymbol{x}^{|3|}$ as

$$x_1^2x_2^2=x_1^2x_2\cdot x_2=x_1x_2^2\cdot x_1=x_1^2\cdot x_2^2=x_1x_2\cdot x_1x_2.$$

Thus, $c_1x_1^2x_2^2$ can be written as

$$\begin{aligned}c_1x_1^2x_2^2=&(c_1-\alpha_1-\alpha_2-\alpha_3)\cdot x_1^2x_2\cdot x_2+\alpha_1\cdot x_1x_2^2\cdot x_1\\&+\alpha_2\cdot x_1^2\cdot x_2^2+\alpha_3x_1x_2\cdot x_1x_2,\end{aligned}$$

where $\boldsymbol{\alpha}=[\alpha_1\ \alpha_2\ \alpha_3]^{\mathrm{T}}\in\mathbb{R}^3$ is a free variable vector which has zero effect on $c_1x_1^2x_2^2$. For $f(\boldsymbol{x})\in\Re_{n,2r}$, there are $N(\boldsymbol{\alpha})$ free variables which have zero effects on $f(\boldsymbol{x})$, and $N(\boldsymbol{\alpha})$ is calculated as (Chesi, 2003)

$$\begin{aligned}N(\boldsymbol{\alpha})=&c(n+r,r)[c(n+r,r)+1]/2\\&-c(n+2r,2r).\end{aligned} \tag{6}$$

Furthermore, Eq.(5) can be written as

$$f(\boldsymbol{x})=(\boldsymbol{x}^{|r|})^{\mathrm{T}}(\boldsymbol{P}_f+\boldsymbol{L}(\boldsymbol{\alpha}))\boldsymbol{x}^{|r|}, \tag{7}$$

where $\boldsymbol{L}(\boldsymbol{\alpha})\in\mathbb{R}^{c(n+r,r)\times c(n+r,r)}$ is a symmetric matrix with $N(\boldsymbol{\alpha})$ free variables and satisfies

$$(\boldsymbol{x}^{|r|})^{\mathrm{T}}\boldsymbol{L}(\boldsymbol{\alpha})\boldsymbol{x}^{|r|}=0,\quad\forall\boldsymbol{\alpha}\in\mathbb{R}^{N(\boldsymbol{\alpha})}.$$

$\boldsymbol{L}(\boldsymbol{\alpha})$ covers all the possible matrices which have zero effects on $f(\boldsymbol{x})$. We define $\boldsymbol{M}(f)$ as

$$\boldsymbol{M}(f)=\boldsymbol{P}_f+\boldsymbol{L}(\boldsymbol{\alpha}). \tag{8}$$

Lemma 1 (Parrilo, 2000) Given $f(\boldsymbol{x})\in\Re_{n,2r}$, the sufficient condition for $f(\boldsymbol{x})\geq 0\ \forall\boldsymbol{x}\in\mathbb{R}^n$ is that, $f(\boldsymbol{x})$

can be rewritten as an SOS form: $f(\boldsymbol{x})=\sum_{i=1}^{k} f_i^2(\boldsymbol{x})$. The necessary condition also holds in the following three cases: (1) n=2; (2) r=1; (3) n=3, r=2.

Theorem 1 Given $f(\boldsymbol{x})\in\Re_{n,2r}$, the following statements are equivalent:

(1) $f(\boldsymbol{x})$ can be rewritten as an SOS form: $f(\boldsymbol{x})=\sum_{i=1}^{k} f_i^2(\boldsymbol{x})$.

(2) There exists $\boldsymbol{\alpha}\in\mathbb{R}^{N(\boldsymbol{\alpha})}$ such that $\boldsymbol{M}(f)=\boldsymbol{P}_f+\boldsymbol{L}(\boldsymbol{\alpha})\geq\boldsymbol{0}$.

Proof (of Theorem 1) Statements (1) to (2). Since $\deg(f(\boldsymbol{x}))=2r$, $\deg(f_i(\boldsymbol{x}))\leq r$ holds for any $f_i(\boldsymbol{x})$, and $f_i(\boldsymbol{x})$ can be written in the linear form of $\boldsymbol{x}^{|r|}$, i.e., $f_i(\boldsymbol{x})=\boldsymbol{C}_i\boldsymbol{x}^{|r|}$, $\boldsymbol{C}_i\in\mathbb{R}^{1\times c(n+r,r)}$. Thus, one has

$$f(\boldsymbol{x})=(\boldsymbol{x}^{|r|})^{\mathrm{T}}\left(\sum_{i=1}^{k}\boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{C}_i\right)\boldsymbol{x}^{|r|}.$$

Let $\boldsymbol{P}_f^{(1)}=\sum_{i=1}^{k}\boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{C}_i$ and $\boldsymbol{P}_f^{(2)}$ be one of any symmetric matrices satisfying Eq.(5). Since $\boldsymbol{L}(\boldsymbol{\alpha})$ covers all the possible matrices having zero effects on $f(\boldsymbol{x})$, we can find some $\boldsymbol{\alpha}\in\mathbb{R}^{N(\boldsymbol{\alpha})}$ satisfying $\boldsymbol{L}(\boldsymbol{\alpha})=\boldsymbol{P}_f^{(1)}-\boldsymbol{P}_f^{(2)}$. Let $\boldsymbol{M}(f)=\boldsymbol{P}_f^{(2)}+\boldsymbol{L}(\boldsymbol{\alpha})$, then Statement (2) is derived.

Statements (2) to (1). If (2) holds, from singular value decomposition, one has $\boldsymbol{M}(f)=\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{U}$. Choose $f_i(\boldsymbol{x})=(\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}\boldsymbol{x}^{|r|})_i$, then Statement (1) is obtained.

In our work, positive polynomial validation is checked by $\boldsymbol{M}(f)$, which can further be solved by semidefinite programming. We choose SeDuMi (http://sedumi.mcmaster.ca/) for LMI (linear matrix inequalities) problems and PENBMI (http://www.penopt.com/) for BMI (bilinear matrix inequalities) problems.

Algorithms for polynomial decomposition

Generating $\boldsymbol{M}(f)$ is called polynomial decomposition in our work, and it is accomplished by three algorithms: generating $H(\boldsymbol{R},d)$, generating $D(\boldsymbol{R},r)$ and finally generating $\boldsymbol{M}(f)$.

1. Algorithm for generating $H(\boldsymbol{R},d)$

Let $\boldsymbol{R}=[r_1\ r_2\ \dots\ r_n]$, $\mathrm{sum}(\boldsymbol{R})=\sum_{i=1}^{n} r_i$, $E_2(\boldsymbol{R})=[r_2\ r_3\ \dots\ r_n]$ which removes the first element of $\boldsymbol{R}$, $E_1(\boldsymbol{R})=r_1$ which gets the first element of $\boldsymbol{R}$, and $t=\min(E_1(\boldsymbol{R}),d)$, then $H(\boldsymbol{R},d)$ is realized via a recursive function as follows:

(1) If the length of $\boldsymbol{R}$ is 1, and $\mathrm{sum}(\boldsymbol{R})\geq d$, then $H(\boldsymbol{R},d)=d$.

(2) If the length of $\boldsymbol{R}$ is greater than 1, and $\mathrm{sum}(\boldsymbol{R})\geq d$, then

$$H(\boldsymbol{R},d)=\begin{bmatrix} t\cdot\boldsymbol{1}_{m_0\times1} & H(E_2(\boldsymbol{R}),d-t) \\ (t-1)\cdot\boldsymbol{1}_{m_1\times1} & H(E_2(\boldsymbol{R}),d-t+1) \\ \vdots & \vdots \\ \boldsymbol{1}_{m_{t-1}\times1} & H(E_2(\boldsymbol{R}),d-1) \\ 0\cdot\boldsymbol{1}_{m_t\times1} & H(E_2(\boldsymbol{R}),d) \end{bmatrix},$$

where $t=\min(E_1(\boldsymbol{R}),d)$, m_i is the row size of $H(E_2(\boldsymbol{R}),d-t+i)$ for $i=0,1,\cdots,t$. Without loss of generality, $(t-i)\cdot\boldsymbol{1}_{m_i\times1}=\varnothing$ when $H(E_2(\boldsymbol{R}),d-t+i)=\varnothing$.

(3) If $\mathrm{sum}(\boldsymbol{R})<d$, then $H(\boldsymbol{R},d)=\varnothing$.

2. Algorithm for generating $D(\boldsymbol{R},r)$

Let $t=\min(\mathrm{sum}(\boldsymbol{R}),r)$, and $\tilde{H}(\boldsymbol{R},k)$ is the top l rows of $H(\boldsymbol{R},k)$, for the row size of $H(\boldsymbol{R},k)$ being $2l$ or $2l-1$.

(1) If $\mathrm{sum}(\boldsymbol{R})=2k-1$, then

$$D(\boldsymbol{R},r)=\begin{bmatrix} H(\boldsymbol{R},t) \\ H(\boldsymbol{R},t-1) \\ \vdots \\ H(\boldsymbol{R},k+1) \\ H(\boldsymbol{R},k) \end{bmatrix}.$$

(2) If $\mathrm{sum}(\boldsymbol{R})=2k$ and $t>k$, then

$$D(\boldsymbol{R},r)=\begin{bmatrix} H(\boldsymbol{R},t) \\ H(\boldsymbol{R},t-1) \\ \vdots \\ H(\boldsymbol{R},k+1) \\ \tilde{H}(\boldsymbol{R},k) \end{bmatrix}.$$

(3) If $\mathrm{sum}(\boldsymbol{R})=2k$ and $t=k$, then $D(\boldsymbol{R},r)=\tilde{H}(\boldsymbol{R},k)$.

3. Algorithm for generating $\boldsymbol{M}(f)$

Consider the $f(\boldsymbol{x})$ depicted in Eq.(4), and denote by $(\boldsymbol{x}^{|r|})_i$ the ith element in the full monomial base $\boldsymbol{x}^{|r|}$, C_i the ith element of $\boldsymbol{C}$, M_{jk} the element in the jth row and the kth column of $\boldsymbol{M}(f)$. Set i=1 and q=1 for initialization. Denote by v_{jk} an indictor function for

$$v_{jk}=\begin{cases}1, & j\neq k,\\ 2, & j=k.\end{cases}$$

Step 1: Let $\tilde{\boldsymbol{R}}=R((\boldsymbol{x}^{|r|})_i)$ and $\tilde{\boldsymbol{D}}=D(\tilde{\boldsymbol{R}},r)$. Determine the row size of $\tilde{\boldsymbol{D}}$ and denote it by l. If l=1, then $j=L_o(\tilde{\boldsymbol{D}})$, $k=L_o(\tilde{\boldsymbol{J}}-\tilde{\boldsymbol{D}})$, $M_{jk}=M_{kj}=v_{jk}C_i/2$, and go to Step 4; if l>1, set m=1, t=0.

Step 2: Let $j=L_o(\tilde{\boldsymbol{D}}_m)$, $k=L_o(\tilde{\boldsymbol{J}}-\tilde{\boldsymbol{D}}_m)$, $M_{jk}=M_{kj}=v_{jk}\alpha_q$, $t=t+\alpha_q$ and $q=q+1$.

Step 3: Set $m=m+1$. If $m<l$, go to Step 2; else, let $j=L_o(\tilde{\boldsymbol{D}}_l)$, $k=L_o(\tilde{\boldsymbol{J}}-\tilde{\boldsymbol{D}}_l)$, $M_{jk}=M_{kj}=v_{jk}(C_i/2-t)$.

Step 4: Set $i=i+1$. If $i>c(n+r,r)$, $\boldsymbol{M}(f)$ is accomplished; else go to Step 1.

CONTROL SYNTHESIS

Lemma 2 (Parrilo, 2000) Let $(f_j)_{j=1,2,\ldots,s}$, $(g_k)_{k=1,2,\ldots,t}$, $(h_l)_{l=1,2,\ldots,u}$ be finite families of polynomials in $\Re_n$. Denote by F the cone generated by $(f_j)_{j=1,2,\ldots,s}$, G the multiplicative monoid generated by $(g_k)_{k=1,2,\ldots,t}$, and H the ideal generated by $(h_l)_{l=1,2,\ldots,u}$. Then, the following properties are equivalent:

(1) The set $\left\{\boldsymbol{x}\in\mathbb{R}^n \,\middle|\, \begin{array}{ll} f_j(\boldsymbol{x})\geq 0, & j=1,2,\cdots,s\\ g_k(\boldsymbol{x})\neq 0, & k=1,2,\cdots,t\\ h_l(\boldsymbol{x})=0, & l=1,2,\cdots,u\end{array}\right\}$ is empty.

(2) There exist $f\in F$, $g\in G$, $h\in H$ such that $f+g^2+h=0$ $\forall \boldsymbol{x}\in\mathbb{R}^n$.

First, we consider the single input control system depicted as

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{g}(\boldsymbol{x})u(\boldsymbol{x}), \tag{9}$$

where $\boldsymbol{f}(\boldsymbol{x})\in\Re_n^n$, $\boldsymbol{g}(\boldsymbol{x})\in\Re_n^{n\times 1}$ and $u(\boldsymbol{x})\in\Re_n$. Assume that the system Eq.(9) satisfies $\boldsymbol{f}(\boldsymbol{0})=\boldsymbol{0}$, $u(\boldsymbol{0})=0$. $u(\boldsymbol{x})$ is designed to stabilize the system Eq.(9) to the original point $\boldsymbol{x}=\boldsymbol{0}$ for any initial state. The control law for the system Eq.(9) is given in Theorem 2.

Theorem 2 If there exists a Lyapunov function $V(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}$, positive polynomials $s_0(\boldsymbol{x})$ and $s_1(\boldsymbol{x})$, and a polynomial $p(\boldsymbol{x})$ with suitable order satisfying the following conditions:

$$\boldsymbol{P}\geq\varepsilon\boldsymbol{I},\ \ \varepsilon\in\mathbb{R}^+, \tag{10}$$

$$\Xi_1=-s_1\alpha(\boldsymbol{x})-p(\boldsymbol{x})\beta(\boldsymbol{x})-l^2(\boldsymbol{x})\in\Sigma_n, \tag{11}$$

$$\lim_{\boldsymbol{x}\to\boldsymbol{0}}\frac{\alpha(\boldsymbol{x})}{\beta(\boldsymbol{x})}=0,\quad \lim_{\boldsymbol{x}\to\boldsymbol{0}}\frac{s_0(\boldsymbol{x})}{\beta(\boldsymbol{x})}=0, \tag{12}$$

where $\alpha(\boldsymbol{x})=\nabla V\cdot\boldsymbol{f}(\boldsymbol{x})$, $\beta(\boldsymbol{x})=\nabla V\cdot\boldsymbol{g}(\boldsymbol{x})$, and $l(\boldsymbol{x})$ satisfies $l(\boldsymbol{x})\neq 0$ $\forall\boldsymbol{x}\in\mathbb{R}_0^n$, then the controller

$$u(\boldsymbol{x})=\begin{cases}-\left(\alpha(\boldsymbol{x})+s_0(\boldsymbol{x})\right)/\beta(\boldsymbol{x}), & \beta(\boldsymbol{x})\neq 0,\\ 0, \ \beta(\boldsymbol{x})=0,\end{cases} \tag{13}$$

stabilizes the system Eq.(9).

Proof From Eq.(10), we have

$$\varepsilon\|\boldsymbol{x}\|_2^2\leq V(\boldsymbol{x})\leq\lambda_{\max}(\boldsymbol{P})\|\boldsymbol{x}\|_2^2,$$

hence, $V(\boldsymbol{x})$ satisfies the positive definition of the Lyapunov function. The derivative of $V(\boldsymbol{x})$ along the system Eq.(9) is

$$\dot{V}(\boldsymbol{x})=\nabla V\cdot\boldsymbol{f}+\nabla V\cdot\boldsymbol{g}u=\alpha(\boldsymbol{x})+\beta(\boldsymbol{x})u.$$

Substituting Eq.(13) to $\dot{V}$ gives

$$\dot{V}(\boldsymbol{x})=\begin{cases}-s_0(\boldsymbol{x}), & \beta(\boldsymbol{x})\neq 0,\\ \alpha(\boldsymbol{x}), & \beta(\boldsymbol{x})=0.\end{cases} \tag{14}$$

From Eq.(14), one can see that if

$$\alpha(\boldsymbol{x})<0,\ \forall\boldsymbol{x}\in\Omega(\boldsymbol{x}):=\{\boldsymbol{x}\in\mathbb{R}^n\mid\beta(\boldsymbol{x})=0,\ \boldsymbol{x}\neq\boldsymbol{0}\} \tag{15}$$

is true, $\dot{V}(\boldsymbol{x})<0$ holds $\forall\boldsymbol{x}\in\mathbb{R}_0^n$. Therefore, if the condition Eq.(15) holds, the control law given in Eq.(13) stabilizes the system Eq.(9) globally. Condition Eq.(15) is equivalent to

$$\{\boldsymbol{x}\in\mathbb{R}^n\mid\alpha(\boldsymbol{x})\geq 0,\ l(\boldsymbol{x})\neq 0,\ \beta(\boldsymbol{x})=0\}=\varnothing. \tag{16}$$

By Lemma 2, Condition Eq.(16) can be further presented as: there exist $s_1,s_2\in\Sigma_n$ and $p(\boldsymbol{x})\in\Re_n$ satisfying

$$s_1(\boldsymbol{x})\alpha(\boldsymbol{x})+s_2(\boldsymbol{x})+p(\boldsymbol{x})\beta(\boldsymbol{x})+l^2(\boldsymbol{x})=0. \tag{17}$$

Obviously, Condition Eq.(17) is equivalent to the condition Eq.(11).

Condition Eq.(11) is checked by $\boldsymbol{M}(\Xi_1)\geq\boldsymbol{0}$, and $\boldsymbol{M}(\Xi_1)$ is decomposed by our algorithm and solved by

PENBMI. Condition Eq.(12) guarantees that $u(\boldsymbol{x})$ is continuous when $\boldsymbol{x}\to\boldsymbol{0}$.

For multi-input control systems, we can obtain similar results. Consider a multi-input system in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{x})u_i(\boldsymbol{x}), \tag{18}$$

where $\boldsymbol{f}(\boldsymbol{x}) \in \Re_n^n$, $\boldsymbol{f}(\boldsymbol{0})=\boldsymbol{0}$, $\boldsymbol{g}_i(\boldsymbol{x}) \in \Re_n^{n\times 1}$, $u_i(\boldsymbol{x}) \in \Re_n$, $u_i(\boldsymbol{0})=0$, for $i=1,2,\ldots,m$. The control law is given in Theorem 3, which is obtained by a similar method as that in Theorem 2.

Theorem 3 If there exists a Lyapunov function $V(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}$, positive polynomials $s_0(\boldsymbol{x})$ and $s_1(\boldsymbol{x})$, and polynomials $p_{1,2,\ldots,m}(\boldsymbol{x}) \in \Re_n$ satisfying the following conditions:

$$\boldsymbol{P} \ge \varepsilon \boldsymbol{I},\quad \varepsilon \in \mathbb{R}^+, \tag{19}$$

$$\varXi_2 = -s_1\alpha(\boldsymbol{x}) - l^2(\boldsymbol{x}) - \sum_{i=1}^{m} p_i(\boldsymbol{x})b_i(\boldsymbol{x}) \in \Sigma_n, \tag{20}$$

$$\lim_{\boldsymbol{x}\to\boldsymbol{0}} \frac{b_i(\boldsymbol{x})\alpha(\boldsymbol{x})}{\beta(\boldsymbol{x})} = 0,\quad \lim_{\boldsymbol{x}\to\boldsymbol{0}} \frac{b_i(\boldsymbol{x})s_0(\boldsymbol{x})}{\beta(\boldsymbol{x})} = 0,\quad i=1,2,\cdots,m, \tag{21}$$

where $b_i(\boldsymbol{x}) = \nabla V \cdot \boldsymbol{g}_i(\boldsymbol{x})$, $\beta(\boldsymbol{x}) = \sum_{i=1}^{m} b_i^2(\boldsymbol{x})$, $\alpha(\boldsymbol{x})$ and $l(\boldsymbol{x})$ are the same as those defined in Theorem 1. Then the controller $\boldsymbol{u}(\boldsymbol{x}) = [u_1(\boldsymbol{x})\ \ u_2(\boldsymbol{x})\ \ \cdots\ \ u_m(\boldsymbol{x})]^{\mathrm{T}}$ with

$$u_i(\boldsymbol{x}) = \begin{cases} -b_i \dfrac{\alpha(\boldsymbol{x}) + s_0(\boldsymbol{x})}{\beta(\boldsymbol{x})}, & \beta(\boldsymbol{x}) \ne 0, \\ 0, & \beta(\boldsymbol{x}) = 0, \end{cases} \quad i=1,2,\ldots,m \tag{22}$$

stabilizes the system Eq.(18).

APPLICATION IN ATTITUDE CONTROL

The complete attitude dynamics of a rigid body obeys the differential equation (Tsiotras, 1998; Dimarogonas *et al.*, 2006):

$$\begin{cases} \dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{J}^{-1}\boldsymbol{u}, \\ \dot{\boldsymbol{\sigma}} = \boldsymbol{G}(\boldsymbol{\sigma})\boldsymbol{\omega}, \end{cases} \tag{23}$$

where $\boldsymbol{\omega}\in\mathbb{R}^3$ denotes the angular velocity vector in a body-fixed frame, $\boldsymbol{u}\in\mathbb{R}^3$ is the acting torque vector, $\boldsymbol{J}\in\mathbb{R}^{3\times 3}$ is the symmetric inertia matrix, and $\boldsymbol{\sigma}\in\mathbb{R}^3$ is the modified Rodrigues parameter vector. Comparing to (Prajna *et al.*, 2004), the kinematic description using the modified Rodrigues parameter remains valid for eigenaxis rotations up to 360°, whereas the description in (Prajna *et al.*, 2004) is not suitable when the eigenaxis rotations exceed 180°. The matrix $\boldsymbol{S}(\cdot)$ denotes a skew-symmetric matrix representing the cross product between two vectors, i.e., $\boldsymbol{S}(\boldsymbol{v})\boldsymbol{\omega}=-\boldsymbol{v}\times\boldsymbol{\omega}$. $\boldsymbol{S}(\boldsymbol{\omega})$ and $\boldsymbol{G}(\boldsymbol{\sigma})$ in matrix forms are given by

$$\boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix},$$

$$\boldsymbol{G}(\boldsymbol{\sigma}) = \frac{1}{2}\left(\frac{1-\boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\sigma}}{2}\boldsymbol{I}_{3\times 3} - \boldsymbol{S}(\boldsymbol{\sigma}) + \boldsymbol{\sigma}\boldsymbol{\sigma}^{\mathrm{T}} \right).$$

We apply Theorem 3 and polynomial decomposition to numerically construct a stabilizing controller for the system Eq.(23). Assume $\boldsymbol{J}=\mathrm{diag}(4, 2, 1)$ for a real system. In our construction, let $\boldsymbol{x} = [\boldsymbol{\omega}^{\mathrm{T}}\ \boldsymbol{\sigma}^{\mathrm{T}}]^{\mathrm{T}}$, choose $\varepsilon=0.1$, $s_1=1$, $p_{1,2,3}(\boldsymbol{x}) \in \Re_{6,3}$ and

$$l^2(\boldsymbol{x}) = \varepsilon_1 \sum_{i=1}^{6} x_i^2 + \varepsilon_2 \sum_{i=1}^{6} x_i^4$$

with $\varepsilon_1, \varepsilon_2 \ge 0$ and $\varepsilon_1+\varepsilon_2 \ge 10^{-4}$. By solving Eqs.(19) and (20) with PENBMI, we obtain

$$\boldsymbol{P} = \begin{bmatrix} 10.99 & 0 & 0 & 8.29 & 0 & 0 \\ * & 8.27 & 0 & 0 & 6.34 & 0 \\ * & * & 6.92 & 0 & 0 & 5.40 \\ * & * & * & 98.17 & 0 & 0 \\ * & * & * & * & 96.95 & 0 \\ * & * & * & * & * & 96.34 \end{bmatrix}.$$

Choose $s_0(\boldsymbol{x})=\beta(\boldsymbol{x})$, then the controller $\boldsymbol{u}(\boldsymbol{x})$ is given definitely in Eq.(22) when $\boldsymbol{P}$ is solved. The proposed controller derives good performance for attitude control as shown in Figs.1a~1c.

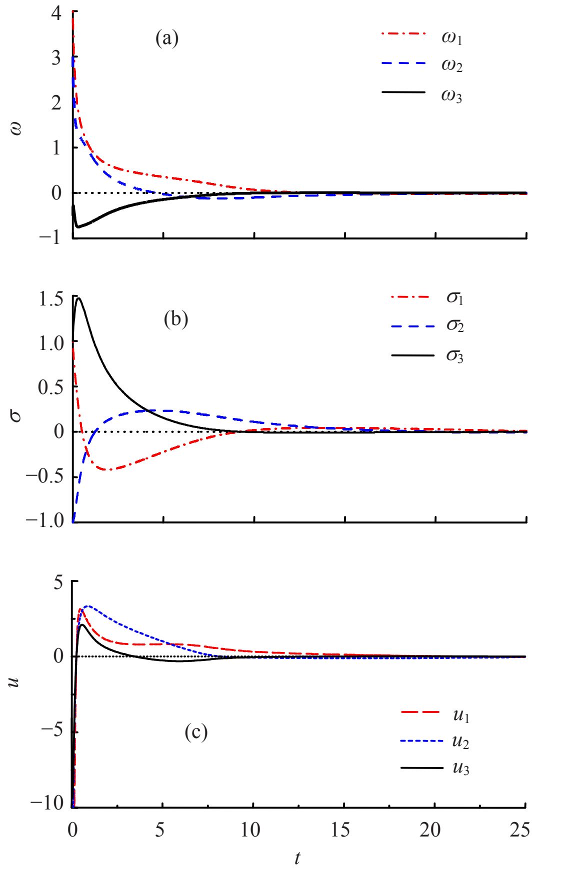

Fig.1 Evolution of ω (a), evolution of σ (b) and control action u (c) for the closed loop system Eq.(23)

CONCLUSION

The proposed polynomial decomposition algorithm is a general-purpose framework for positive polynomial validation. Any polynomial which can be decomposed into sum of squares form can be checked by matrix inequalities after polynomial decomposition. A control law for polynomial nonlinear systems is addressed based on positive polynomial validation. The numerical example shows that the proposed control law achieves effective performance for attitude stabilization control.

References

Ataei-Esfahani, A., Wang, Q., 2007. Nonlinear Control Design of a Hypersonic Aircraft Using Sum-of-Squares Method. Proc. American Control Conf. New York, USA, p.5278-5283. [doi:10.1109/ACC.2007.4282934]

Chesi, G., 2003. Estimating the Domain of Attraction: A Light LMI Technique for a Class of Polynomial Systems. IEEE Conf. on Decision and Control. Maui, Hawaii, USA, p.5609-5614. [doi:10.1109/CDC.2003.1271896]

Chesi, G., 2007. Estimating the domain of attraction via union of continuous families of Lyapunov estimates. *Syst. Control Lett.*, **56**(4):326-333. [doi:10.1016/j.sysconle.2006.10.012]

Chesi, G., Garulli, A., Tesi, A., Vicino, A., 2005. Polynomially parameter-dependent Lyapunov functions for robust stability of polytopic systems: an LMI approach. *IEEE Trans. on Automatic Control*, **50**(3):365-370. [doi:10.1109/TAC.2005.843848]

Dimarogonas, D.V., Tsiotras, P., Kyriakopoulos, K.J., 2006. Laplacian Cooperative Attitude Control of Multiple Rigid Bodies. Proc. IEEE Int. Symp. on Intelligent Control. Munich, Germany, p.3064-3069.

Fisher, J., Bhattacharya, R., 2007. Construction of Lyapunov Certificates for Partial Stability Problems Using Sum of Squares Techniques. Proc. American Control Conf., p.4823-4828. [doi:10.1109/ACC.2007.4282560]

Jarvis-Wloszek, Z., 2003. Lyapunov Based Analysis and Controller Synthesis for Polynomial Systems Using Sum-of-Squares Optimization. Ph.D Thesis, University of California, Berkeley.

Papachristodoulou, A., Prajna, S., 2005. A Tutorial on Sum of Squares Techniques for Systems Analysis. Proc. American Control Conf. Portland, OR, USA, p.2686-2700. [doi:10.1109/ACC.2005.1470374]

Parrilo, P.A., 2000. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D Thesis, California Institute of Technology, Pasadena, CA.

Prajna, S., Parrilo, P.A., Rantzer, A., 2004. Nonlinear control synthesis by convex optimization. *IEEE Trans. on Automatic Control*, **49**(2):310-314. [doi:10.1109/TAC.2003.823000]

Rantzer, A., 2001. A dual to Lyapunov's stability theorem. *Syst. Control Lett.*, **42**(3):161-168. [doi:10.1016/S0167-6911(00)00087-6]

Sturm, J.F., 1999. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, p.625-653.

Tsiotras, P., 1998. Further passivity results for the attitude control problem. *IEEE Trans. on Automatic Control*, **43**(11):1597-1600. [doi:10.1109/9.728877]