J Syst Sci Complex (2022) 35: 413–426

Solving Multivariate Polynomial Matrix Diophantine Equations with Gröbner Basis Method*

XIAO Fanghui · LU Dong · WANG Dingkang

DOI: 10.1007/s11424-021-0072-x
Received: 7 April 2020 / Revised: 30 July 2020

Abstract Different from previous viewpoints, multivariate polynomial matrix Diophantine equations are studied from the perspective of modules in this paper, that is, regarding the columns of matrices as elements in modules. A necessary and sufficient condition of the existence for the solution of equations is derived. Using powerful features and theoretical foundation of Gröbner bases for modules, the problem for determining and computing the solution of matrix Diophantine equations can be solved. Meanwhile, the authors make use of the extension on modules for the GVW algorithm that is a signature-based Gröbner basis algorithm as a powerful tool for the computation of Gröbner basis for module and the representation coefficients problem directly related to the particular solution of equations. As a consequence, a complete algorithm for solving multivariate polynomial matrix Diophantine equations by the Gröbner basis method is presented and has been implemented on the computer algebra system Maple.

Keywords Gröbner basis, matrix Diophantine equation, module, multivariate polynomial.

1 Introduction

With the growing presence of algebra in modern control theory, polynomial equations and matrices have become a useful tool for describing the dynamical behavior and the structure

XIAO Fanghui
College of Mathematics and Statistics, Hunan Normal University, Changsha 410006, China.
Email: xiaofanghui@amss.ac.cn.
LU Dong (Corresponding author)
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China; School of Mathematical Sciences, Beihang University, Beijing 100191, China.
Email: donglu@amss.ac.cn.
WANG Dingkang
KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China.
Email: dwang@mmrc.iss.ac.cn.
*This research was supported by the National Natural Science Foundation of China under Grant No. 12001030, the CAS Key Project QYZDJ-SSW-SYS022 and the National Key Research and Development Project 2020YFA0712300.
◇ *This paper was recommended for publication by Editor MOU Chenqi.*

of linear control systems[1–5]. And by the fractional representation of transfer functions[6], the mathematical synthesis of a control system having a desired property leads to the solution of a diophantine equation over an appropriate ring. Naturally, the computation of the solution for polynomial matrix Diophantine equations becomes an important problem in control theory with many applications that include multivariable stochastic optimal control[7], parametrization of stabilizing controllers[8, 9], robust stabilization[10, 11], disturbance rejecting[12, 13], pole placement[14], model matching[15, 16], H_2 optimal control[17], and so on.

In the past several decades, a lot of methods have been developed to solve the polynomial matrix Diophantine equation in one indeterminate (1-D) which is often employed to solving various control problems for standard systems. For example, Feinstein and Bar-Ness[18], and Lai[19] transfer the Diophantine equation to a set of linear algebraic equations and then solve those equations by applying a sequence of complicated operations on matrices. Chang, et al.[20] used polynomial matrix division method to find a solution of the equation. Wolovich[21] based on the state-space realization of the transfer matrix obtained the unique minimal degree solutions. Fang[22] and Yamada and Funahashi[23], based on the state-space concepts, solved the equation by constructing two constants matrices, which gave the whole class of the solutions and is more straightforward and simpler. Nevertheless, these methods exist some limitations, such as the strict properness of the transfer function matrix or other conditions that the equation needs to be required. Moreover, Karampetakis[24] utilized the generalized inverse of a polynomial matrix to investigate the solution of the matrix Diophantine equation and some new techniques, like the geometric method[25], are also established to solve this problem. Recently, Tzekis[26] has proposed a very interesting method to solve this class of equations.

Although the 1-D polynomial matrix Diophantine equation has been extensively studied and now well understood, many methods for 1-D cases cannot be naturally extended to multivariable cases for nonstandard systems. To the best of our knowledge, there are only a few literatures to study on multivariate (n-D) polynomial matrix equations. Among them, Šebek in [27] has presented an algorithm which is based on elementary reductions in a greater ring of polynomials in one indeterminate, having as coefficients polynomial fractions in the other $n-1$ indeterminates, and makes full use of the Euclidean division. Tzekis, et al.[28] gave an extension of the method as proposed in [26] for the computation of the general solution of n-D polynomial matrix equations with the presentation of a method to address the division of multivariate polynomials.

In this paper, we start from the perspective of the module to explore the general polynomial solution of the matrix equation, that is, regarding the columns of matrices as elements (column vectors) in modules. First, we transfer the polynomial matrix Diophantine equation in the form of $A_1X_1 + A_2X_2 = B$ to the general matrix equation $AX = B$. Considering the module M generated by the columns $\boldsymbol{a}_i$ of A, by the submodule membership, a necessary and sufficient condition of the existence for the solution of the equation is derived, that is, determining if the columns of B are in M. If each column $\boldsymbol{b}_j$ of B can be expressed as a linear combination of the columns of A, i.e., $\boldsymbol{b}_j = \sum_{i=1} x_{ij}\boldsymbol{a}_i$, then the representation coefficients x_{ij} make up a solution of the equation $AX = B$. By means of the powerful features and theoretical foundation of Gröbner bases for modules, not only can the existence and uniqueness for the solution of the equation be determined, but also the solution or general solution of the equation can be computed. Meanwhile, we extend the GVW algorithm proposed by Gao, et al. in [29] which is a signature-based Gröbner basis algorithm[30–32] to the module as a powerful tool for the computation of Gröbner basis for module and the representation coefficients problem directly related to the particular solution of the equation. As a consequence, a complete algorithm for solving multivariate polynomial matrix Diophantine equations by the Gröbner basis method is presented, which will output one of three results: No solution, unique solution and general solution when inputting given matrices A and B. What's more, we have implemented the proposed algorithm on Maple. This algorithm can be used to multivariate polynomial matrix factorizations[33–35].

This paper is organized as follows. In Section 2, some notations and concepts for modules and Gröbner bases are introduced. In Section 3, polynomial matrix Diophantine equations are considered from the idea of modules, and an attempt to solve the equation with the Gröbner basis for modules is presented. Section 4 gives a brief introduction of the extended GVW algorithm on modules and a complete algorithm for solving polynomial matrix equations. An example to illustrate the algorithm is given in Section 5. Finally, we conclude this paper.

2 Preliminaries

In this section, we will introduce some definitions and notations to prepare for the discussion of this article.

Let k be a field, $R = k[z]$ be the polynomial ring over k in the variables $z = \{z_1, z_2, \cdots, z_n\}$. A monomial over R is a product of the form $z^\alpha = z_1^{\alpha_1} z_2^{\alpha_2} \cdots z_n^{\alpha_n}$, where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ is any vector of nonnegative integers. Generally, we use the letters f, g for single polynomials (or elements of the ring R) and boldface letters $\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}$ for column vectors (that is, elements of R^m).

First, we briefly review the concept of modules, and the details can refer to [36]. As is well known, R^m is an R-module, which is closed under addition and scalar multiplication by elements of R, where m is a positive integer. In general, we will consider submodules of R^m to obtain other examples of R-modules. For example, given a finite set of vectors $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_s$ and consider the set of all column vectors which can be written as an R-linear combination of these vectors:

$$M = \{u_1\boldsymbol{f}_1 + u_2\boldsymbol{f}_2 + \cdots + u_s\boldsymbol{f}_s \in R^m : u_i \in R \text{ for } i = 1, 2, \cdots, s\}.$$

M is a submodule of R^m generated by $F = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_s\}$ and we denote it by $\langle \boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_s\rangle$.

In practice, we frequently consider a very important class of modules as follows.

Definition 2.1 Let $(\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_s)$ be an ordered s-tuple of elements $\boldsymbol{f}_i \in R^m$. The set of all $(u_1, u_2, \cdots, u_s)^{\mathrm{T}} \in R^s$ such that $u_1\boldsymbol{f}_1 + u_2\boldsymbol{f}_2 + \cdots + u_s\boldsymbol{f}_s = 0$ is an R-submodule of R^s, called the syzygy module of $(\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_s)$, and denoted by $\mathrm{Syz}(\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_s)$. Where $(u_1, u_2, \cdots,$

$u_s)^{\mathrm{T}}$ indicates the column vector which is the transpose of the row vector $(u_1, u_2, \cdots, u_s)$, or briefly, $(\cdot)^{\mathrm{T}}$ stands for the transpose.

Next, we introduce the theory of Gröbner bases for submodules in R^m, and readers can refer to Chapter 5 in [36].

Let $\succ$ be a monomial order on R and the unit vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_m$ be a standard basis of R^m. Then a monomial in R^m is an element of the form $z^\alpha \boldsymbol{e}_i$ for some i. There are two natural ways to obtain module orders $\succ_m$ on R^m by extending $\succ$ to R^m.

Definition 2.2 Let $\succ$ be any monomial order on R, and assume $\boldsymbol{e}_1 > \boldsymbol{e}_2 > \cdots > \boldsymbol{e}_m$.

1) (POT) We say $z^\alpha \boldsymbol{e}_i \succ_{POT} z^\beta \boldsymbol{e}_j$ if $i < j$, or $i = j$ and $z^\alpha \succ z^\beta$.

2) (TOP) We say $z^\alpha \boldsymbol{e}_i \succ_{TOP} z^\beta \boldsymbol{e}_j$ if $z^\alpha \succ z^\beta$, or if $z^\alpha = z^\beta$ and $i < j$.

Where POT and TOP stand for "position-over-term" and "term-over-position", respectively. For $\boldsymbol{g} \in R^m$, the leading term, leading coefficient, and leading monomial of $\boldsymbol{g}$ with respect to $\succ_m$ respectively are denoted by $\mathrm{LT}(\boldsymbol{g})$, $\mathrm{LC}(\boldsymbol{g})$, and $\mathrm{LM}(\boldsymbol{g})$.

The definition of Gröbner bases for submodules is as follows.

Definition 2.3 Given a monomial order $\succ_m$ on R^m, and let M be a submodule of R^m.

1) $\langle \mathrm{LT}(M) \rangle$ denotes the monomial submodule generated by the leading terms of all $\boldsymbol{g} \in M$ with respect to $\succ_m$.

2) A finite set $G = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_l\} \subset M$ is called a Gröbner basis for M if $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(\boldsymbol{g}_1), \mathrm{LT}(\boldsymbol{g}_2), \cdots, \mathrm{LT}(\boldsymbol{g}_l) \rangle$.

3 The General Solution of Polynomial Matrix Diophantine Equations

Consider the polynomial matrix Diophantine equation

$$A_1 X_1 + A_2 X_2 = B,$$

where $A_1 \in R^{m \times s_1}$, $A_2 \in R^{m \times s_2}$, $B \in R^{m \times p}$ are given polynomial matrices, while the matrices X_1, X_2 are unknown. This equation can be written as

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = B \quad \Longrightarrow \quad AX = B,$$

where $A \in R^{m \times s}$ $(s = s_1 + s_2)$.

Different from previous viewpoints, we start from the perspective of the module to explore the general solution of polynomial matrix equations.

Let $A = (a_{ij})_{m \times s}$, $B = (b_{uv})_{m \times p}$. Write $\boldsymbol{a}_j = (a_{1j}, a_{2j}, \cdots, a_{mj})^{\mathrm{T}}$ for $j = 1, 2, \cdots, s$ and $\boldsymbol{b}_v = (b_{1v}, b_{2v}, \cdots, b_{mv})^{\mathrm{T}}$ for $v = 1, 2, \cdots, p$.

Consider the R-module M generated by the columns of A, that is,

$$M = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s \rangle.$$

There is such a result below.

Theorem 3.1 *Equation $AX = B$ has a polynomial solution iff each column $\boldsymbol{b}_v$ of B is in the R-module M generated by all the columns of A.*

Proof Suppose that the equation $AX = B$ has a solution $X \in R^{s\times p}$. Now assume $X = (x_{lv})_{s\times p}$ and $\boldsymbol{x}_v = (x_{1v}, x_{2v}, \cdots, x_{sv})^{\mathrm{T}}$, then we have

$$A\boldsymbol{x}_v = \boldsymbol{b}_v, \quad v = 1, 2, \cdots, p.$$

Let's change the above expression to anther form to find the connection between the equation and the module. It follows that

$$x_{1v}\boldsymbol{a}_1 + x_{2v}\boldsymbol{a}_2 + \cdots + x_{sv}\boldsymbol{a}_s = \boldsymbol{b}_v,$$

where $v = 1, 2, \cdots, p$.

We can see that each column $\boldsymbol{b}_v$ of B can be expressed as an R-linear combination of the columns of A. Here we call $x_{1v}, x_{2v}, \cdots, x_{sv}$ the corresponding representation coefficients.

Thus, $\boldsymbol{b}_v \in M = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s\rangle$ for $v = 1, 2, \cdots, p$.

Conversely, if $\boldsymbol{b}_v \in M$ for $v = 1, 2, \cdots, p$, then there exist polynomials $\widetilde{x}_{1v}, \widetilde{x}_{2v}, \cdots, \widetilde{x}_{sv}$ in R such that

$$\widetilde{x}_{1v}\boldsymbol{a}_1 + \widetilde{x}_{2v}\boldsymbol{a}_2 + \cdots + \widetilde{x}_{sv}\boldsymbol{a}_s = \boldsymbol{b}_v.$$

Obviously, this implies that the equation $AX = B$ has a solution $X = (\widetilde{x}_{lv})_{s\times p}$ in R.

In particular, when $B = 0_{m\times p}$,

$$x_{1v}\boldsymbol{a}_1 + x_{2v}\boldsymbol{a}_2 + \cdots + x_{sv}\boldsymbol{a}_s = \boldsymbol{0},$$

which corresponds to the syzygy module $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$. It is easy to see that $\boldsymbol{x}_v = (x_{1v}, x_{2v}, \cdots, x_{sv})^{\mathrm{T}}$ as the solution of $AX = \boldsymbol{0}$ is exactly the element in $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$.

Now we give the general solution of the equation, if the matrix equation has a solution.

Proposition 3.2 *The general solution of the equation $AX = B$ is as follows:*

$$X = \widehat{X} + ST,$$

where $\widehat{X}$ is a particular solution of the equation $AX = B$, $S \in R^{s\times q}$ is a matrix consisting of q generators of $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$, and $T \in R^{q\times p}$ is an arbitrary polynomial matrix.

Proof According to the above analysis and the hypothesis, we have

$$A\widehat{X} = B, \quad AS = 0_{m\times q}.$$

Thus,

$$AX = A(\widehat{X} + ST) = A\widehat{X} + AST = B + 0 = B.$$

Based on Theorem 3.1 and Proposition 3.2, there are two things to do. The first is to examine if there is a solution for the equation $AX = B$. That is, we need to determine whether $\boldsymbol{b}_v$ is in M. This is actually the problem for submodule membership. Second, if the equation has a solution, we should give the general solution. Then, we need to obtain the representation coefficients $x_{1v}, x_{2v}, \cdots, x_{sv}$ such that $x_{1v}\boldsymbol{a}_1+x_{2v}\boldsymbol{a}_2+\cdots+x_{sv}\boldsymbol{a}_s = \boldsymbol{b}_v$ and a set of generators for the module $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$. Thanks to the followings from [36], the submodule membership and generators for syzygy modules can be solved by Gröbner bases for submodules.

Lemma 3.3 *Fix any monomial order $\succ_m$ on R^m and let $G = (\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_l)$ be an ordered s-tuple of elements of R^m. Then every $\boldsymbol{g} \in R^m$ can be written as*

$$\boldsymbol{g} = u_1\boldsymbol{g}_1 + u_2\boldsymbol{g}_2 + \cdots + u_l\boldsymbol{g}_l + \boldsymbol{r},$$

where $u_i \in R$, $\boldsymbol{r} \in R^m$, $\mathrm{LM}(u_i\boldsymbol{g}_i) \preceq \mathrm{LM}(\boldsymbol{g})$ for all i, and either $\boldsymbol{r} = \boldsymbol{0}$ or $\boldsymbol{r}$ is a k-linear combination of monomials none of which is divisible by any of $\mathrm{LM}(\boldsymbol{g}_1), \mathrm{LM}(\boldsymbol{g}_2), \cdots, \mathrm{LM}(\boldsymbol{g}_l)$. We call $\boldsymbol{r}$ the remainder on division by G and denote $\boldsymbol{r}$ by $\overline{\boldsymbol{g}}^G$.

Lemma 3.4 *Let G be a Gröbner basis for the submodule $M \subset R^m$ and $\boldsymbol{b} \in R^m$.*

1) *$\boldsymbol{b} \in M$ iff the remainder on division by G is zero.*
2) *A Gröbner basis for M generates M as a module: $M = \langle G \rangle$.*

It is clear that if we get a Gröbner basis, we can judge whether $\boldsymbol{b}_v$ is in M by the division algorithm in R^m. Moreover, the above lemma implies that a Gröbner basis of $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$ is also a set of generators.

What remains urgently to solve now is the problem of representation coefficients which is directly related to the particular solution of the equation. In the next section, we will introduce a practical measure for solving this problem in detail.

Remark 3.5 Since matrix multiplication is not commutative, for linear matrix equations in the ring of polynomials the following three basic types have been distinguished (see [37, 38]): The right-sided equations $A_1X_1 + A_2X_2 = B$, the left-sided equations $X_1A_1 + X_2A_2 = B$ and the two-sided equations $A_1X_1 + X_2A_2 = B$. In general, a different structure of equations calls for different algorithms. However, the three types of equations can be solved by the same method presented in the paper.

In the paper, we consider the right-sided equations and transform it into $AX = B$. Obviously, the left-sided equations can also be transformed into $A^{\mathrm{T}}X^{\mathrm{T}} = B^{\mathrm{T}}$. As for the two-sided equations, some operations are required to make it a similar form. The details are as follows: Treat each element of the unknown matrix X_1, X_2 as an unknown element x_{ij} and expand the matrix equation into a system of linear equations according to x_{ij}, then make all unknown elements and all elements of the matrix B into a column vector $\boldsymbol{x}$ and $\boldsymbol{b}$, respectively, so the equations can be transformed into a new matrix equation $A\boldsymbol{x} = \boldsymbol{b}$. For example:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} + \begin{bmatrix} \overline{x}_{11} & \overline{x}_{12} & \overline{x}_{13} \\ \overline{x}_{21} & \overline{x}_{22} & \overline{x}_{23} \end{bmatrix} \begin{bmatrix} \overline{a}_{11} \\ \overline{a}_{21} \\ \overline{a}_{31} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}.$$

Expand the matrix equation:

$$a_{11}x_{11}+a_{12}x_{21}+\overline{a}_{11}\overline{x}_{11}+\overline{a}_{21}\overline{x}_{12}+\overline{a}_{31}\overline{x}_{13}+0\cdot\overline{x}_{21}+0\cdot\overline{x}_{22}+0\cdot\overline{x}_{23}=b_{11};$$
$$a_{21}x_{11}+a_{22}x_{21}+0\cdot\overline{x}_{11}+0\cdot\overline{x}_{12}+0\cdot\overline{x}_{13}+\overline{a}_{11}\overline{x}_{21}+\overline{a}_{21}\overline{x}_{22}+\overline{a}_{31}\overline{x}_{23}=b_{12}.$$

Then

$$\begin{bmatrix} a_{11} & a_{12} & \overline{a}_{11} & \overline{a}_{21} & \overline{a}_{31} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & \overline{a}_{11} & \overline{a}_{21} & \overline{a}_{31} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ \overline{x}_{11} \\ \overline{x}_{12} \\ \overline{x}_{13} \\ \overline{x}_{21} \\ \overline{x}_{22} \\ \overline{x}_{23} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}.$$

That is, $A\boldsymbol{x}=\boldsymbol{b}$. Thus, the left-sided equations and the two-sided equations can also be solved with the Gröbner basis method.

4 Measure and Algorithm

In this section, we first introduce the GVW algorithm[29], a signature-based algorithm for computing the Gröbner basis, and extend this algorithm to the module R^m, which will be a powerful tool to solve the problem mentioned in the previous section. Then we will give an algorithm for computing the general solution of polynomial matrix Diophantine equations.

4.1 GVW Algorithm on Modules

Gao, et al.[29] have presented a new framework for computing Gröbner bases over polynomial ring R, that is, the well-know GVW algorithm. The algorithm acts on a larger module involving both the ideal and the syzygy module for given polynomials, and computes the strong Gröbner basis for this big module which contains Gröbner bases for both the ideal and the syzygy module. Most importantly, the constructed module expressing the element in the ideal as an R-linear combination of the input polynomials can encode the representation coefficients. Based on these facts and the efficiency of the GVW algorithm, it's wise to extend the GVW algorithm to the module. This important measure helps to simultaneously solve the problem of whether there is a solution and the expression of the general solution.

Now we extend the GVW algorithm for polynomial rings to modules.

All notations are the same as before. Let $\Lambda=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_s)$, we construct a larger module in $R^s\times R^m$:

$$\widetilde{M}=\{(\boldsymbol{u},\boldsymbol{f})\in R^s\times R^m:\boldsymbol{u}\in R^s \text{ and } \boldsymbol{u}^{\mathrm{T}}\Lambda^{\mathrm{T}}=\boldsymbol{f}^{\mathrm{T}}\}.$$

It is obvious that $\widetilde{M}$ is generated by $(\widetilde{\boldsymbol{e}}_1,\boldsymbol{a}_1),(\widetilde{\boldsymbol{e}}_2,\boldsymbol{a}_2),\cdots,(\widetilde{\boldsymbol{e}}_s,\boldsymbol{a}_s)$, where $\widetilde{\boldsymbol{e}}_i$ is the i-th unit vector of R^s, and $\boldsymbol{f}$ is expressed as an R-linear combination of $\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_s$. Here we call

$\boldsymbol{u}$ the representation coefficient vector for $\boldsymbol{f}$ with respect to $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s$. When $\boldsymbol{f} = \boldsymbol{0}$, the corresponding $\boldsymbol{u}$ is an element in $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$. We say $z^\alpha \widetilde{\boldsymbol{e}}_i$ divides $z^\beta \widetilde{\boldsymbol{e}}_j$ if z^α divides z^β and $i = j$.

Fix a module order $\succ_m$ in R^m, and a module order $\succ_s$ in R^s. For convenience, we denote the leading monomials $\mathrm{LM}_{\succ_m}(\boldsymbol{u})$ and $\mathrm{LM}_{\succ_s}(\boldsymbol{f})$ by $\mathrm{LM}(\boldsymbol{u})$ and $\mathrm{LM}(\boldsymbol{f})$ for any element $\boldsymbol{u} \in R^m$, $\boldsymbol{f} \in R^s$. Similarly for the leading term and leading coefficient. For any $\boldsymbol{p} = (\boldsymbol{u}, \boldsymbol{f})$ in $\widetilde{M}$, the $\mathrm{LM}(\boldsymbol{u})$ is called the signature of $\boldsymbol{p}$, denoted by $S(\boldsymbol{p})$.

We below define two types of top-reductions: Regular top-reduction and super top-reduction. In the actual algorithm process, we only need to perform the regular top-reduction.

Suppose $\boldsymbol{p}_1 = (\boldsymbol{u}_1, \boldsymbol{f}_1)$, $\boldsymbol{p}_2 = (\boldsymbol{u}_2, \boldsymbol{f}_2) \in \widetilde{M}$. When $\boldsymbol{f}_1, \boldsymbol{f}_2 \neq \boldsymbol{0}$, $\mathrm{LM}(\boldsymbol{f}_2)$ divides $\mathrm{LM}(\boldsymbol{f}_1)$ and $t\mathrm{LM}(\boldsymbol{u}_2) \preceq \mathrm{LM}(\boldsymbol{u}_1)$, where $t = \mathrm{LM}(\boldsymbol{f}_1)/\mathrm{LM}(\boldsymbol{f}_2)$. If $\mathrm{LM}(\boldsymbol{u}_1 - ct\boldsymbol{u}_2) = \mathrm{LM}(\boldsymbol{u}_1)$ where $c = \mathrm{LC}(\boldsymbol{f}_1)/\mathrm{LC}(\boldsymbol{f}_2)$, we say $\boldsymbol{p}_1$ is regular top-reducible by $\boldsymbol{p}_2$, and the corresponding top-reduction is

$$\mathrm{Red}(\boldsymbol{p}_1, \boldsymbol{p}_2) = \boldsymbol{p}_1 - ct\boldsymbol{p}_2 = (\boldsymbol{u}_1 - ct\boldsymbol{u}_2, \boldsymbol{f}_1 - ct\boldsymbol{f}_2),$$

and super top-reducible otherwise.

In addition, when $\boldsymbol{f}_2 = \boldsymbol{0}$ and $\mathrm{LM}(\boldsymbol{u}_2)$ divides $\mathrm{LM}(\boldsymbol{u}_1)$, $\boldsymbol{p}_1$ is super top-reducible by $\boldsymbol{p}_2$.

Like the GVW algorithm, we define the strong Gröbner basis for $\widetilde{M}$.

Definition 4.1 For a finite subset $\widetilde{G} = \{(\boldsymbol{u}_1, \boldsymbol{f}_1), (\boldsymbol{u}_2, \boldsymbol{f}_2), \cdots, (\boldsymbol{u}_l, \boldsymbol{f}_l)\} \in \widetilde{M}$, if any nonzero element $(\boldsymbol{u}, \boldsymbol{f})$ in $\widetilde{M}$ is top-reducible by some pair $(\boldsymbol{u}_i, \boldsymbol{f}_i)$ in $\widetilde{G}$, then $\widetilde{G}$ is called a strong Gröbner basis for $\widetilde{M}$.

Further, Proposition 2.2 in [29] still holds and the proof is same.

Proposition 4.2 *Given the orders $\succ_m$ in R^m and $\succ_s$ in R^s. Suppose that $\widetilde{G} = \{(\boldsymbol{u}_1, \boldsymbol{f}_1), (\boldsymbol{u}_2, \boldsymbol{f}_2), \cdots, (\boldsymbol{u}_l, \boldsymbol{f}_l)\}$ is a strong Gröbner basis for $\widetilde{M}$, then*

1) $L = \{\boldsymbol{u}_i : \boldsymbol{f}_i = \boldsymbol{0}, 1 \leq i \leq l\}$ is a Gröbner basis for the syzygy module of $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s\}$, and

2) $G = \{\boldsymbol{f}_i : 1 \leq i \leq l\}$ is a Gröbner basis for $M = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s \rangle$ in R^m.

From the above proposition, we know that by computing a strong Gröbner basis for the constructed special module $\widetilde{M}$, one can obtain a Gröbner basis G of module M and a Gröbner basis L of $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$. At the same time, the set U of the representation coefficient vectors $\boldsymbol{u}$ for elements $\boldsymbol{f}_i$ in the Gröbner bases of module M with respect to $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s$ can be gotten.

Next, let's talk about how to compute the strong Gröbner basis for $\widetilde{M}$, i.e., the theoretical foundation of the GVW algorithm.

Suppose $\boldsymbol{p}_1 = (\boldsymbol{u}_1, \boldsymbol{f}_1), \boldsymbol{p}_2 = (\boldsymbol{u}_2, \boldsymbol{f}_2) \in \widetilde{M}$, where $\boldsymbol{f}_1, \boldsymbol{f}_2 \neq \boldsymbol{0}$, and $\mathrm{LM}(\boldsymbol{f}_1)$ and $\mathrm{LM}(\boldsymbol{f}_2)$ are monomials containing the same standard basis vector $\boldsymbol{e}_i$. Let $c = \mathrm{LC}(\boldsymbol{f}_1)/\mathrm{LC}(\boldsymbol{f}_2)$, $\boldsymbol{t} = \mathrm{LCM}(\mathrm{LM}(\boldsymbol{f}_1), \mathrm{LM}(\boldsymbol{f}_2))$, $t_1 = \boldsymbol{t}/\mathrm{LM}(\boldsymbol{f}_1)$, $t_2 = \boldsymbol{t}/\mathrm{LM}(\boldsymbol{f}_2)$. Assume $\boldsymbol{T} = \max\{t_1\mathrm{LM}(\boldsymbol{u}_1), t_2\mathrm{LM}(\boldsymbol{u}_2)\} = t_1\mathrm{LM}(\boldsymbol{u}_1)$. If

$$\mathrm{LM}(t_1\boldsymbol{u}_1 - ct_2\boldsymbol{u}_2) = \boldsymbol{T},$$

then $t_1\boldsymbol{p}_1$ is called the J-pair of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.

Definition 4.3 We say that a pair $(\boldsymbol{u}, \boldsymbol{f}) \in \widetilde{M}$ is covered by $G \subset \widetilde{M}$, if there is a pair $(\boldsymbol{u}_i, \boldsymbol{f}_i) \in G$ such that $\mathrm{LM}(\boldsymbol{u}_i)$ divides $\mathrm{LM}(\boldsymbol{u})$ and $\mathrm{LM}(\boldsymbol{f}) \succ_m t\mathrm{LM}(\boldsymbol{f}_i)$, where $t = \mathrm{LM}(\boldsymbol{u})/\mathrm{LM}(\boldsymbol{u}_i)$.

The main theorem of the GVW algorithm on modules is as follows.

Theorem 4.4 *Given the orders $\succ_m$ in R^m and $\succ_s$ in R^s. Let G be a finite subset of $\widetilde{M}$ satisfies for any monomial $\boldsymbol{T} \in R^s$, there is a pair $(\boldsymbol{u}, \boldsymbol{f}) \in G$ and a monomial $t \in R$ such that $\boldsymbol{T} = t\mathrm{LM}(\boldsymbol{u})$. Then the following are equivalent:*

1) *G is a strong Gröbner basis for $\widetilde{M}$;*
2) *Every J-pair of G is covered by G.*

Proof The proof process is same as that of Theorem 2.4 in [29] except that the polynomial v is replaced by the vector $\boldsymbol{f}$. ∎

It follows from the above theorem that any J-pair that is covered by G can be discarded without performing any reductions. As a consequence, the priori criteria in [29] still hold on the module.

Corollary 4.5 (Syzygy criterion) *For any J-pair $(\boldsymbol{u}, \boldsymbol{f})$ of G, it can be discarded if it is top-reducible by a syzygy.*

Corollary 4.6 (Signature criterion) *Among all J-pairs with a same signature, only one (with the $\boldsymbol{f}$-part minimal) needs to be stored.*

Corollary 4.7 (Rewrite criterion) *For any J-pair $(\boldsymbol{u}, \boldsymbol{f})$ of G, it can be discarded if it is covered by G.*

Based on Theorem 4.4 and the above three criteria, we can give the extended version of the GVW algorithm on modules, that is, input a generator set $F = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s\} \subset R^m$ of module M then output a strong Gröbner basis which contains a Gröbner basis G for module $M = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_S \rangle$, a set U of the corresponding representation coefficient vectors for G, and a set L of a Gröbner basis for $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$.

Since the whole algorithm process is consistent with the GVW algorithm on polynomial rings, we will not repeat the whole algorithm process. Note that for the case of modules, we can't directly construct the trivial principle syzygy, so the syzygy criterion is not well utilized.

Remark 4.8 In fact, for the implementation of the algorithm we have adopted a more efficient way mentioned in [29] to obtain a strong Gröbner basis. That is storing the signature $\boldsymbol{T} = \mathrm{LM}(\boldsymbol{u})$ instead of $\boldsymbol{u}$ during the calculation process, then combining with the approach of recovering the complete $\boldsymbol{u}$-part or strong Gröbner basis from the signature.

4.2 Algorithm for Polynomial Matrix Diophantine Equations

Suppose that $G = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_l\}$ is a Gröbner basis for $M = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s \rangle$, $L = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_q\}$ is a Gröbner basis for $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$ and $U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_l\}$ is a set of the representation coefficient vectors such that $A\boldsymbol{u}_i = \boldsymbol{g}_i$, where $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s$ are the columns of A. According to Theorem 3.1 and Lemma 3.4, if each $\overline{\boldsymbol{b}_j}^G = \boldsymbol{0}$, where $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_p$ are the columns of B, the equation $AX = B$ has a solution. Write

$$\boldsymbol{b}_j = \sum_{k=1}^{l} c_{jk}\boldsymbol{g}_k \quad \text{and} \quad \boldsymbol{g}_i = \sum_{t=1}^{s} u_{it}\boldsymbol{a}_t.$$

Then

$$\boldsymbol{b}_j = \sum_{k=1}^{l}\sum_{t=1}^{s} c_{jk}u_{it}\boldsymbol{a}_t = \sum_{t=1}^{s}\sum_{k=1}^{l} c_{jk}u_{it}\boldsymbol{a}_t.$$

By Proposition 3.2, if L is empty (or in a sense, $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s) = \{\boldsymbol{0}\}$), then the solution is unique. Otherwise, the general solution is of the form: $X = \widehat{X} + ST$ where $\widehat{X}$ is a particular solution of $AX = B$, the matrix $S = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_q) \in R^{s\times q}$ and $T \in R^{q\times p}$ is an arbitrary polynomial matrix.

Now we are ready to formally give the algorithm for solving multivariate polynomial matrix Diophantine equations by the Gröbner basis method.

Algorithm PMDE(A, B)
Input $A \in R^{m\times s}$; $B \in R^{m\times p}$.
Output $\widehat{X}$, a particular solution of $AX = B$;
S, a matrix in $R^{s\times q}$ consisting of q generators for $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$.

1. Set $F = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s\} \subset R^m$, where $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s$ are the column vectors of A;
2. By the extended GVW algorithm on modules, compute a Gröbner basis $G = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_l\}$ for $M = \langle F\rangle$, a Gröbner basis L for $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_s)$, and a set $U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_l\}$ of the representation coefficients such that $A\boldsymbol{u}_i = \boldsymbol{g}_i$ for each $1 \le i \le l$;
3. Compute the remainder $\boldsymbol{r}_j$ and the representation coefficients c_{jk} of $\boldsymbol{b}_j$ on division by G: $\boldsymbol{b}_j = c_{j1}\boldsymbol{g}_1 + c_{j2}\boldsymbol{g}_2 + \cdots + c_{jl}\boldsymbol{g}_l + \boldsymbol{r}_j$ for $1 \le j \le p$, where $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_p$ are the column vectors of B;
4. If there is a $\boldsymbol{r}_j$ is not zero, then return "The equation has no solution";
5. Else, $\widehat{X} = \widetilde{U}C$, where the matrix $\widetilde{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_l)$ and $C = (c_{jk})_{l\times p}$; if L is empty, then return "The equation has a unique solution: $\widehat{X}$";
6. Else, return "The equation has the general solution: $X = \widehat{X} + ST$", where $S \in R^{s\times q}$ consisting of elements in L and $T \in R^{q\times p}$ is an arbitrary polynomial matrix.

Obviously, Algorithm PMDE(A, B) is correct and terminated. The most noteworthy thing is that this algorithm is a complete algorithm. In addition, The proposed algorithm has been implemented on the computer algebra system Maple, and the codes and examples are available on the web: http://www.mmrc.iss.ac.cn/ dwang/software.html.

5 Example

We use the following simple example appeared in [28] to illustrate the main steps in Algorithm PMDE(A, B).

Example 5.1 Consider the matrix Diophantine equation:

$$A_1X_1 + A_2X_2 = B,$$

where

$$A_1 = \begin{bmatrix} z_1+1 & z_2 \\ z_2+1 & z_2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad B = \begin{bmatrix} z_1-z_2+1 & z_1+1 \\ 0 & z_2+1 \end{bmatrix}.$$

This equation can be written as

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = AX = B$$

with

$$A = \begin{bmatrix} z_1+1 & z_2 & 0 \\ z_2+1 & z_2 & -1 \end{bmatrix}.$$

Step 1 Let $M = \langle \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \rangle$ generated by the column vectors $\boldsymbol{a}_1 = (z_1+1, z_2+1)^{\mathrm{T}}$, $\boldsymbol{a}_2 = (z_2, z_2)^{\mathrm{T}}$, $\boldsymbol{a}_3 = (0, -1)^{\mathrm{T}}$ of A. By the extended GVW algorithm in Subsection 4.1, we obtain a Gröbner basis G for M w.r.t $\succ_2$, where

$$\begin{aligned} G =& \{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \boldsymbol{g}_4\} \\ =& \{(z_1+1, z_2+1)^{\mathrm{T}}, (z_2, z_2)^{\mathrm{T}}, (0, -1)^{\mathrm{T}}, (-z_2, z_1z_2 - z_2^2 - z_2)^{\mathrm{T}}\}; \end{aligned}$$

a Gröbner basis L for $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3)$ w.r.t $\succ_3$, where

$$L = \{(-z_2, z_1+1, z_1z_2 - z_2^2)^{\mathrm{T}}\};$$

and a set U of the representation coefficient vectors such that the elements in G can be expressed as R-linear combination of generators $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$, i.e., $\boldsymbol{g}_i = A\boldsymbol{u}_i$, where

$$U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4\} = \{(1,0,0)^{\mathrm{T}}, (0,1,0)^{\mathrm{T}}, (0,0,1)^{\mathrm{T}}, (-z_2, z_1, 0)^{\mathrm{T}}\}.$$

Where $\succ_2$ and $\succ_3$ are the TOP order under the lexicographic order with $z_1 \succ z_2$.

Step 2 Determine whether the equation has the polynomial solution by the submodule membership. That is to determine if each column $\boldsymbol{b}_j$ of B is in M or not. We compute the remainder $\boldsymbol{r}_j$ of $\boldsymbol{b}_j$ on division by $G = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \boldsymbol{g}_4\}$, as well as the representation coefficients c_{jk}, where $\boldsymbol{b}_1 = (z_1 - z_2 + 1, 0)^{\mathrm{T}}$, $\boldsymbol{b}_2 = (z_1+1, z_2+1)^{\mathrm{T}}$. The result is as follows:

$$\begin{aligned} &\boldsymbol{b}_1 = \boldsymbol{g}_1 - \boldsymbol{g}_2 + \boldsymbol{g}_3; \\ &\boldsymbol{b}_2 = \boldsymbol{g}_1. \end{aligned}$$

Obviously, $\boldsymbol{r}_1 = \boldsymbol{0}$ and $\boldsymbol{r}_2 = \boldsymbol{0}$. Thus, the equation has the polynomial solution.

Moreover, by the relations: $\boldsymbol{b}_j = \sum_{k=1}^{l} c_{jk}\boldsymbol{g}_k$ and $\boldsymbol{g}_k = A\boldsymbol{u}_k$, i.e., $B = C\widetilde{G}$ and $\widetilde{G} = A\widetilde{U}$ (where $\widetilde{G}$ and $\widetilde{U}$ are the matrix form of the sets G and U, respectively), we obtain a solution: $\widehat{X} = \widetilde{U}C$, where

$$\widetilde{U} = \begin{bmatrix} 1 & 0 & 0 & -z_2 \\ 0 & 1 & 0 & z_1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Step 3 Determine if the solution is unique. According to L obtained in Step 1 is not empty (i.e., the syzygy module $\mathrm{Syz}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3)$ is nonzero), the equation has more than one solution. Then we give the general solution: $X = \widehat{X} + ST$, where S consists of generators of $\mathrm{Syz}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$, i.e., the elements in L, and T is an arbitrary polynomial matrix.

$$S = \begin{bmatrix} -z_2 \\ z_1 + 1 \\ z_1 z_2 - z_2^2 \end{bmatrix}, \quad T = \begin{bmatrix} d_1 & d_2 \end{bmatrix}, \quad d_1, d_2 \in R.$$

In summary, the matrix Diophantine equations $A_1X_1 + A_2X_2 = B$ has the general solution:

$$X = \widehat{X} + ST = \begin{bmatrix} X_1 \\ \hline X_2 \end{bmatrix} = \begin{bmatrix} 1 - d_1 z_2 & 1 - d_2 z_2 \\ -1 + d_1(z_1 + 1) & d_2(z_1 + 1) \\ \hline 1 + d_1(z_1 z_2 - z_2^2) & d_2(z_1 z_2 - z_2^2) \end{bmatrix}.$$

6 Concluding Remarks

In the paper, a complete algorithm for solving multivariate polynomial matrix Diophantine equations by the Gröbner basis method is presented, which is from the idea of modules to explore matrix Diophantine equations, then by means of powerful features and theoretical foundation of Gröbner bases for modules, the problem for determining and computing the solution of matrix Diophantine equations is successfully solved. Meanwhile, in order to better solve the Gröbner bases for the module and syzygy module, as well as the representation coefficients problem, we have utilized the extended version of the GVW algorithm on modules as a powerful tool. Moreover, as mentioned in Remark 3.5 the two-sided equations $A_1X_1 + X_2A_2 = B$ can also be solved by the Gröbner basis method.

References

[1] Blomberg H and Ylinen R, *Algebraic Theory for Multivariable Linear Systems*, Academic Press, London, 1983.

[2] Fliess M, Some basic structural properties of generalized linear systems, *Systems Control Lett.*, 1990, **15**: 391–396.

[3] Hautus M and Heymann M, Linear feedback: An algebraic approach, *SIAM J. Control and Optimization*, 1978, **16**: 83–105.

[4] Kalman R E, Falb P L, and Arbib M A, *Topics in Mathematical System Theory*, McGraw-Hill, New York, 1969.

[5] Rosenbrock H H, *State-Space and Multivariable Theory*, Nelson, London, 1970.

[6] Desoer C, Liu R W, Murray J, et al., Feedback system design: The fractional representation approach to analysis and synthesis, *IEEE Trans. Autom. Control*, 1980, **25**: 399–412.

[7] Hunt K and Šebek M, Implied polynomial matrix equations in multivariable stochastic optimal control, *Automatica*, 1991, **27**: 395–398.

[8] Kučera V, Closed-loop stability of discrete linear single-variable systems, *Kybernetika*, 1974, **10**: 146–171.

[9] Kučera V, Stability of discrete linear feedback systems, *IFAC Proceedings Volumes*, 1975, **8**: 573–578.

[10] Chen M J and Desoer C, Algebraic theory for robust stability of interconnected systems: Necessary and sufficient conditions, *IEEE Trans. Autom. Control*, 1984, **29**: 511–519.

[11] Vidyasagar M, *Control System Synthesis: A Factorization Approach*, MIT Press, Cambridge, MA, 1985.

[12] Estrada R L and Martınez-Garcıa J, Fixed poles and disturbance rejecting feedback synthesis, *Automatica*, 1999, **35**: 1737–1740.

[13] Kučera V, Disturbance rejection: A polynomial approach, *IEEE Trans. Autom. Control*, 1983, **28**: 508–511.

[14] Koussiouris T and Tzierakis K, Rejection of measurable disturbances with decoupling or pole placement, *Proceedings of the Third European Control Conference*, Rome, Italy, 1995, 2257–2262.

[15] Astrom K, Robustness of a design method based on assignment of poles and zeros, *IEEE Trans. Autom. Control*, 1980, **25**: 588–591.

[16] Kučera V, Exact model matching, polynomial equation approach, *Int. J. Syst. Sci.*, 1981, **12**: 1477–1484.

[17] Hunt K J and Kučera V, The standard H_2-optimal control problem: A polynomial solution, *Int. J. Control*, 1992, **56**: 245–251.

[18] Feinstein J and Bar-Ness Y, The solution of the matrix polynomial equation $A(s)X(s) + B(s)Y(s) = C(s)$, *IEEE Trans. Autom. Control*, 1984, **29**: 75–77.

[19] Lai Y S, An algorithm for solving the matrix polynomial equation $B(s)D(s)+A(s)N(s) = H(s)$, *IEEE Trans. Circuits and Syst.*, 1989, **36**: 1087–1089.

[20] Chang F R, Shieh L S, and Navarro J, A simple division method for solving Diophantine equations and associated problems, *Int. J. Syst. Sci.*, 1986, **17**: 953–968.

[21] Wolovich W A, The canonical diophantine equations with applications, *American Control Conference*, IEEE, 1983, 1165–1171.

[22] Fang C H, A simple approach to solving the diophantine equation, *IEEE Trans. Autom. Control*, 1992, **37**: 152–155.

[23] Yamada M and Funahashi Y, A simple algorithm for solutions of Diophantine equation, *Trans. of the Society of Instrument and Control Engineers*, 1994, **30**: 261–266.

[24] Karampetakis N, Computation of the generalized inverse of a polynomial matrix and applications, *Linear Algebra and Its Applications*, 1997, **252**: 35–60.

[25] Estrada M B, Solving the right Diophantine equation in a geometric way, *Proceedings of the* 34*th IEEE Conference on Decision and Control*, 1995, **1**: 309–310.

[26] Tzekis P, A new algorithm for the solution of a polynomial matrix Diophantine equation, *Appl. Math. Comput.*, 2007, **193**: 395–407.

[27] Šebek M, n-D polynomial matrix equations, *IEEE Trans. Autom. Control*, 1988, **33**: 499–502.

[28] Tzekis P, Antoniou E, and Vologiannidis S, Computation of the general solution of a multivariate polynomial matrix Diophantine equation, 21*st Mediterranean Conference on Control and Automation*, IEEE, 2013, 677–682.

[29] Gao S, Volny IV F, and Wang M, A new framework for computing Gröbner bases, *Mathematics of Computation*, 2016, **85**: 449–465.

[30] Eder C and Faugère J C, A survey on signature-based algorithms for computing Gröbner bases, *J. Symbol. Comput.*, 2017, **80**: 719–784.

[31] Faugère J C, A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), *Proceedings of the* 2002 *International Symposium on Symbolic and Algebraic Computation*, ACM, 2002, 75–83.

[32] Sun Y and Wang D K, A new proof for the correctness of the F5 algorithm, *Science China Mathematics*, 2013, **56**: 745–756.

[33] Lin Z, Ying J Q, and Xu L, Factorizations for n-D polynomial matrices, *Circuits, Systems and Signal Processing*, 2001, **20**: 601–618.

[34] Lu D, Ma X, and Wang D, A new algorithm for general factorizations of multivariate polynomial matrices, *Proceedings of the* 2017 *ACM on International Symposium on Symbolic and Algebraic Computation*, ACM, 2017, 277–284.

[35] Lu D, Wang D K, and Xiao F H, Factorizations for a class of multivariate polynomial matrices, *Multidimensional Systems and Signal Processing*, 2020, **31**: 989–1004.

[36] Cox D A, Little J, and O'shea D, *Using Algebraic Geometry*, Vol 185, Springer Science and Business Media, New York, 2006.

[37] Kučera V, *Discrete Linear Control: The Polynomial Equation Approach*, J. Wiley Chichester, 1979.

[38] Šebek M, Two-sided equations and skew primeness for n-D polynomial matrices, *Systems Control Lett.*, 1989, **12**: 331–337.