計測自動制御学会東北支部 第 272 回研究集会 (2012.5.30)
資料番号 272-10

行列基本変形による多次元 Fornasini-Marchesini 状態空間モデル実現法の次数評価：1 入出力の場合

Order Evaluation of the Elementary Operation Approach to Multidimensional Fornasini-Marchesini State-space Model Realization: the SISO case

○植栗健太 *，　松下慎也 *，　徐粒 *，

○ Kenta Ueguri*，　Shin-ya Matsushita，　Li Xu*，

* 秋田県立大学

*Akita Prefectural University

キーワード： 多次元システム (multidimensional systems), Fornasini-Marchesini 状態空間モデル (Fornasini-Marchesini state-space model), 状態空間実現 (state-space realization)

連絡先： 〒 015-0055　秋田県由利本荘市土谷字海老ノ口 84-4 秋田県立大学 大学院 システム科学技術研究科 電子情報システム学専攻 システム制御工学研究室 植栗健太,
Tel.: (0184)27-2101,　Fax.: (0184)27-2187,　E-mail: m14b002@akita-pu.ac.jp

1. はじめに

多次元 (n-D) システムの伝達関数または伝達関数行列からその状態空間方程式を求める実現問題は, n-D システム理論の基礎的な問題であり, LFR(線形分数表現) によるシステムの不確かさのモデリングや分布型センサーネットワークの実現など多くの分野に応用することができる [1, 2, 3]. 1 つの伝達関数あるいは伝達関数行列に対応する状態空間実現は無数に存在するが, 従来の 1 次元 (1-D) の場合と違い, n-D ($n \geqq 2$) 状態空間実現の次数は伝達関数の次数のみならず, その構造および係数の値にも関連しているため, 最小実現を求めることは極めて困難となる [1, 2, 3]. そのため, いかに低次数の n-D 状態空間実現を求めるかが重要となる.

n-D システムの代表的な状態空間モデルとして, Roesser モデルと Fornasini-Marchesini(F-M)(第二) モデルが挙げられる. Roesser モデルによる n-D システムの状態空間実現問題は多くの研究者によって検討され, いくつかの実現法が提案されている [1). その中でも, 行列基本変形に基づく実現法は, 概念的に理解しやすく, 伝達関数あるいは伝達関数行列の係数による実現への影響を柔軟に考慮することができるため注目されている [4, 5].

同様に F-M モデルについてもいくつかの実現法が提案されている [3, 6, 7]. 最近, 行列基本変形による n-D システムの F-M 状態空間モデル実現法が提案された [8]. しかし, この方法で得られる実現の次数は, 実現を行う際の行列基本変形の順番や独立変数の処理順によって異なる. 従って，いかに低い実現次数を得ることができる処理順を見つけるかが次の課題となっている.

そこで本論文では, まず行列基本変形による F-M 状態空間モデル実現法の一般的な手順を明確に示す. そして, 効率よく最小次数の実現に対応する独立変数の順番を見つけるため, 実際に実現操作を行わずに行列基本変形による F-M 状態空間モデル実現法の次数を評価するアルゴリズムを提案し, その有効性を示す. ただし, 本論文では 1 入出力 (SISO) システ

ムの場合のみを考える.

2. F-M モデルの実現問題

n-D SISO 線形離散システムの F-M(local) 状態空間モデルは次式で与えられる.

$$
\begin{aligned}
&x(i_1+1,\ldots,i_n+1)\\
&= A_1x(i_1,i_2+1,\ldots,i_n+1)+\ldots\\
&\quad + A_nx(i_1+1,\ldots,i_{n-1}+1,i_n)\\
&\quad + B_1u(i_1,i_2+1,\ldots,i_n+1)+\ldots\\
&\quad + B_nu(i_1+1,\ldots,i_{n-1}+1,i_n)\\
&y(i_1,...,i_n)=Cx(i_1,...,i_n)+Du(i_1,...,i_n)
\end{aligned}
\tag{1}
$$

ここで, $x(i_1,\ldots,i_n)\in\mathbb{R}^r$ は (local) 状態ベクトル, $u(i_1,\ldots,i_n)\in\mathbb{R}$, $y(i_1,\ldots,i_n)\in\mathbb{R}$ はそれぞれ入力と出力であり, $A_1,\cdots,A_n\in\mathbb{R}^{r\times r}$, $B_1,\ldots,B_n\in\mathbb{R}^{r\times 1}$, $C\in\mathbb{R}^{1\times r}$, $D\in\mathbb{R}$ は実数行列で, r を F-M モデルの次数という. また, システム (1) を簡単に ($\boldsymbol{A}$, $\boldsymbol{B}$, C, D) で表す場合がある. ただし, $\boldsymbol{A}=(A_1,\ldots,A_n)$, $\boldsymbol{B}=(B_1,\ldots,B_n)$ である.

システム (1) の伝達関数は次式で与えられる.

$$
H(z_1,...,z_n)=D+C\left(I_r-\sum_{j=1}^{n}A_jz_j\right)^{-1}\sum_{j=1}^{n}B_jz_j
\tag{2}
$$

ただし, $z_1,\ldots,z_n$ は遅れ作用素を表す.

n-D (n 変数) 単項式 $cz_1^{k_1}\ldots z_n^{k_n}$ について, 指数 k_i, $i\in\{1,\ldots,n\}$ をこの単項式の z_i に関する次数といい, $k_1+\cdots+k_n$ を全次数という. 特に, cz_i, $i\in\{1,\ldots,n\}$ のように全次数が 1 となる単項式を 1-D 線形単項式と呼ぶ. n-D 多項式は n-D 単項式の有限個の和で表され, n-D 多項式 $p(z_1,\ldots,z_n)$ に含まれる全ての単項式の全次数と z_i に関する次数の最大値をそれぞれ $p(z_1,\ldots,z_n)$ の全次数と z_i に関する次数と呼び, $\deg\, p(z_1,\ldots,z_n)$, $\deg_{z_i}p(z_1,\ldots,z_n)$ で表す. 特に, $c_0+c_1z_1+\cdots+c_nz_n$ のように全次数が 1 となる多項式を n-D 線形多項式と呼ぶ. さらに, n-D 多項式ベクトル $\boldsymbol{p}(z_1,\ldots,z_n)$ は, n-D 多項式を成分に持つベクトルで, $\boldsymbol{p}(z_1,\ldots,z_n)$ に含まれる n-D 多項式の全次数と z_i に関する次数の最大値をそれぞれ $\boldsymbol{p}(z_1,\ldots,z_n)$ の全次数と z_i に関する次数と呼び, $\deg\,\boldsymbol{p}(z_1,\ldots,z_n)$, $\deg_{z_i}\boldsymbol{p}(z_1,\ldots,z_n)$ で表す [4, 5].

$a(z_1,\ldots,z_n)$ と $b(z_1,\ldots,z_n)$ を $z_1,\ldots,z_n$ に関する n-D 多項式とする. n-D 有理関数 $g(z_1,\ldots,z_n)=a(z_1,\ldots,z_n)/b(z_1,\ldots,z_n)$ に対し, $b(0,\ldots,0)\neq 0$ であるとき, $g(z_1,\ldots,z_n)$ は causal であるという [1].

与えられた伝達関数 $H(z_1,\ldots,z_n)$ に対し, (2) 式を満たす ($\boldsymbol{A}$, $\boldsymbol{B}$, C, D) を求める問題を, F-M モデルの実現問題といい, 求めた ($\boldsymbol{A}$, $\boldsymbol{B}$, C, D) を $H(z_1,\ldots,z_n)$ の F-M モデル実現と呼ぶ.

3. 行列基本変形による F-M 状態空間モデル実現法

次の causal な n-D 伝達関数について考える.

$$
H(z_1,\ldots,z_n)=\frac{n(z_1,\ldots,z_n)}{d(z_1,\ldots,z_n)}
\tag{3}
$$

ただし, $n(z_1,\ldots,z_n)$, $d(z_1,\ldots,z_n)$ は, $z_1,\ldots,z_n$ に関する n-D 多項式である. さらに, 一般性を失わず $n(0,\ldots,0)=0$, $d(0,\ldots,0)=1$ とする. このとき $D=0$ となり行列 $\boldsymbol{A}$, $\boldsymbol{B}$, C のみを求めればよい [8].

まず次の形の行列 M_0 を定義する.

$$
M_0\triangleq\begin{bmatrix} d(z_1,\ldots,z_n) & n(z_1,\ldots,z_n)\\ 1 & \mathrm{x}\end{bmatrix}
\tag{4}
$$

ただし, x は最終行と他の行 (もしくは最終列と他の列) の入れ替え, あるいは x の表記を変えるような行列基本変形を禁止するための指標である [8]. M_0 に対して行列基本変形と拡大操作を行うことで,

$$
M=\begin{bmatrix} I_r-\sum_{i=1}^{n}A_iz_i & \sum_{i=1}^{n}B_iz_i\\ C & \mathrm{x}\end{bmatrix}
\tag{5}
$$

の形の行列 M に変形でき, このときの $\boldsymbol{A}$, $\boldsymbol{B}$, C と $D=0$ は $H(z_1,\ldots,z_n)$ の実現となる [8].

最終的に得られる行列 M はいくつかの構造的性質を持つ [8]. その性質に基づき, 次の一般的な実現手順を与えることができる.

まず, 行列 M_0 を

$$
M_0=\begin{bmatrix} 1+\hat{d}(z_1,\ldots,z_n) & n(z_1,\ldots,z_n)\\ 1 & \mathrm{x}\end{bmatrix}
\tag{6}
$$

とする. ただし, $\hat{d}(z_1,\ldots,z_n)=d(z_1,\ldots,z_n)-1$ であり, $\hat{d}(0,\ldots,0)=0$ である. ここで, n-D 多項式 $p(z_1,\ldots,z_n)$ が定数項を持たないとき, すなわち $p(0,\ldots,0)=0$ を満たすとき, 常に z_k, $k\in\{1,\ldots,n\}$ について次の形に分解できる.

$$
\begin{aligned}
p(z_1,\ldots,z_n)&=z_kp_1(z_1,\ldots,z_n)+p_2(z_k)\\
&\quad+p_3(z_1,\ldots,z_{k-1},z_{k+1},\ldots,z_n)
\end{aligned}
\tag{7}
$$

ただし, p_1 は定数項を持たない n-D 多項式, p_2 は z_k に関する 1-D 線形単項式, p_3 は z_k を含まない $(n-1)$-D 多項式である [4]. 以下では簡単のため, 時折 n-D 多項式の独立変数 $(z_1,\ldots,z_n)$ を省略して記述する.

$\bar{M}$ を n-D 多項式行列, $\bar{M}$ の転置を $\bar{M}^T$ とし, 本論文で用いる行列基本変形を次のように定義する [4, 5].

Table 1　変数の処理順による伝達関数 (8) の実現の次数 r の違い

変数の処理順	r	変数の処理順	r
(z_1, z_2, z_3)	9	(z_2, z_3, z_1)	6
(z_1, z_3, z_2)	11	(z_3, z_1, z_2)	11
(z_2, z_1, z_3)	7	(z_3, z_2, z_1)	8

addrow$(\bar{M}, j, i, b(\cdot))$: $\bar{M}$ の第 j 行に多項式 $b(\cdot)$ を掛け, 第 i 行に加える.

addcol$(\bar{M}, j, i, b(\cdot))$: $\bar{M}$ の第 j 列に多項式 $b(\cdot)$ を掛け, 第 i 列に加える.

さらに, $\mathbf{0}$ を適当なサイズの零ベクトルとし, 拡大操作 augment$(\bar{M})$ を次式で定義する [4].

$$\mathrm{augment}(\bar{M}) = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \bar{M} \end{bmatrix}$$

一般的な実現手順

Step 1 : x の位置と表記を変えず, M_0 を全ての成分が n-D 線形多項式となる行列 $\tilde{M}$ に変形する.

- (6) 式で表される M_0 考える. $p = \hat{d}$, $q = n$ とし, これらを z_1 について (7) 式の形に分解する. つまり $p = z_1 p_1 + p_2 + p_3$,$q = z_1 q_1 + q_2 + q_3$ とする.
- M_0 に対し, 次の操作を行う.

$$\tilde{M} = \mathrm{augment}(M_0);$$
$$\tilde{M} = \mathrm{addrow}(\tilde{M}, 1, 2, -z_1);$$
$$\tilde{M} = \mathrm{addcol}(\tilde{M}, 1, 2, p_1);$$
$$\tilde{M} = \mathrm{addcol}(\tilde{M}, 1, 3, q_1)$$
$$= \begin{bmatrix} 1 & p_1 & q_1 \\ -z_1 & 1 + p_2 + p_3 & q_2 + q_3 \\ 0 & 1 & \mathrm{x} \end{bmatrix}$$

- p_1 と q_1 についても同じ操作を行い, 行列 $\tilde{M}$ の各多項式成分において, z_1 を含む項が全て線形単項式になるまで繰り返し行う.
- 行列 $\tilde{M}$ の各成分において, z_1 を含む項が全て線形単項式であるならば, z_2 について同様の操作を行い, z_n まで繰り返す. これらの操作によって, 全ての成分が n-D 線形多項式となる行列 $\tilde{M}$ を得る.

Step 2 : $M \triangleq \tilde{M}$ とし, M を (5) 式の形で表すことで, 行列 $\boldsymbol{A}$, $\boldsymbol{B}$, C を決定する.

しかし, この方法では行列基本変形を行う際の変数の処理順が実現次数に影響を与える. その例として, 3-D 伝達関数

$$H(z_1, z_2, z_3) = \frac{n_1 z_1 z_2^2 + n_2 z_2^2 z_3^2 + n_3 z_2^2 z_3}{1 + d_1 z_2^3 + d_2 z_1 z_2^2 z_3 + d_3 z_3^3} \quad (8)$$

に対し, 異なる全ての変数の処理順で得た実現の次数を Table 1 に示す.

4.　実現法と得られる次数の関係

はじめにこの実現法で得られる実現の次数 r について考える. この実現法の次数は, 最終的に得られる行列 M の行数から指標 x の行数 (もしくは列数) を引いた数となる. すなわち, 最終的に得られる行列 M の大きさを $q \times q$ とすると, $q - 1$ が実現の次数 r となる. また, 最初に定義する行列 M_0 は 2×2 行列であり, この実現法の処理中に複数回の拡大操作を行った結果が $q \times q$ 行列 M となることから, $q = 2 +$ (拡大操作の回数) の関係がある. したがって, 実現の次数 r は

$$r = 1 + (\text{拡大操作の回数}) \quad (9)$$

となる. よって以降では, この実現法で行う拡大操作の回数について考えていく. ただし, 変数の処理順を $z_1, \ldots, z_n$ とする.

まず多項式行ベクトル $\boldsymbol{p}(z_1, \ldots, z_n)$ によって行列 M_0 の最初の行を示す. すなわち

$$\boldsymbol{p}(z_1, \ldots, z_n) \triangleq \begin{bmatrix} d(z_1, \ldots, z_n) \\ n(z_1, \ldots, z_n) \end{bmatrix}^T \quad (10)$$

とし, さらに $l_1 = \deg_{z_1} \boldsymbol{p}(z_1, \ldots, z_n)$ とする.

ここで, $\boldsymbol{p}(z_1, \ldots, z_n)$ は, z_1 に関して

$$\boldsymbol{p}(z_1, \ldots, z_n) = \sum_{k=0}^{l_1} \boldsymbol{p}_k(z_2, \ldots, z_n) z_1^k \quad (11)$$

と表現できる. ただし, $\boldsymbol{p}_k(z_2, \ldots, z_n)$ は, z_1^k の係数ベクトルであり, $k = 0, 1, \ldots, l_1$ である.

最初に 1 番目の変数 z_1 を含む項を全て線形単項式にするために行う拡大操作の回数 t_1 を考える.(11) 式より, z_1 に関する次数を 1 にするために行う拡大操作は, $\boldsymbol{p}_{l_1}(z_2, \ldots, z_n)$ が全次数が 1 以上となる $(n-1)$-D 多項式ベクトルならば, l_1 回行う必要があり, $\boldsymbol{p}_{l_1}(z_2, \ldots, z_n)$ が全次数が 0 となる定数ベクトルならば, $l_1 - 1$ 回行う必要があるといえる. さらに正確に言えば, z_1 についての処理を続けて行うことで, 最終的に

$$\bar{M} = \begin{bmatrix} \bar{M}_{\bar{l}_1}^T & \bar{M}_{\bar{l}_1 - 1}^T & \ldots & \bar{M}_0^T & \bar{M}_{\mathrm{x}}^T \end{bmatrix}^T \quad (12)$$

の形の行列 $\bar{M}$ を得る. ただし,

$$\bar{l}_1 = \begin{cases} l_1 - 1, & \boldsymbol{p}_{l_1} \text{ の全次数が } 0 \\ l_1, & \boldsymbol{p}_{l_1} \text{ の全次数が } 1 \text{ 以上} \end{cases}$$

であり, $\bar{M}_{\mathrm{x}}$ は指標 x が含まれる行ベクトルである. したがって, z_1 の処理で行う拡大操作の回数 t_1 は $t_1 = \bar{l}_1$ であると結論付けられる.

次に 2 番目の変数 z_2 を含む項を全て線形単項式にするために行う拡大操作の回数 t_2 を考える．多項式ベクトル $\bar{\boldsymbol{p}}_j(z_2,\ldots,z_n)$ を，

$$\begin{aligned}&\bar{\boldsymbol{p}}_j(z_2,\ldots,z_n)\\&\quad=\boldsymbol{p}_j(z_2,\ldots,z_n)-\boldsymbol{p}_j(0,\ldots,0)\end{aligned}\tag{13}$$

とする．ただし，$j=0,1,\ldots,\bar{l}_1$ であり，$\bar{\boldsymbol{p}}_j(z_2,\ldots,z_n)$ の $(1,1)$ 成分は，対応する行ベクトル $\bar{M}_j$ の右側から 2 列目に，$\bar{\boldsymbol{p}}_j(z_2,\ldots,z_n)$ の $(1,2)$ 成分は，対応する行ベクトル $\bar{M}_j$ の右側から 1 列目の成分の一部として含まれる．これは，実現法における行列基本変形の手順から確認することができる．また，(13) 式は，$\boldsymbol{p}_j(z_1,\ldots,z_n)$ から定数項を除いているが，本質的には実現操作を行う際に z_1 についての線形単項式を行ベクトル $\bar{M}_j$ に残すことを意味している．

ここで，$\boldsymbol{l}_2\in\mathbb{R}^{1\times(\bar{l}_1+1)}$ とし，$\bar{M}_j$ に対して行列基本変形による実現法を適用すると考える．ただし，$\bar{M}_j$ において，右側から 1 列目，および 2 列目の成分は $z_2,\ldots,z_n$ に関する $(n-1)$-D 多項式であり，それ以外の成分は $1,\ 0,\ -z_1$ のいずれかであるため，右側の 2 列以外の列に対して行列基本変形による操作を行う必要はない．そのため，$\bar{\boldsymbol{p}}_j(z_2,\ldots,z_n)$ に対して行列基本変形による実現法を適用すると考える．すると，$\bar{\boldsymbol{p}}_j(z_2,\ldots,z_n)$ において，z_2 を含む項を全て線形単項式にするために行う拡大操作の回数 $\boldsymbol{l}_2(j)$ は t_1 の評価と同様の方法で行うことができるため，

$$\boldsymbol{l}_2(j)=\begin{cases}l_{2,j}-1, & \bar{p}_{\bar{l}_1}\ \text{の全次数が}\ 0\\ l_{2,j}, & \bar{p}_{\bar{l}_1}\ \text{の全次数が}\ 1\ \text{以上}\\ 0, & l_{2,j}=0\end{cases}$$

となる．ただし，$l_{2,j}=\deg_{z_2}\bar{p}_j,\ j=0,1,\ldots,\bar{l}_1$ であり，$l_{2,j}=0$ の場合は，$\bar{p}_j$ に z_2 が含まれていないことを意味し，拡大操作を行う必要がないため，$\boldsymbol{l}_2(j)=0$ となる．したがって,2 番目の変数 z_2 の処理で行う拡大操作の回数は，各行ベクトルで行う拡大操作の回数の総数となる．すなわち

$$t_2=\sum_{j=0}^{\bar{l}_1}\boldsymbol{l}_2(j)\tag{14}$$

となる．残っている全ての変数の処理で行う拡大操作の回数 $t_i,\ i=3,\ldots,n$ についても，同様の方法で続けて評価することができる．よって，実現の次数 r は (9) 式より次式で与えられる．

$$r=1+\sum_{i=1}^{n}t_i\tag{15}$$

ここで，$z_i,\ i\in\{2,\ldots,n\}$ における拡大操作の回数は $\bar{\boldsymbol{p}}_j$ に含まれる z_i に関する次数に関係しているが，(13) 式より $\bar{\boldsymbol{p}}_j$ は $\boldsymbol{p}_j(z_1,\ldots,z_n)$ から定数項を除いた行ベクトルであるため，$\bar{\boldsymbol{p}}_j$ と $\boldsymbol{p}_j(z_1,\ldots,z_n)$ の z_i に関する次数が等しくなる．すなわち，$\boldsymbol{p}_j$ に対して行列基本変形による実現法を適用すると考えて拡大操作の回数を評価を行っても結果は等しくなると言える．

5.　実現次数の評価法

causal である n-D 伝達関数 $H(z_1,\ldots,z_n)$ に対し，行列基本変形による F-M 状態空間モデル実現法によって得られる実現の次数を評価するアルゴリズムを次のように与えることができる．ただし，変数の処理順を $z_1,\ldots,z_n$ とする．

実現次数の評価法の手順

Step 1 : 多項式ベクトル $\boldsymbol{p}(z_1,\ldots,z_n)$ を次式で定義する．

$$\boldsymbol{p}(z_1,\ldots,z_n)\triangleq\begin{bmatrix}d(z_1,\ldots,z_n)\\ n(z_1,\ldots,z_n)\end{bmatrix}^T\tag{16}$$

また，$V=[\boldsymbol{p}(z_1,\ldots,z_n)]$，$i=0$ とし，拡大操作の回数を表すベクトルを $\boldsymbol{t}=\boldsymbol{0}\in\mathbb{R}^{1\times n}$ とする．

Step 2 : $i=i+1$ とし，$i>n$ ならば，Step.4 へ進む．$i\leqq n$ ならば，以下の操作を行う:

- 行列 V の行数を s で表す．
- 各行に対し，$l_j=\deg_{z_i}V(j),\ j=1,\ldots,s$ を求める．ただし，$V(j)$ は，V の j 行目を表す．
- $V(j)$ を次の形で表す．

$$V(j)=\sum_{k=0}^{l_j}\boldsymbol{p}_{j,k}(z_{i+1},\ldots,z_n)z_i^k\tag{17}$$

　ただし，$\boldsymbol{p}_{j,k}(z_{i+1},\ldots,z_n)$ は j 行目の z_i^k の係数ベクトルを表す．

- 行列 $\hat{V}_j$ を j 行目の係数ベクトルを用いて

$$\hat{V}_j=\begin{bmatrix}\boldsymbol{p}_{j,0}^T & \ldots & \boldsymbol{p}_{j,l_j}^T\end{bmatrix}^T\tag{18}$$

　とする．ただし，$\deg\,\boldsymbol{p}_{j,h}=0,\ h\in\{0,1,\ldots,l_j\}$ ならば，$\boldsymbol{p}_{j,h}$ を $\hat{V}_j$ に含めない．また，$l_j\neq 0$ かつ $\deg\,\boldsymbol{p}_{j,l_j}=0$ ならば，$l_j=l_j-1$ とする．

- 変数 z_i の処理で行う拡大操作の回数 $\boldsymbol{t}(i)$ を

$$\boldsymbol{t}(i)=\sum_{j=1}^{s}l_j\tag{19}$$

とする.

Step 3 : V を次式で再定義し, Step 2 へ戻る.

$$V = \left[\hat{V}_1^T \dots \hat{V}_s^T \right]^T \tag{20}$$

Step 4 : 評価次数 r を次式で求める.

$$r = 1 + \sum_{i=1}^{n} \boldsymbol{t}(i) \tag{21}$$

6. 評価法の有効性の検証

6.1 数値例

(8) 式で与えられる 3-D 伝達関数を考える. この伝達関数に対し, 実現の次数を提案した評価法によって求める. ただし, 変数の処理順を z_1, z_2, z_3 とする.

Step 1: $\boldsymbol{p}(z_1, \dots, z_n)$ を (10) 式の形で定義することで, V は

$$V = \begin{bmatrix} 1 + d_1 z_2^3 + d_2 z_1 z_2^2 z_3 + d_3 z_3^3 \\ n_1 z_1 z_2^2 + n_2 z_2^2 z_3^2 + n_3 z_2^2 z_3 \end{bmatrix}^T$$

となる. また, $\boldsymbol{t} = [0 \quad 0 \quad 0]$, $i = 0$ とする.

Step 2 : $i = i + 1 = 1$ より各操作を行う.

- 行列 V は 1 行であるため, $s = 1$.
- V の 1 行目に対して, $l_1 = \deg_{z_1} V(1) = 1$.
- V の 1 行目は次式で表すことができる.

$$\begin{aligned} V(1) &= \begin{bmatrix} 1 + d_1 z_2^3 + d_3 z_3^3 \\ n_2 z_2^2 z_3^2 + n_3 z_2^2 z_3 \end{bmatrix}^T z_1^0 + \begin{bmatrix} d_2 z_2^2 z_3 \\ n_1 z_2^2 \end{bmatrix}^T z_1^1 \\ &\triangleq \boldsymbol{p}_{1,0} z_1^0 + \boldsymbol{p}_{1,1} z_1^1 \end{aligned}$$

- 行列 $\hat{V}_1$ は次式となる.

$$\hat{V}_1 = \begin{bmatrix} 1 + d_1 z_2^3 + d_3 z_3^3 & d_2 z_2^2 z_3 \\ n_2 z_2^2 z_3^2 + n_3 z_2^2 z_3 & n_1 z_2^2 \end{bmatrix}^T$$

- z_1 の処理で行う拡大操作の回数 $\boldsymbol{t}(1)$ は, $\boldsymbol{t}(1) = 1$ となる.

Step 3 : $V = \hat{V}$ とし, Step 2 へ戻る.

$i = 2, 3$ のとき, 同様の操作を行うことで $t(2) = 4$, $t(3) = 3$ を得る.

Step 4 : 評価した実現の次数 r は次のようになる.

$$r = 1 + \sum_{i=1}^{n} \boldsymbol{t}(i) = 1 + 1 + 4 + 3 = 9$$

この結果は, Table 1 より, (8) 式の伝達関数に対し, z_1, z_2, z_3 の処理順で得られる実現の次数と等しい.

Table 2 実現法と評価法の実行時間の比較

	実行時間 [s]		次数
変数の処理順	実現法	評価法	r
(z_1, z_2, z_3)	0.710	0.089	9
(z_1, z_3, z_2)	0.972	0.117	11
(z_2, z_1, z_3)	0.488	0.081	7
(z_2, z_3, z_1)	0.421	0.075	6
(z_3, z_1, z_2)	0.975	0.127	11
(z_3, z_2, z_1)	0.658	0.091	8
合計	4.224	0.580	–

6.2 プログラムによる実行時間の測定

行列基本変形による F-M 状態空間モデル実現法とその実現法で得られる実現の次数の評価法のアルゴリズムに基づき, それぞれプログラムの開発を行った. プログラミングは, MathWorks 社の MATLAB とそのツールボックスの一つである Symbolic Math Toolbox を用いて行った. 開発を行ったプログラムを用いて, (8) 式の 3-D 伝達関数に対し, 異なる全ての変数の処理順で実行したときの実行時間をそれぞれ 10 回ずつ測定し, その平均を求めた. その結果を Table 2 に示す.

Table 2 から, 評価法の実行時間は実現法の実行時間の 15%程度であることがわかる. さらに, 全ての変数の処理順で評価法を行い, 次数が最小となる変数の処理順で実現法を行ったとしても, 実現法のみの場合の実行時間の 25%程度で次数が最小となる実現を求めることができることが分かる. したがって, 評価の結果に基づき, 次数が最小となる変数の処理順で実現法を行うことで, 計算量の多い実現法を複数回行うことなく, 効率良くこの実現で得られる実現のなかで次数が最小となる実現を得ることができるといえる.

7. まとめ

行列基本変形による F-M 状態空間モデル実現法のより一般的な実現手順を示した. また, 提示した実現手順で生じる実現の次数の変化に対し, 実際に実現操作を行わず, 効率よく最小次数の実現に対応する独立変数の順番を見つけるため, 実現の次数を評価するアルゴリズムを示した. また, 数値例により, その有効性を示した.

参考文献

1) L. Xu, H. Fan, Z. Lin and N. K. Bose: A direct-construction approach to multidimensional realization and LFR uncertainty mod-

eling, Multidimensional Systems and Signal Processing, **19**-3-4, 323/359, (2008).

2) L. Xu, H. Fan, Z. Lin and Y. Xiao: Coefficient-dependent direct-construction approach to realization of multidimensional systems in Roesser model, Multidimensional Systems and Signal Processing, **22**-1-3, 97/129, (2011).

3) H. Cheng, T. Saito, S. Matushita and L. Xu: Realization of multidimensional systems in Fornasini-Marchesini state-space model, Multidimensional Systems and Signal Processing, **22**-4, 319/333, (2011).

4) L. Xu and S. Yan: A New Elementary Operation Approach to Multidimensional Realization and LFR Uncertainty Modeling: The SISO Case, Multidimensional Systems and Signal Processing, **21**-4, 323/372, (2010).

5) L. Xu, S. Yan, Z. Lin and S. Matsushita: A New Elementary Operation Approach to Multidimensional Realization and LFR Uncertainty Modeling: the MIMO Case, IEEE Trans. Circuits and Systems I, **59**-3, 638/651, (2012).

6) L. Xu, L. Wu, Q. Wu, Z. Lin and Y. Xiao: On Realization of 2D Discrete Systems by Fornasini-Marchesini Model, International Journal of Control, Automation and Systems, **3**-4, 631/639, (2005).

7) L. Xu, Q. Wu, Z. Lin and Y. Xiao: A New Constructive Procedure for 2-D Coprime Realization in Fornasini-Marchesini Model, IEEE Trans. Circuits and Systems I, **54**-9, 2061/2069, (2007).

8) 王, 松下, 徐: 行列基本変形による多次元 Fornasini-Marchesini 状態空間モデルの実現, 第 54 回自動制御連合講演会, 豊橋, (2011)