計測自動制御学会東北支部 第 274 回研究集会 (2012.7.13)
資料番号 274-2

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：多入出力の場合

Order Evaluation of the Elementary Operation Approach to Multidimensional Fornasini-Marchesini State-space Model Realization: the MIMO 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, 多くの分野に応用できるため注目されている．しかし, n-D ($n \geqq 2$) システムは従来の 1 次元 (1-D) システムの場合と全く異なる数学的性質を持つため, その最小実現を求めることが極めて困難であり, 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 [1, 2, 3, 4, 5, 6, 7, 8, 9].

最近, 本研究グループでは行列基本変形による n-D Fornasini-Marchesini (F-M) 状態空間モデル実現法を提案した [10]．また，この方法では行列基本変形の手順・独立変数の処理順によって実現の次数が異なる問題を明らかにすると共に, 1 入出力 (SISO) の場合に対し実際の実現操作を行わずにこの実現手順による実現次数の評価アルゴリズムを示した．この結果によって，効率的により低い次数の実現を得ることができる処理順を見つけられるようになった [11]. しかし，これらの成果をより複雑な多入出力 (MIMO) 場合へ拡張することが課題として残されている.

本論文では, MIMO システムの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し, この実現手順で得られる実現の次数を評価するアルゴリズムを提案し, 数値例でその有効性を示す.

2. F-M モデルの実現問題

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

$$
\begin{aligned}
&\boldsymbol{x}(i_1+1, ..., i_n+1)\\
&= A_1\boldsymbol{x}(i_1, i_2+1, \dots, i_n+1) + \dots + A_n\boldsymbol{x}(i_1+1, \dots, i_{n-1}+1, i_n)\\
&\quad + B_1\boldsymbol{u}(i_1, i_2+1, \dots, i_n+1) + \dots + B_n\boldsymbol{u}(i_1+1, \dots, i_{n-1}+1, i_n)\\
&\boldsymbol{y}(i_1, ..., i_n) = C\boldsymbol{x}(i_1, ..., i_n) + D\boldsymbol{u}(i_1, ..., i_n) \qquad (1)
\end{aligned}
$$

ただし，$\boldsymbol{x}(i_1, ..., i_n)$ は状態ベクトル, $\boldsymbol{u}(i_1, ..., i_n)$, $\boldsymbol{y}(i_1, ..., i_n)$ はそれぞれ入出力ベクトル, $A_1, ..., A_n$, $B_1, ..., B_n$, C, D は適切な大きさの実数行列である．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，$\boldsymbol{A} = (A_1, ..., A_n)$, $\boldsymbol{B} = (B_1, ..., B_n)$

である.

システム (1) の伝達行列は次式で与えられる.

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

ただし, $z_1,\ldots,z_n$ は遅れ作用素である. また, r を F-M モデルの次数という.

全次数が 1 となる単項式, 多項式をそれぞれ 1-D 線形単項式, n-D 線形多項式と呼ぶ [11]. また, n-D 多項式 $p(z_1,...,z_n)$ は定数項を持たないとき, 常に z_k, $k\in\{1,...,n\}$ に関して次の形に分解できる.

$$\begin{aligned}p(z_1,\ldots,z_n)&=z_kp_1(z_1,\ldots,z_n)+p_2(z_k)\\&+p_3(z_1,\ldots,z_{k-1},z_{k+1},\ldots,z_n)\end{aligned} \tag{3}$$

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 [6]. 以下では簡単のため, 時折独立変数 $(z_1,\ldots,z_n)$ を省略, または (z) と記述する.

さらに, n-D 多項式ベクトル $\boldsymbol{p}(z)$ は, n-D 多項式を成分に持つベクトルで, $\boldsymbol{p}(z)$ に含まれる n-D 多項式の全次数と z_k に関する次数の最大値をそれぞれ $\boldsymbol{p}(z)$ の全次数と z_k に関する次数と呼び, $\deg \boldsymbol{p}(z)$, $\deg_{z_k}\boldsymbol{p}(z)$ で表す [6, 7].

$a(z)$ と $b(z)$ を n-D 多項式とする n-D 有理関数 $g(z)=a(z)/b(z)$ に対し, $b(0,\ldots,0)\neq 0$ であるならば, $g(z)$ は causal であるという. また, n-D 有理関数行列 $G(z)$ の全ての成分が causal であるならば, $G(z)$ は causal であるという [3].

ここで, $\bar{M}$ を n-D 多項式行列とし, 本論文で用いる行列基本変形を次のように定義する [7].

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, $\mathbf{0}$ を適切なサイズの零行列として拡大操作 augment$(\bar{M})$ を次式で定義する [6].

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

与えられた n-D 伝達行列 $H(z)$ に対し, (2) 式を満たす $(\boldsymbol{A}, \boldsymbol{B}, \mathrm{C}, \mathrm{D})$ を求める問題を, F-M モデルの実現問題という.

3. 行列基本変形による MIMO F-M 状態空間モデル実現法

次の左行列分解表現で与えられる causal な $p\times q$ の n-D 伝達行列について考える.

$$H(z)=D_l(z)^{-1}N_l(z) \tag{4}$$

ただし, $N_l(z)$, $D_l(z)$ はそれぞれ $p\times q$, $p\times p$ の n-D 多項式行列である. さらに, 一般性を失わず $N_l(0,\ldots,0)=\mathbf{0}$, $D_l(0,\ldots,0)=I_p$ とする. このとき $D=\mathbf{0}$ となり $\boldsymbol{A}$, $\boldsymbol{B}$, C を求めればよい [10].

まず次の形の行列 M_0 を定義する.

$$M_0\triangleq\begin{bmatrix}D_l(z)&N_l(z)\\I_p&\mathrm{X}\end{bmatrix} \tag{5}$$

ただし, X は その成分が含まれる行と他の行の入れ替え (列も同様), あるいは X の構造や成分の表記を変えるような行列基本変形を禁止するための $p\times q$ の指標行列である [10]. この M_0 に対して行列基本変形と拡大操作を行うことで,

$$M=\begin{bmatrix}I_r-\sum_{i=1}^{n}A_iz_i&\sum_{i=1}^{n}B_iz_i\\C&\mathrm{X}\end{bmatrix} \tag{6}$$

の形の行列 M に変形でき, このときの $\boldsymbol{A}$, $\boldsymbol{B}$, C と $D=\mathbf{0}$ は $H(z)$ の実現となる [10].

ここで, SISO の場合と比べ, 行列 M_0 の構造がより複雑であるため, 低次数の実現を得るために一般的な実現手順を考えると, 次に示す 2 つの操作が考えられる.

(a) 行に関する操作

この方法は, (5) 式で定義した M_0 に対し,

$$\begin{aligned}&\tilde{M}=\mathrm{augment}(M_0);\\&\tilde{M}=\mathrm{addrow}(\tilde{M},1,j,-z_k);\\&\tilde{M}=\mathrm{addcol}(\tilde{M},1,l,p_{l,1}));\\&\tilde{M}=\mathrm{addcol}(\tilde{M},1,m,p_{m,1});\end{aligned}$$

と同様な操作を繰り返し行うものである. ただし, $p_{l,1}$, $p_{m,1}$ は $\tilde{M}$ の (j,l) 成分ないし (j,m) 成分を z_k, $k\in\{1,\ldots,n\}$ に関して多項式分解したときの n-D 多項式, すなわち (3) 式における p_1 に対応する多項式である.

(b) 列に関する操作

この方法は, (5) 式で定義した M_0 に対し

$$\begin{aligned}&\tilde{M}=\mathrm{augment}(M_0);\\&\tilde{M}=\mathrm{addcol}(\tilde{M},1,j,-z_k);\\&\tilde{M}=\mathrm{addrow}(\tilde{M},1,l,\tilde{p}_{l,1}));\\&\tilde{M}=\mathrm{addrow}(\tilde{M},1,m,\tilde{p}_{m,1});\end{aligned}$$

と同様な操作を繰り返し行うものである. ただし,

Table 1　変数の処理順による伝達行列 (7) の実現の次数 r の違い

	実現の次数 r	
変数の処理順	行に関する操作	列に関する操作
(z_1, z_2, z_3)	11	10
(z_1, z_3, z_2)	12	10
(z_2, z_1, z_3)	7	7
(z_2, z_3, z_1)	7	7
(z_3, z_1, z_2)	11	10
(z_3, z_2, z_1)	9	10

$\tilde{p}_{l,1}$, $\tilde{p}_{m,1}$ は $\tilde{M}$ の (l,j) 成分ないし (m,j) 成分を z_k に関して多項式分解したときの n-D 多項式である.

この二つの操作において, M_0 を定義したとき, 非線形な項が含まれる列数が多い場合は行に関する操作を, 行数が多い場合は列に関する操作を用いることで拡大操作の回数を減らすことができ有効的となる.

以上のことから一般的な実現手順を与えられる.

一般的な実現手順

Step 1 : X の構造と表記を変えず, 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.

- (5) 式で表される M_0 の各成分において z_1 について多項式分解をする.
- M_0 に対し, 行あるいは列に関する操作を行い, 行列 $\tilde{M}$ の成分において, z_1 を含む項が全て 1-D 線形単項式になるまで繰り返し行う.
- $\tilde{M}$ の各成分において, z_1 を含む項が全て 1-D 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 (6) 式の形で表すことで, 行列 $\boldsymbol{A}$, $\boldsymbol{B}$, C を決定する.

しかし, どちらの操作を用いたとしても, 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 伝達行列

$$
\begin{aligned}
H(z_1, z_2, z_3) &= \begin{bmatrix} \dfrac{n_1 z_1 z_2^2 + n_2 z_2^2 z_3}{1 + d_1 z_2 + d_2 z_3} & \dfrac{n_3 z_2^2}{1 + d_1 z_2 + d_2 z_3} \\ \dfrac{n_4 z_2^2 z_3}{1 + d_3 z_1 + d_4 z_3} & \dfrac{n_5 z_1 z_2^2 z_3}{1 + d_3 z_1 + d_4 z_3} \end{bmatrix} \\
&= \begin{bmatrix} 1 + d_1 z_2 + d_2 z_3 & 0 \\ 0 & 1 + d_3 z_1 + d_4 z_3 \end{bmatrix}^{-1} \\
&\quad \cdot \begin{bmatrix} n_1 z_1 z_2^2 + n_2 z_2^2 z_3 & n_3 z_2^2 \\ n_4 z_2^2 z_3 & n_5 z_1 z_2^2 z_3 \end{bmatrix} \qquad (7)
\end{aligned}
$$

に対し, 行あるいは列に関する操作を用いて全ての変数の処理順で得た実現の次数を Table 1 に示す.

この例において, 行列 M_0 を定義すると (7) 式より, 非線形な項が含まれる行数と列数が等しいことがわかる. このような場合, 一般的にどちらの方法を用いれば低次数の実現が得られるかを予測することが困難である. さらに, 非線形な項を含まれる行数と列数の関係から用いる操作を決定しても, 異なる他の操作のほうが低次数の実現を得る場合がある. そのため, 低次数の実現を得るためには, 行に関する操作を用いた実現法と列に関する操作を用いた実現法をそれぞれ全ての変数の処理順で行う必要があり, 計算量が膨大となる. そこで, 以降ではこれらの実現操作を実際に実行せずにそれぞれの操作を用いた実現法で得られる実現の次数を評価するアルゴリズムを検討する.

4.　実現次数の評価

左行列分解表現で表される causal な $p \times q$ の n-D 伝達行列 $H(z_1, \ldots, z_n)$ を考える. 示した実現法で得られる実現の次数 r は拡大操作の回数を用いて

$$r = p + (\text{ 拡大操作の回数}) \qquad (8)$$

と表すことができるため, 各操作法を用いた場合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.

4.1　行に関する操作法を用いた場合

伝達行列 $H(z_1, \ldots, z_n)$ に対応する行列 M_0 は

$$
\begin{aligned}
M_0 &= \begin{bmatrix} D_l(z_1, \ldots, z_n) & N_l(z_1, \ldots, z_n) \\ I_p & \mathrm{X} \end{bmatrix} \\
&\triangleq \begin{bmatrix} \boldsymbol{p}_1^T & \cdots & \boldsymbol{p}_p^T & P_X^T \end{bmatrix}^T \qquad (9)
\end{aligned}
$$

と表すことができる. ただし, $\boldsymbol{p}_j(z_1, \ldots, z_n)$, $j = 1, \ldots, p$ は j 行目の多項式行ベクトルであり,

$$P_X = \begin{bmatrix} I_p & \mathrm{X} \end{bmatrix} \qquad (10)$$

である.

初めに 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総数 t_1 を考える. 多項式行ベクトル $\boldsymbol{p}_j(z_1, \ldots, z_n)$ は z_1 に関して,

$$\boldsymbol{p}_j(z_1, \ldots, z_n) = \sum_{k=0}^{l_{j,1}} \boldsymbol{p}_{j,k}(z_2, \ldots, z_n) z_1^k \qquad (11)$$

と表すことができる. ただし, $\boldsymbol{p}_{j,k}(z_2, \ldots, z_n)$ は j 行目の z_1^k の係数ベクトルで, $l_{j,1} = \deg_{z_1} \boldsymbol{p}_j(z_1, \ldots, z_n)$ である. (11) 式より, $\boldsymbol{p}_j(z_1, \ldots, z_n)$ に対して z_1 に関する次数を 1 にするために行う拡大操作の回数を $t_{j,1}$ とすると, $\boldsymbol{p}_{j,l_{j,1}}(z_2, \ldots, z_n)$ が全次数が 1 以上となる $(n-1)$-D 多項式ベクトルならば, $l_{j,1}$ 回行う必

要があり, $\boldsymbol{p}_{j,l_{j,1}}(z_2,\dots,z_n)$ が全次数が 0 となる定数ベクトルならば, $l_{j,1}-1$ 回行う必要があるといえる. すなわち,

$$t_{j,1}=\begin{cases} l_{j,1}, & \boldsymbol{p}_{j,l_{j,1}} \text{ の全次数が 1 以上} \\ l_{j,1}-1, & \boldsymbol{p}_{j,l_{j,1}} \text{ の全次数が 0} \end{cases}$$

となる. また, これらの多項式行ベクトルは, SISO の場合と異なり [11] p 行あることから, 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総数 t_1 は次式で与えられる.

$$t_1=\sum_{k=1}^{p} t_{k,1} \tag{12}$$

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. z_1 の処理を全て行ったときの行列 $\bar{M}$ の構造を考えると,

$$\hat{M}=\begin{bmatrix}\hat{M}_{t_1}^T & \dots & \hat{M}_1^T & M_1^T & \dots & M_p^T & M_{\mathrm{X}}^T\end{bmatrix}^T \tag{13}$$

の形となる. ただし, $\hat{M}_1,\dots,\hat{M}_{t_1}$ は拡大操作によって新たに生成された行を表し, $M_1,\dots,M_p$ は (9) 式における $\boldsymbol{p}_1,\dots,\boldsymbol{p}_p$ が含まれていた行, すなわち M_0 を定義した時の伝達行列に関する成分が含まれていた行を表す. また, M_{X} は P_{X} が含まれていた行を表す. ここで SISO の場合を考える. SISO の場合では拡大操作の回数を減らすことができるため, 今着目している操作と同じ行に関する処理を用いて実現操作を行う. また, z_1 に関する処理をすべて行うと (13) 式と同様の構造, すなわち拡大操作によって新たに生成された行, 伝達関数に関する成分が含まれていた行, 指標が含まれていた行となる [11]. すなわち, $z_2,\dots,z_n$ の処理で行う拡大操作の回数は, SISO の場合より複雑になるが, 同じ考え方で評価を行うことができるといえる.

これらのことから次の実現の次数を評価するアルゴリズムを与えることができる.

行に関する操作を用いた場合の実現の次数の評価の手順

Step 1 : 左行列分解表現で表される $p\times q$ の伝達行列 $H(z_1,\dots,z_n)$ より, 行列 V を次式で定義する.

$$V\triangleq\begin{bmatrix} D_l(z_1,\dots,z_n) & N_l(z_1,\dots,z_n)\end{bmatrix} \tag{14}$$

また, $i=0$ 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, $l_j=\deg_{z_i}V(j)$, $j=1,\dots,s$ 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.

$$V(j)=\sum_{k=0}^{l_j}\boldsymbol{p}_{j,k}(z_{i+1},\dots,z_n)z_i^k \tag{15}$$

  ただし, $\boldsymbol{p}_{j,k}(z_{i+1},\dots,z_n)$ 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

$$\hat{V}_j=\begin{bmatrix}\boldsymbol{p}_{j,0}^T & \dots & \boldsymbol{p}_{j,l_j}^T\end{bmatrix}^T \tag{16}$$

  とする. ただし, $\deg\,\boldsymbol{p}_{j,h}=0$, $h\in\{0,1,\dots,l_j\}$ 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.
- z_i の処理での拡大操作の回数 $\boldsymbol{t}(i)$ を次式で与える.

$$\boldsymbol{t}(i)=\sum_{j=1}^{s} l_j \tag{17}$$

Step 3 : V を次式で再定義し, Step 2 へ戻る.

$$V=\begin{bmatrix}\hat{V}_1^T\dots\hat{V}_s^T\end{bmatrix}^T \tag{18}$$

Step 4 : 評価次数 r を次式で求める.

$$r=p+\sum_{i=1}^{n}\boldsymbol{t}(i) \tag{19}$$

4.2 列に関する操作を用いた場合

列に関する操作を用いた場合の実現操作は, 行列 M_0 を転置した行列

$$\tilde{M}_0\triangleq M_0^T=\begin{bmatrix} D_l(z_1,\dots,z_n)^T & I_p \\ N_l(z_1,\dots,z_n)^T & \mathrm{X}^T\end{bmatrix} \tag{20}$$

に対して行に関する操作を用いた実現操作をすることと等価である. また, この $\tilde{M}_0$ に対して行に関する操作を用いて実現を求めたとしても, X の成分を含む列の成分を変えるような行列基本変形を行わないため, 先ほど述べた行に関する操作を用いた場合の実現次数の評価の場合と同様に X の成分が含まれる列ベクトルを考える必要はない. すなわち, 先に示した行に関する操作を用いた場合の実現の次数の評価の手順において, Step 1 で定義する行列 V を

$$V\triangleq\begin{bmatrix} D_l(z_1,\dots,z_n)^T \\ N_l(z_1,\dots,z_n)^T\end{bmatrix} \tag{21}$$

とし, 他の操作は同様に行うことで列に関する操作を用いた場合の実現の次数の評価を行うことができる.

列に関する操作を用いた場合の実現の次数の評価の手順

Step 1 : 左行列分解表現で表される $p \times q$ の伝達行列 $H(z_1, \ldots, z_n)$ より, 行列 V を次式で定義する.

$$V \triangleq \begin{bmatrix} D_l(z_1, \ldots, z_n)^T \\ N_l(z_1, \ldots, z_n)^T \end{bmatrix} \tag{22}$$

また, $i = 0$ 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 ∼ Step 4 : 行に関する操作を用いた場合の実現の次数の評価と同じ手順を行う.

5. 評価法の有効性の検証

5.1 数値例

(7) 式で与えられる 2×2 の 3-D 伝達行列を考える. この伝達行列に対し, 先に示した実現手順で得られる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.

行に関する操作を用いた場合の実現の次数の評価

Step 1: 与えられた伝達行列 $H(z_1, z_2, z_3)$ をから行列 V を (14) 式の形で定義すると

$$V = \begin{bmatrix} 1 + d_1 z_2 + d_2 z_3 & 0 & n_1 z_1 z_2^2 + n_2 z_2^2 z_3 & n_3 z_2^2 \\ 0 & 1 + d_3 z_1 + d_4 z_3 & n_4 z_2^2 z_3 & n_5 z_1 z_2^2 z_3 \end{bmatrix}$$

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 2 行であるため, $s = 2$.
- V の各行から, $l_1 = 1$, $l_2 = 1$.
- V の各行を (15) 式の形で表すことで行列 $\hat{V}_1$ $\hat{V}_2$ は次のようになる.

$$\hat{V}_1 = \begin{bmatrix} 1 + d_1 z_2 + d_2 z_3 & 0 & n_2 z_2^2 z_3 & n_3 z_2^2 \\ 0 & 0 & n_1 z_2^2 & 0 \end{bmatrix}$$

$$\hat{V}_2 = \begin{bmatrix} 0 & 1 + d_4 z_3 & n_4 z_2^2 z_3 & 0 \\ 0 & d_3 & 0 & n_5 z_2^2 z_3 \end{bmatrix}$$

- z_1 の処理で行う拡大操作の回数 $\boldsymbol{t}(1)$ は, $\boldsymbol{t}(1) = 1 + 1 = 2$ となる.

Step 3 :

$$V = \begin{bmatrix} \hat{V}_1^T & \hat{V}_2^T \end{bmatrix}^T$$

とし, Step 2 へ戻る.

$i = 2, 3$ のとき, 同様の操作を行うことで $\boldsymbol{t}(2) = 7$, $\boldsymbol{t}(3) = 0$ を得る.

Step 4 : 評価した実現の次数 r は次のようになる.

$$r = p + \sum_{i=1}^{n} \boldsymbol{t}(i) = 2 + 2 + 7 + 0 = 11$$

列に関する操作を用いた場合の実現の次数の評価

Step 1: 与えられた伝達行列 $H(z_1, z_2, z_3)$ をから行列 V を (22) 式の形で定義すると

$$V = \begin{bmatrix} 1 + d_1 z_2 + d_2 z_3 & 0 \\ 0 & 1 + d_3 z_1 + d_4 z_3 \\ n_1 z_1 z_2^2 + n_2 z_2^2 z_3 & n_4 z_2^2 z_3 \\ n_3 z_2^2 & n_5 z_1 z_2^2 z_3 \end{bmatrix}$$

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 4 行であるため, $s = 4$.
- V の各行から, $l_1 = 0$, $l_2 = 1$, $l_3 = 1$, $l_2 = 1$.
- V の各行を (15) 式の形で表すことで行列 $\hat{V}_1 \sim \hat{V}_4$ は次のようになる. ただし, $\deg \boldsymbol{p}_{2,1} = 0$ より $l_2 = 1 - 1 = 0$ となる.

$$\hat{V}_1 = \begin{bmatrix} 1 + d_1 z_2 + d_2 z_3 & 0 \end{bmatrix}, \ \hat{V}_2 = \begin{bmatrix} 0 & 1 + d_4 z_3 \end{bmatrix}$$

$$\hat{V}_3 = \begin{bmatrix} n_2 z_2^2 z_3 & n_4 z_2^2 z_3 \\ n_1 z_2^2 & 0 \end{bmatrix}, \ \hat{V}_4 = \begin{bmatrix} n_3 z_2^2 & 0 \\ 0 & n_5 z_2^2 z_3 \end{bmatrix}$$

- z_1 の処理で行う拡大操作の回数 $\boldsymbol{t}(1)$ は, $\boldsymbol{t}(1) = 0 + 0 + 1 + 1 = 2$ となる.

Step 3 :

$$V = \begin{bmatrix} \hat{V}_1^T & \hat{V}_2^T & \hat{V}_3^T & \hat{V}_4^T \end{bmatrix}^T$$

とし, Step 2 へ戻る.

$i = 2, 3$ のとき, 同様の操作を行うことで $\boldsymbol{t}(2) = 6$, $\boldsymbol{t}(3) = 0$ を得る.

Step 4 : 評価した実現の次数 r は次のようになる.

$$r = p + \sum_{i=1}^{n} \boldsymbol{t}(i) = 2 + 2 + 6 + 0 = 10$$

これらの結果は, Table 1 より, (7) 式の伝達行列に対し, 各操作を用いて z_1, z_2, z_3 の処理順で得られる実現の次数と等しい.

5.2 プログラムによる実行時間の測定

各操作を用いた場合の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. 開発したプログラムを用いて, (7) 式の 3-D 伝達行列に対し, 異なる操作で全ての変数の処理順で実行したときの

Table 2　各操作を用いた実現法の実行時間

	実行時間 [s]	
変数の処理順	行に関する操作	列に関する操作
(z_1, z_2, z_3)	2.334	2.313
(z_1, z_3, z_2)	2.766	1.853
(z_2, z_1, z_3)	1.294	1.506
(z_2, z_3, z_1)	1.279	1.488
(z_3, z_1, z_2)	2.384	1.855
(z_3, z_2, z_1)	1.751	2.315
操作毎の合計時間	11.808	11.330
合計	23.138	

Table 3　各操作を用いた評価法の実行時間

	実行時間 [s]	
変数の処理順	行に関する操作	列に関する操作
(z_1, z_2, z_3)	0.504	0.298
(z_1, z_3, z_2)	0.496	0.302
(z_2, z_1, z_3)	0.460	0.251
(z_2, z_3, z_1)	0.406	0.242
(z_3, z_1, z_2)	0.487	0.299
(z_3, z_2, z_1)	0.400	0.287
操作毎の合計時間	2.753	1.679
合計	4.432	

実行時間をそれぞれ 10 回ずつ測定し, その平均を求めた. 各操作を用いた場合の実現法の実行時間の測定結果を Table 2 に, 評価法の実行時間の測定結果を Table 3 に示す.

Table 2 と Table 3 から, 行に関する操作を用いた場合, 評価法の実行時間は実現法の実行時間の 25%程度であり, 列に関する操作を用いた場合, 評価法の実行時間は実現法の実行時間の 15% 程度であることが分かる. さらに, 全ての変数の処理順で行に関する操作と列に関する操作を用いた場合の評価をそれぞれ行い, 実現の次数が最小となる操作と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, 次数が最小となる操作と変数の処理順で実現法を行うことで, 効率良く次数が最小となる実現を得ることができる.

6.　まとめ

多入出力の場合における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 2 つの実現手順を示した. 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, 数値例によりその有効性を示した.

参考文献

1) J. C. Cockburn and B. G. Morton: Linear fractional representations of uncertain systems, *Automatica*, **33**, 1263/1271, (1997).

2) K. Galkowski: State-space realizations of linear 2-D systems with extensions to the general nD ($n > 2$) Case. LNCIS: Springer, (2001).

3) L. Xu, *et al.*: A direct-construction approach to multidimensional realization and LFR uncertainty modeling, *Multidimens. Syst. Signal Process.*, **19**-3-4, 323/359, (2008).

4) L. Xu, *et al.*: Coefficient-dependent direct-construction approach to realization of multidimensional systems in Roesser model, *Multidimens. Syst. Signal Process.*, **22**-1-3, 97/129, (2011).

5) H. Cheng, *et al.*: Realization of multidimensional systems in Fornasini-Marchesini state-space model, *Multidimens. Syst. Signal Process.*, **22**-4, 319/333, (2011).

6) L. Xu and S. Yan: A new elementary operation approach to multidimensional realization and LFR uncertainty modeling: the SISO case, *Multidimens. Syst. Signal Process.*, **21**-4, 323/372, (2010).

7) L. Xu, *et al.*: 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).

8) L. Xu, *et al.*: On realization of 2D discrete systems by Fornasini-Marchesini model, *International Journal of Control, Automation and Systems*, **3**-4,631/639, (2005).

9) L. Xu, *et al.*: A new constructive procedure for 2-D coprime realization in Fornasini-Marchesini model, *IEEE Trans. Circuits and Systems I*, **54**-9, 2061/2069, (2007).

10) 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).

11) 植栗, 松下, 徐: 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, 計測自動制御学会東北支部 第 272 回研究集会, 272-10, (2012).