Aggregation 法による離散時間システムの低次元化[†]

堀 口 和 己*・浜 田 望*

Model Reduction for Linear Discrete-Time Systems via Aggregation

Kazumi HORIGUCHI* and Nozomu HAMADA*

Recently with the enlargement of plant's scale, model reduction is becoming much more important and many methods are studied in control system design.

The methods of order reduction for linear systems are divided into two groups. One treats transfer functions and the given high-order transfer function is approximately reduced to a low order one. On the other hand, the second method reduces its state space model by state aggregation. The latter method would be preferable in many practical viewpoint.

At present, many reduction techniques have mainly dealt with continuous-time systems and there are few for discrete-time systems.

Considering these situations, this thesis aims at proposing two reduction methods for linear discrete-time systems via state aggregation. We established the following methods:

i) Reduction based on discrete-Schwartz form,

ii) Reduction based on discrete-Routh form.

i) is a time domain approximation method, while ii) is a low-frequency domain one. Both of them assure the stability of reduced-order systems, and are performed by simple algorithm.

Keyword: linear discrete-time system, model reduction, state aggregation, time domain approximation, low-frequency domain approximation

1. 緒 言

近年、システム制御の分野において、プラントの大規模化は増々進む傾向にあり、これに伴ってシステム低次元化問題は一層重要性を増し、多くの研究がなされている^{1),2)}。

[†] 第25回自動制御連合講演会で発表(昭57・11)

* 慶応義塾大学理工学部 横浜市港北区日吉 3-14-1

* Faculty of Science and Technology, Keio

University, Yokohama

(Received April 2, 1983)

(Revised September 28, 1983)

線形システムの低次元化手法は、つぎの2グループに大別される。第1は伝達関数に注目し、高次伝達関数を低次伝達関数で低次元近似するものであり³⁾、第2は状態方程式に注目し、状態の集約化(Aggregation)により低次元モデルを得るものである。方法的には後者のほうが優れた面が多く(特に制御問題への応用に有効である⁴⁾)、最近では後者の立場から低次元化問題を扱う傾向も少なくない^{5),6)}。また、低次元化の対象システムとしては連続系および離散系の両者があるが、既存の手法の大部分が連続系を扱ったもので離散系の低次元化手法を与えたものは比較的少ない。

本論文はこのような背景を踏まえ、線形離散時間システムの低次元化手法を前述第2グループの立場から提案することを目的としている。本手法は本来第1グループに属するものであるゆえ、多入出力系への拡張が困難という欠点は伴うが、たとえば文献6)による可制御性・可観測性の弱い部分を除去する手法に対して、近似目的の具体性、低次元化手順の単純性といった特徴をもっている。

さて、以下に本論文の構成を述べておく。第2章では離散 Schwartz 実現に基づく低次元化手法を展開している。本手法は低次元モデルの単位パルス列を原システムのそれに近づけるという意味で時間域近似である。第3章では離散 Routh 実現に基づく低次元化手法を提案する。本手法は連続系の低次元化手法である Routh 近似法を双一次変換を用いて離散系低次元化手法に変換したもので低周波域近似である。両低次元化手法は共に、低次元システムの安定性を保証する近似法であり、低次元化手順は容易なものである。第4章では、これら低次元化手法の例題を示している。

以下に次章以下で用いる記号の説明をしておく。

対象とする高次元システム:

$$\begin{cases} x_{k+1} = Ax_k + bu_k & x_k \in R^n \\ y_k = cx_k + du_k & A: \text{安定行列} \end{cases} \quad (1)$$

$$\text{伝達関数: } H(z) \triangleq \frac{g(z)}{f(z)} = \frac{\sum_{i=0}^n b_i z^{n-i}}{\sum_{i=0}^n a_i z^{n-i}} \quad (2)$$

低次元近似システム:

$$\begin{cases} z_{k+1} = Fz_k + gu_k & z_k \in R^m \\ \bar{y}_k = hz_k + ju_k \end{cases} \quad (3)$$

$$\text{伝達関数: } \tilde{H}(z) \triangleq \frac{\tilde{g}(z)}{\tilde{f}(z)} = \frac{\sum_{i=0}^m \tilde{b}_i z^{m-i}}{\sum_{i=0}^m \tilde{a}_i z^{m-i}} \quad (4)$$

$f(z)$ の相反多項式:

$$f^*(z) \triangleq z^n f(z^{-1}) = \sum_{i=0}^n a_i z^i \quad (5)$$

行列及びベクトルの部分成分:

$$\begin{cases} A_{(m)} \triangleq A \begin{pmatrix} i=1 \cdots m \\ j=1 \cdots m \end{pmatrix} = UAU^T \\ b_{(m)} \triangleq b(j=1 \cdots m) = Ub \\ c_{(m)} \triangleq c(i=1 \cdots m) = cU^T \\ U \triangleq [I_m \ 0]: m \times n \text{ 行列} \end{cases} \quad (6)$$

2. 離散 Schwartz 実現に基づく低次元化

本章で述べる方法は、離散 Schwartz 実現に基づく時間域近似を目的とした低次元化法である。離散 Schwartz 実現は、連続系 Schwartz 実現に相当する離散時間表現であり⁷⁾、種々のすぐれた特徴をもつ Lattice 構造にて構成されることが知られている⁸⁾。

2.1 離散 Schwartz 実現

以下では、Schur の連分数展開より離散 Schwartz 実現を導き、さらに等価変換により b マトリクスの形を変えることで低次元化のための標準形を導出する。

原システムの伝達関数の分母多項式 $f(z)$ およびその相反多項式 $f^*(z)$ に対しつぎの連分数展開が可能である⁹⁾。

$$\begin{aligned} \frac{f^*(z)}{f(z)} &= \frac{\Delta_n}{1 - \frac{\Delta_n^{-1} \Delta_{n-1} (1 - \Delta_n^2) z^{-1}}{1 + \Delta_n^{-1} \Delta_{n-1} z^{-1} - \frac{\Delta_{n-1}^{-1} \Delta_{n-2} (1 - \Delta_{n-1}^2) z^{-1}}{1 + \Delta_{n-1}^{-1} \Delta_{n-2} z^{-1} - \frac{\Delta_{n-2}^{-1} \Delta_{n-3} (1 - \Delta_{n-2}^2) z^{-1}}{1 + \Delta_{n-2}^{-1} \Delta_{n-3} z^{-1} - \frac{\Delta_{n-3}^{-1} \Delta_{n-4} (1 - \Delta_{n-3}^2) z^{-1}}{1 + \Delta_{n-3}^{-1} \Delta_{n-4} z^{-1} - \cdots}}}} \\ &\quad \cdots \frac{\Delta_2^{-1} \Delta_1 (1 - \Delta_2^2) z^{-1}}{1 + \Delta_2^{-1} \Delta_1 z^{-1} - \frac{\Delta_1^{-1} \Delta_0 (1 - \Delta_1^2) z^{-1}}{1 + \Delta_1^{-1} \Delta_0 z^{-1}}} \end{aligned} \quad (7)$$

上連分数展開に基づき伝達関数 $H(z)$ を次式で表わす。

$$\begin{aligned} H(z) &= b_0 + \frac{\Delta_n z^{-1} F_n(z^{-1})}{1 - (1 - \Delta_n^2) z^{-1} F_n(z^{-1})} [\gamma_n + F_{n-1}(z^{-1}) \\ &\quad \times [\gamma_{n-1} + F_{n-2}(z^{-1}) \cdot [\cdots + F_1(z^{-1}) \gamma_1] \cdots] \end{aligned} \quad (8)$$

ただし,

$$F_i(z^{-1}) = \frac{\Delta_i^{-1} \Delta_{i-1}}{1 + \Delta_i^{-1} \Delta_{i-1} z^{-1} - \cdots} \quad (i=1 \cdots n)$$

ここで、 Δ_i の値は $f(z)$ の係数 a_i より Jury 表¹⁰⁾ を作って得られるパラメータと一致する。また γ_i の値は、 $H(z)$ の分母分子係数 (a_i, b_i) より表形式で逐次的に求められる。逆に、 (Δ_i, γ_i) が与えられたとき (a_i, b_i) を逐次導出することも可能である〈付録1〉。

さてここで、状態変数 X_i ; 入力 U ; 出力 Y を次式:

$$X_i = F_i(z^{-1}) X_{i+1} \quad (i=1 \cdots n-1) \quad (9)$$

$$\begin{cases} U = -\Delta_{n-1}^{-1} (1 - \Delta_{n-1}^2) X_{n-1} + \Delta_{n-1}^{-1} z X_n + \Delta_n X_n \\ Y = \gamma_1 X_1 + \cdots + \gamma_n X_n + b_0 U \end{cases} \quad (10)$$

と定め、 $H(z)$ を状態変数表現したものが次式である。

$$\begin{cases} E z X(z) = A X(z) + b U(z) \\ Y(z) = c X(z) + d U(z) \end{cases} \quad (11)$$

ここで、 $X(z) = [X_1 \cdots X_n]^T$

$$\begin{aligned} E &= \begin{bmatrix} \Delta_0^{-1} & -\Delta_1^{-1} & & 0 \\ & \Delta_1^{-1} & -\Delta_2^{-1} & \\ & 0 & \ddots & -\Delta_{n-1}^{-1} \\ & & & \Delta_{n-1}^{-1} \end{bmatrix} \\ A &= \begin{bmatrix} -\Delta_1^{-1} & & 0 \\ \Delta_1^{-1} (1 - \Delta_1^2) & -\Delta_2^{-1} & \\ & 0 & \ddots & -\Delta_{n-1}^{-1} \\ & & \Delta_{n-1}^{-1} (1 - \Delta_{n-1}^2) & -\Delta_n \end{bmatrix} \\ b &= [0 \cdots 0 \ 1]^T \quad c = [\gamma_1 \cdots \gamma_n] \quad d = [b_0] \end{aligned}$$

以上より、 $H(z)$ の状態および出力方程式を得る。

$$\begin{cases} x_{k+1} = A_s x_k + b_s u_k \\ y_k = c_s x_k + d_s u_k \end{cases} \quad (12)$$

ただし,

$$\begin{aligned} A_s &= E^{-1} A = \begin{bmatrix} -\Delta_0 \Delta_1 & -\Delta_0 \Delta_2 & \cdots & -\Delta_0 \Delta_n \\ 1 - \Delta_1^2 & -\Delta_1 \Delta_2 & \cdots & -\Delta_1 \Delta_n \\ & \ddots & \ddots & \vdots \\ 0 & & 1 - \Delta_{n-1}^2 & -\Delta_{n-1} \Delta_n \end{bmatrix} \\ b_s &= E^{-1} b = \begin{bmatrix} \Delta_0 \\ \Delta_1 \\ \vdots \\ \Delta_{n-1} \end{bmatrix} \\ c_s &= c \quad d_s = d \end{aligned}$$

この表現は、離散 Schwartz 実現と呼ばれている。

つぎに、上実現(12)に等価変換を施すことで、 A マトリクスは不変のまま b マトリクスの形を換え、低次元化のための標準形を導く。次節以下に述べるように、この標準形に基づき時間域近似を目的とする安定な低次元モデルが導出される。システムを m 次元に低次元化するとき、等価変換(変換行列 T)により(12)式の表現を次式に変換する。

$$\begin{cases} x_{k+1} = A_s x_k + b_{sm} u_k \\ y_k = c_{sm} x_k + d_s u_k \end{cases} \quad (13)$$

ただし, $\begin{cases} b_{sm} = T^{-1}b_s = [\Delta_0 \Delta_1 \cdots \Delta_m \ 0 \cdots 0]^T \\ c_{sm} = c_s T = [\gamma_1^m \gamma_2^m \cdots \gamma_n^m] \end{cases}$

この形を Schwartz (m) 実現と呼ぶことにする。

Schwartz ($n-1$) 実現は Schwartz 実現に等しい。

特に, $m=0$ の場合, すなわち,

$$\begin{cases} x_{k+1} = A_s x_k + b_{s0} u_k \\ y_k = c_{s0} x_k + d_s u_k \end{cases} \quad (14)$$

ただし, $\begin{cases} b_{s0} = T^{-1}b_s = [1 \ 0 \cdots 0] \\ c_{s0} = c_s T = [\gamma_1^0 \ \gamma_2^0 \cdots \gamma_n^0] \end{cases}$

の形を Schur 実現と呼ぶことにする。このとき, 変換行列 T は上三角行列となる。 γ_i^m の値を $H(z)$ の分母分子係数 (a_i, b_i) より求めるためには, 連立一次方程式を解く操作が必要となる。ただし, $m=0$ の場合には逐次演算のみでよい (付録 2)。

2.2 Schur 実現からの低次元化

前節で導いた Schur 実現 (14) に基づく低次元化は, つぎのように行われる。

状態の集約化: $z_k = [I_m \ 0] x_k \ (1 \leq m \leq n-1)$

低次元システム:

$$\begin{cases} z_{k+1} = F_s z_k + g_{s0} u_k \\ y_k = h_{s0} z_k + j_s u_k \end{cases} \quad (15)$$

ただし, $\begin{cases} F_s = A_{s(m)} & g_{s0} = b_{s0(m)} \\ h_{s0} = c_{s0(m)} & j_s = d_s \end{cases}$

(注意 1) 原システムにおいて次の関係があるとき

$$\begin{cases} |\Delta_1|, \dots, |\Delta_m| \gg |\Delta_{m+1}|, \dots, |\Delta_n| \\ |\gamma_1|, \dots, |\gamma_m| \gg |\gamma_{m+1}|, \dots, |\gamma_n| \end{cases}$$

m は低次元システムの次数の指標となる。

この理由は, 上条件のもとではシステム (14) において, 状態変数 $(x_{m+1} \cdots x_n)$ が準不可制御・不可観測となることによる。

低次元化アルゴリズムは以下の通りである。

(i) 原システムより伝達関数 $H(z)$ を求め, その分母分子係数をそれぞれ $(a_i, b_i) \ i=0 \cdots n$ とする。

(ii) $(a_i, b_i) \ i=0 \cdots n$ より $(\Delta_i, \gamma_i^0) \ i=1 \cdots n$ を求める。

(iii) $(\Delta_i, \gamma_i^0) \ i=1 \cdots m$ より低次元システム $(F_s, g_{s0}, h_{s0}, j_s)$ が得られる。

(iv) 低次元伝達関数 $\tilde{H}(z)$ の分母分子係数 $(\tilde{a}_i, \tilde{b}_i) \ i=0 \cdots m$ は, $(\Delta_i, \gamma_i) \ i=1 \cdots m$ より (ii) の逆操作で求められる。

(注意 2) 原システム伝達関数 $H(z)$ が与えられ, 低次元伝達関数 $\tilde{H}(z)$ を求める過程は, 上アルゴリズム中 (ii), (iv) による逐次演算だけでよい (付録 2)。

Schur 近似の特徴を以下にまとめておく。

(i) 低次元システムの安定性は保証される。

このことは A_s が安定行列であるための必要十分条件が, $|\Delta_i| < 1 \ (i=1 \cdots n)$ と等価であることによる¹⁰⁾。

(ii) 時間域近似である。

このことは,

$$\begin{cases} h_i = \tilde{h}_i \ (i=0 \cdots m) \\ H(z) = \sum_{i=0}^{\infty} h_i z^{-i} \quad \tilde{H}(z) = \sum_{i=0}^{\infty} \tilde{h}_i z^{-i} \end{cases} \quad (16)$$

が成立することによる (注 1)。

(iii) 低次元化手順が容易である。

このことは, $(a_i, b_i) \leftrightarrow (\Delta_i, \gamma_i^0)$ 間の変換が逐次演算により行われることによる。

2.3 Schwartz (m) 実現からの低次元化

2.1 節で導いた Schwartz (m) 実現に基づく低次元化は以下のように行われる。

状態の集約化: $z_k = [I_m \ 0] x_k$

低次元システム:

$$\begin{cases} z_{k+1} = F_s z_k + g_s u_k \\ y_k = h_s z_k + j_s u_k \end{cases} \quad (17)$$

ただし, $\begin{cases} F_s = A_{s(m)} & g_s = b_{s(m)} \\ h_s = c_{s(m)} & j_s = d_s \end{cases}$

(注意 3) Schur 実現からの低次元化において, 低次元システムの次数 m が任意に設定できるのに対し, Schwartz (m) 実現からの低次元化においては, 低次元システムの次数は m に定められる。低次元システム (F_s, g_s, h_s, j_s) は離散 Schwartz 実現となる。低次元化アルゴリズムを以下に示す。

(i) 原システムより伝達関数 $H(z)$ を求め, その分母分子係数をそれぞれ $(a_i, b_i) \ i=0 \cdots n$ とする。

(ii) $(a_i, b_i) \ i=0 \cdots n$ より $(\Delta_i, \gamma_i^m) \ i=1 \cdots n$ を求める。

(iii) $(\Delta_i, \gamma_i^m) \ i=0 \cdots m$ より低次元システム (F_s, g_s, h_s, j_s) が得られる。

(iv) 低次元伝達関数 $\tilde{H}(z)$ の分母分子係数 $(\tilde{a}_i, \tilde{b}_i) \ i=0 \cdots m$ は, $(\Delta_i, \gamma_i^m) \ i=1 \cdots m$ より逐次的に求められる。

(注意 4) step (ii) は n 次元連立一次方程式を解く操作を必要とする。これに対し, step (iv) は表形式による逐次演算だけでよい (付録 1, Table 3)。

以上の低次元化法の特徴は以下の通りである。

(i) 低次元システムの安定性は保証される。

(ii) 時間域近似である。

(16) 式に対し, $SE = \sum_{i=0}^{\infty} (h_i - \tilde{h}_i)^2$ と定めるとき, 本近似法はつぎのように性質をもつ。

$$h_s = [\gamma_1^m \gamma_2^m \cdots \gamma_m^m]:$$

$$SE(\Delta_1 \cdots \Delta_m \gamma_1^m \cdots \gamma_m^m) = \text{Min}_{\tilde{\gamma}_i} SE(\Delta_1 \cdots \Delta_m \tilde{\gamma}_1 \cdots \tilde{\gamma}_m) \quad (18)$$

(注 1) $\tilde{h}_i = c_{s0} A_{s(m)}^{-i} b_{s0}$ $\tilde{h}_i = c_{s0(m)} A_{s(m)}^{i-1} b_{s0(m)}$ および $c_{s0} A_{s(m)}^{-i} b_{s0} = c_{s0(m)} A_{s(m)}^{i-1} b_{s0(m)}$ ($i=1 \cdots m$) による。最後の式の成立については文献 6) 参照。

低次元化アルゴリズムは以下の通りである。

(i) 原システムより伝達関数 $H(z)$ を求め、その分母分子係数をそれぞれ $(a_i, b_i) i=0 \cdots n$ とする。

(ii) $(a_i, b_i) i=0 \cdots n$ より $(\alpha_i, \beta_i) i=0 \cdots n$ を求める。

(iii) $(\alpha_i, \beta_i) i=0 \cdots m$ より低次元システム $(F_{DR}, g_{DR}, h_{DR}, j_{DR})$ が得られる。

(iv) 低次元伝達関数 $\tilde{H}(z)$ の分母分子係数 $(\tilde{a}_i, \tilde{b}_i) i=0 \cdots m$ は $(\alpha_i, \beta_i) i=0 \cdots m$ より (ii) の逆操作にて求められる。

(注意 3) 原システム伝達関数が与えられ、低次元伝達関数を求める過程は、上アルゴリズム中 (ii), (iv) による逐次演算だけでよい (付録 4, Table 4, 5 参照)。離散 Routh 近似の特徴を以下にまとめておく。

(i) 低次元システムの安定性は保証される。

このことは、離散 Routh 実現されたシステムが安定であるための必要十分条件が、 $\alpha_i > 0 (i=1 \cdots n)$ と等価であることによる¹¹⁾。

(ii) 低周波域近似である。

このことは、次式が成立することによる (注 2)。

$$\begin{cases} r_i = \tilde{r}_i & (i=0 \cdots m) \\ H(z) = \sum_{i=0}^{\infty} r_i \left(\frac{z-1}{z+1} \right)^i & \tilde{H}(z) = \sum_{i=0}^{\infty} \tilde{r}_i \left(\frac{z-1}{z+1} \right)^i \end{cases} \quad (27)$$

(iii) 低次元化手順が容易である。

このことは、 $(a_i, b_i) \leftrightarrow (\alpha_i, \beta_i)$ 間の変換が逐次演算により行えることによる。

4. 例 題

前章までに述べてきた低次元化法の適用例を示す。原システムは伝達関数で与えられ、次式とする。

$$H(z) = \frac{b_0 z^8 + b_1 z^7 + \cdots + b_8}{a_0 z^8 + a_1 z^7 + \cdots + a_8} \quad (28)$$

Table 1 Coefficients of original transfer functions

system (i)			System (ii)		
i	a_i	b_i	i	a_i	b_i
0	1.000E 00	0.000E 00	0	1.000E 00	0.000E 00
1	-1.200E 00	1.000E 00	1	-6.308E-01	2.103E-01
2	1.050E-01	8.990E-02	2	-4.185E-01	1.395E-01
3	4.690E-01	-4.300E-01	3	7.875E-02	-2.625E-02
4	4.680E-03	-3.150E-02	4	-5.700E-02	1.900E-02
5	-1.140E-01	1.000E-02	5	1.935E-01	-6.450E-02
6	1.820E-02	2.560E-04	6	9.825E-02	-3.275E-02
7	7.590E-03	-8.170E-03	7	-1.650E-02	5.500E-03
8	2.000E-03	-1.890E-03	8	2.250E-03	-7.500E-04

(注 2) Routh 近似の性質および双一次変換 (19) より示される。

係数値は Table 1(i), (ii) に与えられる。システム (i) に対しては単位パルス応答を近似する目的で 2 章の方法を、システム (ii) に対しては低周波域近似を目的として 3 章の手法をそれぞれ使い、低次元化を行った結果を Fig. 2, 3, 4 にそれぞれ示す。両者とも比較的良好な近似特性が得られていることがわかる (注 3)。

5. 結 言

本論文では、線形一入出力安定離散時間システムを対象とし、Aggregation法による低次元化手法として、

Fig. 2 Unit pulse responses of original and reduced-order transfer functions (Reduction based on Schur form)

Fig. 3 Unit pulse responses of original and reduced-order transfer functions (Reduction based on Schwartz (m) form)

Fig. 4 Unit step responses of original and reduced-order transfer functions (Reduction based on Discrete-Routh form)

(i) 離散 Schwartz 実現に基づく低次元化手法

(ii) 離散 Routh 実現に基づく低次元化手法

の2手法を提案した。(i)は時間域近似、(ii)は低周波域近似、また両者とも安定性を保証する近似法であり、低次元化手順は容易なものであった。両手法のどちらを適用するかは例題で見たように、対象システムの目的・特性により決定される。

最後に、本研究遂行に当たり貴重なご意見・ご助言を頂きました 高橋進一教授に感謝の意を表わします。

参考文献

- 1) N.R. Sandell, et al.: Survey of Decentralized Control Methods for Large Scale Systems, IEEE Trans. on Automatic Control, **AC-23-2**, 108/128 (1978)
- 2) 前田: 大規模な線形システムの低次元モデルとシステム構造, システムと制御, **22-11**, 655/664 (1978)
- 3) 辻澤・田川: 伝達関数行列に対する簡単化の一手法; 電子通信学会論文誌 A, **J 65-A-5**, 462/469 (1982)
- 4) V.R. Moorthi and R.P. Aggarwal: Suboptimal and Near-Optimal Control of a Load-Frequency-Control System, Proc. IEE, **119-11**, 1653/1660 (1972)
- 5) 神藤・岩住: Aggregation 法による各種低次元モデルの表現法, 計測自動制御学会論文集, **17-1**, 139/146 (1981)
- 6) L. Pernebo and L.M. Silverman: Model Reduction via Balanced State Space Representations, IEEE Trans. on Automatic Control, **AC-27-2**, 382/387 (1982)
- 7) B.D.O. Anderson, et al.: Schwartz Matrix Properties for Continuous and Discrete Time Systems; International J. Control, **23-1**, 1/16 (1976)
- 8) 滝沢・岸・浜田: 状態変数法によるラティスデジタルフィルタの構成, 電子通信学会論文誌 A, **J 65-A-4**, 363/370 (1982)
- 9) N.I. Akhiezer: The Classical Moment Problem, Oliver & Boyd (1965)
- 10) 浜田: Inner と離散時間システムの代数的安定論, 計測と制御, **20-8**, 748/755 (1981)
- 11) M.F. Hutton and M. Friedland: Routh Approxima-

tion for Reducing Order of Linear Time-Invariant Systems, IEEE Trans. on Automatic Control, **AC-20-3**, 329/337 (1975)

12) 伊藤: システム制御理論, 昭晃堂 (1973)

13) 木村: デジタル信号処理と制御, 昭晃堂 (1982)

《付 録》

〈付録1〉 $(a_i, b_i) \leftrightarrow (\Delta_i, \gamma_i)$ 間変換表

変換: $(a_i, b_i) \rightarrow (\Delta_i, \gamma_i)$ を **Table 2**

変換: $(\Delta_i, \gamma_i) \rightarrow (a_i, b_i)$ を **Table 3** にそれぞれ示す。特に、変換表 $a_i \rightarrow \Delta_i$ は Jury 表と呼ばれるものである¹⁰⁾。

〈付録2〉 Schwartz (m) Form への変換

与伝達関数 $H(z)$ を Schwartz (m) 実現する手順, すなわち, (a_i, b_i) より (Δ_i, γ_i^m) を求める手順を以下に示す。まず, つぎの関係が成立する¹²⁾。

$(A_c, b_c, c_c, d_c): H(z)$ の Companion 実現

$(A_s, b_{sm}, c_{sm}, d_s): H(z)$ の Schwartz (m) 実現

T : 変換行列; $(A_s, b_{sm}, c_{sm}, d_s) = (T A_c T^{-1},$

$T b_c, c_c T^{-1}, d_c)$

このとき,

$$T = W_s W_c^{-1} \begin{pmatrix} W_c \triangleq [b_c A_c b_c \cdots A_c^{n-1} b_c] \\ W_s \triangleq [b_{sm} A_s b_{sm} \cdots A_s^{n-1} b_{sm}] \end{pmatrix} \quad (29)$$

上関係に基づき, (a_i, b_i) より (Δ_i, γ_i^m) を求める手順は以下にまとめられる。

【Algorithm: $(a_i, b_i) \rightarrow (\Delta_i, \gamma_i^m)$ 】

(i) a_i より Jury 表に従い Δ_i を求める。

(ii) Δ_i より Schwartz (m) 実現 (A_s, b_{sm}) を作る。

(iii) $c_c W_c, W_s$ を計算する。

(iv) 連立一次方程式: $c_{sm} W_s = c_c W_c$ を解くことにより, $c_{sm} = [\gamma_1^m \gamma_2^m \cdots \gamma_n^m]$ が得られる。

(注意) 特に $m=0$, すなわち Schur 実現の場合, W_s は上三角行列となり連立一次方程式を解く過程 (iv) は後退代入だけとなる。換言すれば, (a_i, b_i) より (Δ_i, γ_i^0) を求める手順は逐次演算のみとなる。

〈付録3〉 2.3 節性質 (ii): (18) 式の証明

Table 3 Calculation of (a_i, b_i) from (Δ_i, γ_i)

Δ_0	a_0^0	$a_i^i = \Delta_i \quad (i=0 \cdots n)$
Δ_1	$a_1^1 \quad a_0^1$	$\begin{cases} a_j^i = a_j^{i-1} + \Delta_i a_{i-j}^{i-1} \\ a_0^i = 1 \end{cases}$
$\vdots$		$(i=1 \cdots n)$
Δ_{n-1}	$a_{n-1}^{n-1} \quad a_{n-2}^{n-1} \cdots a_0^{n-1}$	$(j=1 \cdots i-1)$
Δ_n	$a_n^n \quad a_{n-1}^n \cdots a_1^n \quad a_0^n$	$a_j = a_j^n \quad (j=0 \cdots n)$
r_1	b_1^1	$b_i^i = r_i \quad (i=1 \cdots n)$
r_2	$b_2^2 \quad b_1^2$	$b_j^i = b_j^{i-1} + r_i a_{i-j}^{i-1}$
$\vdots$		$(i=2 \cdots n)$
r_{n-1}	$b_{n-1}^{n-1} \quad b_{n-2}^{n-1} \cdots b_1^{n-1}$	$(j=1 \cdots i-1)$
r_n	$b_n^n \quad b_{n-1}^n \cdots b_2^n \quad b_1^n$	$b_j = b_j^n + b_0 a_j \quad (j=1 \cdots n)$

Table 2 Calculation of (Δ_i, r_i) from (a_i, b_i)

Δ_n	$a_0^n \quad a_1^n \cdots a_{n-1}^n \quad a_n^n$	$a_j^n = a_j \quad (j=0 \cdots n) \quad (a_0=1)$
Δ_{n-1}	$a_0^{n-1} \quad a_1^{n-1} \cdots a_{n-1}^{n-1}$	$a_j^i = \frac{a_j^{i+1} - \Delta_{i+1} a_{i+1-j}^{i+1}}{1 - \Delta_{i+1}^2}$
$\vdots$		$(i=0 \cdots n-1)$
Δ_1	$a_0^1 \quad a_1^1$	$(j=0 \cdots i)$
Δ_0	a_0^0	$\Delta_i = a_i^i \quad (i=0 \cdots n) \quad (\Delta_0=1)$
r_n	$b_1^n \quad b_2^n \cdots b_{n-1}^n \quad b_n^n$	$b_j^n = b_j - b_0 a_j \quad (j=1 \cdots n)$
r_{n-1}	$b_1^{n-1} \quad b_2^{n-1} \cdots b_{n-1}^{n-1}$	$b_j^i = b_j^{i+1} - r_{i+1} a_{i+1-j}^{i+1}$
$\vdots$		$(j=1 \cdots i)$
r_2	$b_2^2 \quad b_1^2$	
r_1	b_1^1	$r_i = b_i^i \quad (i=1 \cdots n)$

(注 3) 実際の計算に際して, 低次元化アルゴリズムの数値的安定性の問題があるが, 本アルゴリズムは Schwartz 実現の低係数感度特性⁹⁾ が示すように, この点に特に問題はないと思われる。

Table 4 Calculation of $(\alpha_i \beta_i)$ from $(a_i b_i)$

	P_0^n	P_1^n	$\dots$	P_{n-2}^n	P_{n-1}^n	P_n^n		β_0	$\bar{b}_0^{n-1}$	$\bar{b}_1^{n-1}$	$\dots$	$\bar{b}_{n-2}^{n-1}$	$\bar{b}_{n-1}^{n-1}$	
α_1	P_0^{n-1}	P_1^{n-1}	$\dots$	P_{n-2}^{n-1}	P_{n-1}^{n-1}	P_n^{n-1}	$\begin{cases} P_j^n = a_j + a_{n-j} \\ (j=0 \dots [n-1/2]) \\ P_{n-j}^n = a_{n-j} - a_j \\ (j=0 \dots [n-1/2]) \\ P_{n/2}^n = a_{n/2} (n:\text{even}) \end{cases}$	β_1	b_0^{n-1}	b_1^{n-1}	$\dots$	b_{n-2}^{n-1}	b_{n-1}^{n-1}	$\beta_0 = \frac{b_0 + \dots + b_n}{a_0 + \dots + a_n}$
α_2	P_0^{n-2}	P_1^{n-2}	$\dots$	P_{n-2}^{n-2}	P_{n-1}^{n-2}	P_n^{n-2}		β_2	$\bar{b}_0^{n-2}$	$\bar{b}_1^{n-2}$	$\dots$	$\bar{b}_{n-2}^{n-2}$	$\bar{b}_{n-1}^{n-2}$	
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$		$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	
α_{n-1}	P_0^1	P_1^1	$\dots$	P_{n-2}^1	P_{n-1}^1	P_n^1		β_{n-1}	b_0^1	b_1^1	$\dots$	b_{n-2}^1	b_{n-1}^1	$\bar{b}_j^{n-1} = b_j - \beta_0 a_j$ ($j=0 \dots n-1$)
α_n	P_0^0	P_1^0	$\dots$	P_{n-2}^0	P_{n-1}^0	P_n^0		β_n	$\bar{b}_0^0$	$\bar{b}_1^0$	$\dots$	$\bar{b}_{n-2}^0$	$\bar{b}_{n-1}^0$	
$\begin{cases} P_0^i = -P_{i+1}^{i+1} \\ P_j^i = -P_{i+1-j}^{i+1} + P_{j-1}^i \quad (j=1 \dots [i/2]) \\ \alpha_{n-i} = \frac{P_0^{i+1} + \dots + P_{[i+1/2]}^{i+1}}{2(P_0^i + \dots + P_{[i/2]}^{i+1})} \\ \begin{cases} P_j^i = \alpha_{n-i} P_0^i - P_0^{i+1} \\ P_{i-j}^i = \alpha_{n-i} (P_{j-1}^i + P_j^i) - P_{j+1}^{i+1} + P P_{i-j+1}^i \quad (j=1 \dots [i-1/2]) \\ (i=0 \dots n-1) \end{cases} \end{cases}$								$\begin{cases} b_0^i = \bar{b}_0^i; \quad b_j^i = \bar{b}_j^i + b_{j-1}^i \quad (j=1 \dots i) \\ \beta_{n-i} = \frac{b_0^i + \dots + b_i^i}{2(P_0^i + \dots + P_{[i/2]}^{i+1})} \\ (i=0 \dots n-1) \\ \begin{cases} \bar{b}_j^i = b_j^{i+1} - \beta_{n-i-1} P_{j+1}^{i+1} \quad (j=0 \dots [i/2]) \\ \bar{b}_{i-j}^i = b_{i-j}^{i+1} - \beta_{n-i-1} P_{j+1}^{i+1} \quad (j=0 \dots [i/2]-1) \\ \bar{b}_{i+1/2}^i = b_{i+1/2}^{i+1} - 2\beta_{n-i-1} P_{i+1/2}^{i+1} \quad (i:\text{odd}) \\ (i=0 \dots n-2) \end{cases} \end{cases}$						

Table 5 Calculation of $(a_i b_i)$ from $(\alpha_i \beta_i)$

α_n	P_0^n	$\begin{cases} a_j = 1/2(P_j^n - P_{n-j}^n) \\ (j=0 \dots [n-1/2]) \\ a_{n-j} = 1/2(P_{n-j}^n - P_j^n) \\ (j=0 \dots [n-1/2]) \\ a_{n/2} = P_{n/2}^n (n: \text{even}) \end{cases}$				β_n	b_0^n	$b_j = \bar{b}_j^{n-1} + \beta_0 a_j$ ($j=0 \dots n-1$)			
α_{n-1}	$P_0^{n-1} \quad P_1^{n-1}$					β_{n-1}	$b_0^{n-1} \quad b_1^{n-1}$	$b_n = \beta_0(a_0 + \dots + a_n)$ $-(b_0 + \dots + b_{n-1})$			
$\vdots$	$\dots \dots \dots$					$\vdots$	$\dots \dots \dots$				
α_2	$P_0^{n-2} \quad P_1^{n-2} \dots P_{n-2}^{n-2}$					β_2	$b_0^{n-2} \quad b_1^{n-2} \dots b_{n-2}^{n-2}$				
α_1	$P_0^{n-1} \quad P_1^{n-1} \dots P_{n-2}^{n-1} \quad P_{n-1}^{n-1}$		$b_0^{n-1} \quad b_1^{n-1} \dots b_{n-2}^{n-1}$	b_{n-1}^{n-1}							
	$P_0^n \quad P_1^n \quad \dots \quad P_{n-2}^n \quad P_{n-1}^n \quad P_n^n$										
$P_0^0 = 1$											
$\begin{cases} P_0^i = \alpha_{n-i+1} P_0^{i-1} - P_{i-1}^{i-1} \\ P_j^i = \alpha_{n-i+1} (P_{j-1}^{i-1} + P_j^{i-1}) - P_{i-j}^{i-1} + P_{i-j-1}^{i-1} \\ (j=1 \dots [i/2]-1) \\ P_{[i/2]}^i = 2\alpha_{n-i+1} (P_0^{i-1} + \dots + P_{[i-1/2]}^{i-1}) - (P_0^i + \dots + P_{[i/2]}^{i-1}) \\ P_j^i = P_{i-j-1}^{i-1} - P_{i-j}^{i-1} \quad (j=[i/2]+1 \dots i-1) \\ P_i^i = -P_0^{i-1} \\ (i=1 \dots n) \end{cases}$											

β_0	$\bar{b}_0^{n-1} \quad \bar{b}_1^{n-1} \dots \bar{b}_{n-2}^{n-1} \quad \bar{b}_{n-1}^{n-1}$
$b_0^0 = 2\beta_n P_0^0; \quad \bar{b}_0^0 = b_0^0$	
$\begin{cases} b_j^i = \beta_{n-i} P_j^i + \bar{b}_j^{i-1} \quad (j=0 \dots [i-1/2]) \\ b_{i-j}^i = \beta_{n-i} P_j^i + \bar{b}_{i-j}^{i-1} \quad (j=1 \dots [i-1/2]) \\ b_{i/2}^i = 2\beta_{n-i} P_{i/2}^i + \bar{b}_{i/2}^{i-1} \quad (i: \text{even}) \\ b_i^i = 2\beta_{n-i} (P_0^i + \dots + P_{[i/2]}^i) - (b_0^i + \dots + b_{i-1}^i) \\ \bar{b}_0^i = b_0^i; \quad \bar{b}_j^i = -b_{j-1}^i + b_j^i \quad (j=1 \dots i) \\ (i=1 \dots n-1) \end{cases}$	

$SE(\Delta_1 \dots \Delta_m \gamma_1^m \dots \gamma_m^m) \triangleq SE_m$ は次式で表現される¹³⁾.

$$SE_m = c_{sm} P_s c_{sm}^T + h_s Q_s h_s^T - 2c_{sm} R_s h_s^T \quad (30)$$

ここで, P, Q, R_s はそれぞれ次方程式の解である.

$$\begin{cases} A_s P_s A_s^T - P_s = -b_{sm} b_{sm}^T & (31a) \\ F_s Q_s F_s^T - Q_s = -g_s g_s^T & (31b) \\ A_s R_s F_s^T - R_s = -b_{sm} g_s^T & (31c) \end{cases}$$

行列方程式(31b)(31c)は容易に解くことができ, 次式となる.

$$\begin{cases} Q_s = \text{diag} \left[\prod_{i=1}^m (1 - \Delta_i^2)^{-1} \prod_{i=2}^m (1 - \Delta_i^2)^{-1} \dots \right. \\ \quad \left. (1 - \Delta_m^2)^{-1} \right] \\ R_s = \begin{bmatrix} Q_s \\ 0 \end{bmatrix} \end{cases} \quad (32)$$

(32)式を(30)式に代入して,

$$SE_m = c_{sm} P_s c_{sm}^T - h_s Q_s h_s^T \quad (33)$$

これに対し, $h \neq h_s$ の場合,

$$\begin{aligned} SE &= c_{sm} P_s c_{sm}^T + h Q_s h^T - 2c_{sm} R_s h^T \\ &= c_{sm} P_s c_{sm}^T - h_s Q_s h_s^T + (h - h_s) Q_s (h - h_s)^T \\ &= SE_m + (h - h_s) Q_s (h - h_s)^T \end{aligned} \quad (34)$$

となる. 上式は(18)式を示している.

〈付録4〉 $(a_i, b_i) \leftrightarrow (\alpha_i, \beta_i)$ 間変換表

変換: $(a_i, b_i) \rightarrow (\alpha_i, \beta_i)$ を Table 4

変換: $(\alpha_i, \beta_i) \rightarrow (a_i, b_i)$ を Table 5 に示す.