



多変数制御系の設計についての考察

著者	清水 俊幸, 久野 啓, 片山 登揚
引用	大阪府立工業高等専門学校研究紀要, 1990, 24, p.19-26
URL	http://doi.org/10.24729/00007869

多変数制御系の設計についての考察

清水 俊幸* 久野 啓** 片山 登揚***

Notes on the Design of Multivariable Control Systems

Toshiyuki SHIMIZU * Kei HISANO** Noriaki KATAYAMA ***

ABSTRACT

This paper deals with the design of multivariable control systems by making use of formula manipulation. Of the process of the design of multivariable control systems, it can be applied in two points. One is to get the transfer function matrix expressions from the given block diagram. The other is to decide the precompensator for the given system to be decoupled.

Key Words: Multivariable Control Systems, Computer Aided Design, Formula Manipulation.

第 1 節 はじめに¹⁻³⁾

多入力多出力の制御系を設計する場合、大きく分けて 2 つの方法がある。1 つは現代制御理論による状態空間法と、もう 1 つは古典制御理論を多変数系に拡張した周波数領域における解析法である。

どちらの手法を用いて制御系の設計を行う場合でも、設計に伴う計算は、複雑かつ膨大なものであり、系の入出力数が増加するにつれて、多種多様なデータの管理は大変な仕事になってくる。そこで、これらのデータ処理を計算機にまかせ、設計者は構成された制御系の評価を行い、次に何をすべきかだけに注意していればよいように、計算機を設計に援用するための CAD システムが開発され、多変数制御理論を実際に応用することが可能になってきた。

しかし、これらのシステムはすべて数値計算によるものであり、制御系のパラメータを記号のままでも与えられた場合には、取り扱うことができない。ところが、制御

系の解析、設計の際にはパラメータを記号のままでも処理したほうが都合の良い場合が多く見受けられる。この種の問題に対しては、数値計算による CAD システムは不向きであり、記号処理に適した CAD システム、すなわち数式処理を用いたシステムを適用することが考えられる。

そこで、本論文では多変数制御系の設計に数式処理を適用した結果について述べる。多変数制御系の設計のなかで、特に非干渉制御に注目する。つまり、古典制御理論を多変数系に拡張するとき、系の内部に相互干渉があってはならない。そのため、相互干渉を近似的に取り除くために、前置補償器を設計することが必要となってくる。また、この前段階として多変数制御系のブロック線図から、入出力関係を伝達関数行列で表現しておくことが必要である。そこで、上記 2 つの過程に数式処理システムを適用することを試みた。以下、第 2 節では今回用いた数式処理システムの概要を述べ、第 3 節では前置補償器を設計するための理論を述べ、数式処理に適用するときのアルゴリズムを示す。さらに、第 4 節では簡単な 2 入力 2 出力系の例と、3 入力 3 出力系であえられる圧延プロセスへの適用結果について、述べる。

第 2 節 数式処理システム⁴⁻⁶⁾

本節では、数式処理システムの概要と今回用いたシス

平成 2 年 4 月 9 日受理

* 広島大学工学部(Hiroshima University, Faculty of Engineering)(本校機械工学科 平成 2 年 3 月 卒業)

** 電気通信大学工学部(The University of Electro-Communications, Faculty of Engineering)

(本校機械工学科 平成 2 年 3 月 卒業)

***機械工学科(Department of Mechanical Engineering)

テムについて簡単に述べる。

数式処理システムとは、近年では計算機代数学という語が広く用いられているように、計算機により数式を数式のままに変形および様々な処理を行い、解析的結果を得ようとするものである。

汎用数式処理システムでは、代数学や解析学の基礎的な計算を扱うことができ、多くの分野で利用できる。しかし、アルゴリズム化されていくことで単純な結果の処理にも、中間結果が膨大になり結果がでないことがある。このようなときには、人間が介して前処理を行わなければならない。このような不備をもってはいるが、制御系のCAD, CAI等に応用されている。

次に、本研究で用いた数式処理システム REDUCE と、その動作環境について簡単に述べる。REDUCE は、1963年にスタンフォード大学のA.C.Hern博士によって開発され、現在世界でもっとも広く使われている数式処理システムの1つである。その開発目的は理論物理学上の長大な数式を処理することであったが、開発者の活発な普及活動により多分野で広く使われている。そのため稼働可能なコンピュータも、パーソナルコンピュータから、ワークステーション、ミニコンまで幅広くなっている。

REDUCEを本研究に利用するために使用したハードウェア及びソフトウェアを表1に示す。また、これらハードウェアとソフトウェアの関係を、概念的に表したものが図1である。なお、図中の四角形の中の単語は、ソフトウェアで用いられるコマンドを表している。

表1 使用したハードウェアとソフトウェア

種別	品名	備考
CPU	NEC PC-9801VM21 + PC-9801-16	68000CPU ボード
RAM	CANOPUS MB68K-11 + MB68K-A21	1MByteRAM MB68K-11に 装着する1M- ByteRAM
モニタ	NEC PC-KD854	640x400ドット
プリンタ	NEC PR-201F2 + PR-201-23	24ピン日本語 シリアルプリン タ トラクタフィ ード
OS	NEC CP/M-86 ver.1.1 CANOPUS CP/M-68K ver.1.2	
言語	StaffLISP	REDUCE記 述言語
ソフト	REDUCE3.2 (StaffLISP)	数式処理シ ステム

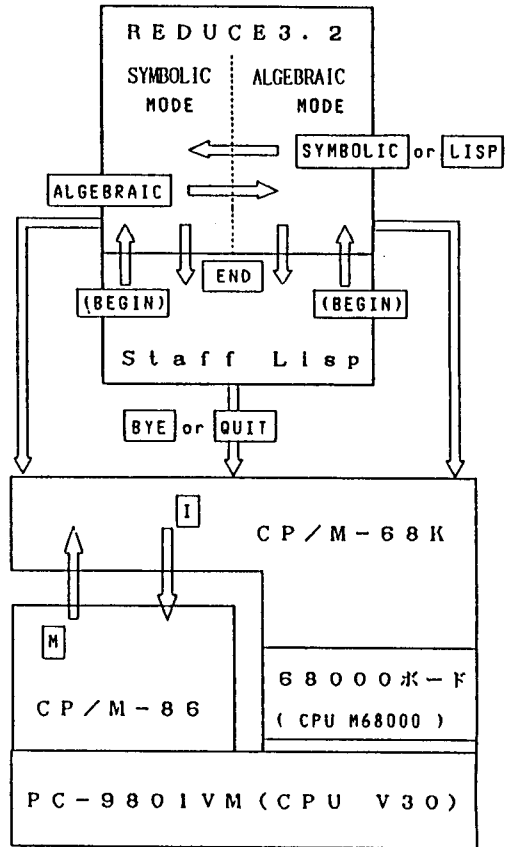


図1 システム概念図

第3節 非干渉制御系の設計について

3.1 前置補償器の設計理論

非干渉制御における前置補償器の設計理論を述べるまえに、多変数制御系について簡単に述べる。

本論文では、図2にあたえられるような構造をしているフィードバック制御系を考察対象としている。ここで $G_p(s)$ は制御対象の伝達関数行列を、 $u(s), y(s)$ はそれぞれ入力ベクトル、出力ベクトルを表している。また $r(s)$ は目標値ベクトルであり、さらに入力変数の数と出力変数の数はともに n で等しいものとする。図2と一

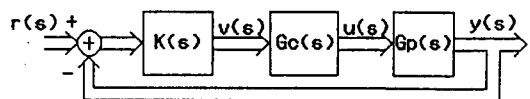


図2 多変数制御系の基本構成

変数制御系との違いは、前置補償器 $G_C(s)$ を置く点にある。前置補償器は主として制御対象の相互干渉を取り除くための、なるべく簡単な構造をもった一種の制御器である。 $K(s)$ は主制御器の伝達関数行列で、これは各ループを単独に制御するための対角行列になっている。なお、図 2 の構図は各操作量がそれぞれ主に影響を与える制御量と一対一に対応しているということを前提にして

いる。次に、前置補償器の設計方法について述べる。本論文では逆ナイキスト配列法をもちいるが、詳細は文献⁷⁾に譲りここでは設計するための要点だけを述べることにする。前述したように、相互干渉を取り除くためには、図 2 で制御対象の伝達関数行列 $G_P(s)$ が与えられたときには、補償要素 $G_C(s)$ を適当に選んで $G(s) = G_P(s) \cdot G_C(s)$ をなるべく対角行列に近づければよい。

このため、Rosenbrock は、 $G_P(0)$ が正則行列であれば $G_C(s)$ は実数実行列として実現するのが実用的かつ簡単であるため、

$$G_C(s) = [G_P(0)]^{-1} \quad (1)$$

を選ぶことを提案している。明らかに、

$$[G_P(s) \cdot G_C(s)]_{s=0} = E_n \quad (2)$$

となり、直流成分においては非干渉化されるが、より一般に $s = j\omega_0$ において非干渉化するには、

$$G_C(s) = [G_P(j\omega_0)]^{-1} \quad (3)$$

とすればよい。ここで、 E_n はサイズ n の単位行列、 j は虚数単位とする。ただし、(3)式右辺の各要素は複素数となるため実数実行列の範囲内で実現することはできない。

従って、以下のような手法を考える。 h_{km} 、を実数として、

$$G_C(s) = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{12} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{n1} & h_{n2} & \dots & h_{nn} \end{bmatrix} \quad (4)$$

とおくと、 $G(s) = G_P(s) \cdot G_C(s)$ の (i, m) 要素 $L_{im}(s)$ は $s = j\omega_0$ で

$$L_{im}(j\omega_0) = \sum_{k=1}^n g_{ik}(j\omega_0) \cdot h_{km} \quad (5)$$

と表される。ただし、 $g_{ik}(s)$ は $G_P(s)$ の (i, k) 要素とする。ここで、 $G_C(s)$ の列ベクトル h_P の要素を制限条件

$$h_{1P}^2 + h_{2P}^2 + \dots + h_{nP}^2 = 1 \quad (6)$$

のもとで、 $G(j\omega_0)$ の第 P 列の非対角要素の絶対値の 2 乗

$$\sum_{i \neq P}^n |L_{iP}(j\omega_0)|^2 \quad (7)$$

が最小になるように選ぶために

$$g_{ik}(j\omega_0) = \alpha_{ik} + j\beta_{ik} \quad (8)$$

とおく。
但し、

$$\left. \begin{aligned} \alpha_{ik} &= \text{Re} [g_{ik}(j\omega_0)] \\ \beta_{ik} &= \text{Im} [g_{ik}(j\omega_0)] \end{aligned} \right\} \quad (9)$$

とする。ここで、行列 A_P を

$$\left. \begin{aligned} A_P &= (a_{km}^P) \\ a_{km}^P &= \sum_{i \neq P}^n (\alpha_{ik} \alpha_{im} + \alpha_{ik} \alpha_{im}) \end{aligned} \right\} \quad (10)$$

とおくとき、 h_P は簡単な計算から

$$A_P \cdot h_P = \lambda h_P \quad (11)$$

を満たさねばならないことがわかる。他方、

$$h_P^t \cdot A_P \cdot h_P = \sum_{i \neq P}^n |L_{iP}(j\omega_0)|^2 \quad (12)$$

が成立することから、(11)式を用いて

$$\sum_{i \neq P}^n |L_{iP}(j\omega_0)|^2 = \lambda h_P^t \cdot h_P = \lambda \quad (13)$$

を得る。ところが、(13)式の左辺は相互干渉の度合を表しており、小さいほうが望ましい。従って、 λ は対称行列 A_P の最小固有値 (非負) をとればよい。つまり、

$G_C(s) = [G_P(j\omega_0)]^{-1}$ は n 個の行列 $A_1, A_2 \dots A_n$ の最小固有値に対する固有ベクトルを並べたものになり、 $G_C(s)$ を求める方法を擬似対角化と呼ぶ。

3.2 数式処理によるアルゴリズム

3.1 で述べた前置補償器の設計方法に数式処理を適用するためのアルゴリズムについて、簡単に示す。

(手順1)

前置補償器の設計の前段階として、与えられたブロック線図から、入出力の関係を1つの伝達関数行列に表現する。つまり、与えられたブロック線図に内部変数を導入して、各ブロックの入出力関係から系全体の入出力関係を $G_P(s)$ として、行列表現する。

(手順2)

複素行列 $G_P(j\omega_0)$ の実部のみと虚部のみとからなる、行列を求める。

(手順3)

⑩式より $A_p (p = 1, 2, \dots, n)$ をもとめ、各行列の最小固有値とそれに対する固有ベクトルを求める。 n 個の固有ベクトルから前置補償器 $G_C(s)$ を決定する。

第4節 実行例およびまとめ

前節で述べたアルゴリズムに従って開発した本システムを、2入力2出力系の簡単な例と、実際のプラントの例にそれぞれ適用した結果について示す。ただし、後者の例については、伝達関数行列の表現(手順1)までとした。

4.1 2入力2出力系への適用例

図3に与えられる2入力2出力系を考える。まず、ブロック線図に図3に示すように、内部変数 $X1$ から $X13$ を仮定する。各ブロックの入出力の関係を表す式を、システムに取り込むために変形し、各々に名前をつけて作成した入力データを図4に示す。

また、計算結果として、伝達関数行列 G_P, G_C の実部よりなる行列、虚部よりなる行列、さらに⑩式より決定される、行列 A_1, A_2 と、前置補償器 G_C を図5から図10までに示す。なお、 G_C については、 $\omega_0 = 1$ を代入した結果を示してある。

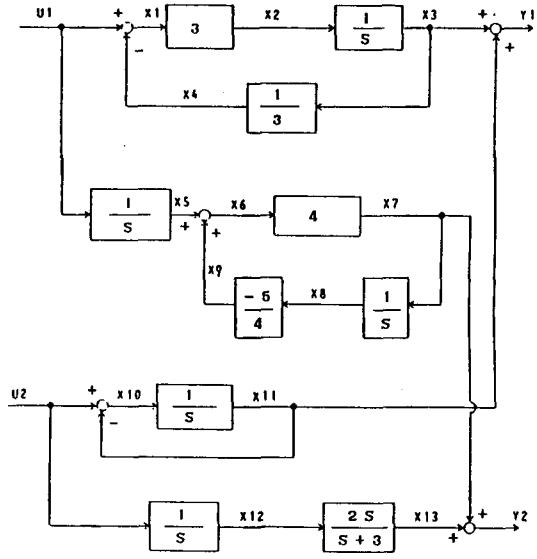


図3 2入力2出力系のブロック線図

```

% naibu hensuu no data ;
xd(1) := u(1) - x(4) - x(1);
xd(2) := 3 * x(1) - x(2);
xd(3) := 1 / s * x(2) - x(3);
xd(4) := 1 / 3 * x(3) - x(4);
xd(5) := 1 / s * u(1) - x(5);
xd(6) := x(5) + x(9) - x(6);
xd(7) := 4 * x(6) - x(7);
xd(8) := 1 / s * x(7) - x(8);
xd(9) := -5 / 4 * x(8) - x(9);
xd(10) := u(2) - x(11) - x(10);
xd(11) := 1 / s * x(10) - x(11);
xd(12) := 1 / s * u(2) - x(12);
xd(13) := 2 * s / (s + 3) * x(12) - x(13);
yd(1) := x(3) + x(11) - y(1);
yd(2) := x(7) + x(13) - y(2);

end;
    
```

図4 入力データ

```

MAT(1,1) := 3 / (S + 1)
MAT(1,2) := 1 / (S + 1)
MAT(2,1) := 4 / (S + 5)
MAT(2,2) := 2 / (S + 3)
    
```

図5 伝達関数行列 G_P の各要素

$$\begin{aligned} \text{MAT}(1,1) &:= 3/(W^2 + 1) \\ \text{MAT}(1,2) &:= 1/(W^2 + 1) \\ \text{MAT}(2,1) &:= (20*(W^2 + 9))/(W^4 + 34*W^2 + 225) \\ \text{MAT}(2,2) &:= (6*(W^2 + 25))/(W^4 + 34*W^2 + 225) \end{aligned}$$

図6 伝達関数行列 G_p の実部の行列要素

$$\begin{aligned} \text{MAT}(1,1) &:= (-3*W)/(W^2 + 1) \\ \text{MAT}(1,2) &:= (-W)/(W^2 + 1) \\ \text{MAT}(2,1) &:= (4*W*(-W^2 - 9))/(W^4 + 34*W^2 + 225) \\ \text{MAT}(2,2) &:= (2*W*(-W^2 - 25))/(W^4 + 34*W^2 + 225) \end{aligned}$$

図7 伝達関数行列 G_p の虚部の行列要素

$$\begin{aligned} \text{MAT}(1,1) &:= (16*(W^6 + 43*W^4 + 531*W^2 + 2025))/(W^8 + 68*W^6 + 1606*W^4 + 15300*W^2 + 50625) \\ \text{MAT}(1,2) &:= (8*(W^6 + 49*W^4 + 735*W^2 + 3375))/(W^8 + 68*W^6 + 1606*W^4 + 15300*W^2 + 50625) \\ \text{MAT}(2,1) &:= (8*(W^6 + 49*W^4 + 735*W^2 + 3375))/(W^8 + 68*W^6 + 1606*W^4 + 15300*W^2 + 50625) \\ \text{MAT}(2,2) &:= (4*(W^6 + 59*W^4 + 1075*W^2 + 5625))/(W^8 + 68*W^6 + 1606*W^4 + 15300*W^2 + 50625) \end{aligned}$$

図8 行列 A_1 の各要素

$$\begin{aligned} \text{MAT}(1,1) &:= (9*(W^2 + 1))/(W^4 + 2*W^2 + 1) \\ \text{MAT}(1,2) &:= (3*(W^2 + 1))/(W^4 + 2*W^2 + 1) \\ \text{MAT}(2,1) &:= (3*(W^2 + 1))/(W^4 + 2*W^2 + 1) \\ \text{MAT}(2,2) &:= 1/(W^2 + 1) \end{aligned}$$

図9 行列 A_2 の各要素

$$\begin{aligned} \text{MAT}(1,1) &:= -0.627\ 01799\ 06 \\ \text{MAT}(1,2) &:= -0.316\ 22776\ 6 \\ \text{MAT}(2,1) &:= 0.779\ 00477\ 5 \\ \text{MAT}(2,2) &:= 0.948\ 68329\ 8 \end{aligned}$$

図10 前置補償器 G_c の各要素

4.2 圧延プロセスへの適用例^{8,9)}

次に、実プラントの例として、鉄鋼プロセスにおける圧延工程を考える。熱間薄板圧延においては、タンデムミルとよばれ、6または7段の圧延機より構成されたシステムで操作されているが、ここでは、そのうちの隣接する2スタンドに注目する。しかも、スタンド間にはルーパとよばれる隣接スタンドのバランサーがあるものとする。ルーパ制御は、2入力2出力系の張力制御とルーパ高さ制御として、これまで詳細に研究されてきた。今回は、さらに下流スタンドの板厚制御を組み込んだ系を考える。このとき、入力は上流スタンドのロール回転数の偏差 $\Delta V (= U1)$ 、ルーパの発生トルクの偏差 $\Delta T (= U2)$ 、下流スタンドのロール間隙の偏差 $\Delta G (= U3)$ であり、他方、出力はルーパ高さの偏差 $\Delta\theta (= Y1)$ 、材料張力の偏差 $\Delta\sigma_b (= Y2)$ 、下流スタンドの出側板厚の偏差 $\Delta h (= Y3)$ となる。3入力3出力系と考えられる。このとき、ブロック線図は図11に与えられる。ラプラス演算子 S 以外は全て定数であるが、各定数の物理的な意味はここでは略す。この系においては、内部変数として $X1$ から $X18$ を仮定する。伝達関数行列の計算結果を図12とブロック線図の記号との対照表を表2に示す。

表2 記号対照表

図11での記号	図12での記号
E	el
e	es
μ	mm
P_G	pg
M	ml
P_σ	ps

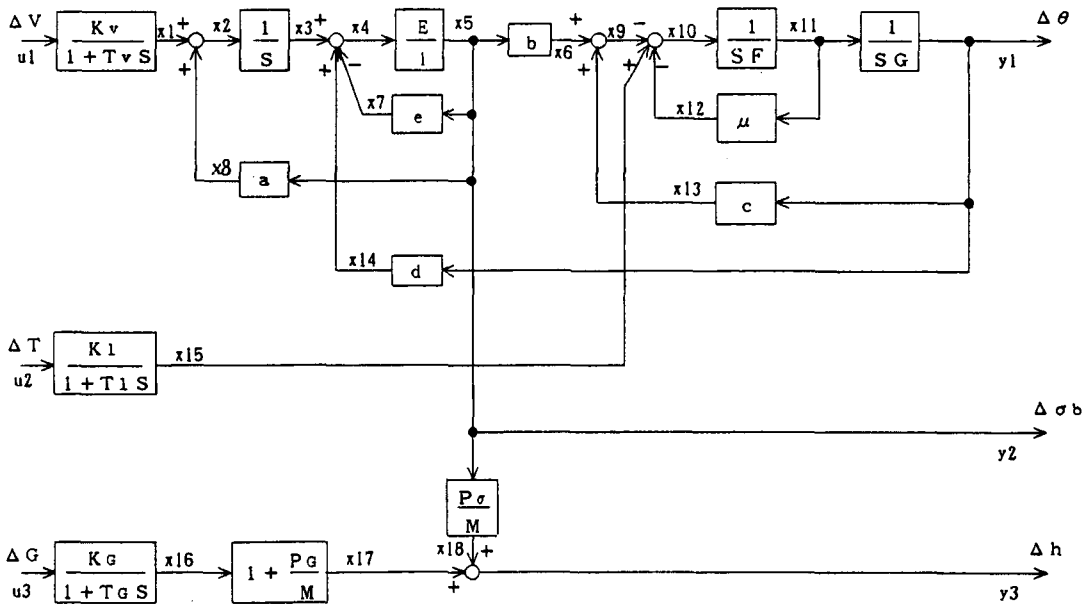


図11 熱延プロセスにおける3入力3出力系のブロック線図

$$\begin{aligned}
\text{MAT}(1,1) &:= (B*EL*KV)/(((F*S + MM)*G*S + C)*A*EL - (B*D*EL + F*G*L*S^2 + G \\
&\quad *L*MM*S)*S - (EL*ES + L)*C*S - (F*S + MM)*EL*ES*G*S^2) \\
&\quad *(S*TV + 1)) \\
\text{MAT}(1,2) &:= ((A*EL - EL*ES*S - L*S)*KL)/(((F*S + MM)*G*S + C)*A*EL - (B*D \\
&\quad *EL + F*G*L*S^2 + G*L*MM*S)*S - (EL*ES + L)*C*S - (F*S \\
&\quad + MM)*EL*ES*G*S^2)*(S*TL + 1)) \\
\text{MAT}(1,3) &:= 0 \\
\text{MAT}(2,1) &:= (-((F*S + MM)*G*S + C)*EL*KV)/(((F*S + MM)*G*S + C)*A*EL - \\
&\quad (B*D*EL + F*G*L*S^2 + G*L*MM*S)*S - (EL*ES + L)*C*S - (F* \\
&\quad S + MM)*EL*ES*G*S^2)*(S*TV + 1)) \\
\text{MAT}(2,2) &:= (-D*EL*KL*S)/(((F*S + MM)*G*S + C)*A*EL - (B*D*EL + F*G*L* \\
&\quad S^2 + G*L*MM*S)*S - (EL*ES + L)*C*S - (F*S + MM)*EL*ES \\
&\quad *G*S^2)*(S*TL + 1)) \\
\text{MAT}(2,3) &:= 0 \\
\text{MAT}(3,1) &:= (-((F*S + MM)*G*S + C)*EL*KV*PS)/(((F*S + MM)*G*S + C)*A*EL \\
&\quad - (B*D*EL + F*G*L*S^2 + G*L*MM*S)*S - (EL*ES + L)*C*S - \\
&\quad (F*S + MM)*EL*ES*G*S^2)*(S*TV + 1)*ML) \\
\text{MAT}(3,2) &:= (-D*EL*KL*PS*S)/(((F*S + MM)*G*S + C)*A*EL - (B*D*EL + F*G* \\
&\quad L*S^2 + G*L*MM*S)*S - (EL*ES + L)*C*S - (F*S + MM)*EL* \\
&\quad ES*G*S^2)*(S*TL + 1)*ML) \\
\text{MAT}(3,3) &:= ((ML + PG)*KG)/((S*TG + 1)*ML)
\end{aligned}$$

図12 熱延プロセスにおける3入力3出力系の伝達関数行列 G_p の各要素

4.3 まとめ

以上の結果より、多変数制御系においてブロック線図から、伝達関数行列表現を得るためには、数式処理システムは大変有用であることがわかった。4.2の例のように、人の手計算では大変複雑で、しかも誤りをおかしやすいが、本システムでは入力するための作業は必要であるが、素早く伝達関数行列の表現を得ることができる。だが、前置補償器を設計するとき、4.1の例のように計算途中までは記号を含んだ結果を得られるのであるが、最終的には、記号のままでは得られない。これは、今回設計に用いた逆ナイキスト配列法のアルゴリズムの中で、最小固有値を決定しなければならないため、どうしても、

数値の大小比較が避けられないためである。したがって非干渉化のアルゴリズムのなかで、大小比較等の数値処理を用いない方法を開発できれば、数式処理の適用範囲は増大する。

今後は、非干渉化された系の図的表現（逆ナイキスト線図および列ゲリシュゴリン帯の表示）や、主制御器 $K(s)$ を組み込んだときのシミュレーションによる応答表示等が可能なシステムに、発展させてゆく予定である。

参 考 文 献

- 1) 斎藤制海, 阿部健一, 高橋岳之; 計算機援用による制御工学, システムと制御, Vol. 29, NO. 12, 785 - 794, (1985).
- 2) 木村英紀; 多変数制御系のCAD, 計測と制御, Vol. 15, No. 12, 935 - 944, (1976).
- 3) 古田勝久, 梶原宏之; 制御系のためのCAD, 計測と制御, Vol. 18, No. 9, 777 - 786, (1979).
- 4) 斎藤制海, 高橋岳之, 平田富夫; 記号処理言語による制御系のCADシステム, シミュレーション, Vol. 3, No. 3, 126-133,(1984).
- 5) 村尾裕一; 数式処理用言語, 数理科学, No. 242, 13 - 21, (1983).
- 6) A.C. Hearn; REDUCE USER'S MANUAL Version 3.2, The Rand Corporation. (1985).
- 7) 伊藤正美, 木村英紀, 細江繁幸; 線形制御系の設計理論, 186 - 221, 計測自動制御学会.(1980).
- 8) 森田徹, 小西正躬, 北村章; 多変数制御を用いた板厚制御の非干渉化, R & D 神戸製鋼技法, Vol. 33, No.2, 51 - 55, (1983).
- 9) 小西正躬, 能勢和夫, 北村章, 中田隆正, 中島繁紀; 熱延仕上ミルにおけるループ系の非干渉制御技術, R & D 神戸製鋼技法, Vol. 36, No.3, 39 - 42, (1986).