

記号処理言語による制御系の CAD システム

齋藤制海・高橋岳之・平田富夫*

ABSTRACT Computer Aided Design (CAD) systems are an important tool when the modern control theory is applied to designing a controller of practical system. Recently various CAD systems of control systems have been developed. However all these systems are based on numerical calculations. Hence they are not available for the problem in which the parameters of mathematical models are given as symbols.

A CAD system which can handle symbolic parameters is necessary for the approach to the structural problem of control system. So the authors are developing a symbolic manipulation CAD system. This paper explains the outline of this system. First in section 1, the motivation is clarified by using simple electric circuits. Secondly section 2 treats the symbolic process language, LISP and symbolic and algebraic manipulation system, REDUCE 3. Thirdly section 3 describes how to construct the CAD system based on REDUCE 3. Finally section 4 deals with the actual examples and shows the availability and several points of this CAD.

1. 緒言

計測技術の発達, マイクロプロセッサの普及および理論体系の整備等により無用の長物といわれてひさしい現代制御論もいよいよ実用化の時代に入ったといわれるようになってきた¹⁾. 現代制御理論を用いた設計手法には複雑かつ膨大な計算が不可避である. したがって, 近年制御系の解析・設計のための計算機援用システム (CAD) が種々開発されている²⁻⁶⁾. しかしこ

これらのシステムは全て数値計算によるもので, 制御系の数学モデルを記号のまま記述することは不可能である.

いま制御対象として図1に示す電気回路を考える. 端子電流を入力 u , 端子電圧を出力 y , コイルに流れる電流 x_1 , コンデンサにかかる電圧 x_2 を状態変数とすると, 状態方程式および出力方程式は

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -r_1/L & 0 \\ 0 & -1/r_2 C \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} r_1/L \\ 1/C \end{bmatrix} u \quad (1)$$

$$y = \begin{bmatrix} -r_1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + r_1 u \quad (2)$$

で与えられる. (1)式のように制御系の数学モデルのパラメータが非数値で与えられるとき, 制御系が可制御であるためのパラメータ間の関係を調べてみよう.

可制御行列 V を計算すると

$$V = \begin{bmatrix} r_1/L & -r_1^2/L^2 \\ 1/C & -1/C^2 r_2 \end{bmatrix} \quad (3)$$

となる. したがってこの制御系が可制御であるための必要十分条件は

$$\det V = -\frac{r_1}{LC^2 r_2} + \frac{r_1^2}{L^2 C} \quad (4)$$

であり, これより r_1, r_2, L, C のパラメータ間に $C r_1 r_2 \neq L$

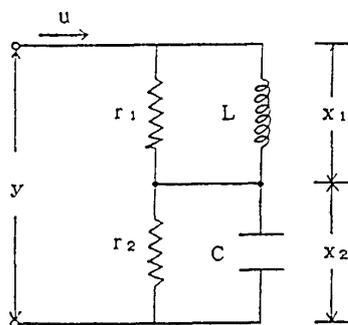


図1 2次系の電気回路

Computer Aided Design of Control System Using Symbolic Manipulation Language. By Osami Saito, Takeyuki Takahashi & Tomio Hirata (Toyohashi University of Technology)

* 豊橋技術科学大学情報工学系

なる関係を満たす必要があることがわかる。逆に(5)式の等号が成立つとき、すなわちインピーダンスが純抵抗となると制御系は非可制御となる。以上の計算は線形制御論を周知の人なら数分もあれば筆算で正確に解くことができるであろう。それでは次に図1と類似である図2の回路の場合を考えてみよう。

この系は4次系であり、そのシステム方程式は

$$\dot{x} = Ax + Bu \quad (6)$$

$$y = Cx + Du \quad (7)$$

$$A \triangleq \begin{bmatrix} -r_1/L_1 & 0 & 0 & 0 \\ 0 & -1/r_2 C_1 & 0 & 0 \\ 0 & 0 & -r_3/L_2 & 0 \\ 0 & 0 & 0 & -1/r_4 C_2 \end{bmatrix} \quad (8)$$

$$B \triangleq [r_1/L_1 \quad 1/C_1 \quad r_3/L_2 \quad 1/C_2]^T \quad (9)$$

$$C \triangleq [-r_1 \quad 1 \quad -r_3 \quad 1] \quad D \triangleq r_1 + r_3$$

となる。この4次系に対して上述の議論と同じようにパラメータ間の可制御条件を求めてみる。可制御行列

$$V = [B \quad AB \quad A^2B \quad A^3B] \quad (10)$$

はこの場合Aが対角行列であるので簡単に求めることができる。しかし(4×4)の密な行列であるVの行列式を筆算で求めるのは容易でないし、得られた結果を検算するのめかなりの手間をとる。

制御系の解析や設計を行なう際にはこの種の煩雑な非数値演算が必要な場合にしばしば出くわし、計算機を利用することを思い立つ。しかし最初にふれたように従来の制御系のCADシステムは数値計算によるものであり、非数値演算を必要とするこの種の問題には役立たない。

現在非数値演算の可能なシステムとしてオランダのTWENTE大学で研究開発された電気回路解析用システムNETFORMの報告があるが⁷⁾、この種の制御系専用CADはいまだみられない。そこで筆者らは非数値すなわち数式を記号のまままで処理することが可能な制御系のCADシステムを開発中であり⁸⁾、現在得られているシステムの概略を紹介する。非数値演算が可能な制御系のCADシステムが稼働すれば、従来のCADには見られなかった新しい利用分野が生まれ、現代制御論がより実用的になると考えている。本文が数式処理の応用および制御系のCADシステムに興味のある読者の一助になれば幸いである。

2. 数式処理システム REDUCE 3

数式を記号のままコンピュータで処理するには、数式の構造を記述するためのなんらかの表現形式(データ構造)が必要となる。現在もっとも広く用いられて

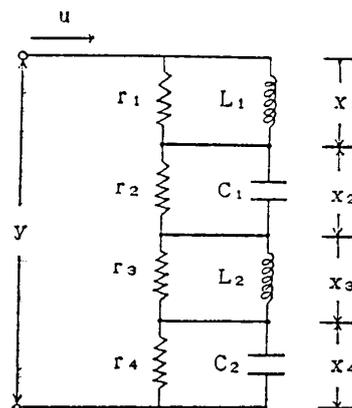


図2 4次系の電気回路

いるものはリスト(List)というデータ構造である。リストとは、ひとことで言えば、要素の並びである。たとえば、 $a+b$ という式は $(+ab)$ という要素の並び(リスト)で表現できる。リストの要素は“+”，“a”のような記号でもよいし、また別のリストでもよい。従って、 $a+b*c$ のような式は $(+a(*bc))$ という具合に表現できる。このようなリスト構造を扱うプログラミング言語に LISP (LIST Processor) がある。FORTRAN や ALGOL を使って数式処理を行なうことは無論可能であるが、これらは数値計算やアルゴリズム記述用に開発された言語であるため、記号処理を簡潔で能率よく記述するには不向きである。

LISP は MIT の McCarthy が1962年に⁹⁾人工知能(AI)向けの言語として開発したもので、FORTRANと同程度の歴史を持つが、近年エキスパートシステム、自然言語処理、数式処理などの記述言語として、第5世代計算機の主力言語 PROLOG とともに急速に注目されるようになってきた。当初 McCarthy によって開発された LISP は LISP 1.5 と呼ばれ、構造的にはすっきりしているが、プログラミングのし易さという点で問題があった。その後様々な改良が試みられ、現在ではいろいろな名前の LISP (いわゆる方言)がある。代表的なものは MACLISP (MIT), InterLISP (Xerox), FRANZ LISP 等である。

LISP は FORTRAN 等の言語にはない幾つかの特徴を持っている。第1に、LISP は関数型言語 (FORTRAN, ALGOL は手続き型言語である。)であり、プログラムは関数定義の集まりになる。その際、関数の再帰的定義が可能である。第2に、LISP は解釈型言語 (BASIC と類似している)であり、プログラムはすべて解釈実行系(インタープリタ)の下で逐次解釈されながら実行される。解釈実行系にとっては LISP に予め備わっている関数も新たに定義された関数も同

じであり、従って数式処理用の関数を LISP に付加していくことにより、LISP を数式処理システムに変えていくといったことが比較的簡単にできる。さらに、LISP 内部ではリストも関数定義もすべてセルというデータ単位によって表現されており、記憶領域は動的に管理されている。これらの詳細については文献^{10)~12)}を参照されたい。

さて計算機を利用して数式処理を行なう研究は MIT の Slagle による不定積分のプログラム SAINT が最初であり、その後天文学や高エネルギー物理学の方面で長大な計算の必要性から種々の数式処理システムが開発されて来た。しかしこれらは多項式演算が主で使い勝手もよくなく一般的な計算には役立たない。その後、上述の LISP 言語をもとに汎用性を持った数式処理システムが開発され出し、その代表的なシステムが MIT の MACSYMA 〔マサチューセッツ大学の REDUCE¹³⁾ および IBM の SCRATCHPAD などである。数式処理機能は MACSYMA が最強であるが、残念ながら簡単に手に入れることができない。一方 REDUCE は MACSYMA ほどの能力は持たないが、開発者 Hearn が普及に熱心でありわずかのコストでだれでも手に入れることができる。現在 REDUCE は第3版がリリースされており主要大学、研究機関等で使用可能である。ところで REDUCE は量子電気力学の数式計算の必要性に端を発して開発された数式処理システムであるが、大雑把にいて現在次の数式演算機能を持つ関数群が用意されている^{14),15)}。

- (1) 多項式・有理関数の展開及び整理
 - ・降べき、昇べき順への並べ換え
 - ・出力変数及び内部変数の並びの順序づけ
 - ・共通項のくくり出し
- (2) 約分、通分等の有理式の簡約化
- (3) 記号式のままの微分、例えば
 - ・ $d \sin x^2 / dx = 2x \cos x^2$
- (4) 種々な形の式に対するパターン照合及び代入
 - ・ 記号変数への多倍長整数、多倍長実数などの数値、あるいは別の数式の代入
 - ・ ある数式の他の数式への置き換え
- (5) ふたつの多項式の GCD (最大公約式) の計算
- (6) 行列の四則演算及び行列式の取り扱い
- (7) 多変数多項式の因数分解
- (8) 簡単な定積分、不定積分

ここでいう関数はちょうど FORTRAN におけるファンクションに対応し、引数 (非数値要素からなる行列

や係数が非数値である多項式でも可) を受け取り、その関数を評価 (実行) し、その結果を関数の値として返す機能を持っている。例えば表1の DET(A) という関数は行列Aを受け取り、その行列式の値を関数の値として返してくるというわけである。

これらの関数を記述するアルゴリズムは、多くの場合数値計算に使用したものをそのまま適用することはできない^{16),17)}。例えば二つの多項式の GCD を求めるのに数値解と同じ Euclid の互除法を適用すると、途中の係数の桁数が爆発的に増大するという数値計算では見られなかった新しい問題に直面する。そこでよく用いられているのが、素数 p を法とする 0 から $p-1$ までの有限体を適用するモジュラー計算法であり、このなかのひとつの Hensel の補題を応用することによって中間結果の爆発を回避している。このように効率的なアルゴリズムの開発が数式処理の大きなテーマのひとつであり^{18),19)}、現在一般的な初等関数の不定積分、極限值操作、さらに我々のシステム制御理論を取り扱う上で必要不可欠な常微分方程式や偏微分方程式の求解などの数式処理用アルゴリズムの研究が活発に行なわれている²⁰⁾。

3. 制御系の CAD

筆者らは、名古屋大学大型計算機センターの FACOM M-382 上で稼働している REDUCE 3 をベースに非数値演算が可能な線形制御系 CAD のシステムを作成中であり以下その概略を述べる。

制御系の CAD システムを作るには(1)~(8)までの REDUCE 3 の数式処理用の関数に線形制御系の解析、設計に必要な関数を付け加えていけばよいわけである。このとき新しい関数は RLISP で記述していく。図3は特性多項式を求める関数を RLISP で書いたプログラム例である。本 CAD システムは現在、多項式及び行列の演算と線形制御系の解析・設計用に表1に示す関数を備えており、各関数は備考に示す機能を持っている。表中 FACTORIZE から TRACE までの関数は REDUCE 3 がすでに持っているものでそれ以降は筆者らが開発した関数である。

```

PROCEDURE CHARACT A;
BEGIN SCALAR N1;
MATRIX !*A;
!*A := A;
N1 := COLUMN!* !*A;
!*A := - !*A;
FOR I:=1:N1 DO !*A(I,I):= S + !*A(I,I);
RETURN DET !*A;
END;
```

図3 特性多項式をもとめる関数のプログラム

数式処理の CAD には

- (I) ユーザが LISP をまったく意識しなくてすむ。
 - (II) 多項式及び行列の入力をやさしくする。
 - (III) 結果は出来るだけ手計算に近くわかりやすい形式をとる。
 - (IV) 処理速度をできるだけ速くする。
- というようなことが要求されるであろう。

(I)については、当初ユーザが安易にシステムを使用できるように関数をメニュー方式にするつもりであったが、これだと応用範囲が限定されてしまう。それに対して現在の関数を定義していく方法だと、CAD システムの実行中にユーザが下記のようなコマンドを入力すれば容易に新しい関数を定義することができる。

FORALL X, Y LET CHA=DET (CONTROL-LABLE (X, Y)); このように表 1 の関数群をもとにユーザが必要に応じて関数を新しく定義し、次々と関数を増殖したほうが応用範囲も広がると考えメニュー方式はとらなかった。しかしその反面、ユーザは現システムを使用するには少なくとも REDUCE が使用できる程度の知識を持っていなくてはならない。そこで今後はメニュー方式と現在の方式の二系統のシステムを構成する予定である。どちらにしても表 1 の関数の合成で表現できない関数は RLISP でプログラムしなくてはならず、出来るだけユーザの負担を軽くするために制御系の解析・設計上の種々の問題に対処できる関数を備えてゆく必要がある。

(II)については、多項式の入力は REDUCE 3 の書式に従うことにし、行列及びベクトルの入力には新しい関数を定義した。REDUCE では行列要素の入力は $A(i, j) := \dots$; という代入文を用いて行なうが、入力ミスが多くなるなど使いづらい。そこで対話的に要素の入力を行なう関数 MATIN を準備した。これを用いた一例を図 4 に挙げておく。この際、行列入力モードにおいて pp というコマンドを入力すれば入力中の行列が清書出力されるので入力ミスをチェックできる。なおユーザにとって入力の煩雑さはシステムの能力を半減させるので機能的なスクリーンエディタを開発中である。(III)については数式の出力結果を筆算の形式に合わせることは一見簡単に思えるが、実はなかなか大変な問題である。たとえば $(x^5-1)/(x-1)$ を約して $x^4+x^3+x^2+x+1$ とするより前者のままの方が望ましいし逆に $(x^2-1)/(x+1)$ は $x-1$ と簡約化した方がよい。数式の表現は一通りではなくどれが

表 1 関数群とその機能

関数名	入力	出力	備考
FACTORIZE	U	多項式	因数分解
DET	A	A	行列式
+, -, *	A, B	A+B, A-B, A*B	四則演算
**(-1)	A	A ⁻¹	逆行列
TP	A	A ^T	転置行列
TRACE	A	tr A	対角和
ADJ	A, i, j	adj A _{ij}	余因子行列
PARTIAL!-EXPANSION	U, S	展開された式	部分分数展開
MATIN IN	A		行列要素の入力
PP	A		行列要素の清書印刷
MPRINT	A		行列要素の清書印刷 (L P)
CHARACT	A	SI-A	特性多項式
TRANSFER[TRANS]	A, B, C	C(SI-A)*B	伝達関数行列
CONTROLLABLE[CNT]	A, B	[B AB ... A B]	可制御行列
OBSERVABLE[OBS]	A, C	[C A C ... A C]	可観測行列
CANONICALCNT	A, B	$\bar{A}, \bar{B}, \bar{C}, T, T^*$	可制御正準形
CANONICALOBS	A, C	$\bar{A}, \bar{B}, \bar{C}, T, T^*$	可観測正準形
.	.	.	.
.	.	.	.

MATIN A(2,2): [A (2×2) の行列の入力]
 MIN> LL1 [コマンド (行列入力モード)]
 A(1,1) := 0 [(1, 1) 要素の現在の値]
 ? -RL1 [(1, 1) 要素の入力]
 A(1,2) := 0
 ?
 .

(<下線部が入力である)

図 4 MATIN による行列の入力例

一番有用であるか計算機には判断できない。現在多項式の出力は REDUCE の方式を踏襲しているが、行列とベクトルをごく自然の形式で出力させるため行列の清書印刷関数を用意した。実際の出力結果は次章の実行例を参照していただきたい。

(IV)については、数式処理システムはミリ秒オーダーの計算機といわれその計算速度の遅さが批判されている。既に述べたように LISP は解釈実行型の言語であるが、コンパイラが開発され計算速度はかなり速くなった。それでも数値計算用計算機に数式処理を行なわせてもその速度にはおのずと限界がある。したがって数式処理専用の計算機を開発すれば計算速度は飛躍的に向上する。現在理化学研究所の FLATS や計算機メーカーなどで LISP 専用マシンの開発が進められている。本学においても情報工学系の計算機アーキテクチャの研究グループがデータフローマシン型の LISP マシンを開発している²¹⁾。本システムを移植することにより演算スピードの改善が期待できる。

4. CAD システムの実行例

本章では 3 章で述べた数式処理による制御系の CAD システムの 2 つの実行例を示しその有効性及び

```

MATRIX CNT;
CNT := CONTROLLABLE(A,B)V
MPRINT CNTV
OUTPUT COMPLETED

MATRIX CNT =
R1/L1 (- R1 )/L1 R1 /L1 (- R1 )/L1
1/C1 (- 1)/(R2+C1) 1/(R2 = C1) (- 1)/(R2 = C1)
R3/L2 (- R3 )/L2 R3 /L2 (- R3 )/L2
1/C2 (- 1)/(R4+C2) 1/(R4 = C2) (- 1)/(R4 = C2)
    
```

図 5(a) 4次系回路の可制御行列

```

M1 := FACTORIZE( NUM(P), AB1)V
FOR I:=0:M1 DO WRITE AB1(I):=AB1(I);

AB1(1) := R4=C2=R3 - L2 ----- ①
AB1(2) := R4=C2=R1 - L1 ----- ②
AB1(3) := R4=C2 - R2=C1 ----- ③
AB1(4) := L2=R1 - R3=L1 ----- ④
AB1(5) := L2 - R3=R2=C1 ----- ⑤
AB1(6) := R3
AB1(7) := R2=C1=R1 - L1 ----- ⑥
AB1(8) := R1
    
```

図 5(c) 行列式の因数分解の結果

```

P := DET CNT;
P := (R3+R1)(R4+C2=L2+R3+R2+C1+R1 - R4=C2=L2+R3+L1+R1 - R4=C2=L2+R3+R2 =
C1=R1 + R4=C2=L2+R3+L1+R1 + R4=C2=R3+R2=C1=L1+R1 - R4=C2 =
R3+R2+C1=L1=R1 - R4=C2=L2+R2+C1+R1 + R4=C2=L2+L1+R1 + R4=C2 =
L2+R3+R2=C1+R1 - R4=C2=L2+R3=L1 - R4=C2=R3+R2=C1=L1+R1 +
R4=C2=R3+R2+C1=L1 + R4=C2=L2+R2=C1=R1 - R4=C2=L2=L1+R1 - R4=C2
+L2+R3+R2=C1+R1 + R4=C2=L2=R3=L1 + R4=C2+R3+R2=C1=L1=R1 - R4=
C2+R3+R2=C1=L1 - L2+R2=C1=L1+R1 + L2+R2=C1=L1+R1 + L2+R3+R2 =
C1=L1+R1 - L2+R3+R2+C1=L1 - L2+R3+R2=C1=L1+R1 + L2+R3+R2=C1 =
L1)/(R4=C2=L2+R2=C1=L1)
    
```

図 5(b) 可制御行列の行列式

の REDUCE の多項式の表現法にも起因する。例えば $a+b+c+d/fg$ という数式の表現と $(afg+bfg+cfg+d)/fg$ なる表現を比較したとき人間にとっては前者のほうが自然の表現と言えるが、計算機は後者の形式で出力を行なう。図 5(b) の結果も分母 $R_4^3 C_2^4 L_2^4 R_3^2 C_1^3 L_1^4$ で分子の多項式を割ればもう少しすっきりした形で出力されるが現在計算機にはそこまで判断できない。

さて $\det V$ の結果は長大な多項式となり、このままでは非可制御となるパラメータ間の関係に関する有効な情報は得られない。そこで $\det V$ の結果を因数分解すると (関数 FACTORIZE) 図 5(c) の出力を得る。すなわち図 5(b) の多項式の分子は図 5(c) に示すように①～⑥の6個の素多項式に分解される。⑥はちょうど2次系回路における(5)式に対応し、かつ①と対称である。また②と⑤、③と④が対称の関係にあることがわかる。①～⑥のいずれかが0になるパラメータ値のとき $\det V=0$ となり、制御系は非可制御となる。このように本 CAD システムを用いることにより1章でふれた4次系の電気回路の可制御条件が極めて明確に得られた。なお図5において実線で囲んだ部分がユーザが入力するコマンドであり、簡潔なプログラムになっているのがわかる。

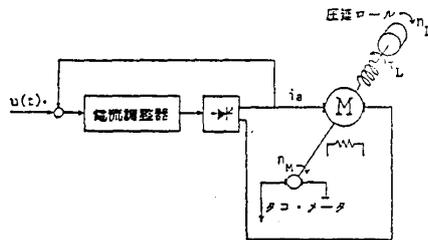


図 6(a) 圧延ミル

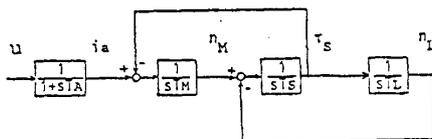


図 6(b) 圧延ミルのブロック線図

問題点を具体的に示す。

4.1 4次系回路の可制御性

第1章で述べた4次系の電気回路の可制御性を本 CAD システムを用いて調べてみる。まず MATIN 関数をもちいて(8), (9)式の行列 A, B を入力する。ついで可制御行列 V を関数 CONTROLLABLE を用いて計算する。図 5(a) がこのときの (4×4) の密なる行列 V の清書印刷である。つぎに V の行列式を関数 DET を用いて求めると、図 5(b) のような長大な多項式が出力される。結果がこのように長大になったのは行列式そのものが複雑であることにもよるが、現在

4.2 圧延ミルの解析

上述の例で記号処理による CAD システムの概念が理解されたと思うが、本 CAD システムの有効性をより深く理解してもらうために以下全般的な実行例を示す。制御対象としては図 6(a) で与えられるような他励直流機に負荷である圧延ロールがシャフトで弾性的に結合された圧延ミルを取り上げる。この制御系のブロック線図は図 6(b) で与えられ、そのシステム方程式は

$$\dot{x} = Ax + Bu \tag{11}$$

$$P := (S^4 * TA * TM * TS * TL + S^3 * TM * TS * TL + S^2 * TA * (TM + TL) + S * (TM + TL)) / (TA * TM * TS * TL)$$

図 7(a) 圧延ミルの特性多項式

$$P := (S^2 * (S * TM * TS * TL + TM + TL) * (S * TA + 1)) / (TA * TM * TS * TL)$$

図 7(b) 特性多項式の因数分解の結果

$$G := 1 / (S^2 * (S * TM * TS * TL + TM + TL) * (S * TA + 1))$$

図 8(a) 圧延ミルの伝達関数

$$\begin{aligned} \text{PART-FRAC(2)} &:= 1 / (S * (TM + TL)) \\ \text{PART-FRAC(3)} &:= (-TA) / ((S * TA + 1) * (TA * TM + TA * TL + TM * TS * TL)) \\ \text{PART-FRAC(4)} &:= (- (S * TM * TS * TL + TA * TM + TA * TL) * TM * TS * TL) / ((S * TM * TS * TL + TM + TL) * (TA * TM + 2 * TA * TM * TL + TA * TL + TM * TS * TL + TM * TS * TL^2)) \end{aligned}$$

図 8(b) 伝達関数の部分分数展開

$$y = Cx \quad (12)$$

$$x \triangleq [n_L \quad \tau_r \quad n_M \quad i_a]^T$$

$$A \triangleq \begin{bmatrix} 0 & 1/T_L & 0 & 0 \\ -1/T_S & 0 & 1/T_S & 0 \\ 0 & -1/T_M & 0 & 1/T_M \\ 0 & 0 & 0 & 1/T_A \end{bmatrix}$$

$$B \triangleq [0 \quad 0 \quad 0 \quad 1/T_A]^T \quad C \triangleq [1 \quad 0 \quad 0 \quad 0]$$

ここで n_L , n_M はロール及び電動機の回転速度, τ_r はトルク, i_a は電機子電流, T_L , T_S , T_M はロール, パネ, 直流機の機械的時定数, T_A は電流調整機の回路時定数である。

[a] 特性多項式

システム行列 A の特性多項式 $p(s)$ は

$$p(s) = \det(sI - A) \quad (13)$$

で定義される。本 CAD システムには(13)式を求める関数 CHARACT が準備されている。(11)式の A を入力し CHARACT を用いて圧延ミルの特性多項式を求めると s の降べき順の多項式として図 7(a) のように出力される。 $p(s)$ を関数 FACTORIZE を用いて因数分解した結果が図 7(b) である。これより特性方程式の根は $S=0$, $S=1/T_A$, $S=\pm\sqrt{T_M+T_L/(T_M T_S T_L)}i$ となり、圧延ミルのシステムは機械時定数 T_M , T_L , T_S がいかなる値をとっても虚軸上に3つの極が存在する構造をしていることがわかる。ただし本 CAD では多項式の平方根や複素根をもつ因数分解は不可能で、この場合 $s^2(T_M T_S T_L) + T_M + T_L$ の2次式の素多項式で出力される。したがって圧延ミルが $\pm\sqrt{T_M+T_L/(T_M T_S T_L)}i$ なる虚根を持つことは図 7(b) の結果より人間が判断してやる必要がある。本来 LISP は会話性にすぐれた言語であるので今後 CAD システムと人間とが結果を見ながら会話できるようにすれば一層強力になると考えられる。

[b] 伝達関数 $G(s)$ の導出

状態方程式より伝達関数 $G(s)$ は

$$G(s) = C(sI - A)^{-1}B$$

で求められる。伝達関数を求める関数 TRANSFER に(11), (12)式の A, B, C を代入すると $G(s)$ は図 8(a) のように S の4次多項式として得られる。さらにこれを部分分数に展開した結果が図 8(b) である。これよりこの制御系は PART-FRAC(2) から PART-FRAC(4) までの伝達関数で表わされる3つの系の並列接続でも記述できることがわかる。

[c] 可制御行列と可制御正準形の導出

可制御行列 V を関数 CONTROLLABLE を用いて求めると図 9(a) のような出力が得られる。この V の行列式は図 9(b) であり、圧延ミルのシステムは1章の電気回路の場合と異なりパラメータが非零ならば必ず可制御となる構造をしていることがわかる。

正準形は制御系の解析・設計に極めて有用かつ重要な数式モデルの記述形式であり、(11), (12)式の可制御正準形を導出する²²⁾。すなわち可制御行列 V の逆行列

$$V^{-1} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{bmatrix} \text{ を求め } T^{-1} = \begin{bmatrix} l_1 A \\ l_1 A^2 \\ l_1 A^3 \end{bmatrix}$$

とすると、 $(T^{-1}AT, T^{-1}B)$ が可制御正準形となる。本 CAD はこの可制御正準形を直接求める関数 CANONICALCNT を持っており図 9(c), (d) が正準形 \bar{A}, \bar{B} の出力である。このときの一次変換行列 T は図 9(e) である。なお可観測正準形も関数 CANONICAL OBS を用いて同様に求めることができる。

[d] 極配置問題

可制御な制御系に状態フィードバックを施すと任意

MATRIX	CNT	=	
0	0	0	$1/(TA+TM+TS+TL)$
0	0	$1/(TA+TM+TS)$	$(-1)/(TA+TM+TS)^2$
0	$1/(TA+TM)$	$(-1)/(TA+TM)^2$	$(-TA+TM+TS)/(TA+TM+TS)^3$
$1/TA$	$(-1)/TA^2$	$1/TA^3$	$(-1)/TA^4$

図 9(a) 圧延ミルの可制御行列

$$P := \text{DET CNT};$$

$$P := 1/(TA+TM+TS+TL)^4$$

図 9(b) 可制御正準形の駆動行列

MATRIX	CNA	=	
0	1	0	0
0	0	1	0
0	0	0	1
0	$(-TM+TL)/(TA+TM+TS+TL)$	$(-TM+TL)/(TM+TS+TL)$	$(-1)/TA$

図 9(c) 可制御正準形のシステム行列

MATRIX	CNB
0	
0	
0	
1	

図 9(d) 可制御正準形の駆動行列

MATRIX	TT	=	
$1/(TA+TM+TS+TL)$	0	0	0
0	$1/(TA+TM+TS)$	0	0
$1/(TA+TM+TS+TL)$	0	$1/(TA+TM)$	0
0	$(TM+TL)/(TA+TM+TS+TL)$	0	$1/TA$

図 9(e) 変換行列 T

以上パラメータが非数値で与えられた制御系への本 CAD の応用例を示した。もちろん本 CAD はパラメータが数値で与えられる制御系にも適用できる。この場合従来の CAD に比べて計算速度は落ちるが、丸め誤差や桁落ちがないという利点が生じる。例えば伝達関数や極配置問題などの数値解法のアルゴリズムは Faddev や正田らのアルゴリズム³⁾を適用するわけであるが、これらは行列のべき乗や逆行列計算を含むので誤差への十分な配慮が必要である。一方数式処理の CAD では計算の過程での誤差はなく、極のバラツキが大きいなど誤差解析からみて微妙な制御系にも有効である。

5. 結 言

以上数式処理による制御系の CAD はまだ現在開発途中であるが、数式処理は研究の段階から実用への時代に来たと感じ、あえて本文を書かせていただいた。数式処理による CAD は数値計算では求めることのできない解析的な解を求めることができる。解析解は数値解に比べて多くの情報を含み、従来の数値解法による CAD にはない種々の利点がある。例えば制御系の構造まで立ち入った解析・設計が可能であり、解析解を求めるため結果に丸め誤差や桁落ちがない。また、極配置のところでも示したようにごく自然のアルゴリズムで設計が可能であるなどである。しかし数式処理の関数機能は当初に比べて強力になってきたが、数値解法に比較すれば制御系の解析・設計にはまだ不十分である。特に数式による常微分方程式の求解、行列の階数の決定などの関数の開発が必要である。

LISP は推論マシンの言語として近年注目を集めて

の極配置が可能なのはよく知られた問題である。圧延ミルは虚軸上に3つの極を持つので、これらを左半面に移動する必要がある。[c]の結果で示したようにパラメータが非零ならば可制御であるので状態フィードバックによって任意の極配置が可能な制御系である。ここではこのときのフィードバックゲインKがみたすべき条件を導出してみる。いま(11)式のシステムに状態フィードバック

$$u = Kx, \quad K \triangleq [K_1 K_2 K_3 K_4] \quad (15)$$

を施すと閉ループ制御系は

$$\dot{x} = (A+BK)x \quad (16)$$

となる。任意に与えられた4個の実軸に対称な複素数 $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ を根にもつ s の多項式は

$$\begin{aligned} \phi(s) &= (s-\lambda_1)(s-\lambda_2)(s-\lambda_3)(s-\lambda_4) \\ &= s^4 + a_3s^3 + a_2s^2 + a_1s + a_0 \end{aligned} \quad (17)$$

である。一方(23)の特性方程式

$$\det(sI-A-BK) \quad (18)$$

を関数 CHARACTER で求めてみると図10の結果を得る。これより特性方程式の s の各次数の係数は K_1, K_2, K_3, K_4 の一次式となることがわかる。したがって各係数が(17)式の係数 a_3, a_2, a_1, a_0 に等しくなるようにゲインKを連立一次方程式で決定してやれば閉ループ制御系の極は $\lambda_1 \sim \lambda_4$ の位置に配置される。

$$P := (S^4 * TA * TM * TS * TL + S^3 * TM * TS * TL * (-K4 + 1) + S^2 * (-K3 * TS * TL + TA * TM + TA * TL) + S * (-K4 * TM - K4 * TL - K2 * TL + TM + TL) - (K3 + K1)) / (TA * TM * TS * TL)$$

図 10 閉ループ系の特性方程式

いる。本 CAD システムも単に制御系の設計の援用だけでなく、制御理論の新しい定理の証明や設計法の開発の補助手段としての利用法もあると考えている。現実に MACSYMA を利用してロバストなオブザーバの設計法の開発の補助手段に使用した報告も見られる²³⁾。今後これらの延長線上に文献や開発事例、特許などの知識データベースとリンクしたエキスパート(専門家)システムへの発展も模索してゆきたい。最後に本文を書く機会を与えてくださった東京工大古田勝久教授に深謝いたします。

参 考 文 献

- 1) 制御理論実用化への道, 計測と制御, 23, 9, 516/524 (1984)
- 2) 木村: 多変数制御系の CAD, 計測と制御, 15, 12, 935/943 (1976)
- 3) 古田, 梶原: 制御系のための CAD, 計測と制御, 18, 9, 777/786 (1979)
- 4) 高橋安人編著: 自動制御計算法, オーム社 (1982)
- 5) K. J. Astrom: Computer Aided Modeling, Analysis and Design of Control Systems-A Perspective, IEEE Control Systems Magazine, May 4/16 (1983)
- 6) D. K. Frederick et al.: Computer-Aided Control System Analysis and Design Using Interactive Computer Graphics, IEEE Control System Magazine, December 19/23 (1982)
- 7) 佐々木: 計算機による数式処理, システムと制御, 26, 4, 201/208 (1982)
- 8) 小松, 斎藤, 北橋: LISP による制御系の CAD システム, 第25回自動制御連合講演会前刷, 471/472 (1982)
- 9) McCarthy et al.: LISP 1.5 Programmer's Manual, MIT Press (1962)
- 10) 黒川利明: LISP 入門, 培風館 (1982)
- 11) 中西正和: 記号処理, 朝倉書店 (1978)
- 12) P. H. Winston: LISP, Addison-Wesley (1981)
- 13) A. C. Hearn: REDUCE 2 User's Manual, U. of Utah UCP-19 (1973)
- 14) 金田: 数式処理とは, 数理科学 8月号, 5/12 (1983)
- 15) 村尾: 数式処理用言語—REDUCE 2—, 数理科学 8月号, 13/21 (1983)
- 16) 後藤, 佐々木: 計算機による数式処理の現状, 情報処理, 18, 8, 830/837 (1977)
- 17) 佐々木, 後藤: 計算機による数式処理と物理学への応用, 日本物理学会誌, 35, 2, 99/107 (1980)
- 18) 佐々木建昭: 数式処理, 情報処理学会(オーム社) (1981)
- 19) 岩波講座 情報科学24巻
- 20) B. D. Saunders: An Implementation of Kovacic's Algorithm for Solving Second Order Linear Homogeneous Differential Equation, Proceeding of the 1981 ACM Symposium on Symbolic and Algebraic Computation, 105/108 (1978)
- 21) 飯田他: 並列処理機能を有する LISP マシンの開発, 知識工学と人工知能, 35-8 (1984)
- 22) 古田, 佐野: 基礎システム理論, コロナ社 (1978)
- 23) J. C. Doyle et al.: Robustness with Observer, IEEE Trans. on Automatic Control, AC-24, 4, 607/611 (1979)