

数式処理系 REDUCE を用いた 制御系の CAD

斎藤制海 菅野幹人 阿部健一

(豊橋技術科学大学)

1. 緒言

最適レギュレータやオブザーバなどの線形制御理論を中心とする制御系の解析、設計技法はほぼ完成の域に達している。これらは計測技術やマイクロプロセッサなどのハードウェアの発達に伴い、産業用ロボット、宇宙航空産業を始め、種々の分野で実用化されるようになってきた。これらの技法を用いた制御系の設計には、ベクトルや多項式演算を始めとする多大な計算を必要とするため、計算機援用設計 (CAD) システムが不可欠となっている。

近年、制御系の CAD システムは BASIC 言語で記述された比較的小規模なものから FORTRAN 等で記述された大規模なものまで種々開発され実用に供している。これらはすべて数値演算によるものであり、制御系のパラメータが記号のままでは与えられた場合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は処理したほうが都合の良い場合が見受けられる。例えば構造可制御性など generic な性質を論じたり、ロバスト性やパラメータ感度を調べたりする場合である。この種の問題に対して従来の数値計算による CAD システムは不向きであり、記号処理に適した CAD システムを必要とする。

本稿では、現在筆者らが開発を進めている記号処理が可能な制御系の CAD システム COUGAR-1 を 2、3 の実行例をまじえて報告する。さらに、現在主流の数値計算による CAD システムとの関係を明らかにするために数値-数式ハイブリッド CAD システムについても報告する。

2. 汎用数式処理システム

計算機を用いて数式処理を行なう研究は、MIT の Slagle による不定積分プログラム SAINT が最初であり、現在では様々な汎用の数式処理システムが開発されている。それらの多くは、数式を扱いやすい

データ形式を持っている LISP 言語によって記述されている。代表的なシステムをあげると、MIT の MACSYMA, Utah 大学の REDUCE、Cambridge 大学の CAMEL, および、IBM の SCRATCHPAD などがある。なかでも、REDUCE は開発者の Hearn が普及に熱心であり、わずかなコストで入手できるので世界的に流布している。現在 REDUCE は第 3.2 版がリリースされており、国内でも主要大学、研究機関などで使用可能である。また、REDUCE はパソコンの PC-9801 に移植できるといふ報告もあり、電卓の如く使用できるようになってきた。

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現在、大雑把に次いで示す 8 つの数式演算機能が用意されている。

- (1) 多項式・有理関数の展開及び整理
- (2) 約分、通分等の有理式の簡略化
- (3) 記号式のままの微分
- (4) 種々の形の式に対するパターン照合及び代入
- (5) 2 つの多項式の GCD (最大公約数) 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) 簡単な不定積分

また、汎用数式処理システムとしては REDUCE の他に MACSYMA がよく知られている。これは REDUCE より機能が一段と高く、上述の機能の他にさらに次の 5 つの機能を有している。

- (9) 極限の計算
- (10) 非線形方程式の解法
- (11) 微分方程式の解法
- (12) Laplace 変換
- (13) 2次元、3次元のプロット

ただし現在は、利用できる計算機は VAX など数種に限られている上に配付価格も高いので日本ではまだ数サイトにしか導入されていない。

今後、数式処理システムは一層高機能かつ身近な存在になると思われる。

3. 数式処理による制御系のCAD

筆者らは、名古屋大学大型計算センターのFACOM M-382 上で稼働しているREDUCE 3.1 (以下、単にREDUCEと書く) をベースに、記号処理演算が可能な制御系のCADシステム COUGAR-1 (Control system design Using General Algebraic language Reduce 1) を作成中であり、現在表1～表3に示す関数を有している。本章ではCOUGAR-1の構築方法、及び設計の際考慮すべき点について説明する。

3-1. COUGAR-1の構築

REDUCEはSLISP(STANDARD LISP)によって数式を処理している。しかし、REDUCEはカッコの羅列であるSLISP で記述されているのではなく、RLISP という中間言語(シンタックスシュガー)によって、その大部分が記述されている。RLISP はPASCAL風の言語であり、SLISP に比べて可読性が高く、プログラムの生産性を上げることができる。

そこで、COUGAR-1の記述は数式や行列のデータ構造をREDUCEが扱っている形式に統一し、RLISP で行なうことにした。(図1は行列の余因子行列を求めるプログラムをRLISP で記述した例である。)これにより、前章で述べたREDUCEが保有している8つの機能をそのまま使用することができることになる。あとは制御系に必要な関数群をRLISP で記述していけば自然にCADシステムが構築できる。COUGAR-1のシステム構成図を図2に示す。

3-2. COUGAR-1の設計指針

本CADシステムの開発にあたり、次の(I)～(III)を設計指針として与えることにした。

- (I) エンドユーザがLISP及びREDUCEをまったく意識しないで基本的な制御系の設計、解析ができるようにする。
- (II) 線形制御系では多項式や行列を取り扱うことが頻繁に起こるのでこれらの入力を簡単に誤りなく行なえるようにする。
- (III) 出力結果をできるだけ手計算の結果に近くわかりやすい表現とする。

```

FLAG( ' (ADJ), 'MATFLG);
PUT( 'ADJ, 'MSIMPFN, 'ADJ);
SYMBOLIC PROCEDURE ADJ(UU, I, J);
BEGIN SCALAR U, V;
  U := MATSM UU;
  V := DELNTH(I, U);
  RETURN FOREACH Y IN V COLLECT DELNTH(J, Y);
END;

```

図1 余因子行列を求めるプログラム

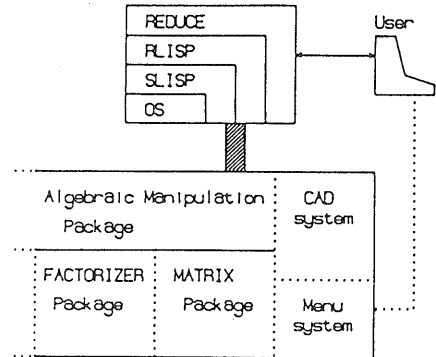


図2 COUGAR-1のシステム構成図

(I) についてはREDUCEをまったく知らないユーザでも簡単にCADシステムを使用できるようにメニュー方式を導入することにした。これは関数の呼び出しから、関数の実行までを会話的に行なうものである。

(II) については、多項式の入力はREDUCEの仕様に従うことにし、行列要素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、

$$A(i, j) := \dots \dots$$

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。そこで行列要素の入力や変更を行なう関数MATINを作成した。この関数は行列要素に対するエディターとなっており、行列要素の入力を行ない、清書出力により要素のチェックができるようになっている。

(III) の数式の出力結果を手計算の結果に合わせることは簡単と思われるが、実はなかなか大変な問題である。

例えば $(x^5 - 1)/(x - 1)$ を約して $x^4 + x^3 + x^2 + x + 1$ とするより前者の方が望ましい場合がある。このように数式の表現は一通りではなくどれが一番有用なのかは現在の計算機では判断できない。今のところ数式の出力はREDUCEの方式を踏襲しており、ユーザ

現在まで実現されている関数

表1 基本関数

関数名	入力	出力	機能
FACTORIZE	多項式 U	分解された多項式	因数分解 (整係数)
GCD	多項式 U, V	多項式	最大公約多項式
DET	正方行列 A	A	行列式の値
+, -, *	行列 A, B	A+B, A-B, A*B	四則演算
**(-1)	正方行列 A	A ⁻¹	逆行列
TP	行列 A	A ^T	転置行列
TRACE	正方行列 A	tr A	対角和
ADJ	行列 A 行, 列 i, j	adj A _{ij}	余因子行列
P!-EXP	多項式 U, 主変数 S	展開された式	部分分数展開 (分母を因数分解して、未定係数法で
MACLAURIN	多項式 U	展開された式	マクローリン展開 各因子を求める)
MATIN	行列名 A		行列要素の入力
PPRINT	行列, 配列名 A		行列, 配列要素の清書印刷
MPRINT	行列, 配列名 A		行列, 配列要素の清書印刷 (L P への出力)
MAT!-TRANS	行列 A	基本変換行列	行列の基本変換
JACOBI	有理関数 U, 主変数 V	ヤコビ行列	ヤコビ行列を求める

表2 解析用関数

関数名	入力	出力	機能
DETCARACT	正方行列 A	det S - A	特性多項式
TRANSFER	システム行列 A, B, C, D	C (S - A) ⁻¹ B + D	伝達関数行列
CONTROLLABLE	行列 A, B	[B AB ... A ⁿ⁻¹ B]	可制御行列
OBSERVABLE	行列 A, C	[C CA ... CA ⁿ⁻¹]T	可観測行列
CANONICALCNT	行列 A, B, C	Ā, B̄, C̄, T, T*	可制御正準形 (1 次変換行列も出力する)
CANONICALOBS	行列 A, B, C	Ā, B̄, C̄, T, T*	可観測正準形
BILINER!-TRANS	有理式 U, 主変数 V	変換された式	双一次変換
FREQUENCY	伝達関数 G(s)	G(jw) , arg G(jw)	ナイキスト及びボード線図も描く
FLOW!-CHART	結線行列 A	伝達関数 G(s)	シグナルチャートから伝達関数を求める
HURWITZ	行列 A, 特性多項式	フルビッツ行列 H	フルビッツの安定判別
JURY	行列 A, 特性多項式	ジュリーの表	ジュリーの安定判別
DESCRIPTOR	行列 A, B, C, D	det D = 0 の時 行列 A, B, C	DESCRIPTOR表現から状態方程式の導出
BODE!-OUT	有理関数 W	FORTRAN ソースプログラム	ボード線図を描く
NYQUEST!-OUT	有理関数 W	FORTRAN ソースプログラム	ナイキスト線図を描く
GRAPH!-OUT	有理関数 W, 主変数 V	FORTRAN ソースプログラム	グラフを描く

表3 設計用関数

関数名	入力	出力	機能
POLE!-ASSIGN	行列 A, B、極	フィードバック行列 K	極配置問題を解く
EXACT!-M!-MATCH	プラントとモデル	補償器	Exact Model Matchingを対話形式で解く
LONG!-DEN(NUM)	伝達関数 G(s)	G(s)の分子 (分母) 系列	伝達関数の分子 (分母) 系列表現
COPRIME	2つの素多項式 A, B	多項式 X, Y	A*X+B*Y = 1 なる X, Y の導出
DIOPHANTINE	多項式行列 X, Y, W	多項式行列 P, R	P*X+R*Y = W なる P, R の導出
DEVISER	有理関数行列 G(s)	G(s) = A**(-1)*B	多項式行列の分解表現
MFD!-L(R)	有理関数行列	多項式行列	左 (右) 既約分解
CON-FLA-EX	伝達関数	連分数展開形	連分数展開
!*MIN!-REAL!-Z	伝達関数行列、主変数 V	システム行列 A, B, C	最小実現

まず MATIN関数を用いて(5),(6)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4がこのときの4×4の密なる行列の消書出力結果である。この問題の場合Vは正方行列であるので $\det V = 0$ になれば

$$\text{rank } V < 4$$

となり、このシステムは非可制御となる。そこで関数 DETを用いてVの行列式を求めると、図5のような長大な多項式が出力される。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ることができないので関数 FACTORIZEによって因数分解すると図6の出力を得る。すなわち図5の多項式の分子は図6①～⑥の6個の素多項式に分解される。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、図3の電気回路の可制御条件が極めて明確に得ることができた。図4～6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、また行列の消書出力はごく自然な形式になっていることが理解できる。

4-2. 伝達関数行列の既約分解

伝達関数行列の既約分解は多入力・多出力系の解析、設計には欠かせない操作の一つである。この種の問題は記号Sを変数とする多項式の演算を種々行なうので、従来の数値計算より数式処理の方が格段に取り扱いやすい。

伝達関数の既約分解は大雑把にいうと次のアルゴリズムで得られる。

$H(S)$ を $m \times r$ のSの有理関数行列とする。このとき $H(S)$ は2つの多項式行列 $AL(S), BL(S)$ を用いて

$$H(S) = AL(S)^{-1} \cdot BL(S) \quad (9)$$

と分解される。 $AL(S), BL(S)$ はそれぞれ $m \times m, m \times r$ 行列であり、

$$AL(S) = \text{diag} [P_1(S), P_2(S), \dots, P_m(S)] \quad (10)$$

ここで $P_i(S)$ は第i行の最小公倍数多項式である。

一般に $AL(S), BL(S)$ は互いに素ではない。そこで $AL(S), BL(S)$ の両者に最大公約多項式行列 $G_1(S)$ の逆行列を左からかければ、 $H(S)$ の左既約分解

$$H(S) = A_1(S)^{-1} \cdot B_1(S) \quad (11)$$

$$A_1(S) = G_1(S)^{-1} \cdot AL(S) \quad (12)$$

$$B_1(S) = G_1(S)^{-1} \cdot BL(S) \quad (13)$$

が得られる。ここで $G_1(S)$ は $m \times m$ 下三角行列であり、 $m \times (m+r)$ の多項式行列 $[AL(S), BL(S)]$ に列に関する基本列変換の積 $M(S)$ を右からかけ

$$[AL(S), BL(S)] M(S) = [G_1(S) \ 0] \quad (14)$$

になるような $M(S)$ を求めれば自動的に得られる。

このとき $M(S)$ を

$$M(S) = \begin{bmatrix} P_1(S) & R_1(S) \\ Q_1(S) & S_1(S) \end{bmatrix} \quad (15)$$

とすれば

$$AL(S) \cdot R_1(S) + BL(S) \cdot S_1(S) = 0 \quad (16)$$

$$A(S) \cdot P_1(S) + B(S) \cdot Q_1(S) = 1 \quad (17)$$

なる方程式が成り立つ。両式は多変数制御系の解析、設計に重要な役割を果たすことは周知の通りである。

COUGAR-1には上述の既約分解を行なうメニューを持っている。いま $H(S)$ を

$$H(S) = \begin{bmatrix} \frac{2S+3}{S^2+3S+2} & \frac{1}{S^2+3S+2} \\ \frac{1}{S+2} & \frac{-1}{S^2+5S+6} \\ \frac{1}{S+1} & \frac{2}{S^2+4S+3} \end{bmatrix} \quad (18)$$

とし、既約分解の具体的な実行例を示す。

まず関数 MATINを用いて(18)式を入力する。続いて MFD!-L (H,S); と入力すると、CRT上に図7に示すようなメニューが表示される。5行目までが(8)、

! *MFD!-L(H,S);

```
H = AL**(-1) * BL = A1**(-1) * B1
          A1 = G1**(-1) * AL
          B1 = G1**(-1) * BL
A1 * P1 + B1 * Q1 = I
AL * R1 + BL * S1 = 0
```

```
(0) H ; (1) AL ; (2) BL ; (3) A1 ; (4) B1 ;
(5) G1 ; (6) P1 ; (7) Q1 ; (8) R1 ; (9) S1 ;
(10) END ; (11) APPEND SUFFIX AND KEEP ALL MATRIX ;
```

Please input matrix number

図7 メニューの表示

(11),(12),(13),(16) 及び(17)式の表示であり、これらの式中に現われる行列には0から9までの番号を割りふってある。ユーザはこのメニューを見て必要に応じて結果を呼び出すことができる。

例えば、入力した行列をチェックしたければ0を入力するとただちに図8が表示される。このとき最後にDo you want to keep this matrix?とメッセージが出力されるので行列を保存する必要があるればYを入力し、それに適当な名前を付ける。この場合は必要がないのでNを入力する。すると再び図7のメニューが出力される。これは10すなわちENDの番号が入力されるまで繰り返される。図9はGが必要なので5を入力し、これを保存するためにYとし、これにGCLDと名前を付けた結果である。また図10はこのときの既約分解の結果のみを示したものである。以上、左既約分解の実行例のみを示したが本CADシステムには右既約分解を行なう関数も備わっている。

```

0
H :=
|-----|
| (2*S + 3)/(S2 + 3*S + 2)  1/(S2 + 3*S + 2) |
| 1/(S + 2)                  (-1)/(S2 + 5*S + 6) |
| 1/(S + 1)                  2/(S2 + 4*S + 3)   |
|-----|
Do you want to keep this matrix ? Y/N

```

図8 入力行列の表示

```

5
G1 :=
|-----|
| 1      0      0 |
| (-1)  3*(S + 2)  0 |
| 2      -3*(S + 1)  S2 + 4*S + 3 |
|-----|
Do you want to keep this matrix ? Y/N
Y
Please input matrix name
GCLD

```

図9 最大公約多項式行列G

```

A1 :=
|-----|
| S2 + 3*S + 2  0      0 |
| (S + 1)/3     (S + 3)/3  0 |
| (-1)          1      1 |
|-----|
B1 :=
|-----|
| 2*S + 3  1  -1 |
| 1        0   0 |
| 0        0   0 |
|-----|

```

図10 左既約分解の結果

5. COUGAR-1の問題点

前章まで非数値演算が可能な線形制御系のCADシステムCOUGAR-1について述べてきたが、まだまだ完成されたシステムではなく様々な問題点を残している。それは、数式処理システムREDUCEを使用することからくる問題、制御系を扱うことから発生する問題、及びコンピュータシステムやその周辺機器による問題などに分けることができる。

この章では本CADシステムの問題点としてつぎの①～④をあげ、今後どのように改善していくか検討する。

- ① REDUCEの数式演算に関するプログラムが有している虫(バグ)
- ② メニュー方式採用による問題
- ③ 行列の清書出力
- ④ REDUCEの不足機能

①について詳説すれば、国内では、数式処理システムはまだマイナーな言語であり使用者も少ない。国外においてはREDUCEよりも2章で述べたMACSYMAのほうが機能的に見て勝っており主流だと思われる。それに帰因するかどうかはわからないがREDUCEには、多くのバグが潜んでおり、我々の作成した関数にも影響を及ぼす。今後は、REDUCEの機能をさらにあげると同時に、少しでもバグを減らす努力が必要である。

②については、COUGAR-1のすべての関数の呼び出しをメニュー方式で行なうと、その関数の応用範囲

が限定され逆に使いにくい場合も生じて来る。例えば、LISPの関数のように関数を幾つか組み合わせて新たな関数として定義する(図11参照)ということが、それぞれの関数が会話的に実行するようにプログラミングされているためにできなくなる。

現在、COUGAR-1からCOUGAR-2へとバージョンをあげるために、このメニュー方式について再検討中である。

```
FOR ALL X,I,J LET
  DET!-ADJ(X,I,J) = DET(ADJ(X,I,J));
```

図11 余因子行列の行列式をとる関数定義

③については、現在の端末CRTやLP用紙の中が80カラムや135カラムに決まっているので、行列の次数が大きかったり、行列要素の式が長いと行列を清書出力しても、システム固有の制限によって行列要素はダンプリストになる。今後は出力方式をさらによいものに変更していきたい。

さらに④についていえば、REDUCEは2章で述べた機能を持っているが、同章で述べたMACSYMAの機能には及ばない。MACSYMAは制御系の設計、解析をするのに魅力的な機能を有しているが、一方ソースリストを公開していない(REDUCEは公開している)などの点で我々が制御系の関数を自由に作るできない。

また一般に数式処理システムはLISP言語を使用しているため、その他の言語例えばFORTRANやBASICなどにくらべて数値計算速度が遅い。そのためグラフィック機能などを付け加えると、どうしても時間的に無駄が生じてくる。そこで次章では数値計算と数式処理の長所をいかしたシステムについて述べる。

6. 数値-数式ハイブリッドCAD

数式処理システムは記号ばかりでなく数値も取り扱うことができる。このとき数値は任意倍長実数で表わされるため誤差に対する影響をまったく考える必要がないが、その反面演算速度は極めて遅い。COUGAR-1を使用して制御系の設計、解析を行なう場合、記号に数値を代入してグラフを書かせたりする必要性がでてくる。さらに数式のCADばかりでなく、

数値のCADを必要に応じて使用することができれば都合のいい場合もある。

そこで筆者らは、数値と数式処理を混在させた数値-数式ハイブリッドCADを検討中であり、パイロットシステムとして、REDUCEとFORTRANをリンクしてグラフを書かせることを試みている。この試みはREDUCEの持つFORTRANコード出力機能とFORTRANの持つX-Yプロッタを動かすためのライブラリを使用して行なっており、現在システム構成は図12のようにになっている。図12を簡単に説明すると、COUGAR-1の起動は、本大学のコンピュータMELCOM M800を使用しN1ネットによって名大のコンピュータFACOM M382を起動して行なっている。グラフの出力は、COUGAR-1の関数GRAPH!-OUT(引数として書かせるグラフの式を持たせる)を実行することで名大のFACOM M382上にGRAPH.FORTというファイルが作成される。このファイルにはグラフを書くためのFORTRANのソースプログラムが格納されているので、このソースプログラムをリモートバッチによってMELCOM M800上のファイルに格納する。そしてMELCOM側でコンパイルして実行し、グラフをX-Yプロッタまたは端末に出力する。この方式だとシステムが分散しているので実際問題としては扱いにくい。しかしMELCOMではREDUCEが動作していないことや、X-Yプロッタはコンピュータシステムによって動作条

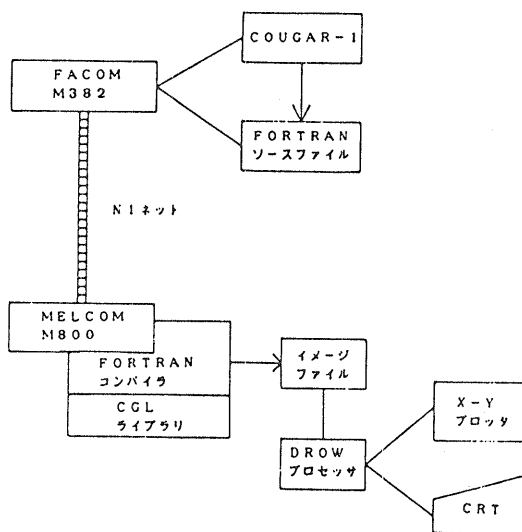


図12 数値-数式ハイブリッドCAD構成図

件が異なることで、現在はやむなくこの方式にあま
んじている。将来はオンデスクでCOUGAR-1を動作さ
せ、使い勝手のよい数値-数式ハイブリッドCAD
を開発したいと考えている。

次に、数値-数式ハイブリッドCADの例を示す。

6-1. システムの安定判別

設計したシステムが安定か不安定かという問題は
非常に重大である。そこでシステムの安定性を数値
-数式ハイブリッドCADで解析してみる。

$$S^4 + 2S^3 + S^2 + (1+K)S + K = 0 \quad (19)$$

のような特性多項式を持つシステムの安定性は、す
べての根が複素数 S の平面上の左半平面内（しかし
虚軸を含まない）に存在するかどうかを確かめれば
よい。これを調べる方法としてフルビッツの安定判
別法がある。これは既に、COUGAR-1が関数 HURWITZ
として持っているので、それを呼び出せば図13の
結果を得る。すべての根が負の実部を持つための必
要十分条件はフルビッツ行列 H のすべての主座行列
式が正となることである。つまり関数 HURWITZで出
力された $H1$ から $H4$ がすべて正となるような K の範囲
を求めればよい。そこで $H1$ と $H2$ は K の範囲がすぐわ
かるが、 $H3$ と $H4$ は難しいのでGRAPH!-OUT関数でグラ
フを描いて解析する。その結果が図14であり K の
範囲が明確に示された。

```

HURWITZ():
  WHAT IS INPUT ?
  Characteristic Polynomial ----> C(HARACT)
  System Matrix A -----> H(MATRIX)

C
  Please Input Characteristic Polynomial
  S**4+2*S**3+S**2+(1+K)*S+K:

CHARACT := S^4 + 2*S^3 + S^2 + S*(K + 1) + K
H :=
  | - 2  K + 1  0  0  - |
  |  1  1  K  0  0  |
  |  0  2  K + 1  0  0  |
  |  0  1  1  K  0  - |
  | - - - - - |
  H1 := 2 > 0
  H2 := -K + 1 > 0
  H3 := -K^2 - 4*K + 1 > 0
  H4 := K*( -K^2 - 4*K + 1) > 0
  
```

図13 フルビッツの安定判別法

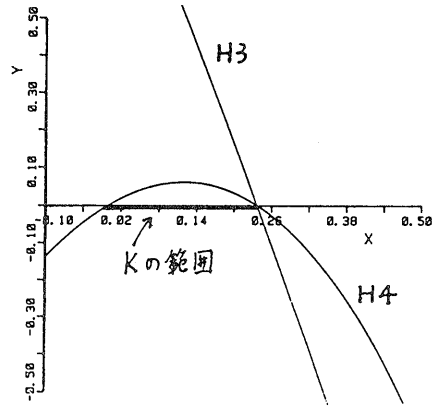


図14 GRAPH!-OUTによるグラフ

7. 結 言

以上、現在筆者らが開発を行なっている数式処理
言語による制御系のCADシステムCOUGAR-1につい
て述べてきた。

本システムは4章の例題でも示したように、従来
の数値解法によるCADでは取り扱いにくい制御系
の構造まで立ち入った設計、解析が可能である。ま
た得られる結果はすべて解析解であり多くの情報
を含んでいるので、制御理論の新しい定理の証明や
設計技法開発の補助的なツールとしても使用するこ
とができる。

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、制御系の解析、設計にはまだ不十分である。
例えば現代制御理論の設計、解析技法には固有値、
固有ベクトルを求める処理を頻りに必要とするが、
現在の数式処理システムは固有値問題は不得意であ
る。しかし多項式演算についてはかなり高度な機能
を有しており4章4-2.で示した伝達関数行列の既約
分解のように多項式演算を基礎とした設計技法は得
意といえる。今後、数式処理に向けた、すなわち多
項式演算による制御理論の見直しの作業も必要とな
ってくる。

参考文献

- [1] 斎藤 他：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 3巻 3号
- [2] 斎藤 他：数式処理言語による制御系CADシ
ステムの開発、「第3回知識工学シンポジウム」