



学生実験用データ解析プログラム作成支援システムの 開発

メタデータ	言語: jpn 出版者: 公開日: 2013-12-05 キーワード (Ja): キーワード (En): 作成者: 大西, 章 メールアドレス: 所属:
URL	https://doi.org/10.24729/00007792

学生実験用データ解析プログラム 作成支援システムの開発

大西 章*

Development of Support System
for Experimental Data Analysis for Students

Akira OHNISHI*

ABSTRACT

A support system is developed for experimental data analysis and chemical process calculation for education. Using this support system, even a beginner may create practical programs by aids of on-line manual on instructions and subroutine modules with CAI manner. This system generates BASIC source program from simple text program combining system's subroutine modules automatically. Modification of this system to other subjects can be made easily by remaking of system files.

Key Word : support system, education, chemical process calculation, CAI

1. はじめに

近年における情報処理機器と利用技術の発達・普及にとともに、高専における情報処理教育についても従来のプログラム言語教育中心の内容から、コンピュータを使いこなせる能力の育成、すなわちオペレーティングシステムについての基礎内容を含めたりテラシー教育の実施法やアルゴリズム教育の方法論等を含めて情報処理教育のありかたや内容について学校学科を問わず検討と試行が精力的に行われている。

一方、情報処理科目以外の科目についてもコンピュータを効果的に利用した教育の導入が強く望まれている。この背景には高度情報化社会への急速な変化もさることながら、最近の学生の各種能力の多様化および学校週5日制の実施と新教育課程の移行にともなう授業時間減に対応していかに教育効果をあげるかという問題の解決が急務なことにあるといえる。

コンピュータ利用教育としてよく知られているのはCAIであるが、著者の知る範囲内では市販されている教材の多くは、説明を単に表示するものやYes/No式の設問形式のものがほとんどであり、優れた参考書に勝るものは見あたらないように思われる。また、現時点ではCAI教育の望ましい実施方法やその効果につ

いてはまだまだ不明な部分も多いとの意見もあるようである。したがって、高専において学生の自習や理解を補助するタイプのCAIを広く導入するにあたっては、独自の教材開発と試行を通じていっそうの検討が必要ではないかと思われる。

もう一つの形のコンピュータ利用教育として、MathematicaやMathcadに代表される支援ソフトウェアの教育への利用が考えられる。この利用形態はわずらわしいプログラミングなしに問題についての疑問やアイデアを思いついたら直ちに確かめられるという優れた機能と学生自身が主体的に学習することを補助できる点で、特に高学年の教育には有効と思われる。もちろん学生が問題解決を進めるにあたって自分でプログラムが短時間で作成でき、深い検討や考察ができるにこしたことはないが、最近の本校工業化学科では学生の個人差が大きく、20行程度の文法や制御構造についての演習が主であるためにこれはなかなか困難な要求である。しかし、最近のプログラミングに対する考え方として、効率化とソフトウェア資産の有効利用を目指す方向にあり、このような情勢に対応して、リテラシー教育、言語教育に加えて各種支援プログラムを利用した教育を積極的に取り入れていくべきと考える。

ところで、以前から高専学生の教育においては、アルゴリズム教育、すなわち、プログラミングに限らず問題を総合的に分析、分解、合成して問題解決ができる能力の育成の必要性が広く指摘されてきた。近年、

1994年4月11日受理

* 工業化学科(Department of Industrial Chemistry)

この問題はいつそう重要となってきた。この訓練は、総合科目である実験実習科目で行うのが最適であり、ここにコンピュータを支援ツールの形態で利用することによる時間的かつ教育内容的効果はきわめて高いといえる。しかしながら、化学科の学生でも手がるに使用でき、かつ、価格の安い教育用のアプリケーションが見つからないのが実状である。

そこで、本報告では、プロセス工学系実験を対象に学生が主体的に問題解決を行う過程を補助し、コンピュータについての知識やプログラミング経験が少なくても実験データの解析やめんどろなプロセス計算が手がるにできることを目標に開発した支援システムについて紹介する。

なお、使用した機種、プログラム言語、OSは日本電気製 PC-9801DA(5MB)、N88-日本語BASIC(MS-DOS版、Ver6.0)、日本語MS-DOS(Ver3.3D)である。

2. システムの特徴

本システムは学生実験において必要な文書作成、グラフ作成、表計算、数値計算、実験課題ガイダンス参照等の多様な処理に対応できることと構造化プログラミングの考え方を身につけさせることを目標に開発した。特徴は以下のとおりである。

- 1) BASICインタプリタ上ですべての処理ができるためコンピュータについての深い知識がなくても使用できる。
- 2) プログラム作成中に BASIC文法をウインドウ画面で参照することができる。
- 3) 実験データの解析に便利なモジュールが用意されており、プログラム作成中に使用法をウインドウ画面で参照できる。
- 4) モジュールでは簡易形式ではあるが引数と関数副プログラムが使用できる。
- 5) モジュールのユーザ登録ができる。
- 6) テキスト作成は基本的に BASIC言語で行うため、ユーザがオリジナルな実用プログラムを効率的に作成できる。

現在モジュールとしては、データファイル、データプロット、関数プロット、方程式の根、最小二乗法、回帰式、補間、補間プロット、数値微分、数値積分、連立一次方程式、連立常微分方程式、表作成が登録されている。モジュールの作成においては高等専門学校工業化学系情報処理科目担当教員らによって執筆され

た情報処理テキスト¹⁾の演習問題が10~20行程度のプログラミングで解けるよう教育レベルと機能に配慮した。

本システムの開発において特に重視した点はN88BASIC言語についての基礎知識さえあれば手がるに利用できることと、決められた様式に従ってファイル群を作成すれば数値的視覚的確認の有効な物理・数学系科目についての教育支援にも使用できる拡張性である。

3. システムの構成

システム全体の構成を図1に、システムファイルの構成を図2に示す。システムは約1200行のN88BASIC言語による本体プログラムと約70個のファイルから成っている。プログラムの主要部はテキストエディタ、BASIC文法とモジュール説明表示制御ルーチン、テキストソース変換ルーチンである。

テキストエディタは、BASIC言語では使用できるキー入力命令がLINE INPUT命令とINKEY\$命令であるため前者の命令を使った基本テキスト入力処理ルーチンと

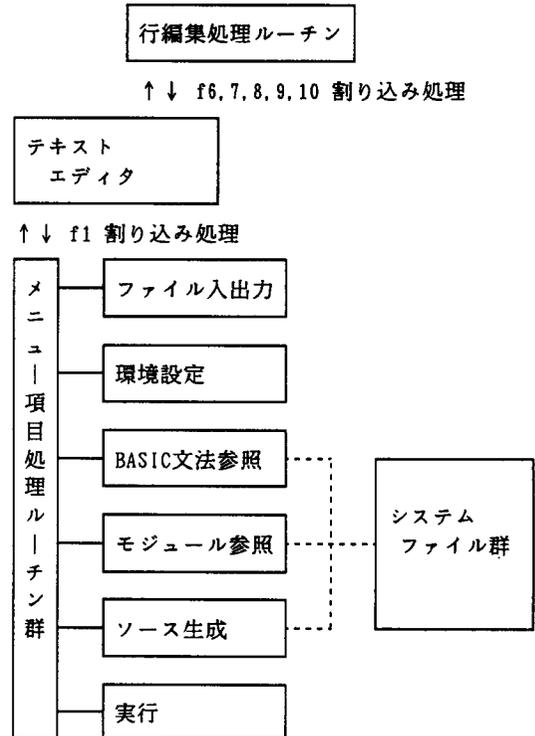


図1 システム構成

ファンクションキー割り込み処理と後者の命令を使った行単位編集処理ルーチンから成っている。編集処理機能にはカーソルキーによるスクロールと行単位複写移動もでき、一般の本格的なスクリーンエディタと比較してもそれ程使用勝手は劣らないと思われる。

本システムでは他の言語や各種科目課題への対応およびモジュール変更等の拡張性を考えてモジュールに関するメニュー項目、説明文、結合用サブルーチンモジュールはすべて図2に示すような文書ファイルとなっている。BASIC文法についての扱いも同様である。

モジュール登録の詳細を図3に示す。1)のモジュールメニューファイルにはモジュール参照時に画面下方

に表示されるメニュー項目が記述されている。2)のモジュール登録ファイルには1)のメニュー項目と対応して、モジュール名、引数、引用サブルーチン名が記述されている。ユーザ作成のモジュールもこのファイルに引数を記述しておけばテキストソース変換時に引数処理命令がソースプログラムに書き込まれる。3)のモジュール仕様説明ファイルは2)で登録されているモジュール名+拡張子(.EXP)のファイル名で一画面内の説明文から成っている。4)の結合用モジュールファイルは2)のモジュール名+拡張子(.SUB)のファイル名でソース作成時に主プログラムにマージされるBASIC副プログラムである。ただし、このプログラムには文番

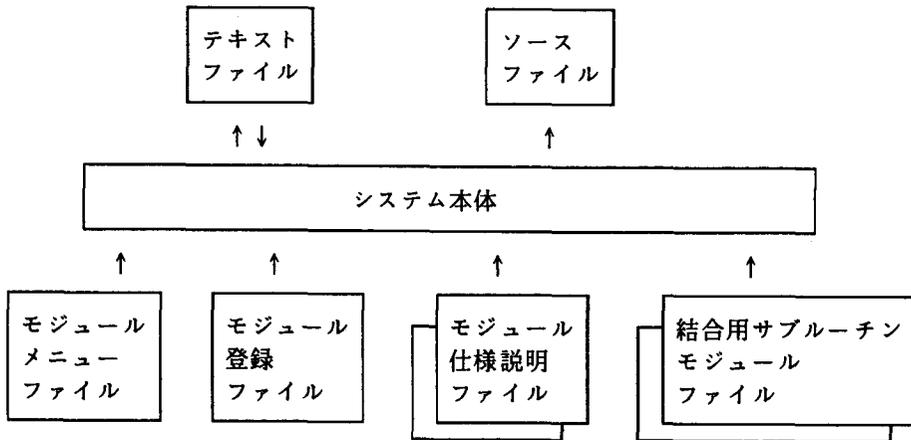


図2 ファイル構成

1)モジュールメニューファイル
[MODULE.LIS]

```

-----
:常微分方程式
:表作成
:子-の正規化
    
```

2)モジュール登録ファイル
[SBMDN.LIS]

```

-----
*RUNGE (FUNC#)
*TBL
*NORMAL (NDM, XDM, XWM) *CERR1
    
```

↑ ↑ ↑
モジュール名 引数 引用サブルーチン

3)モジュール仕様説明ファイル
[NORMAL.EXP]

```

説明文
    
```

4)結合用モジュールファイル
[NORMAL.SUB]

```

*NORMAL
DIM XDM(100), XWM(100)
-----
RETURN
    
```

←使用配列
宣言記述

図3 モジュール登録

号はつけない。また、必要ならば、配列宣言を記述しておく。この配列宣言文は引数処理命令作成の時に参照されるとともに、多重宣言のチェックをした後、ソースプログラムの先頭行にまとめて記述される。図には、新たにデータの正規化モジュールを登録する例について必要な部分を網掛けで示した。

4. システムの仕様

4. 1 使用法と画面例

本システムを起動させると、基本画面である図4のテキスト入力編集画面となる。本システムのテキスト作成は一画面程度(20行程度)を想定しているので行編集機能は画面単位のf6~f9で十分といえるが、データファイル等の任意のアスキー形式ファイルの編集のために効率性を考慮して、f10の行編集モードではカ

ーソルキーによるスクロール、コピー、移動ができるようになっている。また、テキスト作成においては見やすいテキストが作成できるように、空白行が使用できるようにになっている。

f1のメニューを選択すると、画面下2行に図5のメニュー項目が表示される。モジュール参照を選ぶと図6のようにモジュールのメニューが表示される。希望するモジュールを選択すると、画面上方に説明が表示される。このときテキストは説明文の下に移動する。引続き入力画面を選択すると、説明文を表示したままテキスト入力編集画面となる。図7には BASIC文法を参照した時の画面例を示した。

なお、画面の文字は対話性を考慮して、メニュー項目は黄色、テキストは白色、説明文は緑色、入力ガイドと確認項目は水色等に色分けされて表示される。

テキスト名 [D:EX6.TXT] 画面表示行 [1 - 22] テキスト最終行 [11] <

'[例6] 方程式 $f(x) = -x + \ln(x) + 2 = 0$ の根を求める

```
GOSUB *PLOT(FUNC1(X))      : ' 根の範囲をグラフで確認
GOSUB *NIBUN(FUNC1(X))     : ' 二分法ですべての根を求める
END

*FUNC1(X)                  : ' 関数副プログラムの定義
  FUNC1=-X+LOG(X)+2
  RETURN
END
```

f1:メニュー f6:行挿入 f7:行削除 f8:上画面 f9:下画面 f10:編集

図4 テキスト入力編集画面

[メニュー] L:読み込み S:保存 P:印刷 G:ソース作成 R:実行 B:BASIC文法 M:モジュール参照
O:オプション A:入力画面 Q:終了

図5 メニュー項目

テキスト名 [D:EX3.TXT] 画面表示行 [1 - 22] テキスト最終行 [8] <

[補間プロット] (形式1) GOSUB *PLOT(N,X,Y) N:データ数 X,Y:データ(1次元配列)
 (形式2) GOSUB *PLOT

- データを4点Lagrange補間または多項式補間してスムーズ曲線で結ぶ.
 - プログラムの先頭で X,Y の配列宣言 (例) DIM X(100),Y(100) をしておく.
データを N, (X(i),Y(i), i=1,N) に代入してから GOSUB *PLOT(N,X,Y) とする.
 - 引数を省略すると変数として NDM,XDM(510),YDM(510) が使用される.
- ※新規プロット, 重ねプロット, 対数プロット が実行時に選択できる.

・ [例3] データの補間値を求める

```
GOSUB *DFC            : ' データファイルから原データを読み込む
GOSUB *PLOT          : ' グラフで原データを確認
GOSUB *LAGRA         : ' Lagrange補間によって補間値を求める
END
```

[モジュール参照] 1:補間 2:補間プロット 3:数値微分 4:数値積分(式)
 5:数値積分(データ) 6:連立一次方程式 N:次メニュー A:入力画面 f10:復帰

図6 モジュール仕様説明画面例

テキスト名 [D:EX7.TXT] 画面表示行 [1 - 22] テキスト最終行 [9] <

[所定回繰り返す] (形式) FOR J=1 TO 10

処理

NEXT J

- J=1, 2, ..., 10 と変化させて処理を実行します.
処理の中で J= ... としてはいけません.

FOR X=0 TO 1 STEP 0.1 とすると X は 0, 0.1, 0.2, ... と変化します.

```
DIM TD(100),XD(100)
MRK=1                : ' 微分方程式の次数のセット
HRK=0.1:T=0:X=0     : ' サミと初期値のセット
FOR I=1 TO 100
  GOSUB *RLNGE(FUNC1(T,X)=T+X)    : ' 微分方程式を1ステップ解く
  TD(I)=T:XD(I)=X                : ' プロットデータを格納
NEXT I
ND=100:GOSUB *PLOTD(ND,TD,XD)    : ' 結果のプロット
END
```

[BASIC文法] 1:文関数(DEF FNF(X)) 2:判断(IF) 3:所定回繰り返す(FOR)
 4:前判定繰り返す(WHILE) 5:飛越し(GOTO) N:次メニュー A:入力画面 f10:復帰

図7 BASIC文法参照画面例

4. 2 モジュール仕様と引用命令処理

モジュールには引数のない型、変数の引数をもつ型、関数名の引数をもつ型の3種類がある。ここで、変数の引数をもつ型は引数を記述しない使用もできるようになっている。また、作成するテキストができるだけ簡単になるように、細かい機能の選択は基本的には実行時に会話形式で行う形式をとっている。ただし、不都合な場合もあるので、一部のモジュールについてはフラグをセットすれば、主プログラムで設定することが可能である。

変数引数をもつモジュールの場合、モジュール登録ファイルに引数登録をしておけば、任意個数の使用が

できるが、配列については決められた個数の一次元データのみという制約がある。すなわち、モジュール登録ファイルに引数が登録されており、かつ、結合用ファイルに配列宣言があれば、第一引数をデータ数として図8に示す引数処理命令がソースプログラムに作成される。この制約は、言語教育の立場からは問題もあるが、テキスト→ソース変換ルーチンをコンパクトにする、実用性を失わない、初心者を使いやすい点を勘案して設定した。モジュール引用処理ルーチンは、一連の上記ソースを作業領域に作成した後、引用モジュール名と使用配列名を作業領域に登録する。

(テキスト例) データファイルモジュールの引用

GOSUB *DFC(N, X, Y)

1. 実引数→仮引数処理

2. モジュール引用命令

3. 仮引数→実引数処理

```
10 NDM=N
20 FOR IDM=1 TO NDM:XDM(IDM)=X(IDM):NEXT
30 FOR IDM=1 TO NDM:YDM(IDM)=Y(IDM):NEXT
40 GOSUB *DFC
50 N=NDM
60 FOR IDM=1 TO N:X(IDM)=XDM(IDM):NEXT
70 FOR IDM=1 TO N:Y(IDM)=YDM(IDM):NEXT
```

4. 引用モジュール名登録

*DFC *CERR1

5. 使用配列登録

XDM(500) YDM(500)

図8 モジュール引用命令処理

4. 3 関数仕様と定義・引用命令処理

各種数値計算を含むプログラミングには使いがってのよい関数の定義・引用命令が使用できることが不可欠である。そこで、本システムでは FORTRAN言語の関数副プログラムとEXTERNAL宣言に相当する様式と数式処理ソフトウェアのようにモジュール引用命令の中に直接関数を記述することができる様式を採用した。すなわち、例えば $f(x)=\sin(x)$ のグラフを描くには、

```
GOSUB *PLOT(FUNC1(X))
END
*FUNC1(X)
  FUNC1=SIN(X)
  RETURN
END
```

または、

```
GOSUB *PLOT(FUNC2(X)=SIN(X))
```

と記述すればよい(計算範囲は実行時に会話形式で入力するようになっている)。

関数名については FUNC+任意の番号、引数は、変数名については10以内、かつ、変数が配列の場合はその大きさは10までの制約がある。また、N88BASIC言語のため、引数外の変数についてはグローバル変数の扱いとなる。仮引数はソース作成時に作業変数 XFN(i, j) に書換えられ、実引数→仮引数処理が行われる。ここで i は引数番号、j は配列番号(配列でないときは1)である。関数の定義・引用についてのテキスト→ソース変換例を図9に示す。すべての関数定義文は共通のラ

ベル名*FUNC から始まる命令文に書換えられ、引用の際は関数番号IFUNC\$の内容によって所定の関数番号のラベルに分岐する。結果はすべて共通の変数FUNCに戻る。関数の多重定義の場合（関数定義の中で他の関数を引用する場合は多重度を kとして作業変数XFND(k, i, j) が使用され、10重まで可能である。なお、図9で文番号50の処理は一行の算術演算命令文中に複数回の関数引用があることへの対応であり、この場合も10回まで許される。

テキスト	作成されるソースプログラム
Y=FUNC1(X) ----- ----- END	10 IFUNC\$="1" 20 XFN(1,1)=X 30 GOSUB *FUNC 40 X=XFN(1,1) 50 XFN(1,0)=FUNC:Y=XFN(1,0)
*FUNC1(A) FUNC1=A^2 RETURN END	100 *FUNC 110 IF IFUNC\$="1" GOTO *FUNC1 120 PRINT "関数引用エラー" 130 RETURN 140 *FUNC1 150 FUNC1=XFN(1,1)^2 160 FUNC=FUNC1 170 RETURN

図9 関数定義・引用命令処理

4. 4 テキスト例

例1. 複数組のデータを重ねプロットする。

```
*A
GOSUB *DFC
GOSUB *PLOTD
GO TO *A
END
```

*DFC: データファイル入出力、データのキー入力と編集ができる。注釈行が使用できる。

*PLOTD: 新規プロット、重ねプロット、対数プロット点、ライン等が選択できる。

例2. データの補間値を求める。

```
DIM X(100),Y(100)
GOSUB *DFC(N,Y,X)
GOSUB *PLOT(N,X,Y)
GOSUB *LAGRA(N,X,Y)
END
```

引数を使って Y→X の補間をしている。*PLOT:4点 Lagrange補間、または任意の多項式補間によってデータをスムーズ曲線で結ぶ。

例3. 関数 $Y=f(X,P)=(1-X)/(1+(P-1)*X)$ のパラメータPの影響をグラフで調べる。

```
*A
INPUT "P >";P
GOSUB *PLOT(FUNC1(X,P)=(1-X)/(1+(P-1)*X))
GO TO *A
END
```

*PLOT: 関数FUNC1(X,P)において、関数プロットは第1引数について行われる。第2引数以後はパラメータとして扱われる。

例4. メタノール-水系を理想溶液として、ラウールの法則から気液平衡関係を求め、その結果を補間してスムーズ曲線を描く。続いて、文献値または実測値を読み込んで重ねプロットして比較する。

```
DIM XD(20),YD(20)
A1=7.87863:B1=1473.11:C1=230
A2=7.96681:B2=1668.21:C2=228
FOR I=1 TO 11
  X=(I-1)/10
  GOSUB *NIBUN(FUNC1(T,X))
  Y=FUNC2(XXM,A1,B1,C1)*X/760
  XD(I)=X:YD(I)=Y
NEXT I
ND=11:GOSUB *PLOT(ND,XD,YD)
GOSUB *DFC:GOSUB *PLOTD
END
*FUNC1(T,X)
  FUNC1=FUNC2(T,A1,B1,C1)*X
  FUNC1=FUNC1+FUNC2(T,A2,B2,C2)*(1-X)-760
RETURN
```

```

END
*FUNC2(T, A, B, C)
  FUNC2=10^(A-B/(C+T))
  RETURN
END

```

この計算は試行錯誤が必要なため、手計算では長時間を要する学生実験テーマの代表例である。

*NIBUN: 指定された範囲で関数の実根を求める。

5. おわりに

最近における学生の問題解析力、プログラミング力の多様化と学校5日制および新教育課程の実施に伴う効果的教育の必要性に対応するために、実験データの整理、解析やプロセス工学関連科目における各種計算やシミュレーション演習を補助する支援システムを開発した。いわゆるティーチングマシンタイプや完全なレディメイドタイプのソフトウェアではないため、色々な問題に対応でき、学生が自ら主体的に思考して解析を進める時の支援ツールとして、特に高学年教育に有効と考えている。

教育への本格的利用と評価については今後の課題であるが、何人かの学生に使用させてみたところ、実用性についてはおおむね好評であった。よりブラックボックス的なモジュールの希望もあり、これに対応することは容易であるが、問題を総合的に分析、解析、合成できる能力を養う立場からは慎重に行うべきであると思われる。また、もう一つの問題点としては、できるだけ汎用性を考慮して開発したために、モジュール引用における一般的なエラーチェック機能と実行時のエラーチェック機能はあるものの、モジュールごとの細かいエラーチェックがないために、BASICインタプリタで検出されたエラーの原因の特定がユーザの学生には難しいことがあることである。これに対応するにはプロダクション技法⁸⁾の応用が考えられるが現時点ではなかなか難しい問題ではある。

今回開発したシステムではモジュールの説明文の行

数は一画面以内で自由であるが、初心者を対象とした場合、例題+仕様説明を表示するのが好ましいと思われるため、説明表示ウィンドウは1/2程度の範囲で自由にスクロールできるように改良すべきと考えている。また、実行はシステムをいったん終了させてからソースプログラムをロード、実行させているが、教材作成に利用して感ずることは、ウィンドウ画面でテキストの任意の部分を実行でき、実行結果をテキストに取り込める機能があれば使用勝手はさらに向上するのではないかと思われる。

今後の試行を通してさらに適切な改良と拡張を行っていくとともに、効果的なアルゴリズム教育の方法についても検討していきたいと考えている。

最後に、本システムの開発に協力いただいた本校卒業生、高橋晶子さん、佐藤富弘君、岡橋久世さんに感謝いたします。

参考文献

- 1) 大島英次 監修:工業化学のためのFORTRAN77, 日刊工業新聞社(1987)
- 2) 河西朝雄:構造化BASIC, 技術評論社(1985)
- 3) N88-日本語BASIC(86)(Ver6.2)リファレンスマニュアル, 日本電気株式会社(1991)
- 4) 間室 規, 関沢恒夫, 須藤義孝:工業化学実験データの整理法, 日刊工業新聞社(1985)
- 5) 大西 章, 黒岩盛治:大阪府立工業高等専門学校情報処理センター広報, 第4号, pp. 51-60 (1993)
- 6) 若林 茂:高専教育, 第17号, pp. 120-125 (1994)
- 7) 佐藤義隆, 大西 安, 豊成敏隆:高専教育, 第17号, pp. 165-172, (1994)
- 8) 新開純子, 宮林頼夫:高専教育, 第17号, pp. 232-239 (1994)
- 9) 五福明夫:システム制御情報学会誌, Vol. 37, pp. 531-542 (1993)