

ミニコンにおけるプロッタ制御の効率化について†

の場 進†† 本田直人†† 吉岡信夫†† 田辺逸雄††

Abstract

When we use minicomputer system directly connected to a digital plotter, we may spend idle time of minicomputer to plot the calculated results, because the digital plotter is very slow compared with the minicomputer.

We have developed the plotting software system to reduce the idle time, with which the plotter job is processed in two phases. At the first phase, informations necessary to control the plotter action are written on a small magnetic tape. At the second phase, these informations are read and used to control the plotter action. In this phase, another user can use the minicomputer system regardless of the plotter action. Thus the idle time is reduced to 1/50-1/100.

1. まえがき

ミニコンピュータ（以下ミニコンと呼ぶ）の著しい普及にともないその利用方法も多種多様なものになってきつつある。ミニコンの制御のもとでデジタル・プロッタ（以下プロッタと呼ぶ）を利用するのもその1つであるが問題点もある。すなわち、プロッタはその速度が1基本動作につき数ミリ秒とあまり速くないためミニコンとオンライン接続されていると、プロッタが動作中はミニコンはそれに専有されてしまうおそれがある。プロッタによる計算機（CPU）の専有を避けるための方法として通常はプロッタに専用の制御装置をつけ、磁気テープや紙テープにまえて出力された図形情報をオフラインで処理する図1のような方法がとられている。（とくに大型計算機を利用する場合はこの方法がとられている。）

われわれはミニコンにオンライン接続されたプロッタを含むシステムを利用するに際し、プロッタに出力する図形情報をまず外部記憶装置（小型磁気テープ）に出力し、その後、その情報によりプロッタを動作させる仕事とそれとは独立な別の仕事とに同時にミニコンが利用可能となるように考慮し、システム全体の効率をあげることを考えた。このため、ミニコンとプロッタを結ぶインタフェースに割込み機能を持たせた

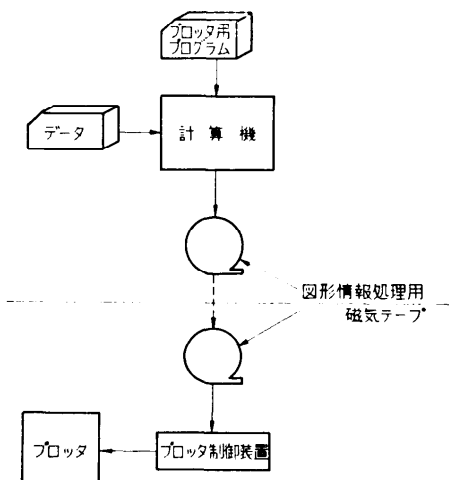


図1 プロッタのオフライン処理

Fig. 1. Off-line plotter system

り、ミニコン用のオペレーティング・システムを修正する必要が生じたが、一応システムとして完成したので以下に報告する。

2. ハードウェアについて

われわれの利用したハードウェアのうち本稿に関係のある部分を図2に示す。ミニコンとしては PDP-11/20、プロッタとしては DPL-602（岩通）が用いられている。前者についてはハンドブックその他^{1),2),3)}に示されているので略す。後者についての主な仕様を表1に示す。PDP-11/20 では図2に示すようにユニ

† Effective use of minicomputer and plotter, by Susumu MATOBA, Naoto HONDA, Nobuo YOSIOKA and Itsuo TANABE(Faculty of Engineering Science, Osaka University).

†† 大阪大学基礎工学部情報工学科

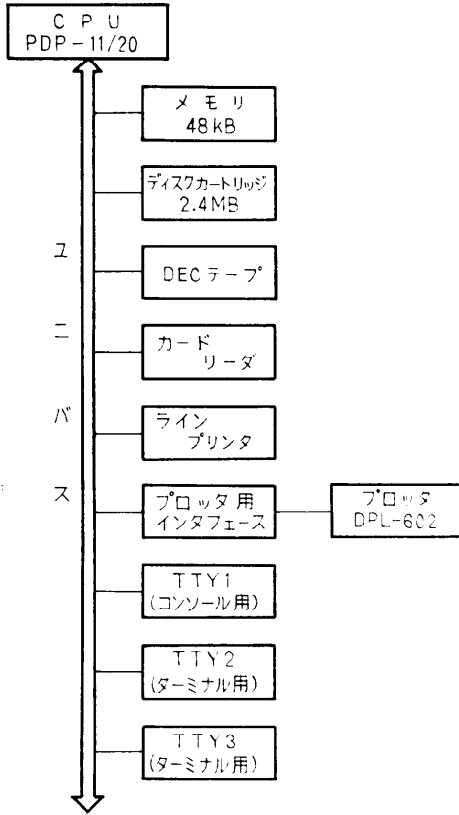


図2 PDP-11/20 の構成図
Fig. 2 System configuration

バスと呼ばれる1本のバスに各装置が結合されるようになっている。そして、それぞれの装置には個有の番地が与えられており、その番地を用いて CPU からアクセスされる。DPL-602 についても同様で、ユニバスと結合するためにインタフェースが必要となるが、このインタフェースにはプロッタの状態を示すステータス・レジスタ (プロッタ・ステータス・レジスタ) とデータ転送のためのバッファ・レジスタ (プロッタ・バッファ・レジスタ) の2つが組み込まれておりそれぞれについて特定の番地が定められている。この場合、インタフェースの有する機能はつぎのようなものである。

- (i) ユニバス上のアドレス信号をデコードする。
- (ii) 割込み要求を発生する。
- (iii) プロッタ自身からの動作完了の信号がないため、内部に遅延回路をもって動作完了信号を発生する。

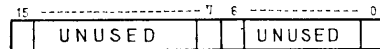
表1 DPL-602 プロッタの仕様
Table 1 Plotter specification.

プロット幅(単位ステップ)	0.1 mm
プロット速度	x 軸, y 軸とも 400 ステップ/秒 ペンの上下 10 回/秒
記録紙寸法	有効記録長 35m 有効記録幅 270 mm
行き帰り誤差	25 cm を往復したとき 0.1 mm 以下
プロッタの有する信号	レディ信号 サイドエンド信号 ペーパーエンド信号
外形寸法	456幅×315高×520奥 (mm)
重量	約 30 kg

- (iv) プロッタに関するエラー信号の発生。
- (v) バッファ・レジスタへの入力を増幅してプロッタを動作させる。

これらの機能を有するインタフェースのブロック図を図3に示す。

プロッタ・ステータス・レジスタは 16 ビットで構成されており、つぎのようにそのうちの4ビットが用いられている。



それらの機能を示すと

Error Bit (15): プロッタの電源がオフ、プロッタと計算機がオフライン、ペンが紙の両端を超えようとする、プロッタの用紙がなくなった、バッファ・レジスタにまちがった値をセットしたといった場合にエラー状態が検出され1にセットされる。このビットへのプログラムによる書き込みはできない。

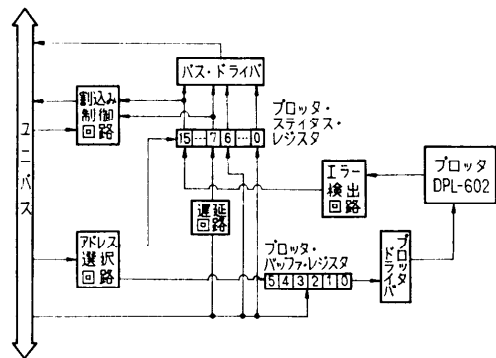


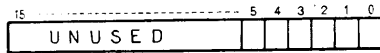
図3 インタフェースのブロック図
Fig. 3 Block diagram of plotter interface

Ready Bit (7): プロッタが動作中は 0 にセットされており、新しく動作可能状態になったとき 1 にセットされる。このビットへのプログラムによる書き込みはできない。

Interrupt Enable Bit (6): プログラムによりこのビットを 1 にセットしておく、プロッタのエラまたは、動作完了時に割込み要求が出される。このビットが 0 のときは割込みは発生しない。

Go Bit (0): プログラムによりこのビットに 1 をセットすると同時にプロッタはそのときのバッファ・レジスタの内容に対応した動作を行なう。このビットの値は保持されず 0 にリセットされる。

プロッタ・バッファ・レジスタも 16 ビット構成であるが実際に用いられるのはつきに示すようにその下位 6 ビットのみである。



ビット 5 はペン・アップ、ビット 4 はペン・ダウン、ビット 3 は X 軸の負の方向、ビット 2 は Y 軸の正方向、ビット 1 は X 軸の正方向、ビット 0 は Y 軸の負方向への 1 ステップ (0.1 ミリ) の動作を示す。

DPL-602 はペンの移動については図 4 に示すような 8 方向に限られているので、その動きに対応するバッファ・レジスタの内容を示すと表 2 のようになる。表 2 以外の値をセットしプロッタを動作させるとエラーが検出される。とくにペンのアップ、ダウンとその他の動作を同時に行うことはできないので注意が必要である。

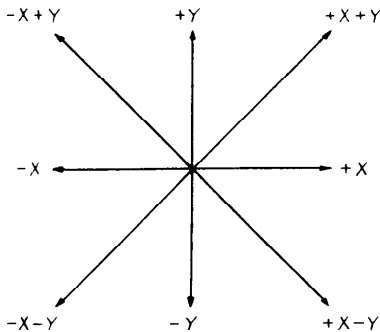


図 4 8 方向のペン移動図

Fig. 4 Basic movements of the plotter

表 2 基本動作に対応するバッファ・レジスタの値
Table 2 Contents of buffer register

ビット	5	4	3	2	1	0
ペン・アップ	1	0	0	0	0	0
ペン・ダウン	0	1	0	0	0	0
-X	0	0	1	0	0	0
+Y	0	0	0	1	0	0
+X	0	0	0	0	1	0
-Y	0	0	0	0	0	1
+X+Y	0	0	0	1	1	0
-X+Y	0	0	1	1	0	0
-X-Y	0	0	1	0	0	1
+X-Y	0	0	0	0	1	1

3. ソフトウェアについて

ミニコンも初期の頃にくらべるとかなり大型化し、オペレーティング・システムのもとで動作するものも多くなっている。われわれが利用したシステム(図 2)は PDP-11/20 DOS (Disk Operating System)⁴⁾のもとで動作可能であり、FORTRAN および ASSEMBLY 言語が使用できる。プロッタを使用する場合の言語としては、その処理が一般に多くの数値計算を必要とするため、FORTRAN 言語を用いることにした。つきにプロッタを利用する場合の概略を述べる。

まず、プロッタを用いる仕事が与えられると後述する図形処理用基本サブルーチンを用いて FORTRAN 言語によりソースプログラムを作成する。このプログラムの処理結果は、図 5 に示すように、2つのフェイズを経てプロッタに出力される。フェイズ 1 では上で作成したソースプログラムを DOS のもとで実行可能な形式にコンパイルしてラン (Run) させ、図形に関する情報 (以下図形情報と呼ぶ) を DEC テープと呼ばれる容量 147 k 語 (16 ビット/語) の小型磁気テープに出力する。フェイズ 2 ではテレタイプ TTY 2 (図 2) からのプロッタの動作開始コマンドにより後述す

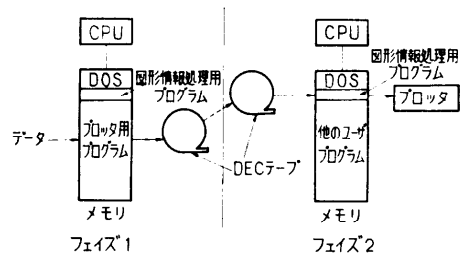


図 5 プロッタジョブの処理

Fig. 5 Operations of plotter job

る図形情報処理用プログラムを働かせ、フェイズ1で作成された図形情報をDECテープから読み込みプロッタを起動させる。これらのフェイズのうちミニコンを専有するのはフェイズ1のみで、フェイズ2では他のユーザがテレタイプTTY1(図2)のコンソールを用いて自由にミニコンを使用できる。フェイズ1とフェイズ2の処理時間の比は1:50~100であるため、プロッタを動作させる仕事のためにミニコンが専有される時間は図形情報を直接プロッタに転送する場合にくらべると1/50~1/100になる。

3.1 図形情報

フェイズ1において図形情報は容量147k語の小型磁気テープに出力されるが、このテープの容量を超えるような図形についてはテープを交換することによって処理可能になっている。図形情報としては、テープの容量をできるだけ有効に利用するために、前述したプロッタの基本動作を、直接表現するのではなく、3語を1つのグループとしてこれにより1つのベクトルを表現する方法をとっている。すなわち、図形がベクトルの集合のような形式に変換されてテープ上に出力されることになる。

この3語の構成のうち、最初の語は制御情報的なもので1, 2, -1, -2のいずれかの値がセットされる。

- 1: ペン・アップ
- 2: ペン・ダウン

- 1: テープに出力された図形に関する情報の完全な終りを示す。
- 2: テープ上の図形に関する情報は一般にいくつかの画面から成っており、これは、これらの画面の区切り記号として用いられる。

2および3番目の語は、プロッタの現在のペンの位置からつぎに移動すべきペンの位置までのX方向およびY方向の相対移動値($\Delta X, \Delta Y$)が0.1ミリを単位とする整数値で示されている。表現可能な最大値は $2^{15}-1$ であり、これはプロッタの動作にすると約327センチの移動に相当する。

たとえば、ある直交座標系のもとで現在のペンの位置が点A(a_x, a_y)にあり、そこから点B(b_x, b_y)に直線を描く場合を考えると、これに対応する図形情報としては、2(ペン・ダウン)、 $b_x - a_x(\Delta X)$ および $b_y - a_y(\Delta Y)$ の3つの値が1つのグループとしてテープ上に出力されていることになる。

図形情報についてこのような表現方法を用いたので1巻のテープには最大約49,000ベクトルを入れるこ

とが可能となり、ミニコンで処理する程度の図形としてはこれで十分ではないと思われる。

3.2 図形情報処理用プログラム

このプログラムはフェイズ2において、前述の図形情報を読みとり、プロッタに動作信号を送る働きをするものである。プロッタの基本動作はペンのアップ、ダウンと0.1ミリ単位の8方向へのペンの移動であるから、図形情報に含まれているXおよびY方向への相対移動値よりプロッタの基本動作をどのように組合わすべきかを決めなければならない。

そこで、X方向およびY方向への相対移動値をそれぞれ $\Delta X, \Delta Y$ で示すことにすると、まず、この2つの符号を調べ表2に示した値のうちからX成分レジスタ(XR)には $+X, -X$ または0を、Y成分レジスタ(YR)には $+Y, -Y$ または0をセットする。そして $\Delta X, \Delta Y$ の絶対値をとり以後の処理を直交座標の第1象限で考えることにする。つぎにプロッタへの動作指令の回数を数えるカウンタ(SC)を定義し、つぎの3つの場合について処理を行う。

$\Delta X=0$ または $\Delta Y=0$ の場合

$\Delta X(\Delta Y)=0$ の場合にはYR(XR)の値がバッファ・レジスタ(図3)にセットされSCには $\Delta Y(\Delta X)$ がセットされる。そして、SCの回数だけバッファ・レジスタに示された動作がくりかえされる。

$\Delta X=\Delta Y$ の場合

バッファ・レジスタにはXRとYRの論理和がセットされSCには $\Delta X(=\Delta Y)$ がセットされ、SCの回数だけバッファ・レジスタに示された動作がくりかえされる。

$\Delta X>\Delta Y$ または $\Delta X<\Delta Y$ の場合

$\Delta X>\Delta Y$ の場合を図6で説明すると、直線OAの勾配は $\Delta Y/\Delta X$ である。したがって、点O(現在のペンの位置)から右へX座標が1単位(0.1ミリ)増加するごとにY座標を $\Delta Y/\Delta X$ だけ増加させればよいことになるが、Y座標も0.1ミリを単位とする整数値しかとれないため真の直線上の点とプロッタのペンの位置との間には差が生じる。そこで、ここではこうして生じる差が0.5単位以下になるようにY座標を1単位ずつ増加してゆくようにした。これをフローチャートで示すと図7のようになる⁵⁾。

$\Delta X<\Delta Y$ の場合もほぼ同様であるが、この場合にはY座標を1単位ずつ増加させながらX座標を修正してゆくようになっている。なお、実際のプログラムにおいては乗除算を含まないで処理するような工夫

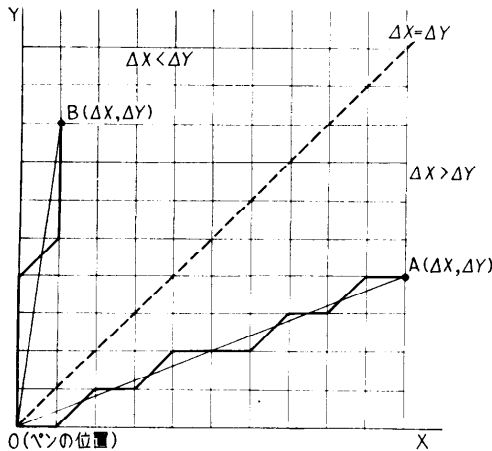


図6 線分の基本動作への分解

Fig. 6 Combination of basic movements

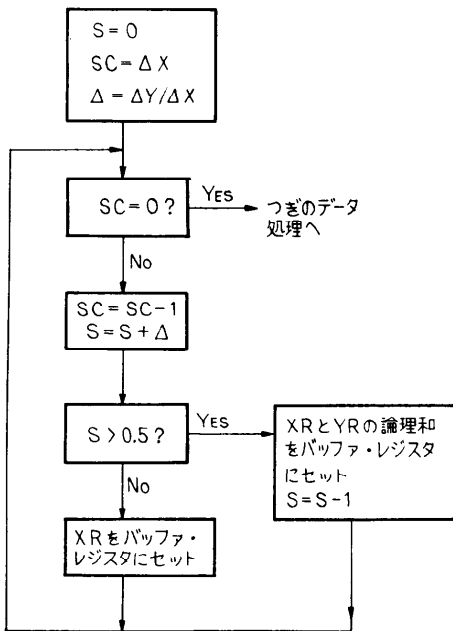


図7 ペン移動のアルゴリズムのフローチャート

Fig. 7 Flow chart of pen movements

がなされている。

この図形情報処理用プログラムは別の独立したミニコンのユーザに影響を与えたり、それから影響を受けたりすることのないように DOS の常駐モニタ領域を約770語大きくしてその部分に入るようになっている。そして、プロッタの1基本動作終了による割込みおよび TTY 2 (図2) からの割込みによって動作する。

表3 コマンドとその機能

Table 3 Commands and their function

コマンド	機	能
Go		図形情報処理用プログラムを起動させる。
Stop	"	を一時中断する。
Continue	"	を Stop 状態から再起動する。
Kill	"	を中断し現在のテープの処理を中止する。
Next		次の画面の図形情報がくるまでテープを読みとばし、Stop 状態に入る。ペンの位置は手動でリセットする。

後者に関するコマンドの一覧を表3に示す。

まず、図形情報が出力された DEC テープをセットし TTY 2 から Go コマンドを入力する。ここで割込みが発生し DEC テープから 256 語 (1 ブロック) を単位とした図形情報の読み込みが行われ前述の方法によってプロッタの基本動作への分解が行われる。1つの基本動作ごとに情報がプロッタへ転送され、プロッタの1基本動作が完了すると割込みによりつぎの基本動作情報がプロッタへ転送される。プロッタへの1つの基本動作の転送からその動作終了による割込みまでの時間はミニコンは別のユーザへのサービスが可能となる。3.1 で示した1つのベクトルはプロッタの基本動作に変換すると一般に数十〜数百ステップになるので、1ブロック分の図形情報をプロッタに出力する時間はかなり長くなる。DEC テープから1ブロック分の図形情報を読み込むためには、数百ミリ秒要するが、読み込み間隔が長いのでこの時間が他のユーザに与える影響はほとんど無視できる。したがって、DOS との関係上 DEC テープから情報を読み込む間は他からの割込みをすべて禁止している。このプログラムの主要な部分をフローチャート表示すると図8のようになる。また、図形情報が出力された小型磁気テープの内容は図9のようになっている。したがって、表3に示したコマンドを用いて *i* 番目の画面を複数回描かせたり、*j* 番目の画面を描いている途中でそれを中断して *k* 番目の画面を描く作業に移るといったことが他のユーザのプログラムとまったく独立に処理できることになる。

3.3 図形処理用基本サブルーチン

前述の図形情報処理用プログラムは2点間を直線で結ぶ働きをするものであり、プロッタのユーザはこのプログラムを直接利用することはできない。そこで、プロッタのユーザ向けにごく基本的なサブルーチンを用意し、それを利用することでプロッタ用プログラム(図1)の作成を容易にするよう考慮した。なお、この場合に使用する言語としては FORTRAN を仮定す

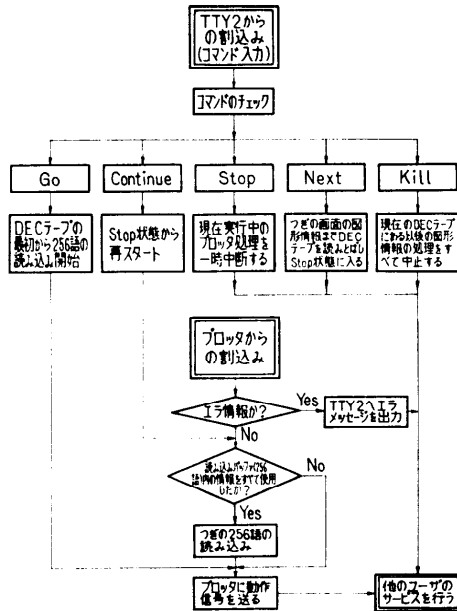


図8 図形情報処理用プログラムのフローチャート
Fig. 8 Flow chart of the plotter processing program

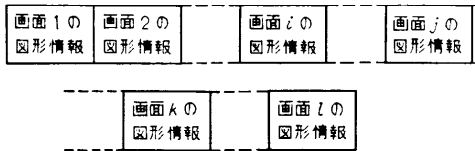


図9 小型磁気テープ上の図形情報

Fig. 9 Graphic Information on Small MT

べて CALL 文で処理することができるようになって
いる。以下にそれらの機能、利用方法について簡単に
述べる。

GRAPHS ルーチン

プロッタに関する種々の初期値を設定するルーチン
でプロッタを利用するプログラムの最初になければな
らない。

GRAPHE ルーチン

プロッタを利用するプログラムの最後に用いるルー
チンである。

PLOT ルーチン

ペンをアップ(ダウン)して現在の位置から指定さ
れた位置まで移動させる場合に使用するルーチンであ
る。最初にプロッタを起動させる場合のペンの位置(絶
対原点)は図10のようになっているものとして処理

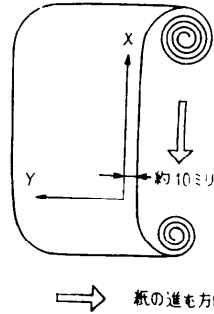


図10 絶対原点

Fig. 10 Origin of the coordinate

される。

ORIGIN ルーチン

ある座標系の任意の点を新しく原点と定義する場合
に使用するルーチンである。

ROUND ルーチン

現在の座標軸を指定された角度だけ回転して新しい
座標軸を定義する場合に使用するルーチンである。

SCALE ルーチン

ここで述べられている各ルーチンのうち座標を示す
値に関して倍率を与えるために使用するルーチンであ
る。

CHARAC ルーチン

指定した位置から指定した傾きで指定した文字を描
く場合に利用するルーチンである。文字として利用で
きるものは FORTRAN で標準的に使用可能なものに
限る。

SKIP ルーチン

1つの画面を描いてから新しい画面へペンを自動的
に移動させる場合に使用するルーチンである。

4. 例題

3.3 で述べた基本サブルーチンを利用した簡単な例
題を図11、図12に示す。図11は2つの画面を描く
ためのプログラムで、1つは $y = x^5 - x^4 - 25.12x^3 + 31.21x^2 + 124.5x - 22.5$ について、 $x = -5$ から $x = 5$ まで 0.125 おきに y の値を計算し直線近似した
もので、もう1つは (0, 60), (60, 0) の2点を結ぶ直
線を X, Y 座標を 5 度ずつ回転させて描かせたものと
文字列を 30 度ずつ回転させて描かせたものである。

なお、この図形を直接プロッタに描かせた場合と本
稿による方法で小型磁気テープに図形情報を出力した
場合の時間はそれぞれ 3,000 秒と 25 秒である。した

FORTRAN V003A

22-MAY-73

PAGE 1

```

C      PLOTTER TEST PROGRAM
C
0001      CALL GRAPHS
0002      CALL CHARAC(31HY=X-X-25.12X+31.21X+124.5X-22.5,31,
1          10.0,50.0,0.0,0.0,7.0)
0003      CALL CHARAC(1H5,1,27.0,58.0,0.0,0.0,3.0)
0004      CALL CHARAC(1H4,1,39.0,58.0,0.0,0.0,3.0)
0005      CALL CHARAC(1H3,1,81.0,58.0,0.0,0.0,3.0)
0006      CALL CHARAC(1H2,1,123.0,58.0,0.0,0.0,3.0)
0007      CALL ORIGIN(100.0,150.0)
0008      CALL PLOT(1,0.0,-80.0)
0009      CALL PLOT(2,0.0,70.0)
0010      CALL CHARAC(5HY*100,5,3.0,65.0,0.0,0.0,5.0)
0011      Y=60.0
0012      1 CALL PLOT(1,-1.0,Y)
0013      CALL PLOT(2,1.0,Y)
0014      IF(Y.EQ.-70.0) GO TO 3
0015      2 Y=Y-10.0
0016      IF(Y.EQ.0.0) GO TO 2
0017      GO TO 1
0018      3 CALL PLOT(1,-60.0,0.0)
0019      CALL PLOT(2,60.0,0.0)
0020      CALL CHARAC(3HX*1,3,55.0,-7.0,0.0,0.0,5.0)
0021      X=50.0
0022      4 CALL PLOT(1,X,1.0)
0023      CALL PLOT(2,X,-1.0)
0024      IF(X.EQ.-50.0) GO TO 6
0025      5 X=X-10.0
0026      IF(X.EQ.0.0) GO TO 5
0027      GO TO 4
0028      6 CALL SCALE(10.0,0.1)
0029      X=-5.0
0030      I=1
0031      7 Y=X**5-X**4-25.12*X**3+31.21*X**2+124.5*X-22.5
0032      CALL PLOT(1,X,Y)
0033      IF(I.EQ.1) I=2
0034      X=X+0.125
0035      IF(X.LE.5.0) GO TO 7
0036      CALL SKIP
C
0037      CALL ORIGIN(100.0,120.0)
0038      DO 10 I=1,72
0039      CALL PLOT(1,0.0,60.0)
0040      CALL PLOT(2,60.0,0.0)
0041      CALL ROUND(5.0,3)
0042      10 CONTINUE
0043      DO 20 I=1,12
0044      CALL CHARAC(5HABCDE,5,15.0,-2.5,0.0,0.0,5.0)
0045      CALL ROUND(30.0,3)
0046      20 CONTINUE
0047      CALL GRAPHE
0048      STOP
0049      END

```

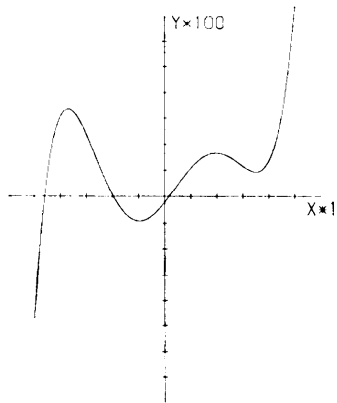
```

BLOCK      LENGTH
MAIN.      00789 (003052)*

```

図 11 プログラム例

Fig. 11 List of example program



$$Y = X^5 - X^4 - 25.12X^3 + 31.21X^2 + 124.5X - 22.5$$

図 12 出力図形の例

Fig. 12 An example of plotter output

がって、この例の場合には本稿の方法によると直接プロッタに描かせる場合にくらべミニコンが専有される割合は約 1/120 となる。

5. むすび

ミニコンとオンライン接続されたプロッタを含むシステムを効率よく利用する一つの試みとして、図形情報を小型磁気テープに出力し、それを割込みを利用して他の独立なプログラムと並列に処理する方法について述べたが、この場合、図形情報としてどのような構

造が適しているかについては深く考えていない。本稿では DOS の常駐部分をすこし大きくして図形情報処理用プログラムを組み込む方法をとったため、図形情報の構造としては比較的簡単なもの (3.1) となった。

図形情報とその処理プログラムとのかねあいについてはシステムの使用環境などにより種々変化すると思われるが、われわれとしてはしばらく使用経験をつんでみて図形情報とその処理プログラムのあり方について考えてみたいと思う。

最後に本システム作成に際し御助力いただいた本学高忠雄教授はじめ、同研究室の方々に深く感謝致します。

参考文献

- 1) DEC: PDP-11 Hand book, Digital Equipment Corp. (1970).
- 2) 石田晴久: 新しいミニコン PDP-11, bit, 臨時増刊号, ミニコン特集, pp. 1352~1356 (1971. 12).
- 3) G. Bell, et al.: A new architecture for mini-computers-The DEC PDP-11, Proc. of 1970 SJCC, pp. 657~675 (1970).
- 4) DEC: PDP-11 DISK OPERATING SYSTEM MONITOR PROGRAMMER'S HAND BOOK, Digital Equipment Corp. (1971).
- 5) 高沢嘉光: ナノピコ教室解答, bit, Vol. 5, No. 3, pp. 270~271 (1973. 3).

(昭和 48 年 6 月 1 日受付)

(昭和 48 年 7 月 14 日再受付)

○ 訂正

下記の原稿につき、以下のとおり訂正いたします。

〔原稿〕 本誌第14巻第9号719～726頁「ミニコンにおけるプロッタ制御の効率化について」(資料)

〔訂正箇所〕 726頁「図12」の右側に右図を加える。

