F 2. 対話形式のディジタル・アナログ・シミュレータ

高田　勝，大槻　説乎，工藤　和彦（九州大学工学部）

1・まえがき

アナログコンピュータは，大型ディジタルコンピュータが開発されてきた現在においても，研究者にとって連続系の解析などに有力な武器となっている．

アナコンは比較的小容量のものが多く，ユーザは機械を独占して，自分で操作し，結果を見て直ちにパラメータの変更やプログラムの修正を行うことができ，又ディジタルコンピュータの場合と違い数値計算の知識も不要である．

しかしスケーリングの必要があり，又使用する要素の精度（特に非線型性を持つ要素）や規模の大きなものの維持には難点がある．

一方ディジタルコンピュータにおいては，精度，容量の点では，アナコンに比べ有利であるが，プログラムやパラメータの変更が，バッチ処理方式では特に面倒であり，ユーザの手に結果がわたるＴＵＲＮ　ＡＲＯＵＮＤ　ＴＩＭＥも無視できない．

前述のアナコンの欠点をとり除くために，従来ディジタル・アナログシミュレータが多く開発されていて，その限りでは成功しているが，いずれも，ディジコンにおいてはバッチ処理方式の中で計算するため，小型機では独占できても大型機ではそれができず，その問題点は依然として残されている．

我々は，最近研究が進められているＴＩＭＥ・ＳＨＡＲＩＮＧ　ＳＹＳＴＥＭ（ＴＳＳ）の中でディジタルアナログシミュレータを用いれば，これらバッチ処理方式によって生じる問題は全て，解決するのではないかと考えた．

すなわちＴＳＳのもとでユーザが機械と一対一で，対話形式でプログラムを作りあげ，計算を実行させるシステムは，アナコンと全く変らない便利さを与え，又大量の計算機の使用を可能にして，これからの計算機の発展段階にも対応できると見られる．

我々はこの種の対話形式を用いたディジタル・アナログ・シミュレータ（略称＊ＤＤＡＨ）を九州大学中央計数施設のＫＩＴＡＣ−５０９０Ｈを用いて製作，実験した．機械の記憶容量の制限のため，その規模は小さいが，上記の特徴をもつのでここにその結果を報告する．

2・機械との対話

操作卓にかじりついて１ＳＴＥＰ，１ＳＴＥＰ機械語で対話しながらプログラムの虫取りをすることが，過去における人間と機械との対話であるが，アナコンにおける配線に相当

する作業としてのプログラシングがアナコンの使用経験者ならば，一度の説明で理解できるような機械語以外の言語を用いなければ，真の意味の機械との対話といえず，ディジタル・アナログ・シミュレータとして使いにくくなる．人間が機械を使うという立場から，プログラムを書き，それを任意の時点で修正でき，出来上った一部分を任意に実行できる．これらの要点を満たす言語が要求される．

1. ブロック線図を見ながら，直ちに入力できること．
2. プログラムの誤まりが，即座に指示され，その修正，又パラメータの変更が容易であること．
3. プログラムを任意の時点で，かつ部分的なテストができること．（これは大きなシステムのシミュレートの際など特に必要である）
4. 数値計算の知識，スケーリングが不要であること．
5. アナユンでの配線と同じく，入力順序がユーザの自由であること，又（日本人にとってＴＹＰＥ　ＷＲＩＴＥＲの使用は余り得意ではないので）入力言語はなるべく簡単明瞭であること．
6. アナユンに苦手なＥＬＥＭＥＮＴを多く備えなるべく高精度，大容量であること．

などが，我々が目標とした言語構造であり，システムである．

3．TSSの中での位置

この計画を進めるのに，シミュレータの処理プログラム（＊ＤＤＡＨ）は昨年のプログラシング・シンポジウムにおいて，九州大学から発表されたＴＳＳの基本方針に従って計画した．

すなわち，プログラムは，リロケーション（ＲＥＬＯＣＡＴＩＯＮ），リエントラント（ＲＥＥＮＴＲＡＮＴ）が可能であり，他のプログラムのデータ領域は，ＲＥＡＤ　ＯＮＬＹでなければならない．

多くのプログラムとデータ領域がメモリに共在することになるが，これらの管理はスーパバイザによって行われる．

ここで計画する＊ＤＤＡＨもこのスーパバイザの管理下で働くことになる．

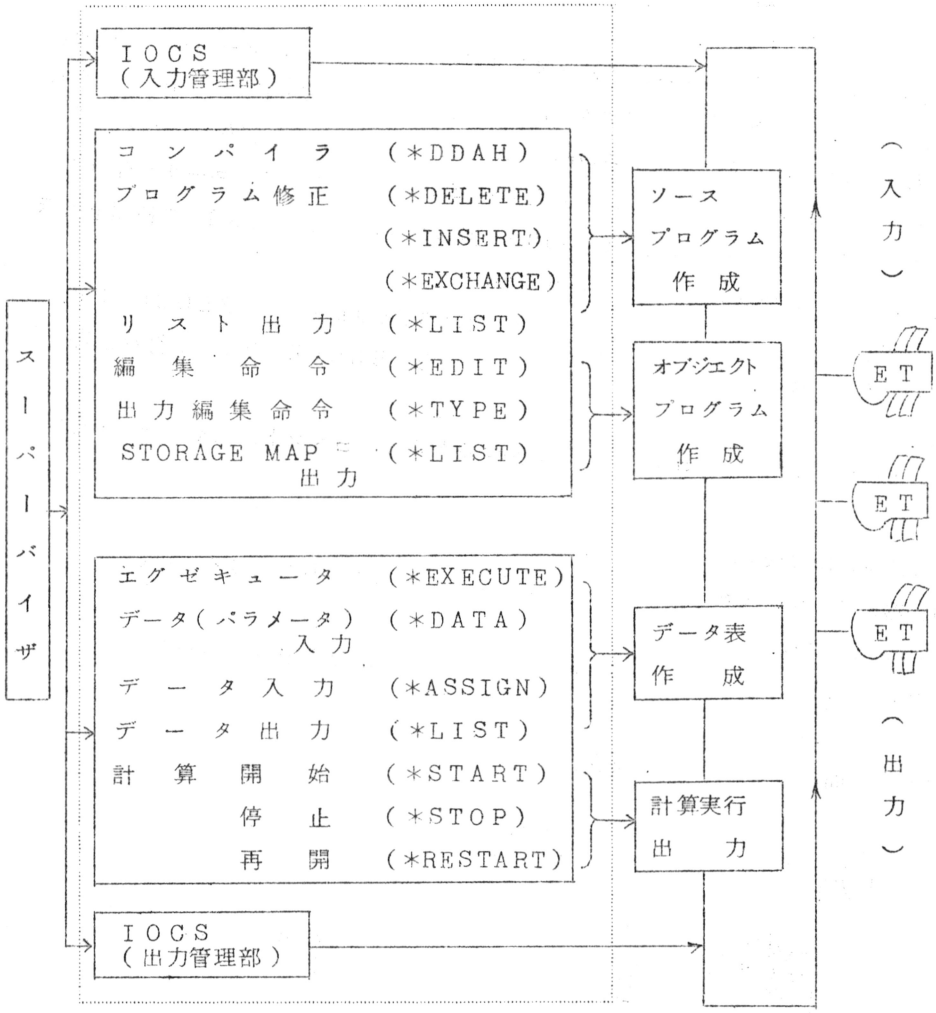
―― コマンドシステムについて ――

機械と対話して，プログラミングする際に，計算させるプログラム（入力プログラムと呼ぶ）とそのパラメータの値の他に，プログラムのフェーズの切換え，プログラムの修正，あるいは，リスト作成などのための指示が必要になる．

これら，入力プログラムとは別のレベルの入力を，スーパバイザを通して処理プログラムに処理させるために，種々のコマンドを用いる．入力に，コマンドを使ってある要求を

出すと，スーパバイザはプログラム内のコマンドルーチンにリンクを行って処理する．

＊ＤＤＡＨプログラムと他のプログラムのうち，関係するＩＯＣＳプログラム，およびスーパバイザとの関係を第１図に示す．スーパバイザは，この図の他のプログラムも同時に管理している．



（第１図）

4．言語構造

2.で述べた要件を満たす．入力プログラム用言語は次のような構造をとる．

まず，処理番号システムを取ったこと．ＴＳＳの中で入力プログラムの処理，入力プログラムを変更する際の処理を容易にするため，最小の入力単位ごとに，処理番号をつけて，必要な情報とともに格納する．

この処理番号は処理プログラムのフェーズが切換えられる時には，スーパバイザを通して受渡しされる．この処理番号システムで，実際は１つのＥＬＥＭＥＮＴ に１つずつ処理番号を与えて格納し，入力プログラムの内容は，最小限この単位ごとに処理される．＊

次にブロック図からの入力を容易にするため各 **ELEMENT** ごとに，ユーザが自由にＥＬＥＭＥＮＴにＢＬＯＣＫ　Ｎｏ.（0～300）を割りつけ，ＥＬＥＭＥＮＴ の種類を指示するため，アルファベットの略号をその後につける．

ＥＬＥＭＥＮＴ間の結びつけ（配線）はそれぞれのＥＬＥＭＥＮＴのＢＬＯＣＫ　Ｎｏ．ＴＹＰＥの入力の後に，そのＥＬＥＭＥＮＴへの入力先のＢＬＯＣＫ　ＮＯ．を符号と共に入力する．

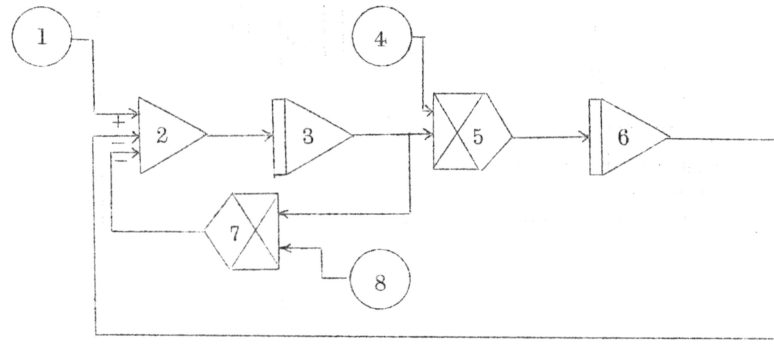
入力プログラムと前述のコマンド類を区別するために，コマンドにはその前に＊をつける．

これについては後で詳しく述べる．

積分器の初期値，定数，ＴＩＭＥ　ＤＥＬＡＹなどはそのＢＬＯＣＫ　ＮＯ．の後に値を入れて入力する．

また，計算を指示する値－プリント間隔，誤差の指示，計算終了の点－などは別にまとめて入力する．

これらの簡単な実例を第２，３図に示す．



（第２図）　２次遅れ系　ブロック図　　$G(S) = \dfrac{K}{1 + T_1 S + T_2 S^2}$

（機械による出力）　　（人間による入力）

○入力プログラム関係

```
<処理番号>      <BLOCK NO. TYPE INPUT>

    1
    2
    ⋮
   16          1,      CN,
   17          2,      SUM,    1, −6, −7,
   18          3,      INT,    2,
   19          4,      CN,
   20          5,      MUL,    3,  4,
   21          6,      INT,    5,
   22          7,      MUL,    3,  8,
   23          8,      CN,
    ⋮
```

○出力プログラム関係　（レコーダに相当する）

```
<処理番号>   <Header>
    ⋮
   27        ITME    INT−1    INT−2……
   28          5, SPACE(8)  4, SPACE(6) 4, ……
    ⋮
```

○パラメータ関係

```
<処理番号>   <BLOCK NO.    パラメータの値>
    ⋮
   32          1,        12. 0,
   33          3,         0. 0,    0. 001,
   34          4,         1. 3,
   35          6,         0. 0,    0. 003,
   36          8,         2. 65,
    ⋮
```

○計算を指示する値

```
<処理番号>   <計算開始点,  終了点,  最小キザミ巾, 出力間隔>
    ⋮
   40        0. 0,   125. 0,  0. 001,    2. 0,
    ⋮
```

（第３図）

＊プログラムの修正を行って挿入が行われるときには，修正開始の上限の処理番号に，１を加えてゆく．そしてその次は，この最後のケタに１を加えるというようにして，処理番号が一致しないように，しかも増加する順序になるようにつける．

5．使用計算機及び言語

1）使用計算機　　ŌＫＩＴＡＣ　５０９０Ｈ

内部記憶容量　　８０００語

外部記憶装置　　磁気テープ　４台

磁気ドラム　１台

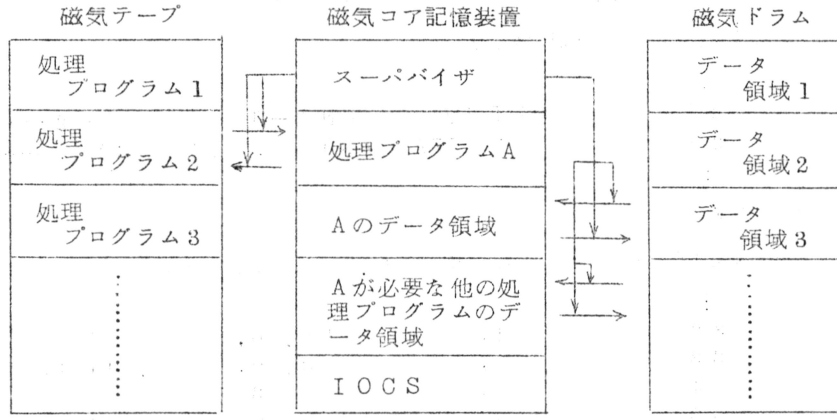
ワード構成　　４２ｂｉｔｓ／ｗｏｒｄ

スーパバイザはコアメモリに常駐し，必要に応じて，ドラム及び磁気テープからプログラム及びデータ領域をコアメモリに移す．

2）使用言語

＊ＩＰＨ　カードベースの機械語で浮動番地が使える．

第４図にその格納の様子を示す．



（第４図）

6．＊ＤＤＡＨの概要

○容　　量

使用ＥＬＥＭＥＮＴ総数　　３００個以内

（ＴＩＭＥ　ＤＥＬＡＹ，函数発生器などは，プログラムにより，大きさの制限が変る）

○積分ルーチン

積分ルーチンには梯形則を修正子とする予測子修正子法を用いる．収束の早さによってキザミの巾を変更する．

入力プログラムをソートすることにより，従来のディジタルーアナログ・シミュレータで用いられていた積分器を含む閉ループの中に１step delay 要素を入れる必

要がないようにしてある．

○ＥＬＥＭＥＮＴ

第５図に使用するＥＬＥＭＥＮＴの種類を示す．この他にユーザによって約１５種のＥＬＥＭＥＮＴのルーチンをつけ加えることができる．

エレメントの分類

種類	略号	入力	COMMENT（ℓ_0：出力）
積分器	INT	$\ell_1 \sim \ell_n$	$\ell_0 = \text{I. C.} + \int(\ell_1+\ell_2+\cdots+\ell_n)dt$
加算器	SUM	$\ell_1 \sim \ell_n$	$\ell_0 = \sum_{m=1}^{n} \ell_m$
符号変換器	NEG	ℓ_1	$\ell_0 = -\ell_1$
掛算器	MUL	$\ell_1 \sim \ell_n$	$\ell_0 = \ell_1 \times \ell_2 \times \cdots\cdots \times \ell_n$
割算器	DIV	ℓ_1 , ℓ_2	$\ell_0 = \ell_1 / \ell_2$
絶対値	ABS	ℓ_1	$\ell_0 = \lvert \ell_1 \rvert$
平方根	SQR	ℓ_1	$\ell_0 = \sqrt{\ell_1}$ $(\ell_1 \geqq 0)$
指数函数	EXP	ℓ_1	$\ell_0 = e^{\ell_1}$
自然対数	NLN	ℓ_1	$\ell_0 = \ln \ell_1$
三角函数	SIN	ℓ_1	$\ell_0 = \text{SIN}(\ell_1)$
	COS	ℓ_1	$\ell_0 = \text{COS}(\ell_1)$
	ART	ℓ_1	$\ell_0 = \text{ART}(\ell_1)$
リレー	OA	ℓ_1, ℓ_2, ℓ_3	$\ell_0 = \begin{cases} \ell_1 & \ell_2 \geqq \ell_3 \\ 0 & \ell_2 < \ell_3 \end{cases}$
	OB	ℓ_1, ℓ_2, ℓ_3	$\ell_0 = \begin{cases} 0 & \ell_2 \geqq \ell_3 \\ \ell_1 & \ell_2 < \ell_3 \end{cases}$
	IR	$\ell_1, \ell_2, \ell_3, \ell_4$	$\ell_0 = \begin{cases} \ell_2 & \ell_1 > 0 \\ \ell_3 & \ell_1 = 0 \\ \ell_4 & \ell_1 < 0 \end{cases}$
LIMITTER	LIM	ℓ_1, ℓ_2, ℓ_3	$\ell_0 = \begin{cases} \ell_2 & \ell_1 > \ell_2 \\ \ell_1 & \ell_2 > \ell_1 > \ell_3 \\ \ell_3 & \ell_3 > \ell_1 \end{cases}$
BANG・BANG	BB	ℓ_1 , ℓ_2	$\ell_0 = \begin{cases} \ell_2 & \ell_1 \geqq 0 \\ -\ell_2 & \ell_1 < 0 \end{cases}$
DEAD・SPACE	DS	ℓ_1, ℓ_2, ℓ_3	$\ell_0 = \begin{cases} \ell_1 - \ell_2 & \ell_1 > \ell_2 \\ 0 & \ell_2 \geqq \ell_1 \geqq \ell_3 \\ \ell_1 - \ell_3 & \ell_3 \geqq \ell_1 \end{cases}$
函数発生器	FA	ℓ_1	$\ell_0 = f(\ell_1)$
時間遅れ	TD	ℓ_1	$\ell_0 = \ell_1(t-T)$ (T:const)
定数	CN	—	$\ell_0 = \text{const}$

（第５図）

7．処理プログラム

処理プログラムはスーパバイザによって，コマンドで指示されて働らく．

これ以後は，コマンドを中心として，各処理プログラムの内容について述べる．

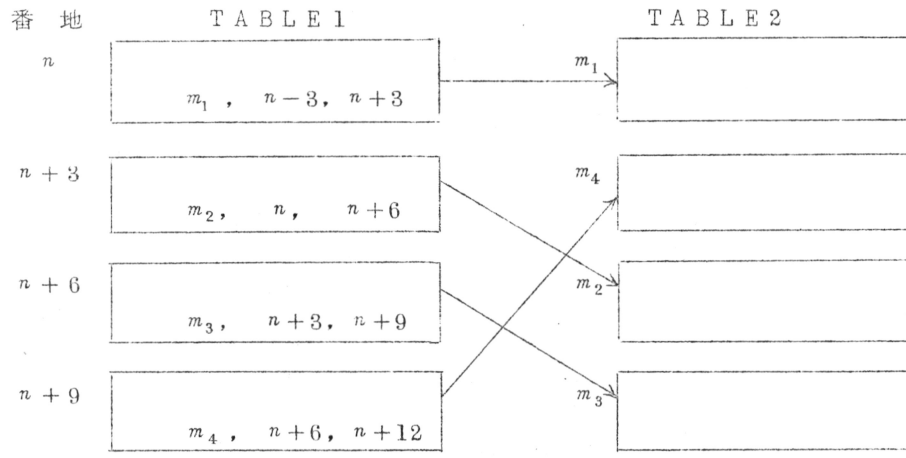
Ⅰ　コンパイラ

コンパイラは２つのＳＵＢ　ＰＲＯＧＲＡＭに分けられ，ソースプログラム作成プログラム（＊ＤＤＡＨ）と，オブジエクトプログラム作成プログラム（＊ＥＤＩＴ）とがある．

Ⅰ－Ａ－１　＊ＤＤＡＨ

入力プログラムが処理単位で入ってくる毎に，処理番号とともに情報をＴＡＢＬＥ１，ＴＡＢＬＥ２に格納する．同時にこのＴＡＢＬＥ１の番地がＢＬＯＣＫ　ＮＯ．ＴＡＢＬＥに書込まれる．ＴＡＢＬＥ１は３ＷＯＲＤＳずつのリストの形を持っていて，他のＥＬＥＭＥＮＴからの情報要求は，まずこのリストの中から探し出され，更にＴＡＢＬＥ２から変数番地，入力ＢＬＯＣＫ　ＮＯ．などがとり出される．

第６，７図にその様子を示す（n，mは番地）．



（第６図）

第７図の各ＴＡＢＬＥが作られソースプログラムとして格納される流れ図を第８図に示す．

Ⅰ－Ａ－２　＊ＤＥＬＥＴＥ

上限の処理番号，下限の処理番号を受取って，それぞれのＴＡＢＬＥ１の番地をしらべる．

ＤＥＬＥＴＥするＴＡＢＬＥ１のＤビットの部分に１を入れる．

上限と下限のそれぞれ上と下の処理単位のＴＡＢＬＥ１の番地をとり出して，それぞれ，前と後の処理単位のＴＡＢＬＥ１の番地の部分に書込んで，リストのつなぎを変更する．

使用するＴＡＢＬＥ

○ＢＬＯＣＫ　ＮＯ．ＴＡＢＬＥ　（全部で１００word）

BLOCKNO.1のTABLE1の番地	BLOCKNO.101のTABLE1の番地	BLOCKNO.201のTABLE1の番地
〃　2　〃	〃　102　〃	〃　202　〃
〃　3　〃	〃　103　〃	〃　203　〃
⋮	⋮	⋮
〃　100　〃	〃　200　〃	〃　300　〃

42　28　14

○ＴＡＢＬＥ１　（３word １単位）

処理番号				
E	D	BLOCK NO.	TYPE	TABLE2の番地
A		前の処理単位のTABLE１番地		後の処理単位のTABLE２番地

42　28　14

A：パラメータが必要なＥＬＥＭＥＮＴのとき，その常数表での格納番地が入る．

D：ＤＥＬＥＴＥされたとき，１が入る．

E：オブジェクトプログラム中に編集されたとき，１が入る．

○ＴＡＢＬＥ２　（word数　＝　入力ＥＬＥＭＥＮＴ数＋１）

E	入力ELEMENT数			V
S	C	BLOCK NO.	この入力ELEMENTのTABLE２の番地	V_1
S	C	〃	〃	V_2
			⋮	
S	C	〃	〃	V_n

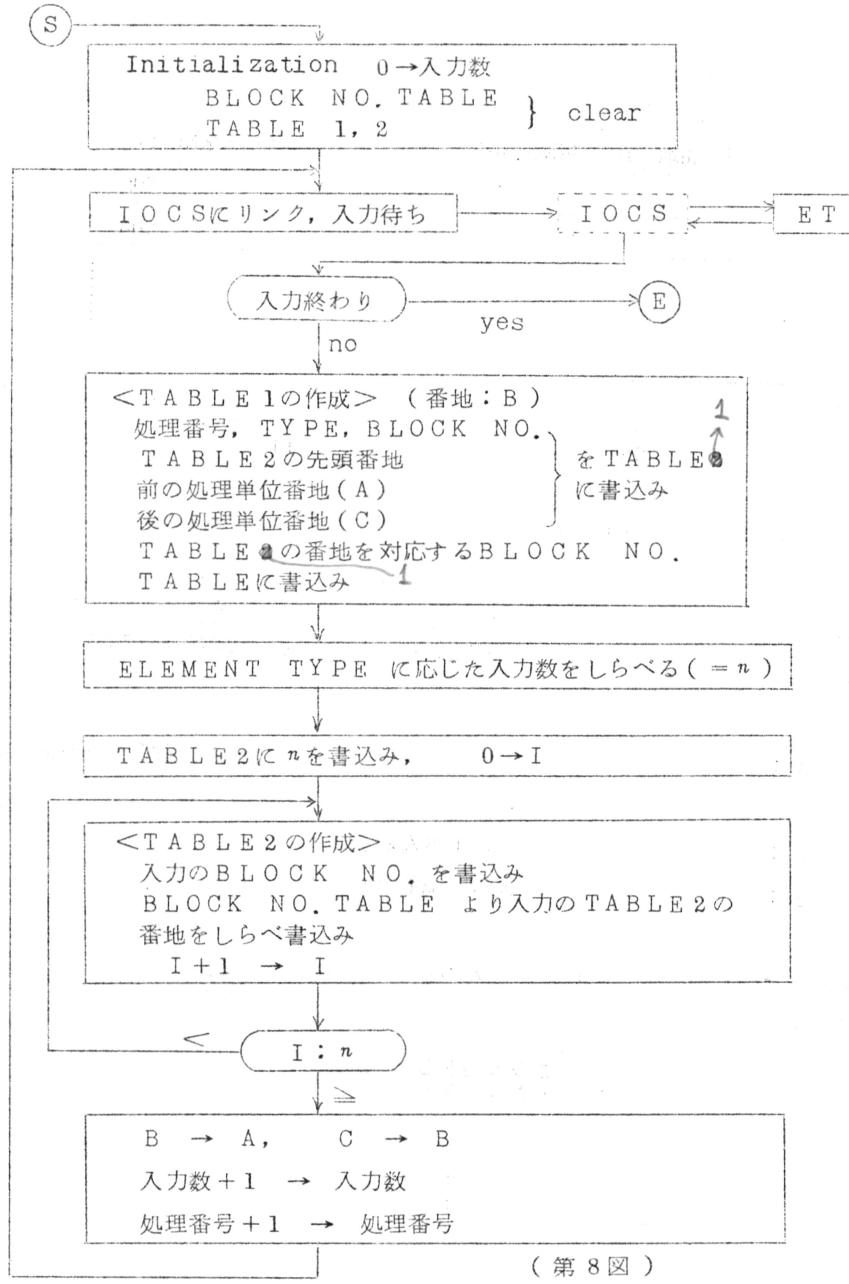
42　28　14

C：この入力ＥＬＥＭＥＮＴが編集されているとき，１が入る．

S：この入力ＥＬＥＭＥＮＴの入力のときの符号

V，V_1～V_n：このＥＬＥＭＥＮＴの出力値を格納してある番地

（第７図）



（第８図）

I－A－3　＊INSERT

上限の処理番号，下限の処理番号を受取って，それぞれのTABLE 1の番地をしらべる．

上限と下限の間に，処理単位が入っていたら，その部分をＤＥＬＥＴＥする．

上限処理番号＋．１　を出力して，入力待ち．

入力は，＊ＤＤＡＨと同様の処理を行う．

入力終りのコードが入ったら，上限と下限の間のリストのつながりを，植えつけておしまい．

Ｉ－Ａ－４　＊ＥＸＣＨＡＮＧＥ

上限の処理番号，下限の処理番号を受取って，それぞれのＴＡＢＬＥ１の番地をしらべる．

上限の処理番号を出力して入力待ち．

入力は，＊ＤＤＡＨと同様の処理を行う．但しＴＡＢＬＥ１の部分は，ＴＡＢＬＥの中味を変更するがＴＡＢＬＥ２は新らしい部分に書込んでゆく．

上限から下限までのＴＡＢＬＥ１，２をこうして変更してゆく．

Ｉ－Ａ－５　＊ＬＩＳＴ

上限の処理番号と，下限の処理番号を受取って，それぞれのＴＡＢＬＥ１の番地をしらべる．

上限のＴＡＢＬＥ１から順にＴＡＢＬＥ１，２をしらべて，＊ＤＤＡＨと逆の変換を行って，ＴＡＢＬＥ１，２の内容を出力する．ＴＡＢＬＥが書込まれている番地も出力する．

これは，＊ＤＥＬＥＴＥ，＊ＥＸＣＨＡＮＧＥなど，プログラム修正をたくさん行ったソースプログラムを整理して出力するのに用いられる．

Ｉ－Ｂ－１　＊ＥＤＩＴ

上限と下限で示された範囲のソースプログラムを正しい順序に編集して機械語になおし，導函数計算のオブジエクト　プログラムを作る．このとき，もし入力先がないＥＬＥＭＥＮＴや，論理的におかしい回路（遅延を含まない閉ループ，遅延を行うＥＬＥＭＥＮＴのみからなる閉ループ）が発見されると，編集不能であることを出力する．

オブジエクトプログラムも，ＲＥＡＤ　ＯＮＬＹの形で作成される．

同時に初期値，データが必要なＥＬＥＭＥＮＴには，常数要求表に，ＢＬＯＣＫ　ＮＯ．およびＴＡＢＬＥ１の番地が書込まれ，常数表の番地が割りつけられ，全ＥＬＥＭＥＮＴに，変数格納番地が割りつけられ，その番地が，ＴＡＢＬＥ１，２に書込まれる．

これを第９，１０図に示す．

○常数要求表

A	R	I	TABLE 2の番地	BLOCK NO.	TYPE	（常数表の番地が入る）
	R	I	〃	〃	〃	〃
	⋮		⋮	⋮		⋮
B	R	I	TABLE 2の番地	BLOCK NO.	0 0	（常数表の番地が入る）
	R	I	〃	〃	〃	〃

（400 words）

R：常数が要求されているかを示すビット

I：常数が入力されたかを示すビット

○常　数　表

A	浮	動	小	数	点	数	値
	〃						
	⋮						
B	浮	動	小	数	点	数	値
	〃						

（1000words）

○変　数　表

C	浮	動	小	数	点	数	値
	〃						
	⋮						
B	浮	動	小	数	点	数	値
	〃						

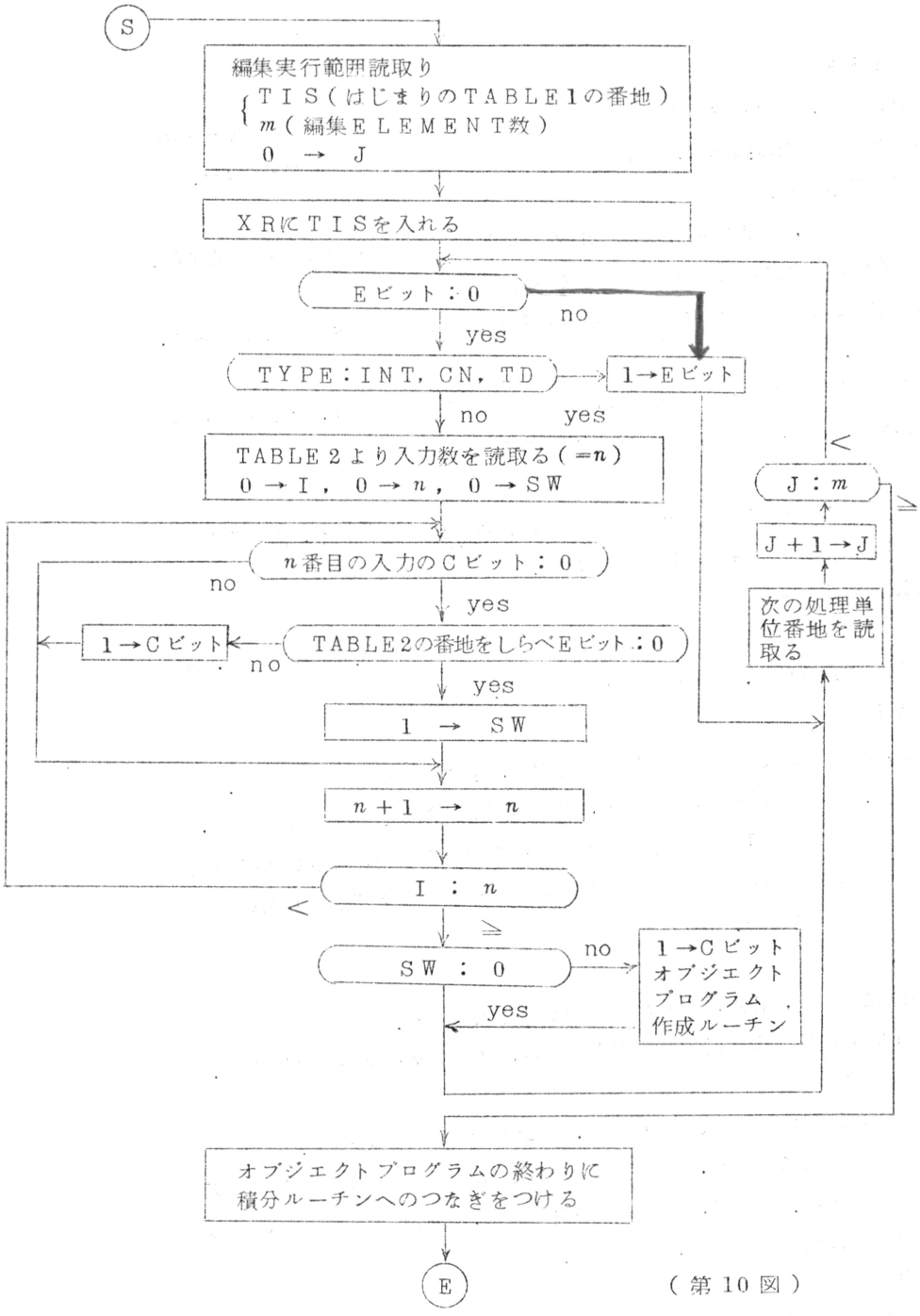
（4000words）

A：TIMEDELAY，CONSTANT，函数発生器，についてのデータを入れる．

B：積分器についてのデータを入れる．

C：積分器の他の全てのELEMENTについてのデータを入れる．

（第9図）


S
編集実行範囲読取り
{ＴＩＳ（はじまりのＴＡＢＬＥ１の番地）
m（編集ＥＬＥＭＥＮＴ数）
0 → J
ＸＲにＴＩＳを入れる
Eビット：0
no
yes
TYPE：INT, CN, TD
1→Eビット
no
yes
TABLE２より入力数を読取る（＝n）
0 → I , 0 → n , 0 → SW
J：m
J＋1→J
次の処理単位番地を読取る
n番目の入力のCビット：0
no
yes
1→Cビット
TABLE2の番地をしらべEビット：0
no
yes
1 → SW
n＋1 → n
I ： n
SW ： 0
no
1→Cビット
オブジェクトプログラム作成ルーチン
yes
オブジェクトプログラムの終わりに積分ルーチンへのつなぎをつける
E

（第 10 図）

Ⅰ−Ｂ−２　＊ＴＹＰＥ

これはアナコンでの，レコーダの部分にあたる．出力させる前に，ＨＥＡＤＥＲを印刷することができる．出力の指示は，出力させたいＥＬＥＭＥＮＴのＢＬＯＣＫ　ＮＯ．と出力させる桁数を指示する．

これらの指示が入ると，出力の編集命令をオブジエクトプログラムとして，格納する．

ＬＩＮＥ　ＯＶＥＲの場合，ＥＲＲＯＲとして示される．

これらの入力が全てＥＲＲＯＲを除かれて格納されると，インジケータがＯＮになって，次の入力待ちになる．

この後処理プログラムのフェーズが変って，＊ＥＤＩＴに移ったとき，インジケータがＯＮになっていなければ，ＥＲＲＯＲとして検出される．

次にエグゼキュートのフェーズになった後を説明する．エグゼキュートも，計算準備のフェーズと，計算実行，出力のフェーズに分けられる．

Ⅱ−Ａ−１　＊ＥＸＥＣＵＴＥ

前のコンパイルにＥＲＲＯＲがないかを，インジケータでしらべる．なかったら，変数，常数領域をＣＬＥＡＲして，入力待ちの状態になる．

Ⅱ−Ａ−２　＊ＤＡＴＡ

積分器の初期値，時間遅れ要素のＤＥＬＡＹ　ＴＩＭＥ，定数の値などを受入れる．

ＢＬＯＣＫ　ＮＯ．とともに値を入力すると，ＢＬＯＣＫ　ＮＯ．ＴＡＢＬＥから，相当する処理単位のＴＡＢＬＥ１の番地を探して前もって割当てておいた常数表に書込む．データは一部だけ変更することもできるから，パラメータを変えて計算したいときは，そこだけ入力してやればよい．

Ⅱ−Ａ−３　＊ＡＳＳＩＧＮ

計算する際の時間軸の始点，終点，プリント間隔などを入力する．

ＤＡＴＡ，ＡＳＳＩＧＮの値が全て正しく入力されたら，インジケータがＯＮになる．

Ⅱ−Ｂ−１　＊ＳＴＡＲＴ

＊ＤＡＴＡ，＊ＡＳＳＩＧＮが全て入力されたか，インジケータをしらべる．

常数表の必要な値を変数表にうつし，前に作られた導函数計算のオブジエクトプログラムを用いて積分を開始する．そして，プリント間隔ごとに，計算した値を出力編集のオブジエクトプログラムに従って編集し，出力する．

Ⅱ−Ｂ−２　＊ＳＴＯＰ

これはコマンドのレベルを持っているが，実際は，キーボード上の定められたキーを押すことにより，計算機は，計算，出力を停止して，入力待ちの状態になる．

Ⅱ−Ｂ−３　＊ＲＥＳＴＡＲＴ

一度ＳＴＯＰさせた計算を再び続けて計算させるときに用いる．

Ⅱ－Ｂ－４　＊ＬＩＳＴ

入力したＤＡＴＡ，ＡＳＳＩＧＮの値をまとめて出力する．計算前か後にパラメータの値をプリントしておくのに使える．

8．ま と め

全処理プログラムを，４月から設計，ＦＬＯＷ　ＣＨＡＲＴを書きはじめて，１１月中に一度できあがった．コンパイラの＊ＤＤＡＨ約１０００ＳＴＥＰ，＊ＥＤＩＴ約２０００ＳＴＥＰ，エグゼキュータ約１５００ＳＴＥＰ程度のものになった．

まだ実験段階ではあるが，対話形式で，プログラミングするのは，非常に作業が楽であり，誤まりが直ちに指摘されるので，修正も簡単であり，操作が楽しくもある．従来のデイジタルアナログシミュレータに用いられていた，積分器を含む閉ループの１ＳＴＥＰＤＥＬＡＹが不要であるので，ユーザに余計な負担が掛からずにすむのは，１つの特徴である．

またパラメータの一部変更が可能であるので，くりかえして同じプログラムで計算することが容易である．

将来への展望として，これは計算機のメモリの制約のために，大きさが限られてしまったが，理論的には，大型機にも拡張できる構造になっているので，その方向も大型機が導入された時に考えてゆきたい．

またＯＦＦ－ＬＩＮＥでのカーブプロッタは，現在もあるが，ＯＮ－ＬＩＮＥでのＧＲＡＰＨＩＣ　ＤＩＳＰＬＡＹ，自動的なパラメータの変更など考えてゆかねばならないと思う．

コーディングの際，番地部は全てＳＣＣ修飾，または相対番地の表現方法をとらねばならないので，いささか煩雑であった．

これはＲＥＥＮＴＲＡＢＬＥなプログラムを記述する際の言語の問題として考えるべきだと思う．

9．文　献

○牛島，有田，大槻，久原

「ＯＫＩＴＡＣ－５０９０Ｈによる タイムシェアリングシステムについて」

’６７年度　プログラミングシンポジウム報告集

○高　田

「プログラムによるＤＩＧＩＴＡＬ　ＤＩＦＦＥＲＥＮＴＩＡＬ　ＡＮＡＬＩＺＥＲ」

情報処理　vol 1 NO. 1 July 1960

○坂井，新美
「ブロックダイアグラムシミュレーションのためのプログラミングシステム」
電子計算機研究会資料　’６７ １月２１日

○R. T. HARNETT, F. J. SANSOM, L. M. WARSHAWSKY
「MIDAS PROGRAMMING GUIDE」
TECHNICAL DOCUMENTARY REPORT NO. SEG－TDR－64－1
’６４ １月

○ROBERT. D. BRENNAN, HARLAN SANO
「PACTOURAS－A DIGITAL ANALOG SIMULATOR PROGRAM FOR THE IBM 1620」
IBM RESEARCH LABORATORY

○牛島，有田
「ＡＬＧＯＬ 入出力実行命令について」
九州大学工学集報 第３８巻 第１号 ４０年３月

本 PDF ファイルは 1968 年発行の「第 9 回プログラミング一シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の捜索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者捜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日 〜 2021 年 3 月 19 日
掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程
https://www.ipsj.or.jp/copyright/ronbun/copyright.html