

B7 ファンクションキーによる会話型 デジタルシミュレーション

矢田光治, 田中 隆(電子技術総合研究所)

平塚尚一(東京都立工業技術センター)

国岡利行(東京理科大学)

1. はじめに

連続系システムの動的過程を解析するには、アナログ計算機を用いる方法が適している。しかし、システムが複雑・大規模になってくるとアナログ計算機による解析は困難となり、デジタルシミュレーションの必要性が生じてくる。

現在デジタルシミュレーション言語³⁾としてMIDASやMIMICやCSMPなど数多く作成され、発表されている。これらはカード入力によるバッチ処理形式がとられているため、プログラムやパラメータの変更が難しくデータの処理性に乏しい。またEASLのように入出力装置にディスプレイを使用してマンマシン・コミュニケーションを良くしようとしているものもある。デジタルシミュレーションを行う場合、会話型のシミュレーション言語を使用すれば、デジタル計算機に不馴れな人でもプログラム作成が容易に行うことができる。

会話型言語¹⁾とはプログラムを記述するステートメントを、タイプライターやディスプレイなどの端末機からオンラインで入力して、計算機と対話しながらプログラムを作成するとき使われるプログラミング言語である。この言語は、人間と計算機の共同作業によって能率よく行えるようにするため、プログラムの誤りを見つける機能と修正する機能をもっている。またプログラムばかりでなく、データについても、その削除、挿入などの機能も備えている。

本報告では、デジタルシミュレーションを会話型で行うとき、ファンクションキー³⁾を用いてコマンドを入力し、XYプロッタでブロック線図を描きながら、シミュレーションプログラムを作成する方法について述べる。

この方法では、入出力タイプライタ上にあるファンクションキーを押すことによってそれに対応した積分器や加算器などがXYプロッタに描かれ、その接続端子の参照を指定することによってブロック線図の配線が組上るものである。したがって、タイプライタのファンクションキーを押すとブロック線図が描かれると同時に、それがシミュレーションプログラムとなり、ブロック線図が完成した後でデータを与えるとシミュレーションの結果が得られるものである。

2. 特 徴

デジタルシミュレーションを行なう場合、人間と計算機とが対話しながらプログラムを作成し実行する会話形式の処理を行なえば作業をより強力に進めることができる。

使用者の立場からすると、シミュレーション言語としては誰もがすぐ計算機を使用できるようなやさしくて使いやすい言語が必要となる。このため使用者の要求を考慮し次にあげる事項に重点をおいて本システムを作成した。

(1) 入力形式を会話型とすることによって計算機と対話しながら試行錯誤的に問題を解くことができ、初心者でも容易に使用できる。また、このシステムではパラメータが入力される度にチェックを行ない、誤りを見つけるとすぐにその旨を知らせてくるため使用者はすみやかに修正を行なうことができる。

(2) 入力言語はブロック線図向きである。このため連続系の解析を行なう数値計算の知識は不要となる。すなわちこの言語では、ブロック線図のブロック名を入力し、続いてそれらを結線するだけでプログラムが作られる。なお、入力順序はブロック名の入力後に結線を行なうこと以外自由である。この方式では使用者が計算機に不馴れでも簡単に扱える。

(3) タイプライタにファンクションキーの機能を備える。ファンクションキーには積分器などのブロックの機能が組込んであり、キーを押すことによって各キーに応じた機能が引き出されてくる。このキーを使用すれば入力が確実となり、タイピングの軽減が得られる。

(4) 端末装置としてタイプライタとXYプロッタを組合せて使用する。タイプライタから入力した情報をブロック記号や線としてXYプロッタに描くことで、プログラム作成の過程を確認できる。同時にXYプロッタに描かれるブロック線図を見ながらプログラミングできるので視覚的な思考の断続ができる。さらに、XYプロッタに計算結果を出力することによりアナログ出力として直感的に結果を知ることができる。この出力はXYプロッタに自動スケーリングで描くことができるのでアナログ計算機のような操作は全く必要ない。

3. 言語使用

3.1 エレメント

このプログラム言語は、コマンド、パラメータ、ブロックアドレス、定数、予約名、演算子、ブロック名、関数等のエレメントから成り立っている。

(1) コマンド

コマンドは使用者がシステムに指示を与えるもので次の8種類がある。

SYMBOL	ブロックを入力する
/SYMBOL	すでに入力されたブロックを修正する
CONNECT	入力されたプログラム間の結線を行う。
/CONNECT	すでに行われた結線の修正を行う
DATA	入力されたパラメータやブロックに値を与える
/DATA	すでに与えた値の修正を行う
REWRITE	プログラムのリストをタイプライタに出力し、ブロック線図をXYプロッタに書き直す。

GO 計算を実行する

これらのコマンドは、タイプライタにコマンド入力可能を示す“*”が出力されたら、いずれを入力してもよい。

(2) パラメータ

パラメータには次の4種類がある。

システムパラメータ

ブロックパラメータ

コネクトパラメータ

関数パラメータ

システムパラメータは、システムから指示を待つメッセージが出力されたときにユーザが入力するパラメータで通常YES OR NOを意味するYまたはNである。

ブロックパラメータは、ブロックをセクションのどこへ配置するかを指示するブロックアドレスである。

コネクトパラメータは、ブロック間を結線するためブロックの位置をブロックアドレスで表わしたものである。

関数パラメータは関数に値を与える定数である。

(3) ブロックアドレス

XYプロッタに書かれたセクション上の位置を表わす番地でセクションの行を示す1~5の数字と、列を示すA~M(Lを除く)の英字から或る2文字の英数字。英字、数字はいずれを先に書いてもよい。

(4) 定数

ここで取扱われる定数はすべて8桁以下の実数型定数である。入力形式は整数型、実数型のいずれでもよく、実数型は固定小数点または浮動小数点で表示される。その大きさの範囲は

$$10^{-77} \leq | \text{定数} | \leq 10^{77}$$

である。

(5) 予約名

予約名はこのシステムで使用する特別の意味を持つ変数名である。これら予約名には次の5つがある。

FT(final time) シミュレーションの打ち切り時間

CP(curve plots) XYプロッタに出力する端子を持つブロック名を指示する

IT(output interval) タイプライタ出力の間隔

XF(X-axis scale factor) 時間軸の縮尺

YF(Y-axis scale factor) 変動軸の縮尺

(6) 演算子

演算子には次の7種類がある。

+, -, /, *, (,), =, 9

(7) ブロック

ブロックは単純な基本機能を有するもので、9種類のブロックが用意されている。ブロックの種類を表1に示す。

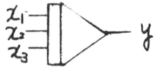
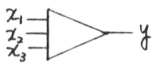

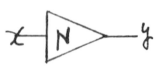
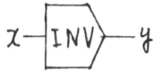
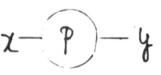
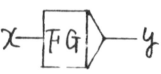

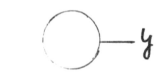
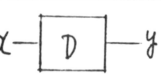
ブロックの種類	ブロック記号	ブロック記号	コメント
積分	INT		$y = y_0 + \int \sum x_m dt$
加算	ADD		$y = \sum x_m$
掛算	MUT		$y = x_1 x_2$
符号変換	NEG		$y = -x$
逆転	DINV		$y = \frac{1}{x}$
係数	POT		$y = dx$
関数発生	FG		$y = f(x)$
信号発生	SS		$y = f(t)$
定数	CON		$y = Const$
一時遅れ	UF		$y = x(x-T)$

表1 ブロックの種類

(8) 関数

関数発生器の関数として SIN, COS, exp, log, ln, arctan, $\sqrt{\quad}$, 絶対値の8種類が、また信号発生器の関数として、ステップ、ランプ、インパルス、正弦波の4種類が用意されている。

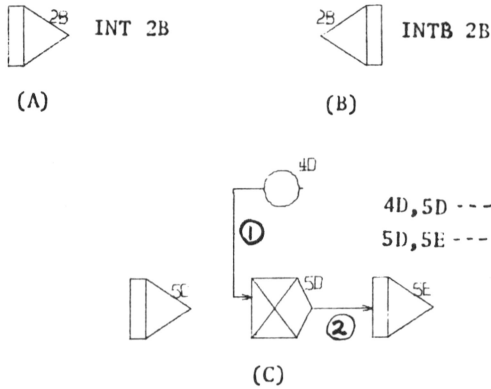


図 2

係数器の係数，積分器の初期値，一時遅れの時間，関数発生器および信号発生器の関数，打ち切り時間，XYプロッタに描く結果の出力端子などである。コマンドを入力するとシステムからメッセージと共にデータの必要なパラメータをタイプアウトする。ユーザはそれに従って次の形で入力する。

[ブロックアドレス]=[定数]

または

[ブロックアドレス]=[関数]

入力したプログラムやデータを修正するには修正用コマンドを使用する。

ブロックの修正は/SYMBOLで行う。すでに描かれたブロックを他のブロックに書換えるにはファンクションキーを押した後，同じブロックアドレスを入力すればよい。抹消するにはブロックパラメータ入力後抹消コマンド/を入力する。

結線の修正は/CONNECTで行う。新たに結線する場合コネクタパラメータを入力するだけでよい。結線を抹消するにはコネクタパラメータ入力後，抹消コマンド/を入力する。

データの一部を修正するには/DATAを入力後，修正したいブロックアドレスにデータを与える。打ち切り時間，出力の指示は予約名を使用して変更する。

3.3 出力形式

この言語の出力形式にはXYプロッタにグラフのみを出力するものと，同時にタイプライタに数字を印字するものの2種類があり，先に述べたコマンドDATAによってユーザが指示することができる。

(i) XYプロッタ

XYプロッタにはブロック線図の任意のブロックを指示することによってそのブロックの出力を描くことができる。グラフは時間軸方向に20cm，変動軸方向に15cmの大きさで，最大5本までの出力を同時に描くことができる。グラフのスケールは自動とユーザ指定のい

ずれでもよく DATA の中で指示される。グラフが描かれると、変動軸方向の最大値、最小値と時間軸の打ち切り時間がそれぞれグラフ上に書かれる。

(2) タイプライタ

タイプライタには、XYプロッタに描かれた出力の値を指数部付きの浮動小数点形式で出力する。なお出力ステップは DATA で与えられる。

4. 使用例

簡単な例として、図3に示すRLC直列回路の過渡解析を行う。この回路の微分方程式は次のようになる。

$$L \frac{d^2 q}{dt^2} + R \frac{dq}{dt} + \frac{q}{c} = E$$

図4はブロック線図を作るためタイプライタから入力するインプットプログラムである。まず、SYMBOLコマンドによってブロックを入力し、続けてCONNECT コマンドでそれらの結線を行う。さらにDATAコマンドによってパラメータに値を与えたり、計算終了時間、出力端子などを指示している。このプログラムによってXYプロッタに描かれたブロック線図を図5に、計算結果を図6に示す。

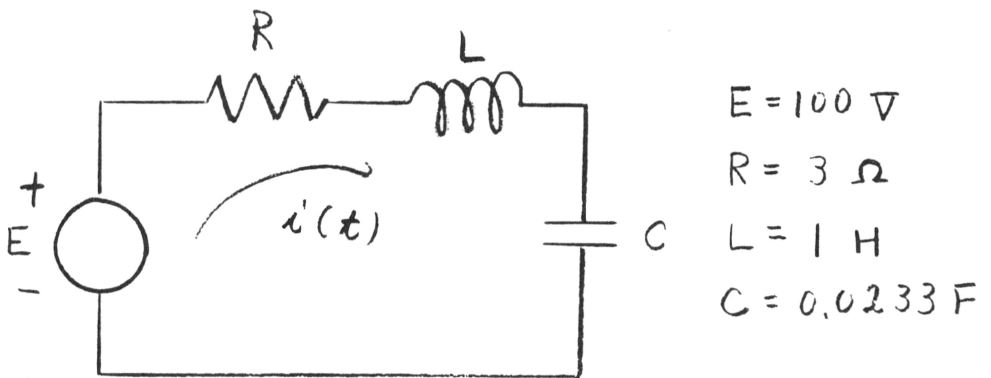


図 3

*** DIGITAL SIMULATION ***

SECTION YES OR NO?

N

READY FOR INPUT

*SYMBOL

con 3c
 negb 5f
 add 3d
 con 4d
 negb 2e
 div 4e
 mutb 2f
 mut 3f
 dinvb 1g
 int 3g
 mutb 5g
 con 1h
 int 3h
 con 5h

*CONNECT

3c, 3d
 3d, 3f
 4d, 4e
 2e, 3d
 4e, 3f
 2f, 2e
 3f, 3g
 1g, 2f
 3g, 3h
 1h, 1g
 3h, 2f
 3g, 5g
 5h, 5g
 5g, 5f
 5f, 3d

*DATA

constant 3c,4d,1h,5h
 3c=100
 4d=1
 1h=0.02533
 5h=5

initial value 3g,3h
 3g=0
 3h=0

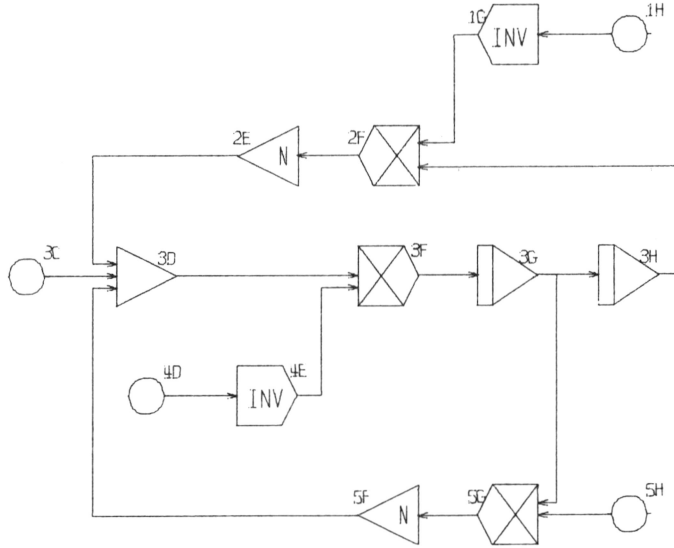
final time (sec) ft
 ft=2

curve plots cp
 cp=3h

scale factor
 auto

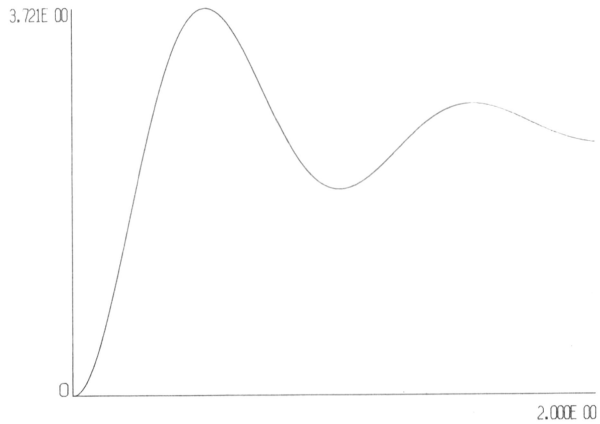
values?
 no

*GO



☒

5



☒

6

☒

4

5. おわりに

会話形式に加えてタイプライタにファンクションキーの機能を持たせたことによってプログラミング作業が非常に楽になり、プログラムの組上る過程を目で見られるのでプログラムミスがすぐに指適できた。タイプライタはこの言語のために設計されたものではなく、一般に用いられる入出力タイプライタのためファンクションキーは各ブロックの頭文字を代用している。システムから見れば何を入力されてもかまわないが、リスト上に意味を持たない文字を印字するとユーザが混乱するので使用するのを避けた。このため逆向きのブロック入力の場合余計な操作を行わなければならなかった。

このシステムではXYプロッタに出力する部分が多いため、慣れてくるとプロッタの動きが遅く感じられてくる。特に入力の際用いるセクションのプロットは時間がかかるので、通常セクションを印刷したコーディング用紙にブロック記号を記入し、セクションのプロットを省略してブロックを入力している。ブロック間の結線はプロッタを見ながら行うのでプログラミングにはさしつかえない。XYプロッタの代りにディスプレイ装置を使用すればブロック入力や結線をすみやかに行うことができるが、CRTに表示できる量に制限があるため大規模なシステムのブロック線図を一度に表示しようとするとう具合が悪い。

この言語は使用法が簡単なため会話形式でなくても機能を損うことはない。したがってバッチ処理による使用も可能である。

今後の課題として、部分的なシステムを個々に保存しておき、それらを結合して大きなシステムのシミュレートを行えるようにしたい。また自動制御系に应用する場合、これらのブロック線図からこの言語用のブロック線図への変換は慣れないと手数がかかるので自動的に変換を行えるよう検討したい。

参 考 文 献

- 1) 矢田, 会話型言語, ソフトウェア科学, Vol. 3, No. 9, 1971.
- 2) 矢田, 田中, 平塚, Function keyによる会話型言語, 昭46年電気学会全国大会.
- 3) Y. Chu, Digital Simulation of Continuous Systems, McGraw-Hill, 1969.

言語	作成者	形 式				計算機	備 考	年号
		入 力	出 力	プログラムの形式	積分方式			
ASTRAL	Parker	カード	プリンタ	コンパイラ	四次	IBM 704	フォートラン演算機 コンパイラ導入	58
	Steim				Runge Kotta			
DYSAC	Rose	カード	プロッタ	コンパイラ	四次	CDC 1604		61
	Horley				Runge Kotta			
MIDAS	Skiles	磁気テープ	プリンタ	タイプ	五次 Miln	IBM 7090 IBM 7094	ディジタル・シミュレーション普及、実用化に貢献	64
	Harnett				Predictor			
PACTOLUS	Sansom	カード	プリンタ	タイプ	Corrector	IBM 1620	小型計算機との	64
	Warsawsky	カード			二次			
EASL	Bernnan	タイプライタ	プロッタ	タイプ	Runge Kotta	IBM 7090 IBM 7094	Man Machine	64
	Sashkim	センススイッチ			Runge Kotta			
MIMIC	Schlesinger	タイプライタ	プロッタ	コンパイラ	Runge Kotta	IBM 7090 IBM 7094	MIDAS改良	65
	Harnett				Miln			
CSMP	Sansom	カード	プリンタ	コンパイラ	Runge Kotta	IBM 7090 IBM 7094		65
	Warsawsky	カード			他 5種			
CSSL	Syn	カード	プリンタ	コンパイラ	Runge Kotta	IBM 360 IBM 1130	ディジタル・シミュレーション言語を標準化したもの	67
	Wyman	カード			他 7種			
CSMP	Brennan	カード	プリンタ	コンパイラ	Runge Kotta			67
	Simulation Software Committee				四次			
CSSL	Committee	カード	プリンタ	コンパイラ	Runge Kotta			67

本 PDF ファイルは 1972 年発行の「第 13 回プログラミング・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、権利者の検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、一定期間の権利者検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>