

UDC 681.32.06:62-52

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-156

自動制御系の動特性を計数型電子計算機に
より模擬するための汎用プログラム

戸川隼人・丸山治雄

1969年3月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

既 刊 資 料

TM-120	円錐管レンズの設計とその応用	1967年11月	山中 龍夫, 奥 岨 澄 男
TM-121	大きなマトリクスの逆行列計算および連立一次方程式の計算のためのプログラミング技術	1967年11月	戸川 隼人, 戸川 保子
TM-122	NAL-7-P ロケットの強度および燃焼試験結果	1967年11月	竹中 幸彦, 古 田 敏 康 小川 敏一, 朝 田 洋 雄 原 恒彦, 田 代 富 克 湯沢 克宜, 伊 藤 富 克 西村 久男
TM-123	“NAL-16・31” および “NAL-25・31” 二段ロケットの振動試験	1967年11月	中井 暎一, 古 関 昌 次 田寺 木一, 高 木 俊 朗 安藤 泰正, 森 田 孝 峯岸 正勝, 菊 地 孝 太田 幹雄
TM-124	吹出式超音速風洞の集合胴圧力制御について	1967年12月	外立 政隆, 近 藤 洋 史 原 亘 利
TM-125	航技研 1m×1m吹出式超音速風洞におけるハーフモデル試験について	1968年 2月	石原 久蔵, 原 亘 利 榊原 盛三, 関 根 英 夫
TM-127	2024-T3アルミニウム合金の3-bay 有孔補強平板の軸荷重による疲労特性	1968年 4月	飯田宗四郎, 猿 本 光 明 斉藤信一郎
TM-130	リフトエンジンの自然吸込みについて	1968年 4月	近 藤 博, 大 城 章 一 郎
TM-131	遷音速タービン翼列二次元試験(Ⅲ)	1968年 5月	近 藤 博, 養 田 光 弘 山崎 紀雄, 古 川 昇
TM-132	行列の最小固有値の一計算法	1968年 5月	戸川 隼人, 戸川 保子
TM-133	フィラメント・ワインディング円筒の強度特性に関する研究	1968年 5月	竹中 幸彦, 斉 藤 浩 一 古田 敏康, 川 島 矩 郎 三本 茂夫, 小 川 敏 一 越出 慎一, 朝 田 洋 雄 態倉 郁夫, 大 竹 邦 彦
TM-134	AGARD 標準模型 HB-1, HB-2 の超音速風洞試験	1968年 5月	斉藤 秀夫, 石 原 久 蔵 原 亘 利, 野 田 順 一 関 根 英 夫
TM-135	歪ゲージの自己加熱による歪ドリフト	1968年 5月	小川 敏一, 遠 藤 修 司
TM-136	“NAL-16・31” および “NAL-25・31” 二段ロケット結合部の曲げ剛性試験	1968年 5月	中井 暎一, 飯 田 宗 四 郎 高木 俊朗, 安 藤 泰 勝 菊 地 孝 男
TM-137	ロケットの三次元の運動の方程式および HITAC 5020 による軌道計算のためのプログラム	1968年 5月	毛利 浩
TM-138	片持板の振動解析に関する考察	1968年 6月	堀 武敏, 越 出 慎 一 林 洋 一
TM-139	薄板構造の疲れき裂伝ば実験	1968年 6月	竹内 和之, 野 原 利 雄 飯田宗四郎
TM-140	二次元スラットおよびスロテッドフラップの実験的研究 (Ⅱ)	1968年 7月	犬丸 矩夫, 北 村 清 美
TM-141	超音速二次元翼列予備実験風洞について	1968年 7月	近 藤 博, 坂 口 一 八山 優, 高 森 晋
TM-142	二次元スラットおよびスロテッドフラップの実験的研究 (Ⅲ)	1968年 7月	犬丸 矩夫, 高 橋 侔
TM-143	端面一体巻きフィラメント・ワインディング容器の静圧強度および疲れ強度	1968年 8月	竹中 幸彦, 朝 田 洋 雄 野口 義 男
TM-144	真ひずみ計の試作	1968年 8月	竹中 幸彦, 朝 田 洋 雄 野口 義 男
TM-145	スピンを伴うロケットの運動を計算するプログラム	1968年 8月	戸川 隼人, 石 黒 登 美 子 山本 浩 通

自動制御系の動特性を計数型電子計算機 により模擬するための汎用プログラム*

戸川隼人**・丸山治雄**

概 要

ブロック図の形式で書かれた自動制御系の動的なシミュレーションを、デジタル計算機を用いて、容易に且つ効率よく実行するための汎用プログラムを作成したので報告する。

この種のプログラムとしては、他の計算機用のものは、これまでも既に数多く試作され実用化されたものがあるが、ここに報告するプログラムは、当所の大型計算機 HITAC-5020 のために作成したもので、更に下記の特色がある。

(1) どんな機種にも適用できること。

これまでに開発されたプログラムは、いずれも特定の計算機に専用のものであったが、本プログラムは basic FORTRAN で書いてあるので、他の機種にも、ほとんどそのまま適用することができる。

(2) 使い易いこと。

形式は CSMP や DSL-90 とほぼ同型で、FORTRAN で書かれたプログラムとの接続や混用が可能であり、ことに FORTRAN 経験者には使い易い。

(3) 機能が豊富

機能的には DSL-90 とほぼ同程度であるが、更に大きな問題まで扱うことができるように工夫した。

(4) グラフ表示

計算結果を XY プロッタで描かせることができる。

(5) 部分的に他のプログラムに流用可能

全体が、「サブルーチンのセット」として構成されているので、他のプログラムに一部分だけを流用することが可能である。

本資料では、このプログラムの設計方針、仕様、計算法、使用法、計算例などについて述べる。

1. ま え が き

本資料は「アナログ計算機的センスでデジタル計算機を使う」ために作成した“SSDDS”という名のプログラムについて述べる。

自動制御系の動的なシミュレーションには、一般的に云えば、デジタル計算機よりもアナログ計算機の方が適しており、実際、これまで、アナログ計算機が、この種の目的によく使われてきた。しかし、非常に大

規模なシステムの解析をアナログ計算機で実行しようとする、種々の困難な問題に遭遇する。第一に、問題の規模に応じた非常に大型の計算機が必要になり、このため多額の経費を要する。第二に、大きな問題になるほど、パッチ盤のワイヤリングやポテショメータのセットが大変になり、ことに、それが正しくセットされているか否かを点検する作業が困難になる。第三に精度の問題があり、アナログ計算機も以前と比較すれば格段に精度的に向上してはいるが、まだ多くの問題を残している。

もしも、これをデジタル計算機で処理すれば、上記の第一の点については、ほとんど問題なく、たとえ

* 昭和44年1月27日受付

** 計測部

ば、ここに報告するプログラムは、積分器2000台、任意関数発生器数百台、乗算機は事実上は無数個、というような巨大なアナコンに匹敵する。第二の点に関しては、たしかにデジタル計算機の方にもプログラミングやデバッグというような、やっかいな仕事があるが、それでもアナログ計算機のパッチ盤と比較すれば、はるかに明快で簡単なものである。第三の点、すなわち精度について云えば、デジタル計算機の方が文字どおり「桁ちがいに」良い。

以上のような利点のある反面、デジタル計算機は欠点として、実物の制御機器類をシミュレーションの中に容易には組み込めないという問題があり、また、アナログ計算機に慣れた人から見ると、デジタル計算機には一種の「便りにくさ」がある。

上記の二大欠点のうち、前者は、構造の相違に基づく問題なので、特別なリンケージを付けない限り解決できないが、後者については、適当なソフトウェアを用意すれば、かなり改善することができる。本資料で報告するプログラムは、この後者の欠点をカバーするためのものである。

使い易さ、使いにくさ、といっても、慣れや主観が入るので一概には云えないが、少なくとも下記の2点に関しては、アナログ計算機の方が有利であろう。

(1) 計算の経過が直接に監視でき、それを見ながらパラメータの設定、変更、などを容易に行なうことができる。要するに、マン・マシン・システムとして、すぐれている。

(2) 制御系の

実物——等価モデル——計算機（演算要素）
の対応が明快で、わかりやすい。

上記(1)に関しては、これも特別なハードウェアを付けられない限り根本的には解決されないが、(2)に関してはソフトウェアでカバーすることができる。本資料で問題にしているのは主として上記(2)の点である。

デジタル計算機を用いて制御系の動的シミュレーションを行なうためには、まず制御系の特性を、連立常微分方程式の形に書き、次にそれを数値的に解くためのプログラムを作らなければならない。一方、これまでの自動制御関係の研究者の多くは、ブロック図と伝達関数、という形でシステムを考えたり解析したりしており、連立常微分方程式を立てるとか、それを数値的に解くとかいうことには、あまり慣れていなかった。アナログ計算機ならば、その点、ブロック図と伝達関数という考えで、そのまま計算機を動かせるので、非常に良かった訳である。

そこで、デジタル計算機を、なるべくこれに近い形で使えるようにするためのソフトウェアを開発した。これを用いれば、ブロックの結線と、特性のパラメータを入れるだけで、デジタル・シミュレーションを行なうことができる。

このような着想は、著者らのものが最初ではなく、既に1960年頃から、主として米国で試みられ、しだいにその価値が認められて来た。この種のプログラムは、その後、非常に多くの大学、研究所、会社等で開発され、1965年に書かれた「デジタル・シミュレーション用語、その批評と利用案内」と題した論文²¹⁾には31種類がリストアップされている。わが国でも、東大宇宙航研¹⁶⁾、九州大学²⁾、京都大学、石川島播磨重工、日立中研⁷⁾、三菱電機¹¹⁾、その他各所で、このようなプログラムが開発されている。また IBM の MIDAS というシステムだけでも既に 100箇所以上のユーザに使用されているという⁸⁾。これらのプログラムのうち、代表的なものを表1に示す。

これらのプログラムのアプリケーションとしては、航空宇宙関係が非常に多い。その数例を表2に示す。いずれも、かなり大規模な計算で、アナログ計算機を増設すべきか、ハイブリッド方式にすべきか、いろいろ検討した上で、経済的理由等により、デジタル計算機を用いることにしたのであろう。

デジタル・シミュレーションのプログラムも、初期のものには、いろいろ不便な所や精度的に良くないものなどがあったが、その後、これらの問題点が、しだいに解決され、使い易く、しかも安心して使えるものになった。その決定版とも云うべきものは、現在、IBM 社の計算センタで一般向に公開して利用されている CSMP, DSL-90, および MIDAS であろう。しかしこれらは IBM の計算機専用であるので今回、当所の計算機にかけられる新しいシステムを開発することにし、その機能および容量などに関しては、上記システムと同等以上の水準を目標にした。日立中研の三巻氏が最近(昭和43年9月)に5020用のシステム³⁴⁾を発表したが、これは積分間隔自動調節機能を持っている事、ソーティングを実施している事など、優れた点がある反面、規模としては小さく、ブロック数は 200個という制限があり、機能や表現方法は非常にアナコン的で、FORTRAN 的自由さが無い。また、石川島播磨重工で開発した BOSS というシステム (UNIVAC 1107 用) を借用し、HITAC 5020 用に書き換えて使用したこともあったが、当所の使用条件に適しない面があり、やはり自分の所で使うものは自分の所で開発する

表 1 これまでに発表された制御系シミュレーション・プログラムの例

発表年代	プログラム名	対象計算機	作成した所	文献
1955	—	IBM 701	USNOTS	
1957	DEPI	DATATRON 205	JPL	3 2
"	DIDAS	IBM 704	ロッキード社	
1958	ASTRAL	IBM 704	コンベア社	
1959	DEPI-4	IBM 704	アリス・シャルマ社	
"	DYANA	IBM 704	GE	
1961	BLODI	IBM 704	ベル研究所	
"	DYSAC	CDC 1604	ウィスコンシン大学	2 5
1962	DES-1	(専用機)	SDS	2 2
"	DYNASAR	IBM 7090	GE	
"	PARTNER	IBM 650	ハネウェル社	
1963	DAS	IBM 7090	マーチン社	3 1
"	—	IBM 7090	三菱電機	
"	DIAN	IBM 7074	アイオワ大学	
"	JANIS	IBM 7090	ベル研究所	
1964	COBLOC	CDC 1604	ウィスコンシン大学	2 9
"	EASL	IBM 7090	エアロスペース社	
"	FORBLOC	—	ウィスコンシン大学	
"	HYBLOC	IBM 7090	"	
"	MADBLOC	IBM 7090	コロラド大学	
"	MIDAS	IBM 7090	ライト・パターソン空軍基地	2 8
"	PACTOLUS	IBM 1620	IBM	2 7
"	PLIANT	IBM 7090	"	
"	SCADS	G-20	カーネギー研究所	
"	SIMTRAN	IBM 7090	WRE	
1965	BLODIB	IBM 7090	ベル研究所	3 0
"	DES-1	SDS-9300	SDS	
"	DSL/90	IBM 7090	IBM	2 3
"	MIMIC	IBM 7090	ライト・パターソン空軍基地	
"	SADSAC	B-5000	USAF	
"	SLASH	B-5000	USAF	
"	UNITRAC	IBM-7094	マーチン社	
"	—	LGP-30	早大	9
1966	CODIS	PRODAC-580	ウェスチングハウス社	3 3
	DDS	HIPAC-103	日立	
	HYBALL	PRODAC-580	ウェスチングハウス社	3 3
	MDAS	IBM-7044	三菱重工	8
	MIDAS-III	IBM-7094	コンベア社	
	Mod-IV	IBM-7040	リットン社	2 6
	SAHYB	—	CETIS	
1967	BOSS	UNIVAC-1107	石川島播磨重工	
	OLBA	(実験機)	東大宇宙研	1 6
	—	OKITAC-5090	九大	2
1968	CSMP	IBM-360	IBM	1 9
	DDAH	OKITAC-5090	九大	3

表 2 航空宇宙関係への応用例

発表者	使用プログラム	応用例	備考	文献
Hurley 他	DYSAC	自動着陸システムの解析, 特に, オート・フレアの研究	機体, 対機速度, オートパイロット, 及びフレア・コンピュータを含む系	2 5
Palevsky 他	DES-1	パルス・ジェットによる姿勢制御の解析	最適な回路常数を決定するために使用しているようである。	2 2
Stein 他	(無名)	ミサイルのオートパイロットの解析		2 4
Syn 他	DSL/90	サターンVの発射から, 第一段終了までのシミュレーション	制御用計算機, レートジャイロ特性, アクチュエータの特性, エンジン特性, 風による外乱を考慮している。	2 3

のが最も良いという結論になった。これまでに作られたプログラムと比較すると, 本プログラムは, 冒頭の要約で簡条書きにしたような利点がある。

この簡条書きの(5)について少し補足説明を加えると, デジタル・シミュレーションのブロック図の処理方法には種々のやり方があり, 実行時にインタープリティブに処理するもの, 計算前に一種のコンパイルのような処理をして実行用プログラムをジェネレートさせるもの, その他各種の方法があるが, 本プログラムでは, すべての要素を, FORTRAN 文法における「FUNCTION 型サブルーチン」として作成し, 全体を FORTRAN コンパイラに処理させる, という独自の方式をとった。したがって, そこにあるのはサブルーチンの集まりであって, 全体をまとめて使用すればシミュレーション用システムとなるが, 部分的に, たとえばノイズ・ジェネレータとかヒステリシス・ループとか, そういう, 単独に使用できるものは他のプログラムの中に流用することも可能である。

この利点, および独自の構成を表現するため, このプログラムに, SSDDS (Systems of Subroutines for Dynamic Digital Simulation) という名前を付けた。DDS あるいは, これに類した名前は, この種のシステムによく用いられている(表1参照)が, 頭に SS と付くところに特色がある。

2. ブロックの設計

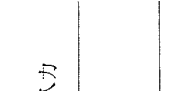
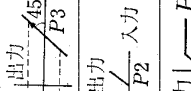
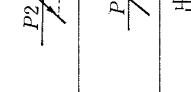

アナログ計算機は, 積分器, 加算器, 乗算器, 関数発生器, スイッチング要素, その他から構成されてい

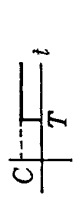
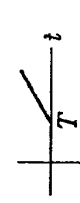
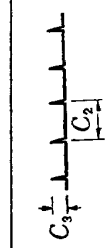

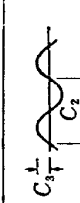


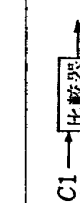
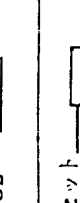
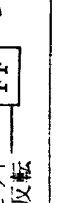
る。したがって, デジタル計算機を「アナログ計算機的センス」で使うためには, 上記の機能に対応するものさえ作ってやれば一応は間にあうわけである。

しかし, 実際に研究者が制御系の解析をやっているのを見ると, 研究者が考えている「制御系モデルのブロック図」と, それをアナログ計算機にかけるための「計算機パッチ盤配線用のブロック図」は必ずしも同一ではない。たとえば, 研究者は最初は, センサ, アクチュエータ, フィードバック機構, 被制御体特性, といった単位でシステムを考え, 次にその個々の特性を適当なモデルで置き換えてゆく。そこで使用される数学的モデルとしては, 線形の特性的をもつ部分に関しては伝達関数, 非線形の部分に関しては入出力の関係図, あるいは, バックラッシュ, ヒステリシスというようなもので, これらを図式的に書いたものが, いわゆる「制御系のブロック図」である。これをアナログ計算機にかける場合には, アナログ計算機の単一の要素の段階まで, このブロック図を分解し, 一段と複雑な「パッチ盤用ブロック図」を作らなければならない。

デジタル計算機で, シミュレーションを行なう場合, 最終目標は「制御系を解析すること」であって, 「アナコンのまねをする」ことではないから, なるべく元の「制御系のブロック図」のまま使用できることが望ましい。先に, 現代の代表的なデジタル・シミュレーション・プログラムとしてあげた CSMP, DSL-90, MIDAS のうち, MIDAS は完全にアナログ計算機そのままの機能しか持たせていないのに対し, CSMP や DSL-90 は, もう少し大きな, まとまった

表3 使用できるプログラムの種類と指定方法

機能	数学的内容 (伝達関数)	書き方 (関数名および引数の並び方)	式数の意味, 使用上の注意事項等 (Xは入力, YOは出力側 の初期値, tは時間)	合計 2000
積分	$\frac{1}{S}$	SEKIBN(X, YO)	$\int_0^t X dt + YO$ が出力となる。	}
時限積分	$\frac{1}{S}$	SEKIMD(X, YO, A, B)	$\left. \begin{matrix} A=1 \\ B=0 \end{matrix} \right\}$ の時だけ積分を行なう。 $\left. \begin{matrix} A=0 \\ B=0 \end{matrix} \right\}$ なら最後の結果をホールド。	
一次遅れ	$\frac{1}{C \cdot S + 1}$	REALPL(X, YO, C)	$C \cdot \dot{Y} + Y = X$ の解 Y が出力となる。	100
二次遅れ	$\frac{1}{S^2 + C_1 C_2 S + C_3}$	CMPXPL(X, YO, C1, C2)	$\ddot{Y} + (2 \cdot C_1 \cdot \dot{Y} + C_2 \cdot Y) C_2 = X$ の解 Y が出力となる。	100
リード・ラグ	$\frac{C_2 S + 1}{C_1 S + 1}$	SLDLAG(X, YO, C1, C2)	$C_1 \cdot \dot{X} + X = C_2 \cdot \dot{Y} + Y$ の解 Y が出力となる。	100
微分	$C \cdot S$	DERIV(X, YO, C)	$\frac{dX}{dt}$ が出力となる。	100
むだ時間	$e^{-c \cdot s}$	DELAYV(X, T)	T 秒前に入った値が出る。500ステップまでの遅延が可能。	20
零次ホールド	$\frac{1}{S} (1 - e^{-st})$	ZRHOLD(X, C)	C=1 ならば, 入力をそのまま出力側に出す。 C=0 ならば, C=0 になった瞬間の入力の値を保持する。	100
任意関数発生	数表の形で与えた関数の一次補間	FUNC1(F, X, N)	} 任意関数 f(x) の数表を, F(1, I) = x _i , F(2, I) = f(x _i) の形で, 定義しておく, その x=X に対する値を補間して出力する。 N は数表のサイズ(点数)。	—
任意関数発生	同, 二次補間	FUNC2(F, X, N)		—
ヒステリシス		HYSTSS(X, P1, P2, P3)	上り坂では Y=P3, 下り坂では X=P2, すき間では最終値をホールドする。なお P1 は出発時の状態 (出力変数値)	100
不感帯		DEADSP(X, P1, P2)	$\left. \begin{matrix} P_2 < X \\ X < P_1 \end{matrix} \right\}$ ならば 出力は X-P2 } その他の所では出力0 $\left. \begin{matrix} X < P_1 \\ X > P_2 \end{matrix} \right\}$ ならば 出力は X-P1 }	—
飽和		GLIMIT(X, P1, P2)	$\left. \begin{matrix} X < P_1 \\ X > P_2 \end{matrix} \right\}$ ならば 出力は P1 } その中間では入力をそのまま出す。 $\left. \begin{matrix} X < P_1 \\ X > P_2 \end{matrix} \right\}$ ならば 出力は P2 }	—
階段化		QNTZR(X, P)	$P \cdot \left[\frac{X}{P} - \frac{1}{2} \right]$ が出力となる。ただし [] はガウスの記号。	—

ステップ入力		STEP(T, C)	$t < T$ ならば 出力はゼロ $t \geq T$ ならば 出力は C	—
ランプ入力		RAMP(T)	$t < T$ ならば 出力はゼロ $t \geq T$ ならば 出力は $t-T$	—
パルス発生器		GIMPLS(C1, C2, C3)	$t < C1$ ならば 出力はゼロ C_2 は パルス間隔, C_3 はパルス高, (パルス幅はゼロ)	—
矩形波発生器		PULSE(C1, C2, C3)	C_1 は パルス間隔, C_2 はパルス幅, C_3 はパルス高	—
正弦波発生器		SINE(C1, C2, C3)	$t < C1$ ならば 出力はゼロ。 C_2 は周期, C_3 は位相 (出力は $\sin(C_2(t-C_1) + C_3)$)	—
雑音発生器	[0, 1]における一様乱数を発生する。	UNNOIS(N)	N は擬似乱数発生のための初期値で, 正の奇数を入れる。 異なる N を用いければ独立な乱数を得る。同一の N なら再現性あり。	100
正規分布の雑音発生器	指定した平均値, 標準偏差に従う正規乱数発生	GRNÖIS(N, C1, C2)	N は上と同じ。C1 は標準偏差, C2 は平均値。	100
3点リレー		FUNCSW(P, X1, X2, X3)	$P < 0$ ならば 出力は X1, $P = 0$ ならば 出力は X2 $P > 0$ ならば 出力は X3	—
2点リレー		RELAY(P, X1, X2)	$P < 0$ ならば 出力は X1 $P \geq 0$ ならば 出力は X2	—
比較器		CÖMPAR(C1, C2, C3)	$C_1 < C_2$ ならば 出力はゼロ $C_1 \geq C_2$ ならば 出力は C_3	—
フリップフロップ		RSTFF(R, S, T, V)	$R > 0$ になれば リセット, $R > 0$ で $S > 0$ になればセット, $R > 0$ 且つ $S < 0$ で $S > 0$ ならば反転。セット時出力は, V, リセットで0	100
(例外) 切換スイッチ		これは CALL で呼び出す。 OUTSW(C, X, Y1, Y2)	$C < 0$ ならば $Y1 = X, Y2 = 0$ $C \geq 0$ ならば $Y1 = 0, Y2 = X$	—

(備考 制限個数欄が—印のブロックは何個使ってもよい。)

単位でブロックを扱えるようになっており、たとえば「二次遅れ要素」というような形でコーディングすることができるようになってきている。

この考え方をおし進めると、「レートジャイロ」とか「油圧ポンプ」とか「人間パイロット」とか、そういう物に対応する典型的なモデルを使っておいて、適当なパラメータを入れれば、ただちに計算できるようなシミュレータを作ることも可能であろう。しかし、簡便で的確なモデルというものは、計算の目的等により、かなりの差があるのが普通であるから、上記のような、完全に「実物的」なシミュレータが、どの程度の利用価値があるか、いささか疑問がある。

むしろ、「ブロック図の中に、任意の形の伝達関数を書ける」ことの方が有意義であろう。このようなデジタル・シミュレータは昨年、九州大学で計画されたが、本格的にこれをやるためには、シミュレータ自身がコンパイラおよび数式解析プロセッサな機能を持つ必要があり、それは非常に手間がかかるので、ついに実現せず、結局今回の当所のシステムと同等程度、すなわち、微積分、むだ時間、一次遅れ、二次遅れに任意のパラメータを入れられる、というだけになってしまった²⁾。

そこで、現在かなり広く使用され、割合に評判の良い、CSMP と DSL-90 を参考にして、表3のような構成にした。

簡単に説明すると、SEKIBN が通常の積分器に相当するもの、REALPL, CMPXPL は、それぞれ

real pole, complex pole の略で、一次遅れ要素、二次遅れ要素を表わし、SLDLAG (lead lag, DERIV (微分), DELAY (むだ時間要素), ZRHOLD (零次ホールド), HYSTSS (ヒステリシス) 等の機能があり、矩形波やノイズなどのジェネレータ、任意関数発生器、フリップ・フロップ、各種のスイッチング機能などがある。

なお、この表に記載された機能のほかにFORTRANのすべての機能が使用できる。たとえば、この表には加算器や乗算器、符号変換器などは、特にあげてないが、後の例で見られるように、数式の形で自由に、何箇所にも書くことができる。また FORTRAN の組み込み関数、たとえば三角関数、双曲線関数、指数対数関数、絶対値、等が使用可能であり、また必要に応じて、ベッセル関数、ガンマ関数などのサブルーチンを組み合わせることも可能である。

3. 計 算 法

まず最も基本になる積分法について述べる。これまでに開発された同種のプログラムで使用されている積分方式は表4のとおりであるが、これらの使用経験について報告された結果を総合すると、つぎのような結論が得られる。

- (1) 単純なオイラの公式では精度が不十分である。
- (2) RKG (Runge-Kutta-Gill)法ならば、大体問題なく使える。
- (3) 誤差評価をするためには、Milne 法などの、

表4 これまでに発表されたデジタル・シミュレーション・システムに採用されている積分公式

積 分 公 式	適用例 (その積分公式を使用できるシステム名)
オ イ ラ 法	DIDAS, DYANA, PARTNER, JANIS, FORBLOC, NADBLOC PLIANT, DBS-1, DSL/90, CSMP
ルンゲ, クッタ法 (RKG 法)	DEPI, ASTRAL, DYSAC, WIZ, HYBLOC, MIMIC EASL, MIDAS, DSL/90, TELSİM, MDAS, DDS CSMP
ミ ル ン 法	MIDAS, MIMIC, DSL/90, CSMP
ア ダ ム ス 法	DYNASAR, CSMP
可変ステップサイズ	MIDAS, MIMIC, SADSAC, SLASH, DDAH, DSL/90 CEMP
何種類かの公式を 選択できるもの	DES-1 (5種類), COBLOC (3種類), DSL/90 (8種類), CSMP

いわゆる予測子修正子法が便利であるので、これらを用いている例も多い。しかし、これらの公式は場合によって異常振動、発散、というような現象を起こす可能性があり、その対策についていろいろ研究されてはいるが、まだ完全には解決されていない。

- (4) 積分間隔(時間キザミ)を可変にする公式が用いられている例があるが、これは用いて便利な場合と不都合な場合がある。
- (5) 結局、あまり決定的な良い方法がないため、4~5種類の公式を用意し、利用者にその選択をまかせる、というのが最近の傾向である。しかし、プログラムを作る側から言えば、これは技術的に非常に困難な問題をもたらす結果になり、ことに今回のシステムのような単純な構成では不都合な面が多い。

以上の事情から、今回は、とりあえず、RKG法だけを用いることにした。最初は上記(5)相当のものを試みたが、各種の公式を、どれでも使えるようにするためには、積分器だけでなくすべての要素を大幅に改造する必要があり、その手間の割に効果はあまり期待できないので、とりやめた。

なお、一次遅れ要素および二次遅れ要素に関しては、RKG法だけによらず、常数係数線形常微分方程式の厳密解を利用できる部分はそれを採用した。これが数値積分の解より格段に精度がよいことは言うまでもない。参考までに、ここの微分の計算式を表5に示す。

微分に関しては現在のところ、仮に

$$\frac{f(t+\Delta t) - f(t)}{\Delta t}$$

を微係数としている。これでは精度的に、あまり好ましくないことは事実であるが、高次の多項式近似による微係数算定は、場合によっては異常値あるいは逆符号値を発生する原因になるので、一応、安全のため、上記の最も簡単な式を用いておいた。これまでに作られたデジタル・シミュレーションのプログラムについて調べてみると、微分機能を持ったシステムは比較的少なく、MIDASでさえも持っていないようである。DSL-90では放物線近似により微係数を算出している^{18), 23)}。

次に任意関数発生器であるが、これまでのシステムについて調査した結果は、DES-1, PACTOLUS, 等は一次補間のみであるが、DYSAC, MIDAS, DSL-90等においては、一次補間と二次補間を選択して使用できるようになっている。(二次補間に統一しなかったのは、微係数の場合と同様、異常値を避けるためである)。また、独立変数の個数については、DES-1が二変数関数を許しているほかは、いずれも一変数に限定している。今回のシステムにおいては、一応、DSL-90なみに、一変数関数の一次補間および二次補間による任意関数発生器を用意したが、当所のライブラリ・プログラムには、高次補間および多変数関数補間用のプログラムを用意してあるので、これらと組みあわ

表5 厳密解を使用したブロックとその計算式

ブロック名	伝達関数	計算式	備考
REALPL	$\frac{1}{CS+1}$	$1 - \exp\left(-\frac{t}{c}\right)$	ステップ入力 に対して適用
CMPXPL	$\frac{1}{S^2 + C_1 C_2 S + C_2}$	$\left\{ \begin{array}{ll} 1 - \frac{1}{2\sqrt{\xi^2-1}} \left\{ \frac{\exp(s_1 t)}{\xi - \sqrt{\xi^2-1}} - \frac{\exp(s_2 t)}{\xi + \sqrt{\xi^2-1}} \right\} & \text{for } \xi^2 > 1 \\ 1 - \left\{ \cos \nu t + \left(\frac{-\lambda}{\nu}\right) \sin \nu t \right\} \exp(\lambda t) & \text{for } \xi^2 < 1 \end{array} \right.$ <p>ただし、引数 S_1, S_2 と上式中の $\nu, \lambda, \xi, s_1, s_2$ の関係は、 $\nu = S_2 \sqrt{1 - S_1^2}$ $\lambda = -S_1 \cdot S_2$ $\xi = S_1$ $s_1 = -S_1 \cdot S_2 + i \cdot \sqrt{S_1^2 - 1} \cdot S_2$ $s_2 = -S_1 \cdot S_2 - i \cdot \sqrt{S_1^2 - 1} \cdot S_2$</p>	同上

注) 榎木義一, 砂原善文: 自動制御工学 I (養賢堂 1965) による。

せて使用すれば、さらに複雑な問題をも扱うことができる。

なお、一様乱数発生には Lehmer の方法^{12),14)} 正規乱数の発生には Box-Muller の方法^{13),15)}を採用した。

4. 入出力形式

ブロック図の接続関係および諸係数をデジタル計算機に入れる方法としては、最近ではコンピュータ・グラフィックスを応用して、ライトペン等で、図を直接にコンピュータにインプットさせる研究などがなされているが¹⁶⁾、このような理想的方式はまだ一般に採用されていないようで、従来のシステムで主として採用されてきた方式としては、

(1) 各ブロック毎に、そのブロックの機能と諸係数、接続関係を示す固定形式のカード（普通は各1枚）を用いる。

(2) 数式の形で、かなり自由なフォームで入れる。というようなものがあり、たとえば MIDAS¹⁷⁾、BLODIB³⁰⁾、DES-1²²⁾などは前者に属し、DSL-90¹⁸⁾、CSMP¹⁹⁾、ASTRALなどが後者の形式によっている。どちらが便利かということは、人の好みとか慣れなどの関係で一概には云えないが、FORTRAN を使っている人ならば後者の方がはるかに便利で使い易く自由度が大きい。今回のシステムでは、(2)の方の形式を採用した。この(2)の形式にも、いろいろな種類があり、以前は、積分器にはすべて番号を付けなければいけない、というような不便なシステムが多かったが、最近では、(今回のシステムも)単に SEKIBN と書くだけで済むように改良されてきている。

次に、計算結果であるが、今回のシステムの、標準的使用法では、40個以内の変数の、指定された時刻における値を、ラインプリンタで印刷することが可能で、また、その内の12個までの値を XY プロッタでグラフ表示することができる。プリント例とプロット例を図1、2に示す。今回のシステムはオンライン・シミュレーションという事は考えていないので、計算経過を実時間で監視することはできない。そのかわり、カードを計算センタに提出しておけば、順番が来ればセンタのオペレータが全部処理してくれて結果が出れば電話がかかって来るという便利さがあり、数多くのパラメータの組み合わせについて大量の計算をする場合に適している。

5. 使用法 (その1 ブロックの表現)

まえがきにも述べたとおり、本システムの構造は、

多数のサブルーチンの集まり、という形式をとっている。FORTRAN のサブルーチンには

{ CALL で呼び出すもの
関数の形で式の中に書けるもの

の2種類があるが(詳細は FORTRAN マニュアル参照)本システムにおいてブロックの表現に使用されているのはすべて後者すなわち「関数型」のサブルーチンである*。そして、その関数の「関数名」が、ブロックの種類を表わし、また、その引数の部分に、入力変数名、パラメータ、および初期条件等を記入するようになっていく。

基本的な使用法の例として、積分器の使い方について少し詳しく説明すると

SEKIBN (F, C)

という関数は、変数 F の(時間変数 t に関する)積分

$$\int F(t) dt$$

を表わし、C はその積分常数、云いかえれば、出力側変数の初期値を表わす。厳密に書けば、出力は

$$\int_0^t F(t) dt + C$$

である。

FORTRAN の文法によれば、式の書き方について次の規則がある。

(1) 出力変数名は左辺に、その計算式は右辺に書かなければいけない。

正しい例 Y=A+B

Y=SEKIBN (F, C)

誤りの例 A+B=Y

SEKIBN (F, C) = Y

(2) 関数は、計算式の中の、どこに書いてもよい。

正しい例 Y=A+SEKIBN (F, C)

Y=SEKIBN (F, C)

+SEKIBN (G, D)

(3) 関数の引数としては、変数、常数、または任意の計算式を書くことができる。

正しい例 Y=SEKIBN (F, 3.6)

Y=SEKIBN (F+G, C)

(4) 上記(2)と(3)を組み合わせた場合として、関数の引数の中に関数を書いてよい。

正しい例 Y=SEKIBN (SEKIBN (F, C), D)

(5) 変数名としては、アルファベットの大文字と数

* ÖUTSW のみは例外

```

***** SSDDS *****

***** TRAJECTORY *****

SIMULATION DATA

END          TIME = 32.00000
TIME        INTERVAL = 1.00000
OUTPUT TIME INTERVAL = 1.00000
PLOT (OPEN OR CLOSE) = OPEN      WIDTH (X) = 10.00000  HEIGHT (Y) = 10.00000 (INCH)
WRITE PLOT LABEL =SSDDS 3

HEAD LABEL

LABEL 1 =Z          LABEL 2 =V
LABEL 3 =MASS       LABEL 4 =X
LABEL 5 =GAMMA      LABEL

INITIAL VALUES SET
COUNT= 3
INITIAL VALUE 1= 13.00000  INITIAL VALUE 2= 3.00000
INITIAL VALUE 3= 27.69000  INITIAL VALUE 4= 1.32645
INITIAL VALUE 5= 62.70200  INITIAL VALUE 6= 240.00000
INITIAL VALUE 7= 9.80000   INITIAL VALUE 8= 229.00000
INITIAL VALUE 9= 2.01061   INITIAL VALUE 10= 0.80000
INITIAL VALUE 11= 0.12499  INITIAL VALUE 12= 0.00000
INITIAL VALUE 13= 0.00000  INITIAL VALUE 14= 0.00000
INITIAL VALUE 15= 0.00000  INITIAL VALUE 16= 0.00000

```

図 1 (a) 計算結果のプリント例(計算例3, 軌道計算)

```

***** SSDDS *****

***** TRAJECTORY *****

END TIME = 32.00000, TIME INTERVAL = 1.00000, OUTPUT TIME INTERVAL = 1.00000, PLOT = OPEN

TIME      Z          V          MASS      X          GAMMA
0.00000  0.13000E+02  0.27690E+02  0.62702E+02  0.30000E+01  0.76000E+02
1.00000  0.52831E+02  0.55158E+02  0.61654E+02  0.14143E+02  0.73052E+02
2.00000  0.11860E+03  0.83381E+02  0.60606E+02  0.35672E+02  0.70855E+02
3.00000  0.21046E+03  0.11235E+03  0.59558E+02  0.69254E+02  0.69077E+02
4.00000  0.32859E+03  0.14207E+03  0.58510E+02  0.11628E+03  0.67572E+02
5.00000  0.47323E+03  0.17256E+03  0.57462E+02  0.17799E+03  0.66264E+02
6.00000  0.64465E+03  0.20384E+03  0.56414E+02  0.25552E+03  0.65105E+02
7.00000  0.84316E+03  0.23592E+03  0.55366E+02  0.34991E+03  0.64063E+02
8.00000  0.10691E+04  0.26884E+03  0.54318E+02  0.46218E+03  0.63118E+02
9.00000  0.13229E+04  0.30260E+03  0.53270E+02  0.59330E+03  0.62253E+02
10.00000 0.16049E+04  0.33724E+03  0.52222E+02  0.74421E+03  0.61457E+02
11.00000 0.19155E+04  0.37280E+03  0.51174E+02  0.91586E+03  0.60719E+02
12.00000 0.22554E+04  0.40929E+03  0.50126E+02  0.11092E+04  0.60032E+02
13.00000 0.26249E+04  0.44675E+03  0.49078E+02  0.13250E+04  0.59391E+02
14.00000 0.30246E+04  0.48523E+03  0.48030E+02  0.15644E+04  0.58790E+02
15.00000 0.34551E+04  0.52475E+03  0.46981E+02  0.18282E+04  0.58225E+02
16.00000 0.39170E+04  0.56536E+03  0.45933E+02  0.21173E+04  0.57693E+02
17.00000 0.44110E+04  0.60711E+03  0.44885E+02  0.24328E+04  0.57191E+02
18.00000 0.49478E+04  0.65003E+03  0.43837E+02  0.27755E+04  0.56715E+02
19.00000 0.54991E+04  0.69420E+03  0.42789E+02  0.31466E+04  0.56265E+02
20.00000 0.60927E+04  0.73965E+03  0.41741E+02  0.35469E+04  0.55837E+02
21.00000 0.67225E+04  0.78646E+03  0.40693E+02  0.39776E+04  0.55431E+02
22.00000 0.73882E+04  0.83468E+03  0.39645E+02  0.44397E+04  0.55044E+02
23.00000 0.80910E+04  0.88438E+03  0.38597E+02  0.49344E+04  0.54676E+02
24.00000 0.88317E+04  0.93565E+03  0.37549E+02  0.54628E+04  0.54325E+02
25.00000 0.96115E+04  0.98856E+03  0.36501E+02  0.60261E+04  0.53990E+02
26.00000 0.10431E+05  0.10432E+04  0.35453E+02  0.66256E+04  0.53670E+02
27.00000 0.11293E+05  0.10997E+04  0.34405E+02  0.72626E+04  0.53364E+02
28.00000 0.12197E+05  0.11581E+04  0.33357E+02  0.79385E+04  0.53072E+02
29.00000 0.13145E+05  0.12186E+04  0.32309E+02  0.86547E+04  0.52792E+02
30.00000 0.14138E+05  0.12813E+04  0.31261E+02  0.94128E+04  0.52525E+02
31.00000 0.15179E+05  0.13462E+04  0.30213E+02  0.10214E+05  0.52269E+02
32.00000 0.16268E+05  0.14137E+04  0.29165E+02  0.11061E+05  0.52023E+02

```

図 1 (b) 計算結果のプリント例(計算例3, 軌道計算)

```

***** SSDDS *****

***** TRAJECTORY *****

LABEL          MINIMUM          (TIME)          MAXIMUM          (TIME)
Z              0.13000E+02      0.00000          0.16268E+05     32.00000
V              0.27690E+02      0.00000          0.14137E+04     32.00000
MASS           0.29165E+02      32.00000          0.62702E+02     0.00000
X              0.30000E+01      0.00000          0.11061E+05     32.00000
GAMMA         0.52023E+02      32.00000          0.76000E+02     0.00000

PLOT LABEL
Z
V
MASS
X
GAMMA
    
```

図 1 (c) 計算結果のプリント例(計算例 3, 軌道計算)

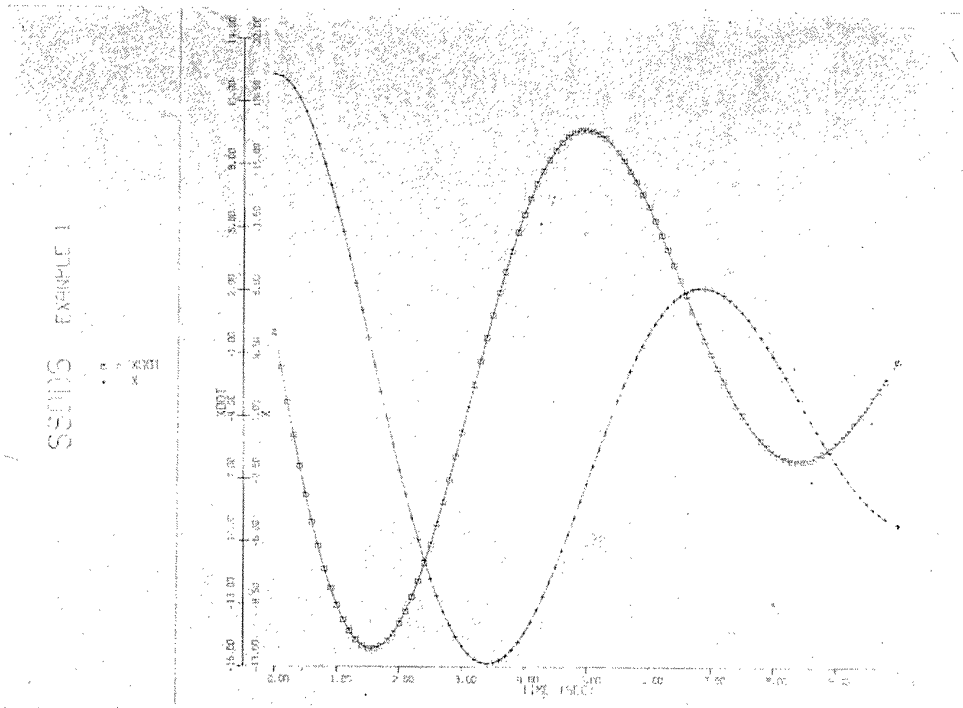


図 2 (a) プロット例 (計算例 2 の結果)

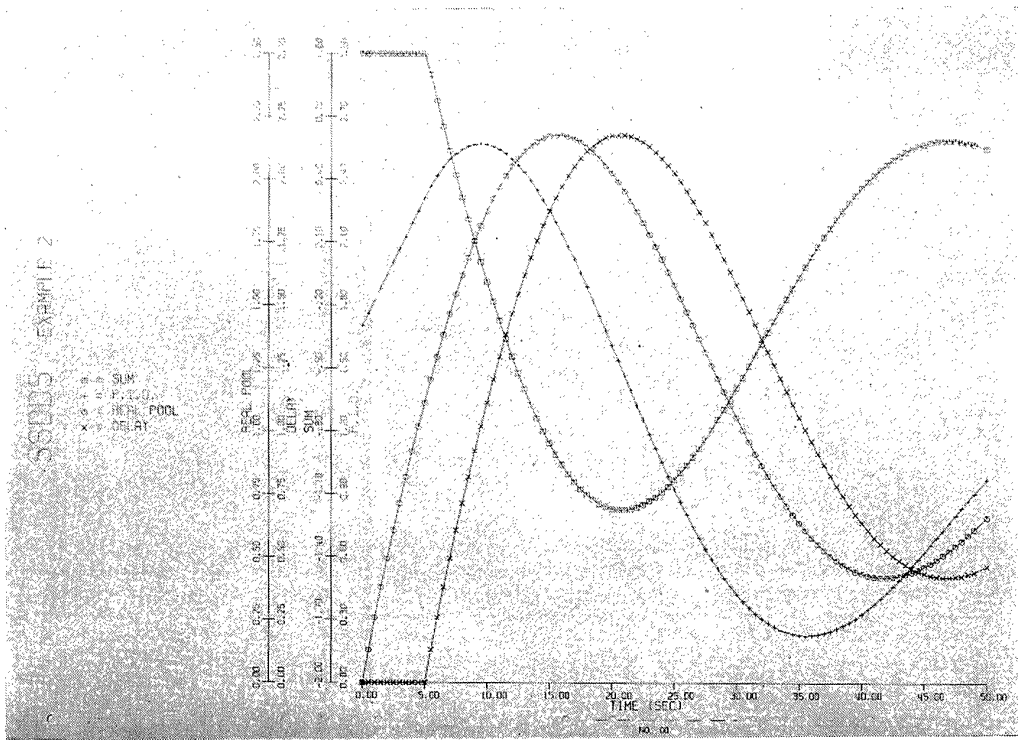


図 2 (b) プロット例 (計算例 2 の結果)

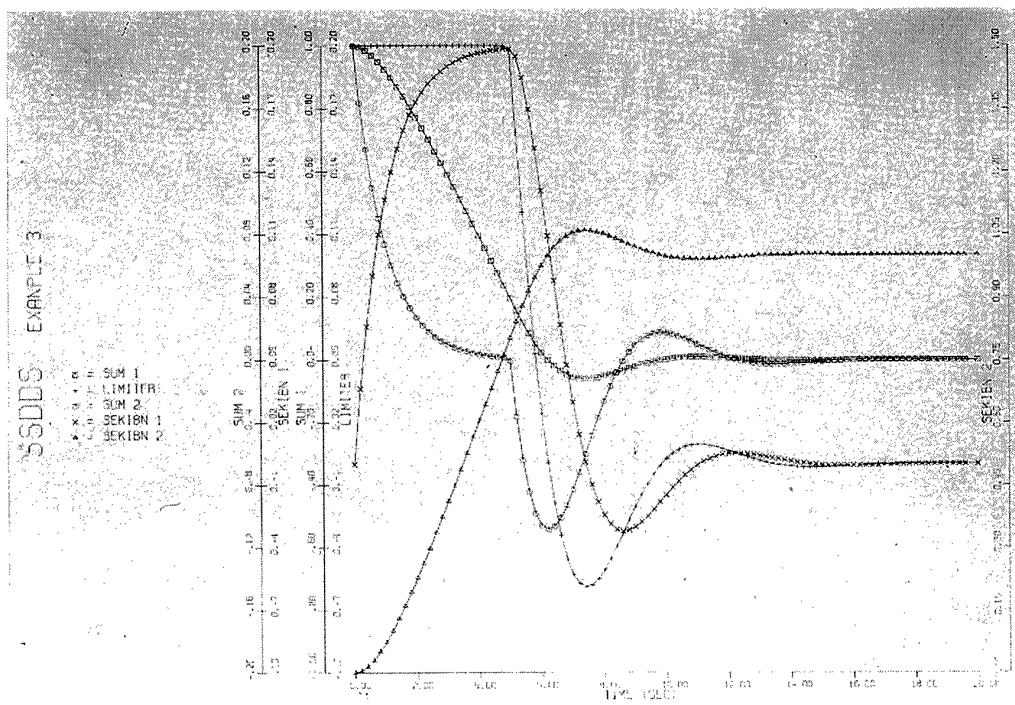


図 2 (c) プロット例 (計算例 2 の結果)

字の、6字までの任意の組み合わせを用いてよい。ただし、先頭の文字は A~H または O~Z でなければいけない（逆に言えば、第1字目には数字または I~N の英字を使ってはいけない）。

正しい例 X
ABC
OUTPUT
YUATSU
F104J

誤りの例 K (I~N)
PRSSURE (6字以上)
5020F (頭が数字)

- (6) 乗算記号には×を用いず、*印を用いる。
除算記号には÷を用いず、/印を用いる。
かっこは () だけが許され何重に使ってもよい。
- (7) 数字で書く常数には必ず小数点を付ける。

正しい例 1.0+A
誤りの例 1+A

本システムにおいては、ブロックの表現として書かれた式は、すべて FORTRAN のステートメントとして扱われ、FORTRAN コンパイラによってそのまま処理される仕組みになっているので、ブロック図の表現に当たっては、上記の文法規則に忠実に従う必要がある。なお詳細に関しては FORTRAN マニュアルを参照されたい。

この文法規則に従って書かれた応用例の二三のものを、図3に示す。(a)が積分器の基本型、(b)は入力

が2個ある例、(c)は乗算器の付いた例、(d)は二重積分の例、(e)はフィードバックのかかった系の一例である。

次に、パラメータを含むブロックの一例として、一次遅れ要素について簡単に説明すると、

$$\text{REALPL (F, YO, C)}$$

という関数は、

$$F \rightarrow \left[\frac{CS+1}{1} \right] \rightarrow Y \text{ (初期値 } Y_0)$$

なるブロックを表わす。C は回路定数のようなものであるから多くの場合はコンスタントであろうが、FORTRAN の文法的に見れば F と平等な引数であるから、必要ならばここに式を書いてもよいし、その値が時間的に変化してもよい。したがって、たとえば、adaptive control system のシミュレーションなどに用いることが可能である。

使用できるブロックの種類と、その引数の使い方を表3に示す。ここにあげたものは、既に完成して十分にテストされたものだけであるが、このほかにも一応作るだけ作ってみたものもあり、また今後とも、逐次充実させて行きたいと考えている。

備考) 右辺の形式に関して、DSL-90 および CSMP においては、FORTRAN よりもかなり強い制限を付けているようであるが、本システムにおいては FORTRAN 文法そのままよくはるかに自由度が大きい。

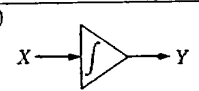
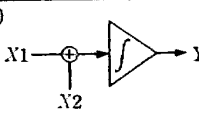
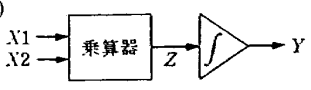
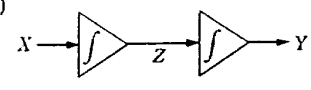
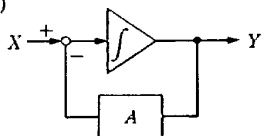
ブロック図	対応する式	別の書き方
(a) 	$Y = \text{SEKIBN} (X, Y_0)$	
(b) 	$Y = \text{SEKIBN} (X1 + X2, Y_0)$	$\text{SUM} = X1 + X2$ $Y = \text{SEKIBN} (\text{SUM}, Y_0)$
(c) 	$Y = \text{SEKIBN} (X1 * X2, Y_0)$	$Z = X1 * X2$ $Y = \text{SEKIBN} (Z, Y_0)$
(d) 	$Y = \text{SEKIBN} (\text{SEKIBN} (X, Z_0), Y_0)$	$Z = \text{SEKIBN} (X, Z_0)$ $Y = \text{SEKIBN} (Z, Y_0)$
(e) 	$Y = \text{SEKIBN} (X - A * Y, Y_0)$	$\text{FIDBAC} = -AY$ $\text{YINPUT} = X + \text{FIDBAC}$ $Y = \text{SEKIBN} (\text{YINPUT}, Y_0)$

図3 積分器の応用例

6. 使用法 (その2 計算全体の制御)

本システムを用いてシュミレーションを行なう場合の、実行時のプログラム全体の構成は図4のようになっている。ここで特に注意すべき点は、FORTRANの文法上の意味における「形式的なメインプログラム」は、図にSSDDSと記されている、既製の、固定されたプログラムであって、ブロック図の表現として利用者が作成する「実質的なメインプログラム」は形式的にはサブルーチンの形をとっており、

```
SUBROUTINE SSDDS1
```

というステートメントで始まり

```
RETURN
```

```
END
```

というステートメントで終る、変則的な構成になっていることである。

このような形式をとる理由は、常微分方程式のサブルーチンを使った経験のある人には容易に了解されると思うが、簡単に説明すると、Runge-Kutta-Gill法による数値積分のプログラムの本体がSSDDSの部分であって*、通常「補助ルーチン」(auxiliary routine)と呼ばれている部分——これが微分方程式の式の形を表わす——がSSDDS1に相当するわけである。ただ、普通の補助ルーチンと異なる点は、積分値や微係数を「補助ルーチンの引数」として表面に出すかわりに、「SEKIBN」というサブルーチンによって受け渡しを行っていることである。

利用者が作成すべき部分は、このSSDDS1という

部分だけである。ここには、前節の要領で、ブロック図を式で表現したものを並べて書くのであるが、それ以外に、

- 積分の範囲 (t の初期値, 終了値)
- 積分の間隔 (数値積分のステップ幅)
- 印刷すべき変数の名前, 順序, 時間間隔
- グラフの記録様式の指定 (サイズ, 見出し)
- 計算打切条件

などを指定しなければならない。その記入要領は表6のとおりで、いずれも形式上は、CALLにより、サブルーチン呼び出す、という形で書き、その引数として、それぞれの指定値を記入する。このようにすると計算時間の上では少しむだな時間を食われるが、使い易さを第一にするために、あえて、この形式を採用した。

書く順序は、

```
SUBROUTINE SSDDS1
COMMON/COUNT/KOUNT/TIMEER/T
CALL TITLE
CALL START
CALL OPEN
CALL INITAL
CALL LABEL
ブロック図を表わすステートメント群
CALL PRINT1
CALL PLOT2
```

* 正確に云えば図4のようにSSDDSの直接支配下にあるTIMEという所が積分を実行する。

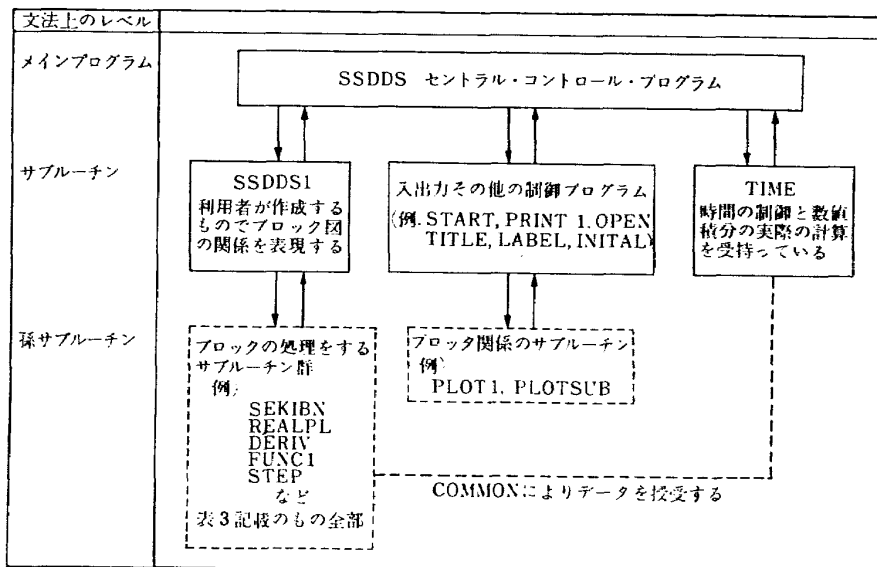


図4 プログラムの構成

表 6 計算の制御に関する指定の方法

目的	書き方	引数の意味	使用法	呼出制限回数	備考	備
計算条件の指定	START (T, D, N)	T : 計算打切時間, $t=0$ から計算を始め, $t=T$ で計算終了にする D : 積分のキザミ幅 Δt N : プリント間隔, 積分 N スラップ進む毎に 1 回, プリントする。		1	必ずつけなければいけない	
印刷すべき変数の指定	PRINT1 (A1, A2, A3, A4, A5, A6, A7, A8)	A1~A8 で指定された変数の値が, この順にプリントされる。 8 個より多くの変数をプリントしたい時は, 5 回までコールできる。k 回目指定された変数の値は第 k 行目にプリントされる。		5	少なくとも 1 回は呼出しておかないと, せっかく計算しても結果が見られない。	
タイトルの指定	TITLE (MŌJI)	MŌJI の部分に, タイトルを, 10H × × × × × × × × × × の形で書いておくと, 計算結果の最初に, このタイトルが印字される。(計算例参照)		1	省略してもよい	
計算結果の見出しの指定	LABEL (I1, I2, I3, I4, I5, I6, I7, I8)	先に PRINT1 で指定した A1~A8 に対応する見出しを, I1~I8 の所に, 10H × × × × × × × × × × の形で書くと, 各ページの最初に計算結果と共に印字される。10H × × × × × × × × × × ならプロットサプレス, 10H 空白 10 個なら印字サプレス。		5	PRINT 1 と同一個数だけ必要	
「プロットを使う」という宣言	OPEN (XINCH, YINCH)	XINCH, YINCH の所に, X 軸および Y 軸の仕上り寸法 (単位はインチ) を書く。但し Y 軸は 10 インチ以下。最初の PRINT 1 で指定された 8 個の変数のグラフィックが, 自動的にスクーリングされて描かれる。		1	プロットしたくなく入れなければよい	
計算中断の指定	FINISH (P, Q, I)	P > Q になったら, 打切予定時間以前でも計算終了とする。I の所は, 4H STOP と書いておけば全ケースの計算終了を意味し, 1H 空白ならば「次のケースに移れ」ということを意味する。		1	省略してもよい	
パラメータの読み込み	INITAL (M, S1, S1, S3, S4, S5, S6, S7, S8)	計算上のパラメータをカードから読み込むために用いる。M は変更回数, S1~S8 はパラメータの変数名で, カード上には, これと同じ順序で, 8F 10.5 の形式でパンチしておく。		原則として 1	パラメータをカードから読み込む場合にのみ使用する。	
時間変数以外の変数を X 軸にとりて描く	PLOT2 (X, Y, I1, I2, I3, I4, I5)	X, Y は仕上り寸法 (OPEN の XINCH, YINCH と同じ) I1 は X 軸にとる変数の番号, I2~I5 は Y 軸の方にとる変数の番号 (PRINT 1 の記載順 1, 2, ……を番号とする)。		1	必要な時だけ入れる	

CALL FINISH
RETURN
END

が標準である。

これらのプログラムは、上記のように、並べて書くだけでよく、特にそれを使う時にだけ呼び出す工夫をしなくても、自動的に、動作すべき時にだけ働くように作ってある。たとえば TITLE, START, INITAL, LABEL, OPEN, PLOT 2 などは計算の開始時点にだけ必要な処理を行ない、その他の時には何もしないで、もとのプログラムに戻る。また PRINT 1 も、最初に START で指定された印刷時点に来た時だけ作動する。計算の初期値は、特に指定しなくても印刷されるから、PRINT 1 を後の方に入れても心配する必要はない。FINISH の判定条件で中途打ち切りになった場合でも最終値はプリントされるから、これも利用者側でめんどろなプログラムを書く必要はない。

なお、本システムにおいては、ソーティング⁵⁾を行っていないので、ブロック図を表わす式は下記の順序に並べておくことが望ましい(その方が精度的に、好ましい結果が得られる、ということが知れている)。

- (1) 関数発生要素
- (2) 積分および遅れ要素
- (3) その他の演算要素

これでプログラムは完成。あとは、これをコーディング・シートに書き、パンチ受付に提出し、SSDDS システムのデッキを付けて計算機に掛けるだけである。FORTRAN を全く知らない人にとっては、以上の説明では、いささか不十分かとも思われるが、「FORTRAN を全く知らず、しかも本プログラムを使用しなければならぬ」という人はごく小数と推定されるので、詳細は著者または計算センタ(プログラム相談室)に照会されたい。

7. 使用法(その3 高等なテクニック)

[グラフに出す方法]

OPEN というサブルーチンと呼ぶと、ラインプリンタに印刷される変数(すなわち、PRINT 1 で指定したもの)が、グラフの方にも出力される。呼び出し方は

CALL OPEN (XINCH, YINCH)

で、XINCH YINCH はサイズの指定であり、それぞれ、X軸(時間軸)およびY軸の長さ(単位はインチ)を書く。X軸の方はロール紙なので、いくらでも長く記録できるが、Y軸の方は10インチ以下が望ましい(最

大で11インチであるが11インチにすると座標の目盛の数値が書けなくなってしまう)。プロットされる変数は自動的にスケールされて、このサイズにあわせてプロットされる。変数がいくつあっても1枚のグラフにまとめて描かれる。この際、変数名と曲線との対応は、打点シンボルによって区別することになっている。シンボルと変数の対応は表7のとおりである。

表7 打点シンボルと変数名の対応

PRINT1の引数として書かれた順序	グラフ上に打点されるシンボル
1	□
2	+
3	⊙
4	×
5	△
6	*
7	◇
8	☆

シンボルと変数名の対応はグラフの左端に表示される(例参照)。以上のようにすると、時間軸 t をX軸にしたグラフしか書けないが、従属変数間の関係を見るため(たとえば phase plane のカーブ $y-\dot{y}$ 曲線)には、PLOT 2 を用いる。これを用いると、PRINT 1 の最初の8箇に限らず、第40番目までの任意の変数をプロットできる。なお I1=4 HTIME としておくと t が X軸になる。

[パラメータの読み込み]

回路定数、計算のパラメータ、積分開始時の初期条件などを、データ・カードを用いて読み込ませることができる。

これには INITAL というサブルーチンを用いる。書き方は、

CALL INITAL (L, A, B, C, D, E, F, G, H) で、8個のパラメータA~Hを読み込むことができる。この際、特に注意すべき点は、「読み込むべき変数の個数がたとえ8個より少ない場合でも、一応表向きには8個として書かなければいけない」という事で、もしもそのような場合に困ったら、中で使っていない変数名

(たとえば Q) でうめておく。例：

```
CALL INITAL(L, A, B, C, Q, Q, Q, Q, Q)
```

逆に、読み込みたい変数が 8 個よりも多い場合には、二つ目以後の CALL INITAL においては、L の部分に 3HETC と書いておく。例：

```
CALL INITAL(L, A, B, C, D, E, F, G, H)
```

```
CALL INITAL(3HETC, X, Y, Z, Q, Q, Q, Q, Q)
```

この場合、データカードは 2 枚 1 組となり、最初のカードが A~H, 2 番目のカードが X~Z を表わす。

この 3HETC 以外の場合の L の意味は次項で説明する。

[2 ケース以上続けて計算する場合]

(i) 前の式と全く異なる計算式を処理する場合

FINISH の第 3 引数 (表 6 参照) を 1H にしておくと、1 ケースの計算が終了したのち、制御プログラムの中の KOUNT という変数が 1 だけ足されて、計算再開の状態になる。これを SSDDS1 の中で

```
COMMŌN/CŌUNT/KŌUNT
```

と書いて取り出して、computed GO TO で分岐させるか、あるいは IF ステートメントで判定すれば、全く異なるブロック図の計算を、いくつでも連続的に処理する事ができる。もちろん、最後のケースだけは FINISH の中に 4HSTOP と書かなければいけない。

(ii) 計算式(ブロック図)は前のケースと同一で、係数あるいは初期値だけを変えて計算する場合

FINISH の第 3 引数は 4HSTOP にして、INITAL の第 1 引数(表 6 の M)の所に、何ケース計算するかを書き込む。こうすると、1 ケースの計算が終了するごとに自動的にパラメータ読み込みルーチンが作動して係数、初期値をセットし、計算状態をすべてイニシャ

ライズして次の計算を開始する。

(iii) 上記 i, ii の組み合わせ

あるブロック図について何ケースかの計算をしたのち、別のブロック図についてまた何ケースか計算する場合には、FINISH の第 3 引数を 1H にして、先に i の項で述べたように KOUNT を用いて、INITAL の第 1 引数をコントロールする。

FINISH と INITAL の関係は図 5 のようになっている。

(iv) 計算続行中に一部分のパラメータを変更する。

FUNCSW(表 3) などを用いるのが原則であるが、これで物足りない場合は次項を参照。

[FORTRAN ステートメントの混用]

本システムは最初から FORTRAN と組み合わせて使用することを目標にして製作してあるので、FORTRAN のプログラムをまぜて使う事は差支えない。ただし下記の点だけは十分に注意されたい。

表 3 記載のブロックのうち、個数制限の付いているブロックは、常に一定の順序で使用しなければいけない。もちろん飛び越してもいけない。

(悪い書き方の例 その 1)

```
IF(A.GT.B)GŌ TŌ 10
```

```
X=SEKIBN(Y,Z)
```

```
U=SEKIBN(V,W)
```

```
GŌ TŌ 20
```

```
10 U=SEKIBN(V,W)
```

```
X=SEKIBN(YY,ZZ)
```

```
20 CŌNTINUE
```

(悪い書き方の例 その 2)

```
IF(A.GT.B) X=SEKIBN(F,G)
```

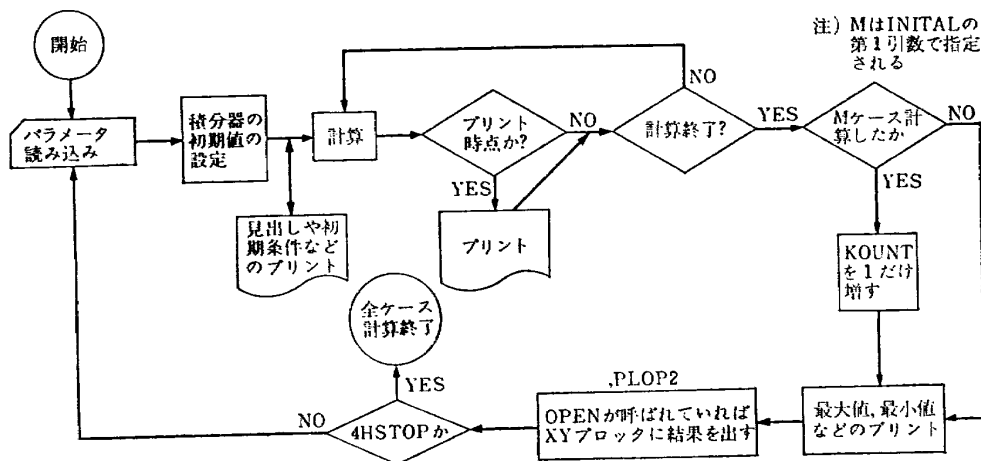


図 5 何ケースも続けて計算するための制御部分のフローチャート

このように、主として注意を要するのは IF を用いた場合の処置だけである。

厳密に云えば、順序一定という条件は、各種類ごとに、その同じブロックをいくつも使う時の順序が一定であればよい。たとえば SEKIBN と DERIV の順序がコースによって逆になっても差支えない。

順序一定という制限は、「積分器などに番号を付けない」という所から生じたものである。番号を付けると書くのがめんどうだから、そのかわりに順序一定にすることにした。しかし微分するにも積分するにも、「今の値」のほかに「昔の値」が必要である。「今の値」は引数に入っているが、「昔の値」の方はブロック処理プログラム本体の方で記憶しており、その今の値と昔の値の対応をつける必要がある。これを、本システムでは呼び出された順序に従って処理しているわけである。

もう一つの注意は、入出力ステートメントを自分で書く場合、ルンゲ・クッタ法の積分公式においては、積分を1スラップ進める間に、補助ルーチン(SSDDS1)を4回使用するという事で、それを忘れると積分やりかけの所でプリントしたりパラメータを変更したりする結果になってしまう。必ず4回単位のカウントをして、しかるべき時に入出力しないとイケない。

[時間変数を使う場合]

先の14ページの記載順序のとおり、冒頭に CÔMMON/CÔUNT/KOUNT/TIMEER/T と宣言しておけば、変数Tは時間を表わす。KÔUNT を用いない場合には

CÔMMON/TIMEER/T

としてもよい。

[記憶装置容量のゆとり]

各サブルーチンの長さは表8のとおりで、全部を使

表 8 サブルーチンに占有される語数

サブルーチン名	語 数	サブルーチン名	語 数
SSDDS	6 5	RELAY	3 9
START	1 7 6	OUTSW	4 2
FINISH	9 2	COMPAR	3 7
TITLE	3 4	RSTFF	3 3 2
OPEN	5 6	FUNC 1	3 1 1
PLOT 2	8 9	FUNC 2	3 2 2
LABEL	9 9	GLIMIT	5 9
INITAL	1 5 4	QNTZR	5 4
TIME	2, 2 3 4	HYSTSS	3 4 3
PRINT 1	1, 3 7 7	DEADSP	6 3
PLOT 1	3, 1 7 9	STEP	3 7
SEKIBN	1 0 9	RAMP	3 7
SEKIMD	1 5 9	GIMPLS	2 9 1
DERIV	2 2 9	PULSE	2 8 1
REALPL	5 3 4	SINE	4 9
SLDLAG	3 6 1	UNNÔIS	2 3 9
CMPXPL	1, 5 9 9	GRNÔIS	3 2 7
DELAY	1 0, 2 8 1	コモンセグメント	2 1 6
ZRHOLD	2 0 3	PLOTSUB	約 5, 0 0 0
FUNCSW	5 7		
標準組込関新名	語 数	これを必要とするサブルーチン名	
EXP	2 0 7	CMPXPL	
ALOG	1 7 4	GRNOIS	
SIN	8 0	GRNOIS	
COS	8 2	GRNOIS, CMPXPL	
L&SIN	2 2	GRNOIS, CMPXPL	
SQRT	5 0	GRNOIS	

用するとすれば約 25000ワードになるが、使用しないサブルーチンをあらかじめ除いて計算機にかければ、もっと少ない語数で済ませることもできる。記憶装置の占有される容量は、そのブロックを使うか使わないかで決まることであって、何回使うかということには関係しない。たとえば積分器は 1 台使っても 10 台使っても 109ワードである。以上のほかにシステムプログラム（ジョブモニタ、IOCS など）が 10000ワードぐらいを使用するので、全部のブロックを使う場合は 32K 語の機械にはかからないが、DELAYを除いてやれば 32K語で十分である。

8. 計 算 例

[例 1 簡単な制御系の過渡特性]

Hurley and Skiles の論文²⁵⁾に、簡単ではあるが実際の例題が示されていて、入力データと計算結果が出ていたので、同じ問題を本システムで解いてみた。発電機を含む系のシミュレーションで、ブロック図およびパラメータは図 6 のとおりである。これを本システムで解くために作成したプログラム (SSDDS 1) の部分は図 7 のとおりで、原論文のプログラム例は特定の 1 ケースだけを解くように書いてあるが、図 7 では一応、すべてのパラメータをデータカードで読み込めるようにしてある。計算結果を図 8 に示す。原論文の結果とも、よく一致している。

積分間隔 0.005秒として 7秒までの現象を計算するに要した時間は 27秒であった（原論文では同じ条件下で 40秒かかったと報告されている。ただし原論文の使用計算機は CDC-1604 である）。

[例 2 異なるブロック図の計算を連続して処理した例]

17ページで述べた KÖUNT の応用例として、下記の 3 種類の問題を連続して処理するプログラム例を図 9 に示す。

i) ダンパーを含む系(図10(i))

ii) むだ時間のある系(図10(ii))

iii) リミッタを含む系(図10(iii))

また、結果のプロット例を図 2 に示す。図 9 のプログラムおよび図 2 のプロット中の、EXAMPLE 1, EXAMPLE 2, EXAMPLE 3 が、上記 i, ii, iii に対応する。

[例 3 軌道計算への応用例]

ロケットの軌道計算への最も簡単な応用例として、質点計算で無制御の場合の計算例（プログラム図 11、計算結果は図 1）を示す。運動方程式は

$$\dot{x} = V \cos \gamma$$

$$\dot{z} = V \sin \gamma$$

$$\dot{V} = \frac{(T-D)g_0}{m} - g_0 \sin \gamma$$

$$\dot{\gamma} = - \frac{g_0 \cos \gamma}{V}$$

$$\dot{m} = - T/I_{sp}$$

ただし

$$D = \frac{1}{2} C_D \rho V^2 S$$

$$\rho = \frac{1}{8} (1 - 0.000022569 z)^{4.253} \quad \text{kg} \cdot \text{s}^2 / \text{m}^4$$

簡単のため T, C_D, I_{sp} は常数とした。記号の意味などは著者の一人が先に発表した報告（航技研資料 TM-109）を参照されたい。同報告の例題と同じ問題について、初段だけの計算を、発射角 $76^\circ, 80^\circ, 84^\circ$ について計算させてみた（結果に若干の差があるのは、ランチャーの長さが違っているためである）。このように、本プログラムは自動制御系の解析ばかりでなく、常微分方程式の問題一般にも適用可能で、非常にプログラミングが容易であり、実用価値がある。

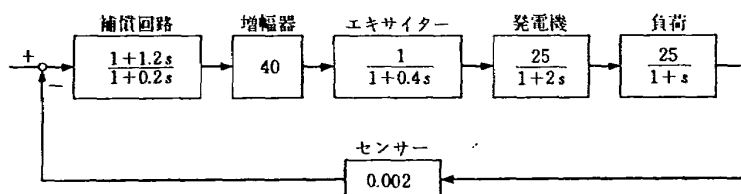


図 6 計算例 1 のブロック図

HARP 5020 COMPILED LIST		SSDDS1	12/20/43
EXTERNAL FORMULA NUMBER	SOURCE STATEMENT	INTERNAL FORMULA NUMBER	
	SUBROUTINE SSDDS1	SSDDS000	1
	CALL TITLE (10HREGULATOR)	SSDDS010	2
	CALL START (10,,0,01,2)	SSDDS020	3
	CALL OPEN (10,,10,)	SSDDS030	4
	CALL INITAL (1,AA,BB,CC,DD,EE,FF,GG,HH)	SSDDS040	5
	CALL INITAL (3HETC,OO,PP,QQ,RR,ZERO,ZERO,ZERO,ZERO)	SSDDS050	6
	CALL LABEL (10HERROR VOLT,10HCOMP.OUT.V,10HAMP.VOLTS ,10HEXCIT.VOL	SSDDS060	
	T,10HGEN.VOLTS ,10HLOAD AMPS ,10H)	SSDDS070	7
	A1=GG+HH*SEKI4	SSDDS080	8
	A2=OO+A1+PP*SEKI1	SSDDS090	9
	SEKI1=SEKI8N (A2,ZERO)	SSDDS100	10
	A3=QQ+A2+SEKI1	SSDDS110	11
	A4=RR+A3	SSDDS120	12
	B1=A4*AA+SEKI2*BB	SSDDS130	13
	SEKI2=SEKI8N (B1,ZERO)	SSDDS140	14
	B2=SEKI2*CC+SEKI3*DD	SSDDS150	15
	SEKI3=SEKI8N (B2,ZERO)	SSDDS160	16
	B3=SEKI3*EE+SEKI4*FF	SSDDS170	17
	SEKI4=SEKI8N (B3,ZERO)	SSDDS180	18
	CALL PRINT1 (A1,A3,A4,SEKI2,SEKI3,SEKI4,ZERO,ZERO)	SSDDS190	19
	CALL FINISH (SEKI4,1.E+20,4HSTOP)	SSDDS200	20
	RETURN	SSDDS210	21
	END	SSDDS220	22

図 7 計算例1のプログラム例

***** SSDDS *****

***** REGULATOR *****

END TIME = 10.96000, TIME INTERVAL = 0.01000, OUTPUT TIME INTERVAL = 0.02000, PLOT = OPEN

TIME	ERROR VOLT	COMP.OUT.V	AMP.VOLTS	EXCIT.VOLT	GEN.VOLTS	LOAD AMPS
0.00000	0.20403E+02	0.12240E+04	0.48960E+05	0.00000E-40	0.00000E-40	0.00000E-40
0.02000	0.20129E+02	0.44096E+03	0.17622E+05	0.15145E+04	0.29234E+03	0.10093E+03
0.04000	0.19527E+02	0.14762E+03	0.59048E+04	0.19912E+04	0.76072E+03	0.40344E+03
0.06000	0.18451E+02	0.21412E+02	0.85648E+03	0.20553E+04	0.12651E+04	0.94373E+03
0.08000	0.16916E+02	-0.42956E+02	-0.17182E+04	0.19366E+04	0.17463E+04	0.17150E+04
0.10000	0.14971E+02	-0.82757E+02	-0.33103E+04	0.17234E+04	0.21759E+04	0.26917E+04
0.12000	0.12687E+02	-0.11102E+03	-0.44407E+04	0.14537E+04	0.25381E+04	0.38388E+04
0.14000	0.10142E+02	-0.13214E+03	-0.52854E+04	0.11479E+04	0.28230E+04	0.51163E+04
0.16000	0.74231E+01	-0.14750E+03	-0.59000E+04	0.82010E+03	0.30249E+04	0.64813E+04
0.18000	0.46163E+01	-0.15752E+03	-0.63007E+04	0.48212E+03	0.31413E+04	0.78901E+04
0.20000	0.18071E+01	-0.16235E+03	-0.64939E+04	0.14480E+03	0.31724E+04	0.92997E+04
0.22000	-0.92112E+00	-0.16215E+03	-0.64861E+04	-0.18185E+03	0.31208E+04	0.10669E+05
0.24000	-0.34922E+01	-0.15718E+03	-0.62873E+04	-0.48863E+03	0.29916E+04	0.11959E+05
0.26000	-0.58376E+01	-0.14780E+03	-0.59120E+04	-0.76739E+03	0.27921E+04	0.13135E+05
0.28000	-0.78981E+01	-0.13447E+03	-0.53790E+04	-0.10111E+04	0.25309E+04	0.14168E+05
0.30000	-0.96257E+01	-0.11777E+03	-0.47107E+04	-0.12143E+04	0.22186E+04	0.15035E+05
0.32000	-0.10984E+02	-0.98314E+02	-0.39326E+04	-0.13725E+04	0.18663E+04	0.15716E+05
0.34000	-0.11949E+02	-0.76806E+02	-0.30722E+04	-0.14833E+04	0.14863E+04	0.16199E+05
0.36000	-0.12510E+02	-0.53961E+02	-0.21584E+04	-0.15453E+04	0.10908E+04	0.16440E+05
0.38000	-0.12668E+02	-0.30507E+02	-0.12203E+04	-0.15588E+04	0.69213E+03	0.16558E+05
0.40000	-0.12435E+02	-0.71596E+01	-0.28639E+03	-0.15259E+04	0.30207E+03	0.16440E+05
0.42000	-0.11834E+02	0.15403E+02	0.61611E+03	-0.14489E+04	-0.68295E+02	0.16138E+05
0.44000	-0.10901E+02	0.36552E+02	0.14621E+04	-0.13319E+04	-0.40895E+03	0.15669E+05
0.46000	-0.96754E+01	0.55728E+02	0.22291E+04	-0.11807E+04	-0.71118E+03	0.15054E+05
0.48000	-0.82068E+01	0.72454E+02	0.28982E+04	-0.10010E+04	-0.96778E+03	0.14316E+05
0.50000	-0.65487E+01	0.86345E+02	0.34538E+04	-0.79919E+03	-0.11732E+04	0.13484E+05
0.52000	-0.47577E+01	0.97117E+02	0.38847E+04	-0.58220E+03	-0.13236E+04	0.12585E+05
0.54000	-0.28923E+01	0.10459E+03	0.41835E+04	-0.35702E+03	-0.14170E+04	0.11648E+05
0.56000	-0.10102E+01	0.10868E+03	0.43472E+04	-0.13062E+03	-0.14531E+04	0.10704E+05
0.58000	0.83257E+00	0.10942E+03	0.43768E+04	0.90311E+02	-0.14333E+04	0.97792E+04
0.60000	0.25836E+01	0.10693E+03	0.42773E+04	0.29953E+03	-0.13605E+04	0.89006E+04
0.62000	0.41956E+01	0.10143E+03	0.40570E+04	0.49141E+03	-0.12393E+04	0.80929E+04
0.64000	0.56272E+01	0.93195E+02	0.37278E+04	0.66107E+03	-0.10752E+04	0.73739E+04
0.66000	0.68443E+01	0.82597E+02	0.33039E+04	0.80449E+03	-0.87496E+03	0.67636E+04
0.68000	0.78203E+01	0.70047E+02	0.28019E+04	0.91860E+03	-0.64596E+03	0.62743E+04
0.70000	0.85370E+01	0.55995E+02	0.22398E+04	0.10013E+04	-0.39616E+03	0.59152E+04
0.72000	0.89847E+01	0.40919E+02	0.16368E+04	0.10516E+04	-0.13379E+03	0.56911E+04
0.74000	0.91617E+01	0.25309E+02	0.10122E+04	0.10693E+04	0.13291E+03	0.56028E+04
0.76000	0.90746E+01	0.96347E+01	0.38539E+03	0.10553E+04	0.39594E+03	0.56471E+04
0.78000	0.87375E+01	-0.56307E+01	-0.22523E+03	0.10115E+04	0.64774E+03	0.58168E+04
0.80000	0.81715E+01	-0.20061E+02	-0.80243E+03	0.94038E+03	0.88140E+03	0.61018E+04
0.82000	0.74005E+01	-0.33268E+02	-0.13307E+04	0.84521E+03	0.10909E+04	0.64885E+04
0.84000	0.64591E+01	-0.44918E+02	-0.17967E+04	0.72983E+03	0.12710E+04	0.69614E+04
0.86000	0.53809E+01	-0.54736E+02	-0.21894E+04	0.59850E+03	0.14179E+04	0.75027E+04
0.88000	0.42035E+01	-0.62514E+02	-0.25005E+04	0.45574E+03	0.15285E+04	0.80937E+04
0.90000	0.29659E+01	-0.68111E+02	-0.27244E+04	0.30625E+03	0.16013E+04	0.87150E+04
0.92000	0.17067E+01	-0.71457E+02	-0.28583E+04	0.15470E+03	0.16357E+04	0.93469E+04
0.94000	0.46395E+00	-0.72552E+02	-0.29021E+04	0.56146E+01	0.16324E+04	0.99705E+04
0.96000	0.72673E+00	-0.71460E+02	-0.28584E+04	-0.13673E+03	0.15931E+04	0.10568E+05
0.98000	-0.18326E+01	-0.68307E+02	-0.27323E+04	-0.26846E+03	0.15205E+04	0.11123E+05
1.00000	-0.28249E+01	-0.63275E+02	-0.25310E+04	-0.38616E+03	0.14381E+04	0.11620E+05

図 8 計算例1の計算結果の一部

```

HARP 5020 COMPILED LIST          SSDDS1                      12/20/43

EXTERNAL FORMULA NUMBER - SOURCE STATEMENT - INTERNAL FORMULA NUMBER

SURROUTINE SSDDS1                      SSDDS000  1
COMMON /COUNT/<COUNT              SSDDS010
DATA ZERO/0.0/,TEN/10.0/,ONE/1.00/   SSDDS020
GO TO (1,2,3),<COUNT              SSDDS030
C ***** EXAMPLE 1                      ***** SSDDS040
1 CALL TITLE (10-EXAMPLE 1 )          SSDDS050  3
  CALL LABEL (10-1XDOT ,10HX ,10H ,10H ,10H ) SSDDS060
  1 ,10H ,10H ,10H ,10H )          SSDDS070  4
  CALL OPEN (TEN,TEN)                SSDDS080  5
  CALL INITIAL (1,AA,BB,DELT,SPIN,ZERO,ZERO,ZERO,ZERO) SSDDS090  6
  CALL START (TEN,DELT,10)           SSDDS100  7
  SINT=SINE (ZERO,ONE,ZERO)         SSDDS110  8
  XDOT=SEKIBN ((SINT-AA*XDOT-BB*X)/TEN,ZERO) SSDDS120  9
  X =SEKIBN (XDOT,SPIN)             SSDDS130 10
  CALL PRINT1 (XDOT,X,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO) SSDDS140 11
  RETURN                              SSDDS150 12
C ***** EXAMPLE 2                      ***** SSDDS160
2 CALL TITLE (10-EXAMPLE 2 )          SSDDS170 13
  CALL LABEL (10-SUM ,10HP,I.D. ,10HREAL POOL ,10HDELAY ) SSDDS180
  1 ,10H ,10H ,10H ,10H )          SSDDS190 14
  CALL OPEN (TEN,TEN)                SSDDS200 15
  CALL START (50.0,0.125,2)          SSDDS210 16
  SUM1=ONE-DELY                      SSDDS220 17
  PID=1.7*(1.+SEKIBN (SUM1,ZERO)/1.4) SSDDS230 18
  POOL=REALPL (PID,ZERO,TEN)         SSDDS240 19
  DELY=DELAY(POOL,5.)               SSDDS250 20
  CALL PRINT1 (SUM1,PID,POOL,DELY,ZERO,ZERO,ZERO,ZERO) SSDDS260 21
  CALL FINISH (PID,1.E+20,1H )      SSDDS270 22
  RETURN                              SSDDS280 23
C ***** EXAMPLE 3                      ***** SSDDS290
3 CALL TITLE (10-EXAMPLE 3 )          SSDDS300 24
  CALL LABEL (10-SUM 1 ,10HLIMITER ,10HSUM 2 ,10HSEKIBN 1 ) SSDDS310
  1 ,10HSEKIBN 2 ,10H ,10H ,10H )   SSDDS320 25
  CALL OPEN (TEN,TEN)                SSDDS330 26
  CALL START (20.,0.01,10)          SSDDS340 27
  SUM1=ONE-SEKI2                     SSDDS350 28
  GLIM=GLIMIT (SUM1,-0.2,0.2)       SSDDS360 29
  SUM2=GLIM-SEKI1                    SSDDS370 30
  SEKI1=SEKIBN (SUM2,ZERO)          SSDDS380 31
  SEKI2=SEKIBN (SEKI1,ZERO)         SSDDS390 32
  CALL PRINT1 (SUM1,GLIM,SUM2,SEKI1,SEKI2,ZERO,ZERO,ZERO) SSDDS400 33
  CALL FINISH (SEKI2,1.E+20,4HSTOP) SSDDS410 34
  RETURN                              SSDDS420 35
END                                  SSDDS430 36
    
```

図 9 計算例2のプログラム

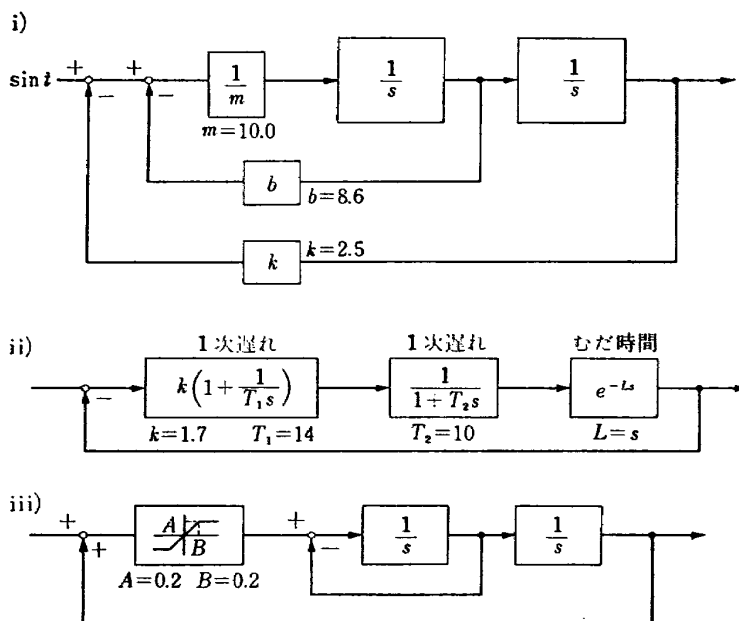


図 10 計算例2のブロック図

EXTERNAL FORMULA NUMBER	SOURCE STATEMENT	INTERNAL FORMULA NUMBER
	HARP 5020 COMPILED LIST SSDDS1 12/21/68	
	SUBROUTINE SSDDS1	SSDDS000 1
	REAL MASS,ISP	SSDDS010
	CALL TITLE (10HTRAJECTORY)	SSDDS020 2
	CALL LABEL (10HZ ,1CHV ,1HMASS ,10HX ,10HGAMMA ,10H ,10H ,10H)	SSDDS030
	CALL OPEN (10.,10.)	SSDDS040 3
	CALL INITIAL (3,Z,XI,V,GAMMA,MASS,TI,GC,ISP)	SSDDS050 4
	CALL INITIAL (3HETC,S,CD,RAG,ZERO,ZERO,ZERC,ZERO,ZERO)	SSDDS060 5
	CALL START (32.,1.,1)	SSDDS070 6
	RHO=(1.-0.000022569*Z)**4.253/8000.	SSDDS080 7
	D=0.5*S*CD*RHO*V**2	SSDDS090 8
	X=SEKIBN ((V*COS(GAMMA)),XI)	SSDDS100 9
	Z=SEKIBN ((V*SIN(GAMMA)),Z)	SSDDS110 10
	V=SEKIBN (((TI-D)*GO/MASS-GO*SIN(GAMMA)),V)	SSDDS120 11
	MASS=SEKIBN ((-TI/ISP),MASS)	SSDDS130 12
	GAMMA=SEKIBN (((-GO*COS(GAMMA))/V),GAMMA)	SSDDS140 13
	GAMMM=GAMMA*RAG	SSDDS150 14
	CALL PRINT1 (Z,V,MASS,X,GAMMM,ZERO,ZERO,ZERO)	SSDDS170 16
	CALL FINISH (ABS(Z),1.E+20,4HSTOP)	SSDDS180 17
	RETURN	SSDDS200 18
	END	SSDDS210 19

図 11 計算例3のプログラム例

9. あとがき

本システムの開発にあたり、先に石川島播磨重工で開発された同種のシステム“BOSS”を参考にさせていただいた。このシステムを御提供下さった、同社制御部長唐沢康人氏をはじめ関係者の方々に心からお礼申上げる。また「良きユーザ」として、数々の有益な助言をされた当所データ処理研究室の鈴木崇弘技官に深くお礼申上げる。

参 照 文 献

- 1) 渋谷政昭；各種の分布をもつ乱数の生成，第3回プログラミング・シンポジウム・報告集(1962)
- 2) 高田 勝，田中秀幸；制御解析のプログラム，第8回プログラミング・シンポジウム報告集(1967)
- 3) 高田 勝，大規説平，工藤和彦；対話形式のデジタル・アナログ・シミュレータ，第9回プログラミング・シンポジウム報告集(1968)
- 4) 的場 進；ブロック線図のデジタル・シミュレーションに関するプログラム作成の一方法，情報処理，第9巻第1号(1968)
- 5) 三巻達夫；ダイナミック・システムのデジタル・シミュレーション，計測と制御，第7巻第4号(1968)
- 6) 田中 明；システム工学部会報告，計測と制御，第7巻第11号(1968)
- 7) 三巻達夫；電子計算機による自動制御系のシミュレーション，計測と制御，第5巻第8号(1966)
- 8) 的場 進；アナログむき入力言語によるシミュレータ，(MDAS)，電気通信学会誌，第49巻第7号(1966)
- 9) 田村康男；ブロックダイアグラムシミュレータと自動プログラム方式の結合，電気学会雑誌，第80巻862号(1960)
- 10) 結城 信，桑田竜一；計算機制御のためのデジタルシミュレータ，第6回自動制御連合講演会前刷(1963)
- 11) 外山守城，松浦卓丈；ブロック線図のデジタル・シミュレーション・プログラム・システム，第6回自動制御連合講演会前刷(1963)
- 12) 西原 宏；Monte Carlo 法，制御工学，第7巻第3号(1963)
- 13) 森村英典；シミュレーション3，数理科学，第2巻第5号(1964)
- 14) 平井平八郎，石桁正士；モンテカルロ法とその応用，制御工学，第9巻第4号(1965)
- 15) 浦 昭二，高橋磐郎，共編；公式と数値計算手順集，(経営数学E-1) みすず書房(1961)
- 16) 大須賀節雄，山内平行；図形入出力を用いたオンライン汎用ブロック解析システム，情報処理学会第8回大会講演予稿集(1967)
- 17) 日本アイビーエム；MIDAS，IBM-APA，No. DC-46
- 18) 日本アイビーエム；DSL/90，IBM-APA，No. DC-054
- 19) 日本アイビーエム；CSMP，IBM データセンター

- ・ユーザーズ・ガイド(1968)
- 20) 森口繁一; 連立常微分方式の解法, 情報処理, 第4巻第1号(1963)
 - 21) J. J. Clancy and M. S. Fineberg; Digital Simulation Languages : A Critique and a Guide, Proc. FJCC(1965)
 - 22) M. Palevsky and J. V. Howell; The DES-1 — A Real Time Digital Simulation Computer, Proc. FJCC(1963)
 - 23) W. M. Syn and R. N. Linebarger; DSL/90— A Digital Simulation Program for Continuous System Modeling, Proc. FJCC(1966)
 - 24) M. L. Stein, J. Rose and D. B. Parker; A Compiler with an Analog Oriented Input Language, Proc. WJCC(1959)
 - 25) J. R. Hurley and J. J. Skiles; DYSAC — A Digitally Simulated Analog Computer, Proc. SJCC(1963)
 - 26) G. F. Forbes; An Engineering Simulator, Proc. ACM National Meeting(1966)
 - 27) R. D. Brennan and H. Sano; PACTOLUS— A Digital Analog Simulator Program for the IBM 1620, Proc. FJCC(1964)
 - 28) H. E. Petersen, F. J. Sansom, R. T. Hartnett and L. M. Warshawsky; MIDAS — How it works and How it's worked, Proc. FJCC (1964)
 - 29) R. M. Janosky, R. L. Shaefer and J. J. Skiles; COBLOC—A Program for All-Digital Simulation of a Hybrid Computer, IEEE Trans. EC-15, No.1 (1966)
 - 30) B. J. Karafin; The New Block Diagram Compiler for Simulation of Sampled-Data Systems, Proc. FJCC(1965)
 - 31) R. A. Gaskill, J. W. Harris and A. L. McKnight; DAS — A Digital Analog Simulator, Proc. SJCC(1963)
 - 32) F. Lesh; Methods of Simulating a Differential Analyzer on a Digital Computer, JACM, vol. 5, No. 3(1958)
 - 33) R. G. Abraham and M. W. Collings; Digital Programs for Simulation, Control Eng. October 1966.
 - 34) 三巻達夫, 並木成之 ; DDS (Digital Dynamics Simulator) 第11回自動制御連合講演会前刷(1968)

TM-146 双発STOL機の飛行試験 (実用運用包囲線および離着陸特性)	1968年8月	幸尾治朗, 岡 遠一 塚野雄吉, 鎌田幸男 小野孝次
TM-147 浮動振子型加速度計の試作	1968年9月	大月正男, 鈴木孝雄 円居繁治
TM-148 吹出しフラップ翼の高揚力特性	1968年9月	小橋安次郎, 宮沢政文 西 武徳
TM-149 NAL-16-TR ロケットの研究試作	1968年10月	ロケット性能研究室
TM-150 リフトジェット VTOL 機の着陸にお ける水平方向の減速について	1968年11月	西村博史
TM-151 フラッタ解析法の再検討 (フラッタ解析法の研究 その1)	1968年12月	市川輝雄
TM-152 ノズル内の突起物によるロケットエン ジンの側分力	1969年1月	榎崎哲二, 中野富雄
TM-153 FA-200 改機の風洞試験 (I)プロペラ付き	1969年2月	広末健一, 坂井紀穂 高橋 宏, 綿貫忠晴
TM-154 FA-200 改機の風洞試験 (II)高揚力装置付き	1969年2月	広末健一, 木村友昭 高橋 宏, 綿貫忠晴
TM-155 極超音速風洞M7 ノズル較正試験	1969年2月	橋本登, 吉沢 昭 穂積弘一

注：欠番は配布先を限定したもの

航空宇宙技術研究所資料156号

昭和44年3月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話 武蔵野三鷹(0422)44-9171 (代表) ☎182

印刷所 一誠社総合印刷株式会社
東京都武蔵野市御殿山1-6-10
