

連続系シミュレーション言語

松坂 知行*

Continuous System Simulation Language

Tomoyuki MATSUZAKA

Abstract

Continuous system simulation languages such as CSPL or CSSL are implemented on large-scale computers, but not on small scale-ones. However, rapid development of micro computers or personal computers has enabled us to realize such languages on small computers. Hence, the author has developed a new simulation language DECS (Differential Equation based Continuous system Simulation program) which can be implemented on not only personal computers, but also large-scale machines. Thus, the users can select computers properly, depending on problem scales, and save computation cost. This paper describes the constitution and application of the language.

1. まえがき

近年、システムの分析・設計段階におけるシミュレーション技術の普及は目を見張るものがある。計算機のハードウェア、ソフトウェアの発展と相まって理・工学分野のみならず社会科学の分野にまで利用範囲が広がっている。これはシステムが大規模、複雑化していく中で設計段階での事前分析・評価が重要な作業となり、分析・評価に精密化が要求される一方、これに応える計算手段、計算技法が発達し、言わば需要と供給の相乗効果でシミュレーション技術の利用範囲が広がっているものと思われる¹⁾。

シミュレーションという言葉は、1940年代の後半にVon NeumannとUlamが確率的試行の問題をモンテカルロ法を用いて解いたことに起因していると言

昭和63年12月15日受理

* 電気工学科・情報システム工学研究所教授

われており、いまでもシミュレーションと言えばモンテカルロ法を指すほどである。モンテカルロ法は主として待ち行列などの確率現象の分析に用いられてきた。

これに対してシミュレーションという用語が適用されてきた分野に工学現象がある。工学現象は基本的には連立常微分方程式でシステムが表現され、デジタル計算機が普及する前は、その求解にアナログ計算機がもっぱら利用されてきた。アナログ計算機は原理的には、時間的、空間的な並列演算器であり、初期のデジタル計算機より遙かに高速な求解が可能であったが、非線形演算や複雑な論理判断、計算精度、スケーリングなどの問題があり、デジタル計算機の普及によりあまり利用されなくなってきた。

一方デジタル計算機はアナログ計算機のこのような欠点をカバーしているが、FORTRANなどのような汎用言語でプログラミングを行なうと、パッチボード上で積分器や加算器を物理的に接続するアナログ計算機のプログラミングに比較してプログラミングが非常に面倒である。そこでこのようなアナログ計算機の利点を生かし、かつ欠点をデジタル計算機でカバーするというハイブリット計算機が考案されたが、値段が高いこととソフトウェアが十分に整備されなかったため、普及されるに至らなかった。

そこで汎用デジタル計算機を用いてアナログ計算機なみの容易さでプログラミングが行なえないかどうかの検討がはじまり、専用シミュレーション言語が開発されるようになった。

2. 連続系シミュレーション言語²⁾

1955年Selfridgeがデジタル計算機によるアナログ計算のシミュレータ、すなわちデジタル・アナログシミュレータ(DAS)を開発して以来、種々の連続系シミュレータ(連続系シミュレーション言語の通称)が開発されてきた。それらをまとめると

(a) ブロック向き言語(block oriented)

DAS
MIDAS
PACTOLUS
CSMP/1130

(b) 方程式向き言語(equation oriented)

MIMIC
DSL/90
CSSL
CSMP/360

などである。

連続系シミュレーションで重要なポイントは2点である。一つは動特性記述部分の処理順番を決めるソーティング、もう一つは積分方式である。1964年に開発されたMIDASはこの二つも機能を備えた点で画期的なシミュレータであった。さらにその後MIDASの改良版PACTOLUSがIBMのR. D. Brennanにより開発された。アナログ計算機で行なわれるシミュレーションでは、動特性を目で追いながら、モデルの制御が実行できるが、PACTOLUSはこの点で連続系シミュレーション言語の労作である。

以上の言語はアナログ演算で用いられているブロック・モデリングを基にしているが、1960年代の後半から従来のアナログ的なブロック・モデリングの他にデジタルプログラミングの機能を附加することの重要性が認識され始め、1965年IBMのSyn, WynnによりDSL/90が開発され、この機能を備えたFORTRANベースの第一号のシミュレータであった。MIMIC、CSSL、CSMP/360は同様な考え方で開発されたものである。

以上述べた連続系シミュレーション言語は大型計算機をベースにして開発されたものであるが、1980年代に入りマンマシーン・インターフェースの優れた16ビットのパーソナルコンピュータが登場しはじめ、パーソナルコンピュータベースのMCSP³⁾⁻⁶⁾、ACSLなどの連続系シミュレーション言語が開発された。

筆者は今回DECS(Differential Equation based Continuous system Simulator)と呼ばれる新しい連続系シミュレーション言語を開発し、いくつかの改良をおこなったので報告する。

3. ソフトウェア構成⁷⁾⁻⁹⁾

DECSはFORTRANにより記述されている。本言語の開発に用いたFORTRANはPROFESSIONAL FORTRANである。F77完全準拠であればこれ以外のFORTRANでもよい。つぎにソフトウェアの構成について述べる。

3. 1 ソフトウェア構成

DECSのソースプログラムは、基本的に次の6つの文より成る。

a) 制御文

プログラムの流れを制御する文で、”>”で始まる。

b) 初期値設定文

状態変数の初期値を設定する文で、これを省略した時は初期値を零と見なす。

c) 定数設定文

システム記述部に出てくる変数の値を設定する文である。

d) 配列宣言文

システム記述部に出てくる配列を宣言する文である。

e) システム記述文

システムを表わす連立常微分方程式を記述する文である。

システム記述文の中には後述するシステムマクロ関数、FORTRAN命令文を記述することができる。

f) サブルーチン文

FORTRANのSUBROUTINE文を記述する文である。

つぎに制御文の説明を行なう。

3. 1. 1 制御文

(1) タイトル文 (>TITLE)

CRT画面の最上段やプリンタの最初の行にTITLEを表示する。

(2) パラメータ文 (>PARAM文)

各シミュレーション実行時のパラメータを設定する。

(3) 時間制御文 (>TIMER)

演算時間刻み、最終時間、スキップ間隔、横スクロール時間を制御する。横方向にスクロールが可能なため過度状態から定常状態まで長い現象を観察できる。

(4) スケール文 (>SCALE)

縦軸のフルスケール、目盛りの種類（均等、対数）を設定する。

(5) 出力制御文 (>PRINT)

出力形式，出力機器，出力すべき変数を指定する。出力機器としてはCRT画面、プリンタ、プロッタ、ディスクなどで、出力形式はグラフ、数値である。

(6) 入力制御文 (>INPUT)

ディスクファイルを読み出す制御文である。これによりあらかじめ作成していた実データ（例えばランダム信号，外部データなど）をシステムに取り込みながらシミュレーションが可能で、より現実的なシミュレーションが可能になる。

これらの制御文の具体的な記述方法は応用例を通して説明する。

3. 1. 2 初期値設定文

状態変数の初期値を設定する文で次の様式で記述される。

```
*INITCON
```

```
  X1=C1
```

```
  X2=C2
```

```

      .
      .
      .
      Xn=Cn
*END

```

ここで $X_1, X_2 \dots X_n$ は状態変数で $C_1, C_2 \dots C_n$ は初期値である。

3. 1. 3 定数設定文

システム記述文に使用される変数の値を記述する文である。

```

*CONSTANTS
      V1=t1
      V2=t2
      .
      .
      .
      Vn=tn
*END

```

ここで $V_1, V_2 \dots V_n$ は変数で $t_1, t_2 \dots t_n$ は定数である。

3. 1. 4 配列宣言文

システム記述文に使用される配列を記述する文である。

```

*ARRAY
      DIMENSION A1(10),A2(100)
      COMMON B1(10),C1/B2(100),B3(10)/
      .
      .
      .
*END

```

3. 1. 5 システム記述文

システムのダイナミックスを記述する文である。システムの挙動は一般に次のような二つの方程式によって記述できる。

$$\frac{dx}{dt} = f(x, u, t) \quad (1)$$

$$y = g(x, u, t) \quad (2)$$

ここで x, u, y はそれぞれ状態ベクトル、入力ベクトル、出力ベクトルである。上式が線形の場合は状態遷移行列を計算することにより状態ベクトル、出力ベクトルを求めることができるが、非線形の場合はこの方法は使えない。DECSでは線形、非線形いずれのシステムにも適用できるように四次 RUNGE-

KUTTA法を用いている。

システム記述文は

*DYNAMICS

$\text{DOT}(X_1)=F_1(X_1, X_2, X_3, \dots X_n, U, T)$

$\text{DOT}(X_2)=F_2(X_1, X_2, X_3, \dots X_n, U, T)$

•

•

•

$\text{DOT}(X_n)=F_n(X_1, X_2, X_3, \dots X_n, U, T)$

*END

のように記述される。

なおシステム記述文の中ではシステムマクロ関数とFORTRAN命令文を記述することができる。

DECSのもっている関数と記述形式を表1から表4に示す。これらの関数は、

a) 信号発生関数

RAMP ランプ関数

STEP ステップ関数

SINE 正弦波関数

RANF 正規乱数発生関数

b) 論理関数

NOT 否定

AND 論理積

OR 論理和

c) スイッチング関数

FCNSW 関数スイッチ

INSW 入力スイッチ

d) 非線形関数

LIMIT リミッタ

DEADSP 不感帯

HSTRSS ヒステリシス

FITTER 折れ線関数

などである。

3. 1. 6 サブルーチン文

サブルーチンを記述する文で、FORTRANのサブルーチン文と同じである。なおサブルーチン文ではシステム記述文で用いられているマクロ関数は使用できない。従来のシステムではサブルーチン文をシステム記述文の中に記述したが、マクロ関数を使用できないため、混同を避けるため独立のブロックと

した。

```
*SUBROUTINE
  SUBROUTINE SUB1(A1)
    DIMENSION A1(10)
    COMMON B1(10), C1/B2(100), B3(10)/
      .
      .
      .
    RETURN
  END
*END
```

4. ハードウェア

DECSを実行させるにはFORTRANの走るハードウェアならパーソナル・コンピュータでも大型機でもよい。現在までに実績のあるものはPC-9801（数値演算プロセッサ付き）、ワークステーションNEWS、大型計算機ACOSS2000である。

5. 翻訳と実行

3. のルールにしたがって書いたソースプログラムはDECSによりFORTRANプログラムに変換される。すなわちDECSは実行形式のFORTRANプログラムを生成するためのプリプロセッサとして働く。したがって全体の実行時間はFORTRANプログラムへの変換時間と生成されたFORTRANプログラムの実行時間との和になる。DECSの文法と生成されたFORTRANプログラムに誤りがなければ直ちに実行されるが、ソースプログラムに文法誤りがある時はソースプログラムが自動的に呼出されるので訂正後再度DECSを起動する。実行形式のFORTRANプログラム自体に誤りがある時はFORTRANコンパイラのエラーメッセージにより誤りを訂正する。

6. 応用例

以下具体的な応用例を通してソースプログラムの書き方、実行結果、方法、を述べる。

図1のようなパラメトリック励磁回路をシミュレーションしてみた。文献(7)よりこの回路の微分方程式は(3)式のように与えられる。

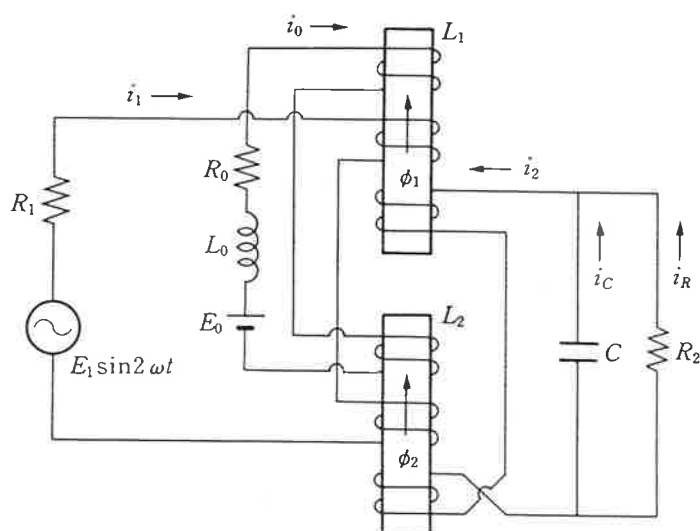


図1 パラメトリック回路

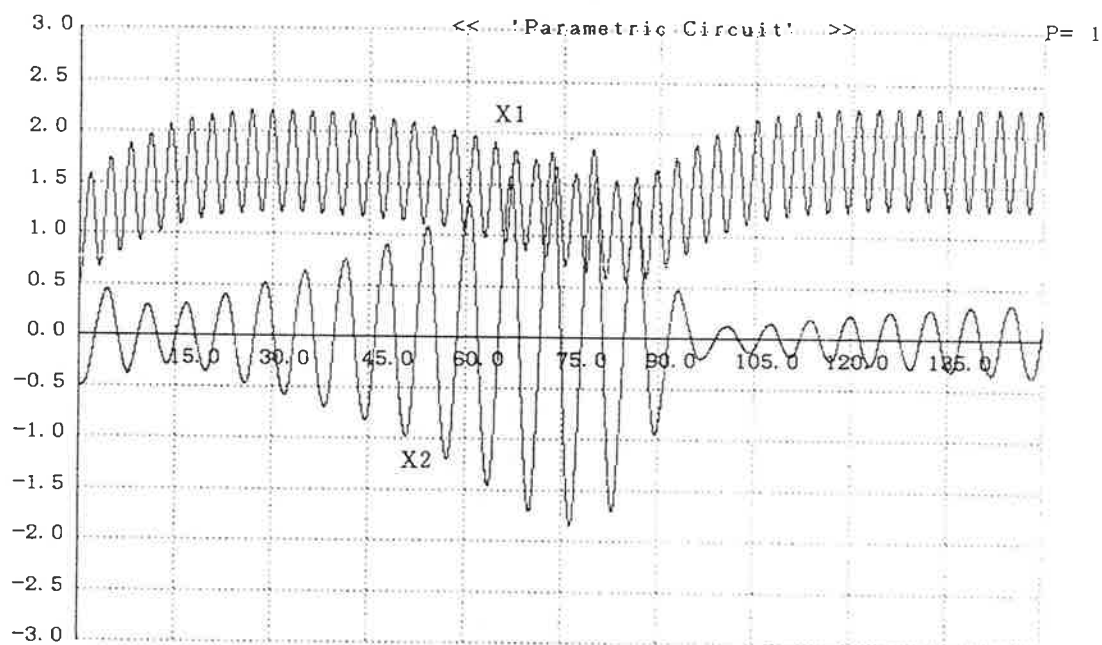


図2 パラメトリック回路の応答

$$\left. \begin{aligned} da/dt + k_1 \{ (a^2 + 3b^2)a - 8u_0 \} / 8 &= B \sin 2t \\ d^2b/dt^2 + k_2 db/dt + (3a^2 + b^2)b / 8 &= 0 \end{aligned} \right\} \quad (3)$$

ここで,

$$\left. \begin{aligned} x1 &= a \\ x2 &= b \\ x3 &= db/dt \end{aligned} \right\} \quad (4)$$

とにおいてソースプログラムを作成すると、

```

1: C      REM PARAMETRIC EXCITATION
2: C      >TITLE ['Parametric Circuit']
3:        >TIMER [150,.02]
4:        >PRINT [D:X1,X2]
5:        *CONSTANTS
6:          K1=.1
7:          K2=.2
8:          B=1
9:          U0=.8
10:       *END
11:       *INITCON
12:          X1=.5
13:          X2=-.5
14:       *END
15:       *DYNAMICS
16:          DOT(X1)=B*SIN(2*TIME)-K1*((X1**2+3*X2**2)*X1-8*U0)/8
17:          DOT(X2)=X3
18:          DOT(X3)=-K2*X3-(3*X1**2+X2**2)*X2/8
19:       *END

```

となる。リストの内容は下記の通りである。

- 2: タイトル文
- 3: タイマー文で、終値時間150、演算刻み0.02
- 4: 出力制御文でディスクにファイル名X1.0、X2.0で出力
- 5:～10: 定数設定文
- 11:～14: 初期値設定文
- 15:～19: システム記述文で、(3)の微分方程式を(4)の状態方程式に直

表 1 信号発生関数

関数	DECS記述	数式記述
RAMP	$Y = \text{RAMP}(X)$	$Y = 0, \quad t < X$ $Y = t - X, \quad t \geq X$
STEP	$Y = \text{STEP}(X)$	$Y = 0, \quad t < X$ $Y = 1, \quad t \geq X$
SINE	$Y = \text{SINE}(P_1, P_2, P_3)$	$Y = 0, \quad t < P_1$ $Y = \text{SIN}\{P_2(t - P_1) + P_3\}, \quad t \geq P_1$
RANF	$Y = \text{RANF}(P_1, P_2)$	P_1 : 平均値 P_2 : 標準偏差

表 2 論理関数

関数	DECS記述	数式記述
NOT	$Y = \text{NOT}(X)$	$Y = 1, \quad X \leq 0$ $Y = 0, \quad X > 0$
AND	$Y = \text{AND}(X_1, X_2)$	$Y = 1, X_1 > 0 \text{ and } X_2 > 0$ $Y = 0, : \text{上記以外の時}$
OR	$Y = \text{OR}(X_1, X_2)$	$Y = 0, X_1 \leq 0 \text{ and } X_2 \leq 0$ $Y = 1, : \text{上記以外の時}$

表 3 スイッチ関数

関数	DECS記述	数式記述
FCNSW	$Y = \text{FCNSW}(X_1, X_2, X_3, X_4)$	$Y = X_2, X_1 < 0$ $Y = X_3, X_1 = 0$ $Y = X_4, X_1 > 0$
INSW	$Y = \text{INSW}(X_1, X_2, X_3)$	$Y = X_2, X_1 < 0$ $Y = X_3, X_1 \geq 0$

表 4 非線形関数

関数	DECS記述	数式記述
LIMIT	$Y = \text{LIMIT}(P_1, P_2, X)$	$Y = X, P_1 \leq X \leq P_2$ $Y = P_1, X < P_1$ $Y = P_2, X > P_2$
DEADSP	$Y = \text{DEADSP}(P_1, P_2, X)$	$Y = 0, P_1 \leq X \leq P_2$ $Y = X - P_2, X > P_2$ $Y = X - P_1, X < P_1$
* HSTRSS	$Y = \text{HSTRSS}(I_C, P_1, P_2, X)$	$Y = X - P_2, X > X_{n-1}$ $Y_{n-1} \leq X - P_2$ $Y = X - P_1, X > X_{n-1}$ $Y_{n-1} \leq X - P_1$ $Y = Y_{n-1}$: 上記以外の時 Y_{n-1}, X_{n-1} : 1 つ前の値 I_C : Yの初期値
FITTER	$Y = \text{FITTER}(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X)$	$Y = Y_1, X < X_1$ $Y = Y_7, X > X_7$ それ以外 (X_n, Y_n) を 結ぶ折れ線関数

*HSTRSSはヒリテリシス関数

表 5 実行速度の比較

機種名	PC-9801VM2		PC9801XL ²			NEWS-MODEL821
CPU	V30		80386			68020
ビット数	16		32			32
クロック数	10 MHz		16 MHz			16.67 MHz
出力	フロッピー・ディスク	ラムディスク	フロッピー・ディスク	ハードディスク	ラムディスク	ハードディスク
速度	2分 48秒 20	1分 03秒 48	2分 0秒 30	37秒 04	21秒 29	30秒80

して記述

図2は実行結果を示し、文献(11)と一致していることが確かめられた。つぎに表5は実行速度の比較である。本表で見られるように16ビット機と32ビット機ではフロッピディスクに出力した場合は、余り差はないが、ラムディスクに出力した場合は約3倍速いことがわかる。またNEWSの速度はハードディスクで比較した32ビットパソコンより20%速い程度である。なお大型機ACOSS 2000で実行してみたところ、実行時間は数秒で終わるがファイル転送時間が10数分かかり、シミュレーションのように何度も実行を繰返す場合には実用的ではない。

なお本稿以外の例ではPWMインバータ駆動誘導電動機、MW級風力発電機、電気機械の過度現象、自動制御系の応答などにも適用してみたが、十分実用に耐えることがわかった。

7. 結論

以上のようにDECSを用いれば連立常微分方程式で記述される動的システムのシミュレーションを行なうプログラムを短時間で自動的に生成でき、時間領域でパラメータを変えながらシステムの時間応答を観測できる。またシステムの規模によりパソコンと大型機を使い分けることができるので計算費用の面からも有利である。DECSの学習は筆者の経験ではFORTRANを知っている学生ならきわめて短時間で可能である。

なお本言語は線形、非線形要素を含む10数例についてテストし考えられるバグを除いてある。

なお大規模システムへ適用する場合には高速化が必要になるが、最近パーソナルコンピュータ上で動作する並列演算ボード(トランスピュータ)やこのボード上で走る並列FORTRANが開発され、このボード単体でも10MIPSの速度が得られると報告されている。こうなるとトランスピュータを並列に接続することにより数10MIPSの速度が得られることが予想され、大型機に迫るシミュレーションエンジンとして十分利用できると思われる。

8. 参考文献

- 1) 石谷：シミュレーション技術の最近の動向、システムと制御、Vol. 29、No. 11、p697-705、1985
- 2) 中西：シミュレーション言語の新しい動向、システムと制御、Vol. 29、No. 11、p706-713、1985
- 3) 松坂：連続系シミュレーション言語の開発、インターフェース、Vol. 8、

p183-190、1982

- 4) 松坂：パーソナルコンピュータによる連続系シミュレーション言語、インフォメーション、Vol. 2、p49-53、1983
- 5) 松坂：パーソナルコンピュータによる連続系シミュレーション言語、シミュレーション、Vol. 5、No. 3、p47-55、1986
- 6) 松坂：パソコンによる動的システムとシミュレーション、工学図書、1986
- 7) T. Matuzaka and S. Ookawa: Microcomputer based continuous system simulation program、EPMESC' 85、3B、p459-471、Macau.
- 8) T. Matuzaka: Continuous system simulation program implemented on a personal computer、IFAC/IMACS International symposium on simulation of control systems、p461-466、Vienna、1986
- 9) 松坂：連続系シミュレーション言語による非線形系のシミュレーション、電子情報通信学会非線形問題研究会資料NLP87-17、1987年7月
- 10) 松坂：パソコンから大型機まで実行可能な連続系シミュレーション言語、SENAC、Vol. 21、No. 3、p76-88、1987
- 11) Chihiro Hayashi: Nonlinear Oscillations in Physical Systems、p277-282