



Title	並列処理方式会話型汎用シミュレータ"HOSS"
Author(s)	牧野, 圭二
Citation	北海道大學工學部研究報告, 129, 45-55
Issue Date	1986-01-31
Doc URL	http://hdl.handle.net/2115/41965
Type	bulletin (article)
File Information	129_45-56.pdf



[Instructions for use](#)

並列処理方式会話型汎用シミュレータ “HOSS”

牧野圭二

(昭和 60 年 9 月 30 日受理)

“HOSS”: An Interactive Multi-Purpose Simulator Using Parallel Processing Techniques

Keiji MAKINO

(Received September 30, 1985)

Abstract

This paper describes the design concept and the structure of HOSS intended to solve continuous and discrete system simulation and signal processing. HOSS is a decentralized hierarchical multiprocessor system including a parallel processing subsystem, PPA, for the simulation of continuous systems. PPA is composed of two master processors and a circular array of thirty-two slave processors connected by a shared memory and two broadcastable buses. It works as a conventional analog computer in solving differential equations. The user can interactively operate HOSS using simulation software, ICOSS (CSMP type) or RISS (GPSS type), for on-line real-time simulation.

1. ま え が き

北海道大学全学共同利用施設汎用シミュレータ施設では⁸⁾ 従来使用してきた大型ハイブリッド計算機の設備更新を機会に、実時間シミュレーションを行うことができる全デジタル式の会話型汎用シミュレータ、HOSS (Hokkaido University High-Speed System Simulator) を新たに開発し³⁾ 導入した。⁵⁾ HOSS では、実時間系のオンライン・シミュレータである ICOSS (Interactive Continuous-System Simulation System; 連続系用)⁴⁾ と RISS (Realtime Interactive System Simulator; 離散系用) とが開発され組み込まれると共に、信号解析や信号処理の問題も取り扱うことができ、モデル化技法やシミュレーション・システムの開発、並列シミュレータの研究なども可能なように考慮がはらわれている。

HOSS は、ハイブリッド計算機の設備更新ということもあって、その処理の中心は連続系のシミュレーションに置かれている。特に、北海道大学で早くから検討並びに提案を行ってきた並列処理方式の連続系シミュレータを基に¹⁾²⁾、実用機として改良した並列演算サブシステム、PPA (Parallel Processor Array)⁶⁾⁷⁾ と、タスクの分割、割付けを自動的に行うソフトウェア・システム (ICOSS) とを実用化している。

連続系シミュレーションの分野では、スケーリングやバッチングなどプログラミングの柔軟性に欠けるものの、実時間性能、問題との対応の良さ、実行時における操作性の良さなどから長い間アナログ（ハイブリッド）計算機が用いられてきた。しかし、ミニコンピュータの出現以来、デジタル計算機のプログラミングの柔軟性を生かし、最大の欠点である実時間性を解決するために、並列システムの研究がミニコン・コンプレックスを中心に進められてきた。しかし、多くの提案があったにもかかわらず、実用機として用いられるシステムは、処理機能単位での演算ユニットの専用化と並列化の方式（Applied Dynamics社のAD-10）、DDA方式（日立電子社のDS-1000）などであり、純粋な並列プロセッサ・システムとしてはHOSS（PPA及びICOSS）が初めてとあってよい。

HOSSは当初から一般研究者に対する実用機として開発が進められ、昭和55年に一応の完式をみた後、広く学内でオープン利用されてきている。

2. 設 計

2.1 問題点の解析

我々は、“汎用シミュレータ”というものを、“個々の利用者が自分の目的に合わせて専用化できる（汎用）、情報処理装置としての性格・機能を有する模擬実験装置（シミュレータ）”であると考えている。これが設計に対する基本拘束条件である（表1、表2）。

表1 利用者が求める利用形態

- ・ 完全なオープン利用
- ・ 実行者の実質的システム資源の専有使用
- ・ システム内部を意識せずにシステムの持つ機能が利用できること
- ・ 道具としての便利さ
- ・ 利用者自身の効率の良さ

表2 汎用シミュレータに求められる能力

- ・ 処理の経過を見ながら処理に介入できる機能
 - システムと人間（実行者）との高度なコミュニケーション能力（オンライン・シミュレーション）
- ・ 他実験装置、測定装置、センサ等との親和性
 - 各種アナログ情報信号、デジタル情報信号の入出力機能
- ・ 対象の規模に依存しない実時間性
 - 高速・高精度の演算処理能力（実時間シミュレーション）
- ・ 利用者の問題に合わせて専用のシミュレータとなしうる機能
 - プログラミング能力
- ・ プログラム、データの蓄積及び再利用できる機能

“シミュレーション”においては、実際の計算処理時間が短いことの必要性は言うまでもないが、単なる“数値計算”とは異なり、試行錯誤の容易さ、すなわち利用者の心理の流れに沿って処理の実行を制御できることにその特質がある。そのためには、シミュレーション実行中における解

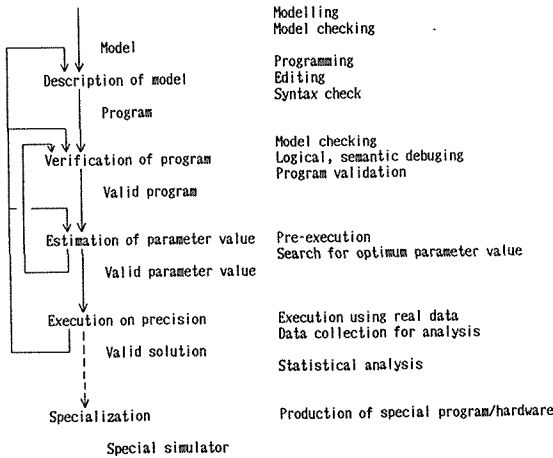


図 1 シミュレーション問題の標準処理パターン

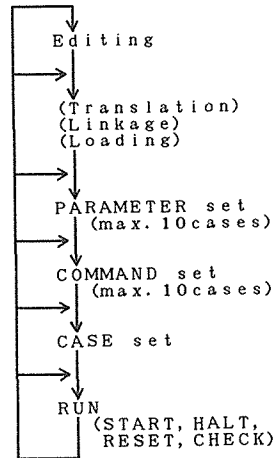


図 2 ICROSS における処理の流れ

の監視，異常検出，停止，修正，再試行等の一連の処理が利用者の意のままに繰り返し連続的に
行なえることが必要である。我々は，この試行錯誤の容易さを実現するために，シミュレーシ
ョン問題を解く過程の分析(図 1)から利用者の必理の流れを解析し，それに基づきシミュレーシ
ョンの実行過程を定めている(図 2)。

この必要性は，連続系のシミュレータとして長い間広く使われてきたアナログ計算機の例から
も明らかである(表 3，表 4)。主として，アナログ計算機の長所はその操作性と演算の高速性に
あり，問題点はプログラミングの柔軟性の欠如にある。一方，汎用デジタル計算機ではこの関
係は逆になり，従来から指摘されてきたアナログ計算機の問題点はデジタル計算機ではほとん
ど解決される。しかし，オンライン実時間シミュレーションの実現は，連続系シミュレーション
言語を用いたとしても難しい。

表 3 アナログ計算機の長所

- 実システムと計算機のシミュレーション・モデルとの対応の良さ
- パラメータの設定及び変更の容易さ及び随意性
- 観測出力点の変更及び追加の容易さ及び随意性
- システム・モデルの部分修正及び追加等の部分変更の容易さ及び随意性
- 問題の規模や複雑さに依存しない演算の高速性
- 演算の実時間性

表 4 アナログ計算機の問題点

- 演算精度
- 固定小数点演算方式によるスケーリング
- 演算の柔軟性
- 解き得る問題の規模の限界
- 大規模問題における解の再現性
- プログラムの蓄積
- 汎用性

その一つの解決方法が我々が目指している全デジタル式シミュレータであり、その成功の一つの基準は、アナログ計算機の持つ問題点の克服と、同時に、アナログ計算機の持つ長所の保存とがいかに実現できるかにある。

2.2 設計方針

以上の分析から以下のような設計方針を立てた。

(1) HOSS は、従来のハイブリッド計算機シミュレータでは困難であった大規模連続系のオンライン実時間シミュレーションを行うことを主目的にした多目的シミュレータとする。さらに、離散系シミュレーション、AD・DA 変換などを含む信号処理等も行うことができるような設計を行う。

(2) 本システムは不特定多数の一般研究者の共同利用に供される実用機であるため、ハードウェアに偏ることなくソフトウェアの完備にも重点を置き、両者が一体化した設計を行う。また、実用機として必要な製作並びに保守の容易さを確保する。

(3) HOSS のハードウェア構成は、機能分散型の階層構造を持つマルチプロセッサ・システムとする。特に、主目的である連続系に関しては、専用の並列演算サブシステムを開発する。その基本演算は 32 bit 浮動小数点演算としハードウェア演算機構を備える。

(4) ソフトウェアは、連続系、離散系共に、会話型オンライン実時間ソフトウェア・システムを作成する。そのモデル記述部分は、汎用性と一般性ことから、それぞれ、CSMP 系統、GPSS 系統とする。

(5) 試行錯誤時における処理間の遷移を容易にするため、シミュレーション問題の“標準処理パターン”に基き処理単位毎にモジュール化を行う。特に、シミュレーション条件（環境）の記述はプログラム本体から分離独立させ、プログラム本体を再翻訳することなくシミュレーション環境のみを単独に変更、再設定して実行できるようにする。

(6) システム利用者とのコミュニケーションを重視し、グラフィック・ディスプレイ装置を有効に使う。そのために、メニュー選択、ガイドライン表示、デフォルト値の各方式を併用する。また、利用者に戸惑いや不安感を与えないように、システムの処理中は処理内容を表示し、エラー表示はエラー内容だけでなく次にどのような動作が選択可能かを表示する。

(7) 連続系においては、特にシミュレーション・ソフトウェア・システムの一部として、モデルや計算法を含む演算実行方法の妥当性、解の安定性などについての評価のための補助手段を工夫考慮し組み込む。

3. HOSS のアーキテクチャ

HOSS は、図 3 に示すように、主計算機レベル、制御プロセッサ・レベル、処理レベルの 3 レベルの階層からなる階層構造を持ったマルチプロセッサ・システムであり、以下のように機能分散することにより各プロセッサの専用装置化を行っている。

主計算機レベルは DEC 社の VAX-11/780 システムで構成されている。本シミュレータ全体の監視と共に、シミュレーションの実行に必要な、ソース・プログラムの作成、編集、翻訳、タスク分割等の準備作業を行う。さらに、シミュレーション後のシミュレーション・データの解析等の処理を行う。

制御プロセッサ・レベルは 4 台のプロセッサからなり、主計算機レベルと処理レベルとの間にあって処理レベルの各サブシステムを制御し主計算機とのインタフェイスとなっている。すなわ

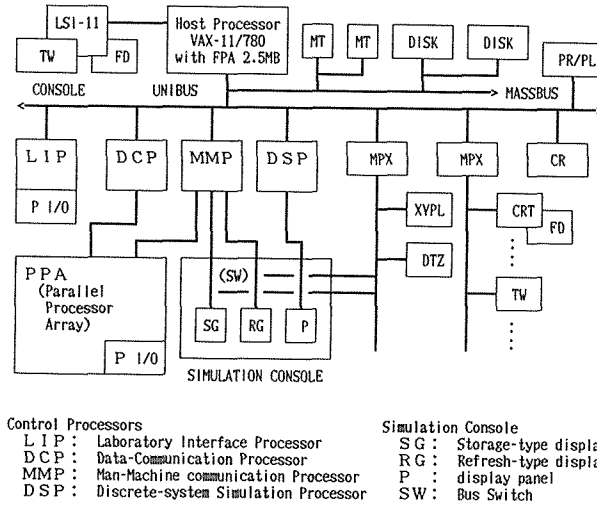


図 3 HOSS の全体構成

ち、ラボラトリ・インタフェイス・プロセッサ (LIP) は、AD・DA 変換器などを装備しており信号処理の専用である。転送制御プロセッサ (DCP) は、PPA への連続系シミュレーション・プログラムの転送と、シミュレーション実行時に PPA で発生される、シミュレーション後の処理で必要となるデータのディスク装置への転送を専門に行う。マン・マシン・コミュニケーション・プロセッサ (MMP) は、グラフィック・ディスプレイへの解の表示と共に、シミュレーション・コンソールからの入力を処理し、人間とシステムとの間の会話機能全般を制御する。離散系シミュレーション・プロセッサ (DSP) は、離散系シミュレーションの実行の制御を行う。現在は実行自体も行っている。

処理レベルは現在二つの部分からなる。一つは並列演算装置 PPA (Parallel Processor Array) であり、もう一つはシミュレーション・コンソールである。PPA はこれ自体階層構造を持つ 34 台のプロセッサからなる 1 次元循環アレイ型のマルチプロセッサ・システムで、連続系のシミュレーションを高速で実行する。シミュレーション・コンソールは、グラフィック・ディスプレイ装置 (SG, RG) を中心に、シミュレーション専用のファンクション・キーを装備したキー・ボードや表示パネル (P) などから構成されている。

HOSS を構成しているプロセッサは、主計算機は DEC 社の VAX-11/780 であり、それ以外のプロセッサは、MAP 16 と呼ばれる三井造船㈱製のマイクロプログラム演算制御方式を用いた標準的な 16 ビット・ミニコンピュータである。MAP 16 は、DEC 社の PDP-11/34 と同等の性能及び類似の機械語命令 (マクロ命令) 体系を有している。

この MAP 16 に対し、指数部 2 進 8 ビット、仮数部 24 ビットの単精度浮動小数点の乗除算が各約 5 μs で実行できるハードウェアによる浮動小数点高速演算機構 (FFP) を開発している。FFP では計算精度向上のための工夫を行うと共に、MAP 16 本体では、FFP 用の各種命令、32 ビット・データを単位として扱う命令、積分演算等のシミュレーション用命令などが特殊機械語命令としてマイクロプログラムにより追加されている。このため、各プロセッサは基本的には 16 ビット・マシンであるにもかかわらず、32 ビット浮動小数点演算を基本演算とする 32 ビット・マシンであるかのように処理を行うことができる。

4. PPA の構造と連続系シミュレーションへの適用

PPA では CSMP タイプの連続系シミュレーション言語の実行が想定されている。処理の並列化は方程式レベルで行い、各演算プロセッサには、オイラ法、2 次のアダムス予測子法、3 次のルンゲークッタ法の 3 種の数値積分法がマイクロプログラム化され、標準演算として組み込まれている。従って、単純には、各演算プロセッサを積分の入力となる代数演算も含めて計算を行う一種の数値積分器とみていることになり、基本的にアナログ計算機のデジタル化と考えられる。

4. 1 PPA の構造

並列演算装置 PPA はマスタースレーブ方式に構成されている(図 4)。各エレメント・プロセッサはシェアド・メモリの他に固有の内部メモリ (LM, ローカル・メモリー) を持っており、それぞれ独立なプログラムを実行でき、MIMD 型(Multiple Instruction and Multiple Data Stream) のシステムである。

PPA は、制御用の 2 台のマスタ・プロセッサ (MPa, MPb)、演算用の 32 台のスレーブ・プロセッサ (SLP 0~SLP 31)、及び、データ交換用の 34 個の 3 ポート・シェアド・メモリ・バンク (SMa, SMb, SM 0~SM 31) とから構成されている。32 台のスレーブ・プロセッサ (SLP) と 32 個のシェアド・メモリ・バンク (SMバンク) とが 1 次元循環アレイ状に交互に並び、このシェアド・メモリ・バンクを通して 2 台のマスタ・プロセッサ (MP) が全体を制御・統括する構造を持っている。また、2 個のシェアド・メモリ・バンク (SMa, SMb) が外部からのプログラムのロード、あるいは外部とのコマンドやデータの交換のために用いられる。また、32 台の SLP のうち 4 台には、P I/O (AD, DA 及び、デジタル入出ポート) が装備されており、通常は演算プロセッサとして働くが、実時間シミュレーションのときには入出力プロセッサとして働く。

PPA は MP に関して対称な構造であり、各 MP は一つおきの SM バンクをアクセスすることができ、これらの SM バンクを通して各 16 台の SLP を制御する。

各 SM バンクはそれぞれ 1 KB ずつの四つの領域、FIX 0, FIX 1, EXC 0, EXC 1 からなっている。FIX 0 と FIX 1 はアドレス固定領域であるが、特に FIX 0 はブロードキャスト可能領域とも呼ばれ、MP が FIX 0 に対応する仮想領域にデータを書き込むと、その MP の支配下にある各

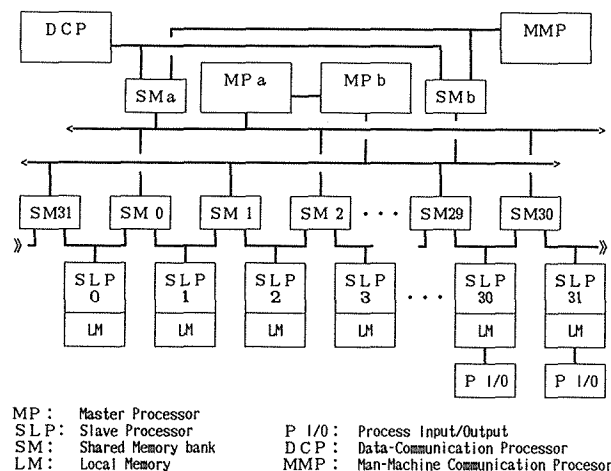


図 4 PPA の構成図

SMバンクのFIX 0領域のそれぞれの位置にブロードキャスト転送が行われる。なお、各MPは、各SMバンク毎のアドレスを直接指定して、各FIX 0領域を個別にアクセスすることができる。一方、EXC 0とEXC 1とはアドレス切替可能領域であり、MPがフラグを切替えると、それぞれの領域の物理アドレスが一斉に入れ換る。

さらに、このブロードキャスト転送時におけるMPによる仮想領域の先頭アドレスへの書き込みは、ブロードキャスト転送と同時に、各SLPへの割り込みを引き起こす。一方、各SLPも1語の仮想領域を持っており、全SLPによる仮想領域への書き込みの完了と共に、制御を受けているMPへの割り込みを引き起こし、同時に、各SLPにより書き込まれたデータの論理和が対応するMPに送られる。また、MP間では、これらとは全く独立して同期を取り合っている。

一方、各SLPは、MPから制御を受けるSMバンクと共に、異なるMP系統に属する隣側のSMバンクもアクセスすることができる。このように、SLP間でのデータ交換は、隣接するSLPの場合はデータの移動なしで直接SMバンクで、また、遠隔SLPの場合はどちらかのMP経由で行うことができる。

4. 2 連続系シミュレーションへの適用

一般に、並列処理システムでは、データ転送が特定の部分に集中しやすく、並列化の効果を打ち消す結果となりやすい。本システムでは、シェアド・メモリ・システム構成上の特長、すなわち、隣接するSLP間でのデータ交換が不要であること、遠隔SLP間でのデータ交換は各MPで半数を行えばよいこと、共通データはブロードキャスト機構で一度に転送できること等を生かし、転送量の減少と転送路の並列、複数化ができるメモリ割り当てを行っている。

プログラムはLMに、変数はSMに割り当てている。特に、各SMバンクでは、FIX 0領域には全域の情報、すなわち、システム変数、分積分出力変数(隣り合うSMバンクに半数ずつ)、パラメータ等を割り当てている。EXC 0とEXC 1のそれぞれの領域には、そのSLPでの演算に必要な変数を入力域と出力域として同じ形で割り当て、MPのフラグの切り換えによっていつも同じアドレスがアクセスできるようにしている。現在、各SLPにそれぞれ6積分器を割り当てている。

シミュレーションの実行は、MP間で同期を取り合いながら、単位フレームの処理を次々と繰り返すことによって進められる。単位フレームは計算フェーズとデータ交換フェーズとからなり、計算フェーズでは各SLPが互いに独立に割り当てられた演算を1積分刻み分を行う。このときMPはSLPの演算と並行して、入力域から解表示用のデータを取り出す。それに続くデータ交換フェーズでは各MPが次の計算フェーズのためにデータの更新を行う。すなわち、遠隔SLP間のデータ転送と、フラグの切替えによるEXC 0とEXC 1との入力(出力)から出力(入力)への役割りの交替である。

5. HOSSのソフトウェア・システム

5. 1 ソフトウェア・システムの構造

HOSSのシミュレーション用基本ソフトウェアとしては、ICOSS(連続系用; CSMP系統)とRISS(離散系; GPSS系統)とが用意されている。どちらも会話型で、オンライン実時間シミュレーションが可能なものである。

両システム共に、シミュレーション問題の標準的問題解決過程を分析した結果得られた“標準処理パターン”(図1)に基づき処理単位毎にモジュール化が行われ、十分な試行錯誤が行えるように処理間の遷移もこれに対応したものになっている(図2)。

このため、両シミュレーション言語共に、モデルの記述部（プログラム本体）は、実行環境の記述（初期値の入力、実行・解出力表示方式）とは完全に分離、独立しており、モデル記述部単独でコンパイルされロード・モジュールに変換される。実行環境はモデル記述部のコンパイル時の情報に基づき独自に編集、登録が行われ、その両者は実行時にダイナミックに結合される。また、実行環境は、実行時に、直接、指定や一部項目の変更ができる。

特に ICOSS コンパイラでは、ソーティングだけでなく、並列処理のためのタスク分割、各 SLP へのタスク割り当てなどの処理が、各 SLP で積分負荷が均等になり隣接するタスクが PPA 上でも極力隣接するように、すべて自動的に行われる。このとき、障害 SLP があるならば、それらを除いて割り当てが行われる。

以上を実行手順の面から見ると、大きく、モデルの記述、実行環境の記述、実行の3段階に分けられる。ICOSS での例を図5に示す。

モデルの記述には、両システムとも、誤り発生時点での誤り検出、修正が行えるように、それぞれに専用のエディタが用意されている。エディタでは、1ステートメントの入力毎にローカルな文法チェックを行い、誤りがあれば即座に修正を求めてくる。また、エディティング終了時にはグローバルな文法チェックを行っている。編集コマンド自体は比較的一般的なものがほとんどであるが、ICOSS では、偏微分方程式を連立常微分方程式に展開する場合などに良く生じる同型方程式の入力を考慮した反復展開用コマンド（\$REPT コマンド）が採り入れられている。

実行環境の記述は、大きく、パラメータの初期値の入力記述と、実行・解表示方法の記述とからなる。これらは、それぞれ最大10ケースまで登録でき、各ケース毎に独立したファイルが作成される。このとき実行・解表示方式の記述では、実行方法と出力方法とに分離されており、特に ICOSS では、出力方法が解データの出力媒体に応じ、グラフィック・ディスプレイ、ラインプリンタ、ディスクファイルのそれぞれを独立に指定できるようになっている。

初期値の設定時には初期値設定を必要とするパラメータ変数名が、また、実行・解表示方法の記述時には設定を必要とする項目名が、システム側からディスプレイ上に表示される。利用者はその表示に沿って値を入力して行けばよく、その場合、各項目にはデフォルト値が用意されている（例えば、ICOSS の初期値の場合には0（零））。さらに、設定値の変更、すでに設定されているケースの複写作成などの編集コマンドが、メニュー選択方式で用意されている。

実行は、初期値と実行・解表示方式の組み合わせを指定した後、シミュレーション・コンソール上の専用キーで進められる。専用キーとしては、START, HALT, RESET, CHECK, SINGLE-STEP (RISS の場合)などが用意されている。なお、初期値や実行・解表示方式の設定値を修正し組み合わせの変更後に再実行する場合、プログラムを再ロードする必要はない。

5.2 実時間シミュレーション

ICOSS, RISS 共に実時間シミュレーションが可能であり、実時間実行時にはそれぞれ PPA あるいは DSP とシミュレーション・コンソールのみで実行される。従って、実時間モードの実行では制御プロセッサ・レベルとのデータの転送はなく、シミュレーション・コンソールと SLP 付属の P I/O を用いて直接外部との交信を行い、他装置の処理速度あるいは他装置との通信速度からの影響を受けない。

また、必然的に、両者共にシミュレーション・クロックの管理は、クロックを一定間隔で進める単位時間方式（固定時間間隔方式）である。クロックのスケールは、実行時に実行・解表示方法記述で設定するが、ICOSS ではテスト・ラン機能で最高実行速度を計測することができ、RISS

HOSS ICROSS V1.0-1

SOURCE LIST

<pre>0001 PROGRAM CHAOS 0002 : 0003 : dx/dt = s*(y - x) 0004 : dy/dt = gx - y - xz 0005 : dz/dt = -bz + xy 0006 : 0007 PARAMETER S, G, B 0008 : 0009 : DX = S * (Y - X) 0010 : DY = G * X - Y - X * Z 0011 : DZ = -B * Z + X * Y 0012 : 0013 : X = INTGR1 (ICX, DX) 0014 : Y = INTGR1 (ICY, DY) 0015 : Z = INTGR1 (ICZ, DZ) 0016 : 0017 END</pre>	<pre>***** CONTROL DATA SET ***** CASE No. 1 EXECUTION : Y PRINT : N FILE : N DISPLAY : Y *** EXECUTION CONTROL *** (1) SIMULATION MODE : NON-REAL (2) INTEGRATION METHOD : RKS (3) DELTA TIME : 0.001 (4) FINISH TIME : 1.7E+38 (5) FINISH CONDITION : N (6) BREAK POINT TIME : N (7) BREAK CONDITION : N *** DISPLAY CONTROL *** (1) DISPLAY TYPE (1,2,3) : 1 (2) DEVICE (REFRESH,STORAGE) : ST (3) ORIGINAL POINT LOCATION : X = 10 Y = 7 (4) OUTPUT DELTA : 20 (5) DISPLAY PERIOD : START TIME = 0.0 END TIME = 1.7E+38 (6) TITLE SIMULATION SOLUTION (7) X-AXIS DOCUMENT 1 VARIABLE NAME : X 2 AXIS CHARACTER COLOR : GREEN 3 ORIGINAL POINT VALUE : 0.0 4 UNIT VALUE : 5 (8) Y-AXIS DOCUMENT NUMBER OF DISPLAY VARIABLE 2 B-1 1 VARIABLE NAME : Y 2 DATA LINE TYPE : SOLID 3 DATA LINE COLOR : GREEN 4 ORIGINAL POINT VALUE : 0.0 5 UNIT VALUE : 5 B-2 1 VARIABLE NAME : Z 2 DATA LINE TYPE : B 3 DATA LINE COLOR : GREEN 4 ORIGINAL POINT VALUE : 0.0 5 UNIT VALUE : 5 (9) Y-AXIS VARIABLE NAME : Y</pre>
--	---

(a) ソース・プログラム

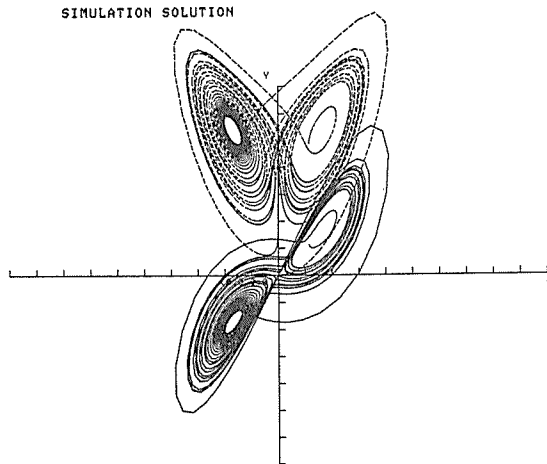
***** PARAMETER LIST *****

CASE No. 1						
1	S	=	12.0			
2	G	=	28.0			
3	B	=	2.6666			
4	* ICX	=	0.1			
5	* ICY	=	0.1			
6	* ICZ	=	0.1			

(b) パラメータ・データ

(c) 制御（実行・解表示）データ

PG.CHAOS CASE PNO= 1,CNO=



(d) 解

図 5 ICROSS の実行例

においてはコンソール・パネル上のホイールで実行時にダイナミックにクロック・レートを変えられることができる。さらに、RISSには、機能ブロック毎に処理を進めるシングル・ステップ・モードが用意されている。

ICOSSのテスト・ラン機能は、最高フレーム実行速度を求める機能であり、この速度がその問題に対するPPA自身の持つ最高実行速度を示す。すなわち、PPAは、指定された時間の間実行を行い、その間自動的に各フレームの実行に要した最大時間を内部で計測し、その結果を表示する。このとき計測は10 μ secの内部クロックで行われる。従って、問題の複雑さ及び各SLPでの負荷の不均衡さ等にかかわらず、この時間より大きな任意の時間をシミュレーション・クロックとして設定した実時間シミュレーションを行うことができる。

6. む す び

北海道大学で開発され、北海道大学全学共同利用施設汎用シミュレータ施設に設置された全デジタル式汎用シミュレータ「高速システム・シミュレーション装置」HOSSについて、その設計目標、ハードウェア・アーキテクチャ、ソフトウェア・システムについて述べてきた。

主目的である連続系実時間シミュレーションのために、マスター・スレーブ方式の並列演算装置PPAがあらたに開発された。PPAでは、ブロードキャスト機構を有するシェアド・メモリでSLPを結合し、データ交換の処理が集中しやすいMPを複数化しそれらを並列に用いることにより、データ転送時間の減少をはかっている。また、その1次元循環アレイ構造によるシステム構成は、対象プログラムの並列割り当てを柔軟性のあるものにし、並列処理システムにとって致命的となりにかねない一部の演算プロセッサの障害に対しても強い構造である。

PPAの性能は、ICOSSによるサークル・テストの結果では、1フレーム・タイムが3次のルンゲ・クッタ法の場合には約1ms、2次のアダムス法の場合には約0.5msである。また、周波数応答は、3次のルンゲ・クッタ法を用いた場合最も良く、3桁の精度で12Hzの応答を得ている。このように、計算精度、解の再現性、取り扱える問題規模が解決され、32ビット浮動小数点演算の採用によりスケール・フリーとなったことは、一般の利用者が共同利用する実用機としては最大の強みとなっている。

ICOSSでは、従来検討が遅れていた、連続系シミュレーション時の解の安定性などの評価のための補助手段を確立する新しい試みとして、rewind, noise controlの両機能を組み込んでいる。rewind機能は、任意時刻までの実行の後、その時点の各変数の値をそれぞれの初期値とみなし、シミュレーション・クロックを逆向き（積分刻み副を $-\Delta t$ ）にして実行し出力表示するものである。一方、noise control機能は、指定した変数に指定したスケールの雑音（一様乱数）をプログラムの変更なしに自動的に重畳しながら実行を進めるものである。両機能共に今後さらに検討を重ねる必要はあるが、このような検証手法の開発と利用法の確立、体系化がシミュレーション技法の一分野として重要である。

本シミュレーションは、システム要素の一つ一つの性能向上、システムとして統一化された有機的結合、ソフトウェアも含めたシステムの設計などにより、高速高精度の実行、容易な試行錯誤、初心者もオープン利用できる使い易さといった所期の目標を達成している。

謝 辞

本北海道大学「高速システム・シミュレーション装置」HOSSは、1978年度及び1979年度の2

年度にわたる文部省特別設備費によって設置され、1980年3月に一応の完成を見た。仕様の決定にあたっては、北海道大学全学共同利用施設汎用シミュレータ施設利用開発小委員会で、それ以前の調査段階まで含めると5年にも及び詳細な検討が行われた。また、HOSSの製作全般にわたっては三井造船株式会社システム本部の積極的協力を得た。HOSS実現までに払われてきた多数の関係各位の多大な努力に対し深く敬畏の意を表すものである。

上記利用開発小委員会の委員として長期にわたる検討期間を通じ多くのアイデアを出され、HOSS実現に熱意を傾けて来られた故小山昭一先生（当時北海道大学工学部精密工学科助教授）に対し、謹んで哀悼の意を表します。

参 考 文 献

- 1) Koyama, S. and Miura, R.: Proc. 8th AICA World Congress, (1976), p. 263~269
- 2) Koyama, S. and Miura, R.: Proc. 7th IFAC World Congress, (1978), p. 781~788
- 3) Makino, K., Koyama, S., Miki, N., Iseki, Y., Kobayashi, H. and Sakai, Y.: Proc. 9th IMACS World Congress, (1979), p. 171~177
- 4) Iino, Y., Makino, K., Iseki, Y., Koyama, S. and Miki, N.: Proc. 9th IMACS World Congress, (1979), p. 261~270
- 5) 牧野圭二, 小山昭一, 三木信弘: 信学技報, Vol. 80, (1980), No. 75, p. 55~66
- 6) Koyama, S., Makino, K., Miki, N., Iino, Y. and Iseki, Y.: Proc. 8th IFAC World Congress, (1981), p. 1715~1720
- 7) 牧野圭二, 三木信弘: 信学技報, Vol. 82, (1982), No. 62, p. 57~68
- 8) “汎用シミュレータ室だより”, No. 1, (1970)~No. 15, (1985), 北海道大学全学共同利用施設汎用シミュレータ施設