JAERI-M
89-023

原子力知能化システム技術の研究
(人間動作シミュレーション・プログラム：HASP)
—昭和63年度作業報告—

1989年3月

浅井　清・上中　淳二*・神林　奨・樋口　健二
久米　悦雄・藤崎　正英**・藤井　実・横川三津夫

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-Mレポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319－11茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　日立高速印刷株式会社

原子力知能化システム技術の研究
（人間動作シミュレーション・プログラム：ＨＡＳＰ）
－ 昭和63年度作業報告 －

日本原子力研究所東海研究所計算センター
浅井 清・上中 淳二*・神林 奨・樋口 健二
久米 悦雄・藤崎 正英**・藤井 実・横川三津夫

（1989年2月1日受理）

日本原子力研究所は，1987年よりＨＡＳＰ（Human Acts Simulation Program）と名付けた人工知能とロボティックスに関する研究を10年計画で開始した。

ＨＡＳＰでは，知能ロボットが自然言語で記述された作業命令を読み，意味を解釈し，自己の行動を計画し，動作列を生成し，装置や機器が保有する情報を採用し，その動作列を精密化してプラント保守作業を遂行する。ロボットの被曝線量計算を含むこれらの過程はすべて論理計算と数値計算によってシミュレーションされる。3次元空間において計算されたロボット行動は，高速画像処理装置を使用して映像化される。

ＨＡＳＰプロジェクトの目標は，(1)知能ロボット設計の基盤技術の開発，(2)プラントの知能化，自動化技術の開発，(3)人工知能関連のシステム化された基盤技術の原子力分野への提供，である。

研究項目には，自然言語理解，ＬＩＳＰによる動作計画，神経ネットワーク手法によるパターン認識，ソリッドモデルによるプラントのモデル化，二足歩行ロボットの動作シミュレーションと映像化，被曝線量計算の高速化を目的とするモンテカルロ・ベクトルプロセッサの概念設計研究がある。

本報告書は，ＨＡＳＰプロジェクトの2年目に達成された研究成果について記述される。

東海研究所：〒319-11 茨城県那珂郡東海村白方字白根2－4
* 外来研究員，ＣＳＫ
** 外来研究員，富士通

A study on Intelligent Nuclear Systems (HASP: Human Acts Simulation Program) — Progress Report 1988 —

Kiyoshi ASAI, Junji UENAKA*, Shaw KAMBAYASHI
Kenji HIGUCHI, Etsuo KUME, Masahide FUJISAKI**
Minoru FUJII and Mitsuo YOKOKAWA
Computing Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received February 1, 1989)

In 1987 Japan Atomic Energy Research Institute has started a ten-years program named HASP, i.e., Human Acts Simulation Program, for artificial intelligence and robotics research.

In the HASP, a human-shaped robot reads and understands orders written in natural language, planning and producing a required sequence of actions, accesses to a device or an instrument recognizing its entity and does the ordered work for plant maintenance. All of these processes including calculation of the radiation exposure of the robot are simulated by logical and numerical computations. The simulated actions of the robot in three-dimensional environments are displayed using a high speed computer for graphics.

The aim of the HASP project is threefold, i.e., (1) to develop fundamental technologies for design of intelligent robots, (2) to develop technologies for automated and/or intelligent plants, (3) to provide researchers and engineers in nuclear field with basic and systematized artificial intelligence techniques.

The research items are natural language understanding, goal planning

---

* On leave from CSK Corp.

** On leave from FUJITSU Ltd.

by Lisp calculus, pattern recognitions by neural network methods, plant modelling by solid modeller, biped robot simulations, graphic display of the robot motion, and a study of design concept of a Monte Carlo vector processor for high speed calculation of the radiation exposure.

In this report research results attained in the second year of the HASP project are described.

Keywords: Artificial Intelligence, Natural Language Processing, Knowledge Base, Neural Network, Pattern Recognition, Robotics, Graphics, Monte Carlo Method, Supercomputers, Simulation

目　　次

Contents

1. 人間動作シミュレーション技術研究の概要

1.1 はじめに

国の原子力開発利用長期計画の改定作業が昭和61－62年度に行われ，62年度に新しい計画が策定された。これにより，原子力委員会のもとに基盤技術推進専門部会を設置し，原子力用の材料，人工知能，レーザ，放射線リスク評価・低減化の四基盤技術の研究開発を推進することになった。日本原子力研究所では人工知能技術としては，センサー知能化，人間動作シミュレーション技術の研究開発に取り組むことになった。

ここで紹介するのは人間動作シミュレーション・プログラム（Human Acts Simulation Program，略してHASP）である。HASPの概念は，Fig.1.1のように，自然文で書かれた命令を計算機のなかで作り出された模擬人間，即ち，ソフト的知能ロボットに与え，ロボット運動によって動作を行い，その過程を実時間で映像化しようとするものである。

この研究の第一の目的は，知能ロボット設計の下敷きとなる技術の開発である。第二にはプラントの知能化・自動化技術の開発である。第三には人工知能関連のシステム化された基盤技術を原子力分野の研究者，技術者に提供することである。

HASP計画は昭和62年度から10年間を予定している（Table 1.1）。62年度の人員は，専任6名（うち外来研究員2名），兼任3名（50%），63年度は，専任6名（うち外来研究員2名），兼任2名（50%）である。

ここではHASPの構成要素となる諸技術の63年度作業の内容の概略について報告する。紙数の都合により，63年度の研究内容に重点を置いて説明しよう。なお，知能ロボットの研究開発では，我が国では電子技術総合研究所・機械技術研究所の報告[1]，最近の知能移動ロボットシンポジウム講演論文集[2]，国外ではオークリッジ国立研究所の研究計画[3]が参考になる。これらはHASPに比べると，いずれもハードウェアよりである。

Table 1.1 Research schedule of Intelligent Nuclear Systems.

研究項目＼年度	62	63	64	65	66	67	68	69	70	71
単体シミュレーション技術の研究										
① 認識・動作技術の研究										
(1) 命令理解	(構文解析)	人工知能用ワークステーション購入（基本推論機能）		特定プラント知識ベース 作業指示書解読，動作指示言語への翻訳システム						
(2) 環境認識	(技術調査)		（位置認識，物体認識）					（情景解析）		（総合化）
(3) 動作判断・動作	(技術調査)	（動作指示言語，歩行）		（手・足・指・関節）					（総合化）	
② 環境設定技術の研究	(技術調査)	CADワークステーション購入（形状モデリング技術）		特定プラントの形状，材質モデリングとデータベース構築						
③ 映像化技術の研究			（形状解読映像化装置導入）					（同高性能化）		
④ 環境変化の影響予測技術の研究		（動的モンテカルロ・コード開発，個人被曝線量）					（集団被曝線量）		（被曝データベース）	
⑤ 超高速計算手法・シミュレーション技術の研究（超高速モンテカルロ装置）	（概念設計）		（詳細設計）	（製作）	ネットワーク化	（実用）				
⑥ システム化技術の研究					（模擬過程動的変更（単））（各種入力動的変更（単））			被曝・工期最適化		
多体シミュレーション技術の研究										
						（複数システムへの構成）		（模擬人間相関モデル開発の着手）		
ヒューマン・ファクタ研究ステーションの設計・試作						（シミュレーション・コード設計・試作）		（実験）		

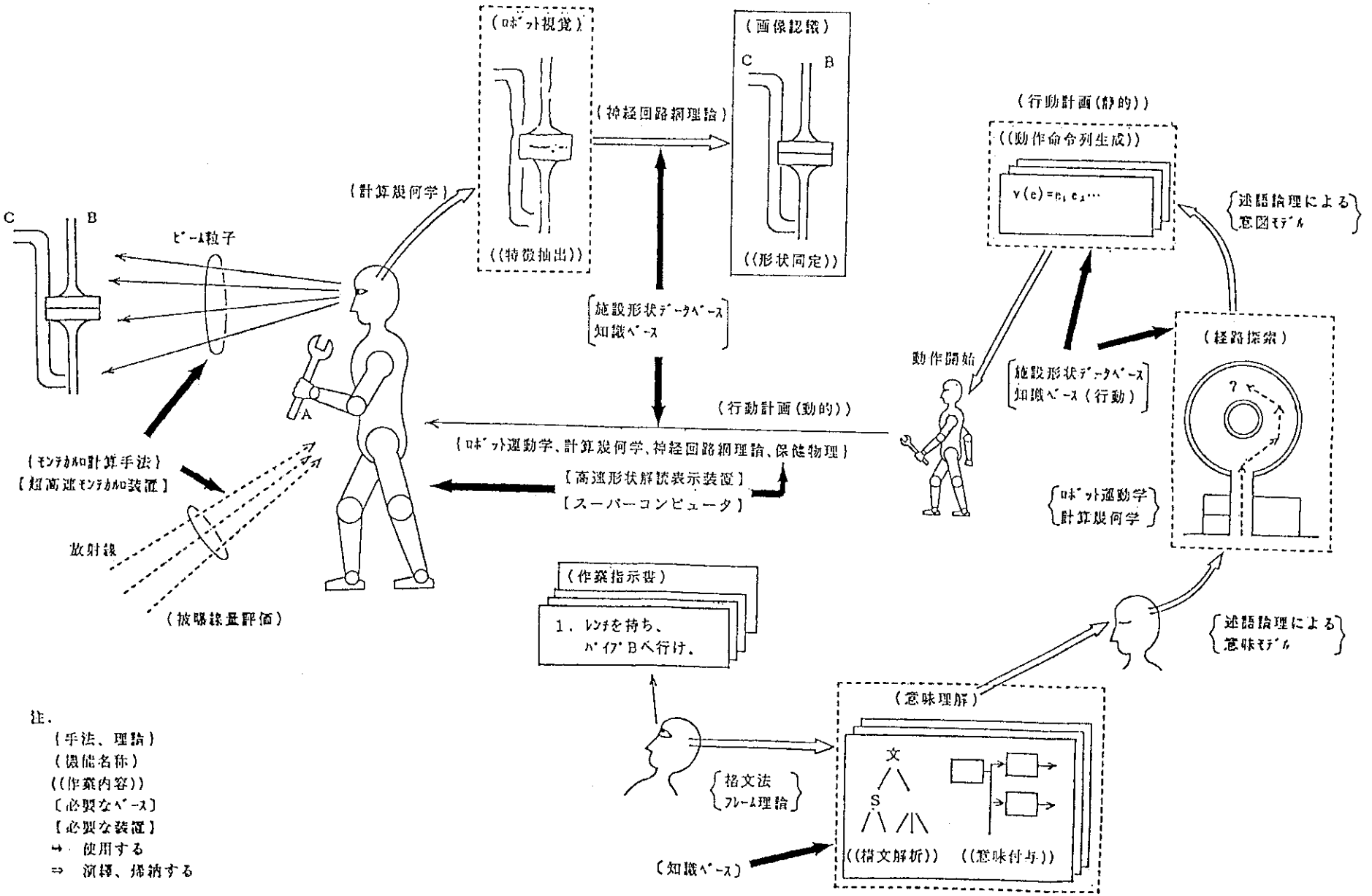


Fig. 1.1 A concept of Human Acts Simulation Program (HASP).

1.2 研究の概略

我々の研究内容を模式図で表現したものがFig. 1.1である。使用する手法，構築すべき知識ベース等を記号で示している。昭和62年度に研究を開始し，この報告が出る頃には二年が経過していることになる。しかし，この報告の原稿を執筆している段階では実質一年の作業期間であった。この一年間に，HASPを構築する要素技術の研究について計画通りの進展が見られた（Table 1.2）。この一年間HASPを構成する各種の要素技術，特に知能ロボットの行動計画の方法論を考察しているうちに，知能ロボットの動作と関連する知識ベース，施設形状データベースの在り方についてひとつの考えがまとまってきた。それを一言でいえば，ロボットが動作する空間と，ロボットとの知能分担によるシステム構築の方法論である。空間が知能を持つとは，従来の知能ロボット設計論からみるとまことに奇異な考え方であり，一般に受け入れられるには時間がかかるであろう。従来の知能ロボット論においては，ロボットに，その動作する空間に関する総ての知識を持たせることが普通である。しかし，従来の方法には，

a) ルールベースで知識を与える手法では，蓄積・探索すべきルール数が多くなり，組合せ爆発を引き起こす，

b) 知識を与えられていない空間内では動作できない，

c) 空間とロボット動作の相関から生まれる総ての状況を事前に知識化して，知能ロボットに与えることはできない，

d) 空間の形状，動作に伴う状況変化等の経験を知識化して学習することが困難である，

などの難点がある。

これらの難点は，知識を状況や人間の身体的動作から切離した静的なものとして，取り扱うことから派生しているように見える。筆者らの見方は，メルロー＝ポンティなどが主張しているとされる次の(1)～(3)のような人間理解の新たな論点[4]と呼ばれるものと軌を一にするもののように筆者には思われる。その論点とは，

(1) 対象についての経験を組織化し統一する際には身体の役割が不可欠のものである，

(2) 行動が規則によらず組織化されるためには，「状況」が不可欠の役割をはたす，

(3) 状況というものを組織化する際には，人間の意図や欲求が重要な役割をはたす，

などである。

HASP計画は，必ずしも人間的動作・思考のシミュレーションを追求しているわけではないが，知能ロボットの動作する空間，装置が人間を対象として作られているために，上記(1)～(3)と類似の考えに自然と行き着く。そこで問題は上記(1)～(3)のような論点をどのように機械化するか，或いは容易に制御可能な機械的表現を得るかということになる。このために筆者らは次のような方法論を採用することとした。

(イ) 施設内の装置，計器等は，その用途・形状に応じて人間（作業員）からどのように取り扱われて欲しいかを示す基本動作要求列を知能ロボットとは独立に付与されている。

(ロ) 知能ロボットが意図を持って当該装置に近づいた場合に基本動作要求列が装置からロボットへ発信される。経験の無いロボットは受信を繰り返すことで取り扱い手順を習得していく（その手法はこれらの研究課題である）。経験のあるロボットには動作すべき手順確

認の手掛りとなる。

(ハ) 知能ロボットは，作業指示書から動作径路及び大まかな動作列を生成するが，これが最上層の意図を表すと見做す。動作径路上の廊下，ドア等も上記(イ)，(ロ)でいう装置である。

(ニ) 上記(ハ)の動作列は大まかすぎるので，さらに細かい基本動作列へと必要に応じ展開する（展開の方法論は今後の研究課題である）。これは意図の中間層的展開である。装置から発信される基本的動作要求列は，この中間層的展開と矛盾してはならない。

(ホ) 装置等に付与した基本動作要求列は，筆者らのモデルでは施設形状データベースの樹状構造化された該当部分に付属情報として与えられる。

(ロ)～(ニ)の動作列への展開は，施設形状データベースを参照して行われる。

ここまでの動作列展開は静的であって，知能ロボットが動作前に行う予備的な動作展開である。次は実際の動作に伴う中間層及び最下層の動作列生成である。

(ヘ) 我々の知能ロボットが知るのは，知識ベース及び施設形状データベースを使ってコンピュータ上で構築したプラント内の形状，装置である（この際の形状同定手法は今後の研究課題である）。

(ト) 装置等に付与した基本動作要求列はサーブリック表現の変形[5]を使用している。その一例をFig. 1.2に示す。これは人間（作業員）動作を１２種程度の基本的動作で記述しようとするものである。サーブリック表現による基本動作のひとつに「調べる」という動作がある。これは最も人間的な動作であって，動作対象となった装置等の要求動作列及び状況によってしか動作を決定できない。その動作も基本動作よりももっと細かいものであって，素動作というべきものである。素動作は最下層の動作である。その例をTable1.3に示す。

(チ) 上記(ハ)～(ト)における動作列展開に使用する手法をTable 1.4に示す。P.R.Cohen & H.J.Levesque[6]，R.C.Schank & J.Meehan[7]の手法の枠組みは第一階の述語論理に近い。

上記(イ)～(チ)の方法を採用することによって，知識ベース，施設形状，ロボット動作は相互に補完的となり，それぞれが独立に機能することはできない。

Table 1.2 Research results attained in 1987 and 1988.

	昭和６２年度	昭和６３年度
1．命令理解	・自然文各種意味理解手法（高木の方法、モンテギューの方法）の調査 ・神経ネットワーク手法の各種応用例の調査 ・機械翻訳システムGRADE の整備 ・核データ評価プログラムの日本語化と機械翻訳	・高木・伊藤の方法による簡易なパーザーの採用 ・日本語による知識ベース構築の検討 ・動作シナリオ（研究炉作業員）の作成と意味理解構造の解析
2．環境設定	・セルラ・アレイ・プロセッサ、レイ・トレーシングなどの画像化装置、手法の調査 ・研究炉一次冷却系設備のリゾット・モデル化及び画像化	・研究炉一次冷却系設備周辺モデルの精密化 ・ロボット躯体モデル化 ・施設形状データベース化
3．環境認識	・視覚、径路捜索、行動計画については未着手 ・動作判断・動作ーロボット歩行プログラム整備、二次元映像化	・径路捜索手法の調査 ・動作判断・動作ーロボット歩行精密化、映像高速化、動作計画展開ツールの開発 ・神経回路網理論による視覚情報処理シミュレーション
4．映像化装置		・低速画像解読表示装置のリース
5．高速モンテカルロ計算装置	・６２年度までスーパーコンピュータに不適の原因調査．原研型演算回路概念の一部の有効性を中性子輸送計算用原子力コードで調査検討．６３年度に概念検討	・原研仕様による高速モンテカルロ計算装置のフィージビリティ・スタディを数社に委託調査
6．高速数値シミュレーション	・原子力コード４本を使用したベクトル・パラレル計算機の高速数値計算の適応性検討	・原研仕様モンテカルロ計算機の有効性シミュレーション
7．環境影響評価		・動的ファントムの有用性検討

Table 1.4 Generation method of action sequences.

階　　層	動 作 列	手　　法
最上層	大まかな動作列 ⇩	P. R. Cohen & H. J. Levesque
中間層	基本動作列 ⇩	R. C. Schank & J. Meehan
最下層	素動作列	ad hoc procedures による数値シミュレーション

Table 1.3 An example of the basic action sequence.

（物を）つかむ
（左手把握中、つかむ）→視る、判断する（探す、歩く）置く、
探す、歩く、判断する、つかむ
①（視る、Ｐ？）　Ｐ＝置く場所の有無
②（判断する、Ｐ？）Ｐ＝置くこと可能と判断する
③（探す、Ｐ？）　Ｐ＝置く場所を視覚で探す
④（歩く、Ｐ？）　Ｐ＝置く場所へ行く
⑤（置く、Ｐ？）　Ｐ＝姿勢を変える必要性？
不必要なら置く
⑥（探す、Ｐ？）　Ｐ＝把握対象物を探す（視覚）
⑦（歩く、Ｐ？）　Ｐ＝把握対象物へ近づく
⑧（判断する、Ｐ？）Ｐ＝対象物を認識する
⑨（つかむ、Ｐ？）　Ｐ＝把握可能と判断する、つかむ

ａ：Ｐ＝置く場所の有無→把握物の形状、大きさ、重量
ビーム放射
把握物分場所の切り取り
ｂ：Ｐ＝判断する　→場所の形状（平坦、凹又は逆に凸でない）

＜知識ベース＞
ａ→　物の大きさ、重量、把握物の形状の記憶
↓
神経ネットによる２次元の形状記憶包含関係判定による広さの判定
ｂ→　場所の形状
・施設形状データベースとロボット位置から
・ビーム放射による形状判断から
（把握物の大きさ程度のビーム放射から形状判断可能か？）、

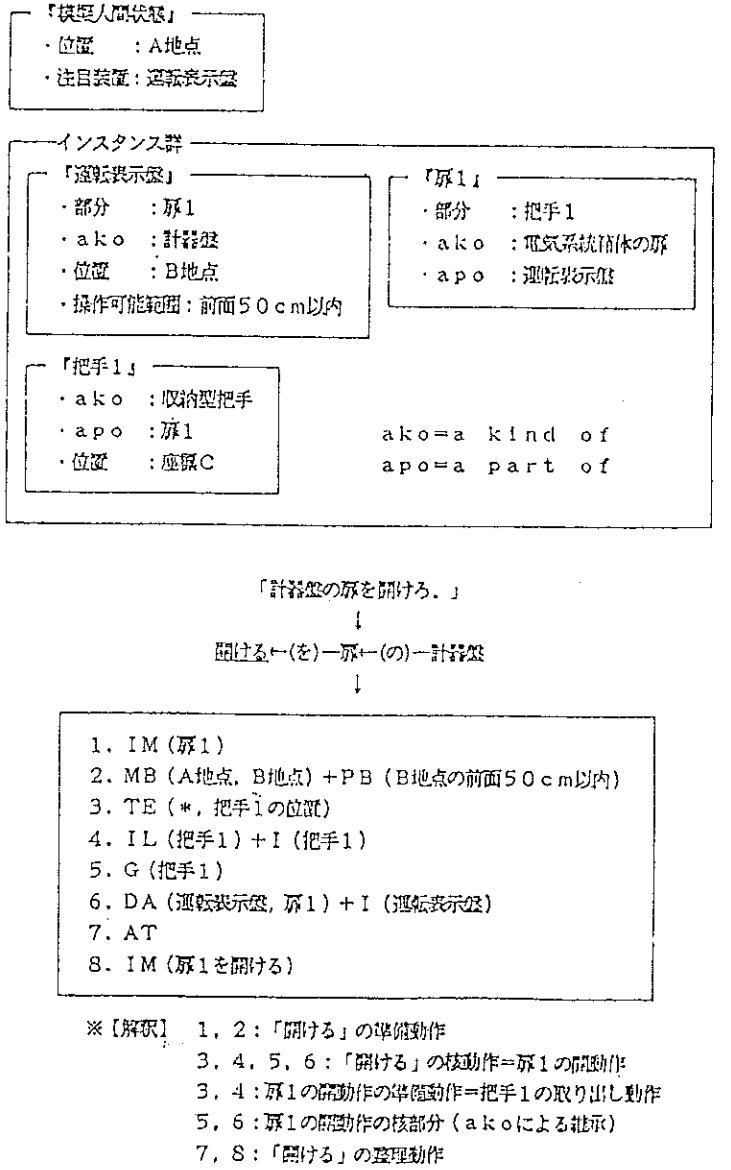


Fig. 1.2 A expression of the action sequence.

1.3　63年度作業の概要

人間動作シミュレーション技術の研究は，原子力知識ベースの研究と超高速数値シミュレーション技術の研究の二つに大きく分けられ，前者は命令理解，環境認識，環境設定技術に関する研究を，後者は動作，映像化，高速数値シミュレーション並びに超高速モンテカルロ計算機の開発に関する研究を行う。

各研究項目の63年度作業の概要を以下に示す。

I．命令理解

命令理解では，自然言語で記述された作業指示文を意味素を用いた意味構造に変換し，そこから動作列に展開する手法をとる。そのため，意味素を抽出する作業と意味素に付随する知識データの構築が必要である。そこで，以下の項目を昭和63年度作業として行った。

(1)　日本語解析プログラムの導入

自然文から意味素をノードに持つ本構造に変換するプログラムとしてCS-PARSERを導入した。さらに，辞書拡充のため，辞書データを入力するためのツールを作成した。

(2)　物語生成プログラムの整備と解析

ロボットが原子炉内で行う意味決定の様式をシミュレートする方法の一例として，喉の渇きや空腹といった簡単なゴールとそれを解決するためのプランを用いた物語生成プログラムMicro tale-spin（R.C.Schank，J.Meehanらによる）をcommon lisp上に移植し，その動作を解析した。このプログラムを実行させる際に現れる動作記述表現を，ロボット動作に結合するための知識構造を検討した。

(3)　炉内作業シナリオの作成と解釈機構の検討

原子炉内作業を示すシナリオを三種類（a.計器室内計器盤点検，b.バルブ保守，c.手動仕切弁保守）作成した。この中でa.に現れる自然文をもとに，装置に固有の知識として，ロボットの基礎動作列への展開機構の検討を行った。

(4)　解釈システムの試作

(3)で検討した指示文の解釈機構に基づき，指示文から基本動作に展開するシステムの試作を行った。

II．ロボットの画像認識（環境認識）

神経回路網（ニューラル・ネットワーク）に関する理論を画像認識に応用するために以下の作業を行っている。

(1)　BPモデルによる手書き数字の認識実験

誤差伝搬学習アルゴリズムを用いたニューラル・ネットワーク・モデル（Back-Propagation model，BPモデルと略す。）を使用して，0～9の手書き数字の認識実験を行った。BPモデルは，比較的簡単に作成，実行できるが，提示するパターンの位置ずれや大きさの違いに弱いことがわかった。

(2)　ネオコグニトロン・モデルによる簡易画像の認識実験

提示パターンの位置ずれや大きさの違いをある程度吸収できるとされているネオコグ

ニトロン・モデルを使用して簡易画像の認識実験を行っている。ネオコグニトロンは特徴抽出機構等のモデル化が難しい上に処理プログラムもＢＰモデルの１０倍以上の行数が必要である（作成中）。

Ⅲ. 二足歩行ロボットの運動学的シミュレーション（動作，映像化）

動作研究では，二足歩行ロボットの歩行動作についての運動学的シミュレーションに取り組んでいる。また，両腕，指の協調動作に関する調査も開始した。

(1) 歩行運動の多様化

６２年度に整備した二足歩行ロボット運動学計算プログラムを用いて，６２年度の平地歩行に加え６３年度は階段歩行，停止・発進動作，歩調・歩幅の変化を取り上げた。これらの計算結果に基づいてコマ取り風の画像を作成した。

(2) 歩行運動の映像化

原子炉建屋内で人間が平地を歩く様子を映像化した。ＣＡＰを使って人間を軀体モデル化し，人間の歩く速度に即した動画を作成した。

(3) 原研版二足歩行ロボット運動学計算プログラムの開発

ヴコブラトビッチの"歩行ロボットと人工の足"という本に，６２年度整備したプログラムの発展形として，腕を自由にした歩行モデルの運動方程式の導出方法が記載されている。６３年度は，これに基づき原研版の計算プログラムを開発した。

(4) ロボット運動学に関連した調査

両腕協調・指の協調動作等に関する運動学方程式の調査を大学に依頼した。

Ⅳ. 施設形状データベース

人間動作シミュレーション技術の研究では，ＪＲＲ－３施設の保守・点検作業を当面の対象としてシミュレーションを行っている。施設形状データベースでは，ＪＲＲ－３建屋をＣＳＧ形式（対象となる物体の形状を幾つかの基本的な立体の集合演算によって表現する形式）によってモデル化し，映像化を行った。以下に６３年度の作業内容を示す。

(1) ＪＲＲ－３建屋の施設形状データの入力作業

６２年度より開始した入力作業を継続して行った。６２年度入力分の一次冷却系配管に加え建屋の壁，床，原子炉本体及び周辺機器の入力を行った。

(2) 機器の導入

シミュレーション結果の映像表示を行うためにソフトウェアとしてモデリング・ソフトＦＵＳＩＯＮを，動画作成システムとして並列計算機ＣＡＰ，Ａ－５０，ビデオ・システム等の機器導入を行った。

(3) 施設形状データ形式変換ソフトウェアの作成

ロボットの視覚認識，自己可動空間認識，被爆線量評価等の計算を将来大型計算機上で行うために，施設形状データ及び画像情報を大型計算機上に移行するソフトウェアを作成した。また，二足歩行シミュレーションの結果からロボットの動画を作成するために必要なソフトウェアも作成した。

(4) 動画作成

上記の導入した機器と(3)で作成したソフトを使用してＪＲＲ－３内の情景と二足歩行シ

ミュレーション結果を合成した動画を作成し，約３分のビデオを作成した。

V. 高速モンテカルロ計算装置

(1) 調　　査

中性子・光子輸送計算の代表的モンテカルロ・コード４本を対象とし，原研提案の有効性を調査した。作業時間の制約から，主としてそのうちの１本である臨界安全性計算コードKENO IVについて精力的に調査を行った。その結果，既存スーパーコンピュータによる実測と外挿によって当初目標であるスカラ計算処理に比較して５～１０倍の処理速度向上の見通しが得られた。

(2) ソフトウェア・シミュレータの作成

モンテカルロ・コードの計算処理を模擬するソフトウェア・シミュレータを作成し，このシミュレータに原研提案の演算機能を組み込んだ場合のコードの処理速度向上率を予測した。シミュレーションの基本単位は，計算コードのＤＯループ，又はまとまったスカラ処理部分である。モンテカルロ装置のハードウェア命令の演算速度は既存スーパーコンピュータによるモンテカルロ・コード計算処理の実測値を内外挿して使用した。シミュレータによる計算時間の予測値と，既存スーパーコンピュータによる実測値とは，計算コードが精確にシミュレーションの基本単位に分割されている場合には数パーセントの誤差で一致する。

本報告書では，２章に命令理解，３章にロボットの画像認識，４章に二足歩行ロボット運動学的シミュレーション，５章に施設形状データベース，６章に高速モンテカルロ計算装置の63年度作業について記述する。

参考文献

1) Y.Shirai " Advanced Robot Technology Project ", Jour. Infor.Proc., vol.9 №2, Sept., 1986.

2) 第４回知能移動ロボットシンポジウム講演論文集，日本機械学会，昭和６３年６月１３日～１４日，東京。

3) J.Barhen et al., " Basic Research on Intelligent Robotics Systems Operating in Hostile Environment : New Development at ORNL", Proc.1984 Nat.' l Topical Meeting on Robotics and Remote Handling in Hostile Environment, 1984.

4) 黒崎" コンピュータに何ができないか "，人工知能学会誌，Dec. 1987.

5) 長町：現代の人間工学，朝倉書店，1986.

6) P.R.Cohen & H.J.Levesque " Persistence, Intention and Commitment ", AI Center and CSLI, March 1987, 解説は向井，談話理解とロジック，人工知能学会誌，May 1983.

7) R.C.Schank & C.K.Riesebeck " Inside Computer Understanding "，自然言語理解入門（石崎訳）総研出版，1986.

2. 自然言語処理と知識ベース

自然言語処理が人間動作シミュレーション（ＨＡＳＰ）においてどのような役割を果たすのかは，今後の研究を通じて問題となる。人間動作シミュレーションの研究は，その名が示すとおり，人間の動作を何らかの理論的裏付けにのっとってシミュレートしようとするものである。その中で，理論に不備な点があれば改良し，よりよいモデルを構築していく。動作という言葉は極めて広義の意味を持つと考えられるが，そのシミュレーションとは次の２点に集約されるであろう。

① 物理的な運動のシミュレーション

② 動作に関連した論理的思考のシミュレーション

ＨＡＳＰの中で自然言語処理が果たす役割とは，おもに②にあげた論理的思考過程のシミュレーションであろう。人間の用いる言語とは，人間がお互いの心理状態，認知状態について情報交換するために発展してきたものと考えると，その役割が明らかになるであろう。

そこで，我々に課せられる問題とは，理論的思考過程をシミュレートするために必要となる理論的裏付けを持った道具を発見していくこと，そして，そのような理論に基づいて人間の持っている知識を定式化していくことである。

従来の自然言語処理では，おもに既に存在する文章を処理することを（理解することを）念頭において進んで来た[1]、そのために構文的な情報をもとに文の構造を示す構文木の導出を狙っていた。しかし，人間は言語を聞いたり，読んだりすることでそのような構文木のみを先に認識しているのではなくて，むしろ，とても概念的なことをただちに認識するものと思われる。昨年度から始まった人間動作シミュレーションの研究の中で自然言語処理の部分は，既存の機械翻訳システムの拡張によって動作シミュレーションにつなげようと考えていた。しかし，概念レベルのシミュレーションの役割を無視していたので良い結果は得られなかった。それは次のような理由による。現在の機械翻訳システムでは，文字列（漢字，仮名，カタカナ，ローマ字）で与えられた文章を形態的な情報や構文的な制約をもとに“概念表現”と呼ばれるものに変換し，それを目標とする言語（英語など）の表現に置き換えている。“概念表現”とは呼んでいるが人間の思考の中にある複雑に絡み合った“概念”とは異なり単純な構文情報を示しているに過ぎない[2]，そのため，本質的な思考のシミュレーションにつなげるためには単純に機械翻訳システムを拡張しただけでは，問題が多すぎるわけである。

昭和６２年度の作業で具体的に行ったことは，機械翻訳システムを用いてＦＯＲＴＲＡＮプログラムから知識を抽出することであった。そこで明らかになったことは，知識を抽出する際に以下の点を明らかにすることであった[3]。

① 明確な（具体的な）最終目的の設定

② 段階的なモデルの作成

③ 意味理解方式の決定と自然言語処理プログラムの整備方法

知識とはきわめて広範囲なものであるから，知識ベースという限られた資源（ハード，ソフトの面から考えて）へと変換するためには，限定された使用目的のもとで扱う知識を抽出するべ

きである。また知識を抽出する際，全体の整合性を保つために巨視的な部分と微視的な部分の両方を調整して行かなければならない。そのため，段階的にモデルを作成し，検証することが重要である。自然言語を基に知識構造を抽出するためには，何らかの自然言語処理ツールが必要であるが，既存の機械翻訳システムを使うことでは，形態素解析と意味解析の融合が難しい。むしろ既存の機械翻訳システムの枠組みを離れたツールを利用して行かなければならないだろう。

上述の問題点を考慮した上で，今年度自然言語処理が扱うべき部分を次のように固定した。すなわち，自然言語が示している“意味”の役割を明らかにすること，そして言語生成の陰にある問題解析過程をシミュレートすることとした。さらに処理プログラムが扱う対象を，研究炉・原子力プラントにおける外観点検に現れる言語表現とし，そこから意味表現の構成要素を抽出することで知識構造を体系化する。

知識の体系化（意味表現の体系化）には，昨年度の調査で有効であると考えられた，意味素と係り受けを基本構造としたものを利用する[4]，この方法によると，具象物の属性表現が“意味素”と呼ばれる基本的要素を用いることで整合性のとれた概念体系を構成することができる。特に，作業対象となる具体的なものの属性をフレームを用いてまとめ上げるためには都合の良いものである。また，意味素を用いることによって，概念レベルの係り受けのスコープをはっきりとしたものにすることができる。

ロボット（あるいは模擬人間が）動作する場合，自分が行うべき達成目標を持ち，状況認識しながら動作を行う。問題解析過程のシミュレーションでは，状況の記述様式，達成目標の記述様式，そして，それらをもとにした問題解決プランの構築方法を明らかにしなければならない。プラント内の外観点検では，視覚と連動した状況確認が主たる作業内容となる。そのためには，作業対象物の把握とその対象物からえる情報を定式化しておく必要がある。このため，作業指示文書に書かれている事柄を上述の意味表現方式に基づき整理すると共に，そこからロボットがなすべきゴールの抽出を試みる。問題解析過程のシミュレーションについては，R．C．Schankらが示したプラン・ゴール・テーマと呼ばれる理論[5)6)]を，我々の問題に適用しようと試み，理論の習得を行った。彼らは，認知過程のシミュレーションと自然言語処理とを結びつけ，人間そのもののシミュレーションに取り組んでいる。彼らの示したプラン・ゴール・テーマ理論は，状況認識と動作のキーとなる目的を認識することに重点が置かれている。プランニングでは，そのプランを引き起こすゴールを概念レベルの意味ネットワークから見つけ出す。そして，それに関わるサブゴールを順を追って評価し，その結果をトレースすることで一つの行動が得られ，エコーバックとして一つの文章表現が得られる。

以下では，上に述べた二つの項目，すなわち問題解析過程のシミュレーションそして自然言語の意味を記述すること，に対し，その理論的背景，人間動作シミュレーションに適用する場合の問題点，作業の進展状況について述べる。

2.1　認知経過と言語生成シミュレーション

本節では，人間の認知過程と言語体系の関係について概観し，人間動作シミュレーションの

研究（ＨＡＳＰ）における自然言語理解の問題に，人間の意図を取り込む手法について述べる。なお，本節では“自然言語処理”という言葉を，おもに“計算機による自然言語処理”として使う。

2.1.1 認知過程と日本語

(1) 認知問題

計算機による自然言語処理とは，文章から意味を抽出して，それを計算機上に移植された人間の頭の中の既存の知識と照合させ，利用していくことだと理解されている。このような考えの背景には，人間の認知過程をモデル化できるという仮説が隠されている。

我々人間が遭遇する文は，ほとんどすべてが見かけ上，全く新しい文であるにもかかわらず，我々はそれが正しい文か否かを判定することができるし，その文の（新しい）意味を理解することができる。N.Chomskyは，そのような言語的知識の獲得が，単なる経験の積み重ねでは説明され得ず，生得的に備わった言語的直感によるものだとした[7]。つまり，全ての新しい実体の認識は，本質的には生得的に仕組まれている生まれながらの“認識”による再認の問題と見なすのである。再認とは，既に人間固有に備わっている概念と，いま取り扱っている実体とを結び付けることによってそれが何を意味しているのかを認識することである。

人間の認知過程を再認として扱うとき，初めて，意味の抽出とその突き合わせが問題となる。意味とは再認にかかわる“型”として，つまり知識として存在し，さらにそのような“型”を取り扱い再構成していく手続きを記述した知識が存在することになる[7]。

このような認知モデルに立脚すると，自然言語処理で必要となる知識とは，表層の文章を写像すべき，意味の型を作っておくこと，そしてそれを種々の問題解決プログラムで利用できる形態に加工する技術を意味する。

さて，人間の言語はそれが活用できる記憶と時間との制限の中で，互いの意志疎通を容易にしようとする中で発展してきた。このため，自然言語は話したり速読する為の機能に富み，論理的な完全性や厳密性を示すためにはつくられていない。例えば，省略文や照応現象，曖昧な単語の存在が当てはまるだろう。つまり人間は，読んだり聞いたりしている中で，失われた情報を復元することが可能なのである。もっと端的にいえば，人間はこれから聞こうとする内容を予想する能力にたけている反面，長い単語列を記憶することは苦手である。そのため理想的な自然言語処理では，統語解析のように文の最初の数語を聞いた時点で単にそれらの単語間の構文的な関係を見つけ出そうとするだけではなく，再認過程に要する“型”つまり事象の因果関係や概念的な結合を考慮する必要がある。

我々は，文理解・生成における基本的な要素を，概念を表す命題的な表現であると考える。命題もしくは述語表現によって得られる事象は，概念的に一つの主張を示しそれらの積み重ねが言語表現，すなわち文章を構成する。さらに，命題が導出される陰には，発話者が到達しようと考えている達成目標が存在する。この目標を解決するために，動作計画を中心とした知識を利用して様々な言語表現が理解・生成されるものと考える。つまり，命題的な概念表現や達成目標の設定が，再認における“型”の一部を構成する。

(2) 状況に強く依存する日本語

ＨＡＳＰで現在取り組んでいる自然言語は日本語である。日本語には再認問題と関連して非常に興味深い特徴がある。自然言語では，言語が成立する前提として，言語の構造と言語の運用形態の存在が重要である。文法論に代表される言語理論ではおもに言語の構造を取り扱っている。しかし本来言語とは言語の構造と運用の両者が一体となって，人間相互の言語によるコミュニケーションを可能としているのであるから，計算機による自然言語処理をおこなう上で，言語の構造の重要性を認めても言語運用のモデル化を省くことはできない。言語の運用形態とは，その言語がどのような状況のもとで活動するものかを定義する。日本語の場合，それは場面依存性が強いということで定義できる。つまり，日本語とは発話される状況によって様々な意味を作り出す言語なのである。そのような場面依存性の強さが，欧米で用いられる言語とはかなり異なった言語体系を形作り，日本語の特徴を作り出している。例えば，日本語には次のような特徴がある；①主語の省略，②文末決定性，③テンスの不規則性，④応答の曖昧さ，⑤句読点の無原則性。これらの特徴は，話し手と聞き手が作り出す状況に，強く依存する日本語の運用形態によってもたらされる[8]。

日本語は，きわめて強い場面依存性があり，そのために日本語を計算機処理するためには，場面の記述（状況の記述）がきわめて重要な課題となる。この状況記述に関する知識が，再認過程における“型”ではないかと考えられる。

2.1.2 意味表現方法

(1) 意味表現

自然言語の意味分野を考えるときに大切なことは，コトバ体系とモノ体系を区別し，常に考慮に入れることである[9]。人間が言語体系を形作る際に，人間の認知活動と外界（モノ体系）との相互作用の上でコトバ体系ができたのである。そのために，語彙の意味領域と現実の物体における意味領域が違った体系になる可能性がある。つまり，人間の認知活動の結果現れた論理的思考の結果，コトバ体系の意味的構成がモノ体系に先んじて形作られることがある。モノ体系における意味の区分が非常に明確な場合，そのことがコトバ体系に反映してモノ体系とコトバ体系の対応付けが明確なものとなる。原子力プラントの作業指示書に現れる語彙にかんしてはこのようなモノ体系とコトバ体系との対応付けが比較的明瞭ではないかと考えられる。

自然言語の体系は，大きく分けて静的な概念（名詞的概念）と動的な概念（動詞的概念）とに分けられる。静的な概念は，上に述べたモノ体系を具体的に表したものであり，その意味分野は具象物に備わっている属性を用いる方法が有効である。これに関して，具体的に作業指示書に現れる語彙を分類し自然言語解析プログラムの辞書体系を構築する際の手法とスケジュールについて2.2節に述べてある。

動的な概念を表すには次のような事実を考察する必要がある。具象物に働きかける概念を示す動的概念には，“マゲル・ネジル（変形）キル・サク（切断），ハズス・ハナス（位置変化），ツカム・ニギル（保持）”等があるが，これらの動作のタイプの違いによって，そこに関与する条件の種類が異なる。各々の条件の内容も動詞によって異なってく

るわけであるから，動的概念の語義領域の記述は，諸条件の組合せの記述ということになる。Table 2.1.1 にあげた切断に関する意味の記述においては，動作，対象物，道具，様態，結果などが動的概念の構成要素としてあげられている。動的概念の諸条件の組合せの記述に関しての構成要素は，単語の中に既に存在しているもので，それが認知過程における再認の“型”として作用すると考えられる。そこで，動的概念と静的概念の構成要素を明らかにし，再認のための“型”となるべく効率的に記述する必要がある。

(2) スキーマ

言語理解の問題に限らず，今日の認知心理学が扱っているほとんど全ての問題は，多かれ少なかれ，“認識＝再認”の公式に当てはまるものである[7]。もちろん，ここでいっさいが再認問題になっているといっても，いわゆる template-matching を意味するわけではなく，特徴の抽出，特徴の組合せ，さらにより上位の概念への統合などのプロセスを想定しているのが普通である。見かけ上，新しい知識や新しい概念が新しく構成されているかのように見受けられるかも知れないが，抽出されるべき特徴や構成されるべき上位概念は，結局のところ，“与えられているもの”として考えていく以外にないのである。意味の取り扱いをする際，現在の認知心理学における“認知＝再認”という公式に沿う場合，そこに現れる知識や意味といった概念は，再認によって突合せするためのテンプレート（template）やテンプレートが複雑に絡み合ったテンプレートが基本となる。そのようなテンプレートは予め用意したり，システムが類似によって適当に判断したり，作り出したりすることによって得られる。そこで文章理解とは，“文から意味を抽出して，それを頭の中の既存の知識と照合させること，そしてそれを利用して高度な処理を行うこと”と換言される。

再認という問題を考えるとき，様々なレベルの再認を取り扱う必要がある。いわゆる template matching（slot filling）も再認の一種である。template をいくつか組み合わせて新しい template を作り出すのも再認である。そして，関係構造を取り扱う template を組み合わせて新しいモデルを作り出すという再認も考えられる。つまり，再認の問題を考えるといっても，非常に低いレベルから構成的なもの，発見的なものまで考え合わせなければならないのである。

認識プロセスが基本的に，再認のプロセスとして取り扱われるということは，未知の入力信号 x が与えられたとき，x を頭の中に用意されている概念の枠組み X に帰属させることにより，“x は結局 X の一種か”と納得することである。ここで概念の枠組み X とは，概念が複雑に絡み合った概念の組合せでも構わない。そのような“構造化された概念の組合せ”のことをスキーマと呼ぶ。R.C.Schank らの示した Conceptual Dependency Theory（ＣＤ理論：概念依存理論）は，このようなスキーマの一つとして考えることができる（Fig.2.1.1）[5)6)]。

(3) Conceptual Dependency Theory（ＣＤ理論）

ＣＤ理論とは，文章を理解あるいは生成するために必要となる，人間の頭の中の“意味表現形式”（スキーマ）をモデル化したものである。ＣＤ理論では，与えられた文の意味というものを，生の文のままでなく頭の中の知識の表現形式に変換するが，この変換は，

表面上は異なる文章表現でも，同じことを意味しているときは同一の“意味表現”と対応付けられる点に特徴がある[5)6)]。

意味表現の研究では表現すべき事柄がとても多い。それらは，物理的事象，心理的事象，目的，物理的原因，心理的原因（すなわち理由），関連する事象や目的に付いての既知の集合，そして事象の結果または反響の予測である。その中でも，中心的な役割を果たすのは，事象の表現である。ＣＤ理論は，事象の一つの核を記述する簡単な構造である。言語で表現されている事象は，同じものを表している場合でも，様々な形態を取ることができる。この曖昧さを吸収するために，事象を記述しているＣＤの形式は常に同じものとする。そのために，ＣＤ理論は，単語をその背後にある概念とそれらの関係で表現する，一種の意味ネットワークの形をとっている。したがって，文の中に含まれている情報をはっきり示し，また意味的に同じ文を同じ形で表現するために，プリミティブズ（primitives；意味的元素）が使われる。このプリミティブズによって表される概念は，自然言語の体系で現れる静的概念と動的概念とに対応して，動的なものと静的なもの（動詞概念と名詞概念）とにわけられる。動的な概念は，

actor　　；行為者
action　　；行為
object　　；対象物
direction　；方向
instrument　；手段
recipient　；受動者

という要素を結合して表現し，静的な概念は，

object　　；対象物
state　　；状態

という要素を結合して表現する。そして，これらの概念に現れるカテゴリーとして次のものをあげている。

PP　(picture producer)；具象物（物体，実体）
ACT　　；行為
LOC　(location)　　；位置
T　　(time)　　；時間
AA　(action aiders)　；行為に対する修飾
PA　(picture aiders)　；具象物に対する修飾

ＣＤの構成は以下のように表される。

どのEVENTも，

一つのACTOR，
そのACTORによって遂行される一つのACTION，
そのACTIONが遂行する一つのOBJECT，
そのACTIONが向けられる一つのDIRECTION

を持つ。

Lispの表現を借りれば，

```
(EVENT (ACTION (ACTOR ?)
               (OBJECT ?)
               (FROM ?)
               (TO ?)))        *  ?はスロットを表す。
```

と表現できる。CDを使うときには，表層の文には記述されていないが，たぶん存在するに違いない行為者や対象物などがあればそれらを仮定する。仮定された（まだわかっていない）EVENTの要素がある場合，解析システムはこれを以前に解析した事実によって置き換えたり，これから解析しようとする事実によって埋めようとする（slot-filling）。これはおもに，人間の予測能力を表現するために用いる。CD表現が，言い替え文に対して有効である点については，次の二つの例文を見ることでその有用性がわかるであろう。

John bought a book from Mary.

Mary sold a book to John.

この二つの文は，わずかに意味は違うかも知れないが，CDによる意味表現上の要素が，文脈にかかわらず少なくとも存在する（Fig.2.1.2）。

ACTIONを記述するには，人間が用いる単語をそのまま利用していたのでは，きわめて数が多くなるため，行為に対して意味的に基本要素となるものを抽出して使用するほうが効率がよい。そのために，Schankらは基本的行為を定めた[5)6)]。彼らの示した基本的行為（primitive actions）は11個あるが，この数については対象とする問題やどこまで詳細な行為の記述を目標とするかによって多少の増減はあるものと思われる。物理的な世界において表現されるべき物事が，きわめて少数の行為を複雑に結合することによって記述できるため，基本的行為を定義することができる。また，多くの自然言語の動詞は行為に直接言及していないため，それをそのまま知識表現の核とすることが難しいため，基本的行為の設定は重要である。

6つの基本的行為

ATRANS	所有権，コントロールなどの抽象的関係の移動を示す。
MTRANS	情報の伝達，移動を示す。記憶内の情報の移動も示す。
PTRANS	対象物の物理的位置の移動を示す。
PROPEL	対象物の物理的な力を加えることを示す。
MBUILD	古い情報から新しい情報を作り出すことを示す。
INGEST	対象物を自分の体内に取り入れることを示す。

手段的行為

ATTEND	注意を向けることを示す。
EXPEL	対象物を自分の体内から排出することを示す。
GRASP	対象物を握ることを示す。
MOVE	動物が自分の体の一部を動かすことを示す。
SPEAK	音声を発することを示す。

ここで手段的行為と名付けてあるものは，CD間の依存性に関連したactionの類別であ

る。あるＣＤが他のＣＤに依存するには，因果関係による場合と，actionが手段を述べるＣＤとして依存するという二つの場合がある。手段的行為とは，後者に対応している。ＣＤ理論では，上に述べた基本的行為を基にして，動的概念と静的概念をある構文ルールや因果タイプを用いて結合し，事象の表現を形作る。これらの詳細な定義に関しては，参考文献に述べてある。

１１個の基本的行為を用いて表現できる事象の範囲そして文脈は，限りのあるものである。しかし，そのような状況は，事象の複雑さのもととなっている対象物の個別性に基づき詳細な情報を対象物に与えること，事象表現としてのＣＤを基本構造とした複雑な因果関係の型を用意すること，そしてＣＤを動機づける原因に関しての理論を用いることで回避することが可能である。

(4) 人間動作シミュレーションにおけるＣＤ理論の役割

ＣＤ表現を用いて意味表現を行うことによって次のような利点が得られる。まず，様々な処理プログラムの間での情報交換をするための“言語”として利用できる点にある。計算機処理を前提とするためには，このような情報の標準化が不可欠である。

また，基本的行為や行為者，対象物スロットを知識ベースへアクセスするキーワードとして利用することができ，処理プログラムをデータベース駆動型とすることができる。データベース駆動形のプログラムの利点は，プログラムの修正において関数に修正を施すのでなく，データベースの属性を変更することとなり，処理過程への影響が少なくなる点にある。ＣＤ表現された事象が，単に事象としての意味をプログラムに反映するだけでなく，事象に含まれる様々な情報が記号としてプログラムへ働きかけることもできるのである。

2.1.3 文脈の効果

ＣＤ表現を用いることで，事象や状態の間の因果関係を表現することができた。それは，人間の認知過程における再認を記述するのに都合のよい形式を持っている。再認をする際に知識として突き合わせるべき型をどの様にして適切に抽出するのかが次の問題となる。我々はこの過程を，文脈情報（因果関係，達成目標，現在進行している動作計画）を用いることで解決しようと考えている。特に，日本語のような文脈や外界の世界に強く依存する言語を対象とした場合，伝統的な統語解析ではもの足りない[10]。文脈情報には，言語の世界，外界，心の世界という三つの世界の間の相互作用が含まれる。そこで文脈の理論では，①事象や状態間の因果関係，②事象が引き起こされた原因あるいは行為者の意図といったものを考えなければならない。計算機処理をする際には，上に述べた事柄を知識ベースとして蓄え，利用していくこととなる。そこで，文脈を扱う際に必要となる知識について考察し，それをもとに知識ベースの構造をさぐる。

(1) 因果関係の計算（行為に付随する知識）

事象や状態間の因果関係には，次の四つの形式がある（Table 2.1.2）[5)6)]。

① 結果の因果関係（result）

事象は，その事象内に含まれるobjectの状態が変化した結果を持つことができる。

② 可能の因果関係（Enable）

状態の変化が起こると，以前にはなかった事象が起こり得るように世の中の条件が変わることがある。これらの状態変化は，潜在的な事象しか可能にしない。すなわち，事象を直接引き起こすのではなく，その潜在的な発生を可能にするに過ぎない。

③ 開始の因果関係（Initiate）

事象や状態の変化が起こったり，ある状態や潜在的な事象が存在した場合には，常にactorがそのことを知らされたり，あるいはそのことについて考えさせられたりする可能性がある。開始の因果関係は，物事に対する人間の思考を取り扱う。

④ 理由の因果関係（Reason）

人間が物事について考え始めると，何かをやろうと決心することが多い（MBUILD）。何かをやろうと決心するのは，それをする理由になる。

これらの因果関係は，一般の文章の中に頻繁に現れ，その文が意味をなしているかどうかといった判断の基準を与える。ＣＤ理論を基にするとこれらの因果関係は効率よく記述することができる。特にある事象が起こったときの“結果”に関する因果関係において，その知識を有限個の基本的行為に基づいて索引付けでき，知識ベースへのアクセスパスとして利用できる。

(2) 例示活動：スクリプト理論（具象物に付随する知識）

私たちが抽象的な概念を理解しようとするとき，自分の頭の中で，それは，例えばどういうだろう”と考えてみるものである。このような認知過程は，例示活動（instantiation）と呼ばれる[7]。例示活動の研究は，スキーマのデフォルト活動や，与えられた文脈に対して適切に対応づく“例”を思い浮かべることに向けられている。特に，人間が文を理解するときに自然に例示活動を行っており，元の文の文句よりも，暗黙のうちに自ら例示したものの方を記憶している点が重要である。

上述の例示活動によって人間は自分の理解を深めるのであるが，理解の妥当性を議論するためには例示によって得られる“適切な例”を問題にしなければならない。スキーマのレベルでは，各スロットを埋める際に種々の制約が存在するはずであり，その制約によって妥当性が検証できる。またスキーマ間の相互活動により一種の文脈が構成され，それによって文脈としての妥当性が検証される必要がある。これは，スキーマのslot fillingや例示活動と共に，提示された例の相互矛盾のチェック，観点の変更，事例の生成などの内的吟味活動によって得られる。このような活動を行うには，一つの事象を対象としたスキーマではなく，日常生活の出来事や諸活動のスキーマ（すなわちスキーマを構造化するスキーマ）が必要である。ＣＤ理論を基にしたこのような問題を取り扱うのがスクリプト理論である[5)6)]。スクリプトでは，人間の諸活動の一般常識を概念的な相互依存関係で記述し，文章理解に於ける“隠された前提”を表現する。

原子炉内作業や様々な作業に着目したばあい，特定の装置に付随して，作業の流れを表すスクリプトを記述することができる。そのようなスクリプトでは，ある具象物があって行為を作用させようとしたときに，どの様な系列で事象が発生するのかが記述される。そのため，スクリプトに関連した知識として，具象物をキー，行為を属性として与えること

で知識ベースを構築することができる。

(3) ゴールとプラン（意図に関する知識）

スクリプト理論では，ある状況に置かれたときになすべき事象を詳細に記述することができる。しかし，なぜその行為が必要であるのか，何のために行うのかといった理由については述べていない。そこで，事象が引き起こされた原因あるいは行為者の意図についての理論が必要である。

一般的に人間が何かをやろうと陽に考える場合，そこには達成目標（ゴール）が存在する。ゴールが存在し，それを解決しようという意図がトップレベルで存在する状況では，行為者は，そのゴールを達成するべくプランを立て実行を試みる。物語に代表される文章では，登場人物に何らかのゴールがあり，それを解決しながら物語が進展していく。つまり文章には，ゴールとゴールによって作られたプランを反映する文脈が存在するのである。そこで，言語処理プログラムにゴール，プランに基づく推論機能を付け加えることで，言語世界において登場人物が使ったプランニングの過程を再構築することが可能であり，このことによって現在何が進行しているかという意味，すなわち登場人物が行う行動の意図を見いだすことが可能となる。R.C.Schankらの示したゴール，プランに関する理論は，このような文脈を解釈するために作られた[5)6)]。

原子炉建屋内で模擬人間が行動する場合も，上述のゴール，プランによる意味付けが可能となると考えられる。ここで例として次のような命令が与えられたとしよう。

“レンチを持ってパイプbへ行け。“

この例文の場合，明示的に示された模擬人間に対するゴールとは，レンチを持つことと，自分がいる場所をパイプbにしなさいというものだけである。パイプbに行った後に何をするかは明らかにされていない。そこでこの段階で模擬人間が持っているゴールとは次のように記述される。

```
((dcont robot wrench)
 (dprox robot robot pipe-b))
```

dcont，dproxはそれぞれ，物の所有に対するゴール，場所の移動に対するゴールとなっている。これらのゴールは，きわめて基本的なものである。Schankらの示した基本的なゴールは，dcont，dprox，dknowである。

(dcont actor object)	対象物の制御権を得ること
(dprox actor object where)	対象物をどこかへ移動させること
(dknow actor infomation)	情報を得ること

これらのD－ゴール（局所的なゴール）は，状態の変化を考えた場合きわめて基本的な要素となっている。次節で述べるマイクロtale-spinプログラムでは状態を記述するCD表現として，

has，is-at，mloc，state，relation，who-has，where-is

のみが与えられている[5)]。そこでは，dcontはhasの属性を変化，dproxはis-atの属性を，dknowはmlocの属性を変化させる役割を持つ。

プランとは，ゴールを達成するために起動される知識である。例えば，上に述べたロボ

ットのdproxゴールを達成するためには以下の事柄をおこなう必要がある。

ゴール：(dprox robot robot pipe-b)

① パイプbはどこにあるのかを知る

サブゴール：(dknow robot (where-is pipe-b)

② 自分が知らない場合，誰かに問い合わせる

プラン：(ask-plan)

③ パイプbの場所がわかったら，自分のいる場所と突き合わせる

サブゴール：(is-prox robot (loc-name-of pipe-b)

④ パイプbの場所が，自分のいる場所でなかったら，そこへ移動する

行為：(ptrans (actor robot)
(object robot)
(from (loc-name-of robot)
(to (loc-name-of pipe-b))

⑤ 行為が成功したら，ゴールは達成される

この例に示した①から⑤までの一連の行動がプランである。プランには，さまざまな条件判断を含むことができる。ここで強調しなければならないのは，プランを立てる時点では，基本的行為を用いて一般的に記述されるが，それを評価する時点では，対象物の属性として与えられているスクリプト記述が評価されるということである。スクリプト記述を評価する時点で新しいゴールが見つけられた場合（例えばレンチが急に必要となったとき），新しいプランを形成することができる。

(4) 人間動作シミュレーションへの応用（知識ベース）

人間動作シミュレーションでは，模擬人間が人間の与えた命令（きっかけ）を基にどのような行動を引き起こすのかをシミュレートする。このような問題を取り扱う場合，上に述べた文脈理解の枠組みは強力な役割を果たすと考えられる（Fig. 2.1.3）。

スクリプト，ゴール，プランといった理論を基にHASPにおける自然言語解析部分を構成するためには，対象とする作業内容にどのようなゴール，テーマがあるのかを抽出し，そして具象物に付随するスクリプトを明らかにすることによって，シミュレータを作らなければならない。そのため，知識ベースの構造とは，以下のようになる。

① 階層化されたゴールの設定

② ゴール評価関数として，プランニング知識を与える

③ 事象，状態間の因果関係を行為，状態記述関数に関して索引付けする

④ 具象物をキーとして行為をロボット動作言語を用いて記述する

次節では，①，②，③を具体的にプログラミングした例を説明する。

2.1.4 マイクロtale-spinプログラム

(1) マイクロtale-spinプログラムの整備

前節で述べた文脈効果を実際にシミュレートするため，CD，plan，goalを基礎理論に持つ簡単な物語生成プログラムを整備し，その動作を解析することとした。我々が対象

とした物語生成プログラムはR.C.Schank，j.Meehanらが示したもので，マイクロtale-spinと呼ばれるものである[5]。我々は，このプログラムをパーソナルコンピュータ上で稼働するcommon lisp上に移植し，その動作を解析する中で，どの様な問題点があるのかを調査した。マイクロtale-spinは，以下のようなコンポーネントを持っている。

① goal evaluator and definition of goal and plan
② simulator
③ memory access function and memory management system
④ initial world data base
⑤ English generation component
⑥ CD generation component

このプログラムの動作の概要は次のようなものである（Fig.2.1.4）。まず，登場人物に解決すべきゴールが人間によって与えられ，その事実をプログラムが認識し，事実データベースに保存される。そして，ゴールに対応するプランが評価され，そこに付随するサブゴールが次々と評価されることによって行動列が生成され，英語に変換される。最終的にゴールが満足された場合には，そこで生成過程は終了し，同一のゴールが見つけられた場合には，満足な結果を得られないまま終了する。プログラムの動作例を付録Aに示す。これらの処理は，2.1.3節で示した結果と可能の因果関係にもとづいている。処理プログラムは，シミュレータが生成した行動，状態を評価するときに，基本的行為，状態記述関数をキーワードとして知識ベースに蓄えられた結果，反応，可能の因果関係を計算する。

因果関係に関する知識へのアクセス

① (get (header cd) 'conseq) ：結果を起動する
② (get (header cd) 'react) ：反応を起動する
③ (get (header cd) 'demons) ：可能な事象を起動する

＊ここで(header cd)は，CD表現された事象，状態のACTION部分を取り出す関数である。

(2) 人間動作シミュレーションへの応用

1) 処理過程の修正（今後の課題）

マイクロtale-spinが生成する物語の性質は，goal，planの定義と動作シミュレーションの定義によって変化させることが可能である。HASPへの適用を考える際には，この要素を修正，拡張していく必要がある。

マイクロtale-spinプログラムが生成する物語の性格を決定しているのは，達成目標がどの様に基本的行為の因果関係を形作るかという部分である（シミュレータ）。例えば，“熊のジョー“が蜂蜜のある楡の木のそばにきたとき，もしお腹がすいていればそれを食べようとする。この例のように，ある状態に対して行為者の持っているトップレベルのゴールが働きかけるために示す反応が，原子炉内作業においても本質的な行動生成の役割を果たすと考えられる。そこで，人間動作シミュレーションにおいて，模擬人間の自然文理解に対しゴール，プランの理論を適応するためには，点検すべき装置の

属性に作業内容を記述し，模擬人間が利用できるようにする。

本節で示したプログラムでは，解決すべきゴールが，“空腹，喉の渇き“のみを提供しているので，この部分も原子炉内作業用に書き換える必要がある。この部分は，プログラムがデータベース駆動形になっているので容易に拡張可能である。また，dcont，dknow，dporxといった基本的なゴールを基に，作業におけるゴール（例えば，こういう操作をしなさいといったもの）を記述し，Dゴールに不足している部分を追加することが必要である。

また，登場人物が扱う具象物の属性とその空間的配置がきわめて簡単に示されているため，各種装置が複雑に入り組んだプラントに対しては，そのままでは効率的な対応ができない。特に，登場人物が場所の移動をする（dproxを評価する）際に，階層的な空間配置表現が必要となる。そのため今後自然言語理解システムを整備していく中で，“旅行エージェント（どこかへ旅行するとき，種々の制約に基づいて，詳細な旅行行程を計画するシステム）“的なプログラムを構築する必要がある。

2）自然言語処理フロントエンドプロセッサとの整合性

本節で紹介された物語生成プログラムは，Common Lisp言語によって書き直されている。HASPで利用を計画している自然言語処理プログラムCS-PARSERもCommon Lisp言語によって記述されている（2.2節参照）。そのため，両者を結合させて自然言語理解部分を構成することは，プログラミング言語からみた移植性を考慮して容易であると思われる。また，辞書データベースが意味的原素を用いて階層的に構成していこうと考えているので，問題解決過程における具象物，基本的行為の属性の引出しが比較的容易ではないかと考えられる。

参考文献

1）石綿，機械翻訳の歩み，言語，17.1，34（1988）

2）郡司，言語学と機械翻訳，言語，17.1，40（1988）

3）上中，神林，核データ評価コードに関する知識構造の調査，JAERI-M 88-143（1988）

4）高木，伊東，自然言語の処理，丸善（1987）

5）R.C.シャンク，C.K.リーズベック編，石崎 監訳，自然言語処理入門，総研出版（1986）

6）R.C.Schank and R.P.Abelson, “Scripts, plans, goals, and understanding.”, Lawrence Erlbaum Associates, Hillsdale, N.J.（1977）

7）安西，佐伯，難波，LISPで学ぶ認知心理学2，東大出版協会（1982）

8）細川，日本語の論理，言語，17.5，62（1988）

9）国広，語義の領域，言語，17.5，40（1988）

10）堀，石崎，文脈理解とAI，人工知能学会誌，3.3，312（1988）

2.2 日本語解析プログラム

本節では，ＨＡＳＰ研究の一部である自然言語処理において昭和６３年度より導入された日本語解析プログラムの利用に関して述べる。

2.2.1 日本語解析プログラムの導入

原研・計算センターでは，ＨＡＳＰ実現における自然言語処理プログラムとして，既存の機械翻訳ソフトウェアを基礎に機能を拡張して利用することを考えている。そのために昭和６２年度は科学技術庁・機械翻訳プロジェクトが開発した科学技術文献の日英翻訳システム（文法記述言語GRAmmer DEscriberの名をとって以下では単にGRADEと表現する）[1]を導入した。しかし，実際に作業を行ない，将来的な拡張性及び機能性等を追求していく上で，やや難があるとの実感を持つに至った[2]。そして検討の結果，ＨＡＳＰを構成する自然言語処理プログラムは少なくとも次の４点を満たしていることが望ましいと考えられた。

① 拡張時の問題として，導入と共に原研がソースプログラムを入手できること。

② 機械翻訳を行なう必要はなく，ＨＡＳＰにとって不要なサブシステムを容易に切り離せる構造になっていること。

③ ＨＡＳＰ全体のサブシステムとして利用するため，その規模は小さいことが望ましく，原研独自のアイデアを反映していくためにハンドリングがしやすいこと。

④ 日本語入力文に対して可能な限り柔軟な適応能力を持っていることが望ましく，構文的に品詞の並び方を規定していく方式よりも，むしろ２単語間の関係（係り受け）を捉えていく方式を得意としていること。

①については商用の機械翻訳ソフトウエアでは困難な条件であるといえる。筆者らの知るところでは，商用化されているシステムは，処理速度・ノウハウの漏洩等を考慮してオブジェクト形式で提供されているものである。それゆえ，機械翻訳ソフトウエアを導入するのであれば，一定のメーカーに偏らず研究用に作られたものに限られることになる。それに該当するのがGRADEである。②についてはHASPが利用しようとしている技術が機械翻訳に限定されるものではなくて，日本語入力を取り込んで咀嚼（そしゃく）することにあるということを示している。ＧＲＡＤＥもその点では，日本語形態素解析プログラム・文法規則トランスレータ・英語形態素生成プログラム等のモジュールに分かれており，基本的には分割可能となっている。ただし，辞書を（品詞情報以外で）分割するのは事実上困難である。③，④は特にＧＲＡＤＥの利用検討を進める上での反省に基づいたものである。ちなみにＧＲＡＤＥは大型計算機・富士通Ｍ－７８０上で稼働したが，正常に機能させるためには処理プログラムが約８MB，辞書用ＶＳＡＭファイルが約１３０MBの資源量を要する大規模ソフトウエアだった（日英システムのみを積算）[3]。このことは，ＨＡＳＰプロジェクトが単独で大型計算機システムを所有している条件であればさほどの問題はないだろうが，計算機システムは原研内で共用のため，特定ユーザーの過度の利用が他の計算機ユーザーに迷惑をかけてしまうことになる（例えば，辞書ファイルの更新でＣＰＵの負担が大きくなる。ＶＳＡＭファイルが大きくなることによってＴＳＳ用ディスクシステムのワーク領域が食いつぶされてしまう。等々）。また，ＧＲＡＤＥはあく

までも日英（＋英日）２カ国語間の機械翻訳を目的としており，その方法として構文情報（木構造）の変換ルールを用いている。確かに構文の変換を用いた機械翻訳は，文型，語順が比較的しっかりしている欧米言語間では有効と考えられる。しかし，日本語は語順についての制約がかなり緩く，文型により文法を規定していくよりも，むしろ助詞を中心とした係り受け関係からの文の構成を捉えているように思われる。

以上の４点を考慮した上で，計算センターでは，昭和６３年度新たに日本語解析プログラム（㈱CSK・CS－PARSER）を導入した。先に掲げた条件に対し，CS－PARSERは次の点が評価されたからである。

① 原研がソースプログラムを入手することが可能である。

② あくまでも日本語入力インターフェイスを意識しており，２カ国語間の機械翻訳に縛られずに前置処理として利用することができる。

③ 基本的なプログラムの規模としてはGRADEと大差ない（約7MB）が，辞書ファイルが無いので全体的な規模としてはＧＲＡＤＥよりもはるかに小さい（基本的な辞書の一例はあるが，利用対象，形態によって利用者側が辞書のあり方を設計していくことになる。）。また，稼働環境がワークステーション（Ｓｕｎマイクロシステムズ・Ｓｕｎ3／２６０）上の設定になっているため，同じ規模のプログラムでも大型計算機上のそれとは比較にならないほど使い勝手がよいといえる。

④ 係り受け規則を用いて解析を進める方式を採用している。

次項以降では，日本語解析プログラムCS－PARSERの導入に関連した昭和63年度作業について記述する。

2.2.2 スケジュールと基本方針

昭和６３年度は，前項で紹介したCS－PARSERを用い，模擬的な点検作業指示の日本語文を入力として与えて，一定の解釈を行なわせるための処理及び辞書データの検討を行なっている。年間の作業スケジュールはTable 2.2.1のとおりである。昭和６３年度の柱となっているのは，表中のａ（CS－PARSERの導入）とｅ（CS－PARSERを用いたデモシステムの試作）である。また，ｂ～ｄはデモシステム試作の準備段階にあたり，昭和６４年度以降のシステムの拡張を含めたデータ作成の基礎と位置づけている。

試作していくシステムの最終的な目標は，人間（日本人）の言葉を人間らしく解釈して行動を起こす過程をコンピュータ上に再現させたいということである。しかし，この通り目標が壮大なものであるため，それを長期的に実現していくため，短期間の目標を段階的に設定して，各段階で得られた経験・技術を蓄積し，統合していかなければならない。昭和６２年度は壮大な目標まで一気に駆け登ることに山を向けすぎたあまり，一般的な知識をどう表現するかということが気になり，かえって壁となってしまった。昭和６３年度からは，先に述べた軽量の日本語解析プログラムCS－PARSERを導入するが，これにあたり関連作業の段階的な目標設定とCS－PARSERの拡張プランを明確にしていくことを重視し，進展を図っていく。以下では，最終目標に近づくために何をどうしなければならないと考えているかを説明する。

(1) 対象分野の限定

ＨＡＳＰでは，最終目標の対象分野の制限として原子炉建屋内点検・保守作業を掲げている。しかし実際問題として，これだけで明確な対象の制限がされているとは認め難く，さらに，質・量として実現目標を具体的に掲示しなければコンピュータシステム化まで到達することはできないと考える。そこで本研究においては，予め対象とする基本文（言い回しは含めない）をTable 2.2.2のような形で有限個数用意し，それらを意図どおり処理できることを当面の最終目的とした。ここで言い回しとは，同一の表現材料に対して表現方法を変化させたものである（例えば「接地線の変色」を「接地線の色が変わっている」，「接地線の色の変化」などと表現すること。）。基本文の内容は①計器室内計器盤点検，②バルブ保守，③手動仕切弁保守の各作業の模擬的な手順を示している。そして，人間側が意図したとおりに文を処理（これを理解と捉える）させるために必要な知識データと，その処理方法を各々検討し，そのメカニズムを模擬するシステムを試作していこうという方針である。このような方針に基づき，昭和６３年度の目標として，まずは①の計器室内計器盤点検作業（基本文数２５）を対象とすることにした。ここで対象にした基本文は，原研内の大型試験装置の運転報告[4]から得た点検項目に関する情報を，現在改造中の原研・研究炉ＪＲＲ−３（Japan Research Reactor−3）の計器室を想定して全て模擬的に作りなおしたものである（②，③については文献[5]で紹介されている実例をほとんど引用した。）

(2) 処理イメージの決定

対象領域を限定し，入力文の作成を行なったところで，（当然の事ながら）試作システムの全体像を具体的に青写真に描いておく必要がある。そのためには，入力文を限定したのと同時に，出力としてどのような表現をシステムが返せば良いのかを充分に検討して入出力仕様を明確に定義することが重要である。このことはＨＡＳＰの他サブシステムとのリンクを考えた上でも最優先すべき問題の一つであると考えられる。試作システムは，命令を表現している入力文を解釈し，複数の基本述語の組み合せにまで展開して外部に引き渡すことができなければならない。模擬人間の詳細な行動を表現する部分はＨＡＳＰにおいては別のサブシステムが受け持つことになるが，試作システムの出力行動列を構成する基本述語は，ＨＡＳＰ全体から見て理にかなったものでなければならない。

Schank等は一連の言語理解研究の中でＣＤ（Conceptual Dependency：概念依存性）を中心に位置づけている[6]。ＣＤでは１１個の基本的行為（Table 2.2.3）が用いられており，彼らはそれらの基本的行為の結合によって表層の動詞（英語）を説明できることを提示している。彼らの対象としているテーマの中心は物語等の理解にあるため，用意された基本的行為をＨＡＳＰのような作業環境にそのまま置き換えてよいかどうかを充分に検討するべきではあるが，基本的行為の概念設定はＨＡＳＰ研究にとって非常に興味深いものである。

また一方では，古くから人間工学・経営工学に関連して様々な作業研究が行なわれているが，Gilbrethは人間の動作中に１８個の基本的動作要素が存在していることを発見し，それをサーブリック（Therblig）と呼んだ（Table 2.2.4）[7][8]。サーブリックはHASPが対象としているような体全体の移動などは考慮されていないものの，（作業台上の）手

作業を中心に目の動きなどを含めて分析されたものであり，参考にできる部分が多いように思われる。

以上述べた「基本的要素」を参考にして，システムの処理イメージを加味した基本動作述語を決定する。これは，対象が限定されるという意味で特殊な利用環境下ではあるが，システムに人間らしい振る舞いをさせるための一知識を表現する要素の提示にあたるものと考えられる。Fig．2.2.1に自然言語レベルで考えた処理イメージの一例を示す。また，同じ処理イメージに対して，サーブリック及びサーブリックもどきの基本的動作要素（Table 2.2.5）を想定したものをFig．2.2.2に示す。

(3) 概念体系及び処理方法の検討

入出力の仕様を充分に検討し，固定した段階で，入力と出力を結びつける仕組みを詳細に分析していく必要がある。これは，入力した文をどういった構造で捉え，環境の情報（装置や模擬人間の持つ属性，並びに対象世界で普遍的な情報など）とどのように照らし合わせるか，そしてどのような情報を新たに派生させ（想起），要求された出力（対話の返答や動作，また環境情報への働きかけなど）へ展開するかまでの詳細を分析していくことにあたる。分析作業の過程の一例をFig．2.2.3に示す。Fig．2.2.3(a)は単語もしくはそれに準ずる概念が命令「計器盤の扉を開ける。」を処理するために持っている知識的データとその体系である。そしてFig．2.2.3(b)の上半分は同じく具体的に対象モデル内に存在する物（インスタンス）のデータであり，下半分がデータを加味した上での処理イメージである。

自然言語処理のみならず，エキスパートシステムなどの分野においても，常識的な判断過程（処理）およびそこに用いられるデータ（知識データ）の表現は重要な意味あいを持っている。高木らは一連の自然言語処理研究の中で数学や化学の文章題を解くシステムの試作を行なっている[9)10)]。彼らは文章題を解くにあたってのプロセスとして，①問題文の小モデル化，②変量間の関係からの問題解決方法の組立，③方程式への展開と求解，の3つのステップを考え，それに用いるための（常識的な知識を含めた）知識をフレームの形で表現している（Fig．2.2.4）。彼らのシステムが対象としている問題領域とHASPが対象とするそれとは直接的には関係がないようであるが，概念階層作成の方法と概念の処理への利用について参考となる部分が多い。また溝口らはエキスパートシステムでよく引き合いに出される表層の経験則（ヒューリスティックス：heuristics）すなわち“浅い知識”をシステム内において自己生成できるような，より本質的な知識（“深い知識”）の利用について報告している[11)]。それと同様に，HASPの目指す自然言語処理も表層の変換知識を抽出することがテーマではなく，それらを順々に生成させることが出来る“より深い知識”（これは“浅い”との相対的関係である）データの表現とその処理方法を提案していくことにある。Fig．2.2.3に示した処理イメージについては，インスタンスに応じたメッセージパッシングを利用した出力行動列の自動生成を意識したものである（図の例は，動詞概念「開ける」を中心として，対象インスタンスへのメッセージパッシングを用いて目標の動作を順次生成させるためのデータを示している。）。

(4) CS－PARSERへの反映

(3)で提起された処理の流れをCS－PARSERに反映させなければならないが，CS－PARSERとの整合性を充分に考慮した上で処理プログラムの追加方法を決定していく。また同時に，机上で考えられた際に処理の流れとともに図的に表現されたデータの構造を，コンピュータシステム上にどのように表現していくかということは重要な課題である。

(3)で行なった分析（Fig．2.2.3）は，見てわかるように，実際にはかなりフレーム的な表現を意識している。また，前にも述べたように，データ構造内に簡単な手続きを記述したスロットを置き，メッセージパッシングによって全体の手続き（行動命令列）を発生することはできないかという考えも合わせて表現したつもりである。

CS－PARSERはLISP言語（KCL：Kyoto Common Lisp）環境下で稼働している。それゆえフレーム表現をリスト構造に置き換えながらシステムの試作を行なっていくには適した環境といえるのではないかと考える。また，辞書ならびに知識データの検索については，将来的には外部ファイルへのアクセスといった形式を予定している。しかし，データの規模が小さい昭和６３年度当初は，将来外部ファイルに移すことを意識しながら主記憶上に展開して，動作を確認することを中心に作業を行なう。

2.2.3 使用語彙の分類

自然言語処理を考える上で，語彙を分類し，その意味あいを考えることは必要不可欠である。はじめに述べたGRADEシステムでは，語彙の分類について，国立国語研究所が刊行している分類語彙表[12]を利用している。分類語彙表では使用頻度が高いとされる約７千語が意味分類されている。意味分類は“ごく一般的な”考え方に基づいて行なわれている。ところが，そこに分類されているデータだけから全ての表層の意味を導き出すのは並々ならぬことがあり，人間的に考えてみると時として奇妙な場合があるものである。

極端な例で申し訳ないが（決して先達の功績にケチをつけようということではない），分類語彙表では「カット」という単語の意味が２種類紹介されている。第１の「カット」は「刺し・切り・削り」であり，第２の「カット」は「美術」である（蛇足ではあるが，前者はナイフやハサミなどで切ったり刺したりすることを意味し，後者は挿し絵のことを意味している。）。ここで，もう１つの「カット」，すなわちテニス・卓球・野球などで用いられる第３の「カット」の存在を考えてみてほしい。乱暴な言い方をして「球を切る」などと使う人もいるにはいるが，この第３の「カット」は「球を切るようにして打つ」ということであり，「球をチョン切る」ことでも「球がキレる」ということでもないはずである。そして意味からわかるように，この「カット」というのは間違いなく第１の「カット」から派生しているものである。そうすると確かに第１の「カット」に対して，持っている道具（ナイフ，バット等）や，スポーツをしている状況なのかどうか，対象物が向かって来るボールなのか等の条件を考慮して第１の意味と第３の意味を判別させることも可能であり，一見そのような処理こそが自然な流れなのではないのかなと悦に入ってしまったりする。しかし筆者は実際問題として，ごく普通の人間がそのようなことまで毎度毎度考えていることをもっともらしく唱えるのはかえって不自然なのではないかと感じる。第３の「カット」が第１の「カット」の派生であるなどということは，

たいていの場合ほとんど重要ではなく，最初から第3の「カット」を「突き・当り・打ち」（分類語彙表の中に実在している分類名）と分類しておく方が現実的に思える。ただし，そのような語の意味的な背景を教えることが目的であるような状況においてはその限りではない。

少し説明が長くなってしまったが，ここで主張したいことは，既に一般化されている語彙分類の扱い方に関して，適用分野（対象）に対してどれだけ有効であるかを充分に検討すべきであるということである。一般化を行なう際に最終的に残しておかなければならない意味（分類ラベルと考えられる）というのは，実は対象があってこそ存在するわけであり，一般的な対象とされている考え方が，自分達のシステムの扱う対象とどのように異なり，また同じなのかということを認識しておかなければならない。対象領域の持つ特異性が意外な落し穴になる場合があるはずである。一般化されてうまく使えそうだという場合にも，やみくもに信じてしまわず，常に疑問を持つことも必要なのではないかと感じる。人間は環境に対する順応性が高く，一般的な意味分類を環境に応じたものに書き換えて保持しており，それを駆使することで人間的な解釈・判断を行なっている（人間らしさが存在する？）のではないかと感じる。

ＨＡＳＰにおいては前に述べたとおり，適用環境と入出力対応を分析して語彙の分類（意味づけ）を行なっていくことを考えている。そしてその過程で，意味分類としてごく一般的と思われる分類語彙表・日本語基本動詞辞書[13)14)]との適合度などを確認して，再度対象分野の特異性を理解した意味づけを考えていく。

2.2.4　問題点と今後の課題

以上，原研・計算センターが昭和６３年度より導入した日本語解析プログラム・ＣＳ－ＰＡＲ－ＳＥＲの利用方法についての基本的な構想を記した。筆者が至らないため，まだまだ検討を要する部分が多いのではないかとも思うが，Ｆｉｇ．2.2.3に示した処理構想を実現できるように，まず２５文中の５文程度をピックアップして小システムの試作を行なう。そして，小システムの動作テストを行なって，知識的なデータとして作成・蓄積すべきものが構想どおりであり，それを基にシステムを拡張していくことができるかどうかを評価していく予定である。

小システムの試作・評価を終えた段階で予定している次の作業（課題）は，今のところ（最小限という意味で）次の２項目である。

① 言い回し文を理解させる。

② 処理可能な文の数を増やす（少なくとも基本である２５文を最終的に全てクリアすること）。

①については，主に動詞の概念階層を細かく設定すること，およびそれに対応した名詞概念の強化を行なう。例えば，「計器盤の扉を開ける。」ということが「扉と計器盤本体の間を模擬人間の手が動きを制限されることなく移動できるように，扉に力を作用させて隙間を作る。」ということと同じ意味を持っていると理解するためには，「開ける」だけではなく，少なくとも「作用する」「押す」「引く」「移動する」などといった概念や実現動作の関係を常に把握できるような概念階層をなしていなければならない。

また②については，昭和６３年度の小システム試作および①で得たノウハウを生かして，概念体系ならびに処理フローを拡張していかなければならない。その際には，既に作ったシステ

ムおよびデータとの整合性が損なわれないように，データと処理フローの全体的な矛盾チェックと，その後の拡張までを考えた管理方法の確立が要求されるのではないかと考えている。

参考文献

1) 科学技術庁科学技術振興局編：日英科学技術文献の速報システムに関する研究成果報告書（1987）
2) 上中，神林：核データ評価コードに関する知識構造の調査，JAERI-M，88-143（1988）
3) 近藤，他：私信（1988）
4) 下村，他：私信（1988）
5) 原子力用バルブ編纂委員会編：原子力用バルブ，日本工業出版（1985）
6) Schank, Riesebeck，（訳）石崎：自然言語理解入門，総研出版（1986）
7) 長町：現代の人間工学，朝倉書店（1986）
8) 加藤：現場のＩＥ(II)－動作分析－，日科技連（1983）
9) 高木，伊東：自然言語の処理，丸善（1987）
10) 阿部，他：簡単な数学文章題を解くシステム，人工知能学会研究会資料，SIG-KBS-8801-5，pp.42-51（1988）
11) 溝口，他：浅い知識と深い知識，人工知能学会研究会資料，SIG-KBS-8801-13，pp.100-105（1988）
12) 国立国語研究所編：分類語彙表　第２６版，秀英出版（1987）
13) 情報処理振興事業協会・技術センター編：計算機用日本語基本動詞辞書ＩPAL（Basic Verbs）－解説編－（1987）
14) 情報処理振興事業協会・技術センター編：計算機用日本語基本動詞辞書ＩPAL（Bacic Verbs）－辞書編－（1987）

Table 2.1.1 Semantic elements for several verbs in semantic group "cutting" [9].

動詞	動作	対象物	道具	様態	結果
切る	切断				鋭い切断面
裂く		薄い物	両手刃物	引っ張る	細片，細長い切口
割る		固い物		打撃を与える	複数の破片
もぐ		本体につながっていながら，形の上で独立している小部分	手	引っ張る	
ちぎる		紙，花びら，パン餅等の柔らかい物			小片

Table 2.1.2 Four types of causation.

① 結果 (result)	② 可能 (Enable)
PP ⟺ ACT r ⇑ → STATE PP ⟸ → STATE PP ⟺ ACT r ⇑ PP ⟺ STATE	→ STATE PP ⟸ E ⇑ → STATE PP ⟺ ACT
③ 開始 (Initiate)	理由 (Reason)
→ STATE PP ⟸ I ⇑ → STATE PP ⟺ ACT	PP ⟺ STATE R ⇑ ACTOR ⟺ MBUILD

Table 2.2.1 Schedule of the fiscal year 1988.

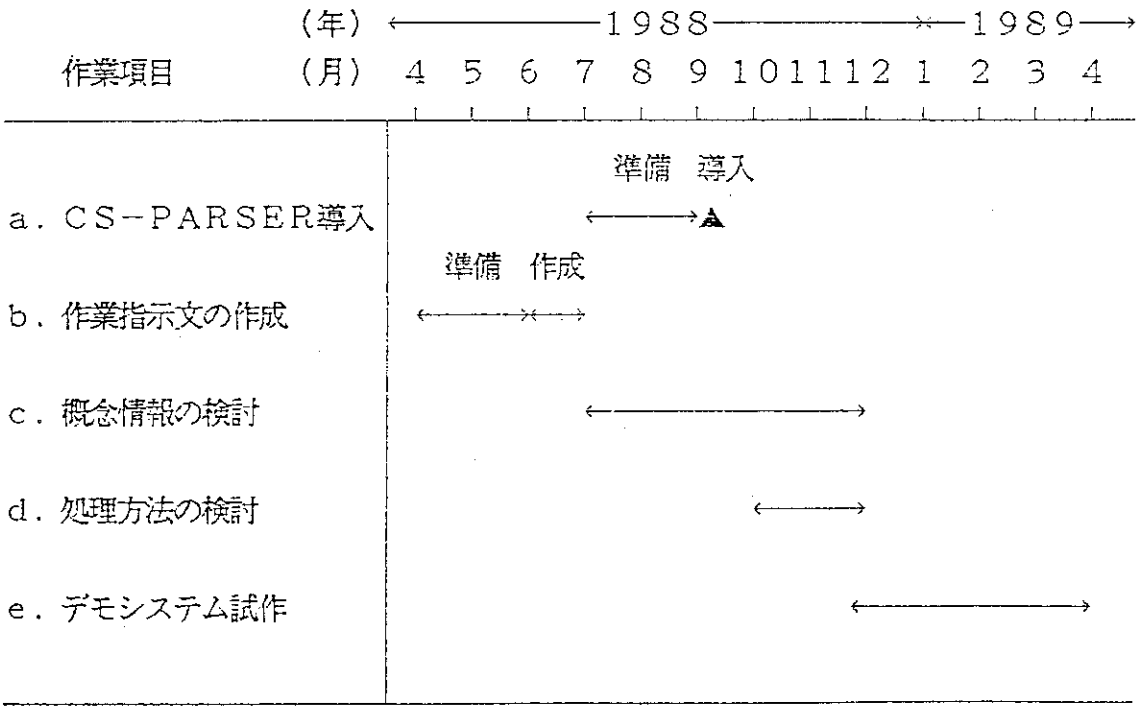


Table 2.2.2 Example for the treated sentences.

- 計器盤の扉を開ける。
- 表示灯およびヒューズの断線を点検し、不良品については交換する。
- 接地線の変色、損傷の有無を目視にて点検する。
- グランドパッキンの摩耗度合いを確認し、不具合が有れば取り替える。
- ボルト・ナットの劣化をチェックする。
- 弁座面等の荷重受け部の表面粗さをチェックする。
- 所要な取合部に、組立時に備えて合マークを入れる。
- ヒンジボルトの六角ナットを１／２程度ゆるめ、パッキンの締付け荷重を解消する。
- 弁体のシートに光明丹を薄く塗布し、弁座に圧着させてから再び弁体を取り出す。

Table 2.2.3 Primitive actions used in CD theory [6),9)].

① PROPEL 物体への物理的な力の作用
② MOVE 身体の一部の移動（所有者の意志による）
③ INGEST 動物が身体内に物体を取り込む動作
④ EXPEL 動物が身体内から物体を取り出す動作
⑤ GRASP 行為者による物体の把握
⑥ PTRANS 物理的な物体や場所の移動
⑦ ATRANS 抽象的な関係の移動
⑧ SPEAK 音を発生する行為
⑨ ATTEND 刺激に対して感覚器官をむける行為
⑩ MTRANS 動物内もしくは動物相互の情報の移動
⑪ MBUILD 古い情報をもとにした新しい情報の構築

Table 2.2.4 Thirbligs and the symbolic expressions[7),8)].

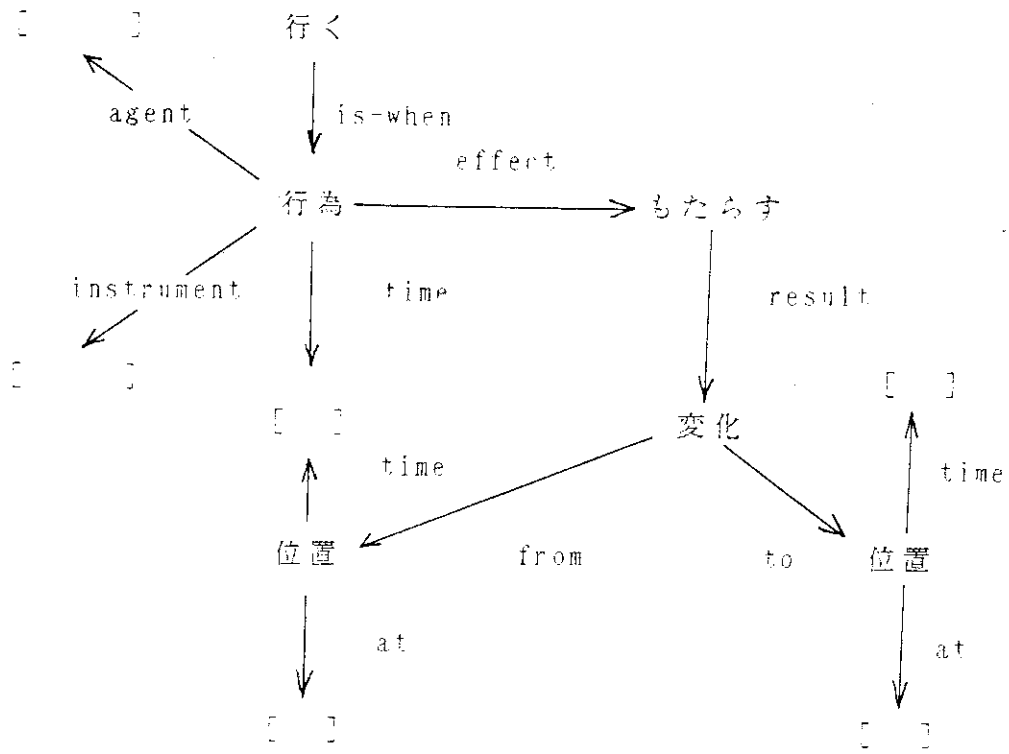
類別	基本要素	基本要素（英語）	略号
↑	① 空手移動する	Transport-Empty	TE
	② つかむ	Grasp	G
1	③ 荷重移動する	Transport-Loaded	TL
	④ 位置を正す	Position	P
	⑤ 組み合わす	Assemble	A
	⑥ 分解する	DisAssemble	DA
類	⑦ 使う	Use	U
	⑧ 手放す	Release Load	RL
↓	⑨ 調べる	Inspect	I
↑	⑩ 探す	SearcH	SH
2	⑪ 見い出す	Find	F
	⑫ 選ぶ	SelecT	ST
類	⑬ 考える	PlaN	PN
↓	⑭ 用意する	PrePosition	PP
↑	⑮ 保持する	Hold	H
3	⑯ 避けられない遅れ	Unavoidable Delay	UD
類	⑰ 避けられる遅れ	Avoidable Delay	AD
↓	⑱ 休む	Rest for overcoming fatigue	R

註）1類：作業を行なうにあたって必要な動作
2類：1類の動作を遅くする傾向のあるもの
3類：作業が進んでいない動作

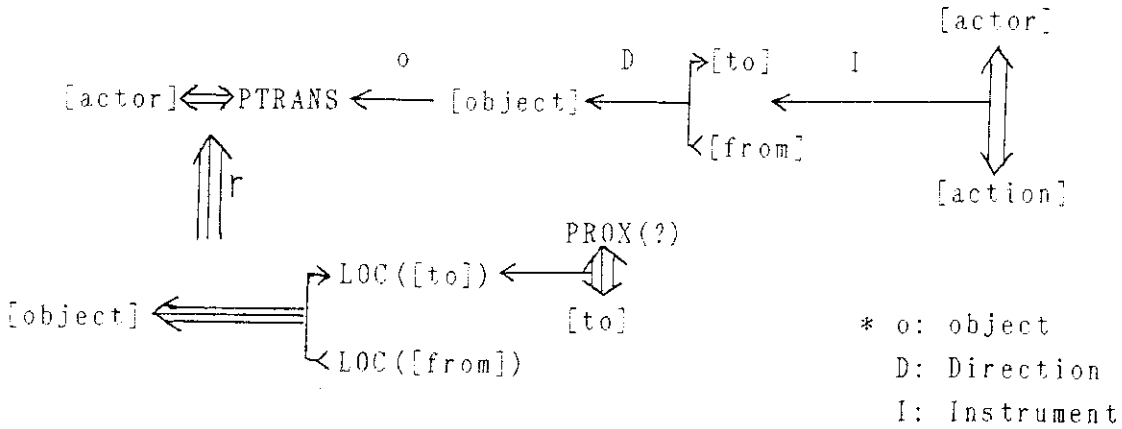
Table 2.2.5 Some basic acts like Thirbligs and the symbolic expressions.

・ 体を移動する	Move Body	MB
・ 体の位置を正す	Position Body	PB
・ 働きかける	InfLuence	IL
・ 基本姿勢をとる	ATtention	AT
・ 報告する	InforM	IM

a. スキーマ表現



b. CD表現



* [ ] はスロットを表す

Fig. 2.1.1 A typical schematic representation of verb "go" (a), and a representation for the same verb in conceptual dependency theory (b).

ジョンが先に本を欲しいと思っていたとき

```
(EVENT (CAUSED (ANTE    ( (ATRANS (ACTOR  JOHN)
                                  (OBJECT MONEY)
                                  (TO     MARY)
                                  (FROM   JOHN))    ))

                 (CONSEQ ( (ATRANS (ACTOR  MARY)
                                  (OBJECT BOOK)
                                  (TO     JOHN)
                                  (FROM   MARY))    )) ))
```

メアリが先に本を売りたいと思っていたとき

```
(EVENT (CAUSED (ANTE    ( (ATRANS (ACTOR  MARY)
                                  (OBJECT BOOK)
                                  (TO     JOHN)
                                  (FROM   MARY))    ))

                 (CONSEQ ( (ATRANS (ACTOR  JOHN)
                                  (OBJECT MONEY)
                                  (TO     MARY)
                                  (FROM   JOHN))    )) ))
```

* ATRANSとは，ものの所有権の移動を表す行為である

Fig. 2.1.2 An example for consistent elements of conceptual dependency theory.

命令："パイプbへ行け．"・・・・ロボットは，パイプbへ行きたいと思う．

```
goal;    (dprox robot robot pipe-b)              "パイプbへ行きたい"

plan;    sub-goal

         (dknow robot (where-is pipe-b))         "パイプbのある場所は？"
         (is-prox actor (loc-name-of pipe-b))    "自分は近くにいるのか？"

         (doit  (ptrans (actor robot)
                        (object robot)
                        (from (loc-name-of robot))
                        (to (loc-name-of robot))  ))

causal relation;

         result

         (is-at robot (loc-name-of pipe-b)) "パイプbの近くにいる"

         reaction

         (eval (get 'pipe-b 'operations)) : "次の動作をしたい"

script;(eval (get 'robot 'ptrans))
       ; (ロボット動作記述言語群)
```

Fig. 2.1.3 A context effect for a simple instruction.

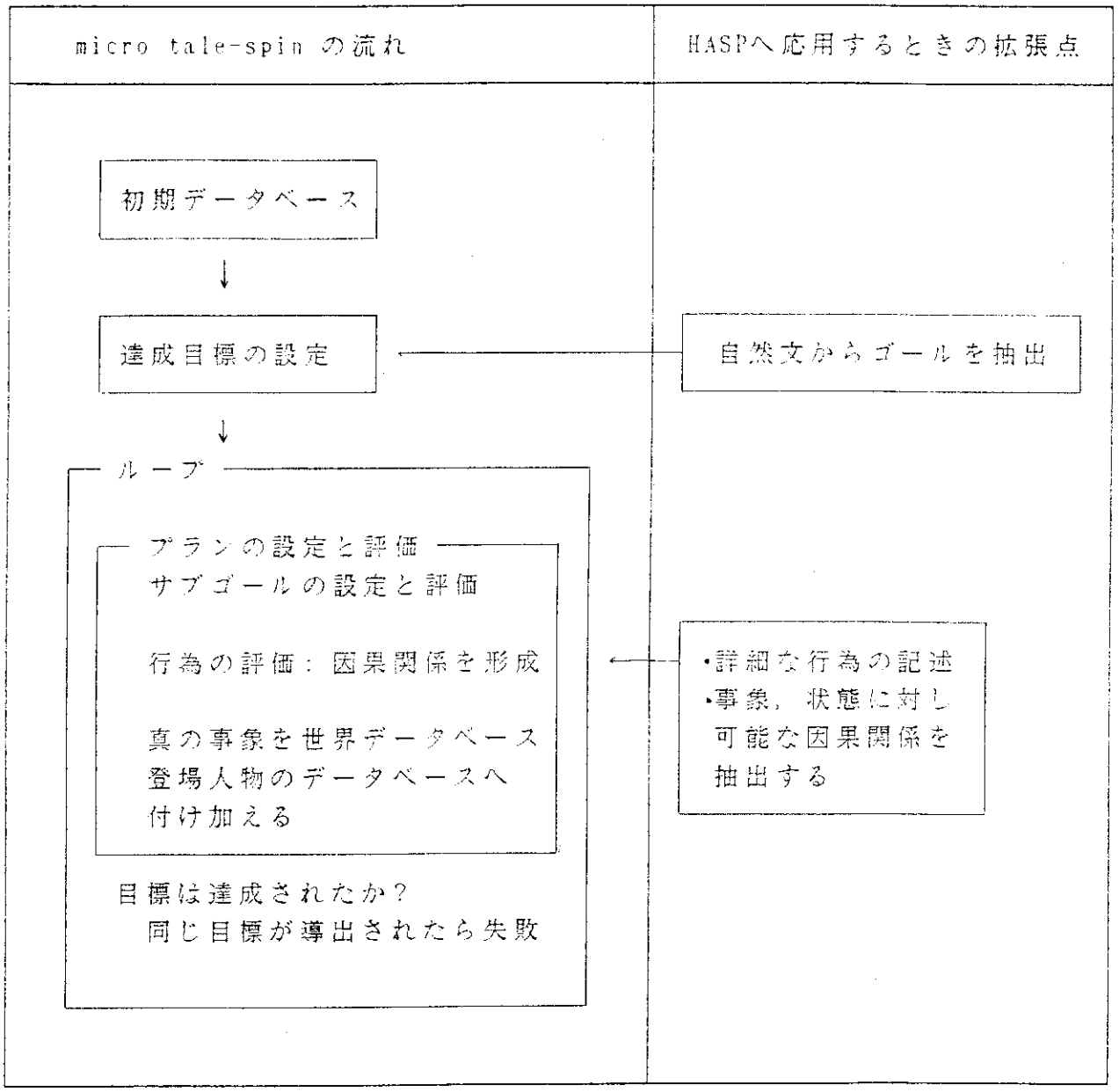


Fig. 2.1.4 Flow diagram of the micro TALE-SPIN.

命令　計器盤の扉を開けろ。

↓

行動
1. 計器盤の具体名を提示する。
2. 対象物の前面に向かう操作位置まで移動する。
3. 把手格納部に向かって把手に触れる位置まで手を移動する。
4. 把手を取り出す。
5. 把手をつかむ。
6. 計器盤内部が見えるまで把手を引く。
7. 基本姿勢をとる。
8. 作業の終了を報告する。

Fig. 2.2.1 I/O design of the demo system (1).

命令　計器盤の扉を開けろ。

↓

行動
1. IM（計器盤の具体名）
2. MB（＊，対象物の位置）＋PB（計器盤操作）
3. TE（＊，把手位置）
4. IL（把手）＋I（把手）
5. G（把手）
6. DA（計器盤本体，扉）＋I（計器盤）
7. AT
8. IM（作業の終了）

Fig. 2.2.2 I/O design of the demo system (2).

動詞概念「開ける」
- 対象物（〜を）：扉に属するインスタンス
- 動作主体：模擬人間
- 核動作　：対象物の開動作
- 準備動作：① 対象物を同定した
  ② 動作主体が対象物を操作できる位置にいる
- 整理動作：① そのままの位置で基本姿勢をとる
  ② 動作の終了を報告する

「扉」
- 部分　　：把手に属するインスタンス
- 開動作　：引数　① 動作主体の状態データを指すポインタ
  手続き　① 動作主体が把手をつかむ
  ② 開動作成就向きへ手を移動する
- 開動作成就向き：（未設定）

「箱体の扉」
- 部分　　：（継承）
- 開動作　：（継承）
- 開動作成就向き：
  扉の法線方向&箱体の外向き

「部屋の扉」
- 部分　　：（継承）
- 開動作　：（継承）
- 開動作成就向き：（未設定）

「電気系統箱体の扉」
- 部分　　：収納型把手に属するインスタンス
- 開動作　：（基本的に継承）
  （準備動作）① 把手がつかめる状態になっていることを調べる
  ② 把手がつかめる状態であれば準備動作は終了
  ② 把手がつかめる状態でなければ把手の取り出し動作を行なう
- 開動作成就向き：（継承）

「収納型把手」
- 取り出し動作：
  引数　① 動作主体の状態データを指すポインタ
  手続き　① 把手がつかめる状態になるよう把手に働きかける

Fig. 2.2.3 (a) Example of data for dealing with the input command "Open the door of the meter unit."

『模擬人間状態』
- 位置　　：A地点
- 注目装置：運転表示盤

インスタンス群

『運転表示盤』
- 部分　　：扉1
- ａｋｏ　：計器盤
- 位置　　：B地点
- 操作可能範囲：前面５０ｃｍ以内

『扉１』
- 部分　　：把手１
- ａｋｏ　：電気系統箱体の扉
- ａｐｏ　：運転表示盤

『把手１』
- ａｋｏ　：収納型把手
- ａｐｏ　：扉1
- 位置　　：座標C

ａｋｏ＝ａ　ｋｉｎｄ　ｏｆ
ａｐｏ＝ａ　ｐａｒｔ　ｏｆ

「計器盤の扉を開けろ。」

↓

開ける←(を)―扉←(の)―計器盤

↓

１．ＩＭ（扉１）
２．ＭＢ（Ａ地点，Ｂ地点）＋ＰＢ（Ｂ地点の前面５０ｃｍ以内）
３．ＴＥ（＊，把手１の位置）
４．ＩＬ（把手１）＋Ｉ（把手１）
５．Ｇ（把手１）
６．ＤＡ（運転表示盤，扉１）＋Ｉ（運転表示盤）
７．ＡＴ
８．ＩＭ（扉１を開ける）

※【解釈】　１，２：「開ける」の準備動作
３，４，５，６：「開ける」の核動作＝扉１の開動作
３，４：扉１の開動作の準備動作＝把手１の取り出し動作
５，６：扉１の開動作の核部分（ａｋｏによる継承）
７，８：「開ける」の整理動作

Fig. 2.2.3(b)　Example of data for dealing with the input command "Open the door of the meter unit."

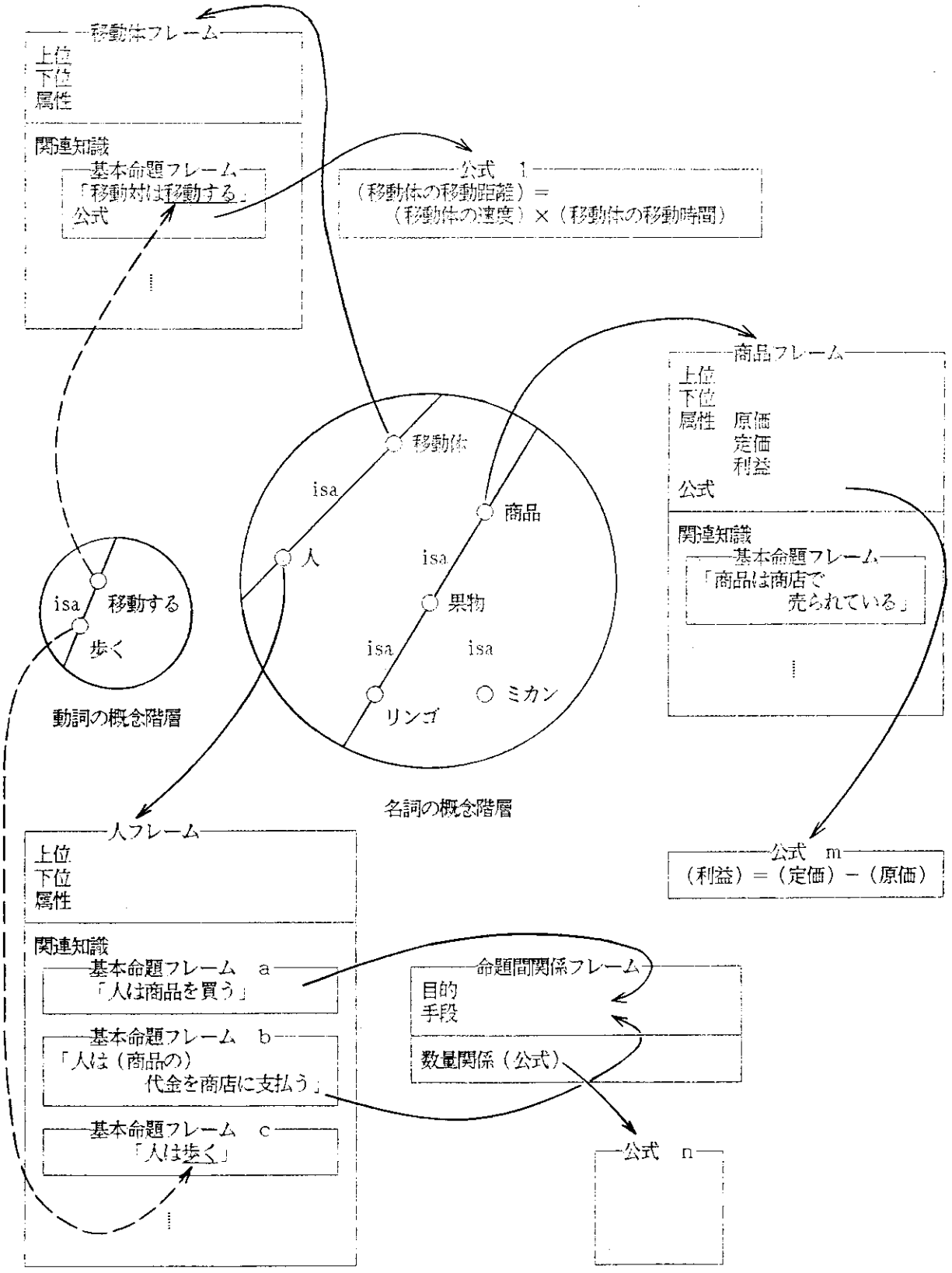


Fig. 2.2.4 Relationship among concepts.[10)]

付録A（2章）

マイクロtale-spinによって作られた物語
登場人物：熊のジョーと鳥のアービン

初期データベース

```
(MLOC (CON (LOC (ACTOR JOE)
                (VAL CAVE)))
      (VAL (CP (PART WORLD))))
 ; Joe was near the cave.
(MLOC (CON (LOC (ACTOR JOE)
                (VAL CAVE)))
      (VAL (CP (PART JOE))))
 ; Joe know that Joe was near the cave.
(MLOC (CON (LOC (ACTOR IRVING)
                (VAL OAK-TREE)))
      (VAL (CP (PART WORLD))))
 ; Irving was near the oak-tree.
(MLOC (CON (LOC (ACTOR IRVING)
                (VAL OAK-TREE)))
      (VAL (CP (PART IRVING))))
 ; Irving knew that Irving was near the oak-tree.
(MLOC (CON (LOC (ACTOR IRVING)
                (VAL OAK-TREE)))
      (VAL (CP (PART JOE))))
 ; Joe knew that Irving was near the oak-tree.
(MLOC (CON (LOC (ACTOR WATER)
                (VAL RIVER)))
      (VAL (CP (PART WORLD))))
 ; The water was near the river.
(MLOC (CON (LOC (ACTOR WATER)
                (VAL RIVER)))
      (VAL (CP (PART JOE))))
 ; Joe knew that the water was near the river.
(MLOC (CON (LOC (ACTOR HONEY)
                (VAL ELM-TREE)))
      (VAL (CP (PART WORLD))))
 ; Honey was near the elm-tree.
(MLOC (CON (LOC (ACTOR HONEY)
                (VAL ELM-TREE)))
      (VAL (CP (PART IRVING))))
 ; Irving knew that honey was near the elm-tree.
(MLOC (CON (LOC (ACTOR WORM)
                (VAL GROUND)))
      (VAL (CP (PART WORLD))))
 ; The worm was near the ground.
(MLOC (CON (LOC (ACTOR WORM)
                (VAL GROUND)))
      (VAL (CP (PART JOE))))
 ; Joe knew that the worm was near the ground.
(MLOC (CON (LOC (ACTOR JOE)
                (VAL CAVE)))
      (VAL (CP (PART IRVING))))
 ; Irving knew that Joe was near the cave.
(MLOC (CON (LOC (ACTOR FISH)
                (VAL RIVER)))
      (VAL (CP (PART WORLD))))
 ; The fish was near the fiver.
(MLOC (CON (LOC (ACTOR FISH)
                (VAL RIVER)))
      (VAL (CP (PART IRVING))))
 ; Irving knew that the fish was near the river.
(MLOC (CON (HUNGRY (ACTOR IRVING)
                   (MODE (POS))))
      (VAL (CP (PART WORLD))))
 ; Irving was hungry.
(MLOC (CON (LIKE (ACTOR IRVING)
                 (TO JOE)
                 (MODE (POS))))
      (VAL (CP (PART JOE))))
 ; Joe thought that Irving liked Joe.
(MLOC (CON (DECEIVE (ACTOR IRVING)
                    (TO JOE)
                    (MODE (NEG))))
      (VAL (CP (PART JOE))))
 ; Joe thought that irving did not deceive Joe.
(MLOC (CON (LIKE (ACTOR JOE)
                 (TO IRVING)
                 (MODE (POS))))
      (VAL (CP (PART JOE))))
 ; Joe thought that Joe liked Irving.
(MLOC (CON (LIKE (ACTOR IRVING)
                 (TO JOE)
                 (MODE (POS))))
      (VAL (CP (PART IRVING))))
 ; Irving thought that Irving liked Joe.
(MLOC (CON (DOMINATE (ACTOR IRVING)
                     (TO JOE)
                     (MODE (POS))))
      (VAL (CP (PART IRVING))))
 ; Irving thought that Irving dominated Joe.
(MLOC (CON (DECEIVE (ACTOR IRVING)
                    (TO JOE)
                    (MODE (NEG))))
      (VAL (CP (PART IRVING))))
 ; Irving thought that Irving did not deceive Joe.
```

物語生成開始

解決すべき状態・・ジョーはお腹がすいている。

```
(MLOC (CON (HUNGRY (ACTOR JOE)
                   (MODE (POS))))
      (VAL (CP (PART WORLD))))
 ; Joe was hungry.
(WANT (ACTOR JOE)
      (OBJECT (HUNGRY (ACTOR JOE)
                      (MODE (NEG)))))
 ; Joe wanted Joe not to be hungry.
(WANT (ACTOR JOE)
      (OBJECT (CONT (ACTOR HONEY)
                    (VAL JOE))))
 ; Joe wanted to have honey.
(WANT (ACTOR JOE)
      (OBJECT (MLOC (CON (LOC (ACTOR HONEY)
                              (VAL (*VAR* UNSPEC))))
                    (VAL (CP (PART JOE))))))
 ; Joe wanted to know where honey was.
(WANT (ACTOR JOE)
      (OBJECT (MTRANS (ACTOR IRVING)
                      (OBJECT (LOC (ACTOR HONEY)
                                   (VAL (*VAR* UNSPEC))))
                      (TO (CP (PART JOE)))
                      (FROM IRVING))))
 ; Joe wanted Irving to tell Joe where honey was.
(WANT (ACTOR JOE)
      (OBJECT (MLOC (CON (MTRANS (FROM IRVING)
                                 (TO (CP (PART JOE)))
                                 (OBJECT (LOC (ACTOR HONEY)
                                              (VAL (*VAR* UNSPEC))))
                                 (ACTOR IRVING)
                                 (MODE (QUES))))
                    (VAL (CP (PART IRVING))))))
 ; Joe wanted Irving to think that Irving told Joe where honey was.
(WANT (ACTOR JOE)
      (OBJECT (LOC (ACTOR JOE)
                   (VAL IRVING))))
 ; Joe wanted Joe to be near Irving.
(MLOC (CON (PTRANS (ACTOR JOE)
                   (OBJECT JOE)
                   (TO OAK-TREE)
                   (FROM CAVE)))
      (VAL (CP (PART WORLD))))
 ; Joe went to the oak-tree.
(MLOC (CON (LOC (ACTOR JOE)
                (VAL OAK-TREE)))
      (VAL (CP (PART WORLD))))
 ; Joe was near the oak-tree.
```

```
(MLOC (CON (MTRANS (ACTOR JOE)
                   (OBJECT (MTRANS (FROM IRVING)
                                   (TO (CP (PART JOE)))
                                   (OBJECT (LOC (ACTOR HONEY)
                                                (VAL (*VAR* UNSPEC))))
                                   (ACTOR IRVING)
                                   (MODE (QUES))))
                   (TO (CP (PART IRVING)))
                   (FROM JOE)))
      (VAL (CP (PART WORLD))))
 ; Joe asked Irving whether Irving would tell Joe where honey was.
(MLOC (CON (PLAN (ACTOR IRVING)
                 (OBJECT (MTRANS (FROM IRVING)
                                 (TO (CP (PART JOE)))
                                 (OBJECT (MTRANS (TIME FUTURE)
                                                 (ACTOR IRVING)
                                                 (OBJECT (LOC (ACTOR HONEY)
                                                              (VAL ?UNSPEC)))
                                                 (TO (CP (PART JOE)))
                                                 (FROM IRVING)
                                                 (MODE (NEG))))
                                 (ACTOR IRVING)
                                 (TIME FUTURE)))))
      (VAL (CP (PART WORLD))))
 ; Irving planned to tell Joe that Irving would not tell Joe where honey was.
(MLOC (CON (MTRANS (ACTOR IRVING)
                   (OBJECT (MTRANS (TIME FUTURE)
                                   (ACTOR IRVING)
                                   (OBJECT (LOC (ACTOR HONEY)
                                                (VAL (*VAR* UNSPEC))))
                                   (TO (CP (PART JOE)))
                                   (FROM IRVING)
                                   (MODE (NEG))))
                   (TO (CP (PART JOE)))
                   (FROM IRVING)
                   (TIME PAST)))
      (VAL (CP (PART WORLD))))
 ; Irving told Joe that Irving would not tell Joe where honey was.
```

```
(MLOC (CON (MBUILD (ACTOR JOE)
                   (OBJECT (CAUSE (ANTE (ATRANS (ACTOR JOE)
                                                (OBJECT WORM)
                                                (TO IRVING)
                                                (FROM JOE)))
                                  (CONSEQ (MTRANS (FROM IRVING)
                                                  (TO (CP (PART JOE)))
                                                  (OBJECT (LOC (ACTOR HONEY)
                                                               (VAL ?UNSPEC)))
                                                  (ACTOR IRVING)
                                                  (MODE (MAYBE)))))))
      (VAL (CP (PART WORLD))))
 ; Joe decided that if Joe would give Irving the worm then Irving might tell
   Joe where honey was.
(WANT (ACTOR JOE)
      (OBJECT (MLOC (CON (CAUSE (CONSEQ (MTRANS (FROM IRVING)
                                                (TO (CP (PART JOE)))
                                                (OBJECT (LOC (ACTOR HONEY)
                                                             (VAL ?UNSPEC)))
                                                (ACTOR IRVING)
                                                (TIME FUTURE)))
                                 (ANTE (ATRANS (ACTOR JOE)
                                               (OBJECT WORM)
                                               (TO IRVING)
                                               (FROM JOE)))
                                 (MODE (QUES))))
                    (VAL (CP (PART IRVING))))))
 ; Joe wanted Irving to think that Irving would tell Joe where honey was
   if Joe give Irving the worm.
(WANT (ACTOR JOE)
      (OBJECT (LOC (ACTOR JOE)
                   (VAL IRVING))))
 ; Joe wanted Joe to be near Irving.
(MLOC (CON (MTRANS (ACTOR JOE)
                   (OBJECT (CAUSE (CONSEQ (MTRANS (FROM IRVING)
                                                  (TO (CP (PART JOE)))
                                                  (OBJECT (LOC (ACTOR HONEY)
                                                               (VAL ?UNSPEC)))
                                                  (ACTOR IRVING)
                                                  (TIME FUTURE)))
                                  (ANTE (ATRANS (ACTOR JOE)
                                                (OBJECT WORM)
                                                (TO IRVING)
                                                (FROM JOE)))
                                  (MODE (QUES))))
                   (TO (CP (PART IRVING)))
                   (FROM JOE)))
      (VAL (CP (PART WORLD))))
 ; Joe asked Irving whether Irving would tell Joe where honey was if Joe gave
   Irving the worm.
```

```
(MLOC (CON (MBUILD (ACTOR IRVING)
                   (OBJECT (CAUSE (ANTE (ATRANS (ACTOR JOE)
                                                (OBJECT WORM)
                                                (TO IRVING)
                                                (FROM JOE)))
                                  (CONSEQ (MTRANS (TIME FUTURE)
                                                  (ACTOR IRVING)
                                                  (OBJECT (LOC (ACTOR HONEY)
                                                               (VAL ?UNSPEC)))
                                                  (TO (CP (PART JOE)))
                                                  (FROM IRVING)
                                                  (MODE (POS)))))))
      (VAL (CP (PART WORLD))))
 ; Irving decided that if Joe would give Irving the worm then Irving would tell
   Joe where honey was.
(MLOC (CON (MTRANS (ACTOR IRVING)
                   (OBJECT (CAUSE (ANTE (ATRANS (ACTOR JOE)
                                                (OBJECT WORM)
                                                (TO IRVING)
                                                (FROM JOE)))
                                  (CONSEQ (MTRANS (TIME FUTURE)
                                                  (ACTOR IRVING)
                                                  (OBJECT (LOC (ACTOR HONEY)
                                                               (VAL ?UNSPEC)))
                                                  (TO (CP (PART JOE)))
                                                  (FROM IRVING)
                                                  (MODE (POS))))))
                   (TO (CP (PART JOE)))
                   (FROM IRVING)))
      (VAL (CP (PART WORLD))))
 ; Irving told Joe that if Joe would give Irving the worm then Irving would tell
   Joe where honey was.
(WANT (ACTOR JOE)
      (OBJECT (CONT (ACTOR WORM)
                    (VAL JOE))))
 ; Joe wanted to have the worm.
(WANT (ACTOR JOE)
      (OBJECT (LOC (ACTOR JOE)
                   (VAL WORM))))
 ; Joe wanted Joe to be near the worm.
(MLOC (CON (PTRANS (ACTOR JOE)
                   (OBJECT JOE)
                   (TO GROUND)
                   (FROM OAK-TREE)))
      (VAL (CP (PART WORLD))))
 ; Joe went to the ground.
(MLOC (CON (LOC (ACTOR JOE)
                (VAL GROUND)))
      (VAL (CP (PART WORLD))))
 ; Joe was near the ground.
```

```
(MLOC (CON (ATRANS (ACTOR JOE)
                   (OBJECT WORM)
                   (TO JOE)
                   (FROM NIL)))
      (VAL (CP (PART WORLD))))
 ; Joe took the worm.
(MLOC (CON (CONT (ACTOR WORM)
                 (VAL JOE)))
      (VAL (CP (PART WORLD))))
 ; Joe had the worm.
(MLOC (CON (LOC (ACTOR WORM)
                (VAL JOE)))
      (VAL (CP (PART WORLD))))
 ; The worm was near Joe.
(WANT (ACTOR JOE)
      (OBJECT (LOC (ACTOR JOE)
                   (VAL IRVING))))
 ; Joe wanted Joe to be near Irving.
(MLOC (CON (PTRANS (ACTOR JOE)
                   (OBJECT JOE)
                   (TO OAK-TREE)
                   (FROM GROUND)))
      (VAL (CP (PART WORLD))))
 ; Joe went to the oak-tree.
(MLOC (CON (LOC (ACTOR JOE)
                (VAL OAK-TREE)))
      (VAL (CP (PART WORLD))))
 ; Joe was near the oak-tree.
(MLOC (CON (ATRANS (ACTOR JOE)
                   (OBJECT WORM)
                   (TO IRVING)
                   (FROM JOE)))
      (VAL (CP (PART WORLD))))
 ; Joe gave Irving the worm.
(MLOC (CON (CONT (ACTOR WORM)
                 (VAL IRVING)))
      (VAL (CP (PART WORLD))))
 ; Irving had the worm.
(MLOC (CON (LOC (ACTOR WORM)
                (VAL IRVING)))
      (VAL (CP (PART WORLD))))
 ; The worm was near Irving.
(MLOC (CON (CONT (VAL JOE)
                 (ACTOR WORM)
                 (MODE (NEG))))
      (VAL (CP (PART WORLD))))
 ; Joe did not have the worm.
```

```
(MLOC (CON (MTRANS (TIME PAST)
                   (ACTOR IRVING)
                   (TO (CP (PART JOE)))
                   (FROM IRVING)
                   (MODE (POS))
                   (OBJECT (LOC (ACTOR HONEY)
                                (VAL ELM-TREE)))))
      (VAL (CP (PART WORLD))))
 ; Irving told Joe that honey was near the elm-tree.
(WANT (ACTOR IRVING)
      (OBJECT (HUNGRY (ACTOR IRVING)
                      (MODE (NEG)))))
 ; Irving wanted Irving not to be hungry.
(MLOC (CON (INGEST (ACTOR IRVING)
                   (OBJECT WORM)))
      (VAL (CP (PART WORLD))))
 ; Irving ate the worm.
(MLOC (CON (HUNGRY (ACTOR IRVING)
                   (MODE (NEG))))
      (VAL (CP (PART WORLD))))
 ; Irving was not hungry.
(WANT (ACTOR JOE)
      (OBJECT (LOC (ACTOR JOE)
                   (VAL HONEY))))
 ; Joe wanted Joe to be near honey.
(MLOC (CON (PTRANS (ACTOR JOE)
                   (OBJECT JOE)
                   (TO ELM-TREE)
                   (FROM OAK-TREE)))
      (VAL (CP (PART WORLD))))
 ; Joe went to the elm-tree.
(MLOC (CON (LOC (ACTOR JOE)
                (VAL ELM-TREE)))
      (VAL (CP (PART WORLD))))
 ; Joe was near the elm-tree.
(MLOC (CON (ATRANS (ACTOR JOE)
                   (OBJECT HONEY)
                   (TO JOE)
                   (FROM NIL)))
      (VAL (CP (PART WORLD))))
 ; Joe took honey.
(MLOC (CON (CONT (ACTOR HONEY)
                 (VAL JOE)))
      (VAL (CP (PART WORLD))))
 ; Joe had honey.
(MLOC (CON (LOC (ACTOR HONEY)
                (VAL JOE)))
      (VAL (CP (PART WORLD))))
 ; Honey was near Joe.
```

```
(MLOC (CON (INGEST (ACTOR JOE)
                   (OBJECT HONEY)))
      (VAL (CP (PART WORLD))))
 ; Joe ate honey.
(MLOC (CON (HUNGRY (ACTOR JOE)
                   (MODE (NEG))))
      (VAL (CP (PART WORLD))))
 ; Joe was not hungry.
```

ゴールは達成された・・ジョーはお腹がすいていない。
物語は終了する。

3. ニューラル・ネットワーク（神経回路網）モデルによる画像認識研究

3.1 はじめに

ニューラル・ネットワークに関する研究は，ここ数年ブームと呼ぶべきフィーバぶりで，従来の人工知能（ＡＩ）研究に取って替わろうとする勢いである。これは，ニューラル・ネットワークに関して理論面での新しい展開があり，その有効性をシミュレーションにより，わかりやすい形で示せるようになったため，従来のＡＩ技術に限界を感じていた多くの研究者がニューラル・ネットワークの新しい可能性（人間の知的活動を解明する手がかりを与える可能性，従来のフォンノイマン型と本質的に異なる新しいニューロ・コンピュータ実現の可能性など）を求めて集まってきたためである。しかし，ニューラル・ネットワーク技術が従来のＡＩ技術にすべてとって代わり得るというものではない。従来のＡＩ技術では十分に扱い切れなかった画像や音声などのパターン認識や論理的に説明しにくい知識処理などに有効な手法として期待されている[1]のが現状で，記号処理を扱える従来のＡＩ技術の重要性は少しも変わっていない。この意味では，ニューラル・ネットワークの研究は，従来のＡＩ研究を補完するものとしてとらえられる。また，記号処理（従来のＡＩ）に対して非記号処理を扱うものとしてとらえることもできる。

人間動作シミュレーション技術の研究においては，筆者は画像認識にこのニューラル・ネットワーク技術を応用することを試みている。画像認識への応用例としては，教師付きの逆伝搬（Back-Propagation）学習則を用いたモデル（以下，ＢＰモデルと略称する。）を使用した研究[2]や教師無しのコグニトロン[3]，教師付きのネオコグニトロン[4][5]，選択的注意のモデル[6]などがある。ＢＰモデルによる画像認識は，画像に入るノイズやぼけに強く，画像の欠けには量と場所によりある程度対応できるが，対象のずれや大きさの変化には弱いということが報告されている[2]。これに対し，ネオコグニトロンでは文字認識において，文字に位置ずれや形のゆがみがあっても，大きさが少し違っていても，簡単な例では認識できることが報告されている[4]。筆者はニューラル・ネットワークに関する経験が無かったため，まず比較的簡単なBPモデルを使用して手書き数字の学習実験を試みた。この内容と結果を3.2節で示す。3.3節では，ネオコグニトロンの手法を使用して簡単な画像認識実験を行う計画とＳ６４年度の計画を簡単に示す。

3.2 Back-Propagation（ＢＰ）モデルによる手書き数字の学習実験

3.2.1 実験の概要

逆伝搬（Back-Propagation）学習則を利用した多層ニューラル・ネットワーク・モデル（以下，ＢＰモデルと略称する。）を使って手書き数字（０～９）の学習実験を行った。逆伝搬学習則に関しては文献7)，8)などに詳しい。この実験では，しきい値に関しても結合の重みに対する学習規則を使用する文献9)の考え方を使った。

(1) 入力データ

Fig. 3.1 に示す 9 × 6 のマス目に 0 ～ 9 の数字のひとつを書き入れ，各マス目を通過する線分量にしたがってこの数字のパターンを数値化した。ただし，数字は一番上の段と一番下の段に必ず線が通過するように書き入れた。各マス目は点線で 4 つの小マス目に区切られ，書かれた線が通過した小マス目の数 0 ～ 4 に数値化される。実験では手書き数字 0 ～ 9 を 2 組，計 2 0 個の数字を学習に使った。Fig. 3.2 に各数字の入力パターンを示す。

(2) ネットワーク構造

Fig. 3.3 にこの実験で使ったネットワークの構造を示す。3 層の，フィードバック結合をもたないネットワークである。入力層は，マス目の数と対応した 9 × 6 ＝ 5 4 個のユニットをもつ。中間層は，入力された文字パターンの特徴をつかむために Fig. 3.4 の 1 2 ユニットを設定した。まず 9 × 6 のマス目を大きく 9 つに区切り，左上，左中，左下，真上，中央，真下，右上，右中，右下の 9 つのユニットを設定した。次に，5 と 6 の文字パターンを区別しやすくするために No. 1 0 のユニットを，7 と 9 の文字パターンを区別しやすくするために No. 1 1 のユニットを，6 と 8 の文字パターンを区別しやすくするために No. 1 2 のユニットを設定した。出力層は，入力された文字パターンが 0 ～ 9 のどの数字であるかを認識するために，上から 1 を認識するユニット，次が 2 を認識するユニット，……，最後が 0 を認識するユニットと合計 1 0 個のユニットを設定した。入力層のユニットと中間層のユニットの結線は Fig. 3.4 に示される。中間層のユニットと出力層のユニットは総当りですべて結線されている。

(3) 誤差逆伝搬学習則アルゴリズム

一般に誤差逆伝搬学習アルゴリズムは，以下のように記述される[9]。

M ：階層の数

u_i^k ：第 k 層の第 i ユニット

i_j^k ：u_i^k への入力の総和

f ：ユニットの特性関数

o_i^k ：u_i^k の出力値

$w_{ij}^{k\,k+1}$ ：第 k 層の第 i ユニットから次の第 k ＋ 1 層の第 j ユニットへの結合の重み

θ_i^k ：第 k 層の第 i ユニットのしきい値

(x_p, y_p) ：学習に使う訓練用の入力と出力パターンの組みの集合 (p＝1, …, N)

上記のように各変数を定義すると，ある入力パターン x が入力されると，第 k 層では以下のように信号が伝搬される。

〔 前進計算 〕

$$\begin{cases} i_j^k = \sum_i w_{ij}^{k-1\,k}\, o_i^{k-1} & \cdots\cdots (3.1) \\ o_j^k = f(i_j^k) = \dfrac{1}{1+e^{-(i_j^k-\theta_j^k)}} & \cdots\cdots (3.2) \end{cases}$$

入力パターンxを入力したとき，（3.1），（3.2）式を使って出力層の第jユニットの出力o_j^Mが計算される。対応する正解出力（教師信号）パターンyの第j成分をy_jとすると，誤差逆伝搬学習では，次の式に従って各結合の重みを修正する。

〔逆伝搬計算〕

$$\omega^{k-1}{}_i{}^k{}_j = \omega^{k-1}{}_i{}^k{}_j + \eta\,\delta^k{}_j o^{k-1}{}_i \quad \cdots\cdots \quad (3.3)$$

$$\delta^M{}_j = (y_j - o^M{}_j)\,f'(i^M{}_j) \quad \cdots\cdots \quad (3.4)$$

$$\delta^{k-1}{}_i = f'(i^{k-1}{}_i)\sum_j \omega^{k-1}{}_i{}^k{}_j\,\delta^k{}_j \ (k=M,\cdots,3) \quad \cdots\cdots \quad (3.5)$$

（3.3）式が重みの修正式であり，ηは小さな正の数である。δ_j^kを一般化された誤差などと呼ぶ。式（3.5）の式はδ_j^kを，ちょうど出力層での誤差$\delta^M{}_j$を出力層への入力として，出力層から入力層の方向へ逆方向に（$f'(i^k{}_j)$を掛けながら）伝播しているような形でk＝Mから3まで順次計算するための漸化式である。これが誤差逆伝播という名前の由来である。

ユニットの特性関数としてロジスティック関数を用いている場合には，$f'(i^k{}_j)$は$o^k{}_j = f(i^k{}_j)$の値を使って

$$f'(i^k{}_j) = o^k{}_j\,(1 - o^k{}_j) \quad \cdots\cdots \quad (3.6)$$

として計算できる。

修正式（3.3）は，ユニット$u^k{}_j$への結合を，$u^k{}_j$の一般化された誤差δ_j^kとその結合が伝えている信号$o^{k-1}{}_i$との積に応じて修正するものであり，誤差が正，すなわち，$u^k{}_j$の活動が足りない場合には，$u^k{}_j$へ正の信号を送っていたユニットからの結合を増やし，負の信号を送っていたユニットからの結合を減らすように働くことが分かる。誤差が負の場合には結合の修正も逆符号になる。

この学習法によって最終的に得られる結合は，訓練セットに関して，ネットワークの出力と正解との2乗誤差の総和

$$R = \sum_p (y_p - o^M(x_p))^2 \quad \cdots\cdots \quad (3.7)$$

を極小にするものであることが知られている。

また，しきい値に関して，各層に一つ常に－1を取る特殊なユニット$u^k{}_0$を置くと，各ユニットのしきい値θ_j^{k+1}をそのユニットからの結合の重み$\omega^k{}_0{}_j{}^{k+1}$に帰着させることができる。こうすることで，しきい値に関する学習規則を結合の重みに対する学習規則に帰着させることができる。

学習のスピードを上げるために，（3.3）式の重みの修正において，以下のようにmomentum term（運動量項）を付け加える方法もある[7]。

$$\Delta\omega_{ij}^{k-1\ k}(n+1) = \eta\,\delta_j^k\,o_i^{k-1} + \alpha\,\Delta\omega_i^{k-1}{}_j{}^k(n) \quad \cdots\cdots \quad (3.8)$$

しかし，今回の実験では学習させる数字を1～9，0とcyclicに与えるため，$\Delta\omega_{ij}(n+1)$

の値は$\Delta\omega_{ij}(n)$の値と関連がないので，この式は用いなかった。単に（3.3）式を用いた。

(4) この実験における学習アルゴリズム

Fig. 3.3 のネットワークに対して，(3)の誤差逆伝搬学習則アルゴリズムを適用させる。

〔変数の定義〕

DATA（k）	：入力パターン，k＝1～54
U（i）	：中間層の第iユニットへの入力の総和，i＝1～12
O（i）	：中間層の第iユニットの出力値
V（j）	：出力層の第jユニットへの入力の総和，j＝1～10
FIG（j）	：出力層の第jユニットの出力値
TEACH（j）	：教師信号パターン
WI（k）	：入力層の第kユニットは，中間層の第1～第9ユニットのいずれかひとつへ結合されている。その結合の重みである。
W10（ℓ）	：入力層の第11～14ユニットから中間層の第10ユニットへの結合の重み，ℓ＝1～4
W11（ℓ）	：入力層の第25～28ユニットから中間層の第11ユニットへの結合の重み
W12（ℓ）	：入力層の第41～44ユニットから中間層の第12ユニットへの結合の重み
WO（i，j）	：中間層の第iユニットから出力層の第jユニットへの結合の重み
WTI（i）	：中間層の第iユニットのしきい値
WO（j）	：出力層の第jユニットのしきい値

上記のように各変数を定義すると，入力パターンDATA（k）が入力されると，次のように信号が伝搬される。

〔前進計算〕

$$U(i)=\sum_{k=(i-1)*6+1}^{(i-1)*6+6} WI(k)*DATA(k) \quad (i=1\sim9) \quad \cdots\cdots \quad (3.9)$$

$$U(10)=\sum_{k=11}^{14} WI(k)*DATA(k) \quad \cdots\cdots \quad (3.10)$$

$$U(11)=\sum_{k=25}^{28} WI(k)*DATA(k) \quad \cdots\cdots \quad (3.11)$$

$$U(12)=\sum_{k=41}^{44} WI(k)*DATA(k) \quad \cdots\cdots \quad (3.12)$$

$$O(i)=\frac{1}{1+e^{-(U(i)-WTI(i))}} \quad \cdots\cdots \quad (3.13)$$

$$V(j)=\sum_{i=1}^{12} WTO(i,j)*O(i) \quad \cdots\cdots \quad (3.14)$$

$$FIG(j)=\frac{1}{1+e^{-(V(j)-WTO(j))}} \quad \cdots\cdots (3.15)$$

（3.15）式のFIG(j)，j＝1～10は，入力された文字パターンに対してこのニューラル・ネットワークがそれぞれ1～9，0である可能性を判定した値となっている。これを正解値（教師信号）と比較し，誤差があればその誤差の最小2乗を極小化するように，ユニット間の結合の重みを修正する。この重みの修正が学習に相当する。

〔逆伝搬計算〕

$$WO(i,j)=WO(i,j)+\eta*Q(j)*O(i) \quad \cdots\cdots (3.16)$$

$$WTO(j)=WTO(j)+\eta*Q(j)*(-1) \quad \cdots\cdots (3.17)$$

ここで，

$$Q(j)=(TEACH(j)-FIG(j))*FIG(j)*(1-FIG(j)) \quad \cdots\cdots (3.18)$$

である。

$$WI(k)=WI(k)+\eta*P(i)*DATA(k) \quad \cdots\cdots (3.19)$$

$$WTI(i)=WTI(i)*\eta*P(i)*(-1) \quad \cdots\cdots (3.20)$$

ここで，

$$P(i)=O(i)*(1-O(i))*\sum_{j} WO(i,j)*Q(j) \quad \cdots\cdots (3.21)$$

であり，（3.19）式はi＝1のときk＝1～6，i＝2のときk＝7～12，……，i＝9のときk＝49～54が対応する。

また，

$$\left.\begin{array}{l}
W10(1)=W10(1)+\eta*P(2)*DATA(11)\\
W10(2)=W10(2)+\eta*P(2)*DATA(12)\\
W10(3)=W10(3)+\eta*P(3)*DATA(13)\\
W10(4)=W10(4)+\eta*P(3)*DATA(14)\\
W11(1)=W11(1)+\eta*P(5)*DATA(25)\\
W11(2)=W11(2)+\eta*P(5)*DATA(26)\\
W11(3)=W11(3)+\eta*P(5)*DATA(27)\\
W11(4)=W11(4)+\eta*P(5)*DATA(28)\\
W12(1)=W12(1)+\eta*P(7)*DATA(41)\\
W12(2)=W12(2)+\eta*P(7)*DATA(42)\\
W12(3)=W12(3)+\eta*P(8)*DATA(43)\\
W12(4)=W12(4)+\eta*P(8)*DATA(44)
\end{array}\right\} \quad \cdots\cdots (3.22)$$

である。

(5) 教師信号

手書きの数字j（j＝0～9）のパターンを入力したときは，jを認識する出力ユニットにのみ1.0を与え，その他のユニットには0.0を与えた。前者に0.9を与え，後者に0.1を与えた方がよいとする考え方もある[7]が，今回の実験では1.0と0.0を与えて，きちんと学習が完了した。

(6) 学習方法

手書きの数字0～9を2組，計20個の入力パターンを使って，（n－1）回学習させ，n回目の認識で正しく判定しているかどうかを調べた。学習は1～9，0，1～9，0の順に入力パターンを与え，これを1回（1サイクル）とカウントした。

(7) 調査内容

この実験では，以下の2つ内容の調査を試みた。

(i) Fig. 3.3のネットワークが,与えた20個の入力パターンをすべて正しく認識する（学習を完了する）ための要件。重みω，しきい値θの初期値や学習定数ηの値が認識誤差の収束状況にどんな影響を与えるかなどを調べる。

(ii) 学習完了状態でネットワークが何を学んだか。結合の重みωとしきい値θの変化（収束値）について調べる。

3.2.2 実験結果

この実験では，(i)学習の完了要件（結合の重みωとしきい値θの初期値や学習定数ηの値が認識誤差の収束状況にどんな影響を与えるか）と(ii)ネットワークの学習事項（学習完了状態における結合の重みωとしきい値θの状況）について調べた。以下にその結果を示す。

(1) 学習の完了要件

1) 結合の重みωとしきい値θの初期値

一般に層間のユニット結合が総当りですべてがつながっている場合，結合の重みωの初期値は（0，1）の乱数が使用されることが多い。これは各層内のユニットが結線において均一なため，結合の重みωの初期値（出発点）を変えることによって各ユニットの独自性を与えようとするためと考えられる。

この実験では，中間層のユニット12個はあらかじめ処理内容が性格づけられているため，結合の重みωの初期値として乱数を与える必要はない。そこで，結合の重みωとしきい値θに関してすべて0.0を与えたケースとすべてに1.0を与えたケースについて比較した。Table 3.1にその結果を示す。Table 3.1は，各学習回数後の数字認識（20個の数字）において，ネットワークの判定（出力）と教師信号（正解）の誤差の最大値を示したものである。（3.3）式のηを適切に選べば，ωとθは0.0を与えても1.0を与えても全体の学習は完了し，認識誤差の収束状況も大差なかった。Table 3.1からは，学習回数が少ない段階（学習回数50回くらいまで）では初期値に1.0を与えたケースが誤差の収束は速かった。しかし，学習回数が多くなってくると初期値に0.0を与えたケースが誤差の収束がよくなっている。これは，この実験では学習完了（すべての入力パターンに対して認識誤差が許容値以下になった）時における結合の重みωと

しきい値θの総平均が1.0より0.0に近いためと考えられる。しかし，まだ詳しい分析は行っていない。Table 3.1の結果からは，ωとθの初期値よりηの値の方が誤差の収束に対してより大きく寄与していることがうかがえる。

2) 学習定数ηの値

Table 3.1では，（3.3）式のηを4.0にしたケースがηを1.0にしたケースより，誤差の収束が速かった。ここでは，ηの値と誤差の収束の関係について調べた結果を示す。Table 3.2は，結合の重みωとしきい値θの初期値をすべて0.0と与えたときのηの値と認識誤差の最大値の関係を示す。ηの値が4.0になるまではηの値が大きいほど収束は速く，認識誤差が小さくなっている。しかし，ηの値として8，16を与えると認識できない数字が出て来て，学習が完了しなくなった。この原因について以下に簡単な考察を示す。

ηは，重みωの修正式（3.3）に表われる学習定数である。（3.3）式の修正項は，k＝M（出力層のユニットにつながる結合）においては（3.4），（3.6）式を使って以下のように展開される。

$$\eta\,\delta_j^M o_i^M = \eta\,(y_j - o_j^M)\,o_j^M\,(1 - o_j^M)\,o_i^{M-1} \qquad \cdots\cdots (3.23)$$

ここで，$0 \leqq o_j^{M-1} \leqq 1$であるため，$o_j^M(1-o_j^M)$は$o_j^M=0.5$のとき最大値0.25をとる。$0 \leqq o_i^{M-1} \leqq 1$であるため，$\eta$が4.0を越えると，誤差$(y_j - o_j^M)$は，拡大されて重み$\omega$を修正する可能性が出てくる。k＝M－1，M－2，…，3と逆伝搬的に重みが修正されていくため，最終的にはO_i^1（入力データ）にまで関係するために，ηの与え方は個々のモデルによって変わってくるが，ひとつの目安としてηは4を越えると誤差の収束に悪い影響を与えるのではないかと考えられる。

(2) ネットワークの学習事項

ＢＰモデルでは，学習の訓練は結合の重みωとしきい値θの変化として表われる。ここでは，この実験で最も認識誤差が小さくなったケース，すなわちωとθの初期値をすべて0.0にし，η＝4.0で学習させ，５０００回の学習後の状態を分析する。

Fig. 3.5に５０００回の学習後の認識状況を示す。ＩＤ＝１～２０がFig. 3.2のNo.1～No.２０の数字入力パターンに対応する。ＩＤ＝１～９は１～９の数字パターン，ＩＤ＝１０は０の数字パターン，ＩＤ＝１１～１９は１～９の数字パターン，ＩＤ＝２０は０の数字パターンが入力されている。図中，ＵとＯは中間層の入力と出力，ＶとＦＩＧはそれぞれ出力層の入力と出力，Ｔは教師入力，ＤＩＦは教師入力（Ｔ）からこのネットワークの判定値（ＦＩＧ）を差し引いた誤差である。２０個の入力パターンすべてについて認識誤差は0.01以下になっている。

Fig. 3.6に５０００回学習後の重み（ＷＩ，Ｗ１０，Ｗ１１，Ｗ１２，ＷＯ），としきい値（ＷＴＩ，ＷＴＯ）の状態を示す。ＷＴＩは中間層の１２個のユニットに対するしきい値である。ＷＴＯは出力層の１０個のユニットに対するしきい値である。ＷＩは，９×６の入力スクリーンの各マス目から中間層に一つずつでている結線（Fig. 3.4参照）についての重みである。Ｗ１０，Ｗ１１，Ｗ１２もFig. 3.4に示される結線についての

重みである。WOは１２×１０で，中間層の１２ユニットから出力層の１０ユニットへの結線についての重みである。

これらの学習状況をわかりやすく図示すると以下のようになる。まず，Fig. 3.7 に各入力パターンが中間層を通過した段階でどのように変化しているかを示す。（ 3.2 ）式の変換後の値であるから $0 \leqq O_i \leqq 1$ である。図中の空白部は 0.0 である。たとえば，No.1（手書き数字１）は，中央列のみが以後の評価対象になるように加工されている。この段階は，各数字の特徴抽出がなされているようである。

次に，出力層の１０個のユニットにつながる結線の重みをFig. 3.8 に示す。出力層の各ユニットは，入力されたパターンが自分の担当する数字（１番目のユニットは１，２番目のユニットは２，……，９番目のユニットは９，１０番目のユニットは０を担当している）かどうかを最終判定する役割をもっている。Fig. 3.8 の(1)を担当しているユニットの重みは，中央列のみを正の評価をし，左右の列に信号の来るパターン（例えば，Fig. 3.2 のNo.2の入力パターン）に対しては負の評価をするようになっている。

しかし，Fig. 3.8 の(4)を担当しているユニットにおいては，上段中央列の信号に対する重みは－5.6 の負の評価となっているが，これは人間の感覚とは異なる。４の数字は，誰が書いても上段中央列に筆を走るはずであり，ここは正の評価をすべき所のはずである。ここが負になっている理由は，Fig. 3.7 との関係を見ないと理解しがたい。たとえば，No.4 の入力パターン（手書き文字４）が入力されるとFig. 3.9 に示す処理が進行する。中間層の出力ではFig. 3.7 のNo.4 に示される信号に変換され，この信号がFig.38の4番目の出力層ユニットにつながる重みで評価される。それらの総和からしきい値を差し引いた値が（3.2）式で変換され，その入力パターンが数字４であるか否かの判定値FIG(4)＝0.99662として出てくる。同様にNo.3の入力パターン（手書き文字３）が入力されるとその入力パターンが数字４であるか否かの判定値はFig. 3.5 より 0.00003 として出てくることがわかる。このように学習結果は，すべての重み ω とすべてのしきい値に分散されて記憶されており，個々の重みを見て議論してもいけない。しかし，この実験のような単純なモデルにおいては，Fig. 3.8 を見ると１は中央列を通る文字で，５は左下（中間層の１０番目のユニット）のマス目を通過しないという特徴があり，０は中段中央列（中間層の５番目と１０番目のユニット）のマス目を通過しない特徴をもつことなどが学習結果として読み取れる。

(3) 計算時間

この実験では，Fig. 3.3，Fig. 3.4 に示すように入力層５４ユニット，中間層１２ユニット，出力層と中間層の結線数６６，中間層と出力層の結線数１２０のネットワーク・モデルを使った。２０個の入力パターンを５０００回学習させるためにかかったＣＰＵ時間は，ＦＡＣＯＭの汎用大型計算機Ｍ７８０で１６.４２秒である。

3.2.3 考　察

前項で示した実験結果の分析から，ＢＰモデルについて以下のことが推論される。

① ＢＰモデルは，文字をビットパターンとしてのみとらえており，中間層に特別なユニット（たとえば，線分検出ユニットなど）を作らない限り，不特定多数者による手書き文字

の認識に使うのは難しい。

② 同様に学習した文字群と大きさの異った文字や書く位置が異った場合はうまく認識できないであろう。

③ 上記②のケースを克服しようとして，ＢＰモデルに異った大きさの同じ文字を学習させたり，同じ文字について書く位置を変えたものを多数学習させようとしてもＢＰモデルはそれらの文字をすべて正しく認識できるように自己組織化できない（学習が完了しない）であろう。このため，②のケースを克服するためには，文字の大きさを入力スクリーンに合わせる，文字の重心を中央列に置くなどの前処理が必要となる。

④ ＢＰモデルは，あらかじめ入力パターンが他の範鋳に入る入力パターンと分離可能な要素をもっているものに対して効力を発揮すると思われる。

Table 3.1 Maximum recognition errors for handwritten figures.

初期値	η	学習回数						
		10	20	30	40	50	500	5000
0.0	1.00	0.94563	0.94532	0.92810	0.90699	0.64343	0.05194	0.01396
1.0	1.00	0.99997	0.94768	0.79216	0.57208	0.39303	0.05941	0.01714
0.0	4.00	0.97170	0.97985	0.99037	0.20494	0.12695	0.02605	0.00775
1.0	4.00	0.99339	0.93453	0.30022	0.10813	0.09168	0.02678	0.00815

Table 3.2 Relations between η and maximum error.

初期値	η	学習回数	
		1000	5000
0.0	0.1	0.16106	0.05386
0.0	0.2	0.09840	0.03669
0.0	0.5	0.05378	0.02191
0.0	1.0	0.03456	0.01396
0.0	2.0	0.02640	0.01124
0.0	4.0	0.01805	0.00775
0.0	8.0	0.99999 *1	1.00000 *1
0.0	16.0	1.00000 *2	1.00000 *2

*1 ：No.17の手書き数字パターン（7）を7と認識できない。

*2 ：すべての入力パターン（0～9）を1と誤って認識する。

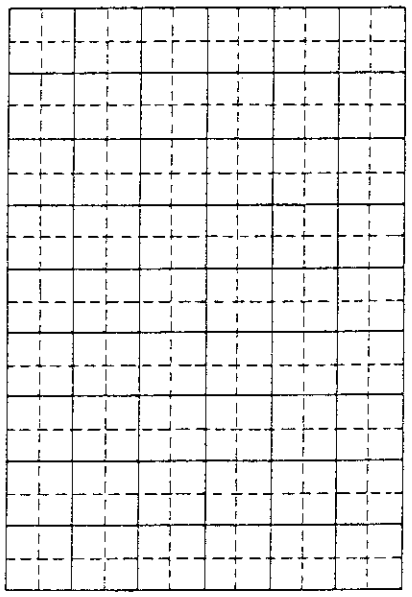


Fig. 3.1 Input screen (9×6).

```
*** INPUT DATA ***
          ND  =     20
          NIT =   5000
          ETA =      4.00
```

(NO. 1)	(NO. 2)	(NO. 3)	(NO. 4)	(NO. 5)	(NO. 6)	(NO. 7)	(NO. 8)	(NO. 9)	(NO. 10)
000200	022221	132222	003000	122222	013221	222223	022220	001330	013200
000200	210002	200002	012000	200000	130002	200012	210012	021002	030130
000200	200003	000003	030200	200000	300000	200020	200002	120002	210012
001200	000012	000230	021200	200000	200000	120020	210011	200132	200002
002100	000030	123310	302100	122320	222210	000020	023320	223230	200002
002000	000310	000031	223222	000012	300012	000020	021310	000020	200002
002000	002100	000003	002000	000002	300002	000120	200011	000210	200021
002000	031000	100003	002000	100021	200021	000200	200002	001200	200030
002000	332222	132221	002000	122210	122310	000100	132231	001000	022200

(NO. 11)	(NO. 12)	(NO. 13)	(NO. 14)	(NO. 15)	(NO. 16)	(NO. 17)	(NO. 18)	(NO. 19)	(NO. 20)
000200	012220	012200	002000	002221	000021	022221	002320	001320	002210
000300	120012	000130	003000	012000	000310	030030	021011	012020	003031
000200	110002	000012	021030	030000	001100	200120	020021	020020	011002
001100	000002	000002	030200	032100	013210	000200	012020	200130	020002
002000	000021	013420	210300	000130	032021	000300	002310	223220	020002
003000	000120	000131	223322	000012	120002	001200	003200	000210	120012
002000	001300	000003	003000	000002	200011	003000	030200	000300	110030
002000	013000	200002	012000	210031	200130	002000	030300	000200	020310
010000	232221	022221	010000	012210	132200	011000	012100	002100	012100

Fig. 3.2 Input figure patterns.

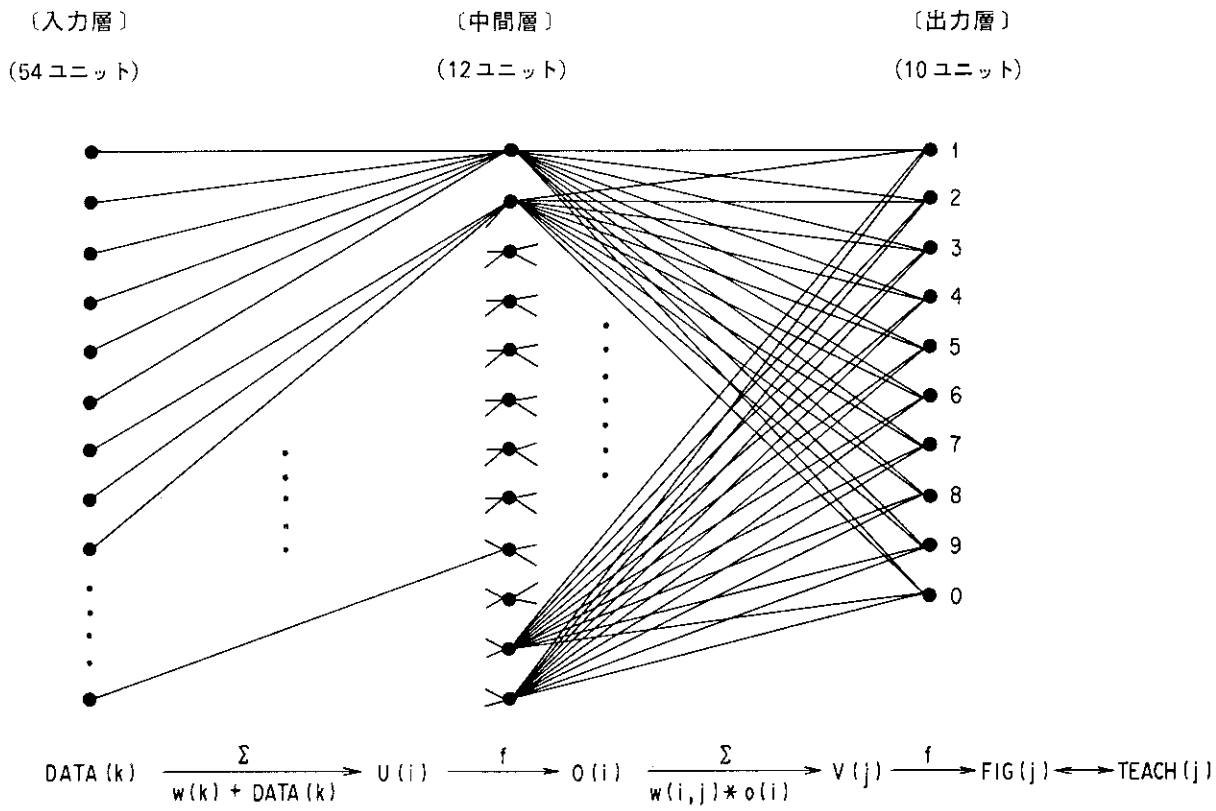


Fig. 3.3 Network structure (54-12-10).

1	2	19	20	37	38
3	4	21	22	39	40
5	6	23	24	41	42
7	8	25	26	43	44
9	10	27	28	45	46
11	12	29	30	47	48
13	14	31	32	49	50
15	16	33	34	51	52
17	18	35	36	53	54

				41	42
		25	26	43	44
		29	28		
11	12				
13	14				

〔入力層〕 〔中間層〕

1 ～ 6 ―――― 1

7 ～12 ―――― 2

13～18 ―――― 3

19～24 ―――― 4

25～30 ―――― 5

31～36 ―――― 6

37～42 ―――― 7

43～48 ―――― 8

49～54 ―――― 9

11 ～14 ―――― 10

25～28 ―――― 11

41～44 ―――― 12

Fig. 3.4 Connected lines between input layer and intermediate layer.

*** ITERATION =5000 ***

ID = 1

	U	O	V	FIG	T	DIF
1	-1.78508	0.14368	5.34125	0.99609	1.00000	0.00391
2	-3.96656	0.01859	-9.94625	0.00005	0.0	-0.00005
3	-5.33535	0.00480	-5.92585	0.00266	0.0	-0.00266
4	8.10434	0.99970	-6.08486	0.00227	0.0	-0.00227
5	8.74924	0.99984	-6.49796	0.00150	0.0	-0.00150
6	7.85245	0.99961	-9.21067	0.00010	0.0	-0.00010
7	-3.43865	0.03051	-6.06519	0.00232	0.0	-0.00232
8	-3.24568	0.03748	-6.47709	0.00154	0.0	-0.00154
9	-3.02466	0.04632	-3.87875	0.00279	0.0	-0.00279
10	-2.51531	0.07479	-18.17456	0.00000	0.0	-0.00000
11	4.77601	0.99164				
12	-0.63294	0.34684				

ID = 2

	U	O	V	FIG	T	DIF
1	5.59885	0.99631	-7.28201	0.00069	0.0	-0.00069
2	-3.96656	0.01859	5.23565	0.99471	1.00000	0.00529
3	10.26728	0.99997	-7.71793	0.00044	0.0	-0.00044
4	4.75404	0.99146	-8.41160	0.00022	0.0	-0.00022
5	-7.60033	0.00050	-7.06017	0.00086	0.0	-0.00086
6	3.86289	0.97943	-18.34272	0.00000	0.0	-0.00000
7	-0.48278	0.38160	-5.17165	0.00564	0.0	-0.00564
8	-0.53951	0.36830	-5.56059	0.00383	0.0	-0.00383
9	-8.25397	0.00026	-17.92030	0.00000	0.0	-0.00000
10	-2.51530	0.07479	-7.58885	0.00051	0.0	-0.00051
11	-3.40618	0.03210				
12	0.91378	0.71377				

ID = 3

	U	O	V	FIG	T	DIF
1	-4.86860	0.00763	-7.52208	0.00054	0.0	-0.00054
2	1.08119	0.74672	-10.65712	0.00002	0.0	-0.00002
3	2.86226	0.94595	5.07183	0.99377	1.00000	0.00623
4	4.75402	0.99146	-10.43685	0.00003	0.0	-0.00003
5	8.79176	0.99985	-7.29161	0.00068	0.0	-0.00068
6	-4.49967	0.01099	-5.53707	0.00392	0.0	-0.00392
7	0.86122	0.70292	-18.30302	0.00000	0.0	-0.00000
8	2.62425	0.93241	-6.48310	0.00153	0.0	-0.00153
9	-5.66224	0.00346	-5.76197	0.00314	0.0	-0.00314
10	-2.51532	0.07479	-10.30230	0.00003	0.0	-0.00003
11	8.28989	0.99975				
12	-0.70200	0.33137				

ID = 4

	U	O	V	FIG	T	DIF
1	7.13745	0.99921	-6.69364	0.00124	0.0	-0.00124
2	18.91588	1.00000	-13.69874	0.00000	0.0	-0.00000
3	-5.33534	0.00480	-13.54080	0.00000	0.0	-0.00000
4	-5.24149	0.00526	5.68566	0.99662	1.00000	0.00338
5	7.35425	0.99936	-10.94504	0.00002	0.0	-0.00002
6	7.85255	0.99961	-5.72046	0.00327	0.0	-0.00327
7	-3.45870	0.03051	-12.83389	0.00000	0.0	-0.00000
8	4.33367	0.98705	-10.47362	0.00003	0.0	-0.00003
9	-3.02466	0.04632	-5.57318	0.00378	0.0	-0.00378
10	5.37625	0.99623	-12.22320	0.00000	0.0	-0.00000
11	4.77600	0.99164				
12	-0.63252	0.34694				

ID = 5

	U	O	V	FIG	T	DIF
1	4.59318	0.98998	-10.88406	0.00002	0.0	-0.00002
2	-2.39148	0.08382	-12.19991	0.00001	0.0	-0.00001
3	1.40030	0.80223	-6.11266	0.00221	0.0	-0.00221
4	4.75404	0.99146	-14.19083	0.00000	0.0	-0.00000
5	5.66673	0.99653	4.83165	0.99225	1.00000	0.00775
6	-4.49968	0.01099	-5.78871	0.00305	0.0	-0.00305
7	0.30409	0.62342	-11.87802	0.00001	0.0	-0.00001
8	3.34642	0.96599	-6.13653	0.00216	0.0	-0.00216
9	5.48980	0.99589	-9.09306	0.00011	0.0	-0.00011
10	-2.51538	0.07479	-12.10934	0.00001	0.0	-0.00001
11	5.16486	0.99432				
12	-0.63246	0.34695				

ID = 6

	U	O	V	FIG	T	DIF
1	13.16350	1.00000	-11.73961	0.00001	0.0	-0.00001
2	11.69760	0.99999	-13.81534	0.00000	0.0	-0.00000
3	4.13873	0.98431	-10.53976	0.00003	0.0	-0.00003
4	4.69241	0.99092	-7.34151	0.00053	0.0	-0.00053
5	4.79633	0.99181	-5.34912	0.00473	0.0	-0.00473
6	-5.39252	0.00453	4.88781	0.99252	1.00000	0.00748
7	-3.11690	0.04242	-20.33260	0.00000	0.0	-0.00000
8	2.78499	0.94186	-6.15631	0.00212	0.0	-0.00212
9	5.48988	0.99589	-9.40138	0.00008	0.0	-0.00008
10	8.20165	0.99973	-6.51233	0.00148	0.0	-0.00148
11	4.29445	0.98654				
12	-0.63253	0.34694				

ID = 7

	U	O	V	FIG	T	DIF
1	5.09984	0.99394	-10.30612	0.00003	0.0	-0.00003
2	-5.86149	0.00284	-6.21910	0.00199	0.0	-0.00199
3	-5.33531	0.00480	-5.95840	0.00258	0.0	-0.00258
4	4.75398	0.99146	-10.52155	0.00003	0.0	-0.00003
5	-2.90431	0.05194	-5.43744	0.00433	0.0	-0.00433
6	-7.27822	0.00069	-16.24643	0.00000	0.0	-0.00000
7	2.11840	0.89268	5.65617	0.99652	1.00000	0.00348
8	0.29736	0.57385	-7.33561	0.00065	0.0	-0.00065
9	-2.81605	0.05646	-5.42628	0.00438	0.0	-0.00438
10	-2.51550	0.07478	-5.45198	0.00427	0.0	-0.00427
11	-3.40619	0.03210				
12	-1.71240	0.15285				

ID = 8

	U	O	V	FIG	T	DIF
1	5.59885	0.99631	-6.33888	0.00176	0.0	-0.00176
2	-5.43432	0.00435	-9.22925	0.00010	0.0	-0.00010
3	5.06427	0.99372	-6.65395	0.00129	0.0	-0.00129
4	4.75403	0.99146	-10.31433	0.00003	0.0	-0.00003
5	5.68634	0.99662	-6.62963	0.00132	0.0	-0.00132
6	-4.49971	0.01099	-7.10938	0.00082	0.0	-0.00082
7	-0.88819	0.29148	-13.70313	0.00000	0.0	-0.00000
8	-1.00518	0.26792	5.06314	0.99371	1.00000	0.00629
9	-8.85617	0.00014	-8.14532	0.00029	0.0	-0.00029
10	0.57829	0.64067	-12.30045	0.00000	0.0	-0.00000
11	8.14480	0.99971				
12	0.12951	0.53233				

ID = 9

	U	O	V	FIG	T	DIF
1	9.91761	0.99995	-5.57904	0.00376	0.0	-0.00376
2	2.59056	0.93025	-16.73674	0.00000	0.0	-0.00000
3	-5.33531	0.00480	-6.22019	0.00198	0.0	-0.00198
4	3.75926	0.97723	-5.47621	0.00417	0.0	-0.00417
5	7.84881	0.99961	-6.70343	0.00123	0.0	-0.00123
6	-5.14065	0.00582	-5.63893	0.00354	0.0	-0.00354
7	-2.07510	0.11134	-8.97261	0.00013	0.0	-0.00013
8	-0.22885	0.44304	-6.79982	0.00111	0.0	-0.00111
9	-2.92041	0.05115	5.02112	0.99345	1.00000	0.00655
10	-2.51532	0.07479	-12.02249	0.00001	0.0	-0.00001
11	7.34694	0.99936				
12	-1.76498	0.14617				

ID = 10

	U	O	V	FIG	T	DIF
1	10.71221	0.99998	-17.20576	0.00000	0.0	-0.00000
2	8.59820	0.99982	-7.28160	0.00069	0.0	-0.00069
3	0.91894	0.71483	-9.95862	0.00005	0.0	-0.00005
4	5.84632	0.99712	-11.86967	0.00001	0.0	-0.00001
5	-2.90432	0.05194	-10.59387	0.00003	0.0	-0.00003
6	-4.49980	0.01099	-5.50674	0.00404	0.0	-0.00404
7	4.11872	0.98400	-8.03529	0.00032	0.0	-0.00032
8	-0.07051	0.48238	-10.82315	0.00002	0.0	-0.00002
9	10.85754	0.99998	-12.44148	0.00000	0.0	-0.00000
10	4.62943	0.99033	5.38217	0.99542	1.00000	0.00458
11	-3.40618	0.03210				
12	1.29717	0.78536				

Fig. 3.5(a) Recognition status after 5,000 learning cycles ($w=0.0$, $\eta=4.0$).

ID = 11

	U	O	V	FIG	T	DIF
1	-1.78510	0.14367	5.48968	0.99589	1.00000	0.00411
2	-3.96664	0.01858	-9.79727	0.00006	0.0	-0.00006
3	-3.87315	0.02037	-5.91510	0.00269	0.0	-0.00269
4	9.25815	0.99991	-6.11243	0.00221	0.0	-0.00221
5	9.54193	0.99993	-6.48916	0.00152	0.0	-0.00152
6	9.56619	0.99993	-9.20031	0.00010	0.0	-0.00010
7	-3.45870	0.03051	-6.14946	0.00213	0.0	-0.00213
8	-3.24570	0.03748	-6.42723	0.00161	0.0	-0.00161
9	-3.02470	0.04632	-6.03121	0.00240	0.0	-0.00240
10	-2.51538	0.07479	-18.08049	0.00000	0.0	-0.00000
11	3.83301	0.97881				
12	-0.63274	0.34689				

ID = 12

	U	O	V	FIG	T	DIF
1	6.64517	0.99870	-8.60590	0.00018	0.0	-0.00018
2	-3.96665	0.01858	5.83479	0.99708	1.00000	0.00292
3	5.47325	0.99582	-6.38923	0.00168	0.0	-0.00168
4	4.75406	0.99146	-7.28474	0.00069	0.0	-0.00069
5	-4.46967	0.01132	-6.32507	0.00179	0.0	-0.00179
6	4.93544	0.99286	-19.17334	0.00000	0.0	-0.00000
7	-0.88762	0.29160	-6.69631	0.00123	0.0	-0.00123
8	1.91178	0.87122	-7.92501	0.00036	0.0	-0.00036
9	-7.16538	0.00077	-19.35091	0.00000	0.0	-0.00000
10	-2.51537	0.07479	-8.50006	0.00020	0.0	-0.00020
11	-3.40620	0.03210				
12	0.93669	0.71843				

ID = 13

	U	O	V	FIG	T	DIF
1	-3.77068	0.02252	-6.17448	0.00208	0.0	-0.00208
2	-3.93453	0.01918	-10.99274	0.00002	0.0	-0.00002
3	-0.15465	0.46142	6.07737	0.99771	1.00000	0.00229
4	5.90785	0.99729	-13.49346	0.00000	0.0	-0.00000
5	7.95174	0.99965	-5.49356	0.00410	0.0	-0.00410
6	-4.49981	0.01099	-13.41061	0.00000	0.0	-0.00000
7	4.11864	0.98399	-12.07868	0.00001	0.0	-0.00001
8	5.69950	0.99666	-6.33585	0.00177	0.0	-0.00177
9	-4.89214	0.00745	-6.32151	0.00147	0.0	-0.00147
10	-2.51537	0.07479	-12.59784	0.00000	0.0	-0.00000
11	9.01521	0.99988				
12	1.29725	0.78537				

ID = 14

	U	O	V	FIG	T	DIF
1	3.15516	0.95911	-8.55134	0.00019	0.0	-0.00019
2	13.88561	1.00000	-7.47161	0.00057	0.0	-0.00057
3	-2.81756	0.05638	-12.54264	0.00000	0.0	-0.00000
4	-9.60900	0.00007	5.30886	0.99508	1.00000	0.00492
5	0.36311	0.58979	-12.72953	0.00000	0.0	-0.00000
6	12.69676	1.00000	-9.16942	0.00010	0.0	-0.00010
7	-2.37683	0.08496	-10.32514	0.00003	0.0	-0.00003
8	4.33284	0.98704	-12.59009	0.00000	0.0	-0.00000
9	-3.02469	0.04632	-9.36271	0.00009	0.0	-0.00009
10	5.57627	0.99623	-5.77572	0.00309	0.0	-0.00309
11	-0.64980	0.34303				
12	0.44921	0.61045				

ID = 15

	U	O	V	FIG	T	DIF
1	7.13745	0.99921	-10.40217	0.00003	0.0	-0.00003
2	-4.20427	0.01471	-8.37513	0.00023	0.0	-0.00023
3	-0.56129	0.36325	-8.25859	0.00026	0.0	-0.00026
4	-1.72903	0.15071	-7.00279	0.00091	0.0	-0.00091
5	0.45247	0.61123	7.54861	0.99947	1.00000	0.00053
6	-4.49980	0.01099	-6.00005	0.00247	0.0	-0.00247
7	-0.84311	0.30088	-7.09557	0.00083	0.0	-0.00083
8	3.90590	0.98027	-7.83180	0.00040	0.0	-0.00040
9	9.62403	0.99993	-8.24151	0.00026	0.0	-0.00026
10	-2.51534	0.07479	-6.46682	0.00155	0.0	-0.00155
11	-0.04937	0.48766				
12	-0.63274	0.34689				

ID = 16

	U	O	V	FIG	T	DIF
1	-1.78514	0.14367	-9.01505	0.00012	0.0	-0.00012
2	1.10646	0.75147	-14.82881	0.00000	0.0	-0.00000
3	5.06400	0.99372	-5.51862	0.00400	0.0	-0.00400
4	4.15992	0.98463	-11.88364	0.00001	0.0	-0.00001
5	6.82127	0.99891	-6.42400	0.00162	0.0	-0.00162
6	-7.16674	0.00077	5.75473	0.99684	1.00000	0.00316
7	0.97742	0.72660	-17.51488	0.00000	0.0	-0.00000
8	0.33560	0.58312	-5.78483	0.00306	0.0	-0.00306
9	10.75328	0.99998	-11.06906	0.00002	0.0	-0.00002
10	3.61401	0.97376	-5.71288	0.00329	0.0	-0.00329
11	6.31943	0.99820				
12	-1.53344	0.17749				

ID = 17

	U	O	V	FIG	T	DIF
1	6.25651	0.99809	-5.85938	0.00284	0.0	-0.00284
2	-3.96661	0.01859	-5.13223	0.00587	0.0	-0.00587
3	-3.87321	0.02037	-10.22298	0.00004	0.0	-0.00004
4	5.21396	0.99459	-9.67879	0.00006	0.0	-0.00006
5	-1.54238	0.17619	-8.31477	0.00024	0.0	-0.00024
6	11.83990	0.99999	-20.97192	0.00000	0.0	-0.00000
7	5.33881	0.99522	4.94263	0.99292	1.00000	0.00708
8	-3.24571	0.03748	-9.18374	0.00010	0.0	-0.00010
9	-3.02471	0.04632	-9.20164	0.00010	0.0	-0.00010
10	-2.51534	0.07479	-10.68214	0.00002	0.0	-0.00002
11	-0.64943	0.34312				
12	0.08826	0.52205				

ID = 18

	U	O	V	FIG	T	DIF
1	6.17945	0.99793	-6.51873	0.00147	0.0	-0.00147
2	-4.04580	0.01719	-11.22079	0.00001	0.0	-0.00001
3	2.76034	0.94050	-8.65574	0.00017	0.0	-0.00017
4	3.69792	0.97582	-11.06373	0.00002	0.0	-0.00002
5	10.15687	0.99996	-7.09215	0.00083	0.0	-0.00083
6	-11.71009	0.00001	-5.73168	0.00323	0.0	-0.00323
7	0.09206	0.52300	-10.90024	0.00002	0.0	-0.00002
8	-4.48618	0.01114	5.91766	0.99732	1.00000	0.00268
9	-3.02470	0.04632	-7.26255	0.00070	0.0	-0.00070
10	0.95158	0.72143	-11.88952	0.00001	0.0	-0.00001
11	7.57855	0.99949				
12	-0.83570	0.30244				

ID = 19

	U	O	V	FIG	T	DIF
1	4.66734	0.99069	-8.51119	0.00020	0.0	-0.00020
2	2.59132	0.93030	-19.03499	0.00000	0.0	-0.00000
3	-3.33539	0.00479	-6.83681	0.00107	0.0	-0.00107
4	0.51791	0.62666	-7.44798	0.00058	0.0	-0.00058
5	4.71806	0.99115	-6.57973	0.00139	0.0	-0.00139
6	-9.09428	0.00011	-7.88338	0.00038	0.0	-0.00038
7	2.17200	0.89771	-5.37120	0.00463	0.0	-0.00463
8	-2.71583	0.06205	-5.74029	0.00320	0.0	-0.00320
9	-3.02471	0.04632	6.04507	0.99764	1.00000	0.00236
10	-2.51534	0.07479	-10.01584	0.00004	0.0	-0.00004
11	7.34690	0.99936				
12	-2.61401	0.06824				

ID = 20

	U	O	V	FIG	T	DIF
1	0.68305	0.66487	-16.61339	0.00000	0.0	-0.00000
2	0.99577	0.73023	-6.19669	0.00203	0.0	-0.00203
3	-0.06975	0.48257	-7.20963	0.00074	0.0	-0.00074
4	-5.23851	0.00528	-5.43751	0.00433	0.0	-0.00433
5	-2.90434	0.05194	-6.71974	0.00121	0.0	-0.00121
6	-11.60779	0.00001	-5.85280	0.00286	0.0	-0.00286
7	3.25535	0.96287	-8.12795	0.00030	0.0	-0.00030
8	2.03955	0.88489	-10.72122	0.00002	0.0	-0.00002
9	1.42222	0.80569	-12.40314	0.00000	0.0	-0.00000
10	4.23310	0.98570	5.69713	0.99666	1.00000	0.00334
11	-3.40616	0.03210				
12	0.93643	0.71838				

Fig. 3.5(b) Recognition status after 5,000 learning cycles ($w=0.0$, $\eta=4.0$).

*** ITERATION =5000 ***

ID = 20

	U	O	V	FIG	T	DIF
1	0.68505	0.66487	-16.61339	0.00000	0.0	-0.00000
2	0.99577	0.73023	-6.19669	0.00203	0.0	-0.00203
3	-0.06975	0.48257	-7.20963	0.00074	0.0	-0.00074
4	-5.23851	0.00528	-5.43751	0.00433	0.0	-0.00433
5	-2.90434	0.05194	-6.71974	0.00121	0.0	-0.00121
6	-11.60779	0.00001	-5.85280	0.00286	0.0	-0.00286
7	3.25555	0.96287	-8.12795	0.00030	0.0	-0.00030
8	2.03955	0.88489	-10.72122	0.00002	0.0	-0.00002
9	1.42222	0.80569	-12.40314	0.00000	0.0	-0.00000
10	4.23310	0.98570	5.69713	0.99666	1.00000	0.00334
11	-3.40616	0.03210				
12	0.93643	0.71838				

	1	2	3	4	5	6	7	8	9	10	11	12
*WTI	1.78509	3.96658	5.33538	-0.50657	2.90434	1.00036	3.45872	3.24572	3.02468	2.51535	3.40617	0.63294
*WTO	0.17470	2.22336	2.75221	1.84875	2.40809	5.43431	-0.42786	3.34721	3.91659	1.95723		
*W10	3.03569	1.01017	0.53671	1.15565								
*W11	1.20685	0.07255	2.97991	0.87041								
*W12	0.36059	0.87790	-0.90069	-0.09316								

*WI

0.50656	-1.98554	1.18329	1.51212	3.73818	2.47017
-1.73623	-0.07921	4.98317	0.03229	3.03569	1.01017
0.53671	1.15565	1.12813	1.05560	2.68322	1.46210
-0.06152	2.18526	-3.24155	1.15380	-0.26785	0.45983
1.20685	0.07255	2.97991	0.87041	1.73572	-1.56532
3.13056	-0.05115	2.15273	-2.66695	-0.85686	-0.89285
0.63429	1.34713	1.82042	-1.13677	0.36059	0.87790
-0.90069	-0.09316	0.56106	0.00150	2.11008	1.67916
0.10431	1.27113	4.13423	-0.77009	-1.52604	-1.08862

*WO

-4.74978	-0.98725	-6.92218	0.40620	1.85867	-3.96322	2.77279	1.92902	1.70465	-1.36086
-2.79787	-2.77420	0.68203	2.66630	-4.85308	3.37514	-4.74901	-5.30345	4.62228	4.08180
-2.57507	5.59766	1.51162	-1.93256	1.31798	2.28538	-8.16144	5.03194	-6.86993	1.09040
1.24018	-0.75083	0.77236	-5.55637	-4.22361	-1.22042	1.67991	-1.48030	-0.70842	-2.42629
2.10367	-5.66511	-0.34755	-0.69049	1.68034	1.65060	-4.64079	1.41136	2.21462	-4.02942
2.71081	3.01977	-4.53792	2.85954	-1.70812	-3.84078	1.19317	-4.78469	-6.26442	-4.58580
-5.00986	-2.70416	1.27006	-4.50259	-1.22934	-4.98122	3.72390	-1.80004	-0.51358	0.92331
-3.63940	0.77796	3.01004	1.37603	1.26262	-1.99067	-2.33286	-4.88027	-2.87005	-1.44627
-1.18577	-1.94552	-3.80297	-3.06578	7.24434	6.03419	-0.20531	-4.32093	-2.85479	2.44961
-0.79843	-1.81671	-4.83326	2.14266	-7.24137	4.60889	-0.64650	2.95313	-3.97950	2.24446
0.99405	-4.75683	0.93359	-0.24302	0.83829	1.83961	-3.36484	2.10263	3.38772	-6.18191
-0.49565	2.41090	3.95766	-1.84427	-1.89859	-6.03625	-4.69321	-1.00633	-0.89529	2.33577

Fig. 3.6 Weights and thresholds after 5,000 learning cycles (w=0.0, η=4.0).

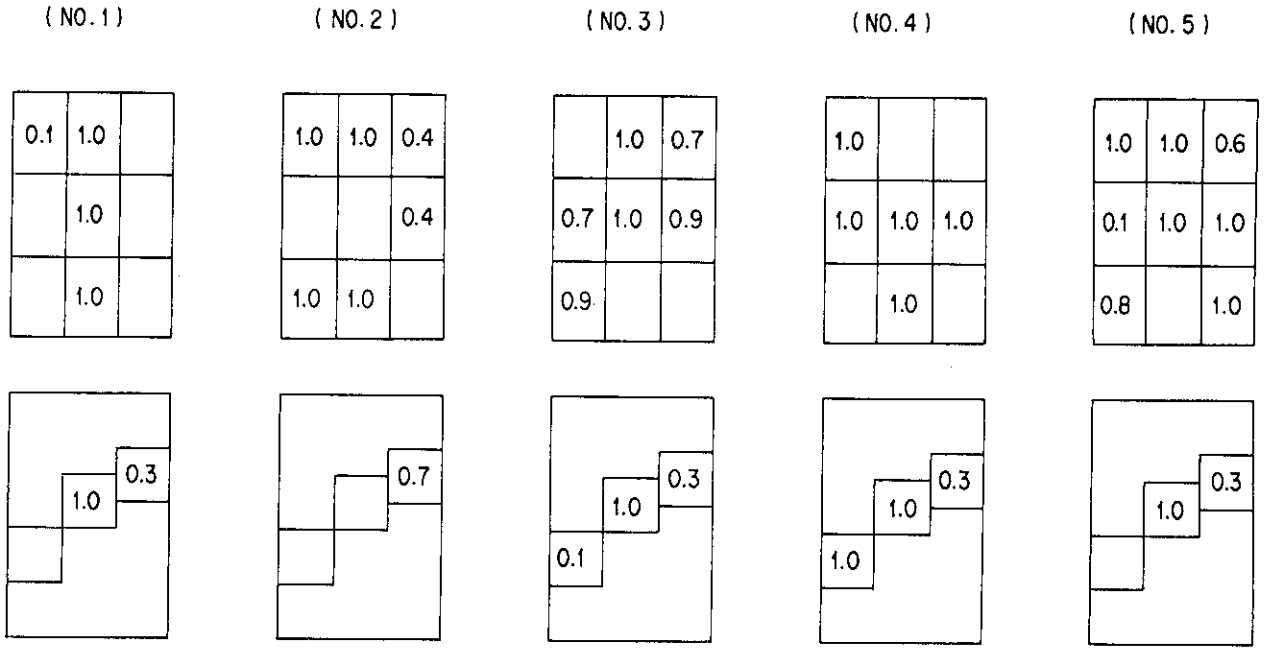


Fig. 3.7(a) Values of O after 5,000 learning cycles (w=0.0, η=4.0).

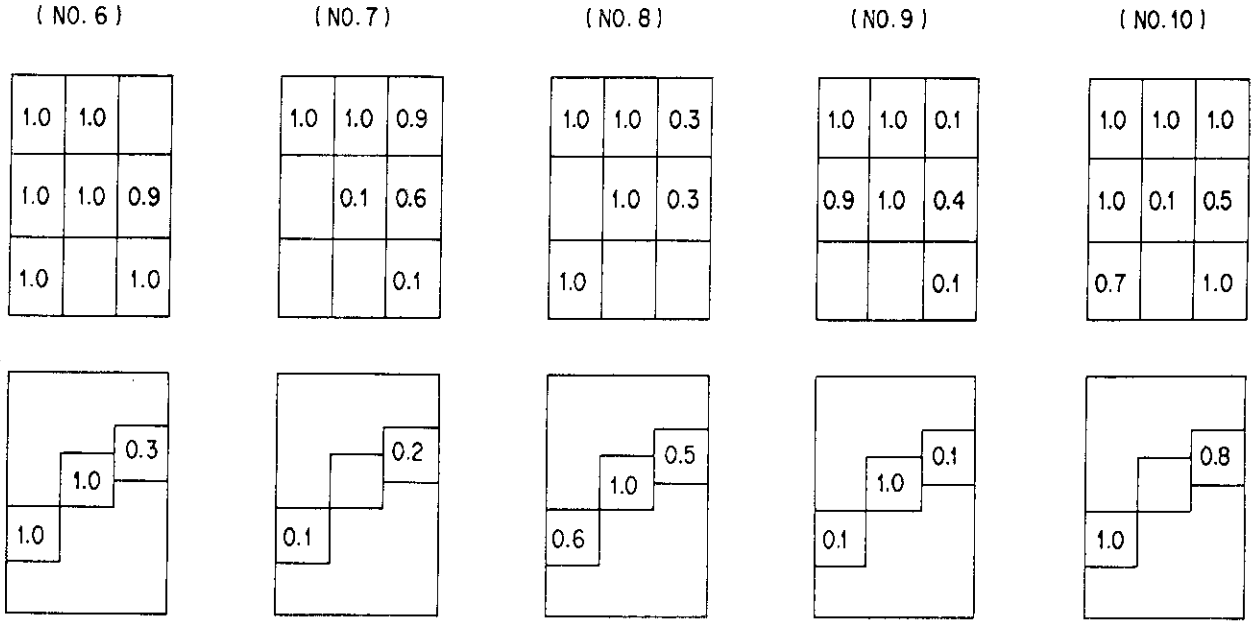


Fig. 3.7(b) Values of O after 5,000 learning cycles (w=0.0, η=4.0).

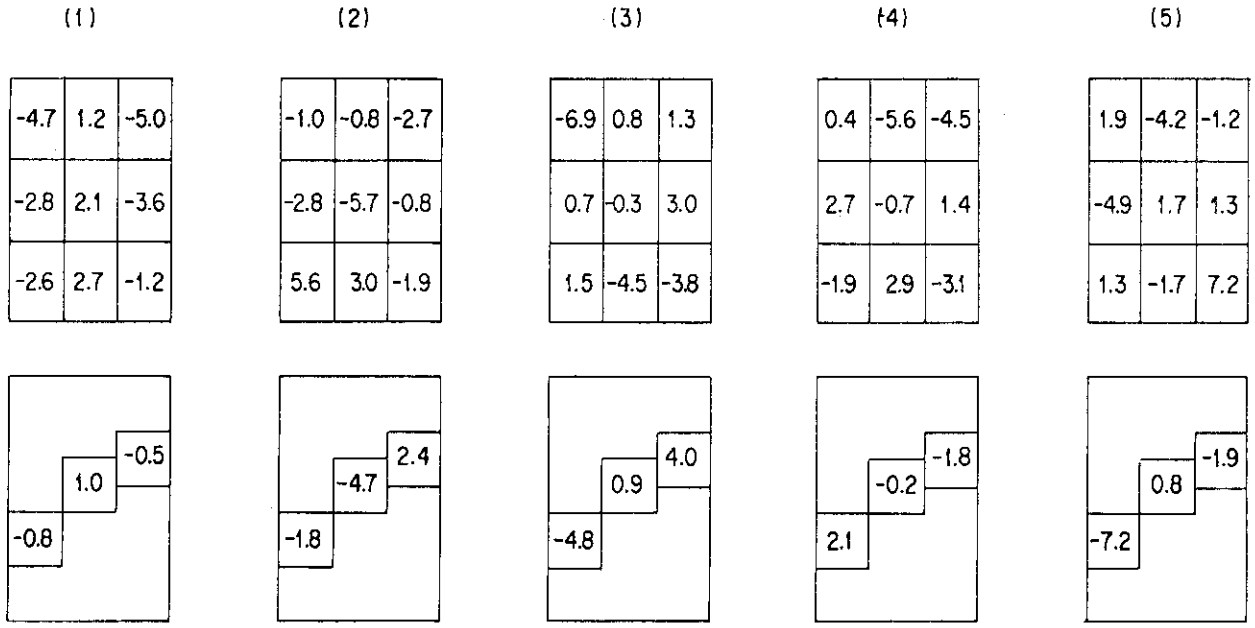


Fig. 3.8(a) Values of W0 after 5,000 learning cycles ($w=0.0$, $\eta=4.0$).

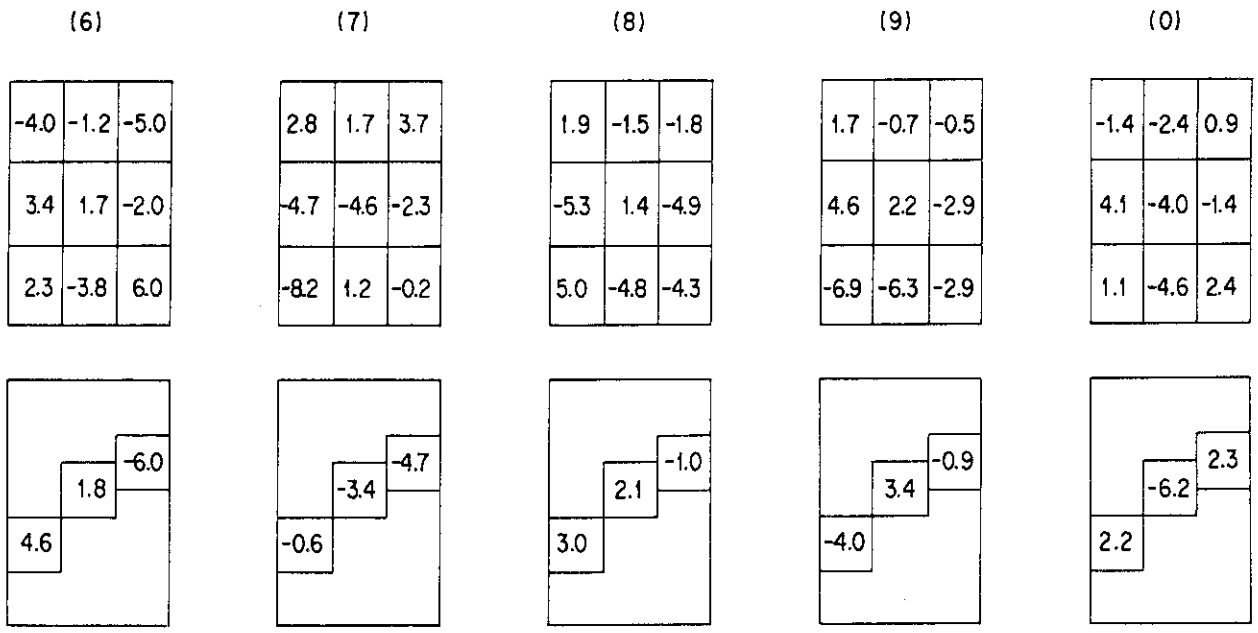


Fig. 3.8(b) Values of W0 after 5,000 learning cycles ($w=0.0$, $\eta=4.0$).

（中間層）

ユニット： j	出力信号：O（1～12）			重み：WO（1～12，4）	
1	1.0	(0.99921)	*	0.4	(0.40630)
2	1.0	(1.00000)	*	2.7	(2.66630)
3	0.0	(0.00480)	*	-1.9	(-1.93256)
4	0.0	(0.00526)	*	-5.6	(-5.55637)
5	1.0	(0.99936)	*	-0.7	(-0.69049)
6	1.0	(0.99961)	*	2.9	(2.85954)
7	0.0	(0.03051)	*	-4.5	(-4.50259)
8	1.0	(0.98705)	*	1.4	(1.37603)
9	0.0	(0.04632)	*	-3.1	(-3.06578)
10	1.0	(0.99623)	*	2.1	(2.14266)
11	1.0	(0.99164)	*	-0.2	(-0.24302)
12	0.3	(0.34694)	*	-1.8	(-1.84427)

$$\text{総和} = \sum_{j} O(j) * WO(j) = 7.53441$$

$$V(4) = \text{総和} - \text{しきい値} : WTO(4)$$

$$= 7.53441 - 1.84875$$

$$= 5.68566$$

$$FIG(4) = 1.0 / (1 + \exp(-V(4)))$$

$$= 0.99662$$

Fig. 3.9 Calculation process for the input pattern No.4.

3.3 今後の計画

(1) S63年度作業

ニューラル・ネットワークに関する理論を画像認識に応用する試みを今年度から開始した。今年度は，パターン認識に有効性を発揮しているBPモデル，コグニトロン，ネオコグニトロンの手法を会得することを主目的とし，各手法を使った簡単なパターン認識実験に取り組んでいる。BPモデルについては，手書き数字の簡単な認識実験を行い，3.2節にその概要を示した。今年度末までにネオコグニトロンの手法を使用した簡単な画像認識実験を行う予定である。

現段階では以下のことがわかっている。BPモデルは比較的簡単にプログラムが作成でき，簡単なパターン認識に対しては学習結果も良好である。しかし，BPモデルは提示するパターンを多数の画素の濃淡情報としてしか利用していないため，提示するパターンが学習させたパターンと同一でも位置をずらしたり，大きさを少し変えて示した場合は同一のものと認識できないことが多いという欠点をもっている。これに対して，ネオコグニトロンの手法では，位置ずれや大きさの変化を吸収できるが，モデル化が難しく，学習のさせ方も試行錯誤的でかなりの時間を要する。ネオコグニトロンでは，提示するパターンを多数の画素の濃淡情報として受け取り，隣接した画素の濃淡の関係から直線，カーブ，かど（多角形の頂点など），交点などの特徴を抽出することが可能である。また，直線の傾きも認識可能である。しかし，これらの特徴ひとつ（たとえば直線の傾きを12段階で判別するためには12個の特徴として表現する）ひとつに対して，モデル化と学習過程が必要なので，各種のパターンを判別するためには気の遠くなるような作業も必要となる。

ニューラル・ネットワークを使用したパターン認識では，現在の所ネオコグニトロンが最も認識能力を秘めている（抽出すべき特徴をひとつずつ組み込むことが可能，ただし使用例としてはまだ数字0～9の認識例しかない）ため，本研究ではこれを画像認識に応用すること試みている。提示する画像は，Fig.3.10に示す基本パターンをぼかしたり，位置ずれ，縮小させたものを考えている。基本パターンは24通りあり，正方形と円の中に時計の針を模擬した矢印がひとつあり，12方向のいずれかを指したものである。

(2) 平成元年度作業

人間動作シミュレーション技術の研究では，ロボットの視覚情報は，視覚センサーから得られたある程度ぼけた白黒画像とTVカメラで得られたカラーの鮮朝画像の両方を対象としている。前者は現在ロボットに搭載されている視覚センサーで得られる画像，後者は将来ロボットに搭載可能になる視覚センサーで得られる画像を想定している。

ロボットは得られた画像から自分の周りの環境を認識するとともに作業対象物やその状態などを認識していく必要がある。これらの認識をすべてニューロ手法で処理することは難しく，既存の認識手法と組み合せてより効率的で強力な認識技術を開発していかねばならない。

そこで，元年度は施設形状データベースの映像化から得られる鮮明なカラー画像を使って，両眼視（2つの視点から得られる）データから立体画像を生成し，各画素に映ってい

る物体の色（ＲＧＢ）情報と物体までの距離情報を使って，対象物の切り出し，特徴抽出等を行う計画である。また，前年度にひきつづき，各種ニューロ手法による画像認識能力の調査，分析も行う計画である。

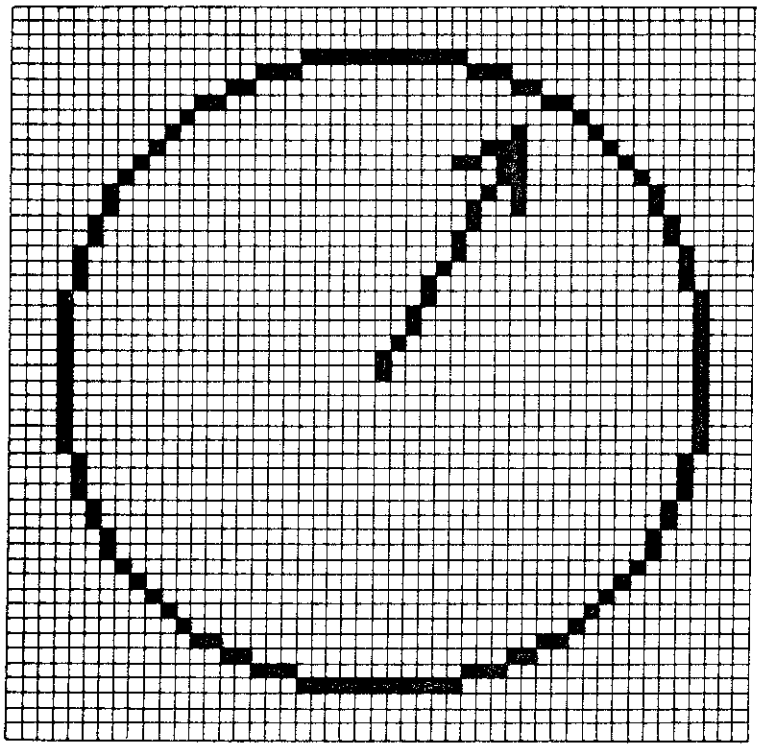


Fig. 3.10(a) Learning pattern (1) for the image recognition experiment.

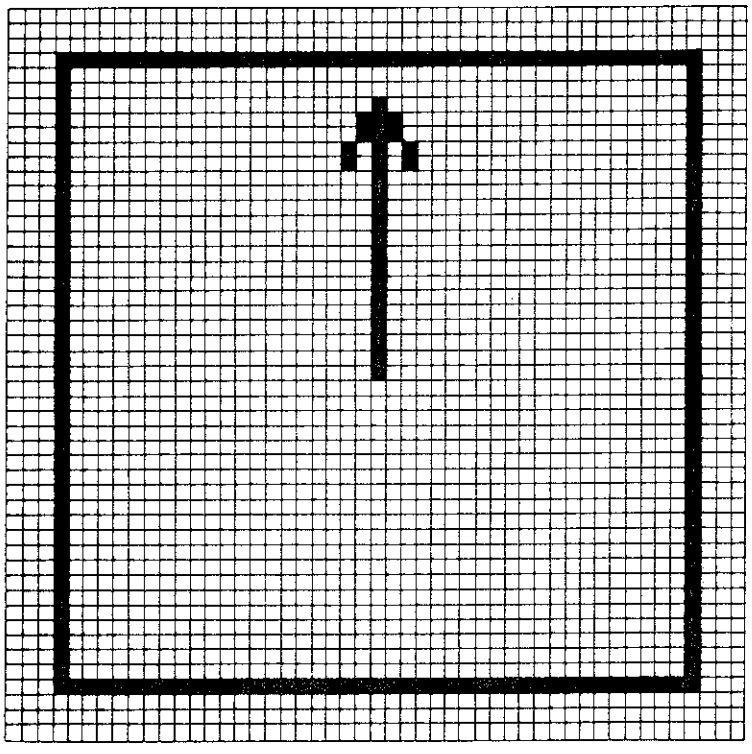


Fig. 3.10(b) Learning pattern (2) for the image recognition experiment.

3.4 おわりに

ニューラル・ネットワーク・モデルによる画像認識研究は，１０月から本格的にスタートしているため，まだ既存の研究を学習し，画像処理分野に適用していく段階がしばらく必要であると思われる。ＮＨＫで研究されているネオコグニトロンの研究も昭和４６年のコグニトロン開発以降の積み重ねからきているが，現在でもまだ手書き数字０～９の認識に関する分野をいろいろな角度から追求しておられるのを見てもこの分野の難しさは理解できよう。

この画像認識研究ではニューラル・ネットワーク技術とニューラル・ネットワーク以外の既存技術の良い点を生かして，視覚情報をより簡易な多層モデルによって的確に認識できるようにしていきたい。

参考文献

1) 中村正弘：輝きを見せはじめたニューロ・コンピュータ，日経コンピュータ，No.169，pp 87－105 （1988．3）．

2) 緑川博子：バックプロパゲーションによる顔画像認識の一考察，情報処理第３６回全国大会，pp．1881－1882 （1988）．

3) 福島邦彦：自己組織機能を持つ多層神経回路，信学論，Vol．J58－D，No.9，pp．530－537 （1975．9）．

4) 福島邦彦：位置ずれに影響されないパターン認識機構の神経回路網モデル，信学論，Vol．J62－A，No.10，pp．658－665 （1979．10）．

5) Fukushima，K．：A Hierarchical Neural Network Capable of Visual Pattern Recognition，Neural Networks，Vol．1，pp．119－130 （1988）．

6) 福島邦彦：視覚パターン認識における選択的注意機構の神経回路網モデル，信学論，Vol．J69－D，No.6，pp．983－1003 （1986．6）．

7) Rumelhart，D．E．，et al．：Learning Internal Representations by Error Propagation，Parallel Distributed Processing，Vol．1，pp．318－364，MIT Press，Massachusetts（1987）．

8) 合原一幸：ニューラル・コンピュータ，東京電機大学出版局，東京 （1988）．

9) 麻生英樹：ニューロ・コンピューティング（原理と概要），情報処理，Vol．29，pp．966－973 （1988．9）．

4. 二足歩行ロボット運動学的シミュレーション

4.1 はじめに

６２年度開始された研究テーマ"人間動作シミュレーション"の研究課題の一つである二足歩行ロボット運動学的シミュレーションについて，６３年度前半の作業内容を報告する。

６３年度の計画としては昨年整備したロボット運動学計算プログラムの発展形としての腕を自由にした歩行モデルに対する二足歩行運動学計算プログラムの開発，Cellular Array Processor（CAP）を使った歩行動作の映像化（Photo.4.1），その他ロボット運動学に関連した調査等がある。今年度前半においては，ＣＡＰを使った歩行動作の映像化のための歩容の多様化に主に取り組んできた。プログラムの開発については今年度後半に行い，ロボット運動学に関連した調査については大学に依頼した。

なお，二足歩行ロボット運動学方程式，二足歩行計算プログラム，作画プログラム等については，文献1)においてすでに発表ずみであるので今回これらに関する説明は省略する。

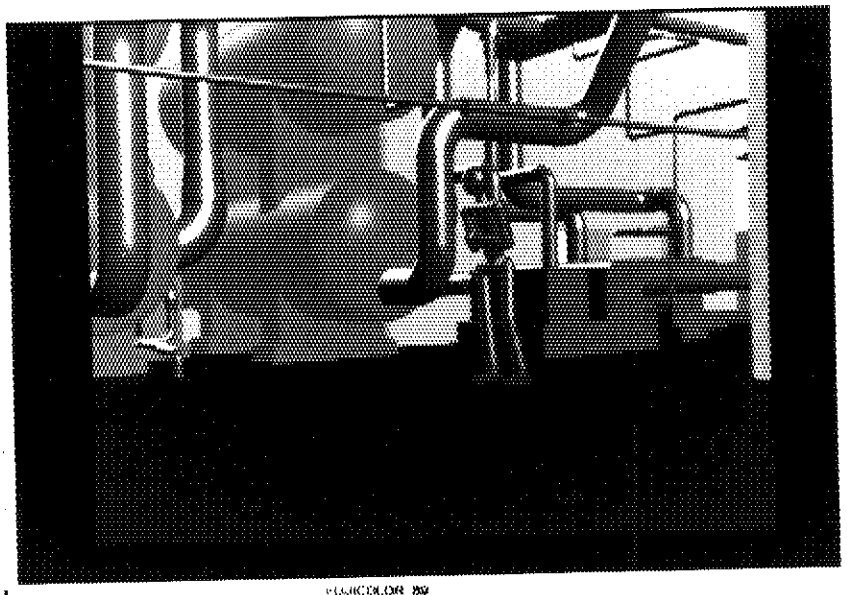


Photo.4.1 Color image of human biped locomotion.

4.2 歩行モデル及び入力データ

本研究で用いている歩行モデルはFig.4.1, Fig.4.2 に示すように人間を7つの関節と12のボディを持つ，腕と首固定のロボットで表現したものである。簡単な歩行動作に関する運動学方程式を解き関節の位置・角度の時間的変化を計算し，計算結果に基づき歩行動作を映像化する。この場合基本歩行パターンとして下肢の関節角の角度 β_{1L}, β_{2L}, β_{3L} を入力し，腰の曲がる角度 ϕ と脚の左右の傾き θ を解として求める。この下肢の関節角のデータは文献2)に折れ線グラフで掲載されているものを用いている。このデータより以下のような形式でデータファイルを作成し入力データとした。なお，歩行は周期的であることを前提としているため左脚のデータのみを入力する。また，本文中脚はLegを，足はFootを意味する。

Line 1	1周期に要する時間（sec.）
Line 2	揺動期の開始タイムステップ番号*
	支持期の開始タイムステップ番号
Line 3	コメント
Line 4	折れ線グラフで基本歩行パターンを与える場合の折れ点の総数
Line 5	折れ点となるタイムステップ番号
Line 6	折れ点での角度座標
Line 7, 8, 9, 10 以下同じ	次の基本歩行パターンをLine 3, 4, 5, 6 と同様に入力する

* 1周期を80分割し，時間メッシュ番号を1, 2, ……, 81とした。

また，Zero Moment Point（Z.M.P.）の移動に関しては支持足のX軸上にあると仮定し，Fig.4.3 に従って T_1, T_2, β_1, β_2 の入力データを与えた。

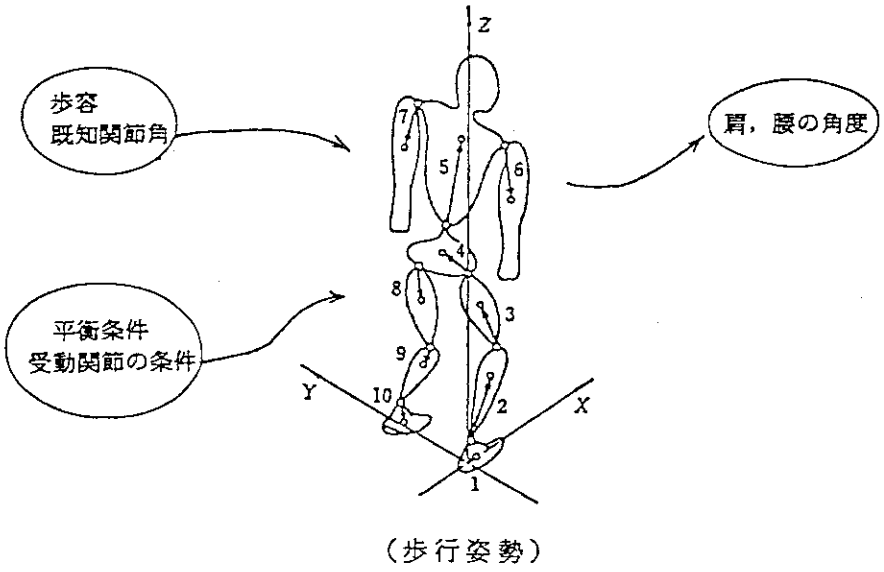


Fig. 4.1 Simulation for human biped locomotion.

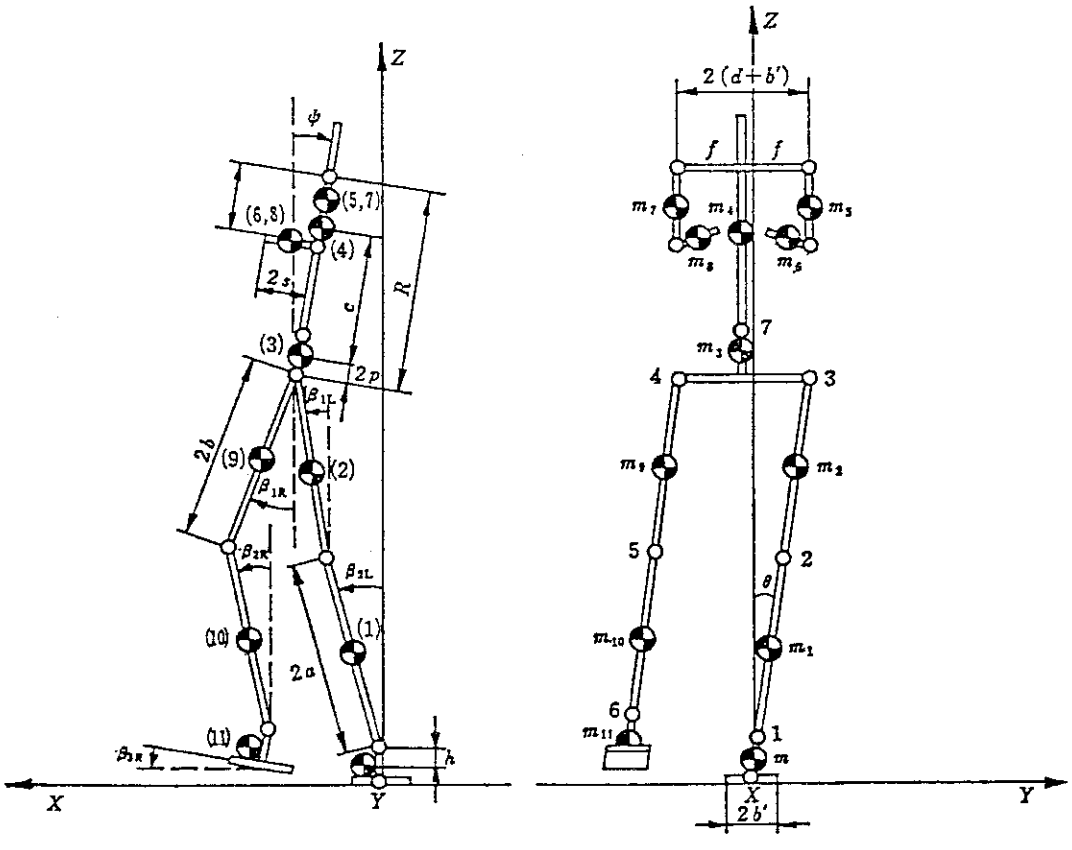


Fig. 4.2 Configuration of human biped locomotion model.

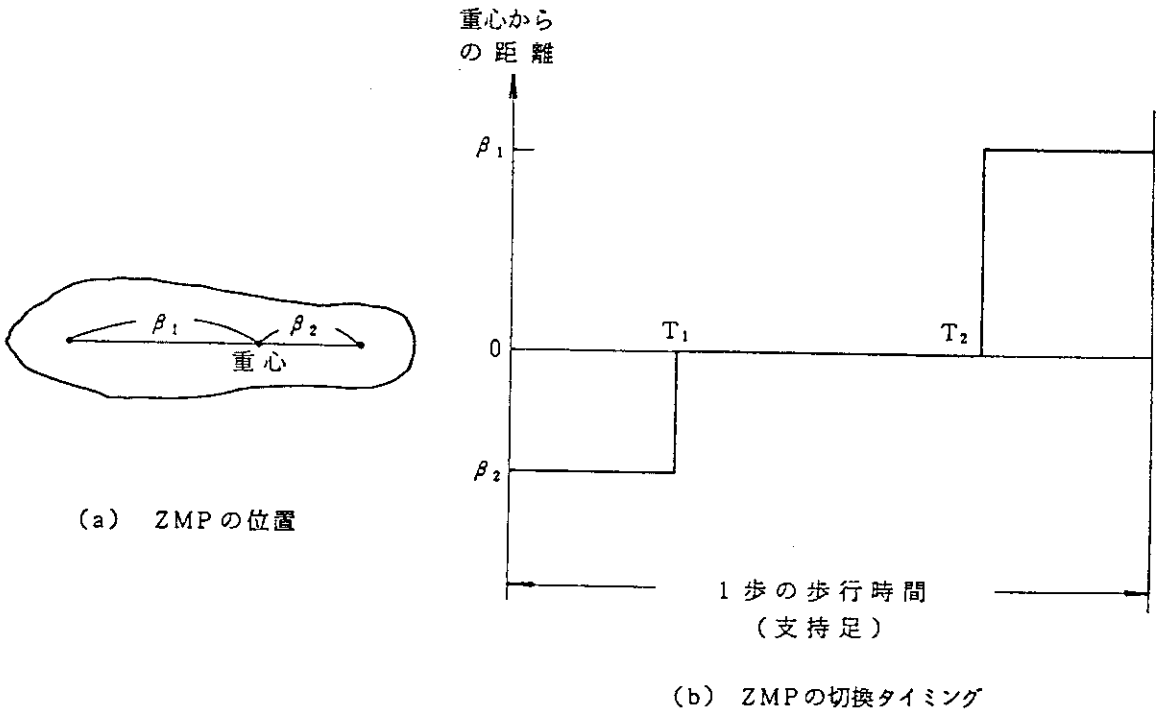


Fig. 4.3 Timing of Z.M.P. transfer.

4.3 歩容の多様化

4.3.1 階段歩行

歩容の多様化としてまず階段歩行を取り上げる。階段歩行に関しては文献2)に上り・下りとも各二種類づつの下肢関節角に対するデータが掲載されている。しかし，実際の歩行シミュレーションのためにはいくつかの問題点があった。その問題点と解決策を以下に示す。

① Z.M.P.の移動に関するデータが記載されていない等，データに関して不明確なところが多い上，現データのみでは階段の高さや幅に関する自由度が小さく，特定の階段に対する歩容しか実現できないという欠点がある。この解決方法については4.3.5節に述べる。

② 数値積分の際の入力初期値についても通常の平地歩行以上に初期値にセンシティブであり，仮に収束したとしても人間の姿勢としては不自然な解（方程式の解として得られる腰や脚の左右のふれの角度が人間の姿勢として不自然であるということ）が得られる場合が多い。そのためZ.M.P.の移動に関するデータの推測と合わせてかなりの時間を試行錯誤に費やした。

③ 画像は二次元であり視点が固定されているため，画像からだけでは左右の足の高さの変化がよくわからない（左足支持の場合と右足支持の場合で足の高さの差が異なるように見える）。計算結果では足の高さの差は左右一致していることから，視点をボディのいづれかのセグメント（プログラムでどのセグメントにするかを指定する，例えば腰や左足）の重心を常に画面の中心に見ているようにプログラムを修正した結果，画像からも左右の足の高さの差が左足支持の場合と右足支持の場合で等しくなっていることが認められた。

これとは別に昨年度の作画方法にミスがあったためこれを修正をした。そのため以下に示す画像は昨年度のものとは異なっている。なお，視点は以前と同じ視点を固定する方式に戻した。

(1) 階段上り1（データ名：GAITA）

腕を固定し足を常に水平に保ち階段を上がる場合。この時の下肢の関節角の入力データ及びZ.M.P.の移動に関する入力データをTable 4.1に，その計算結果の画像をFig. 4.4に示す。ここで図は歩行モデルを横から見た場合である。

(2) 階段下り1（データ名：GAITB）

腕を固定し足を常に水平に保ち階段を下る場合。この時の下肢の関節角の入力データ及びZ.M.P.の移動に関する入力データをTable 4.2に，その計算結果の画像をFig.4.5に示す。なお，このデータは本来腕を自由にした場合のものである上，β_{2L}のデータにミスがあった。また，階段を下る場合足を水平に保つ（$\beta_{3L,R}=0$，Const.)のは不自然であるため，このデータによる画像はあまり適当であるとは言えない。

(3) 階段下り2（データ名：GAITD）

文献2)に記載されている階段下りのもう一つのデータには足の角度のデータが与えられており，このほうがより自然な歩容である。この場合の下肢の関節角の入力データ及びZ.M.P.の移動に関する入力データをTable 4.3に，その計算結果の画像をFig.4.6に示す。

(4) 階段上り2（データ名：GAITU）

(3)に合わせGAITDと同時に掲載されている階段上りのもう一つのデータを用いた場合における下肢の関節角の入力データ及びZ.M.P.の移動に関する入力データをTable 4.4に，その計算結果の画像をFig.4.7に示す。

4.3.2 停止・発進動作[3)～5)]

停止や発進に関する歩行動作を考える時，これらはこれまでの歩容と違い周期的でないため（現プログラムでは解析上周期的であることが前提），それぞれについてシミュレーションするのは困難である。まして，文献2)にも下肢関節角に対するデータ等は無い。

そこで停止・発進を一連の動きと考え，歩行→静止→歩行を1サイクルとしこれらを前半と後半に分割し，停止動作・発進動作とする。具体的には各歩容は0.5秒（1/4周期）近傍で下肢関節角が0に近くなる（つまり直立状態）。このことを利用して各歩容を0.5秒で前後に分割し，間に静止データ（β_{1L}～β_{3L}＝0）を挿入し一連のデータとする。これを各歩容（平地歩行，階段歩行）について作成したが，階段歩行については解が収束せずこのプログラムで計算するのは無理がある。ここでは平地歩行における停止・発進動作のみを取り上げる。

Table 4.5に下肢関節角の入力データ及びZ.M.P.の移動に関する入力データ（データ名：GAITVS）を示す。

本研究では歩容を単脚支持で揺動期と支持期の繰り返しと考えているため，停止中片足状態のため脚がY軸方向に傾いてしまう（傾斜角θ，Fig.4.2参照）。ここで静止状態について考えてみる。まず全ての下肢の関節角を0としたデータ（データ名：GAITS1，Table 4.6）を作成する。この場合常に直立状態ではあるがやはりY軸方向への傾きを持つ。そこで時間ファクター（PER）すなわち周期を変更することによって傾きを変化させる。周期をGAITVSと同様に設定したGAITS1の場合Y軸方向の傾きはGAITVSの場合とほぼ一致した。周期を小さくするに従い傾きも小さくなり，周期が0.2秒（データ名：GAITS）で傾きはほとんど0（Max.0.0029 rad）となった。

これらを利用して，この各データを組み合わせることで停止・発進の動作が作成できる。つまり，静止状態から歩きだしまた停止する場合，

静止(GAITS)→つなぎ(GAITS1)→歩き始め(GAITVS)→平地歩行(GAITV**)
→止まり始め(GAITVS)→つなぎ(GAITS1)→静止(GAITS)

** 昨年度の平地歩行のデータ

のように各データをつなぎ合わせ，接続点での姿勢が一致するようにデータを作成し形にあった必要な時間だけ画像を取り出す。例えば1秒静止したのち4歩歩いて静止する場合，

静止(GAITS)→つなぎ(GAITS1)→歩き始め(GAITVS)→平地歩行(GAITV**)
0～1.0秒　1.0～1.3秒　1.3～2.0秒　0～2.0秒

→止まり始め(GAITVS)→つなぎ(GAITS1)→静止(GAITS)
0～0.3秒　0.7～1.0秒　0～1.0秒

とすればよい。この場合静止状態（完全な直立状態）から歩き始めまで1秒かかっている。この画像をFig.4.8に示す。図は歩行モデルを横から見た場合と正面から見た場合を同時に示し

たものである。

4.3.3 平地歩行における歩調・歩幅変化

平地歩行において1周期に要する時間（歩くペース，以下歩調と呼ぶ）・歩幅の変化を試みる。本研究ではプログラム上，時間ファクター（PER）の変更で歩調を，角度ファクター（SFO）の変更で歩幅を変化させることが可能である。

現在のデータは周期2秒（2歩分）で1歩の歩幅が約31cmである。これは通常の人間の歩行からすれば低速度であり歩幅もやや小さい。そのため歩調・歩幅をこれ以上小さく変化させることは人間の動作から考えるとあまり意味がない。

歩行速度を速くするには時間ファクターを小さくすることによって達成できるが，時間ファクターを小さくするに従って上半身のX軸方向の傾斜角（腰の曲がる角度ϕ，Fig.4.2参照）が大きくなり不自然な形となる。現データではPER＝0.7 すなわち周期1.4秒程度までの歩調しか実現できない。この場合の画像をFig.4.9に示す。

また，歩幅を大きくするためには角度ファクターを大きくすればよい。Table 4.7 に角度ファクターを変化させた場合の歩幅の変化を示す。ここでも時間ファクターを変化させた場合と同じ様に角度ファクターをあまり大きく（ここではSFO＞1.2）すると上半身の傾きが大きくなり人間の姿勢としては不自然な形となる。現データでは角度ファクターは0.8～1.2程度が望ましい。さらにこれらを組み合わせてPER＝0.7，SFO＝1.2とした場合は上半身のX軸方向の傾きが大きくなり不自然な形となる。

4.3.4 方向転換

方向転換，すなわち"曲がる"という動作についても左右の脚の動きが周期的でないため現段階ではシミュレーションできない。そこで今回は単に"曲がる"と言う動作について考察してみる（この"曲がる"という動作に関する二足歩行の研究文献[2)]はあまり見当らない）。

ごく日常的な自然な歩行を考えれば"曲がる"という動作を行う場合1歩目は内側の脚を曲がる方向に踏み出し，2歩目は外側の脚を単に内側の脚に合わせて従属的に動かしている（内側に踏み出すような動作は行っていない）。すなわちFig.4.10に示すように1歩目と2歩目の着地点は同じ向き，3歩目と4歩目，……，(2n−1)歩目と2n歩目（nは自然数）の着地点は同じ向きになる。つまり必ず奇数歩で方向が90度変わり，奇数歩目は内側の脚を曲がる方向に踏み出し，偶数歩目は外側の脚を単に内側の脚を軸として回転し内側の脚と同じ方向を向いて着地する，というような動作と考える事ができる。例えばFig.4.10では5歩で90度向きを変えているが1歩目で30度，3歩目で30度，5歩目で30度というように踏み出し方向を変えている。

先にも述べたが，"曲がる"という動作は研究上必要であるが現在の運動学方程式のみではシミュレーションできない。この動作のシミュレーションのためには新たな運動学方程式を作成する必要がある。

4.3.5　階段歩行における高さ・歩幅変化

4.3.1で用いたデータの場合階段の上りと下りでは歩幅や階段の高さが一致しない。また，ＣＡＰを用いて映像化する際の施設形状にも適合していない。そのため階段歩行における階段の高さと幅を調節することを試みた。データにはGAITDとGAITUを用いた。

まず始めに角度ファクターを変更した場合の高さと幅の変化をTable 4.8に示す。施設内の階段については高さが約２０cm，幅が約２５cm程度のものであるが場所によってまちまちであり，途中で高さの変化する物や勾配の急な物等があり，Table 4.8からもわかるようにこれらに適合した歩容を完成するのは非常に困難である。

単なる角度ファクターの変更だけでは希望の高さ・幅を達成することだけでなく上りと下りを同じ階段にすることも無理である。そこで角度ファクターを各関節角ごとに分けて変更できるように，また支持期と揺動期とに分けて変更できるようにプログラムを一部修正した。これにより上りと下りで階段の高さ・幅は一応一致した。しかし，施設形状にあったようなものはできず（高さ約１３cm，幅約３２cm），不自然な歩容であるうえに階段と足が干渉（融合）してしまうためこの方法もあまり適当であるとは言えない。

次に比較的データ変更の容易な階段上りの下肢関節角のデータを新規に作成し下りの高さ・幅に合わせることを試みた。モデルとするのはGAITDでSFO＝1.0，階段の高さ13.6cm,幅26.2cmのものである。これに合うようにGAITUのデータを参考にデータを作成したところ，若干の不自然さはあるものの高さ・幅とも一致し階段との干渉もない歩容ができた。このときの階段上りの下肢関節角のデータ及びZ.M.P.の移動に関する入力データ（データ名：GAITUU）をTable 4.9に，その計算結果の画像をFig.4.11に示す。

4.4　おわりに

以上これまでに行ってきた内容について述べてきたが，現在のプログラムによるシミュレーションではこれ以上のことは無理である。

研究を進めて行くに当たって，歩容に対するニーズは多岐に渡るが限定された歩容しか実現できないのが現状である。現在のプログラムで研究を進めていくにはデータ不足である上に本運動学方程式では対応のできない歩容や，仮に対応できても不自然な姿勢になってしまうもの，方程式の解が得られないもの等があり，これらに対する新たな運動学方程式の検討[6),7)]が必要である。そして現プログラムではどうしても限界があるため，プログラムの根本的改善等も必要になってくる。これらが今後研究を進めていく上での課題となるであろう。

また，　元年度としては以下のことを計画している。

(1)　歩行パターン生成・逆運動学計算に関する調査とプログラム開発

現プログラムでは，脚の動きは各関節角の動きとして実際の歩行動作のデータを入力しているが，このような人間の歩行パターンだけでなく人工的に歩行パターンを作成することを考える。つまり歩行パターンとして足の軌道を与え，その軌道から逆運動学計算を行い各関節角の動きを計算する。そのための逆運動学計算プログラムを開発する。逆運動学を解くに当たっては，冗長自由度の考慮や拘束条件の検討，軌道の制御等に関する調査が

必要である。また，Z.M.P.の移動方法を再検討し，足の軌道とZ.M.P.の移動に対する最適経路について考察する。

(2) 原研版二足歩行計算プログラムの発展の開発

現プログラムは単脚支持状態についてだけを考えており周期的な歩容のみしか扱えないそのため，本来停止や発進動作等のシミュレーションはできない。そこで両足支持状態を考慮に入れた運動学方程式を取り上げ，拘束条件や境界条件等を変更した計算プログラムを開発する。

(3) ロボット運動学に関連した調査

下記の項目について大学に調査を依頼する。

a) 両腕協調動作・指と腕の協調動作を表すロボット運動学方程式についての継続調査

b) "曲がる"動作について，人間的な場合とロボット的な場合に対するそれぞれの運動学方程式及び方法論に関する調査。

参考文献

1) 石黒美佐子・藤崎正英：二足歩行ロボット運動学的シミュレーション，JAERI-M 88-080，(1988)

2) M. Vukobratović（加藤一郎・山下忠　訳）：歩行ロボットと人工の足，日刊工業新聞社，(1985)

3) 山下忠・谷口隆雄：ヒトの歩行開始動作の解析，計測自動制御学会論文集，Vol.22，No.2・No.3，(1986)

4) 山下忠・谷口隆雄：ヒトの停止動作の解析，計測自動制御学会論文集，Vol.22，No.4 (1986)

5) 古荘純次：動的二足歩行ロボットの制御，日本ロボット学会誌，1巻3号，(1983)

6) 三浦宏文・下山勲：二足歩行ロボットの機構と制御，コンピュートロール，No.9，(1985)

7) 伊藤正美・成清辰雄：拘束のある二足歩行運動の解析と制御，日本ロボット学会誌，1巻3号，(1983)

Table 4.1 Go up stairs 1.

```
BROWSE - J4478.GAIT.DATA(GAITA) - 01.22 --------------- LINE 00000 COLS 001 080
COMMAND ===>                                                  SCROLL ===> PAGE
********************************* TOP OF DATA *********************-CAPS ON-***
  2.0
   13   53
 BETA1
   12
    1    9   13   19   32   41   43   47   51   63   68   81
  0.0        0.2442    0.2442    0.0942    0.1396    0.0      -0.5931  -0.8373
 -0.8373   -0.5652    -0.4187    0.0
 BETA2
   13
    1    9   15   19   29   41   42   45   48   59   60   71   81
  0.0        0.2966    0.2966    0.5582    0.5059    0.0       0.4884   0.4884
  0.1221    0.3489     0.3908    0.2791    0.0
******************************** BOTTOM OF DATA ********************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.20	0.50	0.150	-0.150

Table 4.2 Go down stairs 1.

```
BROWSE - J4478.GAIT.DATA(GAITB) - 01.33 --------------- LINE 00000 COLS 001 080
COMMAND ===>                                                  SCROLL ===> PAGE
********************************* TOP OF DATA *********************-CAPS ON-***
     2.0
   20   60
 BETA1
   19
    1    3    5    7   20   27   29   30   33   35   38   41   44   47   51   55
   58   61   81
  0.0        0.1326    0.1396    0.0523   -0.0523   -0.0523   -0.3768   -0.5582
 -0.6559   -0.6280   -0.4954    0.0       -0.1012   -0.4815   -0.4536   -0.4012
 -0.0419    0.0       0.0
 BETA2
   19
    1    4   12   16   17   19   21   24   27   28   33   35   41   43   47   50
   58   61   81
  0.0        0.1884    0.4919    0.4850    0.4536    0.4850    0.5722    0.5931
  0.6350    0.6629    0.6106    0.5233    0.0       0.1675    0.2233    0.2024
 -0.0454    0.0       0.0
******************************** BOTTOM OF DATA ********************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.20	0.50	0.150	0.030

Table 4.3 Go down stairs 2.

```
BROWSE - J4478.GAIT.DATA(GAITD) - 01.52 ---------------PAUSED AT BOTTOM OF DATA
COMMAND ===>                                              SCROLL ===> PAGE
********************************* TOP OF DATA *********************-CAPS ON-***
    2.0
  20   60
BETA1
  19
   1    2    4    5   17   20   24   31   32   34   37   41   43   46   50   54
  57   60   81
 0.0        0.1000    0.1400    0.1000   -0.0320   -0.3240   -0.4880   -0.6720
-0.6720    -0.6400   -0.4880    0.0      -0.1000   -0.3680   -0.3600   -0.3080
-0.0440     0.0       0.0
BETA2
  18
   1    4    9   15   18   20   24   26   28   38   41   42   43   46   53   57
  60   81
 0.0        0.1360    0.4000    0.6000    0.8000    1.0400    1.0880    1.0720
 1.0000     0.3800    0.0       0.1600    0.3200    0.3200    0.1720   -0.0800
 0.0        0.0
BETA3
   7
   1   13   22   26   28   35   81
 0.0        0.0       0.6080    0.6480    0.6080    0.0       0.0
******************************** BOTTOM OF DATA ******************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.48	0.52	0.170	-0.170

Table 4.4 Go up stairs 2.

```
BROWSE - J4478.GAIT.DATA(GAITU) - 01.99 --------------- LINE 00000 COLS 001 080
COMMAND ===>                                              SCROLL ===> PAGE
********************************* TOP OF DATA *********************-CAPS ON-***
 2.0
 13   53
BETA1
  11
   1    9   13   19   31   41   44   47   52   69   81
 0.0        0.2560    0.2640    0.1040    0.1600    0.0      -0.6600   -0.8560
-0.8560    -0.4000    0.0
BETA2
  15
   1    9   13   20   30   41   42   43   44   45   46   59   63   75   81
 0.0        0.2720    0.2800    0.7000    0.7120    0.0       0.5200    0.6120
 0.6120     0.4800    0.2120    0.5360    0.5600    0.3600    0.0
******************************** BOTTOM OF DATA ******************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.10	0.80	0.050	0.050

Table 4.5 Stop and start.

```
BROWSE - J4478.GAIT.DATA(GAITVS) - 01.42 -------------- LINE 00000 COLS 001 080
COMMAND ===>                                              SCROLL ===> PAGE
********************************** TOP OF DATA *********************-CAPS ON-***
    2.0 DATA OF VUKOBURATOVIC
  37   77
BETA1
  10
   1    7   27   32   39   47   67   73   79   81
-0.129     0.0       0.0       0.101    -0.003     0.0       0.0       -0.273
-0.173    -0.129
BETA2
  10
   1    7   27   35   39   45   47   67   79   81
-0.107     0.0       0.0       0.256     0.464     0.173     0.0       0.0
-0.159    -0.107
BETA3
   8
   1   31   37   42   71   74   79   81
 0.0       0.0       0.257     0.0       0.0      -0.148    -0.009     0.0
******************************** BOTTOM OF DATA ********************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.10	0.95	0.03	0.02

Table 4.6 Stand.

```
BROWSE - J4478.GAIT.DATA(GAITS1) - 01.40 -------------- LINE 00000 COLS 001 080
COMMAND ===>                                              SCROLL ===> PAGE
********************************** TOP OF DATA *********************-CAPS ON-***
    2.0     DATA OF STOP MORTION
   1   41
BETA1
   2
   1   81
  0.0       0.0
BETA2
   2
   1   81
  0.0       0.0
******************************** BOTTOM OF DATA ********************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.00	1.00	0.045	0.045

Table 4.7 Change of stride in usual walking.

角度ファクター SFO	歩幅 (cm)
0.8	25
1.0	31
1.2	37

Table 4.8 Change of height and stride in up and down stairs.

角度ファクター SFO	階段上り		階段下り	
	高さ (cm)	歩幅 (cm)	高さ (cm)	歩幅 (cm)
0.9	11.3	35.8	10.0	24.6
1.0	13.8	39.1	13.0	26.4
1.1	16.6	42.1	16.5	27.8
1.2	———	———	20.1	29.0

Table 4.9 Go up stairs 3.

```
BROWSE - J4478.GAIT.DATA(GAITUU) - 01.51 -------------- LINE 00000 COLS 001 080
COMMAND ===>                                                  SCROLL ===> PAGE
********************************** TOP OF DATA ********************-CAPS ON-***
  2.0
  13   53
BETA1
  11
   1    9   13   19   31   41   44   47   52   69   81
 0.0        0.0400    0.0500    0.1000    0.1400    0.0      -0.3000   -0.8400
-0.8290    -0.1000     0.0
BETA2
  15
   1    9   13   20   30   41   42   43   44   45   46   59   63   75   81
 0.0        0.3000    0.0560    0.2500    0.3500    0.0       0.3000    0.4000
 0.8000     0.7500    0.4000    0.1050    0.0800    0.0700    0.0
******************************** BOTTOM OF DATA ******************-CAPS ON-***
```

T_1	T_2	β_1	β_2
0.28	0.50	0.080	0.050

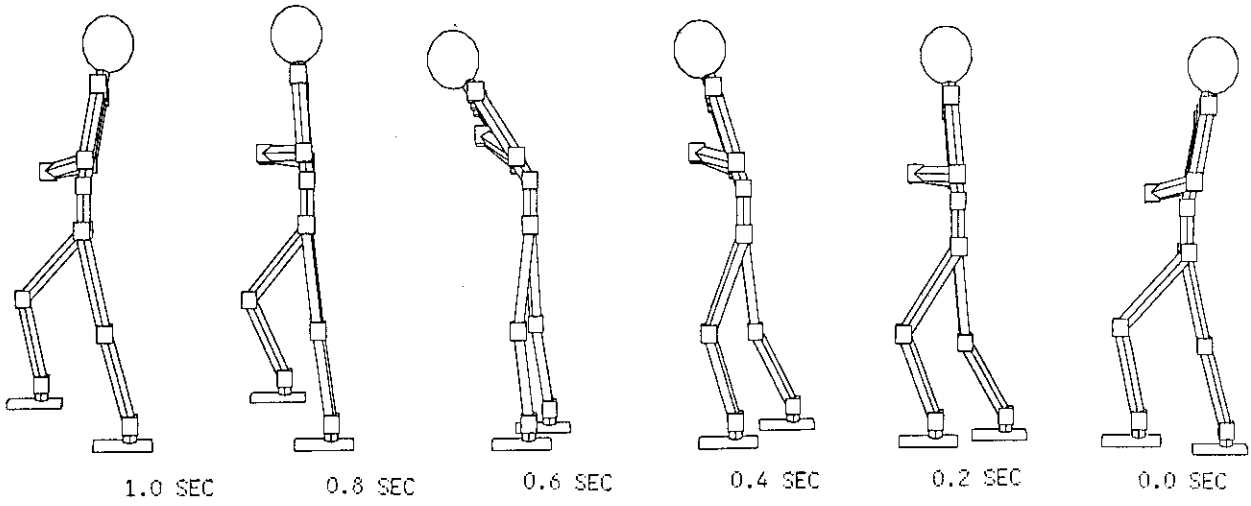


Fig. 4.4 Go up stairs 1.

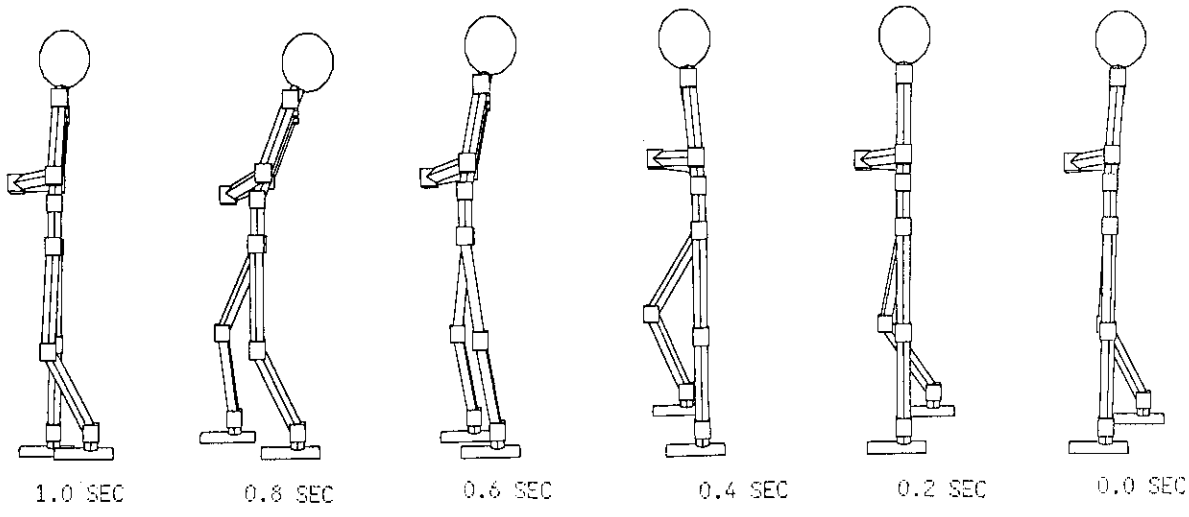


Fig. 4.5 Go down stairs 1.

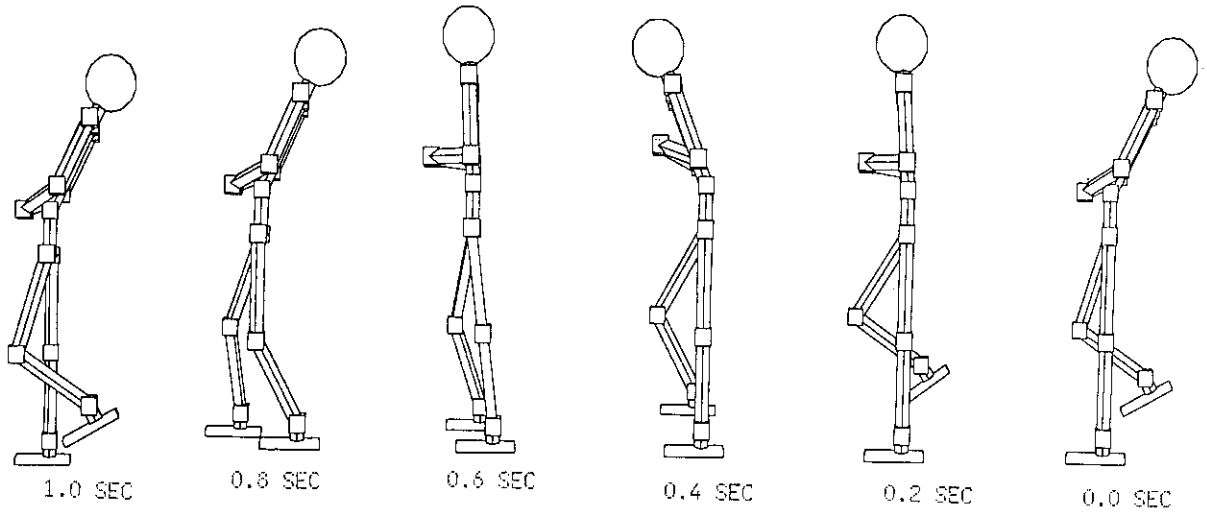


Fig. 4.6 Go down stairs 2.

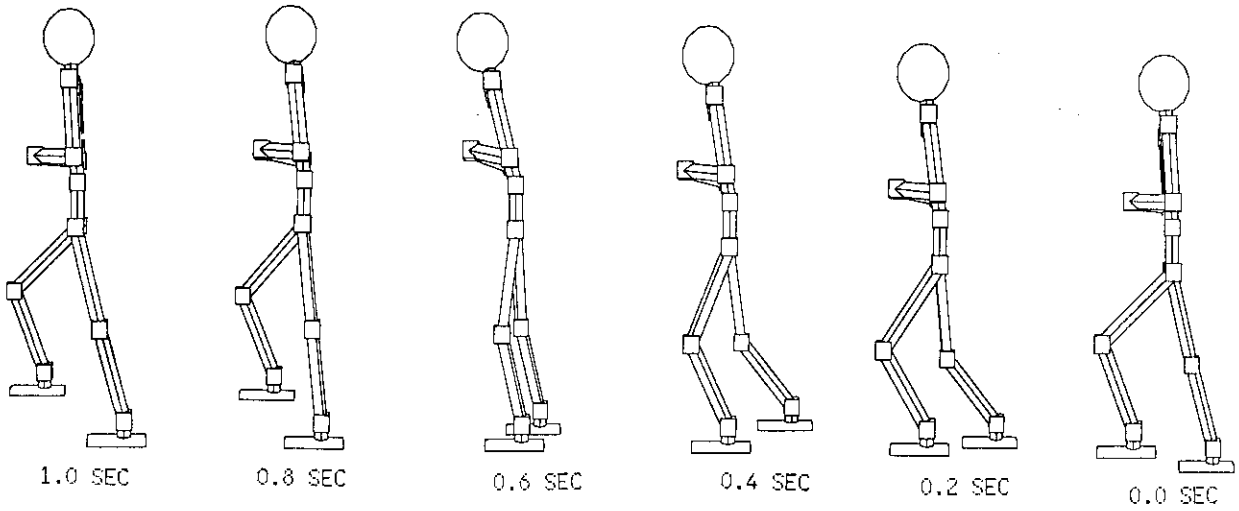


Fig. 4.7 Go up stairs 2.

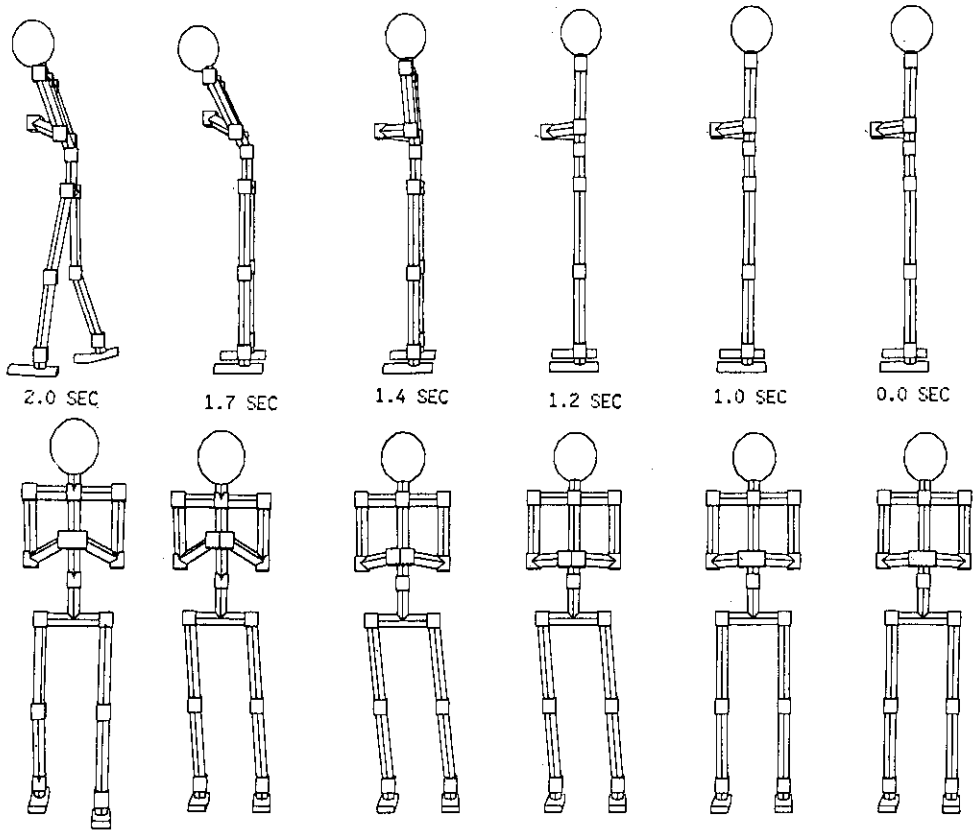


Fig. 4.8 (a) Start.

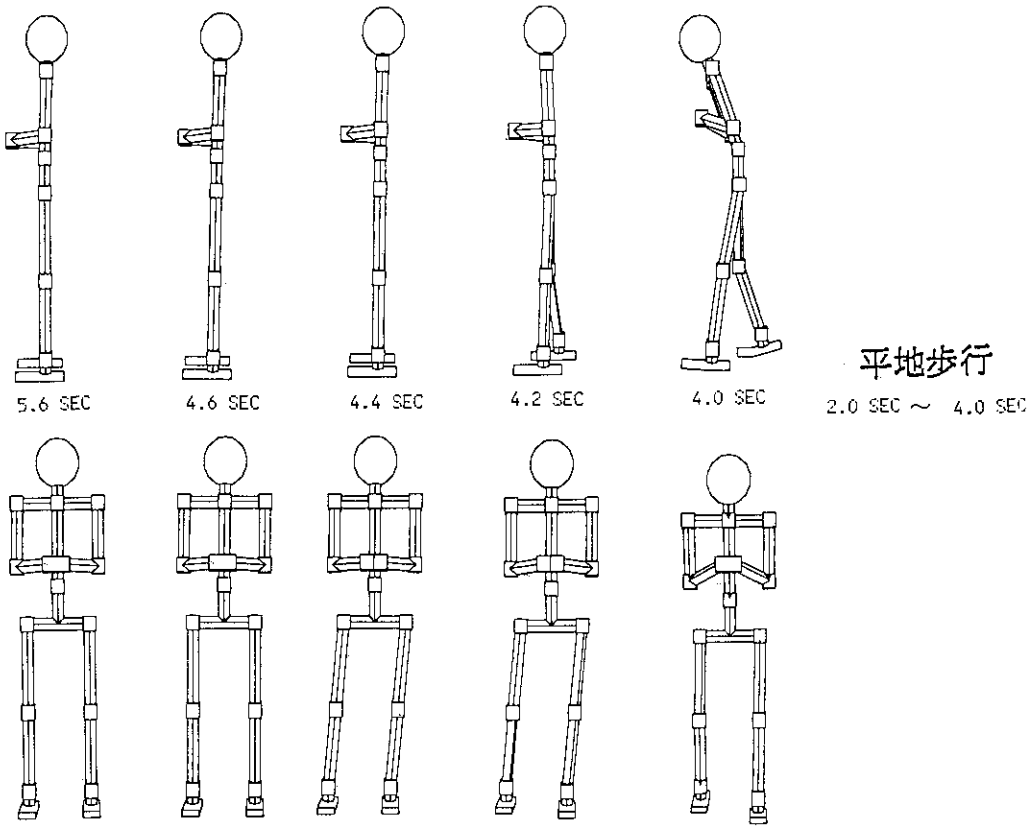


Fig. 4.8 (b) Stop.

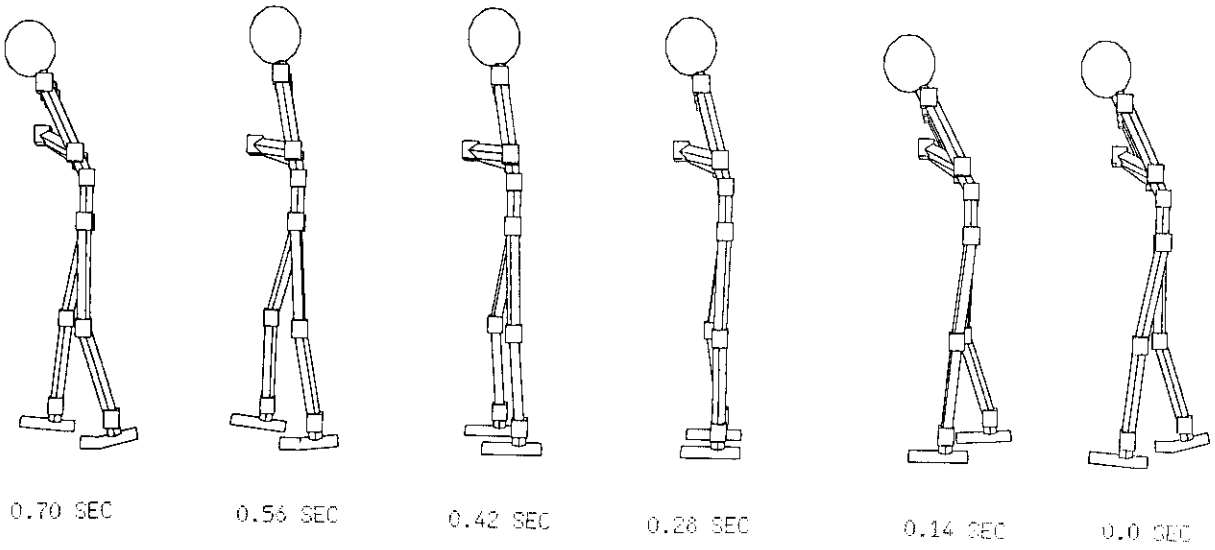


Fig. 4.9 Usual walking (PER=0.7).

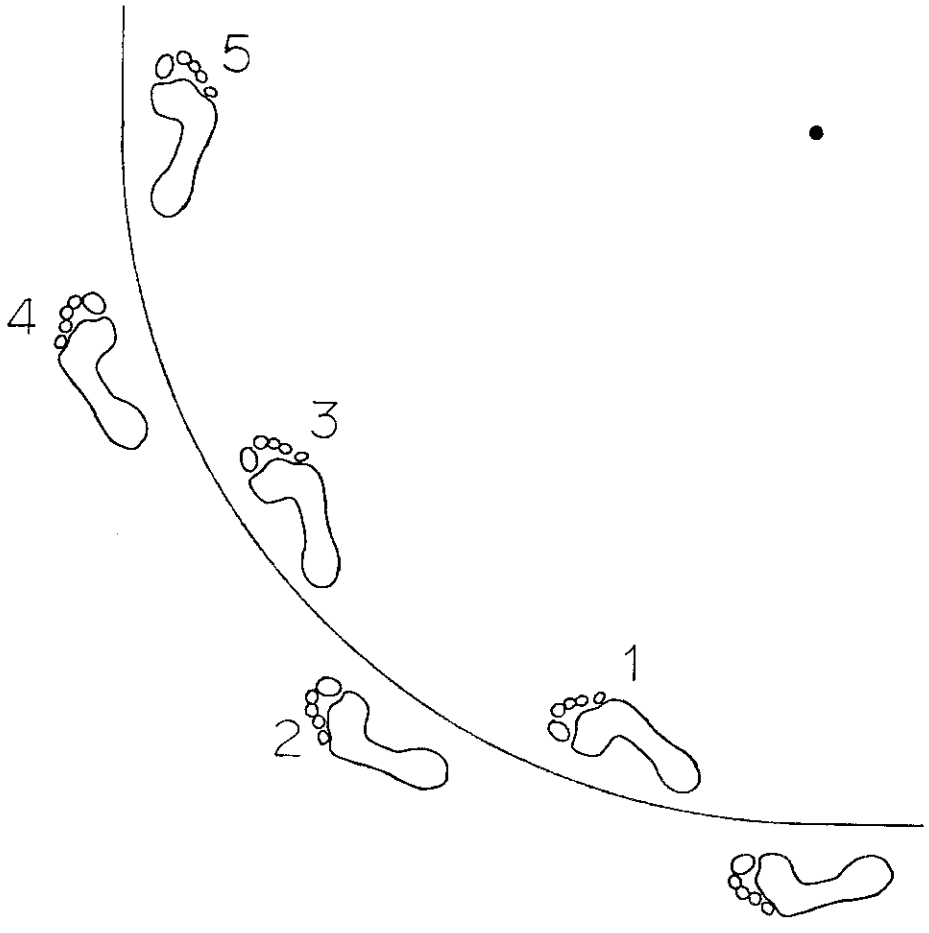


Fig. 4.10 Footmark.

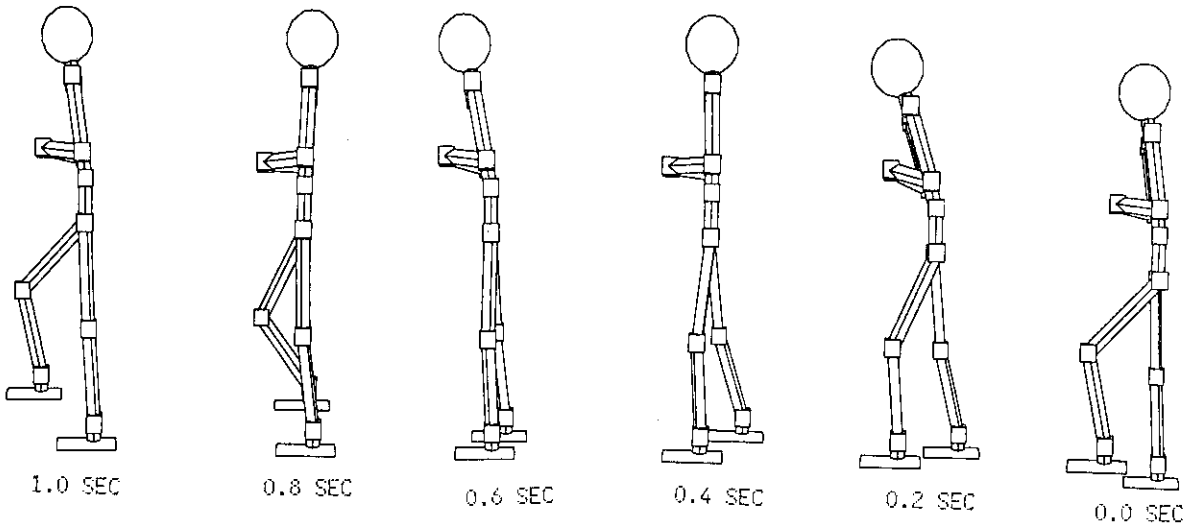


Fig. 4.11 Go up stairs 3.

5. 施設形状データベース

5.1 はじめに

人間動作シミュレーションでは，原子炉施設における作業を対象にさまざまなシミュレーションが行われる。模擬人間が与えられた命令から動作を計画する思考のシミュレーション，作業中の判断材料を得るため環境認識を行う視覚のシミュレーション等である。また，モンテカルロ・コードを用いて原子力施設内作業時のヒバク線量評価も行う。

上記のシミュレーションにおいて，模擬人間が作業を行う環境中に存在する物体の形状データ及び非幾何学的属性値を必要とする。つまり，模擬人間が与えられた命令を理解し動作列を生成する際，設定された環境における経路探索及び物体との干渉チェック等の行動計画が知識ベースを用いて実行されるが，これは施設形状データベースに保存されているデータを使用することにより可能となる。このように模擬人間の作業環境に関する形状データ及び非幾何学的属性値の登録・更新・保存を行うものが施設形状データベースである。ここで，形状データとは各物体を構成するプリミティブ名，各プリミティブ固有のサイズ・パラメータ等であり，非幾何学的属性値とは各物体の名称，色，構成物質名等である。また，シミュレーション結果の確認のために，施設形状データベースを使用して動作中の模擬人間及び環境の映像をディスプレイに表示する。

今回，施設形状データベースを作成するにあたりシミュレーションを行う動作空間として原研3号炉（国産1号炉）を仮定した。原研3号炉は現在61年度～65年度にわたる改造計画に従い，改造工事が行われている。入力作業は62年度より3面図を参考に行った。62年度は一次冷却系配管及び重水タンク等の数個のタンクを入力し，2分程の動画を作成した。63年度は建屋の入力を上半期に行い，二足歩行シミュレーションの結果を含めて3分程の動画を作成した。

本章では施設形状データベースに登録されたデータの構造を5.2節で述べる。映像生成手法として採用されたレイ・トレーシング・アルゴリズムの説明を5.3節で行う。5.4節では，入力の際使用されたモデラーの説明を行う。5.5節で動画作成システムの説明を行った後，5.6節で二足歩行シミュレーション結果の動画化に関する説明を行う。

5.2 データ構造及び物体表現方法

画像処理分野における3次元物体の表現方法としては，B－repによるもの，サーフェイス・モデルによるもの，ソリッド・モデルによるものがある。

ここでB－repとはBoundary representationを意味し，頂点と辺の情報のみをもち，3次元物体の輪郭を表現する手法であり，物体表示をワイヤー・フレームを使って行う場合に使用される。一般にＣＡＤシステムは，この手法により物体表現を行っている。サーフェイス

・モデルとは3次元物体の表面を複数の多角形で表現する手法である。この場合，ソリッド・モデルにより物体表現を行った場合に比べ，ポリゴンの数に比例して増加する保存データ量が大量であり，また映像生成時に大量のメモリを必要とする。反面，映像生成時間は，各ポリゴンの座標データに対し透視変換用のマトリックスとの乗算を行うだけで良く，小さい。しかし，曲面を複数の平面で近似するためにソリッド・モデルをレイ・トレーシング・アルゴリズムを用いて表示した場合に比べ画質は落ちる。

本研究テーマにおいては，3次元物体をソリッド・モデルにより記述した。この手法を用いることにより，3次元物体は，球や直方体等のプリミティブと呼ばれる素立体の集合演算により記述される。このデータ構造はFig.5.1に示されるような木構造となる。環境中の各物体は，論理演算子を持った複数のプリミティブで表現される。各プリミティブに関するデータの例をTable 5.1に示す。この例はある物体を表現するために使用されたシリンダーに関するデータである。底面の中心が（0.0，0.0，0.0），半径10.0，高さ1.0，中心軸がZ軸に沿った円柱を表現している。各物体を構成するプリミティブに関するデータとして，プリミティブの種類，位置，方向，大きさ等を決定するパラメータの値等の形状データが必要である。また非幾何学的情報として構成物質名，色等の情報も必要である。

ソリッド・モデルによって各物体を表現した理由として以下の2点があげられる。原子炉施設内作業時のヒバク線量を計算する際，原研において従来から使用されているモンテカルロ・コードを用いて行う。この場合，計算の対象となる体系を記述する際にソリッド・モデルによるものが適している，という点。また，原子炉施設のように非常に多くの物体が存在する体系を記述する場合，データ量が少ないソリッド・モデルによるものが好ましいという点である。

Table 5.1 An example of primitive data.

Primitive	シリンダー
Primitive Number	1
X	0. 0
Y	0. 0
Z	0. 0
U	0. 0
V	0. 0
W	1. 0
R	10. 0

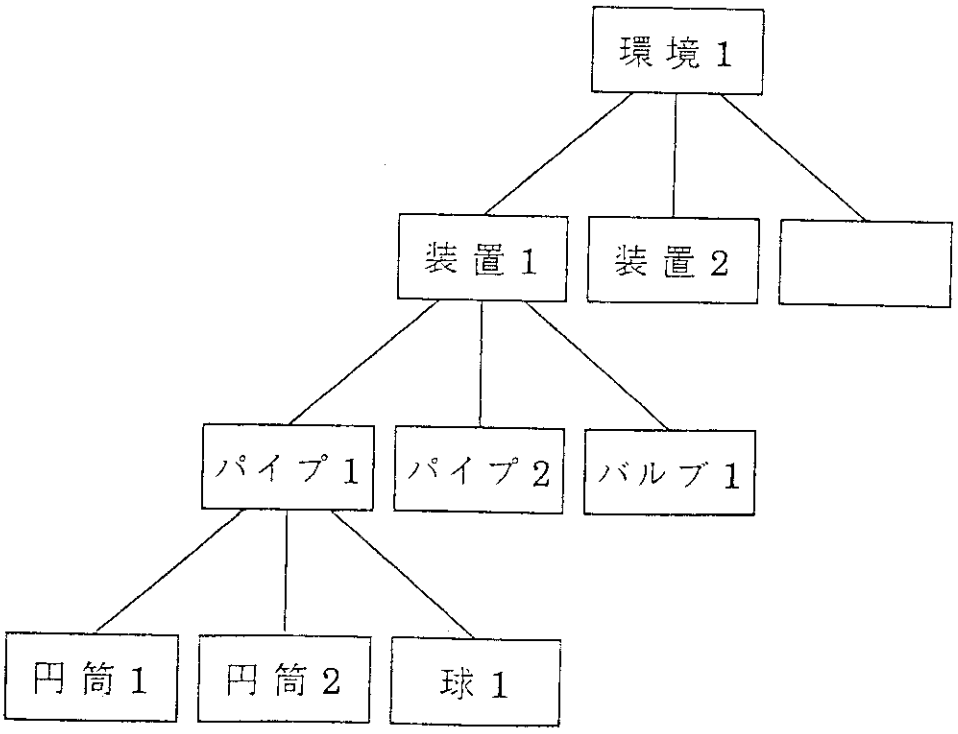


Fig. 5.1 Data construction of the plant data base system.

5.3 レイ・トレーシング・アルゴリズム

今回，映像生成のための手法として採用したレイ・トレーシング・アルゴリズムは，原理は簡単であり，また，鮮明な画像を得られるという利点があるが，反面，多大な計算時間を必要とするために以前は現実的なアルゴリズムではなかった。しかし計算機の速度向上に伴い，10年程前より有力な画像生成アルゴリズムとして注目を浴びるようになってきた。

レイ・トレーシング・アルゴリズムの原理は，人間の目が情景をとらえる様子をシミュレートしたものである。「物体が人間の目に映る」ということは，光源から出発した光が物体に衝突・反射し人間の目に届くことである。この一連の光の動きを模擬する際，理想的には光源から出発した無数の光を追跡し，目に届いた光の強度及び色を調べる。しかし，使用可能な計算機資源の範囲を考慮し，通常は次のような計算を行う。

i) 横　数百×縦　数百　程度のピクセル（画素）で構成されるスクリーンを仮定する。

ii) 指定された視点を始点とし，スクリーン中のあるピクセルを通る直線を考える。（Fig.5.2，直線①）

iii) 直線が既に記述済みの体系内のどの物体と交わるかを調べる。

iv) 物体の衝突点から光源に向けた直線を考える。（Fig.5.2,直線②）この直線上において，物体と光源との間に物体が存在するか否かを調べる。

v) 上記iv)の結果を用いてピクセルに対する光源からの光の強度，色の寄与分を計算する。

上記の手順によりスクリーンを構成するすべてのピクセルに対するR（RED），G（GREEN），B（BLUE）の値を決定しそれに従い画面の表示を行う。

レイ・トレーシング・アルゴリズムにおいては，この他，光の鏡面反射，透過を模擬するために物体との衝突点においていくつかの直線をある条件に従って発生させ体系内の物体との交点を求めることもよく行われる。今回の作業においては，計算時間及びビデオを利用した動画作成時間等の要素を考慮し，反射面，透過面は扱っておらず拡散反射のみが考慮されている。

レイ・トレーシング・アルゴリズムについては，現在多くの解説書が存在するので詳しくはそれを参照されたい。[1)]

レイ・トレーシング・アルゴリズムは，この節の冒頭で述べたように多大の計算時間を要する，という点が欠点である。この欠点を克服するために，これまでさまざまな工夫がなされ，いろいろなアルゴリズムが提案されてきた。本研究テーマにおいて使用されているプログラムでは，空間分割法及び3次元DDA（digital differential analyzer）アルゴリズムと呼ばれるアルゴリズムが採用されている。このアルゴリズムでは直線と物体との交点をみつけるために行う交差テストの回数を減らす工夫がなされている。通常，「直線が計算対象となる体系内のどの物体と最初に衝突するか」という計算は，体系内のすべての物体と交差テストを行い，その距離を比較することにより為される。今回採用した画像生成プログラムでは多大の計算時間を要するこの交差テストの回数を減らすために次のような計算を行う。

i) 予め体系内の空間をサイコロ状に分割し，各サイコロ（VOXELと呼ばれる）内に含まれる物体（プリミティブ）を登録しておく。

ii) 直線と物体（プリミティブ）との交差テストは直線が通過するVOXELに含まれる物体

（プリミティブ）とのみ行う。

iii) 直線が進入するVOXELの番号は直線の傾きから逐次的に求める。

以上のようなソフトウェアによる計算時間短縮の他，本研究においては5.5節で述べる画像生成用並列計算機CAPを用いて，並列計算機による高速処理を行っている。レイ・トレーシング・プログラムでは数十万～数百万本の光線を追跡するが，それぞれの光線に関する計算は独立であり並列処理が可能である。また，並列計算機を使って処理を行う場合，ピーク性能を引き出すために各プロセッサの実行待ち時間が問題になるが，この点は個々のプロセッサに光線をうまく割り当てることにより解決される。詳細は文献2)に詳しい。

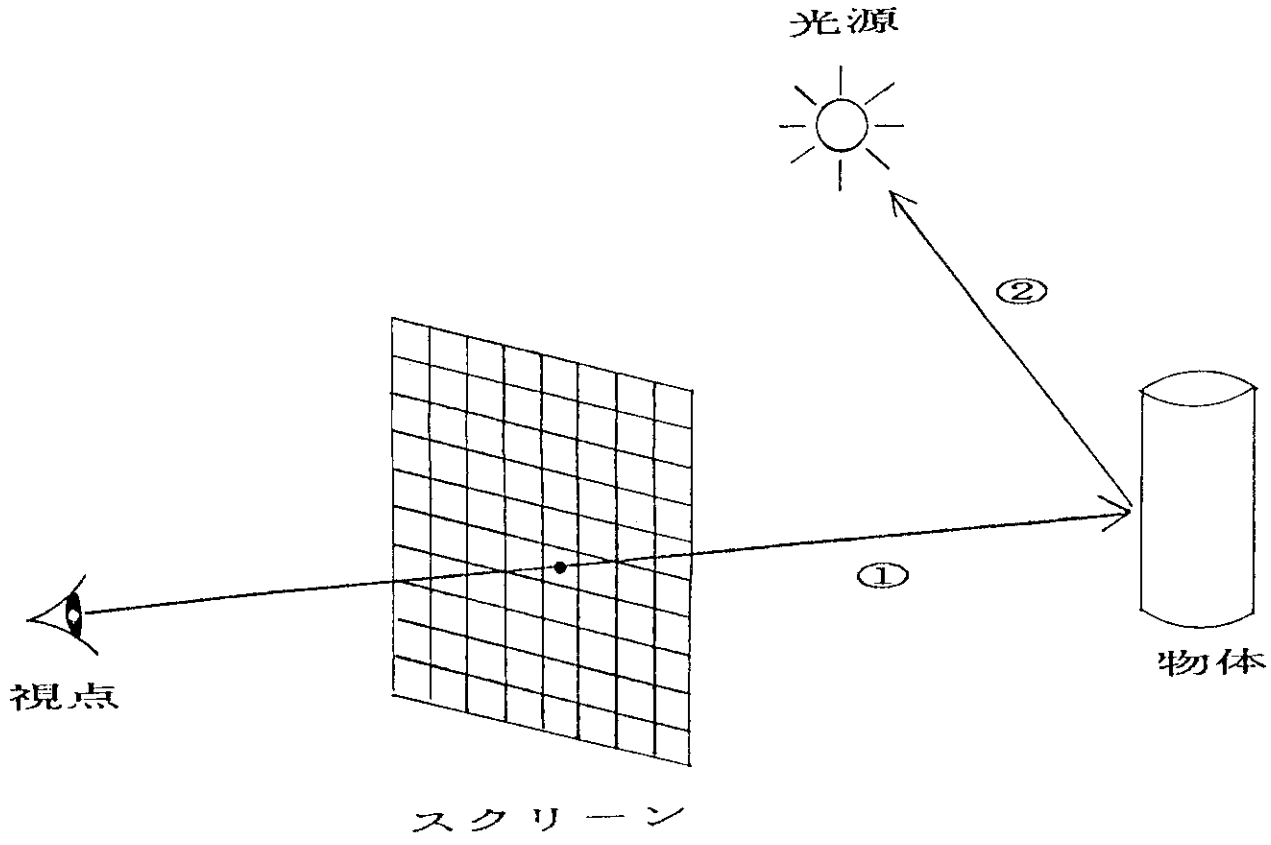


Fig. 5.2 Illustration of ray tracing algorithm.

5.4 モデリング・ソフト FUSION

5.2節で述べたように，本研究において3次元物体はプリミティブの集合演算によるソリッド・モデルによって定義される。入力作業は，三面図（平面図，側面図，断面図）から各装置，機器の情報を掘り起こして行われた。計算機へのデータ入力には，富士通研究所の開発したモデリング・ソフト FUSION[3)] を使用した。モデリング・ソフトとは，人間が思い浮かべたイメージ及び把握した数値データを計算機にインプットするために使用するマン・マシン・インターフェイス・プログラムである。FUSIONは，「モデルの汎用性と再利用性に適した言語型モデリング」と「イメージを具体化するのに適した会話型モデリング」の双方の特長を生かす方針で設計されている。UNIX系ワークステーションの特長であるマルチウインドウを用い，各物体に関する数値の入力，属性の入力を行う。物体を表現する際に用意されているプリミティブは球，板，立方体，楕円球，円柱，楕円柱，円錐，双曲体，放物体の9種類である。さらに，今回配管の曲形部を表現するために円環体をプリミティブとしてつけ加えた。モデリングされる物体はこれらのプリミティブの集合演算で表現される。用意されている演算子は＋，－，＆（共通領域），（　）の4種類である。以上のCSG（Constructive Solid Geometry）の表現の他にFUSIONの特徴として次の2点があげられる。

(1) 属性の継承

色，質感等の非幾何学的性質の定義を行う際，物体階層の上位で指定された属性を下位の部品に伝播させることができる。

<例> 円環体や円柱等のいくつかのプリミティブで表現された配管に対し，色，質感を定義，又は変更することができる。この時，配管を表現している各プリミティブに対しそれらの情報を与える必要はない。

(2) モデルの汎用性・再利用性

物体の大きさや詳細な形状に依存されない抽象化された概念として「クラス」と呼ばれるものを定義することができる。

<例> 円柱や板等のいくつかのプリミティブにより「椅子」をモデリングした場合を考える。ユーザーはこれを「椅子」という名前のクラスとして定義することができる。再利用の際には「椅子」というクラスに高さ，幅等の値を与え，具体的なプリミティブの集合体として扱うことができる。

今回の入力作業は富士通研究所に依頼して行った。モデリングの際，ホスト・コンピュータとしてSUN3／260HMを使用した。原研においてもSUN3／260Cを63年10月に導入し，入力済のデータ及びFUSIONのインストールを行った。

5.5 映像生成用並列計算機 CAP

5.3節で述べたように今回映像生成プログラムにおいて採用されたレイ・トレーシング・アルゴリズムは鮮明な画像が得られる反面多大の計算時間を要するという欠点がある。この欠点を補うためのアルゴリズム上の工夫については5.3節で述べた。以下に説明を行う並列計算機

CAP(Cellular Array Processor)は，富士通研究所において開発され，[4] 本年11月に映像生成用計算機として原研に導入された。計算機プログラムを高速に実行するために考案されているハードウェアとして，現在ベクトル計算機と並列計算機が存在する。それぞれの計算機は，実行されるプログラムに対し，ピーク性能を引きだすことに適した構造を要求する。レイ・トレーシング・アルゴリズムを使った計算には5.3節で述べたように，高い並列性があり，並列計算機による高速化が可能である。今回原研に導入されたＣＡＰ計算機は，レイ・トレーシング・アルゴリズムによる映像生成を行った時，原研計算センターに既設の大型計算機M780の1／2程度の性能を発揮する。

ＣＡＰ計算機はセルと呼ばれるプロセッサ・ユニットを８行８列の二次元格子状に配列した並列計算機であり，今回原研に導入されたものは，128プロセッサを持つ。セルはすべて同じ構成で１６ビットの汎用マイクロプロセッサを中心にローカル・メモリ，ビデオ・メモリ，通信インタフェースなどで構成している。セルの構成をFig.5.3に示す。

またＣＡＰとその映像生成システムの構成をFig.5.4に示す。ホスト・コンピュータとしてFACOM A−50を使用した。プログラムやデータは，ホスト・コンピュータからコマンド・バスを通じて全セルに同じものが送られる。セルがコマンド・バスを用いてほかのすべてのセルにデータを送ることも可能である。

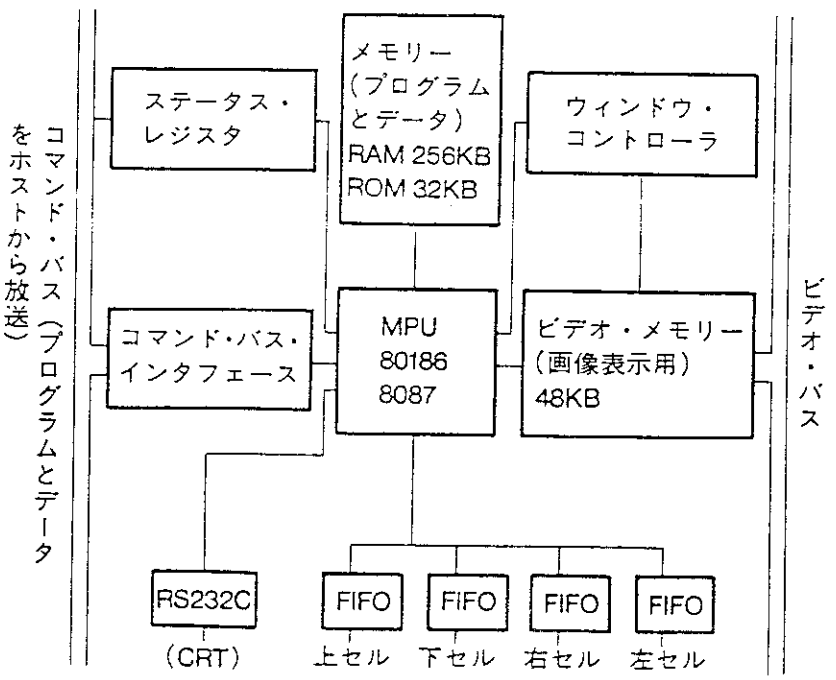


Fig. 5.3 Hardware structure of a cell.

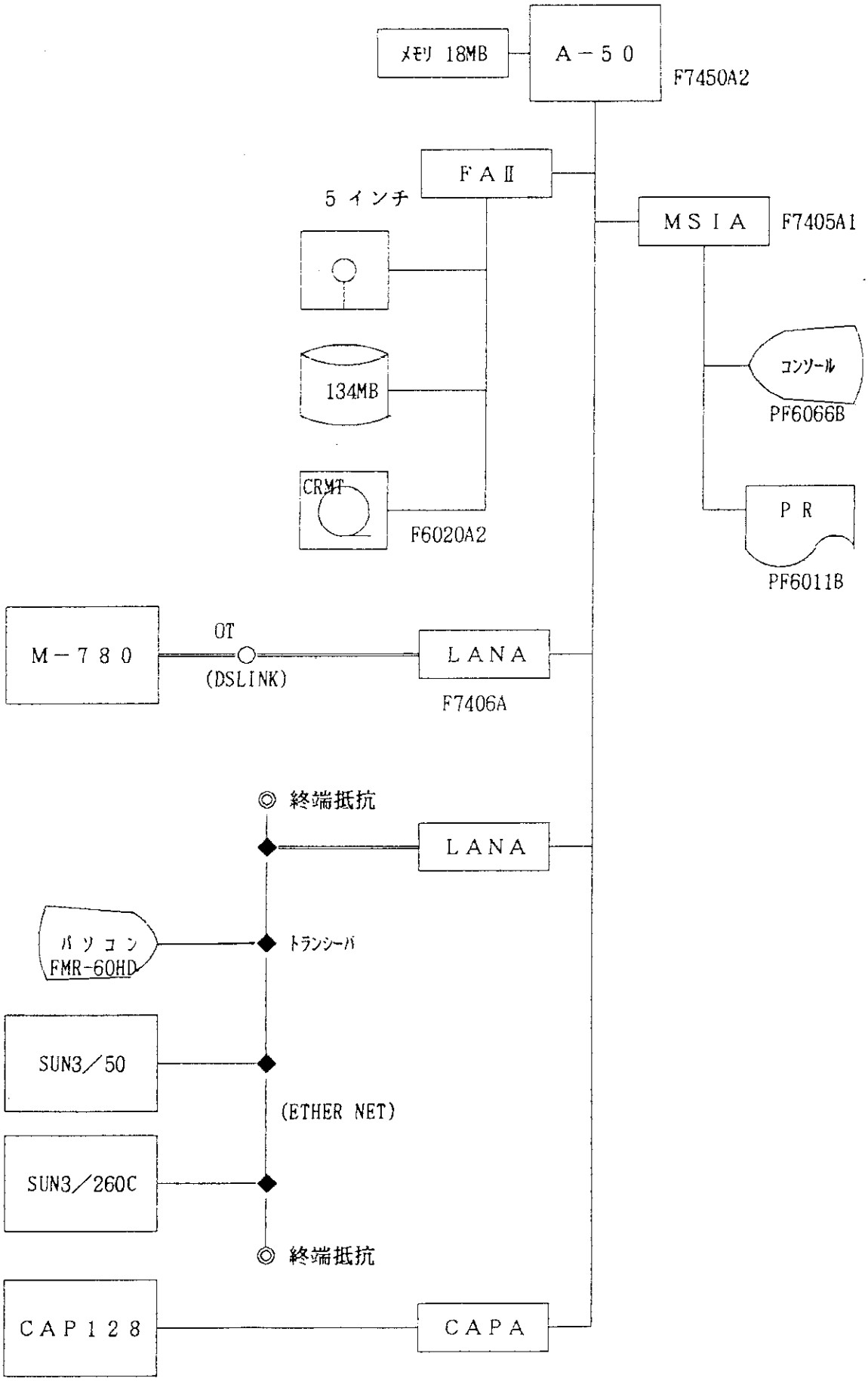


Fig. 5.4 CAP hardware configration.

5.6 二足歩行シミュレーション結果の動画化

６２年度に行ったロボット運動学の計算結果に基づいた二足歩行の動画化について報告する。[5] 動画は総時間３分７秒で，施設環境中で二足歩行する人間動作を光線追跡法で映像化したものである。光線追跡法による静止画像の作成は，６３年１１月に導入したセルラ・アレイ・プロセッサ（CAP－128）で行い，ＶＴＲにコマ録りすることで動画化した。以下にその動画化の概要を述べる。[6]

5.6.1 CAP動画作成システムの概要

並列処理と映像化機能を一体化させた128台構成の計算機CAP－128を導入し，ワークステーションや他の映像機器と組み合わせて動画作成システムを整備した。ハードウェア構成図をFig.5.5に，主な使用ソフトウェアをTable 5.2に示す。M－780は，計算センターに設置された大型計算機で二足歩行動作の計算を行う。A－50とDSLINKで結合し，A－50がM－780の端末として使用できファイル転送も可能になっている。SUN3／260Cはモデリングのためのワークステーションで FUSIONを用いてロボットや施設形状データベースに登録される物体に関する形状定義，色や属性等の定義を行う。また二足歩行計算プログラムをFUSIONに反映するためにインターフェースプログラムも動作する。SUN3／260CとA－50とはTCP／IPで結合され相方向の端末としての使用及びファイル転送が可能である。A－50は，CAP－128のホスト計算機でCAP及びVTR装置の制御プログラムが動作する。ＶＴＲはコマ録り可能な1／2インチβ－CAMを使用している。

5.6.2 動画の作成

動画は１秒間を１５等分にコマ割りして，各コマにＣＡＰで生成した静止画像を録画する事によって実現した。動画作成の手順をFig.5.6に示す。FUSIONで姿勢を変化出来るように定義したロボットと施設環境（ＪＲＲ－３の１次冷却系配管を含む地下室）を統合したモデルに1／15秒毎のロボットの姿勢データ及びシナリオに従ったカメラデータの組を与えることによって，各コマ毎にSUN上で線画を生成する。その後，A－50をホスト計算機にしてＣＡＰに光線追跡（レイ・トレーシング）計算を依頼して静止画像が生成される。この静止画像がA－50の制御の下にVTRに駒撮りされて動画が作成されていく。以下に二足歩行計算結果の反映方法，モデリング，シナリオの決定からカメラデータの作成，ＣＡＰによる画像生成そして動画の作成について述べる。

(1) 二足歩行計算結果の反映

① 二足歩行計算プログラムの整備

６２年度整備したVukobratovicの二足歩行計算プログラムの計算結果をロボットの姿勢に反映するためにさらにプログラムを整備した。第１に時間ステップの変更を行った。Vukobratovicのプログラムでは１歩を１秒で歩行し，１秒を40分割して各時間ステップ毎にロボットの姿勢（各部位の重心座標）を出力していた。これを動画化のために時間ステップを可変にするように修正し，ＶＴＲの最大コマ数である1／30秒で出

力した。第2に出力データの変更を行った。Vukobratovic のプログラムがロボットの各部位（腕，脚等）の質量中心のX，Y，Z座標を出力していたのに対して各部位の曲折角度データ（姿勢データ）を出力するようにした。これらの変更によって各時間ステップ（1/30秒）毎のロボットの姿勢の31組の曲折角度データが出力できるようになった。

今回のVTRは，1秒に15フレーム録画するので31組の曲折角度データを半分に間引いて16組のデータ（0秒から1秒まで0.067秒間隔）を使用する。歩容は，平坦歩行するデータを使用している。

② FUSION用インターフェース・プログラムの作成

ロボットモデルはプリミティブのCSGによる組み合わせや回転・平行移動で表現されている。したがってロボットの歩行動作を表現するためには，基準となるプリミティブ（具体的には首の付け根部分）の絶対座標と各部位（脚や腕等）のプリミティブの回転角度を各時間ステップ毎に与える必要がある。また二足歩行計算プログラムでは，1歩分のデータのみ出力されるので2歩以上歩く場合には左右交互に足が接地していることを考慮にいれた反復計算を行う必要がある。そこで，FUSIONで入力できるような形式で基準となるプリミティブの絶対座標と各部位のプリミティブの回転角度をファイルに出力するプログラムを作成した。

(2) モデリング

モデリングはFUSIONを用いて行っている。モデルは，JRR-3の施設形状モデル，歩行可能なロボットモデル及びその2つを統合した統合モデルである。実際動画に使用したものは，統合モデルであり，施設の中のある位置にロボットを立たせた状態のモデルになっている。施設形状モデルは，62年度からJRR-3の設計図を解析して入力しているもので使用したプリミティブの数は約1400である。ロボットモデルは，プリミティブ数36個で腕，腰，脚，爪先等が動作できるように各プリミティブにX，Y，Z 方向の長さに自由度及び回転の自由度を与えている。統合モデルは，モデルの色や属性（反射率等の設定を行う事によって質感等を表現する）等はパイプ，タンク，壁，ロボット等の材質の違いで変化させている。また壁は，コンクリートの材質感を出すためテクスチャ・マッピングを施している。

(3) シナリオの決定

動画を作成する場合には施設環境内の二足歩行ロボットの動作をどの位置から撮るかというシナリオの決定が重要である。具体的にはカメラの位置及びターゲットの位置（カメラが注目する点）を各タイムステップ毎に決定する必要がある。そこではじめに，シーン構成，各シーンにおけるロボットの歩行経路・時間，大体のカメラ位置，ターゲット位置等の大枠のシナリオを決定する。次に具体的に最適なカメラ位置，ターゲット位置を実際に静止画像を出力して試行錯誤で決めていく。そして最終的なカメラ位置とターゲット位置をFUSION用入力データとして，ロボットの関節曲折角度データとともに1組にして同じファイルに書き込む。施設内の歩行経路及びカメラ位置をFig.5.7に示す。この図は，施設の地下室（天井，外壁を取り除いたもの）を上空から見た時の概略図である。図中の

丸付き数字はロボットの位置，ＥＹＥ数字はカメラ位置，矢印がロボットの歩行経路，点線はカメラの視線の方向を示している。ロボットは①→②，③→④→⑤の方向に歩行する。シーンは次のように構成されている。まずはじめに施設の上空から施設の全体を見ている状態で静止している。そして①の地点までロボット周辺に注目しながら降下してくる。次にロボットが①から②を歩行している様子を外部の視点（ＥＹＥ１）で撮り，同じ経路を今度ロボットの視点（ＥＹＥ２）で撮る。ロボットは②の位置で周辺を見回し（ＥＹＥ３），最後に③方向を見る。この後③から⑤まで歩行する。この時，Fig.5.7に示す背景の壁にさしかかる迄と壁から出てくる様子を外部の視点（ＥＹＥ4，ＥＹＥ5）から撮る。最後に③から⑤までをロボットの視点でタンクの方向を時々見ながら撮る。この時視点が固定の場合は何枚かの静止画出力でカメラ・ターゲット位置は定まるが，上空から視点を降下させるシーンやロボットの視点で周りをみるシーン等は動きを滑らかに見せるためカメラデータ作成プログラムを作成して，計算によってカメラ位置とターゲット位置を定めるようにした。

(4) ＣＡＰによる画像生成および動画作成

ＦＵＳＩＯＮではCAPでの静止画像･動画生成をほとんどマウスによるウインドウ操作だけで簡単に実現できるようにしている。ＣＡＰで画像（１枚の静止画像）を作成するための準備としては，ＦＵＳＩＯＮで統合モデルを部品として参照し，光源と１組のカメラデータを与える。動画を作成するためには，さらに(3)で述べたロボットの姿勢データ及びカメラ位置・ターゲット位置のデータを含んだ動作ファイルをマウス操作によって結合しておけば良い。この他に背景色やアンチエイリアシング処理の有無等の設定を行う。アンチエイリアシング処理とは，ラスタ型ディスプレーを用いて図形表示する場合に斜め線がギザギザになる現象を緩和する処理をいう。また，動画作成を円滑に行うために，以下のマン・マシン・インターフェースが用意されている。第１にＶＴＲの録画開始時間等の設定やＶＴＲ装置の制御（早送り，サーチ等）をウインドウ上で行える。第２に画像情報（各ピクセル毎のＲＧＢ値）をファイル出力する機能もウインドウ上で設定することができる。さらに第３に，これらの準備作業の後Ａ−50との通信路を開いてマウス操作によってデータ転送を行えば，自動的に動作ファイルに含まれるコマの映像を生成してＶＴＲに録画することができる。

(5) 作成工数と資源

動画を生成するための資源と工数について示す。CAP−128の場合にプリミティブ数が約1400のレイトレーシング計算を行っている静止画像が，１フレームあたり２分〜１２分で平均５分で生成する。また，画像生成時間に加えて画像生成用入力データの転送時間及び画像録画時間等に３分程度必要である。この結果，1702フレームで構成される今回作成した動画の総生成時間は229時間となる。またＶＴＲに録画する一方で画像情報をＡ−50に出力している。画像情報は１フレームあたり0.6メガバイトなので合わせて約１ギガバイト必要となった。

Table 5.2 Standard software on CAP system.

ソフトウェア名	ハードウェア名
二足歩行計算プログラム	M－780
モデリングソフトFUSION 二足歩行計算インタフェースプログラム	SUN3 260C
レイトレーシング・プログラム	CAP－128
CAP，VTR制御プログラム 通信関連ソフトウェア	A－50

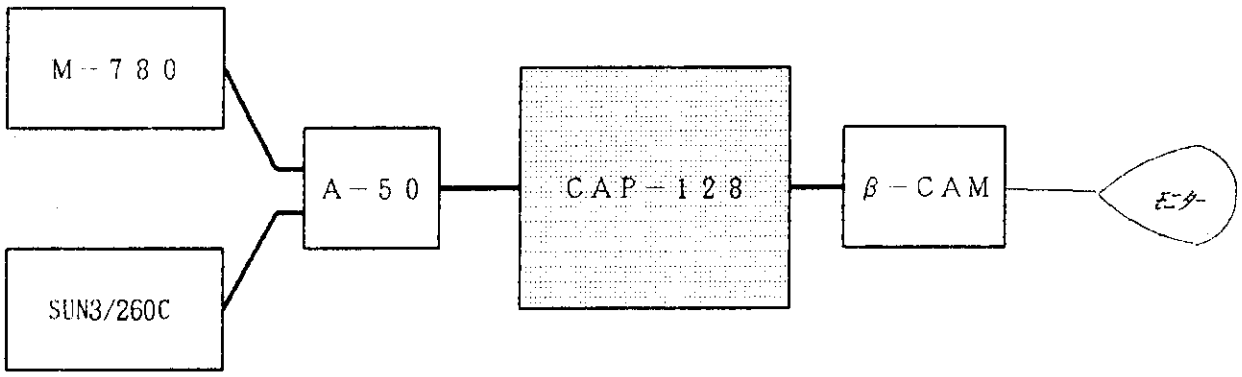


Fig. 5.5 Hardware configuration of video recording system.

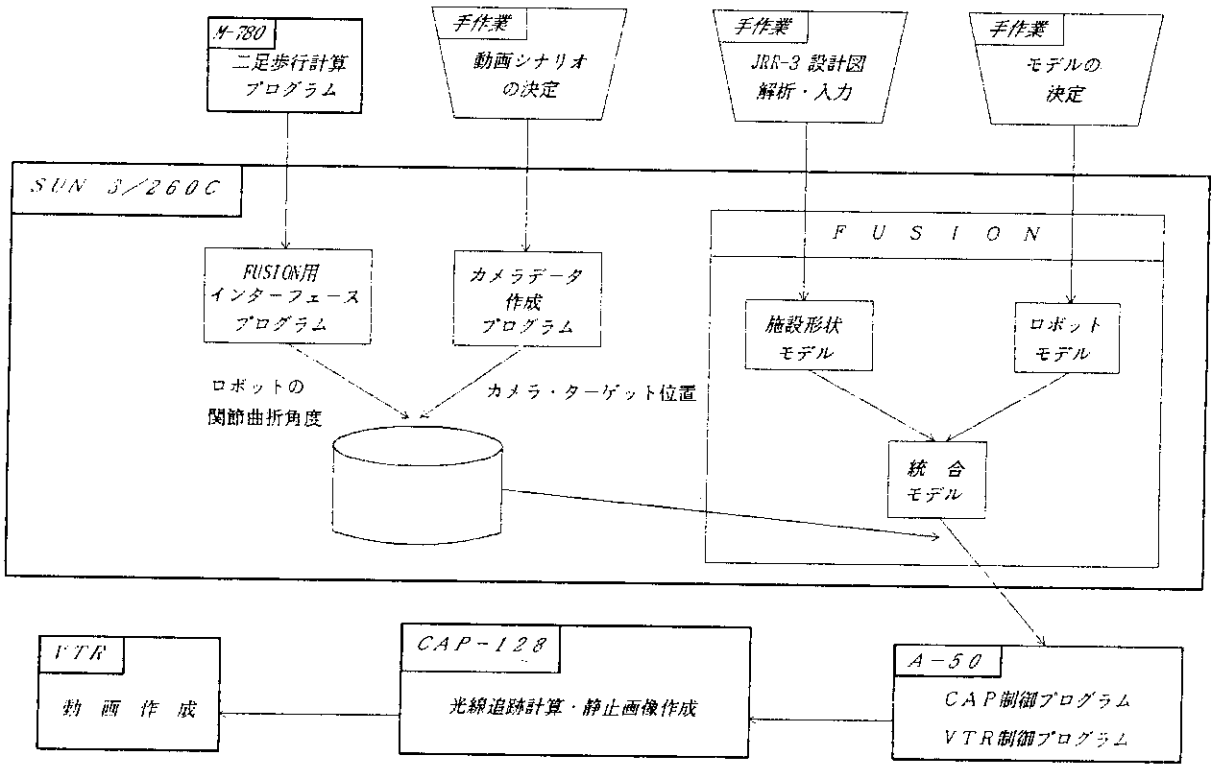


Fig. 5.6 Flow chart for video recording.

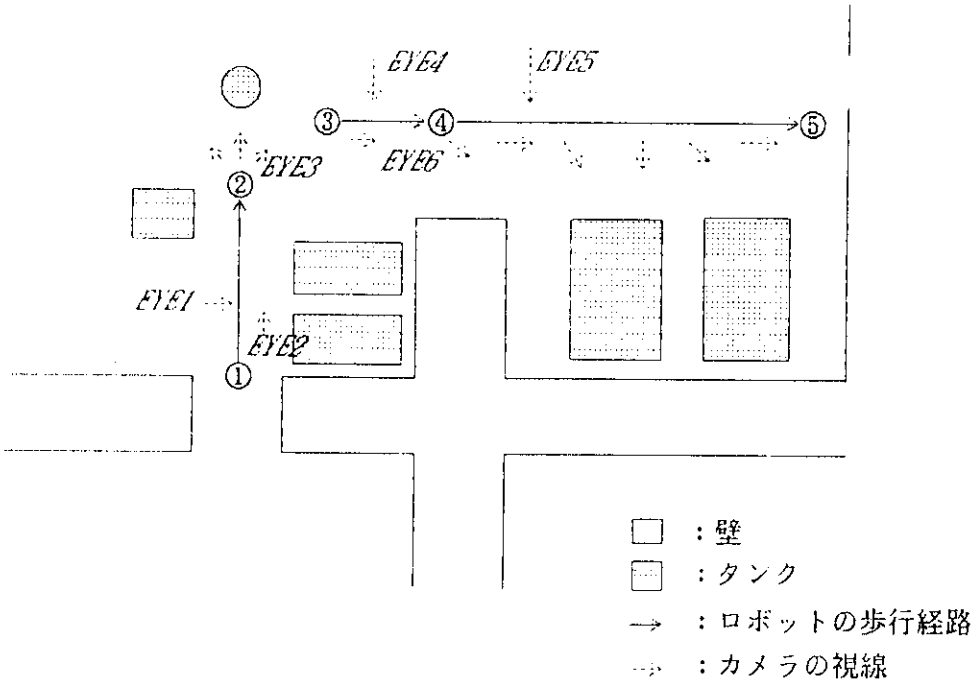


Fig. 5.7 Walking route and camera position.

5.7 施設形状データ形式変換ソフトウェア

模擬人間の行動をシミュレートするためには，大規模な計算が必要となる。たとえば，ロボットの自己可動空間の認識及び決定は，環境中の物体との干渉チェックなどに多大の計算が必要となる。したがって，動作計画は大型計算機上で実行することになるが，シミュレーション結果を映像化するためのソフトウェアは，エンジニアリング・ワークステーション（EWS）と映像生成用並列計算機ＣＡＰを使用して行われるため，大型計算機との整合性を取る必要がある。そのため，現在ＥＷＳ上に存在する施設形状データ及びＣＡＰで作成された映像データを大型計算機で利用可能な以下の形式に変換するソフトウェアを富士通研究所に依頼して作成した。また，変換されたデータは，動作計画だけでなくロボットの視覚認識，被曝線量評価などの計算にも利用される。このソフトウェアにより形状データ及び画像情報をFORTRAN言語で読み込み可能な３つの形式に変換することが可能になった。３つの形式は以下のものである。

(1) ＣＳＧ形式

ＣＳＧ形式は，対象となる物体の形状を幾つかの基本的な立体（プリミティブ）の集合演算（和，差，積）によって表現する。形状定義ソフトウェアFUSIONで定義されているデータはＬＩＳＰ言語で処理するためのＬＩＳＴ形式であり，FORTRAN処理には好ましくない。そのため，１つの物体の形状を木状のデータ構造（木構造）として表現してFORTRAN処理可能とする。すなわち，木構造の葉にプリミティブを，ノードに集合演算を対応させることにより，１つの物体が定義できる。木構造の１つのノードはFig.5.8のようなデータ構造となる。このとき，物体の持つ属性，すなわち色，材質，放射化度などのデータも合わせて保持することが可能である。

(2) Voxel形式

Voxel形式は，施設形状データベースの存在する３次元空間をさいの目状に小分割して，各小片の中の物体の存在を記述したものである。このVoxel形式のデータは，ロボットと環境との大まかな干渉チェックやロボットの移動経路探索に利用することができる。

各Voxelには，物体があるか，空であるかの２値データを与え，全体として３次元のデータ構造とする。空間分割の大きさは，任意の精度で物体を表現できるように，一方向100分割まで変化させることができる。ただし，この制限はプログラムの簡単な修正により変更可能である。

(3) Pixel形式

Pixel形式は，ＣＡＰによって表示された任意の映像に対して，すべての画素についてのＲＧＢ情報を持つ。このため，任意の視点からの２次元情報が得られるため，ロボットの現在位置の確認やロボットの目標物などの確認に利用できる。

１画素のデータは，４バイトでRed，Green，Blueの輝度データにそれぞれ１バイトずつ使用されている。

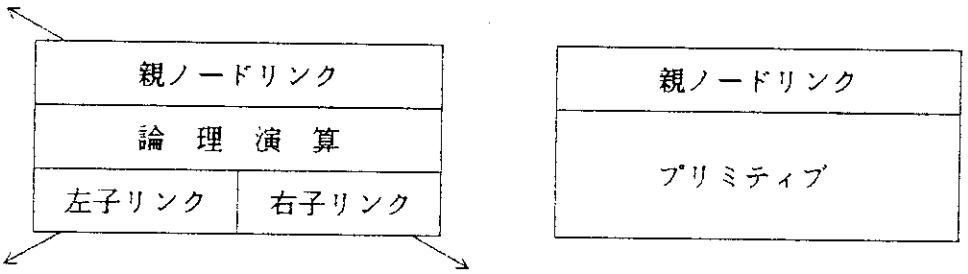


Fig. 5.8 Node construction of tree.

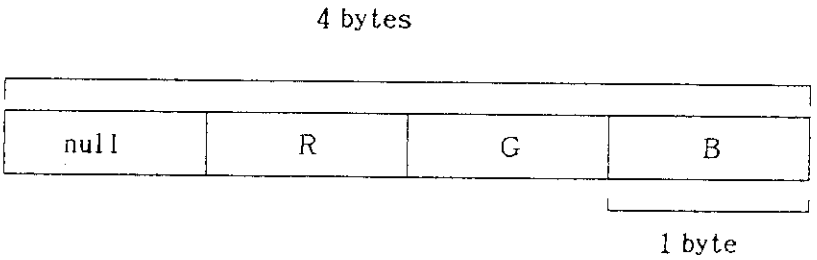


Fig. 5.9 Illustration of pixel data.

5.8 ロボット駆体モデリング・ソフトウェア

昭和62年度まではロボット運動はFig.5.7に示す様にSOLVERと呼ばれるソフトウェアを用いて線画で描いていた。このSOLVERでは，ある時刻のロボットの関節データを入力とした静止画のみが出力可能であった。そこで，今年度は，二足歩行シミュレーションの結果を利用して円滑に動画を作成するシステムを構築した。

ロボット運動の映像化は，FUSIONで定義した二足歩行ロボット（プリミティブで表現）に大型計算機で計算された二足歩行ロボットの各関節の動きを与えることで行われる。二足歩行ロボットの運動学的シミュレーションは多大な計算時間を要するため，この計算は大型計算機上で行われ，シミュレーション結果を大型計算機からFUSIONを実行するEWSへ送る必要がある。

5.4節で述べたようにFUSIONではソリッド・モデルにより物体を表現する。このため，各時刻において変化するロボットの駆体を具体的なプリミティブの集合体として表現する必要がある。また，モデリングに関しては会話形式で行うために，多数の映像から構成される動画作成を行う際，ロボットの関節角度の時系列データからプリミティブの集合体として表現されたロボットを自動的にモデリングする機能が必要である。この機能は以下の三つの手順を経て実現される。

(i) M780上で計算されたロボットの関節角度の時系列データをワークステーションSUN3／260C上に移行する。（Fig.5.11，a）

(ii) 移行されたデータをLISP言語で処理するためのLIST形式のデータに変換する。（Fig.5.11，b）

(iii) LIST形式のデータを読み込み，指定された時刻のロボット駆体をモデリングする。すなわち，ロボットの駆体を表現する各プリミティブに具体的な座標値を与える。（Fig.5.11，c）

以上の機能を持つプログラムを作成し動画の作成を行った。動画作成の手順については5.6節で述べたように，施設形状データベースに登録された環境に関するデータを使用し，JRR－3建屋内を歩行するロボットの動画を作成した。

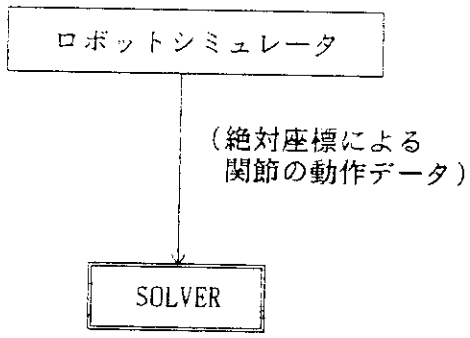


Fig. 5.10 Old system for image generation.

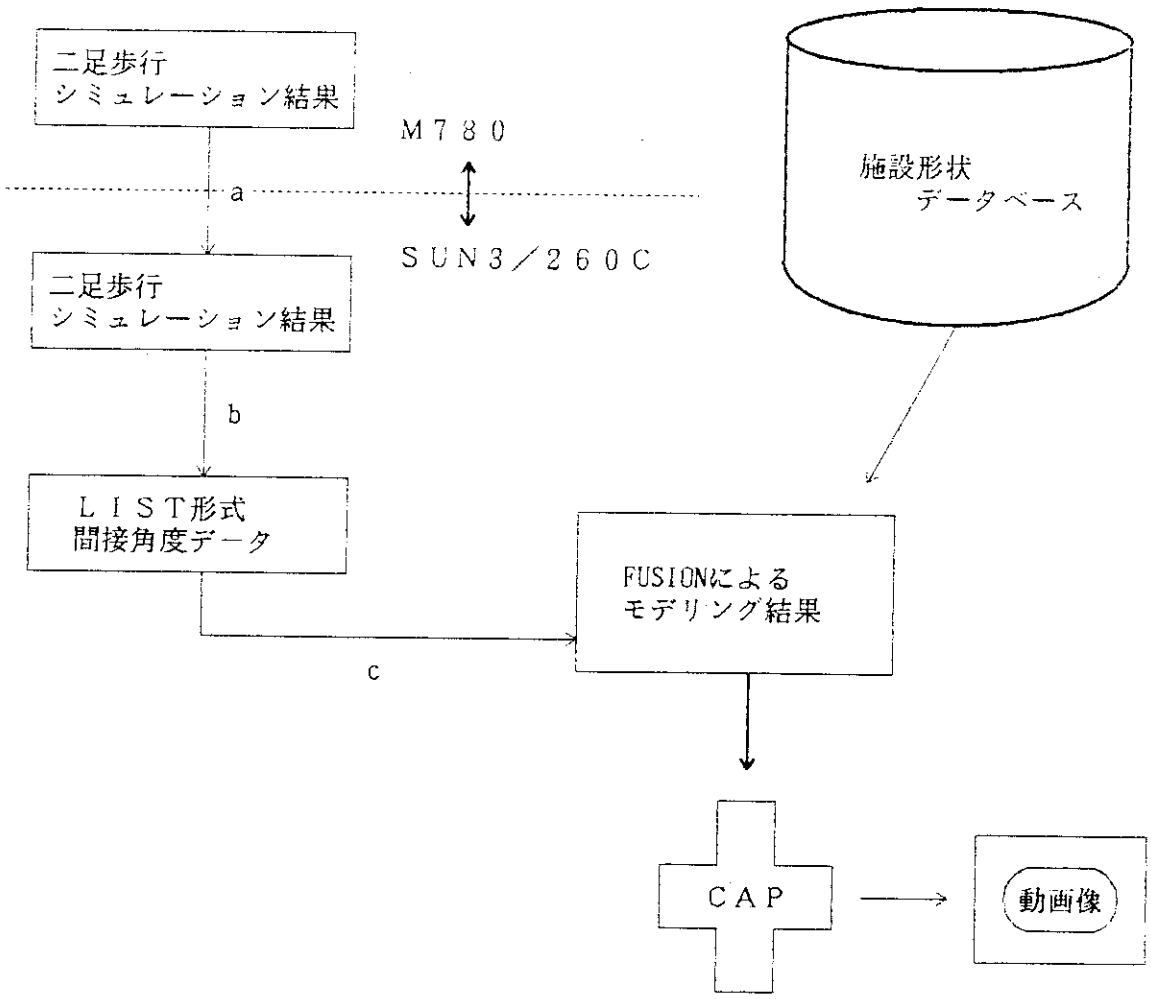


Fig. 5.11 New system for image generation.

5.9 おわりに

63年度は62年度より開始したJRR－3建屋に関する施設形状データベースの入力作業を継続して行った。62年度入力分の一次冷却系配管に加え建屋の壁や床の入力を終了し，現在地階の周辺機器を対象に入力を行っている。周辺機器入力終了後は原子炉上部に位置する監視室等の入力を行う予定である。また，シミュレーション結果の映像表示を行うために，ソフトウェアとしてモデリング・ソフトFUSIONを，動画作成システムとして並列計算機CAP，A－50，ビデオ・システム等の機器導入を行った。また，これらの導入した機器を使用してJRR－3建屋内を二足歩行ロボットが歩く様子を動画化し，約3分間のビデオを作成した。（Photo.5.1）

64年度は，JRR－3建屋地下一階計器室を中心に詳細な機器データ入力を行う。また，導入した動画作成システムについては映像生成技術及び動画作成技術の修得を目指す。さらに現在，模擬人間が行動する際の各動作について装置誘導型のものが検討されているため，行動計画を行う知識ベースと施設形状データベースのインターフェイスについての考慮が今後必要となってくる。このため，施設からロボットに与える情報を施設形状データベースに追加するためのデータ構造の在り方，データの内容について検討する必要がある。

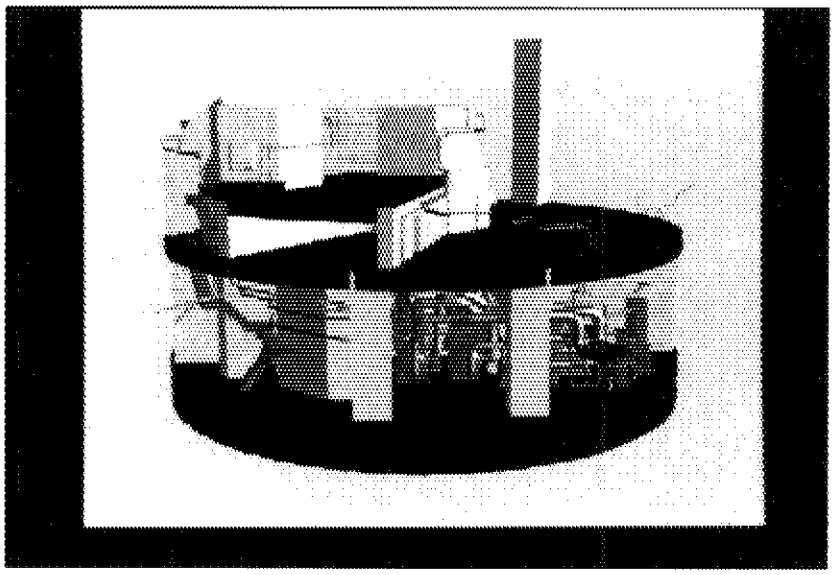


Photo.5.1 A color image of JRR-3 nuclear plant.

参考文献

1) 例えば，甲谷："リアルな画像を作るレイ・トレーシング法"，NIKKEI COMPUTER GRAPHICS，Jul.，1987

2) 村上："セルラアレイプロセッサCAPによるレイ・トレーシング"，情報処理学会グラフィクスとCAD研究会，Jul.，1986

3) 村上："（言語＋対話）型CGモデラ：FUSION／MODELER"，情報処理学会グラフィクスとCAD研究会　第1回集中研究会，Aug.，1987

4) 佐藤："高並列計算機CAPとその応用"，FUJITSU，Vol.38，No.3，1987

5) 石黒美佐子，藤崎正英："二足歩行ロボット運動学的シミュレーション"，JAERI-M，1988.4

6) 村上公一，池坂守夫，白石博："高並列計算機CAP－256によるコンピュータグラフィクス"，FUJITSU VOL.39，No.3，1988

6. モンテカルロ計算装置の概念検討

昭和６３年度は，ベクトル計算機改造型の高速モンテカルロ計算装置についての概念検討を行った。その具体的な作業内容は，原研仕様の改造概念案の有効性に関し，機種の異なるベクトル計算機に対する２件の調査，及びベクトル計算機改造型モンテカルロ計算装置の性能評価用ソフトウェア・シミュレータの作成である。このモンテカルロ装置の目標性能は,ベクトル・プロセッサのスカラ処理能力の５～１０倍である。調査の結果，シミュレータによる評価の方法と結果については，資料が大部であるので別にまとめて報告することとし，ここでは原研提案の改造概念案について説明し，最後に6.9節で調査結果とソフトウェア・シミュレータによるシミュレーション結果について簡単に述べる。なお，調査のテスト・データの対象としたのは，中性子・光子輸送のモンテカルロ計算コードKENO Ⅳ，MORSE−DD，VIM,MCNPであるが，調査期間の制約から結果については主としてKENO Ⅳ，MORSE−DDが主となっている。

6.1 中性子・光子輸送計算の代表的モンテカルロ・コード

中性子・光子輸送問題の数値解法は，決定論的手法と非決定論的手法に大別される。決定論的手法の代表的な数値解法としてＳｎ法，直接積分法，有限要素法などがある。他方，非決定論的手法として，モンテカルロ法が挙げられる。

モンテカルロ法は，決定論的手法に比べて，複雑な幾何形状の体系を精度よく表現でき，体系の幾何形状の次元，取り扱うエネルギーの群数に計算の量がほとんど依存しないなどの利点がある。

日本原子力研究所（以下，原研と省略する）においても，高速臨界集合体における実験の解析，ＪＲＲ－３（Japan Research Reactor−3）の改造における炉心全体の計算，核融合炉工学系の実験の解析，臨界安全解析，そして各種コードの評価などにモンテカルロ法によるコードが使用されている。

モンテカルロ法は母集団から代表粒子をサンプリングし，粒子の振る舞いを模擬することによって物理量の計算を行う手法である。例えば，遮蔽及び臨界計算においては粒子が媒質を構成する核種との反応によって吸収されたり，あるいは体系外へ漏れたりするまでの粒子の振る舞いを模擬することによって粒子束，増倍率などの計算を行う。

計算精度を向上させるためには，サンプリングする粒子の数を増やすと計算時間がかかるのがモンテカルロ法の短所で，計算時間の短縮はモンテカルロ・コードに残されている大きな課題の一つである。この課題には，二つの側面からのアプローチが行われている。一つは，分散低減法の改良を行って計算時間の短縮をはかるアプローチである。もう一つはコードをより高速化の可能性を持つアーキテクチャーのマシン向けに最適化することである。更に進んだアプローチとしてモンテカルロ・コード専用の並列コンピュータ・アーキテクチャーの提案[1]，開発，ベクトル改造型モンテカルロ計算機の提案[2]がある。ここで説明するのは後者についてで

ある。

モンテカルロ法には，中性子エネルギーをいくつかの群に分けて近似する多群理論にもとづくモンテカルロ法と，中性子のエネルギーを連続量として取り扱う連続エネルギー・モンテカルロ法とがあり，前者の代表的なコードとしてKENO Ⅳ [3]，MORSE [4] などがあり，後者の代表的なコードとしてMCNP[5]，VIM [6]などがある。多群理論コードで使用する群定数は媒質の各エネルギー群間における平均実効断面積である。そのため非均質性の高い系においては群定数作成時の精度の問題から中性子のランダム・ウォークの模擬が十分な精度で遂行されない場合が生ずる。非均質体系の解析が多く要求されるにつれてこの要求を満足する連続エネルギー・モンテカルロ・コードの使用も今後増加することが予想される。

6.2 モンテカルロ・コードのベクトル処理の問題点

光子輸送計算については現在のアーキテクチャーのベクトル計算機で高速化されたという報告[7][8]もある。しかし，世界的に広く使用されているモンテカルロ・コードKENO Ⅳ, MORSE, VIM，MCNPなどの実規模施設用かつ汎用プロダクション・コードでの我々のベクトル化経験はTable 6.1に示すとおりであった[9]。この章では高速化達成が困難である原因について説明する。

先ずベクトル処理の原理であるが，Fig. 6.1のように従来のスカラ計算方式ではひとつの演算に4 τ クロック必要となるところを，演算回路を分割して独立に働かせる。そしてここにデータを連続的に流すことによって1 τ クロックで結果を得ることができる。4セグメントはわかりやすさのためで，実際にはもっと多く，例えばFACOM VP−100では13セグメントとなっている。

スカラ計算量のうちベクトル処理できる部分の比率をベクトル化率と呼ぶが，それとコードの速度向上比の関係をベクトル計算機の性能αをパラメータとして表現したのがこのFig. 6.2である。これからわかるとおり，どのように性能のよいベクトル計算機であっても，コードのベクトル化率が低ければ速度向上は達成出来ない。実用的なモンテカルロ・コードKENO Ⅳ，MORSE，MCNP，VIMなどのベクトル化率は60〜70%にとどまっている。したがって速度向上は期待できない。何故ベクトル化率が低いのか，その解決のために必要な機能について説明する。

Fig. 6.3は筆者らがKENO Ⅳコードのサンプル入力に使った臨界実験の体系である[2]。かなり複雑な体系であるからモンテカルロ計算高速化の問題点を論じるには適当であろう。

Fig. 6.4はKENO Ⅳの計算の流れ図である[3]。各ボックスの計算内容はTable 6.2のとおりである。他のモンテカルロ・コードも大まかなところは，これと同じである。これをベクトル化することを考える。その方法は，例えば，この飛行距離を計算する部分に該当する中性子を集めて一度にベクトル処理することを考えてみる。このとき中性子を溜めておく入れ物をバンク(bank)と呼ぶことにする。

そうすると，スカラ計算では古い中性子位置をx，方向余弦(輸送ベクトルとx軸の)をUとして，新しい中性子位置x 1はFig 6.5のように計算される。これをベクトル計算では同じ

くFig.6.5のように複雑になる。間接添字，DOループ初期化，変数の番地計算がオーバーヘッドとして現われてくることがわかる。このようなオーバーヘッドはKENO Ⅳの場合は40%程度計算時間の増加をもたらす。これは大部分スカラ計算されるから，KENO Ⅳの大部分がベクトル計算可能で，その時間が10分の1になったとしても，全体としての速度向上倍率は2倍にしかならない。しかも，このオーバーヘッドは純粋に計算方法の変更によるものであって，物理的問題とは全く無関係であることに注意されたい。同じくVIMの例をFig.6.6に示す[10]。

このような問題を含め，以下の(1)～(9)はいずれも中性子輸送計算のモンテカルロ・コードで顕著に現われる（ただし，多数の中性子を一時に発生させるアナログ的計算では(2)，(3)は必ずしも正しくない）。

(1) スカラ計算で最適化

添字なしのスカラ演算で最適化されていて無駄な計算がなく（15年の歴史！），どのような修正も計算時間の増大に結びつく。

(2) 少ない中性子

誤差が世代数Nの平方根の逆数に比例するから，Nは比較的大きく100～200とし，サンプリング当りの中性子数を300程度と少なくする方法が採用されている。即ち，各事象毎のベクトル長は短い。

(3) 中性子の減少

体系から洩れたり，重みが小さくなって消滅して中性子の少なくなった状態が各サンプリング毎にかなり長く続く。即ち，ベクトル長が減少してゆく（例えばFig.6.7）。

(4) リスト・ベクトル

バラバラな番号の中性子がひとつのバンクに集められて同一処理の対象となるので，中性子の位置，エネルギー群番号等の参照が間接番地（リスト・ベクトル）表現となる。間接番地へのアクセスは時間がかかり，そのため短いベクトル長では，リスト・ベクトル表現での計算時間がオリジナルのスカラ版のそれと同程度になることが多い。ただし，リスト・ベクトル表現よりは直接番地表現のほうがコード全体の計算時間の点から有利な場合もある。

(5) 例外処理のオーバーヘッド

同一バンクのごく少数の中性子について例外処理を必要とする場合がほとんどである。例えばFig.6.8の輸送計算においてx，y，zの新しい位置が古い位置と同じになる場合などがそれである。Fig.6.8の計算では1357万回のうち約1850回発生している。

モンテカルロ・コードの一般形式は次のFig.6.9のようになり，Fig.6.8はこのうちの最内ループに相当する。

Fig 6.8のIF(0)制御行以下の文はまれにしか実行しない計算であるが，現在のベクトル・コンパイラはベクトル・レジスタの連結使用を重視し，無駄な命令を作り出す（ただし，これについては1F(0)の前後で分割してコンパイルするよう改善された）。またベクトル処理でIF文はジャンプ文とはならず，ビット集合の論理和，論理積，否定等の演算となる。IFが2重以上の入れ子となるベクトル処理は，ビット集合演算，収集／拡散演算数が多くなり，ベクトル化による他の演算の高速化を相殺してしまう。

(6) 不要な番地計算

ベクトル化によって添字付変数となり，そのために不要な番地計算を発生させることが多い。例をあげればFig 6.9のループにおいて②の番地計算は①のＩＦ文の後でおこなわれるべきものである（改善されているコンパイラもある）。

(7) 低い計算密度

Fig 6.4の各事象ボックス内のＩＦ文で区切られた計算単位の計算量が小さく，ベクトル効果が出てこない場合が多く，この点からもベクトル化効果が減じられている。

(8) ＤＯループ初期化演算

ループの初期化はスカラ演算で，これも最内ループではかなりのオーバヘッドとなる。

(9) 回帰式演算の出現

統計的総和を求める計算でベクトル化することによって回帰（再帰）性の演算が出現する。これについてはTable 6.3の幾何形状分類で説明しよう。

Table 6.3は先程の例題でKENO Ⅳで臨界計算をしたときのベクトル計算結果である。オリジナルのスカラ計算で３８０秒かかった計算が２６２秒となり，速度向上倍率は1.44倍となった。この1.４４倍を出すのに大変な苦労があった。アセンブラ語で書けば２倍は出るが，それはさらに大変な労力が必要で現実的ではない。２６２秒のうちベクトル計算できなかった幾何形状分類，事象分類，回帰式分類について簡単に説明する。

幾何形状分類とは，番号Ｊの中性子がどの形状の領域，例えば円管，球，etc. に属しているかを分類する操作で，Fortran文ではFig. 6.10(a)のような表現となる。事象分類とは，番号Ｊの中性子が，衝突，あるいは領域通過など，次にどのような事象の対象となるべきかを分類するもので，そのプログラム表現は幾何形状分類とほぼ同じである。回帰式はFig. 6.10(b)のように番号Ｊの中性子について領域N(J)，エネルギーM(J)での，例えば中性子束を計算するときに現われる。このときＪは異なっていても（M(J)，N(J)）は同じ番号対が発生することがあるため，ベクトル演算の対象とできない。これら３種の操作を高速化できない限りはベクトル化によるモンテカルロ・コードの高速化は望めない。

そこで幾何形状パイプラインという２次元的パイプラインを考えてみる（Fig. 6.11）。これは番号Ｊの中性子の領域番号ＩＧ(J)＝Ｋが連続的に与えられたとき，Ｋ＝１，２，…，に該当する中性子を２次元的スタックに分類してゆき，その数をカウントするものである。ベルト・コンベアに乗せられたミカンが大きさの順に次々とカゴに落されてゆくようなものである。Ｋ≠ｎとなる中性子は（ｎ＋１）番目のバンクへ分類され，（Ｋ≠ｎ）としてプログラムで次のループの分類へまわすことができる。このFig. 6.12の事象パイプラインもＫ＝０ or １という点を除いては同じ機能であるが，ただし，このときスタックの数は２つである。回帰式はFig.6.11のパイプラインを使ってM(J)，N(J)で分類し，その後ベクトル処理すれば速くなる。ただし，このときソース・プログラムを少し修正する必要がある。

ベクトル計算機にこれら２つのパイプラインを付加したときのモンテカルロ計算の速度向上倍率の推定例を先程の計算例（Table 6.4，Case1，Case2）で示す。

逆にいうと，球，円管状の金太郎飴のように入れ子の構造や重ね箱の構造で有れば対象

となる領域は1種類しかないから，Table 6.4の幾何形状分類に要する時間，即ち，Computed go to 文によるバンク分けの時間は不要になり，この57秒は半分の29秒に半減する。しかも重要なことは，この残った29秒の時間は，$x^2+y^2<R^2$？などの判定であってベクトル化に適していて，対スカラ演算比で4～5倍の速度向上が期待できる。事象分類についても同じで，領域通過か，領域内滞在（衝突）か，の2種類しかないから63秒が31秒程度に半減する上に，この31秒もベクトル化による高速化が可能である。しかも，このような簡単な幾何形状では，間接番地参照のリスト・ベクトルではなく，直接番地参照の線形ベクトルを使用できて，モンテカルロ・コードのようにベクトル長が短くて計算密度が低い場合はベクトル化部分がさらに2倍程度速くなり，領域毎の中性子を求めないことにすれば，上記条件のもとでの総計算時間は（3+3+20+40）=66秒となり，およそ6倍の速度向上となる。現在論文などで発表されているモンテカルロ計算高速化の例はこのような単純幾何形状のものである。筆者らがベクトル化したKENO Ⅳコードを変えればこのような高速化は可能である（リスト・ベクトルを使っても3.5倍の性能向上が得られた）。しかし，それでは汎用性に欠けるし，このような方法が適用するのも幾何形状がせいぜい1～2種類までで，2～3種類以上の幾何形状では，条件分岐が多くなってベクトル化が必ずしも有利とはいえなくなる。

前述の幾何，事象分類がベクトル化されない限り，いい換えればモンテカルロ・パイプラインがない限り，中性子輸送のモンテカルロ・コードは速くならない。それを別の例で示す。

Fig.6.8はモンテカルロ計算のうちではベクトル処理の性能がでる中性子輸送計算の部分であるが，これで3～4倍の速度向上となる（平均ベクトル長30～70）。KENO Ⅳでは最も性能向上がある散乱角計算部分では，3～5倍の速度向上である。したがって全体としてはさほどの速度向上は期待できない。散乱角計算でさほど速度向上がないのは分母がゼロとなる場合の条件分岐が入るせいでもある。

メモリ増加も大きな問題である。KENO Ⅳ, MCNP, MORSE，VIMの4コードのベクトル化によるメモリ増加はTable 6.1のとおりで，これからみると一度（1バッチで）に400～8000の中性子を使用することは現実的ではない。数10MBのメモリをもつベクトル計算機は存在するが，大きなメモリを要するジョブは夜間ジョブとなって回転時間が長くなる。それよりも10分程度のスカラ計算ですぐ結果を得たくなるのが人情である。以上からベクトル計算では300～500中性子／バッチの指定が多く使用されるが，このとき計算速度を上げるためには，(i)DOループ初期設定と変数番地計算の高速化，又は並列処理，(ii)粒子の遭遇する幾何形状，事象の分類，粒子の存在する領域の判定を高速で行うモンテカルロ・パイプラインなどの機能が非常に重要である。

既に述べたように，ベクトル計算機用に開発した中性子・光子輸送計算用モンテカルロ・コードで高速化が達成されたという報告もある。これらのコードは上述の(i)，(ii)の機能を持つ計算機では計算時間のより一層の短縮が可能となろう。

Table 6.1 Memory and speed increase of Monte Carlo codes.

コード名	ステートメント数（千枚）	メモリ増加	速度向上率	工数（人月）
KENO－Ⅳ	7.6	0.84MB→1.7MB	1.5	30
MORSE－DD	14.3	1.2 MB→5.8MB	1.7	20
VIM	24	1.0 MB→6.3MB	1.4	12
MCNP	34	1.4 MB→7.0MB	1.2	16

Table 6.2 Algorithms in each event box.

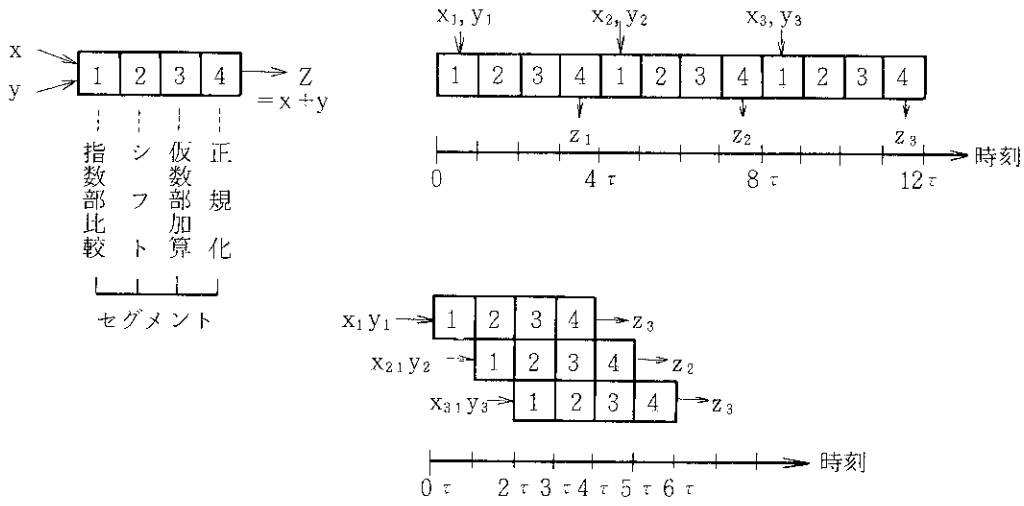
事　象	説　　明
PATH	中性子の輸送距離計算
FINBOX	中性子が反射体からコア内部を通過するときの座標変換，中性子が存在するボックスのx，y，z識別番号の決定
POSIT	計算され輸送距離の中性子が他の領域へ移っているかどうかの判定
IN－CROS	中性子が内側の次の領域へ移っているかどうかを判定し，そうであればx，y，z座標の更新，消費された輸送距離の決定
OUT－CROS	INWARD CROSSINGと同じ。ただし外側
ARRAY	中性子がボックスの境界を通過したかどうかを判定。そうであればX，Y，Z識別番号の更新，中性子が領域外へ出たかどうかの判定
ALBEDO	アルベド反射体に指定されている配列の表面を中性子が通過したときにアルベド反射体から戻ってくる中性子のエネルギー群，重み，方向余弦の計算
XSEC	中性子が衝突を起こしたときの中性子の重みの増減処理，中性子の生成・消滅処理，散乱角の計算，エネルギー群の決定，統計処理用データ収集
	注　この他に一様乱数の生成に5％程度必要

Table 6.3 Categorized computational time (KENO IV).

Category \ Time(sec.)	Current Computation Time
(1) Classification of Geometries.	57.0
(2) Classification of Events.	63.0
(3) Recurrence Expressions.	48.0
(4) Others(scalar).	20.0
(5) Vectorized Expressions.	74.0
Total Time	262.0(sec.)
Speedup Ratio	1.44

Table 6.4 Estimated computational time with Monte Carlo pipelines.

Category \ Time (sec.)	Current Computation Time	Improved Computation Time	
		Case 1	Case 2
(1) Classification of Geometries.	57.0	19.0 (3 times)	11.4 (5 times)
(2) Classification of Events.	63.0	21.0 (3 times)	12.6 (5 times)
(3) Recurrence Expressions.	48.0	24.0 (2 times)	16.0 (3 times)
(4) Others(scalar).	20.0	20.0	20.0
(5) Vectorized Expressions.	74.0	74.0	74.0
Total Time	262.0(sec.)	158.0(sec.)	134.0(sec.)
Speedup Ratio	1.44	2.41	2.84



τ：マシン・クロック（1セグメントの作動に要する基本時間）

(a) 浮動小数点加算の手順（4セグメントの場合）

(b) 逐次処理（4τごとに1結果が得られる）

(c) パイプライン処理（時刻4τごとに1結果が得られる）

Fig. 6.1 Concept of a pipeline operation.

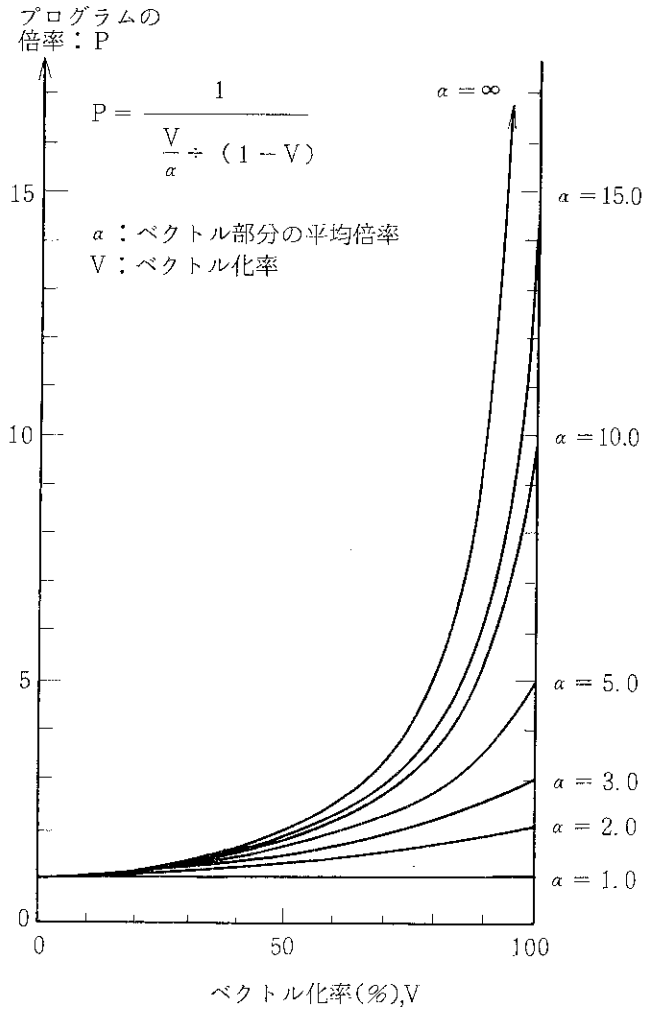


Fig. 6.2 Ratio of vectorization vs. processor performance.

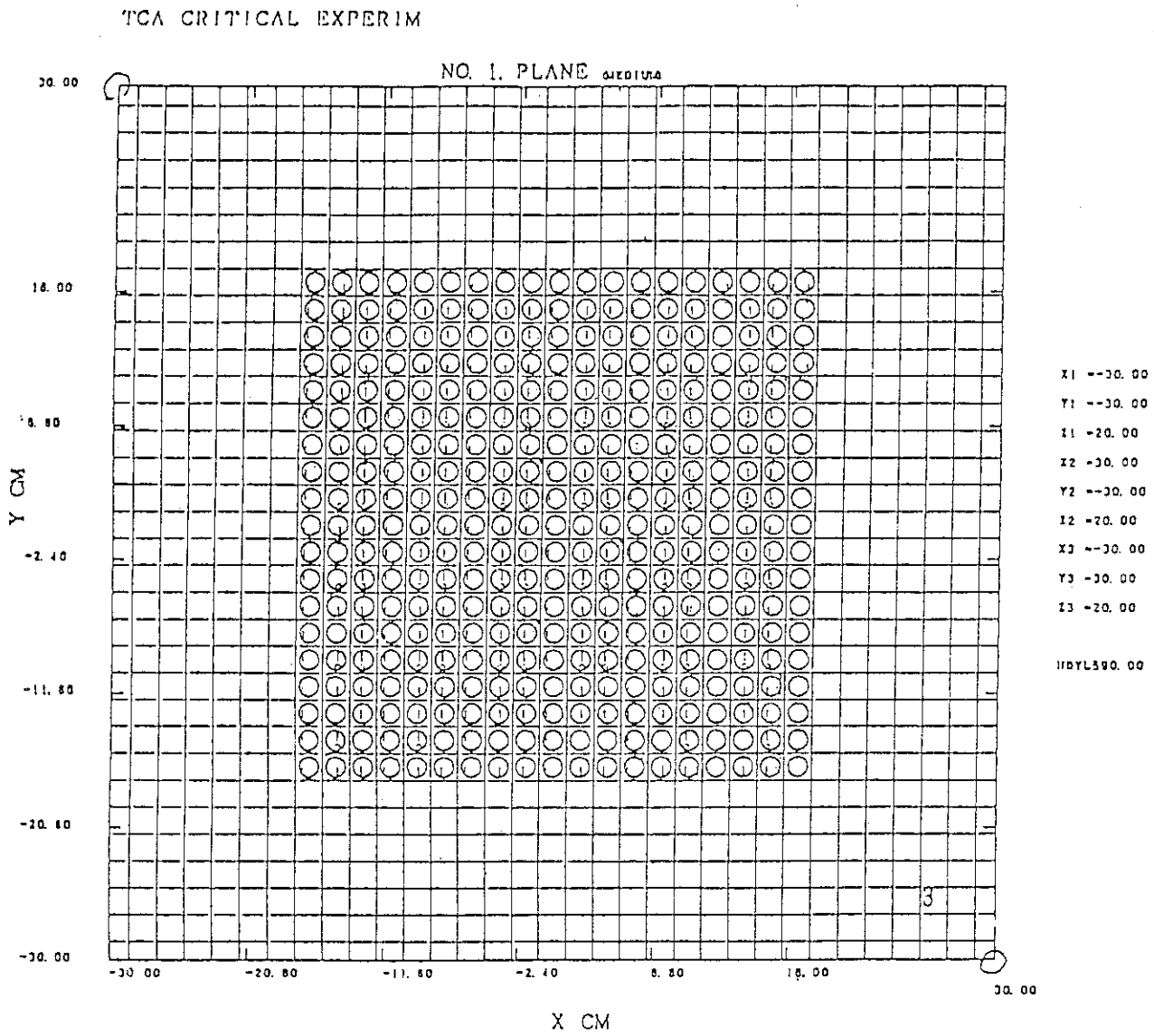


Fig. 6.3 Experimental scheme for KENO IV sample problem-2.

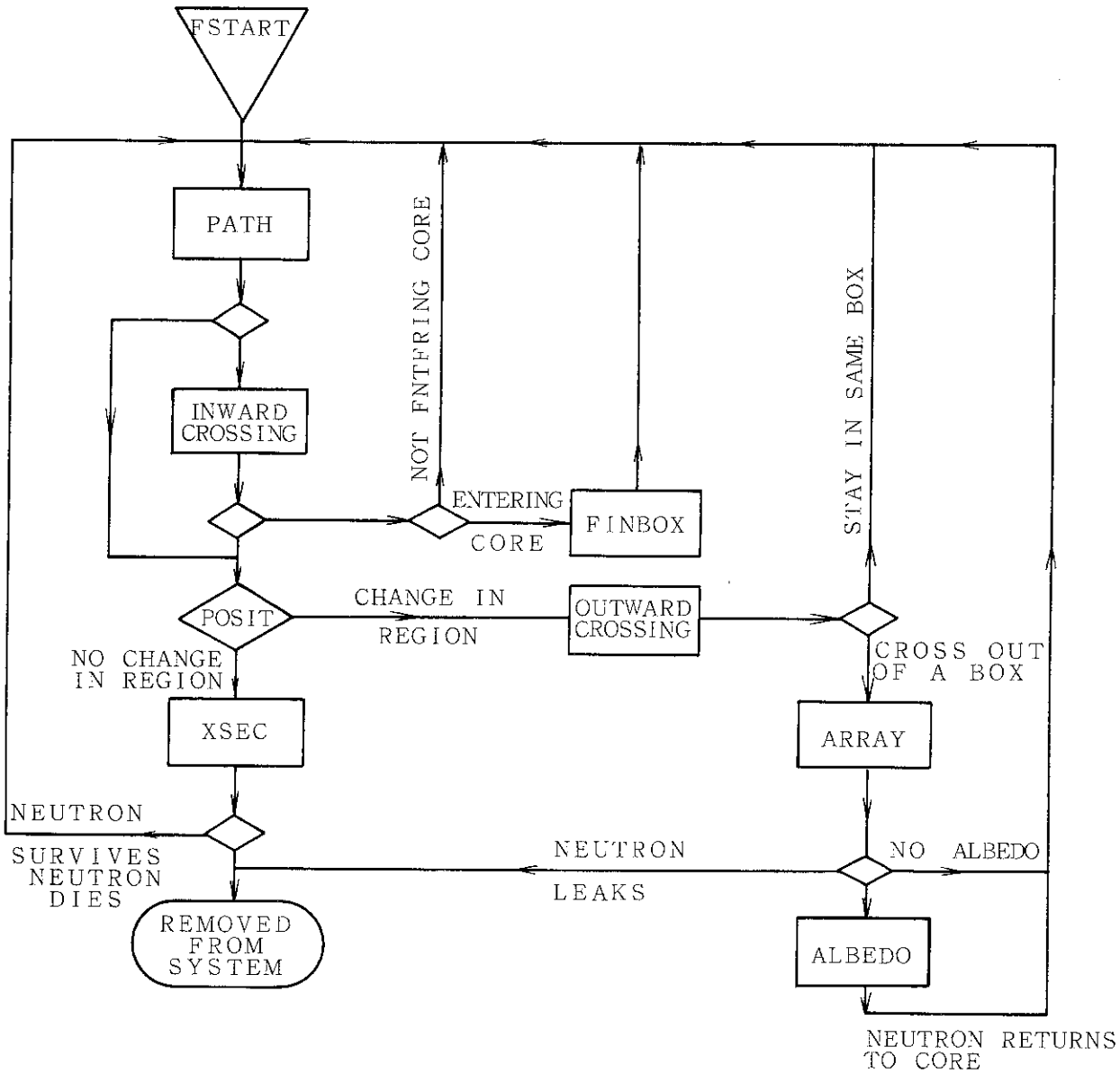


Fig. 6.4 Computational flow of KENO IV code.

```
X 1 = X + P A T H * U
        |
        ↓
      D O  10  I = 1, N P A T H
      J = P A T H B A N K(I)
   10 X 1(J)= X(J)+ P A T H(J)* U(J)
```

Fig. 6.5 Vectorized transport computation for x-axis.

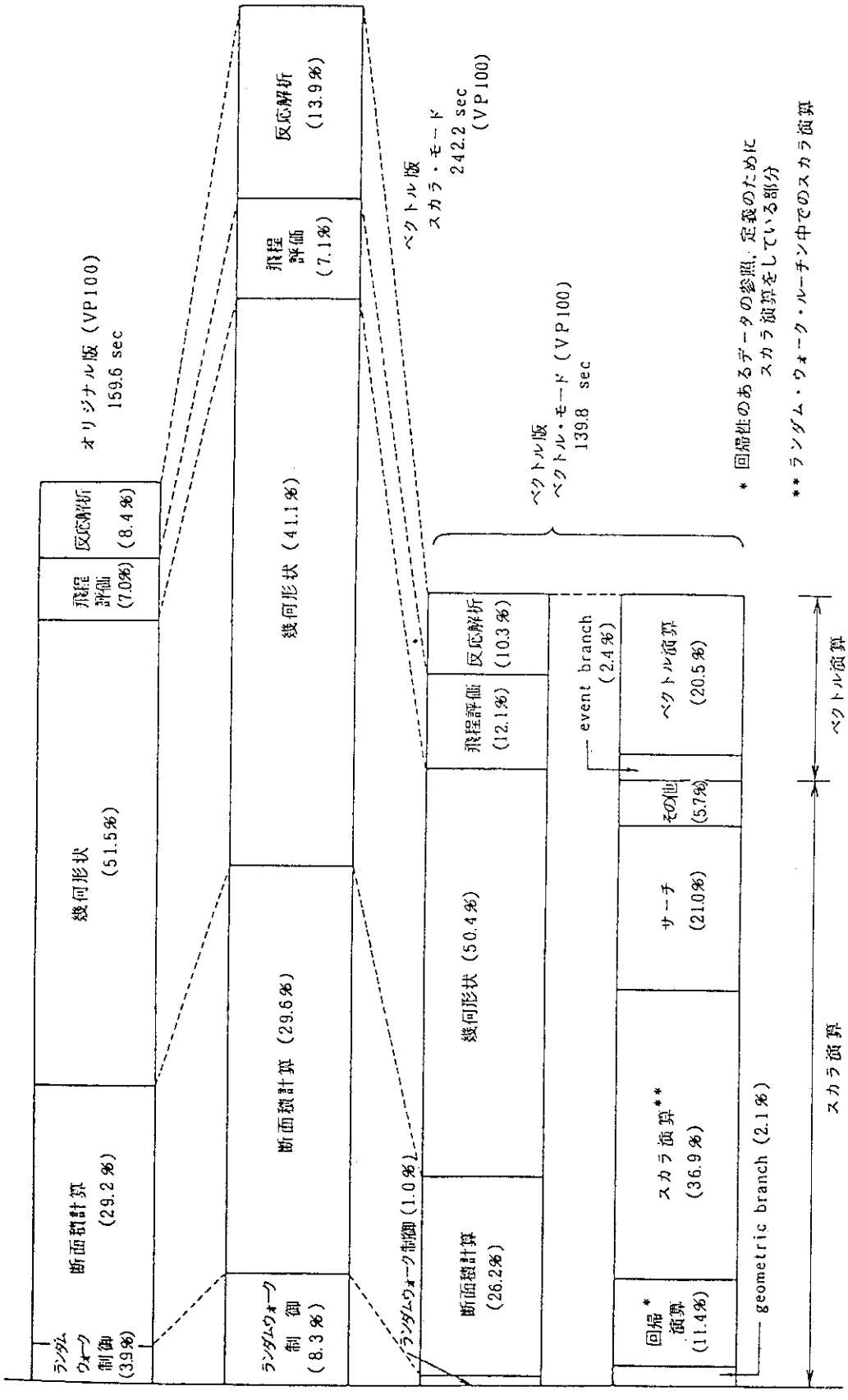


Fig. 6.6 CPU time classified by calculation types (VIM, Problem-2).

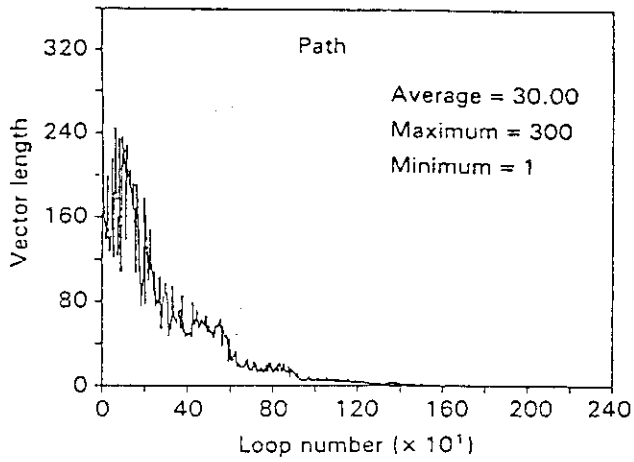


Fig. 6.7 Vector length in one generation (KENO, problem-2).

```
CVOCL LOOP,NOVREC
      DO 9320 IV=1,NPATH
      JV=PATH8A(IV)
      XR(JV)=MAT(X(JV))
CVOCL STMT,IF(99)
      IF(XR(JV).NE.0) THEN
      PTH(JV)=RPTH(JV)*RSIGT(XR(JV),IG(JV))
      ELSE
      PTH(JV)=BIG
      ENDIF
      X1(JV)=X(JV)+PTH(JV)*U(JV)
      Y1(JV)=Y(JV)+PTH(JV)*V(JV)
      Z1(JV)=Z(JV)+PTH(JV)*W(JV)
CVOCL STMT,IF( 0)
      IF(X(JV).EQ.X1(JV)) X1(JV)=X(JV)+SIGN(X(JV),U(JV))*(1.0E-6)
CVOCL STMT,IF( 0)
      IF(Y(JV).EQ.Y1(JV)) Y1(JV)=Y(JV)+SIGN(Y(JV),V(JV))*(1.0E-6)
CVOCL STMT,IF( 0)
      IF(Z(JV).EQ.Z1(JV)) Z1(JV)=Z(JV)+SIGN(Z(JV),W(JV))*(1.0E-6)
 9320 CONTINUE
```

Fig. 6.8 Example of exeptional treatment for neutron transport (KENO IV).

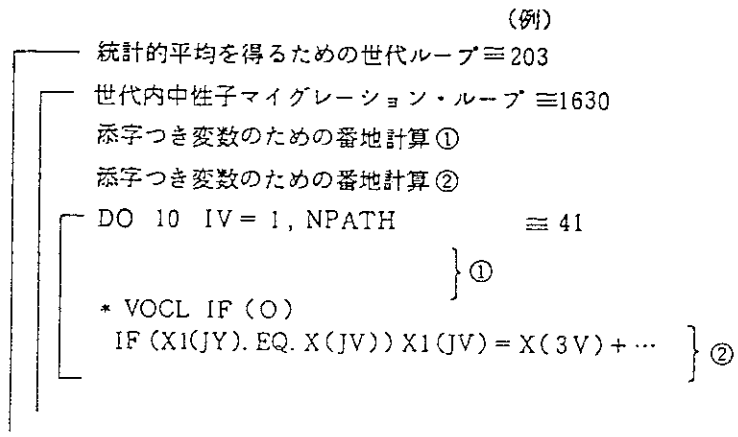


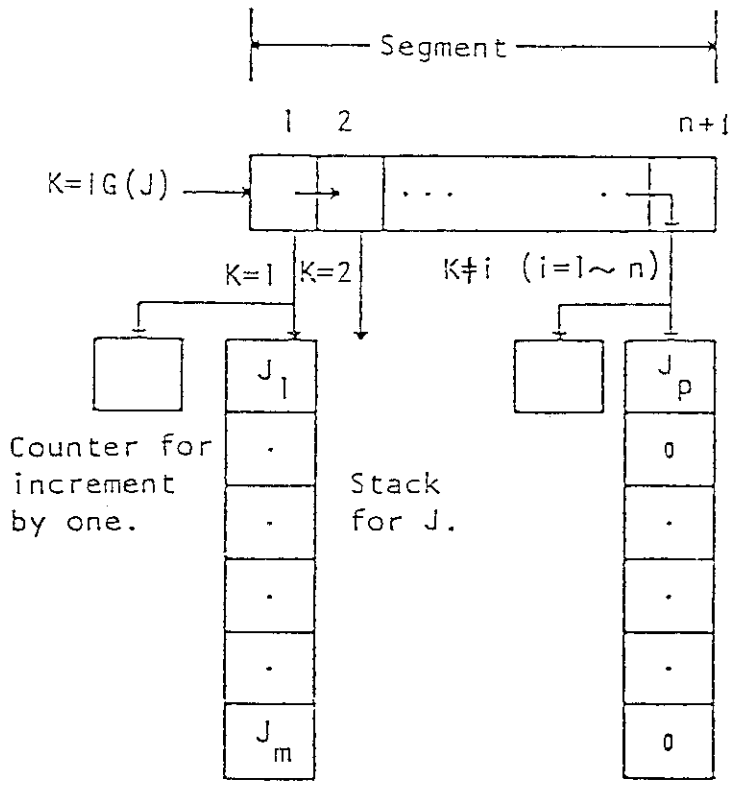
Fig. 6.9 General iterative procedures of Monte Carlo calculation.

```
      GO TO (10, ... ), IG(J)
   10 I = I + 1
      BANK(I) = J
```

Fig. 6.10(a) Fortran statements for region descrimination.

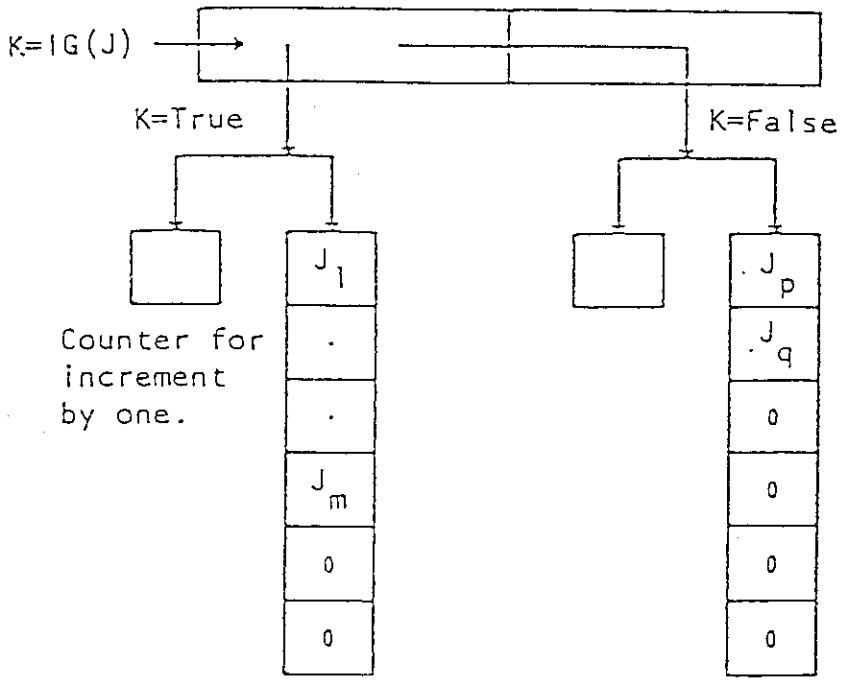
```
F(M(J),N(J)) = F(M(J),N(J)) + ...
```

Fig. 6.10(b) Fortran statement of recurrence expression for statistical summation.



幾何形状パイプライン

Fig. 6.11 Geometric pipeline.



事象パイプライン

Fig. 6.12 Event pipeline.

6.3 KENO Ⅳ, MORSE−DD VIM, MCNPコードのベクトル化版のサンプル入力による計算結果の概要

Table 6.1に4本の中性子輸送，又は中性子・光子輸送計算用モンテカルロ・コードの規模，ベクトル化による速度向上率，ベクトル化工数を示した。ベクトル化による計算速度向上が困難であるため，多くの努力が必要でそのため必然的にベクトル化工数が多くなる。

(1) KENO Ⅳ（多群エネルギー・モンテカルロ・コード）

Table 6.5（文献11，PP. 6）にサンプル入力でのFortuneによるオリジナル・スカラ版の計算コスト推定を示す。Table 6.6（同，PP. 33）にコードの幾何形状判定部分を特殊化・単純化し，かつ単純な入力例のVP−100による計算時間を示す。Table 6.5 KENO Ⅳの場合は計算種別のうち幾何形状分類と事象分類の処理時間が大きな割合を占めている。これらの困難を取り除くための幾何形状分類パイプライン，事象分類パイプライン及びDOループ初期化によるオーバーヘッドを低減する番地設定（APO）演算器の機能を6.5〜6.7節で説明する。

(2) MORSE−DD（多群エネルギー・モンテカルロ・コード）

Table 6.7（文献12，PP. 8）にサンプル入力でのVP−100での主要なサブルーチンの計算時間を示す。Table 6.8（同，PP. 34）に主要サブルーチンの処理時間を示す。Table 6.9（同，PP. 28）でわかるようにベクトル化した部分の計算時間はあまり短くなっていない。その原因は低いベクトル化率と短いベクトル長にある。同じことが他の3本のモンテカルロ・コードについてもいえる。この計算例では幾何形状分類と事象分類の時間比重が小さい。形状がそれほど複雑ではなく，中性子の振舞いも単純なせいである。

MORSEは，円球，直方体等の基本形状（プリミティブ）を組み合わせ所要の形状（ゾーン）を得る技法を採用している。各ゾーン毎に隣接するゾーン番号を保存する配列を持ち，当該ゾーンに隣接するゾーンが新しく判明する毎に，当該ゾーンの配列に隣接ゾーンの番号を記憶する。粒子の飛程と交差するプリミティブの捜索は，先ずこの配列内のゾーンから始める。この学習機能はすべてのゾーンについて不用の交差計算を行うことを避け，計算時間を短縮するためのものである。この部分の一部に6.5〜6.7節で説明する機能を使用して高速化を図ることができる。

(3) VIM（連続エネルギー・モンテカルロ・コード）

Table 6.10〜6.11にサンプル入力でのFortuneによる計算コスト推定と主要ルーチンの機能を示す（文献10，PP. 18〜19）。

Table 6.12と（同，PP. 38）にサンプル入力でのスカラ計算時間とベクトル（VP−100）計算時間及び計算速度向上倍率を示す。Fig. 6.13とFig. 6.14（同，PP. 39〜40）に計算種別による処理時間の比率を示す。取り扱う幾何形状が単純であることから幾何形状分類と事象分類の計算時間の比率は小さい。取り扱う核種の種類がやや多いこと（同，PP. 16），連続エネルギー・モンテカルロ・コードでエネルギー・グリッドが細分化されること（同，PP. 48）から，エネルギー・グリッド・サーチの時間比重が大きくなっている。スカラ版では4バンドに分割によって速く計算されていたエネルギー・グリッド・

サーチの時間がベクトル版では長くなった。ベクトル長を大きくするため，全エネルギーを一度に計算対象としたせいである。ＶＩＭの形状定義方法はMORSEと同じである。

(4) ＭＣＮＰ（連続エネルギー・モンテカルロ・コード）

Table 6.13（文献１３，PP. 11）にサンプル入力での主要サブルーチンのFortuneによる計算コスト推定を，Table 6.14（同，PP. 24）にＶＰ－１００での計算種別による処理時間比を示す。この計算例でも，取り扱う幾何形状が単純であることから，幾何形状分類と事象分類の処理時間は大きな比重を占めていない。ＭＣＮＰはベクトル化不可能な計算時間の割合が大きいが，その原因のひとつに消失粒子（Lost particles）検出のルーチンCHKCELから呼ばれるルーチンLGEVALがベクトル化不可であることによる。ＭＣＮＰは，２４種の基本形状面のいくつかで囲まれた領域（セル）で形状を定義する。CHKCELは粒子の空間座標と粒子を含むべきセルとの間の矛盾を検出するためのルーチンである。CHKCELはＭＣＮＰの多くの部分で頻繁に使用されている。CHKCEL中のＤＯループはLGEVALルーチンを呼び出すが，LGEVALのアルゴリズムはベクトル化できないため，当該ＤＯループもベクトル化不可となる。

無矛盾性検査のために，個々の粒子について，入れ子構造になってその粒子を取り囲むセルの長いリストが作られる。複雑な領域（形状空間）であればリストは長くなり，LGEVAL操作に長い計算時間を必要とする。粒子数が増加すれば形状空間の単純，複雑に関係なくLGEVAL呼出しの回数が多くなる。サンプル入力でのLGEVALのFortuneによるコスト比率は8.2％であるが，その領域定義はこの例題では単純であり，また初期設定粒子数も２５０個と小さい。リストを作成するCHKCELの推定コスト比率は１５％である。複雑形状空間，多い粒子数の場合は，この２つのルーチンの占めるコスト比率が急激に増加する。CHKCELそのものは幾何形状分類パイプライン及び通常のベクトル・パイプラインを使ってベクトル化可能である。

6.8でこのLGEVALのアルゴリズムをベクトル化する機能について説明する。

Table 6.5 Execution cost (number of instructions) of KENO IV code by a sample input.

NO.	ROUTINE	UNITS	LINES	ERR.	EXECUTIONS	COST	%	0.....1.....2.....3.....4.....5.....6.....7.....8..
0001	BEGIN	1	781	0	1	3346848936	83.1	**********
0002	CROS	1	503	0	7271488	675102106	16.8	**********
0003	INPUT	1	178	0	1	1154747	0.0	
0004	XSTAPE	1	243	0	1	1108734	0.0	
0005	FINALE	1	309	0	1	886179	0.0	
0006	CORSIZ	1	142	0	1	839472	0.0	
0007	START	1	431	0	1	838425	0.0	
0008	POSIT	1	70	0	13546	593920	0.0	
0009	NSTART	1	48	0	203	490327	0.0	
0010	KEDIT	1	126	0	1	422589	0.0	
0011	CLEAR	1	11	0	3	346946	0.0	
0012	BOX	1	51	0	1	267815	0.0	
0013	AREAD	1	253	0	36	134081	0.0	
	IREAD				89			
	FREAD				1142			
0014	VOLUME	1	153	0	1	130063	0.0	
0015	FREAK	1	89	0	1	74750	0.0	
0016	KENOG	1	405	0	1	72270	0.0	
0017	FILBOX	1	99	0	1	13298	0.0	
0018	KENO	1	568	0	1	12760	0.0	
0019	FHLPR	1	32	0	3	5451	0.0	
0020	MAIN	1	72	0	1	5160	0.0	
0021	MESAGE	1	18	0	1	4005	0.0	
0022	SAVE	1	20	0	5	3024	0.0	
0023	RDREF	1	113	0	1	1106	0.0	
0024	DATIM	1	44	0	1	1054	0.0	
0025	PULL	1	6	0	204	204	0.0	
0026	JOMCHK	1	485	0	1	152	0.0	
0027	TIMFAC	1	7	0	1	2	0.0	
0028	ALOCAT	1	8	0	1	2	0.0	
0029	STORE	1	32	0	0			
	STORE1				0			
0030	JOM7	1	70	0	0			
⋮		⋮			⋮			⋮
0055	ALBIN	1	196	0	0			
	ALBEDO				0			
0056	AJOINT	1	80	0	0			
			7847	0		4029357578		

Table 6.6 Speedup ratio of a simplified version.

Time (sec.) / Event	Vectorized Version		
	Vector Processor (VP-100)		Scalar Processor (M-380)
	Vector Time	Scalar Time	Scalar Time
PATH	1. 2	4. 8	7. 5
Inward Cross	———	———	———
POSIT	0. 6	———	2. 3
Outward Cross	0. 0	0. 0	0. 0
ARRAY	———	———	———
XSEC	3. 8	9. 2	35. 1
Random Number	0. 3	———	3. 3
Total Time	15. 1 (sec.)		48. 2 (sec.)
Speedup Ratio	3. 04		0. 95

Note. 1) Original scalar version for problem 1 required 46 seconds on M-380.
2) The zero value means a negligibly small computation time.

Table 6.7 Execution cost (number of instructions) of MORSE-DD code by a sample input.

NO.	ROUTINE	UNITS	LINES	ERR.	EXECUTIONS	COST	%	0.....1.....2.....3.....4.....
0001	GG	1	554	0	522555	131953760	46.6	I**********
0002	COLISN	1	144	0	64571	25433780	9.0	I*********
0003	G1	1	210	0	145890	18133266	6.4	I******
0004	RESTOR	1	286	0	8	14295443	5.0	I*****
0005	GOMST	1	119	0	145890	10040414	3.5	I***
0006	MACRO2	1	26	0	9	10023804	3.5	I***
0007	FLUXST	1	96	0	145890	9482850	3.3	I***
0008	TLE7	1	66	0	81319	8782452	3.1	I***
0009	TLE5	1	69	0	64571	6973668	2.5	I**
0010	NXTCOL	1	80	0	71164	5226199	1.8	I*
0011	REARAG	1	114	0	1000	5197577	1.8	I*
0012	GETETA	1	67	0	71164	5194972	1.8	I*
0013	MORSE	1	228	0	1	4141075	1.5	I*
0014	MACRO4	1	46	0	2	4031278	1.4	I*
0015	NBATCH	1	111	0	10	3178942	1.1	I*
0016	APMATX	1	75	0	1	2857458	1.0	I*
0017	SOURCE	1	509	0	5000	2219993	0.8	I
0018	TESTW	1	96	0	75139	2182676	0.8	I
0019	NSIGTA	1	72	0	139297	1950158	0.7	I
0020	BANKR	1	92	0	150912	1634974	0.6	I
0021	OUTPT2	1	28	0	1	1272616	0.4	I
0022	GETNT	1	92	0	15568	1254534	0.4	I
	STORNT				10568			I
	SETNT				1			I
0023	JNPUT	1	654	0	1	897177	0.3	I
0024	MAIN	1	55	0	1	882006	0.3	I
0025	MACRO1	1	62	0	9	675680	0.2	I
0026	MACRO3	1	49	0	3	658602	0.2	I
0027	STBTCH	1	113	0	10	595601	0.2	I
0028	INPUT1	1	601	0	1	563029	0.2	I
0029	GTMED	1	12	0	268439	536878	0.2	I
0030	INPUT2	1	185	0	1	503670	0.2	I
0031	VAR2	1	32	0	3	360972	0.1	I
0032	NRUN	1	560	0	1	323932	0.1	I
0033	SCORIN	1	503	0	1	273980	0.1	I
0034	JOMIN	1	171	0	1	268345	0.1	I
0035	GENI	1	177	0	1	262603	0.1	I
0036	GTNDSK	1	82	0	1	243167	0.1	I
0037	OUTPT	1	213	0	21	238125	0.1	I
0038	MSOUR	1	111	0	10	195200	0.1	I
0039	SOINP	1	187	0	1	72565	0.0	I
0040	SORIN	1	165	0	1	52161	0.0	I
0041	LOOKZ	1	116	0	5000	36682	0.0	I
0042	XSEC	1	331	0	1	16011	0.0	I
0043	GTVLIN	1	72	0	1	15139	0.0	I
0044	ENDRUN	1	38	0	1	7008	0.0	I
0045	ENRGYS	1	98	0	1	1393	0.0	I
0046	STRUN	1	17	0	1	1003	0.0	I
0047	TIMER	1	60	0	14	896	0.0	I
0048	IWEEK	1	36	0	2	232	0.0	I
0049	DATE	1	45	0	2	218	0.0	I

Table 6.8 Categorized computational time (MORSE-DD).

計算内容	現在の処理時間	パイプライン使用時
幾何形状分類	1.1	0.1
事象分類	4.3	0.8
回帰演算	0.7	0.7
その他（スカラ演算）	3.9	3.9
ベクトル処理部分	10.0	10.0
合計	20.0	15.5
速度向上率	1.00	1.35

Table 6.9 (a) Computational time of each subroutine in MORSE-DD code for sample problem-1.

	スカラー処理時 (sec)	ベクトル処理時 (sec)
GG	46.1	25.3
G1	6.1	3.3
RELCOL	9.8	4.9
FLUXST	3.3	3.4
PTHETA	1.4	1.4
EUCLID	0.8	0.7
SDATA	0.9	0.8
その他	14.6	14.1
合計	83.0	53.9

注) 比較したデータはいずれもベクトル版。オリジナルは84秒。

Table 6.9 (b) Computational time of each subroutine in MORSE-DD code for sample problem-2.

	スカラー処理時 (sec)	ベクトル処理時 (sec)
GG	13.7	7.9
G1	5.9	3.3
COLISN	2.5	1.1
GOMST	1.0	0.8
FUUXST	1.0	1.0
NXTCOL	0.8	0.7
MORSE	0.6	0.6
その他	4.6	4.6
合計	30.1	20.0

注) 比較したデータはいずれもベクトル版。オリジナルは21秒。

Table 6.10 Execution cost (number of instructions) of VIM code for a sample problem.

NO.	ROUTINE	UNITS	LINES	ERR.	EXECUTIONS	COST	%	0.....1.....2.....3.....4.....
0001	GGONE	1	948	0	1	962994794	45.9	**********
	GG				8967551			
0002	CROSS1	1	268	0	1	434134176	20.7	**********
	CROSS				466077			
0003	G1ONE	1	360	0	1	169116090	8.1	********
	G1				466077			
0004	REACT1	1	984	0	1	158560682	7.6	********
	REACT				400112			
0005	FLUXX1	1	209	0	1	147462044	7.0	*******
	FLUXX2				10			
	FLUXX				466077			
0006	ELAST1	1	273	0	1	104499443	5.0	*****
	ELAST				362045			
0007	NUTRNG	1	411	0	1	39788739	1.9	**
0008	UNRES1	1	177	0	1	37031487	1.8	*
	UNRES2				1			
	UNRES				202816			
0009	PRINT	1	580	0	10	25760573	1.2	*
0010	BEGET	1	506	0	10	2647739	0.1	
0011	BIRTH1	1	313	0	1	2070031	0.1	
	BIRTH				9010			
0012	GETIR1	1	205	0	1	1251059	0.1	
	GETIR				900			
0013	SPECT1	1	196	0	1	1196285	0.1	
	SPECTL				7163			
0014	POINTR	1	164	0	1	905701	0.0	
0015	SOURCE	1	409	0	11	787962	0.0	
0016	GENI	1	302	0	1	677319	0.0	
0017	JOMIN	1	169	0	1	594771	0.0	
0018	STATS	1	301	0	1	330812	0.0	
0019	CGDENS	1	265	0	1	326826	0.0	
0020	SITES	1	478	0	1	270291	0.0	
0021	READO	1	559	0	1	181400	0.0	
0022	INTRX1	1	141	0	1	171029	0.0	
	INTRAX				9000			
0023	LABX1	1	124	0	1	130852	0.0	
	LABX				6886			
0024	WIPOUT	1	84	0	0	123572	0.0	
	CLEAR				19			
0025	READ1	1	261	0	1	63596	0.0	
0026	SLUGIN	1	164	0	10	57441	0.0	
0027	GINPUT	1	63	0	1	40789	0.0	
0028	PUTM	1	150	0	15	21603	0.0	
	PUTB				38			
0029	START	1	328	0	1	21573	0.0	
0030	GIVE	1	31	0	21	21546	0.0	
0031	MAIN	1	191	0	1	15052	0.0	
0032	SIZEUP	1	174	0	1	10291	0.0	
0033	GETPNT	1	145	0	0	5940	0.0	
	GETN				19			
	DUMP				0			
0034	XSINPT	1	56	0	1	5105	0.0	
0035	PREMC	1	286	0	1	2799	0.0	
0036	TIMER	1	148	0	52	1812	0.0	
0037	BODVOL	1	87	0	23	1158	0.0	
0038	ANALYZ	1	159	0	1	1039	0.0	

Table 6.11 Major subroutines and their functions in VIM code.

ルーチン名	機　　能
NUTRNG	組合せ形状処理系専用のランダム・ウォーク制御ルーチン
CROSS1 (CROSS)	断面積処理系の制御ルーチン
UNRES1 (UNRES)	共鳴確率テーブルを検索し，使用すべき共鳴ピークの実効断面積を求める
REACT1 (REACT)	衝突解析系の制御ルーチン
ELAST1 (ELAST)	非熱領域の弾性散乱を解析し散乱後の中性子のエネルギー，飛行方向を決定する。
G1ONE (G1　　)	組合せ幾何形状系の制御ルーチン
GGONE (GGONE)	組合せ幾何形状系の work house ルーチン
FLUXX1 (FLUXX)	中性子の飛程による中性子束評価のためのルーチン

・括弧内のルーチン名は実際の処理用サブ・エントリ名であり，この場合メイン・エントリは引数のベース・アドレスの設定の機能しか無い。

Table 6.12 Computational time and speedup ratio of VIM code by sample problem-1 and 2.

	オリジナル版		ベクトル版 スカラモード		ベクトル版 ベクトルモード	
	単純形状	複雑形状	単純形状	複雑形状	単純形状	複雑形状
処理時間	103.1	159.1	144.8	242.2	73.8	139.8
時間比	1.0	1.0	1.4	1.5	0.72	0.88
速度比	1.0	1.0	0.71	0.67	1.39	1.13

(処理時間の単位は秒)

Table 6.13 Percentage of execution cost of major subroutines of MCNP code by a sample problem.

ルーチン名	機　能　概　要	コスト比（%）
ACECOL	弾性，非弾性散乱を解析する	0.1
ACENU	核分裂反応から生じる平均中性子数を計算する	― 0.1以下
ACETOT	粒子のエネルギーに対応する媒質の断面積を計算する	7.8
ANGL	粒子の方向余弦と通過する表面の法線ベクトルとのなす角度を計算する	0.4
CALCPS	DETECTOR 方向に散乱される粒子の確率密度を計算する	1.2
CHKCEL	粒子の位置とセル番号の間に矛盾がないかどうかを調べる	15.2
COLIDN	衝突解析処理の制御を行う	0.5
DDDET	衝突地点からDETECTOR までの距離及び方向余弦を計算する	0.8
HSTORY	粒子のランダム・ウオーク過程を制御する	6.0
JBIN	物理量蓄積領域中の使用すべき場所のインデックスをサーチする	2.1
NEWCEL	粒子が次に入るセルを検索する	1.7
SURFAC	セルの表面を通過して次のセルに移る粒子に対してIMPORTANCE SAMPLING, SURFACE SPLITTINGの処理を行う	0.5
TALLY	粒子束評価のために粒子のセル内でのトラック長の累積を行う	4.7
TALLYD	DETECTOR に対する寄与を累積する	3.5
TALSHF	個々のヒストリーの累積データを全体のヒストリーの累積データのテーブルに足し込む	6.9
TRACK	粒子のセル内での飛程長を計算する	29.8
TRANSM	散乱点からDETECTOR までの減衰因子を計算する	2.7
TRNSPT	中間結果の出力の制御を行う	0.1
	合計	84.0

・他に比較的大きなコスト比率をしめるルーチンとしてLGEVAL（8.3%）があるが，これはベクトル化不可能であった。

Table 6.14 Percentage of categorized computational time.

V／S *	分　　類	CPU時間比(%) **	備　　考
V	一般ベクトル演算	25.1	
V	event branch	11.2	
V	search	2.8	
V	geometric branch	2.5	
S	一般スカラ演算	41.6	
S	recurrence	10.0	+
S／V	interface	0.8	++
S	初期設定，結果の編集・出力	6.0	+++

* Vはベクトル演算，Sはスカラ演算を示す。

** 全体のCPU時間に対する個々の項目のしめる割合

\+ 短ベクトル長，ベクトル化困難な外部手続きが原因となるスカラ演算のループ

++ 回帰的なデータの定義・参照関係が原因のスカラ演算

+++ インターフェース整合用ルーチンのオーバーヘッド

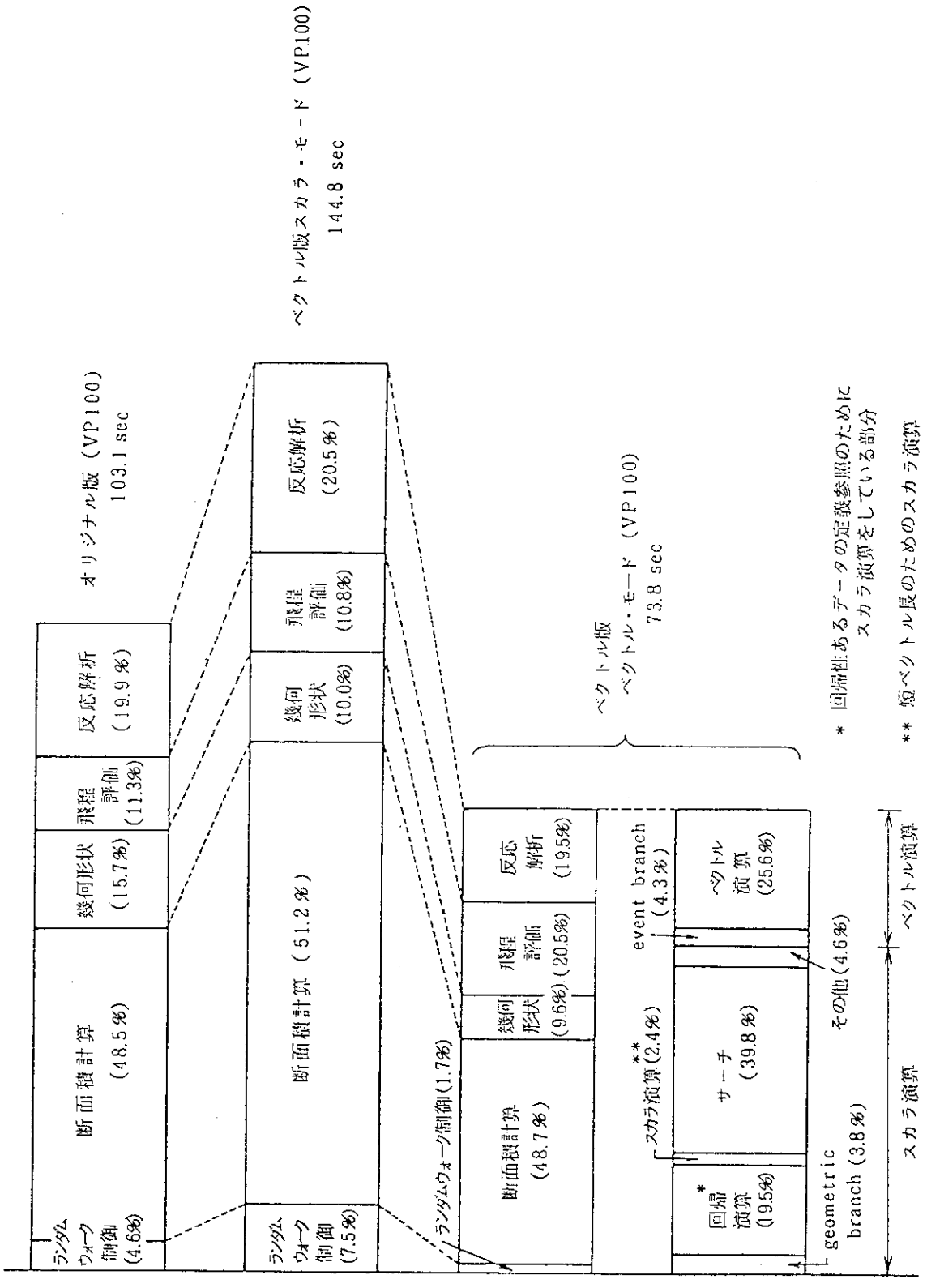


Fig. 6.13 CPU time classified by calculation types for sample problem-1(VIM).

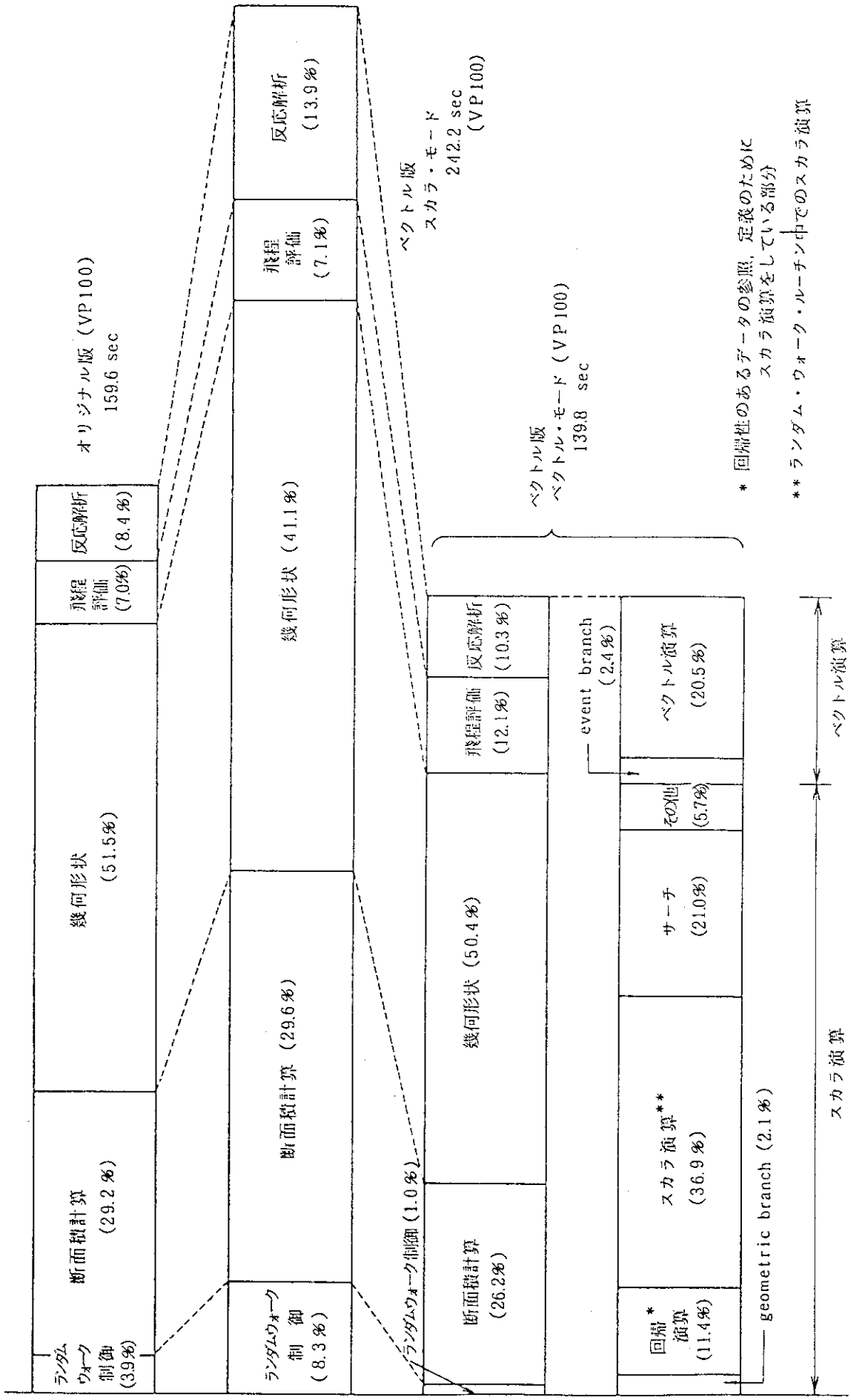


Fig. 6.14 CPU time classified by calculation types for sample problem-2(VIM).

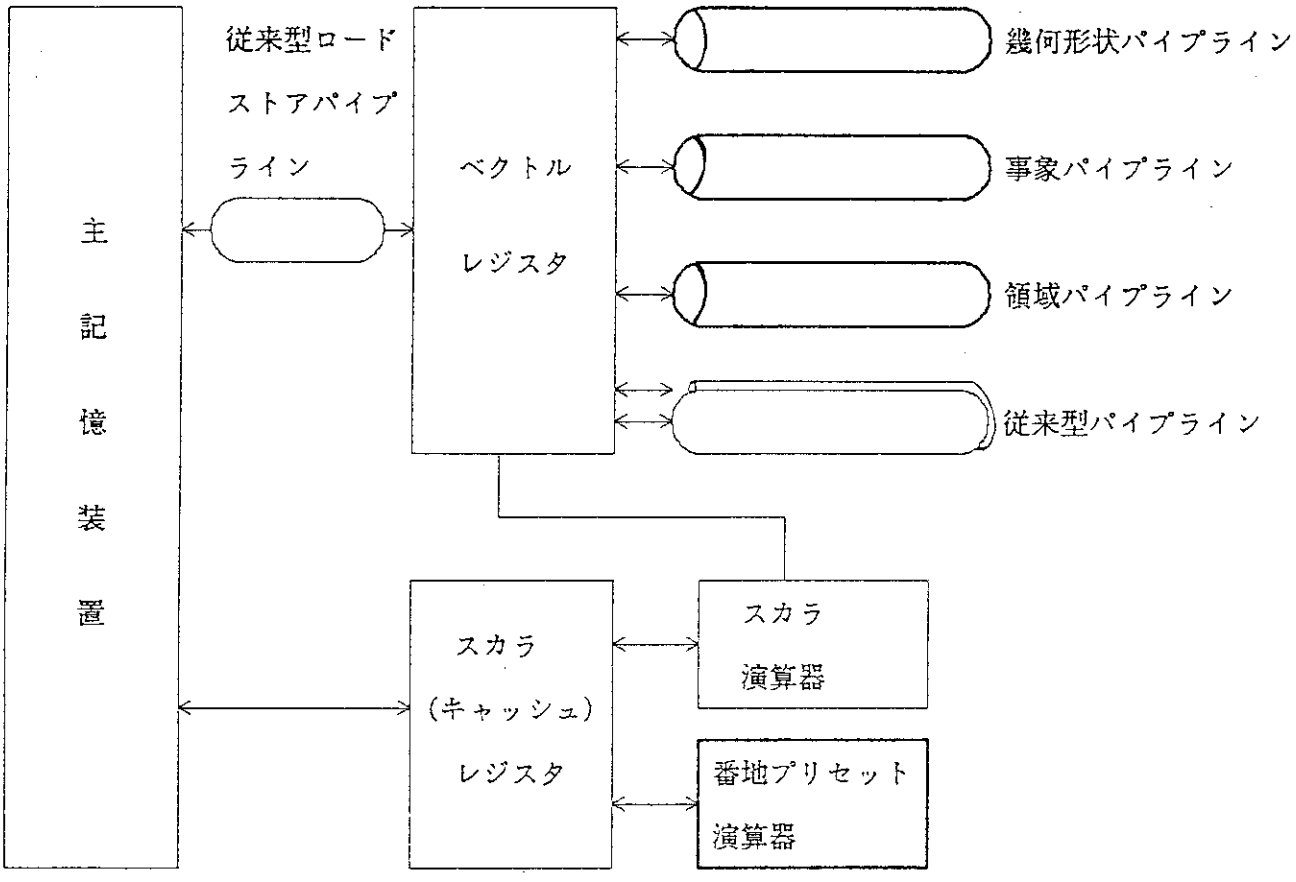


Fig. 6.15 Monte Carlo computer as a revised version of vector processor.

6.4 ベクトル計算機改造型モンテカルロ計算機

6.4.1 ベクトル改造型モンテカルロ計算機の意義

6.2～6.3で説明したように，現在のベクトル計算機のアーキテクチャでは中性子・光子輸送のモンテカルロ計算の高速化に難点がある。それらの難点は，中性子・光子輸送問題に限らず，複雑幾何形状空間内で条件分岐の多い物理現象を粒子モデルによって記述し，計算する問題に共通して現われる。難点解決にベクトル計算機を避けて多数のプロセッサを装備した並列処理計算機による方法も考えられている[1)]。

本報告では，ベクトル計算機改造型のモンテカルロ計算機を提案する。ここで提案するのは「超高速モンテカルロ計算装置」という名の汎用ベクトル計算機である。ベクトル計算機改造型モンテカルロ計算機の利点としては次の(1)～(3)が挙げられよう。

(1) ソフトウェア遺産の継承

コンピュータ・アーキテクチャに依存した原子力コードの開発も重要であるが，既に蓄積されているソフトウェア遺産は厖大なものである。例えばＭＣＮＰコードの開発には１２０人年を費したと公称されている。これらの遺産が小さな修正で高速計算可能なことは大きな価値がある。

(2) 汎用機としての利用

スカラ計算，ベクトル計算，モンテカルロ・パイプラインを使用する計算が混在する多重ジョブ処理が可能である。

(3) 容易な移植性

ベクトル計算機構造型のモンテカルロ計算機の上で開発された原子力コードは少しの修正で他のベクトル計算機，又は汎用機で利用可能である。

6.4.2 付加機能と目標性能

付加する機能，目標性能及び前提を次のように定める。

(1) ベクトル計算機改造型モンテカルロ計算機は，従来のベクトル計算機に，Fig.6.15の番地プリセット演算器，幾何形状分類パイプライン，事象分類パイプライン，領域検査パイプラインを付加することによって粒子型モデルの計算に関するベクトル計算機の弱点の大部分を解消しようとするものである。

(2) これらの演算器，パイプラインの付加によって一台のベクトル・プロセッサ（高速モンテカルロ計算装置）では５～１０倍の性能向上を，１０台のプロセッサを内蔵する計算機（超高速モンテカルロ計算装置）では５０～１００倍の性能向上を目標とする。

(3) 幾何形状分類，事象分類，領域検査パイプラインはそれぞれ並行動作禁止とする。また，これらいずれのパイプラインとも従来型パイプラインとの並行動作は行わない。並行動作禁止の方法は，従来のようにベクトル同期命令で行ってもよく，また回路内部の機能であってもよい。

(4) 番地プリセット演算器は，いずれのパイプライン及びスカラ演算器とも並行動作可能とする。そのための同期命令を新設する。

(5) 幾何形状分類，事象分類，領域検査パイプラインは1クロック当り1個の演算結果を出力する必要はない。

(6) これら付加機能に対応する命令はTable 6.15のとおりである。

(7) 6.5～6.7節で付加機能について説明する。付加機能の説明に使用するアセンブラ語は，提案する新設命令を除いてはFACOM Mシリーズ及びVPシリーズ計算機の命令語[14), 15)]を利用している。

Table 6.15 Added instruction set for Monte Carlo computer.

分類	命令	型	機能
番地プリセット	APWT	RX	番地プリセット演算器のウエイト命令
	APPT	RX	番地プリセット演算器のポスト命令
	APO	RX	指定番地命令の番地プリセット演算器での起動
幾何形状パイプライン	VSC	VR	ベクトル・レジスタ上数値の分類とカウント
	VSGP	VX	分類されたベクトル・レジスタ上数値のストア
	LGC	RX	GCカウンタへの数値のロード
	SGC	RX	GCカウンタ上の数値のストア
事象パイプライン	VEC	VR	ベクトル・レジスタ上の数値の分類とカウント
	VSEP	VX	分類されたベクトル・レジスタ上数値のストア
	LEC	RX	ECカウンタへの数値のロード
	SEC	RX	ECカウンタ上の数値のストア
粒子領域決定パイプライン	VRD	VR	ベクトル・レジスタ上の2値論理式の計算

6.5 番地プリセット演算器（APO：Address Preset Operation演算器）

6.5.1 ＡＰＯの目的

モンテカルロ計算では，個々の粒子は独立に振舞うことが多く，また，まとまった計算の単位は相互に無関係であることが多い。したがって或るＤＯループで計算を行っている時に，このＤＯループとは独立なＤＯループの変数の番地を設定可能な場合が多い。また現在実行中のＤＯループに順序として後続するＤＯループであっても，そのループの回転数（通常は粒子を保存するバンク長）のみが影響を受ける。その場合は，このループについてループ内変数の番地計算を予め実行しておくことが可能である。これのためにＡＰＯ（番地プリセット操作）演算器を付加し，モンテカルロ計算時間の多くを占める変数番地計算時間の短縮を図る。一般にひとつのループとそれに後続するループがソース・プログラム上で利用者によって指定されたとき，後続するループ内変数の番地を前もってセット可能かどうかはコンパイラで判断できる。

(1) コンパイラによって翻訳されたＤＯループの命令列は次のような形式となっている。

ＡＰＯ演算器で実行される③～⑤の命令をＡＰＯ命令と呼び，それ以外の命令をプログラム命令と呼ぶ。ただし，ＡＰＷＴ命令はプログラム命令として実行されることがある。

```
      ①  BC                  addr
      ②  EX      exaddr      B *，又はB **
      ③  APWT    addr        exaddr  0
    * ④  APIB                addr 2  B
      ⑤  APPT    addr        addr 3  B
      ⑥  APO     nextj       activate next preset oper.
   ** ⑦  VPPT
      ⑧  BC
```

ここで，

① ＢＣ　：ＤＯループを実行するかどうかの通常の分岐命令である。ループ回転数の値がゼロであれば②～⑤の命令列をバイパスする。

② ＥＸ　：ＡＰＩＢが実行済であればＶＰＩＢへ分岐する命令を実行する。

③ ＡＰＷＴ：Address Preset Wait 命令である。addr 1の内容がゼロであればno operation 命令である。内容が1であればスピン・ループ命令であり，addr 1の内容がゼロになるまでプログラム命令の実行は待ち状態に入る。このときは④のＡＰＯ演算器で実行されている。

④ ＡＰＩＢ：ＤＯループ内変数の番地計算のためのスカラ命令列である。ＡＰＩＢはベクトル命令を含まないように翻訳する。このＡＰＩＢ命令の最後に②の飛び越し先⑥をセットする命令が入っている。

⑤ ＡＰＰＴ：Address Preset Post 命令である。addr 1の内容を無条件にゼロにセットする。この命令はＡＰＷＴ命令とペアで現われる。

ＡＰＯ演算器はＡＰＯ命令の出現によって起動され，ＡＰＰＴ命令の実行

によって解放される。

⑥ ＡＰＯ　：次のＤＯループの初期設定のためＡＰＯ演算器を起動する。

⑦ ＶＰＩＢ：ＤＯループ内の通常のベクトル命令及びスカラ命令。

⑧ ＢＣ　　：ＤＯループ繰り返しの通常の分岐命令。

6.5.2 ＡＰＯ回路

ＡＰＯ演算器は，Load/Store，Add/Subtract/Multiply，Shift left/Shift right，And/Orなど番地計算に必要な命令及びＡＰＯ演算器に固有のＧ００～Ｇ１５の汎用レジスタから成る。これ以外の命令，レジスタは持たない。

(1) 最初のＤＯループ実行（ＡＰＯ機能使用が指定され，かつ実行されるとき）

(i) ＡＰＯ演算器が使用される場合

```
  EX        exaddr        (exaddr)＝B＊とコンパイル(＊への分岐命令)
 ＊APWT     addr          addr 1←1
  API B                   番地設定命令の実行
  L         G00, addr3
  ST        G00, exaddr   exaddr←B＊＊(＊＊への分岐命令)
  APPT      addr 1        addr 1←0
  AP        nextj         次の番地設定(address preset)命令実行のための
                          ＡＰＯ演算器の起動。Nextj は次のＡＰＷＴの番地
                          (ソースで指定)
```

(ii) プログラム命令

```
  EX        exaddr        (exaddr)=B＊＊
  APWT      addr 1        (addr 1)=0 ならNo. oper.，＝1ならスピン・ループ
  API B                   ┐
  L         G00, addr 3   │
  ST        G00, exaddr   │ B　＊＊なら命令実行をバイパス
  APPT      addr 1        │
  APO       nextj         ┘
→ ＊＊VPIB                ベクトル計算の実行
└ BC        ＊＊          ベクトル計算の繰り返し
    L       G00, addr 2   (addr 2)＝B＊　次回のＤＯループのための
    ST      G00, exaddr   (exaddr)←B＊　初期設定計算を可とする。
```

注意　ＡＰＯ演算器におけるＡＰＷＴ命令の働きはプログラム命令におけるＡＰＷＴとは異なり，addr 1に値1をセットする。また，ＡＰＯ演算器起動後にＡＰＯ演算器が受け付けたＡＰＷＴは，プログラム命令のＡＰＷＴと同じく，ＡＰＰＴの実行が終了するまではスピンループとなる。

(2) 2度目以降のDOループ実行(APO機能指定あるとき)

以下に見れば最初のDOループ実行と同じ論理である。

(i) APO命令

```
  EX      exaddr        (exaddr)
 *APWT    addr 1        addr 1←1
  API B                  番地設定計算の実行
  L       G00, addr 3   (addr 3)=B**
  ST      G00, exaddr   exaddr  ←B**
  APPT    addr 1        addr 1  ←1
```

(ii) プログラム命令

```
  EX      exaddr        (exaddr)=B**であればAPOをバイパス，次
                                   の命令を実行
  APWT    addr 1        ┐
  API B                 │
  L       G00, addr 3   │
  ST      G00, exaddr   │ B**であれば命令実行をバイパス
  APPT    addr 1        │
  APO     next j        ┘
 ┌→**VPIB                 ベクトル計算の実行
 └── BC     **            ベクトル計算の繰り返し
  L       G00, addr 2   (addr 2)=B*
  ST      G00, exaddr   exaddr  ←B*
```

(3) NEXT(J)指定され，かつaddress presetされていないとき

APO機能使用の指定がある1回目のループと同じ手順である。ただし，命令はすべてプログラム命令として実行される。

れは次のような場合に発生する。

```
      DO  10  I=1, N
          :
          :
 10   CONTINUE
C**   NEXT(20)
      IF(NMAX. GT. M)GO TO 40
 20   DO  30  J=1, MN
          :
          :
 30   CONTINUE
```

上記のＩＦ文が成立すれば，プログラムの制御は文番号40の文へ移る。このとき文番号４０で始まるＤＯループはプログラム中の何処かでaddress preset の指定があったとしても，その操作が実行されているとは限らない。

(4) NEXT(Ｊ)指定のないＤＯループ

通常のプログラム命令として翻訳され，実行される。命令列は

ＢＣ
ＡＰＩＢ
ＶＰＩＢ
ＢＣ

のタイプである。

6.5.3 利用上の注意

自身のＤＯループをNEXT(Ｊ)指定することは誤りではない。ただし，当該ループ内でaddress preset に影響ある変数の再定義が行われている場合，又は，当該ループのＶＰＩＢ実行終了後に，実行されたプログラム中の命令によって当該ループのaddress preset に影響ある変数が再定義されているかどうかの判断は利用者の責任とする。

6.6 幾何形状パイプラインと関連命令

6.6.1 幾何形状パイプラインの目的

幾何形状パイプラインの目的は既に6.2，Fig.6.11 等で触れたが，ここでは例によって説明する。Fig 6.16 は番号ＪＶの粒子の属する領域の形状番号ＩGEO(ＪＶ)で粒子群を分類し，粒子がその形状内に存在しているのか，又は領域外へ出ているのかを判断する部分を示している(文献１１，PP. 31)。この部分をベクトル化するには次の手順が考えられる。

(1) ＩGEO(ＪＶ)＝1，2，……，NTYPEに対応するＪＶを高速分類する。この操作をベクトル分類パイプラインで行う。ソース・プログラム上ではVGSORTなる名称の関数表現とする。同種の操作をスカラ演算で表現したものがFig. 6.17(文献１２，PP.29)である。

(2) 上記の分類後は分類されたサブバンク毎にFig. 6.16のＩＦ(・・・)GO TO nn 文

をTFBANK(JV)=. TRUE . と変更する。

(3) 論理変数TFBANK(JV)のすべてを事象分類パイプライン(後述)で高速分類してXSECBA及びCROSB2バンクに入れる。

(4) Fig 6.17 の如くサブバンク分けによって数多く発生するDOループの初期化オーバーヘッドはAPO演算器の利用によって最小化を図る。

幾何形状パイプラインは上記の目的で使用する(Fig.6.18)。

6.6.2 VGSORTの展開形

VGSORT(IGEO(J), B_1, I_1,……, B_n , I_n)

```
J       KGEOM ( I ) /*I              :sequentiral increment*/
LGC             GC00, J1             Load I1 value on geometric pipe
                                     counter C1
 ⋮
LGC             GCnn, In             Load In value on GCnn
VPT                                  Post vec.oper
VWT                                  Wait end of other vec. oper.
L               G00, n               Load vec. length to G00
VLVL            G00                  Load G00 on vec. len. reg.
 ┌VL             V000, KGEOM         Load J on V000
①│
 └VIL            V032, V000, IGEOM   Load IGEOM(J) on V032
VSC             V032                 Vector sort and count
VSGP            V064, B1             Vector store geometric.
 ⋮
VSGP            V224, Bn             Vector store geometric
VPT
VWT
SGC             GC00, I1             Store GC00 in I1
 ⋮
SGC             GC07, In             Store GCnn in In
```

nが8のとき，レジスタ指定最大数は8

nが16のとき，レジスタ指定最大数は16

n＝8又は16の選択はソース・プログラムのコメント的指定文で利用者に選択されてもよい。MCNPなどは24種の形状(面)分類がソース・プログラム上では指定されているが，実際に頻繁に使用されるのは5～6種である。したがってn＝8の指定で大部分の応用では十分であろう。この種の分類方法では実際に必要なバンク数よりもひとつ余計なバンクを設定しておき，目的外の値をそこへ保存しなければならない。回帰式表現の演算ではそれを積極的に

利用して効率を上げ得る場合もある。

利用者が利用可能なnは，６又は１４の固定形とする。

6.6.3　ＶＳＣ，ＶＳＧＰ命令

ＶＳＣ（ＶＲ型）

OP	EOP	R2	R1	

ＶＳＧＰ（ＶＸ型）

OP	EOP	R2	R1	X2	B2	D2

・Ｒ２は００～Ｆまでの範囲で００はGC00を，ＦはGC15を指す。

(1)　ＶＳＣ命令

(イ)　レジスタ指定最大個数が８のとき

Ｖ０３２のベクトル要素の下位３２ビットの内容が１のときは，Ｖ０００の対応するベクトル要素の下位３２ビットの内容をＶ０６４へ移す。このときＧＣ００カウンタの値を１だけ増加させる。

Ｖ０３２のベクトル要素の下位３２ビットの内容が６以上，又は１～５以外の数のときは，Ｖ０００の対応するベクトル要素の下位３２ビットの内容をＶ２２４へ移す。このときＧＣ０７カウンタの値を１だけ増加させる。

(ロ)　レジスタ指定最大個数が１６のとき

Ｖ０１５のベクトル要素の下位３２ビットの内容が１のときはＶ０００の対応するベクトル要素の下位３２ビットの内容をＶ０１６移す。Ｖ０１５のベクトル要素の下位３２ビットの内容が１４以上又は１～１３以外の数であれば，Ｖ０００の対応するベクトル要素の下位３２ビットの内容をＶ２４０へ移す。

(ハ)　生成される命令

直接番地指定のＩGEOM（Ｊ）をロードするとき命令列①は

```
VLM   M000, 0, =F"1"   Vector load mask on Vec. gen. arith. ser.
                       R2=0
VGS  VV000, 0, M000
```

とコンパイルされる。

(2)　ＶＳＧＰ命令

Vector Store Geometric Pipeline elements 命令の実行はＧＣの内容を参考にして実行される。

(イ)　ＧＣの内容が初期設定後も変化しない場合はstore命令は実行されない。

(ロ)　GC_n の値が１だけ増分すると４バイト増分の B_n 番地への格納される。このときベクトル要素の下位３２ビットが０である要素の保存は抑止される。B_n 番地の４バイト増分も抑止される。

```
      DO 9431 IV=1,NPOS
      JV=POSBAN(IV)
  430 IGEO(JV)= IGEOM(K(JV))
      GO TO(460,450,470,460,462,464,466,467),IGEO(JV)
  450 RSQ=X1(JV)*X1(JV)+Y1(JV)*Y1(JV)
      IF(Z1(JV).LE. XX(K(JV),2)    .AND.Z1(JV).GE. XX(K(JV),3)
     1   .AND.RSQ.LE. XX(K(JV),4))     GO TO 739
      GO TO 479
  460 IF(X1(JV).LE. XX(K(JV),1)
     +   .AND.X1(JV).GE. XX(K(JV),2)    .AND.Y1(JV).LE.
     1  XX(K(JV),3) .AND.Y1(JV).GE. XX(K(JV),4)
     +                                   .AND.Z1(JV).LE.XX(K(JV),5)
     2 .AND.Z1(JV).GE. XX(K(JV,.6))     GO TO 739
      GO TO 479
  470 RSQ=X1(JV)*X1(JV)+Y1(JV)*Y1(JV)+Z1(JV)*Z1(JV)
      IF(RSQ.LE. XX(K(JV),2))     GO TO 739
      GO TO 479
  462 RSQX=Y1(JV)*Y1(JV)+Z1(JV)*Z1(JV)
      IF(X1(JV).LE.XX(K(JV),2).AND.X1(JV).GE.XX(K(JV),3).AND.
     +   RSQX.LE.XX(K(JV),4)) GO TO 739
     GO TO 479
  464 RSQY=X1(JV)*X1(JV)+Z1(JV)*Z1(JV)
      IF(Y1(JV).LE.XX(K(JV),2).AND.Y1(JV).GE.XX(K(JV),3).AND.
     +   RSQY.LE.XX(K(JV),4)) GO TO 739
      GO TO 479
  466 TZ1=Y1(JV)*XX(K(JV),3)
      IF(ABS(XX(K(JV),3)).GE.5.0)TZ1=Z1(JV)*XX(K(JV),3)
      IF(ABS(XX(K(JV),3)).LE.2.0)TZ1=X1(JV)*XX(K(JV),3)
      IF(( X1(JV)*X1(JV)+Y1(JV)*Y1(JV)+Z1(JV)*Z1(JV)).LE.XX(K(JV),2)
     +    .AND.TZ1.GE.0.0) GO TO 739
      GO TO 479
  467 NHCYL=ABS(XX(K(JV),5))
      SGN=SIGN(1.0,XX(K(JV),5))
      GO TO(471,472,473,474,475,476),NHCYL
  471 IF(SGN*X1(JV).LT.0.0)GO TO 479
      GO TO 450
  472 IF(SGN*Y1(JV).LT.0.0)GO TO 479
      GO TO 450
  473 IF(SGN*Z1(JV).LT.0.0)GO TO 479
      GO TO 462
  474 IF(SGN*Y1(JV).LT.0.0)GO TO 479
      GO TO 462
  475 IF(SGN*Z1(JV).LT.0.0)GO TO 479
      GO TO 464
  476 IF(SGN*X1(JV).LT.0.0)GO TO 479
      GO TO 464
  739 NXSEC=NXSEC+1
      XSECBA(NXSEC)=JV
      GO TO 9431
  479 NCROS2=NCROS2+1
      CROSB2(NCROS2)=JV
 9431 CONTINUE
```

Fig. 6.16 Fortran statements for particle selection by geometry (KENO IV).

```
*VOCL LOOP,SCALAR
      DO 4010 IV = 1,NGGN¥
         JV      = NGGBK¥(IV)
            IF( (ITYPE¥(JV).LE.0).OR.(ITYPE¥(JV).GT.9) ) THEN
               WRITE(IOUT,2010) ITYPE¥(JV),IR¥(JV),NBO¥(JV)
               CALL PR(1)
               CALL ERROR
                                                          GO TO 4010

            END IF
            RIN¥(JV)  = PINF
            ROUT¥(JV) = -PINF
            GO TO (1101,1201,1401,1301,1401,1601,1701,1801,1711)
     ¥         ,ITYPE¥(JV)
                                                          GO TO 4010
 1101       N1100¥          = N1100¥ + 1
            J1100¥(N1100¥)  = JV
                                                          GO TO 4010
 1201       N1200¥          = N1200¥ + 1
            J1200¥(N1200¥)  = JV
                                                          GO TO 4010
 1401       N1400¥          = N1400¥ + 1
            J1400¥(N1400¥)  = JV
                                                          GO TO 4010
 1301       N1300¥          = N1300¥ + 1
            J1300¥(N1300¥)  = JV
                                                          GO TO 4010
 1601       N1600¥          = N1600¥ + 1
            J1600¥(N1600¥)  = JV
                                                          GO TO 4010
 1701       N1700¥          = N1700¥ + 1
            J1700¥(N1700¥)  = JV
                                                          GO TO 4010
 1801       N1800¥          = N1800¥ + 1
            J1800¥(N1800¥)  = JV
                                                          GO TO 4010
 1711       N1710¥          = N1710¥ + 1
            J1710¥(N1710¥)  = JV
-4010 CONTINUE
```

Fig. 6.17 Example of bank selection by Fortran statements.

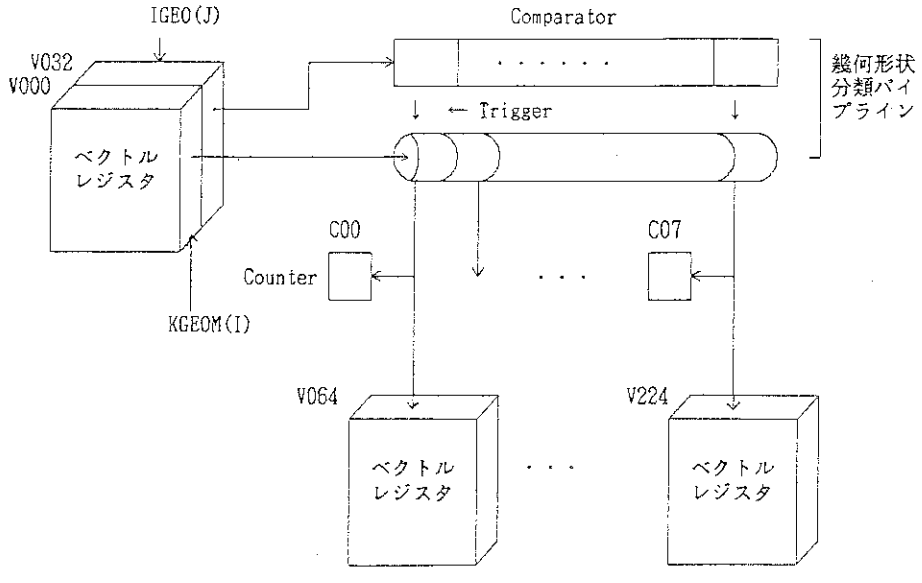


Fig. 6.18 Mechanism of geometric pipeline.

6.7 事象分類パイプラインと関連命令

6.7.1 事象分類パイプラインの目的

目的は論理変数の高速分類と数え上げである。その機能はソース・プログラム上ではVESORTと関数形で使用する。概念図をFig.6.19に示す。

6.7.2 VESORTの展開形

VESORTの展開形は次の形式である。

VESORT(IE(IEE(J), B_1, I_1, B_2, I_2)

```
      LEC    EC0, I1          Load I1 value on event counter EC0
      LEC    EC1, I2          Load I2 value on event counter EC1
      VPT                     Post the end of preceding vec. oper.
      VWT                     Wait until the end of prec. oper.
      L      G00, n           Load vec. length on G00
      VLVL   G00              Load vec. length on vec. len. reg.
①   ┌ VL     V000, IEE        Load J on V000
    └ VIL    V064, V000, IE   Load K on V064
      VEC    V064             Vector event sort and count
      VSEP   V128, B1         Vector store TRUE event elements
      VSEP   V192, B2         Vector store FALSE event elements
      VPT
      VWT
      SEC    EC0, I1          Store EC0 in I1
      SEC    EC1, I2          Store EC1 in I2
```

6.7.3 VEC, VSEP命令

VEC (VR型)

OP	EOP	R2	R1		

VSEP (VX型)

OP	EOP	R2	R1	X2	B2	D2

(1) VEC命令は

(イ) V064の下位32ビットの内容が1のときは，V000の対応する要素の下位32ビットの内容をV128の対応するベクトル要素へ移す。このときEC0カウンタの値を1だけ増加させ，またV192の対応するベクトル要素の下位32ビットの内容はゼロクリア

される。

V064の下位32ビットの内容が1のときは，V000の対応するベクトル要素の下位32ビットの内容をV192の対応するベクトル要素へ移す。このときEC1カウンタの値を1だけ増加させ，またV128の対応するベクトル要素の下位32ビットはゼロクリアされる。

(ロ) 直接番地指定のIEE(J)をロードするときは，

VLM　M000, 0, =F'1'

VGS　V000, 0, M000

とコンパイルされる。

(2) VSEP命令　(Vector Store Event Pipeline elements)

(イ) 初期設定後も値が変化しないECnに対応するVSEP命令の実行は抑止される。

(ロ) ECnの値が1だけ増分すると4バイト増分のBn番地へ格納される。このベクトル要素の下位32ビットが0である要素の格納は抑止される。Bn番地の4バイト増分も抑止される。

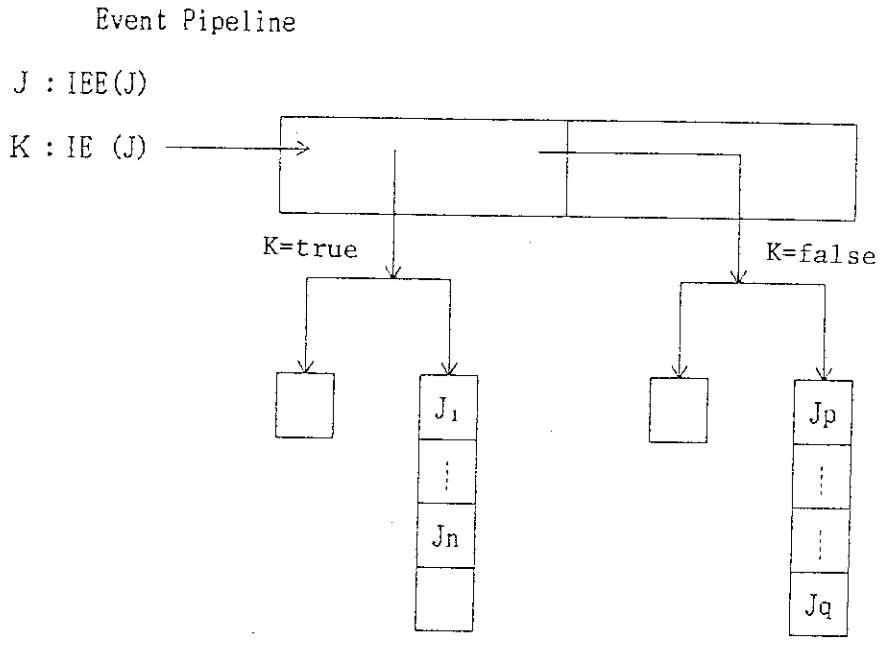
〔利用方法と問題点〕

利用方法：Fig. 6.16のDOループは，粒子JVが2方向円筒，直方体，球，X方向円筒，Y方向円筒，半球，半円筒の領域内に存在しているかどうかを判定し，領域内粒子と領域外粒子をバンクXSECBとCROSB2へ選別格納する操作を記述している。このDOループはComputed go to文を含んでいるのでベクトル化できない。

このループをAddress Preset Operation, Geometric Pipeline, Event Pipelineの機能を用いてベクトル化すれば次のページのようになる。

問題点：次ページのDOループのうちで小さなベクトル長を持つものがある。動的にベクトル長を見てベクトル演算，或いはスカラ演算を選択する命令列を生成するコンパイラもある。その機能を持たないコンパイラではベクトル化によってそのループの演算のみはスカラ演算よりも計算時間が長くなることがある。

```
      DO 10    IV=1, NPOS
      JV=POSBAN(IV)
      IGEO(JV)=IGEOM(K(JV))
 10   VGSORT(IGEO(JV), B1, I1, ···, B8, I8)
C**   NEXT (20)
      IF(I1.LE.0)GO TO 40
 20   DO 30 I=1, I1
      JV=B1(I)
      TFBANK(JV)=.FALSE.
      RSQ= ·····
      IF(Z1(JV).LE.·····) TFBANK(JV)=.TRUE.
 30   CONTINUE
C**   NEXT (50)
 40   IF(I2.LE.0) GO TO 70
 50   DO 60 I=1, I2
      JV=B2 (I)
      TFBANK(JV)=.FALSE.
      IF(X1(JV).LE.·····) TFBANK(JV)=.TRUE.
 60   CONTINUE
C**   NEXT (80)
        ⋮
      DO 200 I=1, NPOS
      JV=TFBANK(I)
      VESORT (TFBANK(JV), XSECBA, NXSEC, CROSB2, NCROS2)
200   CONTINUE
```



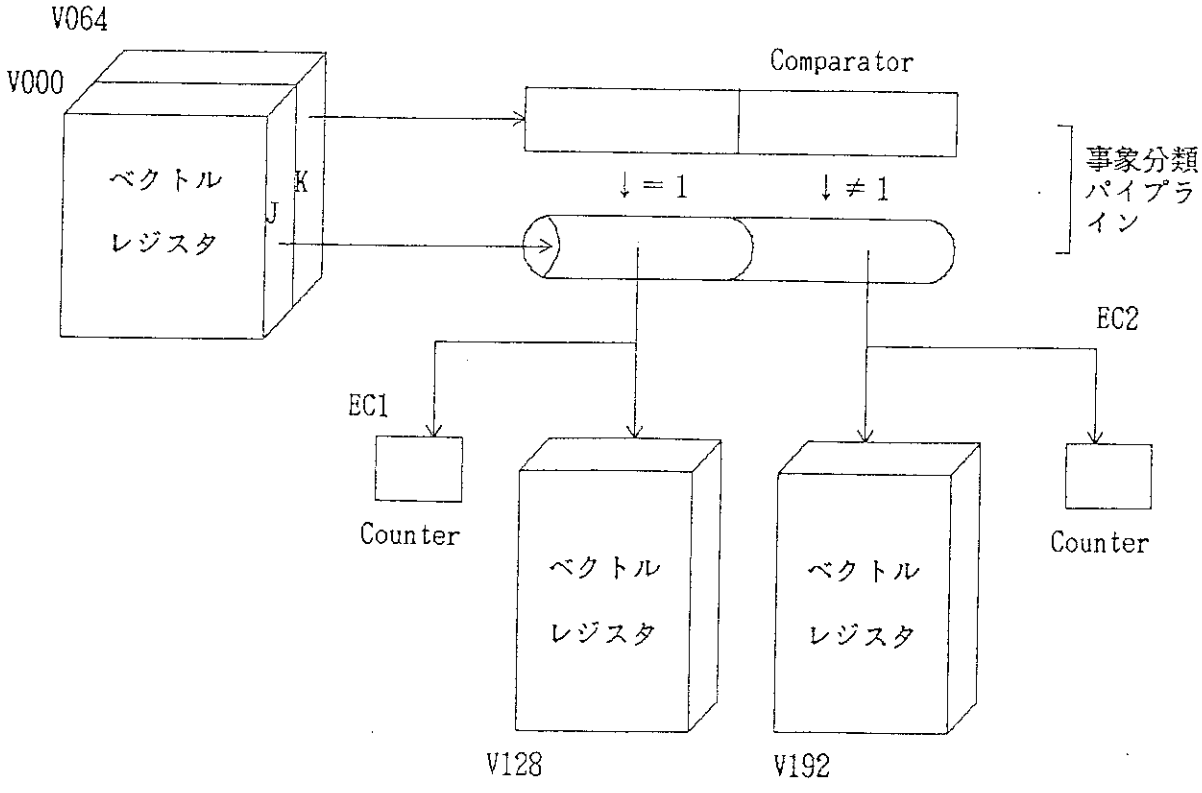


Fig. 6.19 Mechanism of event pipeline.

6.8 粒子領域検査パイプラインと関連命令

6.8.1 粒子領域検査パイプラインの目的

粒子モデルでは，粒子は通常は円球，円筒，直方体，円錐など基本形状（プリミティブ）を組み合わせた領域（ゾーン，或いはセル）が包含関係を成す複雑形状空間内を移動，又は滞留する。領域には予め番号が付けられており，ひとつの領域内にある粒子はその領域の番号を属性として持っている。粒子が空間を移動中に粒子の持つ領域番号と粒子がその存在している領域の番号とが不一致となることがある。この原因としては，入力データで定義されている形状の矛盾，粒子の飛跡線と領域境界との交差計算時の誤差の集積，当該コードのプログラム論理エラーなどが考えられる。不一致を起した粒子はlost粒子(particle)として計算の対象外とする。

Lost粒子を発見するためには，粒子が新しい領域に入る毎に無矛盾性のチェックを行う必要がある。これは，MCNPコードが採用している方法である。

MCNPコードでのセルの定義方法は，セルを取り囲む基本形状定義面を指定する。その際セルを取り囲む面のうちのひとつAの反対側にあるセル番号を同時に入力時に指定しておくこともできる。しかし，それは入力作成者に非常な労力を強いることになる。MCNPマニュアルでは，セルを定義する面のみを入力し，そのエラー・チェックのためには本番の計算前に粒子を飛行させてみることを勧めている。

図形同士の交差が計算誤差によって誤判定されること，例えばFig.6.20の(c)(文献17，PP.963)は十分あり得る。このときは和集合を正しく計算できなくなる。それを避けるためにFig.6.21（同，PP.963）のような様々なノウハウが使われている。

しかし，そのような逃げ手は限界があるとしてより正確な技法が文献16で提案されている。この種の技術伝達のスピードから考えると，一般的技法が確立されるのは相当長い年月を経た後になるであろう。MORSEとVIMは，MCNPとは異なり，基本形状の集合演算でゾーンを定義するが，入力されたゾーン定義データに和集合演算が入っている場合には，入力処理ルーチンが積集合の組合わせに変換する。KENO Ⅳ，MORSE-DD，VIMは粒子の座標位置を優先し，位置から領域を決定する方法を採用しており，粒子毎の無矛盾性のチェックを行っていない。

MCNPの方法は入力データの誤りの発見，コード開発時のプログラム論理の誤りの発見，計算誤差の集積とプログラム論理の不一致による計算時のプログラムの暴走の回避などに有用であると考えられる。粒子領域検査パイプラインは，無矛盾性チェックのアルゴリズムを高速で行うためのものである。

6.8.2 無矛盾性チェックの方法

MCNPコードにおける領域は，それを取り囲む表面の組み合わせをリスト表現して定義される。その例をFig.6.22に示す。リスト表現はFig.6.22に見られる入力データの集合から入力処理ルーチンによって生成される。リスト表現は例1，2及び3に見られる形式である。セルの方向，定義はFig.6.23のとおりとする。このときBoolean operatorの優先順位に従っ

てこのリストＬＧを評価し，その値をLGEVALとする。

LGEVALの値が１なら粒子は正しく定義された領域に存在し，LGEVALの値が０なら粒子は矛盾した領域内に存在する。LGEVAL＝０ならLost粒子である。

正しい領域か否かの判定は，粒子を取り囲む表面の正負の向きの意味が入力データと同じ意味であれば，入力データ中の表面番号を１で置き換え，異なっていれば０で置き換えてリストＬＧを作成する（文献18，PP.12−16）。

例２に対応する粒子のリスト（（０：１：１）：（１０）：（１１））０！を計算することはFig.6.24で定義される２進値treeを評価することと同じである。この計算を高速に行うためにpush down stackのハードウェアを必要とする。

6.8.3 VRD（Vector Region Determination）命令（ＶＲ型）

OP	EOP	R2	R1	

(イ) ＶＲＤパイプラインと呼んでいるが，プッシュ・ダウン・スタックである。その概念をFig.6.25に示す。

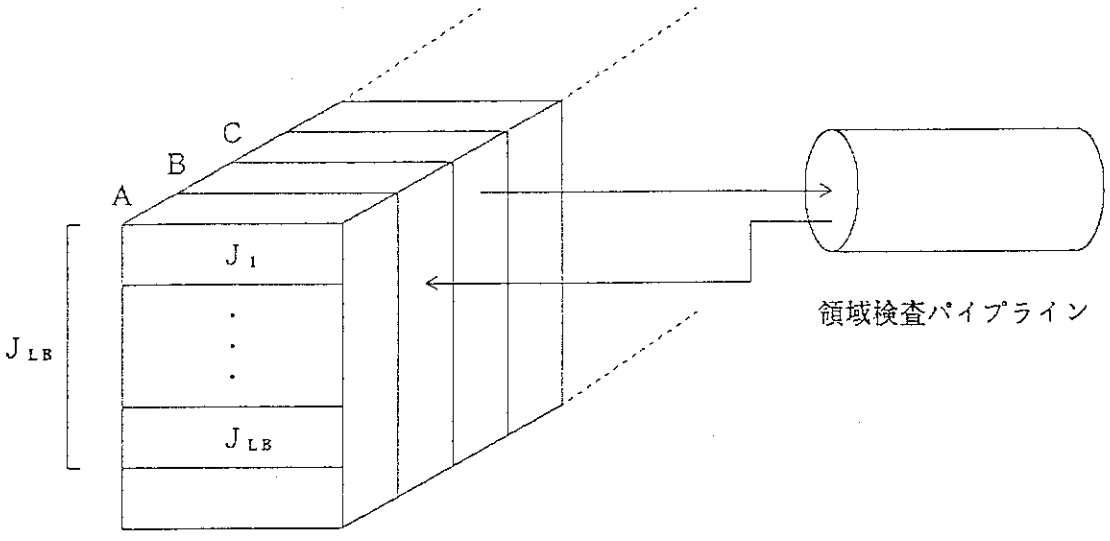
(ロ) 演算対象となる粒子数ｎと粒子群に対応するそれぞれが長さの異なるリストの２つを対象とするため，取り扱いの容易さを求めてｎ個の粒子リストを一次元的に連結する。これは利用者がプログラム上で行うものとする。

(ハ) プッシュ・ダウン・スタックの演算子は次の７種とする。

演算子＝
- ¬　（１０００１）；否定，数字はコード内の表現
- （　（１０００２）；左カッコ
- ）　（１０００３）；右カッコ
- ∧　（１０００４）；論理積
- ∨　（１０００５）；内包的論理和
- ！　（１０００６）；粒子リストの終端
- ？　（１０００７）；全粒子リストの終端

(ニ) 前述のＬＧをプッシュ・ダウン・スタックの動きに従って表現したものがFig.6.25である。当該粒子はLost粒子である。入力データの定義からすると領域はすべてFig.6.22の表面８の内側で定義されているにもかかわらず，！の直前の０は粒子が表面８の外側でも定義されているから。

(ホ) 粒子番号n_1を含むセルの構造判定用リストを（０：１：１）：（０：０），n_2を含むセルの構造判定用リストを０：１：１，・・・とする。計算対象となる空間の幾何形状が複雑であれば，１個の粒子当りのリストは長くなる。これに対し，例えばＶＰ−１００のベクトル・レジスタ３２KBを８分割して使用すると，Fig.6.25のレジスタ・エレメント集合Ａ，Ｂ，Ｃ，・・・それぞれの長さは３２KB÷８分割÷８バイト＝５１２である。粒子１個当りの構造判定用リストの平均長さを２５とする一回のベクトル・ロードで処理できる粒子個数は



例. (a)　Cレジスタ・エレメントから領域検査パイプラインへの入力（長さLL）

LIST：　(0∨1∨1)∨(0∨0)!0∨1∨1!・・・

・・・　((0∨1∨1)∨(1∧0)∨(1∧1))∧0!?

(b)　パイプラインからBレジスタ・エレメントへの出力（LB個）

1、1、・・・、0

(C)　J_1 =30、J_2 =13、・・・、J_{LB}=37のとき RBANKへの出力（LB個）は、

RBANK(30) ←1、RBANK(13) ←1、・・・、RBANK(37) ←0

512÷25=50となる。LISTを作成するときは粒子バンクBANK(I)(I=1, IBANK)をLL≦512となるように分割して使用する。これはソース・プログラムで利用者の責任において行うこととする。その結果粒子数が多ければ，例えば

LB(1)=47, LL(1)=497, LIST(1)=(0 1・・・!・・・!?
LB(2)=30, LL(2)=382, LIST(2)=((1 0・・・ !?

LB(LBMAX), LL(LBMAX)=504, LIST=(LBMAX)=・・・ !?

として利用者の責任において予め作成しておかねばならない。ベクトル・レジスタを4分割できれば1回のVRD命令で処理できるリスト長は2倍に拡大される。

(ヘ) ひとつのVRD命令で処理するリストはLIST(i)である。
ソース・プログラム上での表現は

```
K=1
DO 10 L=1, LBMAX
VRD (BANK(K), LB(L), LIST(L), LL(L), RBANK)
10 K=K+LB(L)
```

このVRD関数をアセンブリ語に展開すると次のようになる。

```
BC
L       G00, LL(L)            Set LL length
VLVL    G00                   to vec.len.reg.
VL      V064, LIST(L)         Load LIST on V064
VRD     V064, V032            Put V064 results on V032
L       G00, LB(L)            Set LB length
VLVL    G00                   to vec.len.reg.
VL      V000, BANK(K)         Load particle no.on V000
VIST    V032, V000, RBANK     Store results in RBANK
BC                            Repeat until LBMAX
```

注. 1) VRD文は，文中にベクトル長さを決定するLLとLBの2つ変数を含んでいるので，ひとつのDOループに展開することはできない。

2) VRDパイプラインに付属するプッシュ・スタックの作り方には，演算子とオペランドを1本のスタックに混在させる方法，これとは違いオペランドと演算子を別々のスタックに入れる方法がある。ここでは後者のほうが，次の理由から望ましい。

・単項演算子を取扱う必要がある。

・翻訳とは異なり，演算結果をスタックに残して置かねばならず，オペランドを別扱いとするほうが便利である。

3) ザブリストの終端記号！がパイプラインへ入力されたとき演算子スタックのポインタがゼロ位置でなければ，ベクトル・レジスタB上の対応する結果はゼロ値，即ちlost粒子とする。前リストの終端？についても同様，ただし，記号？の直前のサブリストのみlost粒子と見なす。両スタックのポインタはゼロにリセットする。

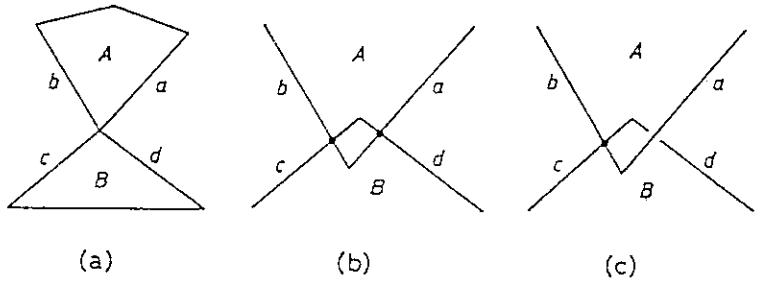


Fig. 6.20 Topological inconsistency caused by numerical errors.

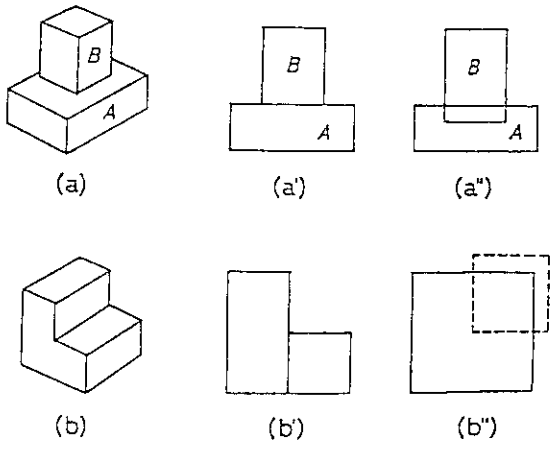


Fig. 6.21 Know-hows of avoiding topological inconsistency.

ルーチンLGEVAL（MCNPコードのルーチン）を理解するために

(1) 100000以上の表面番号

・ logical operator のとき、表面番号ＬＪＡは、100000以上の数。

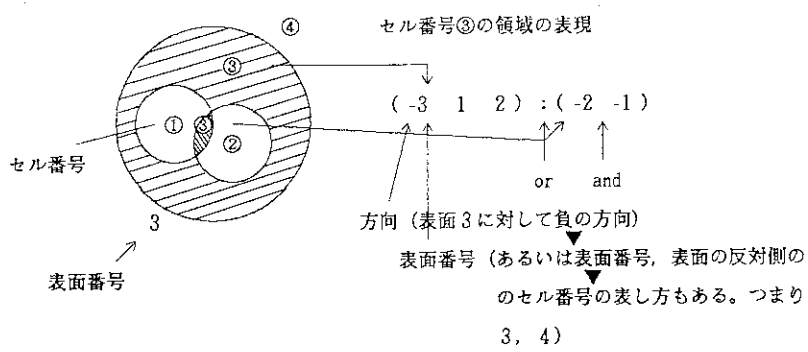
〃　　、ＬＪＡへのポインタＬＣＡは、負である。

ＬＪＡ（　）＝ 100000 ＋ｋｓ

1	(	Logical operatorの評価
2	)	優先順位も決めるための記号
3	:	or
4	#	compliment
	ブランク	and

(2) Logical operatorの使い方の例

・例1



・例2

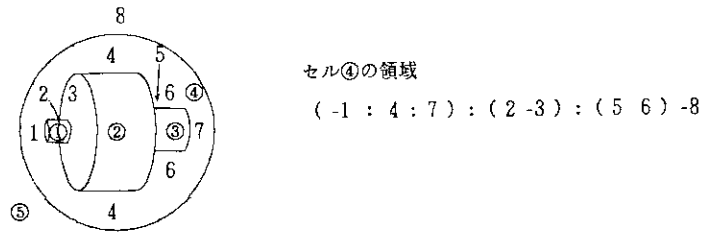
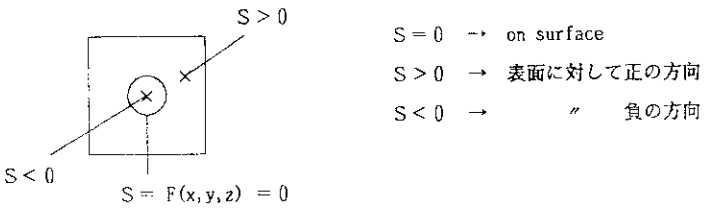


Fig. 6.22 Region definition by surfaces.

セル方向、セルの定義



・ セルの定義

	セル番号	物質番号	物質密度	セルをとり囲む表面番号
ex.	1	3	1.0 E -10	-1　91　-92

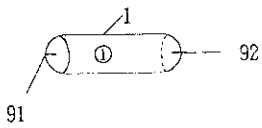


Fig. 6.23 Inside and outside of a surface.

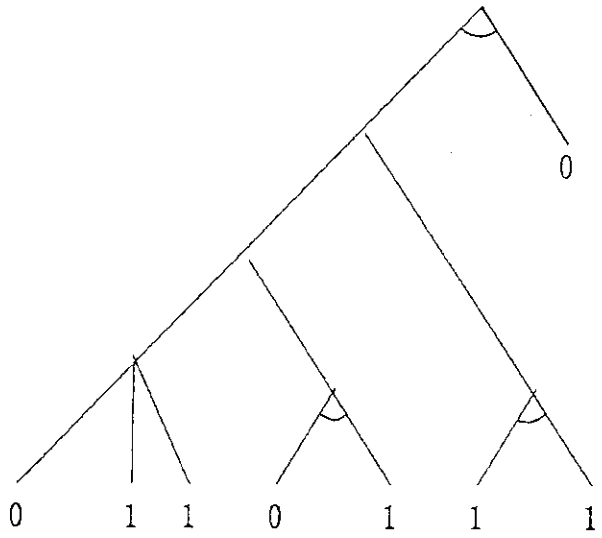


Fig. 6.24 Binary representation of logical expression.

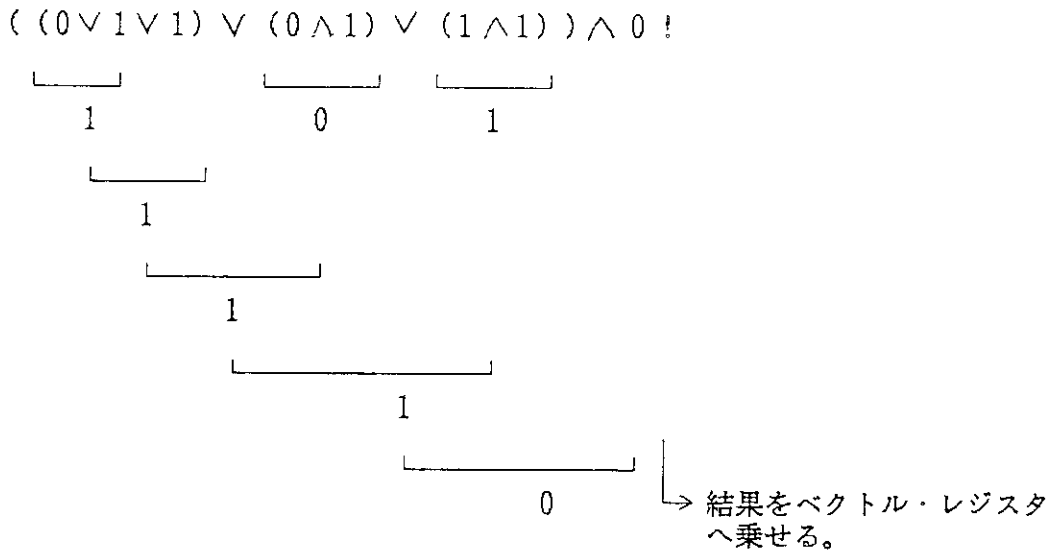


Fig. 6.25 Evaluation of logical expression.

6.9 おわりに

この章の初めに述べたように，調査はA，Bの2種のベクトル・プロセッサを使用して行い，これとは別に，ソフトウェア・シミュレータを作成して原研提案概念の有効性検討を行なった。それら方法の詳細は別に報告することとし，ここでは結果のみを示す。

(1) ベクトル・プロセッサによる調査の結果

ベクトル・プロセッサ及びハードウェア・シミュレータを使用し，KENO Ⅳコードを対象とした調査では次のような結果が得られた。

1) アドレス・プリセット演算器

現在のベクトル・プロセッサは基本的には，スカラ演算とベクトル演算との並列処理が可能である。従って計算の流れが遂次的な場合のアドレス・プリセット演算器の効果はそれほど大きくなく，計算時間の短縮効果は数パーセント以下である。

2) 幾何形状分類パイプライン

この分類パイプラインの存在によって，従来ベクトル化不可であったプログラム部分のベクトル化が可能になる利点は大きい。ベクトル・プロセッサAにおいては，この機能をアセンブラ（機械）語で記述して推定した。即ち，FORTRAN語で翻訳した場合に派生する命令をすべて省いたソフトウェア・ライブラリを作り，これを呼び出すことによって処理性能の向上度を測定した。その効果をFig.6.26に示す。また，このソフトウェア・ライブラリの機能をハードウェア化してもさほど大きな速度向上は得られない見込みである。従って問題は，ソース・プログラムを変更することなく，即ち利用者に負担をかけることなく，ソフトウェア・ライブラリと同じ効果を生み出すようにコンパイラを改造できるかどうかということになる。

3) 事象分類パイプライン

この分類パイプラインの機能は，従来のベクトル・プロセッサでは，マスク・レジスタの機能を使用することで実現されている。新らたに専用のパイプラインとして提案したのは2次元的パイプライン処理によって，従来型ベクトル・プロセッサの演算で必要となるマスク・レジスタのビット数カウント等の処理時間の短縮を図るためである。ベクトル・プロセッサAにおいて，幾何形状パイプラインの特殊な場合としてソフトウェア・ライブラリを使用して推定された。その効果をFig.6.26，6.27に示す。

4) 粒子領域検査パイプライン

このパイプラインは，本質的にはスカラ演算型であるとして，ベクトル・プロセッサBのスカラ演算数を基本として机上計算による推定では，専用演算器を仮定したときの処理速度の向上率は，対象となる計算時間についてのみならば6倍である。ただし，対象モンテカルロ・コードはMCNPである。

以上原研仕様に対する調査結果を簡単に述べた。項目2)～4)は，現状のベクトル・プロセッサのデータ処理方式は1次元処理であり，一定の時間幅でベクトル演算パイプラインに入ってきたデータ数に比例した数のデータが，パイプラインから出力されるようになっている。ところが上記項目2)～4)におけるパイプラインは，2次元処理であり，これを従

来のベクトル・プロセッサの設計方式・命令を組み合せて模擬する限りは大幅な処理性能の向上は望めない。とはいえFig. 6.26に見られる如く，当初目標の5～10倍の範囲の処理速度向上である。さらに，演算回路を多重化することによってこの2倍の処理速度向上の見通しが得られた。

(2) ソフトウェア・シミュレータによる評価

モンテカルロ・コードの計算処理を模擬するソフトウェア・シミュレータを作成し，KENO Ⅳ，MORSEなどのモンテカルロ・コードの計算処理を模擬した。模擬の方法は，(i)モンテカルロ・コードのソース・プログラム上の処理部分を細分割し，分割された各部分（これをブロックと呼ぶ）のスカラ及びベクトル命令数を列挙する。(ii)定められた粒子数・世代数について現実のモンテカルロ・コードによる処理時間の実測値とシミュレータによる模擬処理時間値を比較する。

以上のような方法で現実のモンテカルロ・コードを模擬した結果をTable 6.16に示す。このシミュレータによって，スカラ，ベクトル演算性能の向上が，コード全体の処理時間にどの程度の影響を及ぼすかをより精確に推定可能となった。また演算回路を多重化したときのコードの処理速度向上倍率の推定も可能となった。

Table 6.16 (a) Estimated calculation time of KENO IV code by our simulation program.

〔KENO IV のシミュレータによる評価〕

(1) 物理プロセスの決定

物理プロセスの決定はFortune/Top10 の解析結果から高コストルーチンを選択してその中のDOループで0.5%以上のコストを持つDOループを物理プロセスとして選択した。

ルーチン名	コスト(%)	物理プロセス数	備考
BEGIN	81.9%	37	0.5%以上のコストを持つDOループを選択
VCROS	24.3%	12	0.5%以上のコストを持つDOループを選択
ARBRAN	3.4%	1	Dummy Process.
合計	99.6%	50	

詳細は別冊のKENO IV の物理プロセス属性テーブル参照のこと。

(2) シミュレータでの性能評価

プログラムの実行時間					シミュレータによる予測CPU 時間
	APO	事象	幾何	領域	
★ベクトル版の実行時間 18.37 sec ★相対誤差 2.48% ★オリジナル版の実行時間 16.41 sec	ON	ON	ON		14.8863 sec(-3.9401 sec)
	ON	OFF	OFF		18.5957 sec(-0.2307 sec)
	OFF	ON	OFF		17.3588 sec(-1.4676 sec)
	OFF	OFF	ON		16.2202 sec(-2.6062 sec)
	OFF	OFF	OFF		18.8264 sec(0.0 sec)

シミュレータによる予測CPU 時間の括弧の中の値はモンテカルロ・パイプライン演算器を使用しない場合のシミュレーション時間からの差を示す。

Table 6.16 (b) Estimated calculation time of MORSE code by our simulation program.

〔MORSE のシミュレータによる評価〕

(1) 物理プロセスの決定

物理プロセスの決定はFortune/Top10 の解析結果から高コストルーチンを選択してその中のDOループを物理プロセスとして選択した。

ルーチン名	コスト(%)	物理プロセス数	備考
GG¥	28.3%	15	
G1¥	16.7%	20	
GG	13.7%		RELCO¥ 0450000 の中に含める
RELCO ¥	12.1%	2	
FLUXST	10.9%		RELCO¥ 0450000 の中に含める
EUCLI ¥	2.5%	7	
G1	2.4%		RELCO¥ 0450000 の中に含める
VINTER	2.3%		RELCO¥ 0450000 の中に含める
SDATA ¥	1.9%	4	
PTHETA	1.8%		RELCO¥ 0450000 の中に含める
RESTOR	1.1%	1	Dummy process.
		49	

(2) シミュレータでの性能予測

プログラムの実行時間	演算器の状態				シミュレータによる予測CPU 時間
	APO	事象	幾何	領域	
★ベクトル版の実行時間 12.61 sec	ON	ON	ON		8.2098 sec(-0.8337 sec)
	ON	OFF	OFF		9.0284 sec(-0.0191 sec)
★相対誤差 28.2 %	OFF	ON	OFF		8.5705 sec(-0.477 sec)
★オリジナル版の実行時間 12.98 sec	OFF	OFF	ON		8.7059 sec(-0.3416 sec)
	OFF	OFF	OFF		9.0475 sec(0.0 sec)

シミュレータによる予測CPU 時間の括弧の中の値はモンテカルロ・パイプライン演算器を使用しない場合のシミュレーション時間からの差を示す。

Table 6.16 (c-1) Estimated calculation time of MCNP code by our simulation program.

〔MCNP のシミュレータによる評価〕

(1) 物理プロセスの決定

物理プロセスの決定はFortune/Top10 の解析結果から高コストルーチンを選択してその中のDOループを物理プロセスとして選択した。

ルーチン名	コスト(%)	物理プロセス数	備考
TRACK1	29.7%	16	
CHKCL1	21.4%	12	
ACETT1	8.1%	12	
TALLYD	7.1%	38	
TALLY	6.2%	30	
LGEVAL	4.8%		注1
TALSHF	4.4%	2	
HSTORY	3.4%	39	
TRNSM1	1.9%	14	
NEWCL1	1.9%	6	
JBIN2	1.6%	6	
¥CECAS	1.5%	1	Dummy process.
¥¥STAR	1.4%	1	Dummy process.
COLIDI	0.8%	1	Dummy process.
CALCP1	0.7%	1	Dummy process.
	94.9%	179	

注1

・ LGEVAL はTRACK1及びCHKCL1のDOループ内部で引用されているため，引用しているDOループの手続きの一部として取り扱う。LGEVALを引用しているDOループ（プロセス）は領域検査パイプライン演算器対象DOループ（プロセス）とする。

詳細は別冊のMCNPの物理プロセス属性テーブル参照のこと。

Table 6.16 (c-2) Estimated calculation time of MCNP code by our simulation program.

(2) シミュレータでの性能評価

プログラムの実行時間	演算器の状態				シミュレータによる予測CPU 時間
	APO	事象	幾何	領域	
★ベクトル版の実行時間 6.67 sec	ON	ON	ON	ON	6.49406 sec(-3.5958 sec)
★相対誤差 51.3%	ON	OFF	OFF	OFF	10.0027 sec(-0.0871 sec)
	OFF	ON	OFF	OFF	9.45151 sec(-0.6383 sec)
★オリジナル版の実行時間 4.47 sec	OFF	OFF	ON	OFF	10.0443 sec(-0.0456 sec)
	OFF	OFF	OFF	ON	7.23941 sec(-2.8505 sec)
	OFF	OFF	OFF	OFF	10.0899 sec(0.0 sec)

シミュレータによる予測CPU 時間の括弧内の数値はモンテカルロ・パイプライン演算器を使用しない場合のシミュレータの予測時間からの差である。

Table 6.16 (d-1) Estimated calculation time of VIM code by our simulation program.

〔VIM のシミュレータによる評価〕

(1) 物理プロセスの決定

物理プロセスの決定はFortune/Top10 の解析結果から高コストルーチンを選択してその中のDOループを物理プロセスとして選択した。

ルーチン名	コスト(%)	物理プロセス数	備考
GGONE	34.1	6	
CROSS1	29.1	18	
ELAST1	6.9	21	
G1ONE	6.3	18	
FLUXX1	6.2	1	Dummy process.
REACT1	5.8	24	
GGSONE	4.9	1	Dummy process.
NUTRNG	2.3	9	
UNRES1	1.0	1	Dummy process.
¥¥SPEC	0.6	1	Dummy process.
¥¥INEL	0.6	1	Dummy process.
	98.0	101	

詳細はVIM の物理プロセス属性テーブル参照のこと。

Table 6.16 (d-2) Estimated calculation time of VIM code by our simulation program.

(2) シミュレータでの性能評価

プログラムの実行時間	演算器の状態				シミュレータによる予測CPU 時間
	APO	事象	幾何	領域	
★ベクトル版の実行時間 8.93 sec	ON	ON	ON		8.61875 sec(-0.2870 sec)
★相対誤差 7.9 %	ON	OFF	OFF		8.83871 sec(-0.0671 sec)
	OFF	ON	OFF		8.74146 sec(-0.1643 sec)
★オリジナル版の実行時間 sec	OFF	OFF	ON		8.85016 sec(-0.0556 sec)
	OFF	OFF	OFF		8.90579 sec(0.0 sec)

シミュレータによる予測CPU 時間の括弧内の数値はモンテカルロ・パイプライン演算器を使用しない場合のシミュレータの予測時間からの差である。

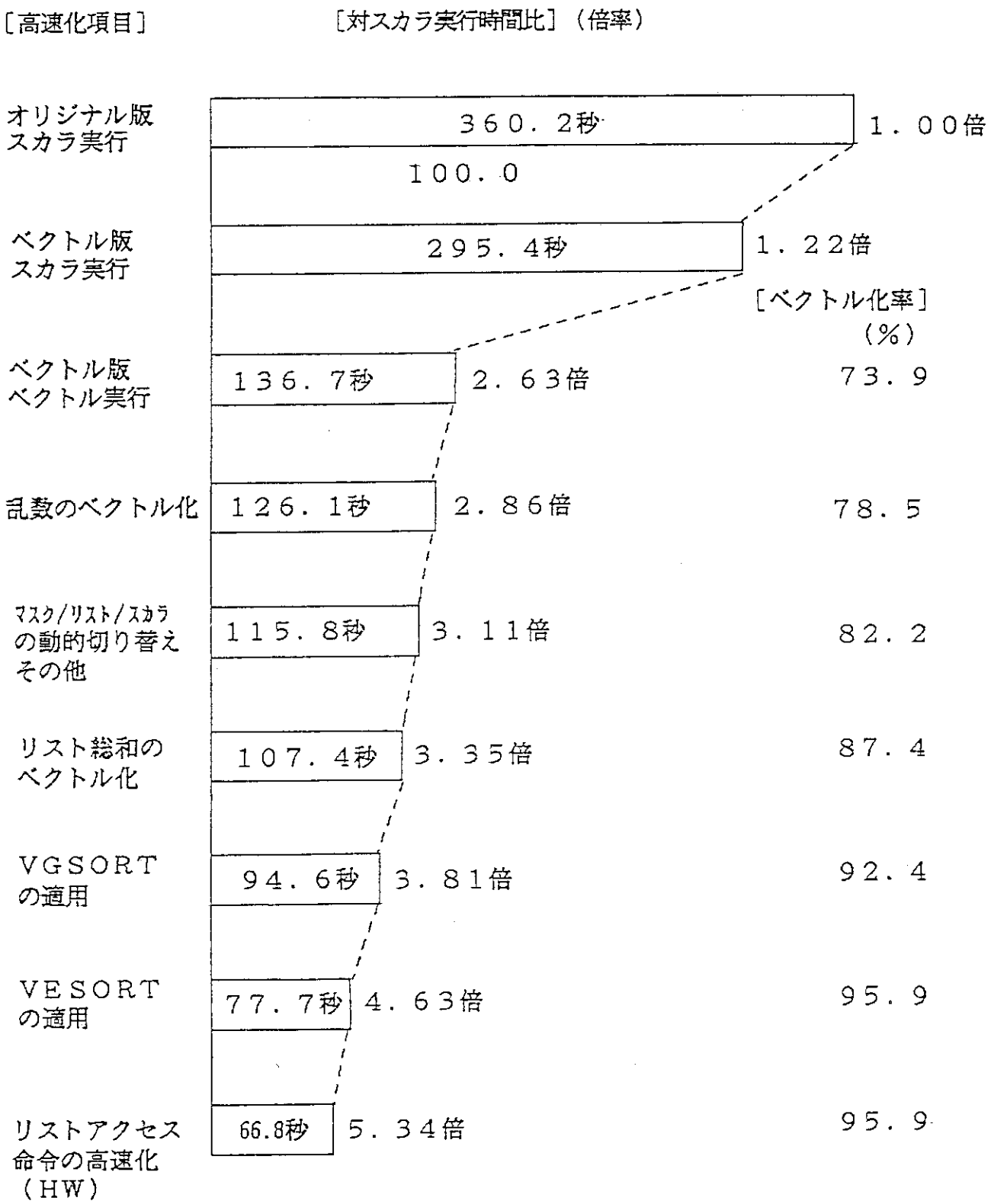


Fig. 6.26 Extimated speedup ratio of KENO IV code by our planned Monte Carlo computer (1200 particles in a batch).

参考文献

1) A.Gotieb et al.:The NYU Ultracomputer-Designing an M2MD shared Memory Parallel Computer, IEEE Trans.Comp., Vol.C.32P №2, 1983.

2) K.Asai et al.:Vectorization of the KENO Ⅳ Code, Nuc.Sci.& Eng., 92, PP.298-307, 1986.

3) L.M.Petrie and N.F.Cross:KENO Ⅳ—An Improved Monte Carlo Criticality Program, ORNL—4938, 1975.

4) M.B.Emmett:The MORSE Monte Carlo Radiation Transport Code System, ORNL 4972, 1975.

5) LASL Group TD—6:MCNP—A General Monte Carlo Code for Neutron and Photon Transport, LA 7396—M, LASL, July, 1978.

6) L.J.Milton:VIM User's Guide, Applied Physics Division, ANL, June, 1981.

7) F.B.Brown and W.R.Martin:Monte Carlo Methods for Radiation Transport Analysis on Vector Computers, Progress in Nuclear Energy, Vol.14, №13, PP.269-299, 1984.

8) W.R.Martin et al:Monte Carlo photon transport on a vecter supercomputer, IBM Jour.Res.Develop., Vol.30, №2, PP.193-202, March, 1986.

9) 浅井，樋口：原子力研究におけるモンテカルロ法とスーパーコンピュータ，第3回ベクトル計算機応用シンポジウム論文集，京都大学大型計算機センター，昭和62年3月。

10) 菅沼，他：連続エネルギー・モンテカルロ・コードVIMのベクトル化，JAERI—M 86-190，日本原子力研究所，1987年1月。

11) K.Asai et al:Vectorization of KENO Ⅳ Code and an Estimate of Vecter-Parallel Precessing, JAERI-M 86-151, 日本原子力研究所, Oct, 1986.

12) 樋口，他：MORSE-DDコードのベクトル化，JAERI-M 87-032, 日本原子力研究所，1987年2月。

13) 栗田，他：モンテカルロ・コードMCNPのベクトル化，JAERI-M 87-022, 日本原子力研究所，1987年2月。

14) FACOM M シリーズ・ハードウェア機能説明書Ⅰ（命令論），61HS-1011-1，富士通㈱，昭和51年10月。

15) FACOM VPシリーズ・ハードウェア機能説明書，79HS-1010-4，富士通㈱，昭和62年5月。

16) Los Alamos Radiation Transport Group X6:Reference for MCNP and the Manual, LA-7386-M, 1981.

17) 杉原，伊理：計算誤差による暴走の心配のないソリッドモデラの提案，情報処理学会論文誌，Vol.28, №9, PP.962-974, 1987.

18) J.S.Hendricks:Calculation of Volumes and Surface Areas in MCNP, LA8113-MS, LASL, 1980.

7. おわりに

「原子力知能化システム技術の研究」の名のもとに行っている人間動作シミュレーション技術の研究も2年経過したことになる。必要な機器，ソフトウェアがそろわず，実質は1年3カ月程度であるが，その間に初期調査は終了し，基盤技術としての本研究の進むべき方向もある程度は見えてきた。第1～6章の内容から推察されるとおり，本年度は研究に実質的な進展が見られた。

各章の分担は次のとおりである。

第1章　浅井

第2章　上中，神林

第3章　藤井

第4章　久米

第5章　樋口，藤崎，横川

第6章　浅井，樋口

また，次年度の作業は次のように予定している。

- 命令理解，知識ベース

炉内作業を表す基本動作列（サーブリック表現）を導出するために必要となる知識データを作成する（このような知識を解釈用知識と呼ぶ）。さらに，前年度整備したMicro tale-spinプログラムを拡張して原子炉内作業のシナリオを生成するプログラムを作成する。

(1) 基本動作生成用（解釈用）知識の概念検討とデータ化

(2) 炉内作業列生成プログラムの試作

- ロボットの画像認識

ロボットの視覚センサーから得られた画像をニューラル・ネットワークの技術のみで処理することは難しいと考えられ，既存の特徴抽出手法等と組み合わせた画像認識手法を検討するため，以下の作業を計画している。

(1) ＢＰモデル，ネオコグニトロン・モデルの画像認識能力の分析と改良法の検討

(2) 対象物体の切り出し，特徴抽出等に関する既存手法の調査，分析

- 二足歩行ロボットの運動学的シミュレーション

(1) 歩行パターン生成・逆運動学計算に関する調査とプログラム開発

現プログラムでは，脚の動きは各関節角の動きとして実際の歩行動作のデータを入力しているが，このような人間の歩行パターンだけでなく人工的に歩行パターンを作成することを考える。つまり歩行パターンとして足の軌道を与え，その軌道から逆運動学計算を行い各関節角の動きを計算する。そのための逆運動学計算プログラムを開発する。逆運動学を解くに当たっては，冗長自由度の考慮や拘束条件の検討，軌道の制御等に関する調査が必要である。また，Z.M.P.の移動方法を再検討し，足の軌道とZ.M.P.の移動に対する最適経路について考察する。

(2) 原研版二足歩行計算プログラムの発展形の開発

現プログラムは単脚支持状態についてだけを考えており周期的な歩行のみしか扱えない。そのため，本来停止や発進動作等のシミュレーションはできない。そこで両足支持状態を考慮に入れた運動方程式を取り上げ，拘束条件や境界条件等を変更した計算プログラムを開発する。

(3) ロボット運動学に関連した調査

下記の項目について大学に調査を依頼する。

a) 両腕協調動作・指と腕の協調動作を表すロボット運動方程式についての継続調査。

b) "曲がる"動作について，人間的な場合とロボット的な場合に対するそれぞれの運動学方程式及び方法論に関する調査。

- 施設形状データベース

(1) 入力作業

６４年度は，主に計器室内の機器を中心に入力を進める予定である。

(2) 技術の修得

導入した動画作成システムについては映像生成技術及び動画作成技術の修得を目指す。

(3) 知識ベースとのインターフェイスについての検討

模擬人間が行動する際の各動作における判断または詳細かつ具体的な動作列の生成については装置誘導型のものが検討されているが，この場合行動計画を行う知識ベースとのインターフェースについての考慮が必要となってくる。すなわち，装置誘導型の判断とは模擬人間が環境中の物体から知識を入手し動作列を生成するものであるが，この情報を施設形状データベースに追加するためのデータ構造の在り方，データの内容について検討する必要がある。

- 高速モンテカルロ計算装置

(1) 詳細設計

装置の詳細設計を行ないハードウェア・シミュレータで評価する。

(2) 実用計算コードによる評価

実用計算コードを対象とし，ソフトウェア・シミュレータによって装置の性能評価を行なう。

謝　　辞

本研究の目的，方法論について，京都大学名誉教授西原宏，東京大学教授吉川弘之，九州大学大型計算機センター研究開発部長松尾文顕の諸先生から貴重な御批判，コメントを戴いた。深く感謝します。

また，モンテカルロ計算装置の概念検討でお世話になった計算機メーカの方々，ソフトウェア・シミュレータ作成でお世話になった富士通株式会社の菅沼正之氏に深く感謝します。

(2) 原研版二足歩行計算プログラムの発展形の開発

現プログラムは単脚支持状態についてだけを考えており周期的な歩行のみしか扱えない。そのため，本来停止や発進動作等のシミュレーションはできない。そこで両足支持状態を考慮に入れた運動方程式を取り上げ，拘束条件や境界条件等を変更した計算プログラムを開発する。

(3) ロボット運動学に関連した調査

下記の項目について大学に調査を依頼する。

a) 両腕協調動作・指と腕の協調動作を表すロボット運動方程式についての継続調査。

b) "曲がる"動作について，人間的な場合とロボット的な場合に対するそれぞれの運動学方程式及び方法論に関する調査。

・ 施設形状データベース

(1) 入力作業

６４年度は，主に計器室内の機器を中心に入力を進める予定である。

(2) 技術の修得

導入した動画作成システムについては映像生成技術及び動画作成技術の修得を目指す。

(3) 知識ベースとのインターフェイスについての検討

模擬人間が行動する際の各動作における判断または詳細かつ具体的な動作列の生成については装置誘導型のものが検討されているが，この場合行動計画を行う知識ベースとのインターフェースについての考慮が必要となってくる。すなわち，装置誘導型の判断とは模擬人間が環境中の物体から知識を入手し動作列を生成するものであるが，この情報を施設形状データベースに追加するためのデータ構造の在り方，データの内容について検討する必要がある。

・ 高速モンテカルロ計算装置

(1) 詳細設計

装置の詳細設計を行ないハードウェア・シミュレータで評価する。

(2) 実用計算コードによる評価

実用計算コードを対象とし，ソフトウェア・シミュレータによって装置の性能評価を行なう。

謝　　辞

本研究の目的，方法論について，京都大学名誉教授西原宏，東京大学教授吉川弘之，九州大学大型計算機センター研究開発部長松尾文顕の諸先生から貴重な御批判，コメントを戴いた。深く感謝します。

また，モンテカルロ計算装置の概念検討でお世話になった計算機メーカの方々，ソフトウェア・シミュレータ作成でお世話になった富士通株式会社の菅沼正之氏に深く感謝します。