JAERI-Research
95-014

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)
—平成5年度作業報告—

1995年3月

秋元正幸・樋口健二・久米悦雄
神林　奨・大谷孝之・海老原健一

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

本レポートは，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，お申し越しください。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。

This report is 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　㈱原子力資料サービス

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）
－ 平成5年度作業報告 －

日本原子力研究所東海研究所情報システムセンター
秋元　正幸・樋口　健二・久米　悦雄・神林　　奨
大谷　孝之・海老原健一

（1995年2月1日受理）

日本原子力研究所は，1987年から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。これは，知能ロボット，知能化プラントの基盤技術を研究開発するものである。その内容は，命令理解システムの試作，視覚認識，2足歩行ロボットの動作シミュレーション，施設形状データベースの作成，被曝線量計算，さらに被曝線量計算等のモンテカルロ計算の高速化を目的としたモンテカルロ計算装置の設計・試作などである。

本報告は，平成5年度のHASPの作業内容について記述する。

東海研究所：〒319－11　茨城県那珂郡東海村白方字白根2－4

JAERI－Research 95－014

A Study on Intelligent Nuclear Systems (HASP : Human Acts Simulation Program) － Progress Report 1993 －

Masayuki AKIMOTO, Kenji HIGUCHI, Etsuo KUME
Shaw KAMBAYASHI, Takayuki OTANI and Ken-ichi EBIHARA

Computing and Information Systems Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received February 1, 1995)

The seventh year progress of the Human Acts Simulation Program, HASP in short, has been presented in this report. The HASP started in 1987 at JAERI as ten-year research and development program of underlying technologies for intelligent robots, intelligent nuclear plants and so on. It consists of the research and development of technologies of knowledge-base, robot vision, robot kinematics／kinetics, plant geometry data-base, dose evaluation and Monte Carlo machine.

Keywords: Artificial Intelligence, Natural Language Processing, Knowledge-base, Pattern Recognition, Robotics, Graphics, Monte Carlo Method, Supercomputer, Computer Simulation

目　　次

Contents

1. はじめに

昭和62年度に原子力委員会が改訂した「原子力開発利用長期計画」においては，原子力技術開発の共通基盤ともなり，かつ他の研究開発分野へも波及効果をもたらす可能性のある技術開発に取組むことが定められた。それらの分野として，原子力用材料，原子力用レーザー，原子力用人工知能及び放射線リスク評価・低減化技術が挙げられ，これらの4分野について，昭和63年度から関連機関によって研究を開始した。原子力用人工知能については，日本原子力研究所（以下，原研）が既に開始していた「人間動作シミュレーション技術の研究」に，「極限環境センシング・システムの開発」を加え，研究開発を進めてきた。尚，平成6年に改訂された新しい「原子力開発利用長期計画」においても人工知能関連の研究を継続実施することとしている。

本報告は,「人間動作シミュレーション技術の研究」（ＨＡＳＰ：Human Acts Simulation Program）に関する平成5年度作業報告である。

さて，平成5年度の成果報告を述べる前に，ＨＡＳＰの研究目的など研究概要，平成4年度までの研究概要，ついで平成5年度の成果概要，今後の計画を述べる。

1.1 ＨＡＳＰの研究概要

ＨＡＳＰの概念は，Fig. 1.1に示すように，自然文で書かれた命令を，コンピュータのなかで作り出された模擬人間，即ち，ソフト知能ロボットに与え，知能ロボットは命令を読み，意味を理解し，自己の行動を計画し，動作列を生成する（命令理解，静的経路探索，ロボット作業計画)。次いで，環境の状況を認識・判断しつつ動作列を精密化して目的の位置に，ロボット動力学／運動学に基づいて移動し（動的経路探索，視覚認識，ロボット動力学），点検作業の対象となる装置や機器が有する情報を収集しながら（センサー融合），それが正常かどうか判断するなど点検作業を遂行する（ロボットによる異常診断，ロボットによる修復作業)。この間ロボットが受ける放射線被曝線量計算を含め，これらすべての過程は論理・数値モデルによってコンピュータ・シミュレーションされる。

以上述べたように，ＨＡＳＰの研究目的は，人間理解や人間そのものの動作シミュレーションではない。人的作業の定性的・定量的評価，人的作業の機械化，複雑人工構造物と人間の役割分担の明確化である。具体的には，高放射線下で動作する知能ロボット設計のための要素技術を開発すること，ロボット知能化の程度との関連において，プラントの知能化・自動化技術を開発すること，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がＨＡＳＰの目的とするところである。

1.2 平成４年度までの経緯

本研究は，二つの研究小テーマに分けて研究を実施してきた。即ち，人間型知能ロボットに要求される知識ベース・システムの研究及びロボット動作等に関する高速シミュレーション技術の研究である。それぞれの研究テーマ別に研究経過を述べる。

1.2.1 原子力知識ベースの研究

本研究テーマでは，日本語で与えられた命令文を解釈し，環境データを用いてロボット動作列を生成する命令理解・行動計画の研究，原子力施設に関する環境データを管理し，シミュレーション結果の可視化を行う環境設定・映像化技術の開発，カメラ画像等のセンサ情報を処理し，環境を理解するための環境認識技術の研究から成る。

本研究は，昭和６２年度から１０年計画で開始しが，昭和６２年度に原子力委員会が改訂した「原子力開発利用長期計画」の原子力基盤技術開発研究「原子力用人工知能」の研究として位置付けられ，当初計画を変更せず昭和６３年から当時の新長期計画の下に研究開発を進めてきた。このうち，本小テーマに対応する研究項目は，命令理解・行動計画の研究として，日本語で与えられた命令文を解釈する命令理解システムの試作，環境設定・映像化技術の研究として，ＪＲＲ－３施設一部の施設形状データベースの試作，環境認識の研究として，ロボットの自己位置及び物体認識技術の研究開発である。

年度毎の研究経過については，昭和６３年度までは，上記計画で研究を進めてきたが，平成元年度に本研究が，個別研究課題「人間動作シミュレーション技術の研究」と科技庁クロスオーバ研究「原子力用人工知能を具備した原子力施設のシステム評価研究」に研究予算とともに二分されたため，研究計画を見直した。平成元年度以降の全体計画は，Table 1.1 に示すとおりである。本小テーマの研究各項目の研究経過，成果とその意義などは，以下のとおりである。

i. 命令理解・行動計画の研究

昭和６３年度

- ＪＲＲ－３炉内点検作業を対象としたロボット動作生成過程の検討
- 動作知識ベースの開発
- 構文解析プログラムＣＳ－ＰＡＲＳＥＲの導入

平成元年度

- 命令文解釈等を行う命令理解システムのプロトタイプの作成

平成２年度

- 問題解決における定性推論の利用手法の検討
- 命令理解システムの体系化及び実用化

平成３年度

- 試作した命令理解システムの性能評価
- 知識表現手法の確立をめざした動詞概念知識構造の作成
- 統語名詞概念知識ベースの論理設計
- ＡＴＭＳ（Assumed Truth Maintenance System）手法による経路探索の効率化や時間の概念を加えた定性推論手法
- 工具の形状と機能を対応付ける機能推論手法の検討

平成４年度

- 共同研究において相手方開発のロボット・ハードへ搭載，改良及び拡張

ii. 環境設定・映像化技術の研究

昭和６３年度

- ＪＲＲ－３一次冷却系配管及び原子炉建屋の施設形状データベースの作成
- 画像生成用並列計算機の導入

平成元年度

- ＪＲＲ－３の主要機器等のデータベース化
- 並列計算機による高速画像生成法の検討

平成２年度

- ＪＲＲ－３の主要機器等のデータベース化及び３次元映像化
- データ量の増加による画像生成速度を向上させるためのデータ構造の階層化

平成３年度

- 他システムとの統合のための機能拡張
- 機器データベースの作成

平成４年度

- 共同研究用ＪＲＲ－３模擬環境の設計・製作
- 模擬環境に関する施設形状データベースの整備

iii. 環境認識技術の研究

昭和６３年度

- 神経回路網理論（ニューロ手法）を用いた物体認識技術の検討
- 誤差逆伝搬学習則モデルを用いた手書き数字の認識実験

- ネオコグニトロン型モデルを用いた簡易画像の認識実験

平成元年度

- 既知環境下におけるマークを利用したロボット・ナビゲーション法の検討
- 両眼立体視による３次元空間認識のためのニューロ手法適用可能性の検討

平成２年度

- ロボットの視覚システム等の調査
- 工具等の特定対象物をシーン中から検出する方法の検討
- 工具画像データベースの開発

平成３年度

- 両眼立体視問題における２段階動的計画法モデルの開発

平成４年度

- ロボットの自己位置の高速認識ソフトの開発
- ２段階動的計画法ソフトの実画像を用いた性能評価
- Simulated Annealing 法による複数画像情報からの物体配置算出手法（物理空間法）の開発及び性能評価
- 巡回点検ロボットに搭載する知的センシング・システムの検討
- ロボット・ハードに搭載するセンシング・システムの詳細設計

iv. 研究成果とその意義

命令理解・行動計画の研究では，日本語による作業指示文を解釈し，ロボットの動作列に高速展開する命令理解システムを開発し，実用性評価を行った。環境設定・映像化技術の研究では，ＪＲＲ－３及び模擬環境に関する施設形状データベースを作成し，ソリッド・モデルを基本とするプラント・モデル化手法を確立した。ソリッド・モデルで記述された環境モデルをレイ・トレーシング法により可視化する実画像近似機能，ポリゴン・データを用いて実時間表示する映像化システムを開発した。環境認識技術の研究では，白黒マークを利用した自己位置認識用の高速処理ソフトを開発した。また，両眼立体視において，従来手法では不可能とされていた順序逆転問題に対して，２段階動的計画法を開発し，その有効性を提案した。さらに，複数視点から撮った画像を使用して，実空間の物体配置を焼きなまし法を適用した直接算出する物理空間法を提案した。

成果の意義として，命令理解・行動計画の研究については，日本語（自然言語）によってロボットに与えられた作業命令を解釈し動作列を生成する命令理解システムを開発し，実用化の見通しを得た。従来は，マン・ロボット・インターフェイスに特殊なロボット言語が用いられ，本システムのような自然言語によるマン・ロボット・インターフェ

イスの実現性を示した例はない。環境モデル開発における，ソリッド・モデルによる３次元物体の記述法は，従来のプラント・モデル化に用いられていたＣＡＤ的手法，即ち３次元物体を頂点及び稜線で記述する手法と異なり，物体を基本立体の集合演算によって精密に記述する手法である。移動ロボットに要求される視覚認識用の精密モデル開発等に利用可能であり，今後の活用が期待される。白黒マークを利用した自己位置認識ソフトは，市販の小型カメラを使用した簡易な方法によって，処理時間・精度の点から実用性をを示した。両眼立体視において開発した２段階動的計画法は，両眼立体視の実用化において避けられない近接小物体の検出を可能するもので，今後の詳細化が期待される。

成果の特徴としては，開発した命令理解システムについて，割り込み命令等の動的に変化する入力を受理可能である。即ち，今後の動的環境変化の認識結果を受理可能なシステム構造を実現している。視覚認識については，床置き白黒マークを利用したロボットの自己位置認識用の高速処理ソフトは実用性を追求した成果である。

これらの成果の波及については，命令理解の研究で開発した命令理解システムが，辞書の拡張・変更によって，適用範囲の拡大が可能であるため，各種ロボットに適用可能な汎用システムとなり得る。また，環境モデル開発において用いた３次元物体の記述法は，その精度が高いことから，移動ロボットと機器との精密な干渉チェック等の処理に利点を持っている。さらに，白黒マークを利用したロボットの自己位置認識手法は，その高速処理性能及び簡易な方法であることから，一般の移動ロボットにも容易に適用可能である。

1.2.2　高速シミュレーション技術の研究

本研究小テーマの目的は，主として数値シミュレーションの高速化に関する研究開発を行うことである。具体的には，コンピュータ上に設定された模擬人間，即ちソフト・モデルとしての知能ロボットの動力学シミュレーション，この模擬人間の被曝線量高速計算手法の研究及び高速モンテカルロ装置の開発である。

本研究も，原子力知識ベースの研究と同様に，昭和６２年度に１０年計画で開始した。本研究小テーマに対応する研究項目は，Table 1.1 に示す研究項目 4)，5)，6) 及び 7) である。この内，5) のロボット・ハードウェアの試作については，予算化が実現しなかった。各研究項目の年度経過，研究成果とその意義などについて以下に述べる。

i. ロボット動作の研究

昭和６３年度

- 二足歩行運動学シミュレーション手法に関する委託調査「人間動作のロボット運動学方程式化に関する調査（手の動き）」

平成元年度

- 直進に関する静歩行シミュレーション・システムの開発

- クロスオーバ研究開始に伴い，プラント知能化に関する委託調査「自律型プラント設計情報の構造化手法に関する調査」
- 感覚と動きに関して委託調査「視覚認識及び両手作業の運動学方程式化に関する調査」
- 歩行に関する委託調査「二足歩行の運動方程式化に関する調査（足の動き）」

平成２年度

- 直進に関する動歩行シミュレーション・システムの開発
- ロボット設計上の問題点把握のため委託調査「設計二足歩行ロボットの設計に関する調査」
- 継続して，プラント知能化に関する委託調査「自律型プラント設計情報の構造化手法に関する調査(II)」

平成３年度

- 各種補償動作に対する動歩行シミュレーション
- 高度な二足動歩行シミュレーション実現のため委託調査「二足歩行方向転換に関する技術調査」

平成４年度

- 発進・停止歩行シミュレーションの解析
- 継続して委託調査「二足歩行ロボットの外乱のある環境下における歩行モデルの調査」

ii. ロボット・ハードウェアの試作

平成４年度

- 視覚，嗅覚，聴覚及び放射線センサー搭載のセンシング装置の設計

iii. 被曝線量計算の研究

昭和６３年度

- 人体に対する実効線量当量算出手法の調査

平成元年度

- MCNP コード入力体系可視化機能の開発

平成２年度

- 人体模型を用いた MCNP コードによる実効線量当量算出システムの開発

平成３年度

- 上記システムによる人体模型を用いたMCNPコードによる実効線量当量算出

平成４年度

- 上記システムの高速モンテカルロ計算装置上への移行

iv. 高速モンテカルロ計算装置の開発

昭和６３年度

- 粒子輸送モンテカルロ・コードの高速処理に必要なハードウェア及びアーキテクチャの概念検討

平成元年度

- 概念検討に基づいた詳細設計

平成２年度

- 詳細設計に基づき装置製作に着手

平成３年度

- 装置製作
- 粒子輸送モンテカルロ・コードの整備

平成４年度

- 装置製作完了（納期の平成５年４月末に先行して８月に搬入）
- 実コードを用いた性能評価

v. 研究成果とその意義

高速モンテカルロ装置の開発では，独自のモンテカルロ・パイプラインと軽装化した従来パイプラインを内蔵した演算装置を４台を搭載し並列処理も可能な高速モンテカルロ装置を開発した。開発した装置の実効性能を評価するため，既存の粒子輸送モンテカルロ・コード（KENO-IV）を用いたベクトル・並列処理を行い，目標性能である１０倍の速度向上率を確認した。二足歩行運動学における動力学モデルの開発では，直進，停止等の安定歩行，安定性に対するロボット・リンクの質量分布の影響，リンクの強度等多様な特性解析が可能な二足動歩行シミュレーション・システムを開発した。

成果の意義としては，従来のスーパーコンピュータで既存の粒子輸送モンテカルロ・コードに対する速度向上率は，１．５倍程度であった。これに対し，高速モンテカルロ装置上においては，既存コード（KENO-IV）に対して，１０倍の速度向上率を達成した意義は大きい。また，二足動歩行の多様な特性解析が可能な二足歩行動力学シミュレーション・システムは，二足歩行ロボットの設計に活用可能である。

成果の特徴のとしては，高速モンテカルロ装置に関し，粒子輸送モンテカルロ計算では２次曲面の集合演算によって記述された３次元領域において粒子の生成・消滅等を追跡する処理に特徴があり，これに伴う演算が計算時間の大部分を占める。しかも，これらの処理は，従来のスーパーコンピュータのアーキテクチャでは高速化不可能でな演算によって構成されることに着目し，幾何形状パイプライン，事象分類パイプライン及び領域判定パイプラインと命名したモンテカルロ・パイプラインを独自に開発した。また，二足歩行の動力学シミュレーションに関しては，動力学的に不安定な二足歩行の安定歩容計算には，歩容を既知として与え，安定化の補償動作求める，補償動作を与えて安定歩容を逆運動学的に求める，あるいはこの混合解析など，多様な特性解析が可能である。

研究成果が及ぼす波及効果としては，まず従来のスーパーコンピュータでは極めて困難であった粒子輸送モンテカルロ・コードの高速処理を可能にした高速モンテカルロ装置は，多大の計算時間を要する粒子シミュレーションを可能にし，分子・原子レベルの挙動等に関する新たな物理現象の解明に資することが期待される。また，二足歩行に関しては，ロボットの移動機構としての二足歩行の研究は，動力学的に不安定な運動を扱う未知の分野であり，実用化されたハードウェアは皆無である。開発したソフト活用することにより，実用二足歩行ロボットに要求されるアクチュエータ性能，リンク強度等の今後開発すべき技術課題を提示することが可能である。

1.3 平成５年度以降の研究計画

平成５年度に，昭和 63 年度から平成４年度までの５年間の研究事後評価及び平成５年度から平成９年度までの研究事前評価が行われ研究計画が見直された。新研究計画を Table 1.2 に示す。尚，研究の一部は，平成元年度から平成５年度までの５年計画で，動燃，理研，電総研，船研及び原研によって実施している原子力基盤技術開発研究クロスオーバ研究「原子力用人工知能を具備した原子力施設のシステム評価研究」（第Ｉ期）の原研担当テーマ「シミュレーション・システムによる知能ロボットの自律型プラントへの適用性評価」として実施している。また，平成６年度から５年間は，同様の機関の参加によって同クロスオーバ研究第II期「自律型プラントの分散協調知能化システムの開発」の原研担当テーマ「知能ロボットの点検用知識ベース構築技術の研究」が研究の一部を成している。次に，研究小テーマの内容を示す。

1.3.1 原子力知識ベースの研究

環境異常の認識，環境情報の学習及び巡回・点検等の手順を経験によって習得する知能ロボットの実現を目指し，本研究では，(1) ロボットの行動・状況判断等の高度化のため，プラント機器とロボット双方へ知識を分散付与する分散型知識ベース構築技術の研究及び (2) ロボットの収集する視覚・聴覚・嗅覚融合センサー情報とロボットに限定付与された知識に基づく環境認識技術及び環境情報学習技術の研究を進めるとともに，(3) 開発した要素技術を統

合し，実環境に即したモデル環境下で，統合技術の適用性を評価する。

従来の知能ロボット設計論では，ロボットにその動作する空間に関するすべての知識を持たせることが普通である。しかし，この方法は小さく閉じた積木の世界では可能であるが，現実世界では，(1) 知能ロボットへの付与知識であるルール数の増大による探索時の組合せ爆発，(2) 知識が与えられていない空間では動作は不可能，(3) 空間とロボットの相関から生じる総ての状況を事前に知識化してロボットへ付与することが困難，(4) 空間形状，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の困難等多くの未解決の課題がある。このような課題を解決し，現実世界で行動可能な知能ロボットを実現するため，ロボットへ付与すべき知識ベース構築法，視覚・聴覚・嗅覚センサー情報に基づく環境認識・環境情報学習に関する手法を研究する必要がある。

i. 命令理解・行動計画の研究

第II期クロスオーバ研究の中核を成す本研究では，点検作業，機器・環境及び環境変化等に関するあらゆる知識をロボットに付与することが不可能であることから，プラント機器及びロボットの知能分担を明らかにし，プラント機器とロボット双方に知識を分散付与する分散型知識ベースを試作し，日本語で与えられた作業指示書に従って移動を開始し，ロボットとプラント機器との交信によって点検作業等の行動に関する知識を随時習得（学習）しながら，ロボットが行動する命令理解システムを作成する。

ii. 環境設定・映像化技術の研究

ロボットの学習機能，即ち分散型知識ベースのプラント機器知識及び視覚センサー情報の再構成から環境情報を学習するための環境モデル生成法を開発し，施設形状データベースに統合する。また，諸機能拡張に応じて，映像生成の高速化を進める。

iii. 環境認識技術の研究

視覚・聴覚・嗅覚センサーから構成されるセンシング・システムの開発・整備，本センシング・システムを搭載可能な移動台車の別個整備及びロボット動作試験用模擬環境において実環境データの収集，上記センシング・システムからの単眼視（色情報）及びレーザー投光データ（距離情報）から環境中の機器を同定し，3角測量の原理によってロボット航行時のロボット自己位置を算出するソフトの開発，上記の単眼視（色情報）及びレーザー投光データ（距離情報）から機器・環境の3次元情報の再構築，聴覚データから異常音の発生位置及び周波数スペクトル分析による異常音同定，嗅覚データによる異臭同定等センサー情報融合処理等の環境認識ソフトの開発を進める。最終的には，上記の命令理解システム，施設形状データベース及び環境認識ソフトを統合し，技術の適用性を評価する。

1.3.2 高速シミュレーション技術の研究

開発済みの二足歩行動力学シミュレーション・システムを活用し，二足歩行ロボットの視覚的な経路状況から，歩容及びトルクを生成するなどフィードフォワード・感覚運動学習等

を取込んだ二足歩行ロボットの機構・制御設計手法を開発し，これを実現するために必要な技術開発を提案する。また，これらの研究を加速するためプロトタイプの二足歩行ロボットを設計・試作する。また，高放射線下で動作するロボット躯体に対する放射線損傷の評価システムを開発し，ロボット動作とロボット躯体の被曝線量評価計算をモンテカルロ装置上で高速処理する技術を開発する。

現在も将来に渉っても，人間の介在を前提としない原子力施設は有り得ないし，もし人間作業を代替できるような人間型ロボットが容易に実現されれば受け入れられることは疑う余地がないと言える。地上の生物をみてもその多くは脚式の移動で自然環境に適合しており，車輪を持つ地上の生物は皆無である。このように環境適合性に優れた人間型ロボットの実現を困難にしているのは，転倒することなく安定性を保持しつつ，多くの自由度に適切な制約条件を課して，人間のように自在で俊敏な歩行を実現するための制御法，軽量で十分な強度を持った材料，トルク・重量比のよいアクチュエータ等に多くの未解決な技術的課題があるからである。従って，このような技術的課題を順次解決し，人間，自然環境，原子力施設等の人工構造物との共生の観点から，人間型ロボットを実現する必要がある。また，原子力環境での行動を想定するため，ロボットの放射線損傷評価も当然必要である。

i. ロボット動作の研究

停止・発進・定常歩行の完全歩行等の特性解析のための開発済二足歩行動力学シミュレーション・システムに，方向転換，外乱補償，不整地歩行等の機能を追加し，システムを高度化する。また，人間の歩行制御等に習った感覚運動学習のフィードフォワード・制御モデル等を開発し，これに基づく歩行制御法の研究を行う。また，高度な二足歩行ロボット実現に必要な技術開発を提言する。最終的には，視覚情報から動作パターンを自動生成する二足歩行の機構・制御法の確立を目指す。

ii. ロボット・ハードウェアの研究

センシング・システムを搭載して環境情報を収集する車輪走行の移動台車を整備し，視覚センサー情報から走行を制御する方法を検討する。また，上記研究を加速するため，二足歩行ロボット・ハードウェアのプロトタイプを設計・試作する。

iii. 被曝線量計算の研究

放射線環境で作業する二足歩行ロボットの系統的・詳細な被曝線量評価を行い，ロボットに要求される耐放射線性の簡便な設計手法を開発する。このため，次の手法を検討する。即ち，歩行中の姿勢変化の影響を評価するため，動的ファントムを用いた時間依存モンテカルロ・シミュレーションの計算手法，多様な線源を持つ実施設の放射線場の設定手法及びＬＳＩ回路等の電子機器の吸収線量の計算法を検討し，高速モンテカルロ装置上で高速処理する放射線損傷評価システムを開発する。

iv. 高速モンテカルロ装置

関連計算コードによる装置の性能評価を行う。

1.4 平成５年度の研究概要

詳細は，第２章以降に述べるとおりであるが，ここではその概要を述べる。

1.4.1 命令理解システム

(1) 従来，簡単なシミュレータによってロボットの運動等を模擬していた部分について，視覚認識，ロボット運動学シミュレーション，映像生成等 HASP において研究開発を行っているソフトウェアと命令理解プロトタイプ・システムとを実際に結合するためのネットワークインタフェースの整備を行うとともに，(2) ２次元メッシュによって記述された移動空間地図を利用した，滑らかな移動経路を導出する経路探索アルゴリズムの開発を行った。これにより，命令理解システムを中核とした統合的なシミュレーションが可能となった。

1.4.2 視覚認識の研究

昨年度概念設計を行った原子力施設巡回点検用センシングシステムの詳細設計及び製作を行い，さらに，このセンシングシステムの制御及び状態表示を行うためのソフトウェアを整備した。

また，環境認識技術の研究の一環として，環境内の機器の特徴点を抽出することによって，移動ロボットの位置を同定するシステムを開発することを目的とし，CCD カメラによって撮影されたカラー画像から，必要とされる領域の線分を抽出するプログラムの作成をおこなった。

1.4.3 ２足歩行シミュレーション

現在の２足歩行ロボットの動作制御は，決められた発進位置から停止位置までの安定歩行パターン等を数値シミュレーションによって予め計算し，そのパターンに追従するように関節トルク等が制御される。従って，歩行中の予期しない外乱等に動的に対応することができない点が大きな問題として残っている。

本年度は，この問題を解決する第１ステップとして昨年度検討したリアルタイム２足歩行制御法を未知の外乱がロボットに作用する環境下での歩行に適用してシミュレーションを行い，この制御法の有効性を確認した。また，この制御法は，歩行中歩容が変化するような場合や方向転換動作にも適応が可能である。

1.4.4 施設形状データベース

施設形状データベースは，ＨＡＳＰにおける各種シミュレーションに必要な環境モデルを生成する役割を持つ。これらの環境モデルの性質はモデルの利用法に依存するが，環境認識においては，ＣＣＤカメラからエッジ情報を抽出し，環境モデルに対応する数学モデルと比較する手法が一般的である。このため，環境認識に必要な２種類の輪郭線モデルを開発した。

３次元空間において視点と視方向を与えたときのＣＳＧ（Constructive Solid Geometry）モデルに対する輪郭線モデルの生成は，複数の基本立体による集合演算を行った際の合成部分の取り扱いが難しいことから，画像処理における問題点の一つとなっている。この問題点

に対するアプローチとして，ＣＳＧモデルではなく，ＣＳＧモデルに使用される各基本立体に対する輪郭線モデルの生成機能を開発した。また，これに加え，ポリゴンの集合体（多面体）として表現された３次元物体に対する輪郭線モデルの生成機能を開発した。

1.4.5　高速モンテカルロ装置

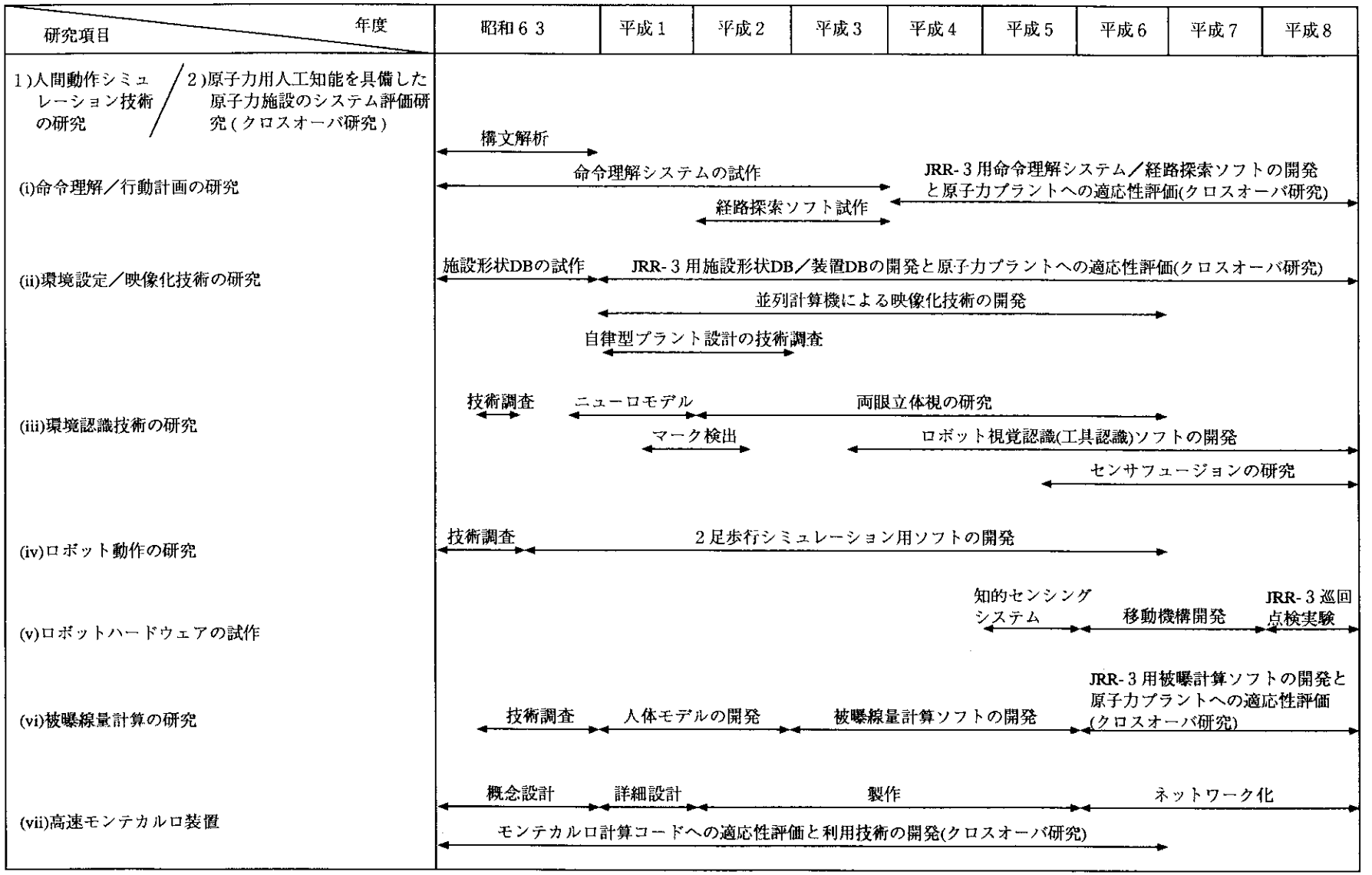
平成４年度に製作を完了した高速モンテカルロ装置の利用技術開発として，本装置上のベクトル処理，マクロ・タスク機能及びマイクロ・タスク機能を用いた並列処理による粒子シミュレーション・コードの高速化手法の開発を行った。

放射線輸送多群モンテカルロ・コードＭＯＲＳＥの高速化では，本装置によるベクトル処理及び並列処理によって，オリジナル・コードのスカラ処理時間に対して９．１倍の速度向上率を達成した。さらに，３次元相対論的粒子シミュレーション・コードｅｍ３ｐｊの高速化においては，本装置のマイクロ・タスク機能を用いた並列ベクトル処理により，オリジナル・コードのスカラ処理時間と比較し，約２５０倍という極めて高い速度向上率を達成した。

1.4.6　統合的シミュレーションシステム

これまで，個別に開発されてきた命令理解，環境認識，２足歩行，映像生成などの各システムを統合し，知能ロボットが命令を与えられてから実際の行動に至るまでの一連の振る舞いを模擬できるシステムを構築することは人工知能研究のワークベンチとしても，ハードウェアとしてのロボットシステムの設計・評価のツールとしても有用である。このため，これまでに開発した各システムをネットワークで結ばれたワークステーション群上に統合化したシミュレーションシステムのシステム構成，インタフェースについて検討し，自然言語による命令を受けてから，自己の作業目標を生成し，経路を探索し，行動するという知能ロボットの一連の振る舞いを映像化する統合的シミュレーションシステムを試作した。

Table 1.1 Research schedule of HASP.



注) 上記の1)は、主にモデル、手法等の基礎的研究開発である。2)は、1)の成果を原研の研究用原子炉JRR-3に適用し、評価するクロスオーバ研究である。

Table 1.2 Research schedule of HASP (revised in 1993).

	平成５年度	平成６年度	平成７年度	平成８年度	平成９年度
クロスオーバ研究	原子力用人工知能を具備した原子力施設のシステム評価研究(第１期)	自律型プラントの分散協調知能化システムの開発(クロスオーバ研究第２期)			
(１)原子力知識ベースの研究					
(i)命令理解／行動計画の研究	開発済システムの実用性評価	分散型知識ベースのアイディア成立性確認	分散型知識ベースの設計／試作及び知識習得基本ソフトウェアの開発		システム統合化／作業知識習得技術の検証
(ii)環境設定／映像化技術の研究	環境情報学習に最適な物体記述法の検討		環境情報の学習に適したデータベース構造の検討及び改良		システム統合化／環境情報習得技術の検証
(iii)環境認識技術の研究	ハードウェア部の開発	色／距離情報からの面情報抽出ソフト開発		視覚／聴覚／嗅覚情報の融合処理による環境認識システムの開発	
					システム統合化及び技術の適応性評価
(２)高速シミュレーション技術の研究					
(i)ロボット動作の研究	シミュレーションシステムの高度化		感覚運動学習に基づく制御法の開発、高度２足歩行ロボットの機構／制御法の設計のまとめ、開発すべき技術の提言		
(ii)ロボットハードウェアの研究	クローラ走行移動台車の整備、視覚情報に基づく走行制御法の検討		２足歩行ロボットプロトタイプの基本設計	２足歩行ロボットプロトタイプの詳細設計	２足歩行ロボットプロトタイプの製作
(iii)被曝線量計算の研究	動的ファントムによる時間依存モンテカルロ計算手法の開発		実施設に即した放射線場設定手法の開発		ロボット躯体への放射線損傷度評価手法開発
(iv)高速モンテカルロ装置	各種原子力コードによる高速モンテカルロ装置の性能評価				
(３)原子力コードの知識モジュール化手法の研究	原子炉設計知的支援システムの概念設計	原子炉設計における知識表現法の検討	計算コード知識情報モジュールの試作	知識モジュール作成支援ツールの開発	

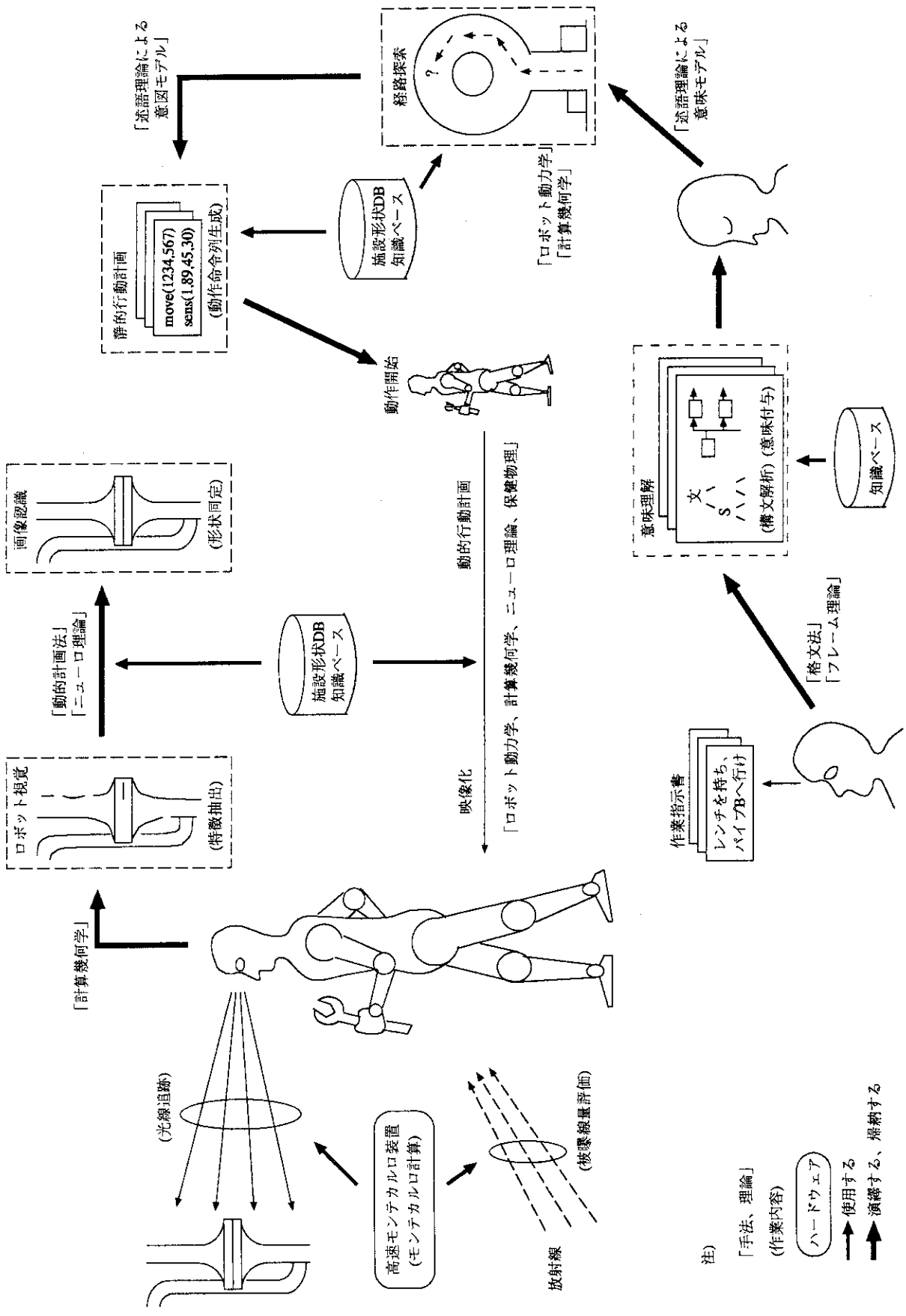


Fig. 1.1 Concept of Human Acts Simulation Program.

2. 命令理解システム

2.1 はじめに

命令理解の研究では，炉内点検作業に関する平易な日本語で記述された命令文から，自然言語解析，行動目標の決定，行動計画及び行動策定等を行いつつ，ロボット運動のソフトウェア・シミュレーションあるいは実際のロボット・ハードウェアの駆動を行うことを目標としている。この研究は昭和 62 年度から着手され，平成 4 年度までに命令理解シミュレーション・システムのプロトタイプの試作を完了し [1, 2]，原研の研究用原子炉 JRR-3 の 1 部を簡略化した仮想的な動作環境における知能ロボットの行動のシミュレーションを行った。

命令理解プロトタイプ・システムは，Lucid Common Lisp [3] 及び C 言語を使って記述されたソフトウェアであり，UNIX ワークステーション S-4/IP （富士通製）上で動作する。このプロトタイプ・システムは，非同期的な命令文の入力（割り込み命令）に対応可能な推論制御機構として，Fig. 2.1 に示した認識-判断-実行サイクルによる巡回的推論エンジンを持つ。この巡回的推論エンジンでは，Fig. 2.1 の認識部分において，命令文の入力等の状況が感知され，判断部分においてシステム及び環境の状態に依存して以下の処理から適切なものを選択することによってシミュレーションが実行される [1, 2]。

- 入力された日本語命令文から，自然言語解析サブプロセス及び行動目標決定サブプロセスにより，知能ロボットの行動目標を特定する。
- 経路探索サブプロセス，手段サブプロセス，発話サブプロセスにより，行動目標を達成するために必要となる知能ロボットの動作系列を生成する。
- 簡単な動作シミュレータによって，得られた動作系列をシミュレートする。

ここで，自然言語解析サブプロセスでは，高木・伊東の構文・意味的な係り受け構造の表現手法 [4] に基づく構文解析ソフトウェア CS-PARSER [5] を利用し，行動目標決定サブプロセス及び動作系列の生成を行う行動計画では，R.C. Schank らの提唱する概念依存性理論に基づくゴール・プランの理論 [6] を用いている。

平成 5 年度の命令理解の研究では，(1) 従来，簡単なシミュレータによってロボットの運動等を模擬していた部分について（Fig. 2.1 参照），視覚認識，ロボット運動学シミュレーション，映像生成等 HASP において研究開発を行っているソフトウェアと命令理解プロトタイプ・システムとを実際に結合するためのネットワークインタフェースの整備を行うとともに，(2) 滑らかな移動経路を導出する経路探索アルゴリズムの開発を行った。これにより，命令理解システムを中核とした統合的なシミュレーションが可能となった。以下では，まず，異なるワークステーション上で動作する各種ソフトウェア間の結合を行うネットワークインタフェースについて述べ，ついで，2 次元メッシュによって記述された移動空間地図を利用した経路探索手法についてまとめる。

2.2 分散処理のためのネットワーク・インタフェース

2.2.1 ネットワーク・インタフェースの概要及び実装方法

命令理解プロトタイプ・システムと HASP において研究開発を行っている視覚認識，ロボット運動学シミュレーション，映像生成等のソフトウェアの統合的なシステム化では，各々のソフトウェアが相当な規模のアプリケーションであるため，ローカル・エリア・ネットワークを利用した分散コンピューティングによるソフトウェアの結合が不可欠である。平成 4 年度には，分散コンピューティング環境整備の第 1 段階として，命令理解プロトタイプ・システムの推論制御部分とユーザ・インタフェース部分を別々のプロセス（コンピュータ上の処理単位）に分離するとともに，ロボットの運動を模議する簡単なプログラム（簡易ロボット・シミュレータ）を作成し，これら 3 つの独立なプロセスをプロセス間通信によって接続するクライアント・サーバ型のソフトウェア形態を整備した [1, 2]。このクライアント・サーバ型のソフトウェア構成を拡張し，命令理解プロトタイプ・システムと他のシミュレーション・ソフトウェアとの接続を行った。各ソフトウェアの接続の様子は，Fig. 2.2 に示したとおりであり，サーバ・クライアントの関係を Table 2.1 にまとめた。

プロセス間の通信手段としては，Internet ドメインのソケット・ストリームによるプロセス間通信を利用した。ソケット・ストリームによるプロセス間通信処理の流れは，(1) 初期化時のソケットの接続，(2) データ送受信（入出力）の実施，(3) 終了時のソケットの切断の 3 つに大別される。初期化時のソケットの接続は，サーバ・プロセス及びクライアント・プロセスのそれぞれについて，以下の手順で行う。

サーバ側:
- step 1: ホストに関する IP アドレス等の情報を得る。
- step 2: ソケットを生成する。
- step 3: ソケットにアドレスを結びつける。
- step 4: 接続要求を待つ数を 1 に設定する。
- step 5: クライアント側からの接続要求を待つ。
- step 6: 接続要求があれば接続する。

クライアント側:
- step 1: ホストに関する IP アドレス等の情報を得る。
- step 2: ソケットを生成する。
- step 3: サーバ側に接続要求を出す。
- step 4: 接続要求が受理されれば接続する。

上記のようにしてソケットが接続されると，以後は，ソケット（ファイル・ディスクリプタ）に対する `read` / `write` システムコールによってデータの入出力が可能となる。終了時には，ソケットに対する `close` システムコールによって接続が切断される。命令理解プロトタイプ・システムの推論エンジン部分は，Common Lisp 言語によって記述されているため，C 言語で記述されたネットワーク・インタフェース・ルーチンを直接呼び出すことはできない。ここでは，Lucid Common Lisp 処理系に用意されている FFI (Foreign Function Interface) 機能を利用して C 言語ルーチンの呼び出しを実現した。Table 2.2 にネットワーク・インタフェースを通じて送受信されるデータをまとめた。

今回の分散処理のためのネットワーク・インタフェース環境の整備によって，３台の異なるワークステーション上で処理される命令理解プロトタイプ・システム，簡易ロボット・シミュレータ，自己位置認識システム，２足歩行ロボット可視化システムを効果的に接続することが可能となった。現状では，２足歩行ロボット可視化システムに割り込み命令を送信することができないため，任意のタイミングでの割り込み命令を使ったロボットシミュレーションは不可能である。また，ロボット・ハードウェアの制御プロセスは単に自己位置認識システムとの接続のためにのみ用意されている。今後，命令理解プロトタイプ・システムの割り込み命令処理機能を活かしたシステムの統合化，そして，センシング・システムの整備に伴い，自己位置認識システムとセンシング・システムとの連携が予定されるため，この部分の接続形態を，命令理解プロトタイプ・システムとセンシング・システムとの接続へ変更すること等が今後の課題としてあげられる。

2.2.2 ネットワーク・インタフェース機能を用いた Lisp パラメータ入力

Lisp プログラムは，Lisp インタプリタによって処理されるため，プログラムで使用するパラメータ，関数，マクロ等をインタプリタに入力することによって，プログラムの実行中に会話的に変更することができる。しかし，命令理解プロトタイプ・システムのように比較的大規模なアプリケーション・ソフトの場合には，パラメータの数が非常に多くなるため，パラメータの変数名を正確に把握することが困難となるため，パラメータの入力作業が極めて繁雑なものとなる。今回，2.2.1 で述べた分散処理のためのネットワーク・インタフェースと XView ツールキット [7] を利用して，メニュー形式で命令理解プロトタイプ・システムのパラメータを与える機能を整備した。

命令理解プロトタイプ・システムの巡回的推論エンジンは，Fig. 2.1 に示すように，認識-判断-実行サイクルを持つ。この推論エンジンの基本プログラムは，以下に示す構成である。

```
(loop
  ;; 認識部分
  (cs2_cognition)                    ; 認識部分本体
  (if3_infermon_after_cognition)     ; 推論過程モニタへの出力
  (if1_lisp_listener)                ; Lisp インタプリタ
  ;; 判断部分
  (cs3_decision)                     ; 判断部分本体
  (if3_infermon_after_decision)      ; 推論過程モニタへの出力
  (if1_lisp_listener)                ; Lisp インタプリタ
  ;; 実行部分
  (cs4_execution)                    ; 実行部分本体
  (i3_infermon_after_cognition)      ; 推論過程モニタへの出力
  (if1_lisp_listener)                ; Lisp インタプリタ
)
```

Lisp 処理系への入力は，関数 `if1_lisp_listener` によって，Lisp エバリュエータに引き渡され，実行される。したがって，Fig. 2.2 に示したユーザインタフェース部分と推論エンジン

部分のソケットを使うことによって，ユーザインタフェース部分から推論エンジンへのデータの受渡しが可能である。

そこで，XView ツールキットで記述されたユーザインタフェース部分に，Lisp パラメータの入力あるいは選択メニューを新たに用意し，Lisp 入力ストリームに対して，S 式（Lisp のデータ表現）を送信する形態を実現した。これにより，100 個以上にのぼる命令理解プロトタイプ・システムのパラメータ入力を効率的に行うことが可能となった。

2.2.3 自己位置認識用マーク選択処理

命令理解プロトタイプ・システムと自己位置認識システムとの接続にあたって，新たに自己位置認識に用いるマークの選択処理をプロトタイプ・システムに追加した。この処理は，本来，自己位置認識システムの 1 部として取り扱われるべきものである。しかし，現状ではロボット・ハードウェアを特定することができずロボット制御プログラムの詳細仕様を決定することが不可能なため，自己位置認識システムの入力データとなるロボットの推定位置を何らかの形で用意する必要がある。このため，今回の統合的なシミュレーション環境の整備では，つねにロボットの位置を把握している命令理解システム側に，暫定的に，自己位置認識用マークの選択処理機能を持たせることとした。

自己位置認識システムでは，(1) ロボットの作業空間にあらかじめマークを配置し，(2) ロボットの推定位置情報とマークの配置情報を利用して，カメラによって撮像すべきマークを選択し，そして，(3) 得られた画像からロボットの現在位置を特定する。今回，命令理解プロトタイプ・システムに追加したマーク選択処理では，マークの配置情報とロボットの推定位置から，以下に示した処理をへて，カメラの可動範囲内にあり，最短距離にあるマークを選び出す。

step 1: すべてのマークについて，ロボットの推定位置からの水平距離，カメラの制御角（θ, ϕ, Fig. 2.3 参照）を計算する。

step 2: 水平距離に関して，マークをソートする。

step 3: 水平距離の短いマークから，カメラの可動角範囲内にあるかチェックし，最初に可動角に関する条件を満たしたマークを選ぶ。

上記のアルゴリズムでは，配置したマークのすべてについて，カメラからの水平距離等の計算を行うため，計算量が膨大なものとなることが予想される。しかし，ハードウェア実験のために用意した模擬環境（約 8 m × 11 m）[2] について，マークの配置を行ったところ，マークの総数は 20 程度であり，総当たりによるマーク選択でも，十分実用的であることが明らかになった。

マーク選択処理を行うプログラムは，C 言語で記述されたサブルーチンであり，Lucid Common Lisp の FFI 機能を使って，命令理解プロトタイプ・システムからプロシジャ・コールの形式で実装した。今後，センシング・システムの整備に伴い，マーク選択処理を適切なソフトウェアに移行していくことが今後の課題である。

2.3 2次元正方メッシュによる経路探索手法

命令理解プロトタイプ・システムでは，ロボットの移動経路の探索を2段階に分けて行っている [1]。一方は，ロボットの作業空間をエリアと呼ばれる部分空間に分割した際に得られるエリア間の位相的接続関係を利用したものであり，他方は，1つのエリア内で適用するものである。後者では，エリアを2次元正方メッシュに分割して，メッシュ間の接続関係から，目的地までのメッシュのリストとして移動経路を作成する。

2次元正方メッシュを利用した経路探索では，ヒューリスティックなルールを利用した縦型探索によって，出発地点から目標地点までのメッシュの接続関係を求める。このアルゴリズムでは，メッシュサイズを適切に選ぶことによって，比較的効率良く探索を行うことができる。しかし，得られるメッシュ・リストを直接ロボットの移動経路として適応すると，当然のことながら，経路がジグザグ状になり，実際のロボット・ハードウェアの移動経路としては好ましくない。そこで，縦型探索によって得られたメッシュ・リストに対して，後処理を施し，直線的な移動経路を生成する機能を追加した。

直線的な移動経路の決定は以下のアルゴリズムで行う（不要メッシュ除去前の経路を当初の【経路】とする）。

step 1: 【経路】の長さが2以下であれば,【経路】を結果として返す。

step 2: 【経路】の先頭のメッシュを【仮出発地】，末尾のメッシュを【仮目的地】とする。

step 3: 【仮目的地】と【仮出発地】との間に他のメッシュが無い場合には,【仮出発地】と,【経路】から【仮出発地】を取り除いた経路についてこの処理を再帰的に適用した結果とをリストにしたものを結果として返す。

step 4: 関数 `sps3_get_via_node_list` を用いて,【仮出発地】と【仮目的地】を直線的に結んだ場合のメッシュのリスト【仮想経由座標リスト】を求める。

step 5: 【仮想経由メッシュリスト】の各メッシュについて匡体干渉チェックを行ない，全てＯＫなら（つまり,【仮出発地】と【仮目的地】との間を直進可能であるなら，関数 `sps3_straightway_p` で判定する),【仮出発地】と,【経路】から【仮出発地】よりも後ろにあるものだけを残した経路についてこの処理を再帰的に適用した結果とをリストにしたものを結果として返す。

step 6: 【仮目的地】よりも一つ手前のメッシュを改めて【仮目的地】として，step 3 に戻る。

簡単に表現すると，移動経路の端からなるべく長い区間で短絡できる部分を捜して，短絡可能な部分の中間メッシュを取り除くということである。関数 `sps3_get_via_node_list` では，対象区間の両端を出発地及び目的地として，出発地から目的地に向かって近づく方向の3点，すなわち,【点1】：x 方向と y 方向に1メッシュ近づく,【点2】：x 方向のみ1メッシュ近づく,【点3】：y 方向のみ1メッシュ近づく，を候補として，これらの各点と目的地を結ぶ

線分の傾きと，出発地と目的地を結ぶ線分の傾きとの差が最も小さい点を選択し，その点を新しい出発地として同様の処理を繰り返し，経由した点のメッシュのリストを求める。但し，半分終わった段階で傾きの算出を各点と出発地を結ぶ線分の傾きに切り替えることによって，対称的な結果を得るようにする。このような処理によって，直線的に結んだ場合のメッシュのリストを求めている。関数 `sps3_straightway_p` では，その経路に添って直進が可能であるかをロボットの匡体が他の物体と干渉しないかどうかのチェックと共に行なっている。Fig. 2.4 に，縦型探索によって得られた経路とともに，このアルゴリズムによって得られた移動経路を例示した。

2.4 まとめ

本章では，平成4年度の命令理解の研究について述べた。今回の研究では，主として，HASPにおける各種シミュレーション・ソフトウェアと命令理解プロトタイプ・システムとの接続を中心としてプログラムの開発整備を行った。Lisp 処理系へのパラメータの受渡しにこのネットワーク・インタフェースを利用したため，命令理解プロトタイプ・システムのウィンドウ・イメージは，Fig. 2.5 に示すように，使い勝手のよいものとなった。さらに，ここで整備した分散処理のためのネットワーク・インタフェースは，極めて汎用的な構造であり，今後整備されるセンシング・システム及び関連ソフトウェアとの接続にも応用可能である。2次元正方メッシュを利用した経路探索手法では，滑らかな移動経路を求めることが可能となり，シミュレーション結果の可視化においても，より現実的なイメージを得ることができた。

参考文献

[1] S. Kambayashi and Y. Abe, *Instruction Understanding for Intelligent Robots in Nuclear Facilities*, in Proc. of Joint Imternational Conference on Mathematical Methods and Supercomputing in Nuclear Applications, edited by H. Künster, E. Stein and W. Werner (Kernforschungszentrum, Karlsruhe, 1993), pp. 398 – 407.

[2] 秋元 正幸 他：**原子力知能化システム技術の研究 – 平成 4 年度作業報告 –**, JAERI-M 94-051, 日本原子力研究所 (1994.3)。

[3] *Lucid Common Lisp/Sun Version 4.0: User's Guide* (Lucid Inc., 1991).

[4] 高木 朗，伊東 幸宏：**自然言語の処理**，丸善 (1987)。

[5] **CSK 日本語解析システム CS-PARSER 使用マニュアル**，(株) CSK (1988)。

[6] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals, and Understanding* (Lawrence Erlbaum Associates, N.J., 1977).

[7] D. Heller, 石川和也 監訳：**XView ver.3 プログラミング・マニュアル**，ソフトバンク (1992)。

Table 2.1 Server-client relation in the network communication between the instruction understanding system (IUS), visualization system for biped locomotion robot, simple robot simulator, controlling process for hardware robot and localization system. Note that the robot hardware is not available currently and the controlling process for the robot hardware is prepared just for the localization system.

Server	Client
User interface of IUS	Inference engine of IUS
Inference engine of IUS	Simple robot simulator
Visualization system for biped locomotion robot	Inference engine of IUS
Controlling process	Inference engine of IUS
Controlling process	Localization system

Table 2.2 Specification for data communication between the instruction understanding system, simple robot simulator, controlling process for hardware robot and visualization system for biped locomotion robot. Note that the robot hardware is not available currently and the controlling process for the robot hardware is prepared just for the localization system

簡易ロボット・シミュレータあるいはロボット制御プロセスへの送信データ		簡易ロボット・シミュレータあるいはロボット制御プロセスからの送信データ	
double x;	目標位置 x (mm)	double x;	目標位置 x (mm)
double y;	目標位置 y (mm)	double y;	目標位置 y (mm)
double z;	目標位置 z (mm)	double z;	目標位置 z (mm)
double th;	目標位置 θ (度)	double th;	目標位置 θ (度)
double errx;	許容誤差 x 成分 (mm)	int opt;	補助フィールド
double erry;	許容誤差 y 成分 (mm)	int id;	コマンド番号
double errth;	許容誤差 y 成分 (度)		
int com;	動作種別		
2足歩行ロボット可視化システムへの送信データ			
double x0;	現在位置 x (mm)	int aflag;	自動映像生成フラグ
double y0;	現在位置 y (mm)	int vflag;	視点調整フラグ
double z0;	現在位置 z (mm)	double vx;	視点方向ベクトル x
double th0;	現在位置 θ (度)	double vy;	視点方向ベクトル y
double x1;	目標位置 x (mm)	double vz;	視点方向ベクトル z
double y1;	目標位置 y (mm)	double ex;	視点位置 x (mm)
double z1;	目標位置 z (mm)	double ey;	視点位置 y (mm)
double th1;	目標位置 θ (度)	double ez;	視点位置 z (mm)
int com;	歩行種別		

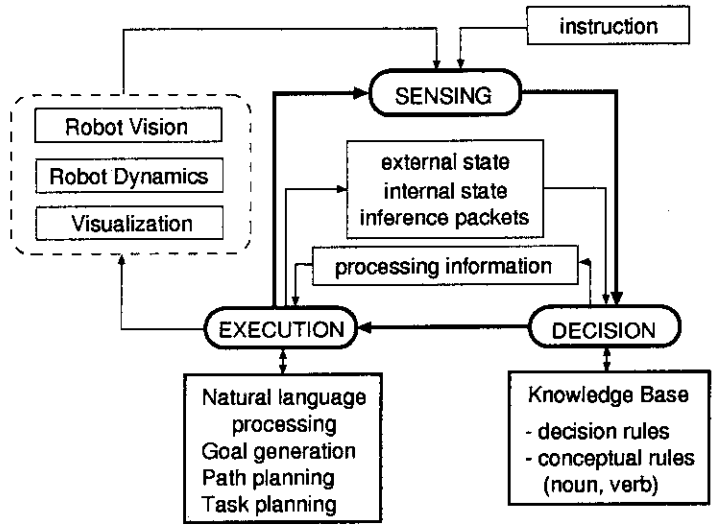


Fig. 2.1 System configuration of the instruction understanding system. Thick arrows denote flow of processing, i.e., the flow of the cyclic inference engine. Thin arrow denote data flow

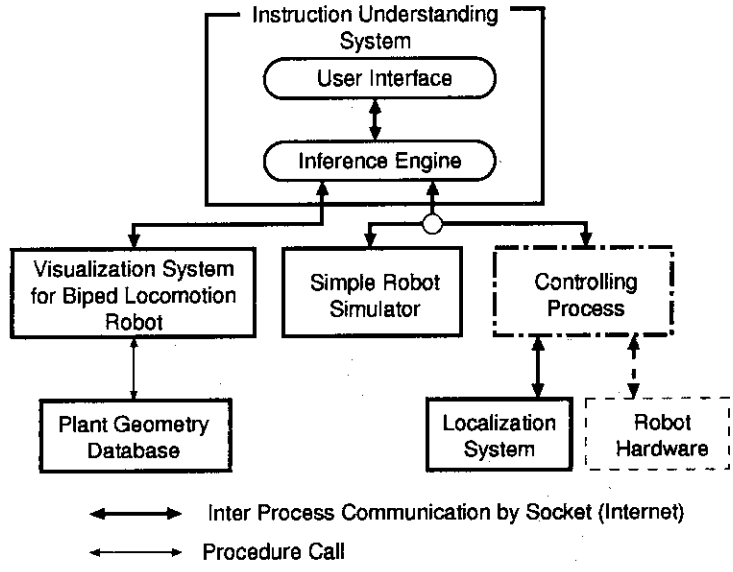


Fig. 2.2 Schematic flow of network communication between the instruction understanding system, visualization system for biped locomotion robot, simple robot simulator and localization system. Note that the robot hardware is not available currently and the controlling process for the robot hardware is prepared just for the localization system. The circle in communication link denotes a switch which is used to choose either the computer simulation or the operation of hardware.

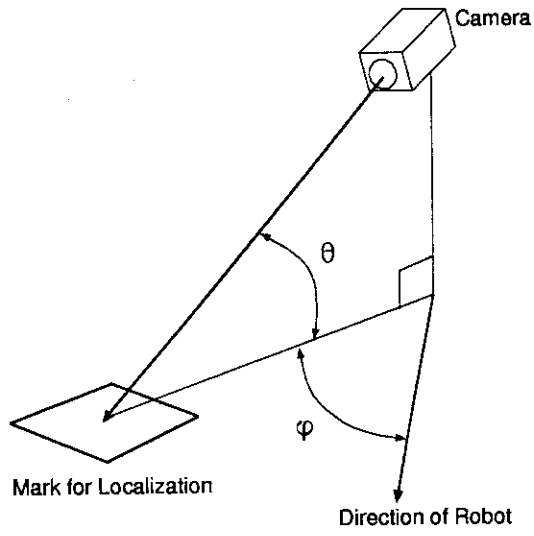


Fig. 2.3 Schematic relation between the camera and mark for localization of the robot. Two angles θ and ϕ are obtained in the instruction understanding system by using estimated location of the robot.

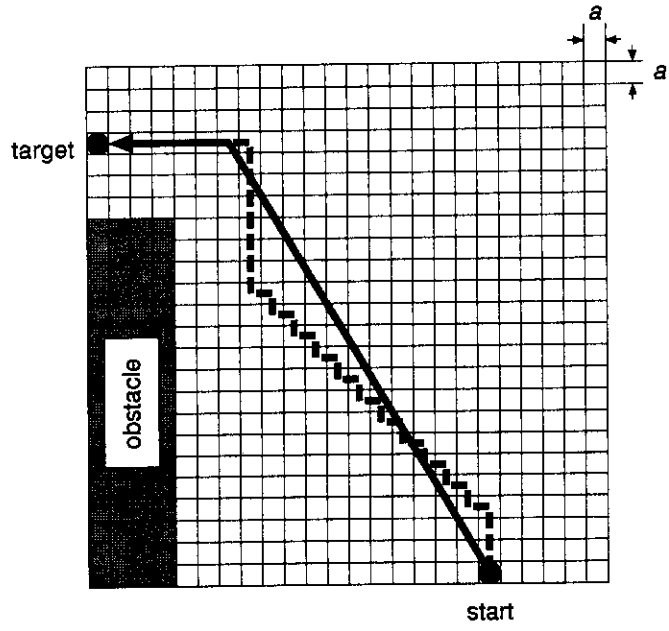


Fig. 2.4 An example of the path planning by using the two dimensional regular mesh with the mesh size a. Dashed arrow represents the initial path obtained by the depth-first search algorithm. Solid arrow represents the final path obtained by the smoothing of the initial path.

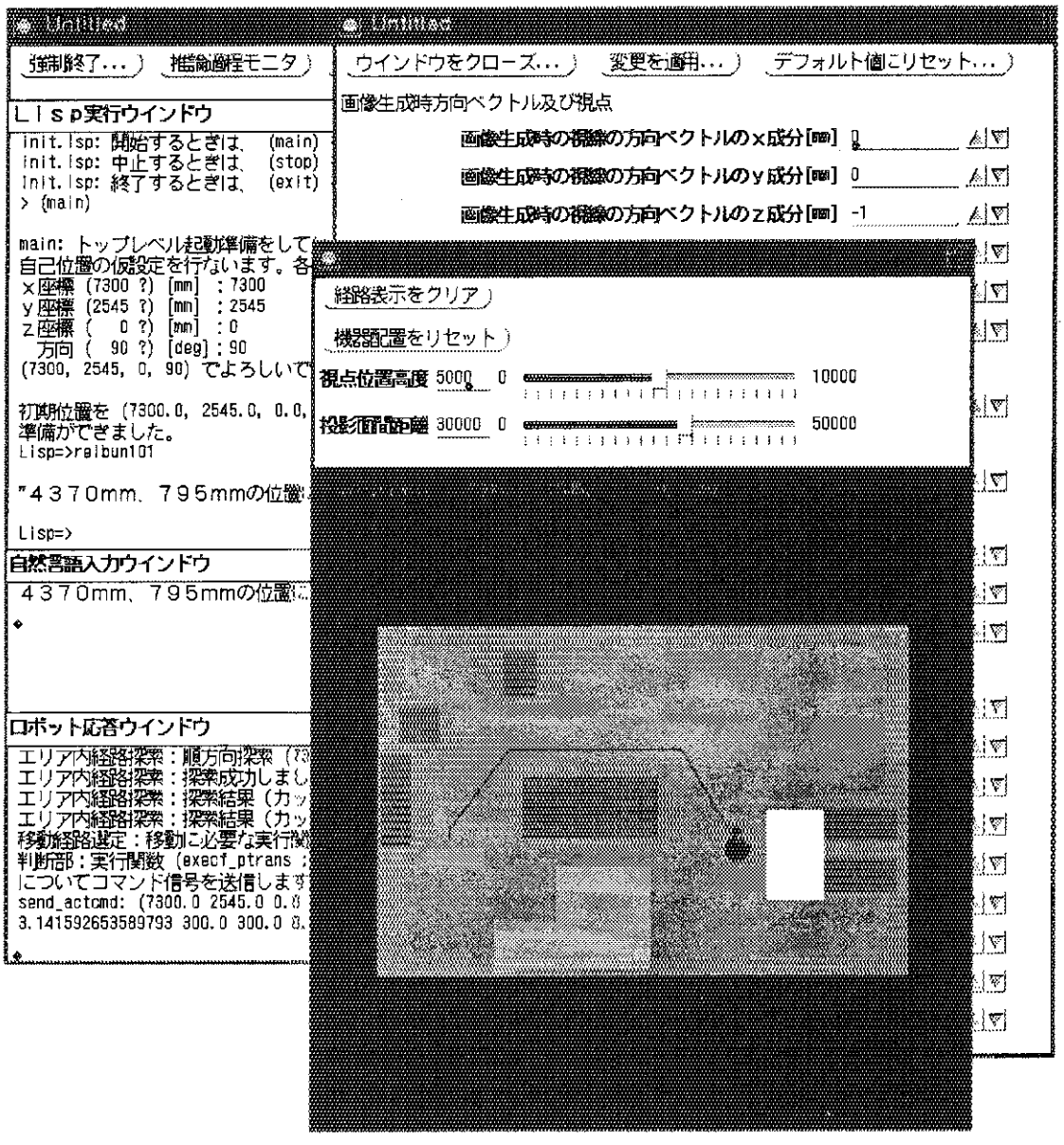


Fig. 2.5 Window image of the prototype of the instruction understanding system. Parameter setting panel and path planning image are also shown.

3. 環境認識技術の研究

3.1 はじめに

人間の代替として，原子力施設等の環境で作業を行なうロボットを考えた場合，そのロボットは，その作業環境を，自ら検出，認識し，さらにその結果に基づいて，次に行なうべき行動を判断する能力を持った自律的な知能ロボットであるべきである。特に，周囲の環境と関係する'検出'や'認識'の能力は，ロボットが行動する上で最も重要な能力である。よって，知能ロボットの設計開発を目標としたとき，知能ロボットによる周囲環境の状態の検出，認識についての研究は，必要不可欠な研究の１つである。そのような理由から，"人間動作シミュレーションプログラム:HASP" の研究では，その一環として，計算機を用いた環境認識技術の研究を行なっている。

HASP の研究では，計算機によって，知能ロボットによる原子力施設内における巡回点検作業のシミュレーションを行なうことによって，知能ロボットを設計開発するための要素技術の研究開発を行なっている。しかし，知能ロボットによる環境状態の認識を考えた場合，必ず，実際の世界との相関を考慮しなくてはならないため，計算機内の世界だけで，環境認識技術の研究を進めて行くことは不可能となる。そこで，ロボットの作業環境の状態の情報を得るために，原子力施設内を巡回点検する知能ロボットに搭載され，周囲環境の状態を検出する装置である "原子力施設巡回点検用センシングシステム (以下，センシングシステムとする)" の製作を行なった。以下，第 3.2 節において，この装置の詳細について記述する。

人間は，視覚，聴覚，嗅覚，触覚等のさまざまな感覚を用いることによって周囲環境の状態を検出し，その検出されたデータに基づいて，その環境状態の認識を行なっている。特に，環境状態を検出するためのこれらの感覚の中において，視覚は，検出する情報の種類や量の多さから，最も汎用的で有用な感覚である。そのため，知能ロボットによる周囲環境の認識も，人間の視覚と同様に，周囲の環境を撮影した画像に基づいて行なわれることが理想的であると考えられる。そこで，将来，知能ロボットに，視覚によって周囲環境の状態を認識するための能力を与えることを目標とし，今回，カラー CCD カメラによって撮影された環境内の物体の画像から，ある特徴量を抽出するソフトウェアの開発を試みた。特に，環境中の物体の画像から，その物体の特徴量を抽出することによって，その物体の同定や認識を行なうことが可能になる。また，逆に，抽出された物体の特徴の位置と，ロボットがあらかじめ持っているその特徴の位置に関する情報から，ロボットの現在位置を同定することも可能となる。このソフトウェアの開発については，第 3.3 節において記述する。

3.2 原子力施設巡回点検用センシングシステムの製作

この節では，原研情報システムセンターにおいて，平成 4 年度に概念設計を行ない，平成 5 年度に製作したセンシングシステム [1] について記述する。

知能ロボットによる巡回点検を模擬する場合，知能ロボットに搭載された各種センサよる周囲の環境状態のセンシングや，計算機を使った周囲環境の状態の認識を模擬することが必要となる。しかし，例えば，CCD カメラによって周囲環境の様子を画像として検出することを考えた場合，コンピュータグラフィックスを用いて生成した画像を用いて，実際の CCD カメラで撮影した画像を正確に模擬することは不可能である。なぜなら，コンピュータグラフィックスの画像は，実際のカメラで撮影した画像に生じるカメラの光学系の歪みや画像転送時のノイズ，また，さまざまな方向からの照明の反射光によってできる微妙な影などを，正確に模擬するとが不可能であるためである。また，将来，知能ロボットを設計することを目指した要素技術開発を行なっていることから，各種センサから得られたデータを処理するためのアルゴリズムは，計算機によって模擬された理想的なデータに対して有効であるよりも，実際の環境から実際のセンサを用いて検出されたデータに対して使用可能なアルゴリズムが望ましいことは明らかである。そこで，HASP における研究の一貫として行なっている環境認識技術の研究において，知能ロボットによるセンシングの実験を行なうために，将来の知能ロボットのセンサとなるセンシングシステムを設計･製作した。

センシングシステムは，知能ロボットに搭載され，ロボットが人間の代替として原子力施設内において巡回点検を行なうときに，周囲環境の状態を検出する装置として設計された。原子力施設内で，人間である作業員は，その感覚器を用いて点検作業を行なっている。そのとき，特に，視覚，聴覚，嗅覚は重要な働きをする。例えば，機器からの漏水や蒸気漏れ，機器の破損，メータの読み取りなどは，すべて視覚によって行なわれる。また，回転機器の異常音の察知，潤滑油の焼けた臭いなどは，それぞれ聴覚，嗅覚によって行なわれる。そこで，今回，製作したセンシングシステムは，知能ロボットの視覚の働きをする視覚的センサ，聴覚の働きをする聴覚的センサ，嗅覚の働きをする嗅覚的センサを備えたものとして製作した。また HASP の研究では，知能ロボットの作業環境として，原子力施設内を想定していることから必要不可欠となる放射線センサも備えている。

センシングシステムの視覚的センサは，人間の視覚のように周囲環境の状態を映像として検出し，また環境内の対象物までの距離を得るために，単板カラー CCD カメラとレーザー光を用いた距離センサを検出機器として使用している。カラー CCD カメラからは，640 × 480 ピクセルの大きさで，8 bit の輝度レベルを持つカラー画像を出力し，距離センサは，カラー画像と同じ大きさで，1 輝度レベルが，実際の距離 2cm に対応するような距離画像を出力する。模擬環境で撮影した，それぞれの画像の例を Fig. 3.1，3.2 に示す。聴覚的センサは，検出装置として指向性を持ったコンデンサマイクロホンを使用し，検出した音の周波数分布を出力する。CCD カメラとレーザー投光装置，マイクロホンは，同一の旋回台に設置され，上下左右のある範囲内の方向を向くことが可能である。嗅覚的センサは，検出部分に 4 種の半導体ガスセンサを使用し，それぞれは，複合臭気，硫化水素，排ガス，可燃性ガスに対して

反応するものである。また，気体分子の分布に影響のある温度，湿度を測定するための温度·湿度センサも備えている。嗅覚的センサは，出力として，半導体センサの抵抗の変化に伴って変化する電圧に関係した値を出力する。そのため，出力された値は，臭気気体の絶対的な量を検出するのではなく，むしろ，ある基準値に対する相対的な値を出力する。実際の人間の嗅覚を考えた場合，人間は，においの成分の絶対的な量を察知しているのではなく，そのにおいが，普通より強いかどうかを察知していることから，この出力方法は妥当なものであると考えられる。放射線センサは，電離箱式サーベイメータを使用し，γ線の量を測定する。このサーベイメータは，原子力施設内で作業員が使用するのものと同程度のものである。そして，これらの各センサ機器や旋回台は，ワークステーション (WS) を通じて制御され，検出されたデータは WS 内のファイルとして保存される。

さらに，このセンシングシステムの制御や状態表示を行なうための GUI (Graphical User Interface) を開発した。Fig. 3.3 に，センシングシステムの旋回台の向きや傾き，音の周波数分布，嗅覚的センサの出力値や温度湿度，さらに放射線量の変化を表示するセンシングシステム状態表示画面を示す。また，このパネルの各種のボタンを押すことによって，手動操作パネル，自動動作パネル，センシングシステム用サーバパネル，前処理設定変更パネルを表示させることができる。手動操作パネル (Fig. 3.4) では，旋回台の回転制御，CCD カメラのレンズのフォーカスやズーム，レーザーや照明の on/off，各種センサの動作を手動で行なうことが可能となる。このとき，各種センサの動作状態は，前処理設定変更パネル (Fig. 3.6) によって設定できる。このパネルでは，レーザー走査のピッチ，走査開始角度，終了角度，距離の測定を行なうときにレーザーの赤色を画像から検出するためのしきい値，聴覚，嗅覚，放射線センサのサンプリング数を設定することができる。また，自動動作パネル (Fig. 3.5) では，旋回台の回転角や照明の on/off に対する目標値を入力することによって，自動的にそれぞれの機器を動作させることが可能である。さらに，センシングシステム用サーバーパネル (Fig. 3.7) を用いることにより，この GUI は，プロセス間通信の形式によって，すでに開発されている命令理解システム及び特殊なマークを用いた自己位置同定システムと接続し，データの送受信を行なうことが可能である。この接続により，センシングシステムは，原理的に命令理解システムに入力された日本語の命令の内容に従って動作することが可能となり，またセンシングシステムの CCD カメラを使用し，特殊なマークを撮影した画像を用いて，移動ロボットの自己位置を同定することも可能となる。

センシングシステムの移動ロボットに搭載される部分 (つまり，各種センサや旋回台) と制御·データ処理用の WS とは，無線器を使った通信が可能である。また，状況に応じて有線による通信も可能である。ここで記述した機器の詳細な仕様については Table 3.1，3.2，3.3 においてまとめる。

Table 3.1 The specification of the sensor part of sensing system

視覚的センサ	CCD カメラ	種別	単板カラー CCD カメラ
		画素数	769 × 494
		解像度	水平 460 本，垂直 350 本
	レンズ	種別	電動ズームレンズ
		焦点距離	$12.5mm \sim 75mm$
		絞り範囲	$F1.8 \sim F360$
		フォーカス	$1m \sim \infty$
		画角範囲	$28.7^\circ.7^\circ \sim 4.9^\circ \times 3.7^\circ$
	レーザー投光装置	光源	半導体レーザー
		水平走査	方　式　ガルバノミラーによる連続走査
			走査範囲　45°
			走査速度　480 Hz
		垂直走査	方　式　ガルバノミラーによる方位制御
			走査範囲　50°
			方位分解能　0.1°
	受光装置		CCD カメラを兼用
	旋回台	種　別	電動 2 軸 (パン，チルト)
		旋回範囲	左右 各 90°
			上下 上 60° 下 90°
		旋回速度	左右 約 15° /s
			上下 約 15° /s
		停止精度	± 1° 以下
聴覚的センサ	マイクロホン	種別	コンデンサマイクロホン
		周波数特性	$100 \sim 10000Hz$
		指向特性	全指向/単一指向/鋭指向 切り替え式
	旋回台		視覚センサと兼用
嗅覚的センサ	ガスセンサ	種別	半導体ガスセンサ
			(基盤型薄膜及び，熱線型焼結半導体センサ)
	温度･湿度センサ	検知方式	自然拡散ガス検知式
放射線センサ	検出装置	種別	電離箱式サーベイメータ
		測定線量	$30keV \sim 2Mev$ の γ 線
		測定範囲	$1\mu Sv/h \sim 100mSv/h$
		時定数	$\mu Sv/h$ レンジ 約 6 秒，
			mSv/h レンジ 約 1 秒

Table 3.2 The specification of the communication part of sensing system

ロボット側通信装置	送信部	映像音声信号	方式	NTSC
			周波数	97.25 MHz (映像)
				101.75 MHz (音声)
			出力レベル	110 dBμ
		データ信号	方式	FSK
			周波数	74.5 MHz
			出力レベル	100 dBμ
			伝送速度	4800 BPS
	受信部	データ信号	方式	FSK
			周波数	43.5 MHz
			入力レベル	50 ～ 90 dBμ
固定局通信装置	送信部	データ信号	方式	FSK
			周波数	43.5 MHz
			出力レベル	120 dBμ
			伝送速度	4800 BPS
	受信部	映像音声信号	方式	NTSC
			周波数	97.25 MHz (映像)
				101.75 MHz (音声)
			出力レベル	60 ～ 85 dBμ

Table 3.3 The specification of the control part of sensing system

ワークステーション	機種	ラップトップ型ワークステーション P2200 (Panasonic)
	CPU	Super SPARC CPU (33 MHz)
	メモリ	メイン 32MB，フレームメモリ 1MB
	OS	PanaOS/S 1.0 (Solaris 1.0 相当 + JLE)

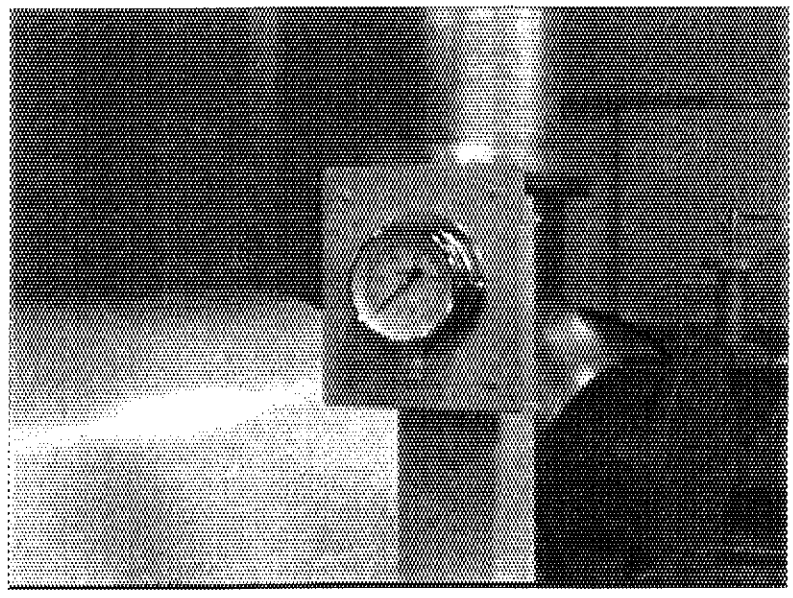


Fig. 3.1 The intensity image taken by the visual sensor

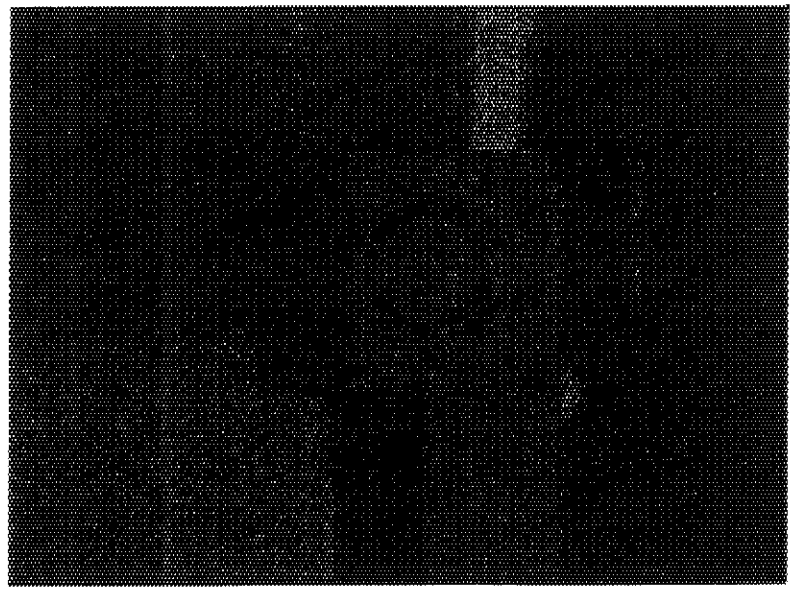


Fig. 3.2 The range image taken by the laser range-finder

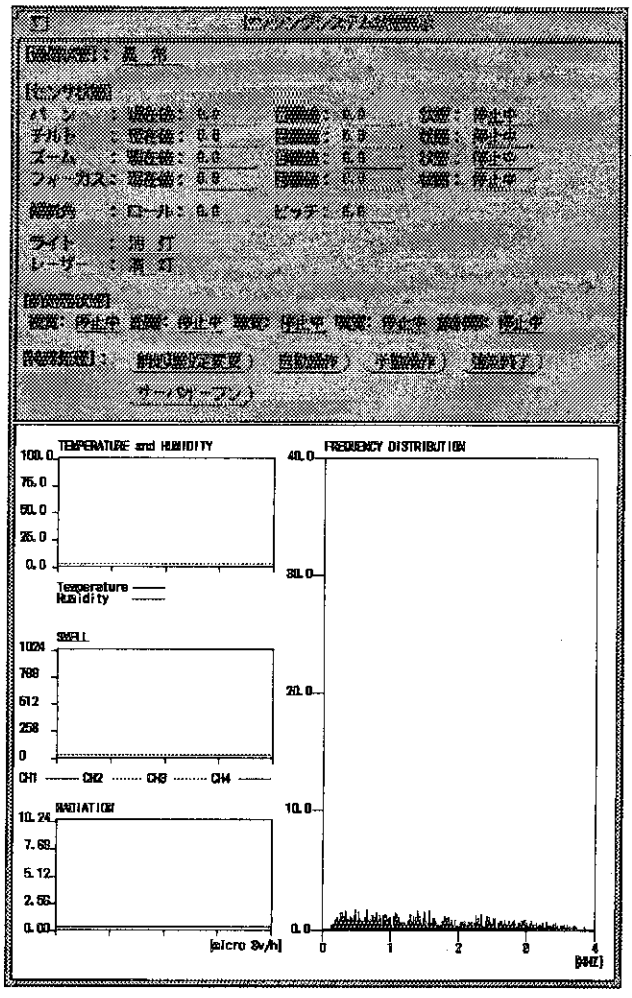


Fig. 3.3　GUI for controlling the sensing system

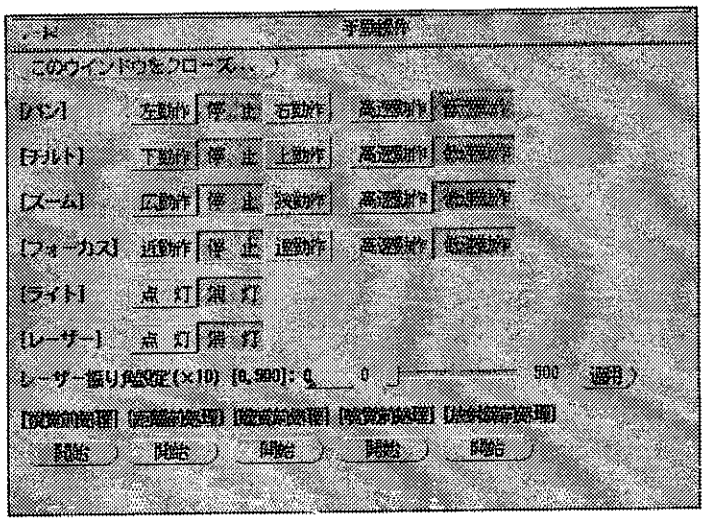


Fig. 3.4 The panel for manually controlling the sensing system

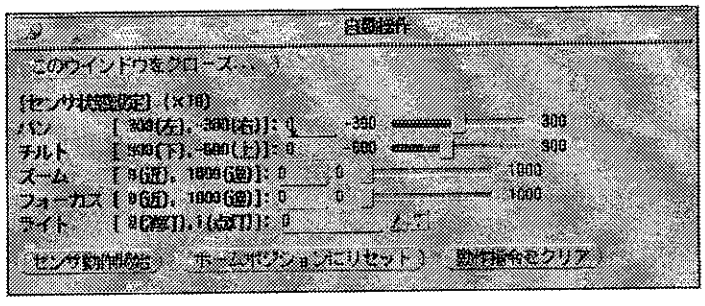


Fig. 3.5 The panel for automatically controlling the sensing system

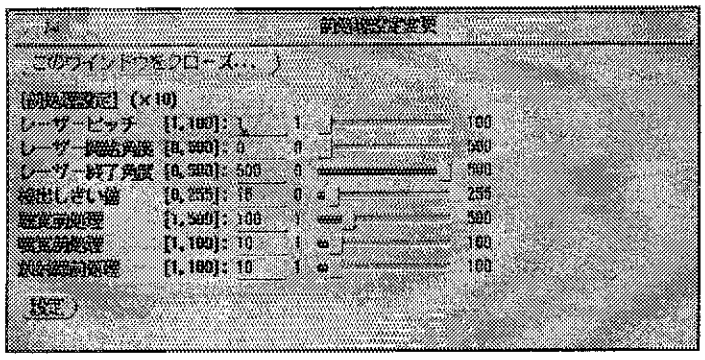


Fig. 3.6 The panel for changing some variables of sensors

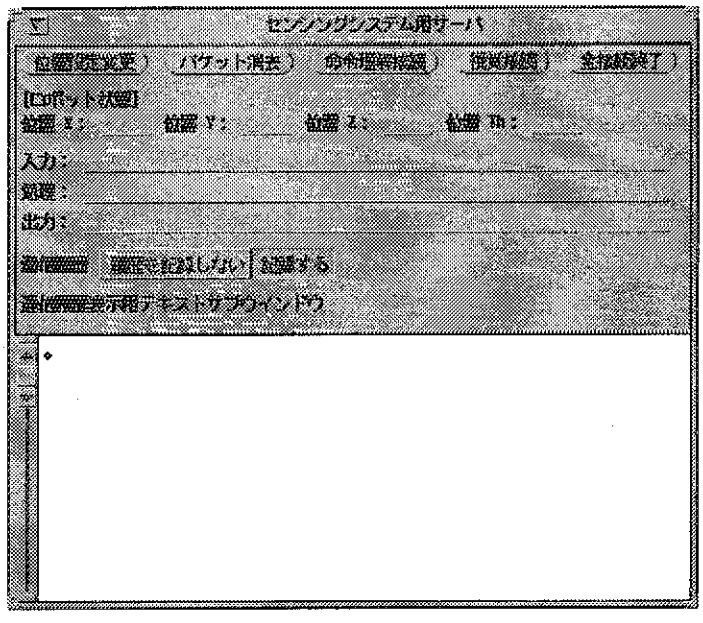


Fig. 3.7 The panel for connecting with other systems

3.3 環境内物体の画像からの特徴抽出について

一般的に，環境内を移動するロボット (移動ロボット) は，エンコーダ等の測定器を用い，環境内における自分自身の現在位置と向きを測定しながら移動している。しかし，移動ロボットの移動方法が，車輪式であろうと，脚式であろうと，環境の床面との接触点における滑りのため，長時間移動することによって，検出誤差が累積され，その位置に関する値の信頼性は著しく減少する。そこで，移動ロボット自身に搭載されたエンコーダ等の測定装置によって測定された移動ロボットの位置と向きを，移動ロボットが持つカメラによって撮影された環境内の対象物の画像から抽出された特徴点と，あらかじめ持っている環境内におけるその特徴点の位置情報を用いて修正することが必要である。つまり，実際に撮影された画像から抽出した特徴点の位置と，推定位置から撮影されるべき画像から抽出されるべき特徴点の位置との差から，移動ロボットの実際の停止位置と向きを，よりよい値で求めることである。

HASP における研究では，すでに特殊なマークを用いた自己位置同定システムが開発されている。しかし，この方法では環境内に特殊なマークを多数設置する必要がある。つまり，ロボットの作業環境をロボットに適した環境に変えることとなる。作業環境を変えることは，ロボットの作業環境を制限することになるため，ロボットの汎用性を損なうこととなる。そのため，特殊なマークを使用して，現在の位置と向きを同定する方法は，将来，開発を目指している知能ロボットに適している方法とは言えない。そこで，より一般的な方法として，CCD カメラによって検出された，環境中に設置されている機器の画像から特徴点を抽出し，その情報から自己位置の修正を行なうことを考える。そのため，本報告書において，CCD カメラで検出された画像から環境中の機器の特徴点を抽出するためのソフトウェアの開発について記述する。

今回の研究では，自己位置を同定するための環境内の物体の特徴として，環境中の機器を設置している台座の稜線の方向及びその稜線が作る頂点を用いることとした。その理由は，原子力施設内のほとんどの機器は，水漏れ等により床に水が溜まった場合を考慮し，台座上に設置されているため，その台座の稜線と頂点を使用することで移動ロボットは環境内のかなり任意の位置で自己位置の同定が可能となることである。また，台座の稜線及びそれらによって作られる頂点を撮影された画像から検出することが，比較的容易であることも１つの理由である。

この方法の概要は以下のとおりである。

(1) エンコーダ等の測定装置の測定結果から推定される移動ロボットの位置から撮影可能な最も近い台座の頂点を選び出し，その頂点がカメラの光軸上にのるようにカメラを向け，頂点の周囲の画像を抽出する。

(2) (1) において得られた画像から自己位置の同定に必要な稜線に対応する直線と頂点に対応するそれらの直線の交点を抽出し，画像内における交点の座標及び交点を作る直線上の任意の１点の座標を算出する。

(3) カメラの位置や向きに関係する適当な座標変換を用い，(2) で抽出した点の座標を，画

像平面内の座標系 (画像座標系) を用いた表現から，環境内に設置された座標系 (絶対座標系) を用いた表現に変換する。

(4) (3) で得られた絶対座標系で表現された，画像内の交点と任意の1点から，絶対座標系で表現された画像内の直線の方向を求め，交点の座標と頂点の座標，直線の方向と稜線の方向とを，それぞれ比較し，その差から移動ロボットの現在位置と向きをより正確に算出する。

環境内の実際の物体を撮影した画像 (実画像) は，コンピュータグラフィックスを用いて生成した理想的な画像とは異なり，光の反射，物体の影，対象物の汚れ，撮影時のノイズ等を含むために，上記の過程において，実画像から自己位置の同定に必要な稜線及びそれらが作る頂点を選び出し，それらを高い精度で抽出することは困難なことである。そこで，画像から特徴点を抽出することを目標として，今回は，特に線画の生成と画像領域の分割の2段階の過程を経て，実画像から稜線に対応する線分を，できる限り高い精度で抽出するソフトウェアの開発を試みた。次節において，カラーの実画像から抽出されたエッジを細線化した画像 (細線化画像) を生成する方法について記述する。なお，本研究では，原研研究炉 JRR-3 の地下一階一次区画を模擬して製作した模擬環境内 [2] の機器の台座を，三板カラー CCD カメラを用いて撮影した画像を処理対象の画像 (Fig. 3.8) として用いた。この画像は，8 bits の輝度値レベル幅を持つ，大きさ 514×384 の画像である。

3.3.1 細線化画像の生成

一般的に，カラー画像から細線化画像を生成する過程は，Fig. 3.9 (a) に示すを方法が用いられている。

しかし，この方法では，初めにカラー画像を白黒画像に変換してしまうため，有用な色情報を用いていないこと，二値化を全画像において行なうため，他のエッジより相対的に輝度値が低いエッジが消えてしまうことなど，いくつかの点で鮮明な線画を得ることができない。また，さらに後の処理において望まれる鮮明な細線化画像を得るためには，ノイズ等によって生じてしまう不必要なエッジや線分を抽出しないようすることも重要である。そこで，これらのことを考慮し，一般的な細線化画像の生成方法をいくつか改良した。

(1) 直接，カラー画像からエッジ抽出を行う

一般に，カラー画像において，エッジとして検出したい境界線が生じるために，次の二通りの原因が考えられる。1 つの原因は，境界線を作る二つの面が同色であるが，表面の明るさの違いによって境界線を作っている場合であり，他方は，それらの面が違う色を持つ場合である。カラー画像を濃淡画像に変換した場合，前者の境界線は輝度値の差として現れ，濃淡画像においても明瞭な境界線として見ることができ，よって，濃淡画像からもその境界線をエッジとして検出することができる。しかし，後者の場合，表面の明るさも違っていれば，前者と同様に，濃淡画像でも境界線が輝度値の差として現れるが，明るさが同程度の時は，濃淡画像に変換することによって，色の違いによる境界線が，明瞭な境界線として見ることができず，カラー画像において，明らかな境界線であった部分をエッジとして検出できなくなっ

てしまうことがある。

そこで，R，G，B の成分の画像，それぞれに直接 sobel オペレータ [3] を作用させることとした。ここで用いた sobel オペレータは，一般的な 3 × 3 のテンプレートである。テンプレートのオペレータを画像に作用させるとは，テンプレートを画像に重ね，各マスに重なる画素の輝度値とそのマスの値を掛け合わせ，さらにすべてを足し合わせることによって，中心画素のエッジ強度を求めることである。よって，この操作を R，G，B，それぞれの成分の画像に対して行なうため，1 つの画素に対して，3 つのエッジ強度を得ることとなる。そこで，それらの強度を足し合わせることによって，得ようとするエッジ画像のエッジ強度とし，このときエッジ強度が 255 より大きくなった場合，その画素のエッジ強度を 255 として，エッジ画像を得た。

Fig. 3.10，3.11 に，濃淡画像からエッジ抽出を行なった場合のエッジ画像と，カラー画像から直接，エッジ抽出を行なった場合のエッジ画像を示す。濃淡画像からのエッジ画像は，明るさの違いによる境界線に対応するエッジを鮮明に抽出しているが，カラー画像において黄色の面と緑色の面が作る境界線に対するエッジが，他のエッジに較べて低い輝度で得られている。また，明るさが同程度の同色の面が作る部分の境界線も，他のエッジに較べて低い輝度のエッジとして得られている。これに対して，カラー画像からのエッジ画像は，前者と比較して画像にノイズが多く含まれるが，前者のエッジ画像において，低い輝度となっていたエッジが，他のエッジと同程度の輝度で得られていることが分かる。よって，カラー三成分の画像それぞれに，sobel オペレータを作用させることにより，明るさが同程度である同色の面が作る境界線や，明るさが同程度であり，色が違う面よって作られる境界線をエッジとして抽出することが可能となった。

(2) 画像を格子に区切り，各格子内において二値化を行なう

画像全体の輝度値に基づいて二値化を行なった場合，画像全体において，相対的に輝度値が大きい画素が 1 となり，そうでない画素は 0 となってしまうことがある。そのため，エッジ画像内のエッジとして抽出された境界線のうち，画像全体を通じて，輝度値が大きいエッジのみが残り，その他のエッジは消えてしまうことになる。この現象を見るために，従来の方法でエッジを検出した画像全体に対して，二値化処理を行なった画像を Fig. 3.12 に示す。この図から分かるように，低い輝度のエッジは二値化において消えてしまっている。しかし，この現象は，エッジを抽出する時に，出来るだけエッジを強調し，またどの部分のエッジも同程度の輝度値を持つようにすることによって，ある程度減らすことが可能である。実際，カラー画像から直接エッジを抽出した画像全体に対して二値化処理を行なった場合 (Fig. 3.13)，上記の現象を見ることはできない。しかし，エッジ検出だけで，この現象を完全になくすことがはできないため，本報告書における線画生成方法が，できる限り多くの画像に対して適応可能である汎用的なものとするために，以下に記述する処理を行なうこととした。

上記の現象は，画像全体の画素の輝度値を比較しているために生じている。つまり，輝度値が低いエッジは，そのエッジ付近の領域内ではエッジとして見ることができるが，画像全体では輝度値が比較的小さいものとみなされ，二値化処理によって消されてしまう。よって，

二値化処理のの対象領域を小さくすることによって，この現象を抑えることができると考えられる。そこで，画像をある大きさの矩形 (今回の場合は，64 × 64 の正方形) に区切り，各々の矩形内において，二値化処理を行なうこととした。このとき用いた二値化のアルゴリズムは，各矩形内における輝度値ヒストグラムを二分するしきい値を，大津の判別分析法 [4] を用いて求め，そのしきい値に基づいて二値化するものである。

Fig. 3.14，3.15 に，従来の方法でエッジ抽出を行なった画像と，カラー画像から直接エッジ抽出を行なった画像，それぞれに対して，格子に切った二値化処理を適応した結果の画像を示す。Fig. 3.14 を見ると，Fig. 3.12 において消えてしまっていたエッジが鮮明に得られていることが確認できる。しかし，Fig. 3.15 からを分かるように，この方法を用いると，Fig. 3.11 に現われた輝度値が低いノイズのみを含む正方形において二値化処理を行った場合，その正方形内においてノイズのみを強調してしまう結果となり，後の処理においてその影響が現われてしまう。そこで，これを回避するために次に節で述べる方法を用いた。

(3) 直接，エッジ画像の細線化を行なう

従来，細線化処理は，二値化した後に行われていた。しかし，そのような処理方法では，Fig. 3.14，3.12 に見られるような，二値化の段階で強調されてしまったノイズに対しても細線化を行なってしまい，その結果が後の線画にも残ってしまう。そこで，このようなノイズが後の処理に対して影響を与えないように，以下の処理を行なうこととした。まず，まだ濃淡画像 (輝度値レベルが，0 ～ 255 の範囲にある画像) である，カラー画像から抽出したエッジ画像に，直接，細線化処理を行なうことによって，撮影時のノイズや対象物の汚れのために，エッジ抽出処理で抽出され，前節で述べた二値化処理により強調されてしまうような孤立した濃淡エッジを点に収束させる。そして，点に収束したノイズを適当なノイズ除去方法によって除去する。この処理において，濃淡ノイズを点に収束させることによって，画像全体からノイズを識別することが容易になり，それらを除去することが容易となる。Fig. 3.16 に，エッジを示す濃淡画像に細線化 [5] を行なった画像を，Fig. 3.17 に，ノイズを点に収束させた画像を示す。さらに，この画像に上述の二値化処理を行ない，点に収束させたノイズを除去した画像を Fig. 3.18 に示す。

以上いくつかの処理を加えて改良した線画生成の過程を Fig.3.9 (b) に示す。また，この線画生成処理の過程に沿って生成される各々の画像と最終的の生成された線画を Fig. 3.19 に示す。

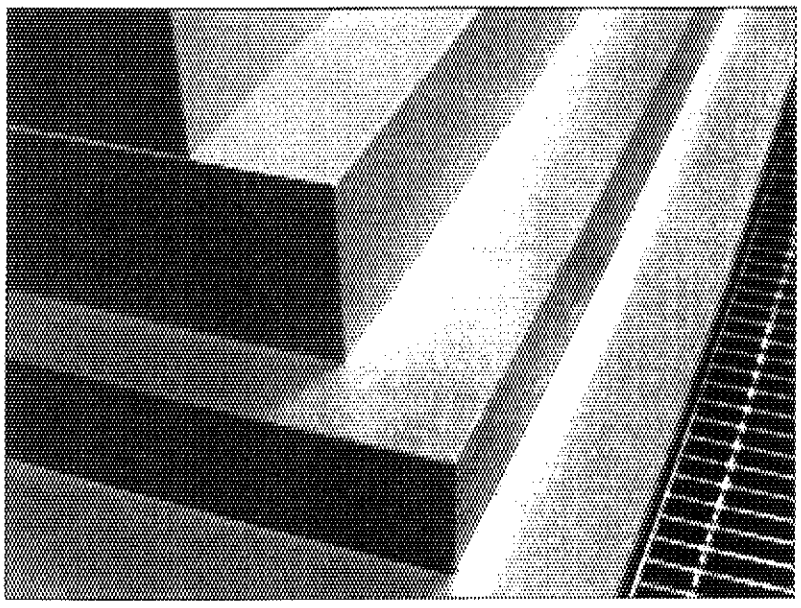
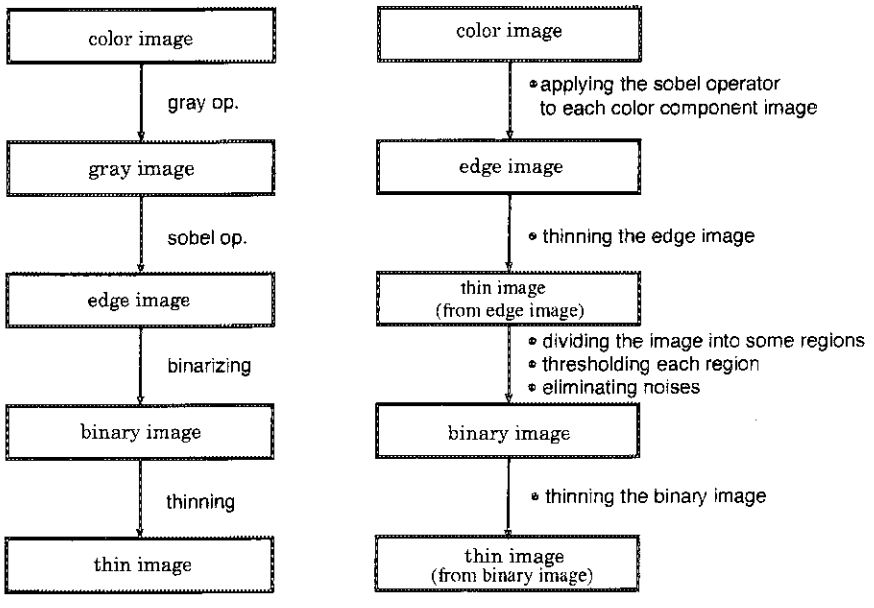


Fig. 3.8 The input image



(a) (b)

Fig. 3.9 The previous method (a) and the improved method (b) to obtain the thinned-edge image

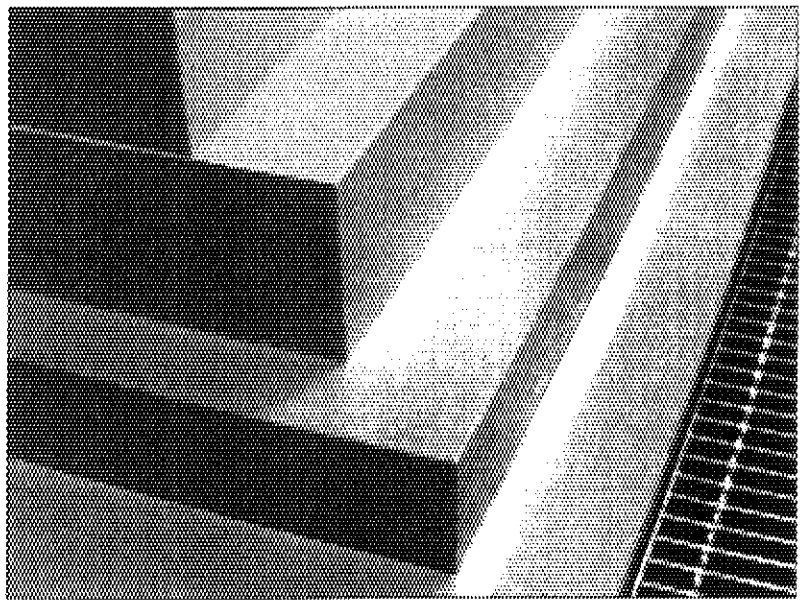
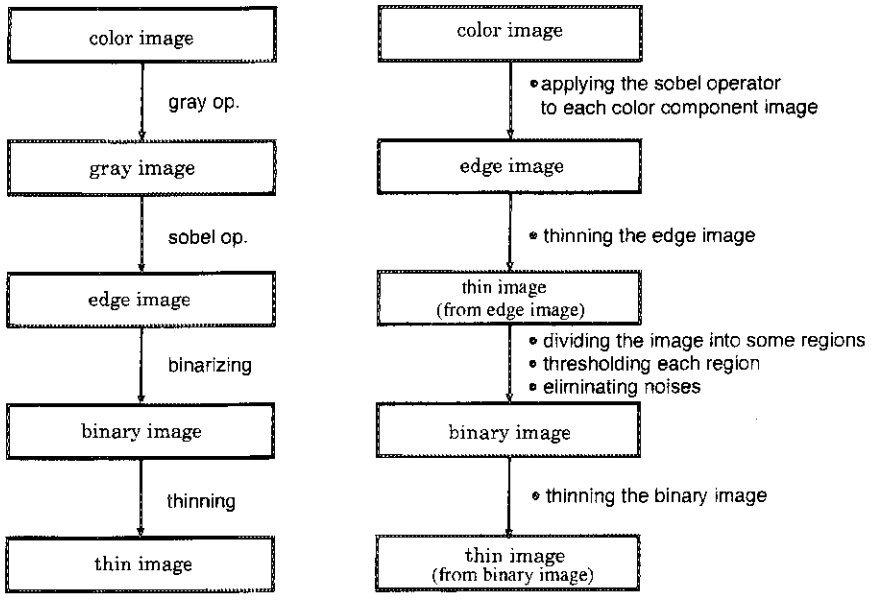


Fig. 3.8 The input image



(a) (b)

Fig. 3.9 The previous method (a) and the improved method (b) to obtain the thinned-edge image

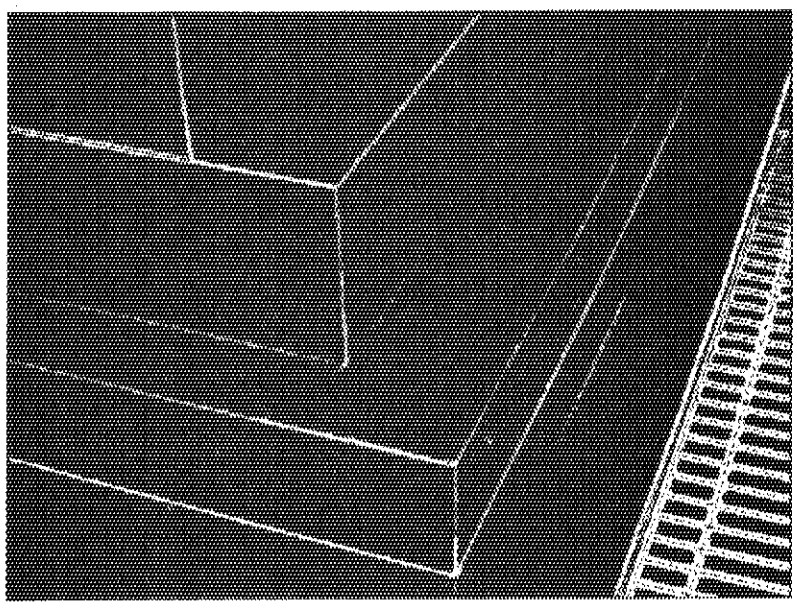


Fig. 3.10 The edge image extracted from the gray-level image

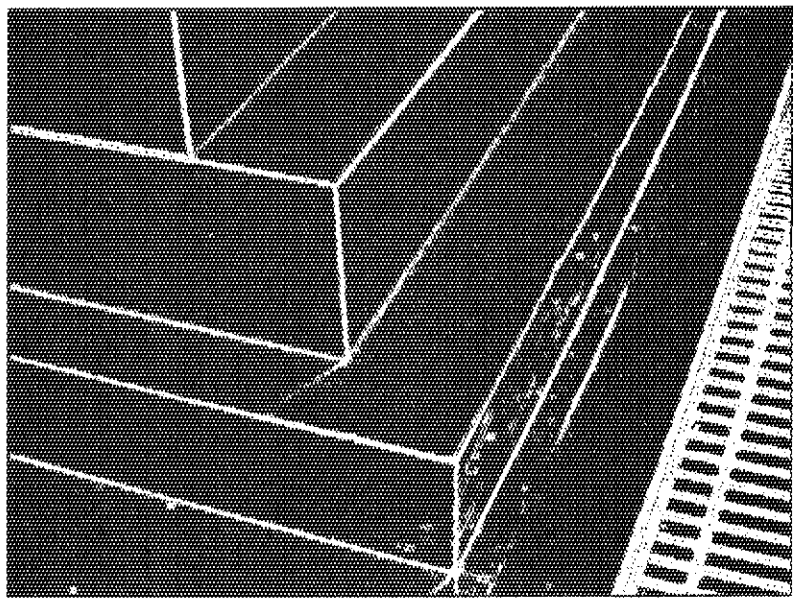


Fig. 3.11 The edge image extracted from the color image

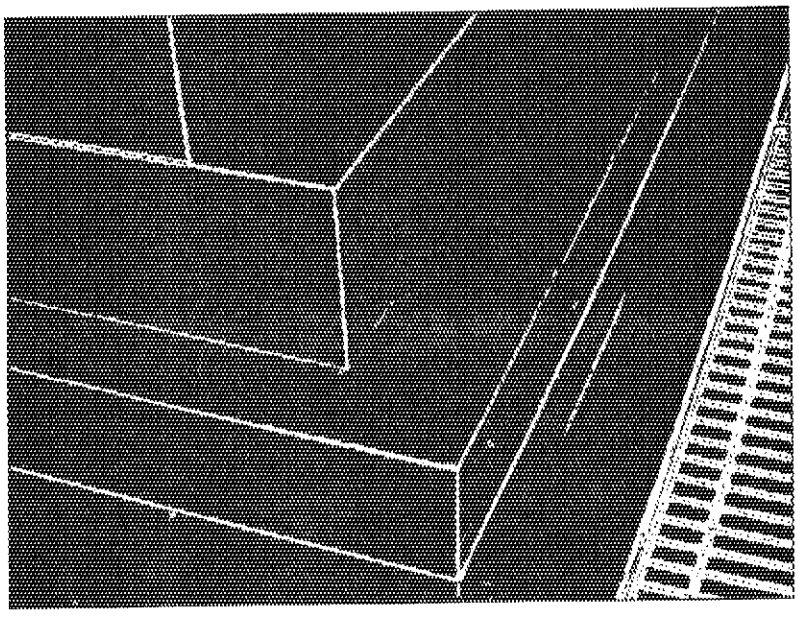


Fig. 3.12 The image obtained by thresholding the edge image extracted from the gray-level image using the previous method

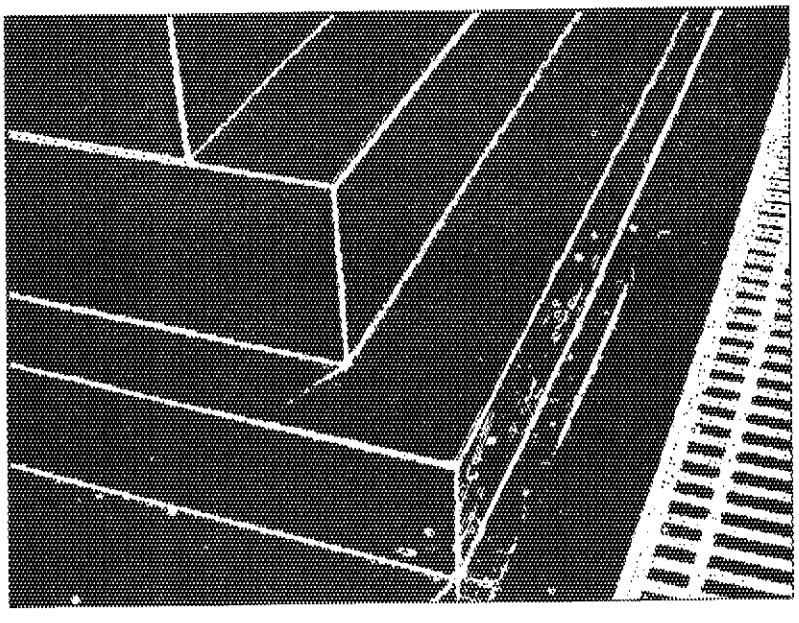


Fig. 3.13 The image obtained by thresholding the edge image extracted from the color image using the previous methods

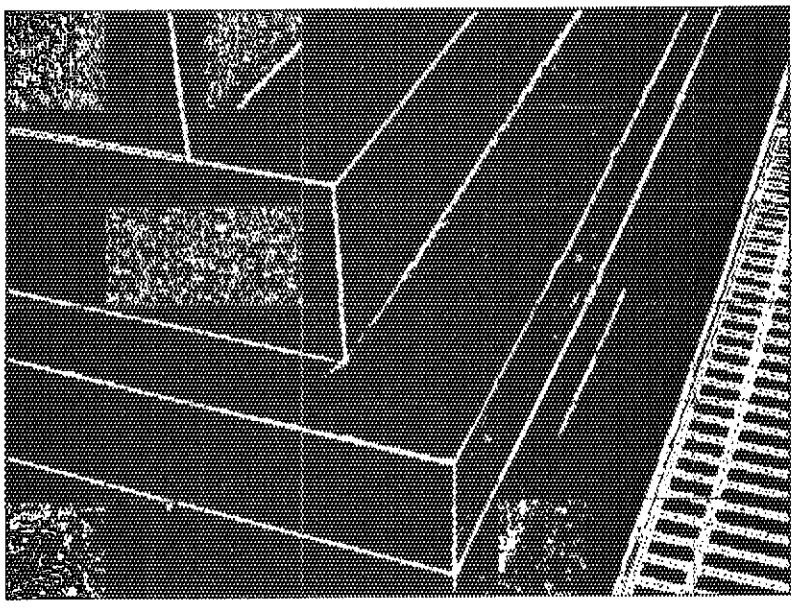


Fig. 3.14 The image obtained by thresholding the edge image extracted from the gray-level image using the improved methods

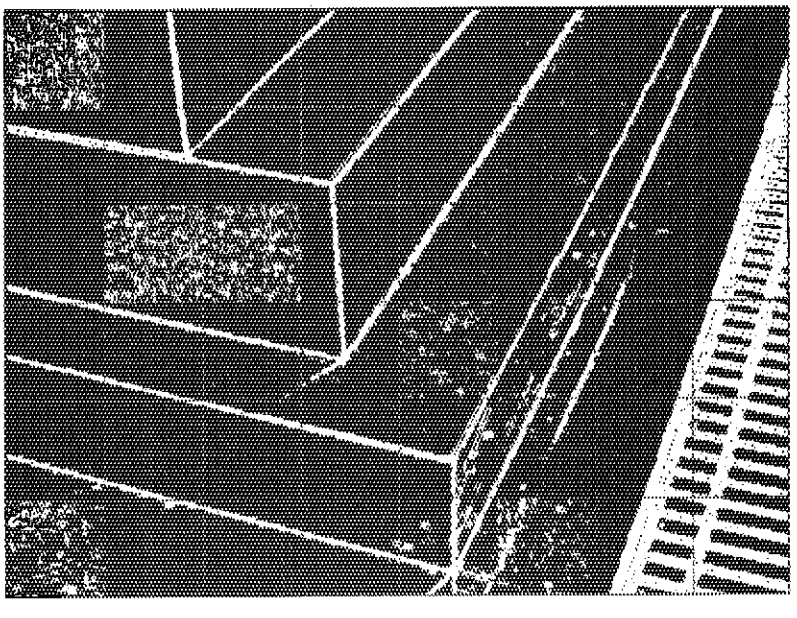


Fig. 3.15 The image obtained by thresholding the edge image extracted from the color image using the improved methods

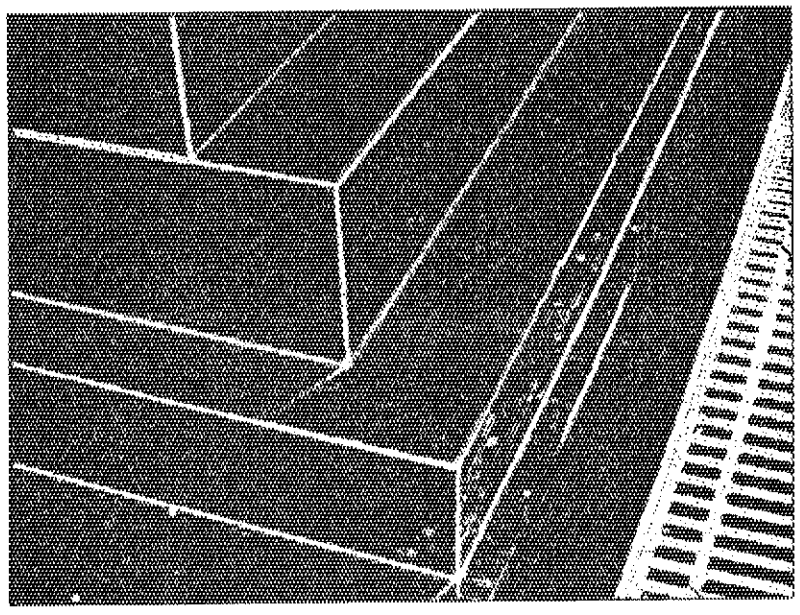


Fig. 3.16 The image obtained by shrinking the noise by thinning the edge image

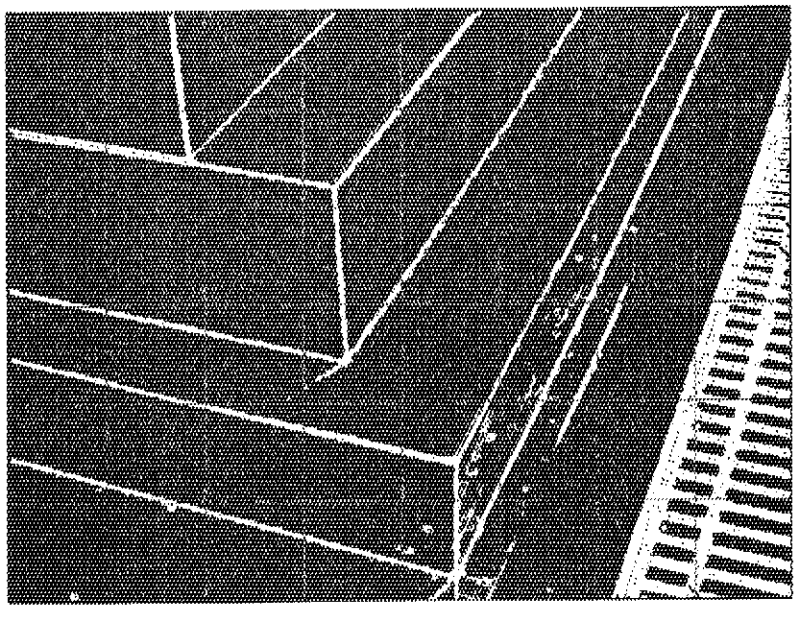


Fig. 3.17 The image obtained by thinning Fig. 3.16 using the improved method

3.3.2 領域抽出

前章において，いくつかの処理を加えることによって，必要であるが除去されてしまうような線分を残し，また，ノイズなどで生じる不要な線分を除去した線画を生成することができた。しかし，必要な稜線に対応する線分と頂点に対応する線分の交点を抽出することを目的とした場合，Fig. 3.19 から，まだ，画像内の線分数が多すぎることが分かる。そこで，色情報を用い，必要な稜線や頂点に対応する線分や交点を含む領域を抽出することを考える。

まず，ここで色について簡単に述べる [6]。厳密に言うと，色は，“色感覚による色” と “色知覚による色” に区別される。色感覚による色は，心理量としての色とそれを感じさせる物理量としての光の分光特性との対応関係を測定することによって，心理物理量として定量的に規定するものである。また，色知覚による色は，心理量として人間の知覚した色を記号，色表等を用いて定性的に規定するものである。さらに，それぞれに対して色を表示するための表色系と呼ばれる体系があり，前者を表示する体系として混色系が，後者を表示する体系として顕色系がある。混色系の代表的なものは，CIE(国際照明委員会) 表色系であり，この中に RGB 表色系が含まれる。また，顕色系の代表的なものは，色の色相，明度，彩度を扱うマンセル表色系である。混色系は，色を定量的，数学的に扱うために便利であるが，表示された色が感覚的にどのように見えるかを評価するには，顕色系が便利である。

通常，色の違いにより画像から特定の領域を抽出する場合，その抽出方法は，人間の感覚的な色の見え方に依存するはずである。そこで，本研究では，人間の感覚的な色の見え方，つまり表色系として顕色系のマンセル表色系を用いて領域抽出を行うことを試みた。マンセル表色系は，色の心理量を色相 (hue)，彩度 (saturation)，明度 (intensity) の 3 属性で定義している。特に色相は，色の違いを区別するための属性であることから，色相を用い，色の感覚的な見え方の違いによる領域抽出が可能となると考えられる。色相は，Fig. 3.20 のような，色相環と呼ばれる 1 つの円の円周上に配置されている。色相の配置方法は，まず，R(赤)，Y(黄)，G(緑)，B(青)，P(紫) の色を基本色相とし，それらが円を 5 等分するように配列され，次にその 5 つの基本色相の中間に，YR，GY，BG，PB，RP となる色が配置される。さらに，隣合った色相の間を 10 等分し，全体が 100 色相で構成されている。そこで，色の違いを用いた画像領域の抽出は，まず，画像の各画素の R，G，B 三色の輝度値をマンセル表色系の色相 (hue)，彩度 (saturation)，明度 (intensity) に変換し，そのあと，色相が，色相環のある範囲に含まれる画像領域を抽出することによって行った (RGB 表色系からマンセル表色系 (HSI 表色系) への変換方法は [6] を参照。)。

特に本研究で用いた，Fig. 3.8 に示す模擬環境の機器の台座付近の画像において，抽出したい台座の色は，感覚的に黄色に見えることから，色相環において YELLOW を含む範囲に含まれる色相値を持つ画素を抽出することにより，台座部分の画像領域が抽出可能となると考えられる。但し，画像には，感覚的に緑色に見える部分も含まれているので，YELLOW を含む領域から GREEN を含む領域は除外することが望ましいと思われる。このことを調査するために，色相環を 20 等分し，各々の領域に含まれる色相値を持つ画素のみを表示した画像を Fig. 3.21 に示す。これらの画像から，色相環の 5YR ～ 5Y の範囲内の色相値を持つ画素のみを含む領域を抽出することによって，望む黄色の台座部分の領域を画像から抽出で

きることが分かる。この方法によって抽出された領域の画像を Fig. 3.22 に示す。この結果により，前章で得られた線画から台座部分を構成する線を抽出することができる。その結果の画像を Fig. 3.23 に示す。

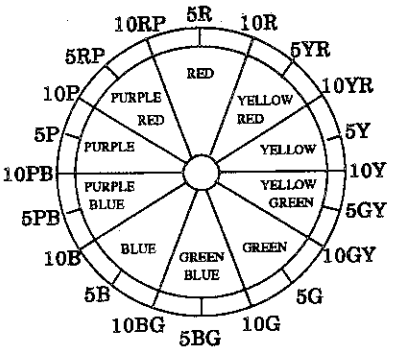


Fig. 3.20 The hue circle

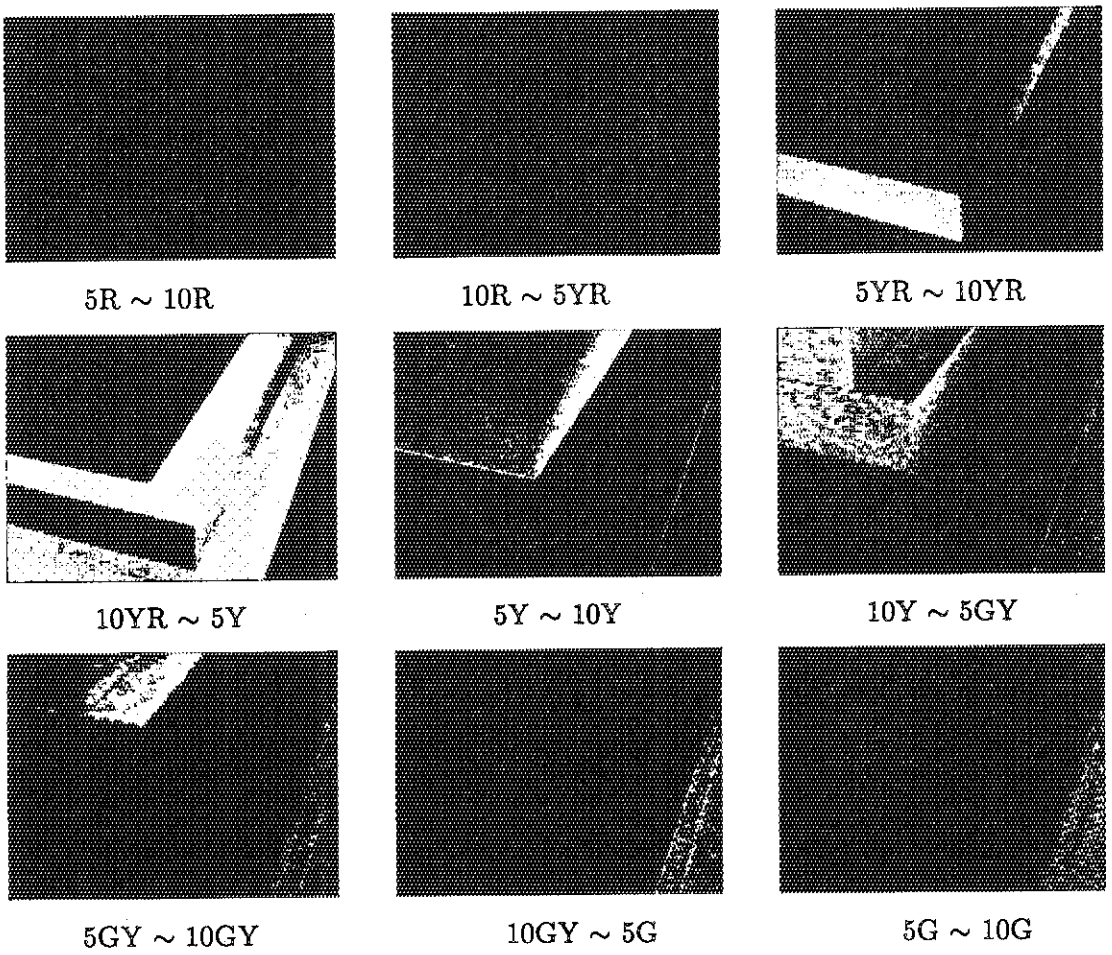


Fig. 3.21 Images obtained by extracting the region based on the hue value of pixels

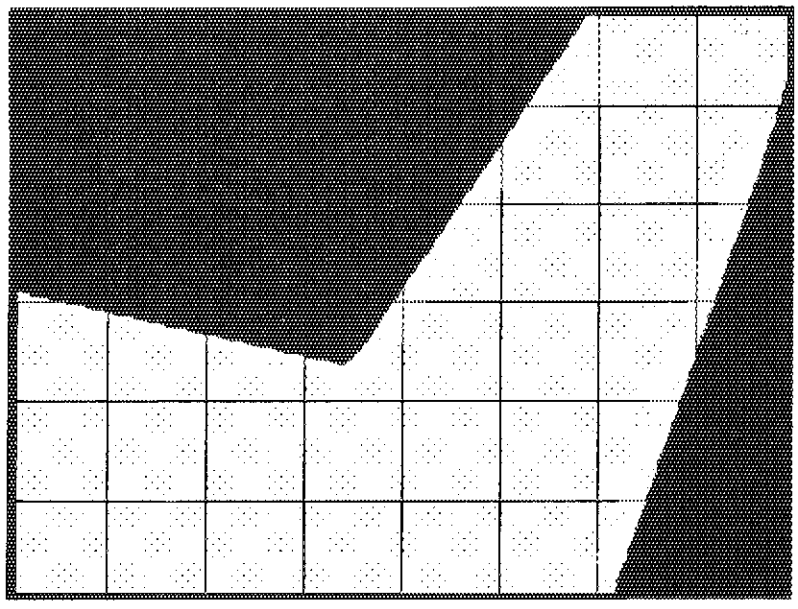


Fig. 3.22 The image of the yellow region

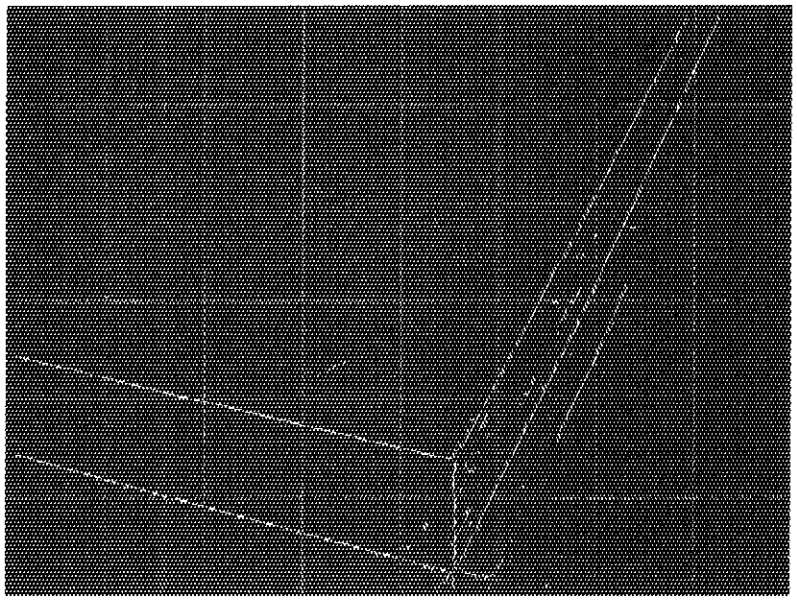


Fig. 3.23 The image obtained by extracting some necessary thinned-edges

3.4 まとめ

センシングシステムについては，今年度製作が完了したので，今後，旋回台や搭載されるCCD カメラなどの性能特性，検出されるセンシングデータの特徴等を調査していく必要がある。

また，環境認識技術の研究の一環として，環境内の機器の特徴点を抽出することによって，移動ロボットの位置を同定するシステムを開発することを目的とし，CCD カメラによって撮影されたカラー画像から，必要とされる領域の線分を抽出するプログラムの作成をおこなった。Fig. 3.24 に，抽出された領域に，抽出された稜線 (空色で示す) を重ねた画像を示す。この図から，作成したプログラムによって，かなりのよく必要とされる線分が抽出されていることがわかる。しかし，プログラム中の領域抽出の部分において，抽出領域の色相の範囲の決定は，使用するカラー画像に依存するものであるため，汎用的な領域抽出方法ではない。そのため，プログラムに汎用性を持たせ，さまざまな画像を入力画像として使用できるようにするためには，入力される画像により，抽出領域の範囲を決定するようなアルゴリズムが必要となる。また，今回は，三板のカラー CCD カメラによって撮影した画像を使用して，プログラムの動作を試したが，上述のセンシングシステムに搭載されている単板のカラー CCD カメラによって撮影された画像に対しても適用可能であるかを調べる必要がある。

さらに，移動ロボットの位置や方向を算出するために必要な特徴点として，画像座標系において，台座の頂点の座標を求める必要がある。しかし，細線化画像では，単に必要とされる線分が抽出されただけであり，直接，細線化画像から，その線分に当てはまる直線の式や直線が作る交点の座標を求めることはできない。よって，幾つかの必要な直線を線分にあてはめ，それらの直線の交点の座標を抽出するアルゴリズムを開発する必要がある。

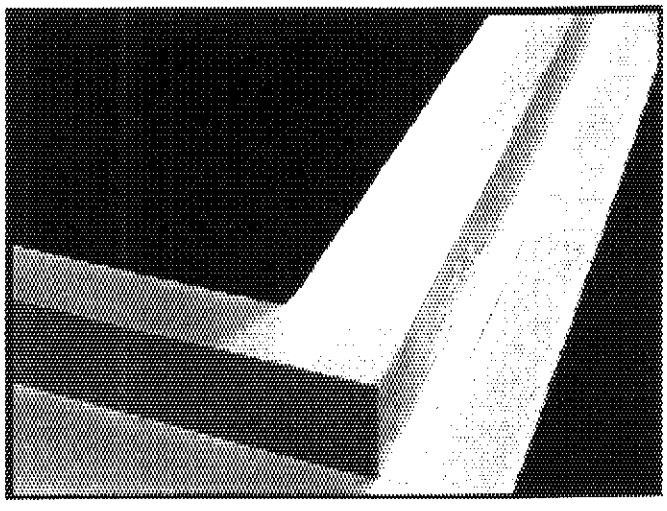


Fig. 3.24 The image obtained by superimposing the thinned-edge image on the yellow region image

参考文献

[1] 海老原　健一, “知能ロボットによる原子力施設内巡回点検用センシングシステムの概念設計” , JAERI-M 93-225(日本原子力研究所, 1993,11).

[2] 秋元　正幸 他, “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) — 平成 4 年度作業報告 — ”, JAERI-M 94-051(日本原子力研究所, 1994,5).

[3] 大谷　孝之, “ディジタル画像処理とその人間動作シミュレーション技術の研究 (HASP) への応用”, 私信 (1990).

[4] 大津　展之, “判別及び最小2乗規準に基づく自動しきい値選定法”, 信学論 (D), Vol. J63-D, pp.349 –356, (1980).

[5] 横井, 鳥脇, 福村, “濃淡図形における特徴点の抽出と細線化の一方法”, 信学論 (D), Vol. 58-D, No.10, pp. 601–608, (1975).

[6] 高木　幹雄, 下田　陽久 監修, “画像解析ハンドブック” (東京大学出版会,1991)

4. 2足歩行シミュレーション

4.1 はじめに

2足歩行ロボットに関する研究では，プラント内巡回点検用人間型知能ロボットの移動機構である2足歩行に関して，動力学的シミュレーションにより，ロボット設計の基盤技術開発を行っている。2足歩行をハードウェアで実現する前に，数値シミュレーションによってロボット設計パラメータの考察を行い，歩行安定性を十分に検討することがここでの目的である。この結果を基にプロトタイプ・モデルの試作，シミュレーション・モデルの妥当性・制御性の検証，ハードウェア上の問題点の検討，及び知能化の検討を行い，命令理解や視覚認識等，他の要素技術を統合した人間型知能ロボットの設計を行う[1]。

これまでに我々が取り上げてきた手法に代表されるような，モデル・ベーストな従来の2足歩行ロボットの歩行制御法は，事前のシミュレーションによって安定歩行パターンを解析し，それをプログラム制御することでロボットを歩行させるものである[2, 3]。この方法では，ロボットが歩行中に自身の状態（状況）を的確に把握し，それに基づいて歩行安定化制御をしているのではなく，ロボットと環境との相互作用が予め分かっているとして，歩行計画（パターン）を作成し，そのパターン通りに歩行を行わせている。そのため，予期できない外乱等に対処することは難しい。

そこで，我々は，歩行パターンの多様化・高度化には，これまでのようなプログラム制御ではなく，ロボットの状態を検出しこれを基に安定化を行うリアルタイム制御法が必要であると考え，平成4年度に新たにリアルタイム2足歩行制御法を提案した[4]。4年度は，この制御法を外乱なしの直進完全歩行に適用し，平地及び階段歩行のシミュレーションにより，この制御法の妥当性を検証した。本年度は，この制御法を未知の外乱がロボットに作用する環境下での歩行に適用し，有効性について検証する。

本章では4.2節で歩行ロボット・モデルと運動方程式，及び歩行安定性の規範であるZMPについて，4.3節で本制御法の概略についてまとめる。また，4.4節では外乱の作用する環境下において，本制御法を適用した歩行シミュレーションの結果と考察について述べる。

4.2 ロボット・モデル

研究の対象とする2足歩行ロボット・モデルをFig.4.1に示す。このモデルは上肢なし人間形モデルで，上体おもり，アクチュエータ等を質点とし，リンク構造部材を剛体とする準質点系モデルである[3, 4]。

一般にロボットをモデル化する場合，構造部材の考慮から剛体系での記述が普通であるが，質量がボディの一部に集中するような場合（例えば，アクチュエータの質量がリンクのそれに比較して非常に大きいような場合）は質点系での記述も可能である。本システムでは剛体

系・質点系のどちらでも記述が可能で，シミュレーション段階ではその大きさや慣性モーメントが不明確であるアクチュエータやセンシング・システム等を質点とし，構造部材を剛体としており，これらの混在するモデルを許している。また，本2足歩行ロボット・モデルは8ボディ，7関節からなり，下肢の前頭面内の動きは両脚（legs）が平行で，足（foot）は常に床面と水平であると仮定し，下肢5自由度（ピッチ軸4自由度，ロール軸1自由度），腰2自由度（ピッチ軸・ロール軸それぞれ1自由度）の合計7自由度を持つ（Fig.4.2)。但し，現段階では直進歩行を前提とし方向転換の自由度は与えていない。

一般に剛体系の運動方程式は式 (4.1) で表せる [5]。

$$\{\boldsymbol{M}\} = [\boldsymbol{A}]\{\ddot{\boldsymbol{\Xi}}\} + [\boldsymbol{B}]\{\dot{\boldsymbol{\Xi}}^2\} + [\boldsymbol{C}]\{\dot{\boldsymbol{\Xi}}\dot{\boldsymbol{\Xi}}\} + \{\boldsymbol{G}\}, \tag{4.1}$$

ここで，

$$\begin{aligned}
\{\boldsymbol{M}\} &= \{\boldsymbol{M}_J, M_{ZMP_x}, M_{ZMP_y}, M_{ZMP_z}\}^T \\
&\quad : \text{各ジョイントとZMPにおけるモーメント}, \\
\{\boldsymbol{\Xi}\} &= \{\alpha, \beta_{1R}, \beta_{1L}, \beta_{2R}, \beta_{2L}, \theta, \psi\}^T \\
&\quad : \text{各ジョイントの角度}, \\
[\boldsymbol{A}] &= [f_A(\alpha, \beta_{1R}, \beta_{1L}, \beta_{2R}, \beta_{2L}, \theta, \psi, XZMP, YZMP)], \\
[\boldsymbol{B}] &= [f_B(\alpha, \beta_{1R}, \beta_{1L}, \beta_{2R}, \beta_{2L}, \theta, \psi, XZMP, YZMP)], \\
[\boldsymbol{C}] &= [f_C(\alpha, \beta_{1R}, \beta_{1L}, \beta_{2R}, \beta_{2L}, \theta, \psi, XZMP, YZMP)], \\
\{\boldsymbol{G}\} &= \{f_G(\alpha, \beta_{1R}, \beta_{1L}, \beta_{2R}, \beta_{2L}, \theta, \psi, XZMP, YZMP)\}, \\
ZMP &= (XZMP, YZMP, 0).
\end{aligned}$$

右辺第1項は慣性項，第2項は遠心力項，第3項はコリオリ力項，そして第4項は重力項である。

我々は歩行安定性の規範として Zero Moment Point（ZMP）規範を用いている。ZMPとは，ロボットに作用するモーメントの総和のピッチ軸及びロール軸まわりのモーメントが共に零になる床面上の点である。このZMPがロボットの足底接地面の形成する多角形の領域（安定領域）内に存在していれば，ロボットは安定に歩行できる [6]。

本研究ではこのZMP規範を採用しているため，運動方程式に対する拘束条件がZMPにおけるモーメントのX・Y成分，すなわち M_{ZMP_x}, M_{ZMP_y} が0であるという2つである。そのため，一部の運動を与え，未知の動作を与えられた運動に対する補償動作として計算する。ここでは，任意の歩容に対する補償動作を上体で行う構成とし，

$$\begin{aligned}
\boldsymbol{M}_0 &= \{M_{ZMP_x}, M_{ZMP_y}\}^T = \{0,0\}^T &&: \text{既知のモーメント}, \\
\boldsymbol{M}_U &= \{\boldsymbol{M}_J, M_{ZMP_z}\}^T &&: \text{未知のモーメント}, \\
\boldsymbol{\Xi}_0 &= \{\alpha, \beta_{1R}, \beta_{1L}, \beta_{2R}, \beta_{2L}\}^T &&: \text{既知の運動（歩容）}, \\
\boldsymbol{\Xi}_U &= \{\theta, \psi\}^T &&: \text{未知の運動（上体動作）},
\end{aligned}$$

とすると，

$$\left\{ \begin{array}{c} \ddot{\theta} \\ \ddot{\psi} \end{array} \right\} = [\boldsymbol{A}_{0U}]^{-1}\left(\left\{ \begin{array}{c} 0 \\ 0 \end{array} \right\} - [\boldsymbol{A}_{00}]\left\{ \begin{array}{c} \ddot{\alpha} \\ \ddot{\beta}_{1R} \\ \ddot{\beta}_{1L} \\ \ddot{\beta}_{2R} \\ \ddot{\beta}_{2L} \end{array} \right\} - [\boldsymbol{P}_0]\left([\boldsymbol{B}]\{\dot{\boldsymbol{\Xi}}^2\} + [\boldsymbol{C}]\{\dot{\boldsymbol{\Xi}}\dot{\boldsymbol{\Xi}}\} + \{\boldsymbol{G}\}\right)\right), \quad (4.2)$$

$$\{\boldsymbol{M}_U\} = [\boldsymbol{A}_{UU}]\left\{ \begin{array}{c} \ddot{\theta} \\ \ddot{\psi} \end{array} \right\} + [\boldsymbol{A}_{U0}]\left\{ \begin{array}{c} \ddot{\alpha} \\ \ddot{\beta}_{1R} \\ \ddot{\beta}_{1L} \\ \ddot{\beta}_{2R} \\ \ddot{\beta}_{2L} \end{array} \right\} + [\boldsymbol{P}_U]\left([\boldsymbol{B}]\{\dot{\boldsymbol{\Xi}}^2\} + [\boldsymbol{C}]\{\dot{\boldsymbol{\Xi}}\dot{\boldsymbol{\Xi}}\} + \{\boldsymbol{G}\}\right), \quad (4.3)$$

ここで，

$$\left[\begin{array}{c} \boldsymbol{P}_0 \\ \hline \boldsymbol{P}_U \end{array} \right]\{\boldsymbol{M}\} = \left\{ \begin{array}{c} \boldsymbol{M}_0 \\ \hline \boldsymbol{M}_U \end{array} \right\},$$

$$\left[\begin{array}{c|c} \boldsymbol{R}_0 & \boldsymbol{R}_U \end{array} \right]\left\{ \begin{array}{c} \boldsymbol{\Xi}_0 \\ \hline \boldsymbol{\Xi}_U \end{array} \right\} = \{\boldsymbol{\Xi}\},$$

$$\left[\begin{array}{c} \boldsymbol{P}_0 \\ \hline \boldsymbol{P}_U \end{array} \right]\{\boldsymbol{A}\}\left[\begin{array}{c|c} \boldsymbol{R}_0 & \boldsymbol{R}_U \end{array} \right] = \left[\begin{array}{c|c} \boldsymbol{A}_{00} & \boldsymbol{A}_{0U} \\ \hline \boldsymbol{A}_{U0} & \boldsymbol{A}_{UU} \end{array} \right].$$

従来の制御法においては，通常上記の常微分方程式は，定常歩行等を前提とした周期解という条件の下に（初期値問題），数値計算によって解かれる [2, 3, 7]。本制御法での解法については次節で概説する。

4.3 2足歩行ロボットの制御法

本制御法 [4] では，歩行ロボットに任意の歩容と安定領域内に設定したＺＭＰ（基準ＺＭＰ）を与え，この歩容に従って下肢の動作を行わせる。上体に関しては，歩行中に実際のＺＭＰを計測（測定ＺＭＰ）することで，上体の補償動作をリアルタイムで生成し，歩行安定化を行う。

本制御法の概念を Fig.4.3に示す。図には，定常歩行時の１歩についてロボットの歩行状況を示す代表的なパラメータの変遷を描いている。１歩の歩行時間 T を n 分割し，$\Delta t\ (= T/n)$ を制御周期として，各点を $t_0, t_1, \ldots, t_n$ とおく。また，現在の状態を t_{i-1} とする。まず，t_{i-1} において計測されたＺＭＰをもとに次のステップ t_i のＺＭＰを予測する。この予測されたＺＭＰ（予測ＺＭＰ）と基準ＺＭＰとの差があるしきい値 δ を越える場合，t_i ステップの実際のＺＭＰがしきい値内に入るように，上体動作を変更する。具体的には，t_i ステップにおいて，与えられた歩容（角度・角速度・角加速度），上体動作の角度・角速度，及び実際に用いるＺＭＰ（基準ＺＭＰとの差がしきい値内に入るように設定したもの）から，式 (4.2) を用いて上体の補償動作の角加速度を算出し，上体動作を修正する。

Fig.4.4に本制御法のフローチャートを，Table 4.1 に歩行中のロボットの動作の生成についてまとめる。以下に各計算法の概略を示す。

(a) 歩容

はじめに定常歩行部の１歩を作成し，これに接続する境界条件を満足する過渡歩行部（発進・停止部）を作成する。歩容の生成は，まず遊脚の通る基本点列を与え（ここでは便宜的に最大高さを与える），これをスプライン関数によって補間し遊脚の軌跡を作成する。腰の高さは一定とし定常歩行時の腰の速度は等速として，逆運動学計算によって下肢の各関節の動作を決定する。過渡歩行部は発進・停止それぞれ１歩とし，発進の下肢動作を行う前及び停止の下肢動作が終了した後に，下肢は運動しないが上体だけ運動を行う歩容静止期を設ける [7]。また，定常歩行部を反復することで任意の歩数を達成する。

(b) ＺＭＰ

Fig.4.3で示したように，ここでは４種類のＺＭＰを定義する。基準ＺＭＰ，測定ＺＭＰ，予測ＺＭＰ，及び上体の補償動作生成に実際に用いるＺＭＰである。

基準ＺＭＰ軌跡は，Fig.4.5に示すように，足の接地位置から決まる安定領域の中央を連続的に移動する軌跡である。なお，このＺＭＰは歩行中の実際のＺＭＰと比較するための単なる基準である。

測定ＺＭＰ，すなわち歩行中の実際のＺＭＰについては，ロボットの運動状態を示す関節角のセンサ情報からＺＭＰ方程式 [6] を用いて算出することができる。しかし，外乱等が作用した場合，これらを全てセンシングする必要があり，ロボット全体にセンサを張り巡らすことになる。これでは，事実上全ての外乱に対処したＺＭＰ算出は不可能である。そこで本研究では，これら全ての情報を含んでいる床反力に着目し，これを実測することによりＺＭＰの算出を行う（但し，シミュレーションでは便宜上関節角情報からＺＭＰ方程式を用いて算出する）。ＺＭＰは床反力の合成力の作用点であることから [5]，足底に複数の力センサを設置し，その床反力の鉛直方向成分からＺＭＰを測定することが可能である。

次のステップの上体補償動作を生成するためには，次のステップでのＺＭＰの予測が必要である。ＺＭＰの予測に関しては，ＺＭＰの移動速度を用いる方法が考えられるが，直接ＺＭＰの移動速度を検出することは困難であり，数値計算に頼らざるを得ない。ＺＭＰの移動の時間微分値を求めることは，シミュレーションにおいては角加速度の微分値が必要になるなど計算時間がかなり増大する。しかし，本制御法では簡易で計算時間のあまりかからない手法が望ましい。そこで，微分値近似＋オイラー法による推定を用いる [8]。これは，式 (4.4) のように微分値を１ステップ前の値で近似し，オイラー法によって推定する方法である。

$$\begin{aligned} ZMP_p(t_i) &= ZMP_m(t_{i-1}) + DZMP_m(t_{i-1})\Delta t\,, \\ DZMP_m(t_{i-1}) &\approx \frac{ZMP_m(t_{i-1}) - ZMP_m(t_{i-2})}{\Delta t}\,, \end{aligned} \tag{4.4}$$

ここで ZMP_p はＺＭＰの予測値，ZMP_m はＺＭＰの測定値である。

上体の補償動作を生成するときに実際に用いるＺＭＰは予測ＺＭＰ（ZMP_p）と基準ＺＭ

P（ZMP_s）から以下のように決定する。

$$ZMP_a(t_i) = ZMP_p(t_i) - \rho(ZMP_p(t_i) - ZMP_s(t_i)) , \tag{4.5}$$

ここで，

ρ ：重み係数.

(c) 上体補償動作

今回提案した制御法では，文献 [3, 7] で示したような多くの計算時間を要する収束計算を行わない。即ち，与えられた歩容，現在の上体動作の角度・角速度及び実際に用いるＺＭＰから，式 (4.2) により次の制御周期における上体動作の角加速度を一意に決定する。

次の制御周期における上体動作の角加速度が決定された後の上体動作の振る舞いは，上体補償動作の各加速度のその後の時間変化に依存する。ここでは，種々の数値シミュレーション実験結果から，この上体補償動作の変遷を以下に述べるように多項式を用いて作成することとした。

i) $0 \le t \le t_c$,

$$\begin{aligned}
\theta(t) &= a_1(t-tc)^{n_c} + a_2(t-t_c)^{n_c-1} + a_3(t-t_c)^{n_c-2} , \\
\dot{\theta}(t) &= a_1 n_c(t-tc)^{n_c-1} + a_2(n_c-1)(t-t_c)^{n_c-2} + a_3(n_c-2)(t-t_c)^{n_c-3} , \\
\ddot{\theta}(t) &= a_1 n_c(n_c-1)(t-tc)^{n_c-2} + a_2(n_c-1)(n_c-2)(t-t_c)^{n_c-3} \\
&\quad + a_3(n_c-2)(n_c-3)(t-t_c)^{n_c-4} ,
\end{aligned}$$

$$\begin{aligned}
a_1 &= \frac{t_c^2\ddot{\theta}_0 + 2t_c\dot{\theta}_0(n_c-2) + \theta_0(n_c^2-3n_c+2)}{2(-t_c)^{n_c}} , \\
a_2 &= \frac{t_c^3\ddot{\theta}_0 + t_c^2\dot{\theta}_0(2n_c-3) + \theta_0 t_c n_c(n_c-2)}{(-t_c)^{n_c}} , \\
a_3 &= \frac{t_c^4\ddot{\theta}_0 + 2t_c^3\dot{\theta}_0(n_c-1) + \theta_0 t_c^2 n_c(n_c-1)}{2(-t_c)^{n_c}} ,
\end{aligned}$$

ii) $t > t_c$,

$$\begin{aligned}
\theta(t) &= 0.0 , \\
\dot{\theta}(t) &= 0.0 , \\
\ddot{\theta}(t) &= 0.0 ,
\end{aligned} \tag{4.6}$$

ここで，

t ：ステップ t_iからの経過時間,
t_c ：θ, $\dot{\theta}$及び$\ddot{\theta}$が 0 になるまでの時間,
n_c ：角加速度曲線の次数,
θ_0 ：ステップ t_iにおける初期角度,
$\dot{\theta}_0$ ：ステップ t_iにおける初期角速度,
$\ddot{\theta}_0$ ：ステップ t_iにおける初期角加速度,

ψについても同様にして作成する。

以上，本制御法をまとめると Fig.4.6に示したブロック線図となる。

4.4 歩行シミュレーション

比較のために，外乱なし直進完全歩行のシミュレーション結果を Fig.4.7に示す [4]。ここで 1 歩 300[mm]，1 歩 1.0[s]，スティック線図は 0.1[s] 間隔で表示している。なお，制御周期は 10[ms] である。この歩行と同一の条件の下に，未知の外乱を作用させた場合 [9, 10] のシミュレーション結果の例を Fig.4.8～Fig.4.12に示す。ベクトル$\boldsymbol{F}_d$はロボット腰部に外乱として作用させた矩形状の外力の大きさと方向を示している。例えば，Fig.4.8は，歩行開始後 3.8[s] 経過した時に，ロボット腰部に進行方向後方へ矩形状の外力を 10[kgf]，0.1[s] 間作用させた場合である。各図から，外力によるＺＭＰ変化を補償し安定歩行が持続していることが分かる。また，この歩行に関して腰部に外力が作用した場合の歩行中補償可能な外力の範囲を Fig.4.13～Fig.4.16に示す。Fig.4.13は進行方向前方から，Fig.4.14は後方から 0.1[s] 間外力を作用させた場合で，Fig.4.15は前方から，Fig.4.16は後方から 1.0[s] 間外力を作用させた場合である（図には外力の横方向からの成分についても同時に示している）。

実際のハードウェア・ロボットへの適用に際して，歩行中のＺＭＰ計測の問題に関しては，Fig.4.17に示すように足底の 4 つ角に小型の 1 軸力センサを設置することで以下のように求められる。

$$\begin{aligned} XZMP &= x_{LF} - l_{xF} , \\ YZMP &= y_{LF} - l_{yL} , \end{aligned} \tag{4.7}$$

$$\begin{aligned} l_{xF} &= \frac{L_x(F_{LR}+F_{RR})}{F} , \\ l_{xR} &= \frac{L_x(F_{LF}+F_{RF})}{F} , \\ l_{yL} &= \frac{L_y(F_{RF}+F_{RR})}{F} , \\ l_{yR} &= \frac{L_y(F_{LF}+F_{LR})}{F} , \\ F &= F_{LR}+F_{RR}+F_{LF}+F_{RF} , \end{aligned}$$

ここで，

$$\begin{aligned} F_{LR} &\quad : \text{鉛直床反力（左後），} \\ F_{RR} &\quad : \text{鉛直床反力（右後），} \\ F_{LF} &\quad : \text{鉛直床反力（左前），} \\ F_{RF} &\quad : \text{鉛直床反力（左前），} \\ (x_{LF}, y_{LF}, 0) &\quad : \text{力センサ（左前）の位置ベクトル．} \end{aligned}$$

例えば，測定誤差等の影響によって，

$$\begin{aligned} l_{xF}+l_{xR} &\neq L_x , \\ l_{yL}+l_{yR} &\neq L_y , \end{aligned} \tag{4.8}$$

となった場合には，

$$\begin{aligned} \tilde{l_{xF}} &= \frac{l_{xF}^2 - l_{xR}^2 + L_x^2}{2L_x}, \\ \tilde{l_{yL}} &= \frac{l_{yL}^2 - l_{yR}^2 + L_y^2}{2L_y}, \end{aligned} \tag{4.9}$$

のように修正する。上記は単脚支持の場合である。両脚支持の場合においては，それぞれの足において上記の方法によりＺＭＰを算出し，これと床反力情報から以下のように求める [11]。

$$\begin{aligned} XZMP &= \frac{F_R x_R + F_L x_L}{F_R + F_L}, \\ YZMP &= \frac{F_R y_R + F_L y_L}{F_R + F_L}, \end{aligned} \tag{4.10}$$

ここで，

F_R ：右足鉛直床反力（合成力），
F_L ：左足鉛直床反力（合成力），
$(x_R, y_R, 0)$ ：右足ＺＭＰ，
$(x_L, y_L, 0)$ ：左足ＺＭＰ．

この測定は，歩行中の動的な力の測定であり，かつロボットのリアルタイム制御を目的としているため，高速な処理過程が要求される。これには一般的なロードセルでも可能であるが，より高い周波数で計測可能であり，高い剛性を持つ計測システムを構成できる圧電型の力センサを用いるのが適当である。

補償動作の計算は上記に示した７歩の外乱なし直進完全歩行の場合で，ＣＰＵ時間が約2.7[s]（ＦＡＣＯＭ M-780/20；スカラ計算）で，１制御周期当たり約 2.8[ms] である。今回のＺＭＰの計算では便宜上式 (4.2) を基に計算しているが，実際は式 (4.7)，(4.9)，(4.10) を用いて容易に計算できるため，計算時間はさらに短縮されるものと思われる。これと同程度もしくはそれ以下のマイクロ・プロセッサ（トランスピュータ）の使用で，コンピュータをロボットに搭載し，リアルタイムでの演算が可能である。

4.5 今後の方針

本章では，４年度に提案した，２足歩行ロボットのリアルタイム制御法「ＺＭＰの動的監視に基づく２足歩行ロボットの制御法」の応用研究について述べた。本制御法を外乱の作用する直進完全歩行に適用し，シミュレーションによってその有効性を確認した。本制御法は，歩行中歩容が変化するような場合や方向転換動作にも適応が可能である。

4.5.1 今後の課題

i. 補償動作の生成にロボットの数式モデルを用いているため，ロボットを正確にモデル化することが必要である。このため，ロボットのモデル化やハードウェアの機械的性質による誤差等に起因した歩行不安定の可能性がある。この問題に対しては，ニューロ・モ

デルやファジイ制御の導入が考えられる。

ii. 事前に修正用ファクターやしきい値などのいくつかのパラメータ・サーベイを必要とするためこれらについて定量化が必要である。これらの値は，歩行速度などのいくつかの歩容パラメータに依存する値であるから，さらにシミュレーションを進めることでこれらの定量化は可能である。

iii. 外乱などによって，上体のみでは補償しきれない場合に対処するための，制御法の開発である。これは，歩幅や周期を変更することによって補償するか，腰の位置の変更でも補償することが可能である。また，上体の胴の長さを長くすることも補償範囲の拡大につながる。

iv. 本制御法を高速歩行や方向転換動作に適応させた場合の，ヨー軸まわりのモーメント補償の問題である。ヨー軸まわりのモーメントの補償は，早稲田大学の WL-12RV[12] のような方法や，腕を付加することによって可能になる。

v. 腰及び足の軌道について。今回は，腰の軌道は一定高を等速（定常歩行時）で動くと仮定し，足の軌跡は接地時の相対速度が０となるようにスプライン関数による補間を行って作成した。これらの軌道をエネルギーやアクチュエータのトルク・パワーなどに関する評価関数の下に最適化し，より安定で自然な歩行とするべきである。

4.5.2 今後の計画

i. 歩行機能の高度化

知能ロボットの移動機構に要求される環境適合性の向上のため，開発済システムに，方向転換，不整地歩行等の機能を追加する。

ii. センシング・システム搭載のクローラ走行ロボットの整備

早期実用化が困難である２足歩行ロボットでの実現に先立ち，センシング・システム，命令理解システムとクローラ型移動機構とを統合し，ＨＡＳＰにおける巡回点検知能ロボットのプロトタイプ・システムを開発する。これにより，各要素間の関係を明確にし，ハードウェア化における問題点の明確化及び感覚運動学習（センシング・システム）に基づく制御法の開発を行い，巡回点検用知能ロボットの基礎的な技術開発を行う。このステップを踏まえることで２足歩行移動機構への移行もスムーズに行える。

iii. ２足歩行ロボット・ハードウェアのプロトタイプの設計・試作

上記で開発したシステムの移動機構を２足歩行ロボットに置き換えるため，２足歩行ロボット・ハードウェアのプロトタイプの設計・試作をすることによって，２足歩行ロボット・シミュレーション・モデルの妥当性及び制御性の検証，ハードウェア上の問題点の検討及び巡回点検用ロボットの移動機構としての２足歩行の有効性を確認する。また，この結果を基に実用を前提とした，２足歩行ロボット設計のための基盤技術を確立する。

参考文献

[1] M.Akimoto, et al., "HASP:Human Acts Simulation Program" *Proc. of Mathematical Methods and Supercomputing in Nuclear Applications (M & C + SNA'93), Karlsruhe, Germany* (1993) pp.420-431.

[2] 高西，加藤　他，"上体補償機構を有する２足歩行ロボット WL-12 の開発" 第５回日本ロボット学会学術講演会 (1987) pp.579-582.

[3] E.Kume and A.Takanishi, "Numerical Simulation for Design of Biped Locomotion Robots" *Proc. of Mathematical Methods and Supercomputing in Nuclear Applications (M & C + SNA'93), Karlsruhe, Germany* (1993) pp.408-419.

[4] 久米，"ＺＭＰの動的監視に基づく２足歩行ロボットの制御法" *JAERI-M 93-253* (1994).

[5] M.Vukobratović, 歩行ロボットと人工の足 (加藤，山下 訳, 日刊工業新聞社，1975).

[6] A.Takanishi, et al., "The Realization of Dynamic Walking by the Biped Walking Robot WL-10RD" *Proc. of ICAR'85* (1985) pp.459-466.

[7] 久米，"人間型ロボットの二足歩行シミュレーション" *JAERI-M 91-047* (1991).

[8] 長島, 数値計算法 (槙書店，1986).

[9] A.Takanishi, et al., "A Control Method for Dynamic Biped Walking Under Unknown External Force" *Proc. of IEEE/RSJ International Workshop on Intelligent Robot and System (IROS'90)* (1990) pp.795-801.

[10] 加藤，高西，久米，"２足歩行ロボットの外乱のある環境下における歩行モデルに関する調査報告" *JAERI-M 93-189* (1993).

[11] Q.Li, et al., "A Biped Walking Robot Having a ZMP Measurement System Using Universal Force-Moment Sensors" *Proc. of IEEE/RSJ International Workshop on Intelligent Robot and System (IROS'91)* (1991) pp.1568-1573.

[12] 高西，寸土　他，"上体運動により３軸モーメントを補償する２足歩行ロボットの開発～高速動歩行の実現～" 第６回知能移動ロボットシンポジウム (1993) pp.1-6.

Table 4.1 Generation of robot motion during walking.

	ロボット動作		
	歩容 (α, β)	前頭面上体動作 (θ)	矢状面上体動作 (ψ)
歩容静止期	歩行開始前にオフラインで生成し，事前にロボットに与えるこのパターンに従い動作 (下記 1)	発進期に支持脚側となる方向に上体を傾ける (下記 3) 初期状態は角度・角速度・角加速度ともに 0	角度・角速度・角加速度ともに 0
発進・定常・停止	〃	今回提案した制御法に従って上体補償動作を生成する（オンライン） a) 現在のＺＭＰの計測 b) 次のＺＭＰの予測 c) 基準ＺＭＰと予測ＺＭＰの偏差がしきい値より大きい 場合上体動作の生成を行う（下記 2,4 を使用） d) a)～c) の処理を制御周期毎に繰り返す	
歩容静止期	〃	上記制御は継続 上体の動作が全て（角度・角速度・角加速度）0 になったら終了	

歩行開始前にロボットに与えられているもの

1. 歩容：

歩容静止・発進（１歩）・定常（歩数に合わせて反復）・停止（１歩）・歩容静止

2. 基準ＺＭＰ軌跡：

安定領域の中央，連続的な軌跡

3. 発進前の歩容静止期における前頭面の上体動作（θ）：

ＺＭＰへの影響が小さい角加速度の変化を与えることで，発進期に支持脚側となる方向に上体を傾ける

4. ＺＭＰ偏差に対するしきい値，重み係数（ＺＭＰ修正用重み係数）

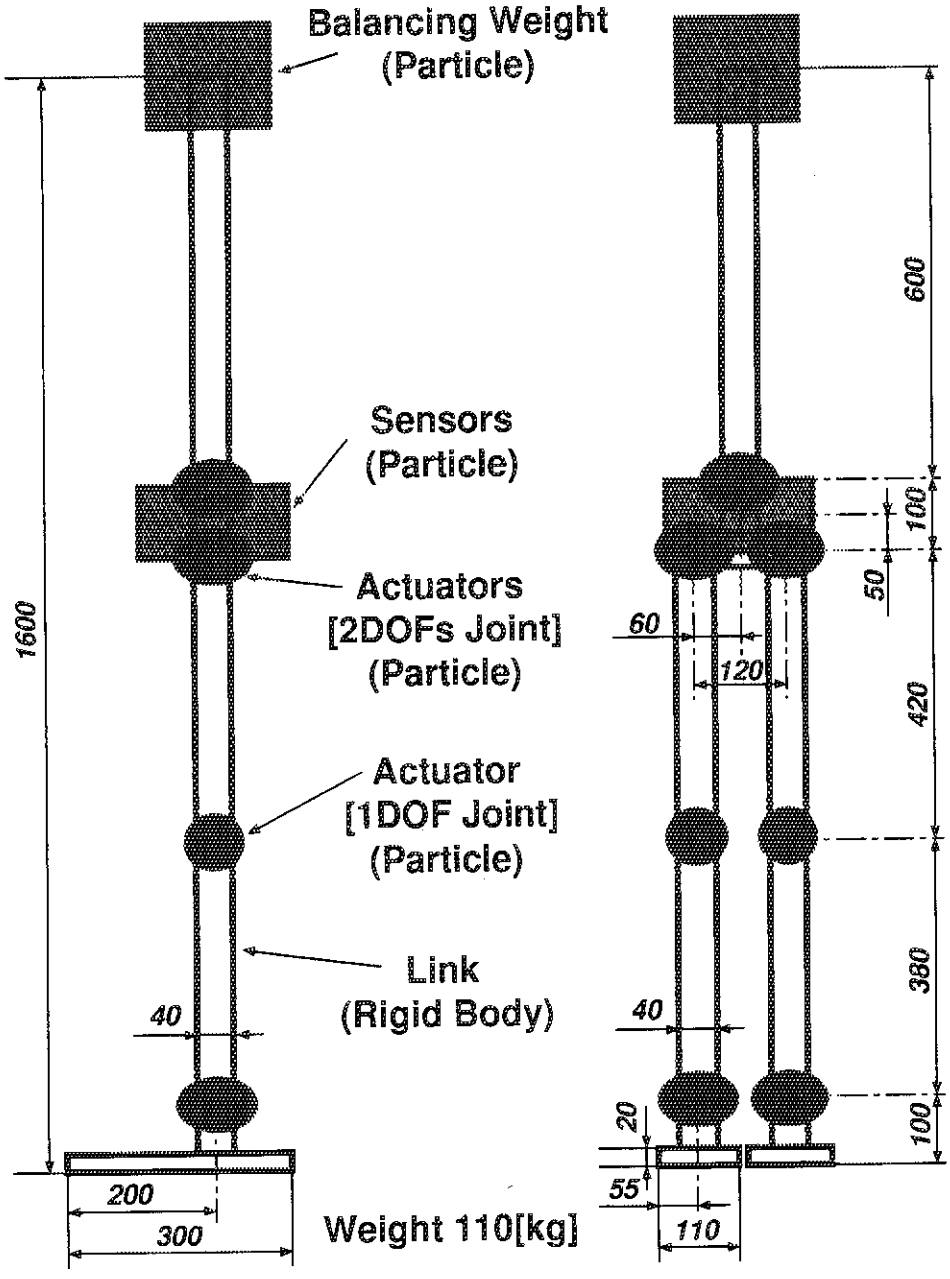


Fig. 4.1 Configuration of robot joints and body elements.

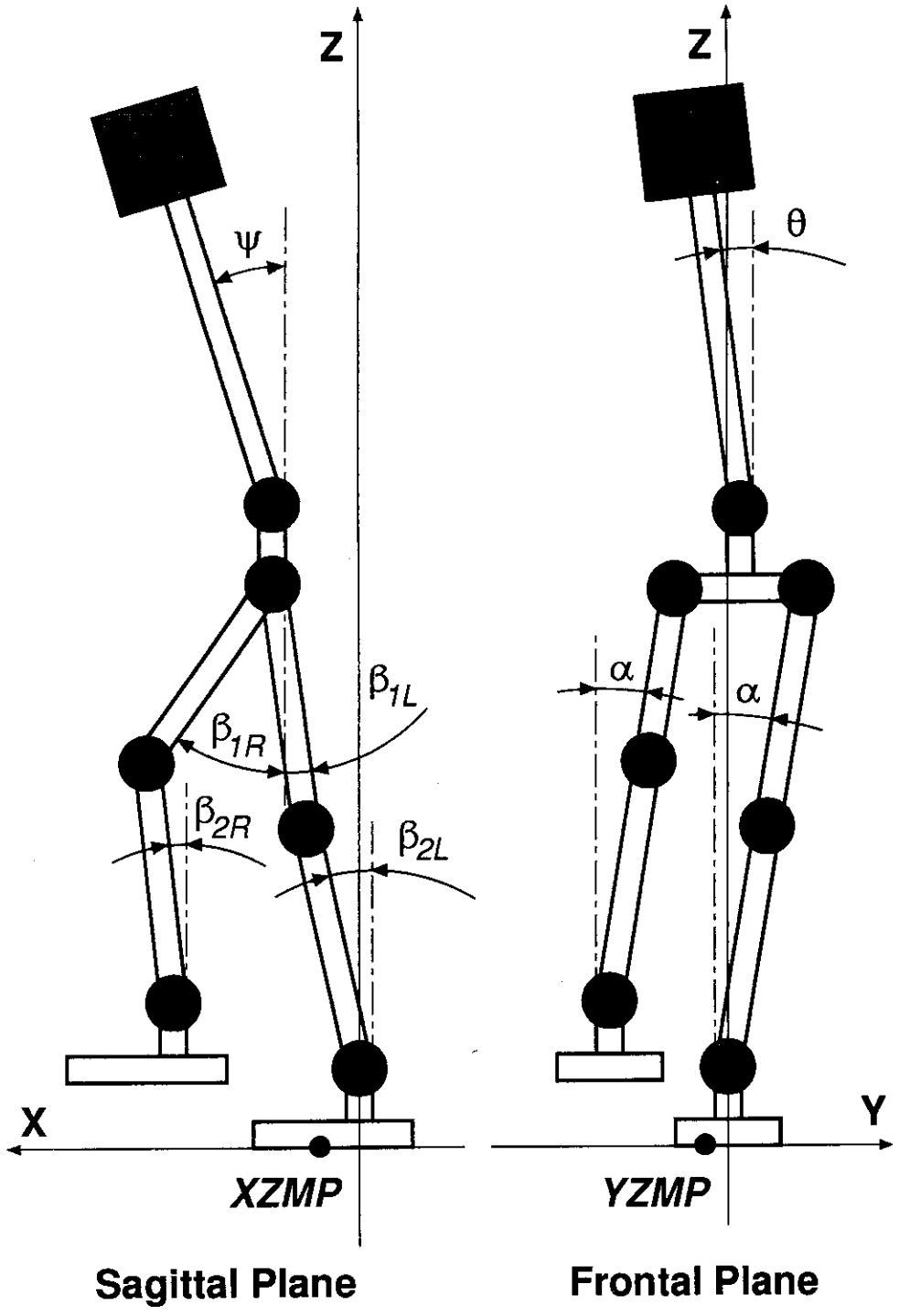


Fig. 4.2 Assignment of DOFs of robot model.

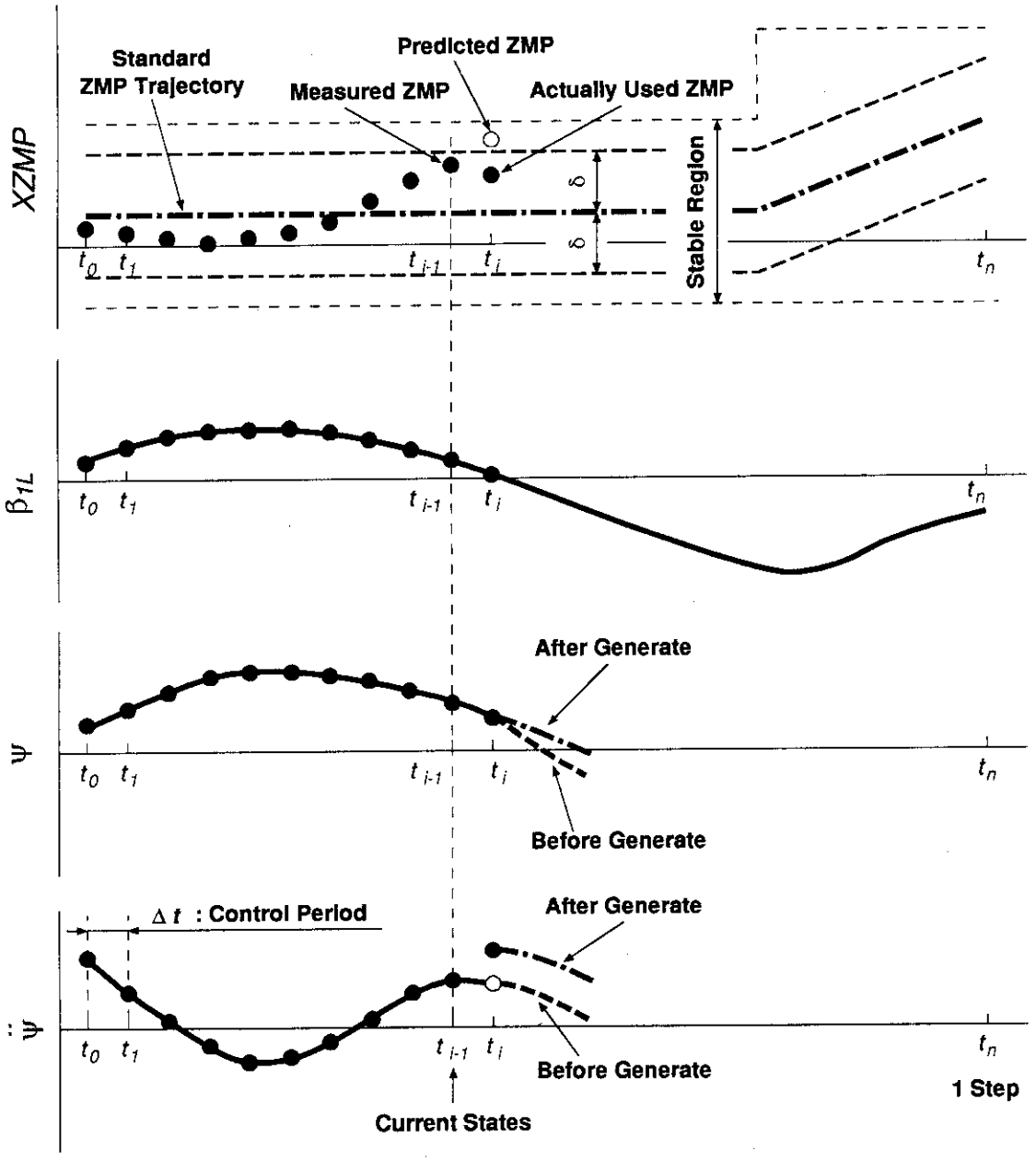


Fig. 4.3 Concept of proposed control method.

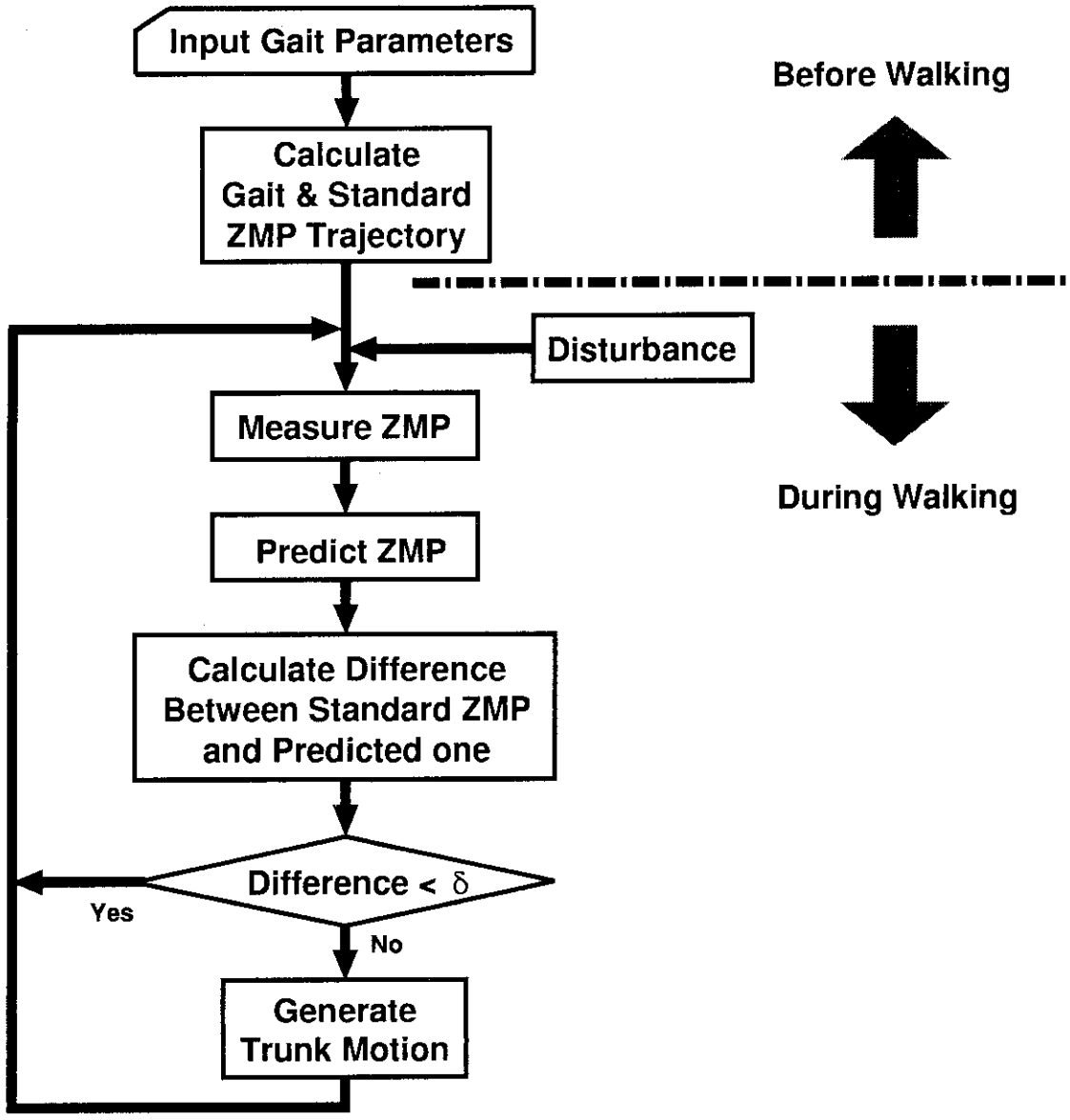


Fig. 4.4 Flow chart of proposed control method.

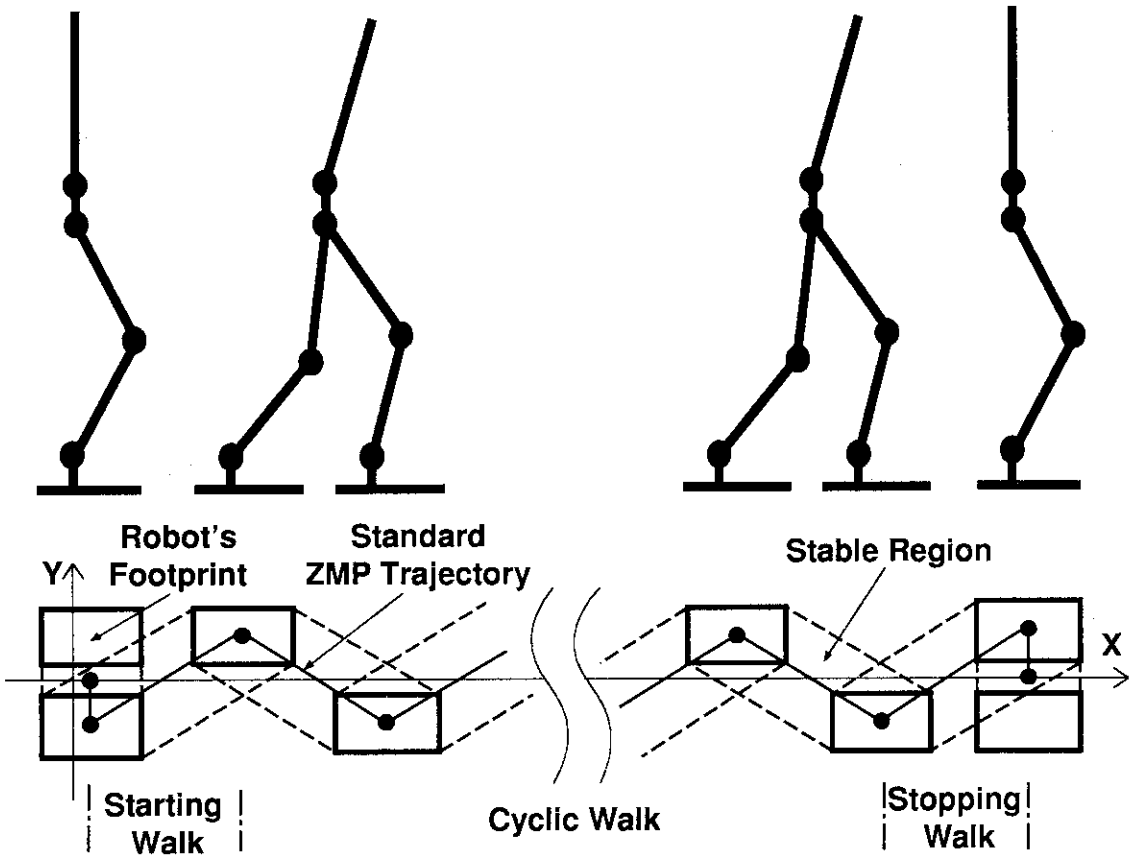


Fig. 4.5 Stable region and standard ZMP trajectory.

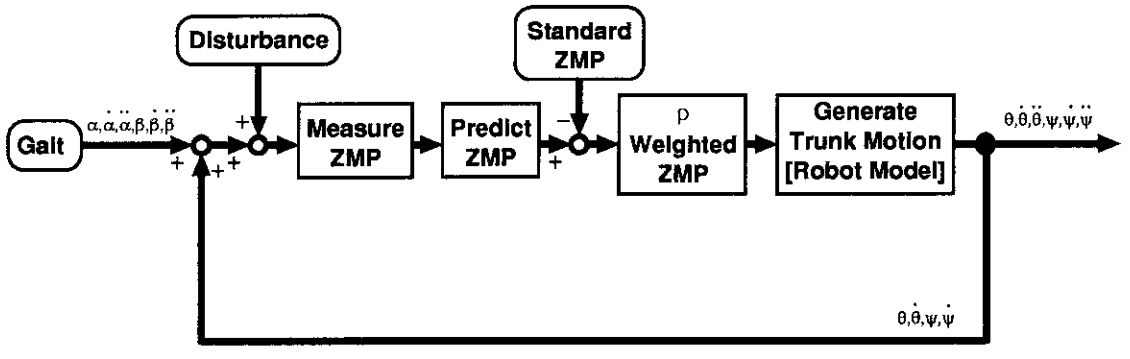


Fig. 4.6 Block diagram of proposed control method.

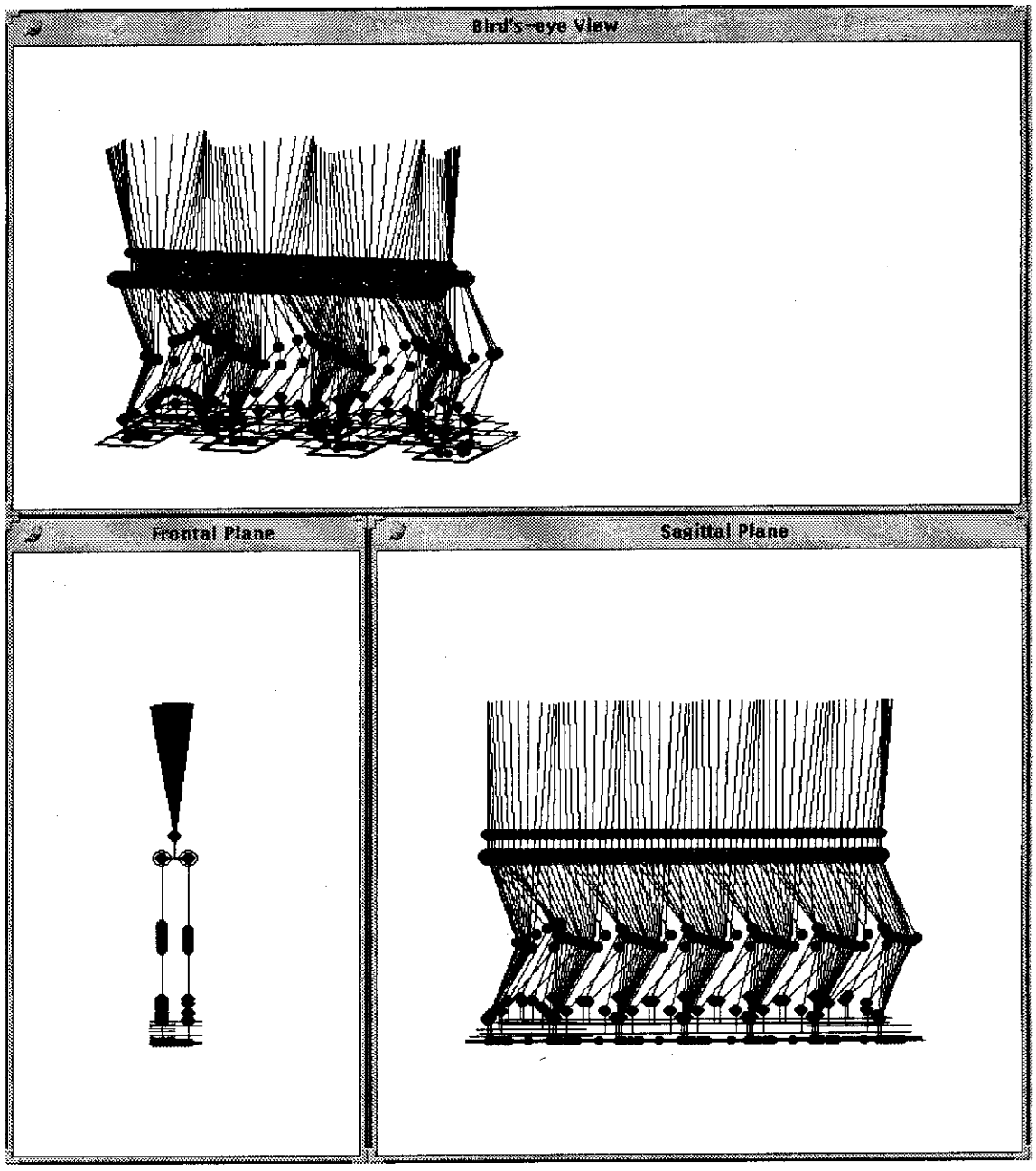


Fig. 4.7 Simulated result of walking without external force. [300 mm/step, 1.0 s/step]

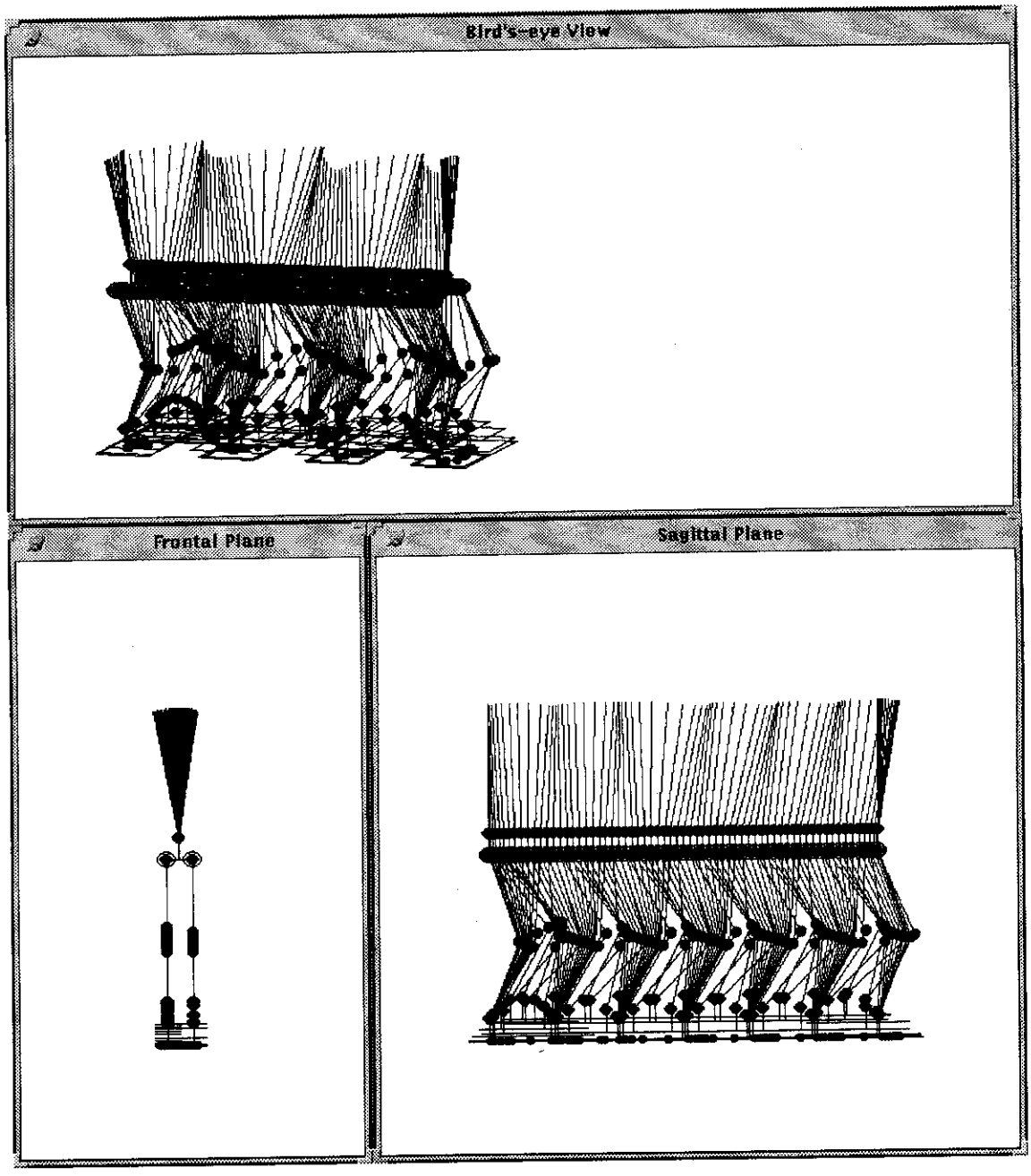


Fig. 4.8 Simulated result of walking under unknown external force.
[$\boldsymbol{F}_d$=(-10.0, 0.0, 0.0) kgf, Time Duration=0.1 s, at 3.8 sec. after walking starts]

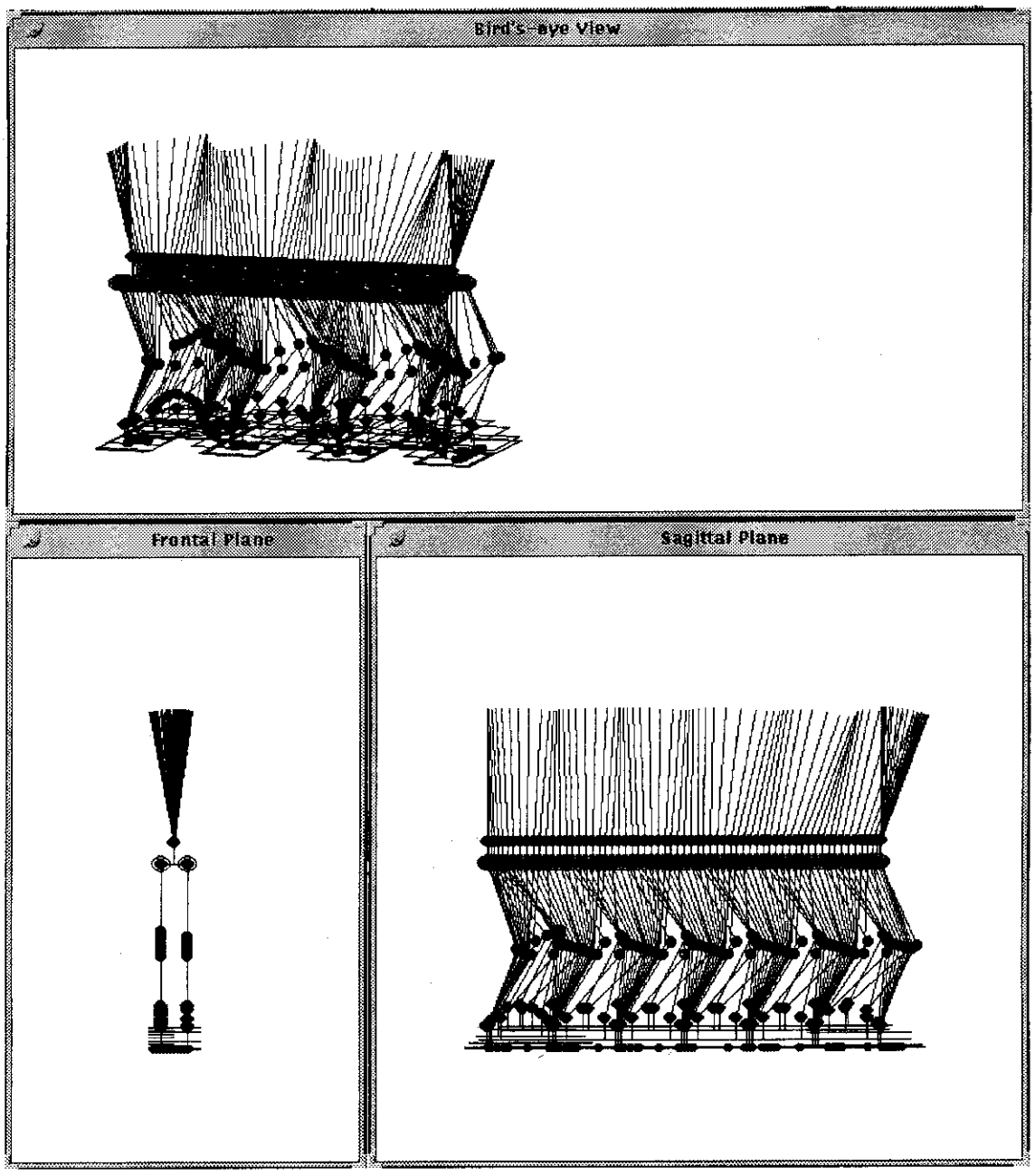


Fig. 4.9 Simulated result of walking under unknown external force.
[$\boldsymbol{F}_d$=(-3.8, 0.0, 0.0) kgf, Time Duration=1.0 s, at 4.5 sec. after walking starts]

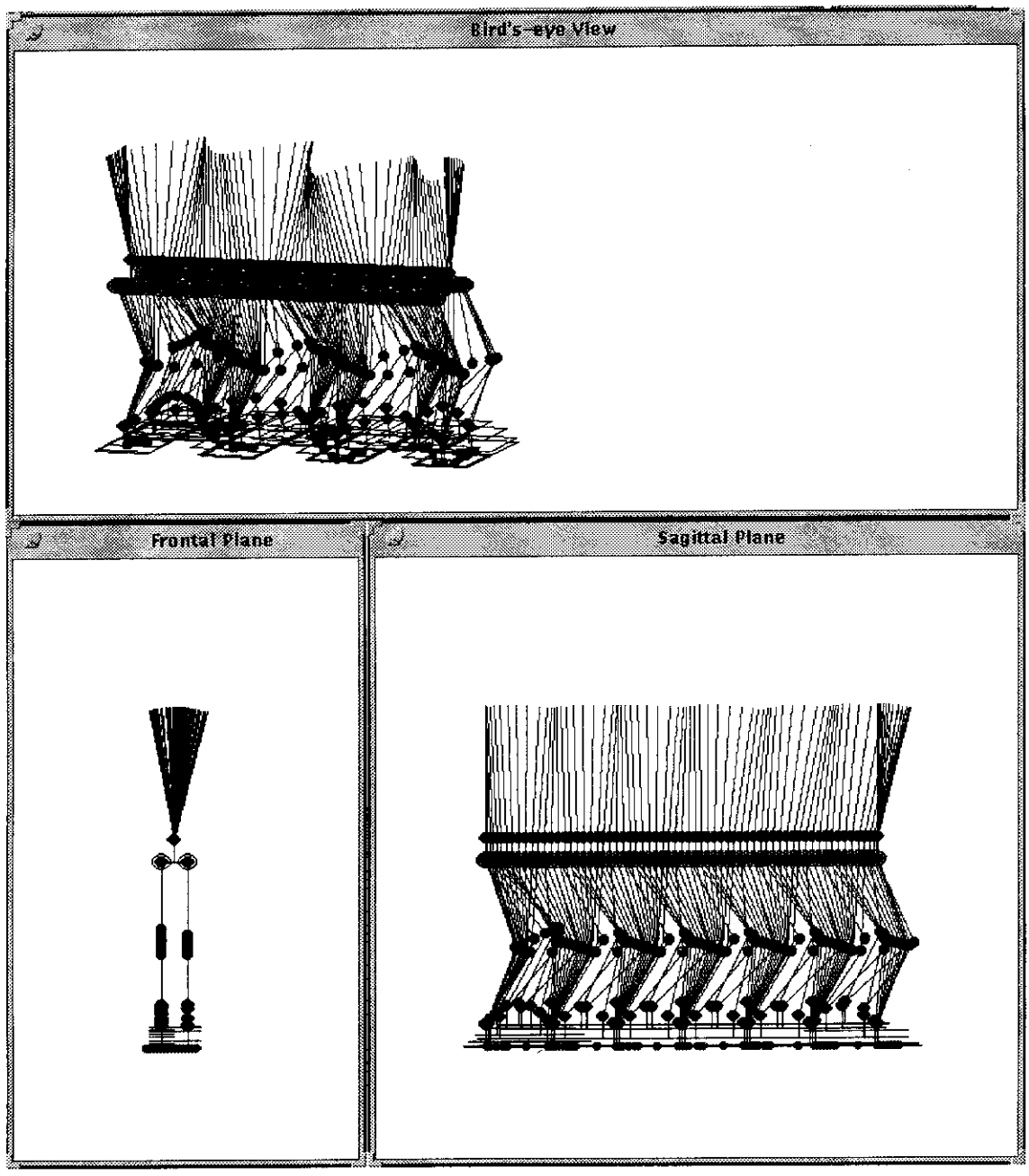


Fig. 4.10 Simulated result of walking under unknown external force.
[$\boldsymbol{F}_d$=(0.0, 5.0, 0.0) kgf, Time Duration=0.1 s, at 4.5 sec. after walking starts]

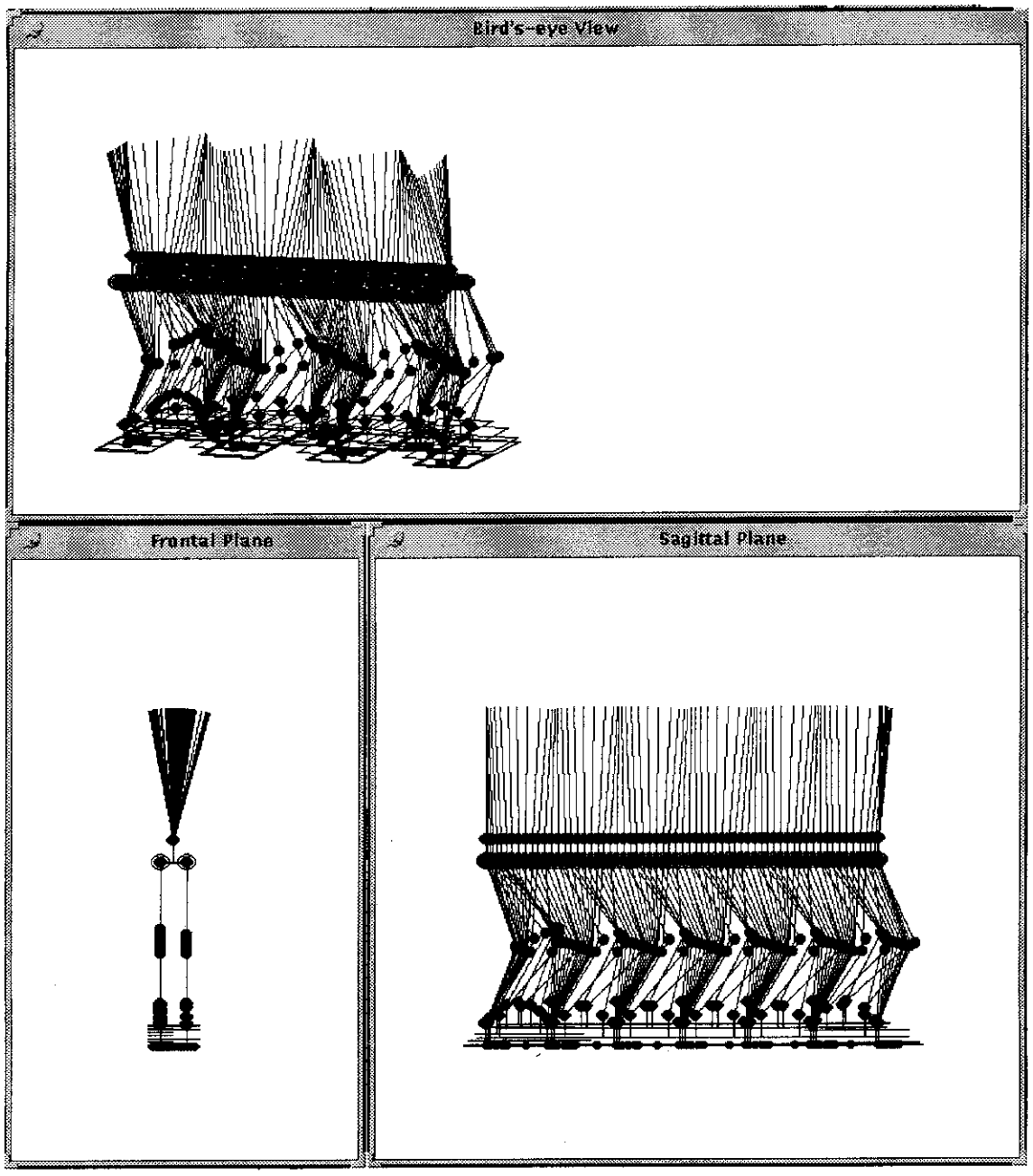


Fig. 4.11 Simulated result of walking under unknown external force.
[$\boldsymbol{F}_d$=(0.0, 2.0, 0.0) kgf, Time Duration=1.0 s, at 4.5 sec. after walking starts]

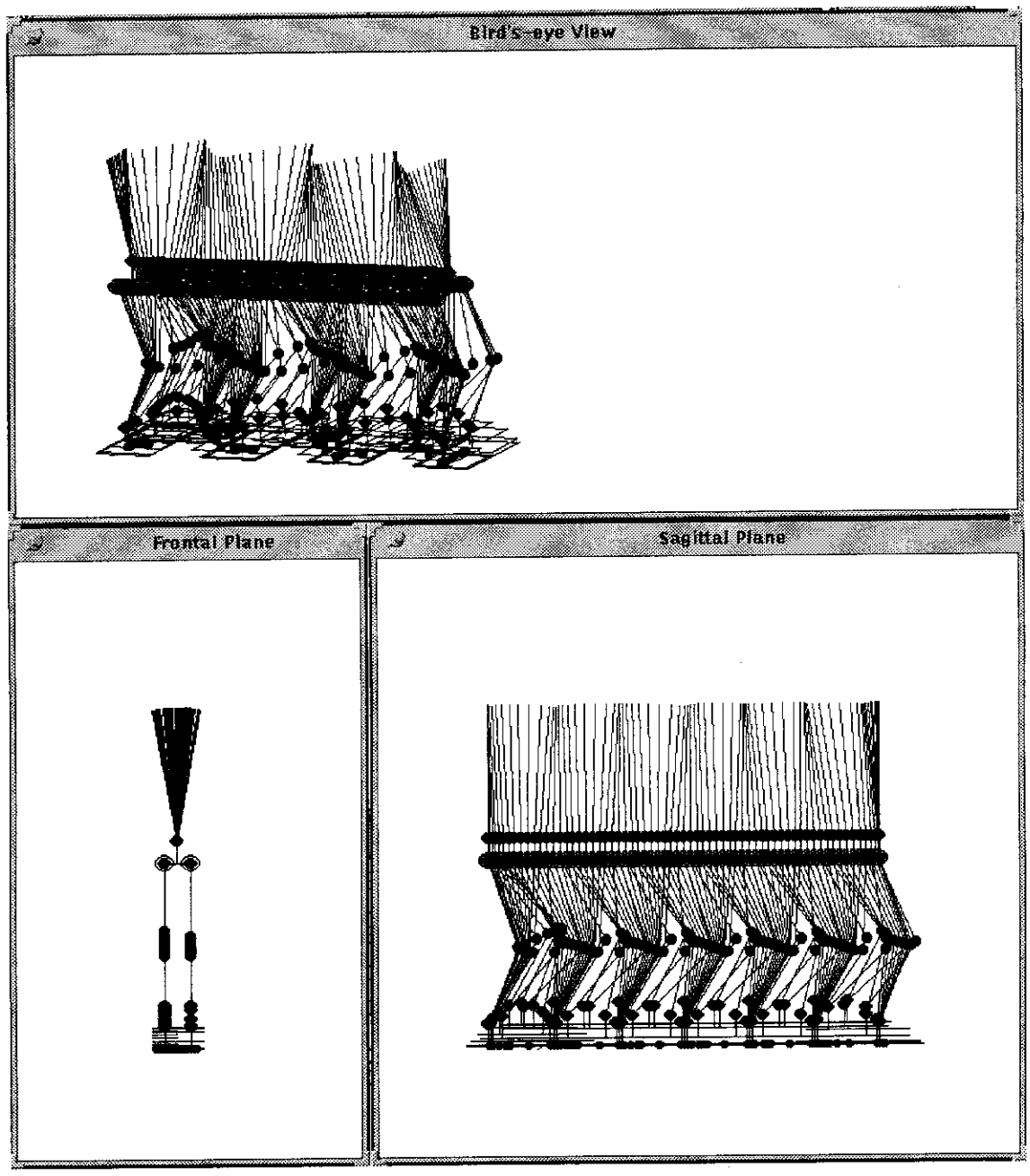


Fig. 4.12 Simulated result of walking under unknown external force.
[$\boldsymbol{F}_d$=(3.0, 3.0, 0.0) kgf, Time Duration=0.1 s, at 4.5 sec. after walking starts]

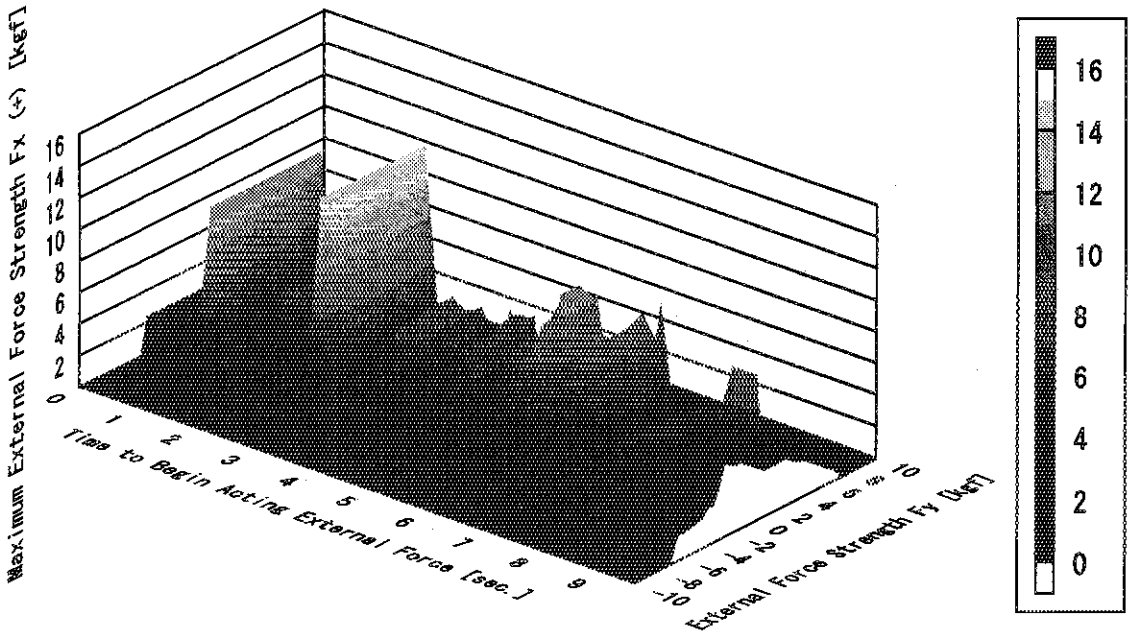


Fig. 4.13 Recoverable external force strength. [F_d=(F_x,F_y,0), Time Duration=0.1 s]

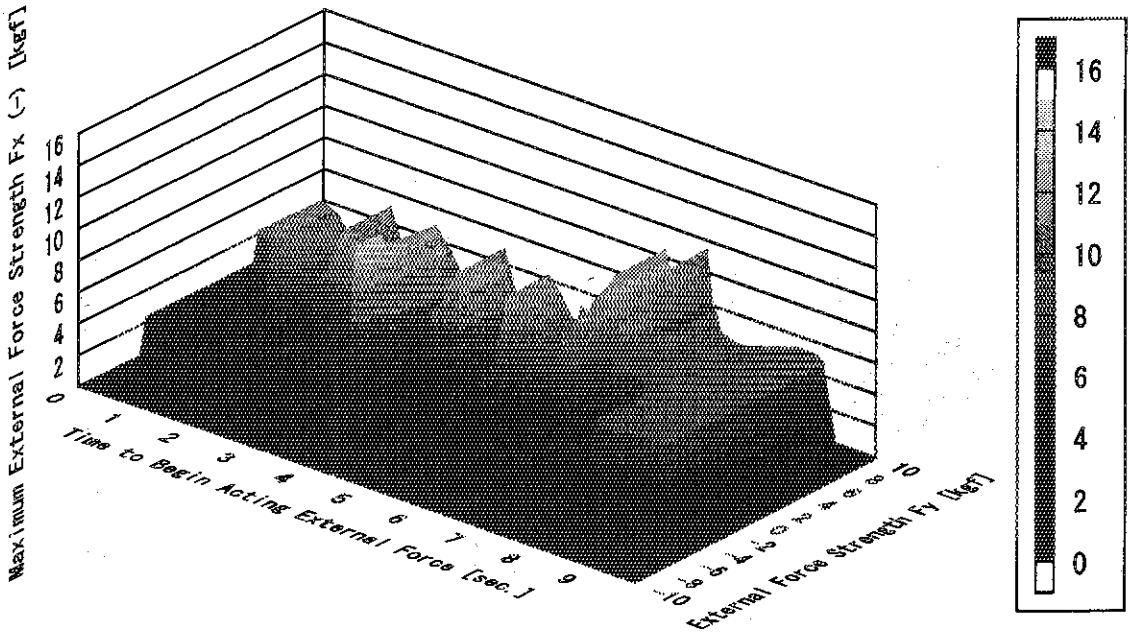


Fig. 4.14 Recoverable external force strength. [F_d=($-F_x$,F_y,0), Time Duration=0.1 s]

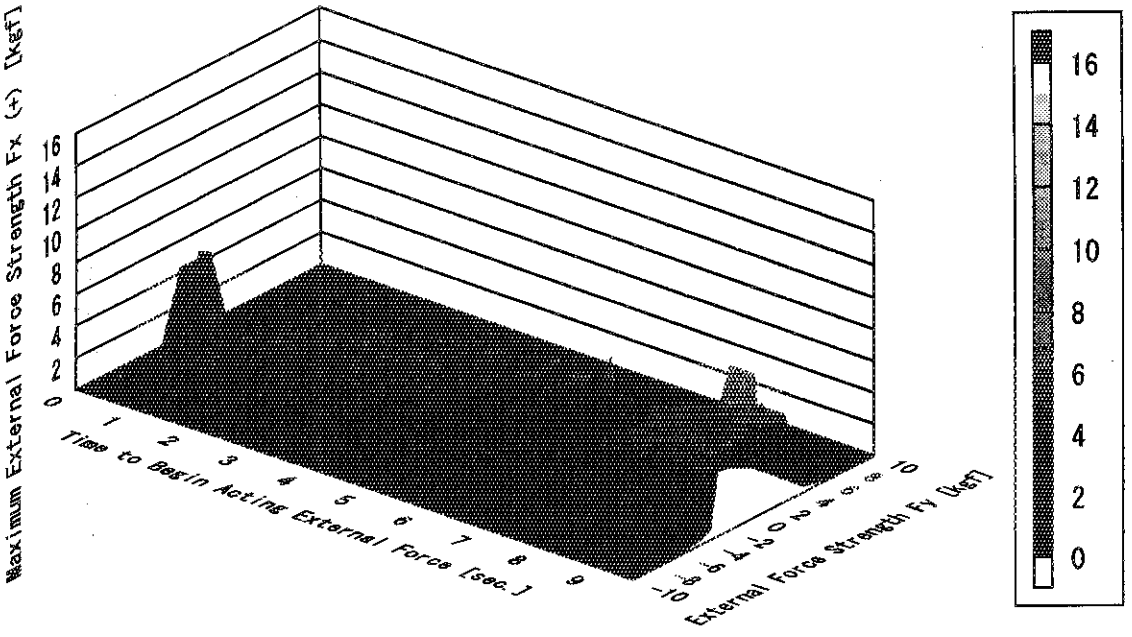


Fig. 4.15 Recoverable external force strength. [$\boldsymbol{F}_d$=(F_x,F_y,0), Time Duration=1.0 s]

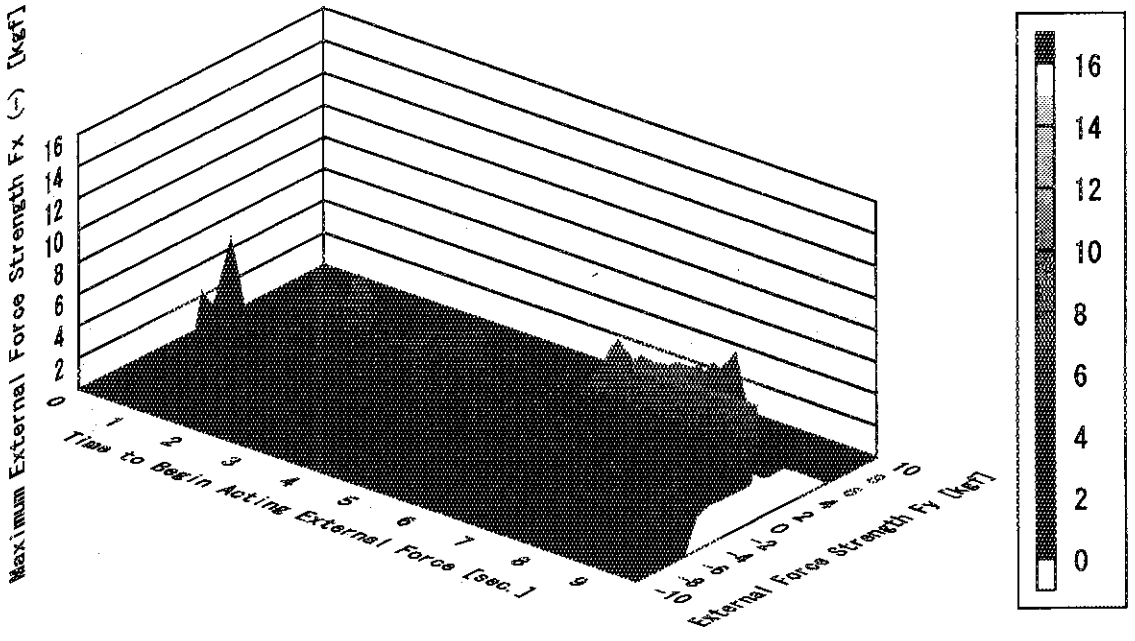


Fig. 4.16 Recoverable external force strength. [$\boldsymbol{F}_d$=($-F_x$,F_y,0), Time Duration=1.0 s]

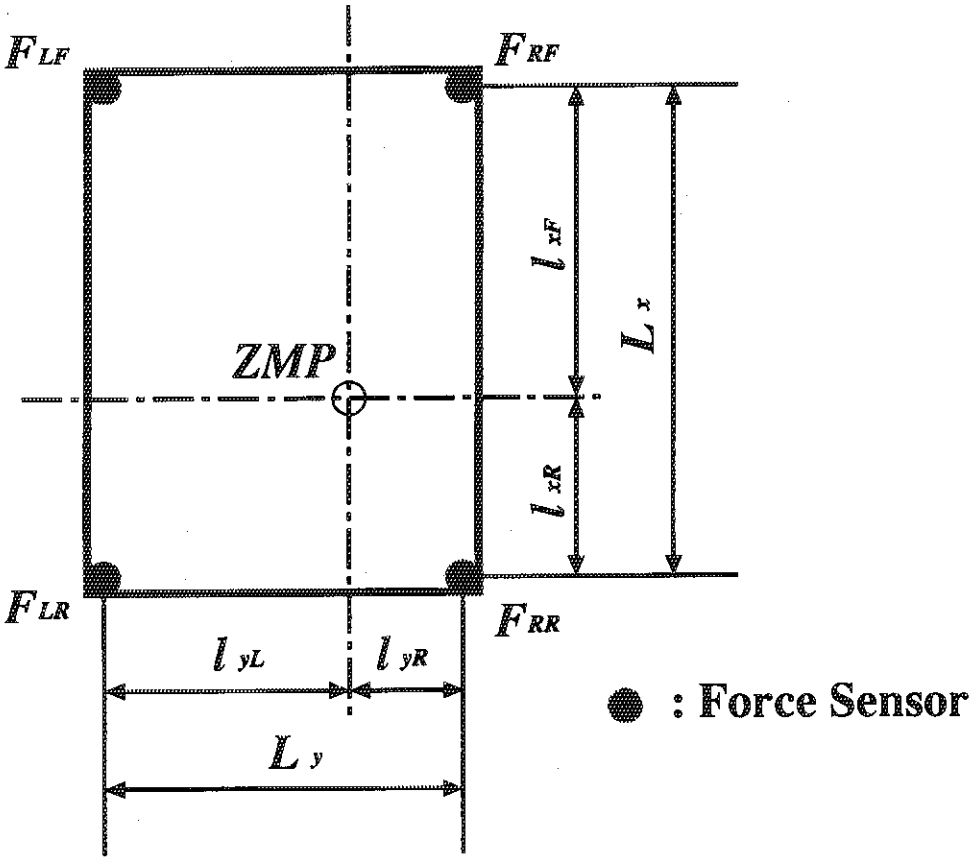


Fig. 4.17 Disposition of force sensors on a sole.

5. 施設形状データベース

施設形状データベースは，HASPにおける各種シミュレーションに必要な環境モデルを生成する役割を持つ。これらの環境モデルの性質は，処理において用いる手法に依存する。環境認識においては，ＣＣＤカメラからの画像から，エッジ情報を抽出し，環境モデルに対応する数学モデルと比較する手法が一般的である。このため，平成５年度は，環境認識に必要な２種類の輪郭線モデルを開発した。

5.1 基本立体に対する輪郭線数学モデルの生成

３次元空間において視点と視方向を与えた時のＣＳＧ (Constructive Solid Geometry) モデルに対する輪郭線数学モデルの生成は，複数の基本立体による集合演算を行った際の合成部分の取り扱いが難しいことから，画像処理における問題点の一つとなっている。平成５年度は，この問題点に対するアプローチとして，ＣＳＧモデルではなく，ＣＳＧモデルに使用される各基本立体に対する輪郭線数学モデル生成機能を開発した。

5.1.1 輪郭線数学モデルの生成を行う基本立体

本機能開発においては，既に情報システムセンターにおいて開発済みの施設形状データベースに登録された機器のうち，一部の機器を対象に，輪郭線数学モデルの生成システムを開発した。ただし，集合演算は考慮せず，機器の表現に用いられている基本立体すべての輪郭線数学モデルを生成することとした。対象とした環境モデルは，ＪＲＲ－３Ｍ（Fig. 5.1）及び模擬環境（Fig. 5.2）である。

現在，施設形状データベースにおいては，３次元物体の記述において，次の１１種類の基本立体を集合演算に用いている。

i. 直方体
ii. 円錐
iii. 円筒
iv. 球
v. 楕円体
vi. 楕円筒
vii. 双曲面
viii. 放物面
ix. 円環体
x. 楕円環体
xi. １／４円環体

今回の機能開発においては，上記基本立体のうち，平面及び二次曲面を中心に輪郭線数学モデルを生成した。即ち，今回，輪郭線数学モデルの生成対象となった基本立体は，直方体，円錐，円筒，球，楕円体及び楕円筒である。

5.1.2 輪郭線を記述する数学モデルの記述法

輪郭線を記述する数学モデルは，視方向ベクトルに直交する平面の一部として定義されるスクリーン上の二次のパラメータ関数とした。例えば，楕円体の輪郭を示す楕円のパラメトリック表現は次の式で表現される。

$$x = r_1 \cos\theta + a, y = r_2 \sin\theta + b$$

5.1.3 各基本立体に対する輪郭線数学モデルの生成法

与えられた視点の座標をV＝(V_x, V_y, V_z)，視方向をE＝(u, v, w) とする。ただし，

$$\mathrm{u}^2+\mathrm{v}^2+\mathrm{w}^2=1 \tag{5.1}$$

である。また，スクリーンを記述する平面を表現する方程式を

$$ux+vy+wz+d=0 \tag{5.2}$$

とする。また，輪郭線を生成する基本立体の方程式を

$$\begin{aligned} f(x,y,z) &= \mathrm{A}\,x^2+\mathrm{B}\,y^2+\mathrm{C}\,z^2 \\ &+ \mathrm{D}\,xy+\mathrm{E}\,yz+\mathrm{F}\,zx+\mathrm{G}\,x+\mathrm{H}\,y+\mathrm{J}\,z+\mathrm{K} \end{aligned} \tag{5.3}$$

とする。即ち，基本立体の表面上の点 (x,y,z) は，f $(x,y,z)=0$ を満たす。

(1) **視点Vからfへの接線**

いま，f上の一点に対し，視点Vから接線を引く。この接線の式を

$$x=\mathrm{l}\,\mathrm{t}+V_x, y=\mathrm{m}\,\mathrm{t}+V_y, z=\mathrm{n}\,\mathrm{t}+V_z \tag{5.4}$$

とする。ただし，

$$\mathrm{l}^2+\mathrm{m}^2+\mathrm{n}^2=1 \tag{5.5}$$

である。

いま，この接線がfに接する接点の座標をC＝(x_1, y_1, z_1) とする。この時のtの値をt_1とすると，式 (5.4) において，t＝t_1とし，これを式 (5.3) に代入し，f $(x_1, y_1, z_1)=0$ と置くことにより，t_1に関する二次方程式を得る。この方程式は，

$$\mathrm{A}\,(\mathrm{l}\,\mathrm{t}_1+V_x)^2+\mathrm{B}\,(\mathrm{m}\,\mathrm{t}_1+V_y)^2+\cdots \quad =0 \tag{5.6}$$

となる。Cは，接点なので，式 (5.6) は，t_1について重根を持つ。つまり，式 (5.6) のt_1に関する判別式は0となる。この判別式をgとすると，gは，l，m，nに関する二次式であり，

$$\mathrm{g}\,(\mathrm{l},\ \mathrm{m},\ \mathrm{n})=0 \tag{5.7}$$

と書ける。

(2) **接線とスクリーンの交点**

式(5.4)で表現される直線と式(5.2)で表現される平面の交点は，式(5.4)を式(5.2)に代入し，tに関して解くことで，求まる。このtをt_2とすると，t_2は，l，m，nの一次式で表現される。つまり，この一次式をhとすると，

$$t_2 = h\ (l,\ m,\ n) \tag{5.8}$$

であり，交点の座標は，

$$x_2 = l\ t_2 + V_x,\ y_2 = m\ t_2 + V_y,\ z_2 = n\ t_2 + V_z \tag{5.9}$$

である。

(3) スクリーン座標系

いま，スクリーンの基底ベクトルを$\vec{p}$，$\vec{q}$とする。$\vec{p}$をz軸に直交するように定めると，$\vec{p}$，$\vec{q}$は次のように書ける。

$$\vec{p}' = (v,\ -u,\ 0),\ \vec{p} = \vec{p}' / |\vec{p}'|,$$

$$\vec{q}' = (uw,\ vw,\ -(u^2+v^2)),\ \vec{q} = \vec{q}' / |\vec{q}'|,$$

また，

$$\vec{o} = (0,\ 0,\ d/w)$$

とすると，スクリーン上の点Xは，

$$X = \vec{o} + \alpha\vec{p} + \beta\vec{q} \tag{5.10}$$

と表現できる。

(4) スクリーン上の輪郭線の方程式

まず，式(5.5)，式(5.7)，式(5.8)，式(5.9)を連立させて，視点Vを始点とし，fに接する直線とスクリーン平面の交点を求める。この結果，この交点X(x_2, y_2, z_2)は，l，mあるいはnの二次式として表現される。例えば，

$$x_2 = j_x(l) \tag{5.11}$$

$$y_2 = j_y(l) \tag{5.12}$$

$$z_2 = j_z(l) \tag{5.13}$$

と書ける。

次に，式(5.10)～(5.13)を連立させて，αとβの関係を求める。$\alpha-\beta$空間における関係式が，輪郭線を記述する方程式である。本システムによって生成されたＪＲＲ－３Ｍ及び模擬環境に対する輪郭線画像を，それぞれFig. 5.3及びFig. 5.4に示す。また，円筒に対して上記処理を行った例を，次に示す。

5.1.4 円筒面に対する輪郭線モデルの生成

(1) スクリーン座標系

前節で述べたように，スクリーンの基底ベクトルを $\vec{p}$，$\vec{q}$ とすると，スクリーン上の点 $\vec{x}$ (x,y,z) は，

$$\vec{x}=\vec{o}+\alpha\vec{p}+\beta\vec{q} \tag{5.14}$$

と表せる。これを成分で表すと次のようになる。

$$x = p_x\alpha+q_x\beta \tag{5.15}$$

$$y = p_y\alpha+q_y\beta \tag{5.16}$$

$$z = o_z+q_z\beta \tag{5.17}$$

ここで，

$$p_x = \frac{v}{\sqrt{v^2+u^2}}$$

$$q_x = \frac{uw}{\sqrt{u^2w^2+v^2w^2+(u^2+v^2)^2}}$$

$$p_y = \frac{-u}{\sqrt{v^2+u^2}}$$

$$q_y = \frac{vw}{\sqrt{u^2w^2+v^2w^2+(u^2+v^2)^2}}$$

$$o_z = \frac{d}{w}$$

$$q_z = \frac{-(u^2+v^2)}{\sqrt{u^2w^2+v^2w^2+(u^2+v^2)^2}}$$

である。

(2) $\alpha-\beta$ 空間における円筒の輪郭線

円筒の式，

$$\begin{aligned}(x-x_0)^2 &+ (y-y_0)^2-r^2\\ &= x^2+y^2-2\,x_0x-2\,y_0y+x_0^2+y_0^2-r^2=0\end{aligned} \tag{5.18}$$

において，

$$G=-2\,x_0$$

$$H=-2\,y_0$$

$$K=x_0^2+y_0^2-r^2$$

とすると，円筒は式 (5.3) の形式で次のように表わせる。

$$f\,(x,y,z)=x^2+y^2+G\,x+H\,y+K \tag{5.19}$$

i) 円筒の上下面の輪郭線

はじめに，円筒の上下面の輪郭線を求める。

視点V（V_x, V_y, V_z)から任意のxy平面($z=z_1$)の円筒上の点(x_1, y_1, z_1)に引いた直線の式を

$$x = (x_1 - V_x)\ t + V_x \tag{5.20}$$

$$y = (y_1 - V_y)\ t + V_y \tag{5.21}$$

$$z = (z_1 - V_z)\ t + V_z \tag{5.22}$$

とすると，これらから，

$$x_1 = \frac{(x - V_x)(z_1 - V_z)}{z - V_z} + V_x \tag{5.23}$$

$$y_1 = \frac{(y - V_y)(z_1 - V_z)}{z - V_z} + V_y \tag{5.24}$$

となり，このx_1, y_1を円筒式 (5.19) に代入すると，f $(x_1, y_1, z_1) = 0$ を得る。

$$\begin{aligned}
f\ (x_1, y_1, z_1) = \ & 0 \\
= \ & \left(\frac{(x - V_x)(z_1 - V_z)}{z - V_z} + V_x\right)^2 \\
& + \left(\frac{(y - V_y)(z_1 - V_z)}{z - V_z} + V_y\right)^2 \\
& + G\left(\frac{(x - V_x)(z_1 - V_z)}{z - V_z} + V_x\right) \\
& + H\left(\frac{(y - V_y)(z_1 - V_z)}{z - V_z} + V_y\right) \\
& + K \\
= \ & (\ (x - V_x)(z_1 - V_z) + V_x(z - V_z))^2 \\
& + ((y - V_y)(z_1 - V_z) + V_y(z - V_z))^2 \\
& + G\ (z_1 - V_z)(x - V_x)(z - V_z) \\
& + G\ V_x(z - V_z)^2 \\
& + H\ (z_1 - V_z)(y - V_y)(z - V_z) \\
& + H\ V_y(z - V_z)^2 \\
& + K\ (z - V_z)^2
\end{aligned} \tag{5.25}$$

この式 (5.25) に，前述の式 (5.15)，式 (5.16)，式 (5.17) で表されるスクリーン上の座標x，y，zを代入してαとβの関係を求める。

$$\begin{aligned}
0 = \ & (\ (p_x\alpha + q_x\beta - V_x)(z_1 - V_z) + V_x(q_z\beta + o_z - V_z))^2 \\
& + ((p_y\alpha + q_y\beta - V_y)(z_1 - V_z) + V_y(q_z\beta + o_z - V_z))^2 \\
& + G\ (z_1 - V_z)(p_x\alpha + q_x\beta - V_x)(q_z\beta + o_z - V_z)
\end{aligned}$$

$$
\begin{aligned}
&+ \quad G V_x(q_z\beta + o_z - V_z)^2 \\
&+ \quad H(z_1 - V_z)(p_y\alpha + q_y\beta - V_y)(q_z\beta + o_z - V_z) \\
&+ \quad H V_y(q_z\beta + o_z - V_z)^2 \\
&+ \quad K(q_z\beta + o_z - V_z)^2 \\
= \quad &(\quad p_x(z_1 - V_z)\alpha + (q_x(z_1 - V_z) + q_z V_x)\beta - (V_x z_1 - o_z V_x))^2 \\
&+ \quad (p_y(z_1 - V_z)\alpha + (q_y(z_1 - V_z) + q_z V_y)\beta - (V_y z_1 - o_z V_y))^2 \\
&+ \quad G(z_1 - V_z)(p_x\alpha + q_x\beta - V_x)(q_z\beta + (o_z - V_z)) \\
&+ \quad G V_x(q_z\beta + (o_z - V_z))^2 \\
&+ \quad H(z_1 - V_z)(p_y\alpha + q_y\beta - V_y)(q_z\beta + (o_z - V_z)) \\
&+ \quad H V_y(q_z\beta + (o_z - V_z))^2 \\
&+ \quad K(q_z\beta + (o_z - V_z))^2 \\
= \quad &(\quad (p_x(z_1 - V_z))^2 + (p_y(z_1 - V_z))^2)\alpha^2 \\
&+ \quad ((q_x(z_1 - V_z) + q_z V_x)^2 + (q_y(z_1 - V_z) + q_z V_y)^2 \\
&+ \quad G(z_1 - V_z)\, q_x q_z \\
&+ \quad G V_x q_z^2 + H(z_1 - V_z)\, q_y q_z + H V_y q_z^2 + K q_z^2)\beta^2 \\
&+ \quad (2 p_x(z_1 - V_z)(q_x(z_1 - V_z) + q_z V_x) \\
&+ \quad 2 p_y(z_1 - V_z)(q_y(z_1 - V_z) + q_z V_y) \\
&+ \quad G(z_1 - V_z)\, p_x q_z + H(z_1 - V_z)\, p_y q_z)\alpha\beta \\
&+ \quad (-2 p_x(z_1 - V_z)(V_x z_1 - o_z V_x) - 2 p_y(z_1 - V_z)(V_y z_1 - o_z V_y) \\
&+ \quad G(z_1 - V_z)\, p_x(o_z - V_z) + H(z_1 - V_z)\, p_y(o_z - V_z))\alpha \\
&+ \quad (-2(q_x(z_1 - V_z) + q_z V_x)(V_x z_1 - o_z V_x) \\
&- \quad 2(q_y(z_1 - V_z) + q_z V_y)(V_y z_1 - o_z V_y) \\
&+ \quad G(z_1 - V_z)(q_x(o_z - V_z) - q_z V_x) + 2 G V_x q_z(o_z - V_z) \\
&+ \quad H(z_1 - V_z)(q_y(o_z - V_z) - q_z V_y) + 2 H V_y q_z(o_z - V_z) \\
&+ \quad 2 K q_z(o_z - V_z))\beta \\
&+ \quad (V_x z_1 - o_z V_x)^2 + (V_y z_1 - o_z V_y)^2 \\
&- \quad G(z_1 - V_z)\, V_x(o_z - V_z) + G V_x(o_z - V_z)^2 \\
&- \quad H(z_1 - V_z)\, V_y(o_z - V_z) + H V_y(o_z - V_z)^2 \\
&+ \quad K(o_z - V_z)^2 \qquad (5.26)
\end{aligned}
$$

となり，任意のxy平面($z = z_1$)で切られる円筒の断面は，スクリーン上でαとβの２次式として表わせる。従って，z_1を円筒の上面または下面のz座標とすれば，上下面の輪郭線の式が作成できる。

ii) 円筒の側面の輪郭線

次に，円筒の側面の輪郭線を求める。

i) で求めた上下面に対する２つの式で表わされる２次曲線に接する２本の直線の式が，円筒の側面の輪郭線の式となる。

(3) 求めた式のパラメータ関数への変換

i) ２次曲線の中心を求める

２次曲線

$$a\alpha^2+2\,h\alpha\beta+b\beta^2+2\,g\alpha+2\,f\beta+c=0 \tag{5.27}$$

の中心の座標は，連立方程式

$$a\alpha+h\beta+g=0 \tag{5.28}$$

$$h\alpha+b\beta+f=0 \tag{5.29}$$

の解として，次のように与えられる。

$$\alpha_0=\frac{-(b\,g-f\,h)}{a\,b-h^2} \tag{5.30}$$

$$\beta_0=\frac{-(h\alpha_0+f)}{b} \tag{5.31}$$

ii) ２次曲線を原点移動する

式 (5.27) の方程式に，

$$\alpha=\alpha'+\alpha_0 \tag{5.32}$$

$$\beta=\beta'+\beta_0 \tag{5.33}$$

を代入すると，i) で求めた中心へ原点が移り式 (5.27) は，

$$a\alpha'^2+2\,h\alpha'\beta'+b\beta'^2+\bar{c}=0 \tag{5.34}$$

となる。ここで，

$$\bar{c}=a\alpha_0^2+b\beta_0^2+2\,h\alpha_0\beta_0+2\,g\alpha_0+2\,f\beta_0+c \tag{5.35}$$

である。

iii) ２次曲線を回転する

さらに，式 (5.34) に対して，

$$\tan 2\theta=\frac{2\,h}{a-b} \tag{5.36}$$

で与えられる角θだけ軸を回転すれば，２次曲線の方程式は，

$$\bar{a}\bar{\alpha}^2+\bar{b}\bar{\beta}^2+\bar{c}=0 \tag{5.37}$$

の形に直せる。ここで$\bar{a}$, $\bar{b}$は，

$$t^2-(a+b)\,t+a\,b-h^2=0 \tag{5.38}$$

の根で，h＞0なら大きい方が$\bar{a}$，h＜0なら小さい方が$\bar{a}$である。

iv) 2次曲線を媒介変数表示する

式(5.35)を媒介変数ϕを用いて表わすと，

$$\bar{\alpha}=-\sqrt{\frac{\bar{c}}{\bar{a}}}\cos(\phi) \tag{5.39}$$

$$\bar{\beta}=-\sqrt{\frac{\bar{c}}{\bar{b}}}\sin(\phi) \tag{5.40}$$

となる。ここで，$0 \leq \phi < 2\pi$の範囲で$(\bar{\alpha}_i, \bar{\beta}_i)$，$i=1,N$をもとめ，

$$\begin{aligned}(\alpha_i', \beta_i') \;=\; & (\bar{\alpha}_i \cos(\theta) - \bar{\beta}_i \sin(\theta), \\ & \bar{\alpha}_i \sin(\theta) + \bar{\beta}_i \cos(\theta))\end{aligned} \tag{5.41}$$

さらに，

$$(\alpha_i, \beta_i) \;=\; (\alpha_i' + \alpha_0, \beta_i' + \beta_0), i=1,N \tag{5.42}$$

を計算して，スクリーン上の座標を求める。この点列を結ぶ線分を順に表示することで2次曲線を描くことができる。

5.2 多面体に対する輪郭線数学モデルの生成

輪郭線モデルの開発においては，上記述べたように，基本立体から直接輪郭線を表現する方程式を生成する機能に加え，ポリゴンの集合体（多面体）として表現される3次元物体の輪郭線を表示する機能を開発した。本機能は，上記5．1に記述したＣＳＧモデル表現に用いられる基本立体の輪郭線生成と異なり，ポリゴン（本データベースにおいては，三角形）の集合体に対する輪郭線の生成である。以下に，この機能を実現するために必要な処理の概要を示す。

5.2.1 頂点リストの作成

ポリゴンは頂点と稜線によって記述できる。各物体を表現するポリゴンを構成する各頂点は，i) ポリゴンがその表面に位置する基本立体（変換前の基本立体）の情報（基本立体の識別子），ii) その頂点からの稜線と相手先の頂点の情報を持っている。

5.2.2 各頂点及び各辺のスクリーンへの投影

各ポリゴンの頂点及び辺を指定されたスクリーンへ投影する。投影法は，点投影（透視変換）とした。

5.2.3 稜線（輪郭線）の抽出

稜線（輪郭線）を抽出するために以下の処理を行う。

(1) 内線消去（内点消去）

スクリーンへ投影した線分のうち，内線を消去する。ここで，内線とは次のとおり。即ち，スクリーンへ投影された二つのポリゴンＰ及びＱがある辺Ｌを共有し，かつ，ＰとＱの向きが同じ（与えられた視点に対し，共に正対している状態）である時，Ｌを内線という。

ここでは，物体を表現する多面体（三角形ポリゴンの集合体）の各辺について，以下の処理を行う。

i. 処理対象となっている辺 AB を一辺とするポリゴンが一つの場合，辺 AB は稜線である。したがって，処理を終了する。辺 AB が二つのポリゴンによって共有される場合，ii）～iv）の処理を行う。

ii. 辺 AB を共有する 2 つの三角形，ABC 及び ABD について，AB と AC の外積及び AB と AD の外積，即ち，二つの三角形の法線ベクトル$\vec{a}$及び$\vec{b}$を求める。

iii. $\vec{a}$と$\vec{b}$の内積を求める。

iv. $\vec{a}$と$\vec{b}$の内積が負である場合，上に述べた定義に従い，辺 AB を内線とする。そうでない場合，辺 AB を稜線とする。

(2) **隠れ線の消去**

使用した隠線消去アルゴリズム [1] は以下のとおり。

三角形 ABC に関して，線分 PQ の観測状態を調べるため，視点 E を頂点とする三角形の各辺 AB , BC, CA を通る平面で構成された無限長のピラミッドを考える (Fig. 5.5)。

ピラミッド内の点で，視点 E より三角形 ABC の後ろにあるものは観測できない。一方，三角形の前やピラミッドの外にある点は，この三角形に関して観測可能である。

Fig. 5.5 において，線分 PQ は，2 点 I と J でピラミッドと交わる。線分 IJ は観測できず，PI と JQ は観測できる。

いま，2 点 $P(x_P, y_P, z_P), Q(x_Q, y_Q, z_Q)$ を通る直線を次式のベクトル方程式で表す。

$$EP + \lambda\vec{r} \tag{5.43}$$

ここで，$\vec{r} = (r_1, r_2, r_3) = \vec{PQ}, r1 = x_Q - x_P, r2 = y_Q - y_P, r3 = z_Q - z_P$ である。3 点 {E, A, B} を通る平面と上式で表される直線の交点に対するλをλ_1とし，同様に，{E, A, C} 及び {E, A, D} に対するλをそれぞれλ_2及びλ_3とする。

すべての i （i＝1，3）について，$\lambda_i \leq 0$ または $1 \leq \lambda_i$の時，線分 PQ は，与えられた視点 E に関し観測可能である。また，ある i について，$0 \leq \lambda_i \leq 1$ の時でも，Fig. 5.6 の場合，PQ は観測可能となる。即ち，Fig. 5.6 の場合，$0 \leq \lambda_1 \leq 1$ であるが，視点 E から PQ は観測可能である。このような場合を想定し，次の処理を行う。

いま，２点A, Bを通る直線を次式のベクトル方程式で表す。

$$\vec{EA} + \mu\vec{AB} \tag{5.44}$$

３点｛E, P, Q｝を通る平面と上式で表される直線の交点に対するμをμ_1とし，同様に，２点A, Cを通る直線に対するμ及び２点B, Cを通る直線に対するμをそれぞれμ_2及びμ_3とする。すべてのｉ（ｉ＝１，３）について，$\mu_i \leq 0$または$1 \leq \mu_i$の時，線分PQは，与えられた視点Eに関し観測可能である。

5.2.4　処理結果の出力

処理結果をファイルへ出力する。表示されている輪郭線のスクリーン座標，世界座標系の座標，線分の色を出力する。なおスクリーン座標で０～１の間にクリッピングしたものを出力する。本システムによって生成されたＪＲＲ－３M及び模擬環境に対する輪郭線画像を，それぞれFig. 5.7及びFig. 5.8に示す。

参考文献

[1] Leendert Ammeraal．Cによるグラフィックス技法，P112～133，オーム社(1993).

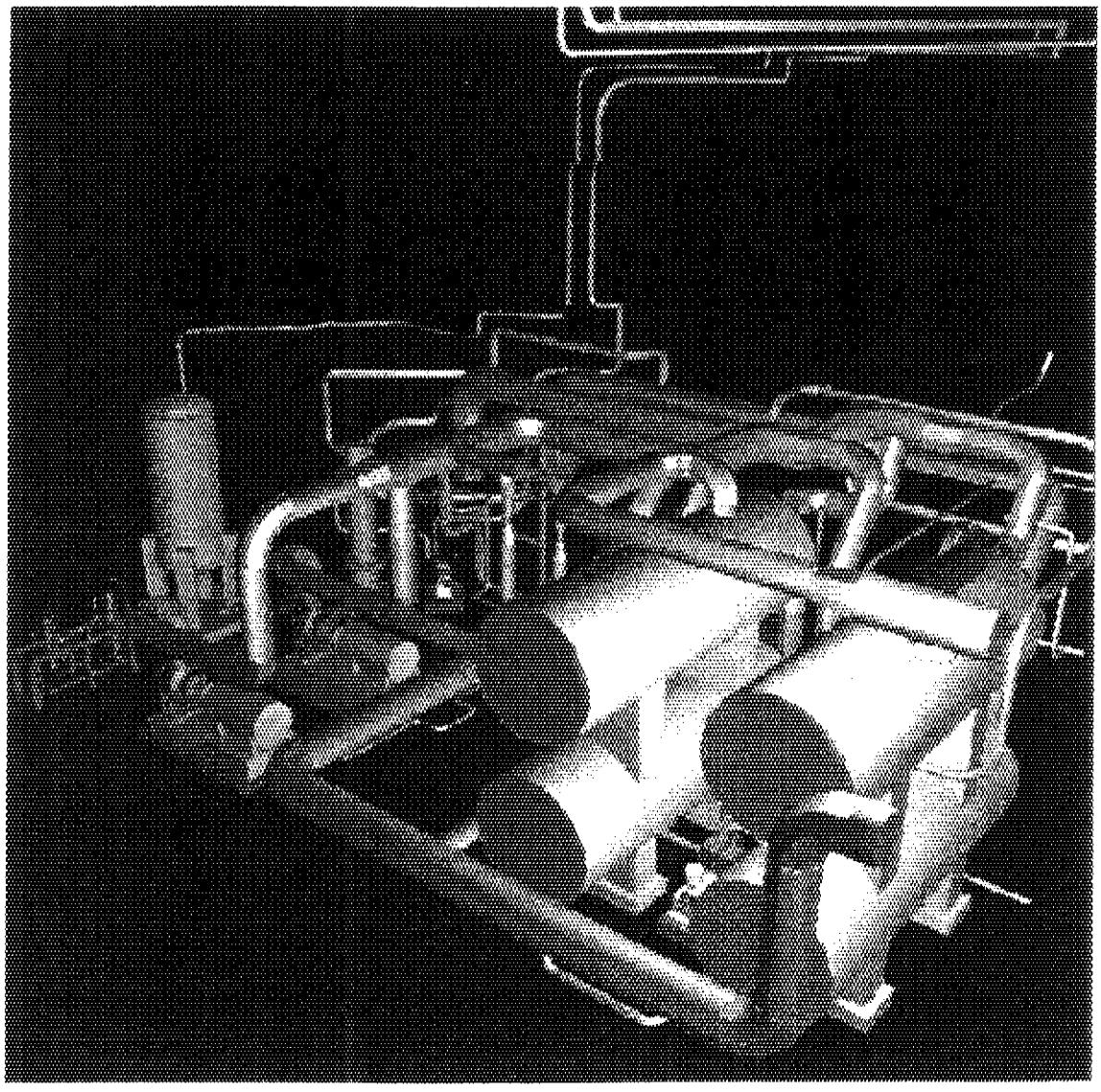


Fig. 5.1 Three dimensional image of JRR-3M.

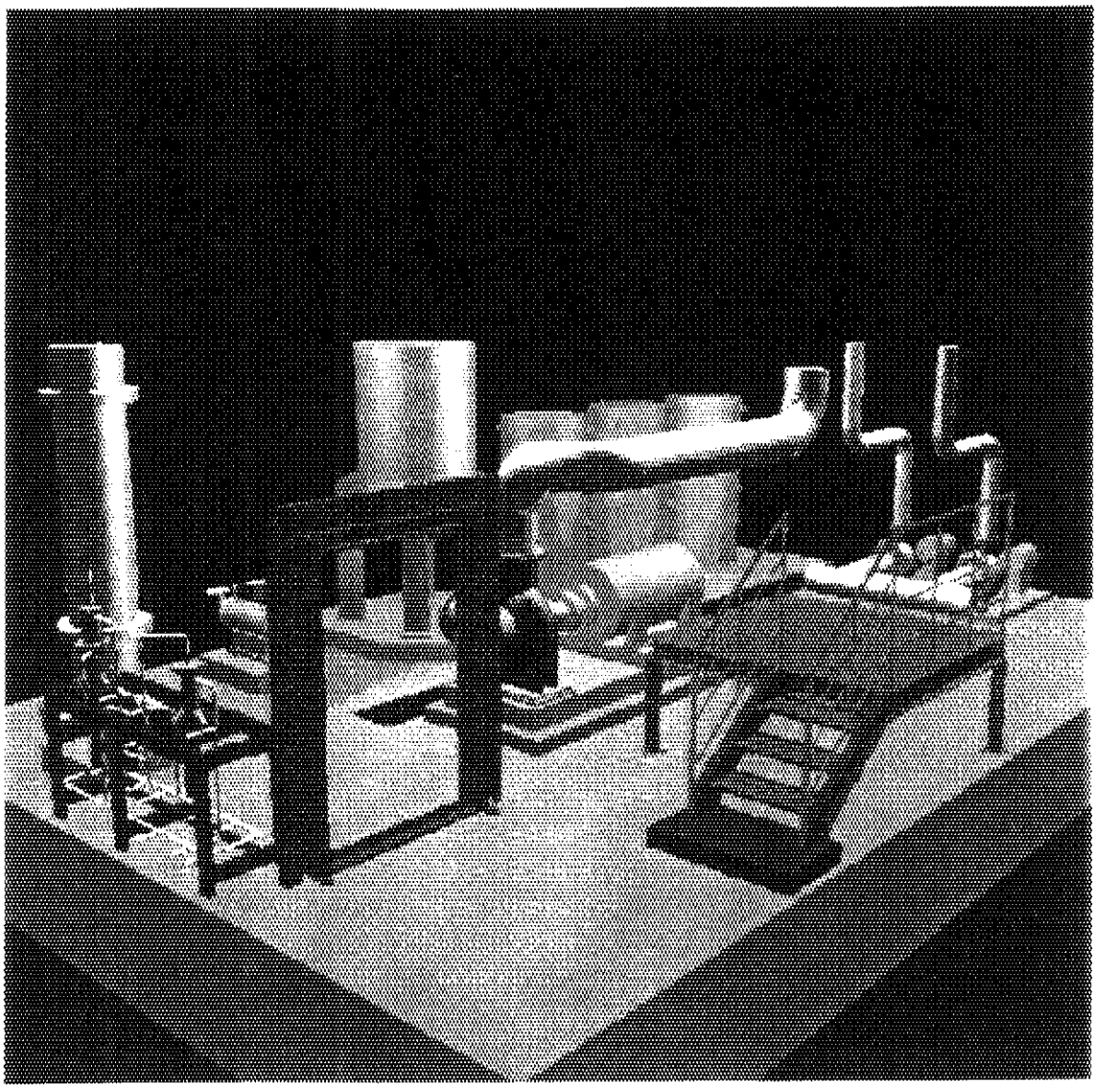


Fig. 5.2 Three dimensional image of mockup environment.

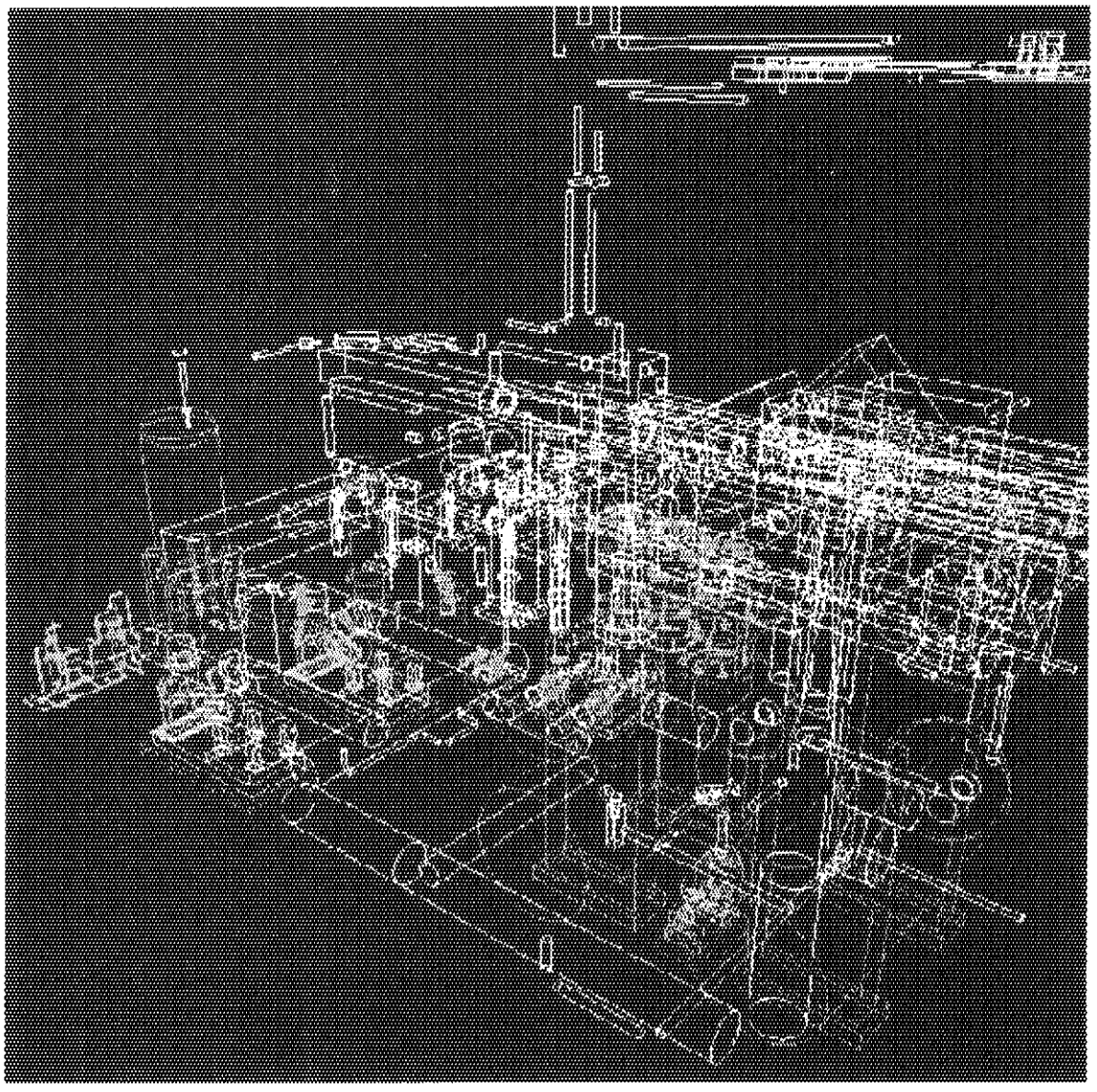


Fig. 5.3 Contour of solid primitives used in the CSG model of JRR-3M.

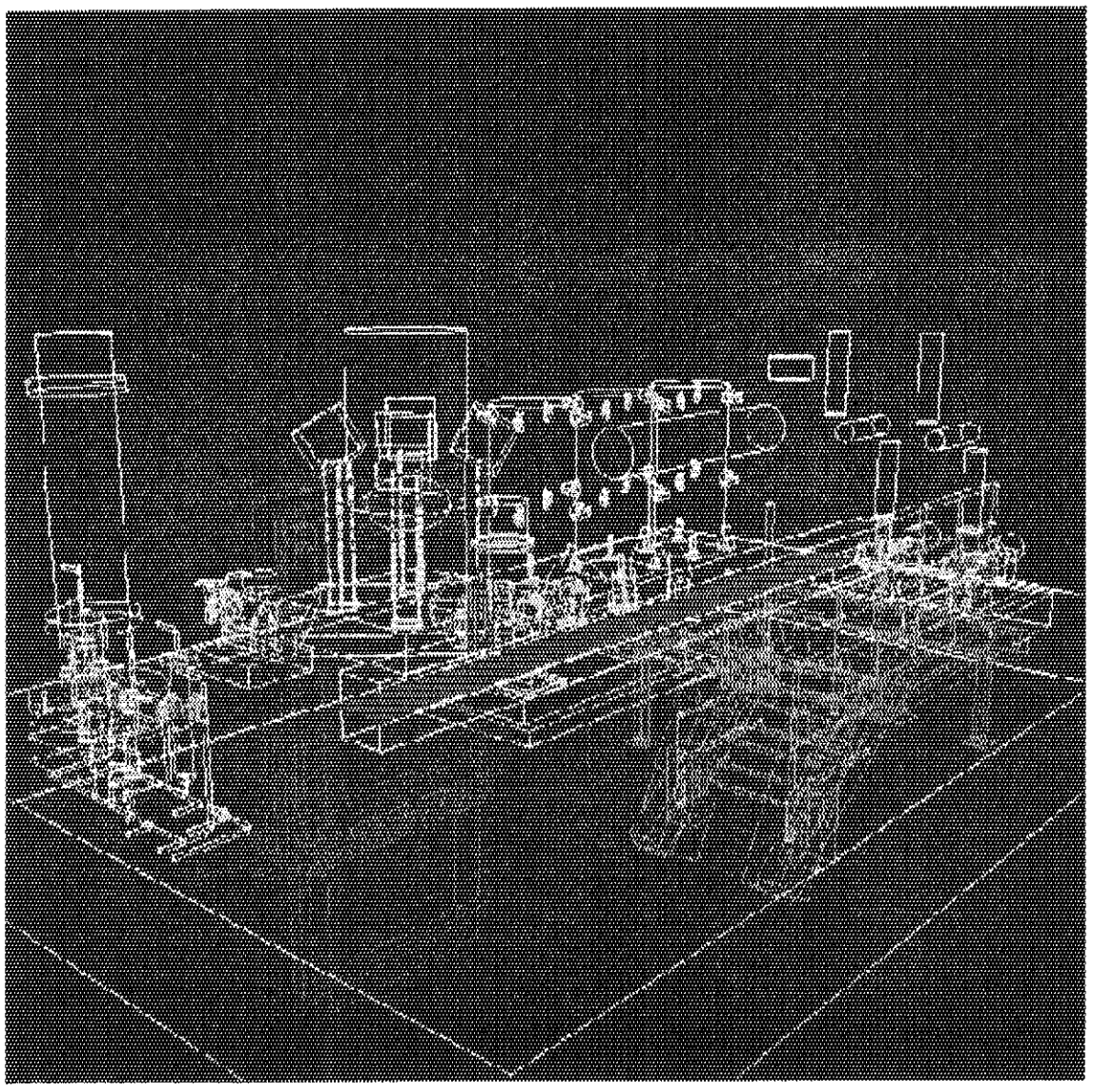


Fig. 5.4 Contour of solid primitives used in the CSG model of mockup environment.

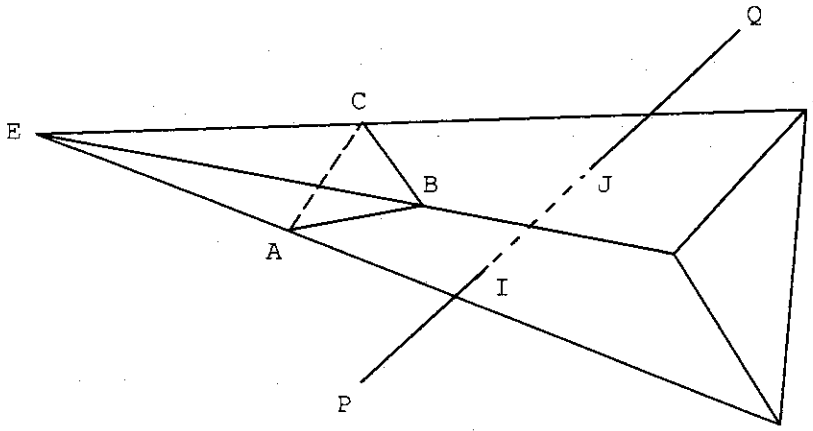


Fig. 5.5 Pyramid used in the investigation of hidden line.

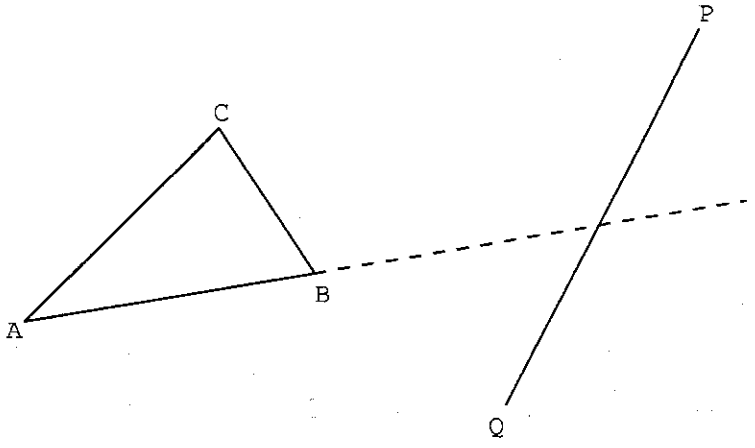


Fig. 5.6 A line segment PQ out of the pyramid.

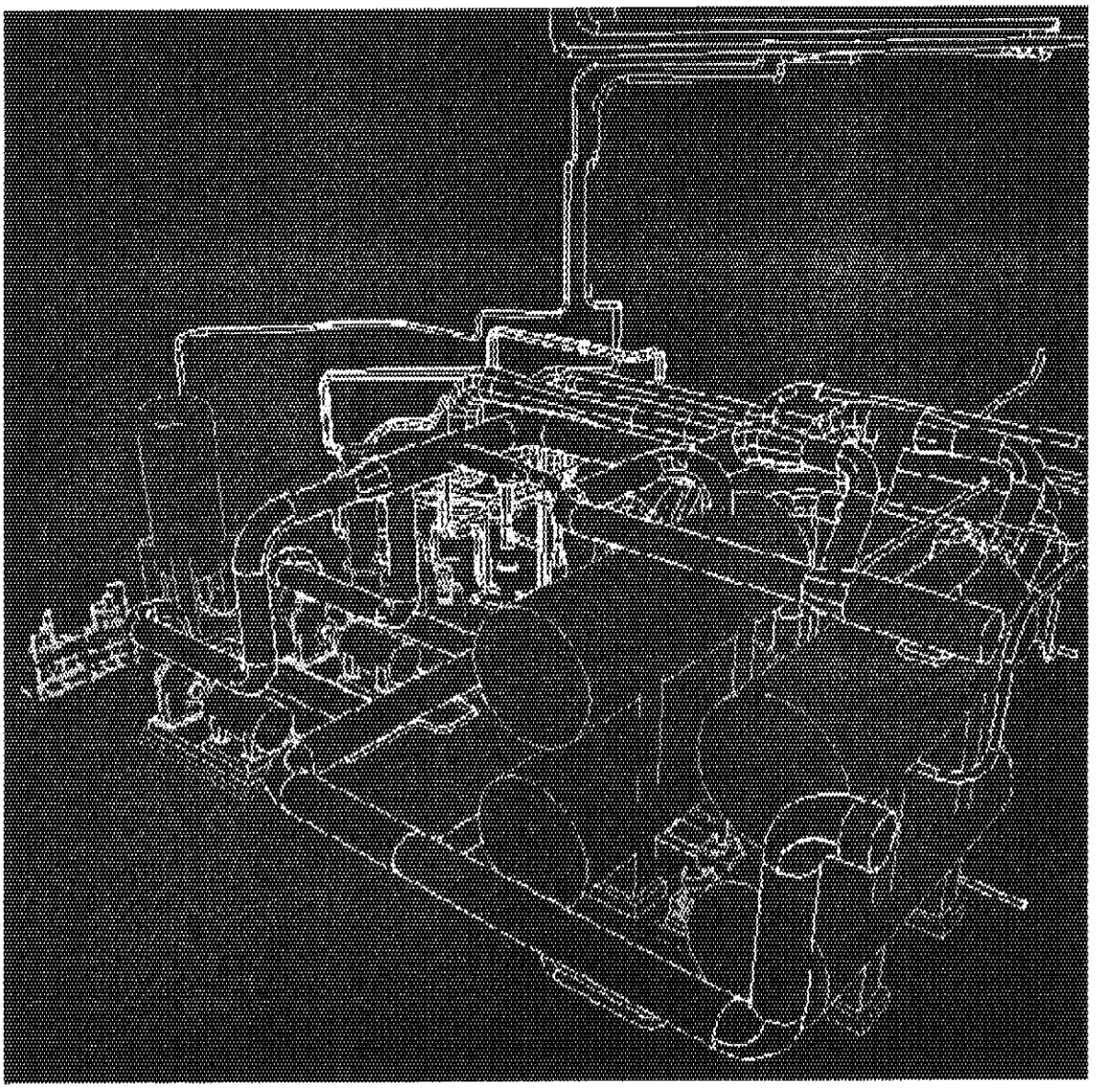


Fig. 5.7 Contour of objects in JRR-3M extracted from a set of polygon.

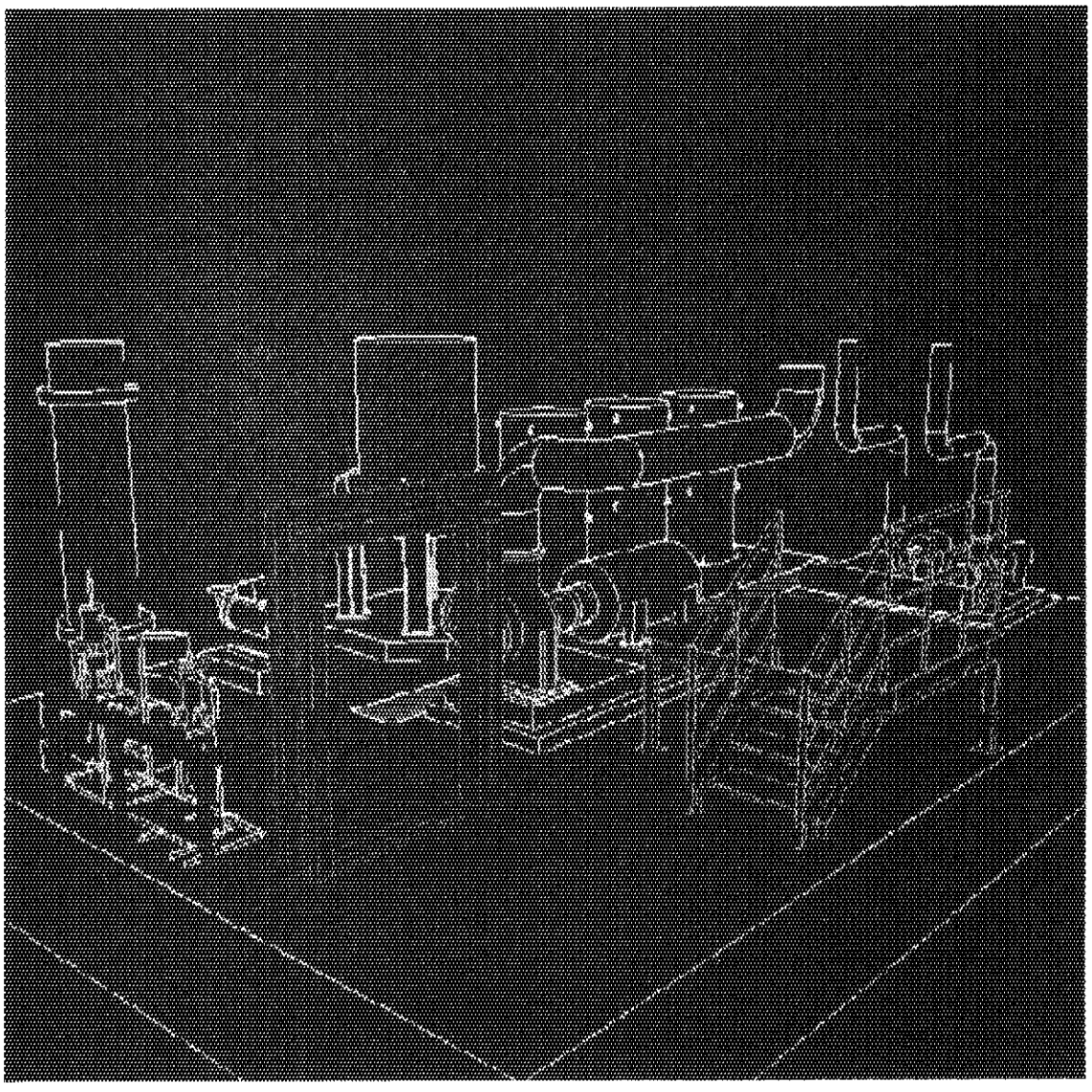


Fig. 5.8 Contour of objects in mockup environment extracted from a set of polygon.

6. 高速モンテカルロ装置

6.1 はじめに

高速モンテカルロ装置 [1] は，平成４年度に製作を完了した。前年度までの報告書において述べたように，本装置は，粒子輸送モンテカルロ・コード（以下，ＭＣコード）高速化のために開発された専用計算機である。既存のスーパーコンピュータをベース・プロセッサとして，新規開発されたハードウェアが付加され，また，既存のハードウェアの一部が改良されている。平成５年度は，本装置の利用技術開発として，本装置上のベクトル処理，マクロ・タスク機能及びマイクロ・タスク機能を用いた並列処理による粒子シミュレーション・コードの高速化手法の開発を行った。６．２節に粒子輸送モンテカルロ・コードの高速化を実現するための本装置のハードウェア上の特長を，６．３節で，本装置上におけるマクロ・タスク機能による粒子輸送モンテカルロ・コードの高速化手法を，６．４節で，MORSE コード及びem3pj コードの高速化事例を述べる。em3pj コードについては，本装置のマイクロ・タスク機能を用いた並列処理を行った。

6.2 高速モンテカルロ装置の特徴

本章では，ＭＣコード高速化のために新規開発あるいは改良された本装置のハードウェア上の特長について述べる。

1) モンテカルロ・パイプライン

本装置は，粒子輸送モンテカルロ計算において多用され，かつ，従来のベクトル計算機においてベクトル処理が困難な処理を高速化するため，モンテカルロ・パイプラインと呼ばれる３種類の特殊パイプラインを搭載している。各モンテカルロ・パイプラインの概念を以下に示す。また，各モンテカルロ・パイプラインの使用方法については，次節で述べる。

i) 幾何形状パイプライン

ＭＣコードにおいては，２次あるいは４次曲面の論理演算によって表現された３次元領域中の粒子を追跡する。この追跡計算において，粒子の現在位置から次の境界までの距離を計算する際，その領域を構成する曲面の種類によって粒子を分類する必要があるが，この処理は，ベクトル化不可能な多分岐の条件分岐文によって行われる。本パイプラインは，従来スカラ処理によって行われていたこの種のＤＯループを高速処理する。

ii) 事象分類パイプライン

ＭＣコードのベクトル処理においては，散乱や吸収といった事象ごとに粒子を集め，ベクトル処理を行う。本パイプラインは，粒子を事象によって高速に分類し，粒子バンクを作成する。

iii) 領域判定パイプライン

２次あるいは４次曲面の論理演算によって表現された３次元領域において，粒子を追跡する際，ある点がどの領域に含まれるかを判定する処理が必要である。本パイプラインは，この処理を行うための二値論理式の評価を高速に行う。

2) 強化されたロード／ストア・パイプライン

ＭＣコードのベクトル処理においては，複数の粒子を同時に追跡する。計算において，各粒子の属性データは間接番地によって参照される。ところが，間接番地参照は，連続番地参照と異なり，主記憶からベクトル・レジスタへのデータ転送が遅く，このため，演算パイプラインに遊びが生じ，ベクトル・プロセッサの性能が低下する。本装置においては，間接番地参照データの転送を行うロード／ストア・パイプラインの能力が改造前のベース・プロセッサと比較し，８倍に強化されている。

3) 並列処理機能

粒子輸送問題においては，各粒子に関する処理は独立であり，この独立性を利用した並列処理による高速化が可能である。本装置においては，４台のプロセッサを用いた並列処理が可能である。並列処理のため，サブルーチン単位で並列処理を行うマクロ・タスク機能とＤＯループを並列処理するマイクロ・タスク機能の２種類が準備されている。マクロ・タスク機能に関する並列化手法については，６．３節で述べる。また，マイクロ・タスク機能については，６．４節で述べる。

6.3 粒子輸送モンテカルロ・コードの高速化手法及び事例

本章では，高速モンテカルロ装置によるＭＣコードの高速化手法の概要を述べる。即ち，本装置のベクトル処理機能及びマクロ・タスク機能による並列処理機能による高速化手法について述べる。

1) ＭＣコードの計算フロー

既存のＭＣコードの計算フローを Fig. 6.1 に示す。扱う粒子の種類や統計値として求める物理量によって計算フローは若干異なるが，大要は，Fig. 6.1 に示すように，粒子を一つずつ取り出し，以下の処理を行う。

(a) 乱数と分布関数等線源に対する条件を用いて，粒子の初期値（位置，飛方向，エネルギー等）を決定する。

(b) 粒子の現在位置から飛方向に沿った領域の境界までの距離を計算する。

(c) 境界までの距離と粒子のパスの長さを比較し，粒子が領域の境界を越えたか否かを決定する。

(d) Flux や Energy deposition 等粒子のその領域への寄与分を計算する。

(e) 粒子が領域の境界を越えた場合，粒子が次に進入する領域を探し，(b) へ戻る。ただし，体系外へ出た場合，漏洩したものと判断し，追跡を終了する。その領域にとどまった場合，次の (f) において衝突のシミュレーションを行う。

(f) 求めた衝突点において，乱数と断面積データを用いて，衝突のシミュレーションを行う。この結果，粒子が散乱したか，二次粒子を生成した場合，さらに追跡を続けるために (b) へ戻る。粒子が吸収された場合，追跡を終了する。

2) ベクトル化手法

上で述べたように，既存のＭＣコードでは，粒子を一つずつ追跡している。粒子のループは，サブルーチン呼び出しや再帰的処理を含むため，ベクトル処理は不可能である。即ち，既存のＭＣコードにおいては，ベクトル処理可能なＤＯループが皆無である。ベクトル化においては，この計算フローを変更し，複数（数百から数千）の粒子を同時に追跡するようコードを書き換える。さらに上記 1) に示す各処理ごとに粒子を分類し，各粒子群をベクトル処理可能なＤＯループによって処理するよう書き換える。

3) モンテカルロ・パイプラインの適用

ＭＣコードを上記手法により，ベクトル化し，さらに本装置が装備したモンテカルロ・パイプラインによって高速化する手法を述べる。

i) 幾何形状パイプラインの適用

Fig. 6.2 は，ベクトル化された KENO-IV コードの一部である。このＤＯループは，粒子が現在属する領域にとどまるか，境界を越えるかを判定するため，幾何形状ごとに粒子を分類し，上記６．３．1) 中の (b) 及び (c) の処理を行っている。このＤＯループは，計算コストが高いにもかかわらず，多分岐の条件分岐文を含むため，従来のベクトル計算機上では高速化できない。この部分に本パイプラインを適用した例を Fig. 6.3 に示す。本パイプラインは，ライブラリ・コールの形式で使用する。呼び出し名は，VGSORT であり，４番目の引き数である８は，分岐の数，ＶＡは分類キー，Ｂ１～Ｂ２は分類された粒子番号の格納用配列である。この例では，本パイプラインの適用により，処理が約５倍高速化されている。

ii) 事象分類パイプラインの適用

Fig. 6.4 は，ベクトル化された KENO-IV コード [2] の一部である。このＤＯループは，粒子を二分岐の条件分岐文によって事象（処理）ごとに分類し，粒子バンクを作成している。この処理は，ベクトル化されたＭＣコードにおいて頻繁に行われる。二分岐の条件分岐文を含むＤＯループは，ベクトル処理可能であるが，条件分岐命令に対して，制御変数（マスク）を用いてベクトル・データの圧縮 (Compress) または伸長 (Expand) 等の処理を行うため，高い速度向上率は望めない。本パイプラインをこのＤＯループに適用し，さらに高速化を図る。この部分に本パイプラインを適用した例を Fig. 6.5 に示す。幾何形状パイプラインと同様，本

パイプラインは，ライブラリ・コールの形式で使用する。呼び出し名は，VESORT であり，Fig. 6.5 におけるＭＣＲＯＳは分類キー，ＣＲＯＳＢ１及びＰＯＳＢＡＮは分類された粒子番号の格納用配列である。この例では，本パイプラインの適用により，処理が約２倍高速化されている。

iii) 領域判定パイプラインの適用

Fig. 6.6 は，ベクトル化されたMCNP コード [3] の一部である。このサブルーチン，LGEVAL は，ある点が複数の曲面によって構成された三次元領域に含まれるか否かを判定する処理を行うが，計算コストが高いにもかかわらず，再帰的処理を含むため，ベクトル化が困難である。本パイプラインは，その他のモンテカルロ・パイプラインと同様，Fig. 6.7 に示すように，ライブラリ・コールの形式で使用する。呼び出し名は，LGEVAL であり，Fig. 6.7 におけるＬＧは評価される論理式の要素が格納された配列である。この例では，本パイプラインの適用により，処理が約１３倍高速化されている。

4) **並列化手法**

上記手法によって高速化したコードを，さらに，マクロ・タスク機能によって並列処理し，高速化する手法について述べる。既に述べたように，マクロ・タスク機能とは，サブルーチン単位で並列処理を行う機能である。粒子輸送問題においては，各粒子に関する処理は独立であり，この独立性を利用した並列処理による高速化が可能である。ベクトル化されたＭＣコードにおいて，粒子に関する計算部分（入力データの読み込みや断面積データの処理等逐次計算部分を除いた部分）をサブルーチン化し，並列処理を行うための関数「ptfork」,「ptjoin」等を挿入することで，マクロ・タスク機能による並列処理が実行される。この書き換えの概念的な説明を以下に示す。

i) 並列化の方針

ベクトル版ＭＣコードの並列化の方法として次の２つの方法が考えられる。

- 世代に関する計算を各プロセッサに分割して並列に実行する
- 各世代における粒子をプロセッサに分割して並列に実行する

遮蔽計算ならば上記どちらの方法でも並列化できるが，前の世代で得られた結果を次の世代で使用する臨界計算では，後者の方法で並列化するのが適切である。この場合，例えば１世代あたりの処理粒子数が４Ｎならば，各プロセッサにはＮ個の粒子が割り当てられる。

ii) 並列化の技法

(a) グローバル・コモンとローカル・コモン

スカラー処理及びベクトル処理と比較し，並列処理では，プログラムにおけるデータの割り付け方法に大きな違いがある。スカラー処理及びベクトル処理では，全てのデータが静的領域に割り付けられるが，並列処理は静的領域または動的領域に割り付けられる。並列処理でのデータの割り付け方法は，Table 6.1 のように分類することができる。すべてのプロセッサにおいて共通に使用されるデータが割り付けられるグローバル・コモンと各プロセッサに

おいて個別に使用されるデータが割り付けられるローカル・コモンの混同は，エラーを引き起こす原因となる。例えば，次の例を考える。ここでは，手続き AAA で変数 A の値を定義し，その値を手続き BBB で参照している。この場合，変数 A はローカルデータとして割り当てられているため，手続き AAA が終了した時点で解放されてしまう。従って，手続き BBB で変数 A を参照するとき，その値は手続き AAA で定義した値であるかどうかは保証されない。

```
SUBROUTINE BBB
B = A
RETURN
ENTRY AAA
A = 12345
RETURN
END
```

上記手続きを正常動作させるためには，SAVE 文を使用する（共通データとする）か，変数 A をコモンに割り付ける必要がある。並列化のためのプログラミングでは共有すべき変数を共有しない，共有すべきでない変数を共有している，あるいは共有変数の誤用している，などによってエラーを引き起こすことが多い。従って，データの割り付けられている場所及び値が保証される時間を十分理解してプログラミングする必要がある。

(b) 排他制御

複数のプロセスが同時に共有変数の値を変えようとすると，正しい結果を得ることができない場合が発生する。このような場合は排他制御を使用して，１つのプロセスのみが共有変数を変更するようにする必要がある。例えば，次のように排他的に処理したい部分をロック関数（PLLOCK と PLUNLOCK）で囲むことによって正しい結果を得ることができる。ロック変数は関数 PLASGN で割り当てられる。

```
GLOBAL COMMON /SUM/A
CALL PLASGN(LTVRAR)
CALL PLLOCK(LTVAR)
   A = A + 1
CALL PLUNLOCK(LTVAR)
```

以上，本節では，ＭＣコードをマクロ・タスク機能による並列処理の概念を述べた。マクロ・タスク並列を実現するための具体的な関数の使用法等より詳しい情報については文献 [1] を参照されたい。本節では，高速モンテカルロ装置上における粒子輸送モンテカルロ計算の高速化手法について述べた。モンテカルロ・パイプラインやマクロ・タスク機能による高速化手法は，プログラム構造やＤＯループの制御変数の大きさ（ループ長）等を良く理解した後，適用する必要がある。前述したように，この作業を行うための支援ツールとして，プログラムの動的解析を行う ANALYZER-P やプログラム最適化ツール FOPP が利用可能である。また，ベクトル・並列計算機である本装置上の高速化においては，ベクトル処理と並列処理の相乗効果を最大にする書き換えが望ましい。この点に注目したより高度な最適化手法及び粒子輸送モンテカルロ・コード以外の原子力コードを本装置のマイクロ・タスク機能によって高速化するための並列処理手法については次節で述べる。

6.4 粒子シミュレーション・コードの高速化事例

前節までに，高速モンテカルロ装置のアーキテクチャ，モンテカルロ・パイプラインの概念及び使用法，これらの機能を用いた粒子輸送モンテカルロ・コード高速化手法について述べた。本節では，実コードの高速化事例として，(1) 放射線輸送多群モンテカルロ・コード MORSE の高速処理結果，(2) マイクロタスク機能を用いた3次元相対論的粒子シミュレーション・コード em3pj の高速化手法及び処理結果について述べる。

6.4.1 放射線輸送多群モンテカルロ・コード MORSE のベクトル並列処理

MORSE コード [4] は，オークリッジ国立研究所において開発された放射線輸送多群モンテカルロ・コードである。放射線輸送問題を解くための豊富な機能が用意されている点，また，ユーザーが計算目的に応じて容易にコードの機能を拡張できるようソース・プログラムに関する詳細なドキュメントが準備されている点が大きな特徴である。性能評価では，人体模型を用いた実効線量当量計算を行った。この計算を行うために，円環体，一般楕円体及び楕円錐が MORSE コードのＣＧパッケージに追加された [5]。この計算においては，あるエネルギーのγ線の平行ビームを模擬する百万個の粒子が追跡される。並列処理は，KENO-IV や MCNP コードと同様，同じ数の粒子を各プロセッサへ分配することで実現した。Table 6.2 に性能評価結果を示す。ベクトル処理における速度向上率は，３．４倍である。このコードについては，(1) 主記憶－ベクトル・プロセッサ間の間接番地参照データ転送能力を強化された高速モンテカルロ装置のロード／ストア・パイプラインと (2) ベクトル長の低いモンテカルロ・コードのベクトル処理のために改善されたコンパイラが効果的であった。モンテカルロ・コードのベクトル処理においては，計算当初，数多くの粒子が体系に存在し，従ってベクトル長（粒子数）も大きいが，計算が進むにつれ，粒子は漏洩あるいは吸収されるため，ベクトル長は減少する。徐々にあるいは急激に減少し，短くなるベクトル長は避けられない。短いベクトル長によるベクトル処理は，ベクトル処理の性能低下を引き起こす。最悪のケースでは，ベクトル長が非常に短いためベクトル処理によってスカラ処理よりも大きな実行時間を要する場合もある。高速モンテカルロ装置においては，ベクトル長が4以下の場合，スカラ処理を行い，ベクトル長が5以上の場合はベクトル処理を行うコンパイラ・オプションが用意されている。性能評価に用いた問題において，コスト比率の高いあるＤＯループは，このオプションによって，約８．２％の実行時間が短縮された。また，コード全体では，同じく，このオプションによって，約７．６％の実行時間が短縮された。

並列処理による速度向上率は，２．６倍である。この問題における並列化率は，９９％であるが，メモリ競合及び並列処理時に生じるオーバヘッドのために，性能が低下した。

6.4.2 3次元相対論的粒子シミュレーション・コード em3pj の並列ベクトル処理

本節では，高速モンテカルロ装置のマイクロ・タスク機能によって並列ベクトル処理を行った粒子シミュレーション・コードの高速化事例について述べる。マイクロ・タスク機能は，ＤＯループや文の集まりを並列処理する機能であり，再帰的処理や関数及びサブルーチン呼び

出しを含むためにベクトル処理が不可能なＤＯループも高速処理可能である。em3pj は，相対論的なプラズマ及びビームの自己無矛盾な挙動を解析するための３次元コードであり，粒子と電磁場の非線型な相互作用を粒子モデルによりシミュレーションする。使用した入力データにおいては，場の電荷密度及び電流密度を計算する４つの多重ループと粒子に働く電磁力を計算する１つの多重ループに９０％以上の計算コストが集中にしている。コードの高速化においては，１６×１６×１６×４０９６程度（Ｘ，Ｙ及びＺ方向のセル数×粒子数）のループ長を持つこれらのループのうち，最外側のループをマイクロ・タスク機能によって並列処理するよう指示行を挿入した他，高速化のためにスカラ処理における最適化，及びベクトル処理における最適化として，(1) 外側ループのアンローリング，(2) バンク・コンフリクトの回避を行った。高速化の内容及び速度向上率に対する考察を以下に述べる。

1) スカラ処理における最適化

スカラ処理における最適化の主な内容は，キャッシュ・ミスヒットの回避である。このコードのスカラ処理では，キャッシュ・ミスヒットが多く，このため，多大の実行時間を要していた。通常，キャッシュは，スカラ処理時のデータの高速アクセスを実現するために用いられる。キャッシュ上に置かれていないデータを用いた場合，実行時間は増大する。キャッシュ上に置くデータのスケジューリングについては，コンピュータ・メーカーあるいは装備しているキャッシュの大きさによってさまざまな戦略があるため，キャッシュ・ミスヒットの回避法については，残念ながら，明解な手法は存在しない。このコードの場合，配列の大きさを変更することで，良好な結果を得ることができたが，その詳細な説明については，煩雑なため本稿では避ける。

2) ベクトル処理における最適化

ベクトル処理における最適化の主な内容は，外側ループのアンローリング，及びバンク・コンフリクトの回避である。

i) 外側ループのアンローリング

外側ループのアンローリングは，ベクトル処理において，主記憶－ベクトル・プロセッサ間のデータのロード／ストアを減らすと共に，メモリ競合を緩和するために行った。ループのアンローリングは，以下に示す f77m4 のコンパイル・オプションまたはベクトル化指示行によって自動的に行うことができる。

コンパイル・オプション； -O unroll=4level

ベクトル化指示行　　； *vdir unroll=4

ループのアンローリングの例を以下に示す。

```
do i=1,L
  do j=1,M
    do k=1,N
      f(k,j)=f(k,j) + ftmp(k,j,i)
    enddo
  enddo
enddo
```

このＤＯループについて，最外側の i で回るループについてアンローリングを行うと，

```
do i=1,L,4
  do j=1,M
    do k=1,N
      f(k,j)=f(k,j) + ftmp(k,j,i  )
                    + ftmp(k,j,i+1)
                    + ftmp(k,j,i+2)
                    + ftmp(k,j,i+3)
    enddo
  enddo
enddo
```

となる。この結果，f(k,j) のロード及びストア回数が１／４になる，と共にメモリ競合を緩和することができる。また，ループ内の計算量も増加する。

ii) バンク・コンフリクトの回避

このコードでは，ベクトル処理における配列参照において，バンク・コンフリクトが生じていた。スーパーコンピュータにおいては，主記憶への高速アクセスを可能にするため，主記憶をバンクと呼ばれるいくつかの小領域に分割し，各バンクへの同時アクセスを可能にしている。Monte-4 の主記憶は，５１２のバンクによって構成される。ところが，このコードにおいては，２のべきの大きさ（＞５１２）の配列を２のべき乗とびに参照しているため，同じバンクへのデータ参照が競合するバンク・コンフリクトが生じていた。そこで，配列のサイズを２ｎ＋１の形式に書き換え，同じバンクへの同時アクセスを避けた。この手法により，バンク・コンフリクトはほぼ解消された。

3) 速度向上率に対する考察

１プロセッサを用いた場合のスカラ処理及びベクトル処理における計算時間及び４プロセッサを用いて並列処理を行った場合の各プロセッサにおける実行時間を表 Table 6.3 に示す。この結果を用いて，同コードの並列処理における速度向上率を解析した。以下に，高速化についての解析結果を示す。スカラ処理における最適化により，約 1,400 秒のコードの実行時間が約 1,000 秒削減された。この大幅な計算時間の減少は，前述したように配列サイズの変更によるキャッシュミスヒットの減少にある。１プロセッサ上のベクトル処理においては，約４０倍の速度向上率を得た。この高い速度向上率は，高いベクトル化率と長いベクトル長に依るものである。並列処理においては，１．８倍の速度向上を得た。本コードの場合，ベクトル処理による速度向上率が高いために，即ち，並列処理部分が既に高速化されているために，並列処理による速度向上率は低い。並列化率は，１プロセッサにおける計算時間をＡ，このうち並列処理される部分の計算時間をＢとすると，Ｂ／Ａである。この問題におけるコードの並列化率は，８３％であった。この値に対する理想的な並列処理による速度向上率は，１００／（（１００－８３）＋８３／４）＝２．６である。しかし，実際に得られた速度向上率は，１０．２／５．７＝１．８である。後者が前者より低い理由は，次のとおり。

i) 並列処理を行う際，タスク生成，マイクロ・タスク手続の起動，同期・排他制御及びループの並列実行の制御のためのオーバーヘッドが生じる。

ii) このコードにおいては，主記憶－ベクトル・プロセッサ間のデータ転送が多く，メモリ・コンフリクトにより性能が低下する。

iii) 並列処理時，キャッシュ・ミスヒットが増大する。これは，並列処理時のデータ管理のために避けられぬ現象である。

本コードの場合は，メモリ負荷を減らすための最適化も含め，Monte-4 に適したコーディングにより，オリジナル・コードのスカラ処理と比較し，約２５０倍という極めて高い速度向上率が得られた。

6.5 おわりに

３つの粒子輸送モンテカルロ・コード及び３次元相対論的粒子シミュレーション・コードの高速モンテカルロ装置上の高速化事例を参考に，本装置のベクトル処理機能及び並列処理機能を用いた高速化手法を述べた。並列処理のための高速化手法は，ベクトル処理のための高速化手法と同様，計算コードの構造あるいは計算処理の内容に大きく依存する。また，運用が開始されたばかりの本装置における並列化の研究については，まだ不十分なため，本稿で述べた手法は,「一般化された手法」とは言い難い。今後，メモリ共有型のベクトル並列計算機である本装置の利用技術開発を進める。

参考文献

[1] K. Asai, K. Higuchi, M. Akimoto, H. Matsumoto, Y. Seo. JAERI Monte Carlo Machine, Proc. of Joint Internat. Conf. on Mathematical Methods and Supercomputing in Nucl. Applications (1993).

[2] K. Asai, K. Higuchi, J. Katakura, Y Kurita. Vectorization of KENO IV Code and an Estimate of Vector-parallel Processing, JAERI-M 86-151 (1986).

[3] 栗田 豊, 菅沼 正之, 樋口 健二. モンテカルロ・コード MCNP のベクトル化, JAERI-M 87-022 (1987).

[4] M. B. Emmett, The MORSE Monte Carlo radiation transport code system, ORNL 4972 (1975).

[5] Y. Yamaguchi. DEEP code to calculate dose equivalents in human phantom for external photon exposure by Monte Carlo method, JAERI-M, 90-235 (1991).

Table 6.1 Datatype in parallel processing.

分　類	宣　言	有効範囲
共通データ	global common , common	プログラム内
タスク内共通データ	local common	タスク内
ローカルデータ	宣言なし	手続き内（サブルーチン）

Table 6.2 Execution time of MORSE code on Monte Carlo machine.

プロセッサ数	1		2	3	4
処理モード	スカラ	ベクトル	ベクトル	ベクトル	ベクトル
処理時間（秒）	374.0	107.6	60.6	47.7	40.7
速度向上率（A）	1.0	3.4	6.1	7.8	9.1
速度向上率（B）	－－－	1.0	1.7	2.2	2.6

注1）表における 374 秒は，オリジナル・コードのスカラー処理時間を示す。
注2）速度向上率Aは，オリジナル・コードのスカラー処理時間に対する速度向上率を，
　　同じくBは，1プロセッサを用いたベクトル処理時間に対する速度向上率を示す。
注3）表における 107.6 秒はベクトル版コードのベクトル処理時間を示す。

Table 6.3 Execution time of em3pj code on Monte Carlo machine.

コードのバージョン	オリジナル	ベクトル		ベクトル並列
プロセッサ台数	1	1	1	4
実行モード	スカラ	スカラ	ベクトル	ベクトル並列
実行時間	1415.9 秒	433.7 秒	10.2 秒	5.7 秒
速度向上率	1.0 倍	3.3 倍	138.8 倍	248.4 倍

注）ベクトル版は，スカラ及びベクトル処理のために最適化されている。

```
      DO 100 IV=1,NPOS
      JV=POSBAN(IV)
      IGEO(JV)= IGEOM(K(JV))
      GO TO(110,120,130,140,150,160,170,180),IGEO(JV)
  110 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 200
       GO TO 300
  120
            .
            .
            .
            .
  200 NXSEC=NXSEC+1
      XSECBA(NXSEC)=JV
      GO TO 100
  300 NCROS2=NCROS2+1
      CROSB2(NCROS2)=JV
  100 CONTINUE
```

Fig. 6.1 FORTRAN statements of KENO-IV code classifying particles by geometries.

```
      len1 = 0
      len2 = 0
      len3 = 0
      len4 = 0
      len5 = 0
      len6 = 0
      len7 = 0
      len8 = 0
      DO 100 IV=1,NPOS
      JV = POSBAN(IV)
      VA(IV) = IGEOM(K_(JV))
  100 CONTINUE

      CALL VGSORT8(VA,POSBAN,NPOS,8,B1,LEN1,B2,LEN2,B3,LEN3,
     & B4,LEN4,B5,LEN5,B6,LEN6,B7,LEN7,B8,LEN8)

*VDIR NODEP
      DO 110 IV=1,LEN2
         jv=b2(iv)
         RSQ=X1(jv)*X1(jv)+Y1(jv)*Y1(jv)
         IF(Z1(jv).LE.XX(K(jv),2)
     1      .AND.Z1(jv).GE.XX(K(jv),3)
     2      .AND.RSQ.LE. XX(K(jv),4)) then
           nxsec=nxsec+1
           xsecba(nxsec)=jv
         else
           ncros2=ncros2+1
           crosb2(ncros2)=jv
         endif
  110 CONTINUE
           .
           .
           .
           .
```

Fig. 6.2 Example of application of the geometric pipeline to KENO-IV code.

```
*VDIR NODEP
      DO 100 IV=1,NCROS1
      JV=CROSB1(IV)
      IF(MCROS(JV).NE.0) THEN
        ICROS1=ICROS1+1
        CROSB1(ICROS1)=JV
      ELSE
        NPOS=NPOS+1
        POSBAN(NPOS)=JV
      ENDIF
  100 CONTINUE
```

Fig. 6.3 FORTRAN statements of KENO-IV code classifying particles by event type.

```
      CALL VESORT(NCROS1,CROSB1,MCROS,ICROS1,CROSB1,NPOS,POSBAN)
```

Fig. 6.4 Example of application of the event pipeline.

```
C            LGEVAL
      FUNCTION LGEVAL(LG,N)
C        EVALUATE LOGICAL EXPRESSION LG OF LENGTH N.
C        REPRESENTATION OF LOGICAL ELEMENTS:
C           (       100001       TRUE           1
C           )       100002       FALSE          0
C           UNION   100003       INTERSECTION   IMPLICIT
      DIMENSION LG(*)
C
      LG(N+1)=100002
      I=0
      L=0
   10 LGEVAL=1
   20 I=I+1
      IF(LG(I).GT.1)GO TO 40
      LGEVAL=MIN(LGEVAL,LG(I))
      IF(LGEVAL.NE.0)GO TO 20
   30 I=I+1
      IF(LG(I).LT.2)GO TO 30
   40 IF(LG(I).NE.100003)GO TO 50
      IF(LGEVAL.EQ.0)GO TO 10
      IF(L.EQ.0)RETURN
      GO TO 60
   50 IF(LG(I).NE.100001)GO TO 80
      L=L+1
      IF(LGEVAL.NE.0)GO TO 20
   60 M=1
   70 I=I+1
      IF(LG(I).EQ.100001)M=M+1
      IF(LG(I).EQ.100002)M=M-1
      IF(M.NE.0)GO TO 70
   80 L=L-1
      IF(I.NE.N+1)GO TO 20
      RETURN
      END
```

Fig. 6.5 FORTRAN statements of MCNP code involving region check procedure.

```
J_(II)=1-LGEVAL(LGC_(IS),N)
```

Fig. 6.6 Example of application of the region check pipeline.

7. シミュレーションのシステム統合

7.1 はじめに

これまで情報システムセンターでは，ＨＡＳＰにおいて，自然言語で書かれた命令を理解し，自己の行動計画（動作列）を生成するシステム，環境の情報を認識・判断するシステム，２足ロボットの動歩行シミュレーションシステム，環境（世界モデル）設定・可視化システムといった，知能ロボットの行動を個別に模擬するソフトウェア群の開発を行ってきた。すなわち，(1) 命令理解システム，(2) 環境認識システム，(3) ２足歩行シミュレーションシステム，(4) 施設形状データベースおよび (5) 映像生成システムである。

これらの個別システムは，将来的にはロボットハードウェアに搭載されて，その行動を制御するために用いられることとなるが，このロボットハードウェアの実際の行動に至るまでの一連の内的および外的な振る舞いを統合的に模擬するシステムを構築することは，人工知能研究のワークベンチとして，あるいはハードウェアとしてのロボットシステムの評価・検討という視点からも極めて重要であると考える。一連のロボット動作のシミュレーションを行うためには，これまでに開発した個別システムをサブシステムとしてワークステーション群および高速モンテカルロ装置上に統合化した全体システム（以下，統合的シミュレーションシステムと呼ぶ）を構築することが必要である。

本章では，シミュレーションのシステム統合におけるシステム構成，各サブシステム間のインタフェース，試作したプロトタイプシステムによるシミュレーション結果および今後の課題等について記述する。

7.2 システム構成

本節では，シミュレーションシステムの構成について記述する。はじめに統合的シミュレーションシステムが持つべき基本的な機能について考察する。次にサブシステムをどのように組み合わせてシミュレーションシステムを構築するかを示す。

7.2.1 基本機能

統合的シミュレーションシステムにおいては，知能ロボットは，例えば「ＰＣＳ補助ポンプを点検しろ。」という命令を与えられると，施設形状データベースを検索することにより，ＰＣＳ補助ポンプの位置を確認し，そこに至るまでの経路を確定し，ロボットの基本性能をもとに全行程の歩容を求める（初期フェーズ：命令理解，静的経路探索，ロボット動作計画）。

そして，確定された歩容に基づいて行動を開始する（行動フェーズ）。行動を開始したロボットは，環境を観測し，自己の位置が計画された行動計画と一致していることを確認しながら移動する（ロボット動力学，自己位置認識）。行動計画からのズレが検出されると，アク

チュエータのトルクを変更すること等により，ズレを修正する。

しかし，ある局面では，このように行動計画に沿って行動するのではなく，行動計画を修正する必要が生じる。すなわち，突然の命令変更（割り込み受付）や環境の異変の検知である。このような場合には，ロボットはいったん動作を停止するとともにオペレータ（人間）にこれを通報し，オペレータの補助を受けて，行動計画を立て直す。

これらのロボットの行動が常にディスプレイに表示されるばかりでなく，ロボットに搭載されたセンサの情報（ロボットから見た環境の映像，臭気，温度，湿度，音波）についても随時表示され，また，ロボット躯体の受ける放射線量も計算される。

以上のシミュレーションで必要となる機能（サブシステム）をまとめると以下のようになる。

i. 命令理解システム

自然言語で入力された命令を解析し，必要なロボットの動作列を構成するシステム。このシステムは，ロボットの行動とともに働き，ロボットの動き，環境の変化に対して動的に動作命令を作成することも考えられるが，それには高度な環境認識および矛盾のない世界モデル自動更新といった研究開発要素が含まれるため，ここでは，静的に命令を解析し，ロボットが動作を始める前にあらかじめ必要な動作列を作成する機能を持つものとする。

ii. ロボット位置認識システム

ロボットに取り付けられたセンサ（傾斜計等）を用い，環境内のランドマークからロボット自身の位置および姿勢（方向）を決定するシステム。ロボット自身の動作の初期位置および履歴を記録することによりロボット位置を計算することができるが，それだけでは誤差が累積していくため，このような機能が必要となる。

iii. 静的経路探索システム

ロボットの現在位置と目標位置から施設形状データベースを用いて，具体的なロボットの移動経路を算定するシステム。このときロボットの動作特性に関するデータ（例えば，最小回転半径，あるいは曲がるためには一旦立ち止まる必要がある等）また，経路の路面情報（経路のどこに階段があり，それは登るのか，下るのか等）も含む必要がある。

iv. 詳細動作列作成システム

静的経路探索システムによって得られた経路情報をもとにして，ロボットの歩容を作成する。以後，このシステムで作成された歩容データに従ってロボットは行動する。

v. ロボット動力学システム

詳細動作列作成システムにより作られた歩容にしたがって，実際のロボットの動作をシミュレートするシステム。ロボット躯体の基本仕様データ（寸法，重量，関節自由度等）もこのシステムに含まれる。

vi. 環境異常認識システム

環境の形状変化，異音，異臭，温度変化等の環境異常を検出するシステム。環境内に異常を検知した場合は，動作を停止し，ただちにオペレータ（人間）に通報する。異常認識の高度化，すなわち異常の原因の同定等については今後の課題とする。

vii. 動的経路探索システム

あらかじめ計画した経路をはずれた場合，たとえば自己位置認識の結果ロボット自身が予定の位置から大きくはずれていることが判明した場合等に，新たに経路を探索するシステム。このシステムは，自己位置認識あるいは環境異常認識の結果をトリガとして経路探索システムを呼び出すことで実現できる。ただし，予定経路上の障害物など環境の変化をトリガとする場合には，施設形状データベースの動的な変更機能が必要になる。また，オペレータからの命令の変更により経路探索をやり直す場合も考えられる。

viii. 映像表示システム

施設形状データベースを用いて環境の映像を３次元的に表示するシステム。このシステムを用いて環境内のロボットの動作を確認したり，ロボットから見た環境の映像（ロボットに搭載されたカメラの映像）を表示する事ができる。

ix. 被曝線量評価システム

環境内の動作中にロボット躯体が受ける放射線量を計算するシステム。

ここにリストアップされた個別の機能を基本的にあるいは部分的に有するソフトウェアシステムについては，既にこれまでに情報システムセンターにおいて開発されており，また，これまでのＨＡＳＰの作業報告，あるいは本報告に記述されているので，ここではこれ以上の説明は省略する。

7.2.2　システムの全体構成

システムは，Fig.7.1に示すように３つのパート，すなわち環境系，思考系および行動・センサ系からなる。環境系では，ロボットのおかれる環境に関するデータベースをもとにして環境に関する知識ベース，経路探索用の環境モデルを生成し，また，映像表示を高速に行うためのデータを生成する（オフライン作業）。映像表示システムによる環境（ロボット含む）の映像化はオンラインでほぼ実時間で行われる。思考系は，オペレータから日本語による命令を受け，命令理解システムによりそれを解析し，経路探索システムを用いてロボットの動作列を生成する。行動・センサ系は実際のロボットの動作をシミュレートしたり，環境の異変を感知したり，ロボットの位置を認識したりする。

これらのモジュールの中で，最も重要な位置を占めるモジュールは思考系である。これは，この系がロボットに対する具体的な命令列を作り出すためである。したがって，行動・センサ系は自己の位置や環境異常の有無を常に報告しなくてはならない。また，思考系はその報告に基づいて適切な命令列を生成し，それを行動・センサ系に伝えなければならない。この点において，思考系は，クライアント・サーバ・システムにおけるサーバの役割を果たしていると考えられる。

環境系は，環境に関するすべてのデータベースを管理し，環境データに変更があった場合は，経路探索用環境モデル，映像表示用環境データ等を更新しなくてはならない。環境系のこの動作は現在のところオフラインで行われ，ひとたびシミュレーションが開始されると，環境系は映像表示システムとして機能する。この点において，環境系データベースの動的（オンライン）変更機能の開発による動的経路探索，環境異常認識の高度化が今後の課題として挙げられる。環境系データベースをオンラインで変更する場合には，そのトリガは思考系からの命令となる。

また，統合的シミュレーションシステムで常に思考系を通して命令およびデータをやりとりすることにしておくと，命令およびデータ等の時間順序性（chronology）を保つことが容易になる。

以上から，思考系をサーバとして，各システムは常に思考系のみと通信することとしている。思考系は，命令バッファを備えており，各システムからのメッセージやオペレータからの新しい命令をバッファ中に蓄える。思考系はバッファから随時メッセージあるいは命令を読み込み，それに対する応答動作を決定し，対応するシステムに送信する。このようにして，思考系はシステム全体の状態を常に把握するようにしている。これは，行動・センサ系をインタラクティブに制御しようとするものである。

7.2.3 知識ベースの分担

思考系がシステムのすべての状態を把握するといっても，何から何まですべて思考系で処理して，行動・センサ系に命令を与えるわけではない。

たとえばロボット動力学システムを考えると，ロボットの姿勢を制御するためにはどのアクチュエータにどれだけのトルクを与えればよいかといった運動学固有の知識がある。姿勢制御はロボット動力学システム固有の機能であり，思考系はこのような各システム固有の知識まで備える必要はない。したがって，知識ベースは，必ずしも思考系にのみ集中することはなく，Fig.7.1に示したように各システムにおいて固有の知識ベースを備えることになる。すなわち，思考系はシステム全体の行動を規定する大局的な判断を下す人工知能であり，各システム固有の知能はそのシステム内にある。したがって，新たな機能（システム）が追加された場合にも固有の知識ベースはそのシステム内に追加され，思考系の知識ベースの変更は最小限にとどめられる。

7.2.4 時間の概念

一般にシミュレーションシステムにおいて時間の概念は重要である。ただし，それが実時間に即した時間概念でなければならないかどうかは，システムに依存する。本シミュレーションシステムにおいては，厳密な時間の概念が必要なのはロボット動力学システムのみである。思考系においても命令列の chronology を保つための時間の概念が必要となるが，この時間概念はロボット動力学システムで求められるような絶対時間ではなく，この事象はあの事象より前であるといった感覚的時間概念で充分である。環境異常認識システムも絶対時間とは無関係なタイムステップを有するのみであり，映像表示システムおよび位置認識システムは瞬

時的で時間の流れに無関係である。

思考系で必要とされる感覚的時間は命令列のバッファがFIFO（first-in first-out）であることによって保証される。バッファがFIFOならば，先に到着した命令は必ず先に処理され，chronologyは保たれる。これに対し，ロボット動力学システムでは，速度および加速度が必要となるため，絶対的な時間スケールが必要となる。したがって，時間のカウントはロボット動力学システムで行い，その時刻を他のシステムに知らしめることとする。このため，ロボット動力学システムはロボットの位置情報に時刻を付加して思考系に報告するものとする。

7.2.5 初期フェーズ

シミュレーションシステムの働きは，大きく，ロボット躯体が動作を開始する以前の働き（初期フェーズ）と動作中の各ステップでの働き（行動フェーズ）に分かれている。

初期フェーズでは，Fig.7.2に示すようにオペレータから与えられた命令（日本語）を解析し，必要となる詳細な動作列を生成する。

i. 命令理解

自然言語処理により命令を理解し，必要な動作列を生成する。移動に関しては，移動目的位置を算出する（例えば，Z地点に行くことによりこの命令は達成される）。

ii. ロボットの初期位置および姿勢の検出

ロボット位置認識システム（ロボットビジョン）とロボットに具備されたセンサ（傾斜計等）を用いて，ロボットの位置および姿勢を検出しておく。これが後の行動計画作成の基点となる（例えば，A地点でロボットは直立している）。

iii. 行動計画および経路探索

命令理解により生成された動作を実行するための詳細な動作列を生成し，静的経路探索によって詳細な（直進，回転，跨ぎ等に分解された）移動経路を生成する（例えば，A地点からZ地点へ移動するためには，A地点からB地点へ直進し，そこで30度回転して，C地点まで直進した後静止して，高さ約10cm，幅約20cmの物体を跨いで，さらに45度回転し，Z地点まで直進する）。

こうして生成された命令列は，一挙に行動・センサ系に与えられるのではなく，1つの命令ごとに与えられていく。

7.2.6 行動フェーズ

ロボットの行動中は，システムは以下のような作業を繰り返す。

i. ロボット動力学に基づく移動

一つの移動命令に対して安定に歩行できる歩容を計算し，その詳細な歩容データに基づいてロボットを移動させる。また，一定の時間ごとにロボットの位置（姿勢）を思考系に報告する。

ii. ロボット動作の表示

環境映像表示画面（ロボットを含む）を更新する。

iii. エラーの検出

安定な歩容が見つけられない等エラーがあった場合，これを思考系に通報し，思考系からの指示を待つ。

iv. 自己位置の検出

自己位置検出システム（ロボットビジョン）により，自己位置を検出する。検出のタイミングは思考系からの命令による。

v. 環境異常の検出

環境に異常が検出された場合，思考系に通報し，新たな命令を待つ。

vi. 割り込み命令

行動中に割り込み命令があった場合，これまでの動作を停止し，新たな命令に従う。

vii. 被曝線量評価

ロボットの位置（姿勢）を入力として被曝線量の計算を行う。ただし，この部分については，現在のところ，まだ統合化がなされていない。このため，現状では何の計算もしないダミーのサブシステムと思考系が情報の交換だけをしている。

Fig.7.3に行動フェーズのチャート図を示す。

7.2.7 サブシステム間のインタフェース

統合的シミュレーションシステムでは，それぞれのサブシステムをサーバプロセスとして計算機上に載せておき，必要に応じてそのプロセスを呼び出すサーバ・クライアント形式を採用している。ただし，どれがサーバで，どれがクライアントというはっきりとした概念はなく，各サブシステムをオブジェクトとして，各オブジェクトにメッセージを送るというオブジェクト指向的な考え方をしている。

オブジェクト間で交換されるメッセージは，思考系からの行動命令および行動・センサ系からのロボット位置，センシング結果である。ロボットの位置情報の中には，ロボットの代表位置や，ロボットに搭載されているセンサの位置といったロボットのコンポーネントの中の特定の一点に関する位置情報とロボットの姿勢の情報のように，ロボットを構成するすべてのコンポーネントの位置が必要となる情報がある。

思考系から発せられるメッセージ（命令）のあて先は，ロボット動力学システム，位置認識システム，環境異常認識システム，映像表示システムである。これらのサブシステムへのメッセージ伝達機構としては，ソケットを用いている。思考系とそれぞれのサブシステムの間に専用のソケットを備えており，したがって，あるサブシステムあての命令が別のサブシステムに伝達されることはない。

7.3 統合化されたシステムによるシミュレーション

統合的シミュレーションシステムは，ほぼこれまで記述してきたシステム設計に基づいて試作され，Fig.7.4に示すように計算機上に実装されている。また，本シミュレーションシステムは，人工知能研究のワークベンチとして一連のシステムの振る舞いを詳細（数値的，論理的）に模擬できるほかに，ロボットハードウェアの評価にも使用できることから，ソフトウェアのみならず本年度製作した原子力施設用センシングシステム（ハードウェア）も一つのサブシステムとして取り扱える。

これによって，オペレータは命令理解システムに日本語で命令を入力することによってロボットを任意の場所に移動させたり，命令の実行途中でロボットの行動を中断させたり，再開させたりすることができる。ロボットは行動時に施設形状データにアクセスして経路を探索し，自己位置を確認しながら２足歩行する。一連の動作は１０分の１秒ごとに計算機のディスプレイ上に映像化され，視覚的に確認することができる。

第２章に示した日本語命令入力ウインドウに日本語命令を入力すると，命令理解システムはこれを解析し，必要な行動列を生成し，経路探索を行う。その後，求められた経路（行動）が一つずつ，ソケットを通してロボット動力学システムに送られる。

ロボット動力学システムは，ロボット歩容作成のためのデータベースを参照しながら，思考系から送られてきた行動を満足する歩容を作成し，思考系に返答する。この歩容データに基づいて，映像表示システムはほぼ１０分の１秒ごとに環境の映像を更新，表示し続ける。Fig.7.5に映像化された環境中でのロボットの例を示す。

また，一つの行動（例えば，直進，跨ぎ越え，階段の昇降など）の始まり，終わりには，思考系から自己位置認識の命令が行動・センサ系に送られ，その結果が思考系に返送される。もし，自己位置認識の結果，ロボットの位置が静的経路探索で計画した位置より大きくずれていた場合には，思考系は次の命令を出す前に，計画した位置に移動するような命令を発する。位置のズレがさほど大きくない場合には，次の移動命令の始点を変更することによって対処している。

7.4 おわりに

本章では，これまで個別に開発されてきた要素技術を統合して，人工知能研究のワークベンチとして，また，ハードウェアとしてのロボットシステムの評価・検討のためのツールとして有用なシミュレーションを実現するためのシステム設計について記述し，また，それをもとに本年度に試作された統合的シミュレーションシステムとその結果，すなわち，日本語命令の入力からロボットによる命令の達成に至る一連のロボットシステムの映像化された振る舞いを示した。

ただし，文中にいくつか述べたように，人工知能研究に有用な統合的シミュレーションシステムとしては，若干完成していない部分が今後の課題として残されている。その中でも主要なものの一つは，「学習」に関するシミュレーションである。環境異常認識システムあるいは

ロボットビジョンにより観察された環境の変化を入力として,（無矛盾性に留意しながら）施設形状データあるいは知識ベースを動的に更新するシステムの開発が統合的シミュレーションシステムの人工知能研究における有用性を格段に向上することは明白である。今後さらに，各要素技術の開発状況を睨みながら，統合的シミュレーションシステムの完成・高度化に向けた考察を行う予定である。

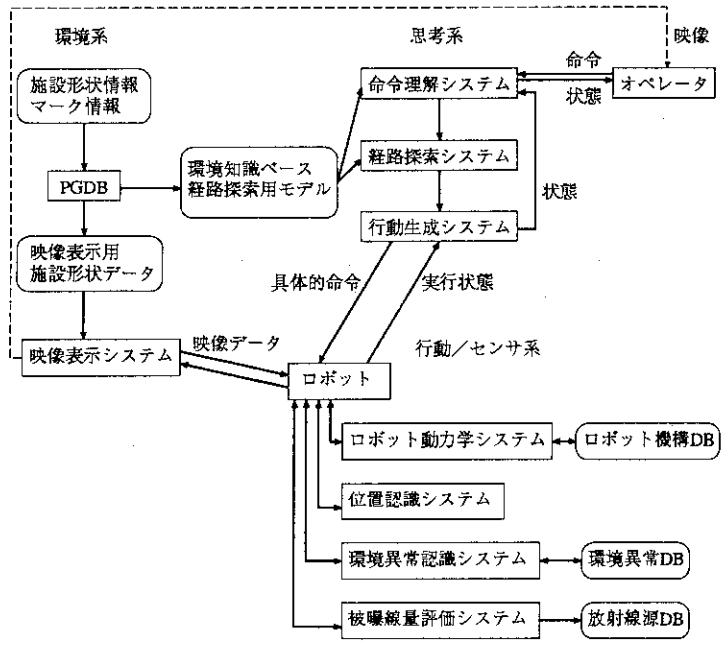


Fig. 7.1 System integration for HASP.

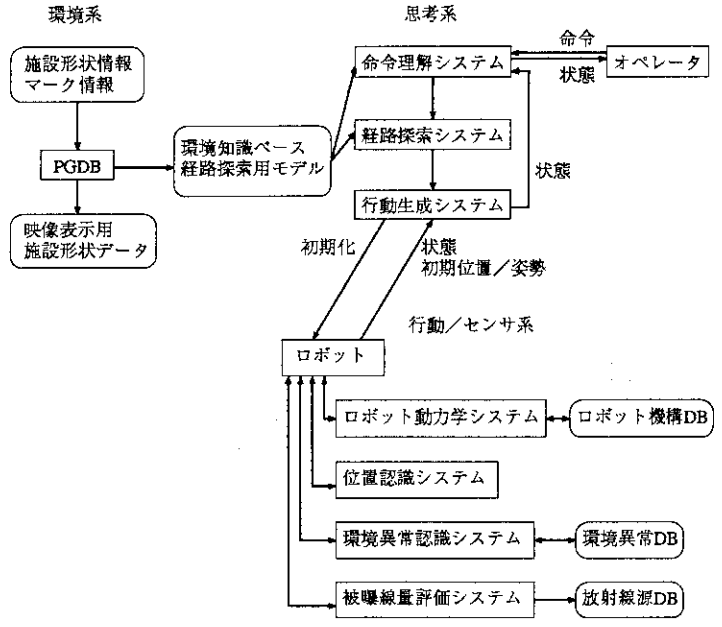


Fig. 7.2 Initial phase of the integrated simulation system.

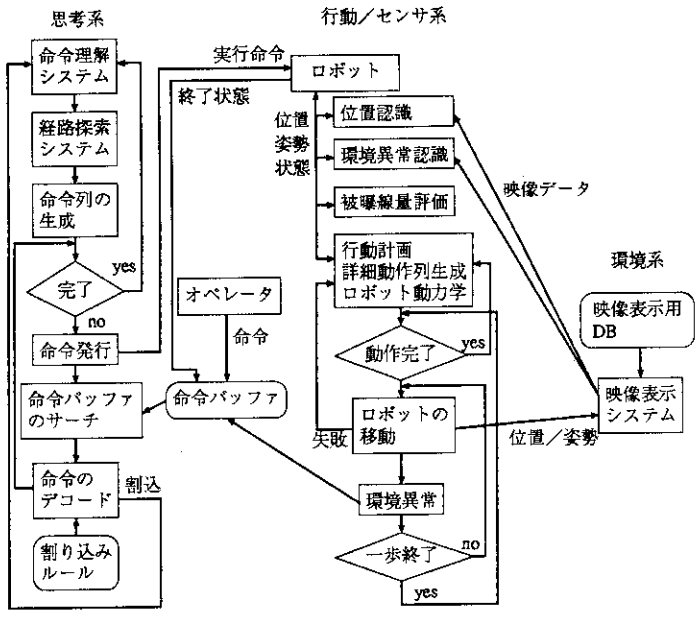


Fig. 7.3 Action phase of the integrated simulation system.

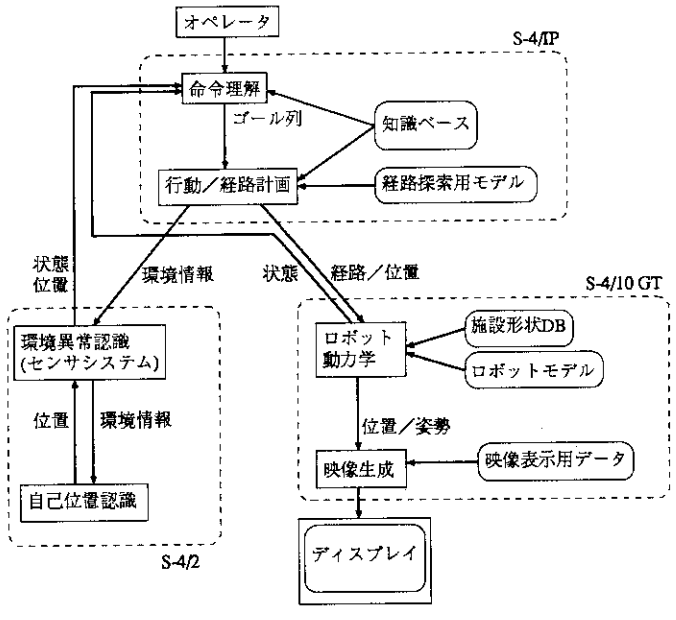


Fig. 7.4 System configuration of the prototype of the integrated simulation system.

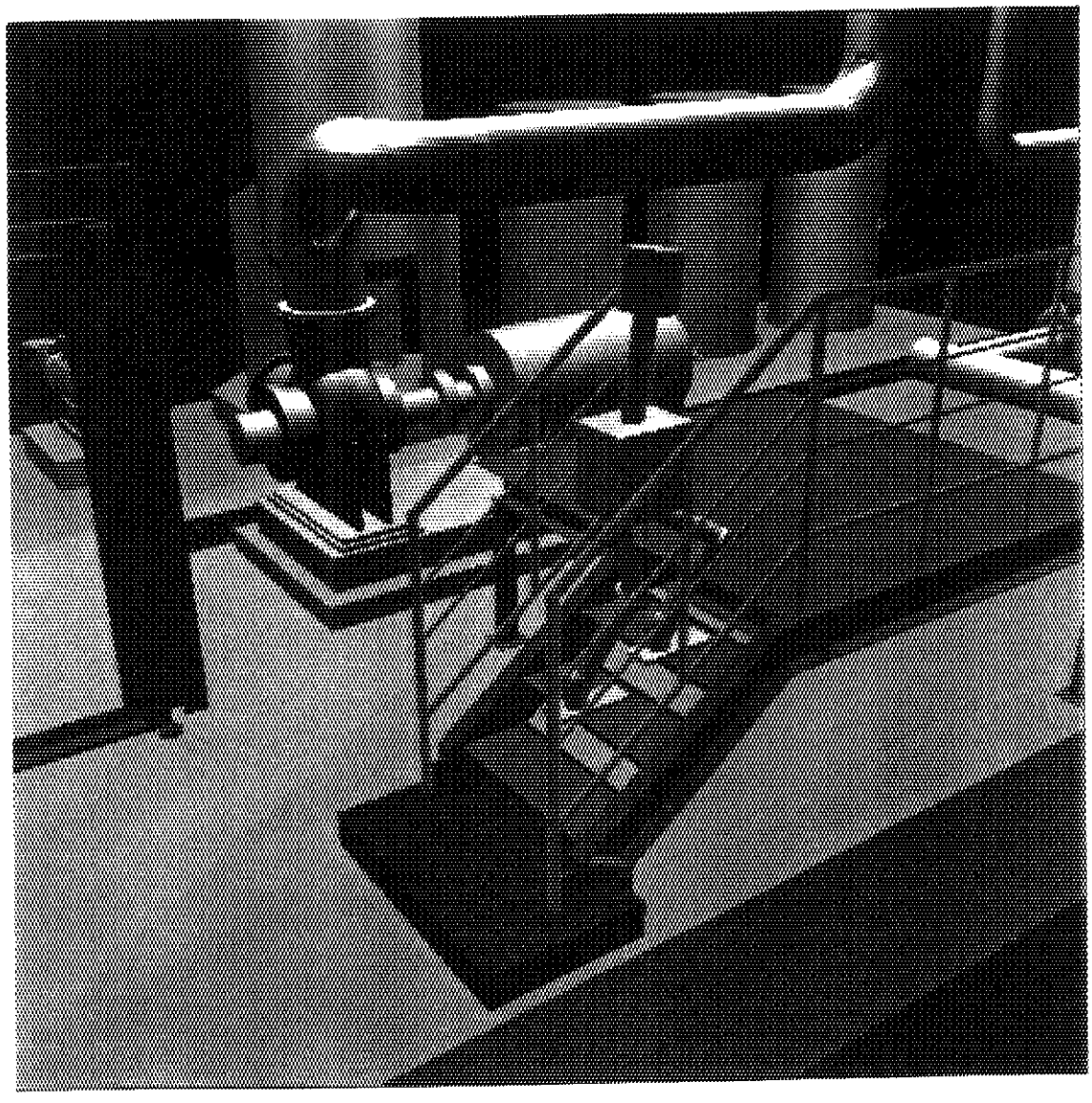


Fig. 7.5 An example of simulated result by using the integrated simulation system.

8. おわりに

ＨＡＳＰの研究開発も７年を経過し，研究も幾分進展をみたものの，知能ロボットの基本機能である知覚，計画・制御，行動を統合した高度な知能ロボット実現には残された課題が多い。各種ロボットで，これまでに実現された知的機能のレベルは決して高いものではない。伝統的なＡＩの「物理的記号仮説」に基づくロボットの知能化は「積木の世界」でのみ有効であり，「フレーム問題」にみられるように，記号的推論は人間の知性の原理とは異なることが明らかになった。また，伝統的ＡＩでは，知性のモデル化に際して，環境との接点である知覚・行動と推論を行う中枢系がそれぞれ別の設計原理で動作するため統合された高度な知性の実現が困難であった。この弱点を克服するためにアプローチがいわゆる「モデルベースト・ロボティクス」である。

ＨＡＳＰの研究開発も基本的にこのアプローチに基づいている。即ち，「環境の幾何モデル」を介して知能化を実現しようとする方法であるが，この方法では，環境の動的変化に対応したロボットの動的行動や環境情報の学習などに多くの課題が残されている。一方，このモデルベースト・アプローチと基本的に異なる「包摂原理に基づくアプローチ」，即ち「逍遥」や「障害物を回避しながら移動する」などの基本的行動を単位として，これらを順次積上げて高度な知性を実現しようとする研究も進展している。このような研究動向に注意を払いつつ，今後のＨＡＳＰの研究開発では，ロボットの「知覚と行動の統合」及び「学習」に重点を置いた研究を展開する予定である。

尚，本報告書の執筆担当者はつぎのとおりである。

第２章	神林 奨
第３章	海老原 健一
第４章	久米 悦雄
第５，６章	樋口 健二
第７章	大谷 孝之
第１，８章	秋元 正幸

謝　　辞

ＨＡＳＰの研究開発に関し，原子力安全システム研究所西原宏氏，九州大学松尾文顕教授からご批判，ご助言を頂いた。厚くお礼を申し上げます。

8. おわりに

ＨＡＳＰの研究開発も７年を経過し，研究も幾分進展をみたものの，知能ロボットの基本機能である知覚，計画・制御，行動を統合した高度な知能ロボット実現には残された課題が多い。各種ロボットで，これまでに実現された知的機能のレベルは決して高いものではない。伝統的なＡＩの「物理的記号仮説」に基づくロボットの知能化は「積木の世界」でのみ有効であり,「フレーム問題」にみられるように，記号的推論は人間の知性の原理とは異なることが明らかになった。また，伝統的ＡＩでは，知性のモデル化に際して，環境との接点である知覚・行動と推論を行う中枢系がそれぞれ別の設計原理で動作するため統合された高度な知性の実現が困難であった。この弱点を克服するためにアプローチがいわゆる「モデルベースト・ロボティクス」である。

ＨＡＳＰの研究開発も基本的にこのアプローチに基づいている。即ち,「環境の幾何モデル」を介して知能化を実現しようとする方法であるが，この方法では，環境の動的変化に対応したロボットの動的行動や環境情報の学習などに多くの課題が残されている。一方，このモデルベースト・アプローチと基本的に異なる「包摂原理に基づくアプローチ」，即ち「逍遥」や「障害物を回避しながら移動する」などの基本的行動を単位として，これらを順次積上げて高度な知性を実現しようとする研究も進展している。このような研究動向に注意を払いつつ，今後のＨＡＳＰの研究開発では，ロボットの「知覚と行動の統合」及び「学習」に重点を置いた研究を展開する予定である。

尚，本報告書の執筆担当者はつぎのとおりである。

第２章	神林 奨
第３章	海老原 健一
第４章	久米 悦雄
第５，６章	樋口 健二
第７章	大谷 孝之
第１，８章	秋元 正幸

謝　　辞

ＨＡＳＰの研究開発に関し，原子力安全システム研究所西原宏氏，九州大学松尾文顕教授からご批判，ご助言を頂いた。厚くお礼を申し上げます。