

人工現実感技術を用いた
プラント運転・保守の
訓練システムに関する研究

平成12年1月

石井 裕剛

人工現実感技術を用いた
プラント運転・保守の
訓練システムに関する研究

平成 12 年 1 月

石井 裕剛

人工現実感技術を用いた プラント 運転・ 保守の訓練システムに関する研究

目次

| | |
|---|-----|
| 論文要旨 | iv |
| 図目次 | vii |
| 表目次 | ix |
| 略語集 | x |
| 第 1 章 序論 | 1 |
| 1.1 研究の背景と目的 | 1 |
| 1.2 本論文の構成 | 3 |
| 参考文献 | 5 |
| 第 2 章 人工現実感技術を用いた訓練環境 | 6 |
| 2.1 人工現実感の研究状況 | 6 |
| 2.1.1 計測サブシステム | 7 |
| 2.1.2 表示サブシステム | 9 |
| 2.1.3 シミュレーションサブシステム | 12 |
| 2.1.4 人工現実感の問題点と課題 | 15 |
| 2.2 原子力発電プラントの運転および保守作業の訓練 | 15 |
| 2.3 人工現実感技術を用いた訓練環境 | 17 |
| 2.4 本研究の着眼点と範囲 | 20 |
| 2.5 まとめ | 23 |
| 参考文献 | 28 |
| 第 3 章 人工現実感技術を用いた機器保守の訓練システム VRMATE の開発 | 29 |
| 3.1 はじめに | 29 |
| 3.2 機器保守の訓練システムに必要とされる機能 | 30 |
| 3.3 構築の対象とする機器保守の訓練環境 | 31 |
| 3.4 仮想空間のシミュレーション手法 | 33 |
| 3.4.1 機器保守作業における訓練生のジェスチャ判定 | 34 |

目次

| | | |
|--------------|--|-----------|
| 3.4.2 | ペトリネットの定義 | 36 |
| 3.4.3 | ペトリネットによるモデル化手法 | 39 |
| 3.4.4 | 仮想空間のシミュレーション手法 | 41 |
| 3.5 | ペトリネット作成支援システム PCSS の開発 | 45 |
| 3.5.1 | PCSS 設計の基本方針 | 45 |
| 3.5.2 | PCSS の概要 | 46 |
| 3.5.3 | PCSS を用いたペトリネットの作成 | 53 |
| 3.6 | 機器保修の自動実演機能 | 55 |
| 3.6.1 | 自動実演機能を構築する際に考慮すべき事柄 | 56 |
| 3.6.2 | 次に発火すべきトランジションを決定するアルゴリズム | 58 |
| 3.7 | VRMATE の開発 | 62 |
| 3.7.1 | VRMATE の概要 | 62 |
| 3.7.2 | VRMATE の使用例と評価 | 65 |
| 3.8 | 考察及び将来課題 | 69 |
| 3.9 | まとめ | 71 |
| | 参考文献 | 74 |
| 第 4 章 | 機器保修の訓練環境の構築支援システム DESCORTE の開発 | 75 |
| 4.1 | 研究の背景と目的 | 75 |
| 4.2 | DESCORTE で構築可能な訓練環境 | 76 |
| 4.2.1 | DESCORTE で構築可能な訓練環境が具備すべき要件 | 76 |
| 4.2.2 | 訓練環境を構築する際に作成する必要がある情報 | 78 |
| 4.3 | 仮想物体の外見と運動の仕方を定めるデータベース | 79 |
| 4.3.1 | 仮想空間で模擬する必要がある物体の運動と物体相互間の拘束関係 | 79 |
| 4.3.2 | DESCORTE で設定可能なパラメータ | 80 |
| 4.4 | 仮想物体の状態遷移の仕方を定めるデータベース | 82 |
| 4.4.1 | 改良ペトリネットの定義 | 82 |
| 4.4.2 | 改良ペトリネットによるモデル化手法 | 85 |
| 4.5 | DESCORTE の開発 | 86 |
| 4.5.1 | DESCORTE の構成 | 87 |
| 4.5.2 | DESCORTE のインタフェース設計 | 89 |
| 4.5.3 | 訓練を実施する際の情報処理手順 | 100 |
| 4.6 | DESCORTE の評価 | 103 |
| 4.6.1 | スイング式逆止弁の分解作業を対象とした訓練環境の構築 | 103 |
| 4.6.2 | DESCORTE の操作性の実験による評価 | 107 |
| 4.7 | DESCORTE のインタフェースの改良 | 110 |
| 4.7.1 | DESCORTE のインタフェースの改良策 | 110 |
| 4.7.2 | インタフェース改良後の DESCORTE の評価 | 113 |
| 4.8 | 考察及び将来課題 | 114 |
| 4.9 | まとめ | 116 |

| | |
|---|-----|
| 参考文献 | 117 |
| 第5章 人工現実感技術を用いた原子カプラント運転の体験型教育システム VENUS の開発 | 118 |
| 5.1 はじめに | 118 |
| 5.2 VENUS に必要とされる機能 | 119 |
| 5.3 VENUS の概要 | 121 |
| 5.3.1 ソフトウェア構成 | 121 |
| 5.3.2 ハードウェア構成 | 122 |
| 5.4 SEAMAID | 123 |
| 5.5 人体モーション合成シミュレータ | 125 |
| 5.5.1 人体モーションの合成技術 | 126 |
| 5.5.2 仮想運転員の動作の分類 | 127 |
| 5.5.3 人体モデルの構成 | 130 |
| 5.5.4 人の動作の計測と計測データの補正 | 132 |
| 5.5.5 動作データベースの作成 | 135 |
| 5.5.6 移動動作の合成 | 139 |
| 5.5.7 操作動作の合成 | 145 |
| 5.5.8 移動動作と操作動作の統合 | 150 |
| 5.6 仮想空間シミュレータ | 150 |
| 5.6.1 制御盤の描画 | 151 |
| 5.6.2 仮想運転員の描画 | 151 |
| 5.6.3 仮想運転員の視点からの映像の描画 | 153 |
| 5.7 その他のシミュレータ | 153 |
| 5.7.1 CRT モニタシミュレータ | 153 |
| 5.7.2 発話合成シミュレータ | 154 |
| 5.8 VENUS の使用例と考察 | 156 |
| 5.9 将来課題 | 158 |
| 5.10 まとめ | 159 |
| 参考文献 | 163 |
| 第6章 結論 | 164 |
| 謝辞 | 168 |
| 著者発表文献リスト | 170 |
| 付録 A 研究の概要（カラー版） | 175 |

論文要旨

近年の急激な電力需要の増加により、原子力発電の果たす役割は、ますます重要になってきている。しかし原子力発電は、その安全性の確保が重要な課題となっており、その中でも、原子力発電プラントを運転する運転員や保守を行う作業員の教育訓練の高度化が不可欠であるとされている。現在、訓練センターで行われているプラント運転員や保守作業員の教育訓練は、機器やシステムの実物大模型を用いた実技訓練が、訓練の重要な部分を占めているが、コスト削減の要求等により十分な訓練設備を設置できないことや、実物大模型等を利用する際には実際の作業に近い危険性が伴うために、訓練のインストラクタが常に付き添う必要があることなどから、実際の作業を体験できる実技訓練は十分に行われていないのが現状である。

そこで本研究では、安全・安価にプラントの運転や機器の保守作業を体験できる新しい訓練環境として、最近、新しいヒューマンインタフェース技術として注目されている人工現実感技術を用いた訓練システムの構築法に関する基礎研究を行うこととした。具体的には、「機器の保守作業」と「プラントの運転操作」の2つの訓練対象に分けて、それぞれの訓練の特質を分析し、人工現実感技術を用いた訓練システムを開発するための新しい手法を提案する。そして、具体的な訓練課題に対して実際に訓練システムを開発し、試用実験による評価を行うことにより、その有効性を確認することとした。

まず、人工現実感技術を用いた機器保守の訓練環境として、原子力発電プラントの構成機器の1つであるスイング式逆止弁の分解作業を対象に訓練システムを開発した。これまでに開発された人工現実感技術を用いた機器保守の訓練システムは、作業の自由度が非常に低く、正しい手順に限り実行可能であるものが大半であった。そこで、本研究では、訓練システムで取り扱える作業の範囲を正しい手順に限定せず、間違った手順でも作業を行うことを可能にすることにより、訓練生が自らの判断で、自由に作業を行える訓練環境の構築を試みた。しかし、訓練生の作業の自由度を上げるに従って、仮想空間内の機器や工具の動きの管理が複雑になる。そこで、本研究では、機器保守の作業中の機器や工具の動きを、物体の状態が順次変化していく離散並行的状態変化と捉え、ペトリネットを用いてモデル化する新たな手法を開発し、機器保守の訓練環境を構築する際に適用した。また、機器保守の作業を対象にモデル化したペトリネットには、同じ構造を持つ小規模なペトリネットが多数存在する傾向があることに注目し、頻繁に使用するペトリネット構造を、必要に応じて複製して使用する作業を支援することにより、ペトリネットによるモデル化作業の労力を削減することを目指したペトリネット作成支援システムを開発した。さらに、訓練生が機器保守の作業を行う際の教示機能として、仮想空間内で機器保守の作業を自動的に実演する自動実演機能のための次手順の評価アルゴリズムを考案し、機器保守の訓練システムに実装した。以上の研究を通じて、作業の自由度が非常に高い機器保守の訓練環境を構築するための新たな手法を提案し、実際に訓練環境を構築することによって、その

有効性を確認すると共に、人工現実感技術を用いた訓練システムの有用性を確認し、同時に問題点の抽出を行った。

次に、上記の機器保修の訓練システムを開発する際に抽出した「人工現実感技術を用いた機器保修の訓練環境の構築には、そのプログラミングに専門的な知識を要し、非常に多くの労力が必要になる」というソフトウェア作成上の問題を解決するために、機器保修の訓練環境の構築支援システムを開発した。まず始めに、構築支援システムのユーザとして、プログラミングについては専門知識を持たないインストラクタを仮定して、そのための構築支援システムの設計と開発を行った。具体的には、以下の3つの特徴を持つ構築支援システムの開発を目指して研究を進めた。

- (1) プログラミングを全く行わなくても自由に機器保修の訓練環境を構築できる。
- (2) 計算機や数学等に関する高度な知識を持たない人でも容易に訓練環境を構築できる。
- (3) 訓練環境を構築する際に必要な情報は全て Graphical User Interface を介して短時間で入力できる。

まず、(1)の目標に対して、実際に保修作業の訓練が行われている複数の機器に対して、仮想空間内に訓練環境を構築する際に作成する必要がある情報を分析し、それらの情報を既に定められたパラメータを変更することによって作成し、訓練環境を構築する手法を考案した。そして、(2)の目標に対して、仮想空間内の物体の状態遷移を視覚的にモデル化するための、理解の容易さとモデル化能力を向上させた改良ペトリネットを考案し、構築支援システムに応用した。また、(3)の目標に対しては、構築支援システムのユーザインタフェース設計の評価と改良を繰り返し、ユーザの認知的負荷を軽減させることにより作業効率の改善をはかり、プログラミングによる従来の訓練環境の構築法に比べ、非常に短時間で機器保修の訓練環境を構築できる支援システムの開発に成功した。

一方、これまでに開発された、人工現実感技術を用いた原子力発電プラントの運転員を教育する訓練システムは、仮想空間内にプラントの制御室を構築し、その中で訓練生自らが制御盤を操作することによって、訓練を行う方式のものが大半であった。そこで、本研究では、この訓練方式をさらに発展させ、プラント監視制御行動中の運転員の行動を人工現実感技術を用いて総合的に情報可視化する手法を開発し、プラント運転の教育訓練に応用する研究を行った。具体的には、プラント運転員の行動と思考過程を計算機上で模擬できるヒューマンモデルを用いることにより、仮想空間内に自らの判断でプラントの異常診断と対応操作を行う仮想運転員を構築し、訓練生がこの仮想運転員の運転作業の様子を観察することにより、プラント運転に必要な知識と技能を修得する、新しい訓練方式の体験型教育システムを開発した。この教育システムの特徴をまとめると以下の様になる。

- (1) 仮想運転員は実際の運転員に近い自然な動作でプラントの異常診断と対応操作を行う。
- (2) 仮想運転員はプラント運転の作業中に、発話によって訓練生に思考過程を伝える。
- (3) 訓練生は仮想運転員が作業を行う様子を自由な視点から眺めることができる。

- (4) 訓練生は仮想運転員の視点からの映像を立体視しながら眺めることにより、プラントの運転作業を仮想体験できる。

以上のような教育システムを開発するために、(1) 訓練シミュレーションを実行する際の計算負荷を分散させるために、教育システムを分散型シミュレーションシステムとして設計し、(2) 仮想空間を描画する際に、訓練生の視点と描画対象との距離に応じて描く詳細度を変化させる Level Of Detail(LOD) の手法を採用した。また、(3) 3次元モーションキャプチャシステムを用いて人間の動作を計測し、仮想運転員の動作データベースを作成し、必要に応じてそれらを変形することにより、歩行動作と、機器を操作する動作を自由に合成できる人体モーションの合成技術を開発した。そして、開発した体験型教育システムに対し、数人の被験者に対して試用実験を行うことにより、原子力発電プラントに関する予備知識を持たない訓練生が、プラント異常発生時の診断と対応操作の方法を理解できることを確認した。

以上、本研究では、人工現実感技術を用いた機器保修の訓練環境及び、プラント運転の訓練環境を構築するための新たな手法を提案し、具体的な訓練課題に対する訓練環境の構築、及び、訓練環境の構築支援システムの開発を通して、その有効性を確認した。

図目次

| | | |
|------|----------------------------------|----|
| 2.1 | 人工現実感の一般的枠組み | 6 |
| 3.1 | スイング式逆止弁の構造 | 32 |
| 3.2 | ペトリネットのグラフ表現 | 37 |
| 3.3 | 発火可能なトランジションの例 | 38 |
| 3.4 | 発火不可能なトランジションの例 | 38 |
| 3.5 | トランジション発火後のトークン配置 | 39 |
| 3.6 | モデル化したペトリネットの例 | 41 |
| 3.7 | 「床の上に置かれたナットを手でボルトにはめる」作業のペトリネット | 42 |
| 3.8 | 訓練シミュレーションにおける処理の流れ | 43 |
| 3.9 | 訓練シミュレーションにおける情報の流れ | 44 |
| 3.10 | ペンを掴む作業をモデル化したペトリネット | 46 |
| 3.11 | ナットを掴む作業をモデル化したペトリネット | 47 |
| 3.12 | 「物」を掴む作業をモデル化したペトリネット | 47 |
| 3.13 | テンプレート作成モードのインタフェース | 49 |
| 3.14 | テンプレート作成モードのメニューの階層構造 | 49 |
| 3.15 | テンプレート結合モードのインタフェース | 50 |
| 3.16 | テンプレート結合モードのメニューの階層構造 | 51 |
| 3.17 | 動作確認モードのインタフェース | 53 |
| 3.18 | 展開されたペトリネット | 55 |
| 3.19 | 訓練生の作業を監視するアルゴリズム | 60 |
| 3.20 | 次に発火すべきトランジションを求めるアルゴリズム | 61 |
| 3.21 | VRMATE のハードウェア構成 | 63 |
| 3.22 | VRMATE のソフトウェア構成 | 64 |
| 3.23 | VRMATE の初期画面 | 66 |
| 3.24 | 逆止弁を横から眺めた映像 | 67 |
| 3.25 | レンチでナットを緩める作業中の仮想空間の映像 | 67 |
| 3.26 | 訓練の補助機能 (各種情報提示) | 68 |
| 4.1 | 改良ペトリネットのグラフ表現 | 83 |
| 4.2 | 旧定義に従ったペトリネット | 84 |
| 4.3 | 新定義に従った改良ペトリネット | 84 |
| 4.4 | ペンで印を付ける作業をモデル化したペトリネット | 86 |
| 4.5 | システム構成 (構築モード) | 88 |
| 4.6 | システム構成 (訓練モード) | 88 |

目 次

| | | |
|------|------------------------------------|-----|
| 4.7 | 構築モードの画面例 | 90 |
| 4.8 | アイコンを張り付けたボタンの例 | 91 |
| 4.9 | 新たに物体を登録する際のインタフェース | 92 |
| 4.10 | 状態を作成する際のインタフェース (外見に関する情報) | 93 |
| 4.11 | 状態を作成する際のインタフェース (初期位置に関する情報) | 94 |
| 4.12 | 状態を作成する際のインタフェース (平行移動に関する情報) | 95 |
| 4.13 | 状態を作成する際のインタフェース (時間に応じた移動法に関する情報) | 96 |
| 4.14 | テンプレート機能 | 97 |
| 4.15 | 新規トランジション作成画面 | 98 |
| 4.16 | イベント設定画面 | 99 |
| 4.17 | ペトリネット表示画面 | 99 |
| 4.18 | ペトリネット拡大縮小ボタンとスライドバー | 100 |
| 4.19 | 環境設定画面 | 101 |
| 4.20 | 操作の実行を確認するウインドウ | 101 |
| 4.21 | 訓練実行時の情報の流れ | 102 |
| 4.22 | スイング式逆止弁の保修訓練環境の画面例 | 106 |
| 4.23 | シャフトの状態遷移のモデル化例 | 108 |
| 4.24 | スイッチでモータの回転を制御する仮想空間の画面例 | 109 |
| 4.25 | インタフェース改良後の DESCORTE の画面例 | 112 |
| 4.26 | DESCORTE の将来構想 | 115 |
| 5.1 | VENUS のソフトウェア構成 | 121 |
| 5.2 | VENUS のハードウェア構成 | 123 |
| 5.3 | SEAMAID のシミュレーション部のシステム構成 | 124 |
| 5.4 | ヒューマンモデルの異常診断過程 | 125 |
| 5.5 | 仮想制御室の制御盤の配置と仮想運転員の移動範囲 | 128 |
| 5.6 | 移動の方向と目的地点で向いている方向 | 129 |
| 5.7 | 仮想運転員の人体モデル | 131 |
| 5.8 | 人体モデルの階層構造 | 132 |
| 5.9 | 人体モデルの各部位のローカル座標系 | 133 |
| 5.10 | 動作データを収集する際のシステム構成 | 134 |
| 5.11 | 動作データの選択から動作データの縮小率の決定までの流れ | 140 |
| 5.12 | 上半身の座標変換 | 143 |
| 5.13 | 制御盤上に配置される機器のテクスチャ | 151 |
| 5.14 | 仮想制御室 | 152 |
| 5.15 | 仮想制御盤 | 152 |
| 5.16 | 仮想運転員の視点からの映像 | 153 |
| 5.17 | CRT モニタ上に描画されるトレンドグラフの例 | 154 |
| 5.18 | CRT モニタシミュレータが模擬する 2次元映像の例 | 155 |
| 5.19 | 仮想運転員の歩行軌跡 | 157 |

表目次

| | | |
|------|-----------------------------|-----|
| 2.1 | FASTRAK の仕様 | 7 |
| 2.2 | 必修作業員が修得すべき課題 | 16 |
| 2.3 | プラント運転員が修得すべき課題 | 16 |
| 2.4 | 実物大模型を用いた実技訓練の問題点 | 17 |
| 2.5 | 人工現実感を用いて訓練環境を構築することの利点 | 18 |
| 2.6 | 人工現実感を用いることによる問題点の改善 | 18 |
| 3.1 | スイング式逆止弁の分解手順 | 32 |
| 3.2 | VRMATE が認識する訓練生のジェスチャ | 34 |
| 3.3 | 「ナットをスパナで締める」作業の単位タスク表現 | 35 |
| 3.4 | ジェスチャが行われたと判定する条件 | 35 |
| 3.5 | ナットが取りうる状態とボルトが取りうる状態 | 40 |
| 3.6 | 「物体の状態」として使用する言葉 | 51 |
| 3.7 | 「他の物体との関係」として使用する言葉 | 51 |
| 3.8 | テンプレートとして既に登録されているペトリネット | 52 |
| 3.9 | ペンで印を付ける作業を単位タスクに分解した結果 | 54 |
| 3.10 | 自動実演を行うために必要な情報 | 58 |
| 3.11 | 変数一覧 | 58 |
| 4.1 | トランジションを発火させる原因として選択可能なイベント | 86 |
| 4.2 | 「印を付ける訓練環境」を構築する場合に入力する物体名 | 90 |
| 4.3 | ペンが取りうる状態 | 92 |
| 4.4 | 自動的に入力されるパラメータ | 92 |
| 4.5 | 登録する仮想物体の名前 | 104 |
| 4.6 | 登録する状態の例 (righthand) | 104 |
| 4.7 | 手に握られた状態にあるペンに対する運動データベース例 | 105 |
| 4.8 | 落下している状態にあるナットに対する運動データベース例 | 105 |
| 4.9 | 回転中のシャフトの状態に対する運動データベース例 | 108 |
| 4.10 | インタフェース改良による効果 | 114 |
| 5.1 | 作成した移動用の動作データの一覧 | 138 |
| 5.2 | 作成した操作用の動作データの一覧 | 139 |
| 5.3 | 仮想運転員の発話内容 | 155 |
| 5.4 | 仮想運転員の行動 | 156 |
| 5.5 | 仮想運転員の発話例 | 157 |

略語集

CAD

Computer Aided Design

DESCORTE

Design Support System for Constructing VR-based Machine-Maintenance Training Environment

FWM

Focal Working Memory

GUI

Graphical User Interface

HMD

Head Mounted Display

ITS

Intelligent Tutoring System

KB

Knowledge Base

LOCA

Loss of Coolant Accident

LOD

Level of Detail

NFS

Network File System

OS

Operating System

OSF

Open Software Foundation

PCSS

Petri net Construction Support System

PWM

Peripheral Working Memory

PWR

Pressurized Water Reactor

RMS

Root Mean Square

SEAMAID

Simulation-based Evaluation and Analysis Support System for Man-machine Interface Design

SGTR

Steam Generator Tube Rupture

STRIPS

Stanford Research Institute Problem-Solver

VENUS

VR-based Education System for Operating Nuclear Power Plants

VIGOR

Virtual Environment for Visual Engineering and Operation

VR

Virtual Reality

VRMATE

VR-based Machine-Maintenance Training Environment

WIMP

Window, Icon, Menu, Pointer

WTK

World Tool Kit

第 1 章 序論

1.1 研究の背景と目的

原子力発電は 21 世紀の重要な電力供給源であるとされている。実際、日本における商業用原子力発電プラントは 1999 年 8 月現在、51 基が運転中であり、日本の総発電量の 40 % 近くを担っている [1]。今後も日本における電力消費量はさらに増加することが予想されるが、火力発電では炭酸ガス排出による地球温暖化の懸念を増大させ、一方、代替エネルギーとしての再生型自然エネルギーの利用についても大幅な容量増大が期待できない現状では、原子力発電は欠くことができないエネルギー源である。しかし、1979 年 3 月にはアメリカ合衆国のスリーマイル島原子力発電プラントにおいて原子炉が熔解する事故が起き、さらに 1986 年 4 月には旧ソビエトのチェルノブイリ原子力発電プラントにおいて、原子炉が爆発する事故が起きるなど、原子力発電は大事故による放射線災害の懸念などから、その安全性確保が重要な課題となっている [2]。

これに対し、近年の原子力発電プラントでは、さらなる安全性の向上、安定稼働を目指し、プラントの構成機器の多重化や安全設備の設置等が行われている。しかし、これまで国内の原子力発電プラントで発生したトラブルの直接的な原因の約 20 % はヒューマンエラーによるものとされており、間接的には人的要因が関わっているものは約 80 % に達すると言われている [3]。従って、プラントを操作する運転員やプラントの定期点検等を行う保守作業員の教育訓練も非常に重要である。

現在、プラント運転員や保守作業員の教育訓練は、座学形式による講義と実物大模型等を用いた実技訓練が併用して行われている。座学形式による講義は基礎的な内容を教示する際には効果的であるが、実際のプラントの運転や、機器の保守作業を行うための十分な技術を身につけるためには、実際の作業を「体験」できる実技訓練が必要不可欠である。しかし、プラント運転員や保守作業員の教育訓練を行う訓練センターでは、訓練を実施する場所の制限やコスト削減の要求等により十分な訓練設備を設置できないことや、実物大模型等を利用する際には実際の作業に近い危険性が伴うために訓練のインストラクタが常に付き添う必要があることなどから訓練時間が制限され、十分な実技訓練はなされていないのが現状である。

一方、近年の計算機技術の急激な進展に伴い、人工現実感 (Virtual Reality; VR) が注目され始めている。人工現実感とは、人間が今、現在、実際に存在している環境以外の仮想環境を、あたかもそれが現実の環境のような感覚を持って「体験」し、かつその仮想世界で行動することを可能とする新しい技術である。人工現実感を用いれば、まだできていない製品を手で触ったり、設計中の家の中を歩き回ったり、分子を手で組み合わせるといった今まで考えられなかった新しい世界が拓けると期待されている [4]。

そこで本研究では、原子力発電プラントの運転や保守作業の教育訓練への人工現実感の

1.1 研究の背景と目的

適用を試みる。人工現実感技術を用いてプラントの運転操作や機器の保守作業の仮想体験が可能な訓練環境を構築することにより、訓練費用の削減や安全性の向上による訓練機会の増加、訓練効率の向上による訓練期間の短縮等が可能になるだけでなく、模型では実現が困難な作業環境における体験訓練が可能になると期待される。

本研究では、人工現実感技術の現状を展望すると共に、これまでに人工現実感技術を応用した訓練システムの研究状況を展望して、人工現実感技術を用いた訓練システムが備えるべき機能や特徴、効率的な訓練環境の構築方法、人工現実感技術を用いた訓練システムによる訓練の実施方法等を提案し、具体的な訓練課題に対する訓練システムの開発、および、訓練環境の構築支援システムの開発を通して、その有効性を確認すると共に、今後、人工現実感技術を用いた訓練システムを実用化するために解決すべき課題をまとめ、将来的に実用的な訓練システムを開発するための基礎技術を確立することを目的とする。

具体的には、本研究では、以下の3つのテーマに関する研究を行った。

- (1) 機器の保守作業を仮想体験できる訓練システムを少ない労力で開発できる手法の開発
作業の自由度が非常に高い機器保守の訓練環境を、これまでの構築手法に比べ少ない労力で構築できる新たな手法として、仮想空間内の物体(仮想物体)の状態遷移を、ペトリネットを用いてモデル化する手法を提案し、さらに、大規模なペトリネットを少ない労力で構築できるペトリネット作成支援システムを開発する。その後、具体的な訓練課題に対する機器保守の訓練システムを開発することにより、提案した手法の有効性を確認し、今後、機器保守の訓練システムを開発、実用化する際の問題点と課題を抽出する。
- (2) 計算機に関する専門的な知識を持たない人でも機器保守の訓練環境を短時間で構築できる支援システムの開発
上記(1)の研究をさらに発展させ、プログラミングを全く行うことなく必要な情報を入力するだけで、仮想空間内に機器保守の訓練環境を構築できる新しい訓練環境の構築手法を提案する。そして、WIMP(Window, Icon, Menu, Pointer)インタフェースを基本としたGUI(Graphical User Interface)を設計することにより、少ない労力で短時間に訓練環境を構築できる構築支援システムを開発する。その際、構築支援システムのユーザインタフェース設計の評価と改良を繰り返し、訓練環境を構築する際のユーザの認知的負荷の軽減を試みる。
- (3) プラント運転員の監視制御行動の情報可視化と体験型教育システムへの応用に関する研究

原子力発電プラントにおける運転員のプラント監視制御行動を、人工現実感技術を用いて総合的に情報可視化するための基礎技術を開発し、それを用いてプラント運転を仮想体験できる体験型教育システムを開発する。具体的には、制御室内におけるプラント運転員の行動を3次元映像として合成できる人体モーションの合成技術を開発し、プラント緊急時の運転員の認知行動を模擬する既開発のヒューマンモデルシミュレータに接続する。さらに異常診断中の運転員の思考過程を発話する機能や仮想運転

員の視点からの映像を立体視しながら体験する機能等を構築することにより、プラント運転員の新しい技能訓練システムを開発する。

1.2 本論文の構成

本論文は、第1章の序論を含め、全体で6章で構成されている。第2章以降の内容を以下に説明する。

第2章では、人工現実感に関する国内外の研究の状況を、人工現実感を実現するシステムの一般的な構成要素である、計測サブシステム、シミュレーションサブシステム、表示サブシステムに分けて展望する。ついで、人工現実感技術を用いた訓練システムに関して、機器の保守作業とプラントの運転操作に分けて、それぞれの訓練の特徴と、人工現実感技術を用いた訓練システムを開発する利点・問題点について述べる。そして最後に、本研究の着眼点と研究の範囲を述べる。

第3章では、原子力発電プラントの構成機器の1つであるスイング式逆止弁の分解作業を対象にした機器保守の訓練システムの開発について述べる。まずはじめに、訓練環境としての機器保守の訓練システムが具備すべき機能について述べた後、仮想物体間の物理的拘束関係を考慮した仮想空間を構築する新たな手法として、仮想物体の状態遷移をペトリネットを用いてモデル化する手法について述べる。そして、ペトリネットによるモデル化の作業を少ない労力で短時間に行うためのペトリネット作成支援システムの開発について述べた後、機器保守の訓練効率を向上させるための自動実演機能について、その実現のためのアルゴリズムと訓練システムへの実装方法について述べ、最後に、開発した機器保守の訓練システムの概要と、評価実験を行った結果及び今後の課題について述べる。

第4章では、人工現実感技術を用いた機器保守の訓練環境を、少ない労力で短時間で構築できるようにするための構築支援システムの開発について述べる。まずはじめに、構築支援システムで構築可能とすべき機器保守の訓練環境の特徴について述べた後、訓練センターで保守訓練の対象となっている複数の機器に対して、それらの保守作業を行うことができる訓練環境を、仮想空間内に構築する際に作成する必要がある情報を分析した結果について述べる。そして、その結果をもとに考案した、プログラミング以外の方法で仮想物体間の物理的拘束関係を考慮した訓練環境を構築する方法について述べる。その後、理解のしやすさとモデル化能力を向上させた改良ペトリネットを用いて仮想物体の状態遷移をモデル化する方法について述べる。次に、開発した構築支援システムのハードウェア構成とソフトウェア構成の概要について述べた後、構築支援システムのユーザインタフェースの概要を説明する。そして、開発した構築支援システムの評価として、第3章で構築した、スイング式逆止弁の分解作業を行うことができる訓練環境を構築支援システムを用いて構築した結果と、構築支援システムを初めて使用する初心者には簡単な訓練環境を構築させた結果について述べる。その後、システムの評価によって判明した、構築支援システムのユーザインタフェースの問題点をまとめ、ユーザの認知的負荷を軽減させる方向でユーザインタフェースを改良した後に、新たな評価実験を行った結果について述べた後、今後の課題について述べる。

1.2 本論文の構成

第5章では、人工現実感技術を用いた原子力発電プラント運転の体験型教育システムの開発について述べる。ここでは、仮想空間内に構築された仮想運転員がプラントの異常診断と対応操作を行う様子を観察し、また、その作業を仮想体験することにより、プラント運転の知識と技術を修得できる新しい訓練方式の訓練システムを開発する。具体的は、まずはじめに、訓練生がプラントの運転操作の仕方を効率的に修得できるようにするために訓練システムが具備すべき機能について述べた後、分散型シミュレーションシステムとして実現した体験型教育システムのソフトウェア構成とハードウェア構成について述べる。その後、体験型教育システムを構成する各シミュレータについて、まず、プラント異常発生時の運転員の行動と思考過程を模擬できる SEAMAID の概要について述べた後、実際の運転員に近い自然な動作を3次元映像として合成できる人体モーションの合成技術について説明する。その後、プラントの異常診断中の思考過程を容易に理解できるようにするための各種補助機能について述べた後、体験型教育システムとして評価実験を行った結果と今後の課題について述べる。

最後に、第6章では、本研究で得られた成果を整理するとともに、今後の研究の発展すべき方向と、解決すべき課題について述べる。

参考文献

- [1] 資源エネルギー庁公益事業部原子力発電安全管理課（編）．原子力発電所運転管理年報．火力原子力発電技術協会，1999．
- [2] 七沢潔．原発事故を問う．岩波新書，1996．
- [3] 井上紘一，幸田武久．ヒューマンエラー．ヒューマンインタフェースシンポジウム'99 講習会資料集，pp. 67-76，1999．
- [4] 舘日章，廣瀬通孝（編）．バーチャル・テック・ラボ．工業調査会，1992．

第 2 章 人工現実感技術を用いた訓練環境

本章ではまず、人工現実感の現在の研究状況を展望し、人工現実感技術の現状と問題点についてまとめる。その後、原子力発電プラントの運転員と保守作業員に対して行われている教育訓練の現状と問題点について述べる。続いて、人工現実感技術を応用した訓練システムのこれまでの研究状況を展望し、人工現実感技術を用いた訓練システムの利点と問題点についてまとめ、最後に本研究の着眼点と範囲について述べる。

2.1 人工現実感の研究状況

一般に、人工現実感を実現するためのシステムは、Fig. 2.1に示すように、人間の動作を計算機に入力するための計測サブシステム、計算機により構成される仮想世界を管理するシミュレーションサブシステム、仮想世界を人間側に臨場感あふれる方法で表示するための表示サブシステムの3つのサブシステムから構成されると捉えることができる [1]。以下では、人工現実感の現在の研究状況を、計測サブシステム、シミュレーションサブシステム、表示サブシステムの3つのサブシステムに分けて展望する。

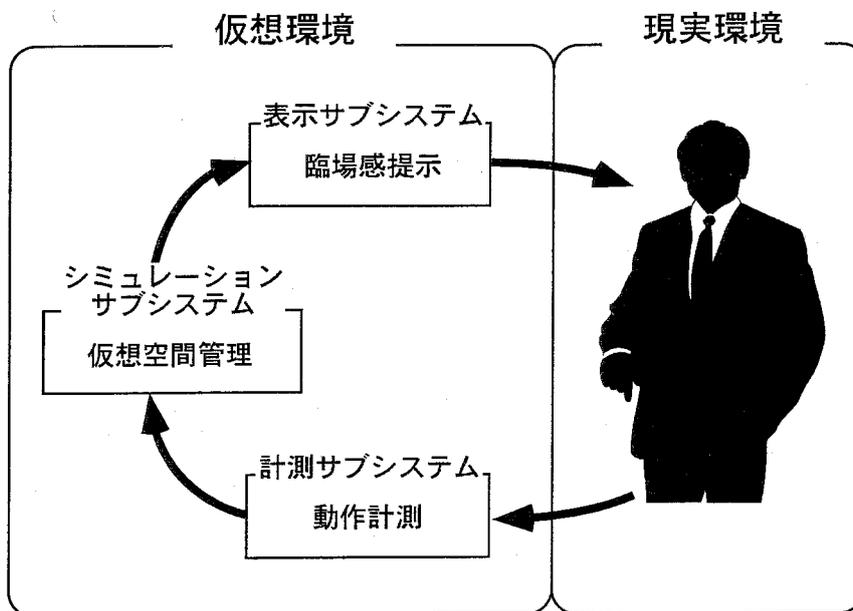


Fig. 2.1: 人工現実感の一般的枠組み

Table 2.1: FASTRAK の仕様

| | |
|----------|---|
| 精度 (RMS) | 位置 : 0.8mm 角度 : 0.15° |
| 分解能 | 位置 : 0.005mm 角度 : 0.025° |
| 測定範囲 | 半径約 76cm の半球内 (精度保障) 半径約 300cm の半球内 (最大) |
| データレート | 120 ポイント/秒 |

2.1.1 計測サブシステム

計測サブシステムでは、人の手の運動や指の運動、体全体の運動だけでなく、表情や発話、さらには、脳波や脈拍のような生体信号までもが計測の対象となる。これらの対象を計測するデバイスは、既に実用化され市販されているものから、現時点ではまだ研究段階にあるものまで、その開発状況はさまざまである。そのような状況の中で、現在の人工現実感を実現するシステムにおいて最もよく利用されているのは、人の運動を計測するためのデバイスである。人の運動を計測するためのデバイスは、位置や角度（姿勢）を計測するためのデバイスと、形状を計測するためのデバイスの2つに大別することができる。

位置や角度を計測するためのデバイス

位置や角度を計測するためのデバイスとしては、米国ポヒマス社が開発したポヒマスセンサが代表的である [2]。これは、磁気を利用して対象の3次元的位置や角度を計測するデバイスで、電磁場を発生させる磁気ソースと、その電磁場を測定する磁気センサ、それらを制御するコントローラで構成される。磁気ソースと磁気センサは共に、互いに直交した3つのコイルで構成され、磁気ソースから発生した交流磁場の時間的変化を磁気センサで計測し、その時に磁気センサに流れる電流が、磁気ソースと磁気センサの相対的位置関係（距離と角度）によって決まることを利用して、磁気センサの3次元位置と角度を計測する。現在市販されているポヒマスセンサの最上位機種である 3SPACE システムの FASTRAK の仕様を Table 2.1 に示す。Table 2.1 により、機器保守等の訓練システムを構築する際に、ポヒマスセンサを訓練生の手に装着することにより、訓練生の手の位置や角度の変化を、ジェスチャとして認識するのに十分な精度で計測できることが分かる。このように磁気センサを用いる方式は、非接触で比較的高精度に位置と角度を計測できるため、多用されているが、近傍に建築構造物などの磁性体がある環境などでは、磁界が乱れ、計測精度が悪化する問題がある。

一方、磁性体等の影響を受けず、比較的広い範囲で3次元位置を計測できるデバイスに、光学式の位置計測デバイスがある [3]。このデバイスでは、2台以上のカメラを用いて対象を撮影し、画像処理と幾何学的な計算を行うことにより、対象の3次元位置を求める。こ

2.1 人工現実感の研究状況

の方法では、画像処理による対象点の位置検出を容易にするために、計測の対象となる位置に、マーカーとして再帰性反射材等を表面に張った小さな球や、赤外線 LED を取り付ける。現時点で販売されているシステムの中には、同時に 100 個以上のマーカーの動きを追跡することが可能なものもあるが、このような光学式の位置計測デバイスでは、計測の対象がカメラの死角に入った場合に、計測精度が極端に落ちるという問題がある。これを避けるためには、設置するカメラの数を増やすという方法があるが、完全には解決できない。

その他、磁気を利用したデバイスや、光学式のデバイスに比べ、安価に位置計測ができるデバイスとして、超音波を用いる方式のデバイス (Logitech 社 3D MOUSE) 等も開発されている。これは、超音波発信器からの信号が受信機に到着するまでの時間を計測し、両者の距離を求めるものであるが、超音波は透過性がないため、発信器と受信機の間には遮蔽物がなく、常に相対する姿勢で測定できることが条件となる。

以上のような 3 次元位置・角度を計測するデバイスは、次に示すような使用方法が考えられる。

1. デバイスを手に装着することにより、手の位置や角度を計測し、手を用いたジェスチャを認識する。
2. デバイスを頭に装着することにより、ユーザの頭の位置や角度に応じて提示する映像をリアルタイムに更新する。
3. デバイスを体全体に装着することにより、体全体を用いたジェスチャを認識する。

体全体にポヒマスセンサを装着し、人間の動作を計測するシステムとして、3次元モーションキャプチャシステム [4] がすでに市販されており、格闘技ゲームや映画作成等に広く使用されている [5]。

形状を計測するためのデバイス

形状を計測するためのデバイスとしては、手の形状 (手の指の曲げ角度) を計測するデータグローブが代表的である [6, 7, 8]。データグローブは、米国 VPL リサーチ社が開発したデバイスで、手袋の表面に、指に沿って光ファイバが配置された構造をしている。この光ファイバには、予め小さな傷が多数つけられており、指の曲げ角度に応じて、光ファイバに進入した光が漏れる仕組みになっている。従って、この光ファイバを通過した光量を計測することにより、指の曲げ角度を知ることができる。データグローブの計測精度は約 5 度で、各指に対して 2 本の光ファイバが配置してあり、指の第一関節と第二関節の曲げを計測できる。

また、同様に手の形状を計測するデバイスとして、光ファイバの代わりに柔軟性のあるベンドセンサを使用したデバイスも開発されている [9]。このベンドセンサは、現時点では非常に高価であるが、光ファイバを用いた場合に比べて高い精度で曲げ角度の計測が可能であり、計測精度は約 0.5 度となっている。

その他の計測デバイス

人工現実感を実現するシステムで使用されるその他の計測デバイスとしては、独国のSPACE CONTROL社が開発した3次元マウス Magellan が有名である [10]。Magellan は、バネで制御された1つのパックと9つのボタンで構成される。パックは手の平で握りながら操作し、位置と回転の6自由度 (X,Y,Z,Yaw, Roll, Pitch) の入力が可能である。Magellan は、データグローブやポヒマスセンサのような計測デバイスに比べ、自然なジェスチャの入力はできないが、計測精度が高く、また、卓上に設置して使用する機器であるため、ユーザが腕を上げなくても使用でき、疲れにくいという利点がある。

2.1.2 表示サブシステム

人間の感覚は、未解明の部分が多く、正確に分類することはできないが、人工現実感の分野では、視覚、聴覚、味覚、嗅覚、触覚のいわゆる人間の五感 [11] と、その他の感覚 (平衡感覚、痛覚等) の6種類の感覚に分類することが多い。人工現実感を実現する場合、これら全てについて実空間と同様の感覚を提示できることが望まれるが、現時点ではまだ実現されていない。特に、味覚と嗅覚に対しては、それらの感覚の発生のメカニズムが解明されていないこともあり、実用的な表示デバイスがほとんど開発されていない状況である。以下では、人工現実感の分野で比較的研究が進んでいると思われる、視覚、聴覚、触覚への感覚提示デバイスの研究状況をまとめる。

視覚ディスプレイ

人間が外界を認識するための情報の大半は、視覚から得られていると言われている。従って、表示サブシステムを構成するデバイスの中でも視覚への提示デバイスが最も重要視され、様々な形式のデバイスが開発されている。現時点で開発されている視覚提示デバイスは、頭部搭載方式、大スクリーン方式、立体ディスプレイ方式の3つの方式に分類できることが多い。

頭部搭載方式の視覚提示デバイスは、一般に Head Mounted Display (HMD) と呼ばれ、2つの小型ディスプレイをそれぞれ右目と左目用に配置し、このディスプレイに左右それぞれの映像を提示する [12, 13]。HMD は、常に目の前にディスプレイが固定されるため、使用者が頭を動かしても常に視界が仮想世界に覆われるため、没入感が得やすい利点がある。HMD は、光学レンズを組み込むことにより、視野角を広くでき、あたかも目の前に大型のスクリーンがあるかのような感覚を得ることができる。また、左右の目に両眼視差を考慮した映像を提示することにより、立体視も可能である。前述のポヒマスセンサと組み合わせることで、頭の動きに応じて提示する映像を更新し、没入感を高めることもできる。しかし、HMD は他の視覚提示デバイスに比べて重く、ユーザが疲労しやすいという問題がある。また、1人用のデバイスであるため、大人数が同時に使用する人工現実感システムには向いていない。

大スクリーン方式の視覚提示デバイスとしては、CAVE が代表的である [14]。これは、体験者の周囲 (前、左、右、下) に4面の強化ガラスでできたスクリーンを設置し、4台の

プロジェクタから各スクリーンに映像を提示するシステムである。このシステムは、多人数が同時に利用できるという利点があるが、広い場所が必要であり、装置が大がかりになるという問題がある。CAVEを簡略化したシステムとして、スクリーンをアーチ状に湾曲させ、広い視野角を確保するシステムも開発されている [15]。

立体ディスプレイ方式の視覚提示デバイスとしては、米国 StereoGraphics 社の CrystalEyes が代表的である [16]。これは、液晶シャッターメガネとディスプレイを組み合わせたシステムである。液晶シャッターメガネは、赤外線信号に応じて左右の液晶シャッターが交互に開閉する仕組みになっており、この開閉のタイミングに合わせてディスプレイに表示する映像を、右目用、左目用に切り替えることにより、両眼視差を考慮した映像を提示し、立体視を実現する。このシステムは、装着が容易であり、安価に多人数で利用できる立体視環境を提供できる利点があるが、ディスプレイ画面の範囲内しか映像を提示できず、広い視野角を提供するためには、大スクリーン方式の視覚提示デバイスと組み合わせる必要がある。

聴覚ディスプレイ

前述のとおり、視覚は人間が外界を認識する際の最も重要な感覚であるが、視覚はその有効範囲が制限されており、特に中心視野の周辺に向かって、視覚の感度は急激に悪くなることが知られている [17]。一方、聴覚は全方向に対して有効であり、その感度も方向によって大きく変化することはない。従って聴覚は、視覚が認識できない範囲を補うものとして重要な役割を果たしている。

臨場感を提示するための聴覚デバイスとして、最もよく用いられているのは2個のスピーカを用いてステレオ音場を生成するシステムである。この方法は非常に簡単に利用できるために、人工現実感を実現するシステムに限らず、古くから様々な分野で使用されている。

3次元音場を生成するシステムとしては米国 Crystal River Engineering 社の Convolvotron (コンボルボトロン) が人工現実感を実現するシステムで最もよく使用されている [18]。これは、音源から耳の鼓膜までの伝達関数と音源波形を畳み込み演算することで、耳の鼓膜の位置の音波形を合成するものである。この方法では、頭の位置が変化すると伝達関数も変化するため、予めデバイスを使用する範囲内の伝達関数を求めておき、頭の位置をポヒマスセンサ等の3次元位置センサで計測して、リアルタイムに伝達関数を変化させる試みがなされている。

触覚ディスプレイ

人工現実感を実現するシステムでは、人間の触覚をさらに詳細な感覚に分類し、各感覚に対して個々に刺激を提示するためのデバイスが開発され始めている。以下では、人間の触覚の内、触圧覚、温度覚、力覚の3つの感覚に対する感覚提示デバイスの研究状況について述べる。

触圧覚は、人間の皮膚にある3つの感覚受容器の内、機械的受容器に由来する感覚であり、体組織の機械的変形によって生じるものである [19]。例えば、指先で物を触った時に指先に物が当たったと感じる場合や、指先で物をなぞった時にざらざらしていることを感じ

2. 人工現実感技術を用いた訓練環境

る場合などがこれにあたる。このような触圧覚に対して刺激を提示するデバイスとしては、指先に小袋を配置し、細管を通じて空気圧により小袋を膨らませ、仮想物体に接触したことを知らせるデバイス [20] や、細ピンをマトリクス状に配置し、それらを様々な周波数で振動させることにより仮想物体の表面の質感を提示するデバイス [21, 22] 等が開発されつつある。しかし、現時点では、提示装置が大きい、他の装置との併用ができない、臨場感に乏しい等の問題があり、触圧覚を完全に提示できるデバイスはまだ開発されていない。

温度覚は、人間の皮膚にある3つの感覚受容器の内、温熱受容器に由来する感覚であり、細かくは温覚と冷覚の2つの感覚が存在する。具体的には、温点、冷点と呼ばれる温度を敏感に感じる点が皮膚上にあり、それらの感覚点が刺激されることにより温度を感じることができる。温度覚に対して刺激を提示するデバイスとしては、セラミック板でペルチェ素子を挟み、ペルチェ素子に流す電流の向きを制御することによって指腹面の冷却と加熱が可能であるデバイスが開発されている [23]。

触圧覚や温覚が体表面の皮膚や粘膜に生じる表面感覚であるのに対し、力覚は筋、関節、靭帯等に生じる深部感覚である。例えば、物を持ち上げた時にその物の重さを感じる感覚や、テニスボール等を握った時にその弾力性を感じる感覚等がこれにあたる。このような力覚に対して刺激を提示するデバイスは様々な種類のものが開発されている [24]。以下では力覚提示デバイスとして代表的な PHANToM、SPIDAR、HapticMaster について述べる。なお、これらのデバイスは力覚を提示する際に、対象の位置や角度の計測も同時に行っており、計測サブシステム、表示サブシステムの両方の性質を兼ね持つデバイスであるが、力覚を提示することを第一の目標として開発されているため、本論文では表示サブシステムに分類した。

PHANToM[25] は SensAble Technologies 社が開発した力覚提示装置で、3個のエンコーダと制御モータを組み合わせ、モータからのトルクをワイヤで駆動部分に伝える構造をしている。仮想物体の形、堅さや柔らかさ、面の荒さや滑らかさ等を感じることができ、主に手術のシミュレーションに使用できるデバイスとして、医学分野で注目されている。

Haptic Master[26] は筑波大学で開発された、6自由度入力のマニピュレータで、可動部が複数のリンクで並列に駆動されるパラレルメカニズムを採用している。Haptic Master の稼働範囲は、直径約 40cm の球内で、各リンクに取り付けたポテンショメータにより角度を計測し、手先の位置と姿勢を求め、物体の硬い表面や、弾性のある表面等を提示する。

SPIDAR[27] は東京工業大学の佐藤らによって開発されたシステムで、指に装着するキャップに4本の糸を張り、その糸の長さをロータリーエンコーダにより計測することにより指の位置を算出し、糸をモータで引っ張り、そのトルクを制御することによって力覚を提示する。SPIDAR に関する研究は、現在も精力的に進められており、8本の糸を用いて同時に両手の指に対して力覚を提示することにより、両手で物体を掴んでいる感覚を提示するデバイスへの拡張 [28] や、可動範囲を広げることにより、腕に対して力覚を提示する BIG SPIDAR 等の開発 [29] も行われている。

以上、力覚を提示する代表的なデバイスについて述べたが、PHANToM や Haptic Master は、手や指の位置を正確に計測し、比較的臨場感の高い力覚フィードバックを提供できるが、可動範囲が狭いという欠点がある。また、SPIDAR は、可動範囲を広く取ることができるが、反面、手や指の位置の計測精度が悪く、臨場感の高い力覚フィードバックが提供

できないという問題がある。また、これら3つの力覚提示デバイスは、手や指のある一点に対して力覚を提示するデバイスである。これに対して、面で力覚を提示するデバイスは研究段階であり、実用的なデバイスはまだ開発されていない。

2.1.3 シミュレーションサブシステム

シミュレーションサブシステムは、計測サブシステムと表示サブシステムの間で、情報の変換と受け渡しを行うサブシステムである。例えば、システムのユーザが、計測サブシステムを介して本を持ち上げるジェスチャを行った場合、表示サブシステムでは、例えば、視覚提示デバイスに対して、仮想空間内の手に相当するモデルが本に相当するモデルを持ち上げる映像を出力しなければならない。また、システムによっては、本の重さをユーザに提示するための信号を、力覚提示デバイスに対して出力する必要がある。このように計測サブシステムからの入力と、表示サブシステムへの出力の因果関係を決定するのがシミュレーションサブシステムである。シミュレーションサブシステムを構築する場合、仮想空間に関する情報を如何に保持し、更新し、表示サブシステムで使用できる形の情報に変換するかが問題となる。

仮想空間に関する情報の保持

シミュレーションサブシステムが保持する必要がある仮想空間の情報は、表示サブシステムで使用するデバイスの種類によって大きく違ってくる。例えば、表示サブシステムで視覚に対してのみ表示を行うのであれば、シミュレーションサブシステムでは、仮想物体の位置や角度、表面の形状や色などの情報を保持すればよい。しかし、表示サブシステムで力覚提示や触覚提示を行うのであれば、仮想物体の重さに関する情報や、仮想物体の材質に関する情報等も保持する必要がある。また、視覚に対してのみ表示を行う場合でも、次に述べる仮想空間の情報の更新方法によっては、仮想物体の重さや材質に関する情報も必要となる。このように、表示サブシステムで使用するデバイスの種類や、仮想空間の情報の更新方法によって、シミュレーションサブシステムが保持する必要がある情報の内容は大きく変化するが、情報の保持方法としては、仮想物体の表面の情報のみを保持する場合と、内部まで含めた情報を保持する場合の2通りの保持方法に分けることができる。

仮想物体の表面の情報のみを保持する方法としては、ポリゴン (Polygon) を用いて仮想物体の表面形状を保持する方法と、テクスチャ(Texture) を用いて仮想物体の表面の色や、質感を保持する方法を併用して用いるのが一般的である [30]。

前者は、仮想物体の表面を多数の多角形 (ポリゴン) の集合として捉え、各多角形の頂点の3次元座標を、仮想物体の表面形状として保持する。この方法は、実現するための計算量が比較的少なく、また使用する際に必要となるメモリの量も少ないため、3次元アニメーションを作成する分野では古くから用いられている方法である。しかし、球などの曲面が多い物体の形状を保持するためには、数多くのポリゴンを用いなくてはならず、また、完全な曲面は保持できないという欠点もある。

後者は、仮想物体の3次元表面と2次元平面を座標変換で対応づけることにより、3次

元に分布する仮想物体の表面の材質や色などの情報を2次元に分布する情報(テクスチャ)として格納する方法である。テクスチャとして物体表面の色分布を指定する方法は比較的容易に写実的な映像を作成できるために広く用いられている。近年は、カメラ等で撮影した実映像を、テクスチャとして用いることにより、非常に臨場感の高い映像を作成する方法も提案されている [31]。

仮想物体の内部の情報も含めて保持する方法としては、仮想物体をボクセルと呼ばれる小さな立方体の集合として捉え、各立方体について、質量や材質、弾性係数、温度等の情報を保持する方法が一般的である [32]。この方法は、仮想物体の変形や切断などを行う必要がある、手術のシミュレーションなどに用いられる場合が多い [33]。また、実空間に近い仮想空間を提供できる利点がある一方で、シミュレーションを行う際の計算量が非常に多く、リアルタイムで処理することが難しいという問題がある。

仮想空間の情報の更新

計測サブシステムからの入力に応じて仮想空間の情報を更新する場合、物体相互間の拘束関係等を考慮して更新する必要がある。例えば、仮想空間内にドアがあり、計測サブシステムからドアを開けるジェスチャが入力された場合、ドアは壁面に対して蝶番によって自由度1で回転し、その可動範囲もドアのストッパまでに制限されなくてはならない。このような物体相互間の拘束関係等を考慮して仮想空間の情報を更新する1つの方法として、シミュレーションサブシステムが、仮想物体の詳細な表面形状や質量、弾性係数等の情報を保持し、物体間の詳細な接触判定や運動方程式等の利用を通して仮想物体の動きを制御する方法も考えられるが、計算量が膨大になり、現時点でのコンピュータの能力では、リアルタイムに処理するのは非常に困難である。従って、物体相互間の拘束関係等を考慮して仮想空間の情報を更新するためには、計算によって物体相互間の拘束関係を調べるのではなく、予め仮想物体の各状態に応じて、物体相互間の拘束関係がどのようになるのかを定めておく必要がある。このような仮想空間における物体相互間の拘束関係を定義する作業は、非常に多くの時間や労力が必要になる。そのため、複雑な仮想空間を構築する場合は、何らかの作業支援システムが必要となる。

これに対し、近年、データグローブや3次元マウス等を用いて、実空間と同様に仮想物体を操作できる仮想空間を構築する作業を支援するシステムがいくつか開発されてきている [34]。特に、GUI(Graphical User Interface)を介して仮想空間を構築できるシステムとしては、「RealMaster」や「dVISE」等が開発されている。「RealMaster」は株式会社ソリッドレイ研究所が開発したシステムで、プログラミングを行わなくても物体の操作ができる仮想空間の構築を可能にしている [35]。また、「dVISE」は英国 Division 社が開発したシステムで、メニューによる対話入力、プログラミングなしに仮想空間の構築、操作を可能にしている [36]。これらのシステムを用いることにより、コンピュータに関する高度な知識のない人でも「ドアのノブを触るとドアが開く」「レバーを操作するとクレーンが動く」といった、単純な仮想空間は容易に構築できる。しかし「RealMaster」は、インタラクティブな仮想空間を構築するために、「イベント」と「アクション」の関係を記述し、各物体の運動の仕方等の細かい設定を行う「シーンファイル」と呼ばれるテキストを作成する必要

がある。また「dVISE」は、「イベント」と「アクション」の関係を全てGUIを介して設定できる機能を備えているが、設定すべき「イベント」と「アクション」の関係が複雑になった場合に、それらを効率的に扱うための機能は備わっていない。そのため、作業手順が非常に複雑なモータやポンプ等の機器を分解・組立できるような仮想空間を構築する場合には、複雑な構築作業を行う必要があり、不可能ではないが多くの時間が必要であり、今後の改善が必要である。

仮想空間の情報の提示

表示サブシステムを介して、ユーザに仮想空間の情報を提示する場合、その情報を如何にユーザが理解しやすい形に変換するかが問題となる。例えば、仮想空間の情報として、ユーザが手に本を持っているという情報があり、これをユーザに提示する場合を考える。この場合、視覚に対して情報提示するためには、本の位置や角度、本の形状に関する情報を人間の視覚で知覚できる映像に変換しなくてはならない。また、力覚に対して情報提示するためには、本の形状や重さに関する情報を、力覚提示デバイスで使用できる電気信号に変換しなくてはならない。一般にこの情報の変換はレンダリングとよばれるが、特に、視覚に対する情報のレンダリング方法が数多く提案されており、仮想物体の前後の位置関係を考慮して3次元映像を描く方法である、Z-バッファ法、スキャン・ライン法、レイ・トレーシング法等がよく使用されている [37]。

Z-バッファ法は、画面を構成する各ピクセルごとにそのピクセルに過去に描かれたポリゴンの奥行き情報を記録するZバッファと呼ばれるバッファを用意する。この方法では、仮想物体を構成する個々のポリゴンを順番にレンダリングし、レンダリングの結果、描かれるピクセルの奥行きと、そのピクセルに相当するZバッファに記録された奥行きを比較し、Zバッファに記録した奥行きの方が大きければ（視点からの距離が遠ければ）、そのピクセルに対してレンダリングした結果を描き、Zバッファも更新する。この方法では、物体を奥行き情報によって事前に分類する必要がなく、順番に関係なく物体をレンダリングできる。比較的アルゴリズムが簡単であり、描画可能なポリゴン数に制限がないため、非常に複雑な物体でも描くことができる。

スキャン・ライン法は、画面上の走査線（スキャン・ライン）と視点を含む2次元平面と仮想物体を構成する各ポリゴンとが交差する線分を計算し、この線分の前後関係を比較し、手前にある線分だけを描く方法である。Z-バッファ法に比べ計算量や必要なメモリ量が少なく、高品質な3次元コンピュータグラフィックスを作成する分野で最も頻繁に使用されている方法である。

レイ・トレーシング法は、光線追跡法あるいは視線探索法ともよばれ、視点から画面上の各ピクセルを通して仮想物体に向かう光線を考え、その光線の仮想物体に対する反射、透過、屈折を追跡し、ピクセルの色を決定する方法である。レイ・トレーシング法は、すべてのピクセルについて個々に計算するため、計算量が非常に多くなるが、高品質な映像を合成できる。また、計算を各ピクセルごとに独立して行えるため、並列計算により高速化することも可能である。

2.1.4 人工現実感の問題点と課題

本節では、人工現実感の現在の研究状況について、計測サブシステム、シミュレーションサブシステム、表示サブシステムの3つのサブシステムに分けて展望した。

計測サブシステムを構成するデバイスの内、人の運動を計測するデバイスに関しては、既に十分使用に耐えうるものが開発されていると言える。特に、ポヒマスセンサやデータグローブは、ユーザのジェスチャを計測するのに十分な精度と速度を持っていると言える。また、表示サブシステムに関しても、視覚や聴覚に対して刺激を提示するためのデバイスは、十分使用に耐えうるものが開発されていると言える。

しかし、一方で、表示サブシステムを構成するデバイスの内、触覚に対して刺激を提示するデバイスは、様々な試みがなされているものの、臨場感が欠如していたり、装置が大規模になるなど、解決すべき問題が多々ある。また、個々の感覚の臨場感の向上に重点を置いて研究が進められている傾向があり、他のデバイスとの併用が考慮されていないため、複数の感覚を同時に提示できない場合が多い。特に、装置が大規模であるために、複数のデバイスを装着すること自体が不可能な場合があり、今後改良する必要がある。

また、シミュレーションサブシステムに関しては、近年の計算機の飛躍的な性能の向上により、大量の情報を高速に処理することが可能になってきているが、その情報を生成するための方法が未だ確立されていないと言える。例えば、物体相互間の拘束関係を考慮した仮想空間を構築するためには、人の手で1つ1つ物体間の関係を定義せざるを得ない状況である。

このように、現時点では、仮想空間を実空間と区別できないほどの臨場感を持ってユーザに提示することは不可能である。しかし、人工現実感技術を用いたシステムを利用する目的によっては、ある程度、臨場感が低くても問題がない場合が多い。従って、今後、人工現実感を用いたシステムを開発する場合、その目的に応じて、計測すべきユーザの情報や、提示すべき感覚の種類を適切に選択し、使用するデバイスの組み合わせを決定することが重要である。

2.2 原子力発電プラントの運転および保守作業の訓練

本節では、原子力発電プラントの運転を行う運転員と保守を行う作業員の能力の維持・向上を目指して実施される、訓練センターにおける教育訓練の現状と問題点について述べる。

原子力発電プラントの保守作業を行う場合、機器の点検と故障個所の修理を確実にを行うために、保守作業員は Table 2.2に示す項目を修得することが要求されている [38]。

これに対し、原子力保守訓練センターでは、新たな保守作業員を育成するための初期訓練コース、既に現場で保守作業を行っている作業員の技術の維持・向上を目指した再訓練コース等を設け、毎年、数百人にも上る訓練生を教育している。この内、特に初期訓練コースを受講する訓練生は、保守作業に対する知識を全く持っていないため、数週間に渡って、座学形式の講義を受け、保守作業の基礎知識を学んでいる。その後、初期訓練コースの後半から実物大模型を用いた実技訓練が行われ、訓練のインストラクタの指導のもとで、様々

2.2 原子力発電プラントの運転および保守作業の訓練

Table 2.2: 保守作業員が修得すべき課題

| | |
|---|-------------------------------|
| 1 | 機器の構造・各部機能に関する知識 |
| 2 | 工具に関する知識と使用する際の技術 |
| 3 | 作業手順の知識（分解・点検・交換・組立・試運転の手順） |
| 4 | 保守作業の工程計画能力（想定外の状況に対する計画立案能力） |
| 5 | 機器等の異常判断能力 |

Table 2.3: プラント運転員が修得すべき課題

| | |
|---|-----------------------------|
| 1 | 原子炉物理、熱水力学等の原子力の基本的な知識 |
| 2 | プラント設備の知識 |
| 3 | プラントの起動と停止操作の手順 |
| 4 | 異常の診断能力 |
| 5 | 異常事象に対する対応操作能力 |
| 6 | 様々な状況に応じて冷静かつ的確に判断し、対応できる能力 |

な工具の使い方や原子力発電プラントを構成する代表的な機器の分解・点検・組立作業の方法等を学んでいる。

一方、原子力発電プラントの運転操作を行う場合、原子力発電プラントの安全、安定運転を確保するために、プラント運転員は、Table 2.3に示すように、幅広い知識と技術、論理的な思考能力を修得することが要求されている [39]。

これに対して、原子力発電訓練センター（NTCやBTC）では、原子力発電プラントの新たな運転員を育成するために、毎年、専門の初期訓練カリキュラムを設け、約22週間に渡る長期の教育訓練を実施している [40]。ここでも、訓練生は、前半は原子力発電の基礎知識や原子力発電プラントの構造および設備、プラントの起動と停止操作の手順、異常発生時の対応方法等を座学形式の講義で学び、後半はプラントシミュレータと実物大模型等を用いた実技訓練を受ける。この教育訓練により、訓練生は最終的に原子力発電プラントを運転できるだけの知識と技術を身につけることができるが、やはり、運転員の養成および能力維持向上のために、シミュレータを用いた実技訓練は中心的な役割を果たしている。特に、訓練の初期段階では、訓練生は初めて原子力発電プラントの制御盤を操作することになるので、プラント制御盤の配置や、各機器の操作方法、実際の操作の手順等を理解するまでに、長期間、実技訓練を行う必要がある。また、プラント運転の訓練で訓練の対象となるプラントの種類は通常1種類であるが、多種多様な異常事象に対して、十分深く理解している必要があることが1つの特徴である。

このように、機器保守の訓練、プラント運転の訓練は共に、実技訓練が訓練の重要な部分を占めているが、実物大模型等を用いた訓練はTable 2.4に示すような問題が存在する。

2. 人工現実感技術を用いた訓練環境

Table 2.4: 実物大模型を用いた実技訓練の問題点

| | | |
|---------------|------|------------------------------|
| 運転・保守 訓練共通 | 問題 1 | 体験訓練のための実物大模型等が必要であり訓練費用が高い。 |
| | 問題 2 | 実物機器を用いるため、訓練環境の変更が困難である。 |
| | 問題 3 | 訓練に比較的広い場所が必要である。 |
| | 問題 4 | 訓練中に緊張感を持続するのが難しい。 |
| | 問題 5 | インストラクタが常に付き添う必要がある。 |
| 保守訓練 | 問題 6 | 実際の保守作業と同等の危険性がある。 |
| | 問題 7 | 訓練を受ける人の数が多く、現状では訓練設備が足りない。 |
| | 問題 8 | 模型では訓練環境を構築することが難しい訓練がある。 |

すなわち、運転と保守に共通した問題として、実物大模型を作成するためのコストが非常に高いこと、訓練環境の変更が困難であること、インストラクタが常に付き添う必要があること等が挙げられる。特に、保守訓練の場合には、訓練の対象となる機器の種類が非常に多く、全ての機器に対して実物大模型を作成するためには膨大な費用が必要となる。また、保守訓練に特有の問題として、実物機器を用いるために実際の保守作業と同等の危険性があること、訓練を受けるべき作業員の数が非常に多く、訓練設備を数多く設置する必要があること、故障した機器を修理するための環境等、模型では再現することが難しい訓練があること等を挙げることができる。特に模型では、壊れた状態にある機器などを、常に同じ状態で繰り返し再現することが難しく、繰り返し訓練を実施することが困難であるという問題がある。

以上述べた理由等により、保守と運転の両方の場合で、十分な実技訓練が実施できておらず、代表的な機器、代表的な異常事象に対する実技訓練のみが行われているのが現状である。従って、これらの問題を解決する新しい訓練方法が必要とされている。

2.3 人工現実感技術を用いた訓練環境

人工現実感の大きな特徴の1つは、実際には存在しない環境に対して、あたかもそれが目の前に存在しているかのような感覚を持って、体験できることである。すなわち、体験したい対象の環境を実空間に「存在するもの」として構築するのではなく、仮想空間内に「情報」として構築することによって、実空間で体験する代わりに、仮想空間で体験できるということである。「情報」として構築された環境は、「情報」が持つ利点を兼ね備えた環境となり、Table 2.5に示す利点を持つ。従って、人工現実感技術を用いて訓練環境を構築することにより、2.2節で述べた、実物大模型を用いて実技訓練を実施する場合の問題点に対し、Table 2.6に示すような改善が期待される。

実際に、人工現実感技術を用いた訓練環境は、これまでに幾つか開発されている。例えば、人工現実感技術を用いた保守作業の訓練を行う環境の構築事例としては、1993年に衛星軌道上で実施されたハッブル宇宙望遠鏡の修理作業のための訓練システムの開発が有名

2.3 人工現実感技術を用いた訓練環境

Table 2.5: 人工現実感を用いて訓練環境を構築することの利点

| | |
|------|----------------------------------|
| 利点 1 | 構築可能な空間の大きさに制限がない。 |
| 利点 2 | 実空間では危険な環境も仮想空間内に構築すれば安全に体験できる。 |
| 利点 3 | 実空間では構築が困難もしくは不可能な環境も構築できる。 |
| 利点 4 | 全く同じ環境を何度も繰り返し再現できる。 |
| 利点 5 | 大抵の場合、環境の変更が実空間に比べ容易に行える。 |
| 利点 6 | 仮想空間の情報を複製することにより複数の環境を容易に構築できる。 |

Table 2.6: 人工現実感を用いることによる問題点の改善

| | |
|------|---|
| 問題 1 | 実物大模型を作成する費用を削減できる。 |
| 問題 2 | 仮想空間の情報を変更することにより容易に訓練環境を変更できる。 |
| 問題 3 | 構築可能な訓練環境の大きさに制限がなく、計算機と入出力デバイスを使用するための場所があれば訓練ができる。 |
| 問題 4 | 実際の作業場所に近い雰囲気 of 訓練環境を構築することにより、緊張感が持続する可能性がある。 |
| 問題 5 | 計算機上に訓練環境を構築しているために、訓練環境にインストラクタの代わりとなる様々な学習支援機能を付加できる。 |
| 問題 6 | 実在しない機器を使用しているために安全に訓練を実施できる。 |
| 問題 7 | 訓練環境の情報を複製することにより、入出力デバイスと計算機を複数設置すれば、複数の訓練環境が得られる。 |
| 問題 8 | 実空間では再現が困難もしくは不可能な訓練環境も構築できる。 |

2. 人工現実感技術を用いた訓練環境

である [41, 42]。この研究は、宇宙空間での非常に危険な作業に対する訓練を、人工現実感技術を用いて安全・安価に実施することを目指したものであり、広視野感星カメラの交換作業や、太陽電池アレイの交換作業、補正レンズ装置の交換作業等、6つの異なった訓練シナリオ用の環境が作成され、実際の任務遂行のために開発された人工現実感技術を用いた訓練環境としては、過去最大の100人を越すフライトチームのメンバーを訓練するために使用され、後のアンケート調査でもその有効性が確認されている。また、三輪らは、変電所係員の安全に対する意識向上を目的として、仮想空間内に変電所全体を構築し、感電等を伴う危険な行為を体験させるシステムを開発している [43]。この研究では、作業環境の臨場感が重視されるとともに、危険個所の学習、各種作業手順に従った作業の体験、各種機器に関する情報提示機能などが実現されている。HMDやデータグローブを用いて変電所内を歩き回ることや機器を操作することもできる。

また、プラント運転員を訓練するためのシステムとしては、吉川ら [44]、松原ら [45] の研究を挙げることができる。吉川らは、原子力発電プラントを運転する際に用いられる知識およびそれらのつながりを仮想空間内に描画することにより、訓練生がプラントシミュレータを運転しながら、常に知識の全体像の観察を可能にすることによって、原子力発電プラントに対する深い理解を効率良く支援することを目指したシステムを開発している。松原らは、ITS(Intelligent Tutoring System)と発電プラントの制御盤を模擬する仮想環境を組み合わせたシステムを開発し、HMDにより立体的な制御室環境を提示するとともに、2次元マウスや3次元マウスを用いることで、仮想空間内で、発電プラントの制御盤の操作やウオークスルーを可能にしている。このシステムでは特に、ITSにより生徒モデルを用いて訓練生の理解度を推定し、教示戦略を変更することで効果的な教示支援を実現していることが特徴である。

以上のように、人工現実感技術を用いた訓練システムが開発されているが、これらの訓練システムを実用化するにあたり、まだ問題が残されている。まず、係員作業員を訓練するためのシステムに関しては、以下に示すような問題がある。

- 仮想空間の中で実施できる作業を、多少の自由度はあるものの正しい手順に限定して訓練システムが構築されている。

仮想空間の中で実施できる作業が、正しい手順に限定される場合、訓練生が試行錯誤して次に行うべき作業を考えることがなくなるため、訓練が受け身になり、作業手順の学習や、作業工程の計画の訓練効率が下がると考えられる。実技訓練が重要視される理由の1つは、訓練生が受け身になってインストラクタの話聞くだけでなく、自ら判断して作業を進める機会を経験させる為である。従って、機器保守の訓練システムを開発する場合、間違った作業でも実施でき、間違った作業を実施した場合に、その結果を経験できる環境を構築することが重要である。

- 訓練システムを開発する際の労力が考慮されておらず、新たな訓練システムを開発するには膨大な時間と労力が必要である。

これまでの機器保守の訓練システムの開発は、「どのような訓練環境を構築すべきか」という問題を中心に研究が進められきた。これにより、ある程度使いやすく、訓練の

2.4 本研究の着眼点と範囲

効果を上げることができる訓練システムを開発する基盤ができつつある。しかし、これまでに提案された訓練システムの開発手法では、訓練システムを開発する際の労力については考慮されていないため、新しい訓練システムを開発する場合、膨大な時間と労力が必要になるという問題がある [46]。しかし、訓練システムを使用するのは、原子力発電を運営する電力会社、もしくは、作業員の訓練を委託される訓練センター等の民間企業が大半である。従って、例え訓練システムが非常に優れており、短時間で訓練の成果を上げることができるものであっても、訓練システムを構築する際に非常に多くの労力と時間、そして費用が必要な場合は、実際の訓練に使用することは難しい。特に、機器保修の訓練では、訓練の対象となる機器の種類が多いため、大きな問題となる。

また、プラント運転員を訓練するためのシステムに関しては、以下のような問題がある。

- 予備知識を殆ど持たない訓練生が最初に使用する訓練システムとしては向いていない。これまでに開発されたプラント運転員を訓練するためのシステムは、訓練生が自ら制御盤を操作し、運転操作を仮想体験させることを念頭に開発されたものばかりであった。すなわち、既存の知識を利用して要求される作業を遂行する訓練に重点が置かれていた。しかし、原子力発電プラントの異常診断を正確かつ迅速に行うには、非常に高度な知識と経験が要求され、十分な知識と経験を持たない訓練生が、試行錯誤的にプラントの異常診断を行っても、正しい思考過程や作業手順を教示されない場合、訓練効率は上がらないことが予想される。従って、プラントの運転作業を体験できるだけの訓練環境は、予備知識を殆ど持たない訓練生を教育するシステムとしては適していない。訓練生の欠けている知識や経験を積極的に補う訓練方法が必要である。

2.4 本研究の着眼点と範囲

本研究では、前節までに述べた、人工現実感の研究状況、原子力発電プラントの運転員と保修作業員に対する訓練の現状、及び、これまでに開発された人工現実感技術を用いた訓練システムの問題点を踏まえ、まず始めに、「作業の自由度が高い機器保修の訓練環境を少ない労力で短時間で構築できる手法を確立すること」を目指し、訓練環境の新しい構築方法に基づいた機器保修の訓練システムを開発を行う。

機器保修の訓練環境を構築する場合、訓練生のジェスチャに応じた仮想物体の運動の仕方を設定する作業が、最も複雑で、時間と労力が必要な作業である。つまり、この作業が繁雑であるために、過去に開発された保修作業員を訓練するための訓練システムでは、訓練生が自由に行動できる訓練環境が構築されていないのであり、また、新たに訓練環境を構築し直す場合でも、膨大な時間と労力が必要になる。従って、訓練生のジェスチャに応じた仮想物体の運動の仕方を容易に少ない労力で設定できるようになれば、訓練生が自由に行動できる訓練環境を容易に構築でき、更に、新たに訓練環境を構築する場合でも、それほど多くの時間と労力は必要なくなると考えられる。具体的には、これまで、機器保修の訓練環境を構築する場合、個々の仮想物体の物理的状態を表す多数のフラグを設け、訓練

2. 人工現実感技術を用いた訓練環境

生のジェスチャに応じた各フラグの変化の方法と、各フラグの状態と訓練生のジェスチャに応じた仮想物体の運動の仕方をプログラミングにより設定していた。しかし、この方法では、構築すべき訓練環境が複雑になった場合に、フラグの数が増加し、その作成、修正、管理が非常に困難となり、機器保修の訓練環境を構築する際に膨大な時間と労力が必要となる。

そこで本研究では、まず、仮想物体の取りうる状態と、訓練生のジェスチャに応じた仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する方法と、作成したペトリネットを用いて複雑な機器保修作業を行うことができる仮想空間をシミュレーションする方法を提案する。ペトリネットは、プラントのプロセス、コンピュータの回路動作、生産ラインの物の流れ、流通における物流など、システムにおける離散的、並列的、並行的な状態遷移のモデル化に用いられている。ペトリネットの特徴は、「グラフィカルな視覚的ツール」、「シミュレーションツール」、及び「数学的方法論」の三つの機能を同時に備えていることであり、ユーザインタフェースの観点からは強力なコミュニケーション媒体としてもその威力を発する。ペトリネットに関する研究は、ペトリネットの生みの親であるカール・アダム・ペトリ (Carl Adam Petri) 博士が1962年に学位論文 [47] を発表して以来、めざましい発展を遂げており、その用途に応じて様々な形で拡張されていて、代表的なものには、「時間ペトリネット [48]」「確率ペトリネット [49]」「カラーペトリネット [50]」等が提案されている。機械の組立工程の表現にペトリネットを用いた研究は数多く発表されているが [51, 52, 53]、仮想空間における物体の物理的制約を含めた作業手順の表現にペトリネットを用いた例は見あたらない。

仮想物体の状態遷移をペトリネットを用いてモデル化することによる利点をまとめると以下のようなになる。

- 仮想物体の状態遷移の仕方をグラフとして視覚的に描画することが可能になり、仮想空間が複雑になった場合でも、その作成、修正、管理が容易になる。
- ペトリネットを構築する際の作業を支援するためのアプリケーションを開発することにより、仮想物体の状態遷移の仕方を定める作業を支援することが可能になる。
- ペトリネットを数学的に処理することにより、作業手順の検索等が可能になり、訓練を支援するための新しい機能を付加することが可能になる。

なお、Table 2.2に示したように、保修作業員が修得すべき課題は、訓練対象物の構造や部品構成、作業手順等の保修作業の知識と、工具や器具の持ち方、作業時の力の入れ方等の保修作業の技能の大きく2つに分けることができ、その両方について修得できる訓練環境を構築することが望ましい。しかし、技能を修得するためには力覚フィードバックの機能を備えた入力デバイスの使用が不可欠であると予想されるが、2.1節で述べたように、現時点では機器保修の訓練システムに使用できるような力覚フィードバックの機能を提供できる入力デバイスはまだ開発されていない。そこで、本研究では、力覚フィードバックの機能を提供できる入力デバイスの開発は将来課題とし、主に、訓練生の視覚と聴覚に対して情報提示を行うことにより、機器保修作業の知識を修得できる訓練システムを開発する。

2.4 本研究の着眼点と範囲

さらに本研究では、機器保修の訓練環境は、実際の訓練の現場ではどのような訓練環境が必要であるかを理解している訓練のインストラクタが構築すべきであると考え、計算機に関する高度な知識を持たないインストラクタが、容易に機器保修の訓練環境を構築できるようにするための支援システムを開発する。具体的には、仮想空間内に配置する機器や工具の、訓練生のジェスチャに応じた運動の仕方を、予め定められたパラメータを設定することにより設計できる手法を考案し、この手法と、訓練生のジェスチャに応じた仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する手法を組み合わせ、プログラミングを全く行うことなく、GUIを介して機器保修の訓練環境を構築できる支援システムを開発する。

次に、プラント運転員の訓練に関して、訓練生がプラントの異常診断と対応操作に関する十分な知識や経験を持っていない場合、「訓練生に複雑な異常診断過程を理解させる新しい訓練方法」として、「他人が作業を行う様子を観察」し、さらに「他人が作業を行う様子を体験」することが、最も効率的に必要な知識を得ることができる方法であると考えた。そこで本研究ではまず、人工現実感技術を用いて、プラント監視制御行動中の運転員の行動を、その思考過程や機器等の詳細な参照行動まで、総合的に情報可視化し、その結果を訓練生に提示することによりプラント運転の訓練効果を上げることを目指した体験型教育システムの開発を行う。ここでも、訓練生の視覚と聴覚に対する情報提示を行うが、具体的には、仮想空間内で人の自然な動きを自由に合成できる人体モーションの合成技術を開発し、原子力発電プラントの異常発生時における運転員の思考過程と行動を、計算機で模擬できるヒューマンモデルシミュレータを用いて、仮想空間内で実際の作業に近い自然な動作でプラントの異常診断と対応操作を行う仮想運転員を構築する。さらに、仮想運転員の視点からの映像を立体視しながら眺めることができる機能を構築する。すなわち、仮想運転員がプラントの異常診断と対応操作を行う様子を、訓練生が「仮想体験」できる訓練システムを開発する。これにより、訓練生が、あたかも自ら制御盤を操作しているような感覚を得ることができ、制御室内の配置や、実際の作業の様子を効率的に理解できると期待される。

関連研究としては、高野らが、原子力発電プラントにおける異常発生時のプラント運転員のチームとしての行動をモデル化する研究 [54, 55] を行っており、多重事故時における4名のプラント運転員（当直長1名と補機運転員3名）の運転員間の会話も含めた行動を計算機上で模擬できるシステムを開発している。このシステムでは、模擬した運転員の行動を2次元映像として観察できる機能が構築されているが、シミュレーション結果を確認する目的で構築されたものであり、運転員の思考過程を訓練生に理解しやすく提示する機能等が構築されていない。また、プラント運転を体験する機能も構築されていないため、訓練生がプラントの異常発生時の診断や対応操作の方法を学ぶためには、システムの機能を拡張する必要がある。

また、土井らは、人工現実感技術を用いて仮想空間内に発電プラントの制御室を構築し、その中で制御盤を操作する仮想運転員の動作を合成するシステム (VIGOR) を開発している [56]。VIGORでは、仮想運転員が制御盤を操作する際の姿勢や視界を評価することにより、仮想空間内に構築された制御盤等の設計を評価する機能が構築されている。しかし、VIGORでは、仮想運転員の行動は、データグローブやマウス等を介して人間が指示する

必要があり、仮想運転員が自らの判断でプラントを操作する機能は構築されていない。

このように、原子力発電プラントの異常発生時における運転員の行動と思考過程を模擬できるヒューマンモデルを利用して、仮想運転員の行動を3次元映像として視覚化し、また、仮想運転員の視点からの映像を体験させることによって、訓練生にプラントの異常診断と対応操作を学習させるシステムはまだ開発されていない。

2.5 まとめ

本章ではまず、人工現実感技術の研究状況を、計測サブシステム、シミュレーションサブシステム、表示サブシステムの3つのサブシステムに分けて展望した。計測サブシステムを構成するためのデバイスの内、人の運動を計測するデバイスは、実用に耐えるものが開発されており、機器保守やプラント運転の訓練システムを構築する際に、使用可能であると考えられる。表示サブシステムに関しては、人の視覚と聴覚に対して情報提示を行うデバイスは、実用的なものが数多く開発されているが、触覚に対して情報提示を行うデバイスは、まだ研究段階であり、臨場感に乏しく、機器も非常に大きく使いにくいいため、訓練システムを開発する際には、使用することが難しいと言える。また、シミュレーションサブシステムに関しては、現在の仮想空間のシミュレーション方法では、物体相互間の拘束関係を考慮した仮想空間を構築するためには、必要な情報を1つ1つ作成する必要がある、非常に煩雑である。このように、現時点では、実空間と区別できないような臨場感の高い仮想空間を構築することは難しく、必要に応じて最適なデバイスの組み合わせを選択する必要がある。

次に、原子力発電プラントの運転および保守作業の訓練の現状について述べ、現在の実技訓練に対する問題点をまとめた。現在の訓練の実施方法では、基本的に実技訓練を実施するための設備と時間、インストラクタが不足しており、訓練生が個人で実技訓練の代わりとなる訓練を実施できる新しい訓練方法が必要である。

一方、人工現実感技術を用いて訓練システムを開発することにより、現在の実技訓練の問題点を改善できることが期待される。しかし、これまでに開発された人工現実感技術を用いた訓練システムは幾つかの問題が存在する。すなわち、保守作業員を訓練するためのシステムは、その訓練システムを構築する際の労力が考慮されておらず、数多くの機器に対する訓練環境を構築する必要がある現状では、そのままでは実用化できない。また、プラント運転員を訓練するためのシステムは、訓練生が自ら作業を行うことを前提とした訓練システムが構築されており、予備知識を持たない訓練生が初めて使用する訓練システムには向いていない。

そこで、本研究では、まず、機器保守の訓練システムを少ない労力で短時間で開発できる新たな手法として、仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する手法と、作成したペトリネットを用いて仮想空間のシミュレーションを行う手法を提案し、実際にこの手法を用いて機器保守の訓練システムを開発する。さらに、本研究では、機器保守の訓練環境を、プログラミングを全く行わずに構築できる手法を考案し、実際にGUIを介して必要な情報を入力することにより、目的の訓練環境を、計算機等の高度な知識を持

2.5 まとめ

たない人でも容易に構築できるようにするための支援システムの開発を行う。

また、プラント運転員を訓練するシステムに関しては、プラントの運転を自らの判断で行うことができない訓練生でも使用できる訓練システムを開発すべきであると考え、原子力発電プラントの異常発生時の運転員の行動と思考過程を模擬できるヒューマンモデルを利用することにより、仮想空間内で、実際の作業に近い自然な動作でプラントの異常診断と対応操作を行う仮想運転員を構築する。そして、訓練生が仮想運転員の作業の様子を観察することにより、実際の作業の様子を把握できる教育システムの開発を行う。この教育システムでは、訓練生は、仮想運転員の視点からの映像を立体視しながら眺めることができ、あたかも自らが制御盤を操作しているかのような感覚を得ながら異常診断の過程を理解できると期待される。

以下、次章からは、第3章において、機器保修の訓練システムを効率的に開発できる新たな手法を適用した機器保修の訓練システムの開発について述べ、第4章において、プログラミングを全く行うことなく短時間で機器保修の訓練環境を構築できる支援システムの開発について述べる。次に、第5章において仮想運転員がプラントの異常診断と対応操作を行う様子を観察できる体験型教育システムの開発について述べ、最後に第6章の結論において本研究を総括する。

参考文献

- [1] 舘田章. 人工現実感. 日刊工業新聞社, 1992.
- [2] A Kaiser Aerospace & Electronics Company. *3SPACE USER'S MANUAL*. 1992.
- [3] 後藤善裕, 白井暁彦, 久米祐一郎, 津田元久, 畑田豊彦. 2台の傾斜対向配置PSDカメラによる手の形状認識. 日本バーチャルリアリティ学会第2回大会論文集, pp. 216-217, 1997.
- [4] POLHEMUS Inc. *ULTRATRAK PRO OPERATING MANUAL*. 1996.
- [5] N. Thalmann and D. Thalmann, editors. *Interactive Computer Animation*, pp. 10-39. Prentice Hall, Hertfordshire, 1995.
- [6] T. Zimmerman and J. Lanier. A Hand Gesture Interface Device. *Proceedings of CHI+GI 1987*, pp. 189-192, 1987.
- [7] VPL Research Inc. *DATAGLOVE MODEL 2 Operation Manual*. 1989.
- [8] VPL Research Inc. *DATAGLOVE TEST and CALIBRATION SOFTWARE for the Apple Macintosh Operation Manual*. 1989.
- [9] 小林広美. ヒューマンインターフェイスは究極か? データグローブの可能性. エレクトロニクス, Vol. 40, No. 3, p. 22, 1995.
- [10] SPACE CONTROL Inc. *MAGELLAN SPACE MOUSE User's Manual*. 1997.
- [11] 松永是. 五感のおはなし. 日本規格協会, 1989.
- [12] I. Sutherland. A Head-mounted Three Dimensional Display. *Fall Joint Computer Conference*, pp. 757-764, 1968.
- [13] 福島省吾, 新井豪, 森川大輔, 下田宏, 吉川榮和. Eye-Sensing Head Mounted Displayの開発. 計測自動制御学会論文集, Vol. 35, No. 6, pp. 699-707, 1999.
- [14] C. Cruz-Neira, D. Sandin, and T. Defanti. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. *COMPUTER GRAPHICS Proceedings*, pp. 135-142, 1993.
- [15] 南雲俊喜, 中山功, 甘利治雄, 岡田幹夫. 小型円筒面スクリーンによる運転・保守作業訓練環境の構築. 日本バーチャルリアリティ学会大会論文集, Vol. 3, pp. 207-208, 1998.

参考文献

- [16] StereoGraphics Corporation. *CrystalEyes Stereoscopic System User's Manual*. 1992.
- [17] 乾敏郎 (編). 視覚情報処理の基礎, pp. 2-19. サイエンス社, 1990.
- [18] E. Wenzel, F. Wightman, and S. Foster. Development of a Three-Dimensional Auditory Display System. *SIGCHI Bulletin*, Vol. 20, pp. 52-57, 1988.
- [19] 星猛, 伊藤正男 (編). 新生理科学体系 (9) 感覚の生理学. 医学書院, 1988.
- [20] 井野秀一, 泉隆, 伊福部達. 感覚フィードバック型ハンドのための圧覚呈示に関する一方式. 第8回日本ロボット学会学術講演会予稿集, pp. 261-262, 1990.
- [21] 池井寧, 若松和史, 池野晃久, 福田収一. 振動型触覚ディスプレイの表示手法の研究. *ヒューマンインタフェース News and Report*, Vol. 10, pp. 283-286, 1995.
- [22] 福田敏夫, 盛田英之, 新井史人, 石原秀則. マイクロ振動子を用いた触覚提示. 日本バーチャルリアリティ学会第2回論文集, pp. 198-210, 1997.
- [23] 井野秀一, 泉隆, 伊福部達. 物体接触時の皮膚温度変化に着目した材質触覚ディスプレイ方式の提案. 計測自動制御学会論文誌, Vol. 30, No. 3, pp. 345-351, 1994.
- [24] 野村淳二, 澤田一哉 (編). バーチャルリアリティ, 第2章. 朝倉書店, 1997.
- [25] T. Massie. Virtual Touch Through Point Interaction. *Proceedings of International Conference on Artificial Reality and Telexistence '96*, pp. 19-38, 1996.
- [26] 浅野武夫, 矢野博明, 岩田洋夫. フォースディスプレイを用いた仮想環境における手術シミュレーションの要素技術開発. 日本バーチャルリアリティ学会論文集, Vol. 1, pp. 95-98, 1996.
- [27] 平田幸広, 猿渡基裕, 佐藤誠. モータ駆動による力覚制御能力を持つ空間インタフェース装置の提案. *ヒューマンインタフェース News and Report*, Vol. 7, pp. 251-257, 1992.
- [28] 石井雅博, 猿渡基裕, 佐藤誠. 両手協調作業のための仮想環境の構築について. *電気学会論文誌 C*, Vol. 115-C, No. 2, pp. 203-208, 1995.
- [29] Y. Cai, M. Ishii, R. Takamatsu, and M. Sato. 3-D Force Display Device for Big Virtual Space, 1995.
- [30] 新藤義昭, 山地秀美 (編). バーチャルリアリティプログラミング. NECクリエイティブ, 1997.
- [31] 廣瀬通孝, 渡辺真二郎, 遠藤隆明. 実写画像を用いた広域仮想空間構築のための車載型撮影装置開発と画像補間手法に関する考察. *ヒューマンインタフェース News and Report*, Vol. 12, No. 2, pp. 157-162, 1997.

参考文献

- [32] 荒田秀樹, 高井昌彰, 高井那美, 山本強. 能動的ボクセル空間における仮想粘土モデリング-モデルの基本理念-. 電子情報通信学会論文誌, Vol. J82-D-II, No. 11, pp. 2008-2016, 1999.
- [33] 橋本渉, 岩田洋夫. ボリュームデータに基づく術中危険領域の力覚表現. 日本バーチャルリアリティ学会論文誌, Vol. 3, No. 4, pp. 197-202, 1998.
- [34] 矢野博明. 仮想環境構築ソフトウェア. ヒューマンインタフェース学会誌, Vol. 1, No. 4, pp. 1-8, 1999.
- [35] 神部勝之. Real Master. 第5回産業用バーチャルリアリティ展・セミナー要録, pp. 87-93, 1997.
- [36] 村上竜一. dVISE. 第5回産業用バーチャルリアリティ展・セミナー要録, pp. 69-80, 1997.
- [37] M.O'Rourke(袋谷賢吉, 大久保篤志訳). 3次元コンピュータ・アニメーションの原理. 株式会社トッパン, 1997.
- [38] 吉田勝俊. 泊原子力事務所原子力訓練センターの概要. 火力原子力発電, Vol. 47, No. 5, pp. 487-494, 1996.
- [39] 早苗勝重, 三森光二郎. 原子力発電所の運転と訓練. 電気学会誌, Vol. 117, No. 8, pp. 528-531, 1997.
- [40] 鈴木康弘, 近藤伸一, 渡部和之, 小林晃. ABWR シミュレータを用いた運転訓練の実績. 火力原子力発電, Vol. 50, No. 5, pp. 28-32, 1999.
- [41] R. Seidel and P. Chatelier, editors. *Virtual Reality, Training's Future?*, chapter 10. Plenum Press, New York, 1997.
- [42] R. Loftin and P. Kenney. Training the Hubble Space Telescope Flight Team. *IEEE Computer Graphics and Applications*, Vol. 15, No. 5, pp. 31-37, 1995.
- [43] 三輪祥太郎, 植田孝夫, 西田正吾. VR を用いた保守教育支援システム. 電気学会論文誌 C, Vol. 115-C, No. 2, pp. 203-211, 1995.
- [44] 菅野太郎, 古田一雄, 吉川信治. プラント運転操作に用いられる知識の可視化による深い理解支援システム. ヒューマンインタフェースシンポジウム'99 論文集, pp. 691-696, 1999.
- [45] Y. Matsubara, S. Toihara, Y. Tsukinari, and M. Nagamachi. Virtual Learning Environment for Discovery Learning and Its Application on Operator Training. *IEICE TRANSACTIONS on Information and Systems*, Vol. E80-D, No. 2, pp. 176-188, 1997.

参考文献

- [46] 新井浩一, 阿部慶子, 上地登. VR 技術を用いた変電所保守員向け集合教育用体感型シミュレータの開発. 日本バーチャルリアリティ学会論文集, Vol. 2, No. 4, pp. 7-16, 1997.
- [47] C. Petri. *Kommunikation mit Automaten*. Schriften des Institutes für Instrumentelle Mathematik, Bonn, 1962.
- [48] 椎塚久雄. 実例ペトリネット, 第4章. コロナ社, 1992.
- [49] 村田忠夫. ペトリネットの解析と応用, 第9章. 近代科学社, 1992.
- [50] K. Jensen. *Coloured Petri Nets*. Springer Verlag, 1992.
- [51] 中島豊四郎, 伊藤宏隆, 南里孝幸. ペトリネットによる伸線プラントシステムの解析. 電気学会論文誌 C, Vol. 114-C, No. 9, pp. 881-887, 1994.
- [52] 玉置久, 椋本進, 西川喜清, 荒木光彦. 一般化ペトリネット・モデルを用いたシミュレーション・ベースド・スケジューリング. 電気学会論文誌 C, Vol. 114-C, No. 9, pp. 888-897, 1994.
- [53] 高橋宏治. ペトリネットの最近の応用 III フレキシブルオートメーションとペトリネット. 電気学会論文誌 C, Vol. 114, No. 9, pp. 869-875, 1994.
- [54] K. Takano, W. Sunaoshi, and K. Suzuki. Intellectual Simulation of Operating Team Behavior in Coping with Anomalies Occurring at Commercial Nuclear Power Plants. *Human-Computer Interaction*, Vol. 2, pp. 1201-1205, 1999.
- [55] 高野研一, 砂押亘, 鈴木国彦. 実機レベルの原子力プラントにおける運転チーム行動モデルの開発. ヒューマンインタフェースシンポジウム'99 論文集, pp. 859-864, 1999.
- [56] M. Doi, N. Kato, N. Umeki, T. Harashima, and K. Matsuda. Visual Engineering System - VIGOR : Virtual environment for visual engineering and operation. *Symbiosis of Human and Artifact*, pp. 435-440, 1995.

第 3 章 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

3.1 はじめに

本章では、人工現実感技術を用いた機器保修の訓練システムの開発について述べる。2.3節で述べたように、これまでに開発された人工現実感技術を用いた機器保修の訓練システムは、訓練生が実施できる作業手順が非常に狭い範囲に限定されているものが大半であった。

そこで、本研究では特に、訓練生が自らの判断で自由な手順で機器保修の作業を進めることができる機器保修の訓練システム (VR-based Machine-Maintenance Training Environment; VRMATE) を開発する。その際、数多くの仮想物体が互いに関係し合う複雑な仮想空間を、容易に構築できる新たな方法として、仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する手法を提案する。さらに、複雑大規模なペトリネットを少ない労力で構築できるようにするためのペトリネット作成支援システム (Petri net Construction Support System; PCSS) を開発する。また、訓練生が自由に作業できる訓練環境を構築した際に、訓練生が効率的に保修の作業手順を学習できるようにするための自動実演機能も構築する。

本章では、以上の研究を通じて、訓練生が自由に作業できる機器保修の訓練環境を少ない労力で容易に構築できるようにするための新たな手法を提案し、人工現実感技術を用いた機器保修の訓練環境の有用性を確認するとともに、提案した訓練環境の構築手法の有効性を評価し、人工現実感技術を用いた機器保修の訓練システムを実用化するために今後解決すべき課題と問題点を見出すことを目的とする。

以下では、まずはじめに、VRMATE を開発するにあたり、訓練生が効率的に訓練を進めることができるようにするために、訓練システムが具備すべき機能について検討する。次に、VRMATE で訓練の対象とする機器保修作業について述べた後、訓練生が実世界と同様に、自らの意志で自由に作業を進めることができるようにするための、仮想空間のシミュレーション手法について、訓練生のジェスチャの判定方法、仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する方法、作成したペトリネットを実際の訓練に使用する方法を順次述べる。そして、機器保修の訓練環境を構築する際の作業効率の改善を目指して開発した、ペトリネット作成支援システム PCSS について、その設計の基本方針を述べた後、システムの概要と、実際の使用例を述べる。その後、訓練生が効率的に機器保修の作業手順を修得できるようにするための自動実演機能について、その実現のためのアルゴリズムを述べる。そして、VRMATE の概要を述べた後、システムの評価を行った結果と、今後の課題について述べる。

3.2 機器保修の訓練システムに必要とされる機能

本節では、訓練生が保修作業に必要な知識を効率的に修得できるようにするために、訓練システムが具備すべき機能や特徴について述べる。

本研究で開発する VRMATE では、機器の保修作業に必要となる知識・技能の内、主に知識を修得することを目的とする。従って、VRMATE では主に訓練生の視覚と聴覚に対して情報提示を行い、現時点での人工現実感技術では情報提示を行うことが困難である触覚、嗅覚等の他の感覚に対しては、情報提示を行わないことにした。また、本研究では、主に以下の5つの項目に重点を置いて VRMATE の開発を進めた。

1. 訓練システムを、予備知識や技能がなくても使用できる。

訓練生が初めて訓練システムを使用する際、その訓練システムを使用するための知識や技能が新たに必要になれば、それらを修得するための時間や労力が必要になり、機器保修作業の訓練に注力できなくなる危険性がある。従って、訓練システムを開発する場合、その訓練システムを使用するために必要な予備知識や技能を最小限にすることは非常に重要である。新たな予備知識や技能を必要としない訓練システムの1つの条件として、「訓練システムに訓練生の意志を伝達するために特別なジェスチャを使わないインタフェースを備えていること」を挙げることができる。これに対し、機器保修の一連の作業を、現実世界の行動に近い方法で実行できるようにすれば、仮想空間内で作業をする際に新たな操作方法を修得する必要がなくなると考えられる。そこで、VRMATE を開発する際、計測サブシステムとして実際の作業に近いジェスチャを入力できる、データグローブとポヒマスセンサを採用し、訓練生の腕と手の運動を計測することにした。VRMATE のハードウェア構成の詳細は 3.7.1項で述べる。

2. 訓練生が自らの意志で自由に作業を進めることができる。

仮想空間内で実施可能な作業の手順が、作業マニュアルに記載されている作業手順のみである場合、訓練生が自ら判断して作業しなくなるため、受け身の訓練となり、訓練意欲が低下する可能性がある。単に作業マニュアルに記載されている作業手順だけでなく、手順に関係のない作業や、間違っただ手順でも作業ができ、また、間違っただ手順で作業した場合も、その結果を体験できる訓練環境を提供すれば、訓練効率が上がると期待される。従って、VRMATE では、思いつく限り自由な手順で作業できる環境を構築することにした。

3. 効率よく機器保修の作業手順を修得できる。

作業マニュアルに記載されている正しい作業手順以外の手順で、機器保修の作業を行うことを可能にすると、機器保修作業の初心者は、知識や経験が不足しているために、作業手順が分からなかったり、間違えたりすることが多々あると予想される。この場合に、訓練を中断したり、間違えたままで作業を進めたりすると、訓練の効率が上がらなかったり、間違えた作業手順を覚えたりする危険性がある。そこで、VRMATE を開発する際には、訓練生が誤りを犯した時や、どうすればよいか分からなくなった

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

ときに、可能ならば間違った時点まで戻り、正しい手順を提示する自動実演機能を構築する。自動実演機能の概要とその実現方法については3.6節で述べる。

4. 仮想空間内で視点を自由に変えることができる。

訓練生が訓練対象機器の構造を容易に把握できるようにするために、仮想空間内で自由に視点を変更できることは重要である。VRMATE では、訓練対象機器の3次元的な構造を自由に眺める機能として、仮想空間内に配置された視点変更オブジェクトをデータグローブを介して操作することにより、自由に視点を変更できる機能を構築する。

5. 訓練対象機器の各部品の名称や、各部品の機能等を学習できる。

VRMATE では、機器の保修作業の手順や機器の構造以外の保修作業に必要な知識も効率的に学習できるようにするために、機器保修の作業を仮想体験できる環境だけでなく、機器の各部の名称を示した図や機器の写真、各部の機能を説明する動画等を参照できる機能を構築する。

3.3 構築の対象とする機器保修の訓練環境

本研究では、VRMATE を開発するにあたり、具体的な訓練の対象を、原子力発電プラントの構成機器の1つである、スイング式逆止弁の分解作業とした。スイング式逆止弁は、流体を一方向のみに流すための弁として、プロセスプラントで広く使用されている機器である。スイング式逆止弁の構造を Fig. 3.1に、その分解の作業手順を Table 3.1に示す。Table 3.1に示した分解作業が終わると、各部品の清掃、検査の後、必要に応じて部品が交換される。組立手順は分解手順のほぼ逆の手順で行われる。本研究では、Table 3.1の手順1から手順6までの分解作業を対象として、訓練環境を構築する。

スイング式逆止弁の分解作業では、再び逆止弁を組み立てる際に、ふたの方向を分解前と同じ方向に戻すために、分解作業を始める前に、ペンを用いてふたと弁箱に「合いマーク」を付けなくてはならない。また、スイング式逆止弁のふたを外す際に、弁の内部に圧力が残っていれば、液体が噴出する危険性があるため、ふたを完全に取り外す前に、ふたボルトのナットを少し緩め、ふたを持ち上げて内圧を確認する必要がある。訓練生がこれらの作業を行わなかった場合、作業手順を間違えたとして修正する必要がある。

なお、本研究では、訓練生が自ら作業手順を判断して作業を行うことができる訓練環境を構築することを1つの目標としているが、今回構築する訓練環境では特に、以下の作業を可能にすることにする。

1. 印 (合いマーク) を弁箱とふたの任意の場所に付けることができる。
2. 仮想物体を床の上の任意の場所に置くことができる。
3. 複数あるナットの内、任意のナットをボルトにはめることができる。

3.3 構築の対象とする機器保守の訓練環境

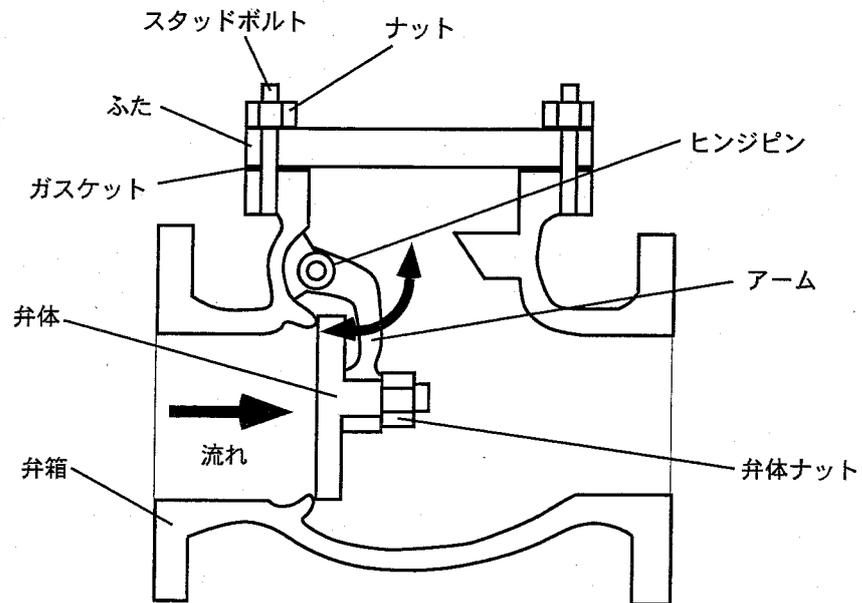


Fig. 3.1: スイング式逆止弁の構造

Table 3.1: スイング式逆止弁の分解手順

| 手順 | 作業内容 |
|----|-----------------------------------|
| 1 | ふたと弁箱にペンで印をつける。 |
| 2 | ふたボルトのナットをレンチで緩める。 |
| 3 | 手を使ってふたを少し持ち上げ、内圧がかかっていないことを確認する。 |
| 4 | 指でふたボルトのナットをすべて取り外す。 |
| 5 | 手を使ってふたを取り外す。 |
| 6 | 弁箱にはまっているスタッドボルトをダブルナットの方法を用いて外す。 |
| 7 | ヘラを用いてガスケットを取り外す。 |
| 8 | アームを持ち上げ、ヒンジピンを抜き取り、アームを取り外す。 |
| 9 | アームのナットを取り外す。 |
| 10 | アームとバルブを分離する。 |

4. 複数あるナットの内、任意のナットを外すことができる。
5. ボルトにはまったナットは、レンチを用いて外すことも、素手で外すこともできる。

3.4 仮想空間のシミュレーション手法

仮想空間内で機器保修の作業を行えるようにするためには、訓練生のジェスチャに応じて、仮想物体が実空間と同様の物理法則に従って運動する必要がある。例えば、スイング式逆止弁の分解作業では、訓練生が手でペンを握るジェスチャをした場合、仮想空間内の訓練生の手に相当する仮想物体(以下、手のモデル)が、床の上に置かれたペンに相当する仮想物体(以下、ペンのモデル)を掴み、その後訓練生が手を動かした時には、ペンのモデルは手のモデルの動きに追従する必要がある。またさらに、訓練生が手を開くジェスチャをした場合には、ペンのモデルは手のモデルを離れ、床に向かって自由落下する必要がある。

このような実空間と同様の物理法則に従った仮想空間を構築する場合、2.1.3項で述べたように、仮想物体の詳細な表面形状や質量、弾性係数等の情報を保持し、物体間の詳細な接触判定や運動方程式等の利用を通して仮想物体の動きを制御する方法も考えられるが、計算量が膨大になり、現時点での計算機の能力では、リアルタイムに処理するのは非常に困難である。従って、機器保修の訓練環境を構築する場合、訓練生のジェスチャに応じた仮想物体の運動の仕方を予め定めておく必要がある。

そこで本研究では、訓練生のジェスチャに応じた仮想物体の運動の仕方を定める方法として、仮想物体の「状態」の概念を利用した方法を採用する。これは、時間の推移に対して連続的に変化する機器保修の作業空間を、仮想物体が取りうる「状態」ごとに離散的に区切って捉えることによりモデル化することに相当する。例えば、「訓練生が床の上のペンを右手で握ることができる仮想空間」を構築する場合、まず、ペンは仮想空間内で「床の上にある状態」と「右手に握られた状態」の2つの状態を取りうると思う。そして、ペンが「床の上にある状態」の時に、訓練生が「ペンを握るジェスチャ」をした場合、ペンの状態が「右手に握られた状態」に遷移すると思う。そして、ペンが取りうる各状態に対して、「床の上にある状態」の時には、ペンは床の上で静止し、ペンが「右手に握られた状態」の時には、ペンが右手の動きに追従すると思う。このように考えれば、訓練生のジェスチャに応じた仮想物体の運動の仕方を定めるためには次のようにすればよいことが分かる。

まず、仮想物体が取りうる状態を列挙し、訓練生のジェスチャに応じた仮想物体の状態遷移の仕方を定める。次に、仮想物体が取りうる各状態ごとに、訓練生のジェスチャに応じた運動の仕方を定める。そして、訓練を実施する際には、訓練生のジェスチャに応じて、仮想物体の状態を遷移させ、仮想物体の状態と訓練生のジェスチャに応じて、仮想物体の運動を制御する。

上記のような方法で訓練生のジェスチャに応じた仮想物体の運動の仕方を定め、仮想空間のシミュレーションを行う場合、以下の事柄を定める必要がある。

1. 訓練生のジェスチャを判定する方法

Table 3.2: VRMATE が認識する訓練生のジェスチャ

- | | |
|-----|---------------------|
| (a) | 手を移動させる。 |
| (b) | 物体に触る。 |
| (c) | 手を握る。 |
| (d) | 手を開く。 |
| (e) | 物体を把持する。 |
| (f) | 把持した物体を放す。 |
| (g) | 物体を把持したまま移動させる。 |
| (h) | 二つの物体を接合する。 |
| (i) | 二つの物体を分離する。 |
| (j) | ある物体をある軸のまわりに回転させる。 |

2. 「仮想物体が取りうる状態」と「仮想物体の状態遷移の仕方」をデータとして記述する方法

3. 実際に訓練を実施する際に、仮想物体の運動を制御する方法

本研究ではまず、1. の訓練生のジェスチャに関して、3.2節で述べたように、訓練生が実際の作業に近い動作で必修作業を進めることができるようにするために、入力デバイスとして、データグローブとポヒマスセンサを用いる。これらのデバイスからは、訓練生の手の形状や手の位置・角度に関する情報が得られるので、それらの情報をもとに、10種類のジェスチャを認識するものとする。ジェスチャの判定方法の詳細については3.4.1項で述べる。また、2. の「仮想物体が取りうる状態」と「仮想物体の状態遷移の仕方」の2つの情報に関しては、機器必修の作業時の仮想物体の状態遷移を離散事象過程と捉え、ペトリネットを用いてモデル化する方法を提案する。ペトリネットによるモデル化方法の詳細は、3.4.2項及び3.4.3項で述べる。さらに、3. の訓練生のジェスチャに応じて仮想物体の運動を制御する方法については、3.4.4項で述べる。

3.4.1 機器必修作業における訓練生のジェスチャ判定

本研究では、VRMATE で訓練生が使用できるジェスチャとして、Table 3.2に示す10種類のジェスチャを認識可能とする。以下、このTable 3.2に示す個々のジェスチャにより仮想空間内で実行される作業を単位タスクと呼ぶ。単位タスクは、訓練生が実行可能な最も細分化された作業となる。Table 3.2に示す10種類のジェスチャを組み合わせることにより、仮想空間内における複雑な機器必修の作業も実行できる。例えば、「ナットをスパナで締める」作業はTable 3.3に示す一連の単位タスクを連続して実行することにより実現される。

VRMATE では、訓練生がジェスチャを入力するデバイスとして、データグローブとポヒマスセンサを右手用、左手用の各2つずつ用いる。このデータグローブは前述のように、

Table 3.3: 「ナットをスパナで締める」作業の単位タスク表現

| 手順 | 動作 | 単位タスク (ジェスチャ) |
|----|------------------|---------------|
| 1 | 手を移動させる。 | (a) |
| 2 | スパナを触る。 | (b) |
| 3 | スパナを把持する。 | (e) |
| 4 | スパナをナットの近くに移動する。 | (g) |
| 5 | スパナをナットにはめる。 | (h) |
| 6 | ナットをスパナで回転させる。 | (j) |
| 7 | スパナをナットから離す。 | (i) |
| 8 | スパナを手放す。 | (f) |

Table 3.4: ジェスチャが行われたと判定する条件

| ジェスチャ | 判定のための条件 |
|----------------|----------------------------------|
| (a) 手を移動させる | ポヒマスセンサからの入力値が変化した場合 |
| (b) 物体に触る | 手を移動させ、手と仮想物体が接触した場合 |
| (c) 手を握る | 5指関節の曲げ角度の合計が、 Th_1 より大きくなった場合 |
| (d) 手を開く | 5指関節の曲げ角度の合計が、 Th_2 より小さくなった場合 |
| (e) 物体を把持する | 仮想物体を触った状態で手を握った場合 |
| (f) 把持した物体を放す | 仮想物体を把持した状態で手を開いた場合 |
| (g) 物体を移動させる | 仮想物体を把持した状態で手の位置を変化させた場合 |
| (h) 2つの物体を接合する | 2つの仮想物体が接触した場合 |
| (i) 2つの物体を分離する | 2つの仮想物体の接触が無くなった場合 |
| (j) 物体を回転させる | 仮想物体を把持した状態で手を回転させた場合 |

手の5指に対して各指の第1関節と第2関節の曲げが計測される。また、ポヒマスセンサも前述のように、3次元位置と角度の合計6種類の数値が計測される。従って、右手、左手それぞれについて、各16種類の数値が入力される。VRMATEではこの16種類の数値から、Table 3.2に示した訓練生のジェスチャを判定するため、Table 3.4に示す規則を用いる。但し、表中の Th_1 と Th_2 の関係は $Th_1 > Th_2$ である。これは、 $Th_1 = Th_2 = Th$ とすると、5指関節の曲げ角度の合計が Th に近い場合に、手を握るジェスチャと手を開くジェスチャが交互に繰り返し認識されてしまう場合があるという問題を回避するためである。

3.4.2 ペトリネットの定義

本項では、「仮想物体が取りうる状態」と「仮想物体の状態遷移の仕方」の2つの情報をモデル化するために使用するペトリネットの定義について述べる。

ペトリネットは有向グラフの1つであり、初期マーキングと呼ばれる初期状態を持つ。ペトリネットを作る基本的なグラフは、有向2部グラフであって、プレースおよびトランジションと呼ばれる2種類のノードからなる。ここで、アークは、プレースからトランジションに接続するもの、あるいは、トランジションからプレースへ接続するもののいずれかである。プレースに非負整数値が割り当てられているとき、プレースはトークンでマーキングされているという。プレース p からトランジション t に向かってアークが接続されている場合、このプレース p をトランジション t の入力プレースと呼ぶ。同様にトランジション t からプレース p に向かってアークが接続されている場合、このプレース p をトランジション t の出力プレースと呼ぶ。あるトランジション t に対して、入力プレースと出力プレースにマーキングされているトークンの数が、共に予め定められた条件を満たす場合、トランジション t は発火可能であるという。トランジションの発火により、予め定義された遷移規則に従い、入力プレース内のトークンが削除され、出力プレースに新たなトークンが加えられる。以下に、VRMATE を開発する際に使用するペトリネットの定義を述べる。

定義 1 P/T-net(Place/Transition net)

P/T-net は6項組 $PN = (P, T, F, W, K, M)$ として定義される。ただし、

- $P = \{p_1, p_2, \dots, p_m\}$ 、 $m = |P| > 0$ はプレースの有限集合。
- $T = \{t_1, t_2, \dots, t_n\}$ 、 $n = |T| > 0$ かつ $P \cap T = \emptyset$ はトランジションの有限集合。
- $F \subseteq (P \times T) \cup (T \times P)$ はアークの集合。
- $W : F \rightarrow \{0, 1\}$ はアークの重み関数。 $W(p, t)$ は p から t へのアークの重み、 $W(t, p)$ は t から p へのアークの重み。
- $K : P \rightarrow \{1, \infty\}$ はプレースの容量関数。 $K(p)$ はプレース p の容量。
- $M : P \rightarrow \mathbb{N}$ はマーキング関数、 $M(p)$ はプレース p に割り当てられるトークンの個数。

ただし \emptyset は空集合、 \mathbb{N} は自然数の集合、 $|X|$ は集合 X の要素の数とする。

定義 2 シングルプレース

$\exists p_i \in P$ が $K(p_i) = 1$ を満たすとき、 p_i を特にシングルプレースと呼び、グラフでは Fig. 3.2 に示すように一重丸で表す。

定義 3 プールプレース

$\exists p_i \in P$ が $K(p_i) = \infty$ を満たすとき、 p_i を特にプールプレースと呼び、グラフでは Fig. 3.2 に示すように一重楕円で表す。

| 名称 | グラフ表現 | 名称 | グラフ表現 |
|-------------|---|-------------|---|
| シングルプレース |  | プールプレース |  |
| 制御プレース |  | ノーマルトランジション |  |
| 自動発火トランジション |  | トークン |  |

Fig. 3.2: ペトリネットのグラフ表現

定義 4 入力プレース、出力プレース

$\exists p_i \in P$ と $\exists t_j \in T$ に対して $W(p_i, t_j) = 1$ を満たすとき、 p_i を t_j の入力プレースと呼び、グラフでは、プレース p_i からトランジション t_j へ矢印を引くことにより表現する。同様に、 $W(t_j, p_i) = 1$ を満たすとき、 p_i を t_j の出力プレースと呼び、グラフでは、トランジション t_j からプレース p_i へ矢印を引くことにより表現する。

定義 5 トークン

$\exists p_i \in P$ に対し、 $M(p_i) = k$ であることを、「プレース p_i にトークンが k 個存在する」と言い、グラフでは Fig. 3.2 に示すような黒丸を、プレース p_i の内部に k 個描くことにより表現する。

定義 6 発火可能

トランジション t は、以下の条件を全て満たす時に発火可能であるという。

条件 1 $\forall p \in \cdot t$ となる p に対して、 $M(p) > 0$ である

条件 2 $\forall p \in t \cdot$ となる p に対して、 $M(p) \neq 0$ ならば $W(p, t) = W(t, p) = 1$ 、もしくは $K(p) = \infty$ である

ただし、トランジション t の入力プレースの集合を $\cdot t$ 、出力プレースの集合を $t \cdot$ で表す。

定義 7 トランジションの発火

トランジション t の発火により以下の遷移規則に従ってトークンが遷移する。

遷移規則 1 $\forall p \in \cdot t$ に対して $M(p) = M(p) - 1$

遷移規則 2 $\forall p \in t \cdot$ に対して $M(p) = M(p) + 1$

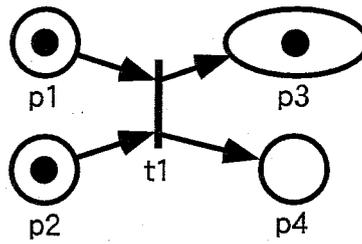


Fig. 3.3: 発火可能なトランジションの例

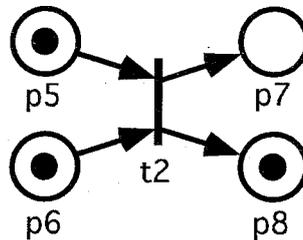


Fig. 3.4: 発火不可能なトランジションの例

また、仮想物体の状態遷移を効率的にモデル化するために、定義2で定義したシングルプレースの内、3.4.3項で述べる規則6の目的で作られたプレースを特に制御プレースと呼び、グラフではFig. 3.2に示すように二重丸で表す。さらに、ペトリネットを用いて仮想空間のシミュレーションを実行する際に、定義6で述べた発火条件が成立しても、予め設定したイベント（仮想物体同士が接触する、訓練生が手を握る等）が起これなければ発火しないトランジションをノーマルトランジションと呼び、グラフではFig. 3.2に示すように一本の棒で表現する。一方、定義6で述べた発火条件が成立すると、イベントの発生に関係なく、必ず発火するトランジションを自動発火トランジションと呼び、グラフではFig. 3.2に示すように二本の棒で表現する。

Fig. 3.3及びFig. 3.4に、上記定義に基づいたペトリネットの例を示す。Fig. 3.3のトランジションt1は、入力プレース p1,p2 共にトークンが存在し(条件1が成立)、出力プレース側のシングルプレース p4 にトークンが存在しないため(条件2が成立)、発火可能である。Fig. 3.3のペトリネットは、トランジションt1の発火によりFig. 3.5に示すトークン配置に遷移する。一方Fig. 3.4のトランジションt2は、入力プレース p5,p6 共にトークンが存在しているが(条件1が成立)、出力プレース側のシングルプレース p8 にトークンが存在するため(条件2が不成立)、発火不可能である。

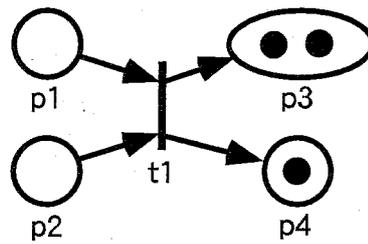


Fig. 3.5: トランジション発火後のトークン配置

3.4.3 ペトリネットによるモデル化手法

本項では、前項で述べたペトリネットを用いて仮想物体の状態遷移をモデル化する方法について説明する。

仮想物体の状態遷移のモデル化は、基本的には、仮想物体が取りうる状態をスペースに、仮想物体の状態遷移をトランジションの発火によるトークンの遷移に対応させることにより行う。以下に、本研究で考案したペトリネットを用いた仮想物体の状態遷移のモデル化の規則を列挙する。

- 規則 1 仮想物体が取りうる 1 つの状態を 1 つのシングルスペース、もしくはプールスペースに対応させる。複数の仮想物体が同時に同じ状態になることが可能な場合はその状態をプールスペースに対応させる。不可能な場合はその状態をシングルスペースに対応させる。
- 規則 2 仮想物体がある状態にあることを、対応するスペースにトークンが存在することで表す。複数の仮想物体が同時に同じ状態になる場合は対応するプールスペースに相当数のトークンを配置する。
- 規則 3 仮想物体の状態遷移を引き起こす訓練生の動作の実行を、トランジションの発火に対応させ、仮想物体の状態遷移をトランジションの発火によるトークンの遷移に対応させる。すなわち、トランジション 1 つが単位タスクに対応する。
- 規則 4 1 つの仮想物体に 1 つのトークンを割り当てる。その対応関係はシミュレーションを通して変化しない。これにより、「トランジションの発火により仮想物体に対応するトークンがアークの矢印の向きに沿って移動する」と考えることができる。
- 規則 5 トークンの総数はトランジションの発火の前後で保存される。これにより、シミュレーションを通して仮想物体が生成したり消滅したりすることはない。
- 規則 6 1 つの仮想物体に 1 つのトークンを割り当てた場合に、ペトリネットによるモデル化が冗長になる場合や、仮想物体の状態以外の特別な条件成立の有無を表現する場合には、制御スペースを用いる。制御スペース内のトークンは、その総数がトランジションの発火の前後で保存されるとは限らない。

3.4 仮想空間のシミュレーション手法

Table 3.5: ナットが取りうる状態とボルトが取りうる状態

| ナットが取りうる状態 | ボルトが取りうる状態 |
|---|---|
| 床の上に置かれている 手に触られている 手に把持されている 手に把持されたままボルトにはまっている ボルトにはめられている 自由落下している | ナットをはめられていない 手に把持されたナットがはまっている ナットがはまっている |

規則 7 手の状態を表すプレースとしては、「手が何もしていない状態」に対応する制御プレースのみを用いる。(VRMATEでは、手の取りうる状態として、「何も持っていない状態」と「何かを持っている状態」の2通りの状態を表現できれば十分であるため)

規則 8 仮想空間のシミュレーションを実行する際に、訓練生のジェスチャやその他のイベントの発生に関係なく、仮想物体の状態が遷移する場合は、その状態遷移を自動発火トランジションの発火によるトークン遷移としてモデル化する。

ペトリネットによる仮想物体の状態遷移のモデル化の例として、「床の上に置かれたナットを手でボルトにはめる」作業（以下、ナットをはめる作業）をモデル化する手順を以下に示す。なお、仮想空間の初期状態としては、床の上に2個のナットが配置され、手は何もしていないものとする。

手順 1 モデル化の対象とする作業に関する仮想物体を抽出する。ナットをはめる作業の場合は「手」、「ボルト」、「ナット」、「床」の4つの仮想物体が作業に関係する。

手順 2 モデル化の対象とする作業を実施した場合に状態が変化する仮想物体を抽出する。ナットをはめる作業の場合は、「手」、「ボルト」、「ナット」の3つの仮想物体の状態が変化する。

手順 3 手順 2 で抽出した仮想物体が取りうる状態を抽出する。ナットをはめる作業におけるナットが取りうる状態とボルトが取りうる状態を Table 3.5 に示す。「手」の状態としては、「何もしていない」状態と「何かをしている」状態の2つの状態を取るものとする。

手順 4 手順 3 で抽出した状態をプレースに対応させる。3.4.2項で述べた様に、同時に複数の仮想物体が同じ状態を取ることができる場合は、その状態をプールプレースに対応させる。ナットをはめる作業の場合は、床の上に複数のナットを置くことを可能にするために、ナットの「床の上に置かれている」状態に対応するプレースをプールプレースとしてモデル化する。

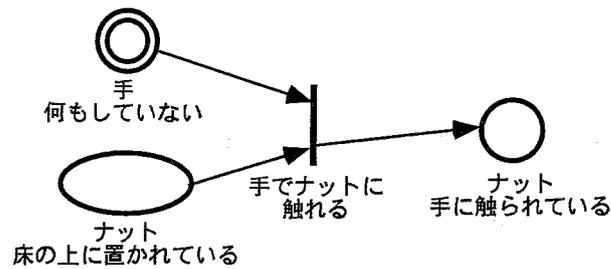


Fig. 3.6: モデル化したペトリネットの例

手順 5 状態遷移前の仮想物体の状態を入力プレースに、状態遷移後の仮想物体の状態を出力プレースとしてトランジションの発火による仮想物体の状態遷移をモデル化する。例えば、手に何も持っていない時に、床の上に置かれているナットを手で触れることにより、ナットの状態が「手に触られている」状態に遷移することをペトリネットモデル化した場合、Fig. 3.6に示すようになる。

手順 6 仮想物体の初期状態に応じて、ペトリネットのトークンの初期配置を決定する。ナットをはめる作業の場合は、手が何もしていないことを表すプレース「手：何もしていない」にトークンを1つ配置し、ナットが床の上に置かれていることを表すプレース「ナット：床の上に置かれている」にトークンを2つ配置する。また、ボルトにナットがはまっていないことを表すプレース「ボルト：ナットをはめられていない」にトークンを1つ配置する。

以上の手順によってモデル化されたペトリネットを Fig. 3.7に示す。

3.4.4 仮想空間のシミュレーション手法

本項では、作成したペトリネットを用いて実際に訓練生のジェスチャに応じて仮想物体の運動を制御する方法について述べる。本研究で開発した VRMATE で、訓練のシミュレーションを実行する際の処理の流れと情報の流れを Fig. 3.8および Fig. 3.9に示す。

基本的には、ペトリネット上の1つのトランジションは、仮想空間内で訓練生が実行できる単位タスクに対応している。例えば、仮想空間内で訓練生がペンに触れた場合、ペトリネット上でその単位タスクに対応するトランジションが発火して、そのトランジションの入出力プレース内のトークンが遷移し、それと同時に仮想物体の状態も変化する。

ここで、ペトリネット上で単位タスクに対応するトランジションの発火条件が成立しない場合は、訓練生がそのトランジションを発火させるようなジェスチャをしてもトランジションは発火せず、仮想物体の状態も変化する。このことを利用することにより、例えばナットで土台に締め付けられたふた等が、ナットを取り外さなければ動かすことができない等の制御が行える。

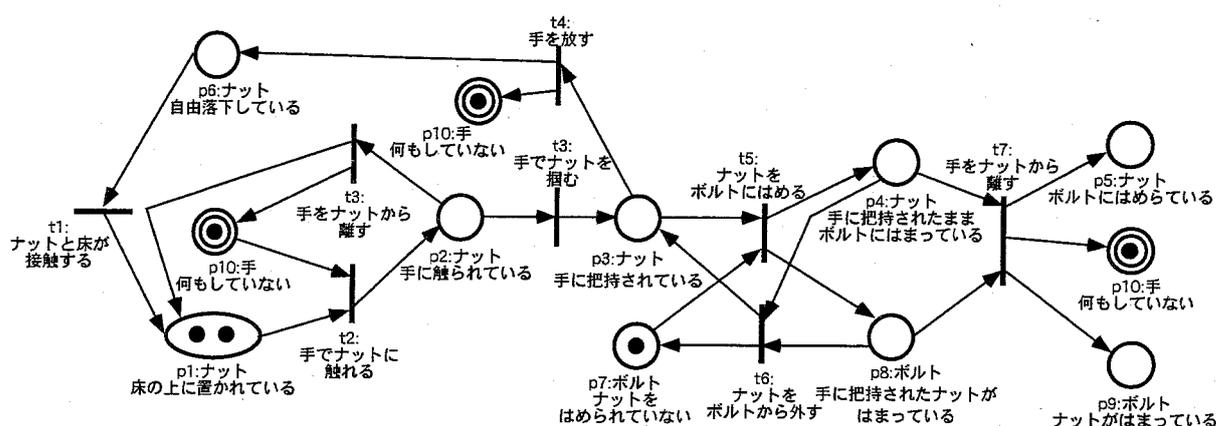


Fig. 3.7: 「床の上に置かれたナットを手でボルトにはめる」作業のペトリネット

このように、ペトリネットによって表現される離散的な状態に関する条件と、仮想空間内で行われる訓練生のジェスチャに関する条件が共に成立した時に初めて、トランジションが発火し、ペトリネットのトークン配置及び仮想物体の状態が変化する。

従って、実際に仮想空間のシミュレーションを行うためには、ペトリネットのシミュレーションに加えて、訓練生のジェスチャを判定し、さらに、判定結果に応じて、実際に仮想物体の運動の制御を行う必要がある。そこで、本研究では、ペトリネット上の各トランジションに対して、タスク関数と呼ぶC言語の関数を用意し、以下の方法で仮想空間のシミュレーションを行うことにした。

まず、全てのトランジションに対して、入出力プレースの条件を判断する。この条件を満たすものが発火可能なトランジションである。次に発火可能なトランジション全てについて、順番に対応するタスク関数を実行する。タスク関数では、以下の処理を行う。

1. 訓練生のジェスチャの認識

それぞれのタスク関数ごとに異なるジェスチャを認識して、予め定めてあるジェスチャと一致しない場合には、何もせずにタスク関数を終了する。ただし、自動発火トランジションの場合には、ジェスチャ認識等は行われずに、次の段階に進む。

2. トランジションの発火条件の判定と発火

トランジションの入出力プレースに入っているトークンに対応する仮想物体の位置関係から、トランジションを発火させるかどうかを判断する。もし、発火条件を満たしている場合には、トランジションを発火させ、トークンが遷移する。

3. 仮想物体の運動の制御

トランジションが発火した場合の仮想物体の運動を制御する。例えば、ある仮想物体を握ったら、その後は仮想物体が手の動きに追従するようにするなどの処理を行う。

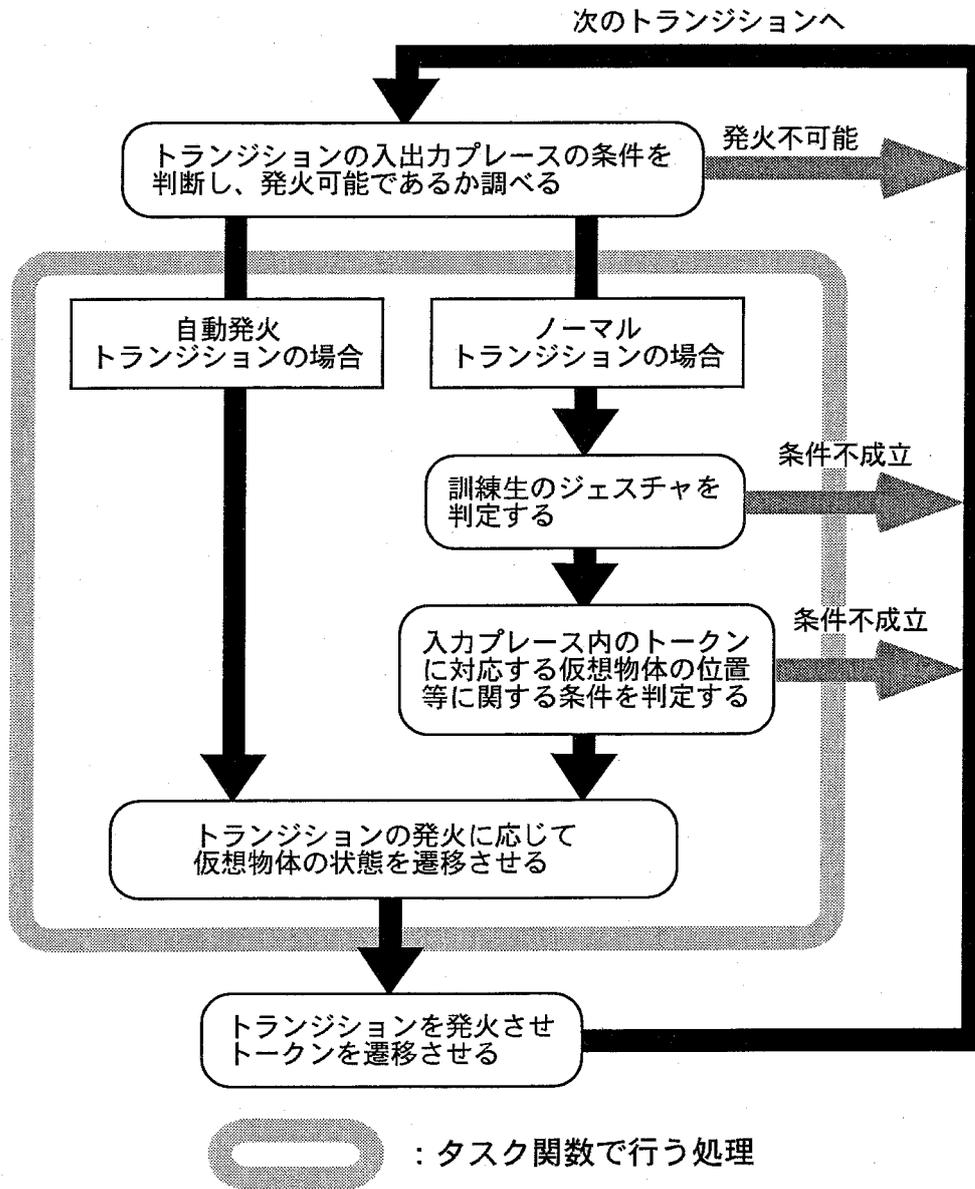


Fig. 3.8: 訓練シミュレーションにおける処理の流れ

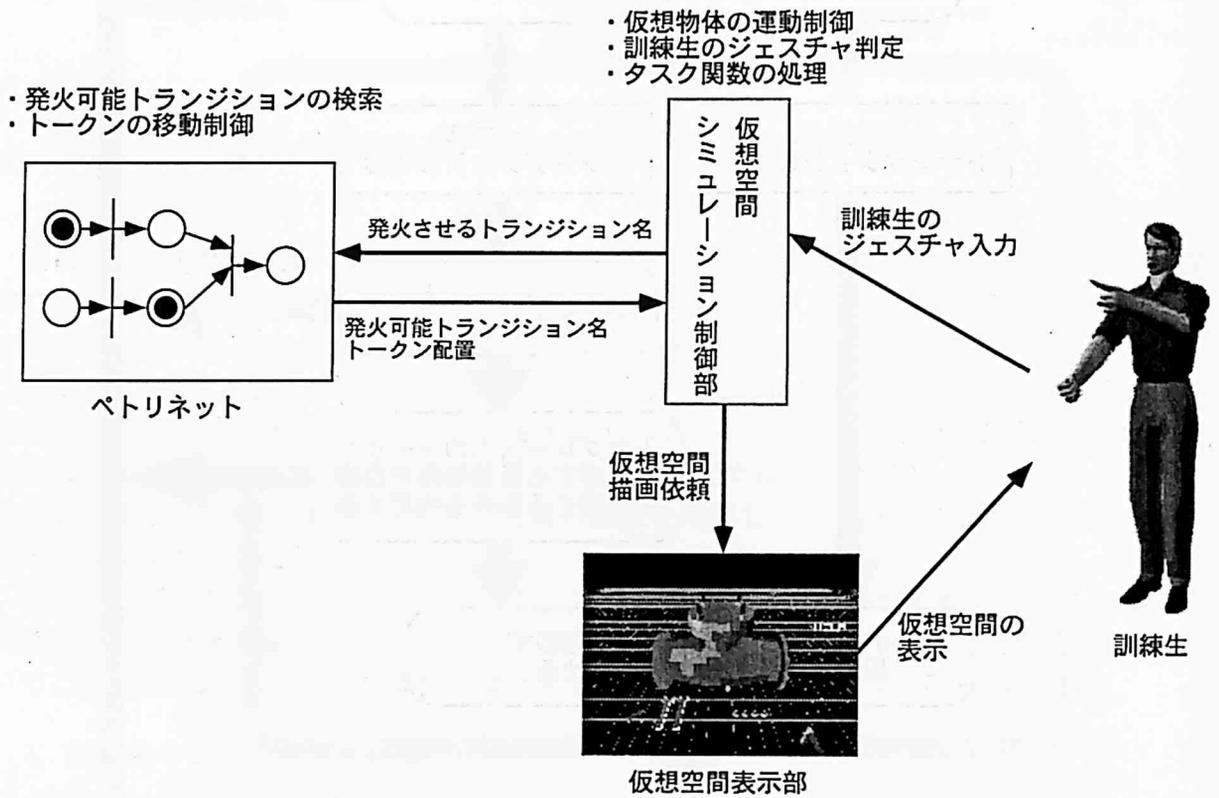


Fig. 3.9: 訓練シミュレーションにおける情報の流れ

以上のように個々の単位タスクに対して1つのタスク関数を割り当てることの利点としては、

1. トランジション単位で関数を作成するため、ペトリネットの変更・追加が容易である。
2. 異なるトランジションであっても、必要な単位タスクが同じであれば、同じ関数を利用できる。

ということが挙げられる。従って、作成する必要があるタスク関数の数は、その訓練対象で生じる単位タスクの数だけでよく、ペトリネットのトランジションの数に比べて格段に少ない数となる。

3.5 ペトリネット作成支援システム PCSS の開発

これまで、訓練生のジェスチャに応じた仮想物体の運動の仕方を容易に記述、管理する方法として、仮想物体の状態遷移の仕方をペトリネットを用いてモデル化し、それを用いて仮想物体の運動を制御する方法について述べた。しかし、ペトリネットが視覚的に表現可能なツールである利点を生かすためには、そのモデル化の方法を確立するだけでなく、実際にペトリネットを視覚的に作成できるソフトウェアを開発することも重要である。特に、モデル化の対象とする仮想空間が複雑になった場合には、作成すべきペトリネットが巨大になり、その作成及び修正に多くの時間と労力が必要となる。そこで、本研究では、ペトリネットの作成を支援するソフトウェアとしてペトリネット作成支援システム PCSS を開発した。

3.5.1 PCSS 設計の基本方針

機器保修の作業を対象にモデル化したペトリネットには、同じ構造のペトリネットを多数組み合わせた部分が存在する場合がある。例えば、本研究で構築の対象にするスイング式逆止弁の保修作業を対象にモデル化したペトリネットの中には、Fig. 3.10や Fig. 3.11に示すようなペトリネットが存在する。Fig. 3.10と Fig. 3.11はそれぞれ、ペンを掴む作業と、ナットを掴む作業をモデル化したペトリネットである。これらの2つのペトリネットは、互いに「ペン」と「ナット」を入れ替えた構造をしており、共に Fig. 3.12に示すペトリネットの「物」の部分に「ペン」や「ナット」を入れ替えた構造であることが分かる。従って、Fig. 3.12に示したような頻繁に使用されるペトリネットを予め作成しておき（以下では予め作成したペトリネットをテンプレートと呼ぶ）、必要に応じてそのペトリネットを修正して利用できるソフトウェアが存在すれば、ペトリネット作成効率が飛躍的に改善されるものと期待される。そこで、本研究では、PCSS を開発するにあたり、ペトリネットを視覚的に構築できる環境を提供するだけでなく、機器保修の作業を対象にモデル化したペトリネットに現れる構造的類似性に着目し、少ない労力で大規模なペトリネットを構築できる機能も提供する。また、ペトリネットを作成する際に、作成したペトリネットが意図した

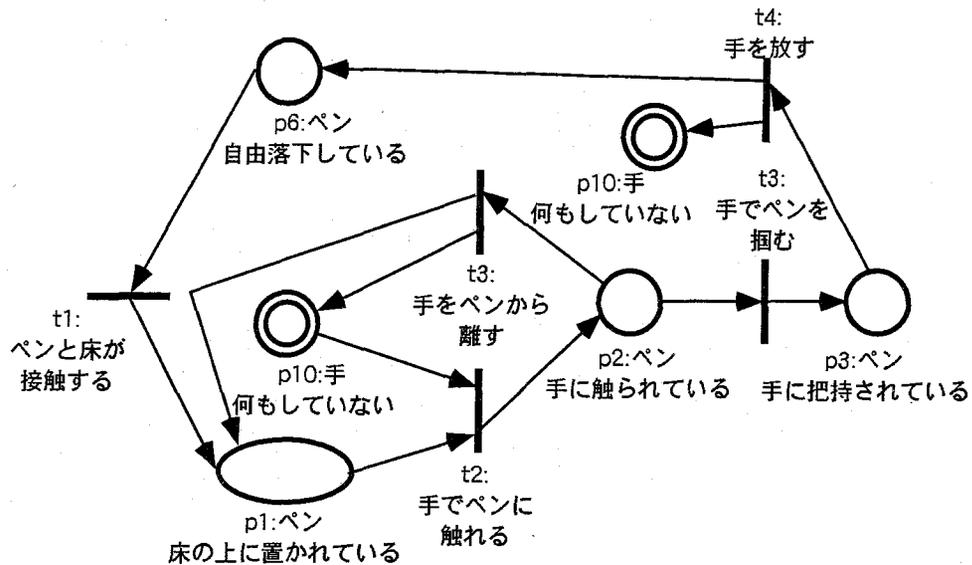


Fig. 3.10: ペンを掴む作業をモデル化したペトリネット

通りに働くかどうかを、容易に確かめることができることは、作業効率を上げるために重要である。従って、PCSS では、ペトリネットを作成中の任意の時点で、ペトリネットシミュレータを用いてその動作を確認できる機能も提供する。

具体的には、PCSS を、グラフィカル・ユーザ・インタフェース構築用の言語 Tcl/Tk[1, 2] を用いて開発した。Tcl/Tk で作成したアプリケーションは、基本的にはプログラムを変更する事なしに、UNIX、Macintosh、Windows の各 OS(Operating System) 上で実行できるが、今回は、シミュレーションサブシステムを UNIX 上で実行することから、PCSS も UNIX 上で開発するものとした。

3.5.2 PCSS の概要

PCSS は、頻繁に使用されるペトリネットの構造を、トランジション、プレース単位で作成し、テンプレートとして登録するモード (テンプレート作成モード)、複数のテンプレートを結合して訓練システムで使用する最終的なペトリネットを作成するモード (テンプレート結合モード)、作成したペトリネットの動作をペトリネットシミュレータを用いて確認するモード (動作確認モード) の 3 つのモードが存在する。

テンプレート作成モード

テンプレート作成モードでは、Fig. 3.13に示すインタフェースを介して、トランジション、プレース単位でテンプレートを作成する。テンプレート作成モードでのメニューの階層構造を Fig. 3.14に示す。新しいテンプレートを作成する場合、ユーザは、画面下部の「テ

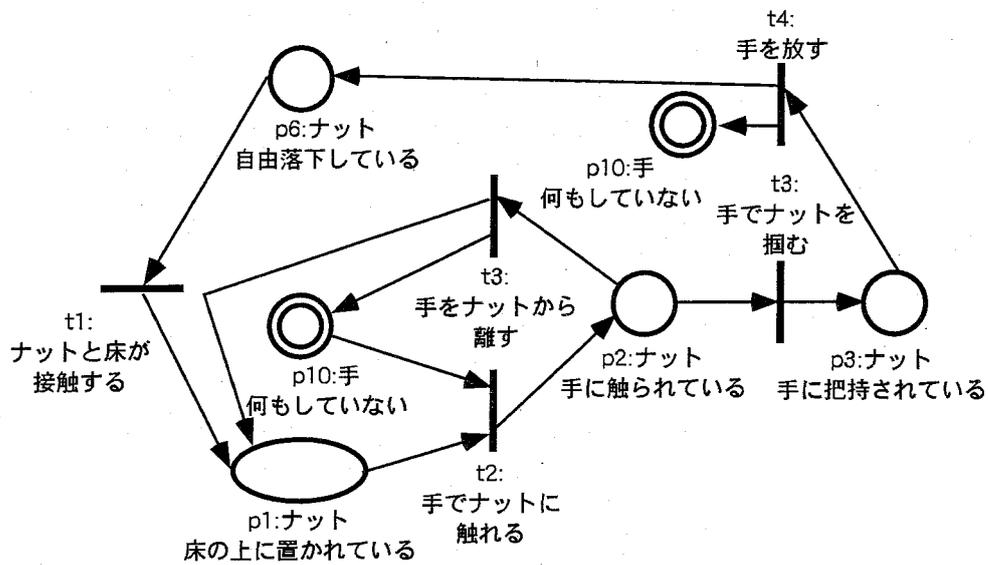


Fig. 3.11: ナットを掴む作業をモデル化したペトリネット

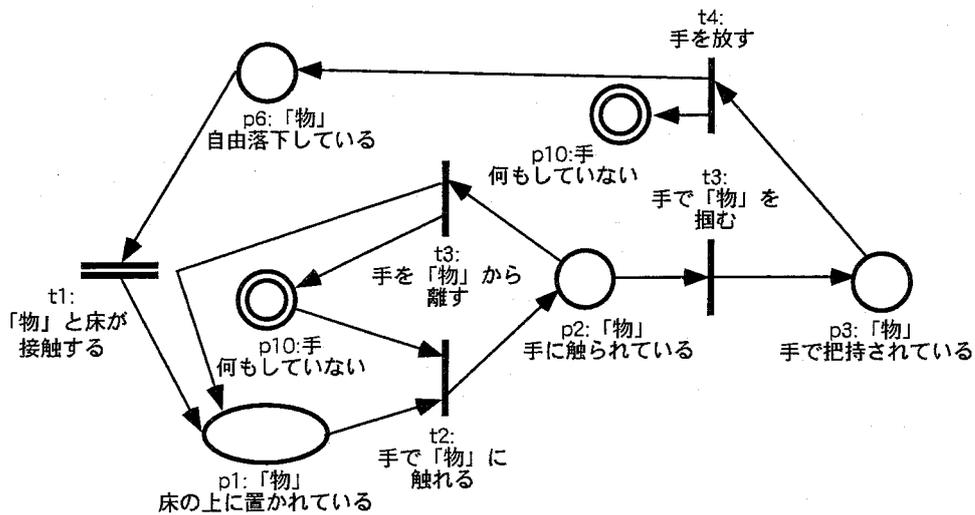


Fig. 3.12: 「物」を掴む作業をモデル化したペトリネット

ンプレート」メニューから「新規作成」を選択する。また、「編集」から「トランジション作成」を選択すると、トランジションの名称と、そのトランジションが発火した場合に実行するタスク関数名が尋ねられるので、キーボードを介して入力する。ここで、トランジションの名称の先頭に「!」を付加すると、そのトランジションは自動発火トランジションとなる。入力が終了すると、画面上に新しいトランジションが名前と共に配置される。同様に「編集」から「プレース作成」を選択すると、プレースの名称が尋ねられるので、同様に入力する。ここでも、プレースの名称の先頭に「!」を付加すると、そのプレースは制御プレースとなり、「&」を付加すると、そのプレースはプールプレースとなる。さらにプレースの名前の中に () で囲まれている部分があると、その部分は、テンプレートを使用する際に別の名前に置き換えられることを意味する。こちらでも、入力が終了すると、画面上に新しいプレースが名前と共に配置される。作成したトランジションやプレースは「編集」メニューから「場所移動」を選択すると、2次元マウスを用いて場所を移動できる。トランジションの入出力プレースを設定する場合は、「編集」メニューから「アーク作成」を選択する。その後、画面上のプレースをマウスで選択し、続いてトランジションを選択すると、最初に選択したプレースはトランジションの入力プレースとして登録され、画面上にアークが描かれる。同様に、「アーク作成」を選択後、画面上のトランジションを選択し、続いてプレースを選択すると、そのプレースはトランジションの出力プレースとなる。作成したテンプレートは、「テンプレート」メニューの「保存」を選択し、テンプレート名と、そのテンプレートの内容を説明する文章を入力すると、テンプレートとして登録され、テンプレート結合モードで訓練システムで使用するペトリネットを作成する際に使用できるようになる。

以上のように、視覚的にペトリネットを作成できるのが PCSS の大きな特徴であり、これにより、仮想物体の状態遷移の仕方を容易にモデル化できると期待される。

テンプレート 結合モード

テンプレート結合モードでは、Fig. 3.15に示すインタフェースを介して、複数のテンプレートを結合し、訓練システムで使用するペトリネットを作成する。PCSS では、「別々のネットにある同じ名称のプレースは同一のものとして扱う」という規則を導入することによりテンプレート間の結合を可能にした。この場合、テンプレートの結合を正しく行うためには、個別にテンプレートを作成する場合でも、仮想物体の同じ状態を表すプレースの名称が、一意に定まる必要がある。従って、本研究では、以下の規則に従って、プレースを命名するものとした。

- プレースは全て「物体の状態」-「他の物体との関係1」-「他の物体との関係2」.....「他の物体との関係n」-「その他特別な記述」-「物体の名称」の形式で名前を付ける。

また、「物体の状態」として使用する言葉と「他の物体との関係」として使用する言葉を、Table 3.6及び Table 3.7に示すように限定し、さらに「他の物体との関係」が複数ある場合には、それらをアルファベット順に並べるものとした。なお、プレースの命名規則の記述の中で、「その他特別な記述」とあるのは、Table 3.6及び Table 3.7に示した言葉だけでは

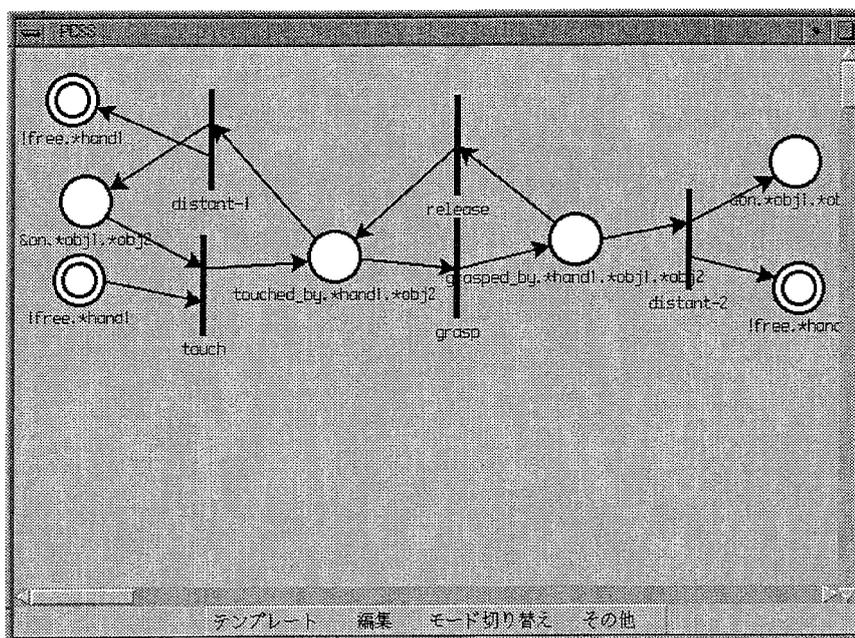


Fig. 3.13: テンプレート作成モードのインターフェース

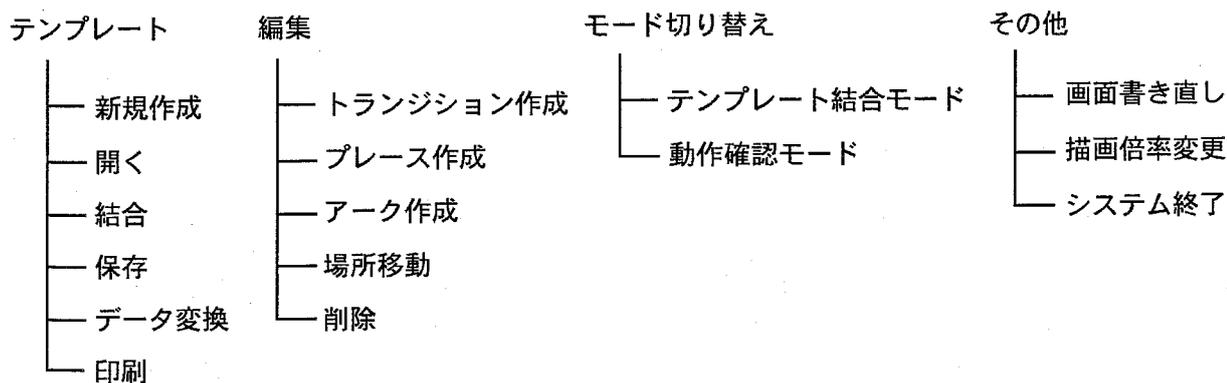


Fig. 3.14: テンプレート作成モードのメニューの階層構造

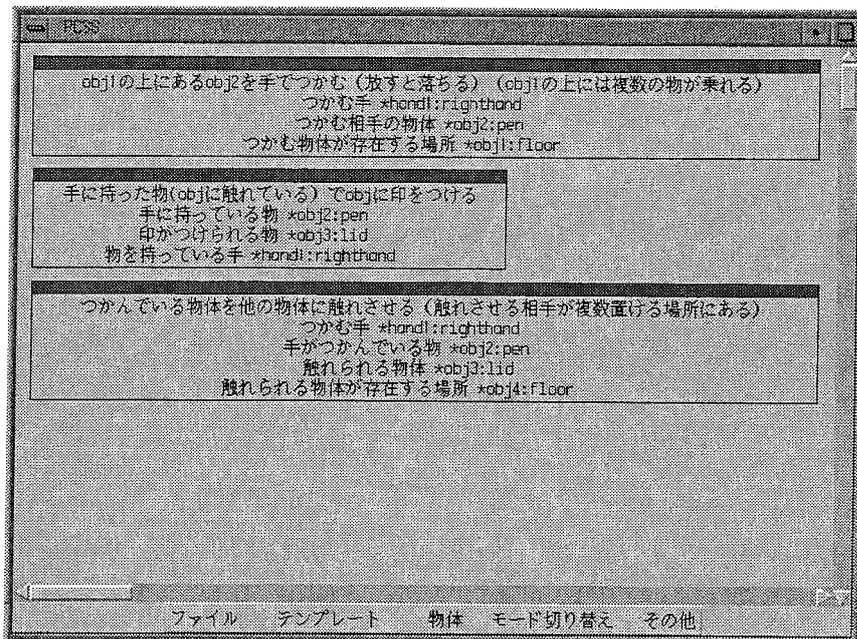


Fig. 3.15: テンプレート結合モードのインタフェース

表現できない状態がある場合に対応するためであり、構築する訓練環境ごとに予め使用できる言葉を定めておくことが望ましい。

テンプレート結合モードでのメニューの階層構造を Fig. 3.16に示す。テンプレート結合モードでペトリネットを作成する場合、まず、ペトリネット内で使用する物体の名称を登録する。ユーザは、画面下部の「物体」メニューから「登録」を選択し、登録する物体名をキーボードを介して入力する。テンプレート作成モードで作成したテンプレートを使用する場合は、「テンプレート」から「テンプレート張り付け」を選択する。すると、テンプレート作成モードでテンプレートを作成した際に、プレースの名称に () を使用した部分に入る物体名を尋ねてくるので、登録した物体名のリストから選択して指定する。同様の手順で複数のテンプレートを張り付け、最終的に PCSS にテンプレートを展開させることにより、最終的なペトリネットを作成できる。なお、PCSS には、既に頻繁に使用されるペトリネットの構造として、Table 3.8に示す 16 個のペトリネットが登録されている。

動作確認モード

動作確認モードでは、Fig. 3.17に示すインタフェースを介して、作成したペトリネットの動作を確認する。ペトリネットシミュレータを用いてペトリネットの動作を確認する場合、トークンの初期配置を決定する必要がある。PCSS の動作確認モードでは、画面下部の「トークン追加」ボタンを押し、続いて画面上のプレースを選択すると、そのプレースにトークンが 1 つ追加され、プレースの上部にトークンの総数が表示される。また、「トー

3. 人工現実感技術を用いた機器保守の訓練システム VRMATE の開発

Table 3.6: 「物体の状態」として使用する言葉

| | 物体の状態 | 意味 |
|---|-------------------|----------------|
| 1 | free | 何もされていない |
| 2 | falling | 落ちつつある |
| 3 | touched_by_(hand) | (hand) に触られている |
| 4 | grasped_by_(hand) | (hand) に握られている |

Table 3.7: 「他の物体との関係」として使用する言葉

| | 他の物体との関係 | 意味 |
|---|-----------------|-------------------|
| 1 | on_(object) | (object) の上に乗っている |
| 2 | in_(object) | (object) の中にある |
| 3 | touch_(object) | (object) に触れている |
| 4 | fit_in_(object) | (object) にはまっている |

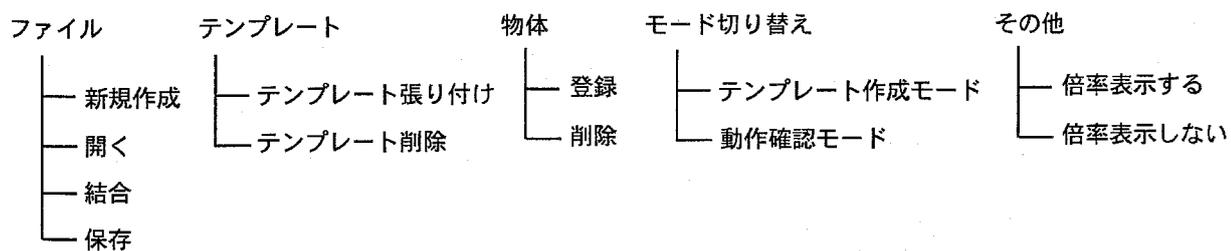


Fig. 3.16: テンプレート結合モードのメニューの階層構造

3.5 ペトリネット 作成支援システム PCSS の開発

Table 3.8: テンプレートとして既に登録されているペトリネット

| | 内容説明 |
|------|---|
| A-1 | (物体1)の上にある(物体2)を(手)で掴む。 (物体2)は手放すと(物体1)の上に落ちる。 物体1の上には複数の物体が乗ることができる。 |
| A-2 | (物体1)の上にある(物体2)を(手)で掴む。 (物体2)は手放しても落ちない。 物体1の上には複数の物体が乗ることができる。 |
| A-3 | (物体1)の上にある(物体2)を(手)で掴む。 (物体2)は手放すと(物体1)の上に落ちる。 物体1の上には1つの物体しか乗ることができない。 |
| A-4 | (物体1)の上にある(物体2)を(手)で掴む (物体2)は手放しても落ちない。 物体1の上には1つの物体しか乗ることができない。 |
| A-5 | (手)に掴んでいる(物体1)を(物体2)に接触させる。 |
| A-6 | (手)に握っている(物体1)で(物体2)に(印)を付ける。 物体1と物体2は既に接触している。 |
| A-7 | 手放された(物体1)が(物体2)の上に落ちる。 |
| A-8 | (物体1)を(手)で取り外す。 |
| A-9 | (物体2)を(手)で持ち上げる。 |
| A-10 | (ボルト)にはまっている(ナット)に(スパナ)をはめる。 ボルトには2つのナットをはめることができる。 |
| A-11 | (ボルト)にはまっている(ナット)に(スパナ)をはめる。 ボルトには1つのナットをはめることができる。 |
| A-12 | (ボルト)にはまっている(ナット)を(スパナ)を使って、 (手)で回す。回し続けるとナットは外れる。 |
| A-13 | (ボルト)にはまっている(ナット)を(手)で回す。 回し続けるとナットは外れる。 |
| A-14 | (手)に握っている(ナット)を(ボルト)にはめる。 |
| A-15 | (ナット)にはまった(スパナ)でダブルナット法で (ボルト)を緩める。 |
| A-16 | (ナット)にはまった(スパナ)でダブルナット法で (ボルト)を外す。 |

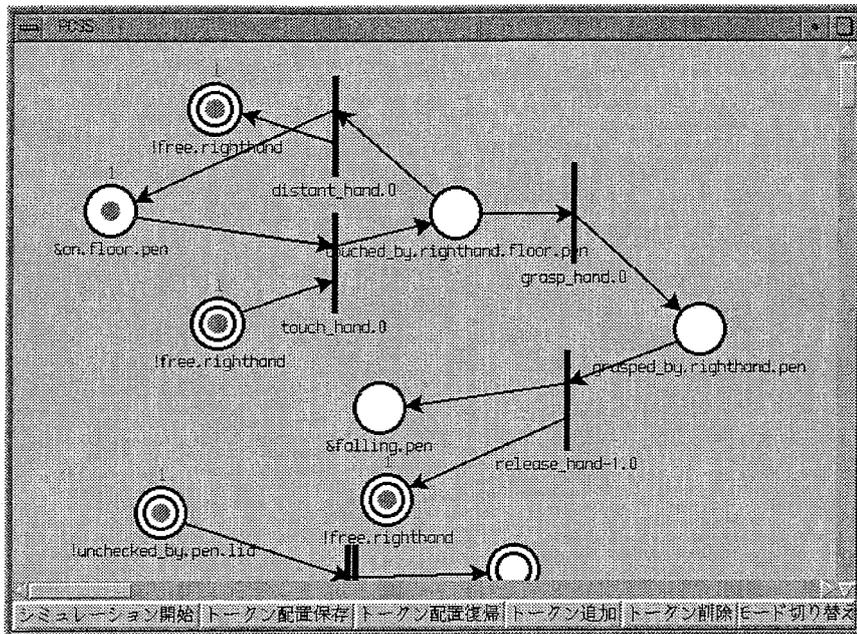


Fig. 3.17: 動作確認モードのインターフェース

「トークン削除」ボタンを押し、続いて画面上のプレースを選択すると、そのプレースからトークンが1つ削除される。「シミュレーション開始」ボタンを押すと、ペトリネットのシミュレーションが始まり、発火可能なトランジションが黒色から赤色に変わる。赤色に変わったトランジションの内、発火させたいトランジションをマウスで選択すると、そのトランジションが発火し、トークンが遷移し、再び発火可能なトランジションの色が赤色になる。画面下部の「トークン配置保存」ボタンを押すことにより、任意の時点のトークン配置が保存でき、「トークン配置復帰」ボタンを押すことにより、保存したトークン配置に戻すことができる。このように、PCSS では、モデル化したペトリネットが意図した通りに動くかどうかを容易に確かめることができる。

3.5.3 PCSS を用いたペトリネットの作成

本項では、PCSS の使用例として、まず、「床に置いてあるペンでふたに印を付ける作業 (以下、ペンで印を付ける作業)」を対象にモデル化したペトリネットを作成する方法を説明する。ペトリネットの作成は、前述のテンプレート結合モードで行われる。

手順1 PCSS に物体を登録する。ペンで印を付ける作業では、床、ペン、ふた、印、手の5つの物体が仮想空間内に配置される。従って、「物体」メニューの「登録」を選択し、これら5つの物体の名称を順次登録する。

Table 3.9: ペンで印を付ける作業を単位タスクに分解した結果

| | 単位タスク |
|-------|--|
| タスク 1 | 手でペンを触る |
| タスク 2 | 手でペンを握る |
| タスク 3 | 握ったペンをふたに接触させる |
| タスク 4 | 握ったペンで印を付ける (ペンはふたに接触した状態のままペンを移動させる) |
| タスク 5 | 手に握ったペンを手放す |

手順 2 構築対象の作業を単位タスクに分解する。ペンで印を付ける作業を単位タスクに分解した結果を Table 3.9に示す。

手順 3 PCSS で既に用意されたテンプレートのリストから使用するテンプレートを選択する。「テンプレート」から「テンプレート張り付け」を選択すると、Table 3.8に示したテンプレートのリストが表示される。ここで、テンプレート A-1には、手順 2で得られた単位タスクのうち、タスク 1とタスク 2及びタスク 5が含まれる。そこで、テンプレートリストから A-1 を選択する。すると、PCSS が、「(物 1) に相当する仮想物体の名称」、「(物 2) に相当する仮想物体の名称」、「(手) に相当する仮想物体の名称」を順に尋ねてくるので、それぞれ、床、ペン、手を物体のリストから選択する。同様に、タスク 3とタスク 4に相当するテンプレートである A-5と A-6をそれぞれ選択し、質問に答える。

手順 4 テンプレートを展開する。PCSS にテンプレートを展開させると Fig. 3.18に示すようにペンで印を付ける作業をモデル化したペトリネットが得られる。

このように、簡単な作業をモデル化の対象とする場合は、殆どの場合、既に作成されているテンプレートを選択し、必要な修正を行うだけで、容易にペトリネットが作成できる。また、必要なテンプレートが作成されていない場合でも、テンプレート作成モードで独自のテンプレートを作成し、PCSS に登録しておくことで、同じ構造のペトリネットを繰り返し作成する必要がなくなり、作業効率が改善されると期待される。実際に、3.3節で述べたスイング式逆止弁の 1 から 6 までの分解作業を対象にしたペトリネットを作成したところ、約 300 回程度の入力作業で目的のペトリネットを作成でき、作成したペトリネットのトランジションの総数が 367 個、プレースの総数が 158 個となった。この作業を PCSS を使用せずに行う場合には、プレースとトランジションの名称及び種類、プレースとトランジションの結合関係等のデータを個々に入力する必要があり、膨大な入力作業が必要なため、正しくデータを入力することが非常に困難になると予想される。

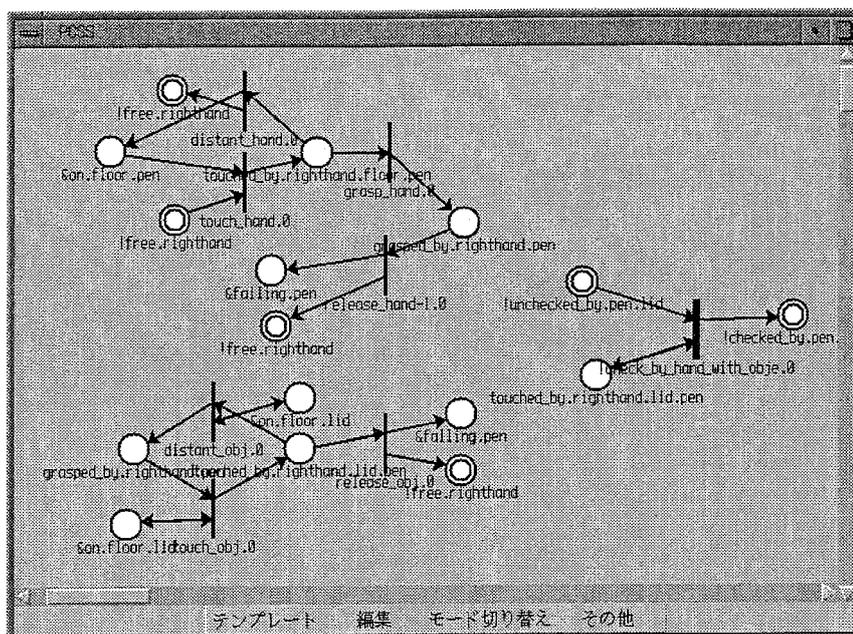


Fig. 3.18: 展開されたペトリネット

3.6 機器保修の自動実演機能

本節では、訓練生の要求に応じて次に実行すべき作業を実演する自動実演機能について述べる。自動実演機能は、訓練生から自動実演の要求があった場合、トランジション単位で次に実行すべき作業を画面上で実演する形をとることとする。続けて自動実演を要求することにより、複数のトランジションに渡る自動実演を観察できる。ここで、次に実行すべき作業とは、訓練生が自動実演機能を使用する前に、間違った作業を行っている場合は、その間違った作業を修正する作業が次に実行すべき作業であり、間違った作業が無い場合は、機器保修の作業を先に進める作業が次に実行すべき作業である。

自動実演機能を実現するためには、訓練生から自動実演が要求された時点における仮想物体の各状態に応じて、次に実施すべき作業を的確に定めるアルゴリズムを考案する必要がある。このような「初期状態と目的状態が与えられた時に、その目的状態に至るまでの状態遷移の過程を検索する」問題に対する研究は、人工知能の分野において様々な試みがなされてきた [3, 4, 5]。代表的なものとしては、Stanford Research Institute で開発されたロボットの行動計画立案用プログラム STRIPS (Stanford Research Institute Problem-Solver) がある。STRIPS では、物体の状態が論理式の積によって記述され、ロボットの作業は、前提条件（その作業を実行する際に成立している必要がある条件）、追加リスト（その作業を実施することにより状態に追加される論理式）、削除リスト（その作業を実施することにより状態から削除される論理式）の3項目によって表現される。このプログラムではまず、初期状態から直接に目的状態が導き出されるかどうかを調べる。導けなければ初期状

態と目的状態の差異を算出し、その差異を減少させる作業を検索する。作業が見つかり、その作業の前提条件が満たされているかどうかを調べる。満たされていれば、その作業が実行され、初期状態と目的状態の差異が減少させられる。満たされていなければ、その前提条件を満足させることを副目標として、初期状態と前提条件との差異を減少させるための作業の検索を繰り返す。STRIPSでは、このように初期状態と目標状態の差を減少する方向の検索を優先的に進めるような検索手段をとることにより、状態遷移の過程の検索に必要な計算量を軽減している。STRIPSは1971年にFikesらによって発表されて以来、作業計画自動生成システムの手本となっているが、深さ優先による作業の検索方法であること、目標を全て同一の優先度で取り扱っていることなどから、初期状態と目標状態の差異を減少させる作業が、必ずしも正しい作業でない場合がある機器の保修作業の作業手順の検索には、そのままでは適用することができない。

そこで本研究では、作業手順を検索する新しいアプローチの方法として、「仕事」の概念を導入した作業手順の検索のアルゴリズムを考案した。具体的には、仕事間の優先順序に関する情報を用いることにより、次に発火すべきトランジションを決定し、その後、対応するタスク関数を実行する。タスク関数では、自動実演の映像を合成するための処理を行うプログラムが作成されており、自動実演モードの時にタスク関数が実行されると、対応するトランジションの入力プレースに配置されているトークンの種類に応じて、仮想物体の動きを合成する。例えば、物体を触るトランジションに対応したタスク関数が実行された場合、トランジションの入力プレースに存在するトークンに対応する仮想物体を調べ、その仮想物体に向かって、手のモデルが移動する映像が合成される。ここで「仕事」を次のように定義する。

「仕事」の定義 手に何も持たない状態を開始時点として、一連の単位タスクを行い、再び手に何も持たない状態になったときに、ペトリネットのトークン配置が開始時点から変化している場合、その一連の単位タスクの組み合わせを「仕事」と呼ぶ。また特に、トークン配置を変化させる「仕事」の内、複数の「仕事」の組み合わせに分解できないものを「最小の仕事」と呼び、以下では、単に「仕事」といえば「最小の仕事」を指すものとする。

例えば、「レンチで1個のナットを緩める」作業は1つの仕事であり、「レンチを手で触り」、「レンチを手で握り」、「レンチをナットにはめて」、「ナットを緩めて」、「レンチをナットから外し」、「レンチを手放す」という一連の単位タスクの組として表現される。しかし、「レンチをつかんでナットにはめ、レンチを手放す」作業は、作業の前後でトークン配置が変化しないため、仕事には該当しない。

以下では、まず、自動実演機能を構築する際に考慮すべき事柄について述べ、その後、本研究で考案した、次に発火すべきトランジションを決定する方法について述べる。

3.6.1 自動実演機能を構築する際に考慮すべき事柄

訓練生から自動実演の要求があった場合、以下に示す規則に従って次に発火すべきトランジションを決定する必要がある。

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

規則1 訓練生がある仕事の途中で自動実演機能を要求した場合 (手に物体を持っている場合)

- 今行っている仕事正しい仕事である場合は、その仕事を先に進める。
- 今行っている仕事間違っ仕事である場合は、その仕事をもとに戻す。

規則2 訓練生が仕事の途中でない時に自動実演機能を要求した場合 (手に何も持っていない場合)

- 訓練生がそれまでに行った仕事に間違いがある場合は、その間違っ仕事を修正する。
- 訓練生がそれまでに行った仕事に間違いがない場合は、次にすべき仕事を先に進める。

また、次のような事柄も、次に発火すべきトランジションを決定する場合に考慮する必要がある。

- 自動実演機能が要求された時点で次に実行すべき仕事は、必ずしも1つには定まらない。例えばナットを外すべきボルトが2本あって、その両方に同じようにナットがはまっている場合、どちらのナットを先に外してもよい。
- ある時点で間違っていた仕事でも、作業を進めると修正する必要がなくなると考えてもよい場合がある。例えば、スイング式逆止弁の分解作業に関して、ペンでふたと弁箱に印を付ける前にボルトのナットを緩めた場合、「ナットを緩める」という仕事は、この時点では間違いであるため修正する必要があるが、その後、作業を続けることにより、ペンでふたと弁箱に印を付けた場合は、その時点から「ナットを緩める」という仕事を修正する必要はないと考えることもできる。
- 作成したペトリネットの性質上、トランジションに、右手のみに関係するもの、左手のみに関係するもの、両手に関係するものがある。そのため、右手がある仕事を行う途中で自動実演機能が要求された場合、右手に関するトランジションを指示せねばならない。
- 機器保修作業における仕事の組み合わせの中には、ある仕事を実行すると、他の仕事の実行できなくなる組み合わせが存在する場合がある。例えば、ナットがはまっていないボルトがある場合、「ボルトにふたをはめる」仕事は実行できるが、「ボルトにナットをはめる」仕事を実行すると、「ボルトにふたをはめる」仕事の実行できなくなる。

本研究では、以上の事柄を考慮して次に発火すべきトランジションを決定するアルゴリズムを考案した。

3.6 機器保守の自動実演機能

Table 3.10: 自動実演を行うために必要な情報

| | |
|-------|--|
| Info1 | そのトランジションが関係する手 (右手、左手、両手の3つの場合があり得る) |
| Info2 | そのトランジションの発火によって進行する仕事 (複数の仕事があり得る) |
| Info3 | そのトランジションが発火した後に Info2 で設定した各仕事を進行させる場合に発火すべきトランジション |
| Info4 | そのトランジションが発火した後にその発火の効果をもとに戻す場合に最初に発火すべきトランジション |
| Info5 | そのトランジションが発火することにより終了する仕事 (存在しない場合もあり得る) |

Table 3.11: 変数一覧

| 変数名 | 意味 |
|-----------|--|
| Tr_R | 右手もしくは両手に関して次に発火すべきトランジション |
| Tr_L | 左手に関して次に発火すべきトランジション |
| Task_todo | 次に実行すべき仕事のリスト (リスト中の仕事の優先度は同じ) |
| Task_cor | 間違いを修正するために実行すべき仕事のリスト (リスト中の仕事は先頭が最も優先順位が高い) |
| Task_proc | 機器保守作業の正しい作業手順に応じた仕事の組のリスト (同じ優先順位の仕事が1つの組として格納される) |

3.6.2 次に発火すべきトランジションを決定するアルゴリズム

本項では、次に発火すべきトランジションを決定するアルゴリズムについて述べる。まず、自動実演の準備として、対象とする機器の保守作業で必要となる仕事をすべて見だし、プログラム内で判別できるように各仕事に名前を付ける。この際、同じ内容の仕事が重複して現れないようにする。次に、各仕事に対して、「その仕事を実行するために最初に発火すべきトランジション」と「その仕事と逆の働きをする仕事」を設定する。更に、ペトリネット上の全てのトランジションに対し、Table 3.10に示すデータを予め設定しておく。なお、以下では説明の便宜上、Table 3.11に示す変数を用いる。

次に発火すべきトランジションを決定するアルゴリズムは、訓練生の作業によってトランジションが発火した際に実行する、訓練生の作業を監視するアルゴリズムと次に発火すべきトランジションを求めるアルゴリズムの大きく2つに分けることができる。以下では、それぞれのアルゴリズムについて説明する。

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

訓練生の作業を監視するアルゴリズムについて

Fig. 3.19は訓練生の作業を監視するアルゴリズム (以下、監視アルゴリズム) である。以下に、監視アルゴリズムを実行する際の処理の流れを説明する。

- 処理 1 訓練生のジェスチャに応じてトランジションが発火する。
- 処理 2 発火したトランジションが進める仕事 (Info2) と、次に実行すべき仕事 (Task_todo) との間に共通の仕事が存在するか調べる。
- 処理 3-1 処理 2 の判定で、共通の仕事が存在すれば、発火したトランジションが、次にすべき仕事を進めることになり、訓練生が正しい作業をしていることになる。この場合は、トランジションの発火によって終了する仕事 (Info5) が存在すれば、その仕事を Task_todo から削除し、さらに仕事を先に進めるトランジション (Info3) を次のトランジションとする。
- 処理 3-2 処理 2 の判定で、共通の仕事が存在しなければ、今度は、Info2 と間違いを修正するために実行すべき仕事 (Task_cor) との間に共通の仕事が存在するか調べる。
- 処理 4-1 処理 3-2 の判定で、共通の仕事が存在すれば、訓練生が過去の間違いを修正していることになる。この場合は、Info5 が存在すれば、その仕事を Task_cor から削除し、Info3 を次のトランジションとする。
- 処理 4-2 処理 3-2 の判定で、共通の仕事が存在しなければ、訓練生の行動が間違っていたことになるので、このトランジションの発火によってなんらかの仕事が終了する場合、その仕事は修正する必要がある。そこで、この終了した仕事と逆の働きをする仕事を Task_cor の先頭に加え、仕事をもとに戻すトランジション (Info4) を次のトランジションとする。
- 処理 5 次のトランジションが左手に関係するものかどうか調べる。
- 処理 6-1 左手に関係するトランジションである場合、次のトランジションを Tr.L に代入する。
- 処理 6-2 左手に関係するトランジションでない場合 (右手もしくは両手の場合)、次のトランジションを Tr.R に代入する。
- 処理 7 Task_todo に仕事が残っていなければ作業手順に従った次の仕事のリスト (Task_proc) の先頭の仕事は Task_todo に代入し、その仕事を Task_proc から削除する。(ここで Task_todo に代入する仕事が残っていなければ、作業は終了である。)
- 処理 8 作業を進めるために次に実行すべき仕事 (Task_todo) と間違いを修正するために実行すべき仕事 (Task_cor) との間に、互いに逆の働きをする仕事が存在すれば、それらの仕事を共に削除する。

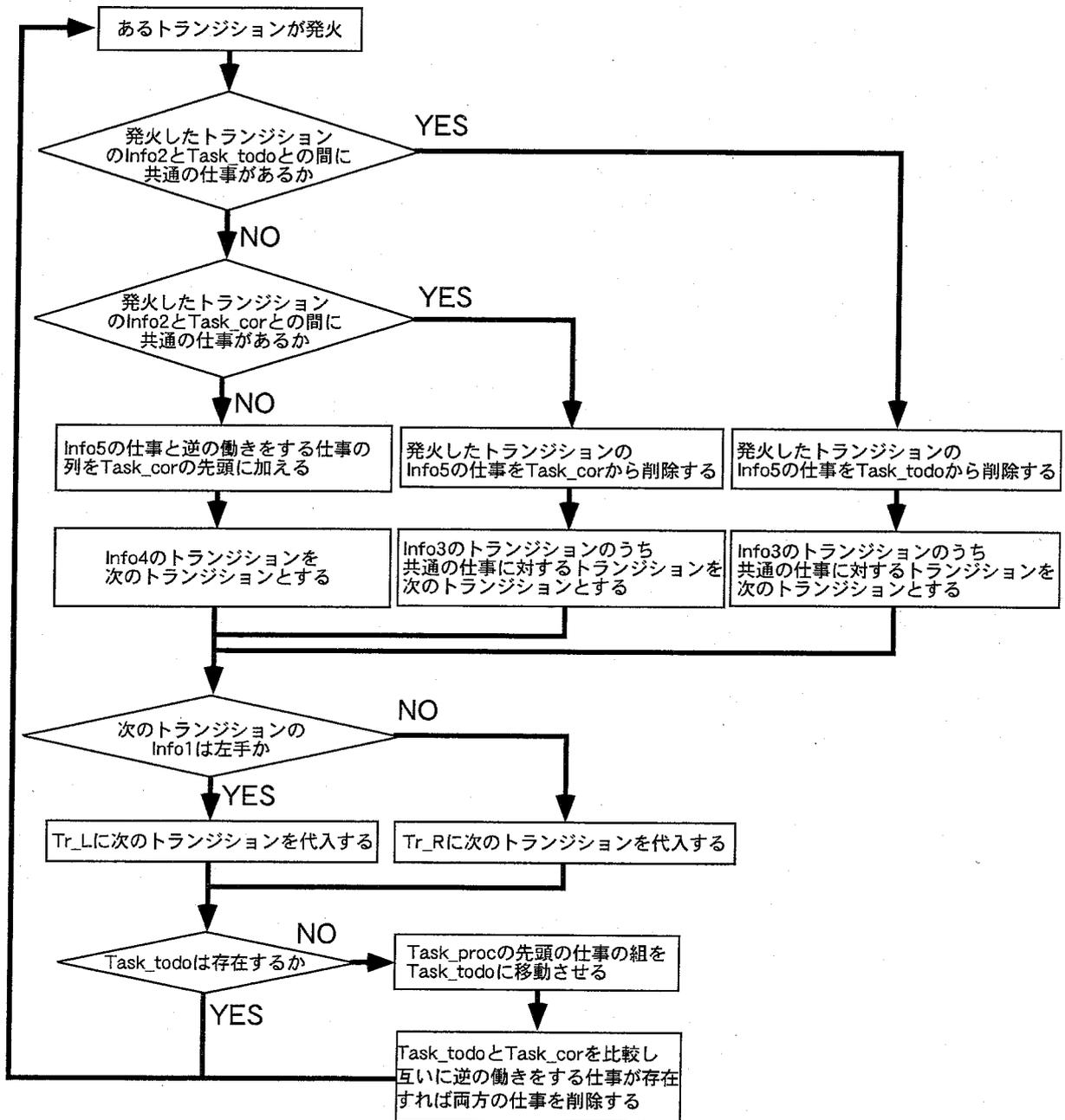


Fig. 3.19: 訓練生の作業を監視するアルゴリズム

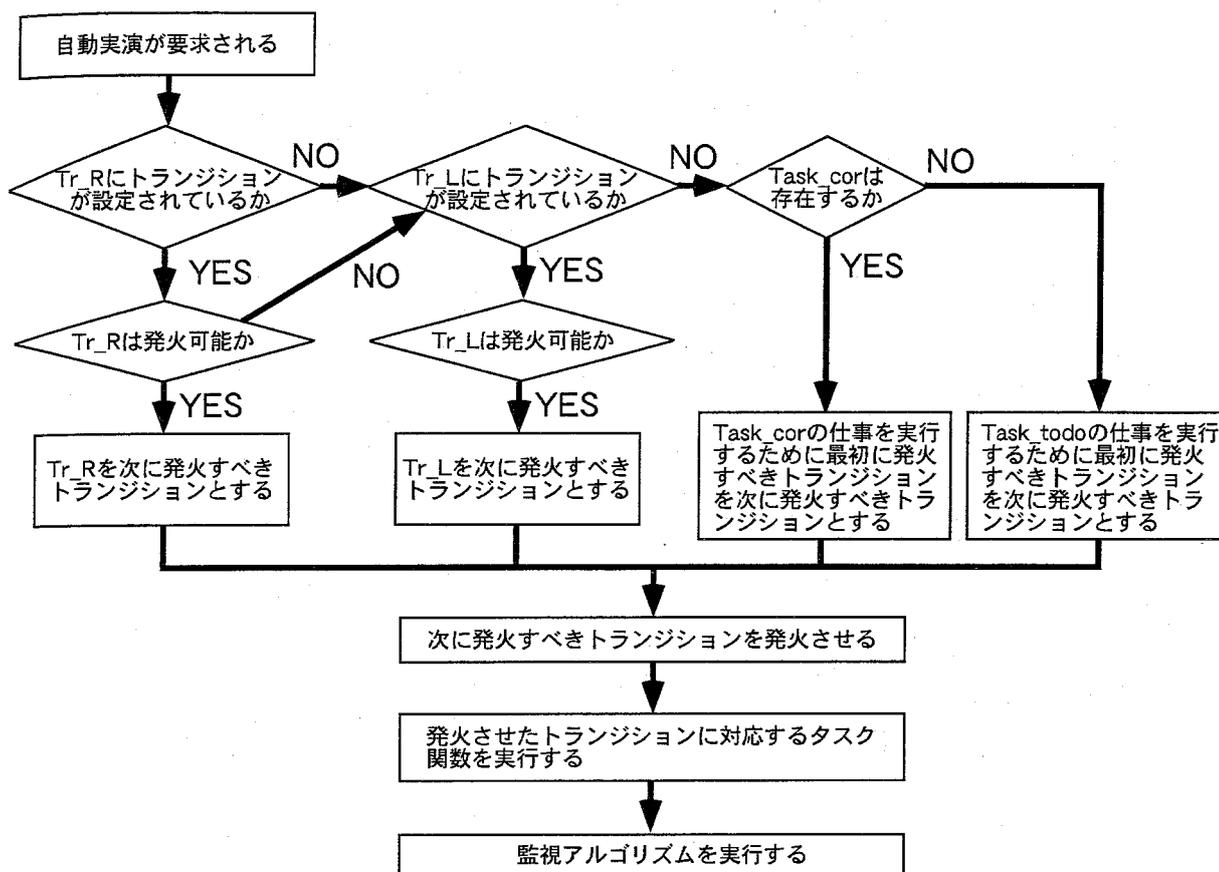


Fig. 3.20: 次に発火すべきトランジションを求めるアルゴリズム

上記の処理 8 について、例えば、訓練生が作業の手順を間違えて、「ペンで印を付ける」仕事の前に「ナットを緩める」仕事をした場合、Task_cor にはその仕事を修正する「ナットを締める」仕事が代入される。しかし、訓練生が「ペンで印を付ける」仕事を終了し、次にすべき仕事として「ナットを緩める」仕事が Task_todo に代入されても、ナットは既に緩まっており、またナットを締め直す必要もないと考えられる。従って Task_cor の「ナットを締める」仕事と Task_todo の「ナットを緩める」仕事は共に消去すべきである。このような場合に対応するために、処理 8 に示したような仕事の消去を行う。

次に発火すべきトランジションを求めるアルゴリズム

Fig. 3.20は次に発火すべきトランジションを求めるアルゴリズムである。以下に、次に発火すべきトランジションを求めるアルゴリズムを実行する際の処理の流れを説明する。

処理 1 訓練生から自動実演が要求される。

3.7 VRMATE の開発

処理 2 Tr_R にトランジションが設定されており、そのトランジションが発火可能であれば、そのトランジションを次に発火すべきトランジションとして発火させ、そのトランジションに対応したタスク関数を実行する。

処理 3 Tr_R にトランジションが設定されていないか、設定されていても発火不可能な場合は、Tr_L にトランジションが設定されているか調べる。Tr_L にトランジションが設定されており、そのトランジションが発火可能であれば、そのトランジションを次に発火すべきトランジションとして発火させ、そのトランジションに対応したタスク関数を実行する。

処理 4 Tr_R と Tr_L に共に発火可能なトランジションが存在しない場合、Task_cor が存在するか調べ、存在する場合は、その仕事を実行するために最初に発火すべきトランジションを次に発火すべきトランジションとして発火させ、そのトランジションに対応したタスク関数を実行する。

処理 5 Task_cor にも仕事が存在しない場合には、Task_todo の仕事を実行するために最初に発火すべきトランジションを次に発火すべきトランジションとして発火させ、そのトランジションに対応したタスク関数を実行する。

処理 6 処理 2～5 でタスク関数を実行した後に監視アルゴリズムを実行する。

以上のアルゴリズムにより、次に発火すべきトランジションが決定され、最終的に、仮想空間内で自動実演が 3 次元映像として合成される。本研究では、以上のアルゴリズムをスイング式逆止弁の分解作業に対して適用し、正しく動作することを確認した。

3.7 VRMATE の開発

本節では、VRMATE の概要について述べた後、VRMATE を第 3 者に試用してもらうことによって、その評価を行った結果を述べる。

3.7.1 VRMATE の概要

本研究で開発した VRMATE のハードウェア構成を Fig. 3.21 に、ソフトウェア構成を Fig. 3.22 に示す。VRMATE も、一般の人工現実感を実現するシステムと同様に、計測サブシステム、シミュレーションサブシステム、表示サブシステムの 3 つのサブシステムより構成される。

計測サブシステムは、データグローブ及びポヒマスセンサで構成される。データグローブとポヒマスセンサは、2.1.1 項で述べたように、訓練生の手を用いたジェスチャを計測するデバイスであり、力覚や触覚は提供されない。これは、本研究で開発する VRMATE が、作業の手順や、機器の構造等、保修作業の知識を修得することを目標としたためである。

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

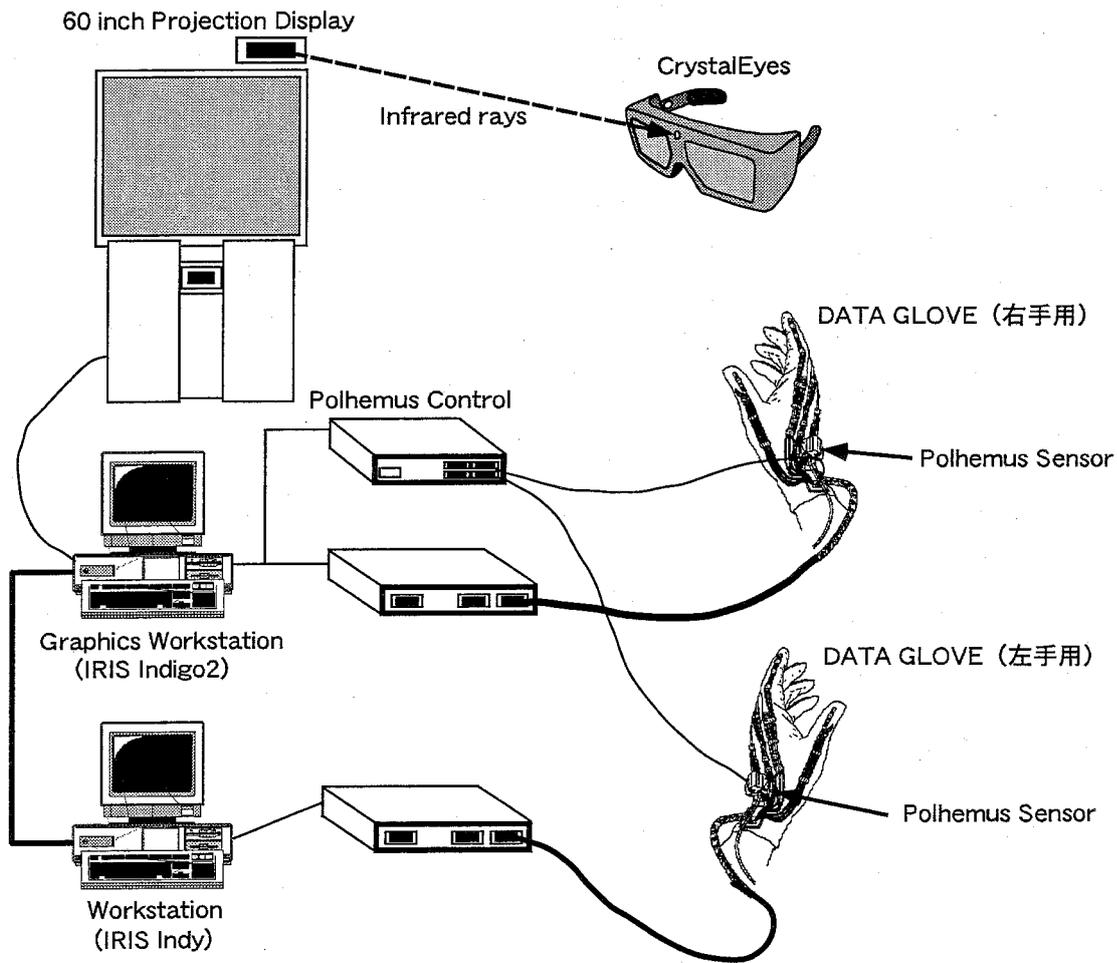


Fig. 3.21: VRMATE のハードウェア構成

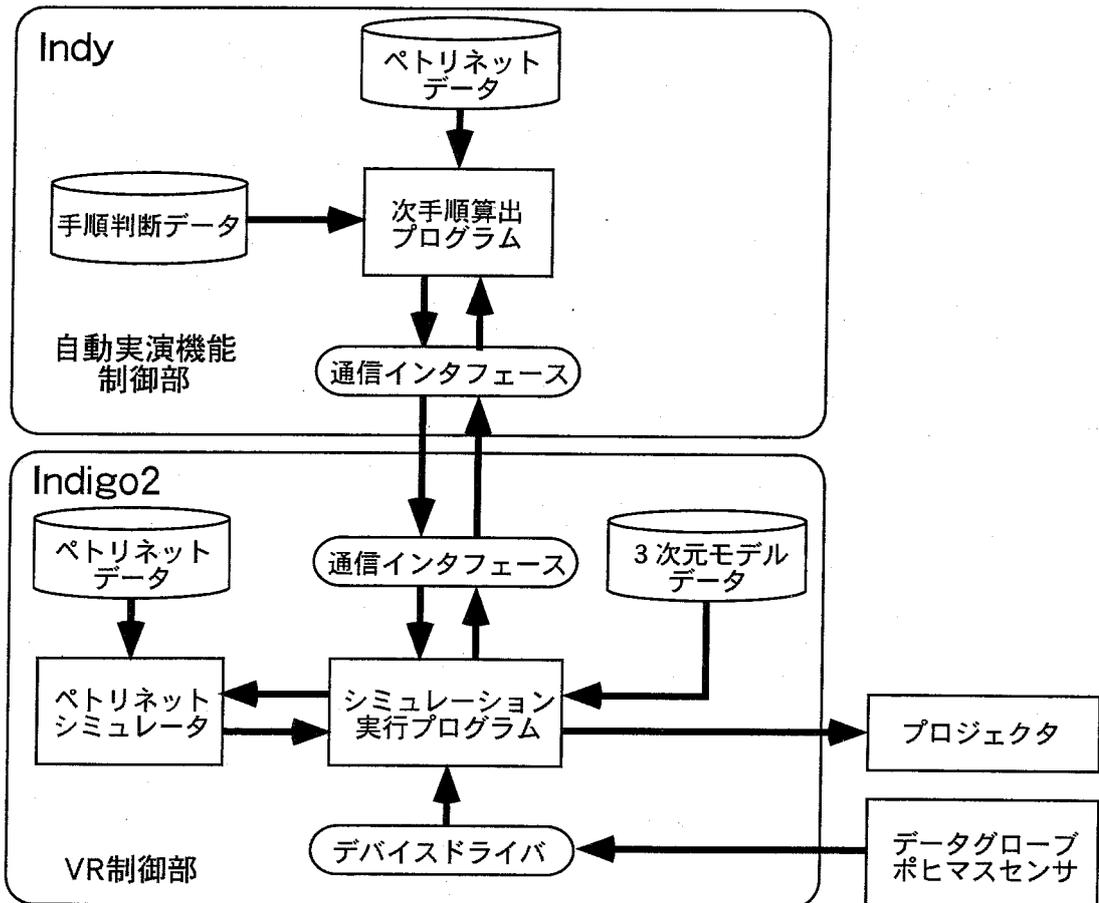


Fig. 3.22: VRMATE のソフトウェア構成

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

表示サブシステムは、視覚に対する表示デバイスとして、大型ディスプレイと立体視用の CrystalEyes を採用し、聴覚に対する表示デバイスとしてスピーカを採用した。訓練生は、CrystalEyes を用いて立体視しながら機器保修を行う仮想空間を眺める。現時点の VRMATE では仮想空間を表示する領域が大型ディスプレイの範囲内に限定されるために、訓練生が仮想空間に没入しにくい可能性があるが、今後、表示デバイスを HMD に変更することによって、訓練生がどの方向を向いても常に眼前に仮想空間が表示されるようにすることも容易に可能である。

シミュレーションサブシステムは、Fig. 3.22に示したように、仮想環境を生成するための VR 制御部と、自動実演機能を提供するための自動実演機能制御部から構成され、計算負荷を分散させるために、それぞれ別の計算機上に実装し、ソケット通信により必要な情報をやりとりする。

VR 制御部は、ペトリネットデータ、3次元モデルデータ、ペトリネットシミュレータ、シミュレーション実行プログラムから構成され、通信インタフェースを介して、自動実演機能制御部と必要な情報をやりとりする。ペトリネットデータとしては、3.5節で述べた PCSS を用いて作成したペトリネットを格納し、3次元モデルデータとしては、市販の CAD(Computer Aided Design) を用いて作成した仮想物体の表面の形状情報をポリゴンとして格納する。また、ペトリネットシミュレータは C 言語を用いて構築され、VRMATE の起動時にペトリネットデータから情報を読み込み、シミュレーション実行プログラムからの指示により、ペトリネットのシミュレーションを行う。シミュレーション実行プログラムは、3次元画像処理用 C 言語ライブラリである World Tool Kit(Sense8 社製)を用いて作成した。World Tool Kit(WTK)は、データグローブやポヒマスセンサからの入力を処理するデバイスドライバを提供し、仮想空間を Zバッファ法で画面に描く機能等を提供する。

自動実演機能制御部は、ペトリネットデータ、手順判断データ、次手順算出プログラムから構成される。次手順算出プログラムは 3.6.2項で述べたアルゴリズムを実行するプログラムで C 言語を用いて構築されており、VR 制御部からの指示により、訓練生の作業の監視と、次に発火すべきトランジションの決定を行う。すなわち、訓練生が自動実演を要求していない場合には、訓練生のジェスチャに対応して仮想物体の状態が変化し、ペトリネットのトランジションの発火やトークンの遷移に関する情報が次手順算出プログラムに送られ、Fig. 3.19に示したアルゴリズムが実行される。また、自動実演のモードに入ると、Fig. 3.20に示したアルゴリズムが実行され、次手順算出プログラムから発火すべきトランジション名が VR 制御部に順に送られる。VR 制御部では、送られたトランジションに対応するタスク関数が実行され、自動実演として 3次元映像が合成される。

以上の構成では、VR 制御部と自動実演機能制御部の両方にペトリネットデータが格納され、2重に処理されることになるが、これは、2つの制御部を単独実行させるためであり、これにより、制御部間の通信の量も減らすことができる。

3.7.2 VRMATE の使用例と評価

VRMATE の初期画面を Fig. 3.23に示す。ここで、画面の右下にある矢印を手で握って回転させると仮想空間が逆止弁を中心に回転し、様々な角度から逆止弁を眺めることがで

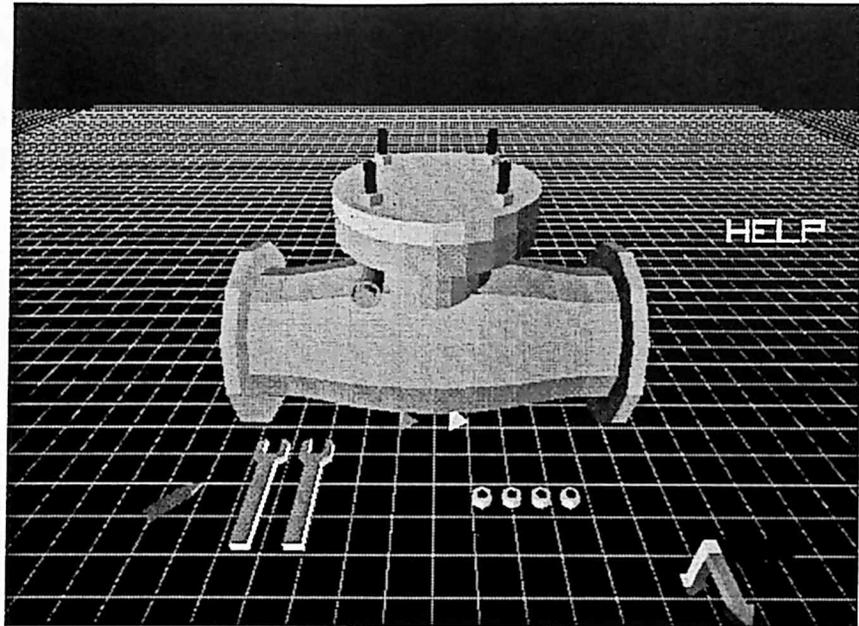


Fig. 3.23: VRMATE の初期画面

きる。また、自動実演機能は、訓練の任意の時点で Fig. 3.23の画面右側にある HELP オブジェクトを手で掴むことにより実行される。Fig. 3.24は、弁の中を覗くために、Fig. 3.23の状態から視点を横方向に移動させた時の映像であり、弁内部にある弁体を見ることができる。また、レンチでナットを緩める作業を行っている時の仮想空間の映像を Fig. 3.25に示す。さらに、機器必修訓練の補助機能として、逆止弁の各種情報を参照する機能を使用した際の画面を Fig. 3.26に示す。Fig. 3.26の左上の画面が逆止弁の各部名称を示す図、右上の画面が逆止弁の働きを説明する動画、右下の画面が逆止弁の写真、左下の画面が訓練用画面である。

また、研究室の大学院生数人を訓練生として、VRMATE を用いて機器必修の訓練を行った結果、得られた感想を以下にまとめる。

1. 間違っただけでも実施できることは、作業手順を学習する際には有効である。しかし、現時点での VRMATE では作業を間違っただけで警告が出されないため、間違えたまま作業を進めてしまう。効果的に学習するには間違えた時点で適切なアドバイスをを行う機能が必要であると思う。
2. 視点の移動が容易にできるため、対象物の構造が理解しやすい。
3. 自動実演機能は役に立つが、自動実演の実行速度を調節できる機能が欲しい。
4. 作業中は腕を上げなくてはならないため、長時間訓練を行うと非常に疲れた。

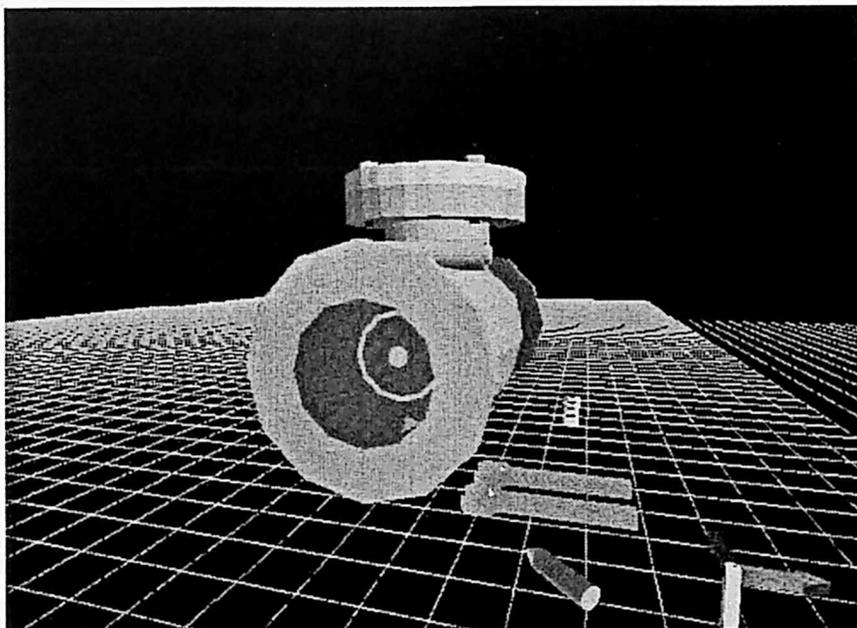


Fig. 3.24: 逆止弁を横から眺めた映像

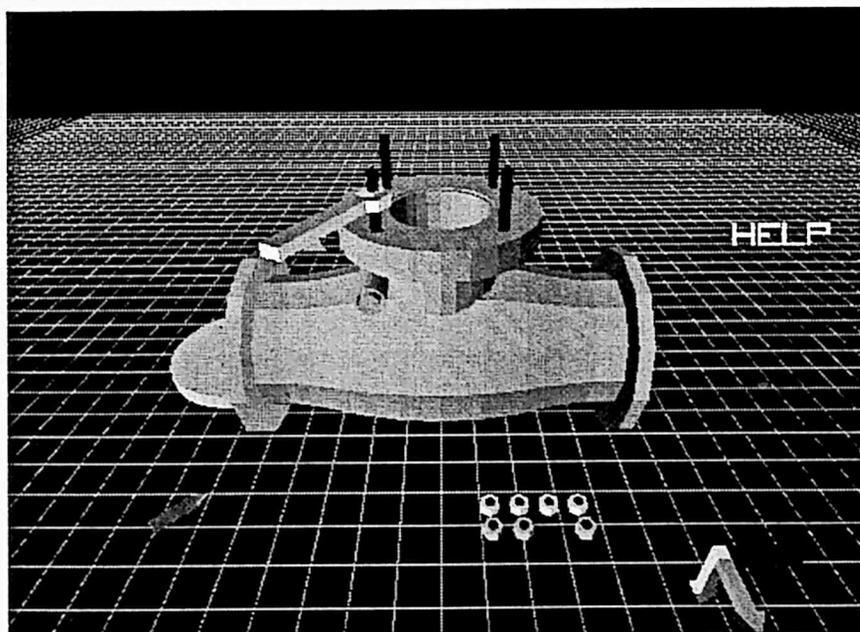


Fig. 3.25: レンチでナットを緩める作業中の仮想空間の映像

3.8 考察及び将来課題

第3者に試用してもらうことにより、VRMATE が、機器保修の作業手順を学習するのに有効であることが確認されたが、幾つかの問題点も明らかになった。以下にその問題点をまとめる。

1. 力覚フィードバックの機能を持たないデータグローブとポヒマスセンサを組み合わせた入力デバイスは、実際の作業に近いジェスチャで訓練が行えるために、訓練を行うためだけの特別な予備知識が少なく済むという利点があるが、VRMATE にジェスチャを認識させる作業自体は行いにくく、訓練生がすぐに疲れてしまうという問題が起きる。従って、今後、機器保修の訓練システムを開発する際には、たとえ機器の構造や作業手順等の知識の修得が目的であっても、ジェスチャを介して作業を行う場合には、力覚フィードバックを提供するデバイスを使用するか、もしくは何らかの方法で力覚フィードバックの欠如を補う機能が必要である。
2. 訓練生が作業手順を学習する際には、自らの判断で自由に作業ができることや自動実演機能が使用できることが効果的であるが、作業を間違えた場合に訓練システムの方から何らかの警告を出す機能も必要である。
3. VRMATE では、訓練生の緊張感を持続させることが難しく、作業に集中させるためのなんらかの監視役が必要である。これには、訓練を実施する際に訓練のインストラクタが付き添う方法も考えられるが、VRMATE が訓練生個人で使用できるという利点が失われる可能性がある。従って、今後は、VRMATE に、訓練生が行った作業に対して点数を付けて評価する機能等を構築し、集中力を持続させる対策を行う必要がある。
4. 今回開発した VRMATE は、訓練のシミュレーションを実行する際のシステムのフレームレート (毎秒あたりに画面を更新する回数) を向上させることに、それほど注目しなかったために、作業の内容によっては、訓練生が違和感を感じてしまうほどにフレームレートが落ちてしまうことがあった。今後は、訓練生のジェスチャの判定や仮想空間のシミュレーションを高速に行う方法を開発し、フレームレートを上げる必要がある。

また、VRMATE を構築する方法にも、幾つか問題点が明らかになった。以下にその問題点をまとめる。

1. 自動実演機能を提供するために予め準備すべき情報の量が非常に多く、大規模な機器保修の訓練に対して自動実演機能を提供することが困難である。また、今回考案した自動実演のためのアルゴリズムは、3.3節で述べたスイング式逆止弁の分解作業に対しては適用できたが、他の機器保修の作業に対して適用できることが証明されていない。今後、自動実演機能を実用化するためには、広い範囲の機器の保修作業に適用できる自動実演のアルゴリズムを開発する必要がある。

3.8 考察及び将来課題

これに対し、ペトリネットのトークン配置を用いて自動実演機能を実現する方法が有望であると考えられる。すなわち、予め正しい手順で作業を行った場合のトークン配置の変化をデータベース化する。そして、自動実演機能が使用された時には、その時のペトリネットのトークン配置を調べ、データベースの中に同じトークン配置が存在する場合には、訓練生は正しく作業を行っているとは判断し、データベースに従って、次に発火させるべきトランジションを決定する。逆に、データベースの中に同じトークン配置が存在しない場合は、訓練生は間違った作業を行っているとは判断し、データベースに存在するトークン配置の内、最も近いトークン配置を選択し、そのトークン配置になるまでのトランジションの発火系列を検索し、次に発火させるべきトランジションを決定するという方法である。この方法では、殆ど全ての機器保守の作業に対して適用できると考えられるが、以下のような問題がある。

- (a) 機器保守の正しい作業手順は必ずしも一通りではない。

機器保守の正しい作業手順は必ずしも一通りではなく、状況によっては同じ優先順位の作業が多数存在する場合がある。例えば、ナットを外すべきボルトが2本あって、その両方に同じようにナットがはまっている場合、どちらのナットを先に外してもよい。従って、予め正しい手順で作業を行った場合のトークン配置の変化をデータベース化する場合、数多くの手順に対するデータベースを作成するか、複数手順が存在する場合にも対応できる新たなデータベース作成方法を考案する必要がある。しかし、前者の方法の場合、データベースが非常に巨大化するという問題がある。

- (b) ペトリネットが巨大になった場合に、発火系列を検索するには膨大な計算が必要になる。

ペトリネットの発火系列の検索法としては、(1) 接続行列を用いて行列方程式を解くことにより各トランジションの発火総数を求める方法 [6]、(2) 可達木を用いて全ての場合について調べる方法 [7, 8] 等が提案されている。しかし、(1)の方法では、求めることのできる解は、目的のトークン配置に至るまでの各トランジションの発火回数の総数だけであり、どのトランジションがどの順番で発火するのかについての情報は得ることができない。一方(2)の方法では、ペトリネットが並行動作可能な対象をモデル化している場合、発火系列の検索に、プレイスとトランジションの総数の指数関数に比例した時間と記憶領域が必要であることが分かっており [9] 巨大なペトリネットに対してそのまま適用するのは、非常に困難である。また、検索の結果、複数の解が得られた場合に、どの解が最も適当な解なのかを判断する方法を考案する必要がある。

従って、今後ペトリネットのトークン配置を用いて自動実演機能を実現する場合、(1) ペトリネットを予め幾つかの小規模なペトリネットに分割する、(2) 発火系列を検索する際に冗長となる部分を予め排除し、発火系列の検索効率を上げる等の対策が必要であると考えられる。

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

2. VRMATE を開発する際には、仮想物体の状態遷移の仕方を定めるペトリネットを構築し、さらに訓練生のジェスチャに応じて仮想物体の運動の仕方を制御するタスク関数を作成する必要がある。ペトリネットの構築に関しては、PCSS を用いることにより、大規模なペトリネットでも、少ない労力で構築可能であると予想されるが、タスク関数を作成する際には、C 言語や WTK に関する高度な知識が必要であり、かなりの労力も必要である。従って、ペトリネットを構築する支援システムだけでなく、タスク関数に相当する機能を構築するための支援システムも今後必要である。

また本研究では、機器の構造や作業手順を教育の対象として VRMATE を開発したが、原子力発電プラントの保修作業員が修得すべき知識や技術は他にも数多く存在する。従って、今後、VRMATE で学習可能な項目を増やしていく必要があるが、その際、訓練生のレベルに応じた適切な訓練のシナリオを設定することは重要である。例えば、VRMATE を用いた機器保修訓練のシナリオとしては、以下のようなものが考えられる。

シナリオ 1 保修作業の訓練を初めて受ける訓練生に対して実施する訓練のシナリオ。機器の構造や各部の機能を理解する。

シナリオ 2 保修作業の初心者に対して実施する訓練のシナリオ。機器の分解、点検、組立の適切な作業手順を修得する。

シナリオ 3 実際の現場で働く作業員を対象に実施する訓練のシナリオ。機器に異常が発生した場合の異常箇所の特定制や故障した機器を修理する技術を修得する。

以上のようなシナリオで機器保修の訓練を実施する場合、シナリオ 1 とシナリオ 2 の訓練を実施するための訓練環境を構築する方法は、本研究で提案したことになる。しかし、シナリオ 3 の訓練を実施するためには、まず、異常が発生した場合の機器の挙動を模擬する必要がある。また、異常箇所の特定制や故障した機器の修理作業を仮想空間内で実施することを可能にするためには、本研究で開発した機器保修の訓練環境に比べて、さらに複雑な訓練環境を構築する必要がある。従って、上記のようなシナリオによる訓練を実現する為には、複雑な訓練環境を容易に構築できるようにする構築環境の整備が必要である。

3.9 まとめ

本研究では、機器保修の訓練環境を構築する新たな手法として、仮想物体の状態遷移をペトリネットを用いてモデル化する手法と、モデル化したペトリネットを用いて仮想物体の運動を制御する方法を提案した。また、大規模なペトリネットを少ない労力で構築可能にするためのペトリネット作成支援システムを開発した。さらに、機器保修の自動実演機能を実現するために、「仕事」の概念を用いた新しいアルゴリズムを考案した。そして、提案した手法を用いて、スイング式逆止弁の分解作業を実際の作業に近い自然な動作で行うことができる機器保修の訓練システム VRMATE を開発した。

3.9 まとめ

まず、訓練生が効率的に保守作業に必要な知識を修得できるようにするために、訓練システムが具備すべき機能を検討し、予備知識がなくても容易に使用できるインタフェースを備えること、訓練生が自らの判断で自由に作業を行うことができること、自由な手順で作業が実行できる訓練環境では自動実演機能が必要であること、訓練生が機器の構造を容易に理解できるようにするために視点を自由に変更できる必要があること等を述べた。

本研究では、VRMATE を構築する際の具体的な訓練対象として、原子力発電プラントの構成機器の1つであるスイング式逆止弁の分解作業を選択した。そして、訓練生のジェスチャに応じた仮想物体の運動の仕方を容易に設定できるようにするために、仮想空間内に物体の状態の概念を導入し、「仮想物体が取りうる状態」と「仮想物体の状態遷移の仕方」の2つの情報をペトリネットを用いてモデル化する本研究独自の手法を提案した。その際、機器保守の訓練環境をモデル化するのに適した本研究独自のペトリネットを定義し、訓練生のジェスチャに応じて仮想物体の運動を制御するタスク関数と組み合わせて用いることにより、仮想空間をシミュレーションする方法について述べた。

次に、ペトリネットを GUI を介して構築できるペトリネット作成支援システム (PCSS) を開発した。その際、機器保守の作業を対象にモデル化したペトリネットには、同じ構造をしたペトリネットを多数組み合わせた部分が存在する場合には、同じ構造をしたペトリネットを多数組み合わせた部分が存在する場合には、頻繁に使用されるペトリネットをテンプレートとして登録し、後に必要に応じて修正して使用できる機能を構築した。そして、開発した PCSS を用いて、実際にスイング式逆止弁の分解作業を対象にしたペトリネットを作成し、PCSS を用いることにより、非常に少ない労力で大規模なペトリネットを構築できることを確認した。

また、訓練生の要求に応じて次に実行すべき機器保守の作業を3次元映像として合成する自動実演機能に関しては、本研究独自のアルゴリズムとして、「仕事」の概念を導入し、仕事間の優先順序に関する情報を用いることにより、訓練生の作業を監視するアルゴリズムと、次に発火すべきトランジションを決定するアルゴリズムの2つのアルゴリズムを考案した。そして、実際に VRMATE に導入することによりその動作を確認した。

開発した VRMATE を、研究室の数人の大学院生に試用してもらうことにより、自動実演機能を備えた機器保守の訓練システムは、保守作業の初心者が作業の手順を修得するのに非常に有効であること、仮想空間内で自由に視点を変えて、訓練の対象機器を観察できることは、機器の構造を理解するのに有効であること等を確認した。しかし、一方で、VRMATE で採用した、データグローブとポヒマスセンサを用いた計測デバイスは、力覚フィードバックの機能が無いため、作業が非常にに行いにくく、訓練生がすぐに疲れてしまうことが判明した。従って、今後、新たに機器保守の訓練システムを開発する場合には、例え、機器の構造や作業手順といった保守作業の知識を教育の対象とする場合でも、力覚フィードバックを提供できるデバイスを使用するか、もしくは何らかの方法で力覚フィードバックの欠如を補う必要がある。

本研究で提案した機器保守の訓練環境を構築する手法は、実際にある程度複雑な訓練環境を構築するために利用できることが確認されたが、今後、数多くの機器に対して訓練環境を構築するためには、さらに効率的な訓練環境の構築方法の確立が必要である。特に、今回、提案した訓練環境の構築方法では、訓練生のジェスチャに応じた仮想物体の運動を制御するためにタスク関数と呼ぶ C 言語の関数を作成する必要があり、訓練環境を構築す

3. 人工現実感技術を用いた機器保修の訓練システム VRMATE の開発

るためには、プログラミングに関する高度な知識が必要となる。今後、人工現実感技術を用いた訓練環境を実用化するためには、プログラミング等の高度な知識を持ち合わせていなくても訓練環境を容易に構築できるようにするための環境の整備が必要である。

以上の課題点を考慮して、次章では、プログラミングを全く行わなくても容易に機器保修の訓練環境を構築できる支援システムの開発を行う。

参考文献

- [1] 浅野理森. はじめての Tcl/Tk. 技術評論社, 1995.
- [2] 宮田重明, 芳賀敏彦. Tcl/Tk プログラミング入門. オーム社, 1995.
- [3] R. Fikes and N. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, Vol. 2, pp. 189–208, 1971.
- [4] D. Sacerdoti. Planning in a Hierarchy of Abstraction Spaces. *Artificial Intelligence*, Vol. 5, No. 2, pp. 115–135, 1974.
- [5] D. Sacerdoti. The Nonlinear Nature of Plans. *International Joint Conference on Artificial Intelligence*, pp. 206–214, 1975.
- [6] 包原考英, 稲葉昭夫, 川瀬俊夫, 鈴木達也, 大熊繁. ペトリネットを用いた組立作業計画. 計測自動制御学会論文集, Vol. 29, No. 4, pp. 461–469, 1992.
- [7] 椎塚久雄. 実例ペトリネット, 第6章. コロナ社, 1992.
- [8] 村田忠夫. ペトリネットの解析と応用, 第4章. 近代科学社, 1992.
- [9] R. Lipton. *The Reachability Problem Requires Exponential Space*. Department of Computer Science, Yale University, New Haven, 1976.

第 4 章 機器保守の訓練環境の構築支援システム DESCORTE の開発

4.1 研究の背景と目的

第 3 章では、スイング式逆止弁の分解作業を対象にした機器保守の訓練システム VRMATE の開発を行い、人工現実感技術を用いた機器保守の訓練システムは、保守作業の初心者が機器の構造や作業手順等を修得する際に効果的に使用できるとの結果を得た。その際、仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する手法を提案し、さらにペトリネットの作成作業を支援する PCSS を開発した。しかし、第 3 章で提案した訓練環境の構築方法では、訓練生のジェスチャの判定方法や、仮想物体の各状態に応じた運動の仕方を設定する際、計算機に関する専門的な知識、特にプログラミングに関する知識が必要であり、1 つの機器に対する訓練環境を構築する場合でも、多くの労力が必要であることも判明した。訓練生が保守技術を修得する必要がある機器は、数十種類に上るとされており、これら全ての機器に対する訓練環境を構築するには、多くの労力が必要となり、多くの時間と人件費が必要となる。さらに、訓練環境を構築する際にプログラミング等の専門知識が必要であれば、それらの知識を持っていない訓練センターのインストラクタは、自分で訓練環境を構築できない。従って、訓練環境の構築をプログラミング等の専門知識を持つエンジニアに依頼するしかなく、実際の現場の要求を反映した訓練環境を構築することが難しくなる。

そこで本章では、プログラミング等の専門知識を持たない人でも、容易に機器保守の訓練環境を構築できるようにするための構築支援システム (Design Support System for Constructing VR-based Machine-Maintenance Training Environment; DESCORTE) の開発を行う。具体的には、以下の 4 つの特徴を持つシステムを開発する。

1. プログラミングを全く行わなくても自由に機器保守の訓練環境を構築できる。
2. 訓練環境を構築するために必要な情報はすべて GUI[1] を介して短時間で入力できる。
3. 計算機やコンピュータグラフィックス、数学等に関する高度な知識を持たない人でも訓練環境を構築できる。
4. 訓練の実施環境も提供する。

1. の目標に対しては、実際の保守作業訓練の対象となっている複数の機器に対して、その機器の保守作業を行うための訓練環境を仮想空間内に構築する際に、どのような情報が必要となるかを分析し、その結果をもとに、プログラミング以外の方法で、訓練環境を構築

4.2 DESCORTE で構築可能な訓練環境

する手法を考案した。2.の目標に対しては、第3章で提案した、仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する手法を改良し、DESCORTE に応用して、視覚的にペトリネットを構築できるインタフェースを構築した。さらに、必要な情報を GUI を介して入力する環境を構築した後、DESCORTE のユーザインタフェースの評価、改良を繰り返すことにより、ユーザの作業時間の短縮を図った。3.の目標に対しては、DESCORTE のインタフェース設計において、ユーザが知らないであろうと思われる専門用語は極力使用しないように勤め、また、代わりとなる言葉が無い場合には、インタフェースにアイコンを使用することにより、絵を見て機能を推測できるようにした。

第2章で述べたように、GUI を介して仮想空間を構築する支援システムは幾つか開発されているが、いずれも仮想空間が複雑になった場合に、構築作業を支援するための特別な機能が備わっておらず、機器保守作業のような、複数の機器が関連した複雑な仮想空間を構築するには向いていない。これに対し、本研究で開発する DESCORTE では、仮想物体の状態遷移の仕方の記述に、視覚的な表現が可能なペトリネットを用いるため、対象が複雑な訓練環境であっても、比較的容易に構築できると期待される。

以下では、まず初めに DESCORTE で構築可能とすべき訓練環境について検討する。次に、訓練環境を構築するために作成する必要がある、状態遷移データベースと運動データベースについて述べた後、それらのデータを入力するための DESCORTE のインタフェースの設計について述べる。その後、開発した DESCORTE の概要と、DESCORTE の評価実験を行った結果について述べる。さらに、評価実験の結果をもとにユーザの認知的負荷を軽減させる方策を考案し、実際に DESCORTE のインタフェースの改良を行い、再び評価実験を行った結果を述べた後、最後に今後の課題と本研究のまとめを述べる。

4.2 DESCORTE で構築可能な訓練環境

4.2.1 DESCORTE で構築可能な訓練環境が具備すべき要件

DESCORTE の設計を行うためには、まず DESCORTE で構築可能な訓練環境の仕様を決定する必要がある。本研究では、VRMATE を開発した際に判明した結果をもとに、構築可能とする訓練環境の仕様を検討し、DESCORTE で構築できる訓練環境には以下の要件を具備させることにした。

1. 実際の作業に似たジェスチャで機器を操作できるインタフェースを備えること

3.2節でも述べたように、予備知識や技能を必要としない訓練システムを構築することは、訓練効率を上げるために重要である。そのためには、現実の作業に似たジェスチャで機器を操作できるインタフェースを備えた訓練システムを開発する必要がある。しかし、VRMATE で採用したデータグローブとポヒマスセンサを組み合わせた入力デバイスは、実際の作業に似たジェスチャで機器を操作できるインタフェースを提供するが、ジェスチャを行う際に訓練生が腕を空中に上げておく必要があり、また、力覚フィードバックを備えていないために作業が行いにくく、訓練生がすぐに疲労し

4. 機器保守の訓練環境の構築支援システム DESCORTE の開発

てしまうという問題があった。そこで DESCORTE では、計測の精度が高いこと、デバイスを使用する際に必要な場所が狭いこと、計測装置自体が安価であること、デバイスを机の上に置いて使用するため訓練生が腕を上げておく必要が無く比較的疲労しにくいこと等を考慮し、Logitech 社の 3次元マウス (Magellan) を用いることとした。

2. 視点の自由な移動が可能であること

VRMATE を評価した際、視点が自由に変更できる機能は、機器の構造を理解する上で有効であるとの結果を得た。従って、DESCORTE でも、3次元マウスを用いて視点を移動させ、機器の構造を観察したり、様々な視点から作業を行ったりする機能を備えた訓練環境を構築可能とする。

3. 訓練環境の臨場感が高いこと

訓練環境の臨場感を向上させる方法としては、使用する表示デバイスの性能を上げる方法と、構築する仮想空間の詳細度を上げる方法の2つの方法が考えられる。使用する表示デバイスの性能を上げる方法としては、力覚フィードバックを提供できるデバイスを採用することや、映像を提示するディスプレイの解像度を上げること等が考えられるが、いずれも実現するためには新たなハードウェアの開発や採用が必要である。一方、構築する仮想空間の詳細度を上げる方法としては、仮想物体にテクスチャを張ることや、効果音を提示すること等が考えられ、これらは時間と労力をかけることにより実現可能である。本研究では、容易に訓練環境の臨場感を向上できる方法として、DESCORTE に、仮想物体にテクスチャを張る機能や、訓練中に効果音を提示する機能を構築する。

4. 仮想空間内での作業を行い易くすること

VRMATE の評価を行った際、使用した表示デバイスに力覚フィードバックが無いために、仮想物体に触ったかどうか分かりにくいいため、保守作業が行いにくく、疲れやすいという結果が出た。しかし、第2章で述べたように、現時点での人工現実感技術では、訓練生の視覚・聴覚以外の感覚に対して情報提示を行うことは困難である。そこで、DESCORTE を開発する際には、この問題を解決するために、訓練生の視覚と聴覚に対して情報提示を行うことを基本として、(1) 奥行き知覚特性の改善、(2) 力覚フィードバックの欠如の補完、の2点に注目し、以下の2点の工夫を行うこととした。

(1) 仮想空間内における奥行き知覚を改善するための工夫

機器保守の作業は一般に3次元空間における機器の操作であるため、訓練生は常に訓練対象機器の各部品の前後関係を知覚する必要がある。訓練生に仮想物体の前後関係を知覚させるには、VRMATE で用いたように、CrystalEyes 等を用いて立体映像を提示する方法が一般的である。CrystalEyes を用いた3次元画像提示は、比較的容易に実現することができ、実空間と同様の感覚で物体の前後関係を知覚することができる利点がある。しかし、CrystalEyes で採用されて

いるような方式で立体視する場合、眼球の輻輳と焦点調節の不一致が生じ、訓練生の眼に生理的な負担がかかる [2]。従って、長時間の使用には向いておらず、また、この方式では立体視ができない人も存在するという問題がある [3]。そこで DESCORTE では仮想物体に影を付けることにより、仮想物体の前後関係を知覚させる方法を採用し、CrystalEyes は必要に応じて使用する、しない、を選択できるようにした。

(2) 力覚フィードバックの欠如の補完

DESCORTE で構築可能な訓練環境では、力覚フィードバックを考慮していないので、訓練生の作業を支援するために、何らかの工夫が必要である。そこで DESCORTE では、訓練中に訓練生が何らかのイベント（物体を触る、物体同士を接触させる等）を起こした場合、先に述べた効果音に加えて、必ずイベント音を提示することとした。力覚フィードバックが無くてもこのような補助的な工夫によって手で物を触った等の感覚を補完できると期待される。

本研究では、DESCORTE を用いることで、上記4つの項目を考慮した訓練環境を構築できるものとする。

4.2.2 訓練環境を構築する際に作成する必要がある情報

3.4節で述べたように、VRMATE では、仮想空間内で機器修理の作業を行えるようにするために、仮想物体の「状態」の概念を導入し、仮想物体が取りうる「状態」ごとに仮想物体の運動の仕方（他の物体との拘束関係等）を記述した。DESCORTE でも同様に、仮想物体の「状態」の概念を導入するが、VRMATE では、訓練生のジェスチャの判定方法や仮想物体の運動の仕方を記述するために、タスク関数をC言語を用いて作成したのに対し、DESCORTE では予め用意されたパラメータを変更することにより、プログラミングを行わずに訓練生のジェスチャの判定方法や仮想物体の運動の仕方を設定できる構築環境を提供する。具体的には、DESCORTE では、訓練環境を構築する際に、以下の2つのデータベースを作成する。

- 状態遷移データベース

状態遷移データベースは、イベントの発生に応じた仮想物体の状態遷移の仕方を規定するデータベースで、VRMATE を開発する際に、ペトリネットを用いてモデル化していた情報に、トランジションを発火させる原因となる訓練生のジェスチャ等のイベントの種類の情報を加えたものに相当する。DESCORTE でも、これらの情報をペトリネットを用いてモデル化する方法を採用し、視覚的にペトリネットを構築できる環境を構築する。今回は、ペトリネットに関して初心者であるユーザでも容易にペトリネットの定義を理解できるようにするためと、ペトリネットのモデル化能力を向上させるために、3.4.2項で述べたペトリネットの定義を改良する。改良したペトリネットの詳細は4.4節で述べる。

● 運動データベース

運動データベースは、仮想物体が取りうる各状態における外見と運動の仕方に関する情報で、DESCORTE では、これらの情報を GUI を介して入力できるインタフェースを構築する。これは、VRMATE を開発した際に、タスク関数として記述した情報にほぼ相当するが、トランジションを発火させる原因となるイベントに関する情報が含まれていない点と、DESCORTE では、これらの情報を入力する際に、設定可能なパラメータの種類が制限されるために、記述の自由度がタスク関数を用いる場合に比べて小さい点が異なる。運動データベースの詳細は 4.3 節で述べる。

4.3 仮想物体の外見と運動の仕方を定めるデータベース

機器保守の訓練環境を構築する際に、仮想物体の各状態における外見や運動の仕方を、予め用意されたパラメータを変更することにより設定する方式（以下、パラメータ設定方式と呼ぶ）を採用する場合、どのようなパラメータを設定可能とするかが、訓練環境を構築する際の作業効率や、構築できる訓練環境の柔軟性に大きく影響すると予想される。本研究では、このパラメータ設定方式を採用するにあたり、柔軟に訓練環境を構築できるようにするために、3.3 節で述べたスイング式逆止弁の分解組立作業、及び、同じく原子力発電プラントの構成機器である横置きポンプの分解組立作業を実施できる訓練環境を構築する際に、仮想空間で模擬する必要がある物体の運動と物体相互間の拘束関係を検討し、DESCORTE で設定可能とすべきパラメータの種類を決定した。このようにして DESCORTE で設定可能なパラメータの種類を決定することにより、保守作業員に対して訓練を行う必要がある全ての機器に対する訓練環境が構築可能になるわけではないが、訓練生が修得すべき基本的な作業が実施可能な訓練環境を構築する場合に必要なパラメータの種類は抽出可能であると思われる。

4.3.1 仮想空間で模擬する必要がある物体の運動と物体相互間の拘束関係

以下に、DESCORTE で少なくとも模擬する必要があると考えられる物体の運動と物体相互間の拘束関係を述べる。

1. 訓練生が手に物体を持っている場合

この場合、手は訓練生のジェスチャに従って自由に運動することができ、握られている物体は、訓練生の手の動きに追随する。つまり、訓練生が手を平行移動させると、握られている物体も同じ速さ、方向で平行移動し、訓練生が手を回転させると、握られている物体も、手を中心にして回転する。従って、仮想空間内でこれらの運動を模擬するためには、訓練生の手の動きに応じて、平行移動と回転を同時に行うように、仮想物体の運動を設定できる必要がある。

4.3 仮想物体の外見と運動の仕方を定めるデータベース

2. 物体が自由落下する場合

訓練生が握っていた物体を手放した場合、物体は自由落下（等加速度運動）する。特に、機器保守の作業では、工具を取り上げ、使用後に再び工具を置くという作業が行われるため、訓練環境でも、仮想物体が自由落下するように設定できる必要がある。しかし、仮想空間内ではこの運動は必ずしも等加速度である必要はない。それは、例えば、等速運動であっても、真下に向かって運動していれば、物体が落下していると知覚することができるからである。以上から、仮想物体の運動の仕方として、等加速度運動もしくは等速度運動するように設定できる必要がある。

3. スパナでボルトにはまったナットを回転させる場合

この場合、訓練生の手、スパナ、ナットは、全て、ナットがはまっているボルトを中心軸とした回転運動をし、さらに回転の方向に応じた並進運動も行う。従って、仮想空間内でこれらの運動を模擬するためには、同時に複数の物体が、ある一つの軸を中心にした回転運動を行うように設定できる必要がある。また、ボルトとナットのように、回転角度に応じて並進運動する場合を模擬するためには、回転角度をもとに、平行移動量を設定できる必要もある。

4. ボルトにはまったふたを外す場合

複数のボルトにふたがはまっていて、そのふたを手で外す場合、このふたと訓練生の手は、ボルトの軸方向に平行な運動に限られる。従って、このような物体相互間の拘束関係を模擬するためには、物体の運動が、他の物体の特定の軸に沿った平行移動に制限できる必要がある。

5. モータの回転を模擬する場合

モータの回転等は、訓練生の手動きに関係なく、常にある一つの軸を中心にした回転運動である。従って、この運動を、仮想空間内に再現するためには、時間の推移に従って、物体の回転角度が変化するように設定できる必要がある。

4.3.2 DESCORTE で設定可能なパラメータ

本項では、4.3.1項での検討をもとに決定した、DESCORTE で設定可能とするパラメータの種類（運動データベースの内容）とそのパラメータを設定可能にすることにより、構築可能となる仮想環境について述べる。設定可能なパラメータは、仮想物体の外見に関する情報と、運動の仕方に関する情報に大別することができる。

物体の外見に関する情報

物体の外見に関しては、3次元形状データ、テクスチャデータ、描画方法の3種類のパラメータを設定可能とした。

- 3次元形状データ

物体の表面形状を格納したファイル名を指定する。この情報を仮想物体が取りうる状態ごとに設定できるようにすることにより、例えば「手を握っている状態から手を開く」といった、物体の形状に変化が伴う動作を、データを差し替えることにより表現できる。

- テクスチャデータ

物体の表面に張り付ける画像ファイル名を指定する。これにより、臨場感のある仮想空間を構築することが可能になる。また、手に触られている物体のテクスチャを、特別な色に設定する等により、訓練生に対する力覚フィードバックの欠如を補完する等の工夫を行うことが可能になる。

- 描画方法

仮想物体が取りうる状態ごとに設定できる仮想物体の描画方法として、物体の表面を表す面を塗りつぶして描画する方法（ポリゴン）、物体の表面を線だけで描画する方法（ワイヤフレーム）、何も描かない（透明）の3つを選択できるようにした。これにより、訓練生に機器の内部の構造を示すことも可能になる。

物体の運動の仕方に関する情報

仮想物体の運動の仕方（位置と角度の変化）は、時間の推移に応じて変化する場合と、3次元マウスからの入力に応じて変化する場合のどちらかを選択できるものとした。以下ではそのそれぞれについて述べる。

- 時間の推移に応じて変化する場合

この場合は、仮想物体の位置と角度を時間の関数として設定できるものとした。例えば、モータの回転は、角度の1つのパラメータのみを時間の1次関数で表現し、他のパラメータは変化しないように設定することで実現される。同様に、物体の自由落下も表現できる。具体的には、DESCORTEのユーザは、予め与えられた式の係数に数値を入力することにより、仮想物体の時間の推移に応じた運動の仕方を設定する。

- 3次元マウスからの入力に応じて変化する場合

3次元マウスからの入力は、訓練生の視点とその方向をもとに定められる座標系を基準にした、x軸、y軸、z軸の平行移動変位 (x, y, z) と、それぞれの軸を中心とした回転変位 (a, b, c) である。ユーザはこれらの入力と仮想物体の位置と角度を表すパラメータとの関係を設定することにより、訓練生のジェスチャに応じた仮想物体の運動の仕方を設定する。例えば、時刻 t における仮想物体の位置を (X_t, Y_t, Z_t) 、角度を (A_t, B_t, C_t) とすると、ボルトにはまり、手に握られているナットの運動の仕方を表現する場合は、式(4.1)のようになる。

$$\begin{cases} X_{t+1} = X_t & A_{t+1} = A_t + a \\ Y_{t+1} = Y_t + ha & B_{t+1} = B_t \\ Z_{t+1} = Z_t & C_{t+1} = C_t \end{cases} \quad (4.1)$$

ただし、 a の値は、ボルトの軸方向を中心にした回転角度を表し、 h は設定可能な定数である。訓練環境を構築する場合は、上記の関係を、項目の選択と数値の入力で行う。本システムではさらに、物体の可動範囲を数値で設定することにより、接触の判定を行わなくても、2つの物体が仮想空間内で重なり合わないようにすることもできる。

- 物体の各状態における初期位置

仮想空間内で物体の状態が変化する際に、物体の位置や角度をある決まった値に変更したい場合が多々ある。例えば、ペンを手で握る場合、ペンの先を下にして握るのが自然である。また、スパナやペンチも握る箇所が決まっている。このような場合に、ペンが手に触られている状態から手で握られている状態に遷移するとき、ペンの位置や角度が、予め定められた位置や角度に必ず変化するように設定できると便利である。そこで DESCORTE では、状態遷移が起こったときに、位置や角度が変化する場合と変化しない場合を選択できるようにし、さらに位置や角度が変化する場合は、座標を指定する基準となる他の物体名と、それを基準とした位置と角度を数値で設定できるようにした。

状態テンプレート

訓練環境の構築の自由度を高めるために、入力可能なパラメータを数多く準備すると、パラメータの設定作業が煩雑になり、かえって訓練環境を構築する上での作業効率を低下させる可能性がある。そこで、仮想物体の状態には、よく似た運動の仕方を設定すべき場合が多数存在することに注目し、よく使用するパラメータの値の組み合わせを、「状態テンプレート」として登録し、これも GUI を介して選択できるようにすることにより、訓練環境を構築する際の労力を削減する試みを行った。

4.4 仮想物体の状態遷移の仕方を定めるデータベース

4.4.1 改良ペトリネットの定義

DESCORTE で用いるペトリネットは、3.4.2項で述べたペトリネットの定義を基本としているが、モデル化能力の向上や、理解のしやすさの向上を目指し、定義を一部変更した。以下では、その定義の変更部分のみを述べる。Fig. 4.1に改良ペトリネットのグラフでの表記法の一覧を示す。

| 用語 | グラフ表現 | 用語 | グラフ表現 |
|-----------------|---|-----------------|---|
| シングルプレース |  | プールプレース |  |
| ノーマル トランジション |  | 自動発火 トランジション |  |
| ノーマルアーク |  | 抑止アーク |  |
| 参照アーク |  | トークン |  |

Fig. 4.1: 改良ペトリネットのグラフ表現

- 制御プレースの廃止

VRMATE では、ペトリネットによるモデル化を効率的に行うために、制御プレースを導入していた。特に、ペトリネットの冗長性を避けるために、手の状態を制御プレースを用いて表していた。そのため「仮想物体 1 つがトークン 1 つに対応し、プレース 1 つが仮想物体の 1 つの状態を表す」という規則が手に対しては守られていなかった。例えば、「ナットが手に握られている」状態を表すプレースは、ナットの状態として「手に握られている」状態と、手の状態として「ナットを握っている」状態の 2 つの仮想物体に対する状態を表していた。この場合、仮想物体の運動の仕方を定めるタスク関数は C 言語を用いて構築されていたため、「ナットが手に握られている」状態を表すプレースにトークンが配置されているという情報をもとに、手とナットの 2 つの運動の仕方を設定することが可能であった。しかし、DESCORTE では、「1 つのプレースが 1 つの仮想物体の状態を表し、1 つのプレースに対して 1 つの仮想物体の運動データベースを定める」ように統一した方がユーザが理解しやすいと考え、制御プレースを廃止し、手に関しても 1 つの状態に対し、1 つのプレースを設けるものとした。例えば、VRMATE の定義に従った Fig. 4.2 に示すペトリネットでは、プレース p1 には運動データベースを 1 つ設定する必要があるのに対し、プレース p3 には運動データベースを、手に対するものとペンに対するものの、2 つ設定する必要がある。一方、改良ペトリネットの定義に従った Fig. 4.3 に示すペトリネットでは、全てのプレースに対し、1 つの運動データベースを設定すればよく、改良ペトリネットの方が理解しやすいと思われる。

- 抑止アークの導入

1 つの箱の中にナット A とナット B が複数個あり、ナット A を全て取り出して初めてナット B が取り出せる状態は、ペトリネットにプレースのゼロテストを行う能力

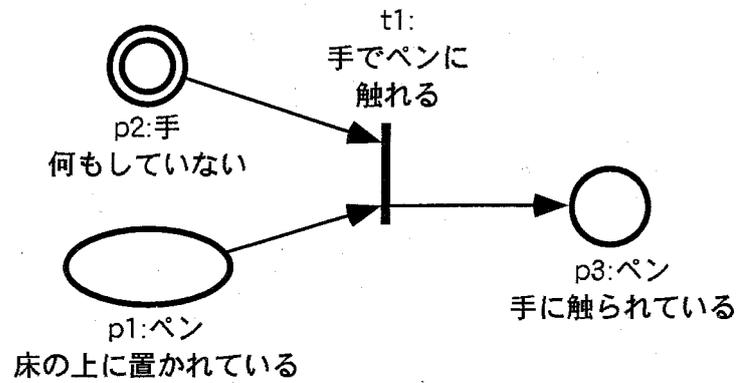


Fig. 4.2: 旧定義に従ったペトリネット

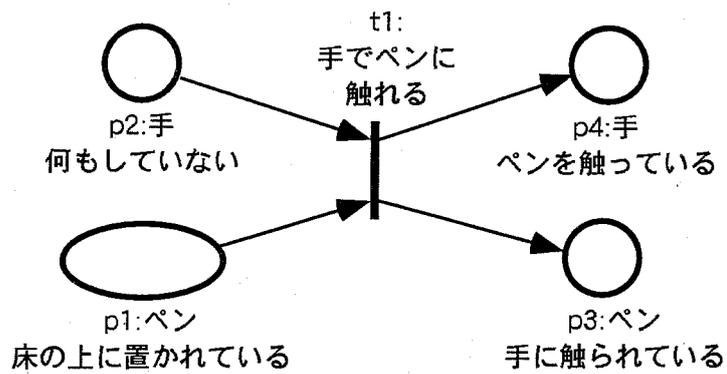


Fig. 4.3: 新定義に従った改良ペトリネット

が無ければ、モデル化できない [4]。これに対するペトリネットの拡張は幾つか提案されているが、本研究では、その概念の容易さを考慮し、新たに抑止アークを導入することとした。グラフでは Fig. 4.1 に示すように、プレースからトランジションへ、矢印の代わりに小円でトランジションにおいて終端する直線を引くことにより表現する。このアークを用いることにより、モデル化できる範囲が広がると同時に、ある物体の状態が、他の物体の状態遷移を抑止する場合（例えば「ナットとボルトでふたを固定する場合」）等をモデル化することが容易となる。

- 参照プレースと参照アーク

機器保守作業を対象に仮想物体の状態遷移をモデル化する場合、あるイベントが発生するための前提条件でありながら、そのイベントが発生しても、それ自身の状態は変化しない場合がある。例えば、ペンで機器に印を付ける場合、印を付けるためには、訓練生がペンを握っていなければならないが、印を付けた後も、訓練生は手を開かない限り、ペンを握ったままである。このような操作をモデル化すると、Fig. 4.4 に示すように1つのトランジション t_1 に対して、同じシングルプレース p_2 が、入力と出力の両方に配置される。この場合、3.4.2項で述べたペトリネットの定義に従えば、トランジションは発火可能であるが、ペトリネットをグラフで書いた場合、出力側のシングルプレースにトークンが存在するため、ユーザが、そのトランジションが発火不可能であると間違えて判断する可能性がある。そこで、本研究では、入力側と出力側に同じプレースが配置される場合、入力側にだけプレースを配置し、Fig. 4.1 に示すように、プレースとトランジションの間を双方向の矢印で結ぶことにより表現する。DESCORTE では特に、このプレースを参照プレースと呼び、また、双方向の矢印を参照アークと呼ぶ。これにより、ユーザは、「入力側のプレースにトークンが存在し、出力側のシングルプレースにトークンが存在しない場合にトランジションは発火可能である」と容易に判断できると思われる。

- トランジションの入出力アークの総数

VRMATE では制御プレースを用いていたため、トランジションの入出力ノーマルアークの総数は必ずしも同じではなかった。DESCORTE ではトランジションの入力ノーマルアークの総数と出力ノーマルアークの総数は必ず等しくなる。

4.4.2 改良ペトリネットによるモデル化手法

改良ペトリネットによる機器保守の作業のモデル化方法は、3.4.3項で述べた方法と基本的には同じである。以下では3.4.3項で述べた方法と異なる点のみを述べる。

変更点 1 3.4.3項では、手に対応する仮想物体については制御プレースを用いてモデル化していたが、改良ペトリネットではシングルプレースでモデル化する。

4.5 DESCORTE の開発

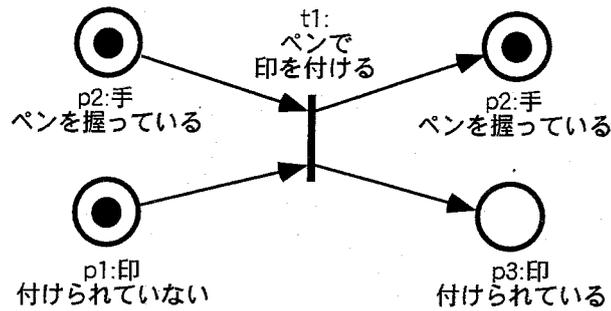


Fig. 4.4: ペンで印を付ける作業をモデル化したペトリネット

Table 4.1: トランジションを発火させる原因として選択可能なイベント

| | イベント | 例 |
|---|----------------|------------|
| 1 | 「物1」と「物2」が接触する | スパナと床が接触する |
| 2 | 「物1」と「物2」が分離する | 右手がペンから離れる |
| 3 | 手を開く | 右手を開く |
| 4 | 手を閉じる | 左手を閉じる |
| 5 | 条件が成立する | ペンを握っている |

変更点 2 3.4.3項では、トランジションとタスク関数が1対1で対応付けられていたが、改良ペトリネットでは、トランジションと、それを発火させるイベントが1対1で対応付けられる。DESCORTEでは、Table 4.1に示すイベントをトランジションの発火の原因となるイベントとして選択可能とする。

変更点 3 3.4.3項では、タスク関数内に効果音を発するための記述を組みこんでいたが、DESCORTEでは、トランジション1つに対し、効果音を格納するファイル1つを対応付け、トランジション発火時に、効果音を発生させるものとする。

変更点 4 訓練開始時の仮想物体の仮想空間内での位置は、VRMATEでは、タスク関数内に格納されていたが、DESCORTEでは、ペトリネットの初期トークン配置における各トークンに対し、仮想空間内での3次元位置を記述する。

4.5 DESCORTE の開発

本節では、本研究で開発したDESCORTEの構成と訓練環境を構築する際に使用するインタフェースについて述べた後、前節で述べた2つのデータベースを用いて実際に訓練を実施する際の情報の処理方法について述べる。

4.5.1 DESCORTE の構成

本研究で開発した DESCORTE は、訓練環境を構築するモード（構築モード）と訓練を実施するモード（訓練モード）が存在する。訓練環境を構築するには必要に応じてこれら 2つのモードを切り替えることにより、仮想空間の状況を確認しながら作業を行う。構築モードで使用するハードウェア、ソフトウェアと、訓練モードで使用するハードウェア、ソフトウェアが若干異なるため、以下ではそのそれぞれについて説明する。

構築モード

訓練環境を構築する場合に使用するシステムの構成を Fig. 4.5に示す。システムは、計測部、情報処理部、情報提示部に分けることができる。計測部は 3次元マウスと 2次元マウス、キーボードで構成され、情報提示部はディスプレイとスピーカで構成される。情報処理部は SGI 社のワークステーション（Indigo2、Maximum IMPACT）で構成され、ソフトウェアとしては、OpenGL ライブラリ [5] と Motif ライブラリ [6] を使用した。OpenGL ライブラリは、3次元画像処理のために開発された C 言語ライブラリであり、Z-バッファ法によるレンダリング環境を提供する。Motif ライブラリは OSF (Open Software Foundation) が提唱している X ウィンドウ上の高レベル GUI を提供するためのライブラリであり、様々な「部品」（ポップアップメニューやプッシュボタンなど）を作成するためのライブラリコールからなる。ユーザは OpenGL ライブラリを用いてレンダリングされた仮想空間をディスプレイを介して眺めながら、3次元マウスと 2次元マウスを操作し必要な情報を入力していく。2次元マウスとキーボードは、主に情報の選択や、数値や名称の入力に用い、3次元マウスは、主に仮想空間内の 3次元位置を指定する場合に用いられる。また、3次元マウスを用いることにより、作業中の視点を自由に変更でき、訓練環境を構築する際に、様々な角度から訓練対象機器の状態を確認しながら構築作業を進めることができる。GUI 環境の詳細は 4.5.2項で述べる。

訓練モード

訓練を実行する場合のシステム構成を Fig. 4.6に示す。訓練を実行する場合のシステムは、一般の人工現実感を実現するシステムと同様、計測サブシステム、シミュレーションサブシステム、表示サブシステムの 3つのサブシステムから構成される。計測サブシステムは 3次元マウスで構成され、訓練生は、3次元マウスのパックとボタンを操作することによって、手の把持・開放・移動等のジェスチャに対応する動作を行う。シミュレーションサブシステムは、ハードウェアとして SGI 社のワークステーション、ソフトウェアとして OpenGL ライブラリを用いた。表示サブシステムは、大型ディスプレイと CrystalEyes、スピーカで構成され、訓練生は、OpenGL ライブラリによって描かれた 3次元画像を CrystalEyes を通して立体映像として見ながら訓練を行うことができる。また HMD を用いて立体視しながら訓練することも可能である。

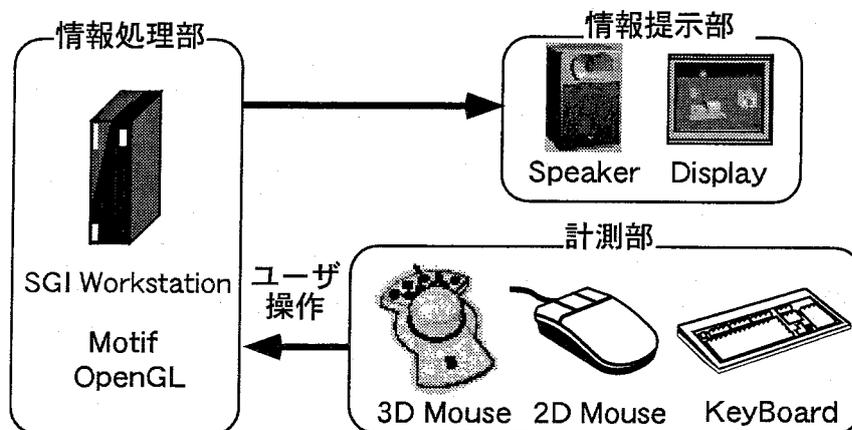


Fig. 4.5: システム構成 (構築モード)

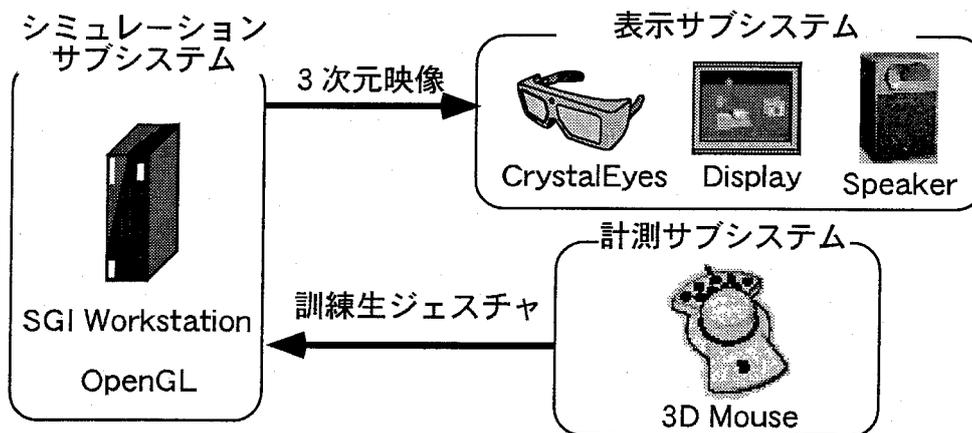


Fig. 4.6: システム構成 (訓練モード)

4.5.2 DESCORTE のインタフェース設計

DESCORTE で訓練環境を構築する際に使用するインタフェースを設計する場合、4.3節及び 4.4節で述べたデータベースを、如何に短時間で容易に入力可能にするかが問題となる。今回、DESCORTE を使用するユーザとしては、訓練センターで機器保守作業の指導を行っているインストラクタを対象としているため、3次元アニメーションやプログラミング等に関する専門知識を持ち合わせていないことを考慮せねばならない。しかし、一方では、最近では計算機の小型化、低価格化が進み、一般家庭にもパーソナルコンピュータ（パソコン）が導入されるようになり、それらの標準的なデータ入力デバイスである、2次元マウスやキーボードを不自由なく使用できる人が増えてきている。さらに、それらのパソコンで標準に使用されているオペレーティングシステムである、Windows や MacOS は、ウインドウ、アイコン、メニュー、ポインティングデバイスの4つの要素からなる「WIMP インタフェース」が標準的な環境である [7]。WIMP インタフェースは 1973 年にゼロックスのパロアルト研究所で作られた Alto マシンが原型になったものであるが、現在のアプリケーションでもほぼ同じ手法や装置が使用されており、これらの操作には慣れている人が多いと思われる。従って、本研究で開発する DESCORTE でも、WIMP インタフェースを介して訓練環境を構築することを可能にすれば、インストラクタが普段使用しているインタフェースとの間に一貫性が生まれ、初めて DESCORTE を使用する場合でも、新たに修得すべき知識や技能を少なくすることができ、訓練環境の構築作業の効率を上げることができると予想される。以上の理由により、DESCORTE では、WIMP インタフェースを基礎にして、運動データベース及び状態遷移データベースを入力、修正するためのインタフェースを設計した。

以下では、4.3節及び 4.4節で述べたデータベースを DESCORTE に入力する手順とその際のインタフェースについて説明する。

DESCORTE の構築モードの画面例を Fig. 4.7に示す。DESCORTE では、Fig. 4.8に示すようなアイコンを 2次元マウスを用いてクリックすることによって、データベースを入力するための様々なウインドウを表示することができる。入力の手順は大きく次の 4段階に分けることができ、全ての段階で、GUI を介して必要な情報を入力できる。

1. 訓練に必要な仮想物体を登録する。

DESCORTE で訓練環境を構築する場合、訓練に用いる物体を最初に登録する。例えば、「右手を用いてペンでふたに印を付ける」訓練環境（以下、印を付ける訓練環境）を構築する場合、Table 4.2に示す 4つの物体を Fig. 4.9に示すインタフェースを介して、キーボードを用いて登録する。

2. 仮想物体が取りうる状態を登録し、運動データベースを作成する。

前段階で登録した仮想物体が取りうる状態を登録する。この作業により、ペトリネットにおけるプレースが自動的に作成される。例えば「印を付ける訓練環境」を構築する場合、ペンが取りうる状態として、Table 4.3に示すような 5つの状態を登録する。そして、それぞれの状態について、Fig. 4.10～Fig. 4.13に示すようなインタフェース

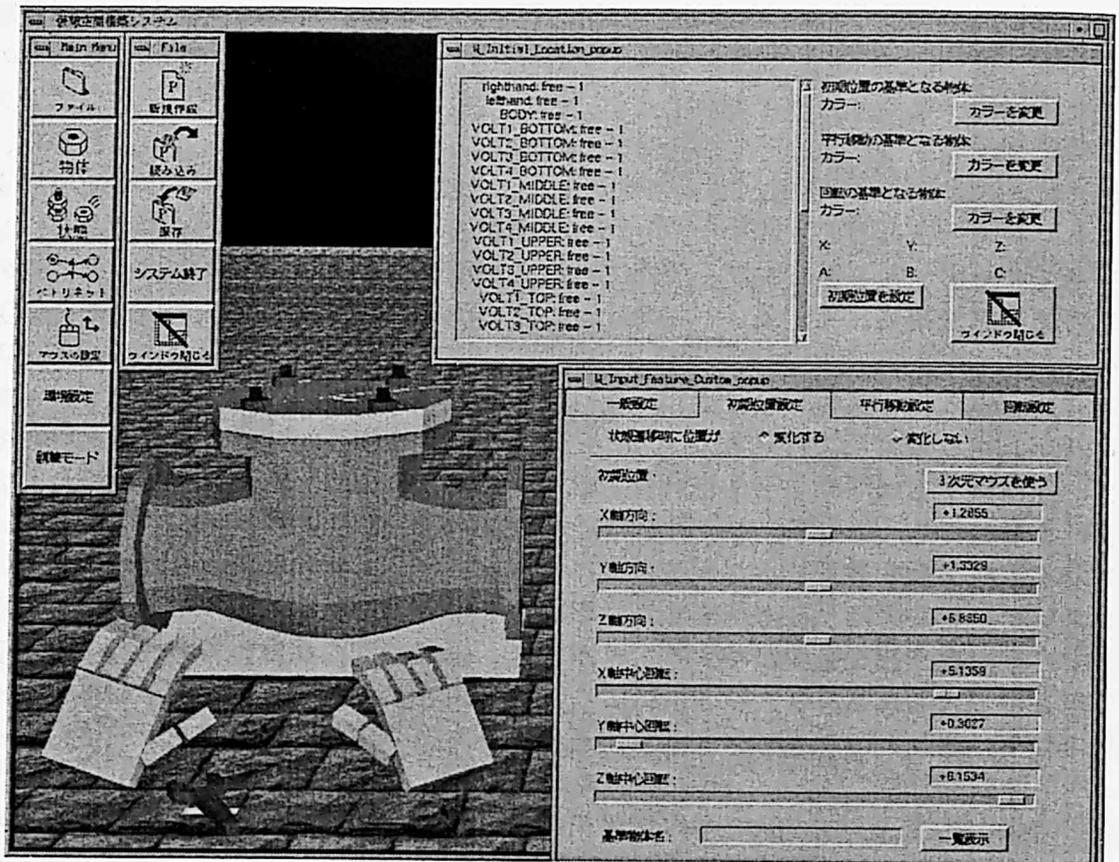


Fig. 4.7: 構築モードの画面例

Table 4.2: 「印を付ける訓練環境」を構築する場合に入力する物体名

| | 入力する物体名 |
|---|---------|
| 1 | 右手 |
| 2 | ペン |
| 3 | ふた |
| 4 | 印 |

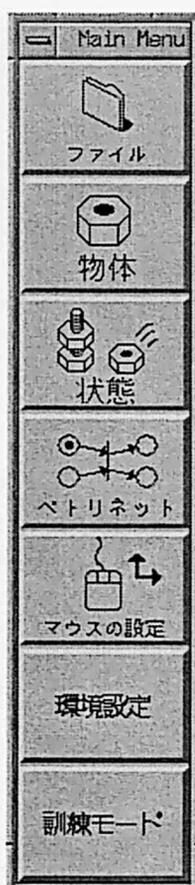


Fig. 4.8: アイコンを張り付けたボタンの例

を介して、ペンの形状データや、各状態における運動データベースを入力する。この際、DESCORTE では、頻繁に入力が必要となる状態に対する運動データベースを、状態テンプレートとして利用できる機能を設けており、Fig. 4.14に示すように、状態テンプレートから、適切なテンプレート名を選択することにより、必要な情報を容易に設定することができる。例えば、新たな状態を作成する際に、テンプレートから「手を開いた状態（自由に動く）」を選択すると、パラメータの各項目に Table 4.4に示すような情報が自動的に入力される。

3. イベントの発生に応じた仮想物体の状態遷移の仕方を設定する。(状態遷移データベースの作成)

ペトリネットを構築することによって、イベントの発生に応じた仮想物体の状態遷移の仕方を設定する。4.4節で述べたように、ペトリネットにおけるプレース1つに対して仮想物体の1つの状態を対応させ、訓練生のジェスチャや他のイベントに応じて物体の状態が変化することをトランジションの発火で表現する。具体的には、ユーザ

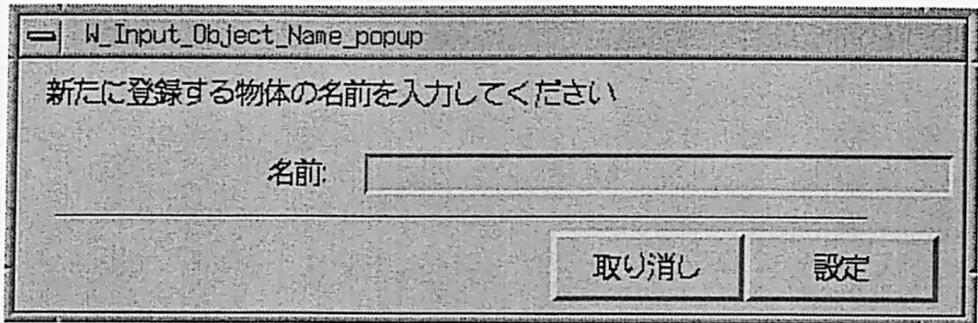


Fig. 4.9: 新たに物体を登録する際のインターフェース

Table 4.3: ペンが取りうる状態

| | 登録する状態名 |
|---|-----------|
| 1 | 何もされていない |
| 2 | 右手に触られている |
| 3 | 右手に握られている |
| 4 | ふたに接触している |
| 5 | 落ちつつある |

Table 4.4: 自動的に入力されるパラメータ

| パラメータ名 | 値 | パラメータ名 | 値 |
|-------------|----------|--------------|----------------|
| 状態名 | (未設定) | 形状格納ファイル | right-open.dxf |
| テクスチャ格納ファイル | hand.rgb | 状態種類 | single |
| 描画方法 | ポリゴン | 大きさ倍率 | 1.0 |
| 平行移動量決定源 | マウス入力 | 回転移動量決定源 | マウス入力 |
| X 軸平行移動 | 可能 | マウス入力種類 (倍率) | マウス入力 X(1.00) |
| X 軸中心回転 | 可能 | マウス入力種類 (倍率) | マウス入力 A(1.00) |
| Y 軸平行移動 | 可能 | マウス入力種類 (倍率) | マウス入力 Y(1.00) |
| Y 軸中心回転 | 可能 | マウス入力種類 (倍率) | マウス入力 B(1.00) |
| Z 軸平行移動 | 可能 | マウス入力種類 (倍率) | マウス入力 Z(1.00) |
| Z 軸中心回転 | 可能 | マウス入力種類 (倍率) | マウス入力 C(1.00) |

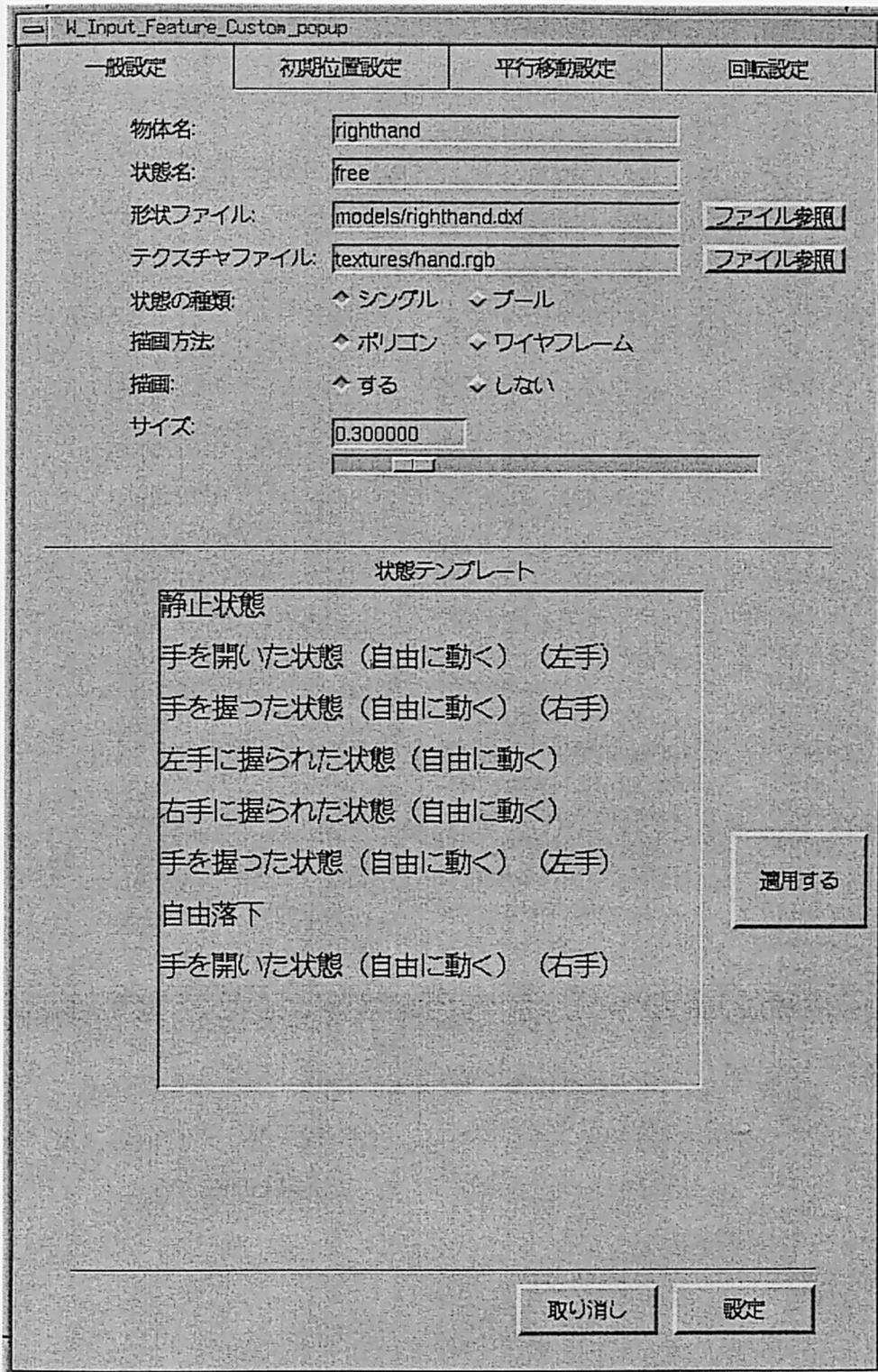


Fig. 4.10: 状態を作成する際のインターフェース (外見に関する情報)

4.5 DESCORTE の開発

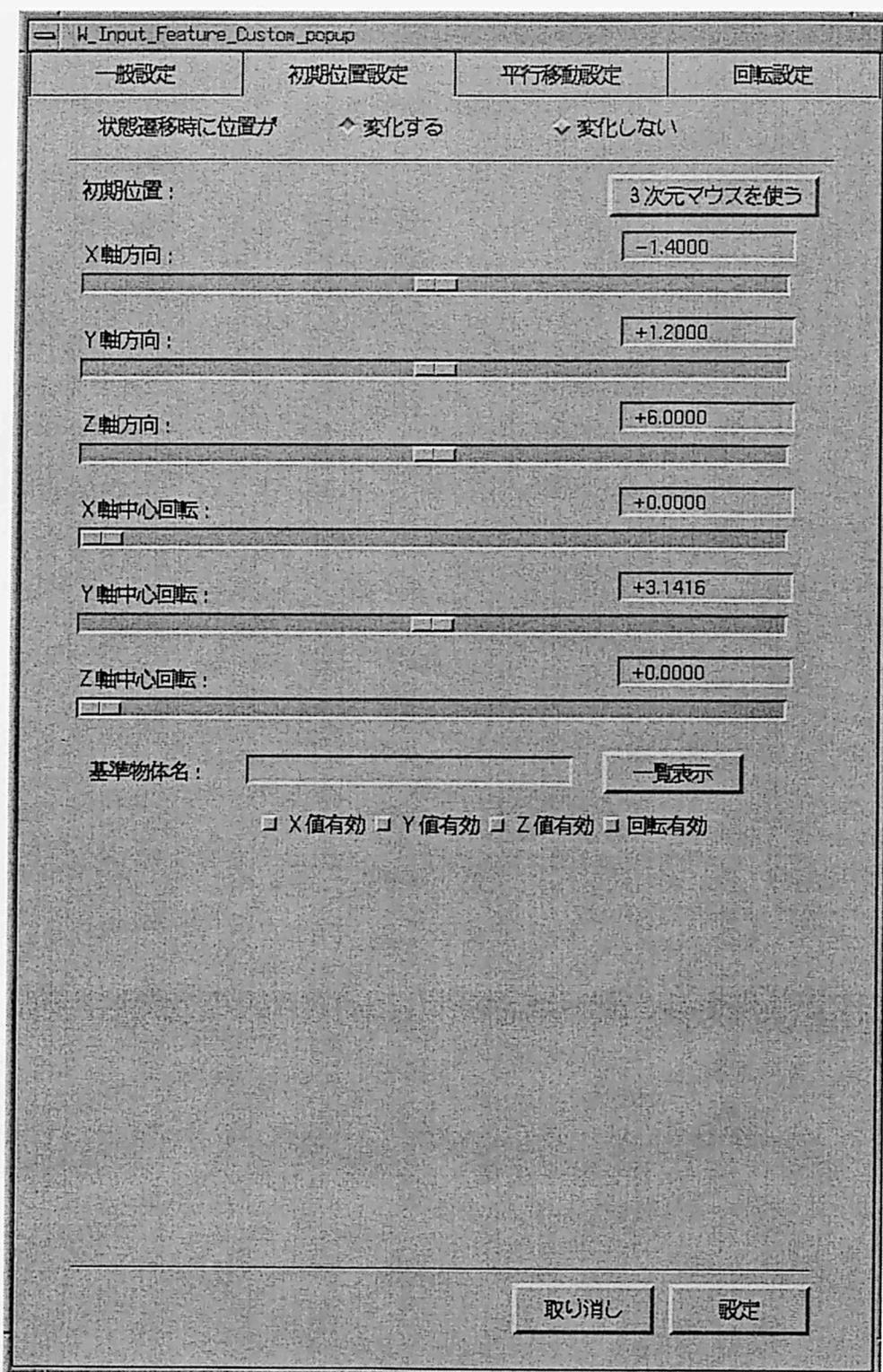


Fig. 4.11: 状態を作成する際のインタフェース (初期位置に関する情報)

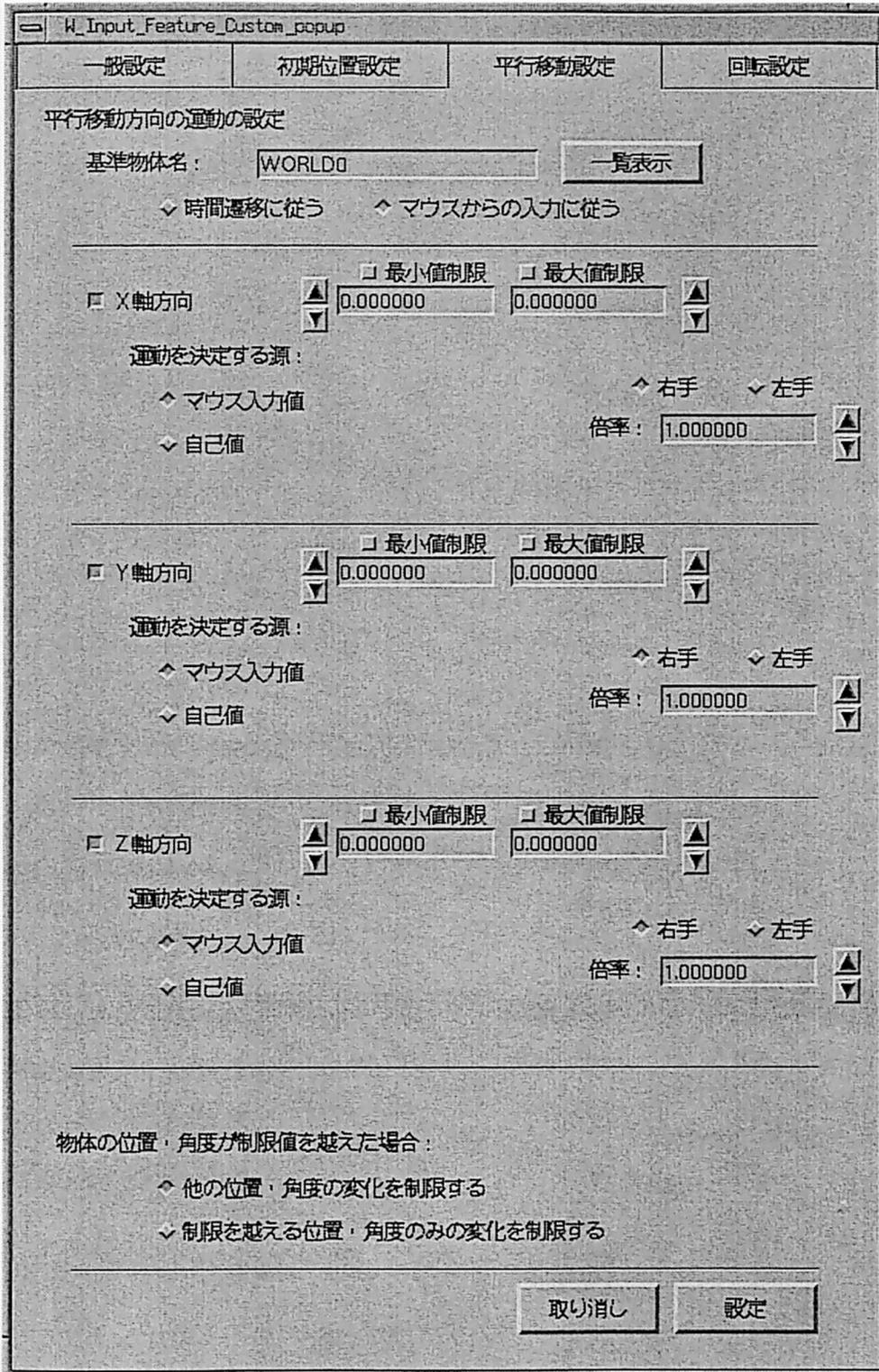


Fig. 4.12: 状態を作成する際のインタフェース (平行移動に関する情報)

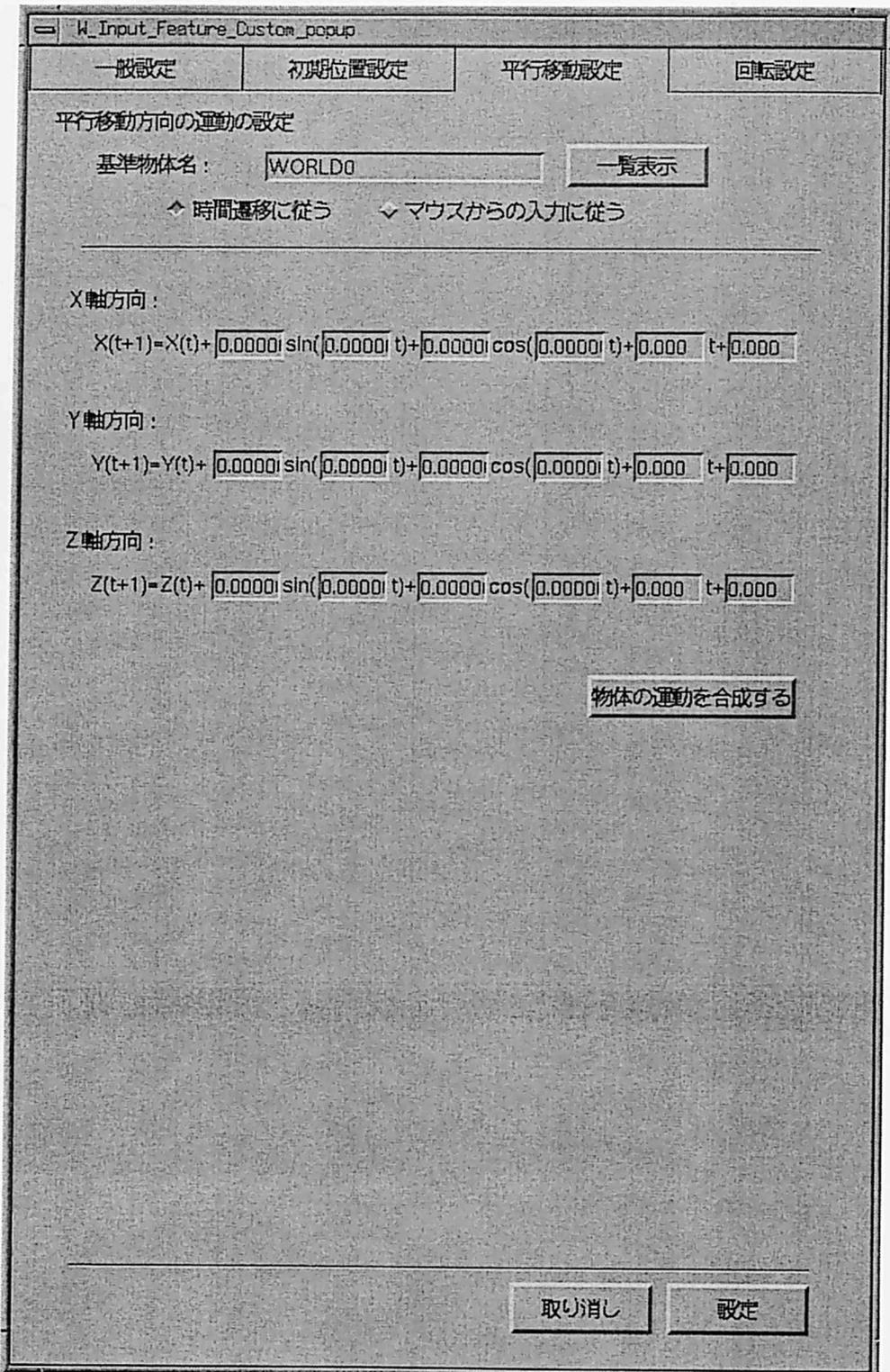


Fig. 4.13: 状態を作成する際のインターフェース（時間に応じた移動法に関する情報）

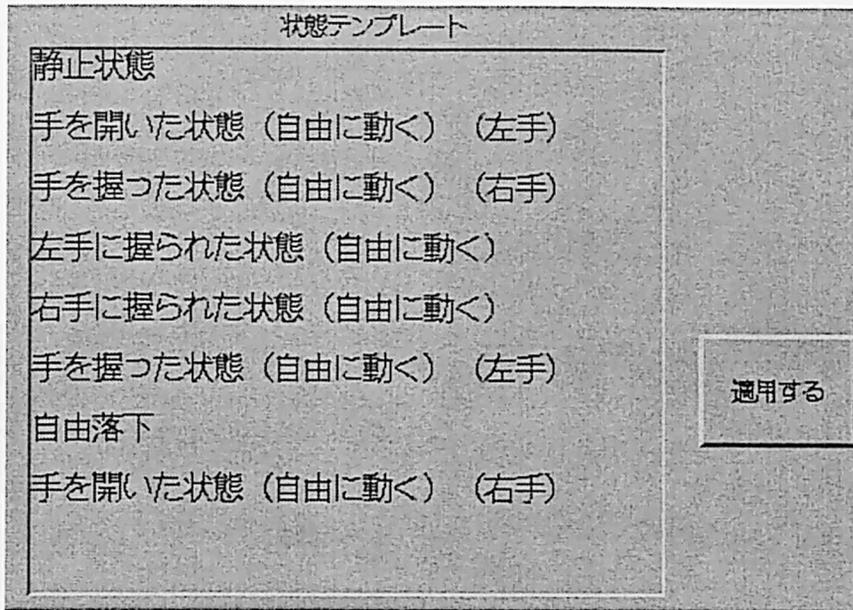


Fig. 4.14: テンプレート機能

は Fig. 4.15 に示すようなインターフェースを通して、仮想物体の変化前の状態と変化後の状態 (4.4 節で述べたようにそれぞれ入力スペース、出力スペースに対応する) を入力し、さらに Fig. 4.16 に示すインターフェースを通して、状態遷移を起こさせる原因となるイベントの設定を行う。構築したペトリネットは、Fig. 4.17 に示すように自動的に配置され、Fig. 4.18 に示すような拡大・縮小ボタンやスライドバーを操作することによって、全体を眺めることもできる。DESCORTE ではさらに、仮想物体の状態遷移が起こるときに発する効果音を個々のトランジションに対して設定できる。

4. 仮想物体の初期状態を登録する。

訓練を実際に行う為には、各仮想物体の初期状態を設定する必要がある。DESCORTE では、ペトリネットにトークンを配置し、配置した各トークンに対して、対応する仮想物体の仮想空間内での初期位置を設定することによって、各仮想物体の初期状態を指定する。DESCORTE では、仮想物体の仮想空間内での位置を指定する場合、3次元マウスを用いて実際に仮想空間内の配置したい位置に仮想物体を移動させることにより、仮想空間内の3次元的位置と角度を指定するものとした。

DESCORTE では、上記の他に特徴的な機能として、Fig. 4.19 に示すような環境設定画面がある。これらのボタンは、上から、仮想物体に影を付けるか否かの選択、テクスチャを貼るか否かの選択、物体をポリゴンで描くか、ワイヤフレームで描くかの選択、床にテクスチャを貼るか否か、の選択を行うためのボタンである。それぞれのボタンの選択により、使用するシステムの性能に応じた3次元画像の描画方法を設定できる。このように、

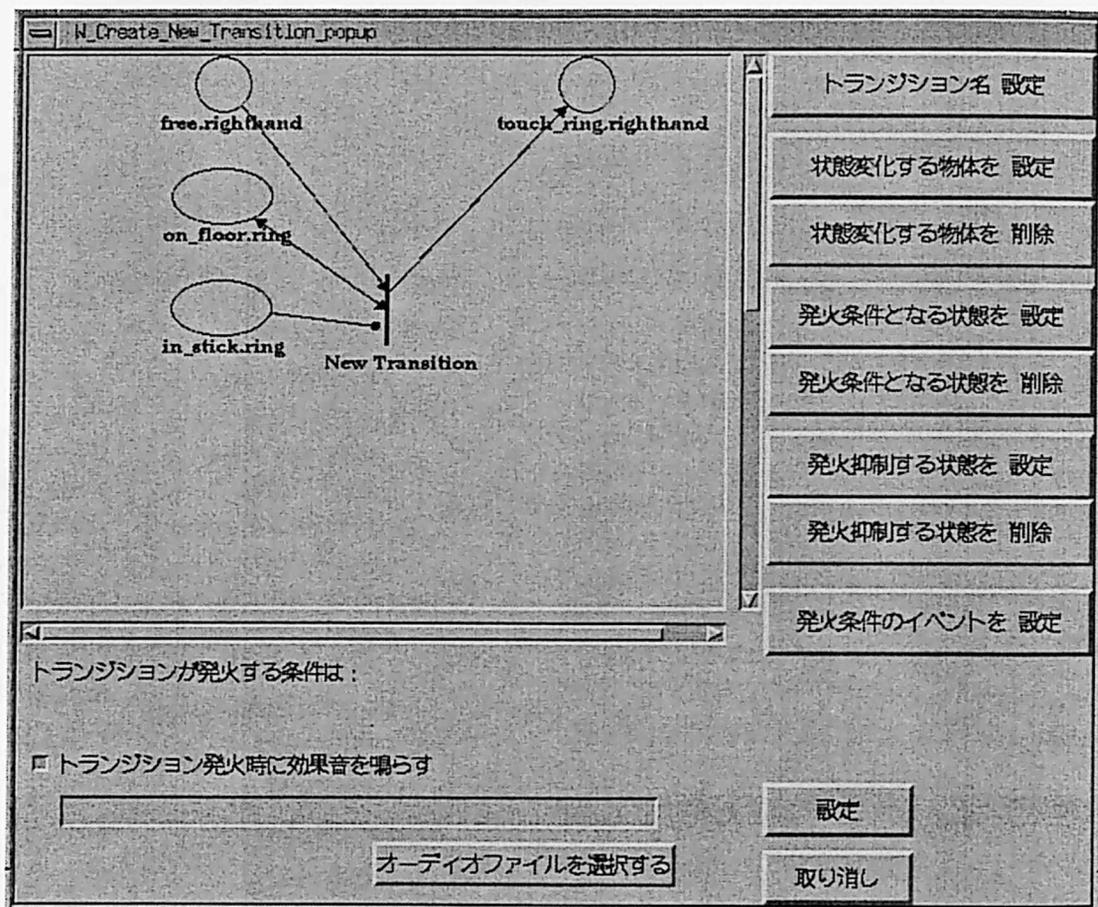


Fig. 4.15: 新規トランジション作成画面

4. 機器保守の訓練環境の構築支援システム DESCORTE の開発

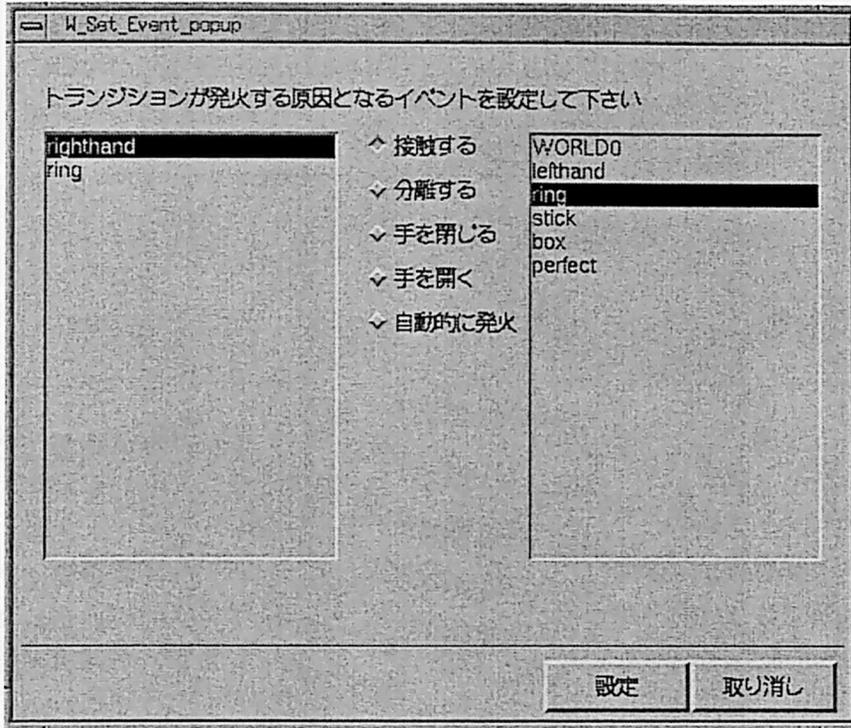


Fig. 4.16: イベント設定画面

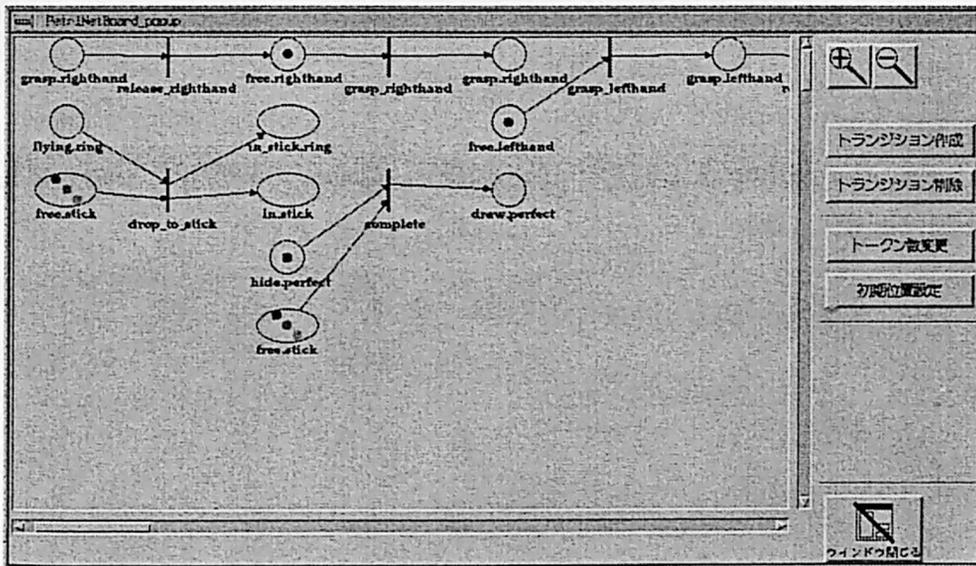


Fig. 4.17: ペトリネット表示画面

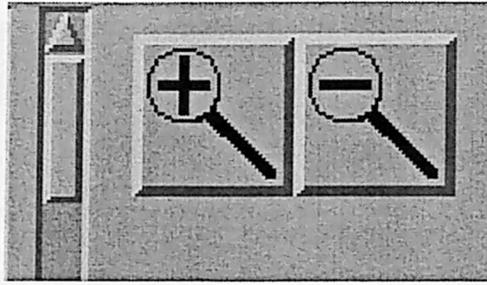


Fig. 4.18: ペトリネット拡大縮小ボタンとスライドバー

ボタンにアイコンを張ることにより、絵を見てそのボタンの機能が類推でき、「ポリゴン」や「ワイヤフレーム」といった専門用語を知らなくても、各種機能が使用できると期待される。

また、新規に仮想空間を構築し直す機能や、DESCORTE を終了させる機能等、実行するとやり直しができない機能に対しては、その機能を実行する前に、Fig. 4.20に示すような確認ウィンドウを表示することにより、ユーザが大きな間違いを犯すのを防ぐ工夫も行っている。

4.5.3 訓練を実施する際の情報処理手順

DESCORTE における訓練実行時の情報の流れを Fig. 4.21に示す。訓練を実行する際には、DESCORTE と同時にペトリネットシミュレータを実行し、発火トランジションと、トークン配置に関する情報をやりとりする。以下に、訓練を実行する場合の情報処理手順を示す。

処理1 発火可能なトランジションの判定

ペトリネットシミュレータから、現在発火可能なトランジションとそのトランジションを発火させる原因となるイベントに関するデータを取り出す。

処理2 訓練生からのジェスチャの入力の判定

訓練生の3次元マウスからの入力を調べる。訓練生からの入力は、ボタンの状態（押されているかどうか）に関するものと、手をどの軸を中心にしてどのくらい回転させようとしているか、手をどの軸方向にどのくらい移動しようとしているか、に関するものが数値として得られる。これらのデータをもとにジェスチャ判定部でどのようなジェスチャに相当するかが判定される。

処理3 発火トランジションの決定

処理1で得られた発火可能トランジションの発火の原因となるイベントと、処理2で得られた訓練生からのジェスチャとを比較し、両者が一致すれば、処理1で得られた

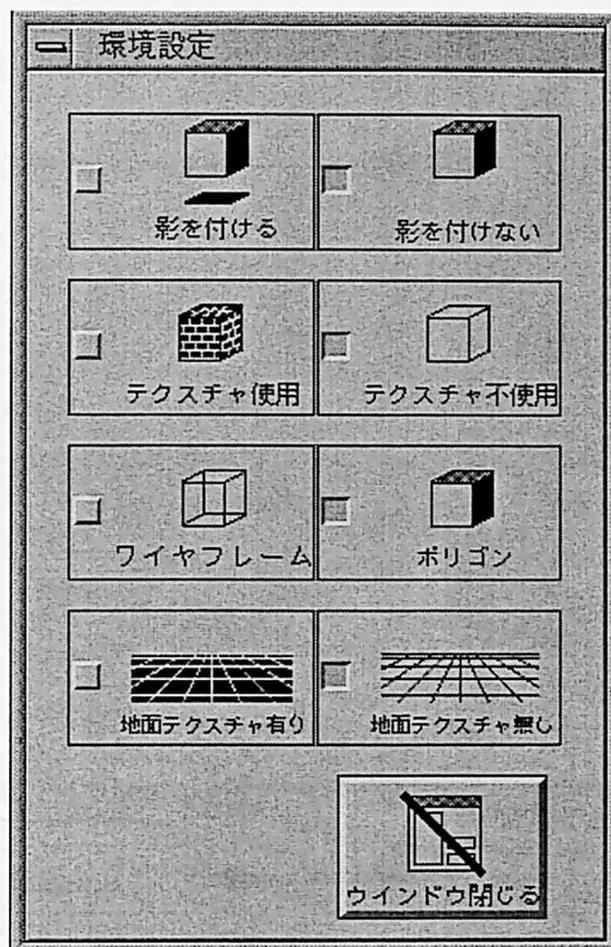


Fig. 4.19: 環境設定画面

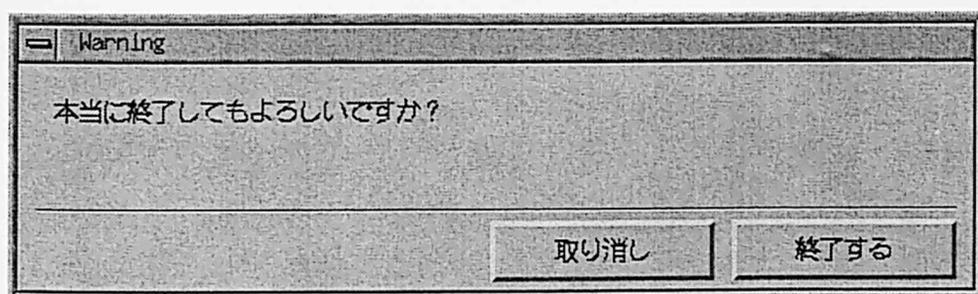


Fig. 4.20: 操作の実行を確認するウインドウ

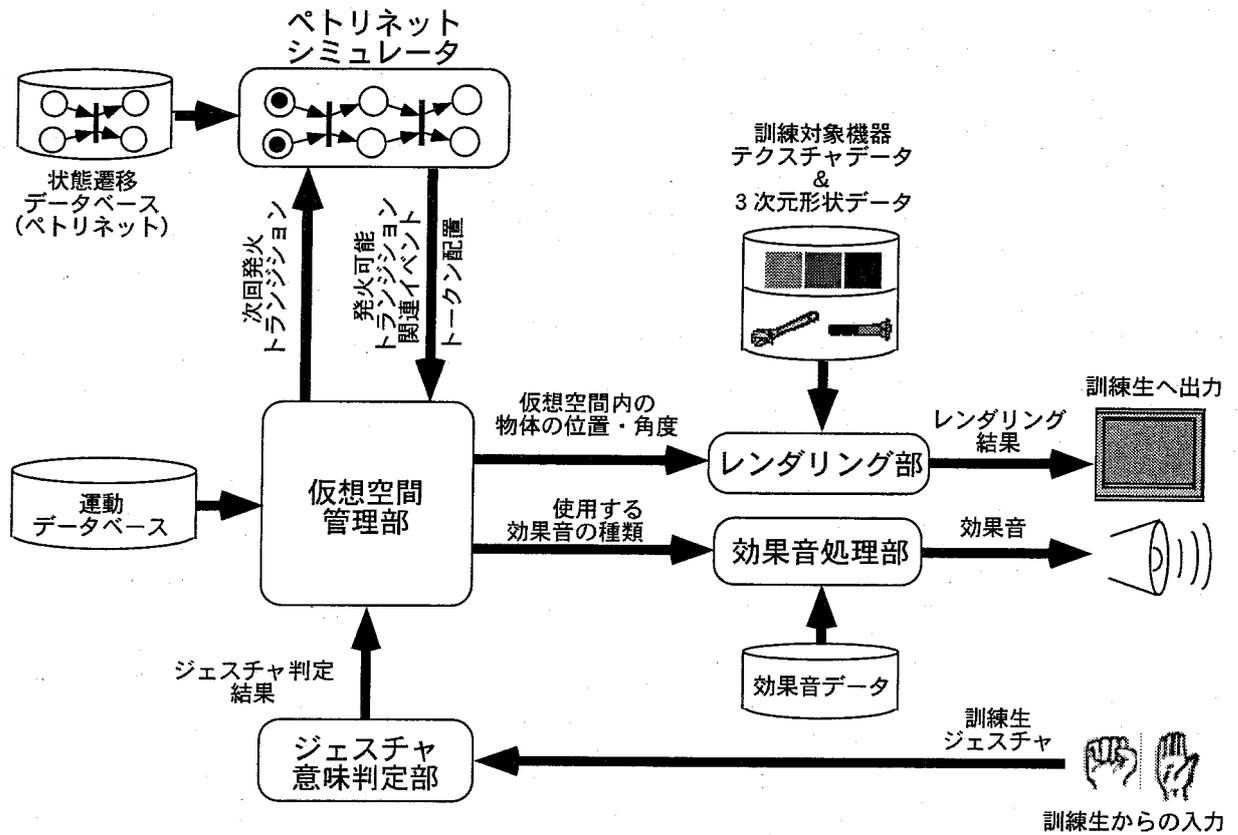


Fig. 4.21: 訓練実行時の情報の流れ

4. 機器保守の訓練環境の構築支援システム DESCORTE の開発

トランジションを発火させるべきトランジションとしてペトリネットシミュレータに渡す。

処理4 トランジションの発火によるトークン遷移の調査

処理3で示したトランジションが発火することによりトークンが遷移し、遷移後のトークン配置が仮想空間を管理する部分に送られる。

処理5 各物体の状態の決定

処理4で得られたトークンの配置と運動データベースをもとに、仮想物体の状態を決定する。

処理6 3次元画像のレンダリングと訓練生への提示

各物体の現在の状態に関する情報はレンダリング部に送られる。その後、訓練対象機器のテクスチャデータと訓練対象機器の3次元形状データとを用いて3次元画像をレンダリングし、訓練生に提示する。

処理7 物体の物理的性質のシミュレーション

ペトリネットにおけるトークン配置の情報と運動データベースをもとに各物体の物理的性質をシミュレーションする。

以上の処理を繰り返すことによって、訓練生のジェスチャを仮想空間に反映させている。

このような内部処理を行って訓練を実施する場合、ペトリネットをシミュレーションするための処理が必要となり、計算機への負荷が増えると予想される。しかし、ペトリネットシミュレータが使用されるのはイベントが発生した時だけである。また、スイング式逆止弁の保守作業を行う訓練を実施する場合は、ペトリネットの処理時間は全体の処理時間の約0.9%に過ぎず、さらに、ペトリネットシミュレータを訓練システムとは別のプロセスで処理する事も可能であるため、大きな問題にはならないと思われる。

4.6 DESCORTE の評価

本節では、DESCORTE の評価を行った結果について述べる。

4.6.1 スイング式逆止弁の分解作業を対象とした訓練環境の構築

プログラミングにより訓練環境を構築する際の作業効率と DESCORTE を用いて訓練環境を構築する際の作業効率を比較するために、VRMATE を開発した際に構築したスイング式逆止弁の分解作業が実施可能な訓練環境を DESCORTE を用いて構築する。

具体的には、訓練対象の3次元形状データとテクスチャデータは既に作成されている状態から作業を行い、3.6節で述べた自動実演機能は DESCORTE では提供する機能が無いため、今回は自動実演機能のための情報作成は行わないものとした。

4.6 DESCORTE の評価

Table 4.5: 登録する仮想物体の名前

| | | | |
|---|-----------|----|-------------|
| 1 | righthand | 7 | top-volt |
| 2 | lefthand | 8 | upper-volt |
| 3 | body | 9 | middle-volt |
| 4 | mark | 10 | lower-volt |
| 5 | lid | 11 | spana |
| 6 | nut | 12 | pen |

Table 4.6: 登録する状態の例 (righthand)

| | | | | | |
|---|--------------------|----|--------------------------|----|----------------------------|
| 1 | free | 7 | grasp-nut-in-top-volt | 13 | grasp-spana-in-upper-volt |
| 2 | grasp-lid-in-space | 8 | grasp-nut-in-upper-volt | 14 | grasp-spana-in-middle-volt |
| 3 | grasp-lid | 9 | grasp-nut-in-middle-volt | 15 | grasp-spana-in-lower-volt |
| 4 | grasp-pen-in-space | 10 | grasp-nut-in-lower-volt | 16 | touch-nut |
| 5 | grasp-nut-in-space | 11 | grasp-spana-in-space | 17 | touch-spana |
| 6 | grasp-nut | 12 | grasp-spana-in-top-volt | 18 | grasp |

Table 4.5～Table 4.8に訓練環境を構築するために入力したデータベースの例を示す。実際にはこれらの情報はテンプレート機能を用いることにより、ほとんどの場合、細部までは入力する必要はない。構築した訓練環境の画面例を Fig. 4.22に示す。

今回の構築作業では、DESCORTE の操作に習熟したユーザが1人で作業を行い、作業開始から作業終了まで約12時間が必要であった。VRMATEを開発する際、自動実演機能を具備していない訓練環境までを、プログラミングにより構築した際に、約3ヶ月の作業時間を要した（自動実演機能を含めると約4ヶ月）ことと比較すると、DESCORTEを用いることにより、訓練環境を構築する際の作業時間が飛躍的に短縮されたと言える。

このように飛躍的に作業時間が短縮された理由は次の様に考えることができる。プログラミングにより訓練環境を構築する場合には、仮想空間内での訓練作業を可能にするための各種アルゴリズムの考察や、プログラムのバグ取り等の作業が必要であり、これらの作業はプログラマが熟練していなければかなりの時間を要する。しかし、DESCORTEを用いて作業を行う場合には、各種アルゴリズムを細部まで考察する必要がなく、また、作業手順が簡略化されたため、間違いを起しにくくなり、バグ取り等の作業量も飛躍的に少なくなり、作業効率が改善された。

4. 機器保守の訓練環境の構築支援システム DESCORTE の開発

Table 4.7: 手に握られた状態にあるペンに対する運動データベース例

| パラメータ名 | 値 | パラメータ名 | 値 |
|--------------|----------------|--------------|---------------|
| 状態名 | grasped | 形状ファイル | pen.dxf |
| テクスチャファイル | pen.rgb | 状態種類 | single |
| 描画方法 | ポリゴン | 大きさ倍率 | 0.3 |
| 初期位置 | 有り | 初期位置基準物体名 | righthand |
| 初期位置 (x,y,z) | (-0.8,0.9,0.0) | 初期角度 (a,b,c) | (0.9,6.0,1.8) |
| 平行移動量決定源 | マウス入力 | 回転移動量決定源 | マウス入力 |
| X 軸平行移動 | 可能 | マウス入力種類 (倍率) | マウス入力 X(1.0) |
| X 軸中心回転 | 可能 | マウス入力種類 (倍率) | マウス入力 A(1.0) |
| Y 軸平行移動 | 可能 | マウス入力種類 (倍率) | マウス入力 Y(1.0) |
| Y 軸中心回転 | 可能 | マウス入力種類 (倍率) | マウス入力 B(1.0) |
| Z 軸平行移動 | 可能 | マウス入力種類 (倍率) | マウス入力 Z(1.0) |
| Z 軸中心回転 | 可能 | マウス入力種類 (倍率) | マウス入力 C(1.0) |

Table 4.8: 落下している状態にあるナットに対する運動データベース例

| パラメータ名 | 値 | パラメータ名 | 値 |
|-------------|---------|----------|--------------|
| 状態名 | falling | 形状格納ファイル | nut.dxf |
| テクスチャ格納ファイル | nut.rgb | 状態種類 | single |
| 描画方法 | ポリゴン | 大きさ倍率 | 0.3 |
| 初期位置 | 無し | | |
| 平行移動量決定源 | 時間 | 回転移動量決定源 | 時間 |
| X 軸平行移動 | 移動不可 | X 軸中心回転 | 回転不可 |
| Y 軸平行移動 | 可能 | 式の係数 | 50.0,0.1,0.0 |
| Y 軸中心回転 | 回転不可 | | |
| Z 軸平行移動 | 移動不可 | Z 軸中心回転 | 回転不可 |

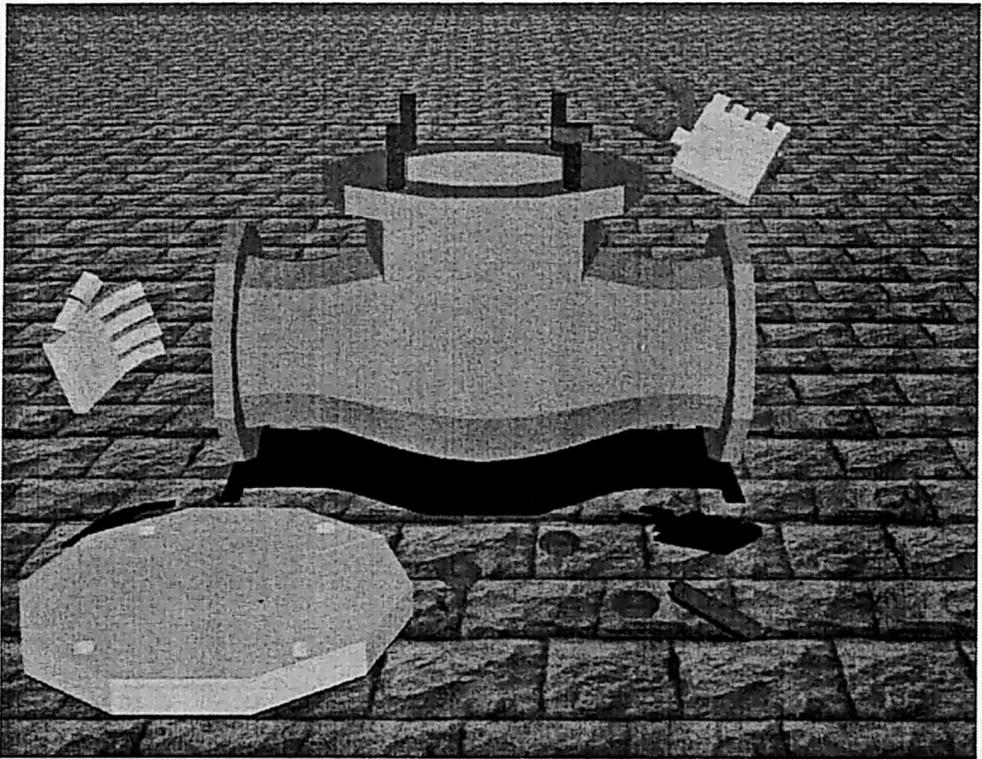


Fig. 4.22: スイング式逆止弁の保守訓練環境の画面例

4.6.2 DESCORTE の操作性の実験による評価

前項で述べた DESCORTE の使用例は、DESCORTE の操作に精通した者による作業の評価であった。そこで本項では、DESCORTE を初めて利用する初心者でも、訓練環境が構築できることを確認するために評価実験を行った結果について述べる。評価は、初心者には簡単な訓練環境を構築させ、その構築に必要とした時間と、作業終了後のアンケート調査により行った。

実験の目的

DESCORTE を初心者にも使用してもらい、初心者でも機器保守の訓練環境を構築できることを確認する。また、DESCORTE の改良すべき点を見い出す。

実験方法

被験者 実験の被験者は、DESCORTE を初めて使用する 20 歳代と 40 歳代の男性 2 人（被験者 A、被験者 B）とした。両者ともプログラミングにより訓練環境を構築した経験は無いが、2 次元マウスとキーボードの操作には十分慣れており、必要な情報を自由なく入力できる。

実験手順 訓練に利用する機器の 3 次元形状データとテクスチャファイル、提示する効果音は、予め適当なアプリケーションを用いて作成されており、被験者には各ファイルの格納場所とその内容について予め説明した。また、構築作業を開始する直前に、DESCORTE の操作に十分習熟した者が、DESCORTE の各機能の説明と操作方法等を約 30 分間説明し、最後に構築作業の例として、「手でペンを握ったり放したりできる訓練環境」の構築作業を見学させた。実験では、作業開始から作業終了までに要した時間を計測し、また、作成した訓練環境データの内容も記録した。アンケートは、構築作業が終了した直後に、感想を自由に述べてもらう形で行った。

被験者に与える課題 被験者が構築する訓練環境は、「手でスイッチを操作することにより、モータのシャフトの回転を制御する訓練環境」とした。仮想空間に配置する物体は、モータ、シャフト（ファン付）、スイッチ、スイッチの台の計 4 つとなり、そのそれぞれについて、3 次元形状データとテクスチャデータを予め準備した。このような訓練環境を構築するために入力する必要がある情報の一例を、Table 4.9 と Fig. 4.23 に示す。仮想物体の状態は、シャフトが「回転中」と「静止」の 2 つの状態をとり、他の物体は状態が変化しない。従って、この構築作業では、ペトリネットの作成は比較的容易であるが、シャフトの回転方法の設定は難しい作業である、と予想した。

実験結果

最終的には 2 人の被験者は共に訓練環境を正しく構築することができた。構築した訓練環境の画面例を Fig. 4.24 に示す。訓練環境構築に必要とした作業時間は、被験者 A では

4.6 DESCORTE の評価

Table 4.9: 回転中のシャフトの状態に対する運動データベース例

| パラメータ名 | 値 | パラメータ名 | 値 |
|-------------|-----------|----------|--------------|
| 状態名 | rotating | 形状格納ファイル | shaft.dxf |
| テクスチャ格納ファイル | shaft.rgb | 状態種類 | single |
| 描画方法 | ポリゴン | 大きさ倍率 | 1.0 |
| 初期位置 | 無し | 平行移動量決定源 | 平行移動不可 |
| 回転移動量決定源 | 時間 | X 軸中心回転 | 回転不可 |
| Y 軸中心回転 | 可能 | 式の係数 | 0.0,10.0,0.0 |
| Z 軸中心回転 | 回転不可 | | |

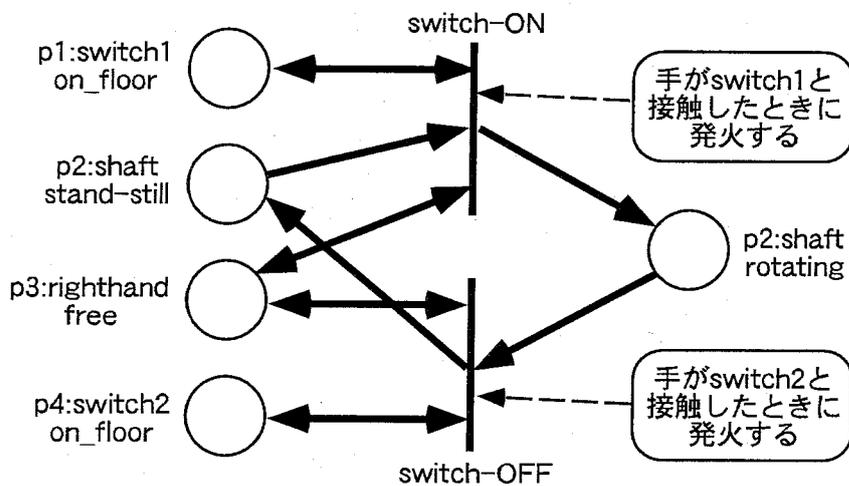


Fig. 4.23: シャフトの状態遷移のモデル化例

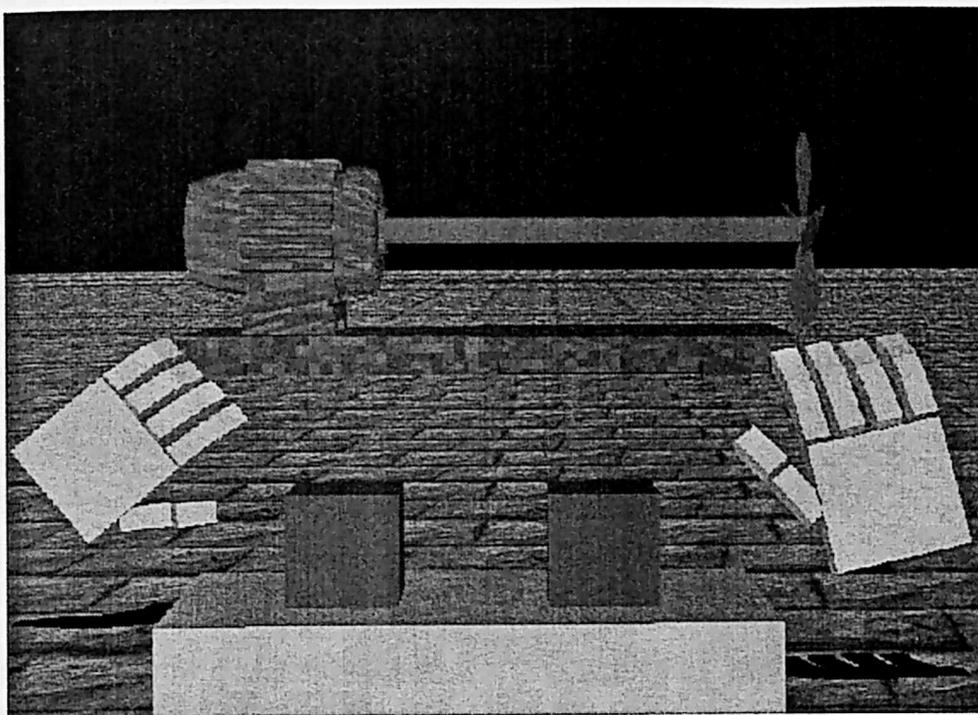


Fig. 4.24: スイッチでモータの回転を制御する仮想空間の画面例

41分、被験者Bでは38分であった。難しいと予想された、シャフトの回転方法の設定も、DESCORTEの各機能を説明する際に実演した方法を真似ることで、混乱することなく設定することができたようである。ただし、被験者Bは、ペトリネットを作成する際に、物体の状態の分け方に少し戸惑ったが、試行錯誤の後、正しいペトリネットを作成することができた。実験後のアンケートにより、両者から次のような感想が得られた。

- DESCORTEを利用して見て、最初は何のボタンがどの機能に対応しているのかが分からなかったが、使っているうちに分かるようになった。
- 3次元マウスが使いにくく、物体の位置合わせが難しかった。
- たくさんのウインドウを開いている時に、仮想空間が隠れて見にくかった。
- 使っているうちに、もっと凝った仮想空間を作りたいと思うようになった。
- システムを使い込むと、もっと自由に仮想空間を構築できるようになると思う。

考察

実験の結果により、DESCORTEを初めて利用する場合でも、単純な訓練環境ならば、試行錯誤しながら最終的には正しく構築できることが確認された。被験者は2人とも、これ

4.7 DESCORTE のインタフェースの改良

までに仮想空間を構築した経験はなく、プログラミングにより仮想空間を構築させた場合、プログラミングの基礎から学ぶ必要があり、多くの学習時間が必要となると思われるが、DESCORTE を用いることにより、せいぜい数時間の練習を行うだけで、目的の訓練環境を構築できると思われる。

今回、構築の対象となった訓練環境は単純であったため、両被験者が構築した運動データベースや状態遷移データベースは、物体の配置位置等を除けば、熟練者が構築するものと全く同じであった。しかし、訓練環境が複雑になると、幾通りものデータベースの記述が可能になるため、構築する人によって構築方法に違いが生じ、作業効率にも違いが生じると予想される。これは、仮想空間構築のテクニックを利用できるかどうかに影響される問題であり、このようなテクニックをユーザが効率的に学ぶことができれば、訓練環境を構築する際の作業時間のさらなる短縮が期待できると考えられる。

一方、アンケートの結果からは、DESCORTE のインタフェースについて、改善すべき点が多々あることが判明した。特に、仮想物体の位置合わせのインタフェースを改良すべきことが判明した。また、3次元マウスを用いて、正確に仮想物体の位置を設定することは、3次元マウスの操作に慣れていない初心者には困難なことも判明した。GUIを介して、2次元マウスを用いて位置合わせを行うことができるインタフェースについても検討すべきであると思われる。また、数多くのウインドウを開くことにより、仮想空間を表示する領域が隠れて見えなくなり、逐一ウインドウを移動させて訓練環境の状態を確認する作業が繰り返し行われているのが目立った。必要な情報を入力する際に、仮想空間を隠さないようなインタフェースにする必要があると思われる。

4.7 DESCORTE のインタフェースの改良

本節では、前節での評価実験により得られた結果をもとに、DESCORTE のインタフェースを改良し、さらにその改良の効果を、新たに実験を行うことにより評価した結果について述べる。

4.7.1 DESCORTE のインタフェースの改良策

本項では、4.5節で開発した DESCORTE のインタフェースの問題点と改良策を、ユーザの認知的負荷の観点から検討する。

1. 仮想物体の位置と角度を指定するインタフェース

インタフェース改良前の DESCORTE では、仮想物体の3次元的位置や角度を指定する際、主に3次元マウスを用いて実際に対象となる仮想物体を目的の位置・角度まで移動させることにより指定していた。しかし、3次元マウスは3次元空間における大まかな位置を示す場合には向いているが、3次元空間の位置や角度を正確に示す際には、3次元マウスのパッドを微妙に調節しながら操作する必要があり、ユーザにかなりの負担がかかることが判明した。さらに、インタフェース改良前の DESCORTE

では、ユーザが視点移動する際には、視点変更モードに切り替えて、3次元マウスを用いて視点移動するインタフェースを構築していた。しかし、例えば、ある仮想物体を他の仮想物体との位置関係を考慮しながら配置する作業を行う際は、これら両仮想物体間の位置関係を常に把握するために、頻繁に視点を移動しなければならず、仮想物体を移動させるモードと、視点を移動させるモードを交互に切り替えて作業を行う必要があった。従って、作業が繁雑になると同時に、常に視点移動モードか仮想物体を移動させるモードかを認識する必要があり、ユーザに負担がかかることも判明した。そこで今回は、3次元マウスは視点の移動のみに使用することとし、仮想物体の3次元位置と角度は2次元マウスを用いて設定するインタフェースに変更する。これにより、視点を移動する操作と仮想物体を移動させる操作を明確に分けて考えることができるようになり、ユーザに対する負担を減らすことができると期待される。

2. 訓練環境を構築する際、数多くのウィンドウの開閉が必要である問題

インタフェース改良前の DESCORTE では、運動データベースや状態遷移データベースを入力するために、様々なウィンドウの開閉を繰り返し行わなくてはならなかった。また、一度に開くウィンドウの数が増えると、仮想空間を描画するための領域が開いたウィンドウにより隠されてしまうため、ユーザは、頻繁にウィンドウを移動させるか、ウィンドウを一度閉じて、仮想空間の状態を確認し、再びウィンドウを開くといった作業を繰り返し行っていた。この方法では、情報を入力する際に必要な操作の回数が増えるだけでなく、仮想空間を観察しながら情報を入力することができなくなるために、ユーザに負担がかかると考えられる。そこで今回は、複数のウィンドウに分かれて配置されていたボタンやスライドバー等の大きさを一回り小さくし、それらをできる限り1つのウィンドウに集約することにより、ウィンドウが画面上を占める割合を減らす試みを行った。また、インタフェース改良前の DESCORTE では、ディスプレイ全面を仮想空間の描画領域として用い、ウィンドウはそれらに上書きする形で表示していたが、今回は、仮想空間を描画する領域とウィンドウを表示する領域をできる限り分けることにより、ウィンドウが仮想空間を描画する領域を隠さないインタフェースに変更する。

3. アイコンを張り付けたボタン等が仮想空間の表示領域を隠してしまう問題

WIMP インタフェースの1つの特徴であるアイコンは、ウィンドウに配置されるボタンに張り付けることにより、そのボタンの機能を類推させたり、各ボタンの機能を覚えやすくしたりする効果が期待できる。インタフェース改良前の DESCORTE でも、システムを初めて利用するユーザが、直感的に各ボタンの機能が分かるようにするために、大きめのアイコンを張り付けていたが、大きなアイコンは DESCORTE を本格的に利用する場合には、仮想空間を描画するための領域を隠してしまうため、使いにくいという意見があった。そこで今回は、仮想空間を見る必要がある場合は極力アイコンを使用しないものとし、多くの場合、機能を説明するテキストを張り付けたボタンを利用し、さらに、少ない占有面積で多くの機能を配置できるプルダウンメニューを採用する。

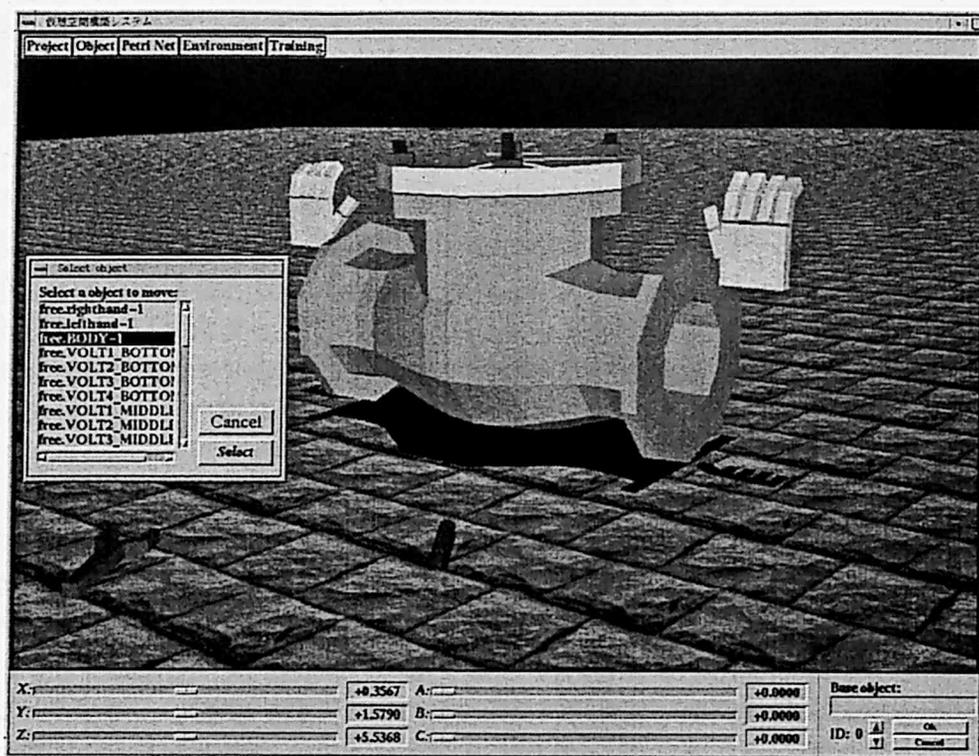


Fig. 4.25: インタフェース改良後の DESCORTE の画面例

4. 訓練環境を構築する際に設定すべきパラメータの数が多問題

4.3.2項で述べたように、インタフェース改良前の DESCORTE では、柔軟に訓練環境を構築できるようにするために、数多くのパラメータを設定可能としていた。これに対し、数多くのパラメータを入力する煩雑さを解消するために、よく使用するパラメータの値の組み合わせを「状態テンプレート」として登録し、後にそれらを選択して使用できる機能を構築していた。これにより、ある程度は作業時間が短縮されたと思われる。今回はさらに、運動データベースを構成する各パラメータについて、頻繁に入力される値を各項目ごとに予め設定しておき、必要な場合にはそれらを修正するインタフェースに変更した。また、インタフェース改良前の DESCORTE では、訓練開始時の初期状態（仮想物体の初期位置や角度、ペトリネットのトークン配置等）をユーザが個々に指示していたが、今回は、ユーザが各仮想物体に対して最初に作成した状態を、訓練開始時における仮想物体の初期状態とし、必要に応じて変更するものとした。これは多くのユーザが、訓練環境を構築する際、訓練開始時の状態から作成する傾向があるため、システムを操作する回数を減らすことができると考えられる。

以上のようなインタフェースの改良を行った後の DESCORTE の画面例を Fig. 4.25に示す。Fig. 4.25に示した画面は、Fig. 4.7に示したインタフェース改良前の DESCORTE の画

面例とほぼ同じ情報と、操作ボタン、スライドバー等を配置しているが、仮想空間を描画するための領域はインタフェース改良後の方がかなり大きくなっている。

4.7.2 インタフェース改良後の DESCORTE の評価

本項では、インタフェースの改良による効果を確認するために、新たに評価実験を行った結果について述べる。

実験目的

DESCORTE のインタフェースを改良することにより、訓練環境を構築する際の作業効率が改善されることを確認する。

実験方法

評価は、4.6節で述べた評価実験とほぼ同じ条件で行った。但し、DESCORTE を繰り返し使用することによる学習効果の影響を避けるため、実験の被験者は、4.6節で述べた評価実験とは別の 20 歳代の男性 2 人（被験者 C、被験者 D）とした。両者とも、DESCORTE を使用した経験は無いが、2次元マウスとキーボードは不自由なく使用できる。

実験結果

2 人の被験者は、共に訓練環境を正しく構築することができた。訓練環境構築に必要とした作業時間は、被験者 C は 20 分、被験者 D では 16 分であった。実験後のアンケートにより、両者から次のような感想が得られた。

- 思ったより簡単に仮想空間を構築することができた。
- ペトリネットには少し慣れが必要だと感じた。
- プログラミングで同じ環境をはじめから構築しようとするすると 3 ヶ月は必要だと思う。
- 物体の動き方を数式で表現しているが「自由落下」「回転」等、抽象的な言葉を選択すると、それに応じた動きを設定できるようにもして欲しい。
- 床を自由に変更できるようになれば、簡単なゲームも作れると思う。

考察

インタフェース改良前の DESCORTE を用いて訓練環境を構築した際の作業時間と、インタフェース改良後の DESCORTE を用いて訓練環境を構築した際の作業時間を Table 4.10 に示す。今回の評価実験では、4.6.2項で述べた評価実験と全く同じ訓練環境を構築したのにも関わらず、約半分の時間で構築が完了している。このように、構築の作業時間が短縮

4.8 考察及び将来課題

Table 4.10: インタフェース改良による効果

| 改良前の作業時間 | | 改良後の作業時間 | |
|----------|------|----------|------|
| 被験者 A | 41 分 | 被験者 C | 20 分 |
| 被験者 B | 38 分 | 被験者 D | 16 分 |

された理由については、多くのパラメータ値に対して、予め頻繁に使用される値を設定したことによる効果が大きいようであった。また、構築作業を観察すると、ユーザがシステムを操作する回数がかなり減少していることも確認できた。これらの要因が影響して、構築時間の短縮に繋がったと考えられる。

また、今回の実験のアンケート調査では、4.6.2項で述べた評価実験の際に得られた「物体の位置合わせが難しい」という意見は得られなかった。物体の位置・角度を設定するインタフェースに、3次元マウスを用いずに2次元マウスを用いるようにした効果が出たものと思われる。

4.8 考察及び将来課題

本研究で開発した DESCORTE は、プログラミングにより訓練環境を構築する代わりに、予め選定されたパラメータを必要に応じて設定、修正する事により、目的の訓練環境を構築する方法を採用した。これにより、プログラミング等の知識が必要なくなり、構築の作業手順も簡略化されたため、作業時間が飛躍的に短縮される結果となったが、これによる代償として、プログラミングによる訓練環境の構築方法に比べ、構築可能な訓練環境が制限されると予想された。しかし、今回、DESCORTE を使用した範囲では、「パラメータ設定の自由度が低いために構築したい訓練環境が構築できない」という問題は現れなかった。これは、DESCORTE で構築した訓練環境の種類がまだ少ないためであるとも考えられるが、一応、予め設定可能なパラメータの種類は現時点で十分であると思われる。ただし、今後、設定可能なパラメータの種類は必要に応じて順次増やしていく必要があると思われる。

一方、DESCORTE を用いて訓練環境を構築する際の作業を支援するための機能は、今後追加していく余地があると考えられる。例えば、今回開発した DESCORTE では、ペトリネットを視覚的に構築できる機能は構築されているが、構築したペトリネットから、目的のトランジションやプレースを検索する機能は構築されていない。構築対象となる訓練環境がさらに複雑になった場合に、構築すべきペトリネットの規模も大きくなることが予想され、そのような場合は、トランジションやプレースを検索する機能が必要になると思われる。また、DESCORTE のインタフェースに関しても、今後、インタフェースの評価と改良を繰り返すことにより、さらなる作業時間の短縮が可能であると思われる。

また、今回の DESCORTE の開発では、訓練生が機器保守の作業手順を効率的に学習できるようにするための自動実演機能については考慮しなかったが、VRMATE を開発した際のシステムの試用実験では、「自動実演機能は作業手順を全く知らない初心者にとっては

4. 機器保守の訓練環境の構築支援システム DESCORTE の開発

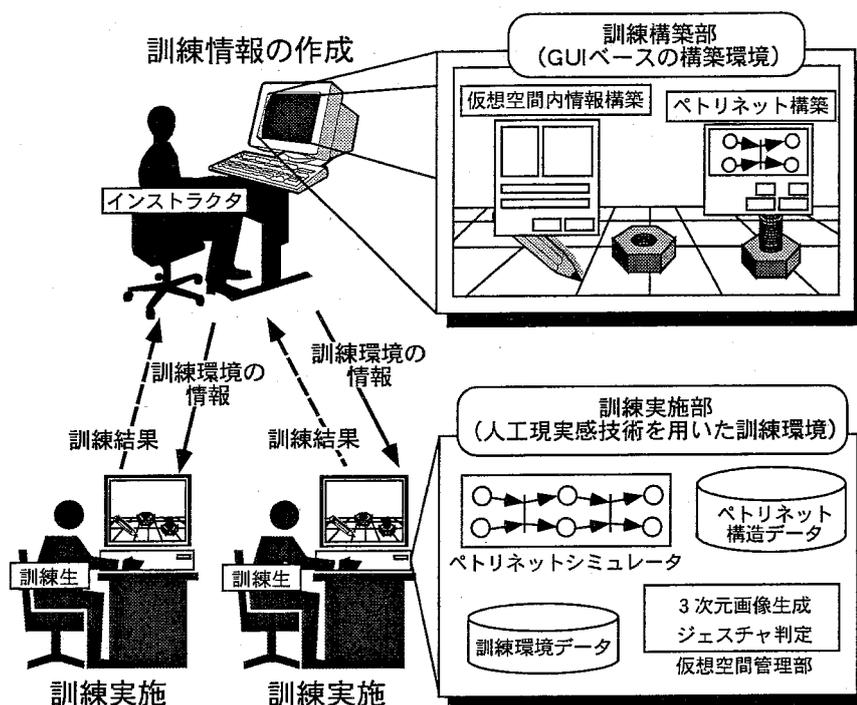


Fig. 4.26: DESCORTE の将来構想

有効な機能である」との結果を得ていることから、今後、自動実演機能を提供するためのデータベースの作成を支援する機能を構築する必要があると考えられる。

本研究で開発した DESCORTE を使用する際の訓練体制としては、今後、機能を拡張することにより、Fig. 4.26に示すように、教育訓練を実施できるようになると考えている。すなわち、次のようなステップで実際の訓練に使用できる。

- (1) まず、訓練環境の構築者（訓練のインストラクタを想定）は、DESCORTE を用いて、複数の訓練生用の訓練環境をその GUI を介して構築する。
- (2) それぞれの訓練生には、それぞれの訓練環境用のデータが提供されるので、それぞれの仮想空間内で個別に訓練を行う。
- (3) それぞれの訓練生の訓練の進行状況について、その訓練状況のデータが自動的に記録され、インストラクタに各々の訓練生の履歴データが送信される。
- (4) インストラクタは、送信された履歴データを解析して、それぞれの訓練生の修得レベルに応じた適切な教示を訓練生にフィードバックする。

なお、現時点での DESCORTE では上記の内、(3)、(4) を可能にする機能が構築されておらず、今後の課題である。

4.9 まとめ

本章では、計算機に関する高度な知識を持たない人でも、容易に人工現実感技術を用いた機器保修の訓練環境を構築できるようにするための、訓練環境の構築支援システム DESCORTE の開発を行った。

本章ではまず、DESCORTE で構築可能とすべき訓練環境の要件について検討し、実際の作業に似たジェスチャで訓練できること、視点を自由に変更できること、臨場感の高い訓練環境であること、仮想空間内での作業を行いやすくするため機能を備えていること等が必要であると述べた。

次に、プログラミング以外の方法で、訓練環境を構築可能にするために、仮想物体の取りうる状態における外見や運動の仕方を定める「運動データベース」とイベントの発生に応じた仮想物体の状態遷移の仕方を定める「状態遷移データベース」を構築する必要があることを述べた。

運動データベースに関しては、実際の訓練の現場で訓練対象となっている複数の機器に対して、それらの機器の保修作業を行うことができる訓練環境を構築するために設定可能である必要があるパラメータを分析することにより、DESCORTE で設定可能とすべきパラメータの種類を決定した。また、状態遷移データベースに関しては、第3章で提案した仮想物体の状態遷移の仕方をペトリネットを用いてモデル化する手法を改良し、初心者にも理解しやすく、よりモデル化能力の高い改良ペトリネットを提案し、DESCORTE で訓練環境を構築するインタフェースに応用した。

次に、開発した DESCORTE のシステム構成について述べた後、運動データベースと状態遷移データベースを入力するためのインタフェースについて述べた。そして、実際に DESCORTE を用いて訓練環境を構築する際の手順を説明した。本研究では、DESCORTE のインタフェースを、近年幅広く使用されているパソコンの OS で採用されている WIMP インタフェースを基礎にして設計した。

続いて、DESCORTE の評価を行った。まず、DESCORTE の熟練者による使用実験として、第3章で構築したスイング式逆止弁の分解作業を行うことができる訓練環境を、DESCORTE を用いて構築した。その結果、プログラミングにより訓練環境を構築する場合に比べて、飛躍的に作業時間が短縮されることを確認した。また、初心者による DESCORTE の使用実験として、初めて DESCORTE を使用する人に簡単な訓練環境を構築してもらい、思考錯誤しながらでも、最終的には正しい訓練環境を構築できることを確認した。しかし、一方で、DESCORTE のインタフェースで改良すべき点が多々あることも判明した。

そこで、DESCORTE を使用する際のユーザの認知的負荷を軽減させる方針でインタフェースの改良を行った。そして改めて評価実験を行うことにより、DESCORTE のインタフェースを改良したことにより、作業時間がさらに短縮されることを確認した。

参考文献

- [1] 菊池安行, 山岡俊樹 (編) . GUIデザイン・ガイドブック. 海文堂, 1995.
- [2] 今井拓史. 家庭用ヘッド・マウント・ディスプレイ -懸念される人体への影響. 日経エレクトロニクス, No. 618, pp. 67-83, 1994.
- [3] 福島省吾, 森川大輔, 吉川榮和. 両眼視差による瞳孔反応の Eye-Sensing HMD による実験研究. 計測自動制御学会論文集, Vol. 35, No. 2, pp. 191-199, 1999.
- [4] J. L. Peterson. ペトリネット入門, pp. 190-212. 共立出版, 1984.
- [5] 相川恭寛. OpenGLプログラミング・ガイドブック. 技術評論社, 1995.
- [6] 兜木昭男, 木下凌一, 栄谷政己, 林秀幸, 安川悦子. X-Window OSF/Motifプログラミング. 日刊工業新聞社, 1994.
- [7] 田村博 (編) . ヒューマンインタフェース. オーム社, 1998.

第 5 章 人工現実感技術を用いた原子力プラント 運転の体験型教育システム VENUS の 開発

5.1 はじめに

本章では、原子力発電プラントの異常診断と対応操作を教育対象とする人工現実感技術を用いた体験型教育システム (VR-based Education System for Operating Nuclear Power Plants ; VENUS) の開発について述べる。

2.2節で述べたように原子力発電プラントの運転員は、幅広い知識と技術、論理的な思考能力を修得していることが要求される。特に原子力発電プラントの異常発生時における異常診断と対応操作を行うための知識や技術を修得することは、プラントの安全性を確保するために非常に重要である。原子力発電プラントの異常診断と対応操作を行う場合、制御盤上に配置された数多くの機器から、プラントの様々な状態を表すパラメータを参照し、それらの結果を総合して判断することにより異常原因を推定し、次に参照すべきパラメータの種類や、次に実行すべき操作等を決定する。従って、原子力発電プラントの異常診断と対応操作を自ら判断して行えるようになるためには、ある程度の経験を積んでいる必要があり、プラント運転の初心者が、何の支援も無しに自ら制御盤を操作して体験訓練を行うのは困難である。そこで本研究では、プラント監視制御行動中の運転員の行動を、人工現実感技術を用いて、その思考過程や機器等の詳細な参照行動まで総合的に情報可視化し、その結果を訓練生に提示することにより、訓練生に欠けている知識や技術を新たに修得させることに重点を置いた教育システムの開発を行うことにした。

具体的には、原子力発電プラントの様々な異常事象を模擬できるプラントシミュレータと、プラント異常発生時における運転員の行動と思考過程を模擬できるヒューマンモデルシミュレータを利用し、プラントシミュレータによって模擬されるプラントの様々な状態に応じて、ヒューマンモデルシミュレータによって生成される運転員の行動や思考過程を訓練生に理解させることにより、プラントの異常診断や対応操作の方法を学習させるものとした。

この場合、プラントシミュレータが模擬するプラントの挙動やヒューマンモデルシミュレータが模擬する運転員の行動や思考過程を、如何に訓練生に理解しやすい形で提示するかが問題となる。本研究ではまず、プラントシミュレータが模擬するプラントの挙動に関しては、仮想空間内に原子力発電プラントの制御室を3次元モデルとして詳細に構築し、プラントシミュレータの計算結果をリアルタイムに制御上の機器に反映させることにより、3次元映像として可視化するものとした。また、ヒューマンモデルシミュレータが模擬する運転員の行動に関しては、仮想空間内に構築された仮想運転員が、実際の運転員の動作

5. 人工現実感技術を用いた原子力プラント運転の体験型教育システム VENUS の開発

に近い自然な動作で、仮想空間内の制御盤を操作する様子を3次元映像として合成することにより可視化するものとした。さらに、ヒューマンモデルシミュレータが模擬する運転員の思考過程に関しては、仮想運転員が自ら思考過程を発話するという形で可聴化するものとした。

また、仮想空間内に仮想運転員を構築する場合、仮想運転員の姿勢情報を用いれば、仮想運転員の視点位置と方向を求めることができる。そこで本研究では、訓練生が CrystalEyes を用いて立体視しながら、仮想運転員の視点からの映像を「体験」できる機能を構築する。この機能を用いることにより、訓練生はあたかも自分で制御盤を操作しているような感覚を得ることができ、プラントの異常診断と対応操作時の作業の様子を容易に理解できるようになると期待される。

以下では、まず始めに、VENUS が具備すべき機能を検討する。次に、本研究で開発した VENUS の概要として、そのソフトウェア構成とハードウェア構成について述べた後、VENUS を構成する各シミュレータについて述べる。そして、VENUS の動作例と、VENUS を数人の被験者に試用してもらった結果について述べ、最後に今後の課題を述べる。

5.2 VENUS に必要とされる機能

訓練生が原子力発電プラントの異常診断や対応操作等の作業の様子を容易に理解できるようにするために、VENUS が具備すべき機能を以下にまとめる。

1. 訓練生が制御盤の配置や、制御盤上の各種機器の配置等を容易に理解できる。

原子力発電訓練センターで初期訓練カリキュラムを受ける訓練生は、原子力発電プラントの制御盤がどのように配置されているのか、制御盤上の各種機器がどのように配置されているのか等を理解していない者が多い。従って、VENUS が、このような制御室の実際の様子を把握するための支援機能を備えることは重要である。そこで、VENUS では、教育対象となる原子力発電プラントの制御室を3次元モデルとして仮想空間内に細部まで構築し、3次元マウスを用いることにより、自由に視点を移動しながら制御室の様子を観察できる機能を構築する。

2. 原子力発電プラントの異常発生時の制御盤の挙動を観察できる。

原子力発電プラントにおける異常発生時の制御盤の挙動を自由に観察できることは、プラントの異常診断や対応操作の技術を身につけるために重要である。これまで開発されたプラント運転を教育対象とした訓練システムでは、異常発生時における制御盤の挙動を、簡略化した2次元映像として提示する場合が多かった。しかし、実際に制御盤を操作する時と同じ見え方で制御盤を観察できる方が、異常発生時のプラントの挙動を正しく把握できると考えられる。そこで、VENUS では、異常事象を模擬できる原子力プラントシミュレータを使用し、そのシミュレータの計算結果をリアルタイムに仮想空間内に構築した制御盤上の機器に反映させることにより、訓練生が、3次元映像として制御盤の挙動を観察できるようにする。

5.2 VENUS に必要とされる機能

3. 仮想運転員が自らの判断で制御室内を歩き回り、異常診断と対応操作を行う様子を観察できる。

訓練生が原子力発電プラントの異常診断や対応操作の方法を学習する場合、第3者が異常診断や対応操作を行う様子を観察する方法が最も実際の作業の様子を把握しやすい方法であると考えられる。そこで、VENUS では、三菱電機が開発したマンマシンインタフェース設計評価システムである SEAMAID[1] を仮想運転員の頭脳として利用し、プラントの状態に応じて知的に判断し、自ら行動を決定する仮想運転員を構築する。そして、訓練生が自由な視点から仮想運転員の作業の様子を観察できる機能を構築する。また、VENUS では、仮想運転員の視点からの映像を CrystalEyes を用いて立体視しながら眺める機能を構築し、訓練生があたかも自分で制御盤を操作しているような感覚を得ることができるようにする。

4. 訓練生は仮想運転員が運転操作する様子を観察するだけでなく、仮想運転員の思考過程も同時に理解できる。

2.3節で述べたように、原子力発電プラントの運転訓練では、その異常診断時における考え方（思考過程）を理解することが重要である。SEAMAID では、プラントの運転員の行動だけでなく、思考過程も模擬しているため、仮想運転員の頭脳として用いることにより、プラント異常発生時における異常診断と対応操作の様子を、行動として合成するだけでなく、異常診断の際の思考過程を訓練生に提示する機能も構築できる。そこで、VENUS では、仮想運転員がプラントの診断中に思考過程を発話する機能を構築する。この機能により、仮想運転員が制御盤を操作する様子を観察し、同時に仮想運転員が考えている内容を理解することが可能となり、原子力発電プラントの異常診断の過程をより深く理解できるようになると期待される。

5. 仮想運転員は実際の運転員の運転操作時の動作に近い自然な動作で異常診断と対応操作を行う。

訓練生がプラントの異常診断と対応操作を行う様子を正しく把握するためには、仮想運転員が実際の運転員の運転操作時の動作に近い動作で異常診断と対応操作を行う必要があると考えられる。そこで本研究では、3次元モーションキャプチャシステムを用いて実際の人間の動作を計測し、仮想運転員の動作データベースを構築する。そして、SEAMAID が生成する仮想運転員の行動に関する情報をもとに、必要に応じて動作データを変形し、3次元映像として仮想運転員の動作を訓練生に提示するものとする。

本研究では、以上に述べた5つの機能に重点をおいて VENUS を開発した。なお、訓練生が自ら制御盤を操作し、その結果を体験できる機能は、訓練生が効果的に訓練を進めるためには必要な機能であると考えられるが、2.3節で述べたように、既にそのような機能を備えた訓練システムは開発されているため、本研究ではこのような機能は構築せず、訓練生が仮想運転員の作業の様子を観察する機能に重点をおいて研究を進めた。

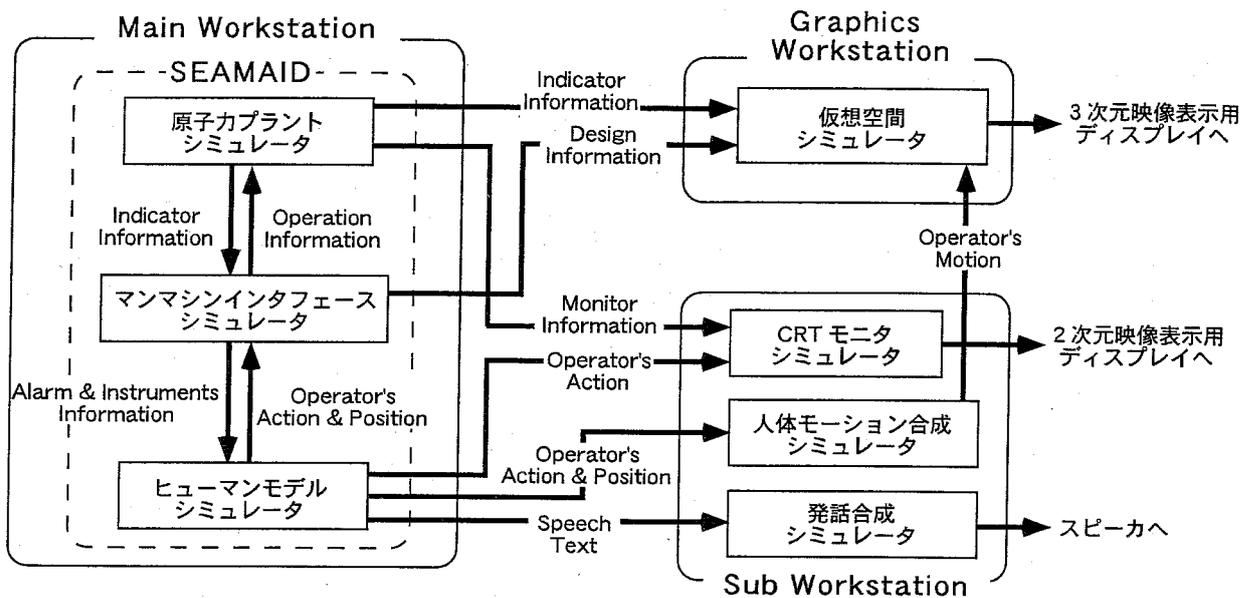


Fig. 5.1: VENUS のソフトウェア構成

5.3 VENUS の概要

本研究では、VENUS を、システム全体を複数のシミュレータに分けて実装する、分散型シミュレーションシステムとして設計した。これにより、大量の計算を要する大規模なシミュレーションを、比較的安価な計算機を複数使用することにより負荷を分散させて実行できるため、コストを削減でき、また、システムも容易に拡張できる利点がある。以下では、開発した VENUS の概要を述べる。

5.3.1 ソフトウェア構成

VENUS のソフトウェア構成を Fig. 5.1 に示す。VENUS は、SEAMAID、人体モーション合成シミュレータ、仮想空間シミュレータ、CRT モニタシミュレータ、発話合成シミュレータで構成される。

SEAMAID は、今回、仮想運転員の頭脳として採用した既開発のシミュレータで、異常事象を模擬できる原子カプラントシミュレータ、制御盤の設計情報を管理するマンマシンインタフェースシミュレータ、仮想運転員の行動と思考過程を模擬するヒューマンモデルシミュレータで構成される。VENUS を実現するために必要な原子力発電プラントの設計情報、仮想運転員の行動や思考過程に関する情報は、全てこの SEAMAID から得ることができる。すなわち、本研究はこの SEAMAID のシミュレーション結果を人工現実感技術を用いて総合的に情報可視化することにより、体験型教育システムを開発することになる。

5.3 VENUS の概要

SEAMAID の詳細は 5.4 節で述べる。

人体モーション合成シミュレータは、ヒューマンモデルシミュレータから仮想運転員が移動する先の制御室内の位置や、操作する機器の種類、機器の操作の仕方等の情報を受け取り、仮想運転員の人間らしい自然な動作を合成し、結果を仮想空間シミュレータに送る。人体モーション合成シミュレータの詳細は 5.5 節で述べる。

仮想空間シミュレータは、原子力プラントシミュレータ、マンマシンインタフェースシミュレータ、人体モーション合成シミュレータから、それぞれ、原子力発電プラントの現在の状態に関する情報、制御盤の設計情報、仮想運転員の位置と姿勢に関する情報を受け取り、制御室と仮想運転員を 3 次元映像として描画する。仮想空間シミュレータの詳細は 5.6 節で述べる。

CRT モニタシミュレータは、制御盤上に配置される機器の 1 つである「CRT モニタ」と呼ばれる機器に表示される、原子力発電プラントの状態に関する詳細な情報を、仮想空間シミュレータが 3 次元映像として描画する代わりに、2 次元映像として訓練生に提示し、仮想空間シミュレータの計算負荷を軽減させる働きをする。CRT モニタシミュレータの詳細は 5.7.1 項で述べる。

発話合成シミュレータは、ヒューマンモデルシミュレータから仮想運転員の思考過程を表す発話内容を受け取り、仮想運転員の音声を合成するシミュレータである。発話合成シミュレータの詳細は 5.7.2 項で述べる。

5.3.2 ハードウェア構成

VENUS のハードウェア構成を Fig. 5.2 に示す。VENUS の計測サブシステムは 3 次元マウスで構成され、表示サブシステムは 3 次元映像を提示するためのディスプレイ、2 次元映像を表示するためのディスプレイ、立体視するための CrystalEyes、仮想運転員の発話を提示するためのスピーカから構成される。なお、今回の訓練システムでは、訓練生が 2 種類のディスプレイを同時に眺める必要があるために、HMD を表示デバイスとして用いることは困難である。また、VENUS のシミュレーションサブシステムは、SEAMAID を実行するための Main Workstation、CRT モニタシミュレータ、人体モーション合成シミュレータ、発話合成シミュレータの 3 つのシミュレータを実行するための Sub Workstation、仮想空間シミュレータを実行するための Graphics Workstation の 3 つの Workstation から構成され、各 Workstation 間はネットワークにより繋がれている。

Fig. 5.2 では、Sub Workstation と Graphics Workstation が共に 1 台ずつ使用されているが、VENUS は、分散型シミュレーションシステムとして設計しているため、容易にシステムを拡張できる。例えば、Graphics Workstation を新たに用意し、各 Graphics Workstation 上で仮想空間シミュレータを実行することにより、既存の Graphics Workstation 上に表示する 3 次元映像の画質を落とすことなく、同時に複数の視点から仮想制御室を眺める環境を実現できる。これにより、1 人の訓練生が仮想運転員の視点からの映像を立体視しながら眺めると同時に、別の訓練生が仮想運転員の作業の様子を別の方向から眺めることも可能である。

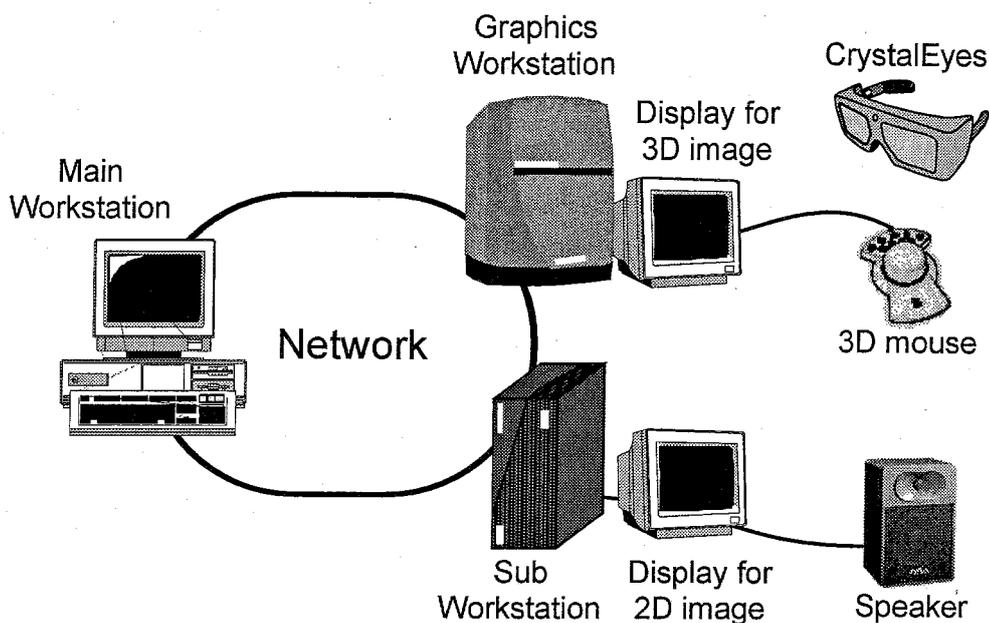


Fig. 5.2: VENUS のハードウェア構成

5.4 SEAMAID

SEAMAID は、計算機シミュレーションにより、原子力発電プラントの制御盤のインタフェースを評価するために開発されたシステムで、原子力発電プラントの異常発生時におけるプラント運転員の行動と思考過程を模擬する機能を持つ。

SEAMAID は、Fig. 5.3に示すように、原子カプラントシミュレータ、マンマシンインタフェースシミュレータ、ヒューマンモデルシミュレータの3つのシミュレータで構成される [1, 2]。

原子カプラントシミュレータは、3ループ PWR 型のプラントシミュレータで、様々なプラントの異常事象をリアルタイムで模擬できるシミュレータである。また、マンマシンインタフェースシミュレータは、主にプラントの制御盤の設計情報を管理し、運転員の情報処理行動を模擬するヒューマンモデルシミュレータとプラントシミュレータの間でプラント情報やプラント操作情報の橋渡しを行う。ヒューマンモデルシミュレータは、Reason の提唱するモデル [3] を発展させた運転員認知モデルをフレームワークとし、実際のプラント運転員がプラントの異常事象を診断する際の行動を計測した実験の結果に基づき構築されている [4]。SEAMAID での運転員認知モデルは、人間の情報処理を行う記憶領域を、意識の当たらない周辺記憶領域 (PWM : Peripheral Working Memory)、意識的な情報処理が行える焦点記憶領域 (FWM : Focal Working Memory)、長期記憶に相当する知識ベース (KB : Knowledge Base) から構成し、外界から知覚系を経て入力された情報は、一旦 PWM に入力され、その情報の目立ち度、現在処理を行っている情報との文脈一致度、その情報

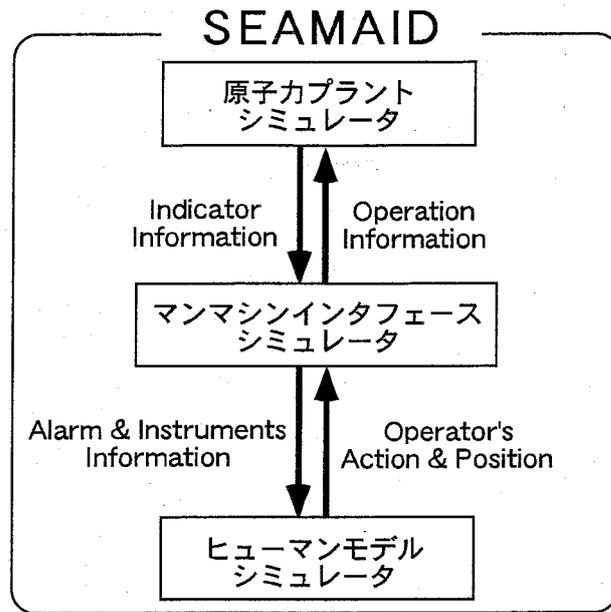


Fig. 5.3: SEAMAID のシミュレーション部のシステム構成

を用いる頻度、意識焦点の当たっている情報との類似度等に応じて、FWMに入力される。そしてFWM内の内部状態がKBに入力され、KB内部で、検索対象との類似性及び、呼び出し頻度に応じて活性化された要素がPWMに出力される。このように人間の情報処理をPWM→FWM→KB→PWMというループで模擬している [1]。

具体的にはヒューマンモデルシミュレータは Fig. 5.4に示すような流れで異常診断を行う。以下に、その内容を時系列順に説明する。

- 手順1 警報の発生、またはパラメータの異常変動によって、異常の発生を検知する。この時の警報やパラメータ逸脱を第一症候と呼ぶ。
- 手順2 原子力発電プラントの異常事象に関する知識ベースを用いて、第一症候に応じた異常仮説を立てる。
- 手順3 立てられた異常仮説が発生した場合に変動するパラメータ群（関連パラメータと呼ぶ）とその変動の傾向を予測する。
- 手順4 手順3で予測した関連パラメータの実際の状態を確認する。
- 手順5 確認したパラメータの状態と当初の予測とを比較し、その一致・不一致によって、思考中の仮説の確信度（仮説に対する自信の度合い）を増減させる。
- 手順6 思考中の仮説の確信度によって、その仮説を棄却するか、採用するか、または更なるパラメータの参照を行うかを定める。更なるパラメータの参照を行う場合には、手順4に戻る。仮説を棄却する場合には、手順2に戻る。

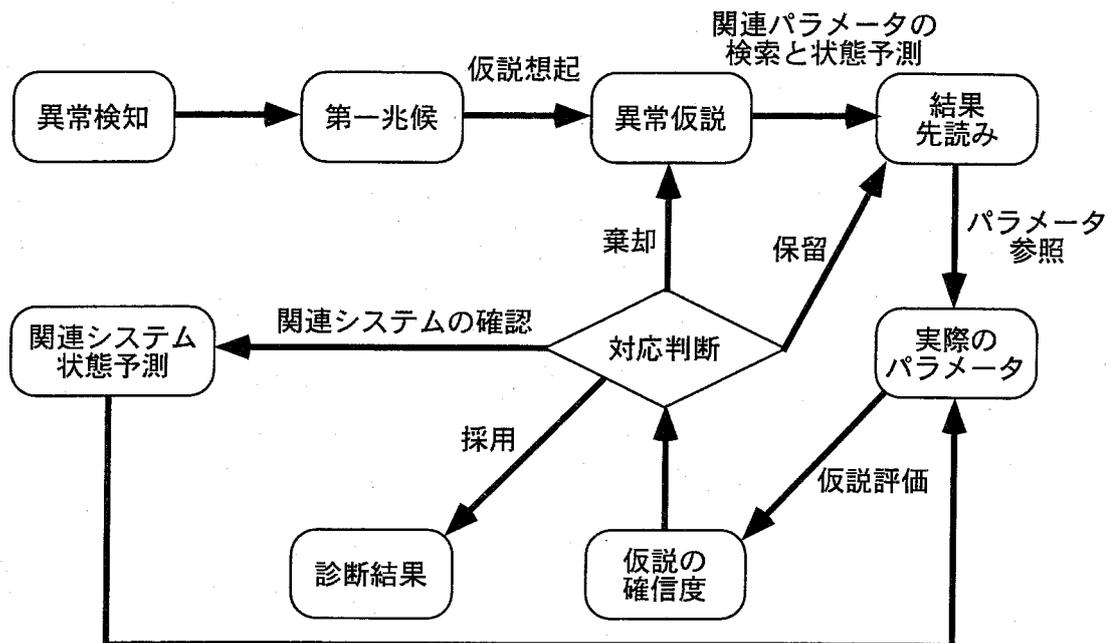


Fig. 5.4: ヒューマンモデルの異常診断過程

手順7 仮説を採用する場合には、他の異常事象が起こっていないことを確認する（異常仮説成立の十分条件確認と呼ぶ）。

手順8 異常仮説の十分条件を確認した後、その異常仮説を診断結果として出力する。

以上のようなヒューマンモデルシミュレータは、運転員の行動と思考過程をよく再現できることが確認されている [5]。VENUS では、ヒューマンモデルシミュレータにおいて、次に移動すべき場所や次に操作すべき機器、発話すべき内容などが生成され、人体モーション合成シミュレータや、発話合成シミュレータ等に送られる。なお、ヒューマンモデルシミュレータが次に操作すべき機器を決定する場合、対象の機器に仮想運転員の手が届くかどうか判断され、手が届かない場合は、まず、手が届く範囲に仮想運転員を移動させてから、対象の機器を操作する指示を人体モーション合成シミュレータに送る。従って、次に述べる人体モーション合成シミュレータでは、操作対象の機器に仮想運転員の手が届くかどうかの判断を行う必要はない。

5.5 人体モーション合成シミュレータ

人体モーション合成シミュレータは、ヒューマンモデルシミュレータから、仮想制御室内の移動する位置、操作する機器、操作の仕方といった情報を受け取り、仮想運転員の動

作を合成し、結果を仮想空間シミュレータに送る。本節では、実際の作業に近い仮想運転員の動作を3次元映像として合成するための人体モーションの合成方法について述べる。

5.5.1 人体モーションの合成技術

現在、計算機を用いて人の動作を3次元映像として合成する方法は、主に次の3つの方法が使用されている [6, 7]。

- (1) 3次元モーションキャプチャシステムを用いて実際の人動きを計測し、3次元映像として再生する [8]。
- (2) 計算機を用いて、体の形状、筋肉の強さ等を考慮した正確なシミュレーションを行い、3次元映像を合成する [9, 10]。
- (3) 人の一連の動きの内、重要となる姿勢だけをアニメータが作成し、それらの姿勢の間を計算機を用いて補間する [11]。

(1)の方法は、比較的容易に利用でき、また十分自然な3次元映像を作成することが可能であるため、映画やコンピュータゲームの世界で幅広く利用されている。しかしこの方法では、合成したい動作ごとに、実際に人がその動作を演じ、計測しなければならない。従って、合成する必要がある動作の種類が少ない場合に適した方法である。(2)の方法は、様々な人の動作の合成が可能であるが、現状では合成した3次元映像は実際の人動きと比べて自然ではなく、また、合成する際に膨大な計算が必要であり、リアルタイムに人の動作を表示する必要があるシステムでは使用することが難しい。(3)の方法では、多種多様な動作を十分に自然な動作として作成することも可能であるが、非常に多くの時間と労力が必要であり、制作コストが最も高い方法である。このように、現時点では、様々な人の動作を3次元映像として簡単に合成できる手法は開発されておらず、3次元映像を使用する用途に応じて適切な合成方法を選択することが重要である。

本研究では、訓練生に仮想運転員が作業している様子を観察させ、プラントの運転操作をどのように行うのかを理解させることが目的であるため、VENUSで合成する仮想運転員の動作は、実際の人間と見間違えるほど自然である必要はなく、訓練生が見て違和感を感じない程度に自然であることで十分であると考えた。ここで、「訓練生が見て違和感を感じない程度に自然な動作」とは、具体的には以下の条件を満たす動作であると考えられる。

条件1 仮想運転員の各関節の可動範囲が、実際の人各関節の可動範囲を逸脱しないこと

条件2 仮想運転員の各部位の動作速度が、実際の人に可能な動作速度を越えていないこと

条件3 仮想運転員の各部位の速度変化が、実際の人に可能な速度変化を越えていないこと

実際に仮想運転員の動作を合成する場合、これらの条件を満たしているかどうかを判断する方法が必要となるが、これらの条件を定量的に評価するためには、実際に人の動作を

5. 人工現実感技術を用いた原子力プラント 運転の体験型教育システム VENUS の開発

多数計測し、人の関節の可動範囲や可動速度等を求める必要があり、多くの労力が必要であると考えられる。そこで本研究では、上記3つの条件を考慮しながら仮想運転員の動作の合成方法を考案するが、合成した動作の評価は、人の主観的な評価、すなわち、人が実際に合成した動作を観察することによって、自然であるか否かを判断し、不都合があればそれを修正するものとした。具体的には、本研究では、仮想運転員の動作が、制御室内の移動と制御盤の操作に限定されることから、合成する必要がある動作の種類が比較的少なく済むと判断し、上記(1)の方法を基本として、以下の手順で仮想運転員の動作を合成する。

手順1 人の動作の計測

3次元モーションキャプチャシステムを用いて人の動作を計測し、計測誤差等の補正を行う。人の動作の計測方法については、5.5.4項で述べる。

手順2 動作データベースの作成

手順1で得られた計測データから必要な部分を切り出し、必要に応じて変形を行い、仮想運転員の動作を合成する際に使用するデータベース(動作データベース)を作成する。動作データベースの作成方法については、5.5.5項で述べる。

手順3 動作の合成

動作データベースから目的とする動作に最も近い動作データを選択し、それを変形して目的の動作を合成する。動作の合成方法については、5.5.6項及び5.5.7項で述べる。

次項以降では、まず、VENUSで合成する必要がある仮想運転員の動作を分類する。その後、仮想運転員の人体モデルについて述べた後、人の動作を計測する方法と計測したデータから動作データベースを作成する方法について述べ、本節の最後に、動作データベースを用いて仮想運転員の動作を合成する方法について述べる。

5.5.2 仮想運転員の動作の分類

本項では、VENUSで合成する必要がある仮想運転員の動作を分類する。原子力発電プラントの制御室において、運転員が制御盤上に配置された機器を操作する場合、まず、操作する機器に手が届く地点まで移動し、次に機器を操作する。従って、VENUSで合成する必要がある仮想運転員の動作は、「制御室内の移動」と「機器の操作」の大きく2つに分けることができる。

制御室内の移動

VENUSでは、仮想運転員はFig. 5.5に示す制御室内を移動する。従って、仮想運転員が、ある制御盤の前から別の制御盤の前に移動する際に、その間に障害物は無く、仮想運転員の移動の経路は直線のみ限定される。

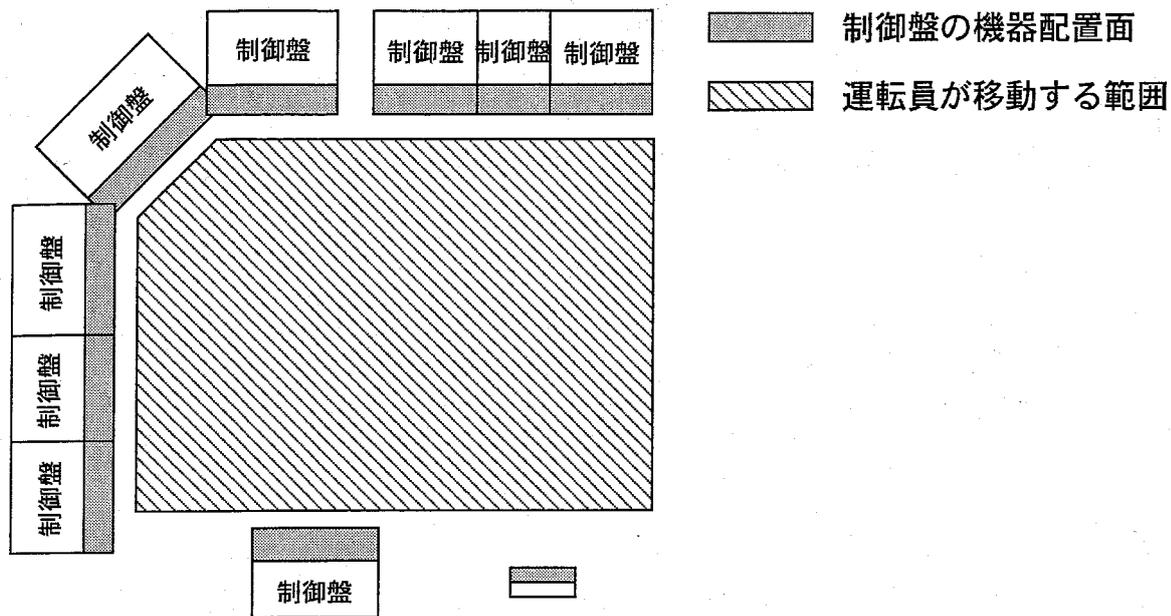


Fig. 5.5: 仮想制御室の制御盤の配置と仮想運転員の移動範囲

人の歩行動作を考えた場合、その歩数が1歩もしくは2歩の場合は、同じ動作が繰り返されることはないが、その歩数が3歩以上の場合は、途中、直進の動作が繰り返されると捉えることができる。そこで本研究では、仮想運転員が2つの地点間を移動する時の動作を、距離が短い場合（歩数が1歩もしくは2歩の場合）と長い場合（歩数が3歩以上の場合）に分けて考える。なお、仮想運転員は歩行の前後では直立の姿勢をとっているものとする。

- 距離が短い場合（1歩もしくは2歩の場合）

1歩で目的の地点にたどり着くということは、まず直立の姿勢から片足を踏み出した段階で、その着地点がすでに目的の地点であるということである。そして、残っているもう片方の足は最初に踏み出した足に添えるように着地する。2歩で目的の地点にたどり着く場合も同様に、2歩目に踏み出した足の着地点が目的の地点になる。

次に移動方向と目的の地点で向いている方向に応じて歩行動作を分類する。ここで移動方向とは、最初の直立の姿勢で体全体が向いている方向のベクトルと、出発点から目的の地点までのベクトルとの相対的な角度のことである。Fig. 5.6のAに示すように、ここでは移動方向は -180 度から 180 度までの範囲となる。また、目的の地点で向いている方向とは、Fig. 5.6のBに示すように、歩行動作が終わった時点で体全体が向いている方向のベクトルと移動方向との相対的な角度であり、同様に -180 度から 180 度までの範囲となる。

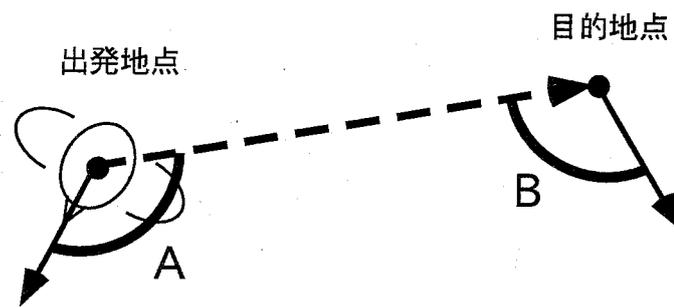


Fig. 5.6: 移動の方向と目的地点で向いている方向

以上の考察により、距離が短い場合の歩行動作としては、1歩と2歩のそれぞれについて、移動方向と、目的地点で向いている方向に応じて、個々に動作データを作成する必要があることが分かる。ただし、 -180 度から 0 度までの動作データは、 180 度から 0 度までの動作データの体を反転させたものと考えられることができるので、作成するのは移動方向と目的地点で向いている方向、共に、 $0\sim 180$ 度の動作データのみとなる。

- 距離が長い場合（3歩以上の場合）

人が3歩以上で移動する場合、その一連の動作は、まず移動方向に体を向けながら片方の足を振り出す「歩き始め」の動作、その後、移動距離に応じた回数、左右の足を前後に振り出す「直進」の動作、最後に、目的地点、方向に体を静止させる「歩き終わり」の動作の3つの動作から構成されると捉えることができる。以下では、歩き始め、直進、歩き終わりの3つの動作についてそれぞれ説明する。

- 歩き始め

本研究で「歩き始め」とは、直立の姿勢から移動方向に向かって足を踏み出しつつ体を回転させ、2歩歩いた後に、体の向いている方向が移動方向と一致し、体の姿勢が直進の動作の始めの姿勢と一致するまでの動作とする。VENUSでは、仮想運転員が全ての方向に対して移動する可能性があるが、 -180 度から 0 度までの動作データは、 180 度から 0 度までの動作データの体を反転させたものと考えられることができるため、「歩き始め」用の動作データとして、 0 度から 180 度の各方向に歩き出す動作データを作成する必要がある。

- 直進

人が直進する動作は、片足を踏み出す動作を左右逆にしながら交互に繰り返す動作と捉えることができる。本研究では、この交互に繰り返される一連の動作の内、後方の足だけが地面と接触している状態から、前方の足が着地し、体が前進するにつれて足の前後が入れ替わり、そして、最初に後方にあった足が前に振り出されるまでの動作を「直進の単位動作」と呼ぶ。仮想運転員が直進す

る動作を合成するには、右足が最初に前に来る直進の単位動作と、左足が最初に前に来る直進の単位動作が必要となるが、これら2つの動作は左右対称であるため、実際には右足が最初に前に来る直進の単位動作の動作データのみを作成する。

- 歩き終わり

人がある程度長い距離を歩く場合、歩き始めや途中の直進の段階では、目的の地点に向かって歩くことを意識しているだけで歩幅の調整は行っておらず、歩き終わりの段階で初めて歩幅の調整を行っていると考えられる。また、最終的に体を向ける方向に関しても、歩き終わりの段階で、方向の調整を行っていると考えられる。また、人の歩行動作では、その移動距離に応じて、右足で目的の地点に着地し、左足を添える場合と、左足で目的の地点に着地し、右足を添える場合の2つの場合があると考えられる。従って、仮想運転員の歩き終わりの歩行動作を合成する場合には、歩き終わりの方向に応じた0度から180度までの動作データを、右足を最後に添える動作と、左足を最後に添える動作の両方の動作に対して作成する必要がある。

機器の操作

VENUSでは、仮想運転員が機器を操作する動作として、ボタンを押す動作とレバーを前後にスライドさせる動作の2種類の動作を合成する。

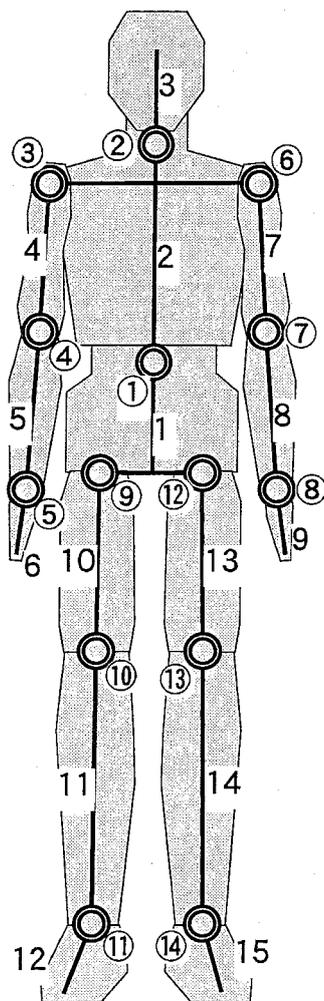
まず、レバーを操作する動作は、「レバーまで腕を移動させる動作」と「レバーを前後に移動させる動作」の2つの動作が連続したものと捉えることができる。一方、ボタンを操作する動作は、「ボタンまで腕を移動させる動作」のみで構成されると捉えることができる。従って、機器の操作に関して、VENUSで仮想運転員の動作として合成すべき動作は以下の2種類となる。

1. ある地点から別の地点へ手を移動させる動作。これには直立の状態からある地点へ腕を伸ばす動作も含まれる。
2. レバーを前後にスライドさせる動作。

5.5.3 人体モデルの構成

VENUSで用いる仮想運転員の人体モデルを Fig. 5.7に示す。人間の可動関節の総数は100個に近いが、仮想運転員の動きを合成する際、これら全ての関節の動きを再現するのは困難である。従って、本研究では、仮想運転員が有する関節を決定するにあたり、全身の関節の中から自然な動きを表現するのに最低限必要と思われる14箇所の関節を選択した。Fig. 5.7では◎印が関節の位置を表す。

各関節で接続されている2つの体の部位（リンク）のうち、片方のリンクがもう片方のリンクに対して、関節を中心とした x,y,z 軸回りの回転の3つの自由度を持つ。また、仮想運転員の人体モデルを構成する15個のリンクに、Fig. 5.8に示すように、腰を頂点とし



リンク名称

- 1:腰(Hip)
- 2:胴(Body)
- 3:頭(Head)
- 4:右上腕(RArm0)
- 5:右下腕(RArm1)
- 6:右手の平(RArm2)
- 7:左上腕(LArm0)
- 8:左下腕(LArm1)
- 9:左手の平(LArm2)
- 10:右太股(RLeg0)
- 11:右脛(RLeg1)
- 12:右足(RLeg2)
- 13:左太股(LLeg0)
- 14:左脛(LLeg1)
- 15:左足(LLeg2)

関節名称

- ①:腰の関節
- ②:首
- ③:右肩
- ④:右肘
- ⑤:右手首
- ⑥:左肩
- ⑦:左肘
- ⑧:左手首
- ⑨:右股関節
- ⑩:右膝
- ⑪:右足首
- ⑫:左股関節
- ⑬:左膝
- ⑭:左足首

Fig. 5.7: 仮想運転員の人体モデル

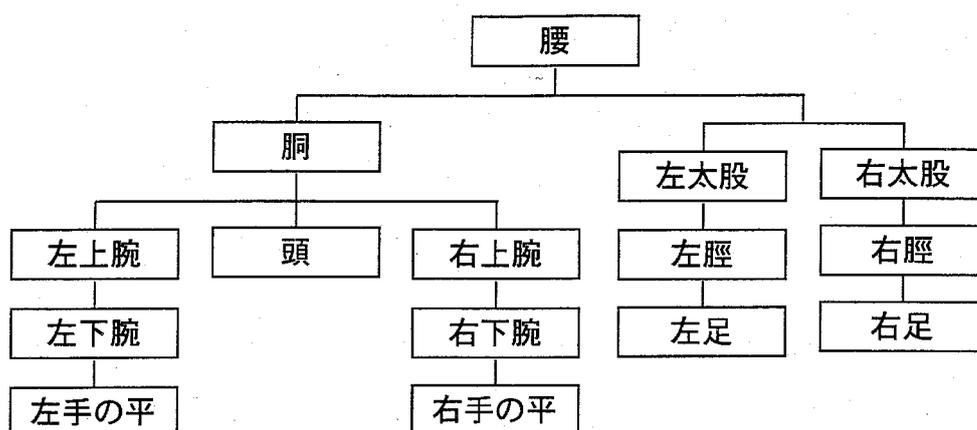


Fig. 5.8: 人体モデルの階層構造

た階層構造を定める。さらに、各リンクに対し、上位リンクとの接続部分である関節を原点とし、常にリンクと同じ位置関係を保つローカル座標系を Fig. 5.9 に示すように定める。これにより、仮想制御室に対して定めたワールド座標系から腰のリンクのローカル座標系への座標変換と、隣接する2つのリンクのローカル座標系間の座標変換（合計15個の座標変換）を用いることにより、人体モデルの姿勢を記述できる。また、ワールド座標系のx軸とy軸により決定される平面を人体モデルの動きを合成する際の水平面（地面）とし、z軸方向を地面から上向き垂直方向とする。

5.5.4 人の動作の計測と計測データの補正

本項では、3次元モーションキャプチャシステムを用いて人の動作を計測する方法、計測したデータの雑音を除去する方法、モーションキャプチャシステムの磁気センサの取り付け位置誤差を補正する方法について述べる。

人の動作の計測

動作データを収集する際のシステム構成を Fig. 5.10 に示す。本研究では、仮想運転員の15個のリンクの動きを合成するために、磁気センサを、人間の頭、胴、腰、左右上腕、左右下腕、左右手の平、左右太股、左右脛、左右足の合計15箇所に取り付けた。なお、計測により得られるデータには磁気センサの位置と方向があるが、本研究では仮想運転員の動作を各リンクの方向をもとに合成するため、計測データとしては方向のみを用いる。また磁気方式のモーションキャプチャシステムは周囲にある磁性体の影響を受けやすいため、計測は屋外で行った。

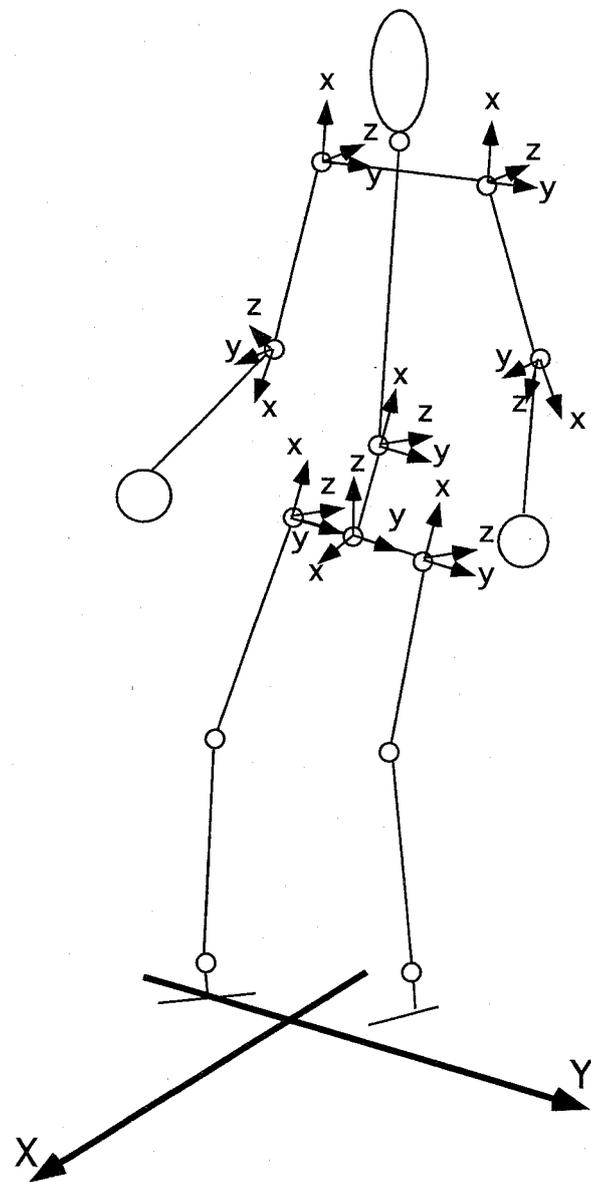


Fig. 5.9: 人体モデルの各部位のローカル座標系

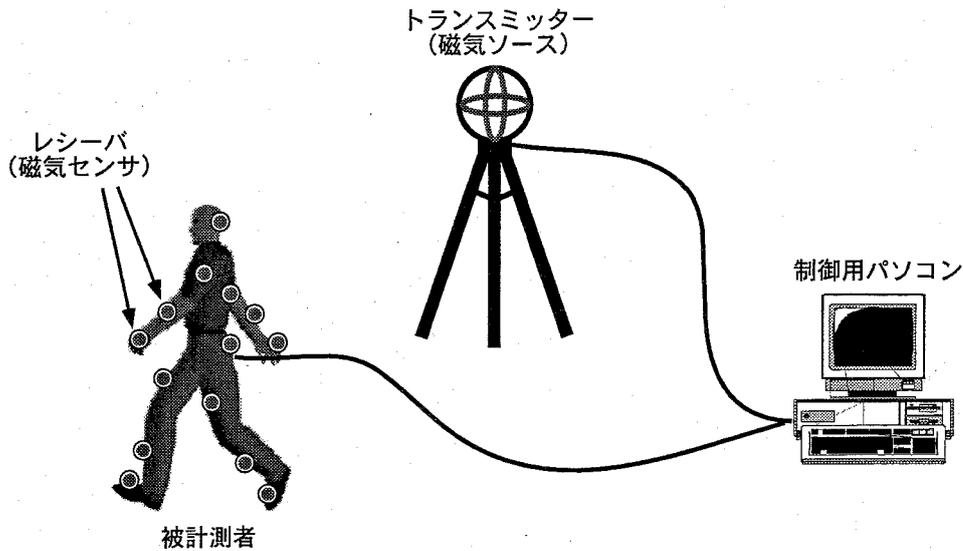


Fig. 5.10: 動作データを収集する際のシステム構成

計測データの雑音の除去

人の動作を計測して得られたデータには、使用したモーションキャプチャシステムの性能上、データの不連続点や、測定誤差等の雑音が含まれているため、計測データの平滑化を行う必要がある。前述のように、今回の人体モーションの合成には、磁気センサの方向のみのデータを用いるので、平滑化は、得られた計測データのYaw、Pitch、Rollの角度のそれぞれについて移動平均をとることとする。実際に幾つかの計測データについて3点、5点、7点の移動平均をとった後の計測データを用いて、人の動作を仮想空間内に合成したところ、3点では十分に雑音が除去されなかったが、5点の移動平均を行うと十分に滑らかな動作となった。また7点の移動平均をとった計測データで合成した人の動作は、5点のものとはほとんど変化がなかった。そこで、本研究では、雑音除去のために5点の移動平均を用いた。

磁気センサの取り付け位置誤差の補正

人に磁気センサを取り付けて動作を計測する場合、磁気センサの方向と取り付け部位の方向を完全に一致させることは難しく、実際にはずれが生じる。従って、このずれを何らかの方法で補正する必要がある。本研究では、動作を計測する際に被計測者に直立の姿勢になってもらい、その時に得られた計測データが直立時の姿勢を表すと仮定して計測データの補正を行う。以下にその補正方法を述べる。

まず始めに腰の姿勢を補正する方法を説明する。腰は5.5.3項で述べた人体モデルの階層構造で頂点になっているので、地面に対して定めたワールド座標系から見た腰の姿勢を表す回転行列を求める必要がある。ワールド座標系から見た、腰の姿勢を回転行列で表した

5. 人工現実感技術を用いた原子カプラント 運転の体験型教育システム VENUS の開発

ものを ${}^W R_{Hip}$ 、腰に張り付けた磁気センサの姿勢を回転行列で表したものを ${}^W R_{HipSensor}$ 、腰から腰に張り付けた磁気センサへの姿勢のずれ（取り付け位置誤差）を回転行列で表したものを ${}^W R_{Hip \rightarrow Sensor}$ とすると次式が成り立つ。

$${}^W R_{Hip} \cdot {}^W R_{Hip \rightarrow Sensor} = {}^W R_{HipSensor} \quad (5.1)$$

従って、被計測者が直立姿勢を取ったときの計測データ（腰に張り付けた磁気センサの姿勢）を ${}^W R_{HipSensor}^{Calib}$ 、この時の腰の姿勢を ${}^W R_{Hip}^{Calib}$ とすると、この時、 ${}^W R_{Hip}^{Calib}$ は直立時の腰の姿勢を表す値になると仮定して、位置誤差 ${}^W R_{Hip \rightarrow Sensor}$ は、次式で定数として求めることができる。

$${}^W R_{Hip \rightarrow Sensor} = ({}^W R_{Hip}^{Calib})^{-1} \cdot {}^W R_{HipSensor}^{Calib} \quad (5.2)$$

よって、式(5.1)及び式(5.2)により、補正後の腰の姿勢を次式で求めることができる。

$${}^W R_{Hip} = {}^W R_{HipSensor} \cdot ({}^W R_{HipSensor}^{Calib})^{-1} \cdot {}^W R_{Hip}^{Calib} \quad (5.3)$$

次に胴の姿勢を補正する方法を説明する。胴は、5.5.3項で述べた人体モデルの階層構造で腰の1つ下の階層になっているので、腰のローカル座標系から見た腰から胴への姿勢変化を求める必要がある。まず、式(5.3)のHipをBodyに変えることで

$${}^W R_{Body} = {}^W R_{BodySensor} \cdot ({}^W R_{BodySensor}^{Calib})^{-1} \cdot {}^W R_{Body}^{Calib} \quad (5.4)$$

が得られる。ここで、腰のローカル座標系から見た腰から胴への姿勢変化を ${}^{Hip} R_{Hip \rightarrow Body}$ とすると次式が成り立つ。

$${}^{Hip} R_{Hip \rightarrow Body} = {}^W R_{Body} \cdot ({}^W R_{Hip})^{-1} \quad (5.5)$$

${}^W R_{Hip}$ と ${}^W R_{Body}$ は、式(5.3)及び式(5.4)で求められるので、 ${}^{Hip} R_{Hip \rightarrow Body}$ を求めることができる。他のリンクについても同様に求めることができる。

5.5.5 動作データベースの作成

本項では、5.5.4項で得られた計測データを利用して、5.5.2項で述べた、仮想運転員の動作を合成するために必要となる動作データを作成する方法について述べる。

移動用の動作データの作成

移動用としては、前述のように、直進、歩き始め、歩き終わり、1歩もしくは2歩の歩行に対する動作データを作成する。

- 直進の動作データの作成

5.5.4項で述べた方法で補正した計測データから、一步分のデータを切り出す。歩行動作を合成する場合、直進の動作データを左右逆にしながら連続して表示するが、通常、切り出したデータは、最初の姿勢と最後の姿勢が完全には一致しないので、一致するようにデータを修正する。ここでは、最初の姿勢を基準にして、最後の姿勢を最初の姿勢の左右を反転させたものに一致させることにする。

最後の姿勢における Fig. 5.8に示した階層構造の親子関係にある2つのリンク A 及び B の間の座標変換を ${}^W R_{Last}$ 、最初の姿勢の左右を反転させた後の姿勢における同じリンク間の座標変換を ${}^W R_{Rev_First}$ とする。また、回転行列 R を $M(Yaw, Pitch, Roll)$ で表すことにする。ここで、 Yaw 、 $Pitch$ 、 $Roll$ はそれぞれ z 軸、 y 軸、 x 軸回りの回転角を表す。 ${}^W R_{Last}$ から ${}^W R_{Rev_First}$ への姿勢変化を表す回転行列 ${}^L R_F$ を求めると次式のようになる。

$${}^L R_F = ({}^W R_{Last})^{-1} \cdot {}^W R_{Rev_First} = M(Y_{LF}, P_{LF}, R_{LF}) \quad (5.6)$$

次に、切り出したデータに含まれる姿勢の数を N 、 x 番目の姿勢のリンク A 及び B の間の座標変換を R_{x-th} とし、修正後の直進の歩行におけるリンク A 及び B の間の座標変換 R_{x-th}^{Modify} を以下のように定義する。

$$R_{x-th}^{Modify} = R_{x-th} \cdot M\left(\frac{Y_{LF}}{N}x, \frac{P_{LF}}{N}x, \frac{R_{LF}}{N}x\right) \quad (5.7)$$

最初の姿勢では $x = 0$ であり、 $R_{0-th}^{Modify} = R_{0-th}$ となり、最後の姿勢では、 $x = N$ であり、

$$R_{N-th}^{Modify} = R_{N-th} \cdot M(Y_{LF}, P_{LF}, R_{LF}) = {}^W R_{Rev_First} \quad (5.8)$$

となり、式 (5.7) によって、最後の姿勢を最初の姿勢を左右反転させたものに一致させることができる。こうして得られたデータを直進の動作データとする。

● 歩き始めの動作データの作成

直進の動作データを作成した場合と同様に、補正した計測データから、2歩目までのデータを切り出す。作成する歩き始めの動作データの最初の姿勢は直立でなくてはならない。また、歩き始めの動作データの最後の姿勢は、直進の動作データの最初の姿勢を、移動方向分だけ回転させた姿勢と一致してはならない。ここでは、歩き始めの動作データの最後の姿勢を、直進の動作データの最初の姿勢に一致するように修正する。

歩き始めの最初の姿勢における親子関係にある2つのリンク A 及び B の間の座標変換を ${}^W R_{First}^{Start}$ 、最後の姿勢における座標変換を ${}^W R_{Last}^{Start}$ 、直進の動作の最初の姿勢を移動方向だけ回転させた姿勢におけるリンク A 及び B の間の座標変換を ${}^W R_{First}^{Straight}$ 、直立姿勢におけるリンク A 及び B の間の座標変換を ${}^W R_{Stand}$ とする。式 (5.6) と同様に、以下の式を求める。

$$M_1(Y_F, P_F, R_F) = ({}^W R_{First}^{Start})^{-1} \cdot {}^W R_{Stand} \quad (5.9)$$

$$M_2(Y_L, P_L, R_L) = ({}^W R_{Last}^{Start})^{-1} \cdot {}^W R_{First}^{Straight} \quad (5.10)$$

そして、修正後の歩き始めの動作におけるリンク A 及び B の間の座標変換 R_{x-th}^{Modify} を次式で定義する。

$$R_{x-th}^{Modify} = R_{x-th} \cdot M \quad (5.11)$$

ただし、

$$M = (Y_F(\frac{N-x}{N})^{d1} + Y_L(\frac{x}{N})^{d2}, P_F(\frac{N-x}{N})^{d1} + P_L(\frac{x}{N})^{d2}, R_F(\frac{N-x}{N})^{d1} + R_L(\frac{x}{N})^{d2})$$

$d1$ 、 $d2$ は 1 以上の適当な整数である。 $d1$ 及び $d2$ は、修正した後の 3 次元映像を見て、最も最適だと思われる値を決定した。

● 歩き終わりの動作データの作成

歩き終わりの動作データは、歩き始めの動作データとは逆に、最後の姿勢が直立であり、最初の姿勢が直進の動作の最後の姿勢に一致してはならない。具体的には、補正した計測データから最後の直立の姿勢から 2 歩さがった姿勢までを切り出し、歩き始めの動作データを作成した時と同じ方法で動作の修正を行う。

● 1 歩または 2 歩で到達する動作データの作成

1 歩または 2 歩で到達する動作データは、最初の姿勢と最後の姿勢が共に直立でなくてはならない。修正の方法は、歩き始めの動作データを作成した時と同じである。

Table 5.1: 作成した移動用の動作データの一覧

| 動作の分類 | 各動作データの種類 | データ数 |
|-------|---|-------|
| 1歩で到着 | <ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ ・目的地点で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ | 合計 98 |
| 2歩で到着 | <ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ ・目的地点で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ | 合計 98 |
| 歩き始め | <ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ | 合計 7 |
| 直進 | <ul style="list-style-type: none"> ・1歩歩くデータ | 合計 1 |
| 歩き終り | <ul style="list-style-type: none"> ・目的地点で向いている方向について0度から180度まで30度ステップ ・最後に踏み出した足の左右 ・歩幅について普通の歩幅と小さい歩幅の2つ | 合計 28 |

以上述べた方法で作成した移動用の動作データの一覧を Table 5.1に示す。移動方向と目的地点で向いている方向については、共に動作データは30度毎に用意する。さらに変形の度合いを小さくするために、1歩と2歩で到達する動作データと歩き終わりの動作データについては、歩幅が小さいものと普通のものの2種類の動作データを用意した。

操作用の動作データの作成

運転員が機器を操作する場合、足の位置が固定されており、最初の姿勢や最後の姿勢が他の動作の姿勢と一致する必要がないため、機器を操作する動作データの作成は以下に示す手順で容易に作成できる。

- 手を移動させる動作データの作成

補正した計測データから、手が移動元の位置から移動先の位置まで移動するデータを切り出し、動作データとする。

- レバーをスライドさせる動作データの作成

補正した計測データから、レバーをスライドさせるデータを切り出し、動作データとする。

以上述べた方法で作成した操作用の動作データの一覧を Table 5.2に示す。制御盤上の代表点は格子状に横方向に3点、縦方向に3点並べた、合計9点とする。操作対象まで腕を

Table 5.2: 作成した操作用の動作データの一覧

| 動作データの分類 | 各動作データの種類 | データ数 |
|----------------|----------------------------------|-------|
| 操作対象まで腕を移動する動作 | 制御盤の格子点の9点に手がある姿勢と直立姿勢の組み合わせ間の動作 | 合計 99 |
| 操作対象を実際に操作する動作 | 制御盤の格子点の9点上にあるレバーを前後にスライドする動作 | 合計 18 |

移動する動作については直立姿勢を地点0番とし、 $0 \rightarrow 1$ 、 $0 \rightarrow 2$ 、 \dots $0 \rightarrow 9$ の動作データと $1 \rightarrow 0\dots9$ 、 $2 \rightarrow 0\dots9$ 、 \dots $9 \rightarrow 0\dots9$ の合計99個の動作データと、1~9の地点においてレバーを前後にスライドさせる動作 $9 \times 2 = 18$ 個の動作データの、全部で117個の動作データを用意する。また、後述する、体に近い位置にある機器を操作する動作を合成する際に使用する、「右肘を曲げた状態にある姿勢」を作成し、この姿勢のデータを Bend と呼ぶ。

動作データからの腰の位置の算出方法

作成した動作データを用いて3次元映像を合成する場合、仮想運転員の足の動きに応じて、描画する仮想運転員の位置を変化させる必要があるため、作成した動作データから、リンクの階層構造の頂点にある腰の位置の変化を求めなくてはならない。本研究で用いた腰の位置の計算方法を以下に示す。

手順1 左右の足のつま先及び踵に対して、腰からの相対的な鉛直方向の位置を求め、位置が最も低い点を接地部分とする。

手順2 接地部分から腰の位置へのベクトル \vec{A} を計算する。

手順3 \vec{A} にワールド座標系の原点から手順1で求めた接地部分までのベクトル \vec{B} を加えて、ワールド座標系における腰の位置を計算する。

手順4 次の姿勢の接地部分を調べ、接地部分が手順1で求めた接地部分と同じ場合は、手順2に戻る。違う場合は、手順1で求めた接地部分から新たに求めた接地部分へのベクトル \vec{C} を計算する。

手順5 手順4で求めた \vec{C} を手順3で求めた \vec{B} に加え、新たな接地部分のワールド座標系の原点からの位置を求める。

5.5.6 移動動作の合成

本項では、前項で述べた方法により作成した動作データを用いて、移動の動作を合成する方法について述べる。移動の動作を合成する場合、まず、移動の方向と距離に応じて適切な動作データを選択する。その後、選択した動作データの変形を行い、動作を合成する。

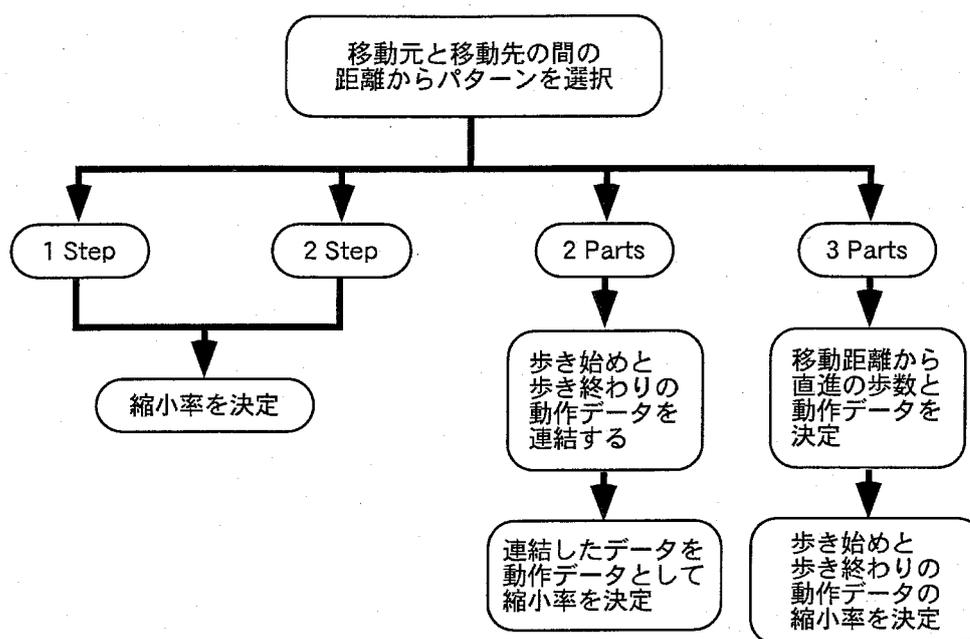


Fig. 5.11: 動作データの選択から動作データの縮小率の決定までの流れ

方向と距離に応じた動作データの選択

本研究では5.5.2項で述べたように、その移動距離が長い場合は、移動動作を1つの動作ではなく、歩き始め、直進、歩き終わりといった複数の動作を組み合わせることにより合成する。また、予め準備する動作のデータベースに関しても、その方向や移動距離によって複数の種類の動作を用意している。また、移動距離が短い場合でも、その移動距離や到達地点で人体モデルが向いている方向によって複数の動作データを用意している。

動作データの選択から動作データの移動距離の縮小率の決定までの流れを Fig. 5.11に示す。移動動作を合成するためには、まず、移動方向と距離に応じて適切な動作データを選択する必要がある。

最初に、移動元と移動先の2地点間の距離 (Len) に応じて、以下の4つのパターンに分類する。

1Step 1歩で移動先にたどり着く。

2Step 2歩で移動先にたどり着く。

2Parts 歩き始めと歩き終わりの動作データを直接連結した範囲で移動先にたどり着く。

3Parts 歩き始め、歩き終わりの間に直進のデータを含まなければ移動先にたどり着けない。

移動元と移動先の2地点間の距離に応じた具体的な分類方法を以下に述べる。

5. 人工現実感技術を用いた原子カプラント 運転の体験型教育システム VENUS の開発

1. 合成する動作の移動方向に応じて、歩き始め、歩き終わり、1歩でたどり着くもの、2歩でたどり着くものの4種類の動作データを全て選択する。具体的には、動作データは、30度ごとに用意されているので、合成する動作の移動方向の±15度以内に入る動作データを選択する。
2. 移動方向に応じて選択された、歩き始めの動作データの移動距離を Len_{First} 、普通の歩幅の歩き終わりの動作データの移動距離を Len_{LastL} とすると、

$$Len_{First} + Len_{LastL} < Len$$

の条件に当てはまる場合はこの移動動作を 3Parts のパターンとする。

3. 移動方向に応じて選択された、2歩で普通の歩幅でたどり着く動作データの移動距離を Len_{2StepL} 、小さな歩幅の歩き終わりの動作データの移動距離を Len_{LastS} とすると、

$$Len_{2StepL} < Len \leq Len_{First} + Len_{LastL}$$

の条件に当てはまる場合はこの移動動作を 2Parts のパターンとする。さらにここで、 $Len_{First} + Len_{LastS} < Len$ の場合は、歩き終わりの動作データは普通の歩幅のものを選び、そうでない場合は小さな歩幅の動作データを選択する。

4. 移動方向に応じて選択された、1歩で普通の歩幅でたどり着く動作データの移動距離を Len_{1StepL} 、2歩で小さな歩幅でたどり着く動作データの移動距離を Len_{2StepS} 、2歩で普通の歩幅でたどり着く動作データの移動距離を Len_{2StepL} とすると、

$$Len_{1StepL} < Len \leq Len_{2StepL}$$

の条件に当てはまる場合は、この移動動作を 2Step のパターンとする。ここで、 $Len_{2StepS} < Len$ の場合は、2歩で普通の歩幅でたどり着く動作データを選択する。そうでない場合は小さな歩幅の動作データを選択する。

5. 上記以外の場合は $Len \leq Len_{1StepL}$ なので、1Step のパターンとなる。1歩で小さな歩幅でたどり着く動作データの移動距離を Len_{1StepS} とすると、 $Len_{1StepS} < Len$ の場合は1歩で普通の歩幅でたどり着く動作データを選択する。そうでない場合は小さな歩幅の動作データを選択する。

選択されたパターンが 1Step、2Step の場合、動作データの縮小率は Len を、選択された動作データの移動距離で割ったものである。2Parts のパターンの場合は、歩き始めと歩き終わりのデータを連結したデータを1つの動作データと考え、 Len をその動作データの移動距離で割ったものが縮小率となる。

3Parts のパターンについてはさらに細かく考える。人は通常、移動の際に歩き終わりの部分において距離の調整を行うものと考えられるので、基本的に、移動距離を調整するの

は歩き終わりの部分とする。また、3Partsのパターンの場合、他の3つのパターンとは異なり、要求されている移動距離に応じて直進の動作が繰り返される。そのため、直進部分の歩数も求める必要がある。そこで3Partsのパターンの場合、以下のように動作データ及びその移動距離の縮小率を決定する。

1. まず、歩き始めの基本データの移動距離は変えないと考え、直進の部分の歩数を1つずつ増やしていく。そして、それによって歩き終りの部分で足の左右の着地順が変化するので、それを考慮に入れて、歩き終りの基本データは普通の歩幅のものを選ぶ。これらの合計の移動距離が Len を越えるまで直進の歩数を増やす。
2. 歩き終りの基本データの移動距離の縮小率 $ratio$ を算出する。これは、直進の移動距離を Len_M 、直進の歩数を num とすると、

$$ratio = (Len - Len_{First} - Len_M \times num) / Len_{LastL} \quad (5.12)$$

である。もし $ratio$ がある適当な値 $Rlimit$ 、例えば $Rlimit = 0.8$ よりも大きい場合はこのデータを選択する。 $Rlimit$ よりも小さい場合は、歩き終りの基本データを小さい歩幅のものに変えて $ratio$ を計算する。なぜなら、 $ratio$ が小さいということは変形の度合いが大きいということであり、普通の歩幅の基本データを採用すると動作が不自然になってしまうからである。小さい歩幅の歩き終りの基本データの移動距離を Len_{LastS} として式 (5.12) の Len_{LastL} を Len_{LastS} に置き換えて同様に計算する。 $Rlimit \leq ratio \leq 1$ の場合はこのデータを選択する。

ここで、 $ratio > 1$ の場合は直進の歩数を1つ増やす。この時、足の左右が切り替わるため、選択する基本データも変わることには注意する。そして再び上述の方法により $ratio$ を計算する。 $Rlimit \leq ratio \leq 1$ の場合はこの基本データを選択する。

3. 2. で適切な基本データを選択できなかった場合は、歩き始めの基本データの移動距離も縮小する。1. が終わった段階で算出されている直進の歩数を Num として、 $ratio1$ を以下のように計算する。

$$ratio1 = (Len - Len_M \times Num) / (Len_{First} + Len_{LastL}) \quad (5.13)$$

また、2. の最後の段階で算出されている直進の歩数、つまり、歩き終りの基本データの歩幅を小さいものとして $ratio$ が1より小さくなるように選択された直進の歩数を $Num2$ として、式 (5.13) と同様に、

$$ratio2 = (Len - Len_M \times Num2) / (Len_{First} + Len_{LastS}) \quad (5.14)$$

を計算する。 $ratio1$ と $ratio2$ を比較し、 $ratio1$ が大きい場合は歩き終りの基本データとして普通の歩幅のものを、そうでない場合は小さい歩幅のものを選ぶ。

ここまでの手順により適切な動作データ及び歩き始めと歩き終わりの部分の適切な縮小率が決定される。

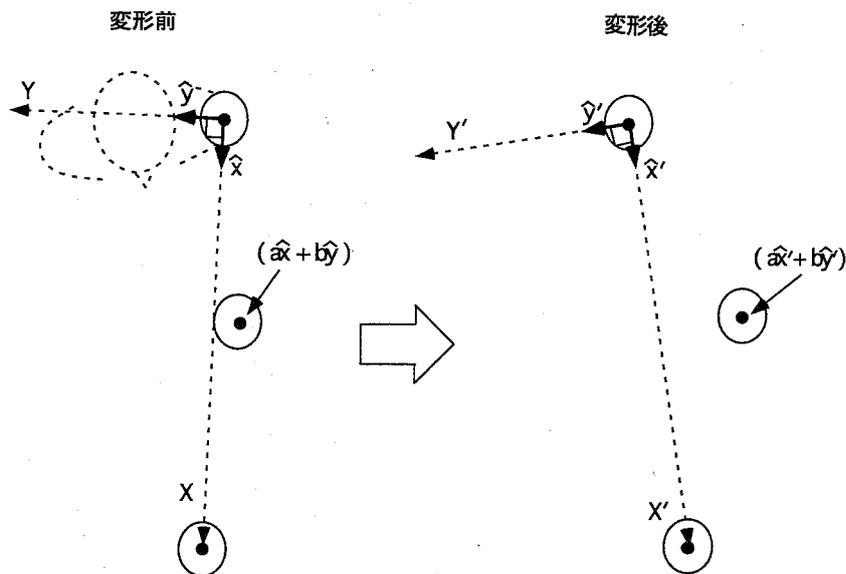


Fig. 5.12: 上半身の座標変換

動作データの变形

移動動作をその方向や距離に応じて自由に合成するためには、移動方向の調整や移動距離の縮小といった動作データの变形が必要である。このとき、上半身(腰よりも位置が上の部位、腰も含む)は、移動動作中に外界の物体と接触することがないため、その動きが自然であれば、各部位の位置をそれほど正確に考慮する必要はないが、下半身(腰よりも位置が下の部位、腰を含まない)は、足が地面と接触しているため、移動動作中に足が滑ったように見えないように動作を变形する必要がある。従って、以下では動作の变形方法を、上半身と下半身に分けて述べる。

まず、人体モデルの上半身の各部位について、注目している部位のローカル座標系に姿勢変化を与えた時の座標系を Σ_{Part} 、さらに移動方向と移動距離の变形を加えた時の座標系を Σ'_{Part} とする。 Σ_{Part} と Σ'_{Part} の関係をワールド座標系のz軸の正から負の方向へ眺めた図をFig. 5.12に示す。図中は、左上腕に注目して、その变形前のローカル座標系と变形後のローカル座標系を示している。

まず、 Σ'_{Part} の原点の座標を定める方法を説明する。以下では、特に断らない限り、ワールド座標系のx-y平面のみで考える。つまり、各座標系の原点のワールド座標系におけるz座標は同じものとする。

1. まず、Fig. 5.12の左側の図において、動作データの最初の姿勢における Σ_{Part} の原点から最後の姿勢における原点へのベクトルの方向をX軸とし、X軸方向の単位ベクトルを \hat{x} とする。次に、X軸をワールド座標系のz軸に対して90度回転させたものをY軸とし、Y軸方向の単位ベクトルを \hat{y} とする。そして移動動作中の Σ_{Part} の原点の座標をこれら2つの単位ベクトルで表したものを $(a\hat{x} + b\hat{y})$ とする。

- 次に、変形後の動作データについて、1.と同様に、最初の姿勢の Σ'_{Part} の原点から最後の姿勢の Σ'_{Part} の原点へのベクトルを X' 軸とし、 X' 軸方向の単位ベクトルを \hat{x}' とする。また X' 軸をワールド座標系の z 軸に対して90度回転させたものを Y' 軸とし、 Y' 軸方向の単位ベクトルを \hat{y}' とする。
- 以上定めた a 、 b 、 \hat{x}' 、 \hat{y}' を用いて、変形した移動動作中の Σ'_{Part} の原点の座標を $(a\hat{x}' + b\hat{y}')$ と定める。

次に Σ'_{Part} の姿勢を定める方法を述べる。 Σ'_{Part} の姿勢は、 Σ'_{Part} を定める x 軸、 y 軸、 z 軸の3軸の内、2軸の方向が定まると求めることができる。まず Σ_{Part} の z 軸の原点を基準とする単位ベクトルの先端の座標について考える。

- 動作データの最初の姿勢の Σ_{Part} の原点を始点とした z 軸方向の単位ベクトル（以下、 z 軸の単位ベクトル）の先端から最後の姿勢の z 軸の単位ベクトルの先端へのベクトルを X 軸とし、 X 軸方向の単位ベクトルを \hat{x} とする。次に X 軸をワールド座標系の z 軸に対して90度回転させたものを Y 軸とし、 Y 軸方向の単位ベクトルを \hat{y} とする。そして、移動動作中の Σ_{Part} の z 軸の単位ベクトルの先端の座標を $(a\hat{x} + b\hat{y})$ と表す。
- 変形後の動作データについても同様に X' 軸と Y' 軸を定め、 \hat{x}' 、 \hat{y}' をその単位ベクトルとする。そして、変形後の移動動作中の Σ'_{Part} の z 軸の単位ベクトルの先端の座標を、 a 、 b を用いて $(a\hat{x}' + b\hat{y}')$ と定める。
- 変形後の移動動作中の Σ'_{Part} の x 軸の単位ベクトルの先端の座標も同様に定める。
- 変形後の動作データの Σ'_{Part} の x 軸の単位ベクトルの先端の座標をワールド座標系の x - y 平面へ正射影した点を Q とする。そして、単位ベクトルの先端の座標を正射影したものが点 Q に一致する Σ'_{Part} の原点を基準とする単位ベクトルを求め、これを Σ'_{Part} の x 軸の単位ベクトルとする。
- 同様に、 Σ'_{Part} の z 軸が Σ'_{Part} の x 軸と直交していることを考慮しながら Σ'_{Part} の z 軸の単位ベクトルを求める。
- Σ'_{Part} の x 軸と z 軸の単位ベクトルから、 Σ'_{Part} の姿勢を求める。

以上のようにして、移動距離と移動方向の変形後の上半身の各部位の位置と姿勢を定める。

下肢の姿勢の決定

下肢の姿勢を決定する場合、まず、太股 (Leg0) の姿勢を決定し、次に地面 (ワールド座標系の x - y 平面) と接触している部位である足 (Leg2) の姿勢を決定し、最後に脛 (Leg1) の姿勢を決定する。なお、以下の説明で用いる各リンクのローカル座標系の x, y, z の各軸は、Fig. 5.9に示した各リンクのローカル座標系の x, y, z の各軸を表している。

5. 人工現実感技術を用いた原子カプラント 運転の体験型教育システム VENUS の開発

1. まず、太股 (Leg0) について考える。Leg0 のローカル座標系 (Σ_{Leg0}) の原点 (股関節) を、上半身の姿勢を決定した時に既に定まっている腰の姿勢を用いて求める。

次に、Leg0 の Σ_{Leg0} の x 軸を求める。Leg0 は Leg1 (脛) のローカル座標系 (Σ_{Leg1}) の原点に繋がっていることを利用し、上半身の姿勢を求めた時と同じ方法で求めた Σ_{Leg1} の原点をワールド座標系の x-y 平面へ正射影した座標と、先に求めた Leg0 の Σ_{Leg0} の原点をワールド座標系の x-y 平面へ正射影した座標を結んだベクトルを、Leg0 の Σ_{Leg0} の x 軸を x-y 平面へ正射影したものに等しいとする。

そして、Leg0 の長さを Σ_{Leg0} の x 軸を表すベクトルの長さとして、そのベクトルの先端の座標のワールド座標系の x-y 平面への正射影から Leg0 の方向を求める。 Σ_{Leg0} の z 軸に関しては、上半身の姿勢を求めた方法と同様にして求める。

以上の方法により、太股 (Leg0) の姿勢が求まる。

2. 次に足 (Leg2) の姿勢を決定する。まず、変形前の動作データの腰 (Hip) のローカル座標系 Σ_{Hip} の z 軸を、ワールド座標系の x-y 平面へ正射影したものと、変形後の動作データの腰のローカル座標系 Σ'_{Hip} の z 軸を、ワールド座標系の x-y 平面へ正射影したものとをなす角を求め、これを θ とする。

次に、変形する前の動作データの Leg2 のワールド座標系から見た姿勢に対し、ワールド座標系の z 軸回りに θ だけ回転させ、このときの姿勢を変形後の Leg2 の姿勢とする。

3. さらに Leg2 のローカル座標系の原点を求める。まず、注目している Leg2 が空中にある場合、上半身の姿勢を求めた方法と同様にして Leg2 のローカル座標系の原点を求める。一方、Leg2 が空中から地面に接触した場合、その接触地点が滑らないように接触地点を固定し、その接触地点と先に求めた Leg2 の姿勢から、Leg2 のローカル座標系の原点を求める。
4. 最後に脛 (Leg1) の姿勢を求める。Leg0 の姿勢を求める際に求めた Leg1 のローカル座標系 Σ'_{Leg1} の原点を、ワールド座標系の x-y 平面へ正射影した点から、先に求めた Leg2 の Σ'_{Leg2} の原点を x-y 平面に正射影した点へのベクトルを、Leg1 の Σ'_{Leg1} の x 軸を表すベクトルのワールド座標系の x-y 平面への正射影とし、後は Leg0 の姿勢を求めた方法と同様にして、Leg1 の姿勢を決定する。

以上のようにして、移動距離と移動方向の変形後の下半身の各部位の位置と姿勢を求める。

5.5.7 操作動作の合成

本項では、機器を操作する動作を合成する方法について述べる。機器を操作する動作を合成する場合、まず、操作対象の機器の位置に応じて、適切な動作データを選択する。そして操作対象の機器に手を移動させ終えた時の姿勢をまず求め、次に、動作の開始から操作対象の機器に手を移動させるまでの間の動作を求める。そして、レバーを操作する際には、

さらに機器を操作する動作を求める。これ以降、地点 a から地点 b まで腕を移動させる動作データを $TO_{a,b}$ 、地点 x においてレバーをスライドさせる動作データを LEV_x とする。

操作対象の位置に応じた動作データの選択

はじめに、操作対象の位置に応じて使用する動作データを選択する。直立の状態から手を伸ばす動作のデータベースは、5.5.5項で述べたように、手を伸ばす位置に応じて9種類存在する。まず、この9種類の動作データ ($TO_{0,x} \{x|x=1\cdots 9\}$) に対して、腕を移動し終わった時の手の平の位置の座標を求める。こうして定まった9点の座標の中で、腕を移動させる前の操作対象物の位置に最も近い点の番号を a、到達地点の操作対象物の位置に最も近い点の番号を b とし、 $TO_{a,b}$ の動作データを選択する。

また、レバーをスライドさせる動作の場合には、レバーの位置に応じた9種類の動作データ ($LEV_x \{x|x=1\cdots 9\}$) に対し、レバーの初期位置を求める。そして、合成する動作のレバーの初期位置に近い地点の番号を x とし、 LEV_x の動作データを選択する。

操作対象に到達時の姿勢の決定

次にボタンやレバーに手を伸ばして、目標物に手の平が接触した状態の体の姿勢を決定する。

まず始めに手の平の姿勢を決定する。ボタンを押したときの手の平の姿勢は、ボタンを押す動作データの最後の姿勢をワールド座標系から見たものをそのまま使用する。一方、レバーを握るときには、レバーの把手の方向に合わせるように変形する。本研究では、簡単化のため、仮想運転員がレバーをスライドさせる方向は、仮想運転員が向いている方向に限定する。

次に、人体モデルの他の部位の姿勢を決定する。まず、腰より下の下半身の姿勢は、動作データの最後の姿勢をそのまま使用する。一方、上半身に関しては、まず、頭は動作データの最後の姿勢をそのまま使用する。左腕全体（左上腕、左下腕、左手の平）については、直立の状態の胴に対する姿勢を保持し、常に胴に対して真っ直ぐのばしたまま横に添えるものとした。これは、目標の操作を行うのは右腕であり、左腕はそれほど重要ではないという考えに基づいている。また、実際に合成した動作を見ても、不自然さは感じられない。

続いて、胴、右上腕、右下腕の姿勢を決定する。基本的には、操作対象が遠くにあり、手が届かなければ、胴を手が届くまで前に倒し、逆に操作対象が近くにある場合は、肘を曲げて距離を調整する。

最初に、操作対象の機器が近くにある場合を考える。操作対象の機器が近くにあるということは、動作データにおいて右上腕のローカル座標系の原点と右手の平のローカル座標系の原点の間の距離を Len_0 、操作対象の機器に手の平が接触したときの、手の平の姿勢を決定した後の右上腕のローカル座標系の原点と、右手の平のローカル座標系の原点の間を Len_1 とした場合、 $Len_1 < Len_0$ になるということである。

人体モデルが立つ位置と操作対象の機器の位置が決まれば、右上腕のローカル座標系の原点の座標と、右手の平のローカル座標系の原点の座標は定まるので、求めるべきものは右肘の座標である。以下では右肘の座標を求める方法を述べる。

5. 人工現実感技術を用いた原子カプラント 運転の体験型教育システム VENUS の開発

1. まず、肘関節の角度 α (肘から肩へのベクトルと肘から手首へのベクトルのなす角度)を求める。右上腕、右下腕の長さは決まっているので、 α は、それらと Len を用いて、余弦定理より求めることができる。
2. 次に、右肘を曲げた状態にある姿勢のデータである Bend を、肘関節の角度が α になるように変形する。その時の Bend の肘の座標は、変形する前の姿勢の、肩、肘、手首の3点によって決定される平面上にあるようにする。また、手首の座標は、変形する前の姿勢における肩から手首へのベクトル上にあるようにする。ここで、肘を曲げた後の肩から手首へのベクトルを \vec{L}_1 とする。
3. 次に、肩と到達すべき手首の位置を結ぶベクトルを \vec{L}_2 とする。 \vec{L}_1 と \vec{L}_2 のなす角度を β とし、 \vec{L}_1 と \vec{L}_2 の外積を \vec{n} とする。そして、肘の角度を α に変形した後の Bend の肘の座標を \vec{n} を回転軸として β だけ回転させると、求めるべき肘の座標となる。

一方、操作対象の機器が遠くにあり手が届かない場合には、胴をローカル座標系の y 軸回りに回転して、 $Len_0 = Len_1$ になるように前方に倒す。その後、右上腕と右下腕の姿勢を上述の方法で決定する。

操作対象に到達するまでの姿勢の決定

これまでに述べた方法により、動作データの選択と、最初の姿勢及び操作対象に到達時の姿勢が決定されている。ここで、最初の姿勢は、直立、もしくは、以前に手を伸ばしている操作対象に対して操作を終了してその操作対象から手を離す直前の姿勢である。次に、選択された動作データを変形し、最初の姿勢と操作対象に到達時の姿勢の間の動作を合成する。以下でその手順を説明する。

1. 選択された動作データについて、最初の姿勢における親子関係にある2つのリンク A 及び B の間の座標変換 (以後、単に最初の姿勢と呼ぶ) を R_{orgF} 、最後の姿勢を R_{orgL} 、最初の姿勢におけるワールド座標系を基準とした右手の平のローカル座標系の原点の位置ベクトルを Pos_{orgF} 、最後の姿勢における原点の位置ベクトルを Pos_{orgL} とする。また、変形後の姿勢について、最初の姿勢を R_F 、最後の姿勢を R_L 、最初の姿勢における右手の平のローカル座標系の原点の位置ベクトルを Pos_F 、最後の姿勢における原点の位置ベクトルを Pos_L とする。最初と最後の姿勢における、変形前の姿勢と変形後の姿勢の差をそれぞれ R_{diffF} 、 R_{diffL} 、位置ベクトルの差を Pos_{diffF} 、 Pos_{diffL} とすると、以下の式が成り立つ。

$$R_F = R_{orgF} \cdot R_{diffF} \quad (5.15)$$

$$R_L = R_{orgL} \cdot R_{diffL} \quad (5.16)$$

$$Pos_{diffF} = Pos_F - Pos_{orgF} \quad (5.17)$$

$$Pos_{diffL} = Pos_L - Pos_{orgL} \quad (5.18)$$

但し、式 (5.15) 及び式 (5.16) の右項は、回転行列の積を表し、式 (5.17) 及び式 (5.18) の右項は、3次元ベクトルの差を表す。

2. 変形前の動作データにおいて、最初の姿勢から最後の姿勢までの右手の平のローカル座標系の原点の移動距離を Len 、最初の姿勢から x 番目の姿勢の右手の平のローカル座標系の原点の移動距離を len 、そして $ratio = len/Len$ とする。
3. まず、腰から下の姿勢及び胴、頭、右手の平の姿勢は次のように決定する。
変形前の姿勢を R_{x_org} 、変形後の姿勢を R_x として、 R_x を次式で求める。

$$R_x = R_{x_org} \cdot (R_{diffF} \times (1 - ratio) + R_{diffL} \times ratio) \quad (5.19)$$

この式で、 $R \times ratio$ は、回転行列 R を $M(Yaw, Pitch, Roll)$ で考え、 M の各成分を $ratio$ 倍することを意味し、 $+$ は、 M の各成分を加えることを意味する。左腕については前述のように胴に添えたまま動かさない。

4. 次に右上腕と右下腕の姿勢を決定する。まず、 x 番目の姿勢における右手の平のローカル座標系の原点の位置 Pos_x を計算する。変形前の右手の平のローカル座標系の原点の位置を Pos_{x_org} として、 Pos_x を次式で求める。

$$Pos_x = Pos_{x_org} + Pos_{diffF} \times (1 - ratio) + Pos_{diffL} \times ratio \quad (5.20)$$

ただし、 $\times ratio$ 及び $\times (1 - ratio)$ は、ベクトルの各成分に対して $ratio$ 及び $1 - ratio$ を掛けることを意味する。

しかし、このままでは Pos_x と肩との距離 $Len1$ が、変形前の肩から手首までの距離 $Len0$ に比べて大きい場合や小さい場合がある。そこで、 $Len1 \leq Len0$ の場合は前述の方法で腕を曲げ、 $Len1 > Len0$ の場合は、肩から Pos_x へのベクトルを \vec{L} として、 $\vec{L}l = \vec{L}/Len0 \times Len1$ を計算し、肩を始点とした $\vec{L}l$ の先端の座標を右手の平のローカル座標系の原点として腕を曲げる。

操作対象を操作する際の姿勢の決定

前述のように、ボタンを押す動作はボタンの位置まで手を移動するだけの動作としているので、ここでは、レバーをスライドさせる動作についてのみ述べる。

まず、LEV_x の動作データにおいてレバーをスライドさせた時の右手の平の移動距離を Len 、操作途中における移動距離を len とし、 $ratio = len/Len$ とする。以下では説明を容易にするため、仮想運転員がワールド座標系の $x-z$ 平面に平行な方向を向いているものとして考える。前述のように、仮想運転員がレバーをスライドさせる方向は、仮想運転員が向いている方向に限られるのでレバーがスライドする間、その xz 座標が変化し、 y 座標は変化しない。最初のレバーの位置を $(x0, y0, z0)$ 、最後のレバーの位置を $(x1, y0, z1)$ として、操作途中のレバーの位置を (x, y, z) として、以下の式で計算する。

$$(x, y, z) = (x_0, y_0, z_0) + (x_1 - x_0, 0, z_1 - z_0) \times ratio \quad (5.21)$$

式 (5.21) で求めた操作途中のレバーの位置を右手の平の位置に合わせ、右手の平の姿勢を、前述の操作対象に到達時の姿勢を決定した方法と同様にして求め、さらに手首の位置が肩に近ければ腕を曲げ、遠ければ胴を傾けた後に腕の姿勢を決定する。

指の動きの決定

指の動きに関しては、3次元モーションキャプチャシステムでは計測できないので、以下の方法で合成する。

5本の指それぞれについて、媒介変数を t_n ($n=0$: 親指、 $n=1$: 人指し指、 $n=2$: 中指、 $n=3$: 薬指、 $n=4$: 小指) として指を伸ばした状態を $t_n = 0$ 、完全に曲げた状態を $t_n = 1$ 、そして5本の指の状態を $F(t_0, t_1, t_2, t_3, t_4)$ と表すと、ボタンを押す時の指は人指し指が伸びており、他の指は完全に曲げた状態なので $F(1, 0, 1, 1, 1)$ 、レバーを握っている時の指は全ての指が完全に曲がっているため $F(1, 1, 1, 1, 1)$ となる。以下、指の動きを合成する方法について説明する。

最初の姿勢が直立の場合は、 $F(0, 0, 0, 0, 0)$ から $F(1, a, 1, 1, 1)$ へと線形に変化させる。ここで操作がボタンを押す場合は $a = 0$ 、レバーをスライドさせる場合は $a = 1$ である。

最初の姿勢がボタンを押している場合、

- 次もボタンを押す場合は $F(1, 0, 1, 1, 1)$ のまま変化させない。
- 次がレバーの場合はまず $F(1, 0, 1, 1, 1)$ から $F(0.5, 0.5, 0.5, 0.5, 0.5)$ へと変化させる。これはレバーを握るためにすべての指をある程度まで開く必要があるからである。その後 $F(1, 1, 1, 1, 1)$ へと変化させる。
- 次が直立の場合は $F(0, 0, 0, 0, 0)$ へと変化させる。

最初にレバーを握っている場合は、まず、 $F(1, 1, 1, 1, 1)$ から $F(0.5, 0.5, 0.5, 0.5, 0.5)$ へと変化させる。そして、

- 次がボタンの場合は $F(1, 0, 1, 1, 1)$ へと変化させる。
- 次もレバーの場合は $F(1, 1, 1, 1, 1)$ へと変化させる。
- 次が直立の場合は $F(0, 0, 0, 0, 0)$ へと変化させる。

以上のようにして、機器操作時の指の動きを合成する。

5.5.8 移動動作と操作動作の統合

本項では、前項までに述べた方法を用いて、実際にヒューマンモデルシミュレータからの指示に応じて、仮想運転員の動作を合成する方法について述べる。

仮想運転員の動作を合成する際、人体モーション合成シミュレータは、SEAMAID を構成するヒューマンモデルシミュレータから、仮想運転員の移動先の座標と方向、操作する機器の座標と操作方法に関する情報をネットワークを介して受け取る。この時、操作対象の機器に、仮想運転員の手が届くかどうかは、ヒューマンモデルシミュレータの方で判定済みであるため考慮する必要はない。

実際に仮想運転員の動作を合成する際には、合成する仮想運転員の動作の内容を、動作の指示を受けた直前の動作の種類に応じて、以下のように変更する。

直前の動作が機器操作であった場合は、仮想運転員は操作対象の機器に手を伸ばしたままであるので、

- 次の動作が機器操作の場合は、直立には戻らずに、5.5.7項で述べた方法を用いて、連続して対象の機器を操作する動作を合成する。
- 次の動作が移動の場合は、一度直立の姿勢に戻ってから、5.5.6項で述べた方法を用いて移動する動作を合成する。

直前の動作が移動であった場合、

- 次の動作が機器操作の場合は、そのまま、機器を操作する動作を合成する。
- 次の動作が移動の場合は、そのまま、移動する動作を合成する。

以上のようにして、ヒューマンモデルシミュレータからの指示に応じて仮想運転員の動作を合成する。合成した結果は、ネットワークを介して、仮想空間シミュレータに送られる。

5.6 仮想空間シミュレータ

仮想空間シミュレータは、仮想空間内に、制御室、制御盤、及び、仮想運転員の動作を3次元映像として描画する。本節では、制御盤の描画方法、仮想運転員の描画方法及び、仮想運転員の視点からの映像を合成する方法について述べる。

仮想空間内に教育対象とする原子力発電プラントの制御室を描画する場合、制御室内に制御盤がどのように配置されるのか、制御盤上に機器がどのように配置されるのかといった、制御室の設計情報が必要になる。これらの情報は、SEAMAID を構成するマンマシンインタフェースシミュレータが保持しているが、VENUS では、SEAMAID を実行するワークステーションと、仮想空間シミュレータを実行するワークステーションを別々にしているため、これらの情報は、ワークステーション間でファイルを共有する、NFS(Network File System) を用いて共有し、仮想空間シミュレータで用いることにした。

なお、マンマシンインタフェースシミュレータは、制御盤の設計を変更するための専用のエディタを備えており、GUI を介して、容易に制御盤の設計を変更することができる。

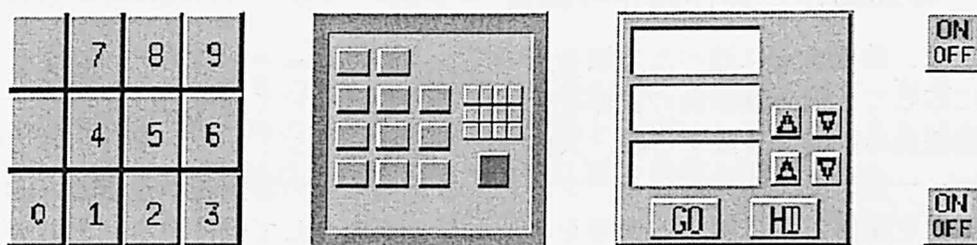


Fig. 5.13: 制御盤上に配置される機器のテクスチャ

これにより、制御盤の設計を自由に変更し、それに応じて仮想運転員がどのように制御盤を操作するかを観察することも可能である。

5.6.1 制御盤の描画

VENUS が教育の対象とする原子力発電プラントは、その制御室に 10 個の制御盤が配置され、それらの制御盤上に約 20 種類の機器が、合計約 500 個配置されている。前述の通り、訓練生が仮想空間内の機器を見ることにより、原子力発電プラントの状態を把握できるようにするためには、ある程度詳細に、機器及びその数値表示等を描かねばならない。しかし、これらを全てポリゴンを用いて詳細に描く場合、非常に描画能力が高い計算機が必要となる。そこで本研究では、機器の詳細はポリゴンを用いず、Fig. 5.13に示すようなテクスチャを用いて描画し、さらに、訓練生の視点と個々の機器との距離に応じて、描く機器の詳細度を変化させる、LOD(Level Of Detail) の手法を適用した。具体的には、訓練生の視点と各機器との距離が遠い順番に、1. 何も描かない、2. 機器の枠のみ描く、3. 数値表示以外の機器の全体を描く、4. 機器の数値表示まで詳細に描く、の 4 段階に分けて機器を描画するものとした。また、仮想制御盤上の機器の数値表示を更新する場合、原子力プラントシミュレータからは、更新された機器の情報のみ、1 秒ごとに受け取るものとした。この手法により、全ての機器の数値表示等まで詳細に描く場合と比べて、約 1.5 倍（訓練生と制御盤の距離が近い場合）から 5 倍（訓練生と制御盤の距離が遠い場合）の描画速度を得ることができた。Fig. 5.14に本手法で描画した制御室を、Fig. 5.15に制御盤を示す。制御盤上の機器には、現在の原子力発電プラントの状態を表す数値が表示されていることを確認できる。

5.6.2 仮想運転員の描画

仮想運転員の 3 次元形状モデルは、腕の長さや足の大きさ等、実際の人の各部位のサイズを計測し、その結果をもとに CAD を用いて作成した。これにより、仮想運転員を構成するポリゴンの総数は約 700 個となった。また、仮想運転員の動きを見やすくするために、仮想運転員の各部位には、テクスチャを張った。

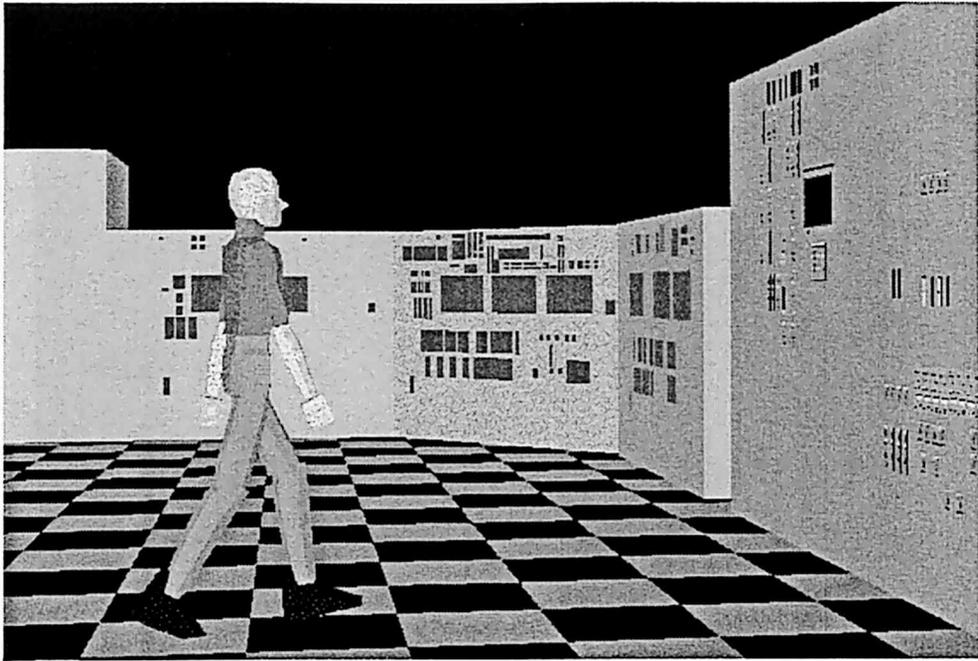


Fig. 5.14: 仮想制御室

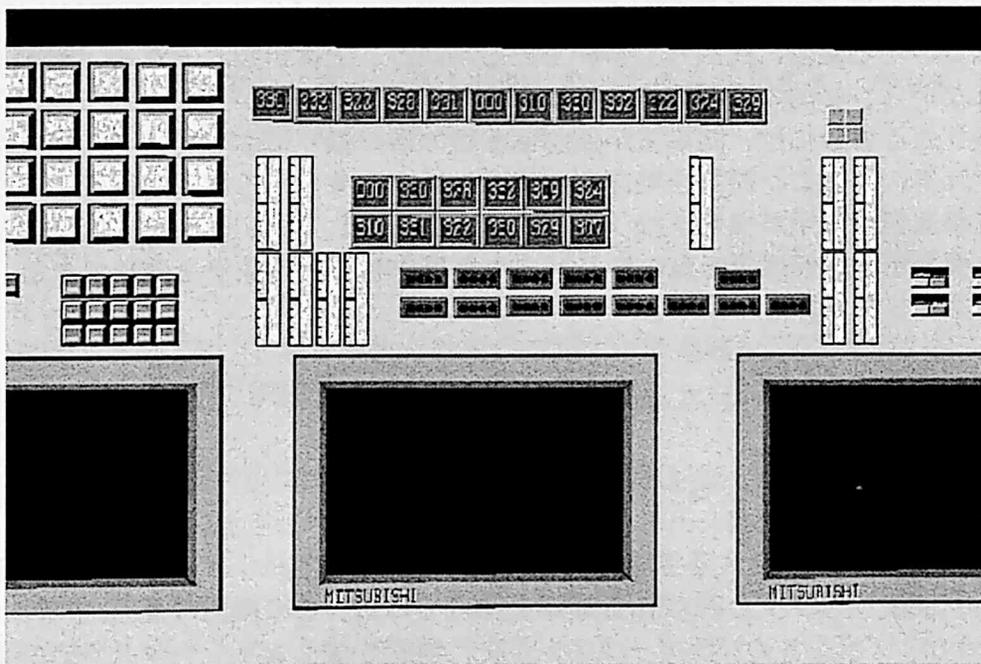


Fig. 5.15: 仮想制御盤

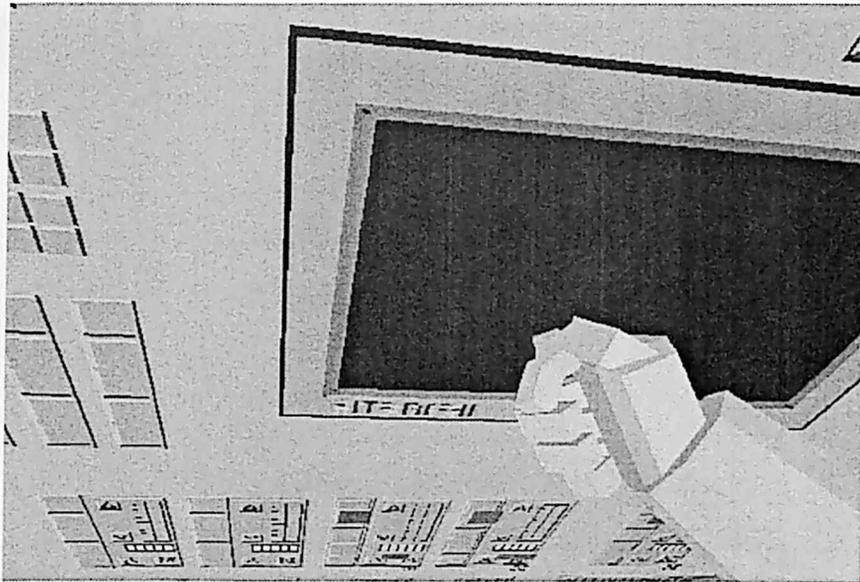


Fig. 5.16: 仮想運転員の視点からの映像

5.6.3 仮想運転員の視点からの映像の描画

仮想運転員の視点からの映像は、基本的には人体モーション合成シミュレータにおいて求められた仮想運転員の各姿勢における、頭の位置と方向をもとに合成される。例えば、仮想運転員がある機器に向かって歩いている時には、上体が揺れるため、制御室が上下左右に揺れながら、目的の機器が徐々に大きくなっていく映像が合成される。また、仮想運転員が振り返る場合には、制御室が仮想運転員の頭の振りに応じて徐々に横にずれる映像が合成される。さらに、機器を操作する場合には、対象となる機器を視野の中心に合わせながら仮想運転員の手が機器を操作する映像が合成される。Fig. 5.16に仮想運転員の視点からの映像を示す。仮想運転員の手がパネルを操作している様子を確認できる。

5.7 その他のシミュレータ

5.7.1 CRT モニタシミュレータ

VENUS が教育の対象とする原子力発電プラントの制御盤は、プラントの状態をまとめて表示する「CRT モニタ」と呼ばれる機器を備えている。この機器の画面上には、Fig. 5.17 に示すような、プラントのパラメータの時間的な推移を表したトレンドグラフのような複雑な映像が表示される場合がある。このような複雑な映像を、仮想空間内に配置された CRT モニタ上に、3次元映像として詳細に描画する場合、計算機に多大な負荷が掛かる上に、訓練生には非常に見づらい映像になることが予想される。そこで本研究では、この CRT モニ

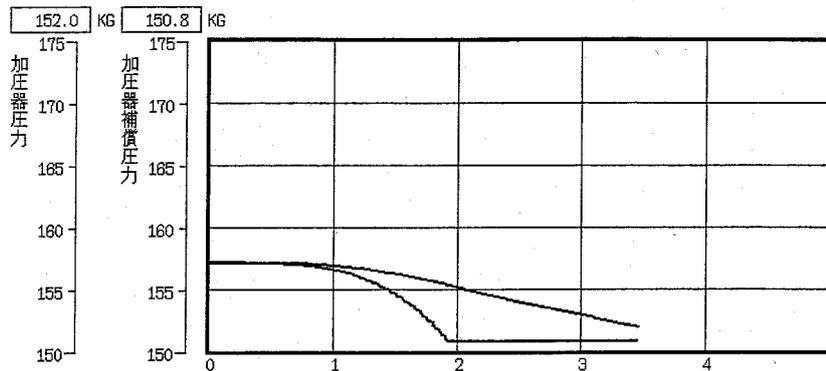


Fig. 5.17: CRT モニタ上に描画されるトレンドグラフの例

タに関してのみ、数値やグラフ等の詳細な描画は仮想空間内で行わず、3次元映像を提示するディスプレイとは別の2次元映像を表示するための専用のディスプレイを用意し、その画面上に2次元の映像として描画することにした。

具体的には、CRT モニタシミュレータは、ヒューマンモデルシミュレータから、仮想運転員の行動に関する情報を受け取り、仮想運転員が眺めている CRT モニタの種類に応じて、2次元映像をリアルタイムに変更する。Fig. 5.18に CRT モニタシミュレータが表示する2次元映像の例を示す。

さらに、VENUS では、CRT モニタ上で仮想運転員が眺めている箇所をリアルタイムに矢印で示す機能を構築した。この機能により、訓練生は、仮想運転員が CRT モニタを眺めているときに、具体的にどのパラメータを参照しているのかを知ることができ、より具体的に、プラント診断や対応操作の過程を理解できると期待される。

5.7.2 発話合成シミュレータ

発話合成シミュレータは、訓練生が仮想運転員の行動を観察するだけでは理解できない、原子力発電プラントにおける異常事象を診断する際の思考過程を、「仮想運転員が自らの考えを発話する」という形で訓練生に提示する。具体的には、主に Table 5.3に示す場合に発話するものとした。

具体的には、仮想運転員が何時、どのような内容の発話を行うかの設定は、ヒューマンモデルシミュレータ内に記述されており、発話合成シミュレータは、ヒューマンモデルシミュレータから、リアルタイムに仮想運転員が発話すべき内容を受け取る。近年は、ViaVoice といった合成音声を使用できるシステムも開発されているが、これらは、UNIX Workstation 上では使用できないため、ViaVoice を使用するためには、新たに VENUS にパソコンを組み込む必要があり、コストが高くなるという問題がある。そこで今回は、UNIX Workstation で仮想運転員の発話を合成できる方法を採用した。すなわち、今回は、仮想運転員が発話すべき内容が原子力発電プラントの異常診断と対応操作に関する内容に限定されているこ

5. 人工現実感技術を用いた原子カプラント 運転の体験型教育システム VENUS の開発

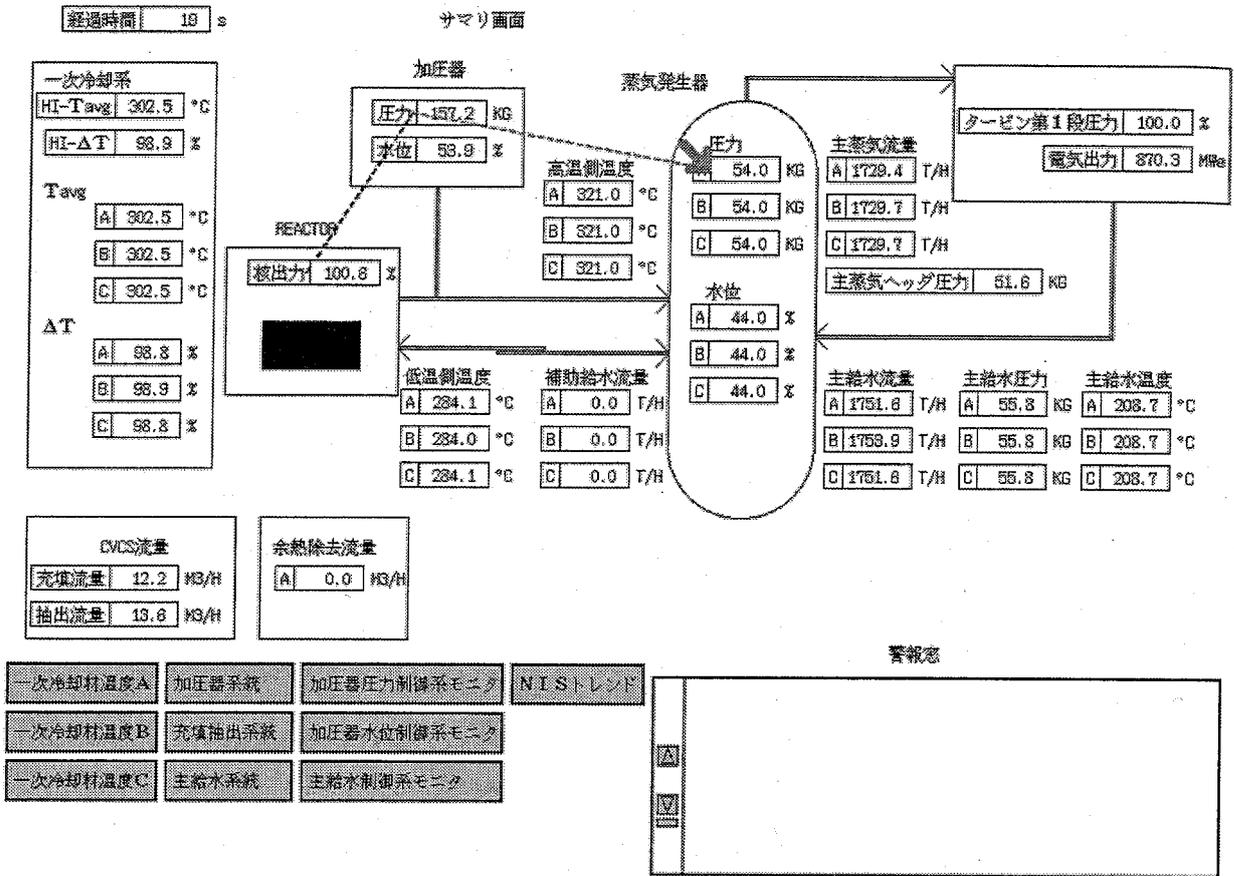


Fig. 5.18: CRT モニタシミュレータが模擬する 2 次元映像の例

Table 5.3: 仮想運転員の発話内容

| 仮想運転員が発話する条件 | 発話の例 |
|------------------|----------------------|
| プラントのパラメータを調べる場合 | 加圧器圧力を調べます |
| 異常が発生したことを認知した場合 | 異常が発生しました。診断を開始します |
| 異常仮説を想起した場合 | LOCA が発生したとの仮説を立てます |
| パラメータの状態を予測した場合 | この仮説では加圧器圧力が大きいはずです |
| パラメータに対し何か判断した場合 | 加圧器水位は予想どおり小さくなっています |
| 異常仮説を棄却した場合 | LOCA ではなさそうです |
| 異常仮説を採用した場合 | LOCA が発生したと確信しました |

5.8 VENUS の使用例と考察

Table 5.4: 仮想運転員の行動

| 時間 (分:秒) | 行動 (イベント) |
|----------|----------------------|
| 00:00 | (シミュレーション開始) |
| 00:00 | A-AG 水位の確認操作を始める |
| 00:20 | A-AG 水位の値を確認する |
| 00:22 | プロセスモニタ注意警報の発生を確認する |
| 00:22 | LOCA が発生したとの仮説を立てる |
| 00:22 | 格納容器ガスモニタの確認操作を始める |
| 00:41 | 格納容器ガスモニタの状態を確認する |
| 00:52 | 格納容器じんあいモニタの確認操作を始める |
| 01:25 | 格納容器じんあいモニタの状態を確認する |
| ... | ... |
| 02:10 | 加圧器水位の変化傾向の確認操作を始める |
| 02:24 | 加圧器水位の変化傾向を確認する |
| 02:31 | LOCA が発生したと断定する |
| 02:31 | 加圧器水位と圧力を維持する操作を始める |
| ... | ... |
| 03:50 | 加圧器水位と圧力を維持する操作を終える |
| 03:50 | 緊急負荷降下操作を始める |
| ... | ... |
| 05:06 | 緊急負荷降下操作を終える |
| 05:14 | (原子炉がトリップする) |

とに注目し、仮想運転員が発話する内容は、予め人間が発話した声をオーディオファイルとして録音しておき、必要に応じて再生することにより、仮想運転員の発話を模擬することにした。

実際には、予め準備するオーディオファイルを、名詞、動詞、副詞、その他に分割して録音し、発話を合成する際には、適切なオーディオファイルを複数選択し、それらを連続して再生することにより、予め準備すべきオーディオファイルの総数を減らす工夫を行った。これにより、例えば LOCA(Loss Of Coolant Accident) の異常事象を診断するシミュレーションを実行するために必要なオーディオファイルの総数は約 90 種類となった。

5.8 VENUS の使用例と考察

VENUS の動作例として、原子力プラントシミュレータに LOCA の異常を発生させた場合の仮想運転員の行動と発話内容を Table 5.4 及び Table 5.5 に示し、さらにその時の仮想運転員が仮想制御室内を移動した軌跡を Fig. 5.19 に示す。

5. 人工現実感技術を用いた原子力プラント 運転の体験型教育システム VENUS の開発

Table 5.5: 仮想運転員の発話例

| 時間 (分:秒) | 発話内容 |
|----------|-----------------------------|
| 00:00 | A-AG 水位を調べます。 |
| 00:20 | A-SG-水位は変化していません。 |
| 00:22 | プロセスモニタ注意警報が発生しました。 |
| 00:24 | 診断を開始します。 |
| 00:25 | 今考えられる異常は LOCA です。 |
| 00:27 | LOCA が発生したとの仮説を立てます。 |
| 00:29 | 格納容器ガスモニタを調べます。 |
| 00:41 | 格納容器ガスモニタの値は大きくなっています。 |
| 00:52 | 格納容器じんあいモニタの変化傾向を調べます。 |
| 01:25 | 格納容器じんあいモニタは過去に増加し現在も増加中です。 |
| ... | ... |
| 02:10 | 加圧器水位の変化傾向を調べます。 |
| 02:24 | 加圧器水位は減少中です。 |
| 02:31 | LOCA が発生したと確信しました。 |
| 02:34 | 対応操作に入ります。 |
| ... | ... |

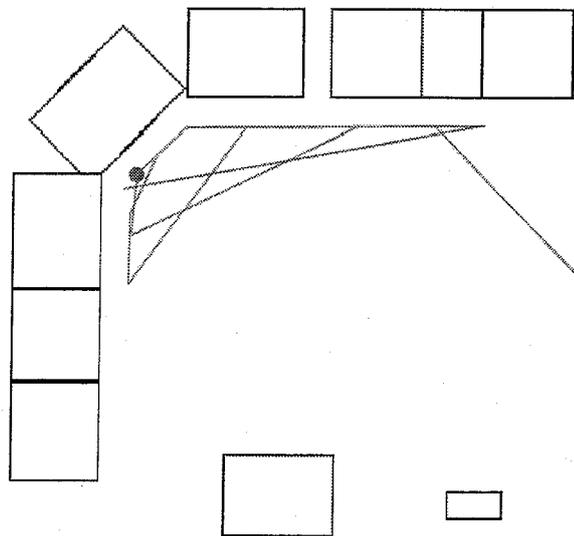


Fig. 5.19: 仮想運転員の歩行軌跡

5.9 将来課題

また、本研究で開発した VENUS を用いることにより、訓練生が、原子力発電プラントの異常診断と対応操作の様子を観察でき、診断の内容を理解できることを確認するために、研究室の3人の大学院生に VENUS を使用してもらった。その結果、使用後のアンケートにより、以下のような感想を得ることができた。

1. 仮想運転員の視点からの映像を眺めることにより、自分で制御室の中を歩き回っているような感じがした。
2. 仮想運転員の視点を体験する機能はおもしろい。
3. 仮想運転員が何を考え、何をしようとしているのかよく分かった。
4. システムを使用しているうちに気分が悪くなってきた。
5. 仮想運転員が発話しなかったら、何をしているのかよく分からなかったかもしれない。
6. 画面がコマ落ちするので少し見にくかった。

これらの結果により、VENUS の使用者が仮想運転員の動作を眺め、発話を聞くことにより、仮想運転員の思考過程と行動の内容を理解できることが確認できた。また、「仮想運転員の視点を体験する機能はおもしろい。」という意見が得られたことにより、VENUS は、訓練生の学習意欲を促進する可能性があることも判明した。

一方、「システムを使用しているうちに気分が悪くなってきた。」との意見が得られたが、特に、仮想運転員が仮想空間内で振り返る場面を見るときに、気分が悪くなるとの意見が得られた。これは、身体への動揺と視覚情報との間に不一致が生じるためであると推測される [12] が、VENUS を用いることにより、訓練生の気分が悪くなるならば、訓練生の学習意欲を削ぐ可能性があるため、訓練生の気分が悪くならないように VENUS のインタフェースを改良する必要がある。また、「画面がコマ落ちするので少し見にくかった。」という意見が得られたが、VENUS で使用した計算機の性能上、現在の描画アルゴリズムでは負荷が高くなった場合に、かなり画面の更新速度が遅く感じるようである。これを避けるには、さらに高性能な計算機を用いるか、描画アルゴリズムをさらに改良する必要がある。

5.9 将来課題

本研究で開発した VENUS を実際の訓練の現場で使用するためには、以下の項目を順次実現する必要がある。

1. 訓練生自らが仮想制御盤を操作し、その結果を体験できる機能の構築

現時点の VENUS を用いることにより、訓練生は、原子力プラントの異常発生時における、診断と対応操作の作業の様子を把握できると期待されるが、これに加えて、作業の様子を把握した後に、訓練生自らが制御盤を操作することにより、獲得した知識の確認ができるようになれば、さらに学習効果が上がると思われる。

5. 人工現実感技術を用いた原子力プラント運転の体験型教育システム VENUS の開発

2. 仮想運転員が診断可能な異常事象の追加

現時点では、SEAMAID が診断と対応が可能な異常事象が LOCA と SGTR(Steam Generator Tube Rupture) の 2 種類であるため、VENUS で仮想運転員が診断と対応が可能な異常事象もこれら 2 種類である。訓練生が診断および対応技術を修得すべき異常事象は多岐に渡るため、現在の VENUS が対応できる異常事象の種類だけでは不十分である。今後、診断と対応が可能な異常事象の種類を増やす必要があるが、既に SEAMAID 内のヒューマンモデルシミュレータには、異常事象に対する処理関数が構築されており、異常事象に対する新たな知識をヒューマンモデルシミュレータの知識ベースに加えるだけで、診断と対応が可能な異常事象の種類を増やすことができ、それほど多くの労力は必要ないと思われる。

3. VENUS のインタフェースの改良

前述のように、VENUS を用いることにより訓練生の気分が悪くなれば、訓練生の学習意欲を削ぐ可能性があり、問題である。従って、今後、訓練生が VENUS を使用しても気分が悪くならないように、インタフェースを改良する必要がある。しかし、このようないわゆる VR 酔いは、その原因が正確には分かっておらず、訓練生が全く気分が悪くならないインタフェースを構築するのは現時点では困難であると思われる [12]。従って、訓練生が VENUS を使用する時間を制限する等の方向で対策を考える必要があると思われる。

4. VENUS の使用による学習効果の評価

今回、VENUS を使用してもらった結果により、使用者が仮想運転員の行動を自由に眺め、仮想運転員の思考過程を理解できることが確認されたが、実際に、訓練センターで訓練を受ける訓練生が、VENUS を用いることにより学習効果が上がるかどうかの評価はなされておらず、今後の課題である。

5.10 まとめ

本章では、原子力発電プラントの異常診断と対応操作を教育対象とした、人工現実感技術を用いた体験型教育システム VENUS について述べた。

本研究では、訓練生が原子力発電プラントの異常診断や対応操作等の作業を容易に理解できるようにするために、まず、VENUS が具備すべき機能について検討し、主に以下の 5 つの項目に重点をおいて開発を進めることにした。

1. 仮想空間内に原子力発電プラントの制御盤を 3 次元モデルとして詳細に構築し、訓練生が制御室の様子を容易に把握できるようにする。
2. 原子力プラントシミュレータの計算結果を仮想空間内の機器にリアルタイムに反映させ、プラント異常発生時の制御盤の挙動を観察できるようにする。

3. 仮想運転員が自らの判断で制御室内を歩き回り、異常診断と対応操作を行う様子を自由に観察できるようにする。
4. 仮想運転員がプラントの診断と対応操作を行っている時の思考過程を訓練生がリアルタイムに理解できるようにする。
5. 仮想運転員は実際の運転員の動作に近い動作でプラントの異常診断と対応操作を行うようにする。

具体的には、VENUSは、計算負荷を分散させるために、分散型シミュレーションシステムとして設計し、SEAMAID、人体モーション合成シミュレータ、仮想空間シミュレータ、CRTモニタシミュレータ、発話合成シミュレータの5つのシミュレータで構成した。

まず、SEAMAIDは仮想運転員の頭脳に相当するシミュレータで、実際のプラント運転員がプラントの異常事象を診断する際の行動を計測した実験結果をもとにして構築されており、実験により、運転員の行動と思考過程をよく再現できることが確認されている。

人体モーション合成シミュレータは、仮想運転員の制御室における行動を3次元映像として合成するシミュレータで、本研究では、仮想運転員のプラントを運転する際の動作を、実際の人の動作に近いものにするために、以下の手順で仮想運転員の動作を合成する方法を考案した。

1. 3次元モーションキャプチャシステムを用いて人の動作を計測し、動作データベースを構築する。
2. 動作データベースから目的の動作に最も近い動作データを選択し、これを変形することによって、目的の動作を合成する。

以上の方法により、移動の方向と移動距離に応じて自由に歩く動作と、ボタンを押す動作、レバーをスライドさせる動作が合成可能になった。

仮想空間シミュレータは、仮想空間内に制御室と仮想運転員を3次元映像として描画するシミュレータで、数多くの機器を高速に描く為に、訓練生の視点と描画対象の機器の間の距離に応じて、描く詳細度を変化させるLODの手法を採用し、全ての機器を詳細に描く場合に比べて、最大で約5倍の高速描画が可能になった。また、仮想運転員の視点からの映像は、人体モーション合成シミュレータによって合成された仮想運転員の姿勢をもとに合成した。

また、発話合成シミュレータでは、運転員の思考過程を訓練生に提示するために、予め人間が発話した声を録音したオーディオファイルを、ヒューマンモデルシミュレータからの指示に応じて再生することにより、仮想運転員の発話を模擬した。その際、予め録音する発話内容を、名詞や動詞といった幾つかの言葉に分けて録音し、必要に応じて複数のファイルを連続して再生することにより、予め用意すべきオーディオファイルの数を減らす工夫を行った。

そして、VENUSを、研究室の大学院生に使用してもらうことにより、VENUSの使用者が仮想運転員の行動と思考内容を理解できること、訓練生の学習意欲を促進する可能性が

5. 人工現実感技術を用いた原子カプラント 運転の体験型教育システム VENUS の開発

あることが判明した。しかし VENUS を長時間使用すると気分が悪くなる場合があることも判明し、他人の行動を人工現実感技術を用いて「体験」させる場合、ユーザの気分が悪くならないように、何らかの対策をとる必要があることも判明した。

最後に、VENUS を実際の訓練に使用するために実現すべき将来課題として、訓練生自らが仮想制御盤を操作して新しく得た知識を確認できる機能の構築、VENUS で対応可能な異常事象の追加、VENUS を用いることにより訓練生気分が悪くならないようにするためのインタフェースの改良、VENUS を実際の訓練生に使用してもらうことによる学習効果の評価等を行う必要があることを述べた。

参考文献

- [1] 中川隆志, 仲谷善雄, 佐々木和則, 吉川榮和, 高橋信, 古田富彦, 長谷川明. マンマシンインタフェース評価のための原子力運転員行動シミュレータの開発. 計測自動制御学会論文集, Vol. 33, No. 8, pp. 834-842, 1997.
- [2] H. Yoshikawa, T. Nakagawa, Y. Nakatani, T. Furuta, and A. Hasegawa. Development of an analysis support system for man-machine system design information. *Control Engineering Practice*, Vol. 5, No. 3, pp. 417-425, 1997.
- [3] J. Reason. *Human Error*, pp. 97-147. Cambridge University Press, 1990.
- [4] Wu Wei, 高橋信, 吉川榮和. 原子炉異常診断過程の認知プロセスのモデル化とシミュレーション. 第12回ヒューマン・インタフェース・シンポジウム論文集, pp. 343-348, 1996.
- [5] Wu Wei, 中川隆志, 吉川榮和. プラント運転員の異常診断行動のモデリングとヒューマンモデルシミュレーションによる人間認知信頼性(HCR)曲線の導出法の研究. ヒューマンインタフェース学会論文誌, Vol. 1, No. 2, pp. 11-24, 1999.
- [6] M. Cavazza, R. Earnshaw, N. Thalmann, and D. Thalmann. Motion control of virtual humans. *IEEE Computer Graphics and Applications*, Vol. 18, No. 5, pp. 24-31, 1998.
- [7] J. Hodgins. リアルになったコンピュータアニメ. 日経サイエンス, No. 320, pp. 46-53, 1998.
- [8] B. Delaney. The mystery of motion capture. *IEEE Computer Graphics and Applications*, Vol. 18, No. 5, pp. 14-19, 1998.
- [9] D. Chen and D. Zeltzer. Pump it up : Computer animation of a biomechanically based model of muscle using the finite element method. *Computer Graphics*, Vol. 26, No. 2, pp. 89-98, 1992.
- [10] D. Brogan, R. Metoyer, and J. Hodgins. Dynamically simulated characters in virtual environments. *IEEE Computer Graphics and Applications*, Vol. 18, No. 5, pp. 2-13, 1998.
- [11] B. Robertson. Toy story : A triumph of animation. *Computer Graphics World*, Vol. 18, No. 8, pp. 28-38, 1995.

参考文献

- [12] 中川千鶴, 大須賀美恵子. VE 酔い研究および関連分野における研究の現状. 日本バーチャルリアリティ学会論文誌, Vol. 3, No. 2, pp. 31-39, 1998.

第 6 章 結論

本研究では、原子力発電プラントの保守作業員とプラント運転員を訓練するための人工現実感技術を用いた訓練システムに関する研究を行った。まずはじめに、保守作業員を対象にした訓練システムに関して、作業の自由度が高い機器保守の訓練環境を仮想空間内に構築するための新たな手法を提案し、具体的な訓練課題に対する訓練システムとして VRMATE を開発した。そして、提案した手法を用いることにより、これまでの構築手法に比べ、少ない労力で訓練環境を構築できることを確認した。その後、プログラミングを行わなくても機器保守の訓練環境を自由に構築できる新たな手法を提案し、プログラミングによる訓練環境の構築手法に比べ、非常に短時間で訓練環境を構築できる支援システム DESCORTE を開発した。次に、プラント運転員を対象にした訓練システムに関して、原子力発電プラントの異常発生時における運転員の行動と思考過程を模擬できるヒューマンモデルの計算結果を可視化することにより、仮想空間内に構築された仮想運転員がプラントの運転を行う様子を自由に観察できる訓練システム VENUS を開発した。そして、数人の被験者に VENUS を試用してもらうことにより、プラント運転に関する予備知識を持たない初心者が、仮想運転員がどのように考えて異常事象を診断しているのか、どのような装置を操作しているのかを理解できることを確認した。

以下に、本論文で行った研究をまとめる。

第 2 章では、まずはじめに、人工現実感の現在の研究状況を展望し、現時点で何が技術的に実現されているか、何が技術的な制約として問題になっているかを明らかにした。次に、訓練センターにおける保守作業員やプラント運転員に対して実施している訓練の現状と問題点についてまとめた。その後、人工現実感技術を用いた訓練システムの研究状況について、これまでに開発された保守作業員とプラント運転員を訓練するためのシステムの概要と問題点についてまとめた。その結果、(1) これまでに開発された機器保守の訓練システムは、その訓練環境を構築する際の労力を削減することに注目していなかったために、数多くの機器に対する訓練環境を構築することが困難であること、(2) 機器保守の訓練システムを開発する際には、プログラミングに関する高度な知識が必要なため、訓練を行うインストラクタには、容易に訓練環境を構築できないこと、(3) これまでに開発されたプラント運転員を訓練するためのシステムは、訓練生がある程度の知識を備えていない場合は使用できないことに注目し、本研究では、以下の 3 つの研究を行うことにした。

1. 機器保守の訓練環境を構築する際の労力を削減できる新しい訓練環境の構築手法の提案と具体的な訓練課題に対する訓練システムの開発
2. プログラミングを行わなくても機器保守の訓練環境を自由に構築できる新たな手法の提案と機器保守の訓練環境の構築支援システムの開発

3. プラント監視制御行動中の運転員の行動を情報可視化し、その結果を観察することによりプラント運転に必要な知識や技能を修得する体験型教育システムの開発

第3章では、人工現実感技術を用いた機器保修の訓練システムの開発について述べた。まず、訓練生が効率的に訓練できるようにするために、機器保修の訓練システムが備えるべき特徴についてまとめた。次いで、機器保修の訓練環境を少ない労力で構築できる新しい手法として、仮想物体の状態遷移の表現法にペトリネットを応用したモデル化を行い、作成したペトリネットを用いて仮想空間をシミュレーションする手法を提案した。さらに、ペトリネットを構築する作業効率を改善するために、ペトリネット構築支援システム(PCSS)を開発した。PCSSでは、ペトリネットをGUIを介して視覚的に構築することが可能であり、過去に作成したペトリネットを再利用できる機能も備えている。本研究では、PCSSを用いてスイング式逆止弁の分解作業を対象にモデル化したペトリネットを構築した。その結果、PCSSを用いることにより、少ない労力で大規模なペトリネットが容易に構築できることを確認した。

次に、訓練生が機器保修の作業手順を効率的に学ぶことができるようにするために、「仕事」の概念を用いた自動実演機能を実現するためのアルゴリズムを考案し、機器保修の訓練システムに実装した。その後、原子力発電プラントの構成機器の一つであるスイング式逆止弁の分解作業を行うことができる訓練システムVRMATEを開発し、実際に複数の被験者による試用実験を行い、保修作業の初心者でも作業の手順や機器の構造を理解するのに有効なことを確認すると共に、VRMATEの実用化に向けて更に解決すべき課題を抽出した。

第4章では、C言語等のプログラミングを全く行うことなく機器保修の訓練環境を構築できる支援システムDESCORTEを開発した。はじめに、DESCORTEで構築可能とする訓練環境が備えるべき特徴について述べた。そして、実際の保修作業の訓練の対象となっている複数の機器に対して、その機器の保修作業を行うための訓練環境を仮想空間内に構築する際に、どのような情報が必要となるかを分析し、機器保修の訓練環境を構築するためには、仮想物体の外見や運動の仕方を定める情報と仮想物体の状態遷移の仕方を定める情報を構築する必要があることを述べた。そして、機器保修の訓練環境を、プログラミングを行わずに構築できる方法として、第3章で提案した仮想物体の状態遷移をペトリネットを用いてモデル化する手法を改良したものと、訓練生のジェスチャに応じた仮想物体の運動を、予め定められたパラメータを変更することにより設定する手法を融合させた、新しい訓練環境の構築方法を提案した。

そして、一般に広く普及しているパソコンのオペレーティングシステムで採用されているWIMPインタフェースを基本として、機器保修の訓練環境を構築する際に必要となる情報を入力する際に使用するDESCORTEのインタフェースを設計した。その後、開発したDESCORTEの評価として、第3章で構築した機器保修の訓練環境と同じ訓練環境を、DESCORTEを用いて構築し、訓練環境を構築する際の作業時間効率が飛躍的に改善されることを確認した。しかし、一方で、主に訓練環境を構築する際のDESCORTEのインタフェースに改良すべき点が数多く存在することも判明した。そこで、インタフェースを改良するための方策として、ユーザの認知的負荷を軽減させる観点でインタフェース設計の

改良を行った後に、再び DESCORTE の評価を行い、訓練環境を構築する際の作業時間効率が、インタフェース改良前に比べ、さらに改善されることを確認した。

第5章では、人工現実感技術を用いたプラント運転の体験型教育システムの開発について述べた。ここでは、訓練生に欠けている知識を積極的に獲得させることを目標として、プラント監視制御行動中の運転員の行動と思考過程を人工現実感技術を用いて情報可視化し、その結果を用いてプラント運転の教育訓練を行う体験型教育システム VENUS を開発した。具体的には、VENUS を分散型シミュレーションシステムとして開発し、安価な計算機を複数使用する訓練環境を開発した。その際、3次元モーションキャプチャシステムを用いて実際の人々がプラントを操作する動作を計測して動作データベースを作成し、必要に応じて動作データを変形することにより、歩行動作と機器を操作する動作を柔軟に合成できる人体モーション合成手法を開発した。そしてプラント異常発生時における運転員の行動と思考過程を模擬できるヒューマンモデルシミュレータを用いて、仮想空間内で実際の運転員に近い動作で、プラントの異常診断と対応操作を行う仮想運転員を構築した。さらに、仮想運転員の思考過程を発話させる機能を構築し、訓練生にプラントの異常診断の過程を深く理解させる機能も構築した。そして、VENUS を複数の被験者に試用してもらい、訓練生が、プラントの異常診断と対応操作の内容を理解できること、学習意欲を促進する可能性があることなどを確認した。

以上、本研究では、人工現実感技術を用いた機器保修の訓練環境及び、プラント運転の訓練環境を構築するための新たな手法を提案し、具体的な訓練課題に対する訓練環境の構築、および、訓練環境の構築支援システムの開発を通して、その有効性を確認した。

本研究で得られた成果は、今後、新たな訓練システムを開発する際に有効であるだけでなく、他の分野にも応用可能である。例えば、第3章で提案し、その有効性を確認した、仮想物体の状態遷移をペトリネットを用いてモデル化する手法は、三菱総合研究所が開発中である VR 応用保守訓練システムに適用される予定であり、現在、本研究で開発した PCSS のプログラムの一部が移植作業中である。また、第4章で提案した、プログラミングを行わずに訓練環境を構築する手法は、機器保修の訓練環境を構築する際に限らず、一般的な作業環境を構築する際にも応用できる。このような、人が仮想空間内に配置された物体と自由にインタラクションできる環境は、ヒューマンインタフェースの観点から眺めると、「直感的なインタフェース」として重要であり、高齢者には使いにくいと言われるパソコンを直感的に使用できるインタフェースの開発や、複雑大規模化した工学システムを直感的に操作できるインタフェースの開発等に応用できると考えられる。

また、第5章では、実際を人の動きに近い仮想運転員の自然な動作を合成できる、人体モーションの合成技術を開発したが、このような3次元映像として人の動作を合成する技術は、様々な分野に応用可能である。例えば、映画やコマーシャルを作成する際に仮想役者として利用することが考えられる。また、駅の案内システムや、自動販売機の擬人化インタフェースを開発する際に利用することも考えられる。さらに、今回開発した体験型教育システムでは、仮想運転員はプラントの異常診断と対応操作を行う能力しか備えていないが、将来的に仮想運転員に人とコミュニケーションする能力を付加することにより、人と仮想運転員が協力して作業を行うコラボレーションを実現するための要素技術となることも期待される。

6. 結論

このように、本研究で得られた成果は、様々な分野に応用することが可能であるが、今後、解決すべき課題も残されている。例えば、人工現実感を実現するシステムを開発する際、人とシステムとの接触点である入出力デバイスの選択肢が非常に少ないことは大きな問題である。第2章で述べたように、人工現実感を実現するシステムで使用するための様々なデバイスが開発されつつあるが、長時間使用してもユーザが疲労しにくく、自然な動作で使用可能であり、作業自体も行いやすいデバイスはまだ開発されていない。

パソコンが一般家庭まで普及し、各家庭が高速ネットワークで繋がれ始めている。各家庭がネットワークで繋がれることにより、ネットワークを介した配布が可能なソフトウェアは、開発されると同時に世界中で使用可能になる。従って、本研究で開発したような人工現実感を実現するためのソフトウェアも世界中に配布することが可能であるが、問題は、人工現実感技術が広範に利用できる安価なハードウェアがまだ普及していないことである。人工現実感の概念が提唱されてから、新聞やテレビ、インターネット等で話題になりながらも、人工現実感が一般家庭まで浸透していないのはハードウェアがまだ普及していないためであると考えられる。今後の人工現実感技術の普及には、ソフトウェアの改良だけでなく、コストダウンのための、ハード、ソフト両面の技術開発努力が要請される。

謝辞

本研究は、著者が京都大学工学部に在学していた1995年から現在に至るまで、京都大学大学院エネルギー科学研究科吉川榮和教授のご指導のもとに進めてきたものです。吉川榮和教授には、研究の方向性や進め方から、卒業論文や修士論文の書き方に至るまで、大変お忙しいにも関わらず、終始、熱意あふれるご指導とご鞭撻を賜りました。また、本論文をまとめるにあたって、終始適切で細やかなご指導を頂きました。ここに深甚なる感謝の意を表する次第であります。

京都大学大学院エネルギー科学研究科、手塚哲央助教授には、著者が吉川研究室に配属された当時、本研究室に講師として在任しておられ、本研究の基礎となる人工現実感やペトリネットに関する様々なご教示、ご指導を頂きました。また、本論文をまとめるにあたり、予備検討段階で、適切なお指導とご助言を頂きました。深く感謝の意を表します。

また、本論文のご査読、ご審査を頂くとともに、適切なお指導とご助言を頂きました京都大学大学院エネルギー科学研究科笠原三紀夫教授、京都大学大学院情報学研究科片井修教授、および関係諸先生方に対しましても深く感謝の意を表します。

京都大学大学院エネルギー科学研究科、下田宏助教授には、大変お忙しい中にも関わらず、本研究を遂行する上で、熱心な議論を通して貴重なご意見、ご助言、様々なご支援を頂きました。また、本論文を執筆するにあたり、様々なご指導と、激励を頂きました。さらに筆者が博士後期課程に進学するにあたり、様々な相談に乗って頂きました。深く感謝の意を表します。

東北大学大学院工学研究科、高橋信博士は、著者が吉川研究室に配属された当時、本研究室に助手として在任しておられ、研究者としてあるべき姿、学問に対する取り組み方を熱くご指導を頂きました。また、東北大学に赴任されてからも、様々なご指導を頂きました。深く感謝の意を表します。

松下電工株式会社、福島省吾博士には、本研究を遂行するにあたり、数多くのご意見やご助言を頂きました。また、筆者が博士後期課程に進学するにあたり、様々なご配慮、ご理解を頂き、ご厚情あふれる激励を頂きました。心から感謝申し上げます。

京都大学大学院エネルギー科学研究科博士後期課程、Wu Wei さんには、原子炉プラント、ヒューマンモデルなど、本研究を遂行するにあたり、様々なご意見、ご助言を頂きました。また、研究室のよき先輩として、公私にわたり、様々な相談に乗って頂きました。心から感謝申し上げます。

本研究を進めるにあたり、ご協力を頂きました谷友美さん、藤岡美紀さん、高塚真理さんをはじめとする吉川研究室の方々に感謝いたします。特に、小澤尚久君、大林史明君には、同じ博士後期課程の学生として、研究者として、よき友として、過酷なスケジュールに耐える私を支えて頂きました。また、柏健一郎さん(現在 日本ビューレットパッカード株式会社)、三谷拓也さん(現在 日本ビューレットパッカード株式会社)、安藤寛君(現在 富

6. 結論

士通株式会社)、市口誠道君、小牧大輔君には、本研究の遂行にご協力頂きました。ここに深く感謝いたします。

最後に、家庭にあっても、終始精神面、生活面で支えてくれ、筆者の研究生活を可能にしてくれた妻の令乃奈ほか家族にも深く感謝いたします。

著者発表文献リスト

第 2 章

総説論文

- [1] 下田宏, 石井裕剛, 山本倫也, 吉川榮和: プラント運転・保守の教育訓練への Virtual Reality 適用研究の展望, ヒューマンインタフェース学会誌, Vol. 1, No. 4, pp. 35-42, 1999.

第 3 章

原著論文

- [1] 吉川榮和, 手塚哲央, 柏健一郎, 石井裕剛: 仮想空間における機器保守訓練シミュレーション, 日本原子力学会誌, Vol. 39, No. 12, pp. 72-83, 1997.

国際会議議事録

- [1] H. Ishii, K. Kashiwa, T. Tezuka and H. Yoshikawa: Development of Machine Maintenance Training System using Petri Net and VR Environment, Proceedings of Cognitive Systems Engineering in Process Control, pp. 138-143, 1996.
- [2] H. Ishii, K. Kashiwa, T. Tezuka and H. Yoshikawa: A Study on Constructing a Machine-Maintenance Training System Based on Virtual Reality Technology, Proceedings of the 5th International Topical Meeting on Nuclear Thermal Hydraulics, Operations and Safety, pp. DD2-1 - DD2-6, 1997.
- [3] T. Tezuka, M. Yamamoto, H. Ishii and H. Yoshikawa: A Study on VR-based Plant-Maintenance Training System, Journal of Tsinghua University (Science and Technology), Vol. 37, No. S3, pp. 53-64, 1997.

口答発表

- [1] 石井裕剛, 柏健一郎, 手塚哲央, 吉川榮和: 人工現実感技術を用いた機器保守訓練システムへのペトリネットの適用, 第 40 回システム制御情報学会研究発表講演会予稿集, pp. 239-240, 1996.

- [2] 手塚哲央, 柏健一郎, 石井裕剛, 吉川榮和: 仮想空間における機器保守訓練システムの開発, Human Interface News and Reports, Vol. 11, No. 2, pp. 245-252, 1996.
- [3] 石井裕剛, 柏健一郎, 手塚哲央, 吉川榮和: 仮想空間における機器保守訓練システムのためのペトリネット構築支援システム, Human Interface News and Reports, Vol. 11, No. 2, pp. 253-260, 1996.
- [4] 手塚哲央, 石井裕剛, 柏健一郎, 吉川榮和: 仮想空間における機器保守訓練シミュレーションシステム, 日本原子力学会 1997 年秋の大会予稿集, 第 2 分冊, p. 347, 1997.

第 4 章

原著論文

- [1] 石井裕剛, 手塚哲央, 吉川榮和: 人工現実感技術を用いた機器保守の訓練環境構築支援システムの開発, 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 1, pp. 303-312, 1999.

国際会議議事録

- [1] H. Ishii, T. Tezuka and H. Yoshikawa: A Study on Design Support for Constructing Machine-Maintenance Training System by Using Virtual Reality Technology, Proceedings of the 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, pp. 389-394, 1998.
- [2] H. Ishii, T. Tezuka and H. Yoshikawa: A Study on Design Support System for Constructing Machine-Maintenance Training Environment Based on Virtual Reality Technology, Proceedings of 1998 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2635-2640, 1998.
- [3] H. Ishii, T. Tezuka and H. Yoshikawa: User-Interface Design of the Support System for Constructing Virtual Environment, Proceedings of the 8th International Conference on Human-Computer Interaction, Vol. 2, pp. 1182-1186, 1999. (Invited Paper Presentation)

口答発表

- [1] 石井裕剛, 手塚哲央, 吉川榮和: 人工現実感技術を用いた機器分解組立訓練システムの設計支援に関する研究, 第 13 回ヒューマン・インタフェース・シンポジウム論文集, pp. 33-38, 1997.

- [2] 石井裕剛, 手塚哲央, 吉川榮和: 人工現実感技術を用いた機器保守訓練環境の構築支援システムの開発, 日本原子力学会 1998 年春の年会要旨集, 第 2 分冊, p. 239, 1998.
- [3] 石井裕剛, 手塚哲央, 吉川榮和: 仮想空間設計支援システムの開発, Human Interface News and Reports, Vol. 13, No. 2, pp. 237-242, 1998.
- [4] 石井裕剛, 手塚哲央, 吉川榮和: 人工現実感技術を用いた機器分解組立訓練システムの設計支援に関する研究 (第 2 報), 第 14 回ヒューマン・インタフェース・シンポジウム論文集, pp. 479-484, 1998.

第 5 章

原著論文

- [1] 石井裕剛, Wu Wei, 下田宏, 吉川榮和: 人工現実感技術を用いた原子力プラント運転の体験型教育システムの開発, ヒューマンインタフェース学会投稿中.
- [2] 石井裕剛, 安藤寛, 下田宏, 吉川榮和: 仮想空間内におけるプラント運転員の自然な人体モーションの生成, システム制御情報学会投稿準備中.

国際会議議事録

- [1] H. Ishii, W. Wu, D. Li, H. Ando, H. Shimoda and H. Yoshikawa: A Basic Study of Virtual Collaborator - The First Prototype System Integration -, Proceedings of the 4th International Symposium on Artificial Life and Robotics, Vol. 2, pp. 682-685, 1999. (Invited Paper Presentation)
- [2] H. Ishii, W. Wu, D. Li, H. Shimoda and H. Yoshikawa: Development of a VR-based Experienceable Education System -A Cyber World of Virtual Operator in Virtual Control Room -, Proceedings of the 3rd World Multiconference on Systemics, Cybernetics and Informatics and the 5th International Conference on Information Systems Analysis and Synthesis, Vol. 1, pp. 473-478, 1999.
- [3] S. Shimoda, H. Ishii, W. Wu, D. Li, T. Nakagawa and H. Yoshikawa: A Basic Study on Virtual Collaborator as an Innovative Human-Machine Interface in Distributed Virtual Environment - The Prototype System and Its Implication for Industrial Application -, Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics, pp. V-697 - V-702, 1999. (Invited Paper Presentation)
- [4] N. Ichiguchi, D. Komaki, H. Ishii, H. Shimoda and H. Yoshikawa: Experimental Analysis on Human Walking Animation of Virtual Robot in 3D Visual Space, Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics, pp. VI-918 - VI-923, 1999. (Invited Paper Presentation)

- [5] H. Shimoda, H. Ishii, W. Wu, D. Li, T. Nakagawa and H. Yoshikawa: Virtual Operator in Virtual Control Room - The Prototype System Implementation -, Proceedings of the 18th European Annual Conference on Human Decision Making and Manual Control, 1999.
- [6] H. Yoshikawa, H. Shimoda, W. Wu, H. Ishii and K. Ito: Development of Virtual Collaborator as an Innovative Interface Agent System between Human and Plant Systems: Its Framework, Present Status and Future Direction, Proceedings of the 5th International Symposium on Artificial Life and Robotics, pp. 693-698, 2000.

口答発表

- [1] 吉川榮和, 下田宏, Wu Wei, 石井裕剛, 小澤尚久: 新しいインタフェース環境「バーチャルコラボレータ」の実験研究 (1) - 全体構想 -, 日本原子力学会 1999 年春の年会要旨集, 第 2 分冊, p. 278, 1999.
- [2] 市口誠道, 安藤寛, 石井裕剛, 下田宏, 吉川榮和: 新しいインタフェース環境「バーチャルコラボレータ」の実験研究 (5) - 人体モーションの合成 -, 日本原子力学会 1999 年春の年会要旨集, 第 2 分冊, p. 282, 1999.
- [3] 石井裕剛, Wu Wei, 李徳衡, 下田宏, 吉川榮和: 新しいインタフェース環境「バーチャルコラボレータ」の実験研究 (7) - プロトタイプシステム -, 日本原子力学会 1999 年春の年会要旨集, 第 2 分冊, p. 284, 1999.
- [4] 下田宏, 福島省吾, 小澤尚久, 青竹雄介, 梅田直樹, 二階堂義明, Wu Wei, 石井裕剛, 吉川榮和: 新しいインタフェース環境「バーチャルコラボレータ」の実験研究 (8) - デバイス開発と将来構想 -, 日本原子力学会 1999 年春の年会要旨集, 第 2 分冊, p. 285, 1999.
- [5] 石井裕剛, Wu Wei, Li Deheng, 下田宏, 吉川榮和: 人工現実感技術を用いた原子力プラント運転の体験型教育システムの開発, 電気学会研究会資料, 原子力研究会, NE-99-7-12, pp. 19-24, 1999.
- [6] 小牧大輔, 市口誠道, 石井裕剛, 下田宏, 吉川榮和: VR による人の歩行動作の効率的表現法の研究, ヒューマンインタフェース学会研究会報告集, Vol. 1, No. 2, pp. 19-24, 1999.
- [7] 市口誠道, 石井裕剛, 小牧大輔, 下田宏, 吉川榮和: 仮想空間におけるアフォーダンスを利用した人体モーションの合成, ヒューマンインタフェースシンポジウム'99 論文集, pp. 211-216, 1999.
- [8] 石井裕剛, Wu Wei, 李徳衡, 下田宏, 吉川榮和: 新しいインタフェース環境「バーチャルコラボレータ」の実験研究 (第 2 報) - 体験型教育システムへの応用 -, 日本原子力学会 1999 年秋の大会予稿集, 第 2 分冊, p. 468, 1999.

- [9] 吉川榮和, 下田宏, Wu Wei, 石井裕剛, 小澤尚久: バーチャルコラボレータ: 新しい
ヒューマン・マシン・コミュニケーションアプローチ, 計測自動制御学会第38回学術
講演会予稿集, Vol. 2, pp. 581-582, 1999.

付録 A 研究の概要（カラー版）

人工現実感技術を用いた プラント運転・保守の訓練システムに関する研究

石井 裕剛

1

研究の概要

原子力発電プラントの運転員や保守作業員の人的信頼性を向上
知識・技能の効果的な伝承方法

人工現実感による仮想体験と情報可視化

- ◆機器保守の訓練システムに関する研究
仮想体験環境を少ない労力で短時間に構築できる手法を開発
- ◆プラント運転の訓練システムに関する研究
人工現実感を用いて運転員のプラント監視制御行動を可視化

2

研究の背景：人工現実感

- ❖人工現実感(Virtual Reality; VR)
実在しない環境を現実のような感覚を持って体験できる
新しいヒューマンインタフェース技術
- ❖人工現実感の応用分野
 - ◆設計支援
 - ・仮想製品の試用、インテリアデザイン
 - ◆医療分野
 - ・手術支援、患部可視化・可触化
 - ◆臨場感通信
 - ・遠隔会議、テレイグジスタンス

本研究のテーマ：人工現実感の教育訓練分野への応用

3

研究の背景：訓練センターにおける教育訓練の現状

- ❖プラントの保守作業の訓練
 - ◆座学形式の講義と実技訓練を併用
 - ◆数百人規模の訓練生
 - ◆訓練対象の機器の種類が大
- ❖プラントの運転操作の訓練
 - ◆座学形式の講義と実技訓練を併用
 - ◆多種多様な異常事象
 - ◆幅広い知識と技術、論理的な思考能力



保守作業訓練



プラント運転操作訓練

4

研究の着眼点(1)

❖訓練センターでの教育訓練

◆作業を実際に体験できる実技訓練が重要

- 実物大模型は非常に高価
- 実作業と同等の危険性
- インストラクタが必要



十分な実技訓練が行われていない

人工現実感を用いた**仮想体験**により訓練効果を上げる

❖教育訓練への人工現実感の適用による効果

- ◆訓練費用の削減や安全性の向上による訓練機会の増加
- ◆訓練効率の向上による訓練期間の短縮
- ◆模型では実現が困難な作業環境における体験訓練

5

研究の着眼点(2)

❖保守作業を教育対象とした訓練システム

◆訓練環境を構築する際の作業の効率化

仮想空間の更新の仕方の設定作業



時間と労力が必要

- 仮想物体の状態遷移をペトリネットを用いてモデル化
- プログラミングを行わずに訓練環境を構築

❖運転操作を教育対象とした訓練システム

訓練生が自ら制御盤を操作



訓練方式を発展

- プラント監視制御行動中の運転員の行動を可視化
- 制御室内に没入し、運転員の行動を仮想体験

6

研究の目的

研究の目的

- ◆ VR技術を用いた訓練環境を構築するための新たな手法を提案し、実用的な訓練システムを開発するための基礎技術を確立する

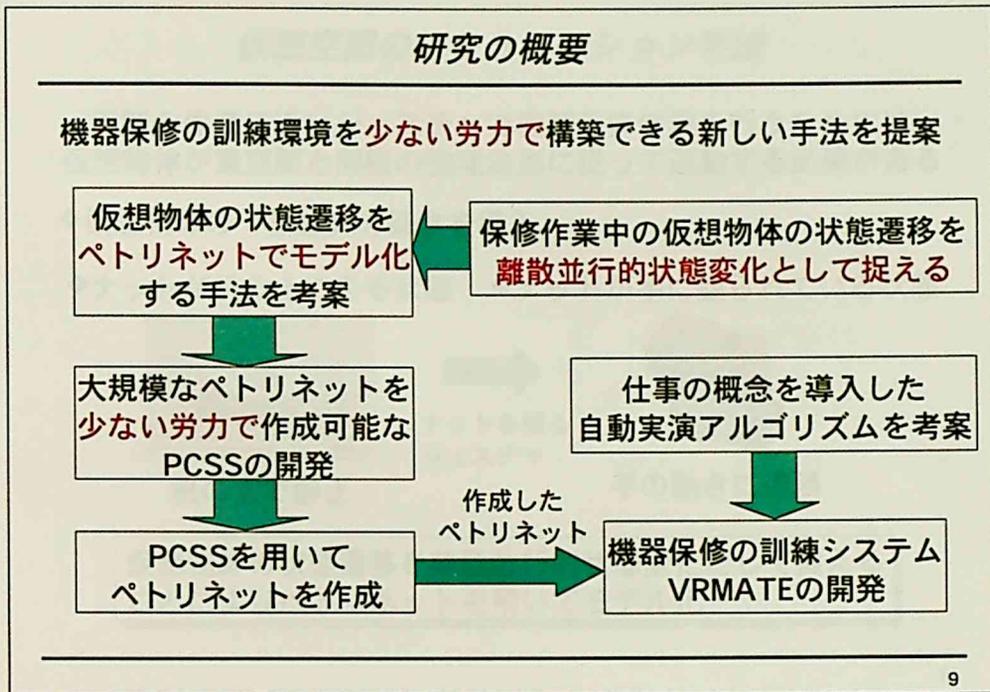
研究テーマ

- (1) 機器保守作業を仮想体験できる訓練環境を少ない労力で構築できる手法の開発
- (2) 計算機に関する専門的な知識を持たない人でも訓練環境を構築できる支援システムの開発
- (3) 運転員のプラント監視制御行動を観察し、プラント運転を仮想体験できる教育システムの開発

7

人工現実感技術を用いた 機器保守の訓練システムVRMATEの開発

8



- 機器修理の訓練システムに必要な機能**
-
1. システムを使用するための特別な知識や技能を必要としない
◆実際の作業に近いジェスチャで機器操作が可能
 2. 自由に作業を進めることができる
◆マニュアルに記載された作業手順以外の作業も実行可能
 3. 機器修理の作業手順を修得できる
◆機器修理の作業を実演する自動実行機能を構築
 4. 訓練対象の機器の構造を理解できる
◆自由に視点を変更できる機能を構築
- 訓練の対象**
スイング式逆止弁の分解作業
-
- 10

仮想空間のシミュレーション手法

実際の作業に近いジェスチャで機器保守作業を行うためには仮想物体が実空間と同様の物理法則に従って運動する必要がある

❖仮想物体の「状態」の概念を導入

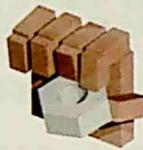
◆ナットが机の上にある状態 ◆ナットが手に握られている状態



机の上で静止



ナットを握る
ジェスチャ



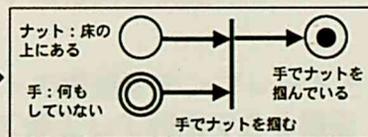
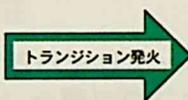
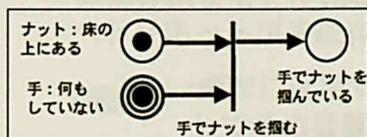
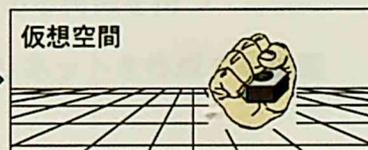
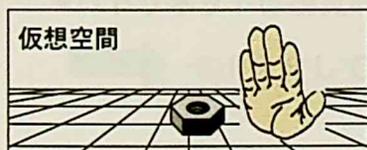
手の動きに追随

仮想物体の状態遷移を**離散並行的状態変化**として捉えることにより**ペトリネット**を用いてモデル化

ペトリネットによる仮想物体の状態遷移のモデル化手法

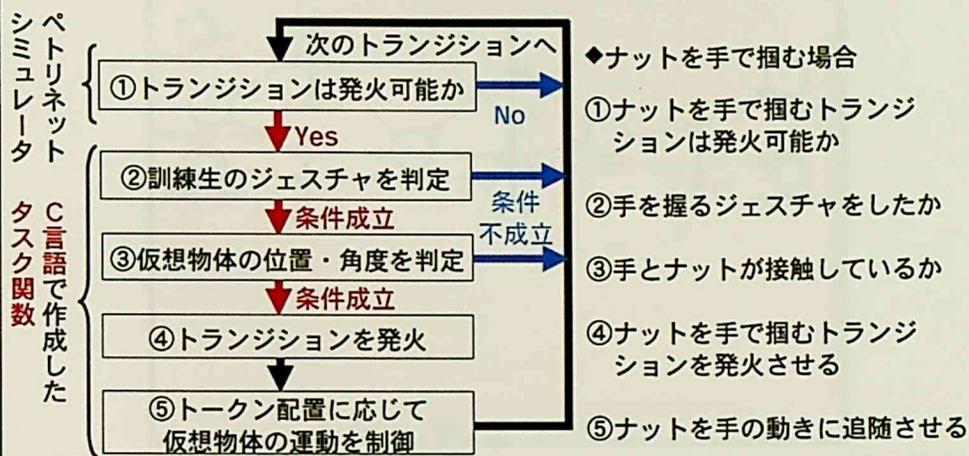
1. 仮想物体の1つの状態を1つのプレースで表現する
2. 仮想物体がある状態にあることを対応するプレースにトークンが存在することで表現する
3. 訓練生の動作の実行をトランジションの発火に対応させる

➡ 単位タスク



ペトリネットを用いた仮想空間のシミュレーション

- ◆ジェスチャと仮想物体の位置・角度に応じてトランジションを発火
- ◆ペトリネット上のトークン配置に応じて仮想物体の運動を制御



13

ペトリネット作成支援システムPCSSの開発

ペトリネットを作成する際の労力を軽減するためにPCSSを開発

◆PCSS開発の着眼点

- ◆ペトリネットは視覚的に表現可能である特徴を持つ

➡ GUIを介して視覚的にペトリネットを作成する機能

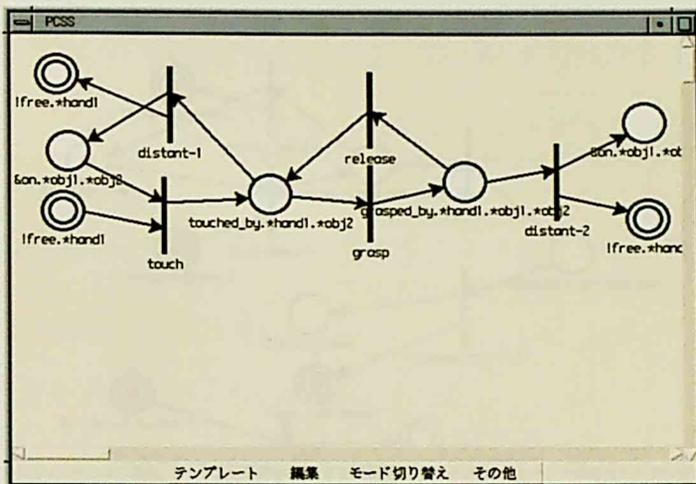
- ◆保守作業をモデル化したペトリネットには同じ構造のペトリネットが多数存在

➡ 頻繁に使用するペトリネットの構造をテンプレートとして予め登録し必要に応じて修正して利用できる機能

14

PCSSの概要 (テンプレート作成モード)

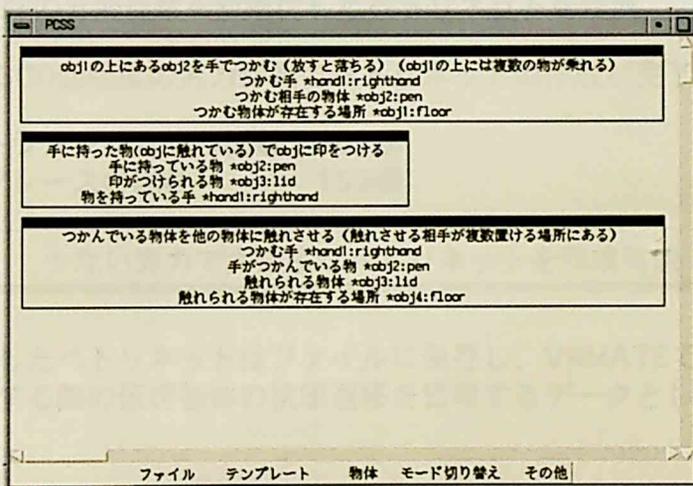
◆ペトリネット構造をプレス・トランジション単位で作成



15

PCSSの概要 (テンプレート結合モード)

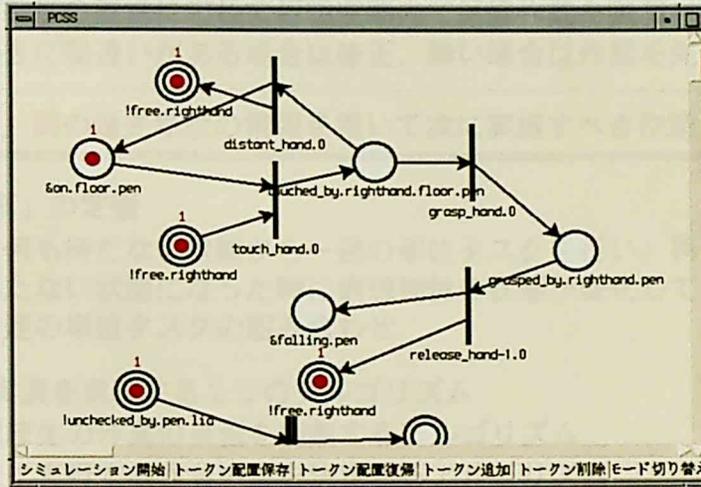
◆テンプレートを結合して大規模なペトリネットを作成



16

PCSSの概要 (動作確認モード)

◆ペトリネットの動作を確認



17

PCSSの使用例

❖逆止弁の分解作業を対象にしたペトリネットを作成

◆約300回程度の入力作業でペトリネットの作成が完了

- トランジションの総数 367個
- プレースの総数 158個

少ない労力で大規模なペトリネットを作成可能

作成したペトリネットはファイルに保存し、VRMATEで訓練を実施する際の仮想物体の状態遷移を管理するデータとして使用

18

機器保守の自動実演機能

❖自動実演機能

- ◆訓練生の要求に応じて仮想空間内で保守作業を実演
- ◆過去に間違いがある場合は修正、無い場合は作業を先に進める

「仕事」間の優先順位の情報を用いて次に実施すべき作業を決定

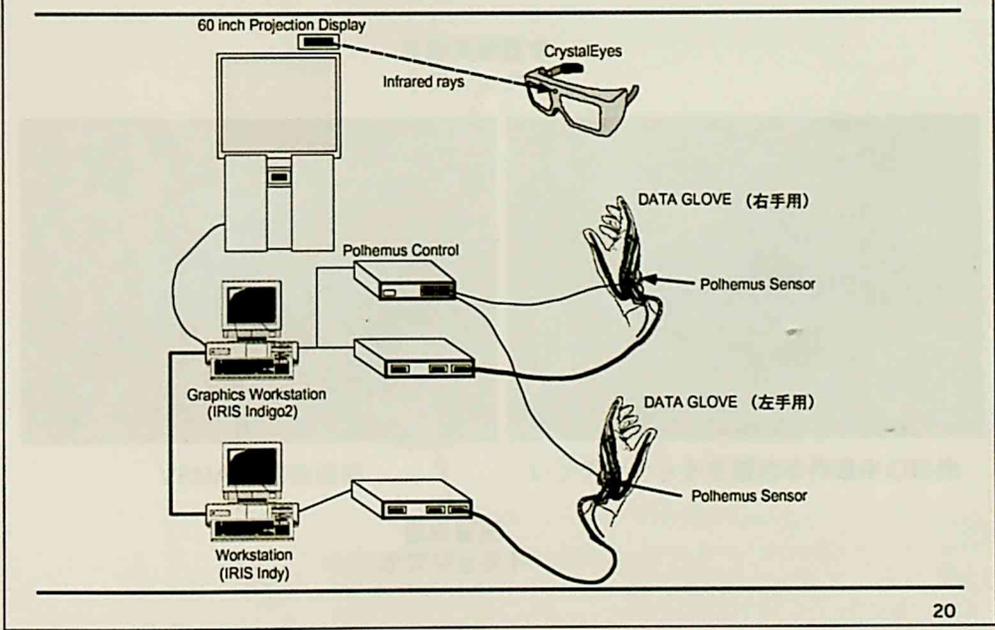
❖「仕事」の定義

手に何も持たない状態から一連の単位タスクを行い、再び手に何も持たない状態になった時に仮想物体の状態が変化している場合の一連の単位タスクの組み合わせ

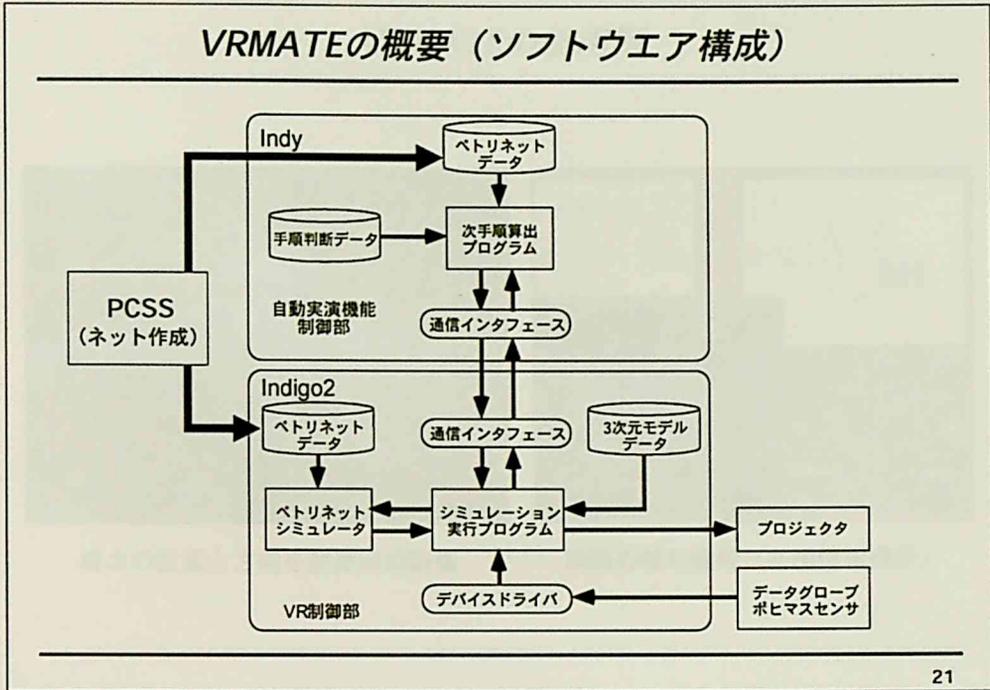
❖自動実演を実現する2つのアルゴリズム

- ◆訓練生の作業の良否を判断するアルゴリズム
- ◆次に実行する作業内容を決定するアルゴリズム

VRMATEの概要 (ハードウェア構成)



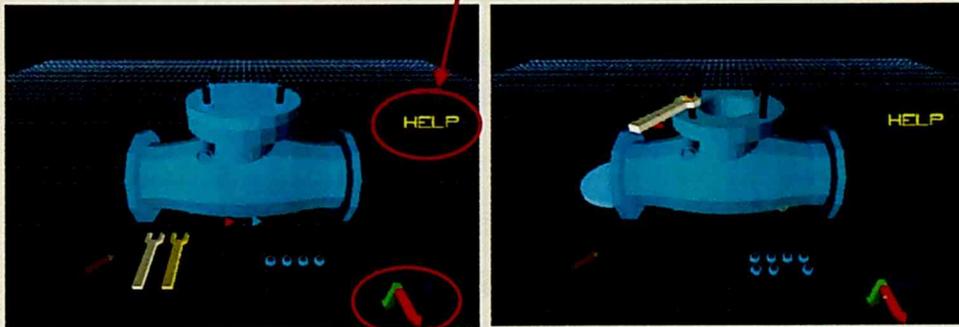
VRMATEの概要 (ソフトウェア構成)



21

VRMATEの使用例

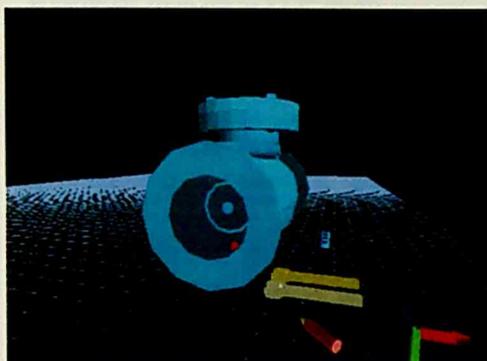
自動実演要求
オブジェクト



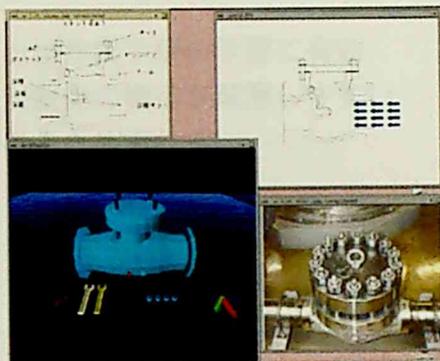
視点変更
オブジェクト

22

VRMATEの使用例



視点の位置と方向を変更後の映像



訓練の補助機能（各種情報提示）

23

VRMATEの評価

大学院生 4 人を訓練生として機器保修の訓練を実施

❖実験の目的

人工現実感技術を用いた機器保修の訓練システムの有効性を確認すると共に今後解決すべき課題を抽出する

❖実験の方法

- ◆仮想物体の詳しい操作方法は説明しない
- ◆自動実演を使用する方法は説明
- ◆訓練生は自らの判断で自由に作業を実施
- ◆体験訓練の終了後、自由に感想を述べてもらう

24

評価の結果

◆訓練生から得られた感想

- 間違った作業でも実施できることは作業手順の学習に有効
- 視点の移動が容易にできるため対象物の構造が理解しやすい
- 作業中に腕を挙げている必要があるため疲れる

◆訓練の様子を観察することにより判明

- 自動実演機能は作業手順を全く知らない初心者には有効
- 具体的な作業方法を説明しなくても作業が行えていた
- 力覚フィードバックがないため必要以上の力が入り疲労しやすい

25

考察および将来課題(1)

❖データグローブやポヒマスセンサを用いた入力デバイスについて

- ◆特別な予備知識を必要としないインタフェースを提供できる
- ◆力覚フィードバックがないために訓練生が疲労しやすい

 力覚フィードバックを提供するデバイスを使用するか、
力覚フィードバックの欠如を補う必要がある

❖自動実演機能について

- ◆作業手順を学習する際に有効である
- ◆現時点でのアルゴリズムは作成すべき情報量が多い
- ◆逆止弁の分解作業以外には適用できない可能性がある

 一般的な機器保守作業に適用できる自動実演の
アルゴリズムを考案する必要がある

26

考察および将来課題(2)

❖提案した訓練環境の構築手法について

◆仮想物体の状態遷移の仕方を定める作業

- ・ペトリネットを用いてモデル化する手法の開発
- ・ペトリネット作成支援システムPCSSの開発

 少ない労力で短時間で仮想物体の状態遷移の仕方を定めることが可能になった

◆仮想物体の運動の仕方を定める作業

- ・C言語を用いてタスク関数を作成
- ・多くの時間と労力が必要

 より少ない労力で短時間に構築できる手法の開発が必要

27

まとめ

❖機器保守の訓練環境を少ない労力で構築できる新たな手法を提案

- ◆仮想物体の状態遷移をペトリネットを用いてモデル化する手法
- ◆ペトリネット作成支援システムPCSSの開発
- ◆スイング逆止弁の分解作業を対象に訓練システムVRMATEを開発



- ◆仮想物体の状態遷移の仕方は少ない労力で設定することが可能
- ◆仮想物体の運動の仕方を定めるタスク関数を作成する作業が繁雑
- ◆より少ない労力で短時間に訓練環境を構築できる手法が必要

28

機器保守の訓練環境の構築支援システム
DESCORTEの開発

29

研究の概要

訓練環境の構築から訓練実施まで一貫して行える支援システムを開発

❖開発の方針

- ◆プログラミングを全く行わなくても訓練環境を構築できる
- ◆初めて使用する場合でも訓練環境を容易に構築できる
- ◆訓練の実施環境も提供する

❖開発の手順

1. 訓練環境を構築する際に作成する必要がある情報を検討
2. 必要な情報を入力するためのGraphical User Interfaceを設計
3. インタフェースの評価実験
4. 評価結果をもとにインタフェースを改良

30

DESCORTEで構築可能な訓練環境が具備すべき要件

VRMATEを開発した際に判明した事柄をもとにDESCORTEで構築可能な訓練環境の仕様を決定

1. 実際の作業に似たジェスチャで機器を操作できる
 - ◆Logitech社の3次元マウスMagellanを採用
2. 視点の自由な移動が可能
3. 仮想空間内での作業が行いやすい
 - ◆奥行き知覚を改善するための機能
 - ・仮想物体に影を付ける機能
 - ◆力覚フィードバックの欠如の補完
 - ・イベントを起こした場合に音を発する機能

31

DESCORTE開発の基本方針

実際の作業に似たジェスチャでの機器操作を可能にするにはジェスチャに応じた仮想物体の運動の仕方を予め定める必要がある

| | VRMATEの開発 | DESCORTE |
|--------------|---------------|--------------------------|
| ① 仮想物体の運動の仕方 | タスク関数内にC言語で記述 | GUIを介して予め選定されたパラメータの値を設定 |
| ② 仮想物体の状態遷移 | ペトリネットでモデル化 | GUIを介してモデル化したペトリネットを入力 |
| ③ ジェスチャ判定 | タスク関数内にC言語で記述 | GUIを介してイベントの種類を選択 |

❖①の情報
運動データベース

❖②及び③の情報
状態遷移データベース

32

運動データベース(1)

❖設定可能なパラメータの種類

訓練センターで実施されている機器保守作業の訓練環境を構築する際に設定できる必要がある情報を分析して決定

| | 仮想空間内で模擬する内容 | 設定できる必要がある物体の運動 |
|---|---------------------|---------------------------------------|
| 1 | 手で物体を握って移動 | •手の動きに応じて平行移動と回転を同時に行う |
| 2 | 物体が自由落下 | •等加速度運動 |
| 3 | スパナでボルトにはまったナットを緩める | •同時に複数の物体が1つの軸を中心に回転 •回転角度に応じた並進運動 |
| 4 | モータの回転 | •時間に従って回転角度が変化 |

33

運動データベース (2)

❖DESCORTEで運動データベースとして設定可能なパラメータ

物体の外見に関する情報

- ◆3次元形状データ
 - 形状ファイル名を指定
- ◆テクスチャデータ
 - 画像ファイル名を指定
- ◆描画方法
 - ポリゴン
 - ワイヤフレーム
 - 透明
 3つから選択

物体の運動の仕方に関する情報

- ◆時間の推移に応じて変化する場合
 - 物体の位置と角度を時間の関数として設定
(式の係数に数値を入力)
- ◆3次元マウスからの入力に応じて変化する場合
 - マウスからの入力値と物体の位置と角度の変化の関係を設定
(項目の選択と数値の入力)
- ◆各状態における初期位置
 - 3次元マウスを用いて位置と角度を指定

34

状態遷移データベース(1)

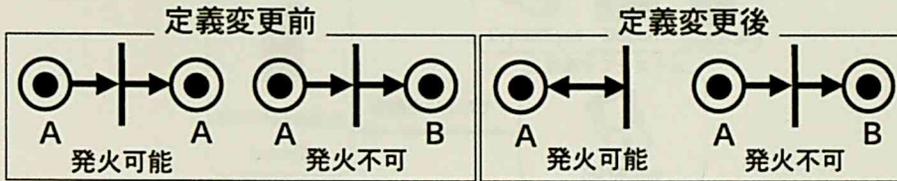
❖ユーザが入力する情報

1. 仮想物体の状態遷移をモデル化した改良ペトリネット
2. トランジションを発火させるイベントの種類

❖ペトリネットの主な変更点

◆参照アークの導入

➡ ユーザの視覚的認知負荷を軽減し間違いを予防

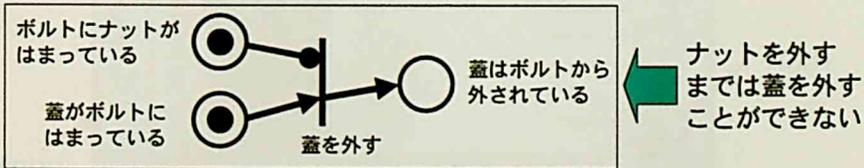


35

状態遷移データベース(2)

◆抑止ブレース・抑止アークの導入

➡ 物体間の拘束関係がモデル化し易くなる

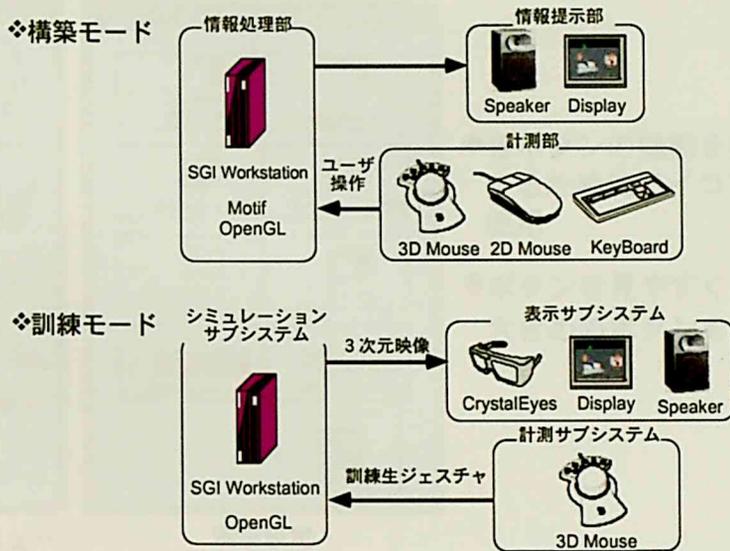


❖選択可能なイベント

| | イベント | 例 |
|---|----------------|------------|
| 1 | 「物1」と「物2」が接触する | スパナと床が接触する |
| 2 | 「物1」と「物2」が分離する | 右手がペンから離れる |
| 3 | 手を開く | 右手を開く |
| 4 | 手を閉じる | 左手を閉じる |
| 5 | 条件が成立する | ペンを握っている |

36

DESCORTEの概要 (ハードウェア構成)

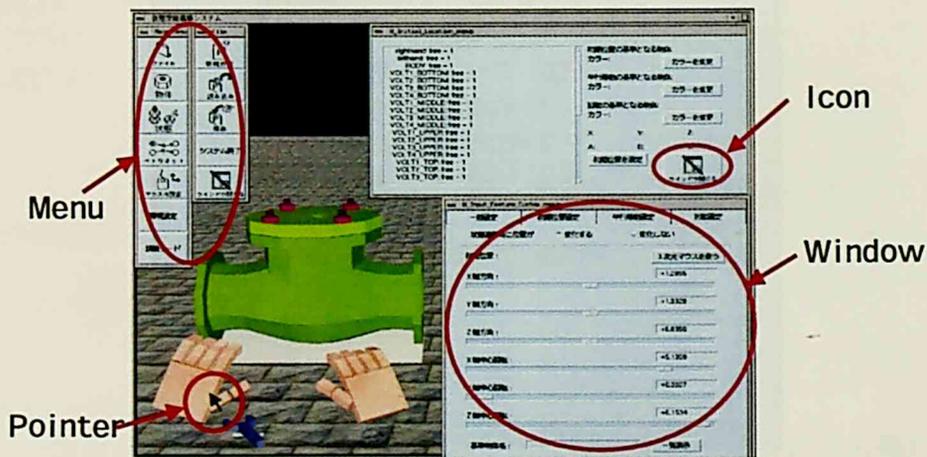


37

DESCORTEの概要
(情報を入力する際のインタフェース1)

❖WIMPインタフェースを基本として設計

➡ ユーザが普段使用しているインタフェースとの間で一貫性を保つ



38

DESCORTEの概要 (情報を入力する際のインタフェース2)

Main Menu

- ファイル
- 物体
- 状態
- ベトリネット
- マウスの設定
- 環境設定
- 試験モード

メニュー

環境設定

| | |
|------------------------------------|------------------------------------|
| <input type="checkbox"/> 影を付ける | <input type="checkbox"/> 影を付けない |
| <input type="checkbox"/> テクスチャ使用 | <input type="checkbox"/> テクスチャ不使用 |
| <input type="checkbox"/> ワイヤフレーム | <input type="checkbox"/> ポリゴン |
| <input type="checkbox"/> 地図テクスチャ有り | <input type="checkbox"/> 地図テクスチャ無し |
| <input type="checkbox"/> ウィンドウ閉じる | |

環境設定

- ◆各ボタンの機能を推測可能にするためにアイコンボタンを使用
- ◆ボタンを見やすくするために大きめのボタンを使用

39

DESCORTEの概要 (情報を入力する際のインタフェース3)

状態テンプレート

静止状態

手を開いた状態 (自由に動く) (左手)

手を握った状態 (自由に動く) (右手)

左手に握られた状態 (自由に動く)

右手に握られた状態 (自由に動く)

手を握った状態 (自由に動く) (左手)

自由落下

手を開いた状態 (自由に動く) (右手)

適用する

取り消し 設定

頻繁に使用するパラメータの組み合わせを予め登録できるテンプレート機能

40

DESCORTEの評価実験(1) (複雑な訓練環境の構築)

VRMATEの開発の際に構築した訓練環境をDESCORTEを用いて構築

❖実験の目的

- ◆プログラミングによりVRMATEを開発した際の作業効率とDESCORTEを用いて訓練環境を構築した際の作業効率を比較

❖実験の条件

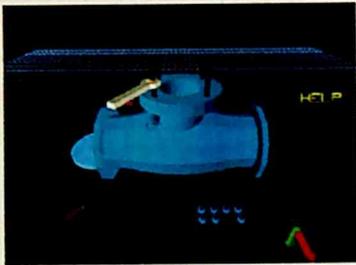
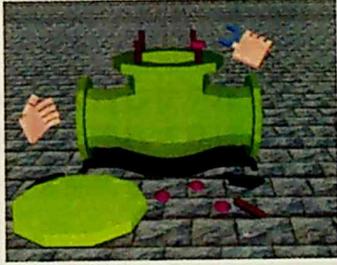
- ◆DESCORTEの操作に習熟したユーザが1人で環境を構築
- ◆構築開始から終了までの時間を計測
- ◆仮想物体の形状ファイル、テクスチャ等は作成済み
- ◆自動実演機能は構築しない

❖構築の対象とする訓練環境

- ◆スイング式逆止弁の分解作業
- ◆約100回のジェスチャにより作業が終了

41

実験結果

| | VRMATE | DESCORTE |
|-------------|---|--|
| 作業者 構築時間 | プログラマ2人 約3ヶ月 | ユーザ1人 約12時間 |
| 訓練画面 |  |  |

❖作業時間短縮の要因

- ◆訓練作業を可能にするためのアルゴリズムを考案する必要がない
- ◆作業工程が単純化されたために間違いが減少

42

DESCORTEの評価実験(2) (単純な訓練環境の構築)

DESCORTEを初めて使用する初心者には簡単な訓練環境を構築させる

❖実験目的

初心者でも訓練環境を正しく構築できることを確認する

❖実験条件

◆被験者2名(被験者A:20歳代、被験者B:40歳代)

・DESCORTEの使用経験なし

・2次元マウスとキーボードを不自由なく使用できる

◆操作方法を約30分間説明後、「手でペンを握る仮想環境」の構築作業を観察

◆作業開始から終了までの時間を計測、終了後アンケートを実施

❖構築対象

・スイッチを操作しモータの回転を制御する訓練環境

43

実験結果

❖実験結果

◆両被験者共に正しく訓練環境を構築できた

◆構築所要時間

・被験者A:41分 被験者B:38分

◆アンケートの結果

・3次元マウスが使いにくく、物体の位置合わせが難しかった

・ウィンドウを開いている際に仮想空間が隠れて見にくかった

・使っているうちに、もっと凝った仮想空間を作りたいと思うようになった



・初めて使用する場合でも正しく訓練環境を構築

・インタフェースに改善すべき点があることが判明

44

DESCORTEのインタフェースの改良

❖評価実験の結果をもとに、ユーザの認知的作業負荷を軽減させる方針でインタフェースを改良する

1. 仮想物体の位置・角度を指定するインタフェース

◆視点移動と物体の位置・角度を指定する方法を明確に分ける

- ➡ ・3次元マウスは視点変更のみに使用
- ・2次元マウスを用いて仮想物体の位置・角度を指定

2. 数多くのウインドウを開閉する必要がある問題

◆データベースを入力する際のシステムの操作回数を減らす

- ➡ ・ボタン等を小さくデザイン
- ・できる限り1つのウインドウに集約

45

DESCORTEのインタフェースの改良

3. アイコンを使用したボタンが仮想空間の表示領域を隠す問題

◆仮想空間を観察しながらの作業を可能にするために仮想空間を表示する領域を大きく確保

- ➡ ・アイコンの使用を仮想空間を見る必要がない場合に制限

4. 設定すべきパラメータの数が多い問題

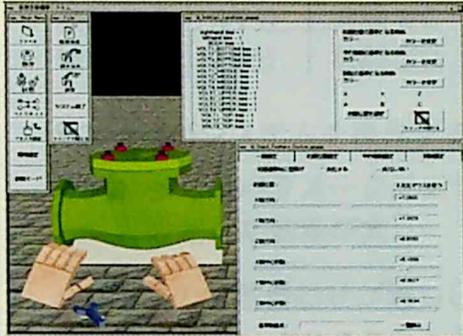
◆データベースを入力する際のシステムの操作回数を減らす

- ➡ ・運動データベースを構成する各パラメータに予め値を設定
- ・ユーザが最初に作成した仮想物体の状態を、訓練開始時の仮想物体の状態として設定

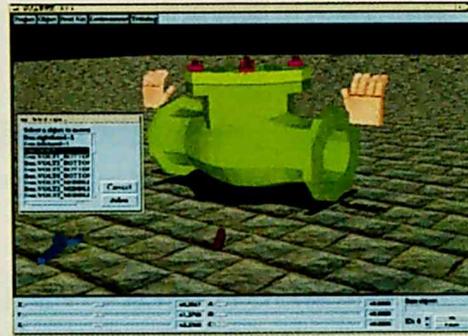
46

改良後のDESCORTEのインタフェース

インタフェース改良前



インタフェース改良後



47

インタフェース改良後のDESCORTEの評価実験

❖実験目的

- ◆インタフェースの改良による効果进行评估する

❖実験条件

- ◆インタフェース改良前のDESCORTEの評価実験と同じ
- ◆被験者2名（被験者C、被験者D）共に20歳代

❖実験結果

- ◆両被験者共に正しく訓練環境を構築できた
- ◆構築所要時間
被験者C：20分 被験者D：16分
- ◆アンケートの結果
 - 思ったより簡単に仮想空間を構築することができた
 - ペトリネットには少し慣れが必要だと感じた
 - 物体の動きを数式で表す代わりに「自由落下」「回転」等抽象的な言葉で設定できるようにしてほしい

48

考察

❖ インタフェース改良前と改良後の作業効率の比較

| 改良前 | | 改良後 | |
|------|-----|------|-----|
| 被験者A | 41分 | 被験者C | 20分 |
| 被験者B | 38分 | 被験者D | 16分 |

作業時間がさらに短縮

- ◆ ユーザがシステムを操作する回数がかなり減少

➡ 多くのパラメータに対して頻繁に使用する値を予め設定した効果が大

- ◆ 「物体の位置合わせが難しい」という意見は得られなかった

➡ 2次元マウスを用いた物体の位置・角度合わせのインタフェースへ変更した効果が現れた

まとめ

プログラミングを行わなくても機器保修の訓練環境を構築できる支援システムDESCORTEを開発

- ❖ 訓練環境を構築する際に必要となる情報を分析

- ◆ 状態遷移データベース
- ◆ 運動データベース

- ❖ WIMPインタフェースを基本として設計

- ◆ プログラミングによる構築手法に比べ短時間で構築可能
- ◆ 初心者でも簡単な訓練環境なら構築可能
- ◆ インタフェースに改良すべき点があることが判明

- ❖ インタフェースの改良

- ◆ ユーザの認知的作業負荷を軽減
- ◆ 更に短時間で訓練環境を構築できることを確認

DESCORTE実用化への課題

- ❖ データベースを作成する際のインタフェースのさらなる改良
 - ・ ペトリネット内のトランジションやプレースを検索する機能
 - ・ ボタン1つで仮想物体の位置合わせができる機能
- ❖ 訓練を実施する際の支援機能
 - ・ 訓練生の作業の結果を記録する機能
- ❖ 自動実演機能を提供するためのデータベース作成支援機能

51

人工現実感技術を用いた原子カプラント運転の
体験型教育システムVENUSの開発

52

研究の概要

プラント監視制御行動中の運転員の行動を
人工現実感を用いて総合的に可視化



仮想運転員の監視制御行動を観察することにより
訓練効果を上げる教育システムVENUSを開発

- ◆プラント緊急時の運転員認知行動を模擬できる既開発の
ヒューマンモデルシミュレータを仮想運転員の頭脳として採用
- ◆人の自然な動作を合成する人体モーション合成技術を開発
- ◆運転員の思考過程を発話する機能
- ◆運転員のプラント監視制御行動を仮想体験する機能

53

VENUSに必要とされる機能

- ❖原子力発電プラントの運転操作
 - ◆異常が発生した場合に、迅速に原因を特定し対応操作を行う

「何時、どのように考えて、どのように実行するか」の理解が必要

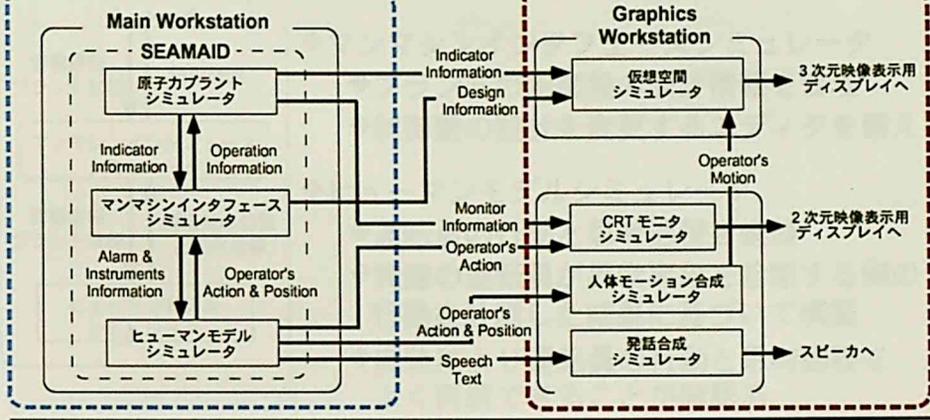
1. 実際の運転員と同様の動作で運転操作を行う様子を観察できる
 - ◆運転員の行動と思考過程を模擬するヒューマンモデルを採用
 - ◆運転員の自然な動作を合成する人体モーション合成技術を開発
2. プラント運転員の思考過程も理解できる
 - ◆仮想運転員が思考過程をリアルタイムに発話する機能
3. プラント運転操作を仮想体験できる
 - ◆仮想運転員の視点からの映像を立体視しながら眺める機能

54

VENUSの概要 (ソフトウェア構成)

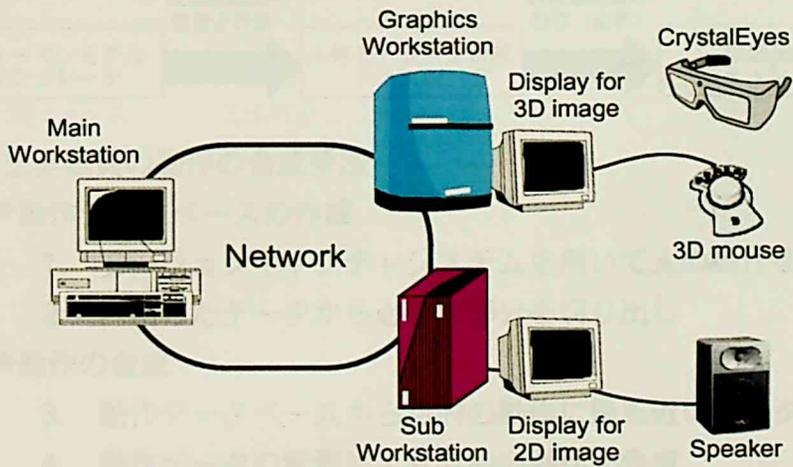
- ❖分散型シミュレーションシステムとして構築
 - ◆シミュレーションを実行する際の計算負荷を分散
 - ◆拡張性が向上

中川 (三菱電機) Wu (京都大学) らが開発

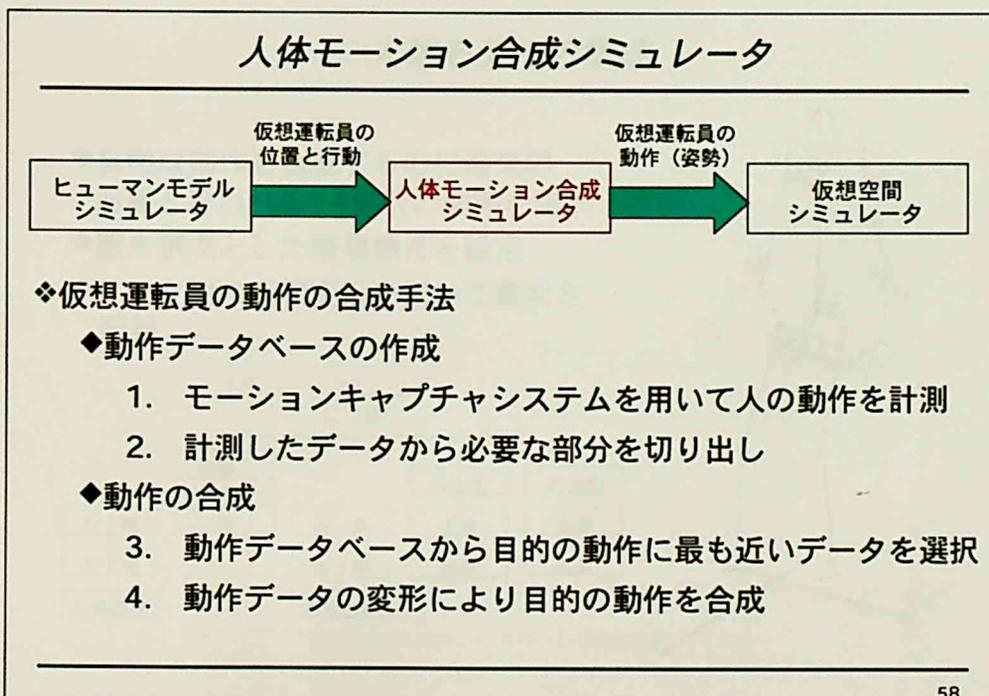
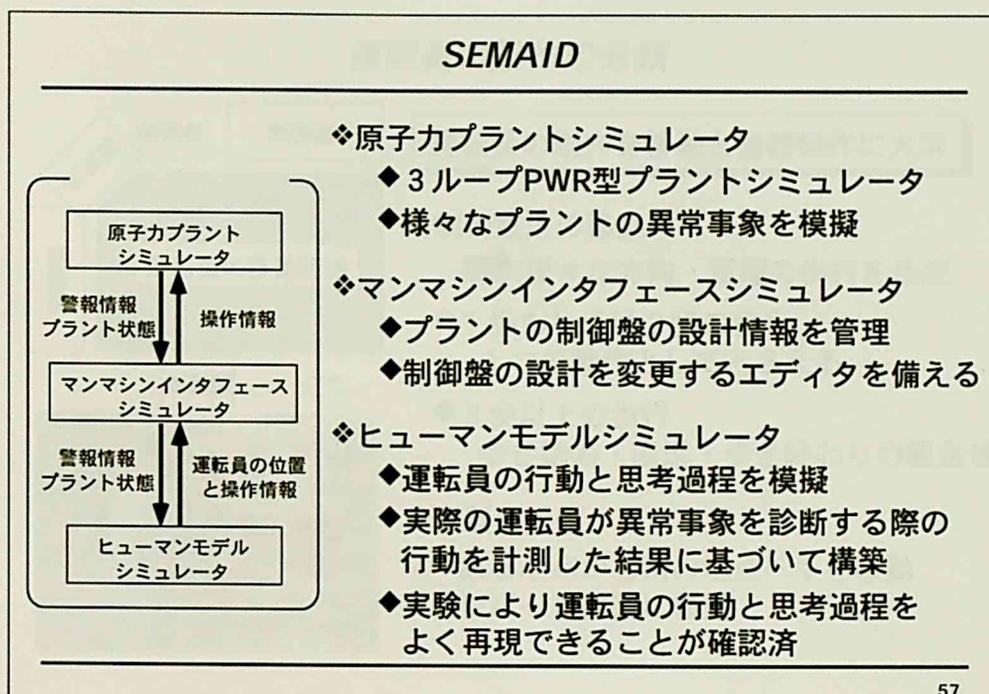


55

VENUSの概要 (ハードウェア構成)



56



運転員の動作の分類



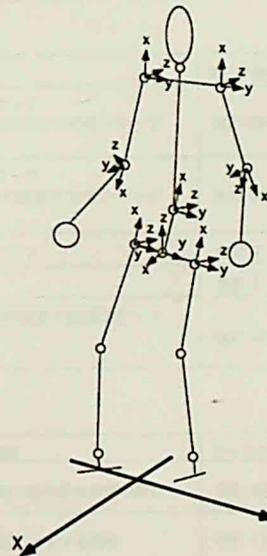
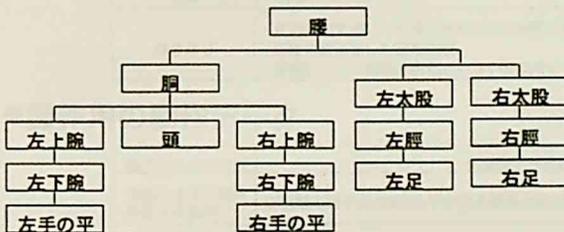
運転員の動作を移動と機器操作に大別

- ❖ 制御室内の移動
 - ➡ 様々な方向・距離の歩行を合成
 - ◆ 1歩もしくは2歩の歩行
1つの動作としてとらえる
 - ◆ 3歩以上の歩行
歩き始め・直進・歩き終わりの組合せ
- ❖ 機器の操作
 - ◆ ある地点から別の地点へ手を移動
 - ◆ レバーを前後に移動

59

人体モデルの構成

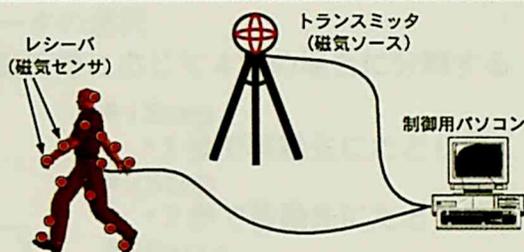
- ◆ 自然な動作を合成するのに最低限必要と思われる14箇所の関節を選択
- ◆ 腰を頂点とした階層構造を設定
- ◆ リンク間の座標変換を用いて姿勢を記述



60

人の動作の計測

- ◆磁気センサ取り付け位置
頭、胴、腰、上腕、下腕、手の平、太股、脛、足 (計15箇所)



- ◆計測データの雑音の除去
各関節の回転角度について5点の移動平均

- ◆磁気センサの取り付け位置の補正
人が直立姿勢にあるときの計測データを基準に補正

$${}^W R_{Hip} = {}^W R_{HipSensor} \cdot ({}^W R_{HipSensor}^{Calib})^{-1} \cdot {}^W R_{Hip}^{Calib}$$

${}^W R_{Hip}$: 補正後の腰の姿勢 ${}^W R_{HipSensor}^{Calib}$: 直立姿勢時のセンサの姿勢
 ${}^W R_{HipSensor}$: 磁気センサの姿勢 ${}^W R_{Hip}^{Calib}$: 直立姿勢時の腰の姿勢

61

計測データの切り出し

◆移動用の動作データ

| 動作データの分類 | 各動作データの種類 | データ数 |
|----------|--|-------|
| 1歩で到着 | ・移動方向について0度から180度まで30度ステップ ・目的地で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ | 合計 98 |
| 2歩で到着 | ・移動方向について0度から180度まで30度ステップ ・目的地で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ | 合計 98 |
| 歩き始め | ・移動方向について0度から180度まで30度ステップ | 合計 7 |
| 直進 | ・1歩歩くデータ | 合計 1 |
| 歩き終り | ・目的地で向いている方向について0度から180度まで30度ステップ ・最後に踏み出した足の左右 ・歩幅について普通の歩幅と小さい歩幅の2つ | 合計 28 |

◆操作用の動作データ

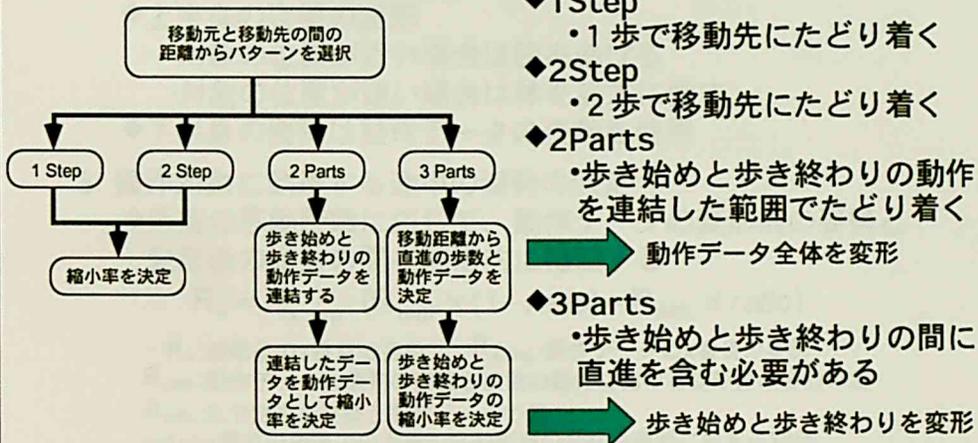
| 動作データの分類 | 各動作データの種類 | データ数 |
|----------------|----------------------------------|-------|
| 操作対象まで腕を移動する動作 | 制御盤の格子点の9点に手がある姿勢と直立姿勢の組み合わせ間の動作 | 合計 98 |
| 操作対象を実際に操作する動作 | 制御盤の格子点の9点上にあるレバーを前後にスライドする動作 | 合計 18 |

62

移動動作の合成(1)

❖方向と距離に応じた動作データの選択

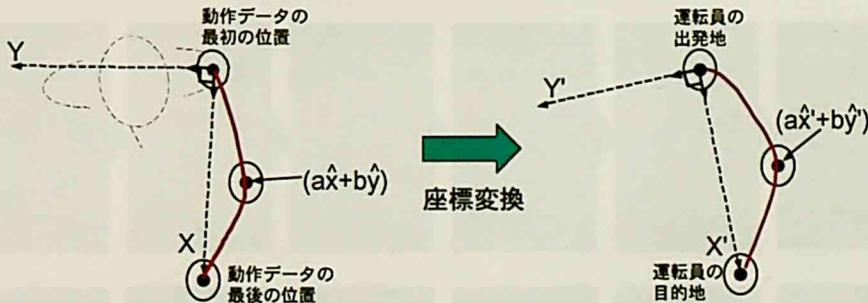
仮想運転員の歩行動作を移動距離に応じて4つの場合に分類する



63

移動動作の合成(2)

❖選択した動作データの移動距離と移動方向を変形する



X:動作データの最初の関節位置から最後の関節位置までのベクトル

Y:Xベクトルを90度回転したベクトル

X':動作開始地点から目的地点までのベクトル

Y':X'ベクトルを90度回転したベクトル

◆各部位の位置・姿勢をXY座標系からX'Y'座標系に座標変換する

64

機器操作動作の合成

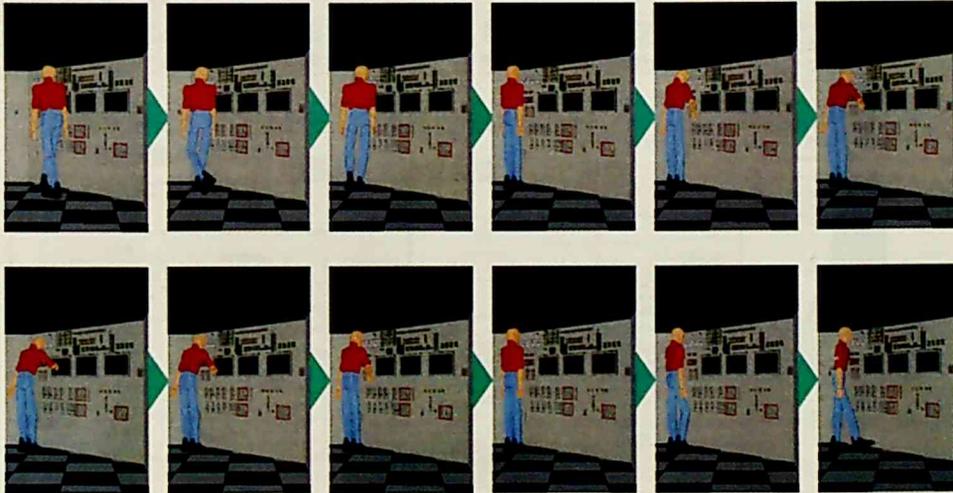
1. 操作対象の機器の位置に応じて動作データを選択
2. 操作対象に到達時の姿勢
 - ◆上半身の姿勢の変形
 - ・対象の位置が近い場合は腕を曲げる
 - ・対象の位置が遠い場合は体を前方に倒す
 - ◆下半身の姿勢は動作データの姿勢を使用
3. 操作対象に到達する途中の姿勢の決定
 - ◆手首の移動距離に応じて、動作データの変形前の姿勢と変形後の姿勢の間を線形的に補間する

$$R_x = R_{x_org} \cdot (R_{diff} \times (1 - ratio) + R_{diffL} \times ratio)$$

R_x : 変形後のx番目の姿勢 R_{x_org} : 動作データのx番目の姿勢
 R_{diff} : 動作データの最初の姿勢と直前の動作における変形後の姿勢の差
 R_{diffL} : 2.で定めた姿勢と動作データの差
 $ratio$: x番目の姿勢での手首の移動距離の全移動距離に対する割合

65

動作合成例



66

仮想空間シミュレータ

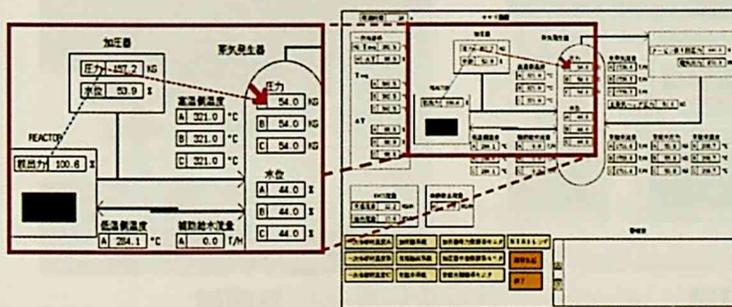
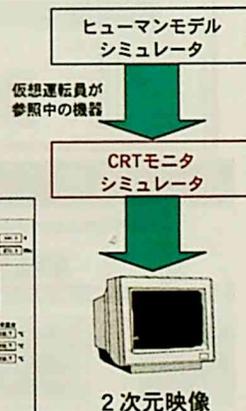


- ❖制御室と制御盤の描画
 - ◆10個の制御盤上に合計で約500個の機器を描画
 - ◆LOD(Level of Detail)の手法を適用
- ❖仮想運転員の描画
 - ◆形状モデル：人の腕の長さ等を計測しCADを用いて作成
 - ◆仮想運転員の各部位にテクスチャを張り付ける
- ❖仮想運転員の視点からの映像の描画
 - ◆人体モーション合成シミュレータの計算結果を利用
 - ◆CrystalEyesを用いた立体視

67

CRTモニタシミュレータ

- ❖CRTモニタ
 - ◆プラントの状態をまとめて表示（トレンドグラフ等）
 - ◆プラントの状態を把握する際に重要な役割を果たす
- ❖仮想空間に3次元映像として詳細に描いた場合
 - ◆計算負荷が高い
 - ◆画像が見にくい



68

発話合成シミュレータ

❖発話の合成方法

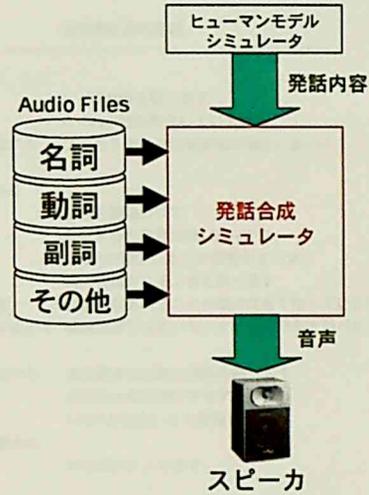
予め人の発話を録音して作成したオーディオファイルを再生

❖発話内容

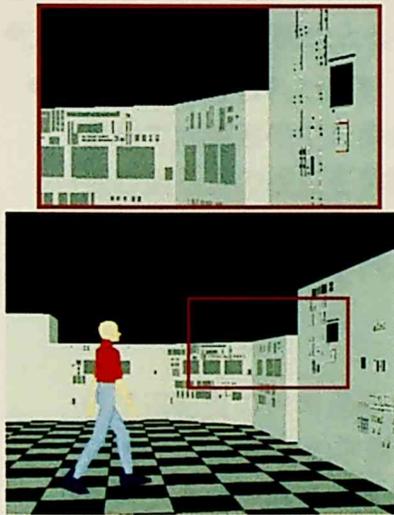
仮想運転員の行動を観察するだけでは理解できない内容

仮想運転員が発話する条件

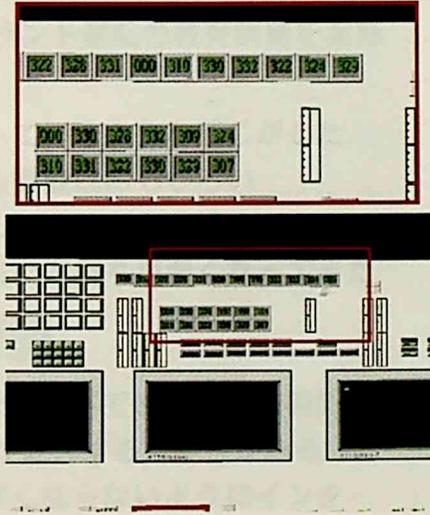
1. プラントのパラメータを調べる場合
2. 異常が発生したことを認知した場合
3. 異常仮説を想起した場合
4. パラメータの状態を予測した場合
5. パラメータに対し何か判断した場合
6. 異常仮説を破棄した場合
7. 異常仮説を採用した場合



VENUSの使用例 (シミュレーション中の画面)



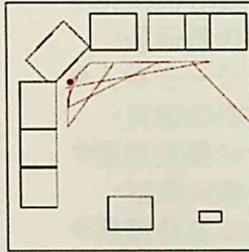
制御室



制御盤

VENUSの使用例 (仮想運転員の行動と発話)

❖LOCA(Loss Of Coolant Accident)の異常が発生した場合



運転員の歩行軌跡

| 時間 (分:秒) | 仮想運転員の行動 | 仮想運転員の発話 |
|-------------|----------------------|-----------------------|
| 00:00 | (シミュレーション開始) | |
| 00:00 | A-AG水位の確認操作を始める | A-AG水位を調べます |
| 00:20 | A-AG水位の値を確認する | A-SG水位は変化していません |
| 00:22 | プロセスモニタ注意警報の発生を確認する | プロセスモニタ注意警報が発生しました |
| 00:22 | LOCAが発生したとの仮説を立てる | |
| 00:22 | 格納容器ガスモニタの確認作業を始める | |
| 00:24 | | 診断を開始します |
| 00:25 | | 今考えられる異常はLOCAです |
| 00:27 | | LOCAが発生したとの仮説を立てます |
| 00:29 | | 格納容器ガスモニタを調べます |
| 00:41 | 格納容器ガスモニタの状態を確認する | 格納容器ガスモニタの値は大きくなっています |
| 00:52 | 格納容器じんあいモニタの確認作業を始める | 格納容器じんあいモニタの変化傾向を調べます |
| ... | | |
| 02:10 | 加圧器水位の変化傾向の確認作業を始める | 加圧器水位の変化傾向を調べます |
| 02:24 | 加圧器水位の変化傾向を確認する | 加圧器水位は減少中です |
| 02:31 | LOCAが発生したと断定する | LOCAが発生したと確信しました |
| 02:31 | 加圧器水位と圧力を維持する操作を始める | |
| 02:34 | | 対応操作に入ります |
| ... | | |

71

VENUSの使用例 (訓練生の感想)

❖大学院生3人を訓練生としてプラント運転の教育訓練を実施

◆訓練生から得られた感想

- 自分で制御室の中を歩き回っているような感じがした
- 仮想運転員の視点を体験する機能はおもしろい
- 仮想運転員が何を考え何をしているのか理解できた
- システムを使用しているうちに気持ちが悪くなってきた



- 仮想運転員の思考過程と行動を理解できることを確認
- VENUSは訓練生の学習意欲を促進する可能性がある
- システムの使用者の気分が悪くならないようにインタフェースを改良する必要がある

72

まとめ

仮想運転員がプラントの異常診断と対応操作を行う様子を観察・体験できる教育システムを開発

❖VENUSを複数のシミュレータに分けて実装

◆SEAMAID

・運転員の行動と思考過程を模擬

◆人体モーション合成シミュレータ

・実際の作業に近い自然な動作を合成

◆仮想空間シミュレータ

・描画対象との距離に応じて描く詳細度を変更(LOD)

◆発話合成シミュレータ

・仮想運転員の発話を模擬



◆仮想運転員の思考過程と行動の内容を理解できることを確認

73

VENUS実用化への課題

❖訓練生自らが仮想制御盤を操作し、結果を体験できる機能の構築

◆作業の様子を観察した後、獲得した知識を確認

❖仮想運転員が診断可能な異常事象の追加

◆現システムで診断可能な異常事象はLOCA及びSGTR

◆ヒューマンモデルシミュレータへのプラント診断知識の追加

❖VENUSのインタフェースの改良

◆気分が悪くならないようにインタフェースを改良

❖VENUSの使用による学習効果の評価

74

結論 (1)

原子力発電プラントの保守作業員と運転員を対象にした人工現実感技術を用いた訓練システムに関する研究

❖研究の背景と目的

- ◆設備、訓練時間の制限等により十分な実技訓練ができない
- ◆プラントの運転操作や保守作業の教育訓練へのVRの適用

❖人工現実感技術を用いた訓練環境

- ◆人工現実感の研究状況
 - ・視覚・聴覚以外の感覚提示デバイスの選択枝が少ない
 - ・仮想物体の運動の仕方を定める作業が非常に繁雑
- ◆人工現実感技術を用いた訓練システム
 - ・保守作業を対象：訓練環境を構築する際の労力が大きい
 - ・運転操作を対象：自ら制御盤を操作する訓練方式

75

結論 (2)

◆研究の着眼点と範囲

- ・少ない労力で訓練環境を構築できる新しい構築手法の提案
- ・容易に短時間で訓練環境を構築できる支援システムの開発
- ・運転員の行動を総合的に情報可視化するシステムの開発

❖人工現実感技術を用いた機器保守の訓練システムVRMATEの開発

- ◆仮想物体の状態遷移をペトリネットを用いてモデル化
- ◆ペトリネット作成支援システムPCSSを開発
- ◆「仕事」の概念を用いた自動実演のアルゴリズムを考案
- ➡ ◆仮想物体の状態遷移は少ない労力で設定可能
- ◆タスク関数を作成する作業が繁雑

76

結論 (3)

- ❖ 機器保守の訓練環境の構築支援システムDESCORTEの開発
 - ◆ 機器保守の訓練環境を構築する際に必要となる情報を分析
 - ◆ DESCORTEのインタフェース設計 (WIMPインタフェース)
 - ◆ ユーザの認知的負荷を軽減させる方針でインタフェースを改良
 - ➡ ◆ 非常に短時間で機器保守の訓練環境を構築可能
 - ◆ システムを初めて使用する際でも訓練環境を構築できる
- ❖ 人工現実感技術を用いた原子力プラント運転の体験型教育システムVENUSの開発
 - ◆ 運転員の行動と思考過程を模擬できるヒューマンモデルを使用
 - ◆ 人体モーションの合成技術を開発
 - ◆ 仮想運転員の思考過程を発話により訓練生に提示
 - ➡ ◆ 仮想運転員の思考過程と行動を理解できることを確認

77

研究成果の他分野への応用と今後の課題

- ❖ 研究成果の他分野への応用
 - ◆ 仮想物体の状態遷移をペトリネットを用いてモデル化
 - ➡ VR応用体験型CAIシステム (三菱総合研究所)
 - ◆ プログラミングを行わずに仮想空間を構築
 - ➡ 直感的な操作が可能なインタフェースの構築
 - ◆ 人体モーション合成技術
 - ➡ 映画やコマーシャル、擬人化インタフェースへの応用
- ❖ 今後の課題
 - ◆ ユーザへの生理的負荷を軽減した入出力デバイスの開発
 - ◆ コストダウンのためのソフトウェアとハードウェアの技術開発

78