

NAIST-IS-DD1561202

博士論文

製造業向けロボットを智能化する  
自律システムアルゴリズムの研究

野田 哲男

2016年9月16日

奈良先端科学技術大学院大学  
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に  
博士(工学) 授与の要件として提出した博士論文である。

野田 哲男

審査委員：

小笠原 司 教授 (主指導教員)

杉本 謙二 教授 (副指導教員)

高松 淳 准教授 (副指導教員)

丁 明 助教 (副指導教員)

# 製造業向けロボットを智能化する 自律システムアルゴリズムの研究\*

野田 哲男

## 内容梗概

製造業分野では少品種大量の製品を高品質にライン生産してきたが、2000年代に入って多様化する市場ニーズに応えるため、人の知能をより積極的に援用する人セル生産方式を導入した。しかし人セル生産方式の生産能力はライン方式に比べて低位で、作業者スキル由来の品質ばらつきは直近の課題であり、中長期的には生産人口減少問題と無縁ではない。そこで本研究では、ライン方式と人セル方式のそれぞれのメリットを集めたセル生産ロボット方式を実現するためのロボット智能化技術に取り組んだ。そのなかで知能としてとりわけ自律性が重要なことがわかった。よって本論文では製造業向けロボットに自律性を付与するシステムアルゴリズムを開発し、以下の結果を得たことを論述する。

まず、人の自律性を特徴づける試行錯誤による習熟を、未知の目的関数の最適化問題と定式化して解く独自の「能動型探索アルゴリズム」を提案する。これは、その問題設定と、その試行で得られるであろう情報量・情報が得られないリスク量を同時に考慮して次の試行を最適化する点とに独創性がある。実証実験の結果、熟練者の調整で動作時間 1045.33[ms] のロボット軌道が、約 56% の 590.22[ms] に短縮したこと、さらに熟練者による加工機制御パラメータ調整などの同種の困難性を持つ問題への水平展開に成功したことで、有用性が示される。

次に、組立工程で切望されるバラ積み状態の部品供給に取り組む。これはロボット学にとってランダム・ビン・ピッキングという未解決の古典的難題である。こ

---

\*奈良先端科学技術大学院大学 情報科学研究科 博士論文, NAIST-IS-DD1561202, 2016年9月16日.

の問題を複数の小問題に分割して解くシステム解法と，作業計画の試行錯誤を実行する「持ち替えグラフアルゴリズム」とを提案する．これらは既に有りそうで実は無かった方法である点，実行可能解を網羅的に算出する点に独創性がある．実証システムにおいて，数 [g] から数 10[g] の質量の実際の小型電機製品を構成する 11 種類以上の金属およびプラスチックの部品を，最速 2 秒台後半の時間間隔にて指定された姿勢に整列させられ，新規部品の追加に要するエンジニアリング時間は部品一つ当たり 0.5 日であることから，有用性がある．

#### キーワード

製造業, ロボット, 知能化, 自律システム, アルゴリズム

# Autonomous System Algorithms to Generate Intelligent Industrial Manufacturing Robots\*

NODA Akio

## Abstract

Reduction of the production population is a structural problem that cannot be avoided in the world. At the manufacturing sector, automated line production system for mass production with high quality have been carried. At the beginning of 2000s, the market needs had begun to diverse, then so-called human cell production system had been in limelight as a solution for high-mix production. However, the production capacity of the human cell production system is lower than those of the full line method, and the quality deviation that depends on the skill and experience of the worker is the most recent problem. Also it is no longer unrelated to avoid the influence of the production population decline in the medium to long term.

So in this study, the authors aim the development of intelligence of the robot for the practical application of the robotic cell production system as a collection of benefits of line systems and human cell systems. Especially the autonomy of humanity plays an important role in solving each problem. In this paper, the author describes his original system algorithms that grant autonomy to the manufacturing industrial robot, and their effects.

First, the author focuses on the proficiency as the prime feature of human autonomy serving as trial and error. He has proposed the original autonomous system algorithm "the active search algorithm".

---

\*Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1561202, September 16, 2016.

The originality is in its formulation of the challenge as an optimization problem of the unknown objective function, and its optimization method for next sampling.

Experimental results discovered that it gives the effect of robot trajectory optimization. A trajectory which operating time of the 1045.33[ms] adjusted by the skilled human operator was shortened to 590.22[ms] of the about 56 %. The algorithm was applied to the same kind of problems and its effect was also confirmed.

The second topic is bulk parts feedings that is absolutely a problem in automated assembly systems. This is known as a classic challenge of random bin picking problem for robotics. The author has proposed his original system design theory and the "re-grip graph algorithm" to run the trial and error of the operation to achieve the bulk parts feedings of multiple complex shaped components in exactly the same robot system.

The originality is in its system approach and exhaustive planning of feasible solutions.

In the demonstration systems, the actual eleven or more metal parts and plastic parts, which mass are about several grams up to a few dozen grams, constituting the small electric appliances could be aligned in the specified attitude with four robots. Speed was under three seconds interval at the fastest. While engineering time was 0.5 days for a new component.

**Keywords:**

Manufacturing Industries, robot, Intelligentize, Autonomous Systems, Algorithm

# 目次

|            |  |           |
|------------|--|-----------|
| <b>第1章</b> | <b>序論</b>                                | <b>1</b>  |
| 1.         | はじめに . . . . .                           | 1         |
| 2.         | 背景と動機 . . . . .                          | 1         |
| 3.         | 問題の本質の同定 . . . . .                       | 3         |
| 4.         | 研究開発のグランドデザインと本論文の主題 . . . . .           | 5         |
| 5.         | 本論文の構成について . . . . .                     | 6         |
| <b>第2章</b> | <b>これまでの知能化研究の分析と残された課題</b>              | <b>7</b>  |
| 1.         | はじめに . . . . .                           | 7         |
| 2.         | 知能化研究の展開と限界 1980年代後半 . . . . .           | 7         |
| 3.         | 知能化研究の展開と限界 1990年代後半から2000年代半ば . . . . . | 9         |
| 4.         | 知能化研究の展開と限界 2000年代後半 . . . . .           | 12        |
| 5.         | 知能化研究のさらなる課題について 2010年代以降 . . . . .      | 13        |
| 6.         | さらなる課題を解く研究開発のグランドデザイン . . . . .         | 14        |
| 6.1        | 課題を同定するトップダウン・ボトムアップ分析の概要 . . . . .      | 16        |
| 6.2        | トップダウン・ボトムアップ分析の初段階 . . . . .            | 17        |
| 6.3        | トップダウン・ボトムアップ分析の中間段階 . . . . .           | 20        |
| 6.4        | トップダウン・ボトムアップ分析の結果確認段階 . . . . .         | 21        |
| 6.5        | トップダウン・ボトムアップ分析の結論 . . . . .             | 24        |
| <b>第3章</b> | <b>技術プラットフォーム「自律型セル生産ロボットシステム」</b>       | <b>27</b> |
| 1.         | はじめに . . . . .                           | 27        |
| 2.         | 関連研究 . . . . .                           | 28        |
| 3.         | 開発した技術プラットフォーム . . . . .                 | 31        |

|            |  |           |
|------------|--|-----------|
| 3.1        | チョコ停対応行動計画 . . . . .                         | 31        |
| 3.2        | レイアウト最適化 . . . . .                           | 32        |
| 3.3        | エラー発生リスク分析 . . . . .                         | 34        |
| 3.4        | ハンドライブラリおよびハンド . . . . .                     | 35        |
| 3.5        | 最適作業動作生成 . . . . .                           | 36        |
| 3.6        | エラーリカバリ . . . . .                            | 38        |
| 3.7        | 教示作業 . . . . .                               | 40        |
| 3.8        | 3次元画像センシング . . . . .                         | 42        |
| 3.9        | システムインテグレーション . . . . .                      | 43        |
| 3.10       | 動作習熟 . . . . .                               | 44        |
| 4.         | 技術プラットフォームによるロボット化困難作業の克服 . . . . .          | 45        |
| 4.1        | 柔軟物ハンドリング . . . . .                          | 45        |
| 4.2        | バラ積み部品供給 . . . . .                           | 45        |
| <b>第4章</b> | <b>動作習熟機能を実現する自律システムアルゴリズム</b>               | <b>47</b> |
| 1.         | はじめに . . . . .                               | 47        |
| 2.         | 関連研究 . . . . .                               | 48        |
| 3.         | 開発したアルゴリズム . . . . .                         | 51        |
| 3.1        | 問題の定義 . . . . .                              | 51        |
| 3.2        | 逐次サンプリング点最適化を備えた能動型探索アルゴリズム<br>の詳細 . . . . . | 55        |
| 3.3        | 逐次サンプリング点選択最適化の評価値計算式について . . . . .          | 58        |
| 3.4        | 能動型探索アルゴリズムの実装について . . . . .                 | 67        |
| 3.5        | 能動型探索アルゴリズムの様々な適用事例について . . . . .            | 68        |
| 4.         | ロボットマニピュレータの動作習熟への適用 . . . . .               | 70        |
| 4.1        | ロボットの教示作業の現状 . . . . .                       | 70        |
| 4.2        | 動作軌道上の経路点決定の困難さ . . . . .                    | 71        |
| 4.3        | 能動型探索アルゴリズムの適用 . . . . .                     | 71        |
| 5.         | ある金属加工機の加工速度最大化への適用 . . . . .                | 79        |
| 5.1        | 課題と適用結果 . . . . .                            | 79        |



|            |                                      |            |
|------------|--------------------------------------|------------|
| 5.2        | 制御パラメータ開発プロセスの変革 . . . . .           | 85         |
| 6.         | サーボパラメータ調整における多目的最適化などへの適用 . . . . . | 88         |
| 6.1        | サーボパラメータの自動調整について . . . . .          | 89         |
| 6.2        | 同アルゴリズムの適用性の検討 . . . . .             | 90         |
| 6.3        | 誤差因子の処理方法の検討 . . . . .               | 103        |
| 6.4        | 計算時間の短縮化の検討 . . . . .                | 104        |
| 7.         | 初期探索に対する考察 . . . . .                 | 109        |
| 8.         | まとめ . . . . .                        | 112        |
| <b>第5章</b> | <b>ロボット化困難作業を克服する自律システムアルゴリズム</b>    | <b>117</b> |
| 1.         | はじめに . . . . .                       | 117        |
| 2.         | 関連研究 . . . . .                       | 118        |
| 3.         | 課題と提案方式 . . . . .                    | 120        |
| 4.         | 開発したアルゴリズム . . . . .                 | 122        |
| 4.1        | 部品の単離 . . . . .                      | 122        |
| 4.2        | 平面上の安定姿勢 . . . . .                   | 126        |
| 4.3        | 部品の把持計画 . . . . .                    | 127        |
| 4.4        | 部品の持ち替え：持ち替えグラフアルゴリズム . . . . .      | 130        |
| 4.5        | 部品の転がし操作 . . . . .                   | 132        |
| 5.         | 一般形状部品の多品種供給システムの設計方法 . . . . .      | 133        |
| 6.         | 構築した実機例とその動作 . . . . .               | 135        |
| 7.         | 他方式との比較実験結果 . . . . .                | 137        |
| 8.         | まとめ . . . . .                        | 139        |
| <b>第6章</b> | <b>開発成果のまとめ</b>                      | <b>141</b> |
| 1.         | 技術プラットフォーム . . . . .                 | 141        |
| 2.         | 実証システム . . . . .                     | 146        |
| 2.1        | 自動組立のためのバラ積み部品供給システム . . . . .       | 147        |
| 2.2        | 自動組立システム . . . . .                   | 149        |
| 3.         | 実証結果 . . . . .                       | 150        |

|                   |            |
|-------------------|------------|
| 4. 考察 . . . . .   | 150        |
| <b>第7章 結論</b>     | <b>151</b> |
| 1. おわりに . . . . . | 151        |
| 2. 展望 . . . . .   | 153        |
| <b>参考文献</b>       | <b>155</b> |
| 参考文献 . . . . .    | 155        |

# 目次

|      |   |    |
|------|---|----|
| 1.1  | 製造業におけるニーズとソリューション選択肢の変遷 . . . . .                    | 4  |
| 2.1  | 型彫り放電加工機のインテリジェント制御システム . . . . .                     | 8  |
| 2.2  | ファジィ制御型彫り放電加工機 V シリーズ . . . . .                       | 9  |
| 2.3  | 著者らが提唱した「インターネット電脳工場」の概念図 . . . . .                   | 10 |
| 2.4  | 著者らが提案した“組み込み EAP” の概念の実装図 . . . . .                  | 11 |
| 2.5  | ロボットを用いる生産システムのコスト内訳の一例 . . . . .                     | 13 |
| 2.6  | 知能ロボットの概念モデル . . . . .                                | 15 |
| 2.7  | ロボットアプリケーションのトップダウン・ボトムアップ分析 . . . . .                | 16 |
| 2.8  | 自律型セル生産ロボットシステムを実現する技術プラットフォームの実装モデル: 機能構成図 . . . . . | 23 |
| 2.9  | 適応行動カテゴリ . . . . .                                    | 25 |
| 2.10 | 自律行動カテゴリ . . . . .                                    | 25 |
| 3.1  | チョコ停対応行動計画のアプリケーション試作例 . . . . .                      | 31 |
| 3.2  | レイアウト最適化計算例 . . . . .                                 | 33 |
| 3.3  | エラー発生リスク分析の概念 . . . . .                               | 34 |
| 3.4  | ハンドライブラリおよびハンドの設計について . . . . .                       | 35 |
| 3.5  | 不確定性を解く最適作業動作生成アルゴリズム: 表記法の説明 . . . . .               | 36 |
| 3.6  | あるロボット作業における最適作業動作の解析例 . . . . .                      | 37 |
| 3.7  | 物体把持理論について . . . . .                                  | 38 |
| 3.8  | 物体把持時の握力 (アクチュエータ負荷) の時間変化の違い . . . . .               | 39 |
| 3.9  | 正常把持, 異常把持の様子 . . . . .                               | 39 |
| 3.10 | 特定作業のエラーリカバリ例 . . . . .                               | 40 |

|  |    |
|--|----|
| 3.11 教示を支援するヒューマンインタフェース例 . . . . .  | 41 |
| 3.12 3次元物体の画像認識例 . . . . .   | 42 |
| 3.13 自律型ロボットセル生産システムのインテグレーション例 . . . . .                                  | 43 |
| 3.14 動作習熟の概念と効果 . . . . .  | 44 |
| 4.1 未知の制約条件付きの未知の目的関数最適化の概念図 . . . . .                                     | 54 |
| 4.2 次のサンプリング点の最適化 . . . . .  | 55 |
| 4.3 能動型探索アルゴリズムの全体フロー . . . . .  | 57 |
| 4.4 探索結果 (左上:線形分離, 右上:単円, 左中:複円, 右中:離れ円, 下:L<br>字形) . . . . .              | 61 |
| 4.5 線形分離 (上:格子点の判定エラー数の推移, 下:マージン内 ( $ f  < 1$ )<br>の未探索格子点数の推移) . . . . . | 62 |
| 4.6 単円 (上:格子点の判定エラー数の推移, 下:マージン内 ( $ f  < 1$ )<br>の未探索格子点数の推移) . . . . .   | 63 |
| 4.7 複円 (上:格子点の判定エラー数の推移, 下:マージン内 ( $ f  < 1$ )<br>の未探索格子点数の推移) . . . . .   | 64 |
| 4.8 離れ円 (上:格子点の判定エラー数の推移, 下:マージン内 ( $ f  < 1$ )<br>の未探索格子点数の推移) . . . . .  | 65 |
| 4.9 L字型 (上:格子点の判定エラー数の推移, 下:マージン内 ( $ f  < 1$ )<br>の未探索格子点数の推移) . . . . .  | 66 |
| 4.10 動作習熟させる軌道の概要 . . . . .  | 72 |
| 4.11 ロボットが避ける障害物 . . . . .   | 72 |
| 4.12 実験システムの概要 . . . . .   | 73 |
| 4.13 動作習熟の進行 . . . . .   | 74 |
| 4.14 動作習熟前後のロボット軌道 . . . . .   | 75 |
| 4.15 探索中の経由点の変化 . . . . .  | 76 |
| 4.16 探索した空間の様子 (2次元表示) . . . . .   | 77 |
| 4.17 探索した空間の様子 (2次元表示) . . . . .   | 78 |
| 4.18 実験の設定 . . . . .   | 80 |
| 4.19 アルゴリズムの実装例のフローチャート . . . . .  | 82 |

|      |  |     |
|------|--|-----|
| 4.20 | 探索の進行例 (1)                                 | 83  |
| 4.21 | 探索の進行例 (2)                                 | 84  |
| 4.22 | 探索の進行例 (3)                                 | 86  |
| 4.23 | 実験の設定                                      | 91  |
| 4.24 | 実験に用いたサーボ制御装置の制御ブロックダイヤグラム                 | 91  |
| 4.25 | 評価指標                                       | 92  |
| 4.26 | データセット                                     | 97  |
| 4.27 | 目標仕様甘めの設定による実験結果                           | 98  |
| 4.28 | 目標仕様辛めの設定による実験結果                           | 99  |
| 4.29 | 目標仕様甘めの設定における探索結果                          | 101 |
| 4.30 | 目標仕様辛めの設定における探索結果                          | 102 |
| 4.31 | 能動探索アルゴリズムの実装例                             | 105 |
| 4.32 | CPU 数と改善率の関係                               | 108 |
| 4.33 | 7パラメータ中の3パラメータを用いて表示した識別境界の非線形性の様子         | 110 |
| 4.34 | 初期探索点の違いによる探索性能の変化                         | 111 |
| 5.1  | 提案する一般化物体のビンピッキングシステムの構成                   | 121 |
| 5.2  | 物体を把持するためのロボット動作と装置レイアウトの前提条件              | 123 |
| 5.3  | ロボットハンドと把持対象物体のモデルによる GrapabilityMap の算出方法 | 124 |
| 5.4  | 平面上の静定状態の計算のための最小凸包の計算例                    | 127 |
| 5.5  | 物体把持計画                                     | 129 |
| 5.6  | 持ち替え計画の計算に用いた部品とロボットハンドモデル例                | 131 |
| 5.7  | 持ち替え計画の計算結果による状態遷移モデル例                     | 131 |
| 5.8  | 物体の平面上の転がし操作計画                             | 132 |
| 5.9  | ビンピッキングのためのシステムの使い分け方法                     | 134 |
| 5.10 | ビンピッキングの実証システム                             | 136 |
| 5.11 | 実験に用いた 11 種類の様々な形状・材質の実部品                  | 136 |

|     |  |     |
|-----|--|-----|
| 6.1 | 開発した要素技術による自律型セル生産ロボットシステムのイン<br>テグレーション . . . . . | 143 |
| 6.2 | セル生産 (組立) の工程例 . . . . .                           | 146 |
| 6.3 | バラ積み部品供給セル . . . . .                               | 148 |
| 6.4 | 部品トレイの例 . . . . .                                  | 148 |
| 6.5 | ブレーカ組立セル . . . . .                                 | 149 |
| 6.6 | 組立対象製品の例 . . . . .                                 | 149 |

# 表 目 次

|     |                                       |     |
|-----|---------------------------------------|-----|
| 2.1 | 知能ロボットアプリケーションの同定 . . . . .           | 17  |
| 2.2 | トップダウン・ボトムアップ分析の初段階 . . . . .         | 19  |
| 2.3 | トップダウン・ボトムアップ分析の最終段階 . . . . .        | 21  |
| 2.4 | トップダウン・ボトムアップ分析の最終段階 . . . . .        | 22  |
| 4.1 | 2種類の要求仕様についての実験結果一覧 . . . . .         | 100 |
| 4.2 | CPU コア数を増やした際の計算時間短縮効果の計測結果 . . . . . | 107 |
| 5.1 | 従来手法とのシステム性能比較 (N は部品の種類の数) . . . . . | 137 |





# 第1章 序論

## 1. はじめに

本論文の主題は、第4章と第5章に示す自律システムアルゴリズムである。同アルゴリズムの研究開発に辿り着くまでには、各時代の要請の変遷とそれを解決するための研究開発の経緯があった。そこで、なぜ3章以降の開発が必要になったかを明確に定義づけるため、解説論文的な記述にはなるが、本章と第2章でこれらを示すことにする。

第3章ではそれらを踏まえて開始した、自律型セル生産ロボットシステム技術プラットフォームの開発プロジェクトの概要について述べる。これはオープンイノベーションを活用した数十人規模の一大プロジェクトであった。

第4章および第5章で、同プラットフォームに対する要求のなかで最も困難性の高い課題を解決するための自律システムアルゴリズムについて報告する。

## 2. 背景と動機

生産人口の減少は、まず日本が、その後、世界が必ず直面する構造問題である。これに対し、あらゆる分野に自動化を万遍なく普及させることが、安全安心で豊かな未来を維持発展させる方法の一つとなる。

製造業分野に目を向けると、動力を利用するようになった産業革命以来、生産を自動化して安価で高品質な製品を大量にライン生産することで、豊かな生活の実現に貢献してきた。ライン生産方式というソリューションは、設備投資コスト・運用コストを長い期間をかけて回収することが前提であり、つまり概してそれらのコストは大きくなるデメリットがあるものの、一度構築して安定して運用でき

るようになれば、高速で安定した生産の自動化を維持することにより、同じ品質の製品を大量に製造することができることがメリットとなる。ライン生産方式に投入されるロボットは、この目的のためにその機能を最適設計され、製造されている。そのようなロボットの行動計画は、ライン構築技術者がライン構築時に知恵を絞って設計し、生産活動開始後は、ロボットはあらかじめ定められた単純な位置決め動作を高速に繰り返すことのみで特化していた。これは、動物が条件反射で運動していることに相当する。1990年代にはいると、二次元ビジョンセンサが普及しはじめ、多少の位置ずれをロボット自ら修正する意味での知能化が実用化した。やはり単純な位置決め動作を高速に繰り返すことに主眼が置かれていて生産ラインの一部への導入にとどまっていた。

2000年代に入ると市場ニーズが多様化しはじめ、短期間のうちに生産機種を切り替える必要が生じた。このような状況の変化に低コストで柔軟に実施できるセル生産方式と呼ばれるソリューションが新たな選択肢として注目され、同方式をフル活用したEMS( Electronics Manufacturing Service) 事業者も躍進した。しかし、セル生産方式の生産能力はライン方式に比して低位である。さらには作業者の熟練度に依存する品質のばらつきが直近の課題として取り沙汰されるようになり、これらを解決するために労働者の選抜制をとるなどの方法が取られた。さらには景気の変動に応じて雇用を増減することなどもあわせて、これらの非人道的な対応が社会問題となった。また、このような人手をあてにした方式は中長期的には生産人口減少の影響と無縁ではない。

やがて2000年代後半になると、冒頭で述べた生産人口減少問題がようやく人口に膾炙するようになり、実際に統計的にも日本の人口が減少しはじめ、少子高齢化、生産人口の減少が現実のものとして議論されるようになった。同時に新興国の工業化と市場化の進展により国際競争が激化し、半導体製品・電気電子製品の設計製造技術の革新、景気変動・為替レートの変容・市場ニーズの激変に対応するための市場オフショア生産と地産地消と生産拠点国内回帰の選択の混乱、および市場のグローバル化といった複雑な事象が急速に同時進行したことで、次世代のものづくりの方式が求められ始めていたのが、この2000年代後半の展開であった。

著者らの研究グループでは複雑な状況に対応するためのソリューションの選択肢を増やすため、2006年に新規の研究プロジェクトをスタートさせ、ライン方式、人セル方式、それぞれのメリットを集めたセル生産ロボット方式の実用化を目指すことにした。ライン生産方式は上述のごとく品種の生産期間は長いことを前提とし少品種大量生産のために最適化され、生産される品質も安定している。ライン生産のなかで産業ロボットは、ワークや治具をピック・アンド・プレースする単純作業に投入されていた。一方、人セル生産方式は生產品種の切り替えが数か月未満で行われることを前提とし、生産速度よりも機種切り替えコストを重視した最適化の結果となっていた。生産速度はラインに比して低いがセルを並列化することで補うという考え方であった。品質は人に依存してばらつきが大きくなっていた。

これに対し、本研究で提案するセル生産ロボット方式では、品種切り替えは迅速、生産速度はそこそこだが品質は安定しているという良いところ取りを目指し、生産速度の増減は、セルの並列数を変化させることで対応しようという概念となる。ここまで述べてきたことを概念的に図 1.1 にまとめる。

### 3. 問題の本質の同定

もともとロボット単体で生産セルを構成するという基本的な考え方は、すでに1980年代の後半に提案されている [1]。当時、人工知能研究もエキスパートシステムと呼ばれる実用化が見られるようになり、バックプロパゲーション則の発見によるニューラルネットワークや、直接法 Fuzzy 推論則による Fuzzy 制御の応用、遺伝的アルゴリズムの応用などが進展しはじめた時代背景とあいまって、生産分野でも、工作機械など熟練者の手に委ねられてきた生産財の知能化研究と実用化が進められてきた。

しかしながら、それらの研究開発成果は、自動化には成功するものの、この手の知的制御システムを構築するにあたり、熟練作業モデルを構築するためのシステムの開発、および熟練作業モデリングの困難さは課題として残ることになった。さらには、1990年代後半から2000年にかけて生産システムの規模が年々大

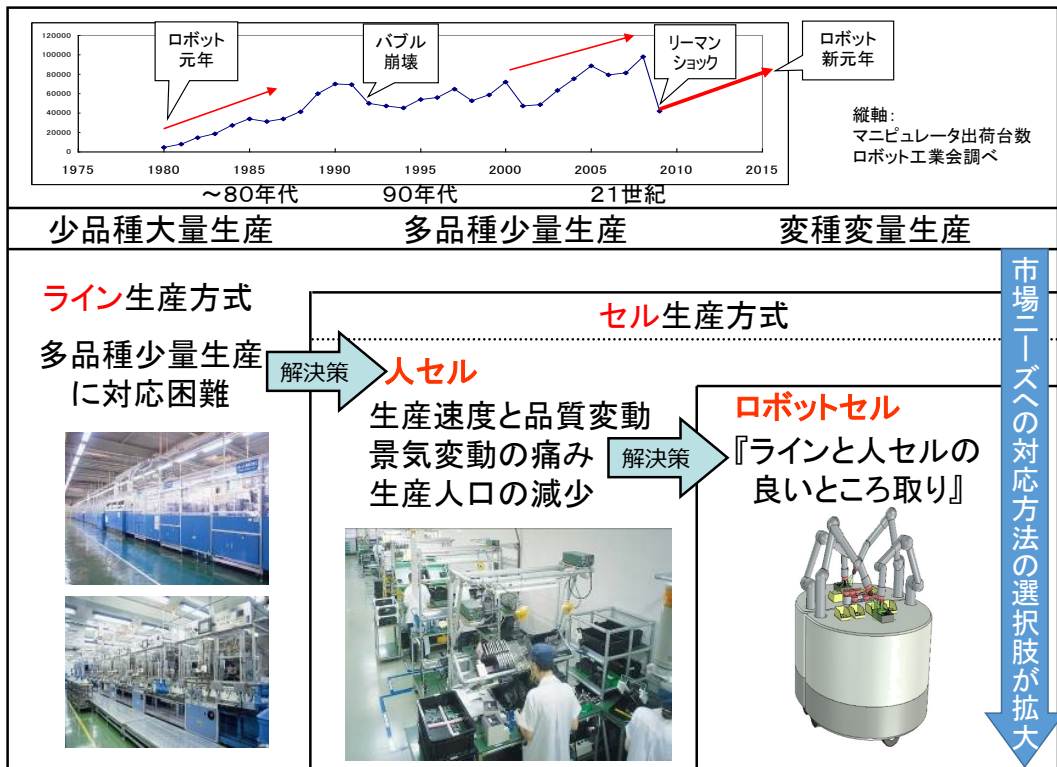


図 1.1 製造業におけるニーズとソリューション選択肢の変遷

大きく複雑化しはじめ、システム構築が属人的技能にとどまっている点も大きな問題となり始めていた [2]。2000 年代後半になると、産業現場の実務上の問題として、エンジニアリングコストの大きさがシステム構築のボトルネックとなることが、はっきりと認識されるようになった。エンジニアリングは、属人的な知識集約作業の典型例であり、リードタイムの長さもコストの一部として計上される。

そこで、著者らの研究グループではこのエンジニアリングコストの削減のため、問題の分析を実施し、本質的な解決策として必要な知能化機能を同定した。これを、第 2 章で詳報する。

通常、ロボットの知能化というと、外界センサーフッドバックにより従来できなかった複雑な動作軌道を実現することに主眼が置かれるが、著者らの研究グループの独創性は、現場のニーズに直接立脚した点にある。産業用ロボットは生産財

である。このため、知能を持った新しいロボットが安全・高速・安定な繰り返し作業を、従来方式となんら変わりなく実現することは大前提である。そのもとで、生産品目切り替えや誤った部品の混入などの大きな外乱に柔軟に対応すること、繰り返し作業への習熟といった自律性を発現する必要がある。つまり複雑な作業が可能となったとしても動作速度が落ちてしまったり、投資コストが大きくなってしまったりすると、現場では受け入れられない。従来型の生産システムが採用され、極端な場合は作業者を雇い入れるといった解決策が取られることになる。

また、エンジニアリングコストの削減に注目することは、これまでにない視点であり独創性が高い。

#### 4. 研究開発のグランドデザインと本論文の主題

このもとで、知能化技術の要素を開発し、技術プラットフォームとして統合することにした。このことを第3章で詳報する。そこでは、それぞれの課題の解決のための知能化技術の開発を進めていることを述べる。

研究開発を進めるなかで、人の知能のうち、とりわけ自律性が重要な役割を担っていることがわかったので、製造業向けロボットに自律性を付与する各システムアルゴリズムを開発することにし、これを本論文の主題とした。

まず、人の自律性の最たる特徴の試行錯誤による習熟を、未知の目的関数の最適化問題として定式化して独自の自律システムアルゴリズム「能動型探索アルゴリズム」を提案し、実際のシステムで効果を確認した。また、同種の困難性を持つ問題に対し効果を確認した。これは第4章で詳報する。

次に、自動化システムにおける製品組立作業ではバラ積み状態の部品供給が必ず問題となる。そこで多品種部品のバラ積み供給を実現するシステム設計理論と動作の試行錯誤を実行する「持ち替えグラフアルゴリズム」を提案し、実際の現場で効果を確認した。これは第5章で詳報する。

## 5. 本論文の構成について

次章以降を，上述した内容に沿って，以下の6つの章で構成する．

第2章では，自動化システム研究の現状を分析し，課題を明らかにする．

第3章では，課題を解決するための技術プラットフォーム「自律型セル生産ロボットシステム」の全容について述べる．

第4章では，課題として重要な自律性の発現について，動作習熟のための自律システムアルゴリズム「能動探索アルゴリズム」について述べる．

第5章では，おなじく自律性の発現について，バラ積み部品供給のための自律システムアルゴリズム「持ち替えグラフ」について述べる．

第6章では，実際の製品組立作業を実行するセル生産ロボットシステムを構築して開発技術を検証した結果を報告する．

第7章で，全体を通じての結論を述べる．

## 第2章 これまでの智能化研究の分析 と残された課題

### 1. はじめに

本章では、これまでの智能化研究のエポックを分析し、それが要請された時代背景、得られた成果、残された課題を整理する。

### 2. 智能化研究の展開と限界 1980年代後半

1980年代後半に人工知能の研究開発において、エキスパートシステムと呼ばれる実用化が見られるようになり、さらにはバックプロパゲーション則の発見によるニューラルネットワークや、直接法 Fuzzy 推論則による Fuzzy 制御の応用、遺伝的アルゴリズムの応用などが進展しはじめた時代背景とあいまって、生産分野にこれらを応用し、工作機械など熟練者の手に委ねられてきた生産財の制御操作の、智能化研究および実用化が発展しはじめた。

著者らも形彫り放電加工機と呼ばれる主として金型を生産するための金属加工の知的制御システムを開発した [3]。同システムの構成を図 2.1 に示す。同図にしたがって説明を加える。オフライン制御において加工開始前の制御パラメータの選定を実施し、加工開始後には、オンライン制御部において安定な加工状態を維持するように制御パラメータを調整し、同時に、加工速度が高速なものになるように制御パラメータの最適化を自動的に実施する構成である。

また、このシステムのうち、オンライン制御部分については製品化を行った [4,5]。製品の写真を図 2.2 に示す。これは従来、熟練作業者が加工開始前の制御パラメータの選定を、加工開始後には加工状態の維持と最適化を行っていたものを自動化

するもので、発表後即座にライバルメーカの追撃を受けるなど市場で好評を博した。その後、中小企業向け自動化機械開発賞、近畿地方発明表彰奨励賞を受賞し、今日に至るまで改良を加えつつ標準搭載機能として搭載され続ける製品となった。

このシステムでは、オフライン制御部では If-Then ルール群により熟練作業者の制御パラメータ選定方法を書き下している。背反するルールについては、優先順位により排他する推論ルーチンを独自に開発して用いた。If-Then ルール群の開発は熟練作業員へのインタビューを繰り返して改良を加えて完成させた。

一方、オフライン制御部では、加工中の主軸の動き、加工電流・電圧の変化の特徴量を入力として、直接法のファジィ推論により加工状態の良否、加工速度最適化の可能性を推定しながら、制御パラメータの変更操作を実施する構成である。熟練者の変更操作をモデル化するため、制御装置のパラメータ変更に関する Application Program Interface を設計して C 言語にてスク립トライクにプログラミングする環境を開発、熟練者へのインタビュー、実際の作業状態の観察を繰り返して改良を重ね、完成に至った。これにより、熟練者が加工機につきっきりで関し操作するのと同等の加工が自動で可能になった。

研究開発および製品化という成果としては成功裏に完了したものの、本論文の主題として繰り返し述べるとおり、この手の知的制御システムの構築には、熟練作業員モデルを構築するためのシステムの開発、および熟練作業員モデリングの困難さは課題として残ることになった。

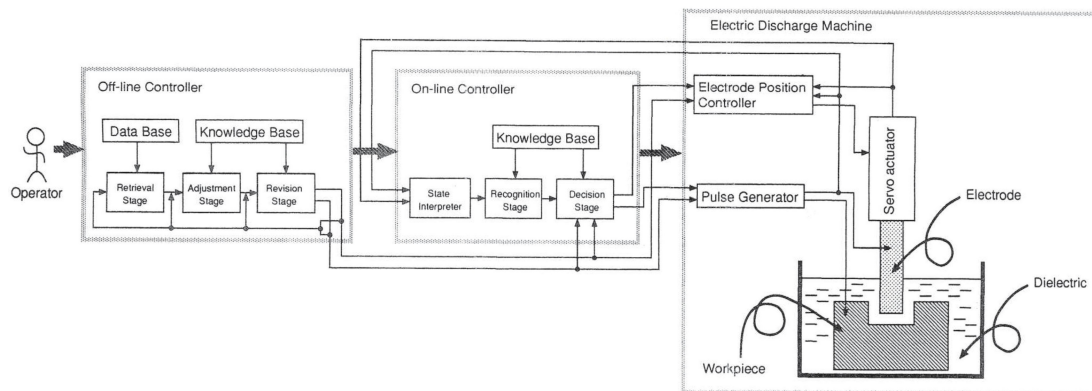


図 2.1 型彫り放電加工機のインテリジェント制御システム [3]



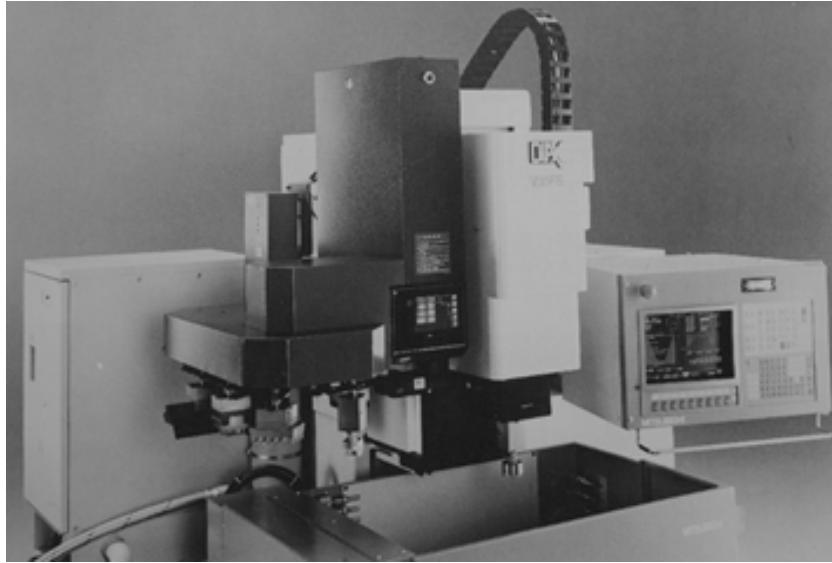


図 2.2 ファジィ制御型彫り放電加工機 V シリーズ (写真提供:三菱電機)

### 3. 知能化研究の展開と限界 1990 年代後半から 2000 年代半ば

1990 年代後半から 2000 年代半ばにかけては、イントラネット、インターネットが普及しはじめた時代であり、製造業への応用も検討が進められた。例えば著者らの研究グループでは、工作機械およびその数値制御装置を製造する業界内の複数の企業により、OSE 協議会 (Open System Environment) を立ち上げ、そのネットワーク WG がインターネットを援用した製造業のビジネスモデルを変革する提案を行った。これを、図 2.3 に示す。その後、この概念は自社 FA (Factory Automation) 事業の統合コンセプトに発展し、アライアンスや製品の展開が進められている。

同時期に日本ロボット工業会では、異なるメーカーの産業用ロボットを混在させた生産システム構築を目的とする ORiN (Open Robot interface for Network) と呼ばれるソフトウェアの共同開発プロジェクトを推進した [6-11]。その後、ORiN は工場内の各種装置に対して、メーカー、機種の違いを超え、統一的なアクセス手

段と表現方法を提供する通信インタフェースとして現在も発展を続けている [12].

このようなシステムを構築するにあたって問題となるのは、新たなネットワークシステムと従来の生産設備を混在させてシステム構築する新たな知識とノウハウであり、それに費やされるコストの大きさである。ここでいう新たなネットワークシステムとは、企業経営の管理する期間情報系，日々の生産計画立案系，そこから生産現場に対する生産指示を取り扱う生産実行系 (MES: Message Execution System)，生産現場の進捗を収集し生産実行系に返す SCADA 系 (Supervisory Control And Data Acquisition)，現場のセンサ，アクチュエータなどをつなぐフィールドバス系のことである。

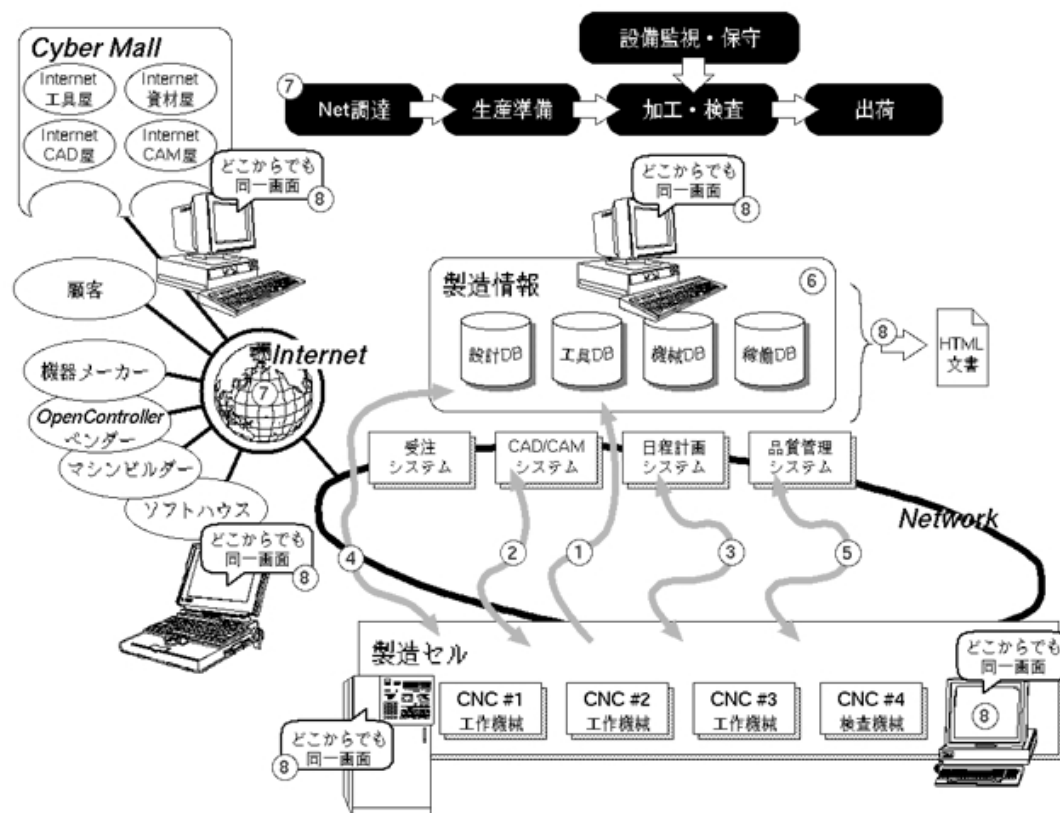


図 2.3 著者らが提唱した「インターネット電脳工場」の概念図

例示したとおり、多くのシステムが存在し、見るからに複雑である。つまり、その複雑さに呼応してリテラシー（そのシステムを構築し運用するため必要な知識の意味で使用）も、増大していく。その結果、このような企業内システムを構築するにあたり、系統ごとの専門技術者が系統の数だけ投入される事態となりコストがかさむ。トラブルシュートの局面では責任の切り分けも問題となる。このようなシステム統合は、EAI: Enterprise Application Integration と呼ばれている。

これを解決する一助として著者らの研究グループでは、リテラシーフリー化を実現する方法としての“組み込みEAI”なる概念を提唱し、その実装例としてMESインタフェースS/Wモジュールを開発し、制御装置製品や検査装置製品に組み込み搭載した。これにより、現場の制御装置の技術者が自己の知識で生産実行系とのシステム統合が可能となった。このことは生産装置と密結合する制御装置からの情報収集が可能となったことにより、生産の詳細な進捗とエネルギー消費の関係を管理するシステムに発展した [13]。

ただし、生産システムの末端は不確実な物理現象と対峙していて、それを御するための人手によるエンジニアリングが残された課題となった。

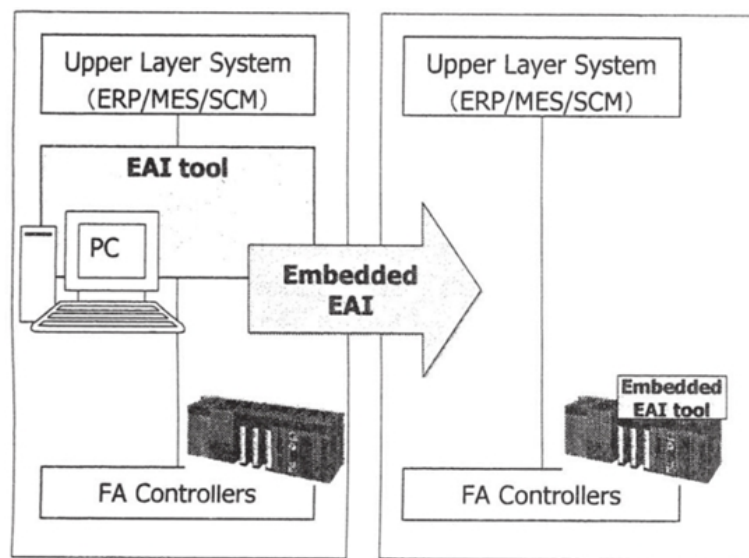


図 2.4 著者らが提案した“組み込みEAI”の概念の実装図

## 4. 知能化研究の展開と限界 2000年代後半

1章で述べたように、2000年代後半から、生産システムにおいては、頻繁な生産機種切り替えと需要の垂直変動への対応に悩んでいて、人手によるセル生産を複数並べて、柔軟さと生産能力を得る方策に出た。しかし品質、雇用、生産国リスクといった新たな問題にあえぐことになった。そこで著者らの研究グループでは、主として電気電子製品の組立作業を主ターゲットとするロボットによるセル生産の実現に取り組むことにした。

もともとロボット単体で生産セルを構成するという基本的な考え方は、すでに1980年代の後半に提案されていた [1]。しかし、同じセル内で、部品供給、組立、検査をするための研究は、実は2000年代後半の最近に行われるようになった [14]。ただし、これらの研究では、ロボットの主なタスクは、ワーク・治具・特殊なツールの運搬に限定されていて、価値を生み出す肝心の巧みな作業は、専用に設計製造された治具と特殊ツールが担っており、それらの設計は誰でもできるものではなく、システムエンジニアリングコストとなってシステム価格を押し上げている。著者らは、あらためて社内および顧客の生産プラントの調査と生産技術者へのインタビューを実施したところ、セル生産ロボットシステムを普及させるために根本的に解決すべき重要な課題は、やはりシステムエンジニアリングコストの削減であることがわかった。

図2.5は、典型的なロボットシステムの見積りの一例である。図の左半分の、設計、据え付け調整は、人が稼働するエンジニアリングコストである。右半分のロボット本体と、周辺（ロボット以外の周辺装置類の略）が、ハードウェアのコストである。つまり、エンジニアリングコストが生産システム構築コストの半分を占め、ロボット以外の治具、特殊ツール類が3分の1を占めている。これらのコスト構造が、投資効果の損益分岐点を押し上げている。

このコスト構造を変化させることが、ロボットシステムが投資効果を生む場面が広がる。このことは、これまで知られていなかった。ロボットの知能化研究はこれまで数多く取り組まれており、その一部が実用化段階に入っていたものの、そのような知られていない課題への取り組みは前例がなかった。

このようなセル生産ロボットシステムを実現するための知能化技術の研究開発

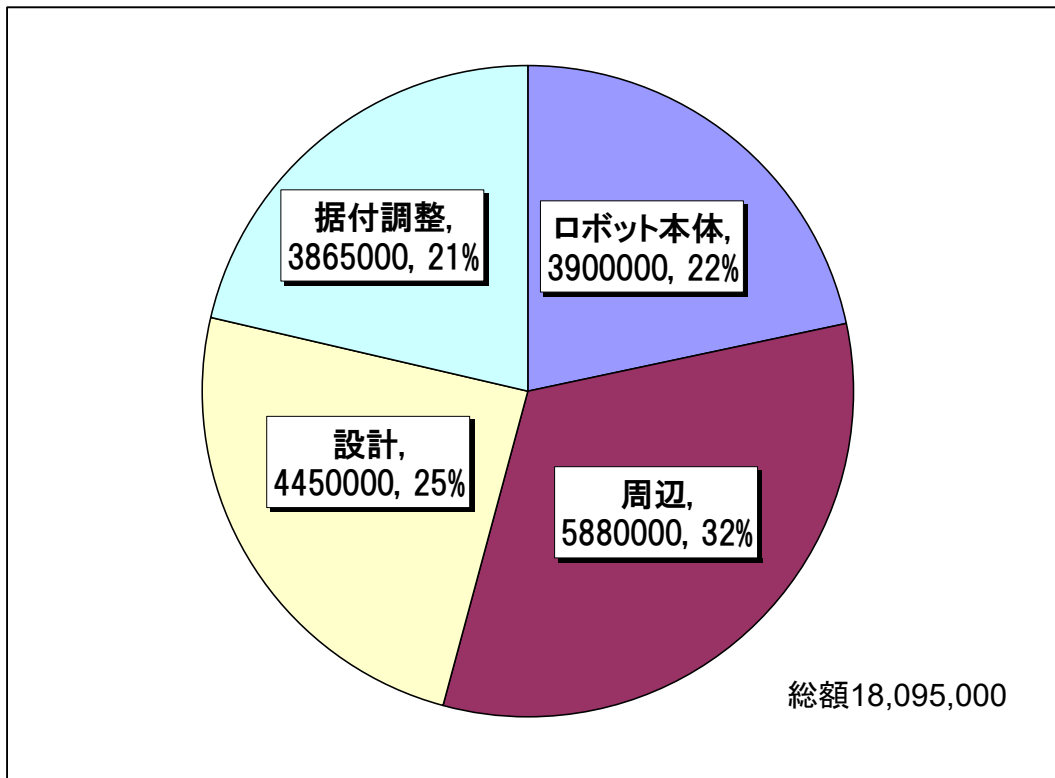


図 2.5 ロボットを用いる生産システムのコスト内訳の一例

に至るグランドデザインを本章後半に論述する。また、3章にてグランドデザインにもとづいて開発された技術プラットフォーム「セル生産ロボットシステム」について、概要を述べることにする。

## 5. 知能化研究のさらなる課題について 2010年代以降

従来のライン方式，人セル方式いずれにおいても，そのメリットの醸成には，人の知能が重要な役割を果たしている。知能のなかでも，与えられた状況に対して自ら考えて行動し，試行錯誤しながら学習する人の自律性がとりわけ重要である。グローバルな産業競争力を持ったセル生産ロボット方式の実用化のためには，もう一段進めてロボットを自律化する必要がある。自律化は機械学習理論上の未踏領域，およびロボット学上の未解決問題を含むチャレンジングなテーマである。

そこで、本論文では、人が介在することで発生しているエンジニアリングコストの削減をはかるための機能開発の基礎となる自律システムアルゴリズムの研究について、論述する。これは、第4章で詳報する。

やや視点を変えた課題として、生産システムにおいてはロボット化が困難な作業工程が依然として存在しているのも事実である。その最たるものは部品供給工程の自動化である。そこで本論文では、これまで人手に委ねられてきた部品供給工程の自動化を実現するための自律システムアルゴリズムについて、論述する。こちらは第5章で詳報する。

## 6. さらなる課題を解く研究開発のグランドデザイン

前節までで述べた通り、セル生産ロボットに求められる機能は、人の持つ柔軟さである。これまでこのような機能は知能化と呼ばれている。

本節では、セル生産ロボットシステムを実現するための、ロボット知能化技術研究のグランドデザインについて述べる。これは4節の話題と深く、関連している。

まず、ロボットの動きのサイクルを模して、知能ロボットを構成する機能要素の概念モデルを図2.6に示す。

図中、ロボットは図の情報に位置するセンサおよびセンサシステムにより、環境と作業対象ワークをセンシングし自らが置かれている状況を認識する。その結果を記憶に加味して移動あるいは作業するための行動計画が作られる。行動計画に含まれる目標値に向かって制御システムがエンドエフェクタを駆動し、実際に環境に変更を加える。以降このループを継続していく。

図2.6の六角形には、様々な分野の多くの理論・技術で構成されている。このようなシステムを実現するための技術群の開発を目指す。

なお、本論文では、これら一連の技術群を取捨選択してパッケージ化したものを、技術プラットフォーム「自律型セル生産ロボットシステム」と呼ぶことにする。

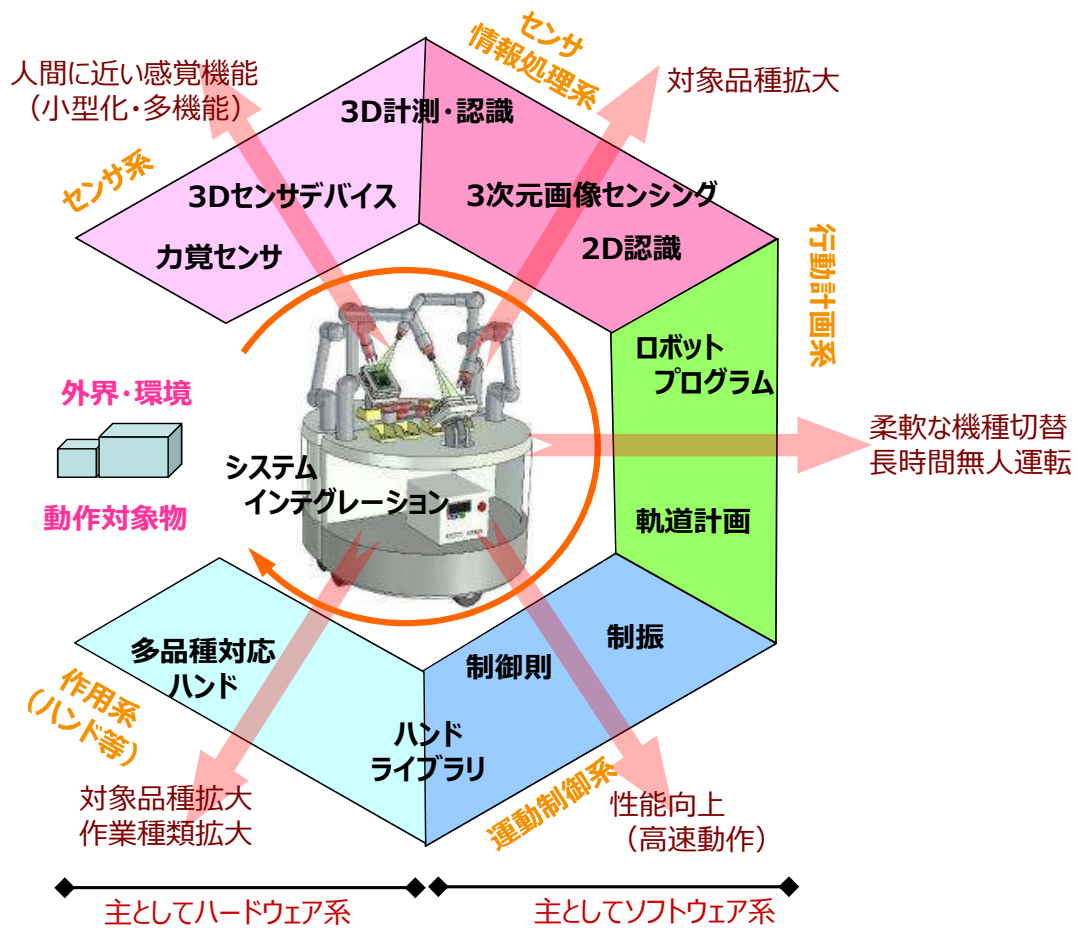


図 2.6 知能ロボットの概念モデル

## 6.1 課題を同定するトップダウン・ボトムアップ分析の概要

図 2.7 は、具体的な要素技術を抽出するための方針を示している。図 2.7 に示すように、まず対象とする知能ロボットアプリケーションを選定する。次にそのアプリケーションを実現するための機能を分割していき、可能な限り、小さな機能にブレークダウンする。

次に、ブレークダウンされた機能の分解粒度を整えながら、同一機能をまとめて機能モジュール化する。

最後に、機能モジュール群をインテグレーションしたときに、もともと想定した知能ロボットアプリケーションを構築できるかを確認し、その汎用性、仕様変更に対するロバスト性を確認することで、機能モジュール群が、技術プラットフォーム足りえるかを検証するものである。

以下、図 2.7 に沿って、作業を進めていく。

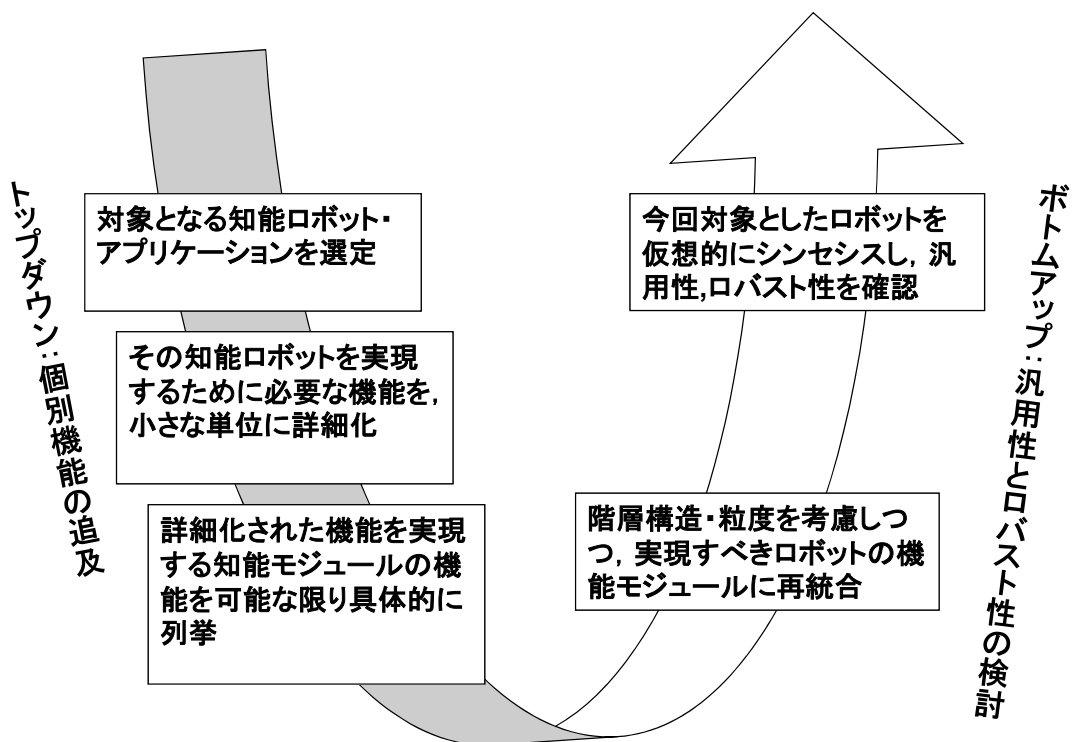


図 2.7 ロボットアプリケーションのトップダウン・ボトムアップ分析



## 6.2 トップダウン・ボトムアップ分析の初段階

まず、今回、対象とする知能ロボットアプリケーションを同定したものを、表 2.1 に示す。この表のアプリケーションは、日本ロボット学会編のロボット工学ハンドブックに記載された、製造業向けの産業用ロボットが適用されている工程の例を、引用した。

表 2.1 知能ロボットアプリケーションの同定

|   |  |   |  |  |   |                            |  |   |  |   |
|---|--|---|--|--|---|----------------------------|--|---|--|---|
| パレット・整列積載などで供給される部品の仕分・整列   |  |   |  |  |   |                            |  |   |  |   |
| ばら積み状態で入着した部品の仕分・整列   |  |   |  |  |   |                            |  |   |  |   |
| 部品同士を嵌合・締結して組み立てる   |  |   |  |  |   |                            |  |   |  |   |
| 組立途中における部品の移動   |  |   |  |  |   |                            |  |   |  |   |
| 専用加工機工程とワークの受け渡し  |  |   |  |  |   |                            |  |   |  |   |
| 研磨・バリ取り・切断などの機械加工   |  |   |  |  |   |                            |  |   |  |   |
| 液状のものを塗りつける加工   |  |   |  |  |   |                            |  |   |  |   |
| ボタン押しなどの簡単な検査   |  |   |  |  |   |                            |  |   |  |   |
| 後工程への引渡し  |  |   |  |  |   |                            |  |   |  |   |
| ロボット・アプリケーション   |  |   |  |  |   |                            |  |   |  |   |
| (マ<br>パ<br>テ<br>レ<br>ッ<br>ア<br>ル<br>・<br>ト<br>・<br>整<br>列<br>積<br>載<br>・<br>供<br>給<br>) | (マ<br>バ<br>テ<br>ラ<br>リ<br>ア<br>ル<br>・<br>積<br>み<br>・<br>コ<br>ン<br>ベ<br>ア<br>リ<br>ン<br>グ<br>) | (マ<br>マ<br>テ<br>リ<br>ア<br>ル<br>・<br>ハ<br>ン<br>ド<br>リ<br>ン<br>グ<br>) | (一<br>般<br>組<br>立<br>・<br>嵌<br>め<br>合<br>い<br>・<br>締<br>結<br>) | (そ<br>の<br>他<br>組<br>立<br>・<br>ピ<br>ツ<br>ク<br>&<br>プ<br>レ<br>ー<br>ス<br>) | ロ<br>ー<br>ド<br>・<br>ア<br>ン<br>ロ<br>ー<br>ド | 専<br>用<br>装<br>置<br>へ<br>の | 研<br>磨<br>・<br>バ<br>リ<br>取<br>り<br>・<br>切<br>断 | 溶<br>接<br>・<br>塗<br>装<br>・<br>ハ<br>ン<br>ダ<br>付<br>け | 塗<br>装<br>・<br>塗<br>布<br>・<br>簡<br>易<br>検<br>査 | (マ<br>マ<br>テ<br>リ<br>ア<br>ル<br>・<br>ハ<br>ン<br>ド<br>リ<br>ン<br>グ<br>) |

一方、各アプリケーションを実現するためのロボットシステムを構築するステップは、次の3つであると考えた。

1. 教示 ロボットシステムを構築する際の作業のうち、特に、ロボットを使用するための設計・据え付け調整作業を象徴して、「教示」と称する。作業シーケンスプログラミング、ハンド設計、動作習熟、軌道教示、精査計測処理の設定などの機能が、ここに分類される。
2. 作業動作 ロボットが生産活動を実施する際の、作業動作を高度化するためのセンサ情報処理や、制御則などの要素機能、それらを組み合わせたアプリケーション機能を、「作業動作」と称する。アーム移動、倣い移動、剛体物把持、柔軟物把持、ネジ締め動作、検査動作、センサ計測処理などの機能が、ここに分類される。
3. エラーリカバリ これまで述べてきたチョコ停<sup>1</sup>を抑制し、チョコ停発生時には自律的に復旧動作を行うための機能を、「エラーリカバリ」と称する。衝突・異物挟み込みによるアーム移動失敗、ハンド駆動失敗、把持失敗、載置失敗、組み付け失敗、検査失敗、部品供給ミスなどのエラー状態の検知とそこから復旧機能が、ここに分類される。

トップダウン・ボトムアップ分析の初段階として、表2.1に示したロボットアプリケーションと、上述の「教示」「作業動作」「エラーリカバリ」の分類の中に含まれる機能を掛けあわせて、表2.2を作成する。表中の○印が、縦横で関係を有する箇所である。横軸方向に読み取れば、左端に記された機能が、どのロボットアプリケーションで必要とされるかを表す。縦軸方向に読み取れば、上端に記されたロボットアプリケーションを実現するために必要な機能群を読み取ることができる。

---

<sup>1</sup>生産システム分野で使用される用語で、生産システムあるいは部品が壊れていない状態であるにもかかわらず生産が停止してしまう現象のこと。例えば搬送経路で部品が引っかかって止まっている状態を指し、作業員が引っかかっている部品の向きを整えると直ちにチョコ停は解消する。生産が一時的にせよ停止すること、およびその解消のために人員が動くことから、生産効率ダウンおよび生産コストをアップさせる要因のひとつであり、対策が望まれている。



### 6.3 トップダウン・ボトムアップ分析の中間段階

トップダウン・ボトムアップ分析の次の段階では、横軸の左端の各機能を作業小分類と呼ぶことに、各作業小分類の機能を実現するための課題とその解決策を分析する。

例えば、表 2.2 の、教示の項の作業小分類「作業シーケンスプログラミング」あるいは「動作習熟」の行については、何が問題になっているかという課題と、その解決策となる技術課題を追記していく。具体的には次のようである。

| 作業小分類          | 課題   | 解決策 (機能)   |
|----------------|--|--|
| 作業シーケンスプログラミング | (1) 作業シーケンスの組み立てが煩雑<br>(2) 作業シーケンスからロボット言語への変換，コーディング，教示点への展開，パス設定が煩雑        | (1) 作業シーケンス自動生成<br><br>(2) 概略作業パス自動生成  |
| 動作習熟           | (1) 直行率改善：ロボストに作業が成功する位置・姿勢を探ってそれを教示せねばならない。<br>(2) タクトタイム削減：ビデオ分析，勘に頼った試行錯誤 | (1) 動作内容記録，機械学（モーション系）による動作習熟，軌道姿勢アシスト<br><br>(2) 動作内容記録，機械学習（モーション系，シーケンス系）による動作習熟，軌道生成アシスト |

このような分析を、表 2.2 のすべての行について実施した。

トップダウン・ボトムアップ分析のさらに次の段階は、表 2.2 に追記した数多くの解決策を、機能としてまとめ上げる作業である。似た機能を寄せ集め、かつ実装時の粒度を整えながら、代表機能の名称をつけて統合していく。

この作業を実施した結果を、表 2.3 の左側に示し、右側には、技術プラットフォームを構成する機能として適切な粒度で実装可能なモジュールに整理しなおしたものを示す。

この作業の結果、7 群 35 種の機能モジュールが、当初想定したロボットアプリケーションを実現するために必要であることがわかった。

表 2.3 トップダウン・ボトムアップ分析の最終段階

| 解決策(機能)                   | 解決策(モジュール構成)          | 解決策(自己提供機能)      | 解決策(プロジェクト内提供予想機能)        |
|---------------------------|-----------------------|------------------|---------------------------|
| 作業シーケンス自動生成               | オフライン教示モジュール          | ハンドライブラリ         | 作業シーケンス自動生成               |
| 概略作業パス自動生成                |                       | エラーリカバリ行動制御      | 概略作業パス自動生成                |
| 軌道生成アシスト<br>(オフラインシミュレータ) |                       | エラー検知            | 軌道生成アシスト<br>(オフラインシミュレータ) |
| エラー発生分析                   |                       | エラー発生分析          |                           |
| エラーリカバリ行動制御               | オンライン教示モジュール          | 複合情報GUI          | マルチグリッパ型ハンド(H/W)          |
| エラー検知                     |                       | テレオペレーション        | 手先カメラ(HW)                 |
| 機械学習                      |                       | 自律制御             | ディスプレイ付ティーチングペンダント(HW)    |
| 動作内容記録                    |                       | 制御則構築支援ツール       | 力覚処理                      |
| マルチグリッパ型ハンド(H/W)          |                       | 機械学習             |                           |
| 手先カメラ(HW)                 |                       | 動作内容記録           |                           |
| ディスプレイ付ティーチングペンダント(HW)    | エラーリカバリモジュール          | エラーリカバリ行動制御      | マルチグリッパ型ハンド(H/W)          |
| 力覚処理                      |                       | 複合情報GUI          | 手先カメラ(HW)                 |
| センサ補正                     | 部品ピッキング用<br>物体認識モジュール | テレオペレーション        | ディスプレイ付ティーチングペンダント(HW)    |
| 自律制御                      |                       | 密な距離データからの干渉チェック | 3次元物体認識                   |
| 制御則構築支援ツール                |                       | 把持物体の状態認識        | 2次元物体認識<br>(部品位置、物体位置ずれ)  |
| 3次元物体認識                   | ハイブリッド視覚補正モジュール       | センサ補正            | 2次元画像認識                   |
| 2次元物体認識<br>(部品位置、物体位置ずれ)  |                       | 自律制御             | 3次元物体認識                   |
| 2次元画像認識                   | エラー検知用視覚モジュール         | 物体の状態認識          | 2次元画像認識                   |
| 3次元部品ピッキング用物体認識           |                       |                  |                           |
| 2次元・3次元ハイブリッド視覚補正誘導       |                       |                  |                           |
| 認識アプリケーション構築支援ツール         |                       |                  |                           |
| 把持物体の状態認識                 |                       |                  |                           |
| 干渉チェック                    |                       |                  |                           |
| 複合情報GUI                   |                       |                  |                           |
| ハンドライブラリ                  |                       |                  |                           |

## 6.4 トップダウン・ボトムアップ分析の結果確認段階

前節までで、7群 35 種の機能モジュールが、当初想定したロボットアプリケーションを実現するために必要であることがわかった。さらにモジュールの粒度と分類を再検討して 4 群 12 種の機能モジュールに整理しなおした。またシステムインテグレーションを実施するためのソフトウェアの開発が追加で必要になることがわかったため、開発課題にシステムインテグレーションソフトウェアを加えることにした。

トップダウン・ボトムアップ分析の最後の段階は、ここまでの作業で得られたモジュールが、もともと想定していたロボットアプリケーションに対応することができるかを確認することである。この作業を実施したものを表 2.4 に示す。○印

表 2.4 トップダウン・ボトムアップ分析の最終段階

| モジュール名 \ アプリ名  | マテリアル・ハンドリング<br>(パレット・ハンドリング<br>整列供給) | マテリアル・ハンドリング<br>(バラ積み・コンベア供給) | 一般組立て<br>(嵌め合い・締結) | その他組立て<br>(ピック&プレース) | 専用装置への<br>ロード・アンロード | 研磨・バリ取り・切断 | 塗装・塗布・<br>溶接・ハンタ付け | 簡易検査 | マテリアル・ハンドリング<br>(払い出し) |
|--|---------------------------------------|-------------------------------|--------------------|----------------------|---------------------|------------|--------------------|------|------------------------|
| オフライン教示<br>(チョコ停対応行動計画, エラー発生リスク分析,<br>最適作業動作生成, ハンドライブラリ) | ○                                     | △                             | ○                  | ○                    | ○                   | △          | ○                  | ○    | ○                      |
| オンライン教示<br>(教示作業, 動作習熟)                                    | ○                                     | ○                             | ○                  | ○                    | ○                   | ○          | ○                  | ○    | ○                      |
| エラーリカバリ<br>(行動制御, 教示作業)                                    | ○                                     | △                             | ○                  | ○                    | ○                   | △          | △                  | ○    | ○                      |
| 3次元画像センシング<br>(部品ピッキング用物体認識)                               | ○                                     | ○                             |                    | ○                    | ○                   |            |                    |      | ○                      |
| (ハイブリッド視覚補正)   | ○                                     | ○                             | ○                  | ○                    | ○                   | △          | ○                  |      |                        |
| (エラー検知用視覚)   | △                                     | △                             | ○                  | ○                    | ○                   | △          | △                  | ○    |                        |
| (認識アプリケーション構築支援)   | ○                                     | ○                             | ○                  | ○                    | ○                   | △          | △                  | ○    | ○                      |
| システムインテグレーション  | ○                                     | ○                             | ○                  | ○                    | ○                   | ○          | ○                  | ○    | ○                      |

は、関連が必要十分であることを表し、△印は、アプリケーションの設定によっては関連することを表す。

図からわかるとおり、4群のモジュールにより、想定したロボットアプリケーションをカバーできる。

さらに、ここまでで抽出した4群12種の機能モジュール間でデータ授受が成立するかの確認を実施した。この様子を、図2.8に示す。同図を概念モデルに対する実装モデルと呼ぶことにする。なお図中のピンク色の曲線は主として生産システム稼働開始前に動作する機能部、青色の曲線は稼働開始後の機能部を示している。このモジュール間データ授受の確認作業のなかで各機能モジュール間を結ぶ矢印のデータの具体的な内容、および、そのデータフォーマットの設計を実施し、実際にシステムがインテグレーションできる設計であることを確認した。

これ以降、実装モデルにしたがって技術プラットフォームを開発し、その技術

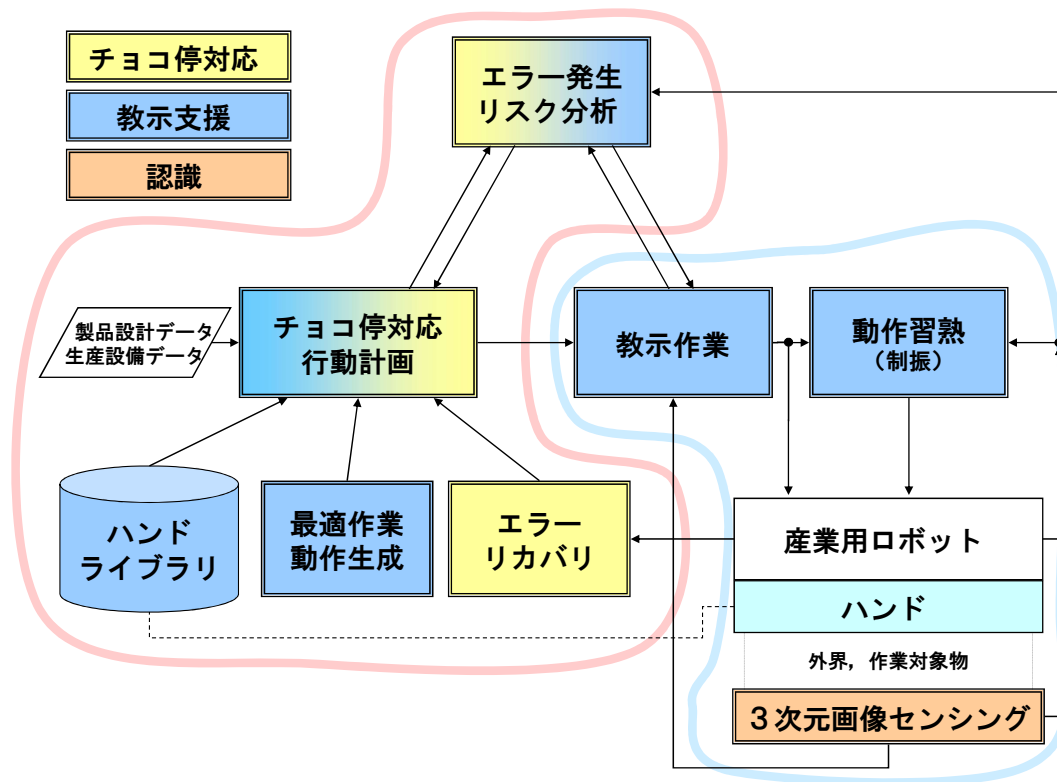


図 2.8 自律型セル生産ロボットシステムを実現する技術プラットフォームの実装モデル: 機能構成図

プラットフォームを用いて実際の製品ワークを用いて組立動作を実施する実証システムの構築を試みることにより、各要素技術および技術プラットフォームの効果を確認していく。

## 6.5 トップダウン・ボトムアップ分析の結論

前節までの結果から、トップダウン・ボトムアップ分析の結果、技術プラットフォームの開発に関して、チョコ停対応行動計画、レイアウト最適化、エラー発生リスク分析、ハンドライブラリおよびハンド、最適作業動作生成、エラーリカバリ、教示支援、動作習熟、3次元画像センシング、システムインテグレーション、の開発に取り組むことにした。システムインテグレーション機能については、動作速度と信頼性が保障されたロボット製品コントローラと、知能化機能を動作させるために高速演算できるパソコンを併用して生産システムを構成することで、信頼性と機能性を両立できる。ただしシステムインテグレーションには複数のシステムそれぞれのリテラシーが必要となるため、リテラシーフリー化 [2] を目指すことにする。このとき、これらの要素技術は図 2.6 に対して、2つの機能カテゴリの拡張として整理できる。

一つ目は適応行動カテゴリである。これは外界・環境の変化に対応する機能のカテゴリで、ここに分類される機能を図 2.9 に示す。

二つ目は、自律行動カテゴリであり、これを図 2.10 に示す。これは適応行動カテゴリだけでは解決しない問題に対応するために必要である。人間は困難な課題に突き当たったとき、最後の最後は、自律行動、すなわち、自発的な試行錯誤で、これを解決している。

そこで本論文ではロボットに自律行動を発現させるためのアルゴリズムを開発することにした。特に、次の2つの課題に取り組むことにした。

課題1は、動作習熟である。この問題の困難性は、未知の目的関数の最適化問題に集約される。本論文における解決策は、独自の能動型探索アルゴリズムである。これについて第4章で詳述する。

課題2は、バラ積み部品供給である。この問題の困難性は、ランダム・ビン・ピッキング問題に集約される。本研論文における解決策は、独自のシステム論的アプローチと持ち替えグラフアルゴリズムである。これについて第5章で詳述する。



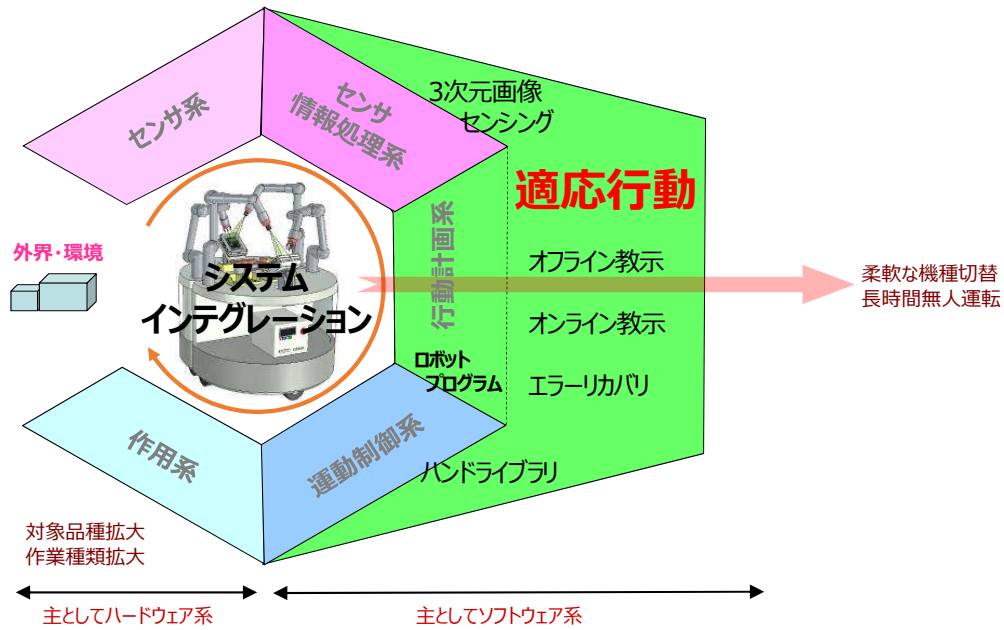


図 2.9 適応行動カテゴリ (緑色ハッチング部分)

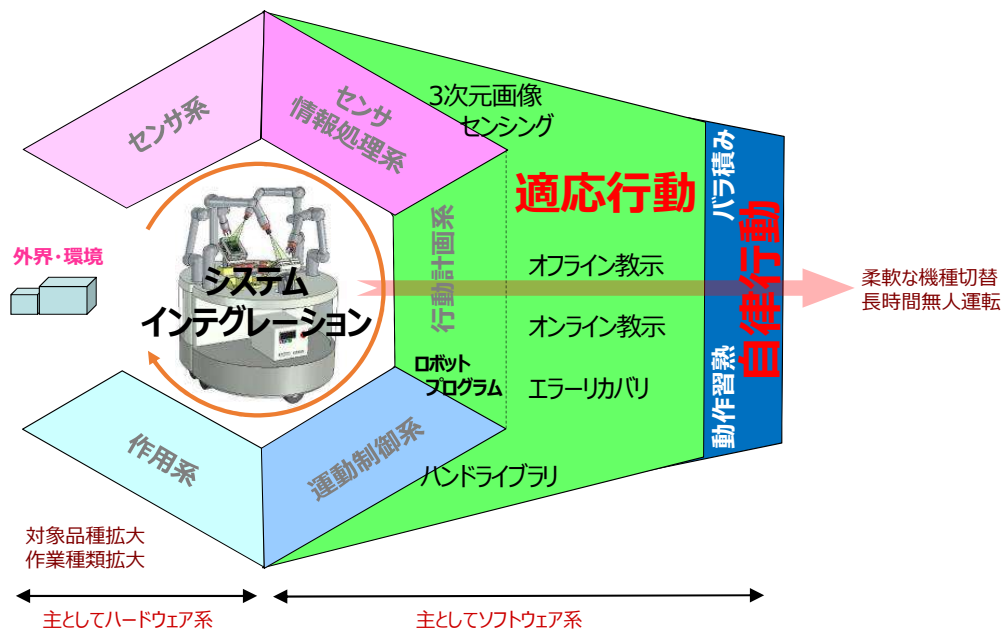


図 2.10 自律行動カテゴリ (濃青色ハッチング部分)



# 第3章 技術プラットフォーム「自律型セル生産ロボットシステム」

## 1. はじめに

本章では、前章までで述べた背景と経緯・動機・課題の分析結果に立脚した研究開発としての、主として組立作業を対象とする知能化技術プラットフォーム「自律型セル生産ロボットシステム」のグランドデザインについて述べる [15].

産業用ロボットは完成しているもので、もはや研究することがないと言われている。しかしながらモノづくりと製造業向けの産業用ロボットを含む生産システムを取り巻く状況が急速に変化しはじめ、これまで評価されることのなかった機能とコストが、ロボットに求められるようになってきたことは、これまで述べてきたとおりである。

この問題は工学システムと人の関係性に起因するシステム論の根源的課題の一種であり、著者はロボット知能化技術の開発はこれを解決するために用いるべきとの立場をとる。ロボット知能化は、不確実で予測困難な要素を含む複雑作業をロボット化するために用いられるとの印象がある。本論文では、この技術が担う役割は、むしろ機能とコストについての根源的課題の解決に対する有効性にあることを提言する。

前章でも述べたが、生産システムを構築する際の設計・据え付け調整に代表されるエンジニアリングには、各システム要素に対するリテラシーとノウハウ、つまり多くの人的コストが必要で直接的に全体コストに跳ね返る。また、自動化システムの受発注は、要求する機能が完成することを仕様とする「機能請負」が通常で、「機能」が実現されるまで手戻りが発生し検収が完了しない。手戻りで工期が不安定に変動することはリスクと考えられ、全て発注コストに織り込まれる。

このようなコストの問題が解決すれば、グローバル規模でエンドユーザの新たな投資が発生し、投資案件が増加すれば、エンドユーザにシステムを納める事業者(システムインテグレータ)が隆盛する。エンジニアリングコストが低減することはシステムインテグレータの取り分を単純に減らすのではなく、前述の利益変動リスクを低減させるので、むしろ事業維持上のプラス要因と捉えられる。次世代の産業用ロボットを、上記のような正のスパイラルを支える日本発の世界戦略商材にしたい。製造業向けロボットの研究開発の目的はここに極まる。

## 2. 関連研究

近年、ロボットセンサシステムの高性能化・低価格化とあいまって、製造業向けのロボットにおけるロボット知能化技術の研究開発と実用化が進展した。国内ではロボットメーカ各社の取り組み(例えば [16])、複数の大学の取り組み(例えば東北大学 [17])がある。経済産業省(以下、METI)および新エネルギー・産業技術総合開発機構(以下、NEDO)の研究開発事業でも、産業用ロボットの知能化技術への取り組みがある。

海外の国策としては、欧州の様々な産学官連携に加え、米大統領がカーネギーメロン大学での演説 [18] に至った事情には考えさせられる。その後の製造業国内回帰施策とその効果は一進一退ではあるが、継続的に効果を測定しながら対策が打たれ続けていることは注視すべきである。アジアでも中国の新松机器人自動化股分有限公司や広州数控設備有限公司に代表されるロボットメーカが製品やビジネス展開について進化を加速している。同国国策については2012年3月27日付けの中国科学技術部の智能製造科学技術発展第12次5カ年特別規則的通知 [19] に、現状は輸入に頼っている製造技術について、“智”能化技術開発に注力し、モノづくり力を高めることが記されている。その後、国務院通達として2015年5月に交付された「中国製造2025(Made in China 2025)」には、35年先を見据えた最初の10年間の製造業の高度化に関する総合的計画が示されていて、2025年までに製造強国の仲間入りし、その後は2035年までに製造強国の中レベルへ、2049年ま

でに製造強国の先頭グループへ至る大目標が描かれる。これを調査したまとめ資料としては例えば内閣府の検討会資料 [20] や [21] がコンパクトでわかりやすい。本通達は中国国内でも相当なインパクトがあったようで、曲解した反応を含め活動状況が伝えられつつある (例えば [22])。一方、ドイツにおいては Industrie4.0 が国策として進められていて [23]、日本ではこれを相当に警戒する反応が出ている。しかしながらこれは、いわゆるオープン・クローズというオーソドックスな戦略をとっているにすぎない。オープンな部分は製造業の B2B 取引の電子化通信プロトコルを標準化して協調領域とし、そのプロトコル上に乗るデータは各企業ごとにクローズして競争領域とするものと推察される。つまり競争領域こそが本丸であり、産業用ロボットなど従来からの製造業向けの生産財の開発競争は継続していくことになる。

民間では、KUKA 社 (LWR 機から iiwa 機に至る開発が典型例 [24])、ABB 社 (同じく FRIIDA 機から YuMi 機に至る流れ [25])、欧州の産学連携プロジェクト (ROSETTA, SMErobot, ミュンヘン工科大学を中心とする CoTeSys, KUKA とドイツ航空宇宙センター DLR の関係など)、米国のベンチャー企業 Rethink Robotics(元 MIT の Brooks 氏が創業) や Universal Robotics(知能化ソフトに注力) をはじめ、特に 2016 年に至る数年で工場内で作業者と近接して作業するロボットの製品化例が数多く見られるようになり、新しい生産システムの姿とされる一方で、安全のために動作速度を落としていることで生産効率の観点から疑問視する声も多い。いずれにせよ、新しい生産システムの形態の模索が世界中で行われている。また、ソフトウェアの動きとして Neocortex(知能化ソフトに注力 [26]) や、Willow Garage 社が創始し有志による開発・活用が続いているソフトウェアプラットフォーム ROS(Robot Operating System) においても、Southwest Research Institute による ROS-Industrial という産業用ロボット向けのコレクションパッケージが追加された [27]。ROS-Industrial にはオムロン傘下に入った Adept, 安川電機米国法人らが参加している。Microsoft の Robotics Developer Studio と、低価格 3 次元センサ Kinect は、パソコンの高い計算能力と高効率な開発環境を、ロボットシステムに安価に援用できる [28]。

日本国内でも、経済産業省 [29] や日本ロボット工業会における新しい動き [30]

があり、首相官邸を発信源とするロボット新戦略 [31] を受けた動きが活発化していることはよく知られている。

著者は、2006年から「経産省・NEDO次世代ロボット知能化技術開発プロジェクト(平成20～23年度)」「NEDO戦略的先端ロボット要素技術開発プロジェクト(平成19～22年度)」に参画、オープンイノベーションによる製造業向け産業用ロボットの知能化技術の開発に取り組んでいる [32,33]。そこでは、ロボットによるセル生産を実現するための知能化技術を開発するとともに、前述のエンジニアリングコストの削減のために同技術を活用することを提案、実システム上で、効果を確認した。

なかでも、エンジニアリングコストの削減に関係が深いものは行動計画系の要素技術である。これまで設計技術者、据え付け調整技術者が経験と勘で実現していた熟練の技を、知能化技術で代替あるいは技術者・作業者の感覚を拡張して作業支援することで、自動化システムの設計時・据付調整時・運用時・移設再調整時に発生するエンジニアリングの自動化・効率化・ノウハウレス化を狙っている。繰り返し述べるが、ロボットの生産活動時だけでなく、エンジニアリング作業時にこそ、知能化技術は有効である。

特記事項としては、従来の製造業向けの産業ロボットシステムの安定性と高速性をそのまま活かしたいので、従来型生産システムに対し、複数の知能化機能をアドオンするシステム形態をとった点である。

この技術プラットフォームは、外界センサ、アクチュエータ、エンドエフェクタといったハードウェア要素、センサ情報処理、行動計画、制御といったソフトウェア要素から構成されている。ソフトウェア要素には、ロボットの据え付け調整時の教示作業を支援する機能、ロボット稼働中のセンサフィードバック制御機能などを含めた。この技術プラットフォームを用いて、実際に組立作業を自動化するシステムを構築したところ、部品点数30数点の電機製品を人間の2倍程度の時間で組み立てることができた。また、システム構築コストの削減に関して、教示予定作業時間8時間を1時間に短縮する効果、特定のエラー発生時に自律的に復旧する効果を得た。以下、前掲の図2.8にしたがって、開発した技術プラットフォームの各機能モジュールについて、説明する。

### 3. 開発した技術プラットフォーム

#### 3.1 チョコ停対応行動計画

生産システムの作業動作プランナー [34,35], 複数台ロボットのインタラクティブな作業工程のためのブロックプログラミングツール [36] を開発した.

生産システムのシミュレータでは, 一般的なロボット動作, マテリアルハンドリング, 剛体部品の干渉チェックに加え, 弾性変形および塑性変形を取り扱うことができる. 作業プランナーは, RRT 法 (Rapidly-exploring Random Trees) により, 動作軌道生成, 剛体部品, 弾性変形, 塑性変形を伴う部品の組み付け軌道生成を行なえることを確認した. プログラミングツールでは, ロボットの複数動作をプログラミングした上で一塊の作業工程として登録し, それら作業工程を登録パレットから選んでプログラミングした後, それらの順序を入れ替えたり, 複数ロボット間で同期をとったりするプログラミングが可能で, 動作シミュレーションの後, ロボット言語を吐き出すことが可能である.

一方, 生産システムの構築現場では依然として PLC (Programmable Logic Controller) を中心に据えて, 制御系を構築することが主流である. PLC ではラダー言語によるシーケンス制御プログラミングが行われている. ラダー言語プログラムのデバッグ法は科学的な定説がない. そこで著者らの研究グループでは独自の解析手法を開発し, 産業現場で実稼働するラダーに適用したところ, 効率的かつ見落としなく解析できる効果を確認した [37].

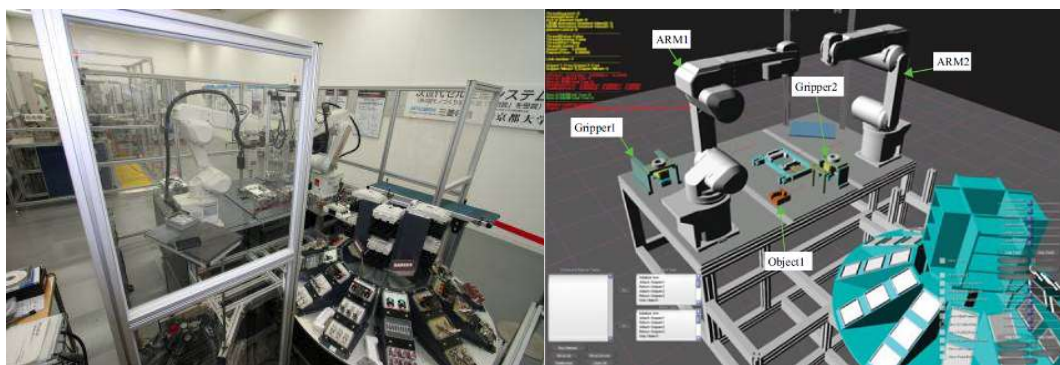


図 3.1 チョコ停対応行動計画のアプリケーション試作例

### 3.2 レイアウト最適化

レイアウト最適化は「チョコ停対応行動計画」に含まれる機能である。

ロボットを用いた生産システムのレイアウト配置は難しい。レイアウトを決めて、いざロボットの教示作業(後述)を始めると、ロボットの動作範囲の都合上、ロボットが作業姿勢を取れなくなり、最悪の場合、レイアウトの手直しが行なわれる。もちろん教示はやり直しになる。また、タクトタイムを期待して、ロボットの配置を変更が検討されることは多いが、全体として何が起きるかはやって見ないとわからない。レイアウト配置に関して、作業や管理者の勘と度胸に頼らない合理的な判断を。システム構想時にフロントローディングして、上記のような手戻りと無駄を少なくしたい。そこで、著者は、この課題を(i)タクトタイムの最小化、(ii)教示しやすさの最大化、(iii)占有面積の最小化、の3つの指標を目的関数とする多目的最適化問題に帰着させることで、上述の課題の解決に取り組んでおり、実際のロボットシステム構築案件で、有効性を確認した[38-43]。例えば、これ以上ロボットの据付間隔を縮めてもタクトタイムはたいして縮まないが、教示の困難さが急激に増大する、ということが定量的に判断できる。図3.2は、[40]に記載された2台のロボットで協調作業を行なうセル生産ロボットシステムのレイアウトをいう課題に対して、床面積・サイクルタイム・可操作性という3つの指標についての多目的なレイアウト最適計算結果である。上方の3次元グラフは3つの指標に対するパレート解の配置を表しており、それらパレート解の典型例としてのSolution A~Dの実際を、下方に平面図で示している。平面図における1の矩形、2の矩形がロボットで、その他の番号の矩形は生産システムに必要な機器類、グレーハッチングの四角はダミースペースである。



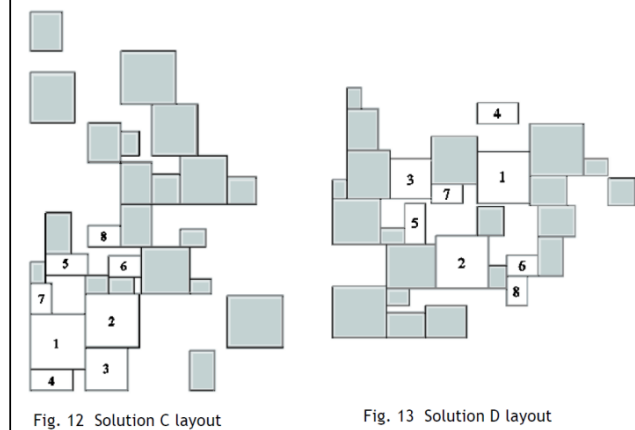
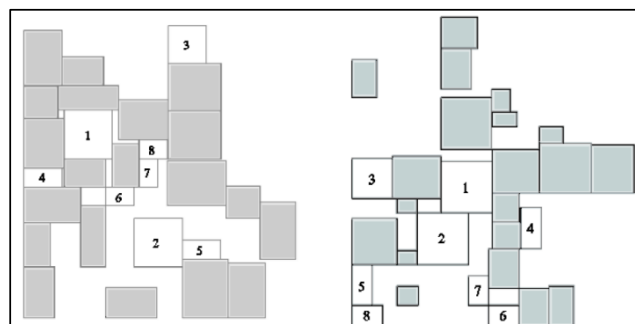
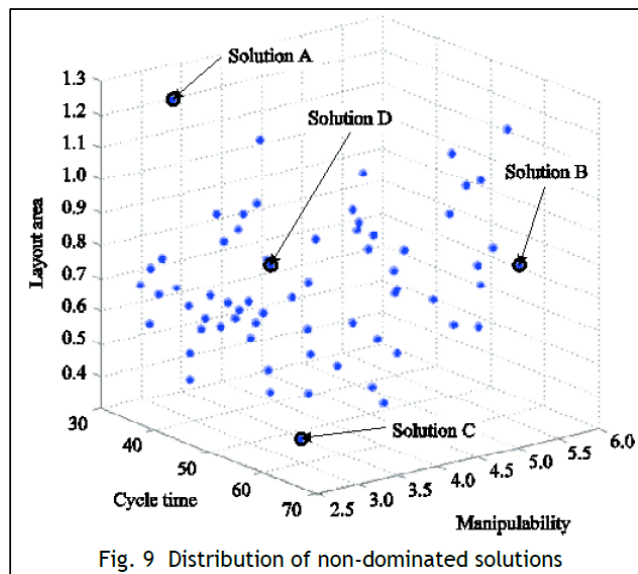


図 3.2 レイアウト最適化計算例 [40]

### 3.3 エラー発生リスク分析

ロボットが作業中に、いわゆるチョコ停が発生し、生産が停止することがある。エラー発生リスクを事前に予測し、エラーが発生しにくい作業手順をプログラミングしたい。しかもプログラミング作業者の経験と勘に頼るのではなく、データに基づいた合理的判断をしたい。そこで本研究では、ベイジアンネットワークを時間的に拡張したダイナミックベイジアンネットワーク (DBN) を用いて、ロボットのエラー状態 (あるいは非エラー状態) を動的に推定・解析するアルゴリズムを開発した [44]。これは近年提案された確率ロボティクス [45] と関連がある。

例えば、図 3.3 の右上に示した、3枚のピースからなるパズルの組みF作業を考えてみる。その組立手順は、同図に示すような6通りの順序が可能である。6通りのそれぞれのケースでエラーの起こりやすさは異なる。BのピースをAあるいはCよりも先に組みつけると、Bが正しい組み付け位置からずれて組み付けられる確率が高まる。このため、その後のAあるいはCの組み付け時にエラーが発生する確率の増加につながってしまう。エラー発生が少ない組立手順は、同図に示したグレー色のハッチングを施した2種類のパスである。本機能では、このような2種類のパスを統計データに基づく確率計算により求めている。計算結果からエラーの発生が少ない組立手順を選ぶことが可能となる。また、実際のエラーの検知箇所、復旧手順の示唆を得ることが可能となる。

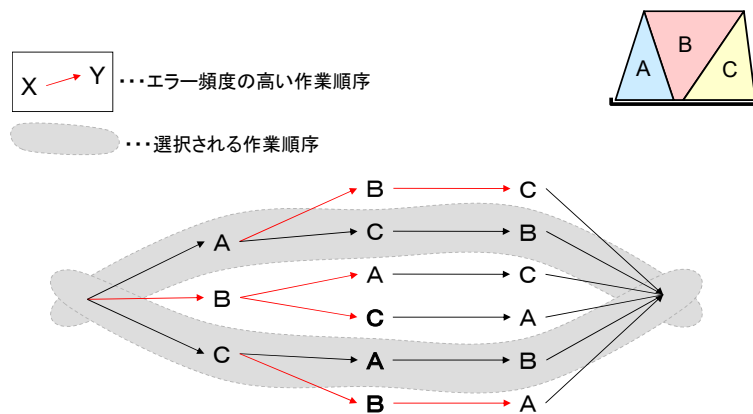


図 3.3 エラー発生リスク分析の概念

### 3.4 ハンドライブラリおよびハンド

ロボットハンドにより対象物を把持する際の過渡現象を解析し、対象物体の初期位置ズレとハンドの指の制御方法との関係から、把持の成否を推定する機能である [46–50].

図 3.4 は、図左上に示された、客先条件としての部品形状および許容すべき把持動作開始初期のハンドと部品の位置ずれ誤差（これは使用するセンサやロボットの実現精度により、与えられる客先条件の一種である）と、最終的な把持位置とが与えられたとき、本ハンドライブラリが保有しているハンド機構および把持戦略の候補のなかから最適なものを導出する動作を行えることを示している。最終的に、図中の右上の把持戦略（ハンドの指の動作軌道の計画など）あるいは与えられる部品形状の把持に関して、汎用性の高いハンド機構が出力される。

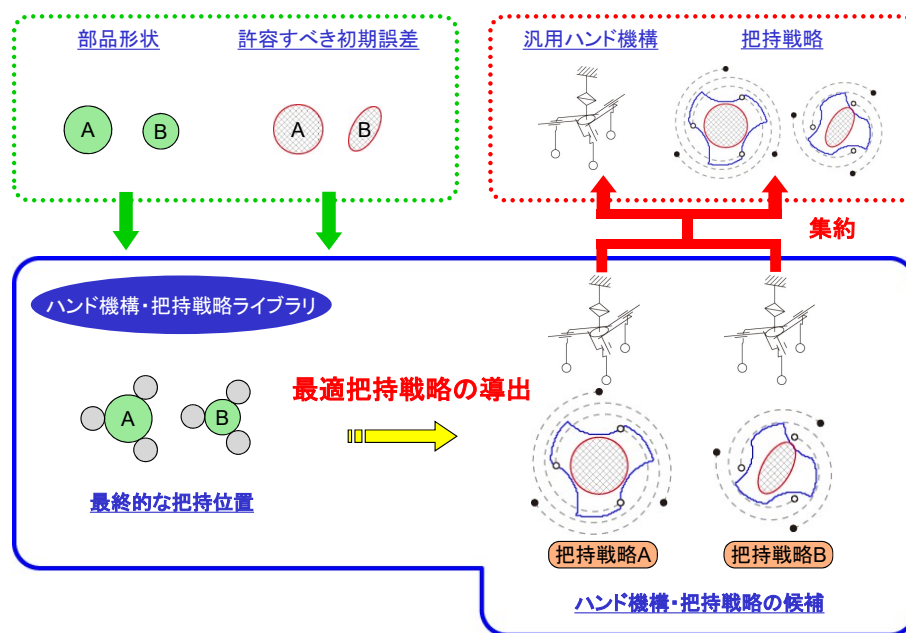


図 3.4 ハンドライブラリおよびハンドの設計について

### 3.5 最適作業動作生成

自動化システムの受発注は要求する機能が完成することで検収される「機能請負」が通常で、手戻りで工期が不安定になることを見越して全て上乗せした見積書が作られる。複雑な組立作業を「機能請負」する場合、作業者が試行錯誤でロボットに作業動作のシーケンスを教示しているが、センサによる状況認識とその結果に基づいて行動を分岐させる高度なプログラミングは、だれでもできる作業ではない。

そこで本研究では、人が時間をかけて得る組み付け作業動作シーケンスを先取する手法を開発することで、巧みな行動分岐プログラミングの自動化、ノウハウレス化を狙った。組み付けの巧拙は不確実性への対処の巧みさに依存しているとの観点から、組み付け過程での概念状態に不確実性を導入し、そのエントロピを徐々に減少させながら、ゴールを目指す作業設計のアプローチを行った、実際のロボットにて、部品挿入作業を例題とする作業動作が成功することを確認した [51].

この理論の例題として、図 3.5 に示す表記法を用いて、図 3.6 のようなペグ・イン・ホール作業を例題にして、最適作業動作の解析を行った例を示す。なお、写真でわかるように、動作の観測をしやすいようホール側は側面を切り開いてある。

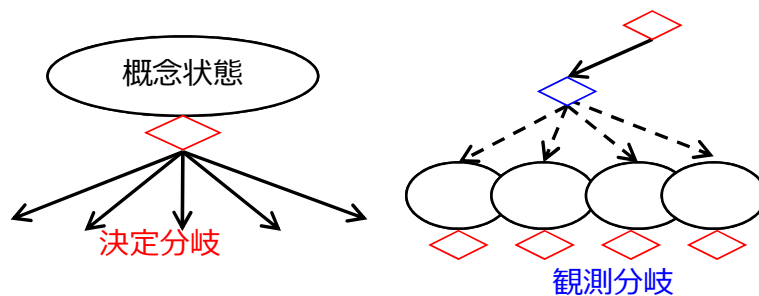


図 3.5 不確実性を解く最適作業動作生成アルゴリズム：表記法の説明

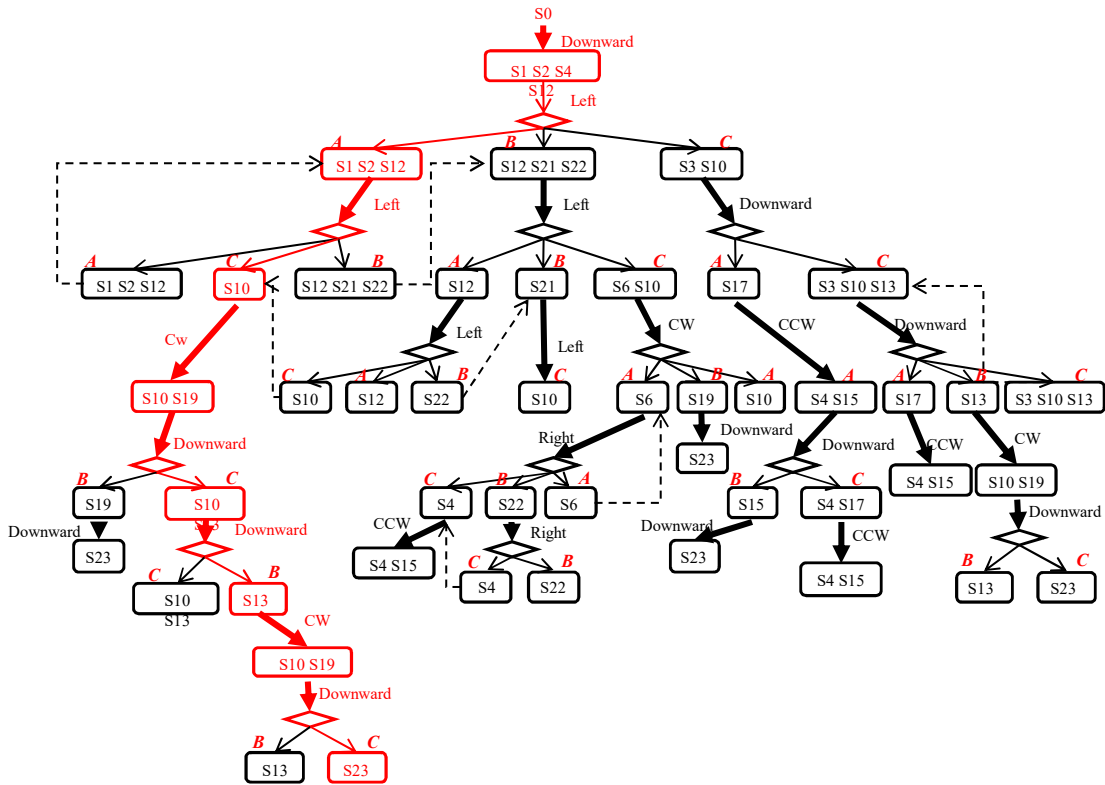
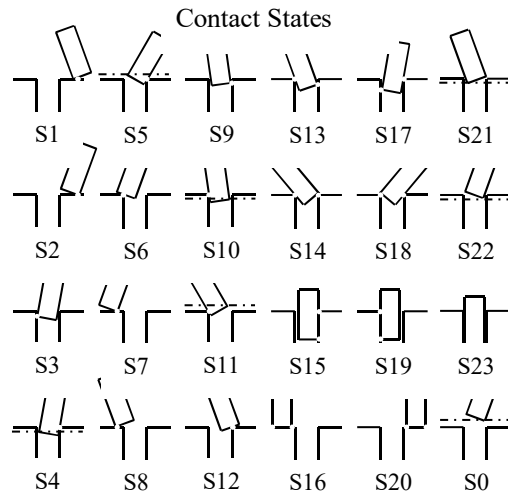
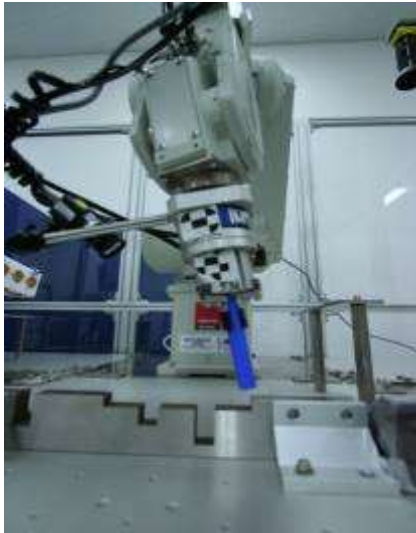


図 3.6 あるロボット作業における最適作業動作の解析例

### 3.6 エラーリカバリ

ロボットを動かし続けると、いわゆるチョコ停が発生し生産活動が停止する。作業者はロボットの再教示を実施してエラー再発防止に努めるがこれは属人的な熟練作業の典型例である。そこで作業の効率化と作業に関する知識の再利用性・継承容易性を高めるため、ロボットによる製品組み作業の記号構造と、ロボット教示に要する知識を分類し、演繹学習と事例ベース推論の手法により対話型知識獲得と獲得した知識を再利用するシステムの開発を進め実製品組立を例題にして効果を実証した [52-57]。

また、図 3.7 に示すような物体把持理論の研究の過程で、生産現場で広く用いられる平行グリップハンドが基礎とする把持理論は「力による拘束」であり、物体を握り始めて把持が完成していく短い時間のなかでの過渡的な現象において、図 3.8 に示すごとく、把持に成功する場合と失敗する場合（図 3.9）とで、ハンドのアクチュエータ負荷の時間変化に差異が見られることがわかった。このことが、エラーリカバリ動作のきっかけとして利用でき、あるいは作業の成否のログを収集することに使えたのである。この発見が電動ハンド製品の応用例につながった。

さらに、図 3.10 に示すように、コネクタ挿入作業、ねじ締め作業のエラー検知方法を見出し、そこからのエラーリカバリ動作の発現方法を発見した [36]。


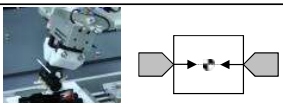
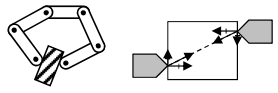
|          | Form Closure  | Force Closure  |   |
|----------|---|--|---|
|          |   | 力による拘束   | 力とモーメントによる拘束  |
| 原理       | 対象物の自由度を完全に奪って静定させる   | 力を釣り合わせて静定させる  | 力とモーメントを釣り合わせて静定させる   |
| 現象の解析    | 準静的に解析  | 動力学的に解析  | 動力学的に解析   |
| 演算量      | 少ない   | 中間   | 多い  |
| 機構       | 中間(自由度が多い)  | 単純   | 複雑  |
| 安定性(対外力) | 安定  | 力を制御すれば、安定   | 制御可能範囲が狭く不安定になりやすい  |
| 用途       | 産業用ロボットの多指ハンド設計と制御 (cf. 3つ爪)  | 産業用ロボットの平行グリップハンド設計と制御   | ヒューマノイドによる把持物体の操り   |
|          |  |  |  |

図 3.7 物体把持理論について

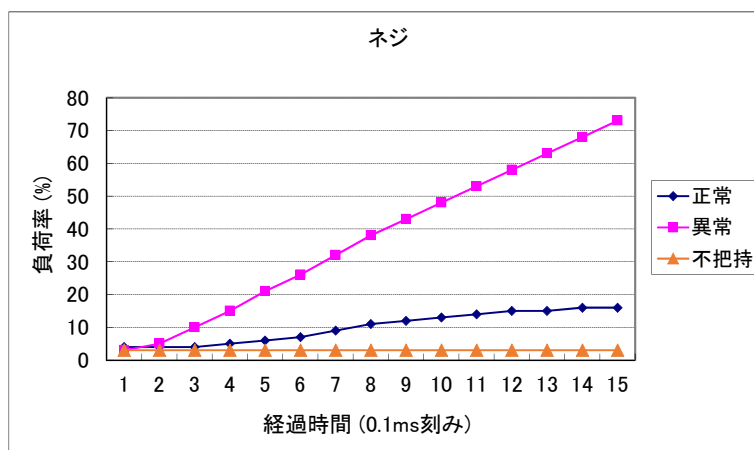


図 3.8 物体把持時の握力 (アクチュエータ負荷) の時間変化の違い

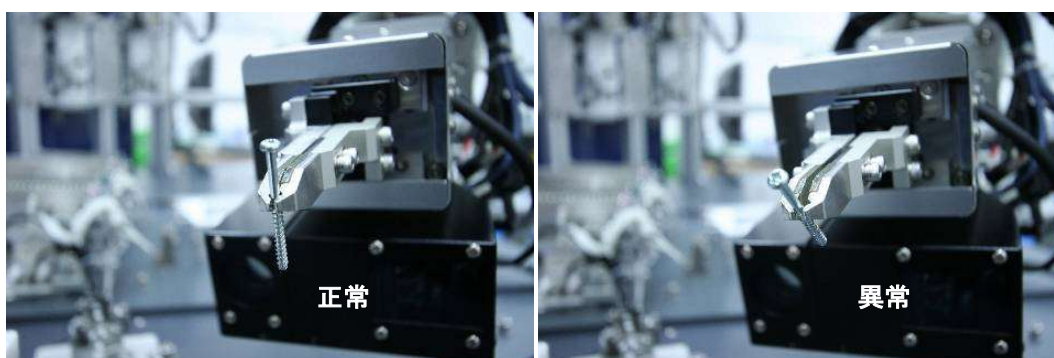


図 3.9 正常把持, 異常把持の様子

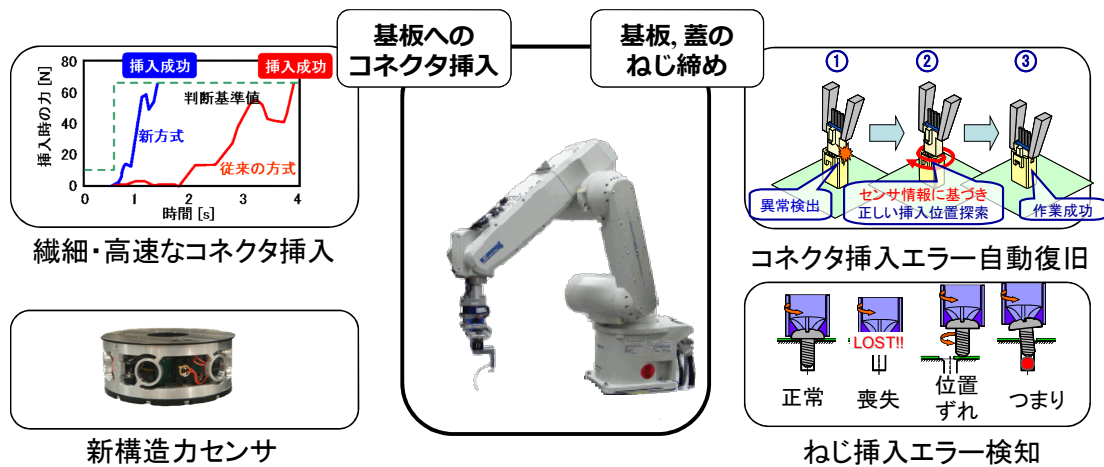


図 3.10 特定作業のエラーリカバリ例

### 3.7 教示作業

産業用ロボットは、その据付調整時に、ロボットの動きを、逐一教え込む必要がある。これは、産業用ロボットの動作プログラムの大部分が、ティーチングプレイバック方式によるからである。この作業は、「教示」と呼ばれており、図4に示すように、教示作業者が、ティーチングボックス(あるいはティーチングペンダント)と呼ばれる操作箱を握り、主として目視を頼りに実施している。教示点1点毎に作業時間を要し、点数は100点以上に及ぶことがある。この結果、作業者の技量に応じて教示作業時間とロボットに教え込まれた動作にばらつきが生じる。つまり、この教示作業時間の短縮とノウハウレス化が課題となる。著者らのグループでは、まず、ロボットによる部品組み付け動作の教示作業内容の構造を分析した。その結果、教示作業の進行状況を教示作業者が知覚できるように提示することが教示作業支援に有効であることがわかった。そこで、ロボットに取り付けた力覚センサ情報を含む複合情報を GUI (Graphical User Interface) を用いて、効果的に教示作業者に提示する枠組みを提案、実システムを開発し、その効果を確認した [58–60]。

さらに、ロボットを用いた生産システムはそのライフサイクルのなかで経年変化・故障修理・移設・複製といったイベントを迎え、システムに外乱が加わり教示位置ズレ復旧が問題になる。そこで著者らは教示データの構造についても分析



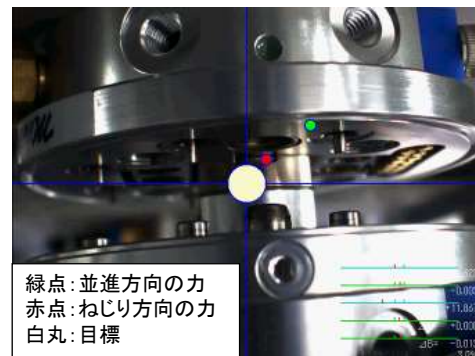
し、教示作業を再定義した。これまでの教示作業は式 (1) の左辺のロボット座標系のもとでの  $n$  点の教示点に外乱が発生するたび、 $n$  点全ての教示をやり直していた。著者らの新手法は、簡単に言えば、次式の右辺の外乱前後で不変な教示点集合を、外乱前後の状態の校正行列  $M$  で変換して、左辺の本来の教示点集合を、 $n$  点の再教示無しに得るというものである。

$$({}^r q_1, {}^r q_2, {}^r q_3, \dots, {}^r q_n) = M_{w \rightarrow r}({}^w q_1, {}^w q_2, {}^w q_3, \dots, {}^w q_n) \quad (3.1)$$

このもとで、ロボットに取り付けられた力覚センサを用いて、外乱前後に周辺環境をタッチして位置関係を認識、作業者のスキルによるばらつきのない校正行列  $M$  を迅速に得る。この枠組みのもと CAD データの援用技術、力覚センサとビジョンセンサを併用したロボットシステムの位置復旧支援技術を開発し実機にて効果を確認した [61]。



産業用ロボットの教示作業



開発したヒューマンマシンインタフェース

図 3.11 教示を支援するヒューマンインタフェース例

### 3.8 3次元画像センシング

ロボットが作業対象物と自己との関係性を認識する機能を得ることで、ロボット化困難作業の克服と据え付け調整に係るエンジニアリングコストの削減が期待できる。

カメラによるビジョンセンサが、その筆頭である。単眼カメラによるセンシングは古くから工業化していたが、2010年代後半に入ると、ステレオカメラあるいはパターン投光機と単眼カメラによる距離画像を取得して画像処理、認識処理を行なう3次元画像センシングが実用域に入った。

著者らの研究グループでは、組立対象部品のバラ積み状態の認識を目的として、独自の3次元ビジョンセンサからの距離画像情報を処理してバラ積み部品をつまみ上げるための把持点を認識するモジュールとしてピッキング用物体認識モジュールを開発した(図3.12)。ビジョンセンサを用いた認識プログラミングにおいても、コストの観点から迅速化は重要である。そこで3次元ビジョンセンサから得られた距離画像と2次元画像情報を融合し、教示作業に有効な作業ポイントを対話的に提供する・ハイブリッド視覚補正モジュールを開発した。

一方、広義の画像センシングとして、物体把持および操作の高度化を目的とする触覚アレイセンサは有望株である。著者らの研究グループでは力覚センサと触覚アレイセンサとのセンサフュージョンにより、把持対象物自体および把持の状態を推定する研究に取り組み、一定の知見を得た [62]。

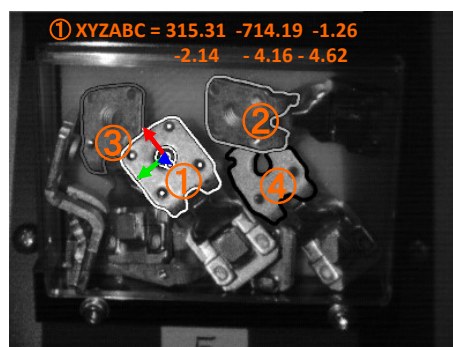


図 3.12 3次元物体の画像認識例

### 3.9 システムインテグレーション

システムインテグレーションは研究開発成果の総合的な実力が問われる重要機能である。そこで、品質保証試験を経た専用ハードウェアにより手堅く作られたレガシーなFAシステムに、新しい価値をもたらす智能モジュール・サブシステムを追加することで、産業用ロボットシステムが従来通りの安定性を保ちながら進化する。このような考え方で統合システムを構築することにし、新旧のサブシステム間を結合するためのシステムインテグレーション・ソフトウェアを開発した。図3.13の太い矢印の位置で本ソフトウェア群が動作する [32].

近年、ロボット用のソフトウェアの使い回しを容易化するため、RTミドルウェア (Robot Technology Middleware) [63] や、ROS(Robot Operating System) [64] の開発が進められている。本研究では、RTミドルウェアを用いて各機能を実装するとともに、レガシーシステムとの間のインタフェース部分については、RTMとレガシーシステム間をブリッジする機能を独自開発した。

本論文の第6章にて実証システムのシステムインテグレーションを詳述する。

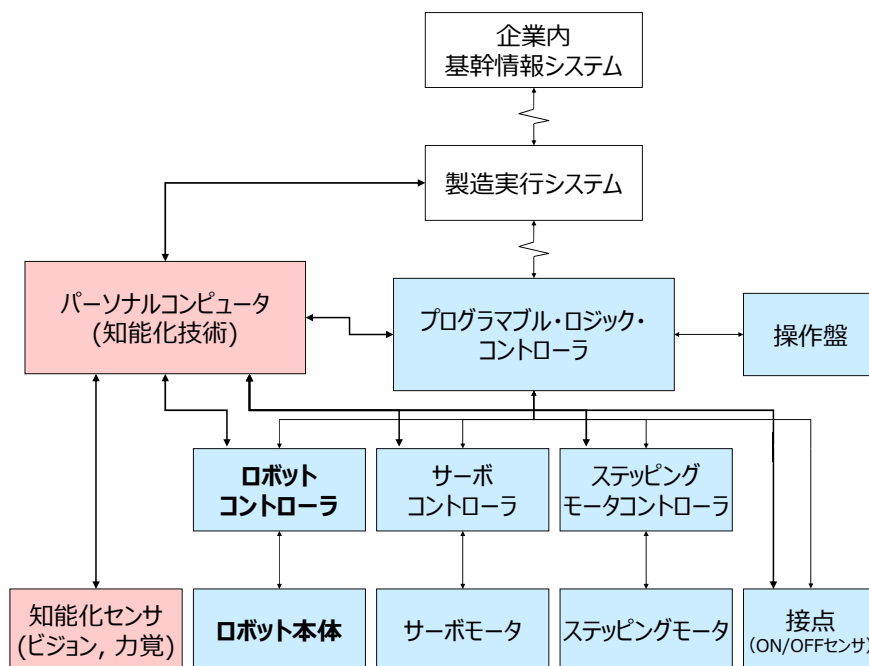


図 3.13 自律型ロボットセル生産システムのインテグレーション例

### 3.10 動作習熟

上述してきた知能化技術では実現し得ない機能がある。それは、人がにわかには解決困難な課題につきあつた時の創意工夫により生み出される機能である。これを「自律性」と呼ぶことにする。例えば、人はある作業を繰り返すとき、その作業に習熟して速度や成功率が上がっていく。著者らのグループでは、このような現象のメリットを、与えられたプログラムを繰り返すだけのロボットに取り込むことを考えた。人による習熟は、「次にどのような試行をすれば自分は、より習熟できるか」を常に考え続けることで、効率的に進行する。そこで、ある時点までに得られている観測信号を用いて次の試行の最適化を行ないながら探索試行を繰り返して、系の性能を上げていく能動探索アルゴリズムを開発した。これをロボットをはじめ、制御パラメータを調整する必要がある複雑な系に適用し、効果を確認 [65] している。

本技術により、ロボットをはじめモデル化困難な複雑な制御系においてパラメータを熟練者が調整するしかない局面で、ノウハウレス化と調整時間の短縮が期待できる。

同アルゴリズムは本論文の主題であり、第4章にて詳述する。

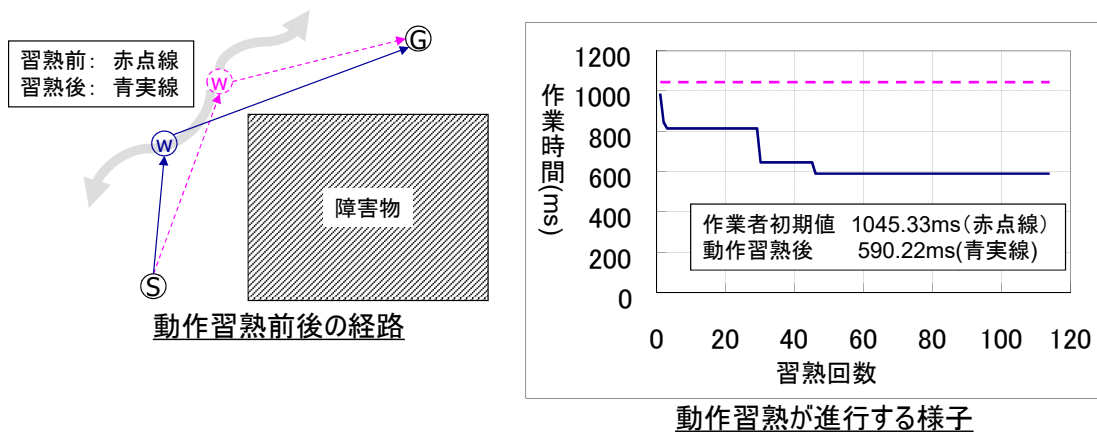


図 3.14 動作習熟の概念と効果

## 4. 技術プラットフォームによるロボット化困難作業の克服

以上、述べてきた要素技術のインテグレーションにより、従来はロボット化困難とされた作業をも克服できるかどうかを検証するため、下記の2題に取り組んだ。

### 4.1 柔軟物ハンドリング

従来の製造業向けのロボットの動作はいわゆる「位置決め」を基本とし、その動作プログラミングもティーチング・プレイバックと呼ばれている。物体を把持する位置、障害物を避ける退避位置、物体を搬送する軌道についてはその経由点の位置を、現物あわせてロボットコントローラに記憶しておき、生産運転時に記憶している位置を順番通りに再生していく。毎回位置が変わる柔軟物を取り扱うためには品種ごとの専用設計される機構と制御系が必要であり、その設計・調整作業にかかわるコスト増大を招き、また経年変化や想定外の形状ばらつきなどにより動作の安定性に課題が生じている。そこで本研究では、3次元ビジョンセンサ、力覚センサからの計測信号を認識処理することで対象物の状態を推定し、把持して搬送、エラー発生時にはリカバリ動作する機能をロボット技術プラットフォームに追加した。その結果中間状態を保持する汎用性の高いジグは用いたものの、後述する実証システム上でコネクタ付きケーブル部品の製品への組み付け作業が成立することを確認した [36]。

### 4.2 バラ積み部品供給

自動組立システムにおいて、部品箱の中に部品を無作為に多数投入したバラ積み状態で供給することが、部品保管・運搬の効率上、切望されている。これまでは、部品の種類数と同じ数だけ、パーツフィーダと呼ばれる高価な専用供給装置を用いて部品を整列する必要があるが、複雑な形状の部品の供給については、その整列を人手に頼らざるを得なかった。これはロボット学の古典的難題のひとつとして知られているランダム・ビン・ピッキング問題そのものである。

本研究では、汎用機種4台のロボットによる流れ作業で、バラ積み部品箱から取り出して、部品パレット上に部品を整列して、後段の自動組立工程へ供給する技術を開発し、電気電子組立における部品供給の問題を解決した。同技術を用いた実証システムにおいて最速3秒台の周期にて1台目のロボットが取り出した小さくて細かい小型電気電子製品の部品を、残り3台のロボットで持替えながら3次元空間内の任意姿勢へ整列できることを確認した。

その研究開発過程で、部品操作の作業手順の計画で「自律性による試行錯誤」が必要なことがわかり、これを実現する自律システムアルゴリズムの研究開発に取り組んだ。

これらが本論文の2つ目の主題であり、第5章で詳細に論述する。

# 第4章 動作習熟機能を実現する自律システムアルゴリズム

本章は、自著ジャーナル投稿論文 [65] の内容に、「次のサンプリング点選択の最適化」のための評価値計算式の設計などについて加筆したものである。

## 1. はじめに

本章では、自律的な習熟機能をもたらす自律システムアルゴリズムの開発について述べる。従来の製造業向けロボットは、その据え付け調整時に教示した内容を精確に繰り返すので生産能力は一定であるのに対し、人セルでは作業者が担当作業を繰り返すうちにその作業に習熟して作業速度を向上させ、かつ作業ミスが減らしていく。このような人の習熟機能は自律性に支えられている。自律性とは自ら考えて行動計画することであり、習熟においては単に教えられたことを丸覚えするだけでなく、自身で考えて試行錯誤を行っていると考えられる。

この試行錯誤を機械学習のスキームで考えると、学習データセットが外部から与えられるのではなく、試行錯誤では学習機械自ら探しだしているという違いがある。例えば、生徒が与えられた大量の教材を丸暗記する勉強法をとるのではなく、生徒自ら、教師に対して自分が苦手なところを次々に質問して正解を得て、自らの弱点を潰していくイメージである。

本研究では、この試行錯誤による習熟を、未知の目的関数を探索するときの次の試行の最適化として定式化して、独自の自律システムアルゴリズム「能動型探索アルゴリズム」を開発した。

なお、ここでいう「未知の」とは、その関数の型式がノンパラメトリック回帰式においてパラメータが「未知」であることを指している。

同アルゴリズムは、過去の経験の記憶のモデルから次の一手の候補を準最適化して試行し、得られた結果を記憶モデルに追加することにより、次々と試行を繰り返すものである。このアルゴリズムを実システムに適用したところ、ロボットシステムの動作速度の向上だけでなく、金属加工機の加工条件調整など、制御パラメータを調節する局面に水平展開できることがわかった。

複雑かつ未知の入出力特性を持つ工業プロセスの制御パラメータを最適化する際、対象のシステムに対して適当な入力を与えて出力を得る操作(サンプリング)を複数回実施して、入出力特性を探りながらパラメータを調整する必要がある。

この手の問題に対しては、それぞれの対象システムごとに、いわゆる熟練作業者が存在し、その熟練者が自らの経験と知識を頼りに試行錯誤を繰り返しているのが通例である。

最適化の結果は個人の技量に依存し、かつ確率的にばらつく。この手の問題が産業用途の多くの場面で見いだせるため、このような熟練作業を自動化することは、短期的には調整コストを削減し中長期的には熟練者を含む生産人口の減少対策につながるため、産業的価値がある。同時に理論から適用範囲が明確化された技術に昇華させるためには、何でも解けるという主張ではなく、適用範囲の確認と明示が重要であるので、本研究ではいくつかの適用事例を示すことにした。

本章では、上述の問題を未知の目的関数の最適化に帰着させ、その解法として開発した能動学習のスキームに基づいて自律的にパラメータを探索するアルゴリズムの構築と、その適用事例について述べる。

## 2. 関連研究

未知の目的関数の最適化は、やってみるまでわからないという古典的な難題のひとつであり、効率よく解くための様々な取り組みがなされてきた。例えば実験計画法 [66] では、効率よくシステムの入出力特性をサンプリングする方法、サンプリング結果から最適動作点を抽出する方法、その妥当性を評価する手法を、提供している。最適計画 (例えば [67] の pp.247-259) では、サンプリング集合の選び方のよしあしを測る数理的な尺度を最適化する意味での効率的なサンプリングを



実現している。

しかしながら、これらの手法において、正解を含んだ制御パラメータの定義域を選び出す作業は、実験者の手に委ねられる。システムの入出力特性が線形であれば、それは比較的容易な作業であるが、システムの非線形性が強い場合は、途端に難しくなる。熟練した実験者がこれらの方法で成功するのは、経験的に正解が含まれる範囲をある程度知っていること、および、非線形性の強いシステムであっても実験範囲を狭くすれば、その範囲内においては線形近似値と実値のずれが小さくなり、線形なシステムとみなして取り扱えることによる。

実験範囲選択の課題に対し、応答局面法 [68] では、実験範囲を自動的にスライドしていく手法を提供している。しかしながらサンプリング回数はそれなりに多くなること、および、システムの強い非線形性の克服には別の工夫が必要となる。

一方、システムの入出力特性をサンプリングした標本集合があらかじめ得られている場合は、その集合に対し目的関数を回帰する機械学習アルゴリズムを用いて最適化を実施する方法が知られている。特に階層型のニューラルネットのバックプロパゲーション学習則 [69–71] がよく知られており、近年はディープラーニング手法 (例えば [72]) が注目を集めているが、回帰関数の誤差を減らすためには十分な量の標本集合が必要となるため、サンプリングにコストがかかる問題に対する適用は困難である。

サンプリングコストの課題に対し、あらかじめ標本集合を取得しておくのではなく、サンプリングを逐次的に、かつ自動的に繰り返すスキームとして強化学習 [73] があるが、次のサンプリング点を効率的に決定する方法は強化学習のスキームの外側にある。

そこで、本研究では、広い実験範囲の中から、より少ないサンプリング回数で非線形な未知の目的関数を自律的に最適化するという課題に対し、動的に次のサンプリング点を逐次選択しながら目的関数を最適化する能動学習 [74,75] のスキームに基づく方法を用いることにする。このもとで、自律的に次に試行すべき準最適サンプリング点を選択し、得られた標本集合に基づいて、対象となる未知の目的関数モデルの学習を繰り返しながら、最適な制御パラメータを探索するアルゴリズムを開発した。その最大の特徴は、ある時点で得られている情報から、次の

サンプリング点の選択を最適化する方法にある。

さて、一般的に、能動学習の基本戦略は、解くべき問題の設定により、例えば2つに大別される。ひとつめは、サンプリングにはコストがかからないものの、そのサンプルに対し、例えば合否のラベルをつける作業にコストがかかるという問題設定で、受動的かつ大量に得られるサンプルのうち、どうすれば、ラベルをつけるべきサンプルの数を減らせるかを論ずるもので、Selective Sampling [76]と呼ばれる。もうひとつは、次のサンプリングを学習アルゴリズム自らが能動的に生成する方法で、Query Learning [77]と呼ばれている。今回開発したアルゴリズムは、後者に分類される。

ここからは、未知の目的関数の最適化の能動学習的解法という設定の先行研究と、本研究とを対比させる。

例えば [78, 79] では、目的関数のモデリングにラジアルベシス関数あるいはサポートベクターマシン [80] を用いている。そして、次のサンプリング点を仮に定め、そのサンプリングをしたことにより得られるであろう情報量に相当する量を計算して、そのサンプリング候補点の評価値とすることで、複数のサンプリング候補点の中から評価値最大のを次に選ぶべきサンプリング点とする最適化を行っている。

また、能動的にどれを選んでラベルをつけるかというスキームに属するものではあるが、[81] はサポートベクターマシンの性質を利用して、次のサンプリング点候補に0あるいは1それぞれのラベルをつけたと仮定した場合ごとの推定モデルを計算し、それら2つの推定モデルが描く識別境界の差分の最小値を最大化する評価方法により、次のサンプリング点を最適化する方法を例示している。[82] は、そのような識別境界の差違を求めるヒューリスティックを改良している。なお、サポートベクターマシンで逐次学習を実現するにあたり、[83] はサポートベクターマシンの学習進行時に標本集合を増減させた場合の性質を調べている点が参考になる。次のサンプリング点の候補の評価を、そのサンプリングにより得られるであろう情報量を独自の算法で演算し、さらに、そのサンプリングを得るために必要なコストも考慮に入れて次のサンプリング点を最適化するものに [84] がある。

ただし、著者らの調査で、これらのヒューリスティックでは有効なサンプリング点候補が選べない場合があることがわかっている。簡単に言えばこれらの先行研究では、得られるであろう情報量が予測されているが、予測した情報量が得られないリスクが考慮されていないからである。この結果、大きな情報量が得られると予測されても実際にサンプリングすると情報量ゼロのケースが有り、そのようなケースが連続すると、その間、学習が全く進まないのである。

そこで、本研究では以上のことを考慮して、目的関数に対する制約条件および未知の目的関数をモデリングするための関数を一般化した形式、および、現実的な実装の両面でアルゴリズムの構築を試みる。本稿では、情報が得られないリスクと、得られると期待される情報量を同時に考慮するヒューリスティックを擁するオリジナルアルゴリズムを抽象表現で説明した上で、同アルゴリズムの適用事例とアルゴリズムの実装例について詳細について述べ、制御パラメータの調整作業が自動化される効果について考察を加える。

## 3. 開発したアルゴリズム

### 3.1 問題の定義

ある種の工業プロセスで、その制御パラメータを調整することで、プロセス時間の短縮、収量の増加といった評価指標が変化することは、これまで述べてきたとおりであり、本章ではこの問題の定式化を試みる。

例えば、ある種の金属加工では、制御パラメータを変化させることで、加工速度が変化し、プロセスが実行不能になるパラメータ組み合わせが存在する。定性的には、加工現象に対して投入する加工エネルギー量が徐々に大きくなるように、パラメータを変更していくことで加工速度が増大していくが、あるところで加工の物理現象内でエネルギーが吸収しきれなくなって加工が振動的になり、加工不能に陥ってしまうと考えられる。問題は、パラメータと加工速度の関係、パラメータと加工不能領域の関係が未知かつ複雑で、にわかにはプロセスのモデル化ができないため、最適な制御パラメータをリーズナブルなコストで選び出すことが困

難な点にある。熟練作業者は自身の経験と記憶を頼りに調整作業を進めていると想像される。

ここで今一度、本アルゴリズム設計に至るまでの問題設定を述べる。対象とする工業プロセスの良否を支配する制御パラメータが存在し、その最適化問題を考えている。パラメータを調整すると工業プロセスの良否を判定する連続値で与えられる指標が変化し、これを最大化もしくは最小化したい。この指標は制御パラメータを少しずつ変化させると少しずつ変化するが、その関数は解析的には表現できず、ノンパラメトリック関数で回帰するしか無いという点で未知の目的関数となる。また、制御パラメータを変化させていると、思わぬところで工業プロセスは突然実行不能に陥る。つまりパラメータ空間内に実行可能・不能を隔てる識別境界が存在することになる。この識別境界も解析的には表現できず、ノンパラメトリック関数で回帰するしか無いという点で未知の関数である。経験的に未知の目的関数の最良値は、制約条件を与える識別境界付近に存在している。このとき、未知の関数を回帰するためのデータセットは全く与えられておらず、そのサンプリングコストは高く、全探索は困難で、できるだけ少ないサンプリングで最良のパラメータを探し当てたい。このため、制御パラメータを変更しながら探索するしかないが、この探索を自動的に逐次実行する工夫をして、最適化問題として組み立てようというのである。

このもとの、この問題を、次のように定義する。

- $m$ 次元の引数  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  をもつ未知の目的関数  $y = F(\mathbf{x})$  があり、 $F(\mathbf{x})$  が実行可能となる未知の制約条件  $f(\mathbf{x}) \geq 0$  を伴う。
- あるサンプリング点  $\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{mi})$  でサンプリングすると、真の  $F$  と  $f$  から  $y_i = F(\mathbf{x}_i)$ ,  $\text{sign}(f(\mathbf{x}_i))$  を観測、標本  $\mathbf{s}_i = (\text{sign}(f(\mathbf{x}_i)), y_i, \mathbf{x}_i)$  を得る。
- 標本  $\mathbf{s}_i$  新たな  $\mathbf{s}_n$  として加え、標本集合  $\mathbf{S}_n = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$  が更新される。
- 標本集合  $\mathbf{S}_n = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_j, \dots, \mathbf{s}_n\}$  が得られているとき、 $F$  が実行不能になった  $\mathbf{s}_j$  を除いた部分集合により、目的関数の推定モデル  $y = F(\mathbf{S}_n, \mathbf{x})$  を学習する（回帰モデルを計算する）。経験・記憶に相当。

- 同時に標本集合  $S_n$  のもとでの制約条件の推定モデル  $f(S_n, \mathbf{x}) \geq 0$  を学習する（回帰モデルを計算する）。経験・記憶に相当。
- 推定モデル群を道標にしてサンプリングを繰り返し、 $\mathbf{x}^* = \arg \max_{\mathbf{x}} F(S_n, \mathbf{x})$  なるサンプリング  $\mathbf{x}^*$  を求む。

この定式化により、制御パラメータの最適化作業は、サンプリング点  $\mathbf{x}_i$  を逐次的に、かつ適切に選ぶことによって、できるだけ少ないサンプリング数で、制約条件  $f(\mathbf{x}) \geq 0$  を満たす未知の目的関数  $F(\mathbf{x})$  を最大化する  $\mathbf{x}$  を、目的関数の推定モデル  $F$ 、制約条件の識別境界の推定モデル  $f$  を頼りにして探し当てたいという問題に帰着される。 $F$  や  $f$  の回帰モデルが得られることが最終目的ではなく、制御パラメータの組み合わせについて、実験したもののなかから最も好ましいものを選択することが目的となる。この概念を図 4.1 に示す。

図中、XY 平面上で制御パラメータの組み合わせ  $\mathbf{x}$  が変化し、それとともに目的関数  $F(\mathbf{x})$  の値が Z 軸上で変化している。同時に制約条件  $f(\mathbf{x}) \geq 0$  が  $\mathbf{x}$  の正常域と異常域を識別境界で分割している。

このとき、次のサンプリングを適切に選択し標本集合に追加することで、目的関数の推定モデル  $F$  の正確化、制約条件の識別境界の推定モデル  $f$  の正確化が同時に進行し、次のサンプリング点としての最適性を加速させながら目的関数に対して最適な  $\mathbf{x}$  を得ることが期待される。

なお、システムに投入するエネルギー量を制御パラメータで増減するプロセスにおいて一般的なシステムの入出力特性として予想されるのは、なだらかに増加する目的関数が制約条件の近傍で最大になることであり、制約条件の推定モデルにより推定される識別境界の推定モデル  $f(S_n, \mathbf{x}) = 0$  の正確さが、重要であると考えられる。したがって制約条件の識別境界付近を重点的にサンプリングすることで効率的な探索が期待される。実際、本アルゴリズムの複数の適用事例でも  $F$  は比較的になだらかで、それに比して  $f$  は非常に複雑なものであった。

最大の問題は、次のサンプリング点をどのように逐次的に選択するか、である。熟練作業者の熟練の本質は、探索的に実験試行を進めていくときの、この逐次サンプリング点最適化に長けていることにあると考えられる。

本研究では、このような選択を能動的に繰り返しながら、最適な制御パラメータを探索するアルゴリズムを開発することにし、これを能動型探索アルゴリズムと呼ぶことにする。

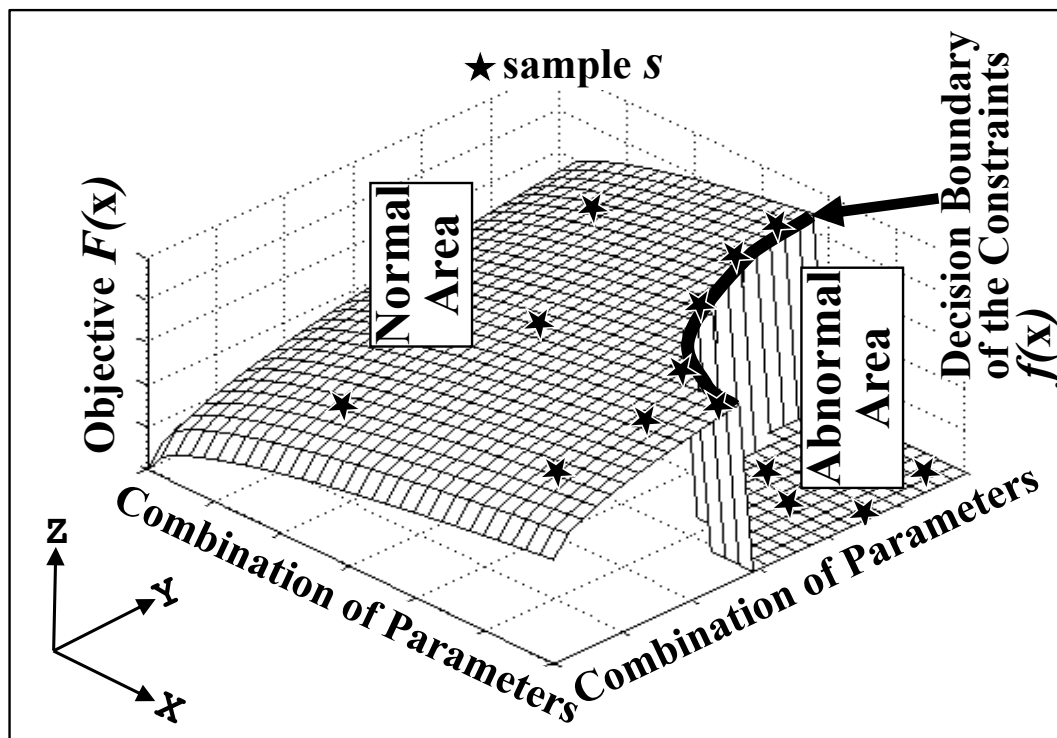


図 4.1 未知の制約条件付きの未知の目的関数最適化の概念図

### 3.2 逐次サンプリング点最適化を備えた能動型探索アルゴリズムの詳細

まず、抽象化した形式で、能動型探索アルゴリズムを説明する。具体的な実装例は本章 3.4 節で示す。本アルゴリズムの重要な機能として、逐次サンプリング点最適化に関する部分を、図 4.2 に示す。

図中の左側の図形は、制約条件  $f(\mathbf{S}_n, \mathbf{x}) \geq 0$  の識別境界をプロットしたものである。このとき、次のサンプリングを実施して標本集合が更新された後の、識別境界  $f$  の変化は、 $\text{sign}(f(\mathbf{x}_{n+1})) \geq 0$  となる場合、あるいは  $\text{sign}(f(\mathbf{x}_{n+1})) < 0$  となる場合の、いずれになるかで異なる。標本  $\{\text{sign}(f(\mathbf{x}_{n+1})) \geq 0, y_{n+1}, \mathbf{x}_{n+1}\}$  を加えた標本集合  $+\mathbf{S}_{n+1}$  および他方の  $-\mathbf{S}_{n+1}$  による識別境界を、それぞれ  $h, g$  とする。

ここで  $f$  と  $h$  の距離を定義する。これを識別境界の存在確率分布の差異と考えて、カルバック・ライブラー情報量 [85] を用いて、図 4.2 の右側に示すように、その距離  $A_{fh}(+\mathbf{S}_{n+1})$  とする。これを、そのサンプリングの結果、 $f$  が  $h$  に変化したこと得られる情報量であると仮定する。 $f$  から  $g$  に変化した場合の情報量  $A_{fg}(-\mathbf{S}_{n+1})$  も同様とする。

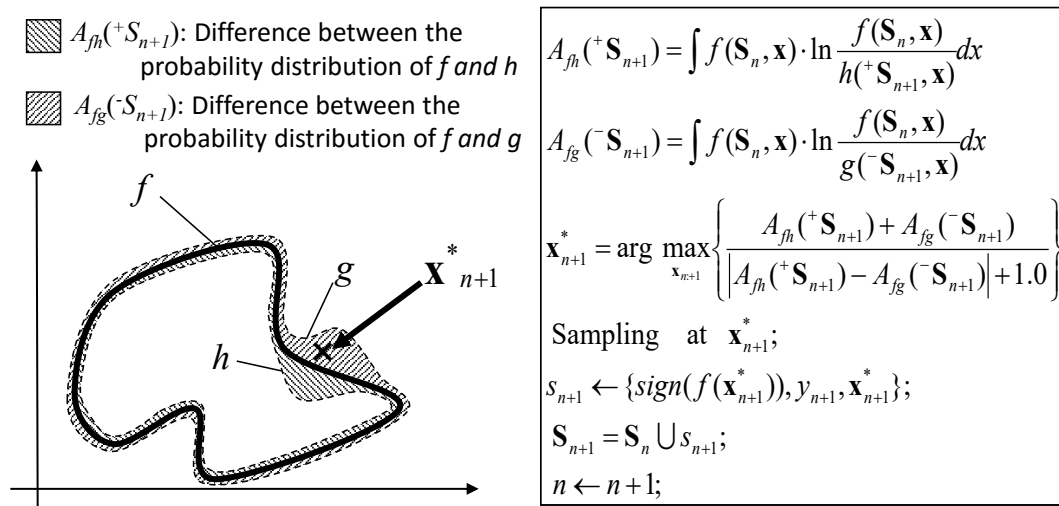


図 4.2 次のサンプリング点の最適化

$A_{fg}$  と  $A_{fh}$  の和は、そのサンプリング候補により得られると期待される情報量であるとする。一方、 $A_{fg}$  と  $A_{fh}$  の差は、得られる情報量の差であるので、差が大きいということは、そのサンプリング候補により情報が得られないリスクが大きいことを示しているとする。すると、ゼロ割りを避けるための小さな数 1.0 を分母に加えた

$$\frac{A_{fg} + A_{fh}}{|A_{fg} - A_{fh}| + 1.0} \quad (4.1)$$

を、そのサンプリング候補の評価値を得るためのヒューリスティックとすることで、本研究が目的とする次のサンプリングの準最適解

$$\mathbf{x}_{n+1}^* = \arg \max_{\mathbf{x}} \left\{ \frac{A_{fh}(+\mathbf{S}_{n+1}) + A_{fg}(-\mathbf{S}_{n+1})}{|A_{fh}(+\mathbf{S}_{n+1}) - A_{fg}(-\mathbf{S}_{n+1})| + 1.0} \right\} \quad (4.2)$$

を選び出すことが可能となる。

次に、能動型探索アルゴリズムの全体構成を図 4.3 に示す。上述のサンプリング点最適化以外に、いくつかの機能が必要となるので順を追って説明する。まず、初期サンプリング (Initial Search) についてである。これには、例えば、乱数により、任意の点を選び出してサンプルするか、あるいは直交表 (Orthogonal Array) を用いてサンプリング点群を設計すれば良い。直交表は、複数の入力パラメータを変化させた入出力特性のデータ取りをしたいときに用いる数表であり、一言で言うと、定義域に対して網羅性、かつ一様性を持った実験を、少ない実験回数で実現する実験計画のために用いられるものである。具体的には、各パラメータの実現値を数段階に量子化して、例えばパラメータの定義域を二等分した場合は定義域の左端に 0、中点に 1、右端に 2 と番号を付け、各パラメータについての 0, 1, 2 の組み合わせ方を数表にしたものである。パラメータを量子化したとしても、すべての組み合わせを実験するとコストがかかるため、実験が網羅的かつ一様に行えるという指標で全ての組み合わせの中からいくつかの組み合わせだけを選び出して、数表に記載してある。直交表には様々な種類がありアルファベット L と組み合わせ数を並べて L27 などと呼んで区別する。より詳しい直交表の解説は例えば [67] の pp.87–pp.91 に示されていて、同書 pp.121–pp.124 には一つのパラメータを 3 段階に変化させて組み合わせる 3 水準系の直交表とその一例として L27 についての解説がある。話を戻して、例えば直交表 L27 を用いると、制御パラメー



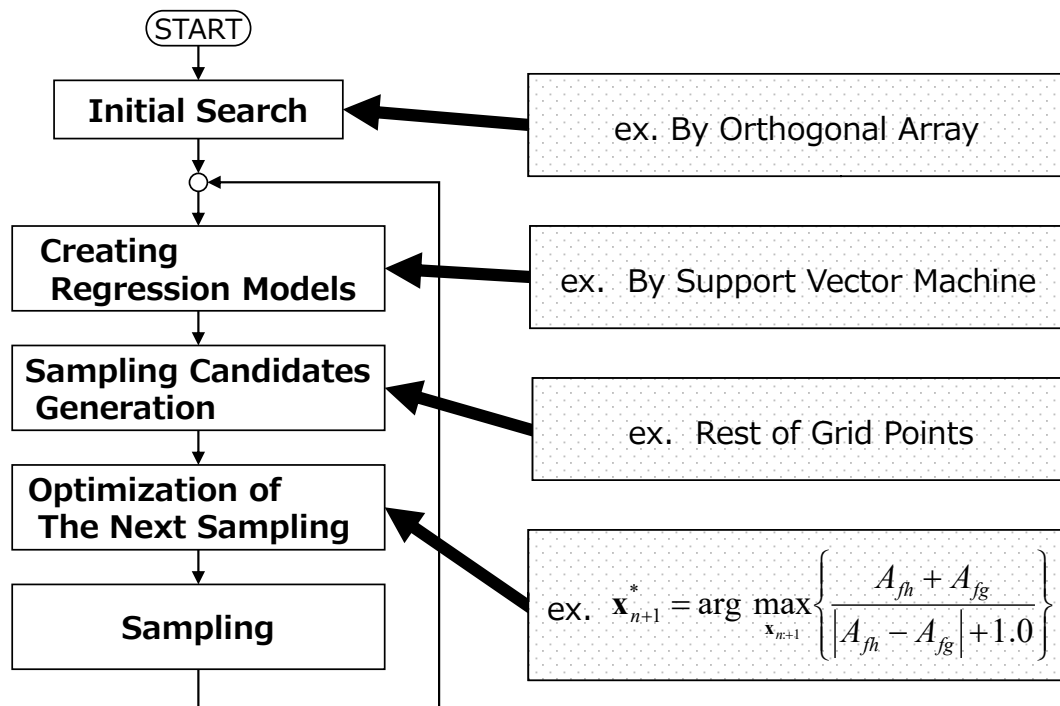


図 4.3 能動型探索アルゴリズムの全体フロー

タの組み合わせについて，最大で 13 種類の制御パラメータについて各パラメータごとに 3 つの水準を振った組み合わせ総数  $3^{13}$  のうちの 27 通りの組み合わせを選ぶことができる．これに従って 27 回のサンプリングを実施して，初期の標本集合を得る．

ここからは，逐次探索が開始される．まず回帰アルゴリズムにより推定モデルを計算する (Creating an Regression Model)．そして次のサンプリングの候補を発生し (Samping Candidate Generation)，その中から最適なものをひとつ選択する (Optimization of the next sampling)．これを用いて実際にサンプリングを実施する (Sampling) と標本集合が更新され，推定モデルも更新し，以降，同様のことを繰り返す．

### 3.3 逐次サンプリング点選択最適化の評価値計算式について

ここで、逐次サンプリング点選択の最適化のための評価値計算式について、考察を加える。前節で述べた同式のコア部分は次のとおりである。

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} \frac{\text{得られる情報量}(\mathbf{x})}{\text{情報が得られないリスク量}(\mathbf{x})}$$

直感的には、「得られる情報量」が最も多い  $\mathbf{x}$  を選ぶことで最適に思える。しかし、数値実験結果は、本研究が独自に提案する「情報が得られないリスク量」の項が重要であることを示している。

本節ではこれを考察するため、2次元の探索空間  $S = \{\mathbf{x} | \mathbf{x} = (x_1, x_2), 0 \leq x_i \leq 1\}$  における未知の制約条件の識別境界  $f(\mathbf{x})$  を以下の箇条書きの5つとし、これら5つのそれぞれの場合に対して (i) から (iii) の3つのケース、(i) 「得られる情報量／情報が得られないリスク量」（提案する算式のケース）、(ii) 「得られる情報量／1.0」（得られる情報量のみを計算するケース）、(iii) 「1.0／情報が得られないリスク量」（情報が得られないリスク量のみを計算するケース）を比較してみる。

- 線形分離

$$f(\mathbf{x}) = -x_2 - x_1 + 0.5 \text{ の上下}$$

- 単円

$$f(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.09$$

つまり中心 (0.5,0.5) 半径 0.3 の円の内外

- 複円

$$f(\mathbf{x}) = (x_1 - 0.9)^2 + (x_2 - 0.9)^2 - 0.16 \text{ および } f(\mathbf{x}) = (x_1 - 0.6)^2 + (x_2 - 0.6)^2 - 0.04$$

つまり中心 (0.9,0.9) 半径 0.4 の円の内外および中心 (0.6,0.6) 半径 0.2 の円の内外の和領域

- 離れ円

$$f(\mathbf{x}) = (x_1 - 0.9)^2 + (x_2 - 0.9)^2 - 0.16 \text{ および } f(\mathbf{x}) = (x_1 - 0.6)^2 + x_2^2 - 0.09$$

つまり中心 (0.9,0.9) 半径 0.4 の円の内外および中心 (0.6,0.0) 半径 0.3 の円の内外の和領域

- L 字形

座標値 (0.5, 0.25), (0.75, 0.25), (0.75, 0.75), (0.25, 0.75), (0.25, 0.5), (0.5, 0.5) の各点を結ぶ多角形の内外

これらの識別境界が未知であるとし、ある  $\mathbf{x}$  でサンプリングをすることで、 $f(\mathbf{x})$  の符号のみが判明することにする。初期探索は直交表 L9 を使用し探索空間各軸、隅、真ん中、隅の 3 水準づつ、合計  $3 \times 3 = 9$  点を実施する。

各識別境界の算式ごとの最終探索結果を、図 4.4 に示す。各図は今回の計算機実験のために開発したコンピュータソフトウェアの表示画面のスクリーンショットである。格子点数は X 軸, Y 軸それぞれの定義域を 20 等分する 21 点  $\times$  21 点 = 441 点である。赤点、青点は試行により得られる真値を示し、赤い領域と青い領域の識別境界はサポートベクター回帰した推定モデルで、各格子点の小さな四角はサポートベクターを表す。他の物より大きな四角は最後の試行点である。識別境界の両脇にある黒点は  $0.99 < |f| < 1.01$  の領域を黒くプロットしたもので、おおむね  $|f| = 1$  すなわちマージン付近を示す。

また、各算式 (i)(ii)(iii) による探索推移の違いを、図 4.5, 図 4.6, 図 4.7, 図 4.8, 図 4.9 に示す。各グラフの横軸はサンプリング回数である。

上段のグラフの縦軸は、真の識別境界による格子点の青赤の判定とその時点での識別境界の推定モデルによる判定の差で、これをエラー数と呼ぶことにする。今回の探索例ではエラー数は小さいほうが良いことになる。各グラフには前述の 3 つのサンプリング点選択最適化式 (i)(ii)(iii) それぞれについてのエラー数をプロットしてある

下段のグラフの縦軸は少し解説が必要である。サポートベクターマシンで描いた識別境界のマージン  $|f| = 1$  により囲まれた面積の大きさは、探索が進むに連れ小さくなってほしい。また単に小さくなるだけでなく、探索済の格子点付近で小さくくなってほしい。マージン内領域が小さくなるのは、探索が真の識別境界付近に集中していることの証左である。逆に言えば探索が済んでいない領域で小さくなくても必然性・蓋然性がない。そこで、識別境界のマージン内の

未探索の格子点数を数えることで探索の進行具合を代表する特徴量とし、これを縦軸にプロットした。

これらの探索結果から分かる通り、(i)(iii) と (ii) を比較すると「情報が得られないリスク量」の項の有無で探索結果は大きく変わる。探索速度が改善することがわかる。(i) と (iii) を比較すると「得られる情報量」は探索速度の改善に若干ではあるが寄与することがわかる。ただし (i) と (iii) のいずれが優れているか、なぜ優れているかは、今後、統計学上の理論解析をした上で実際の探索例における探索推移の比較し考察を加えることにしたい。

関連研究として、松原ら [86] は次のサンプリング点の最適化指標として以下のような算式を用いている。目的関数はロボットの前に存在する物体の未知の輪郭である。ロボットで物体にタッチすることで物体の輪郭が存在する位置をサンプリングする。このとき最適化指標は概略としては「輪郭の存在の不確かさ、すなわちそこをサンプリングして改善されるであろう情報量の大きさ」と「そこをサンプリングするためのコスト=タッチ動作のロボットのトラベル量」の差で計算している。ロボットに取り付けられたタッチセンサが反応した座標値が取得できるため、狙ったサンプリング点以外の情報が得られることは、本研究とは問題設定がやや異なる。

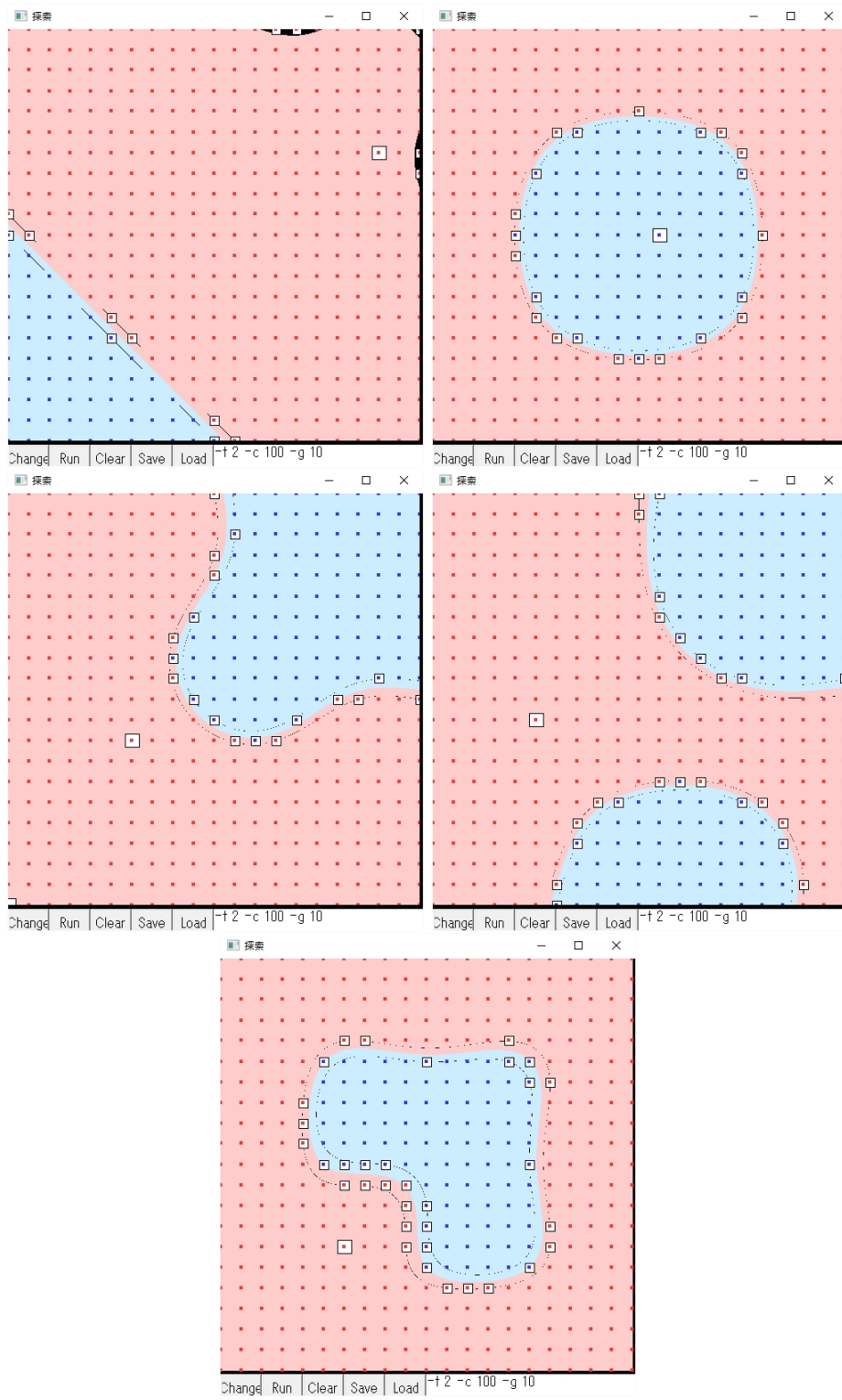


图 4.4 探索結果 (左上:線形分離, 右上:单円, 左中:複円, 右中:離れ円, 下:L字形)

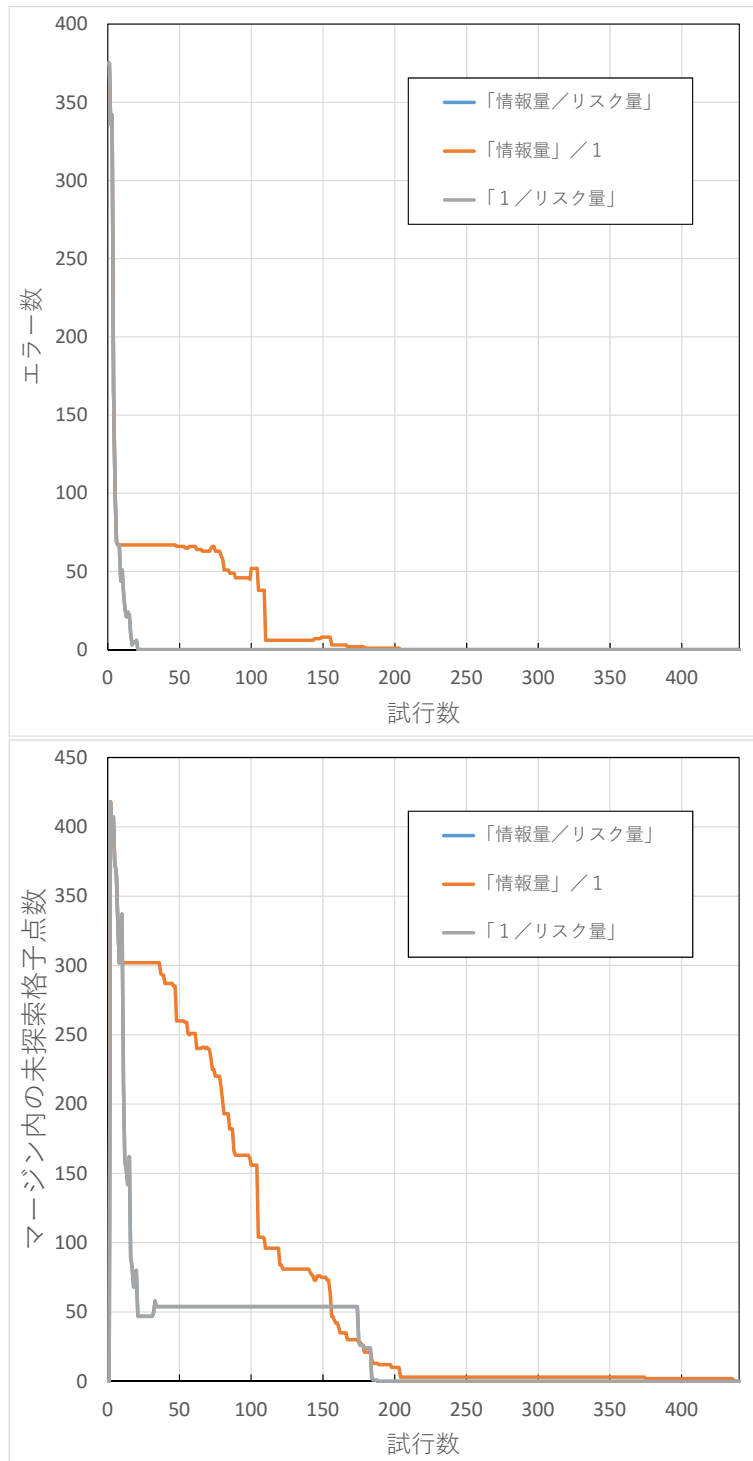


図 4.5 線形分離（上：格子点の判定エラー数の推移，下：マージン内 ( $|f| < 1$ ) の未探索格子点数の推移）．情報量／1 以外のプロットは重なっている．

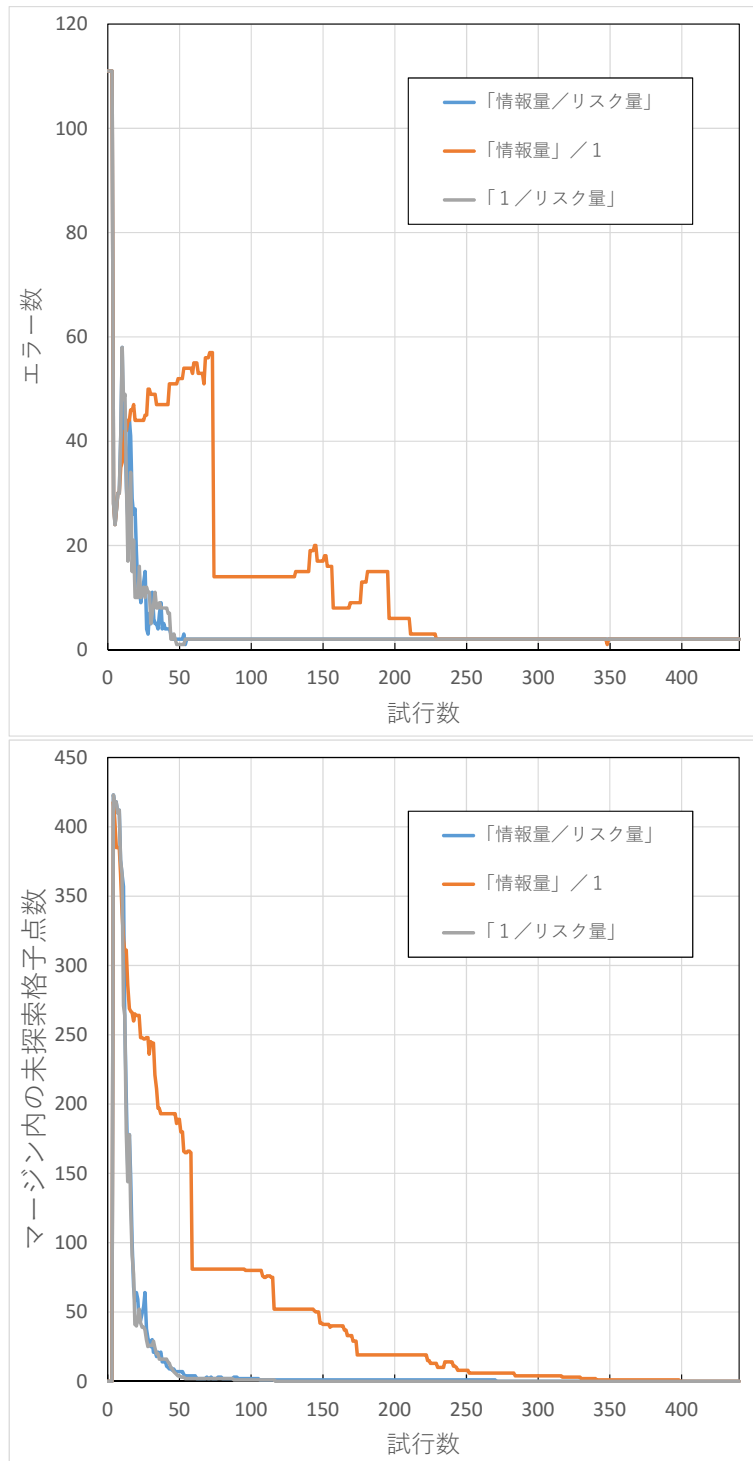


図 4.6 単円（上：格子点の判定エラー数の推移，下：マージン内 ( $|f| < 1$ ) の未探索格子点数の推移）

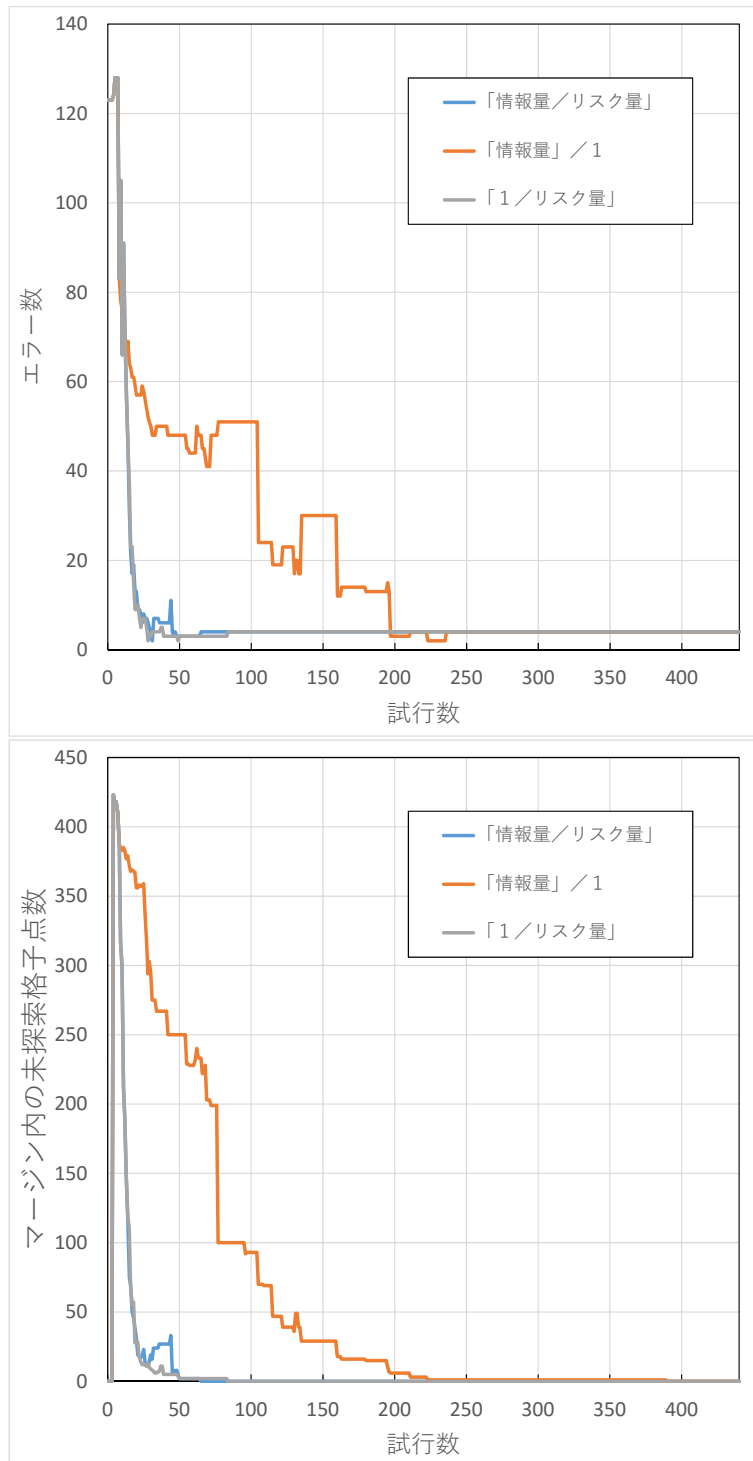


図 4.7 複円（上：格子点の判定エラー数の推移，下：マージン内 ( $|f| < 1$ ) の未探索格子点数の推移）



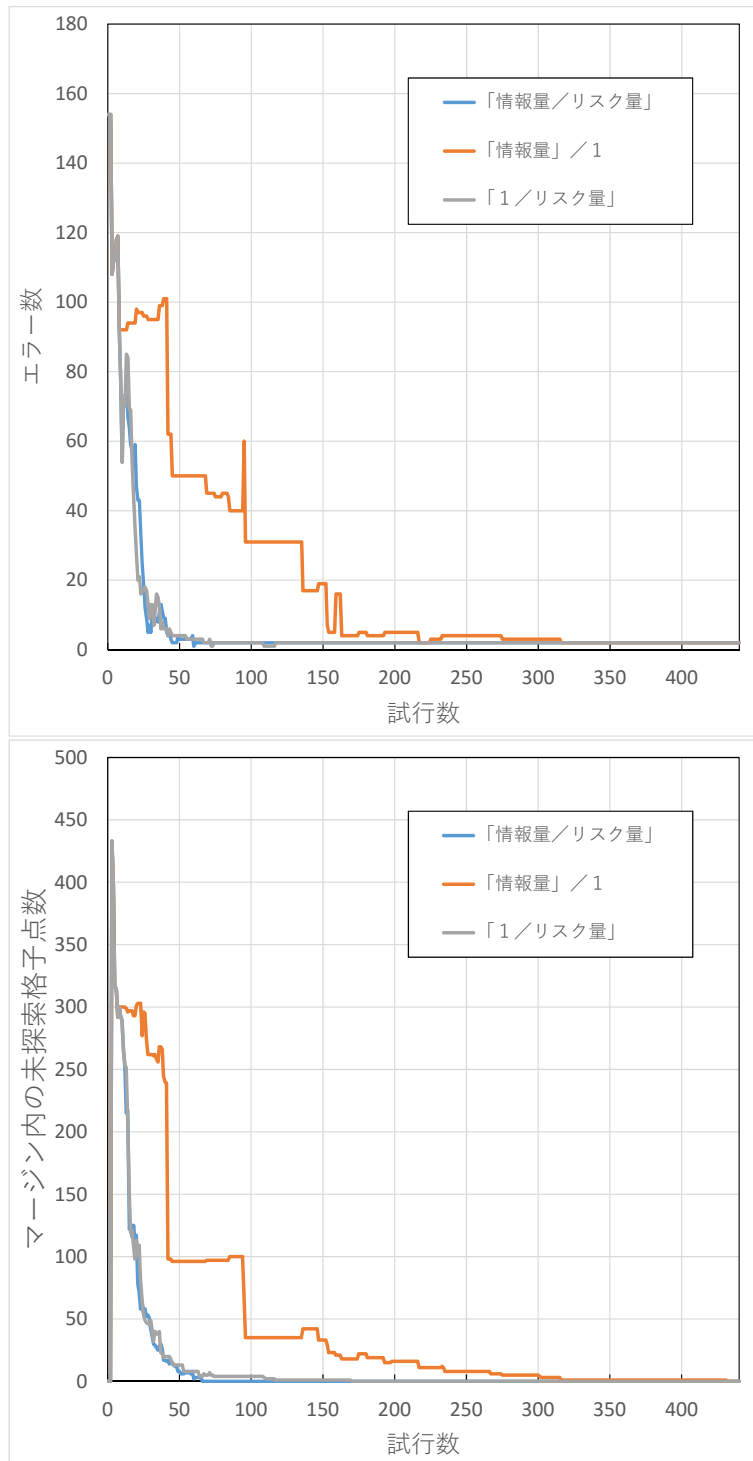


図 4.8 離れ円（上：格子点の判定エラー数の推移，下：マージン内 ( $|f| < 1$ ) の未探索格子点数の推移）

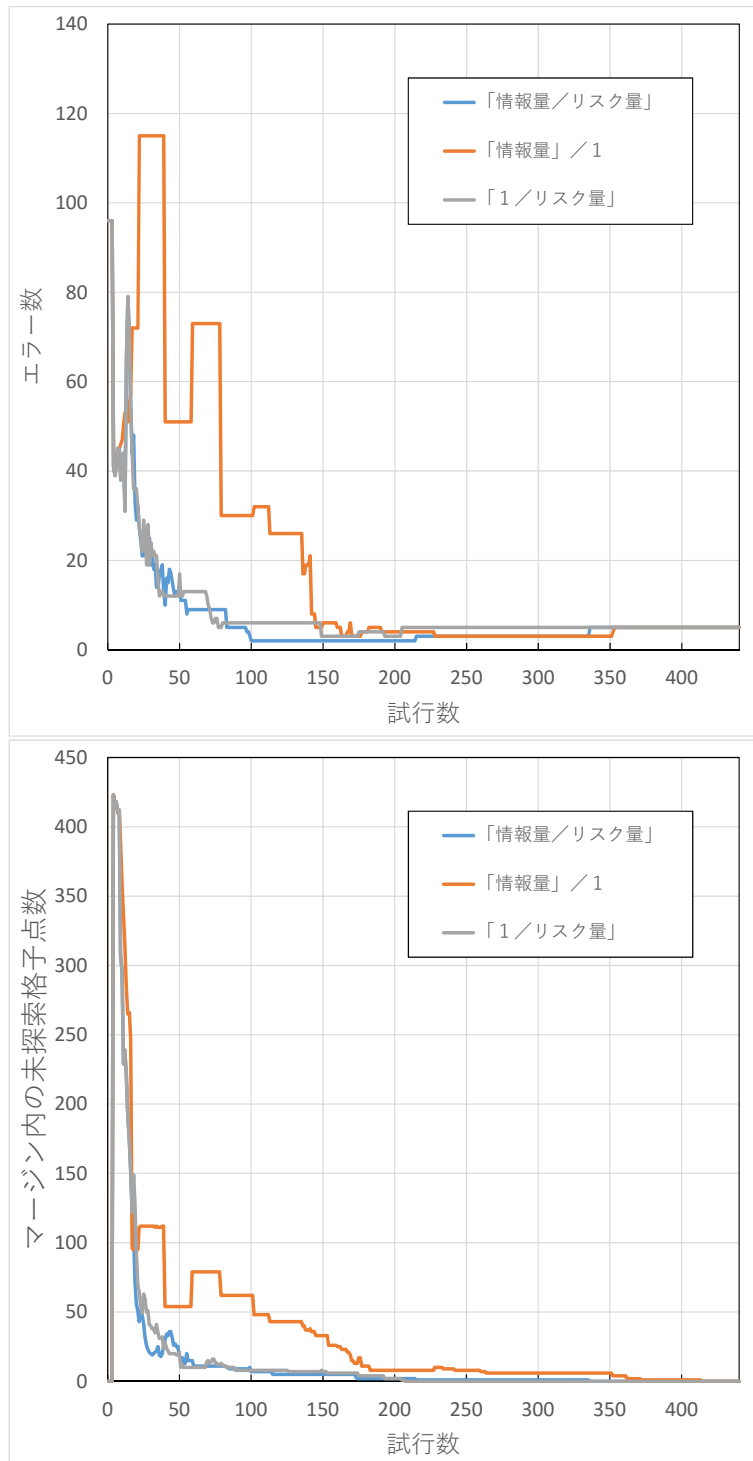


図 4.9 L字型（上：格子点の判定エラー数の推移，下：マージン内 ( $|f| < 1$ ) の未探索格子点数の推移）

### 3.4 能動型探索アルゴリズムの実装について

本アルゴリズムを計算機に実装する際の論点は、六つある。

一つめは前述の目的関数および制約条件の推定モデルである。これは一般的な回帰アルゴリズムで実現できる。近年、非線形な入出力特性の回帰アルゴリズムとして、カーネル関数を用いたサポートベクターマシン [80] が広く応用されている。特に複雑な境界を描く  $f$  にサポートベクターマシンを用いることは自然である。サポートベクターマシンを実装したフリーソフトはいくつか存在し、著者らの実験システム初期版は [87] を用いて実装していた。一方、比較的なだらかな  $F$  には重回帰分析を用いると良い。

二つめはカルバック・ライブラー情報量で定義した2つの識別境界の距離をどうするかである。最も簡単な実装は、パラメータ空間内に格子点を配置し、2つの識別境界に挟まれる格子点の数を数えることである。

三つめは、期待される情報量の算式である。通常は、先に示したものをそのまま用いればよいが、

$$\frac{A_{fg} + A_{fh}}{|A_{fg} - A_{fh}| + 1.0} F(\mathbf{S}_n, \mathbf{x}_{n+1}) \quad (4.3)$$

のごとく、目的関数の推定値を加味するなど、問題の性質に合わせてカスタマイズすることもできる。

四つめは初期探索をどうするかで、これはランダムに数点選ぶか、直交表を用いるのが簡単である。この初期探索点の選び方にはひとつ条件があって、サンプリング後に正常域および異常域に属する標本、属すると推定される標本が、それぞれの域に少なくともひとつ含まれることが必要である。さもないと識別境界を描くことができない。そこで、この状態になるまで初期探索をランダムに継続するなどの工夫をすれば良い。

五つめは探索の終了方法である。本論文が取り扱うシステムは、その入出力特性のランドスケープが不明であることから、全探索が完了しないと真の最適点に到達したことが判明しない。この問題に対し、作業者は、あらかじめ、これぐらいの性能が出ればOKとするという作業目標値を立てて調整作業を開始して、そこに到達すれば作業を打ち切ることにしている。本アルゴリズムを適用する場合も同様に作業目標値を数値入力しておき、目標到達時に打ち切るのが実装方法の

一つである。

六つめは、図 4.3 の「候補点の発生 (Sampling Candidate Generation)」の方法である。上記 3 つめで述べた情報量算式の値が大きい候補点は、実は識別境界付近の候補点である。識別境界から離れた候補点では、情報量算式の分母が大きくなり計算値が下がるためである。制約条件の推定モデルにサポートベクターマシンを用いると、制約条件の推定モデルの識別境界からの距離がサポートベクターマシンでいうところのマージンとして得られるので、識別境界から一定範囲に入る候補点だけを取り出すことが容易である。さらに識別境界  $f$  付近の候補点集合を選び出し、それを  $F$  の推定値でソートして最大値を持つ候補点を選ぶこともできる。このように有望な候補点集合のみに絞って情報量算式を計算することは計算量削減に寄与する。

その他、実装上のノウハウとしては、目的関数  $F$  に複数の最適化指標の合成関数を与えたり、制約条件の識別境界  $f$  にサンプリング結果の実行可能・不能判定だけでなくサンプリング中の不安定現象の発生を実行不能と割り当てるなど、適用対象にあわせて様々に工夫をすることで、問題に合わせて多目的最適化の実装が得られる。また、実験に用いた実装ではサポートベクターマシンはソフトマージン SVM とし、ペナルティ  $C$  を 2000 とした。カーネルはラジアルベース関数  $K(x_i; x_j) = e^{-\gamma \|x_i - x_j\|^2}$  を用い、その  $\gamma$  について標本を識別しきれない場合に動的に 2 倍にする実装とし初期値は 0.01 とした。

### 3.5 能動型探索アルゴリズムの様々な適用事例について

本節で、能動型探索アルゴリズムを適用した事例を簡単な説明とともに列挙し、次節以降に産業用ロボットの動作速度、金属加工機の加工速度、非線形性の強いメカの位置決め制御装置の制御パラメータ調整問題について、後章で詳述する。

一つめは製造業向けの産業用ロボットの動作速度を改善する動作習熟への適用で、本章 4 節で、詳しく述べる。これは熟練作業者が実施している産業ロボットの教示作業に関するものである。ロボットの動作軌道上の経由点を制御パラメータとみなして、 $F$  にロボット動作時間を、 $f$  に障害物との干渉など動作困難な状態の有無を割り当てた。これにより、熟練者が調整した経由点による動作時間 1045.33[ms]

が 590.22[ms] に、つまり約 56%に短縮される経路点が発見された。熟練者はロボット姿勢の 3次元空間のなかで経路点を設定するのに対し、本アルゴリズムは各関節の動作速度の違いを考慮した多次元空間 (6軸ロボットなら六次元) のなかで最適解を探索することでこのような差異が生まれると考えている [88,89]。

二つめは、金属加工機の加工速度に関する制御パラメータの調整作業の自動化 [90,91] である。これは加工制御のパラメータを、加工不能になる組み合わせを避けながら加工速度を最大化するというもので、従来は熟練作業者の手に委ねられてきた。 $F$  に加工速度を、 $f$  に加工不能域を割り当てたところ、熟練作業が自動化・ノウハウレス化し工期が短縮化・安定化する効果を得た。本章 5 節で詳しく述べる。

三つめは、非線形性の強いメカの位置決め制御装置の制御パラメータ調整問題である。これは産業機械でよく使用される位置決め装置のいわゆるサーボパラメータの調整作業を自動化しようというもので、整定時間、オーバーシュート量、整定ずれ量許容範囲内といった相反する目標値の多目的最適化や外乱に対するロバスト化など、現場で要求される実戦問題に適用可能性を検討したところ、ある条件のものとして良好な結果が得られた。本章 6 節で詳しく述べる。

四つめは、一つ目とおなじ金属加工機の加工精度に関する制御パラメータの調整作業の自動化 [92] である。これは加工断面形状の歪みが小さくなる制御パラメータを探索するものである。この問題では、加工速度を  $F$  に、加工断面形状の特微量の合成関数を  $f$  に割り当てたところ、加工断面形状の歪が小さく高速に加工できる制御パラメータの調整作業が自動化され、熟練作業者が約 10 時間を要していた工程について約 5 時間で熟練作業者が満足する結果が得られた。

五つめは、ロボットハンドの把持動作の制御パラメータの調整作業 [47] である。把持動作の開始から完了までの時間を  $F$  に、把持の成否を  $f$  を割り当てた。この調整作業には熟練者がいないため、本アルゴリズムによる自動調整は [47] に示された新しいハンド設計理論の実用化に寄与すると考えている。

六つめは、ある工業プロセスの制御パラメータの調整作業 [93] であり、プロセスの進行速度を  $F$  に、プロセスの安定度合いを  $f$  に割り当てた。上述の例との違いは、プロセスの安定度合いの判定に誤りが含まれていることである。標本集合

の安定度合いの1つが誤りであると仮定して判定結果を反転させて回帰した  $f$  をサンプリング数だけ作成し、それらの積集合を確からしい安定領域として、これらをよりどころに探索し最終結果の選び出しを行った。これは、本アルゴリズムを、問題ごとにアレンジして適用できることを示す事例でもある。結果、20回の試行で熟練者が満足する制御パラメータを得るシステムが構築できた。

その他、移動ロボットの移動経路の設計問題に対する適用事例、トイプロブレムを用いた本アルゴリズムの性質の調査事例などがあり、これについては別報したい。

## 4. ロボットマニピュレータの動作習熟への適用 [88,89]

### 4.1 ロボットの教示作業の現状

組立作業に供する産業用ロボットのプログラミング=教示は、いまだティーチング・プレイバック方式を基本とする。つまり、ごく一部のセンサフィードバック制御の活用を除き、その動きの大半は、作業者により、あらかじめプログラミングされたものである。プログラムデータは作業のシーケンス（手続型のロボット言語）と具体的な動作軌道の座標値（いくつかの座標系のもとで与えられるベクトル値）からなり、いずれもロボットの据付調整時および機種切り替え時に、作業者が長い時間と手間を掛けて丹念に与えており、これをロボットの教示作業と呼んでいる。具体的な座標値の決定はロボットとその動作環境の持つ不確定性および予測困難性を現物合わせで解決しようというもので、作業者は生産設備を前にしてティーチングペンダントあるいはティーチングボックスと呼ばれるリモコン操作箱を手にとってロボットを手動操作し、ロボットの手先が目的の位置・姿勢をとるように、あるいは目的の軌道を描くよう動作軌道の始点、終点、複数の経由点を設定していく。同時に、ロボットと環境の干渉回避と動作タクトタイムの短縮が考慮される。作業者は、あたかもゲームセンタのぬいぐるみ獲りゲームをプレイするがごとく、ロボットと環境の位置関係を、正面から、横から、目視確認しながら、ロボットの位置・姿勢決めを行っている。

## 4.2 動作軌道上の経路点決定の困難さ

動作軌道の始点と終点は、ロボットが組立作業を実行する場合、環境と物理的に接触する必要性から、相対位置関係により決定されてしまい、作業者は迷うことなく製造誤差等の現物あわせに専念することになる。この作業の支援については、別の研究で取り組んでいる [58–61]。

本章での問題は、動作軌道上に存在する障害との干渉回避を目的とする経路点の選択方法である。ここでいう障害との干渉とは、物理的な障害物、把持物体との干渉、測定中のビジョンセンサの視野、その他センサの測定妨害などを指している。空間内の任意の2点を滑らかにつなぐ動作軌道の決定は二点境界値問題に帰着され解析的に解くことが難しく、現状は上記の干渉回避を含め、作業者の手により成り行きで軌道の設計がなされている。しかし、作業者は、目視で手先の通過する位置を設計基準にしている。残念ながら、手先の移動距離が短いことや障害物との距離が近いことは、必ずしも動作時間の短いこととは一致しない。前者は3次元空間内での最適性であり、後者はロボットの各関節ごとの最高速度や姿勢に依存して変化する慣性モーメントなどのロボットの身体性でスケールされる六次元以上の多次元空間内の最適性が求められるからである。つまり、この問題は解析的に解くことが困難で、作業者が実験して制御結果を確かめながら最適化するしかないシステムの一つであり、これを自動的に最適するためには、制御パラメータを自律的に探索するアルゴリズムが有効である。

## 4.3 能動型探索アルゴリズムの適用

能動型探索アルゴリズムを用いて、図 4.10 に示す例題を解く。図中の赤丸が軌道の始点、青丸が終点であり、ロボットは始点から終点に移動する。このとき、図 4.11 のごとく、ロボットの前に存在する物理的な障害物を避けるため、黄丸で示す経路点を設ける。始点と終点は固定で、経路点は干渉を避けられるという前提で可変である。ロボットは市販の製造業向けの産業用ロボットであり、6[Kg]の可搬質量を持つ標準的な機種である。

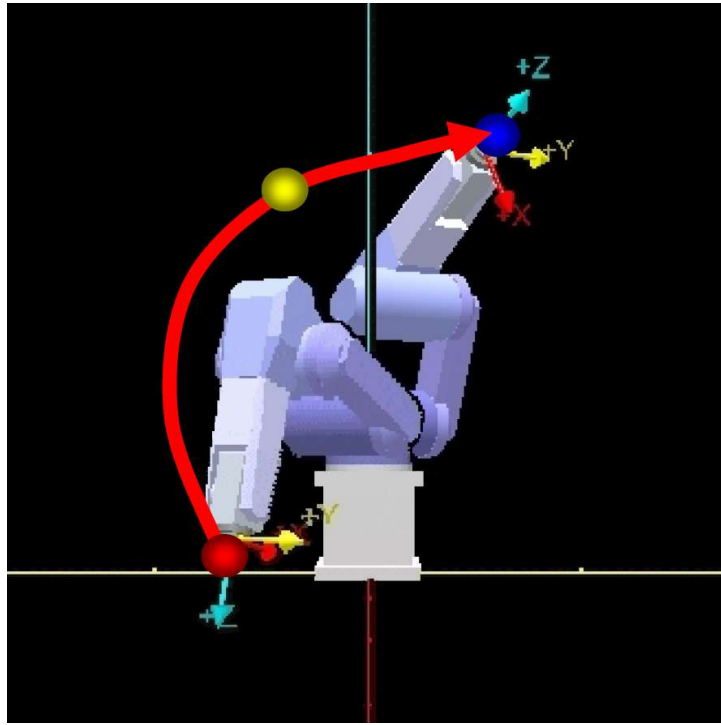


図 4.10 動作習熟させる軌道の概要

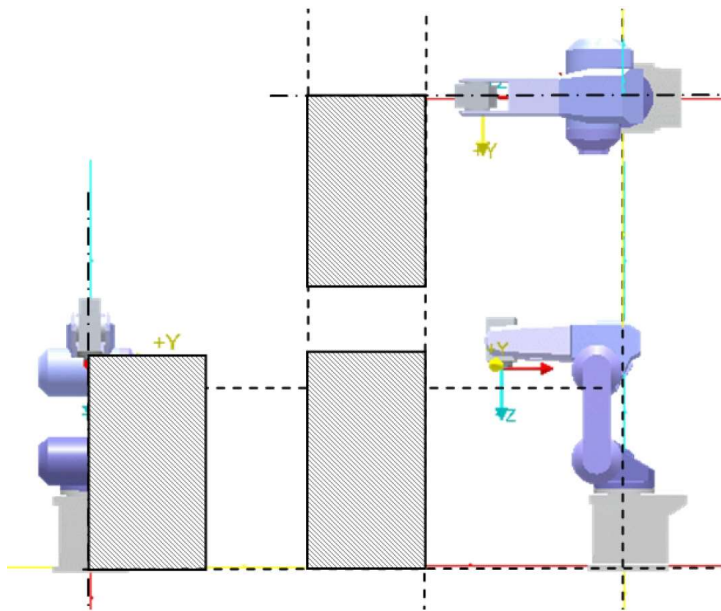


図 4.11 ロボットが避ける障害物



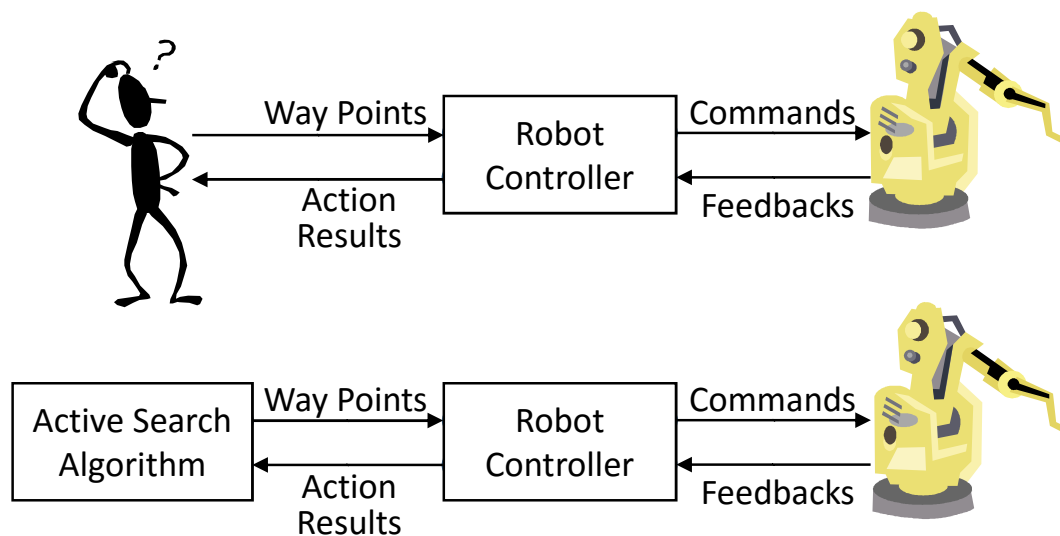


図 4.12 実験システムの概要

この条件の下，経由点を経つつ，始点から終点に至るまでの移動時間を最小化したい．このとき，能動型探索アルゴリズムにおける  $F(\mathbf{x})$  にロボットの移動時間を， $f(\mathbf{x}) > 0$  に障害物と干渉しないことを割り当てる． $\mathbf{x}$  は経由点のロボット各軸の関節角度とした．

能動型探索アルゴリズムはパーソナルコンピュータ上のソフトウェアで実装した．事故防止のため障害物はロボットに付随するロボットコントローラ内に仮想的に与え，ロボットコントローラ内で干渉・非干渉の判断を行った．能動型探索アルゴリズムからロボットコントローラへ経由点座標を送り，ロボット動作起動コマンドと送ると，ロボットは所定の動作を行ってホームポジションへリターンする．そののち，ロボットコントローラから始点終点間の動作時間と，仮想物体との干渉の有無がパーソナルコンピュータに返される．パーソナルコンピュータとロボットコントローラは，イーサネット<sup>®</sup> 上の TCP/IP 通信を用いて通信を行っている．図 4.12 に実験システムの概要を示す．

探索空間は関節角座標系で与え，3次元で9水準×17水準×25水準，総組み合わせ数3825通りの組み合わせの空間の中から，動作時間の短い経由点を探索する．初期実験点は直交表L27を用いて与えた．

探索を実施したところ，初期27点後の，46手目で全空間内の最適点に至るこ

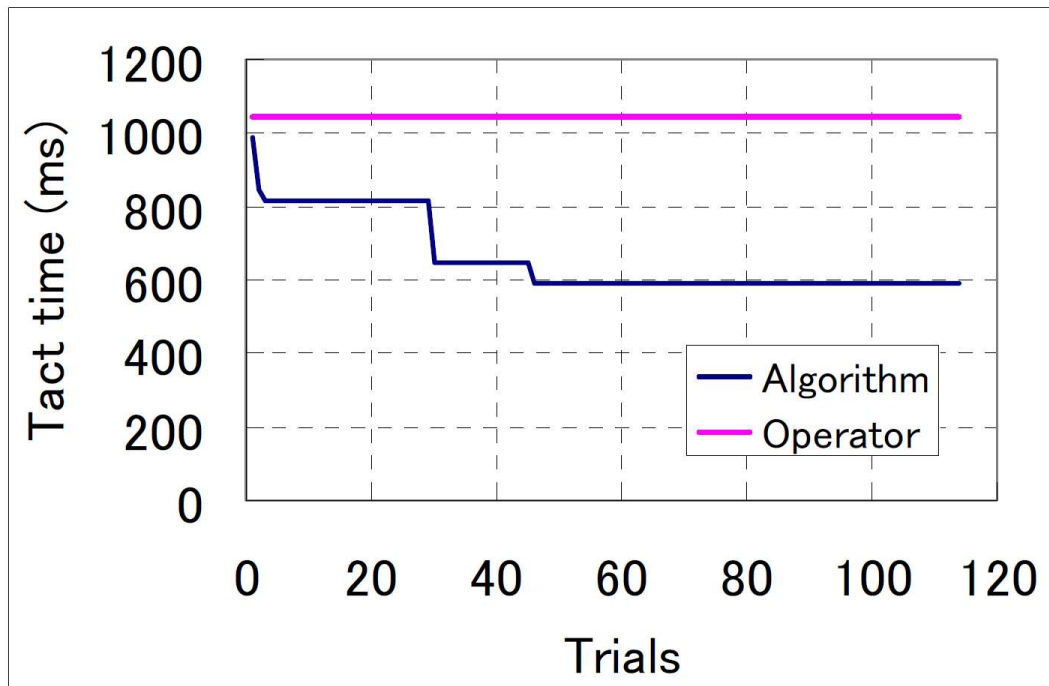


図 4.13 動作習熟の進行

とがわかった。念のため、3825 通りの全空間内の探索を行って、真の最適点に到達していることを確認している。探索の進行を図 4.13 に示す。実現された動作時間の習熟の効果は、作業者初期値 1045.33ms（赤い線）に対し、590.22ms(青い線)であり、実行時間が約 56%に短縮された。

図 4.14 は、動作習熟前後におけるロボット軌道の変化を示す。習熟前は、障害物をかわせるところまで腕を振り上げきってから水平旋回を強める軌道になっている。それに対し習熟後は腕を振り上げ終わる前に水平旋回を強めていることがわかる。習熟後の軌道のほうが、ロボットにとって高速動作が可能な動きであるということになる。

探索途中の経路点の変化を図 4.15 に示す。本アルゴリズムは広範囲に探索を進めていくことがわかる。

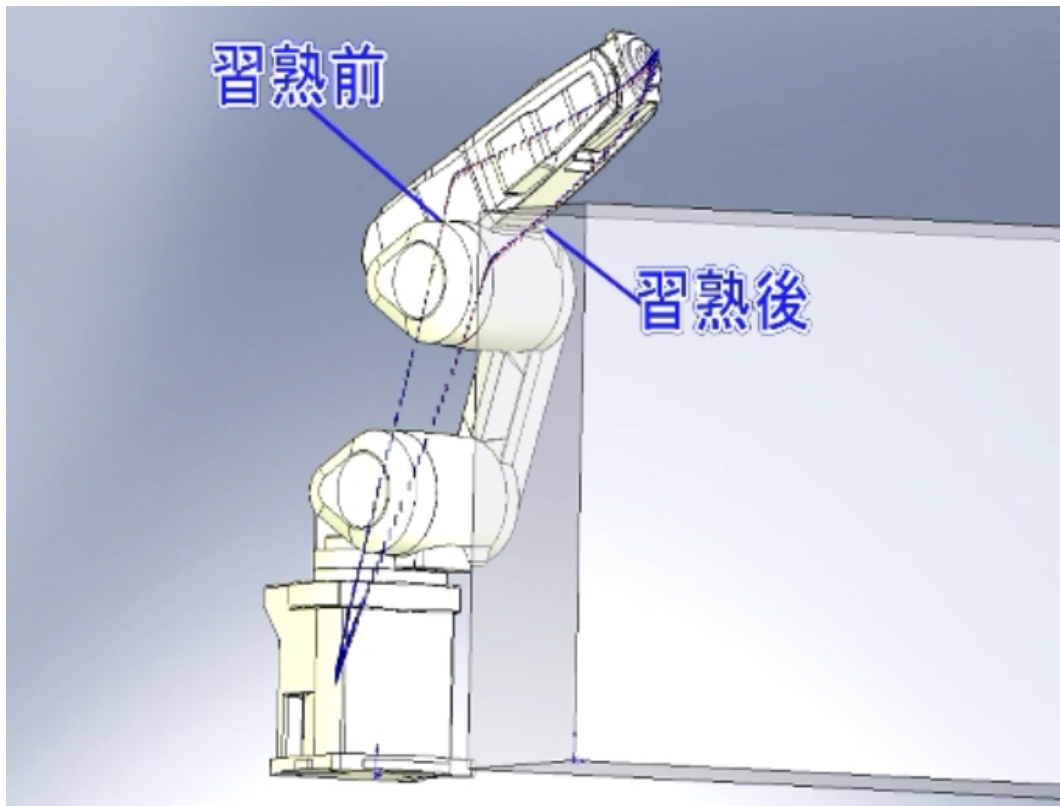


図 4.14 動作習熟前後のロボット軌道

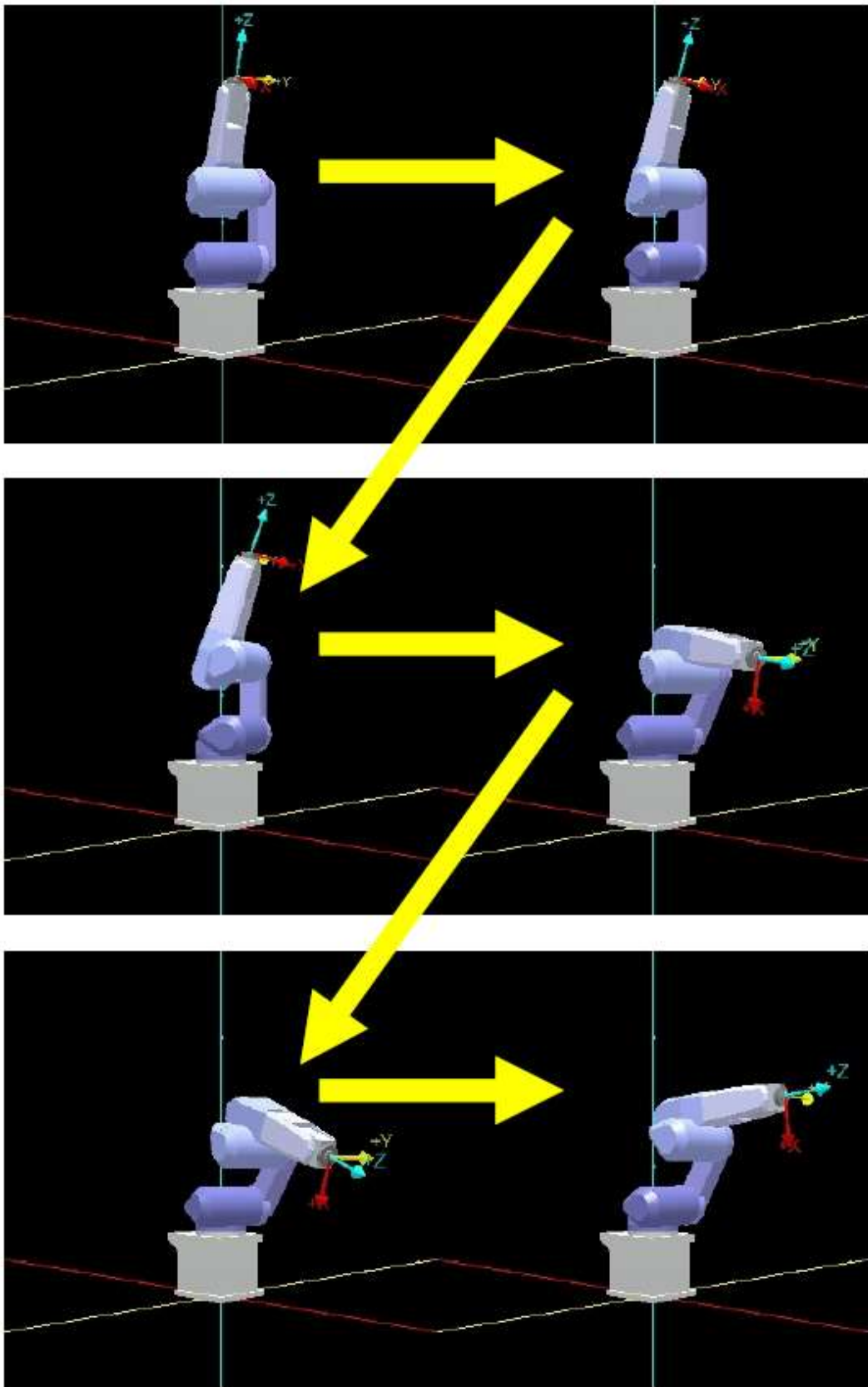


図 4.15 探索中の経路点の変化

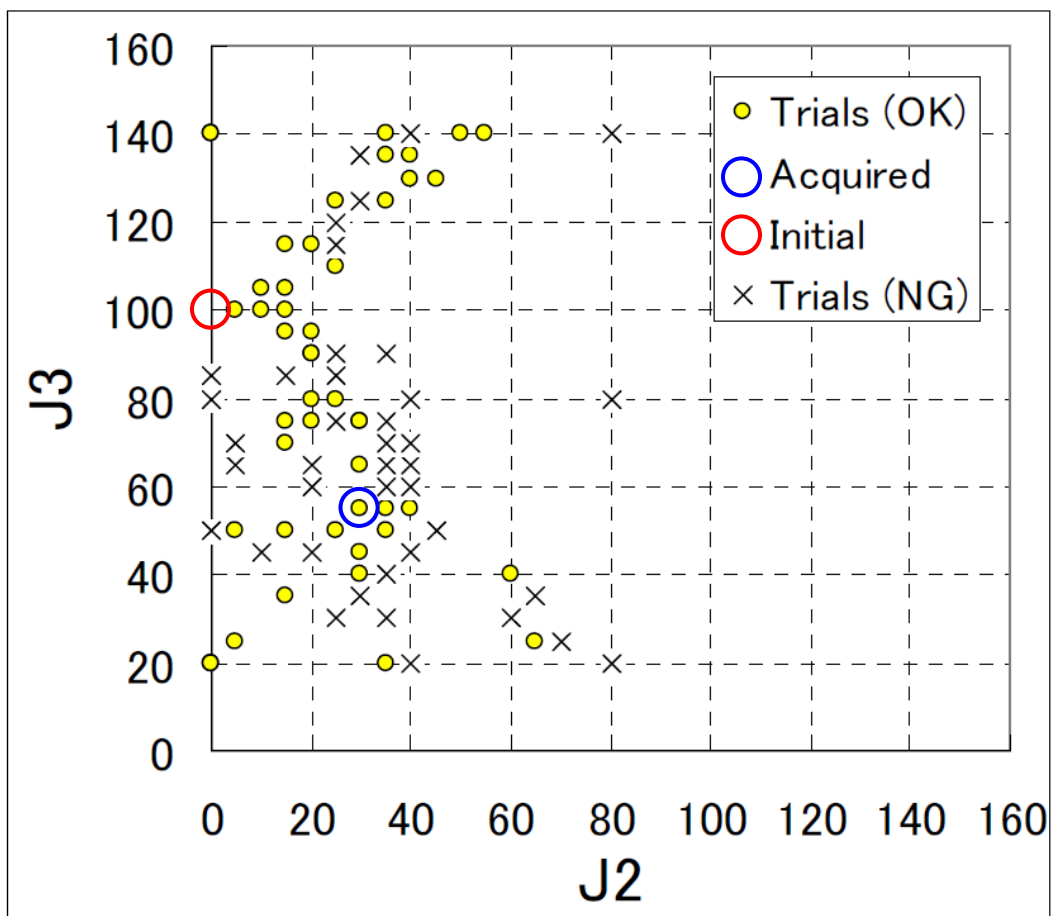


図 4.16 探索した空間の様子（2次元表示）

図 4.16 に、より具体的な探索の様子を示す。これは、3次元空間内の各探索点を2次元平面上にプロットしたもので、赤丸が習熟前の経由点、青丸が習熟後の最適な経由点、黄丸が干渉の発生しなかった経由点、×印は、干渉の発生した経由点を、それぞれ示しており、複雑な非線形な現象の中を、探索が進んでいくことがわかる。

また、同じ空間を、3次元表示したものを図 4.17 に示す。図中の球体が標本であり、添えた数字は何手目かを示している。

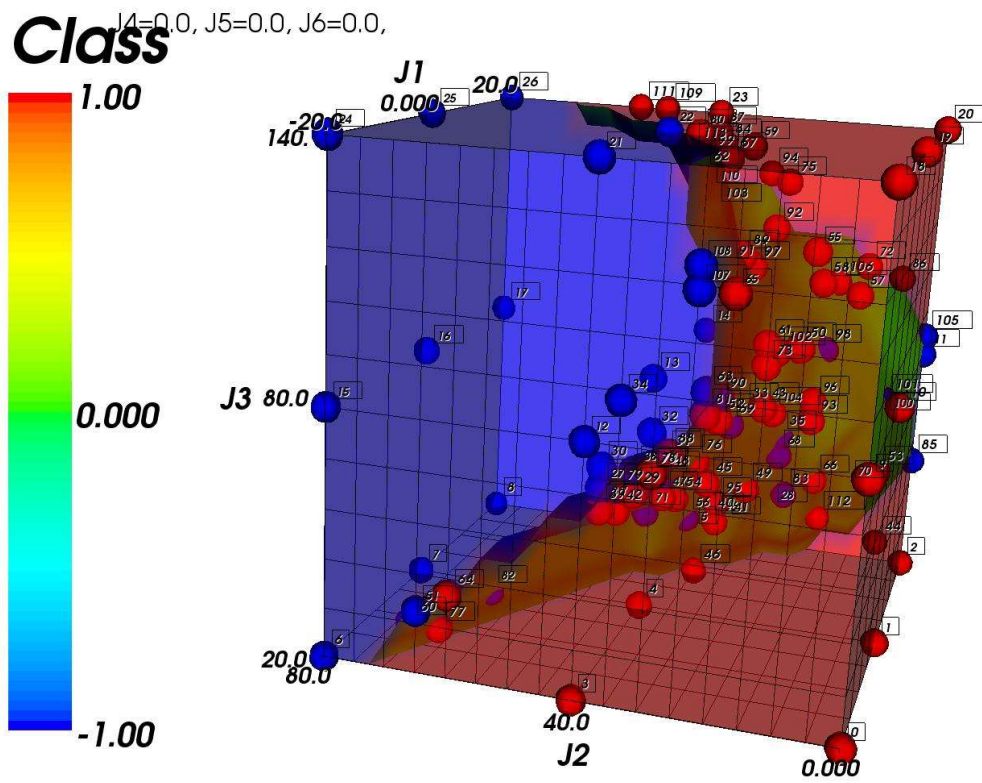


図 4.17 探索した空間の様子 (2次元表示)

## 5. ある金属加工機の加工速度最大化への適用

### 5.1 課題と適用結果

本節で取り上げる加工機は金型の加工等に用いられており、投入する加工エネルギーの増減(パルス電流波形)、加工軸送り、加工液流量といった制御パラメータが17種類存在する。加工対象物と工具について寸法をそれぞれ28通りと7通りに量子化した上で材質6種類づつを組み合わせたもの各々について制御パラメータを調整して、好ましい加工結果が得られる制御パラメータ組み合わせを発見する作業が必要となる。

このような制御パラメータ調整作業は熟練作業者の手に委ねられている。加工速度を目的関数  $F(\mathbf{x})$  とし、加工可能領域を制約条件  $f(\mathbf{x}) \geq 0$  として、最適な制御パラメータ  $\mathbf{x}^*$  を求める。

図4.18の上半分はこの様子を示す。図中、熟練作業者は加工機(Processing Unit)に加工対象物(Work Pieces)を取り付けたのち、加工制御装置(Machining Controller)に初期の制御パラメータ(Parameters)を打ち込んで加工を開始する。加工後、その加工結果を観察し加工結果が好ましいものになるように制御パラメータを変更して、再び加工を実施することを繰り返している。

これに対し、能動型探索アルゴリズムを適用する場合を図4.18の下半分に示す。能動型探索アルゴリズムを用いて、熟練作業者が行っている、初期パラメータ投入、加工結果を見て制御パラメータの加工制御装置に投入する動作を代替する。

探索範囲については、熟練者が通常作業で調べている範囲内を本アルゴリズムを用いて探索することになっている。

加工開始・停止、加工結果取得といった実験手順のバッチ処理プログラム、および能動型探索アルゴリズムは、Windows OSのパーソナルコンピュータ上のソフトウェアにて実装し、加工制御装置とはイーサネット®上のTCP/IP通信によるローカルエリアネットワークを介して加工制御装置のリモートコントロールポートとデータ回線を接続した。このデータ回線を通じて、制御パラメータ変更コマンド、加工結果取得コマンドとそのレスポンスにより、加工機運転操作に関して熟練作業操作と同等の操作を自動化し、制御パラメータ調整に関して熟練作業

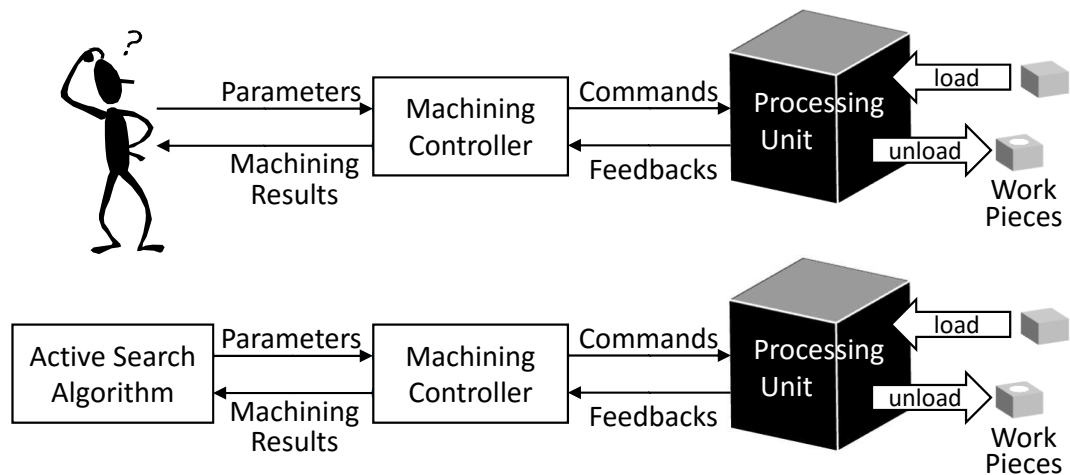


図 4.18 実験の設定

者と能動型探索アルゴリズムを比較できるようにした。なお、この場合でも作業対象物の取り付け取り外しは作業者が行っている(相応のコストをかければ取り付け取り外し作業の自動化装置は製作可能である)。

本実験の前提として、加工制御装置の中身、すなわち加工プロセスの制御器と加工機可動部の位置決め制御器には今回は手をつけていない。もちろん、これら制御器自体の選択を間違えている可能性は指摘できる。しかしながら本アルゴリズムによる調整が必要な対象は、そもそも制御対象の特性が基礎方程式等でモデリングできず数学的な最適設計手法が適用しにくく、実験した範囲内で良い物を選び出すしかない対象を想定していて、制御器の選択、制御器と制御対象のマッチングについては、その時点でのベストエフォートで適合しているものと判断していること、本論文で提案するアルゴリズムは制御器のパラメータを操作するものであり、人手で調整するパラメータを持つ制御器であれば制御器自体の形式には依存しないものであることを付記する。本論文は、この与えられた制御システムについての調整作業を自動化、ノウハウレス化するのが狙いである。現時点でも、および今後も制御パラメータが理論的な設計で一意に決められず作業による現物合わせのパラメータ調整箇所があるかぎり、適用対象の理論的な解析・設計手法と本件手法は並立しようと考えている。

本件事例における  $f$  の設計は次のようにした。この加工機では助走加工をした



のち本加工に移る。その途中のいずれかで、加工不能に陥いる／加工が不安定になる／正常に加工完了する／判定に失敗する現象が発生する。そこで、 $f$  の値を次のように定義した上で、 $f \geq 0$  ならば正常域、 $f < 0$  ならば異常域とした。

- $f = 2$ : 助走加工が正常，かつ，本加工が正常
- $f = 1$ : 助走加工が異常，かつ，本加工が正常
- $f = 0$ : 助走加工が判定失敗，かつ，本加工が正常
- $f = -1$ : 本加工が異常
- $f = -10$ : 加工停止

ここで能動型探索アルゴリズムの具体的な実装例のフローチャートを図 4.19 に示す。図の左側が、初期探索を含む能動型探索の全体シーケンスである。図の右側のハッチング部分が次の試行の最適化，すなわち逐次サンプリング点最適化部分のシーケンスである。制御パラメータはその定義域を量子化して組み合わせる。各組み合わせごとに加工結果の  $f$  値，および  $F$  値が試行結果がサンプリングされる。これらは配列  $S$  で管理している。これで、制御パラメータの組み合わせが配列  $S$  のインデックスで指定されるので、ポインタ配列  $I$  にインデックスを格納し実験順序を管理する。直交表による初期実験順序は、実験開始前に  $I$  に代入され、その後は、サンプリングごとに逐次サンプリング点最適化が行われ  $I$  は逐次書き換えられ、次々とサンプリングが繰り返される。

この逐次サンプリング点最適化の実装においてはサポートベクターマシンを用いてすべての標本について  $f$  の回帰モデルを計算している。および加工が正常に完了した標本について重回帰分析を用いて  $F$  の回帰モデルを計算している。そして、まだ試行をしていない格子点 (= 試行の候補点) に対して回帰モデルによる推定値を配列に代入する。次に、まだ試行をしていない格子点について、 $f$  の絶対値で昇順に  $I$  内のインデックスをソートする。その後  $|f| < 1$  となっている最大順序インデックスに目印  $M$  をつける。そしてその目印までの範囲の順序インデックスを  $F$  の推定値で降順にソートする。これで、次に実験すべき候補点が発生されたことになる。

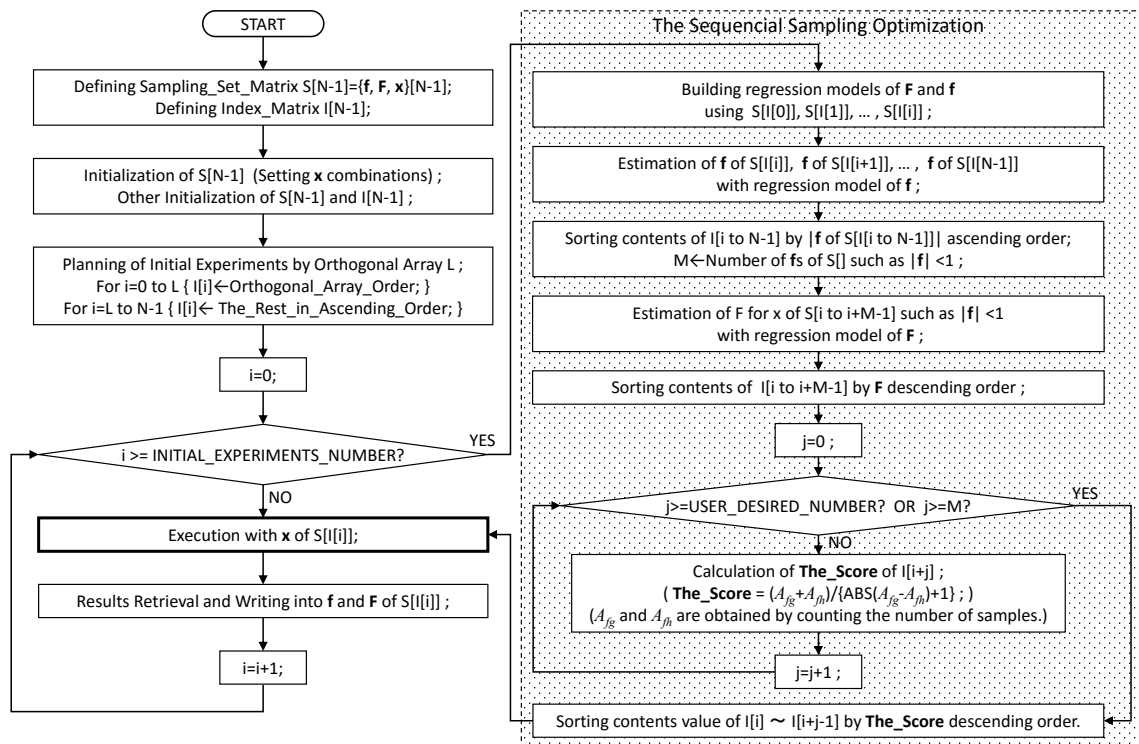


図 4.19 アルゴリズムの実装例のフローチャート

つぎに、発生させた候補点について、スコアを計算する。これは、4.3 式の情報量算式を用いている。スコアは候補点のうち、予め定めた個数だけ、あるいは M 個以下だけ計算することになっている。

最後は、上記で計算したスコアで降順にインデックスをソートし、先頭に来たものを次のサンプリング点としている。その後、実際に試行を行う箇所は、図 4.19 の左側の下方、太い枠で囲ったステップである。

上記のうち、スコアの最大値を求めるのではなく、スコアでサンプリング順序をソートする計算は冗長ではあるが、これを行うことで、複数のサンプリングを一気に行うこともできる。

図 4.18 に示した実験装置と図 4.19 に示したアルゴリズム実装を用いて、全 17 種のうち熟練作業者が通常の調整作業で用いる 7 つのパラメータについて、各 3 水準、組み合わせ総数 2187 点の空間内を探索した。同加工機では、とりわけ制約条件が複雑な形状を描く。そこで目的関数  $F(\mathbf{x})$  は重回帰分析を、制約条件  $f(\mathbf{x}) \geq 0$

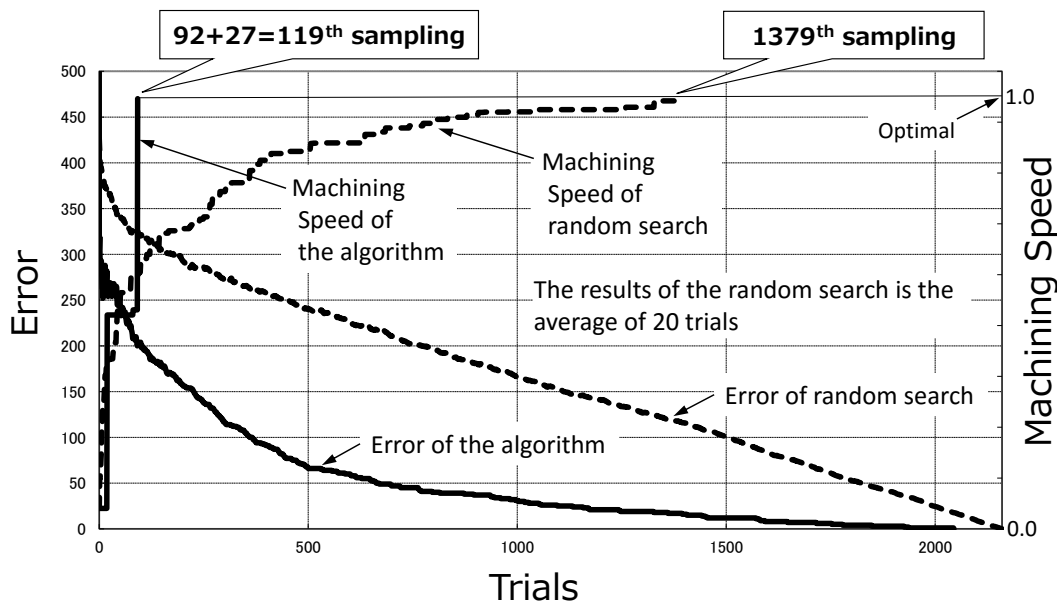


図 4.20 探索の進行例 (1)

はサポートベクターマシンで推定モデルを構築した。識別境界間の距離は内包する格子点を数える方法とし、初期実験点は直交表 L27 を用いて与えた。

この結果を図 4.20 に示す。横軸は探索試行回数で、縦軸は左側が 2187 点の識別真値とその時点の識別境界による格子点の識別エラー数で、下に行くほど小さく、好ましい方向である。右側が加工速度 (最高速度を 1 として  $[0,1]$  に正規化) で、上に行くほど大きく、好ましい方向である。図 4.20 は、本アルゴリズムが能動型探索 92 手+初期探索 27 手の計 119 サンプル目で最適条件を発見していることを示している (Machining Speed of the algorithm の線)。図中の「ランダム探索 (random search)」というのは、一度サンプルした点は記憶しながら次のサンプルをランダムに発生させるアルゴリズムで探索することを 20 回繰り返しその平均値をプロットしたもので、作業者の最悪ケースを模して本アルゴリズムとの比較のために用いており、1379 サンプル目で最適値に到達している。探索が進むにつれ対比アルゴリズムのそれ (Error of random search) よりも高速にエラーが減少し  $f$  が正確になる (Error of the algorithm)。

実際の作業者との比較は後述する。なお、この実験は 3ヶ月をかけて全 2187 通りの加工を実施してデータを取得し、最適組み合わせが判明した状態で計算機シ

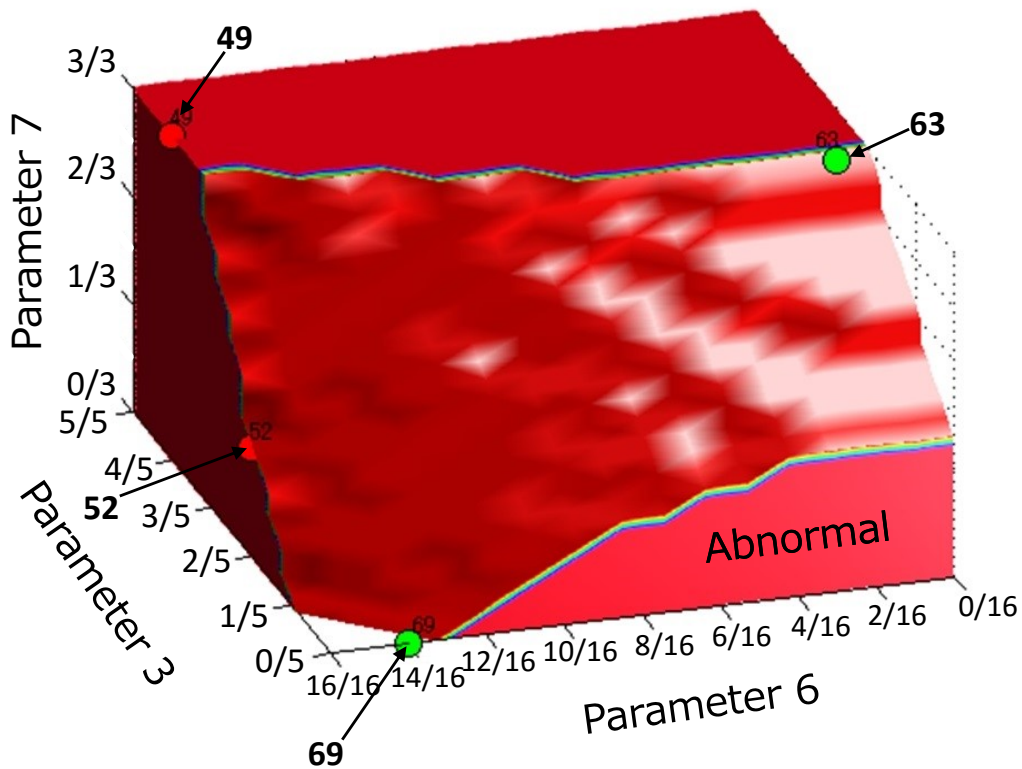


図 4.21 探索の進行例 (2)

ミュレーションで実施した。

さらに図 4.21 には、別の探索例を示す。こちらは7パラメータで、パラメータ 1 を 3 水準、パラメータ 2 を 3 水準、パラメータ 3 を 6 水準、パラメータ 4 を 6 水準、パラメータ 5 を 4 水準、パラメータ 6 を 17 水準、パラメータ 7 を 4 水準とし、組み合わせ総数  $3 \times 3 \times 6 \times 6 \times 4 \times 17 \times 4 = 88128$  点の空間を、実際に加工しながら探索した様子を示している。図は、7つのパラメータのうちの3つのパラメータ軸を用いて識別境界の推定モデルを3次元表示している。

図 4.21 の表示に用いたのは 6 水準のパラメータ 3、17 水準のパラメータ 6、4 水準のパラメータ 7 である。各パラメータの値域は  $[0,1]$  に正規化して表示している。つまり、水準数が 6 のパラメータ 3 の各水準の正規化値は、 $0/5, 1/5, 2/5, 3/5, 4/5, 5/5$  の 6 種類である。水準数が 17 のパラメータ 6 のそれは、 $0/16, 1/16,$

… , 16/16 の 17 種類となる。水準数が 4 のパラメータ 7 のそれは、0/3, 1/3, 2/3, 3/3 の 4 種類となる。非表示のパラメータについては、パラメータ 1 を、0/2, 1/2, 2/2 のうちの 2/2 に、パラメータ 2 を、0/2, 1/2, 2/2 のうちの 2/2 に、パラメータ 4 を、0/5, 1/5, …, 5/5 のうちの 4/5 に、パラメータ 5 を、0/3, 1/3, 2/3, 3/3 のうちの 3/3 に、それぞれ固定した。

図中の異形物体の波打った表面が正常な加工結果となる正常域と異常な加工結果となる異常域の識別境界の推定モデルで、小さな球体は 0 サンプル目から数え始めて 63, 69 サンプル目の正常域の標本、および 49, 52 サンプル目の異常域の標本である。これらのうち 69 が、あらかじめ熟練者が数週間をかけて選んでいたパラメータ組み合わせによる加工速度の 1.31 倍の速度に自動加工時間 5 時間 30 分で到達した最速標本であり、十分な性能改善がなされたと判断し、71 で探索を打ち切った。

図 4.22 は同じ探索結果を、識別境界 (図中 Decision Boundary と表記) の表裏を見渡すように、非表示軸も含む 7 次元空間で視線を変えて表示したものである。非表示のパラメータについては、パラメータ 1 を、0/2, 1/2, 2/2 のうちの 2/2 に、パラメータ 2 を、0/2, 1/2, 2/2 のうちの 2/2 に、パラメータ 4 を、0/5, 1/5, …, 5/5 のうちの 5/5 に、パラメータ 5 を、0/3, 1/3, 2/3, 3/3 のうちの 3/3 に、それぞれ固定してある。標本に添えられた探索順番が示す通り、識別境界の表裏をなぞるように探索が進んでいく様子が見て取れる。

## 5.2 制御パラメータ開発プロセスの変革

制御パラメータ開発プロセスというのは、客先で加工機が使用されるとき、加工対象物の材質、形状寸法が指定されるが、その指定に適した制御パラメータの数表をあらかじめ作成するプロセスのことである。熟練作業者は、加工対象物を変更しながら朝から晩まで何 10 種類もの加工組み合わせについて加工実験を繰り返す。このため本アルゴリズムも同じ回数活用することで自動化の効果が期待される。図 4.18 に示した実験システムを熟練作業者に預けたところ、日々の作業のなかで繰り返し能動型探索が行われ、作業プロセスが変化した。

以下、本アルゴリズム適用による作業プロセスの変化について述べる。熟練作

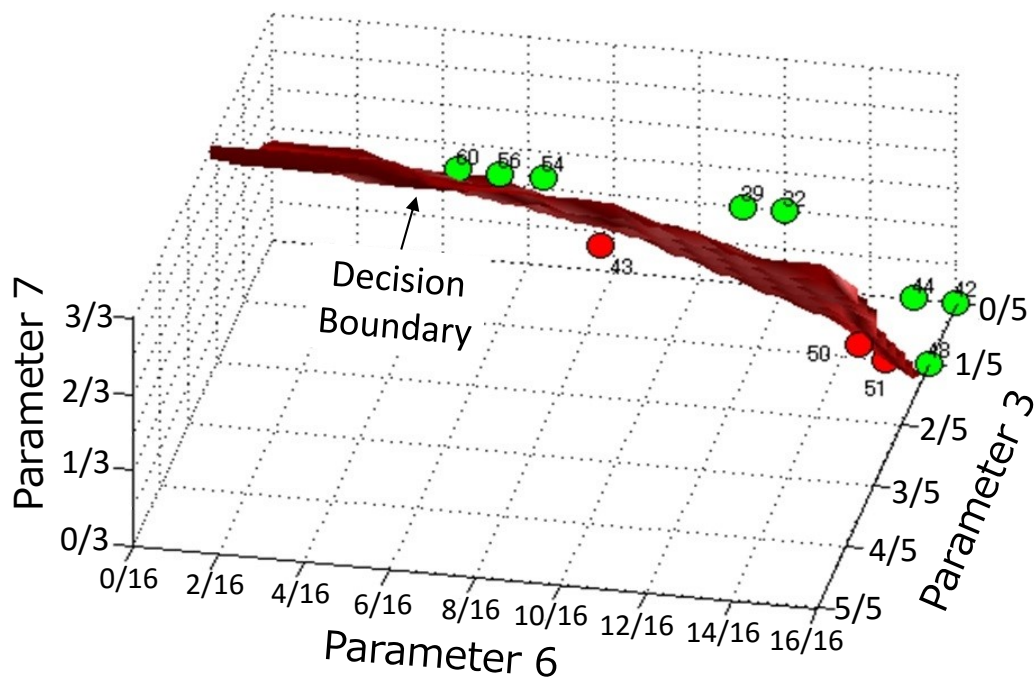


図 4.22 探索の進行例 (3)

業者が手作業する従来のケースと、今回の能動型探索アルゴリズム搭載自動実験システムを用いたケースとで、時間短縮効果を比較する。その際、自動化実験システムによる効果と、能動型探索アルゴリズムによる効果の分離を試みるため、まず、熟練者の作業時間、次に、作業者が指定した制御パラメータで自動加工できる自動実験システムの作業時間、最後に能動型探索アルゴリズム搭載自動実験システムを用いた作業時間を比較し、それぞれの制御パラメータ開発プロセスの変化について考察する。

#### 熟練作業による従来のケース

- プロセス時間 [合計：最長3ヶ月]  
 組み合わせ探し・データ取り (0.5 日) → 失敗 → 組み合わせ探し・データ取り (0.5 日) → 失敗 → … 繰り返し数十回 … → 確認試験 (2 日～3 日) → 検証 (2

日～3日) →完成

- 考察

作業者によって組み合わせの決定までの時間がばらつく (経験と知識が必要). 加工状態を見るためや組み合わせ変更のために機械の前に付きっ切りとなる. ひとつの組み合わせの良し悪しを判断するのに半日を要し, それを数十回繰り返している.

### 熟練作業者の思考+自動実験システムのケース

- プロセス時間 [合計: 1～2週間]

組み合わせ自動実験 (1日) →結果確認・再データ取り (0.5日～3日) →確認試験 (2日～3日) →検証 (2日～3日) →完成

- 考察

加工対象物の取り付け等の実験セットアップ作業は, 人手で実施. 実験をセットアップする作業者は誰でも良く, その後, 無人運転可能. 実験中の加工結果の良し悪しは自動的に記録される.

『組み合わせ探し→データ取り』の繰り返し回数の減少 (数10回→1～5回).

### 能動型探索アルゴリズム+自動実験システムのケース

- プロセス時間 [合計: 約1週間]

自動探索 (2日) →データ取り (2日～3日) →検証 (2日～3日) →完成

- 考察

加工対象物の取り付け等の実験セットアップ作業は人手で実施するも, 開発期間の短縮と安定化および調整のノウハウレス化が実現.

一人の作業者に対し加工機を複数用意することで作業の並列化が容易で外注作業も可能.

実証を担当した熟練作業者らが「夢のよう」「魔法」「自分を見ているようで気持ち悪い」と感嘆.

## 6. サーボパラメータ調整における多目的最適化などへの適用

産業現場で用いられる制御システムには調整すべきパラメータが存在する。制御パラメータの調整コストを下げるため様々なオートチューニング手法が実用化されてきたが、モデル化が困難な制御対象に対して制御性能が厳しく要求される場合には、いわゆる熟練作業者が制御対象の応答を見ながら現物あわせで調整しているのが現状である。

このような調整作業は、やってみるまで結果がわからない。作業者の熟練度などにより作業の結果と時間にムラが発生し、迅速かつ低コストなシステムインテグレーションを阻害する典型的な難題のひとつと捉えられている。その困難さの本質は、未知の目的関数の最適化問題として定式化できるのである。

著者らは機械学習 [74] とりわけ能動学習 [75] のスキームに基づいて、未知の目的関数を最適化する独自の能動型探索アルゴリズムの開発を進めている。同アルゴリズムの提案、パーソナルコンピュータ程度のハードウェア上でソフトウェアとして実装可能であること、ある金属加工機の加工制御パラメータ開発期間を短縮する効果があることを既報した [65]。問題の困難さの本質の解法であるからには同種の困難さをもつ問題への水平展開が可能と考えられる。よって今後、様々な現場の要求を含めた事例への適用性を調べるのが課題となる。

例えば、産業メカトロニクス分野で用いられる位置決めサーボシステムのゲイン調整、すなわち制御パラメータの調整に関して、まさに冒頭で述べた課題が発生している。そこで本稿では本アルゴリズムがこの問題にも適用可能かを検討、同時に多目的最適化、誤差因子に対するロバスト化、計算時間の短縮化といった現場の要求を考慮しうるかを報告する。

6.1 節でサーボパラメータのオートチューニング問題の背景について述べる。

6.2 節では、同アルゴリズムの適用性を検討する。概略は次のとおりである。多目的最適化、および移動ストロークの変化に対するロバスト化を考慮して提案アルゴリズムをソフトウェア実装して適用したところ、目標評価仕様 2 種に対する最適値探索を行う計算機実験において、それぞれ 7 試行目、15 試行目に複数の要



求仕様を満たしながら移動ストロークの変化に対してロバストな最適値に到達することを確認した。

6.3 節では、実際のパラメータ調整の現場では様々な誤差因子に対するロバスト化が必要となる場合が多いので、その対策について検討する。

6.4 節では、提案アルゴリズムの計算時間の短縮化を検討した結果を述べる。

## 6.1 サーボパラメータの自動調整について

サーボパラメータの自動調整は古くから取り組まれてきた課題であり、1980 年後半から定点観測的に動向調査が報告されている [94, 95]。それらの基本原理は様々であり、例えば制御対象のモデルを同定して解析的にパラメータを決定する方法から、特定の特徴量に注目したもの、ヒューリスティックによるものまで、数多くの手法がある。市場に投入される商用の制御システム製品や制御システム設計のソフトウェアツール類についても自動調整機能は必須といえる状況であり、整定時間の短さが厳しく要求されるのに制御対象のメカの剛性が低くて移動速度で稼げないなどの特殊な場面でない限り、自動調整機能は便利に利用されている。

その一方で、前述の特殊な場面に遭遇した場合には、人手による調整が実施されており、その自動化が求められている。そこでは、現場ニーズを満たすため、制御対象にダメージを与える発振現象を避けながらの安全な調整、より高ゲインの追求、自動調整に要する時間の短縮、自動調整の適用可能範囲の拡大、などを目標として、企業間の競争領域として研究開発、製品化開発が続けられている。能動型探索アルゴリズムをサーボパラメータの自動調整問題に適用することは、上に述べたような人手で調整しないといけない特殊な場面を想定していることになる。

本研究ではそのような場面を、パラメータの変化に対する制御性能の変化を制約条件付きの目的関数をノンパラメトリックモデルで逐次構築しながら同時にパラメータの最適値を探索するという、未知の目的関数の最適化問題として捉えることにした。

なお、解法が確立した場面から外れているからこそ特殊といわれる場面であるから、その類例と適切な解法との対比はにわかには困難である。適用事例を積み

重ねてながらもまずは傾向を分析していくことが必要で、その過程の中で提案するアルゴリズムが適用可能あるいは適用不適切な場面や条件を明確化していくことが重要であると考えている。もちろん同アルゴリズムが全ての場面でサーボパラメータの自動調整に適していないという結論が導かれることは厭わない。

## 6.2 同アルゴリズムの適用性の検討

前章で述べた能動型探索アルゴリズムが、位置決めサーボシステムの制御パラメータ調整に効を奏するかを確認するにあたり、市場で使用されている位置決めシステムの主要構成のみに抽象化した実験を行うことにした。すなわち位置決めシステムの最小構成要素であるサーボコントローラ、モータ、ボールねじ、軸受け、リニアガイド他の機構を組み合わせた実験装置を構築し、制御パラメータを与え、位置決め動作を実行し、動作状態を計測する系を組んだ。定性的な話になるが、実験装置のメカニカル要素の各部材は高剛性素材を使用せず、組立作業、据付作業についても微調整などに格別の注意を払わずにおいたため、ガタや動きの渋さが部分によって異なるなど、いくばくかの非線形性をはらんでいると考える。このような系に対し、制御パラメータと位置決めストロークを設定し、それらの全組み合わせで位置決め動作させ、パラメータ空間内の全探索データを取得した。なお同じ制御パラメータに対する動作の再現性は今回の実験では問題なく担保されている。

まず、今回の実験装置の概略構成を図 4.23 に示す。これは制御パラメータの調整について、同一コントローラ、同一メカのまま、熟練作業者が行っている作業を能動型探索アルゴリズムに切り替えて実験する構成である。

つぎに位置決めサーボの制御ブロック線図を図 4.24 に示す。こちらはいわゆる 2 自由度制御系である。フィードフォワード部について、位置の P 制御、速度の P 制御、および機械モデル  $1/J \cdot s$  で構成し、フィードバック部について、位置の P 制御、速度の PI 制御で構成してある。

フィードフォワード部について、位置応答ゲインを  $\omega_{pc1}$ 、速度ゲインを  $\omega_{sc1}$  とし、 $\omega_{sc1}$  については  $\omega_{sc1} = 4 \cdot J \cdot \omega_{pc1}$  とし  $\omega_{pc1}$  に従属して自動的に定まることとした。

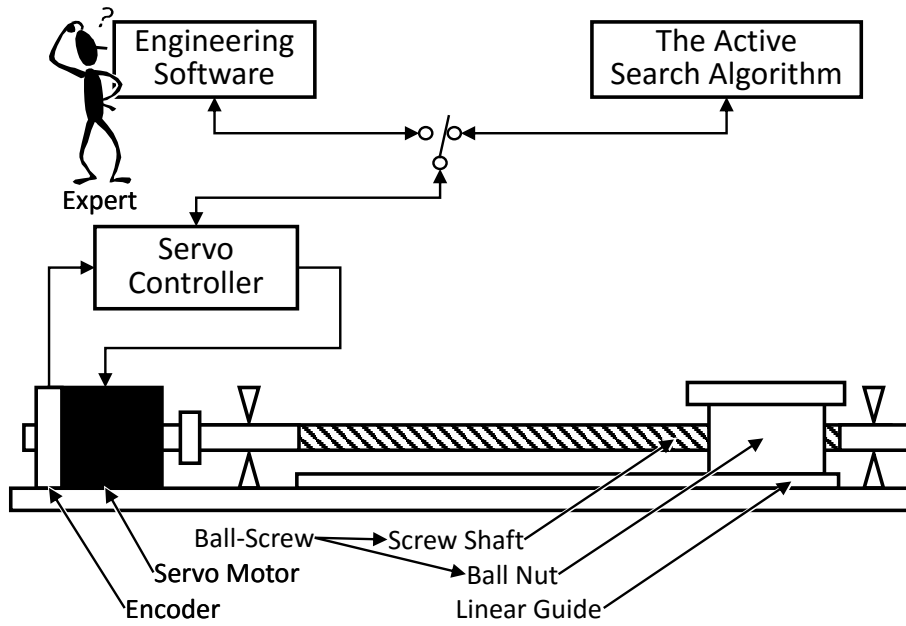


図 4.23 実験の設定

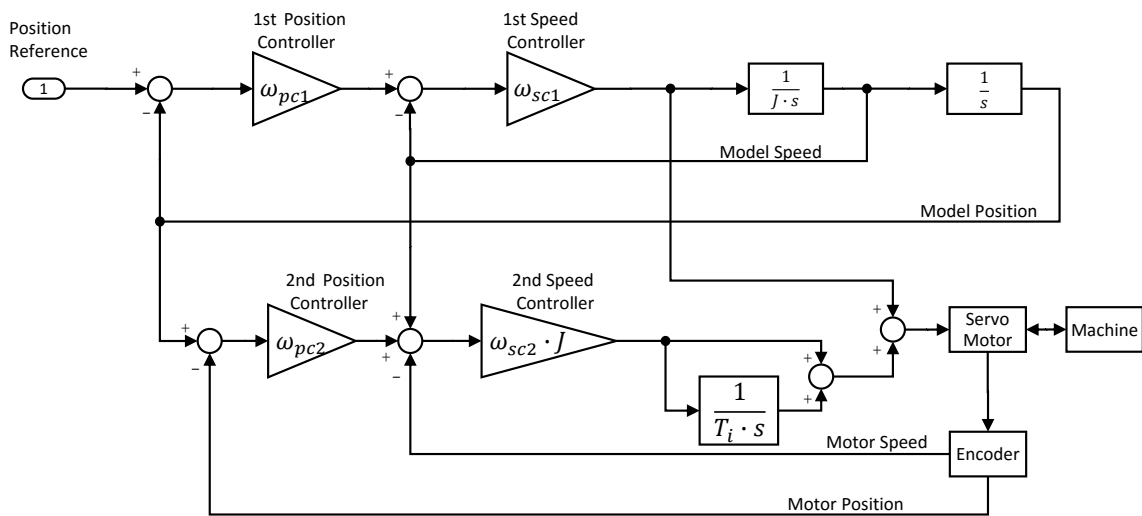


図 4.24 実験に用いたサーボ制御装置の制御ブロックダイアグラム [96]

またフィードバック部について、位置応答ゲインを  $\omega_{pc2}$ 、速度ゲインを  $\omega_{sc2}$ 、速度積分時間を  $T_i$  とする。

上述の実験装置において、適当な制御パラメータを設定し、適当な移動スト

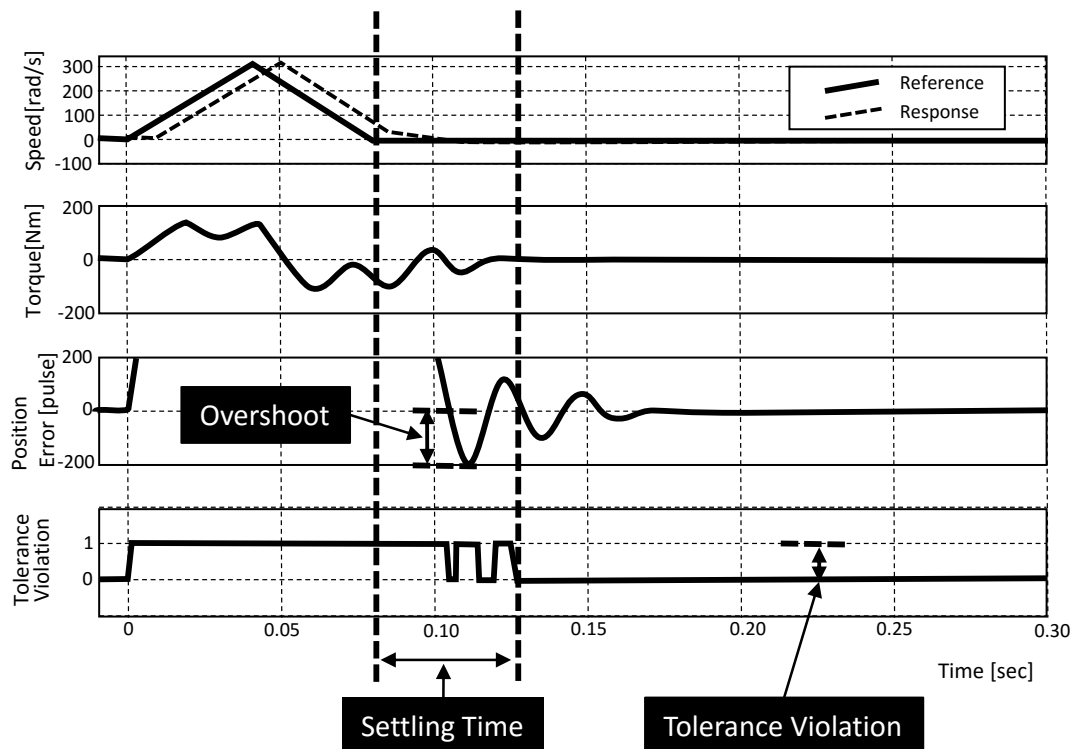


図 4.25 評価指標

ロークを経て位置決めする動作を実行すると図4.25のような波形データが観測され、与えた制御パラメータおよび移動ストロークの組み合わせに対する評価指標値を計測することができる。評価指標としては「許容整定範囲割れ」「整定時間」「オーバーシュート量」があり、これら評価指標は未知の制約条件付きの未知の目的関数を描く。各評価指標を説明する。

- 許容整定範囲割れ：0 または 1 いずれかの数値。

指令位置に対する偏差が別途与えられる許容範囲内に収まっていない状態のこと。以下の基準に照らすと図4.25の観測波形は「許容整定範囲割れ」1と判定される。

0：位置決め指令入力以降、位置偏差が最初に許容範囲内に入ってから、次回指令入力まで許容範囲外になることがない場合（良い）。

1：位置決め指令入力以降、位置偏差が最初に許容範囲内に入ってから、次

回指令入力まで許容範囲外になることがある場合（悪い）。

- 整定時間 [msec] : 位置決め指令入力以降, 位置偏差が最後に許容範囲内に入るまでの時間 (小さいほど良)。
- オーバーシュート量 [pulse] : 指令入力以降, 最初に許容範囲内に入ってから位置偏差の行き過ぎ最大値 (小さいほど良)。

「許容整定範囲割れ」「オーバーシュート量」についてはモータ軸に取り付けられた1回転あたり13万パルス ( $2^{17} = 131,072$ [pulse]) の分解能を持つエンコーダで計測する。今回の実験装置では, ボールねじのリードは10[mm]であり, その1回転でナット側は10[mm]移動するので, エンコーダの1[pulse]はナット側が0.07629[ $\mu$ m]移動することに相当する。このとき, 許容される整定幅を50[pulse]と与えれば, 目標位置に対し3.815[ $\mu$ m]以内のズレ量で整定した時「許容整定範囲割れ」は発生していないことになる。またオーバーシュート量を50[pulse]と計測すれば, 実際には4[ $\mu$ m]ほどオーバーシュートしていたことになる。ただし, これらの数値はモータ軸に取り付けたエンコーダの回転角度を計測しているので, モータ軸からボールねじ, ボールねじと嵌合しているナット, ナットから先の部材といった伝達機構を経た機械端がこれらの値通りに動くものではないが今回は問題とはならないレベルである。機械端の動きをリニアエンコーダなどで計測しフィードバックループを組んで機械端の位置決め精度を高めることはもちろん可能であるが, ここでの議論とは直交する話題である。

以下, 本論文では, 上述の位置決め実験装置に対して, 操作可能な制御パラメータのうち, フィードバック系の応答と安定にかかるゲイン  $\omega_{pc2}$  と  $\omega_{sc2}$  は事前に参考文献 [96] に示した方法で適正に調整されているという前提で, 「オーバーシュート量」と「整定時間」の特性を決めるゲイン  $\omega_{pc1}$  と  $T_i$  を, 提案アルゴリズムにより探索的に自動調整し, 指定された目標仕様を満たす最適化について考察することにする。

## 調整に対する要求仕様について

現場の制御パラメータ調整では上述の評価値に対する位置決め目標仕様が指定される。これを位置決め前の移動ストローク、位置決め場所(6.3章にて後述)などの誤差因子に対してロバスト化する必要がある。典型的な市場用途をかんがみて目標仕様辛め、目標仕様甘めの2種類を用意した。甘め、辛めの目標仕様は次のとおりである。

- 目標仕様甘め

整定時間 50[ms] 以内

オーバーシュート量 85[pulse] 以内

許容整定範囲 85[pulse] 以内

一般産業用途（ピック&プレース装置，包装機械，etc.）

- 目標仕様辛め

整定時間 35[ms] 以内

オーバーシュート量 50[pulse] 以内

許容整定範囲 50[pulse] 以内

高速高精度用途（チップマウンタ，工作機械，etc.）

## 今回の実験方法について

本稿では前述のとおり，位置決め実験装置を用いて制御パラメータの定義域内と位置決めストローク組み合わせ空間の全探索を実施しデータを取得した。これらのデータを目標仕様によって整理してデータセットを作り，計算機のみによる探索実験を実施した。位置決め実験装置を，同じ制御パラメータで動作させたときの再現性は目標仕様に対して十分に高く，あらかじめ全探索により作ったデータセットを計算機のみで探索しても，実際に制御パラメータを振りながら実験装置を動かして探索しても，得られる結果は同等である。

データセットは、移動ストローク別に4つのファイル（移動ストローク 1[mm], 5[mm], 20[mm], 50[mm]）があり、目標仕様別に2セット

一方、探索の対象となる制御パラメータは上述のとおり  $\omega_{pc1}$  と  $T_i$  の2種である。

これらの2つのパラメータを 10水準  $\times$  10水準=100点で組み合わせ、前述の4つのストロークで動作させ、2種類の目標仕様に従って評価値を算出、先の各ファイルに格納した（ある制御パラメータ、あるストロークにより得られたひとつの観測波形について、甘め辛めの目標仕様によりデータを整理して、甘め辛め2つのデータセットそれぞれに格納する）。

### 多目的最適化の実装について

能動型探索アルゴリズムを今回の問題における多目的最適化に適用するための実装の考え方について述べる。具体的には、最適化目標が「許容整定範囲割れ」、「整定時間」、「オーバーシュート量」の3つあり、動作ストロークも4種類あるため、アルゴリズムを計算するなかで、これらの事項をどのように取り扱うかである。

制御パラメータは自分で操作できる制御因子であるが、移動ストローク 1mm, 5mm, 20mm, 50mm は受け入れるしかない（自分で操作できない）誤差因子と考えることにする。移動ストロークは4つあるので、各制御パラメータ組み合わせ毎に4つの実測値が得られるが、それらの最悪値を採用することで移動ストロークの変動に対してロバストな評価値とする。

一方、各評価値に対する実行不能判定は、優先順位をもって実施する。具体的には「許容整定範囲割れ」 $>$ 「整定時間」 $>$ 「オーバーシュート量」とする。そして最も厳しく判定したい「許容整定範囲割れ」の0, 1を用いて、未知の制約条件の識別境界の推定モデル  $f(S_n, \mathbf{x}) = 0$  を更新し、最良の試行結果を選ぶ際の優劣の判定に上述の優先順位を用いる。例えば「整定時間」が改良されれば「オーバーシュート量」が悪くなっているても良い解が見つかったと判断することになる。上述のとおり制約条件の推定モデルにはSVMを用いた。なお、目的関数の推定モデルについては、探索空間が2次元で狭いため探索の高速化に寄与しないため、今回の適用ではスコア計算に加えないことにした。

なお、上記の方法で元データを前処理したところ、目標仕様辛めのデータ群については、「許容整定範囲割れ」していないデータの中に、整定時間が目標の範囲に入っているものがなかったことを特記しておく。

また、探索を始めるに当たり、初期実験点を与える必要がある。直交表 L4 あるいは D 最適計画などに従い、制御因子の定義域の 4 隅の 4 点を与えることにする。

## 適用結果

まず、前節の手順に従い、実測値から求めた 2 つのデータセットを図 4.26 に示す。これが全探索のデータとなる。非線形性の存在がわかる。

以降、実際に実機を動かす代わりにプログラムでこれらのデータの所定の箇所を読み出して、計算機のみによる数値実験を行う。パーソナルコンピュータは 10 万円程度の価格で、OS は Microsoft 社の Windows シリーズである。プログラミング、コンパイルには同社 Visual Studio の C++ 言語を用いた。



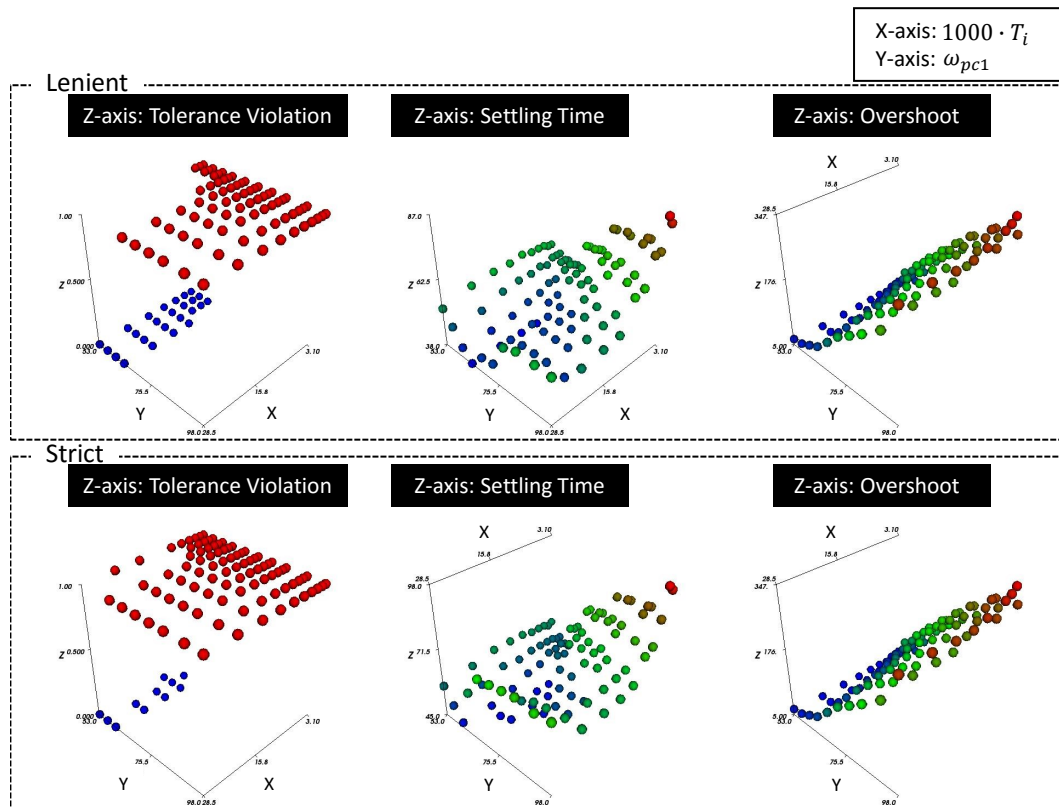


図 4.26 データセット

図 4.27 および図 4.28 は，探索結果をプロットしたものである。

横軸は試行回数，左縦軸が整定時間(小さいほど良い)，右縦軸がオーバーシュート量(小さいほど良い)，○印の折線がその時点までにおける最良の「整定時間」，□印の折線がその時点までにおける最良の「オーバーシュート量」である。各曲線はフラットな部分が描かれるが，これはその試行回においてそれ以前の最良解(前述の優先順位を考慮)を超える解が見つからなかったことを示している。解が改良されない回でも各推定モデルの正確性は改良されている。図から分かる通り，初期点4点を含めて7試行目および15試行目に最適点に到達していることが確認できる。図 4.28 では6試行目で、「オーバーシュート量」が悪化した。「整定時間」が改良されたサンプルが，前述の優先順位に従って最良解として採用されている。

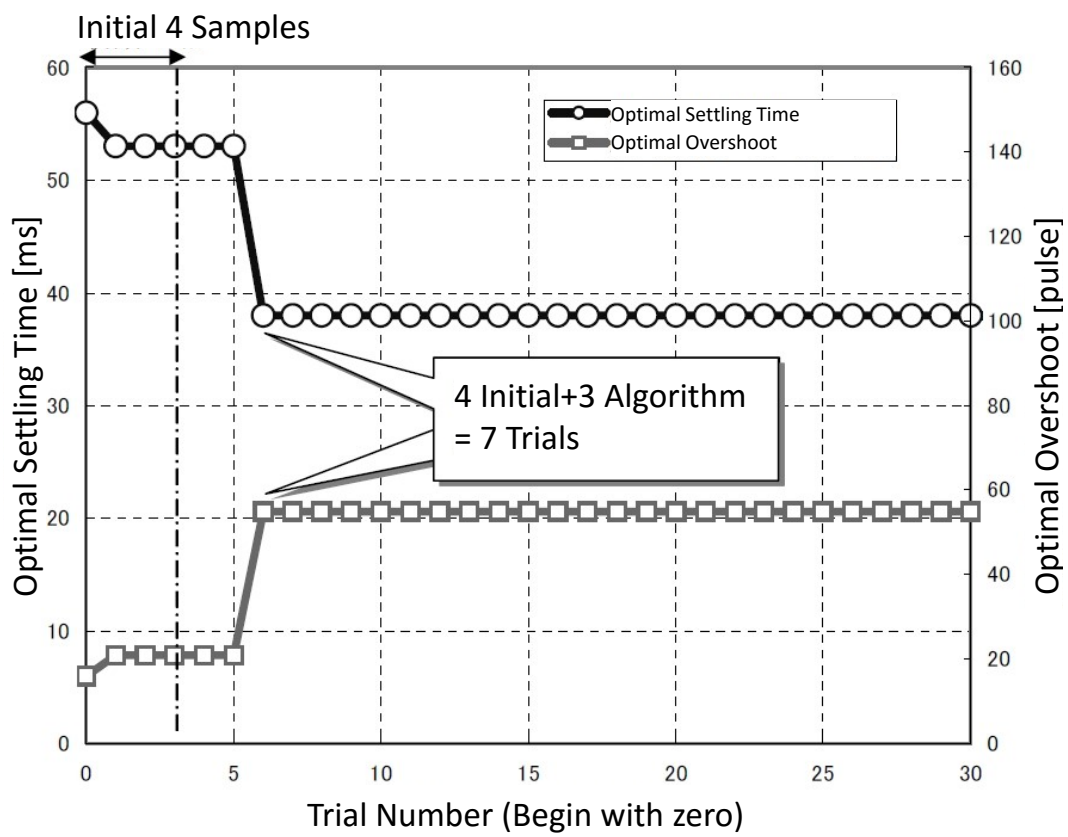


図 4.27 目標仕様甘めの設定による実験結果

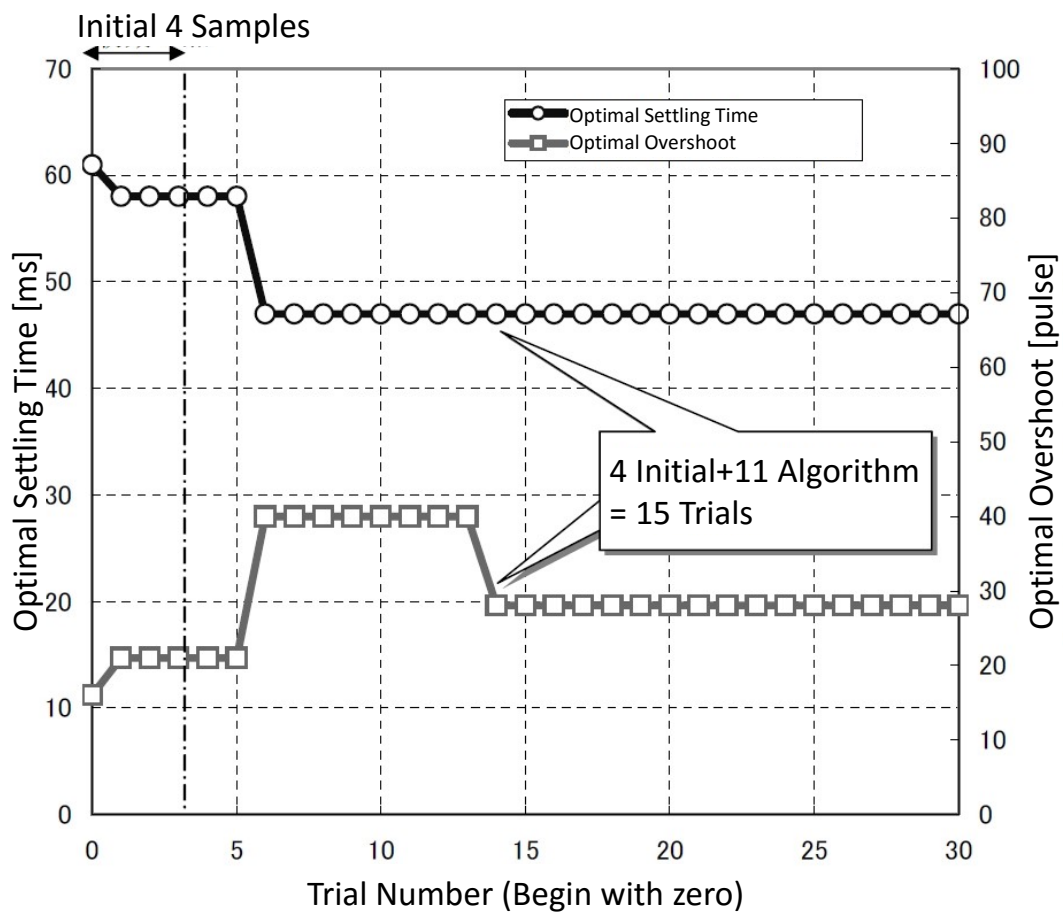


図 4.28 目標仕様辛めの設定による実験結果

実験結果を表 4.1 にまとめる.

なお, 辛めの評価仕様において「整定時間」目標値 35[ms] 以下のところ, あらかじめ計測した全データの中では 47[ms] が最小値であった.

提案する探索アルゴリズムでは, それまでの時点で観測されたデータ群を用いて識別境界モデルを推定する. 今回の計算機のみの数値実験では全探索済のデータのうち, その時点まで読みだしたデータを用いて推定モデルを作る. すべての全探索済データを用いた識別境界モデルが真のモデルとなる. 試行途上で推定モデルは試行するたびに真のモデルに近づいていく. SVM は, ソフトマージン SVM としサンプル点が増えてきて複雑になり, それらをひとつでも正しく識別分離出来なくなったときに SVM のパラメータを動的に変更してサンプル集合を完全に識別できる値に増大する実装にしてある. 複雑な識別を可能とするパラメータは SVM の計算コストが高いため, このように段階的に複雑な識別を実現することは計算コストを適切に抑制させると考えている.

ここで探索の進行の様子を仕様甘めのケースで図 4.29 に示す. これは 20 試行目のサンプル集合とそれを用いて描いた識別境界の推定モデルで, 上方の 4 分の 3 程度の領域が「許容整定範囲割れ」している領域, 下方 4 分の 1 程度の領域が「許容整定範囲割れ」していない領域である. 丸印は各サンプル点で, その横の数字が探索番号 (0 から 19 まで) で, 例えば 6 は「許容整定範囲割れ」していない領域に属し, 11 は「許容整定範囲割れ」領域に属する. 探索を開始すると, まずはじめに  $\omega_{pci}$  及び  $T_i$  の定義域の 4 隅を初期点として探索番号 0,1,2,3 と進み, その後は次のサンプル点選択を最適化しながら 4, 5, 6 と進み, 最適点に到達する.

表 4.1 2 種類の要求仕様についての実験結果一覧

|                     | Lenient Settings |                      | Strict Settings |                      |
|---------------------|------------------|----------------------|-----------------|----------------------|
|                     | Result           | Reference            | Result          | reference            |
| # of Trials         | 7                | —                    | 15              | —                    |
| Tolerance Violation | OK               | ( $\leq 85$ [pulse]) | OK              | ( $\leq 50$ [pulse]) |
| Settling Time [ms]  | 38 (OK)          | $\leq 50$            | 47 (NG)         | $\leq 35$            |
| Overshoot [pulse]   | 55 (OK)          | $\leq 85$            | 40 (OK)         | $\leq 50$            |

この結果を見る限りは  $\omega_{pc1}$  の定義域の右端に達しているので  $\omega_{pc1}$  の定義域を広げて探索するとより良いパラメータが見つかる可能性があることがわかる。同様に仕様辛めのケースを図 4.30 に示す。

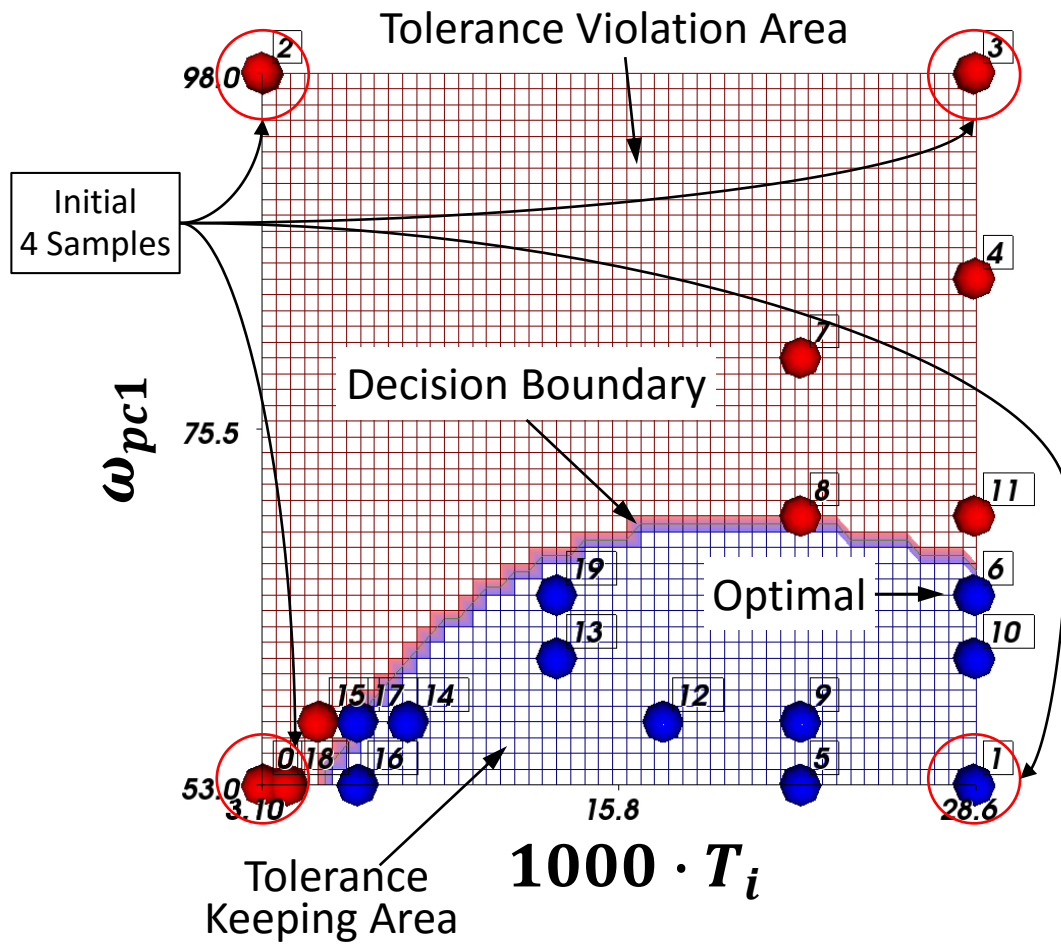


図 4.29 目標仕様甘めの設定における探索結果

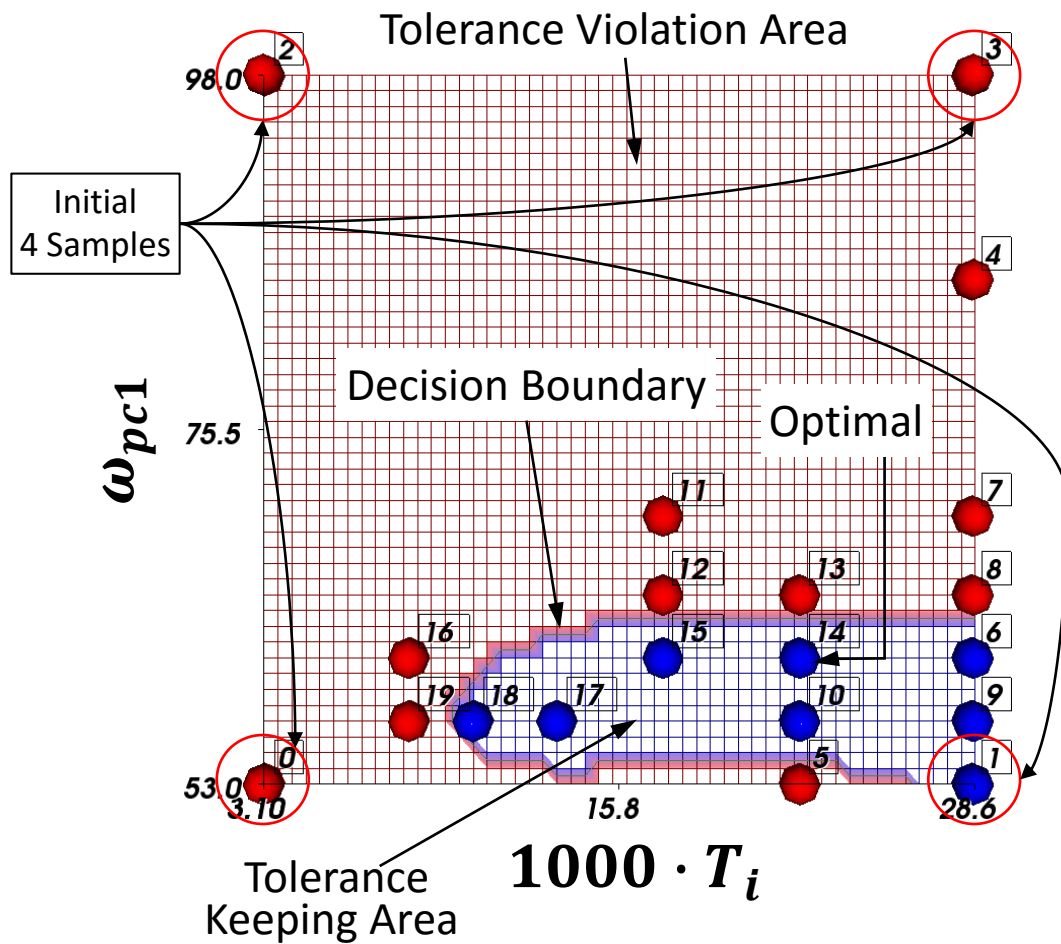


図 4.30 目標仕様辛めの設定における探索結果

### 6.3 誤差因子の処理方法の検討

本稿において、誤差因子は移動ストローク長であり、複数の移動ストロークに対してロバストな制御パラメータを探索する効果を得るため、各ストロークの最悪値を採用している。

現実世界では、さらなる誤差因子が存在する。具体的には、位置決めサーボシステムによる位置決め制御では、位置決め評価場所の移動（ボールねじ根元→真中→先端等）も考慮する必要がある。これは位置決め装置のメカニカルな非線形性により位置決め評価場所が変わると特性が変化しているからで、現場では一般的に発生する事象である。すなわち、これが新たに考慮すべき誤差因子となる。やはり、制御パラメータのロバスト性を考えて、異なる位置決め評価場所の最悪値を採用することでこの誤差因子を考慮できる。

ただし、対象プラントの構造の都合上、移動動作が自動化できない場合がある。このようなときには、ユーザが手動で移動を行うことになる。しかしながら、ユーザが調整システムの指示にしたがって、根元→先端、再度、先端→根元に手動移動するといった操作を行うのは煩雑である。そもそも調整の自動化が目的であるのに作業者が機械の前に拘束されるのは難があり、実際に現場からも同様の意見は根強い。このとき、せめてすべての評価場所への移動が1回のみになるような方式に出来ないかを検討する。具体的には、手動で「根元」に移動させて自動動作、その後手動で「真中」に移動させて自動動作、手動で「先端」に移動させて自動動作、そして調整完了という手順である。

そこで、この問題を制御因子が2入力:  $x_1, x_2$  , 誤差因子が1入力:  $x_3$  で、評価指標が1出力:  $y$  の系として

$$y = F(x_1, x_2, x_3)$$

とモデル化する。もし制御因子による効果  $f$  と、誤差因子による効果  $g$  が独立していた場合、すなわち、次式

$$y = F(x_1, x_2, x_3) = \xi_1 \cdot f(x_1, x_2) + \xi_2 \cdot g(x_3)$$

のような形式になっていた場合には、実験結果から  $f$  と  $g$  を分離するので、 $f$  と  $g$  の値域の大きさの差に留意しながら  $f$  の値の大小差に意味があるように  $f$  の

最適化を実行すればよく、「すべての評価場所への移動が1回のみになるような方式」にすることが可能となる。

その一方で、上の式の形式になっていない（みなせない）場合には、実験結果から  $f$  と  $g$  を分離しえない。つまり、「すべての評価場所への移動が1回のみになるような方式」にすることは、最適化の結果に妥当性を欠くことになる。以上の議論の具体的な実装方法としては、各評価場所への移動を1回のみにした実験を実施したのちに、 $f$  と  $g$  が独立していると言えるか否かを検定した上で、最適化完了とするか、さらに実験が必要かを判断する方法が考えられる。

## 6.4 計算時間の短縮化の検討

まず能動型探索アルゴリズムの計算手順を説明する。変数を次のように定義する。

|              |                 |
|--------------|-----------------|
| 初期サンプリング点数   | $n_{initial}$   |
| 探索サンプリング試行数  | $n_{trial}$     |
| 次のサンプル点の候補数  | $n_{candidate}$ |
| 次の候補の発生時間    | $T_{generate}$  |
| 次の候補のスコア計算時間 | $T_{score}$     |
| サンプリング時間     | $T_{sample}$    |
| 推定モデル再構築時間   | $T_{remodel}$   |

すると、図 4.31 から能動型探索アルゴリズムの計算時間  $T_{total}$  は、次式で表される。

$$\begin{aligned} T_{total} = & n_{initial} \cdot T_{sample} + T_{remodel} \\ & + n_{trial} \cdot (T_{generate} + n_{candidate} \cdot T_{score} \\ & + T_{sample} + T_{remodel}) \end{aligned}$$



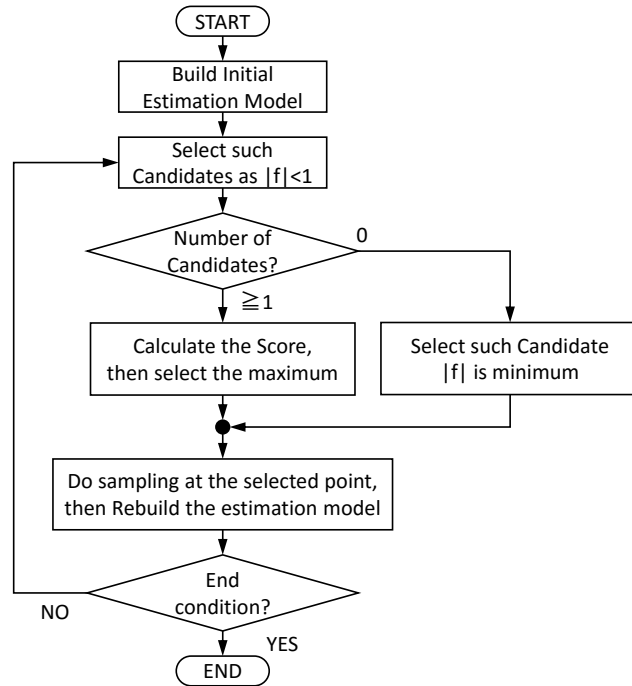


図 4.31 能動探索アルゴリズムの実装例

ここで初期探索点数は探索試行数に比して十分小さいとし無視する。

$$\begin{aligned}
 T_{total} \approx & T_{remodel} \\
 & + n_{trial} \cdot (T_{generate} + n_{candidate} \cdot T_{score} \\
 & + T_{sample} + T_{remodel})
 \end{aligned}$$

さらに試行1回分の差異を考えるため  $n_{trial}$  で微分する。

$$\begin{aligned}
 T_{total} \propto & T_{generate} + n_{candidate} \cdot T_{score} \\
 & + T_{sample} + T_{remodel}
 \end{aligned}$$

ここまで式変形して残った4項のうち  $T_{generate}$  は推定モデルの順方向の計算が主であり計算コストは他項に比して小さい。さらに残った3項のうち、 $n_{candidate} \cdot T_{score}$  と  $T_{remodel}$  を比べると、 $T_{score}$  には2回分の  $T_{remodel}$  相当時間が含まれるため、明らかに  $n_{candidate} \cdot T_{score} \gg T_{remodel}$  となり残り2項に比して小さくなる。結局、1

回の試行に要する時間を支配するのは、

$$n_{candidate} \cdot T_{score} \quad \text{および} \quad T_{sample}$$

である。

ここで  $n_{candidate} \cdot T_{score}$  と  $T_{sample}$  の大小関係を考えてみる。  $T_{sample}$  は対象プロセスを実際に動作させて結果を得る時間であり、対象プロセスごとに多いに異なる。著者らのこれまでの適用事例では数時間に及ぶものから今回の位置決めシステムのように秒単位のものまで存在する。この  $T_{sample}$  が短くなってくると  $n_{candidate} \cdot T_{score}$  が相対的にボトルネックになりはじめる。

$n_{candidate}$  の最大値は全ての未試行点の数であり、これが計算時間最大のケースとなる。計算時間が最大とならないよう、制約条件の識別境界から一定の距離にある未試行点のみ選ぶ、あるいは目的関数の推定モデルで未試行点による目的関数の推定値を計算し、推定値降順に上位の一定数個までのみに絞り込むなどの方法で  $n_{candidate}$  を減らす方法がある。ただしこの方法の欠点は最適化を考慮する範囲が小さくなり真の最適点を逃がし結果的に試行回数の増大を招く可能性が発生することにある。

実は、  $n_{candidate}$  を増やして次のサンプル点をどこにするか時間をかけてじっくり考えると試行数が減り、  $n_{candidate}$  を上述の方法で減らすと考える時間は減るが試行数は増えるという実験例が得られている。ということは、  $n_{candidate} \cdot T_{score}$  と  $T_{sample}$  の大きさが近いときには、  $T_{total}$  を減らそうとするとき、次の一手を選ぶのにあれこれ考えてじっくり時間をかけて試行回数を少なくするほうが良いのか、長考せずに次々にサンプリングするほうが良いのかというジレンマが発生する。このことは別の機会に報告したい。

さて、  $T_{score}$  の内訳に注目すると、それは次のような順序で実施される計算から構成されている。まず、その実験点候補で実験した結果による識別境界の変化の予測のため、制約条件を満たした場合／満たさない場合のそれぞれについてSVMにより識別境界を計算する。すなわち二次計画問題を2回解いている。さらに、現時点で得られている識別境界と上記で求めた2つの識別境界とに挟まれたそれぞれの多様体の体積を積分してその実験点候補に対するスコアの各項を得ている。識別境界が3つあるのでそれらに挟まれる空間は2つになる。本アルゴリズムで

表 4.2 CPU コア数を増やした際の計算時間短縮効果の計測結果

| Computer Type           | Single-Core                           |       | Dual-Core         |      | Quad-Core       |      |      |      |
|-------------------------|---------------------------------------|-------|-------------------|------|-----------------|------|------|------|
| CPU                     | Pentium4 3.2GHz<br>(Hyper Thread Off) |       | Core2 Duo 2.13GHz |      | i7-4770 3.40GHz |      |      |      |
| Memory                  | 2GB                                   |       | 1GB               |      | 8GB             |      |      |      |
| OS                      | Windows2000 SP4                       |       | WindowsXP SP2     |      | Windows7 SP1    |      |      |      |
| # of Thread in Program  | 1                                     | 2     | 1                 | 2    | 1               | 2    | 3    | 4    |
| Mean Time[ms/candidate] | 1,276                                 | 1,267 | 991               | 522  | 210             | 106  | 75   | 57   |
| Mean Reduction Rate     | 1.00                                  | 0.99  | 1.00              | 0.53 | 1.00            | 0.50 | 0.36 | 0.27 |

は、これらの計算コストの大きな操作をサンプル点候補の数だけ繰り返しており、計算時間がかかるのは当たり前という話になる。以下、このスコア計算高速化の一手法を検討する。次のサンプル点候補ごとのスコア計算は、まったく独立して並列計算できる。そこで、このスコア計算スレッドをマルチスレッド化して、マルチコア・コンピュータで実行することにより  $n_{candidate}$  を見かけ上小さくすることによる計算速度の向上率を探る。

実験として、スコア評価スレッドを Windows OS の `_beginthreadex()` および `_endthreadex()` 関数でマルチスレッド化し、`SetThreadAffinityMask()` 関数で起動されたスレッドを特定の物理コアに均等に分散して割り付けるプログラミングを実施した。その上でシングルコア CPU およびマルチコア CPU を持つコンピュータを用意しプログラミング上の並列スレッド数を増減させて演算時間を計測した。時間取得は `GetTickCount()` 関数を用いており分解能は 15.6[ms] であり、同じ計算を 100 回繰り返して計測すると 1 回あたりの分解能は 0.156[ms] となる。この実験では、今回の応用事例の探索空間が小さく演算時間の違いが出にくいと考え、既報文献 [65] に示した組み合わせ総数 88,128 のデータを用いることにした。その探索においては各試行において制約条件の識別境界推定モデルの近傍 (SVM のマージン < 1) に候補点が約 6,000 点発生する。それらを目的関数の推定モデル (具体的には重線形回帰分析モデル) による推定値で上位 100 点に絞り込んでスコア計算を実施している。探索試行途中の連続 10 試行分のスコア計算時間を計測して、スコア計算候補点 1 点あたりの計算時間の平均値を算出、スレッド数 1 のケースに対する短縮率を求めた。短縮率というのは、短縮率 0.5 な

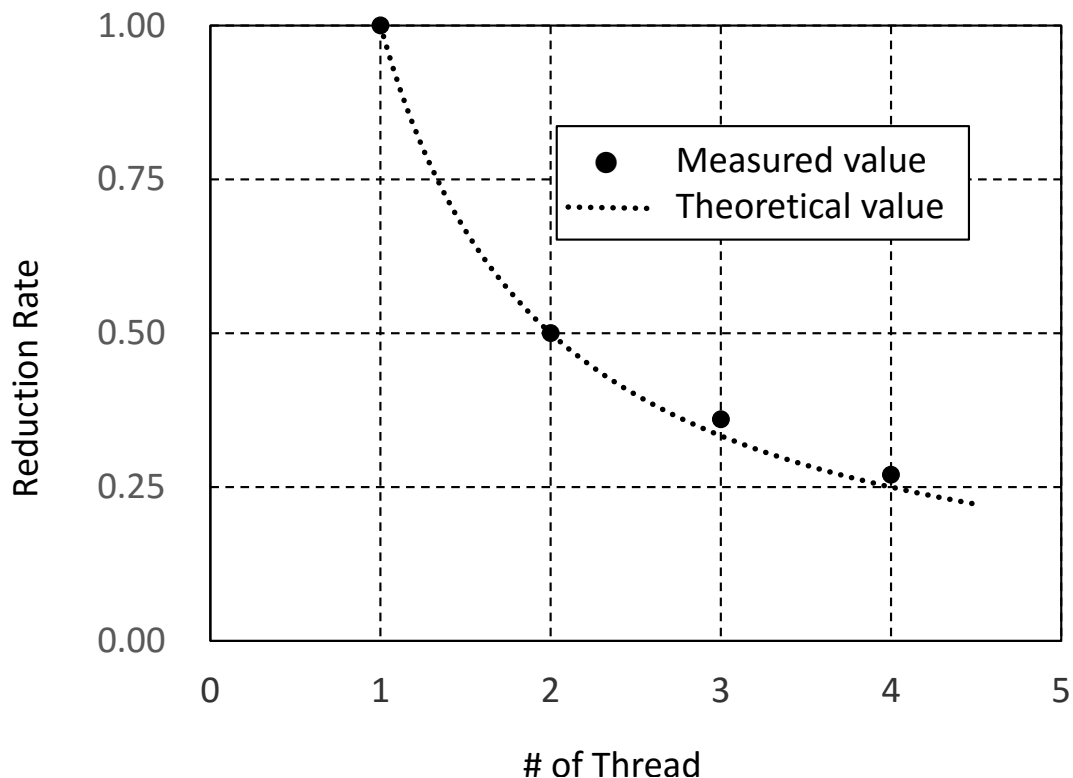


図 4.32 CPU 数と改善率の関係

ら計算時間が  $1/2$  に、同  $0.25$  なら同  $1/4$  になるという定義である。結果を表 4.2 にまとめる。

まず表 4.2 においてシングルコア CPU マシンとデュアルコア CPU マシンの計測結果を比較する。スレッド数を変化させたとき、シングルコア・コンピュータ上ではスレッド数を変化させても計算時間に変化は無い。デュアルコア・コンピュータ上でスレッド数 2 とすることで、スコア評価にかかる計算時間は約  $1/2$  へ短縮している。

次に、クアッドコア CPU マシンの結果を考察する。表 4.2 に示したプログラミング上のスレッド数を 1, 2, 3, 4 と変化させた場合の短縮率 (小数点以下第 2 位まで) を図 4.32 上にプロットしてみる。同図の横軸はスレッド数、縦軸はスレッド数 1 に対する短縮率である。これらの座標空間内に実験により計測された結果を黒丸で、 $n$  スレッドなら計算時間は  $1/n$  になるという理論値を点線でプロットする。

このとき、黒丸は理論値より若干上方に位置するため、計測値は理論値よりも短縮率が悪いことになる。これはスコア計算において並列化しえない部分が残っているからである。

以上の結果、CPU 物理コア数  $n$  以上の場合に並列スレッド数を  $n$  個にすることで  $n_{candidate} \cdot T_{score}$  の計算時間をおおむね  $1/n$  へ短縮しうると予想される。もちろん  $n$  を  $n_{candidate}$  以上に増やしても計算時間は  $T_{score}$  以下にはならない。

## 7. 初期探索に対する考察

本アルゴリズムでは初期探索として直交表を用いて、そこからさらに探索を進めている。言い換えると、直交表だけでは探索が完了しない例を解いているといえる。先に示した 2187 通りのケースの識別境界の非線形性を、7 パラメータのうち 3 つを用いて図 4.33 に示す。この図から、直交表による実験範囲の絞り方に困難性があることがわかる。さらに、山登り法に類する探索方法を用いた場合には、図中右上に示した最適点に到達するためには、図中の細かく波打った表面、手前に飛び出た鞍部を乗り越えていく際に、ローカルミニマにトラップされることが予想される。

今回開発した能動型探索アルゴリズムは、(i) 初期探索の方法、(ii) サンプルング候補発生の方法により探索性能が変化することがわかっている。

まず (i) について、本論文の 4.1 節で用いた 2187 通りの組み合わせのデータを用いて、初期探索点の与え方による差異について調べた事例を示す。まず、ランダムに 27 点の初期点を与えたのち、能動型探索アルゴリズムによる試行を実施することを 1,000 回実施し、各試行回数目に到達した最大加工速度の曲線を 1000 本集める。そして、これら曲線群の中から、同じ試行回数目における加工速度の最大値、最小値を求める。これを、初期探索点として L27 を用いていたものとあわせ、横軸を初期点 27 点実施後の試行回数、縦軸を正規化加工速度でプロットしたものを図 4.34 に示す。最上部を走る細い実線が初期値ランダムの場合の最大値のケース、最下部を走る一点鎖線が初期値ランダムの場合の最小値のケースであり、初期値 L27 のケースがそれら 2 つの曲線の、ほぼ中間を走る太い実線であ

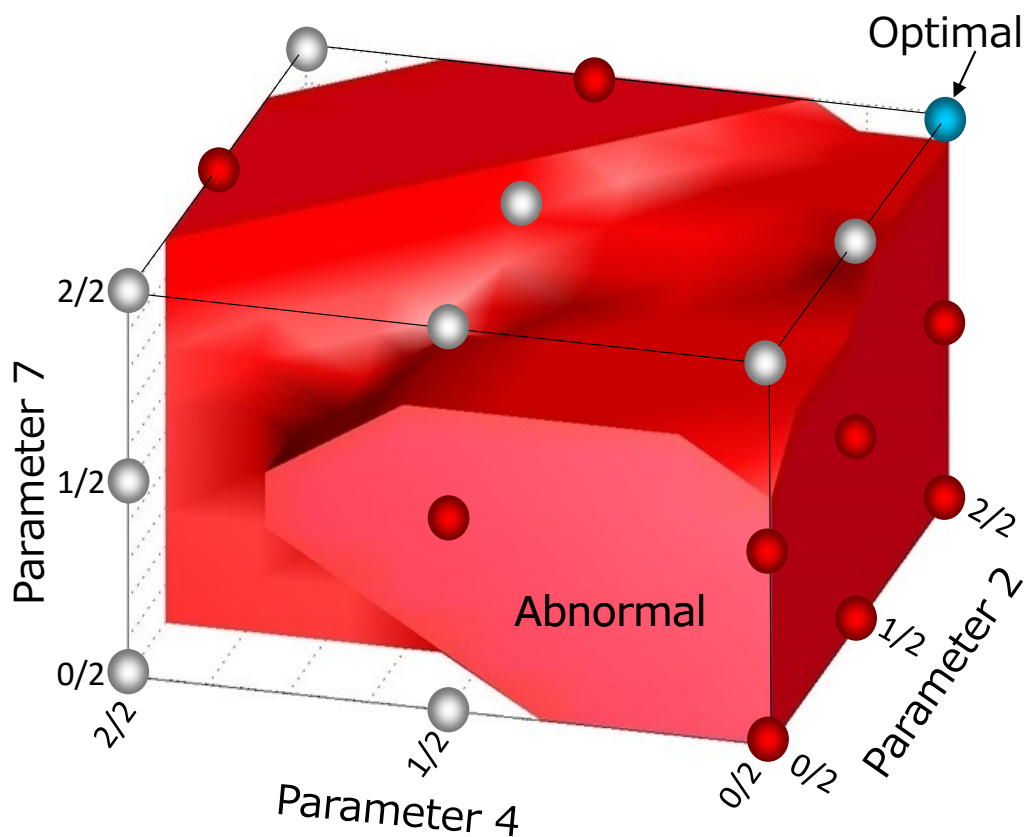


図 4.33 7パラメータ中の3パラメータを用いて表示した識別境界の非線形性の様子

る。曲線が下に折れるのは、あるパターンで最大値が見つかった後に残りのどのパターンも最大値に達していない場合が存在するからである。

このグラフからは、初期値の選び方で、その後の探索が早くなったり、遅くなったりすることがわかる。一方、L27初期探索は良くもなく悪くもなく中庸な結果をもたらしていることがわかる。

初期値問題の特殊なケースについては有効な対策がある。ひとつは離れ小島の異常域があったときで、初期点でこの離れ小島をサンプルし損ねていると本アルゴリズムでは最後まで探索がそこに近づかないため、時折りランダムなサンプリングを混ぜてみることである。もうひとつは離れ小島があるなど対象の特性に知見が得られている場合には、それを含む初期点を加えておくことである。これは

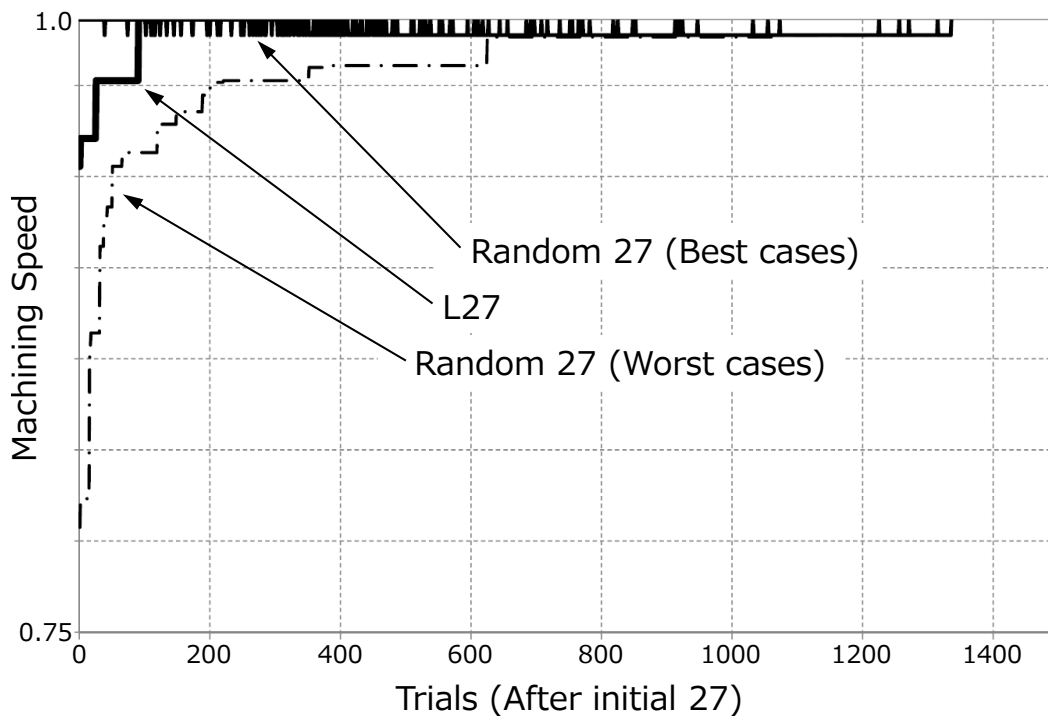


図 4.34 初期探索点の違いによる探索性能の変化

事前知識の利用による学習の加速であり、ベイズ理論的に正しいといえる。熟練者の熟練度は、この事前知識の多寡に拠るものであろう。もっとも、対象プロセスの特性が未知であるからこそ自動探索する機能を備えた能動型探索アルゴリズムの価値がある。

(ii) については、回帰モデルの近似度合いのパラメータ (例えばサポートベクターマシンのカーネル  $K(x_i; x_j) = e^{-\gamma \|x_i - x_j\|^2}$  の  $\gamma$ ) を大きくするとマージン内の未試行候補点数およびサポートベクトル数が増え、情報量算式の計算に時間がかかるようになる。サンプリングに5分間、次のサンプリング点最適化に30分間というような時間配分になったとき、「考えている間にサンプリングしたほうが良い結果が出るのではないか」というジレンマが発生する。実際に実験してみると、じっくり考えてからサンプリングしたほうが試行回数は少なくなるが、トータル時間は長くなる現象が観測された。また、サンプリングの並列化や逐次サンプリング点最適化計算の並列化についても、興味深い結果が得られつつある。

## 8. まとめ

本章では、試行錯誤による習熟という現象を、未知の目的関数の最適化問題に帰着させ、それを解くための能動型探索アルゴリズムを開発したこと、および、その適用事例により同アルゴリズムの有効性を述べた。

ロボットの動作軌道計画に対する適用事例では、熟練作業者が教示した初期軌道を、同アルゴリズムが56%に時間短縮する軌道を発見した例を得た。総組み合わせ数 $9 \times 17 \times 25 = 3825$ 通りの探索空間から46手目で最適点に到達した。

加工機制御パラメータ調整作業への適用事例では、実際に加工機を稼働させての全探索データ取り実験に3ヶ月を要したパラメータ組み合わせ総数2187点の最適化問題において、本アルゴリズムとの対比のために用いた「1回試行したところは記憶しておいて、それ以上選ばないことにした、記憶付きのランダム探索アルゴリズム」が、平均1379回で最適点を発見したのに対し、本アルゴリズムは、119サンプリング目で最適点に到達した。本アルゴリズムによる自動探索は最適解に収束し、実際に、熟練者が納得する好ましい制御パラメータが得られた。組み合わせ総数88128点のケースでは、熟練者と同等の5時間30分で、熟練者の1.31倍の性能の制御パラメータが得られた。これが本アルゴリズムの第1の効果である。第2の効果は、これまで熟練者が行っていた作業が、いわゆる段取作業を残して自動化・ノウハウレス化されることである。これにより、最長3ヶ月でばらつく制御パラメータ調整作業が約1週間に短縮・安定化された。

また、産業メカトロニクス分野で用いられる位置決めサーボシステムの制御パラメータ調整、すなわち制御パラメータの調整において発生している未知の目的関数の最適化問題に対して、著者らの開発した機械学習の枠組みに基づく能動型探索アルゴリズムの適用可能性を検討した。

- 最適化目標が3つという多目的最適化の考慮、および受け入れるしかない誤差因子としての動作ストロークの変化に対するロバスト性の確保を考慮しながら、最適パラメータを発見しうる結果を得た。

具体的には、「許容整定範囲割れ」、「整定時間」、「オーバーシュート量」の3つの目標に対し、位置決め制御における制御パラメータの総当たり組み合



わせ数 100 点 (各点移動ストローク 4 種) の実測データ 2 セットを用いて多目的最適化の実装例として目標仕様の優先度を考慮した探索実験を実施したところ、それぞれ 7 試行目、15 試行目に、全探索データ内の、移動ストロークの変更に対してロバストな最適値に到達することを確認した。

- 誤差因子に対するさらなるロバスト化が必要となる場合を取り扱えることを示すため、ボールネジ全ストローク中の根元、中程、先端といった評価場所の変化に対するロバスト化の対策について考察した。「すべての評価場所への移動が 1 回のみになるような方式」のみを採用することは、最適化に関して妥当性を欠く場合があると考えられる。
- サーボパラメータの自動調整は、制御対象の類型化とそれに対する調整手法の体系化が進んでおり、通常は既存の調整手法で十分な性能が得られている。これに対し、既存の調整手法が届かない場面、すなわち制御結果に対する要求仕様が高度な特殊な場面についての自動調整が求められている。今回用いた実験装置がどこまで特殊なケースを再現しているかは、特殊なケースの類型が困難であることもあって定量的な比較は困難ではあるが、与えられる装置、与えられる目標仕様について誤差因子を考慮しながらの自動調整が実現している。
- 能動探索アルゴリズムの次サンプル点最適化処理を、並列計算による処理の高速化を検討した。並列計算により処理時間は短縮されることがわかったが、サンプリングに要する時間が短いケースでこそ、この高速化は意味を持つ。

以上より、サーボパラメータの自動調整に対する提案する能動型探索アルゴリズムの適用性について、多目的最適化、誤差因子に対するロバスト化、計算時間の短縮化といった現場の要求の考慮を含みつつ、一定のエビデンスを示すことはできたと考える。

今後、未知の目的関数の最適化という問題を共有する様々な事例に対し、さらなる適用事例の積み増しを通じて、提案アルゴリズムの特質を明らかにしながら、各事例ごとの要求仕様に対する実装方法を類型化したい。先行研究成果をはじめ、

他の手法が効果を発揮する場面ではそちらを使えば良い。能動型探索アルゴリズムをこそ使うべき場面を同定し、だれしもが理論ではなく技術として簡単に使えるための方法論の確立を目指す。

繰り返しになるが、本アルゴリズムが対象とする工業プロセスの良否を支配する制御パラメータが存在し、その最適化問題を考えている。パラメータを調整すると工業プロセスの良否を判定する連続値で与えられる指標が変化し、これを最大化もしくは最小化したい。この指標は制御パラメータを少しずつ変化させると少しずつ変化するが、その関数は解析的には表現できず、ノンパラメトリック関数で回帰するしか無いという点で未知の目的関数となる。また、制御パラメータを変化させていると、思わぬところで工業プロセスは突然実行不能に陥る。実行不能に陥る識別境界が存在し、この識別境界も解析的には表現できず、ノンパラメトリック関数で回帰するしか無いという点で未知の関数である。経験的に未知の目的関数の最良値は、制約条件を与える識別境界付近に存在している。このとき、未知の関数を回帰するためのデータセットは全く与えられておらず、そのサンプリングコストは高く全探索は困難で、できるだけ少ないサンプリングで最良のパラメータを探し当てたい。このため制御パラメータを変更しながら探索するしかないが、この探索を自動的に逐次実行する工夫をして、最適化問題として組み立てている。この手の最適化問題が産業現場では少なからず存在するためこの手の問題に対する適用事例を積み重ねたいと考える。

さらに近年ベイズ学習について、マルコフ連鎖モンテカルロ法や変分ベイズ法に代表される従来は困難とされてきた事後確率密度分布関数の効率的近似計算方法の実用化により、制御目標指令値の算出 [97] など様々な応用が提案されつつある。提案アルゴリズムはこのベイズ学習の実用的な一実装法である可能性があり、今後、その関係を明らかにしていきたい。

具体的には、工業プロセスのサンプリング時間と比較してリーズナブルな時間内で計算可能なアルゴリズム実装となることを重視し、制約条件の識別関数を2クラスのサポートベクターマシンで決定論的に計算している。対象とする工業プロセスの制御パラメータについてもデジタル制御装置に係るもので離散化して計算している(もっとも実験計画法の分野でもパラメータの離散化は常識である)。

ベイズ学習的立場を取れば目的関数と制約条件の存在可能性を確率密度関数で表し、さらに確率密度関数間の距離を定義して本アルゴリズムを構築することになる。これは図 4.2 に示した型式に近いと考えられる。



# 第5章 ロボット化困難作業を克服する自律システムアルゴリズム

本章は、自著ジャーナル投稿論文 [98] の内容に、加筆したものである。

## 1. はじめに

セル生産ロボットシステムを含む全ての自動化システムを製品組立作業に適用するとき、必ず問題になるのは、部品供給問題である。とりわけ問題解決への期待が高いのは、バラ積み状態の部品を供給することである。これはロボット学にとっては、ランダム・ビン・ピッキング問題という古典的難題として知られており、今日まで限定的な条件下での解法しか得られていなかった。本研究では多品種の部品を切り替えながら同一ロボットシステムでバラ積み供給を実現するシステム設計理論を提案する。この理論は、バラ積み状態の部品を、ロボットがつまみ上げて単離させるサブシステムと、複数のロボットで部品を持ち替えて、指定された姿勢に整列させるサブシステムを直列接続する実装を導出するもので、持ち替えの行動計画がポイントとなる。本研究では、ロボットによる持ち替え作業を計画する独自の自律システムアルゴリズム「持ち替えグラフ」を開発した。同アルゴリズムは、ロボットシステムの姿勢制約を考慮しながら、与えられた部品の平面上の安定姿勢、ロボットによる把持姿勢をノードとし、各姿勢間の遷移可否をアークとするグラフを作成した上で、ロボットの行動計画を算出し、作業者が試行錯誤しながらロボットプログラミングする作業を自動化するものである。

自動組立システムにおける課題のひとつは、部品供給の自動化である。とりわけ、バラ積み状態の複雑な形状の部品を合理的に供給することは、自動組立開闢以来の永年の課題である。1940年のパーツフィーダと呼ばれる専用の部品供給装

置の発明・その後の実用化・改良・普及の長い歴史 [99] に加え、ロボットが加工組立工場に広く普及し始めた 1980 年代には、ロボットによるバラ積み部品供給自動化への期待が高まったが、ほどなくして、容易には解けない問題であることがわかってきた [100]。その困難性の本質はランダム・ビン・ピッキングと呼ばれるロボット学上の古典的難題に集約され、今日に至るまで、この問題を解くための取り組みは枚挙にいとまがない。

本研究では、このようなバラ積み部品供給の自動化においても先例のない、同じシステムで複雑な形状の部品を多品種取り扱うという課題に取り組んだ。本章では一連の研究の中から、原理原則に立ち返って、この課題を合理的に解くためのシステム論的アプローチと、それを実現する要素技術を提案する。さらに、提案方式を実装することで、質量数 10 [g] 以下の 11 種類の部品を、特殊な治具、ハンドを用いず、最速 3 秒台間隔で、バラ積み状態から整列・組立できるロボットシステムを得た事例について述べる。

## 2. 関連研究

FA 分野の自動化製造システムにおいて部品供給をどう実現するかは、不可避かつ普遍的な課題である。現時点でも、技術的に合理的な部品供給自動化が困難な場面では、作業者がバラ積み状態からパレット上に整列させる作業を担当し、何枚ものパレットを自動組立機に投入する手段がとられている。コスト管理の観点からは、製品としての付加価値を生まない整列作業に人件費が投入されている。作業者自身にとっても、長時間、指定された速度を維持しながら、作業ミスのないように、単調な動作の繰り返しを強いられるという、きつい作業である。

部品供給の自動化手段としては、パーツフィーダが知られている。これは 1940 年の米国 Syntron 社の William V. Spaurlin 氏の発明に端を発すると云われており [99]、1950 年代以降に多くの改良が加えられ普及が進んだ。現在でも、比較的安価、かつ高速な部品供給を実現できるため、組立分野だけでなく、様々な業界で広く使われている。

その動作原理は次のようなものである。商用電源により駆動される電磁石と板

ばね等により，ボウル状の容器に微小な振動を与える．与えた振動がボウル内壁に沿って設けられた螺旋状のガイド通路に進行波を発生させて，ボウル内にランダムに蓄えられた部品をガイド通路上に一行に進行させる．ガイド通路に固定された一品物の細工（ツーリング）群に部品を触れさせて整列し，整列しきれない部品はツーリングによりガイド通路からボウル底部に篩落とされる．最終的にガイド通路上に生き残った部品が，ボウル内のガイド通路から分岐する直進フィーダ上に進み，そこで同じ姿勢で整列する．つまりは確率的な現象を利用する装置である．

このような動作原理であるから，部品の種類と同等の数のパーツフィーダが必要となり，工場内のスペースを占める．工場内で取り扱う部品品種が増えてくると，工場内の保管ヤードに，ボウルとツーリングが大量に保管され，使用部品の切り替えの際に，これらを付け替えながら使用することが通例である．なにより部品ごとにボウルおよびツーリングの設計・調整が必要になるため，機種切り替え時にコストがかかる．パーツフィーダの商談を開始すると，納品まで例えば1ヶ月程度のリードタイムが発生し，対象とする部品形状が複雑になると，ツーリングも複雑になり，エンジニアリングコストとなって跳ね返ってくる．さらに，ツーリングの設計と部品形状のばらつきとの相性で，運用時に部品が通路上で詰まってしまういわゆるチョコ停が頻発，作業者による復旧が必須となる．

パーツフィーダの主たる課題の解決策は，安定動作するパーツフィーダを迅速かつ確実に設計・製作することにある．これを目的とする研究開発事例に [101–105] がある．実際のところチャレンジングなテーマであり，未だ実用化には至った事例は聞こえてこない．依然としてパーツフィーダの設計・製作・運用には，多くを人手に頼っているのが現状である．

これらの取り組みと並行して，ロボットシステムによる解決策も長らく注目されてきたことは前述のとおりである．一連の取り組みのアプローチ方法の基本原理は，センサで物体の位置姿勢を認識し，マニピュレータが認識結果に基づいて，基準軌道を補正しながら，バラ積み部品を操作することにある [106–108]．これまでの数多くの学術的な取り組み，ロボットメーカー，ビジョンメーカー各社らの取り組みが結実し，2010年代に入って，実用化例が見られるようになってきた．

しかしながら，Marvel [109] らが分析しているように，その適用可能範囲は未だ限定的である．特に，実際の工場で扱うような，複雑な形状の部品対応に難がある．このようなロボットによるランダム・ビン・ピッキング（以下ビンピッキングと略す）は，ロボット学における難題の1つとして知られている．

このとき，本研究では，複雑な形状の部品の多品種供給の自動化という問題にフォーカスし，これを効率的に解く方法を模索してきたが，改めてこの問題の困難性を再考したところ，それらが，後述する2つの課題に集約されることを発見した．そこで，本章では，一連の研究成果の中から，従来のアプローチの基本原則たる位置姿勢認識という大きな問題に全力を注ぐのではなく，問題の困難性を複数の小問題に分割して解くというシステム論的アプローチにより，この問題を合理的に解くための方法と要素技術の提案を行なう．

### 3. 課題と提案方式

ビンピッキングは，バラ積みされた部品の姿勢を高速・高精度に認識するロボットビジョンの問題として扱われることが多く，部品の姿勢を高精度で認識する技術 [106–108] は確立してきた感がある．しかし，部品姿勢を高精度に認識したとしても，次の2つの課題がビンピッキングの実現を拒んでいる．

1つめの課題は，部品形状の認識の汎用性と処理時間のバランスにある．単純な円筒形状の部品であれば，多くの場合，側面を把持することで整列が可能である．平面状の部品であれば，裏か表を判断すれば整列が可能である．しかし，例えば多面体の場合，バラ積み状態で起こりうる姿勢状態は多数ある．そのため，高精度な姿勢認識や操作戦略の計算コストが膨大になり，現実的な処理時間で作業ができない．プリミティブな形状に特化することで速度，精度を向上させた認識アルゴリズム [110–112] の提案もあるが，当然一般形状に対する汎用性がない．この汎用性と処理時間の両立が課題である．

2つめの課題は，ロボットやハンドの器用さにある．もし，部品の姿勢認識から操作戦略を高速かつ正確に計算できたとしても，ロボットの動作範囲には限界がある．さらにはロボットハンドの物体操作性能にも限界がある．単純に言う



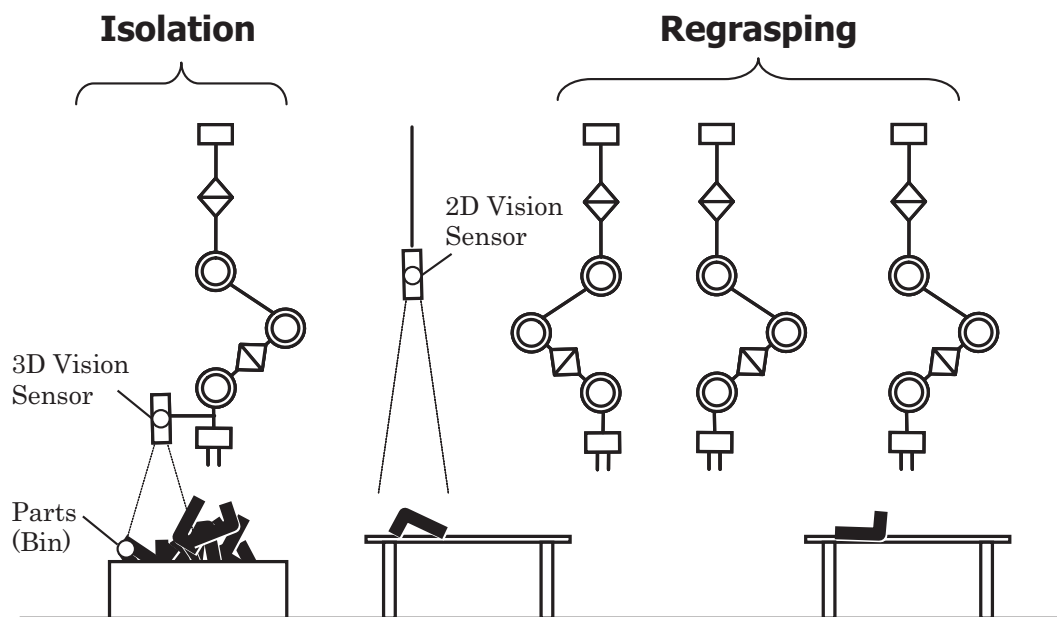


図 5.1 提案する一般化物体のビンピッキングシステムの構成

人並みの器用さはロボットにはなく、汎用的に物理操作を実現できない。この課題の解決は容易ではなく、従来のビンピッキングでは、円筒や平面状の部品、あるいは部品の内側面を内掴みするような「簡易に操作できる部品」が中心的に扱われることとなり、多品種供給への応用が困難であった。

著者はシステム論的アプローチとして、ビンピッキングをこれら2つの課題に対する2つの小問題と捉えることにした。1つめの問題は、バラ積み部品を汎用的かつ高速に取り出し、平面台に置く“単離”と設定した。2つめの問題は、未知の姿勢の単離部品を任意の姿勢への“持ち替え”と設定した。

これらの設定を擁する著者のシステム論的アプローチの概念を、具体的なシステムの形式で表現したものを図5.1に示す。同図では、4台のロボットのうち、最も左側に描いた1台で、1つめの問題、すなわちバラ積み状態から平面台に“単離”する作業を実施し、その平面台を中継点にして、残りの3台が部品を受け渡ししてパレット上に整列する2つめの問題、すなわち“持ち替え”の作業を実施することになる。バラ積み状態の部品の位置と姿勢は連続量である。その連続量を「平面上に静定させること」、「部品を把持すること」で有限個の状態に離散化し、

ロボットのセンサでその離散化状態を識別して、ロボットによる部品操作で離散状態間を遷移させられることが、提案方式が成功するポイントである。

このとき前者の問題には従来の高精度な物体の姿勢認識を応用できるが、前述のように計算コストがかかる。物体の姿勢認識をベースとする場合、さらに、周辺環境との干渉回避、ロボットの動作経路計画も必要になり、計算コストがかさむ。さらに、ロボット動作が複雑になり、作業ごとのタクトタイムにばらつきが発生しやすい。こういった点が作業のボトルネックとなる。そこで、新たに、「バラ積み部品の単離に特化した方法」[113]を開発した。その結果を本章4.1節において述べる。

また後者の問題には、専用設計のロボットやハンドを使うことも考えられるが、専用設計であることは部品形状に応じて再設計が必要になることから、本研究の目的とは相容れない。実用のためには、通常工場で使われるロボットアームやハンドの組み合わせで、ソフトウェアのパラメータを組み合わせることにより、部品を汎用的に持ち替えたい。そこで、汎用の装置を組み合わせたシステム部品の持ち替えるための個々の操作手順と、それらを用いて全体の行動を計画するアルゴリズムを開発した。これについて本章4.2節から4.5節において述べる。そして、それらによるシステム設計方法を本章5節において述べる。本章6節で実機実装例を示し、その評価結果を本章7節で述べる。

## 4. 開発したアルゴリズム

### 4.1 部品の単離

バラ積み状態の部品を単離するため、ロボットハンドモデル（以下、ハンドモデル）と距離画像をマッチングして、周辺と干渉しない把持可能なハンド姿勢を計算するアルゴリズムを提案する。この方法を応用する前提条件として、図5.2に示すように、4自由度の運動で把持を行なう。そして、把持のためのハンドの進入方向と、距離画像を計測するセンサの視線方向を一致させる。ただし、図5.2では鉛直下向き方向への把持アプローチを図示しているが、部品箱の形状や配置によっては、鉛直下向きとは異なる方向へ把持アプローチすることも可能である。

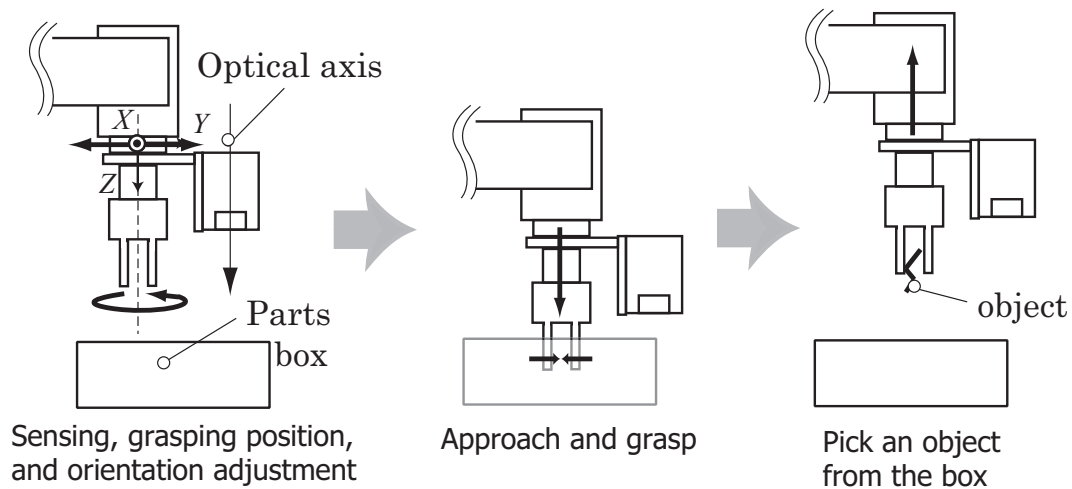


図 5.2 物体を把持するためのロボット動作と装置レイアウトの前提条件

従来の方では6自由度の動きで物体を把持していたところ、この方法では、部品箱に対して、まっすぐアプローチしていくような4自由度の動きだけで部品を把持することになり、実際それは成立するのである。

ハンドモデルは、物体と接触する領域（接触領域）、物体と衝突する可能性のある領域（干渉領域）をそれぞれ2値のマスク画像で表現する。図5.3に例を示す。ハンドは、二つの爪が連動して挟んでつかむ、産業用途として一般的なグリッパ型である。この場合、ハンドの爪の隙間を接触領域  $H_t^i$ 、爪の断面を干渉領域  $H_c^i$  と表現している。ここで  $i$  は、画面内回転に対応したハンドモデルの姿勢インデックスを表す。  $i = 1 \dots T_r$  で、パラメータ  $T_r$  によりハンドモデルの姿勢数が決まる。例に示すハンドの場合、180度の画面内回転をしたハンドモデル同士は一致するので、  $T_r = 3$  の場合、0度、60度、120度の3種類の姿勢のハンドモデルを利用するという意味になる。

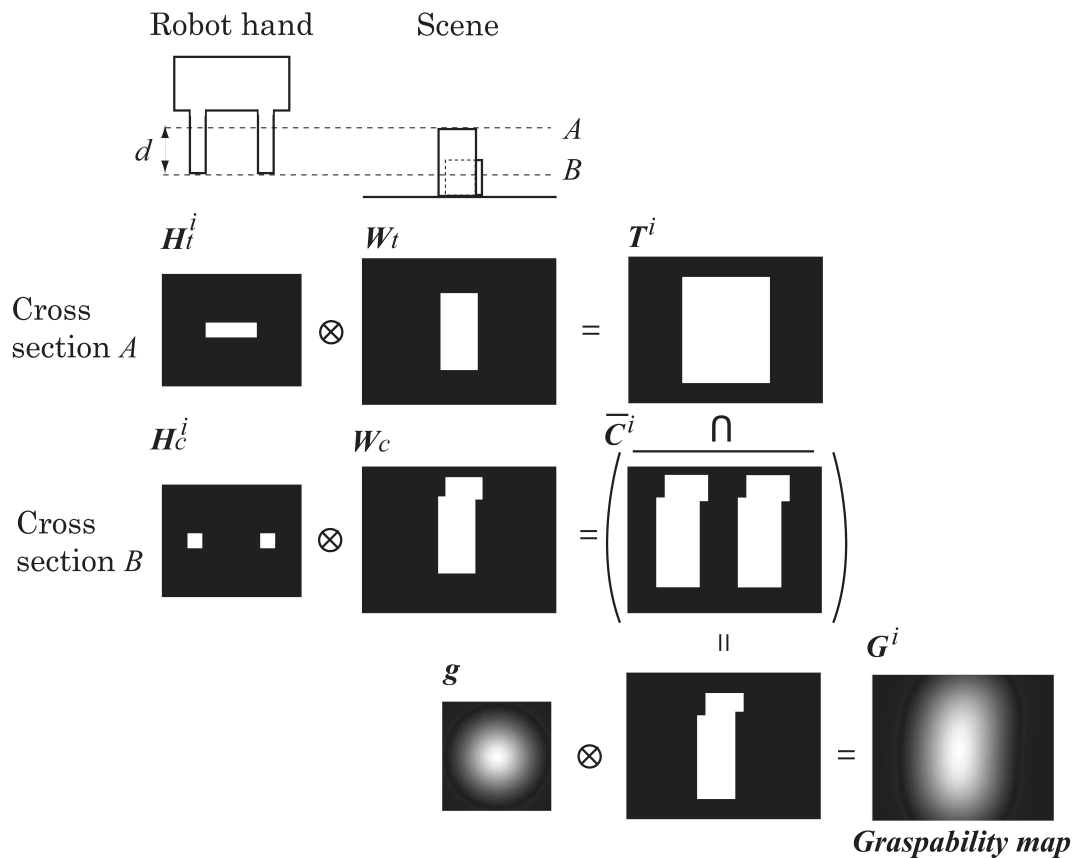


図 5.3 ロボットハンドと把持対象物体のモデルによる GraspabilityMap の算出方法 [113]

次に、シーンを撮影した距離画像から、対象となりそうな領域を選択する。図 5.3 では、距離画像の高い部分にある領域を抽出したものを候補領域  $W_t$  としている。この領域抽出は、例えばエッジベースのセグメンテーション後、最も高いセグメントを選択するなど、一般的な画像処理で実現する。接触領域  $H_t^i$  と、候補領域  $W_t$  を畳み込んだ結果  $T^i$  は、姿勢  $i$  のハンドが  $W_t$  と接触できる領域を示している。この領域内にハンドが存在すれば、候補領域が把持できる可能性があるが、ハンドと周辺環境が衝突する可能性を考慮できていない。挟持型のハンドの場合、物体を把持するため、候補領域の表面に対して、アプローチ時に一定の深さ  $d$  で進入する必要がある。そこで、ハンドが深さ  $d$  進入した場合の断面より高い位置に物体が存在する領域を  $W_c$  のようにマスク化する。この  $W_c$  と干渉

領域  $H_c^i$  を畳み込んだ結果  $C^i$  は、姿勢  $i$  のハンドが  $W_t$  を把持する際に、周辺環境と衝突する領域を表す。したがって  $(T^i \cap \bar{C}^i)$  は、姿勢  $i$  のハンドが物体を周辺と衝突せずに把持できる位置となる。

最終的に、 $(T^i \cap \bar{C}^i)$  に対してガウシアン  $G$  を畳み込み込んだ結果を GrapabilityMap と定義する。GrapabilityMap は領域中心に近いほど高い値を持つマップになるため、このピークを探索することで、基本的に物体重心位置に近く、安定把持がしやすい把持位置を高速に探索することができる。さらに各姿勢  $i$  に対して GrapabilityMap を計算し、その最も高いピーク値を探索することで、ハンドの姿勢も決定することができる。

なお、ハンドの形状寸法と深さ  $d$  などを、実測あるいは図面から得てモデリングすることで、吸着型や多指型など、様々な形式のハンドに対して、上記と同様な議論を展開できる [113]。

以上の計算により、単純な2次元の画像処理だけで、把持位置・姿勢、把持時の干渉判断をし、周辺環境との干渉回避、動作経路も効率的に設計することができる。詳細な定式、パラメータ、実装方法については [113] に譲る。

さて、この方法には、次のような利点がある。

ロボットが6自由度で動作する場合、ロボットと周辺環境が干渉するかどうかの予測は一般的には困難である。これに対し、4自由度で動作する場合には、ロボットに、予定している把持アプローチ方向の姿勢をとらせながら、部品箱の数点（四角い部品箱なら4隅）に、その爪先を移動させれば、周辺と干渉しないかが予測できるため、簡単に、安全な動作軌道を設計できる。いいかえると、把持アプローチが、部品箱に対して、まっすぐ進んで、まっすぐ戻るだけの動作になるため、ハンドと部品箱が干渉するケースが単純になる。この結果、部品箱の形状の設計、把持姿勢をとれる範囲のリミッター設定などに、教示作業者が現場で苦心する必要がなくなり、その負担が軽減される。このことは、部品形状によらず効率的に部品を単離するシステムの、迅速な構築に対して有利に働く。

また、基本的に同じ動きで把持アプローチするので、把持動作に関するタクトタイムがほぼ一定になる。これは生産システムとしての運用管理上の利点となる。

## 4.2 平面上の安定姿勢

部品が平面上でどのように静定するかを，3次元形状モデルをもとに計算する．

まず，部品の3次元形状を内包する最小凸包を求め，その後，部品の重心から最小凸包を構成する各面に降ろした垂線の足がその面内に入るとき，その面を下にした状態で部品は静定することがわかる．このことは，先行研究 [114] や，本研究に関連する共同研究 [115] に記載されていて，静定姿勢の計算手法自体については本研究は独自性・新規性を主張しない．

本研究の主張は，部品が実空間内でとりうる自由な位置・姿勢が，この現象により量子化され，量子化された状態がロボットシステムで識別可能であることを利用している点にある（なお，一連の共同研究でも静定状態計算をさらに発展させ，平面上に静定した部品のその姿勢における安定性についての理論的解析を行い，静定した状態からの操作に関する基礎理論を導出している [116–119]）．

理論的に導出された複数の静定状態と，現実世界における部品の静定状態をなかで識別する方法について本研究では，部品が平面上に静定した状態を平面の情報から眺めたときの部品外形形状の推定モデルと，実際に2次元ビジョンセンサで同じ状況を観測し，センサ情報処理により認識した部品の外形形状と推定モデルをモデルマッチング計算することにした．同じロボットシステムで同時に扱う部品について全ての静定姿勢と外形形状が識別できるかを計算し，識別できるものだけを識別してその先の工程にすすめることとし，識別できない状態はバラ積み整理工程からラインアウトさせればよい．具体的なラインアウト方法としては部品を静定させる平面を，いわゆる「ちゃぶ台返し」をして平面から部品を払い落とすための物理的な反転メカを作ることで容易に実現できる．

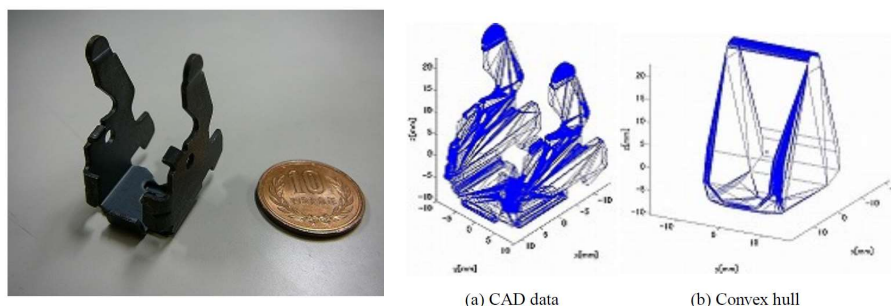


図 5.4 平面上の静定状態の計算のための最小凸包の計算例 [115]

### 4.3 部品の把持計画

部品をロボットハンドでどう把持するかの研究は様々な視点から研究開発が進められてきたが、把持姿勢の計画については 2010 年代にはいって、いくつかのエポックがある。例えばコロンビア大学のグラスデータベースや GraspIt! と呼ばれる把持計画ソフトウェア、その後の OpenRAVE の登場などである。これら一連の紹介は原田の解説記事 [120] に詳しい。

これらの研究は、多指ハンドによる把持計画について汎用性を保つことにも力点が置かれていて、産業用途で多用される構造が簡単な平行チャックハンドの場合には、得られる計画が冗長である。

そこで部品を把持する姿勢の計画について、本研究では次のようなアルゴリズムを開発した [121]。この把持計画の前提条件としてハンドは、平行チャックハンドのみとする。つまり、ハンドの指は 2 つで、それぞれが対向して同時に平行移動する動きをして、指の開閉動作を実行する。指先の爪は直方体形状とし、直方体の面が部品に接するものとする。これらのハンド形状は 3D モデルで与えられる。一方、部品についても 3D モデルで与えられる。ハンドの指を全開にしたとき、部品のどこかには全開にした間隙に納まる大きさの部位があるものとする。

1. 部品を構成する面について、平行チャックハンドの爪をかけることのできる平行面の組み合わせが存在する方向、かつ部品を構成する面に垂直な方向

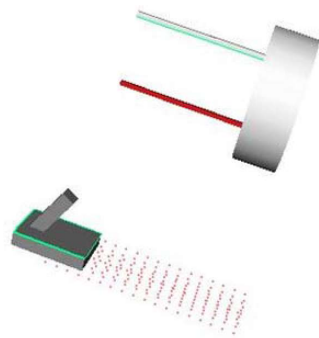
を選び出す。これにより，部品座標系で有限個の，部品とハンドの相対姿勢の集合が抽出される。

2. 各平行面について，ハンドの片方の指が平行面の片側を触ることのできる，ハンドの特徴点位置が存在しうる空間を部品座標系で求める。これを空間 A とする。ただし空間は離散化されたハンドの特徴点が存在しうる格子点の集合で表現される。
3. 各平行面について，ハンドのもう片方の指が平行面の片側を触ることのできる，ハンドの特徴点位置が存在しうる空間を部品座標系で求める。これを空間 B とする。これも格子点の集合で表現される。
4. 各平行面について，爪面以外のハンドのどこかが部品と衝突するときハンド特徴点位置が存在しうる空間を部品座標系で求める。これを空間 C とする。おなじく格子点の集合である。
5. 空間 A，空間 B，空間 C の和，差演算することで，平行チャックハンドで部品を把持できる空間 D を得る。
6. 空間 D の重心位置を，最終位置として出力する。

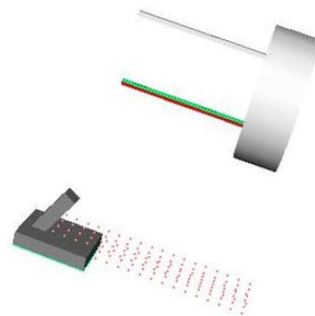
以上の計算を，上述の部品の平行面の組について実行することで，その平行チャックハンドでその部品を把持可能な部品とハンドの相対姿勢が複数得られる。上述しているように各量子化の距離は，設計パラメータとなる。図 5.5 は，以上の計算を図示したものである。

また，後述する「持ち替え」において，平面上に静定した部品をつまみ上げる行動を計画する際には，空間 C をハンドと平面が干渉する現象を加味して拡大すれば良いし，2 台のロボットで部品を受け渡す際も，同様に空間 C の制約を変更すればよい。

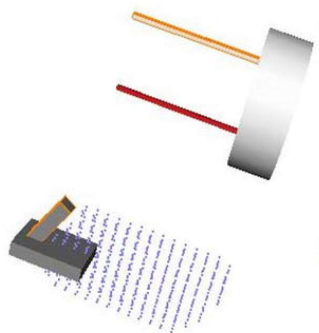




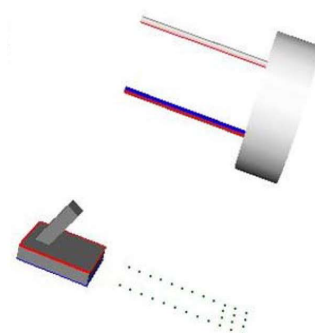
(a)上爪面とワークの  
接触空間A



(b)下爪面とワークの  
接触空間B



(c)爪面以外とワークの  
接触空間C



(d)把持が成立する空間D

図 5.5 物体把持計画 [121]

#### 4.4 部品の持ち替え：持ち替えグラフアルゴリズム

単離した部品の姿勢から、あらかじめ定められた一意の把持姿勢に部品を持ち替えるための、持ち替え行動の事前計画アルゴリズムについて述べる。この方法では、図 5.6 のような、ロボットハンドと部品の 3 次元形状モデルと、それぞれの干渉判断が可能な物理シミュレーション（例えば OpenRAVE など）を利用する。事前準備として、物理シミュレーションに基づき、部品を床に単離したときに、部品がとる安定姿勢を計算しておく。さらに、部品に対して把持可能なロボットハンドの姿勢も計算しておく。それらをノードとすることで、図 5.7 に示す状態遷移グラフを作成することができる。この状態遷移グラフのノードには、前述した部品の安定姿勢（図中の  $\text{sf0}$  ～  $\text{sf8}$ ）と、部品に対するロボットハンドの把持可能な姿勢（図中の  $0$  ～  $19$ ），さらには、最終的に整列する際の部品に対するロボットハンドの把持姿勢（図中の  $0$  ～  $19$  のうちの  $3$ ）が含まれている。各ノード間の直線は、ノード間の状態遷移が実現すること、つまり部品把持からの姿勢変更や持ち替えが実現することを表している。ノード間の状態遷移が実現しない場合はその直線も存在していない。経路点数を最小化するグラフを作成することで、最も効率的な作業工程を設計できる。

なお、平面上でとりうる姿勢の導出から、グラフの作成に至るまでの計算は自動で行われ、その計算時間は、例えば図 5.6 のデータでは、2014 年時点で 10 万円程度の Windows PC にて、10 数 [min] 程度である。また、モデリングしきれないケーブルと環境の干渉の都合により、自動計算で得られた安定姿勢やグラフ上の、実際には実現不可能なノードを、手作業で減らして、途中から再計算することが可能である。

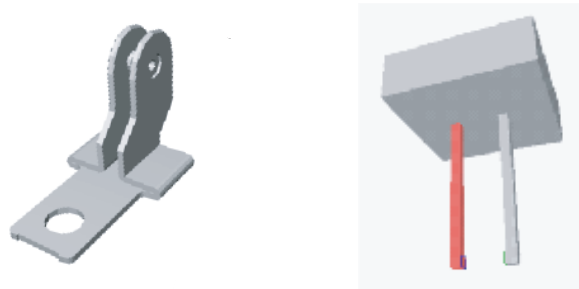


図 5.6 持ち替え計画の計算に用いた部品とロボットハンドモデル例

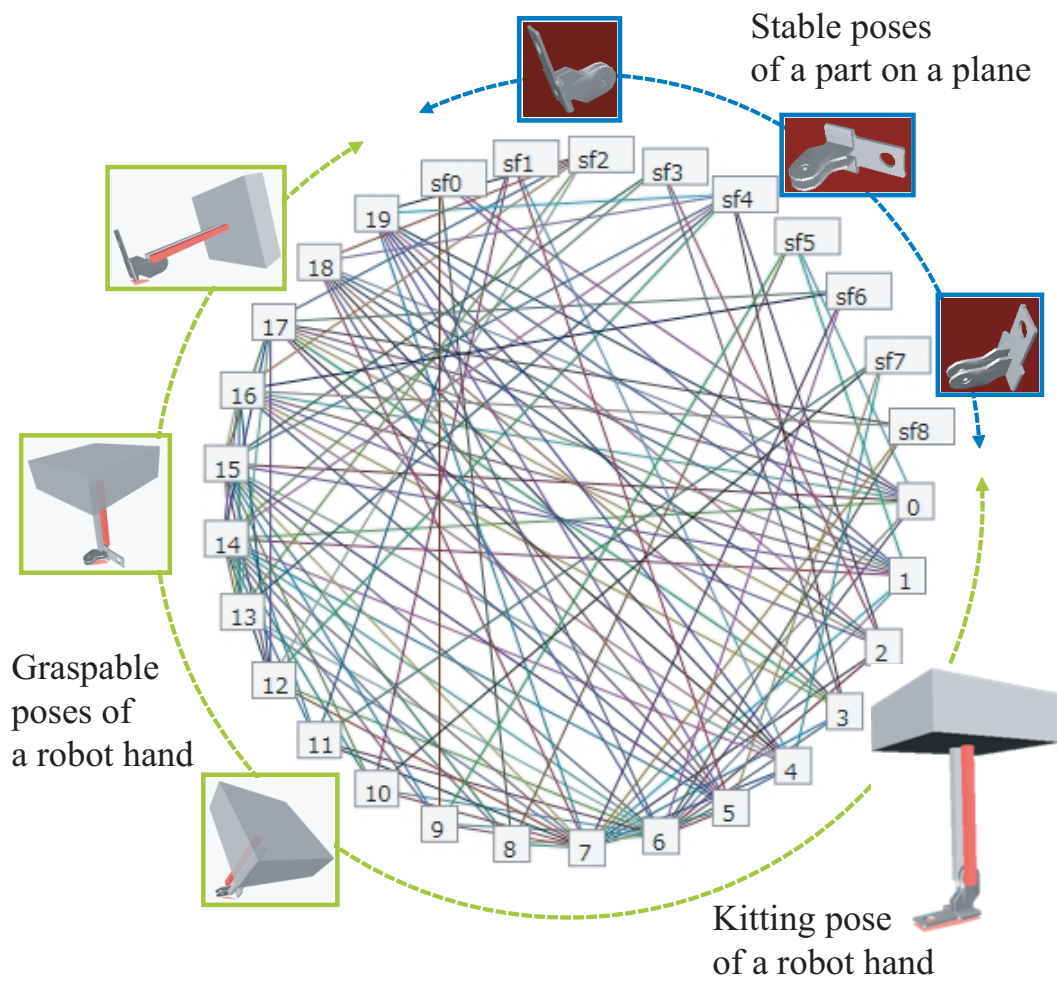


図 5.7 持ち替え計画の計算結果による状態遷移モデル例

## 4.5 部品の転がし操作

持ち替えグラフを利用すると、ある静定状態からある把持姿勢に複数手の持ち替えて到達できるか否かを計画できる。このとき、持ち替え手数が多くなるケース、到達不能であるケースが、実際のシステムを組んだときに問題になる場合がある。

平面上で、部品をロボットハンドについて転がし向きを変えることで、持ち替え手数を減ずる、あるいは到達可能にすることができる場合があり、これを関連研究 [122, 123] で開発中である。図 5.8 は、平面上で部品をハンドで突ついたときに成功する状態遷移を抽出したものである。この図の例では平面と部品の摩擦係数  $\mu$  の変化で結果が異なる様子が示されている。今後、この機能の効果検証を進める予定である。

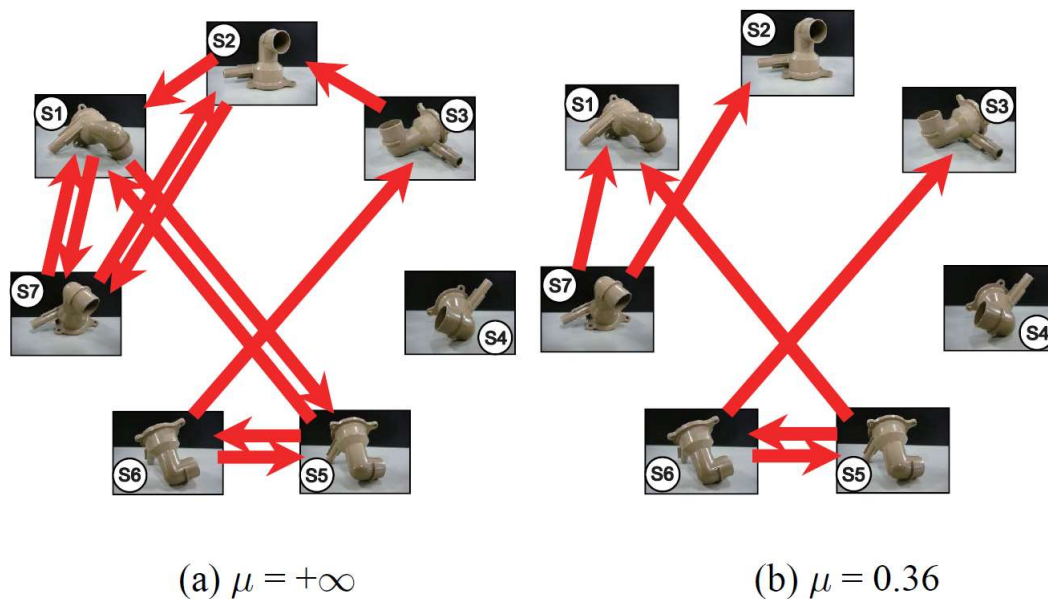


図 5.8 物体の平面上の転がし操作計画 [123]

## 5. 一般形状部品の多品種供給システムの設計方法

前節までで述べた部品の単離，部品持ち替えにより，一般形状物体に対応する部品供給システムを構築することができる。前掲の図 5.1 は実際のシステムでもあり，4 台のロボット，3 次元ビジョンセンサ，2 次元ビジョンセンサを使った例である。左から順に 1~4 台目のロボットとする。このとき 1 台目のロボットと 3 次元ビジョンセンサが，バラ積みされた部品を平面台上に単離する。平面台上に設置した 2 次元ビジョンセンサが部品の姿勢を認識する。そして，2 台目から 4 台目のロボット間で部品を持ち替えて，部品を整列できる姿勢にする。最終的に，4 台目のロボットが任意の姿勢に部品を整列する。

このシステム例では，部品の姿勢不定性が大きい状態への対峙から，高精度に位置決め整列された状態へ，段階的にプロセスが進んでいく。状態を変化させる作業を，複数の工程に分割してパイプライン処理することにより，例えば，バラ積み状態から物体の姿勢認識をして直接整列に持っていくシステムで発生するボトルネックを解消する効果が得られる。具体的には，実行の並列化による時間短縮（姿勢認識計算，ロボット動作），システム構築時の作業によるエンジニアリングの単純化（ロボットと環境の干渉判断，把持・組み付けのための位置決め動作教示，搬送経路計画）の効果が得られる。また，各工程の処理が単純な計算と動作になるので，タクトタイムが変動しにくい。このことは，部品の形状が変わっても，高速に安定した処理を実現しやすいということにつながる。

一方でシステム構成としては比較的大きくなるというデメリットもある。このシステム例と同じ処理（単離と持ち替え）はロボットが 1 台でも 2 台でも実現可能であるが，その分，計算・動作時間がかかる。本章で後述するが，アプリケーションの要求仕様に応じて最適な構成は変わる。なお，ロボットの台数を減らす方策として，関連研究（例えば [123]）において，平面上の部品をロボットハンドで小突いて転がして姿勢を変える操作を使うことで，持ち替え必要数を減らし，動作時間を増やさず台数を削減する試みに取り組んでいる。

以上の構成における処理詳細を示す。3 次元ビジョンセンサを搭載した 1 台目のロボットが，前述した部品単離アルゴリズムをもとに，部品供給箱上のバラ積み部品を一つ取り出し，平面上に単離する。複雑な形状をした部品でも平面上に放

置すれば、とりうる姿勢は少数の有限個にとどまる。平面上に設置された2次元ビジョンセンサにより、部品の姿勢を認識する。これを持ち替え行動の事前計画アルゴリズムの初期姿勢とする。次に部品を整列するときのロボットハンドと部品の関係を最終姿勢として、状態遷移グラフの最短経路をとるように持ち替え計画を計算しておく。平面上で部品がとりうる姿勢に対して計算された持ち替え計画の動作をするよう2台目から4台目のロボットに事前教示しておく。これにより、平面上で物体の姿勢が認識され次第、持ち替え・整列を実行できる。原理上は、モデルのある剛体であればどのような形状の部品でも操作することができる。

また、ロボット群の配置レイアウト設計については、関連研究（[124]など）において、多目的最適化手法を研究開発している。

このシステムは、複数の異なる複雑な形状を持つパーツを高速に整列することを目的に構成している。ロボットの台数を減らしても同じ作業を実現できるが、持ち替え動作の並列度が下がるため、作業時間が上乘せされる。扱う部品が平面状、円筒状のような単純形状の場合、従来のおりの部品姿勢認識のほうが効率的である。そこで著者らは、図5.9のような使い分けを行なう。部品形状が単純な場合は、姿勢認識 [107] に基づく直接整列を行なう。部品形状が複雑な場合は、単離と持ち替えを通じて整列を行なう。多品種の部品供給において複雑な部品形状が含まれる場合も、後方でシステムを構築することで、すべての部品種に対応できる。

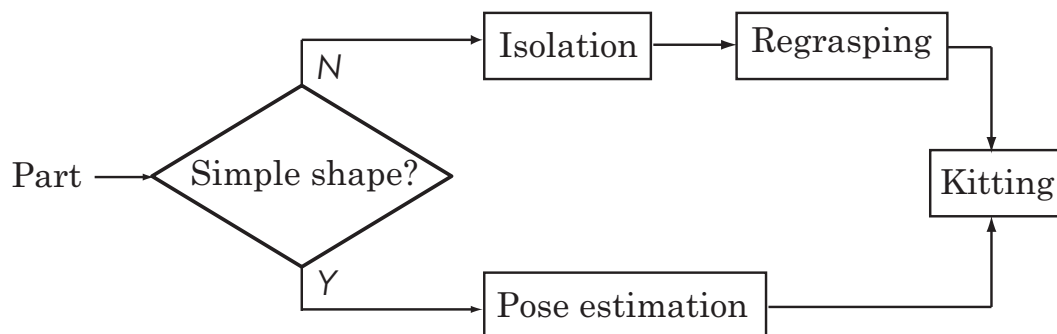


図 5.9 ビンピッキングのためのシステムの使い分け方法

## 6. 構築した実機例とその動作

ここまで説明したシステム論的アプローチと要素技術に基づき、著者らは実際に図 5.10 に示すシステムを構築した。産業用遮断機の部品 11 種類をそれぞれ供給箱に無作為に入れ、システムにより、取り出しからキッティングトレーへの整列を自動化した。扱う部品を図 5.11 に示す。これらは、ある実製品を構成する部品で、質量は数 [g] から数 10 [g] であり、円筒、平面状のものから、突起のあるもの、多面体のものなど、複雑な形状を含む。ロボットは 1 台目が三菱電機製 RV-6SD, 2~4 台目が同 RV-2SQ, 3 次元ビジョンセンサが同 MELFA-3D Vision, 2 次元ビジョンセンサが Cognex 社製 EZ-140, ロボットハンドが 1 台目が TAIYO 社製電動ハンド ESG1, 2~4 台目が Shunk 社製エアハンド RH806 相当品, である。スペックは各社情報を参照されたい。なお、エアハンドは、生産現場で一般的に使われているタイプを使用した。1 台目を電動ハンドにしているのは、把持エラーを把持の瞬間に判定する実験のためである。エアハンドを用いた場合でも、平面上の 2 次元ビジョンセンサによる認識の段階でエラー検知はできる。

本システムにおいて、対象ワークの形状に特化した特殊な設計を行なうことを極力避けた。具体的にはロボットハンドの爪について 1 台目のロボットのものをワークが取りやすいように比較的長く設計した程度で部品に特化した設計はしていない。またロボットとキッティングトレー以外のジグや機構を使っていない。部品箱は生産現場で一般的に部品供給に使われるもので、仕切りにより分けられた各区画に各部品が種類ごとに無作為に投入されている。キッティングトレーは部品を組み付ける溝やピンが配置されているが生産現場で使われるキッティングトレーの一般的設計例である。

このシステムを動作させたところサイクルタイムは 3.26 [s] ~ 15.40 [s], 平均 6.45 [s] で整列できた。サイクルタイムの変動には作業ミスが発生した場合の時間を含む。今回の実験において作業ミスが発生してもパーツフィーダのように人手が介入しての復旧すなわちチョコ停は発生せず、作業時間が伸びるだけであった。実験中の最速サイクルタイムは 3 [s] 未満で、操作の簡易さから、平面状のワークが最速であった。

以上の結果の妥当性を、次節にて、従来方式との比較をしながら、検証する。

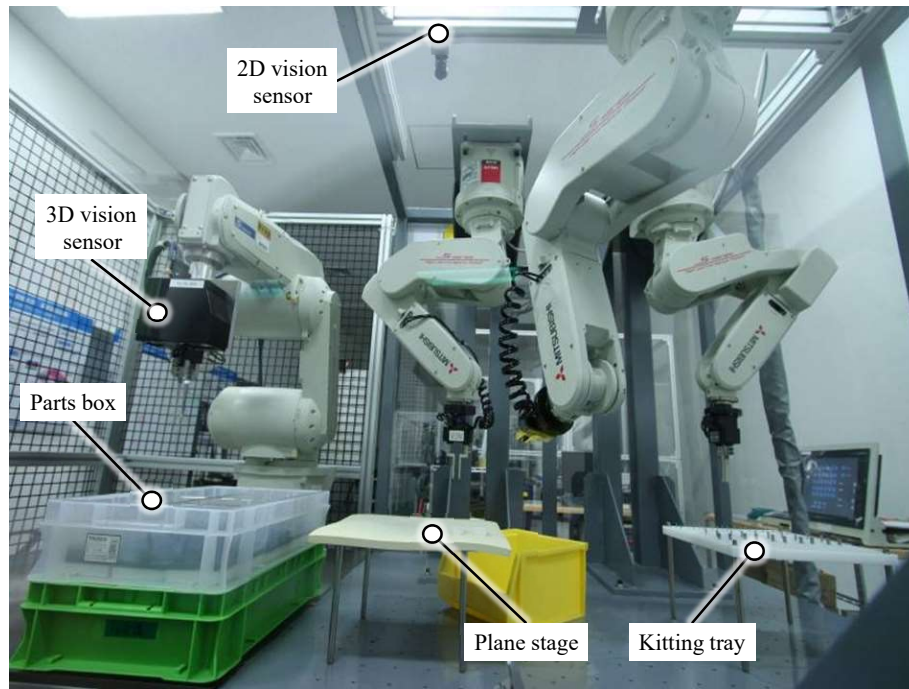


図 5.10 ビンピッキングの実証システム



図 5.11 実験に用いた 11 種類の様々な形状・材質の実部品



表 5.1 従来手法とのシステム性能比較 (N は部品の種類の数)

|                              | 提案システム       | パーツフィーダ              | 従来のロボットシステム | 作業者                                     |
|------------------------------|--------------|----------------------|-------------|---|
| 汎用性                          | OK           | NG                   | NG          | OK                                      |
| サイクルタイム (second)             | 3.26 ~ 15.40 | 1~2.5                | 3.5~10      | 1~3                                     |
| コスト (円)                      | 10,000,000   | $N \times 1,000,000$ | 3,500,000   | 1,500,000~2,000,000 (per year)          |
| Lead time for product change | 3 days       | 1 month              | 2 weeks     | 1 hour (starting), 2 weeks (mastership) |

## 7. 他方式との比較実験結果

ここでの比較対象は、同じ作業を人手でおこなった場合、パーツフィーダを使ってシステムを構築した場合、バラ積み状態からの物体認識をもとに持ち替えなしで整列する場合（従来のロボットシステム）の3つとした。比較の観点は、実際の生産現場で評価される汎用性、サイクルタイム、コスト、再立ち上げ時間とした。以上に基づく比較結果を、表 5.1 に示す。なお、従来のロボットシステムにおける物体の姿勢認識には Drost の手法 [106] を利用した。以下、表 5.1 の内容を説明する。

汎用性は、部品の形状に対する汎用性を指す。パーツフィーダは、複雑形状の部品において、整列部分が設計できないものが発生した。また、従来のロボットシステムの場合、持ち替えができずに対応できない部品があった。人手と提案手法ではすべての部品の供給ができた。なおパーツフィーダや従来のロボットシステムで扱えるのは全部品種類の 50% 以下であった。

サイクルタイムは、一つの部品を取り出して整列するまでの作業時間の平均を指す。提案手法、従来のロボットシステムには、作業失敗の場合もサイクルタイムに加算している。また、パーツフィーダ、従来のロボットシステムでは、全ての部品は扱えないので、扱える部品のみ平均的作業時間で判断している。結果、人手が最も高速であった。自動化手法では、同じ部品で比較した場合、パーツフィーダが最も高速で、続いて提案手法、従来のロボットシステムという順番であった。従来のロボットシステムは物体の姿勢認識、把持のために生成される

ロボット動作それぞれの時間が部品形状によって大きかった。提案手法は、多手法が扱えない複雑形状の部品を扱う場合に、処理時間が大きくなる傾向があった。

コストは、システム構築に必要な初期投資を指す。扱う部品点数が少ない場合、パーツフィーダが最も安価な自動化方法になる。しかし、およそ10種類以上の多品種部品を扱う場合は、従来のロボットシステム、提案手法のほうが安価になる。

なお、中長期的に必要なコストの一種として、取り扱う部品の数が増えていくと、パーツフィーダは、追加機の設置床面積、あるいはボウル、ツーリングの保管場所体積が増加する。提案方式と従来のロボットシステムは、基本的にソフトウェアが変化するのみで自身の設置床面積は変化しない。

再立ち上げ時間とは、生産機種の変更などにより扱う部品形状が変わった場合に、システムを再立ち上げするのにかかる時間を指す。提案手法では、持ち替え作業計画アルゴリズムに基づき、持ち替え作業を再教示する部分を中心に、部品を1点追加するのに約0.5日、トータルで例えば約3日であった。人手の場合、作業自体は約1時間で立ち上がった。しかし、サイクルタイム並みの速度で作業ができるよう習熟するまでに、概ね2週間の実地経験が必要であった。従来のロボットシステムではソフトウェアの再調整、特に物体認識～整列までの教示・調整に多くの時間がかかった。これは、ロボット単体で複雑な作業を実現するのが難しいのと、持ち替え作業計画アルゴリズムのような教示支援方法がないことによる。パーツフィーダはハードウェアの再設計が必要なため、最も時間がかかっている。

以上をまとめると、簡単な形状の部品を10種類以下で扱う場合は、パーツフィーダによる自動化システムの効率が良く、複雑形状を含む一般形状部品を多品種扱う場合は提案手法が最も効率が良いという結果となった。そのため、提案手法は生産機種を切り替えながらのマスプロダクションに対して特に有効であり、少量生産では人手やパーツフィーダのほうが効率的である場合がある。ただし少量生産でも部品が複雑な場合は、パーツフィーダは利用できないが、提案手法は応用可能である。

提案手法のサイクルタイムは、現段階で人手による生産速度よりも遅いことから、場合によっては、自動化の検討に進めない懸念がある。こういった場合には、

夜間に提案手法により整列済パレットを多数作っておく，システムを並列動作させるなどの運用上の対策を実施することで，自動化の検討を進められることを付記する．

## 8. まとめ

本章では，バラ積み部品供給の自動化において同じシステムで複雑な形状の部品を多品種取り扱うという課題に対し，問題の困難性の本質を再考したシステム論的アプローチに基づいて，バラ積み物体の単離，持ち替えという工程でビンピッキングを実現するための要素技術とシステム構成法について述べた．これら一連の仮説に対し，実験結果は，従来手法が扱えない複雑形状を含む全 11 種類を，本手法は操作できることを示している．つまり，その形状モデルのデータが存在する剛体について，一般形状の物体の自動供給をはじめて実現しうる可能性がある．また，本章では，本手法の適用可能範囲と，他手法との使い分けに言及した．



## 第6章 開発成果のまとめ

本研究で開発した「技術プラットフォーム：自律型セル生産ロボットシステム」を用いて、実製品組立を実行するセル生産の「実証システム」を構築した結果をまとめる。

### 1. 技術プラットフォーム

繰り返し述べている通り、本研究においてはロボットの自律化や知能化の産業価値は生産システムにおけるエンジニアリングコスト削減に主眼がある。本論文では、機種切替時の作業コストの低減、運用時の長時間連続操業によるロスコスト削減を実現し、生産現場へ導入できるロボットセル生産システムの実現を目指した。最終的に統合システムにおいて、教示時間短縮3分の1以下となること、ロボットが同一動作を繰り返す際のタクトタイム短縮し44%となったこと、チョコ停（装置故障を伴わない生産停止）が発生した際には多少のタクトタイムの増加を伴いながら自動復帰動作を実現すること、10種以上の小型電気電子製品を構成する部品についてばら置き状態からの組み付け実行が実現すること、をそれぞれ確認した。

なお、これらの評価項目は一連のロボット知能化プロジェクト当初に最終目標課題として掲げていたものである。もちろん、これは普遍的な評価ではないことは認識している。そこで現在、開発成果を徐々に製品展開して市場評価を収集中であり、その様々な条件の下での普遍的な効果を確認中である。

さて、開発した技術プラットフォームは、複数の知能モジュールから構成される。これらはインテグレーションソフトウェアにより実際のハードウェア群と結合され、生産システムが構築される。図6.1にその一例を示す。以下、(1)～(5)の

各群に分類しながら各モジュールの内容を簡単に振り返る。なお、各知能モジュール名について「A モジュール」は単に「A」と記す。また、ロボットが存在しない状態や稼働停止中の状態をオフライン、ロボットを目前にしての作業をしている状態や実稼働中の状態をオンラインと称している。

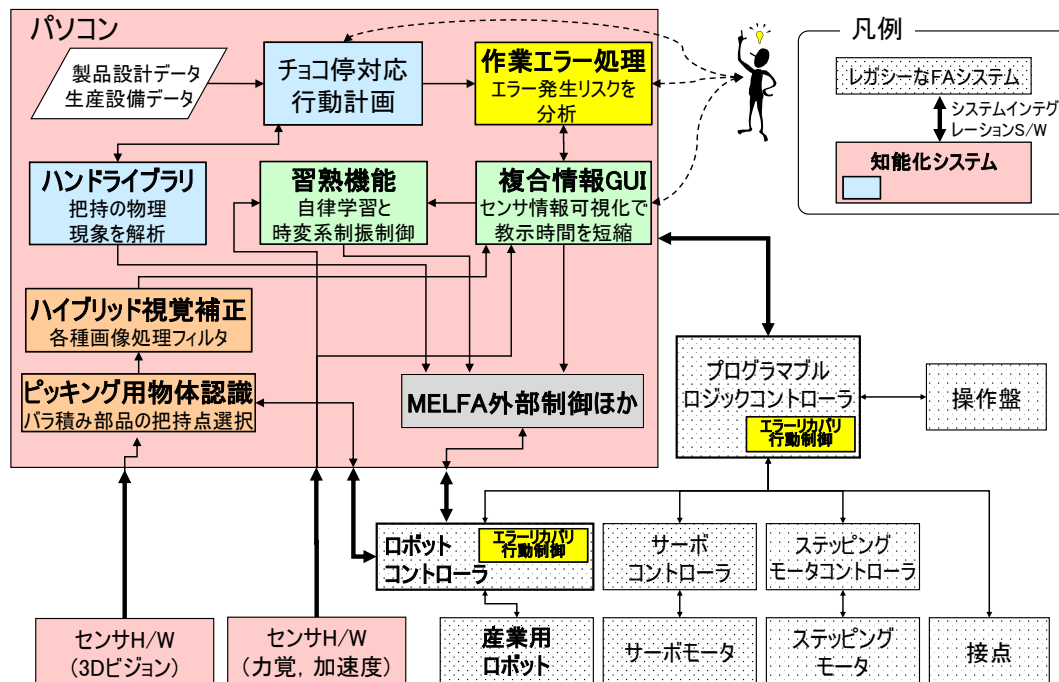


図 6.1 開発した要素技術による自律型セル生産ロボットシステムのインテグレーション

### 1. システムインテグレーションソフトウェア群

手堅く作られたレガシーな FA システムに、新しい価値をもたらす智能モジュール・サブシステムを追加することで、産業用ロボットシステムが従来通りの安定性を保ちながら進化する。このような考え方で統合システムを構築することにし、新旧のサブシステム間を結合するためのシステムインテグレーションソフトウェアを開発した。図 6.1 の太い矢印の位置で本ソフトウェア群が動作する。

### 2. オフライン教示支援智能モジュール群

- チョコ停対応行動計画：生産システムの作業動作計画，複数台ロボットの作業工程のインタラクティブなプログラミングが課題。複数台ロボット動作のシミュレータ，ブロックプログラミングツールをモジュール化。

- ハンドライブラリ：ロボットハンド設計製作と制御則プログラミングを迅速化し，さらには手戻りを減らすことが課題．ロボットハンドにより対象物を把持する際の過渡現象を解析し，対象物体の初期位置ズレとハンドの指の制御方法との関係から，把持の成否を推定するモジュール．
- 作業エラー処理：作業エラーを抑制できる作業シーケンスのプログラミング手法およびエラーリカバリプログラムを理論的根拠に基づいて得ることが課題．ロボットの作業中に生じるチョコ停に対して，ベイジアンネットワークを時間的に拡張したダイナミックベイジアンネットワーク (DBN) を用いて，ロボットのエラー状態 (あるいは非エラー状態) を動的に推定・解析するモジュール．

### 3. オンライン教示支援知能モジュール群

- 複合情報 GUI：ティーチングボックスと呼ばれるリモコン箱を手に持ったオペレータがロボットに動き方を教える教示作業は，作業者の技量に依存して，作業時間と作業結果が変動し，そのノウハウレス化，時短化が課題．作業の進行状態を作業者に知覚させることで作業者の技量に寄らず作業時間を短縮し結果を安定化させる GUI を，ヒューマンインタフェース理論に基づいて導出したモジュール．
- 習熟機能：同一作業を繰り返すことで，その作業に習熟していく人間作業者の能力をロボットに取り込むことが課題．タクト短縮が可能な軌跡等を求める独自の能動型探索アルゴリズムで構成したモジュール，および独自のインプットシェーピング法により，時変系の多軸同時高速動作後の残留振動を抑制する制御指令を生成するモジュール．

### 4. エラーリカバリ支援知能モジュール群

- 複合情報 GUI：ロボットシステムの復旧・移設・複製・改修時に，エンジニアリング時間を短くすることが課題．作業対象物に対し，ロボットがどのような相対位置姿勢にあるかを，センサ計測値を処理して作



業者に知らせることで、ロボットの故障時、移設時、複製時などにおける再教示作業時間を短縮するモジュール。

- エラーリカバリ行動計画：チョコ停からの自動復帰が課題。チョコ停発生を検知した後、自動復帰行動を生起するロボット言語あるいはシーケンスプログラム中のモジュール(スタブ)。ここから復帰行動に分岐して、タクトタイムの増加を伴いながらも自動復帰することを確認。レガシーな FA システムのなかで動作する。

#### 5. 認識支援知能モジュール群

- ピッキング用物体認識：組立対象部品のバラ積み供給において、バラ積み状態の認識が課題。独自の3次元ビジョンセンサからの距離画像情報を処理して、バラ積み部品をつまみ上げるための把持点を認識するモジュール。
- ハイブリッド視覚補正：ビジョンセンサを用いた認識プログラミングの迅速化が課題。3次元ビジョンセンサから得られた距離画像と2次元画像情報を融合し、教示作業に有効な作業ポイントを対話的に提供するモジュール。

## 2. 実証システム

ロボットによるセル生産の典型例として、以下に示す2つの実証システムを構築した。これら2つのシステムの関係であるが、工場内で、「バラ積み部品供給システム (図 6.3)」が、部品トレイ上に部品を整列させ、その部品トレイが「自動組立システム (図 6.5)」に搬送され、部品から製品へと組み立てられる、というような普遍的かつ汎用的なセル生産ラインレイアウトを想定している。

自動組立システムでは、図 6.2 に示すような、複数の工程が、ひとところで実施される。これは、まさにセル生産システムである。

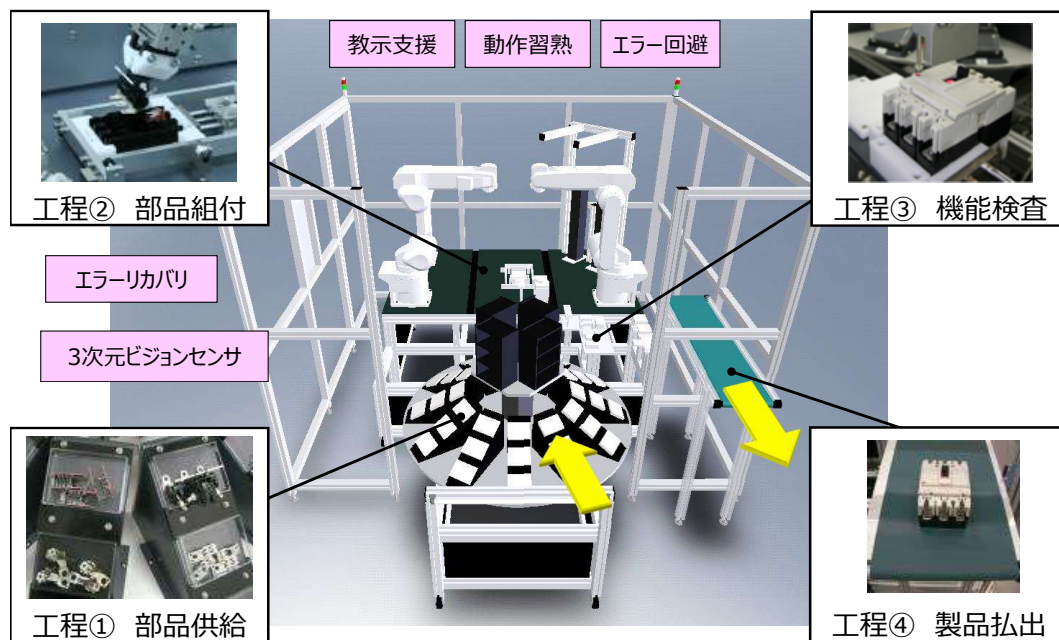


図 6.2 セル生産 (組立) の工程例

## 2.1 自動組立のためのバラ積み部品供給システム

自動組立システムでは，部品供給が問題になる．現状は，パーツフィーダと呼ばれる専用装置，あるいは人手に頼っており，機種切替時のリードタイム，コスト，フットプリントなどに問題がある．これまで，ロボットによる自動化も試みられてきたが，最近ようやく吸着パッドで吸い付けられる板金物，軸物，穴があって内掴み可能なものなど，部品の形状を限定した上での実用化が見えてきた段階にある．そこで今回，ロボットによるバラ積み部品供給を，統合システムの例題とした．

図 6.3 に構築した統合システムを示す．4 台のロボットを用いて，バラ積み部品箱内の部品を外掴みで取り出した後，部品を持ち替えて，その姿勢を整え，太線枠内に示した部品トレイ上に整列させていく．人手に頼るしかなかった複雑な形状の部品を，最速 3 秒台で整列させられることを確認した．このとき，ロボット台数とタクトタイム，対象部品形状の複雑さには関係があることもわかった．このシステムで，複合情報 GUI，ピッキング用物体認識，習熟機能，作業エラー処理，ハンドライブラリなどの知能モジュールを用いてシステムを構築，バラ積み部品整列機能が実現するとともに，システムに対して，立上，稼働，移設，復旧の局面を与えたとき，各知能モジュールにより「機種切替が迅速」「長時間無人連続運転可能」の各機能が発揮されることを確認した．

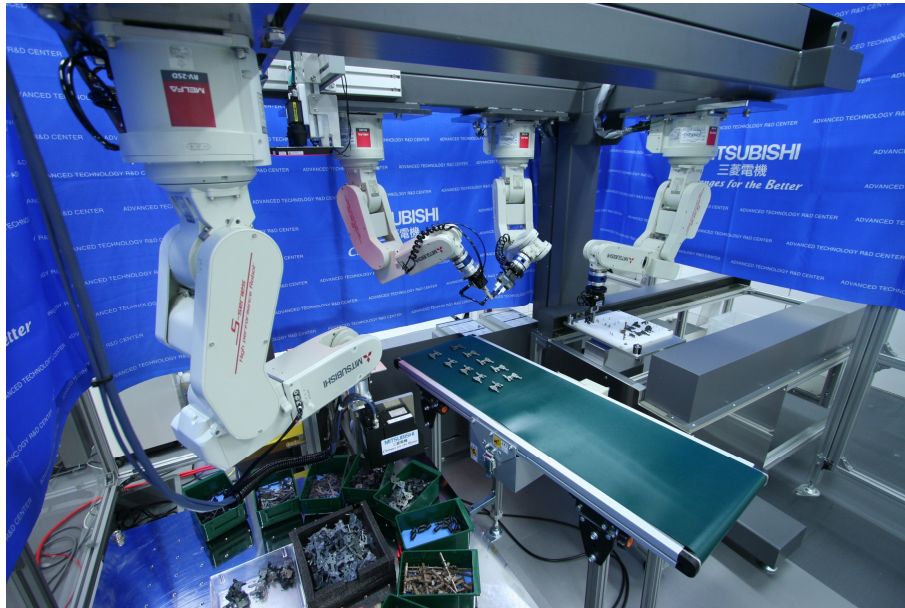


図 6.3 バラ積み部品供給セル

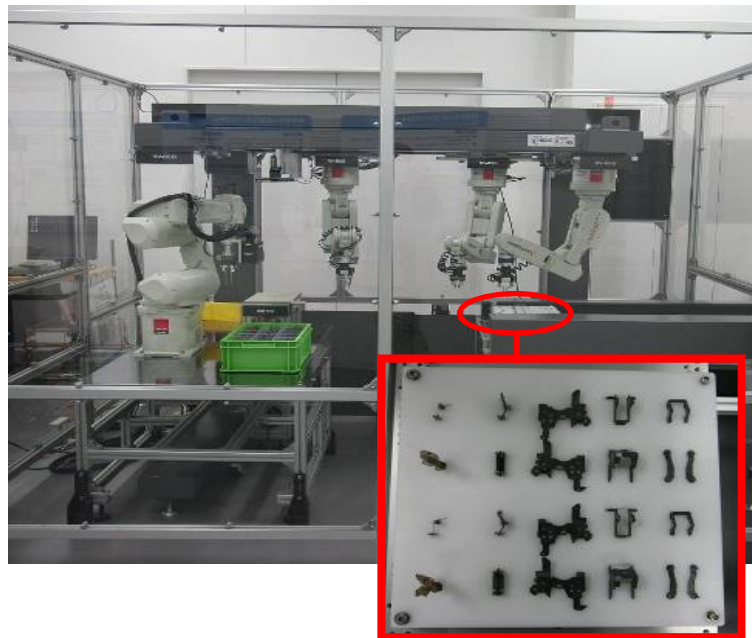


図 6.4 部品トレイの例

## 2.2 自動組立システム

約10cm立方から60cm立方程度のサイズの、部品点数 数10点程度、多品種中量生産の電気電子製品の自動組立を例題にした。図6.5にその写真を示す。この統合システムにおいて、部品の供給(図中手前の円盤状の部品台車から)、組立、検査、払出(右下のコンベアによる)という一連の生産活動が、1カ所で実施される。すなわち、典型的なロボットによるセル生産システムの構築事例である。図4に、対象ワークの例を示す。この統合システムでも、先のシステムと同様に知能モジュールによる効果を確認した。



図 6.5 ブレーカ組立セル



図 6.6 組立対象製品の例

### 3. 実証結果

教示機能について、従来手法を用いた教示作業時間に対し、教示作業時間が従来手法の1/3以下になった。かつ同一作業を繰り返すときのタクトタイムが初期状態に比べて10%短くなることを確認した。

チョコ停を誘発する頻度が高い原因(規定外のワークの混入、位置ずらし等)を、人為的に検証システムに与えたとき、チョコ停止の事前回避あるいは多少のタクトタイムの増加を伴いながらも自動復帰が実現することを確認した。

### 4. 考察

実証システムにより、技術プラットフォーム「自律型セル生産ロボットシステム」の有効性が、個別要素技術ごとに確認された。

開発した技術プラットフォームは、統合的にも適用可能であるし、要素機能を分解しての使用も可能である。このため、セル生産ロボットシステムを構築する際のみならず、従来型のライン生産の単工程を担当するロボットに対しても適用可能である。

これにより、生産現場でよく使われるワークのロード・アンロード作業、コンベアトラッキング、力覚応用、バリ取りなどのシステムを迅速に構築するためのキット(ハードウェアとソフトウェアのセット)を「アプリケーションパッケージ」と銘打った製品展開 [125] が進められている。

これらの製品展開により、特に電気電子の組立アプリケーションについて市場評価が高まりつつあり、このことは当初目的を総合的に満足する効果が確認されつつあることを示していると考えられる。

# 第7章 結論

## 1. おわりに

製造業の分野における社会的要請に応えるため、本研究では、ライン方式と人セル方式のそれぞれのメリットを集めたセル生産ロボット方式を実現するためのロボット知能化技術の研究開発プロジェクトを牽引・推進してきた。その研究過程においてロボットが備えるべき知能として、とりわけ自律性が重要なことがわかった。よって本論文では製造業向けロボットに自律性を付与するシステムアルゴリズムを開発したことを述べた。

本研究全体を特徴づける独自性は、上述のごとくの志にもとづいて、そもそも何がどうなっていたらよいのかを原理原則に立ち返って探究していること、ロボット知能化技術を器用なロボットを創るために使うのではなく、ロボットを使った生産システムのエンジニアリングコストを削減するために使っていることにある。

まず、人の自律性を特徴づける試行錯誤による習熟について、これを『未知の目的関数の最適化問題』であると定式化して解く独自の「能動型探索アルゴリズム」を提案した。これは、そのチャレンジングな問題設定自体と、アルゴリズムの計算の中で、“その試行で得られるであろう情報量・情報が得られないリスク量を同時に考慮して次の試行を最適化する”点とに独創性がある。実証実験の結果、熟練者の調整で動作時間 1045.33[ms] のロボット軌道が、約 56% の 590.22[ms] に短縮したこと、さらに熟練者による加工機制御パラメータ調整などの同種の困難性を持つ問題への水平展開に成功したことで、有用性が示される。

次に、組立工程で切望されるバラ積み状態の部品供給を解決するための自律システムアルゴリズムに取り組んだ。これはロボット学にとって『ランダム・ビン・

ピッキングと呼ばれる未解決の古典的難題』である。この問題を複数の小問題に分割して解くシステム解法と、作業計画の試行錯誤を実行する「持ち替えグラフアルゴリズム」とを提案した。これらは、よく知られた難題に対して既に有りそうで実は無かった解き方を導入した点と、実行可能解を網羅的に算出する点とに独創性がある。実証システムにおいて、数 [g] から数 10[g] の質量の実際の小型電機製品を構成する 11 種類以上の金属およびプラスチックの部品を、最速 2 秒台後半の時間間隔にて指定された姿勢に整列させられ、新規部品の追加に要するエンジニアリング時間は部品 1 つ当たり 0.5 日であることから、有用性がある。著者らの研究グループが対象としたような質量の部品を用いてバラ積み供給状態から組み上げる実用化されたセル生産ロボットシステムの成功事例は、世界的に類例がない。

本研究を含むロボット知能化技術研究開発プロジェクトの全体成果を統合した実証システムでは、部品点数 30 数点の電機製品を、人間の 2 倍程度の時間で組み立てることができる。また、そのシステム構築コストの削減に関して、教示予定作業時間 8 時間を 1 時間に短縮する効果を得ている。

現在、ロボット知能化技術研究開発プロジェクト成果は、徐々に製品化され市場に投入されている。例えば、システム構築ソフト、センサ類などを、用途ごとにパッケージ化した提案型製品 [125] の展開について前章で述べたとおりであり、その市場評価の伸長から近年の製造業の変貌に 대응するという当初の目的を達成しつつあると考える。



## 2. 展望

2010年代後半に入った現在，世界中で製造業を革新するための国策レベルのテコ入れが進められており，その一部は効果の兆候が見え始めている．民間における事業戦略では，ロボットの巨大な市場に対応するためのリソースシフトが散見され始めた．

技術面では，欧米でロボット用のソフトウェア開発が急進し，ベンチャーも起業しはじめている [126]．また，安全柵を取り払う研究に注目が集まり始めており，ロボットメーカー各社からコラボレーションロボット，略してコボット (Cobot: Collaborative robotの略) と呼ばれるジャンルの新製品発表が相次いでいる状況ではあるものの，真の課題の解決策としての効果を疑問視する声も根強い．実際，これがキーテクノロジーとして市場を席卷する決定打となるキラーアプリケーションはいまだ見られない．

これらに対し日本は，これまでのアドバンテージを活かして，開発した次世代ロボット技術の実用化を進めつつ，雇用と所得への寄与を前提とする革新的なロボット技術の研究開発の中長期的計画立案とその遂行に，ただちに着手すべき段階に入っていると考えられる．

時を同じくしてロボットユーザ側も新しいロボットシステムの登場を期待しはじめている．その動きの一端として，Amazon社が主催するストックヤードにおける商品積み替え模擬作業を競う Amazon Picking Challenge (<http://amazonpickingchallenge.org/>) と，Airbus社の主催する航空機製造ヤードにおける穴あけ工程を模擬した作業を競う Airbus Shopfloor Challenge (<http://www.airbusgroup.com/int/en/people-careers/Working-for-Airbus-Group/Airbus-Shopfloor-Challenge-2016.html>) の両コンテストが挙げられる．

産業用ロボットは，その誕生から30年来，設計概念が変化していない．具体的には，ロボット手首のフランジ面が，高速高精度に目標位置姿勢を再現する機能に特化して，機械系，電気系，情報系が最適設計されてきた．その意味において産業用ロボットは成熟し，同時にロボットを活用できる範囲が明確化し，それを拡大させながらの普及を果たした．

しかしながらロボットを活用できる範囲が，一定の範囲に収束しているのも事

実である。さらなる発展のためには、ロボットを利用する目的を今一度再考し、その目的を果たすためのロボットシステムの成り立ちを、材料と制御からゼロ・ベースで組み立て直す新たなイノベーションにチャレンジする必要があると考えている。製造業向けの産業用ロボットのみならず、サービスロボット、災害ロボットにも共通の課題が存在し、前述のイノベーションの成果は、機械システム全般の革新にも波及しうる。前述のコンテストの出題内容と上位入賞チームの解決策を注意深く観察するとイノベーションに至るヒントが隠されていると思えてならない。米国製造業革新施策、独 Industrie4.0 および中国製造 2025 の真の本丸も同じところを目指してくるに違いない。

一方、製造業が抱えている課題をより高い視点から俯瞰すると、日本を先頭にして世界中で進む人口動態変容への対応、世界のいたるところで発生する未曾有の大災害による生産拠点破壊とグローバルサプライチェーン寸断の迅速な再構築、および、IoT(Internet of Things) と FinTech(Financial technology) の深化の結果もたらされる景気と需給バランスの垂直変動への対処といった過酷な状況が遍在している。

これらに対峙するための最適解を走りながら考え続け、ロボット屋として答を出すことが期待されている。それは、原理原則に立ち返って問題を定式化し、その解法の本質をアルゴリズム化あるいはハードウェア設計し、ロボットで実装し、物理的な作用として発現させることにほかならない。これは永遠の課題である。その大前提は、基幹産業のひとつとして、雇用を生み、所得を個人に還元し、人生をその目的の達成と手段選択の自由を阻害する様々なリスクから切り離し、それらを維持可能にするという理念である。

## 参考文献

### 参考文献

- [1] Zupancic, J., Rudel, D., Cibej, I., Bajd, T., Kralj, A.: “Flexible Robotic Cell For Hybrid Circuits Assembly”, IECON '88, Industrial Electronics Society, Vol.4, pp.1022–1027, (1988)
- [2] Akio Noda: The Literacy-Free Manufacturing System, Proceedings of The 5th International Conference on Machine Automaton ICMA2004, pp.597–602, (2004)
- [3] Akio Noda, Tomoko Sendai: Intelligent Control Systems for Machine Work Taking The Case of Electric Discharge Machine, Proceedings of JAPAN / USA Symposium on Flexible Automation 1992, Vol.2, pp.1411–1415, The American Society of Mechanical Engineers, (1992)
- [4] 放電加工機ファジィ制御（三菱電機）, 日本知能情報ファジィ学会 ファジィ応用事例データベース, <http://www.j-soft.org/example-database/482/>
- [5] ファジィ制御 CNC 形彫り放電加工機（三菱電機）, 日本知能情報ファジィ学会 ファジィ応用事例データベース, <http://www.j-soft.org/example-database/767/>
- [6] 水川真・松家英雄・松元明弘・小山俊彦: 産業用ロボットにおけるネットワークインタフェースの標準化活動—ORiN: Open Robot Interface for the Network—, 日本ロボット学会誌, 18 卷 4 号, pp.468–471, (2000)

- [7] M Mizukawa, Toshihiko Koyama, Toshihiro Inukai, Akio Noda, Naoyoshi Kanamaru, Yasuhiko Noguchi, Nobuyuki Otera: Proposal of open-network-interface for industrial robots (ORiN) and its experimental evaluation, Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Vol.2, pp.689–694, IEEE, (2001)
- [8] M Mizukawa, H Matsuka, T Koyama, T Inukai, A Noda, H Tezuka, Y Noguchi, N Otera: ORiN: Open Robot interface for the Network, The Standard Network Interface for Industrial Robots and its Applications, Proc. of International Symposium on Robotics (ISR), paper ref. no.45, (2002)
- [9] 水川真: 産業用機器・ロボットのオープンネットワークインタフェース, 計測と制御, 42 巻 7 号, pp.566–571, (2003)
- [10] 水川真: 産業用機器・ロボット標準ネットワークインタフェース: ORiN, 日本ロボット学会誌, 21 巻 6 号, pp585–590, (2003)
- [11] 木村利明: 工場内ネットワークによる情報活用の方法, 精密工学会誌, 73 巻 8 号, pp.888–892, (2007)
- [12] <http://www.orin.jp/>
- [13] 加知光康, 吉本康浩, 北上真二, 小泉 寿男: 生産情報との連携による工場エネルギー管理システムとその実装評価, 計測自動制御学会論文集, Vol. 50 No. 4 p. 319–327, (2014)
- [14] 松岡眞, 清水武, 北島功朗, 渡部透: ロボットの活用によるセル生産システムの進化, 自動制御連合講演会講演論文集 49(0), pp.29-29, <http://ci.nii.ac.jp/naid/130004599979/>, (2006)
- [15] Akio Noda, Ken'ichi Tanaka, Haruhisa Okuda, Tatsuya Nagatani, Yasuo Kitaaki, Yukiyasu Domae, Hiroki Dobashi, Yasuyoshi Yokokohji, Kohei Kurono, Yukio Horiguchi, Hiroaki Nakanishi, Tetsuo Sawaragi, Keisuke Iso-mura, Hideo Utsuno, Hiroshi Matsuhisa, Shin'ichi Kazama, Takehisa Kohda:

Intelligent Robot Technologies for Cell Production System, Proceedings of 2010 International Symposium on Flexible Automation, The Institute of Systems, Control and Information Engineers & The American Society of Mechanical Engineers, JPS-2562, (2010)

- [16] ロボナブル：年末企画 分野別に振り返るロボット業界 2011,PART4 国際ロボット展, 2011.12.28, <http://www.robonable.jp/special/2011/12/part4.html>, (2011)
- [17] 小菅一弘: ロボティクスと実ロボット開発, 第12回計測自動制御学会システムインテグレーション部門講演会予稿, 1E2-3, pp.320-321, (2011)
- [18] <http://www.whitehouse.gov/blog/2011/06/24/developing-next-generation-robots>
- [19] 智能制造科技发展“十二五”专项规划, 中华人民共和国科学技术部, <http://www.most.gov.cn/tztg/201204/W020120424327129213807.pdf>, (2012)
- [20] 澤田朋子：ドイツ Industrie 4.0 と中国製造 2025 (2/2), 内閣府・総合科学技術・イノベーション会議・戦略協議会・システム基盤技術検討会 (第2回) 配布資料, <http://www8.cao.go.jp/cstp/tyousakai/juyoukadai/wg.html>, <http://www8.cao.go.jp/cstp/tyousakai/juyoukadai/system/2kai/shiryo3-6-2.pdf>
- [21] 邵永裕, 中国国務指南「中国製造 2025」の戦略高層と将来展望, みずほグローバルニュース, 2016 MAY&JUN vol.85, pp.16-17, (2016)
- [22] 呉明憲：「中国製造 2025」の目玉, 週刊ダイヤモンド, 2016.07.02 号, pp.21, (2016)
- [23] 澤田朋子：ドイツ Industrie 4.0 と中国製造 2025 (1/2), 内閣府・総合科学技術・イノベーション会議・戦略協議会・システム基盤技術検討会 (第2回) 配布資料, <http://www8.cao.go.jp/cstp/tyousakai/juyoukadai/wg.html>,

<http://www8.cao.go.jp/cstp/tyousakai/juyoukadai/system/2kai/shiryo3-6-1.pdf>

- [24] <http://www.kuka-robotics.com/en/products/addons/lwr/>, <http://www.kuka-lbr-iiwa.com/>
- [25] <http://www.abb.com/cawp/abbzh254/8657f5e05ede6ac5c1257861002c8ed2.aspx>, <http://new.abb.com/products/robotics/yumi>
- [26] <http://www.universalrobotics.com/neocortex>
- [27] <http://rosindustrial.org/>
- [28] <http://www.microsoft.com/robotics/>
- [29] 経済産業省: ロボット技術導入事例調査 ロボット技術導入事例集, [http://www.meti.go.jp/meti\\_lib/report/2011fy/E001537.pdf](http://www.meti.go.jp/meti_lib/report/2011fy/E001537.pdf), (2011)
- [30] 日本ロボット工業会: ロボットエンジニアリング業界活性化検討会議, [http://www.jara.jp/x8\\_engineering/index.html](http://www.jara.jp/x8_engineering/index.html), (2011)
- [31] 「ロボット新戦略」, <http://www.meti.go.jp/press/2014/01/20150123004/20150123004.html>, (2014)
- [32] 野田哲男, 奥田晴久, 田中健一, 永谷達也, 北明靖雄, 堂前幸康, 榎木哲夫, 横小路泰義, 堀口由貴男, 幸田武久, 宇津野秀夫, 松久寛, 水山元, 小森雅晴, 泉井一浩, 西脇眞二: 次世代セル生産を実現するロボット知能化技術, 第10回計測自動制御学会システムインテグレーション部門講演会予稿集, 1L3-4, pp.766-769, (2009)
- [33] 田中健一, 榎木哲夫: セル生産を実現するロボット知能化技術開発の展望, 第10回計測自動制御学会システムインテグレーション部門講演会予稿集, 1J2-2, pp.657-659, (2009)

- [34] 具□謨 (□=王へんに景, Kyong-mo Koo), 近野敦, 内山勝: セル生産用複数ロボットモーションプランナーの開発, 第11回計測自動制御学会システムインテグレーション部門講演会, 1J3-1, (2010)
- [35] 具□謨 (□=王へんに景, Kyong-mo Koo): Motion Planning for Flexible Object Handling by Multiple Redundant Manipulators (複数の冗長マニピュレータによる柔軟物ハンドリング動作計画), 東北大学博士論文, <http://hdl.handle.net/10097/61480>, <http://ci.nii.ac.jp/naid/500000563278>, (2011)
- [36] Rintaro Haraguchi, Yukiyasu Domae, Koji Shiratsuchi, Yasuo Kitaaki, Haruhisa Okuda, Akio Noda, Kazuhiko Sumi, Takayuki Matsuno, Shun'ichi Kaneko, and Toshio Fukuda, Development of Production Robot System that can Assemble Products with Cable and Connector, *Journal of Robotics and Mechatronics*, Fuji Tech. Press, Vol.23 No.6, pp.939–950, (2011)
- [37] 仲井勤, 野田哲男: 依存関係抽出に基づくシーケンス制御プログラム解析, 計測自動制御学会論文集, Vol.50, No.1, pp.24–29, (2014)
- [38] 泉井一浩, 村雲泰, 末光一成, 西脇眞二, 野田哲男, 永谷達也: シーケンスペア表現を用いたロボットセル生産システムのレイアウト最適設計法, *精密工学会誌*, Vol.77, No.8, pp.781–787, (2011)
- [39] Kazuhiro Izui, Yutaka Murakumo, Issei Suemitsu, Shinji Nishiwaki, Akio Noda, Tatsuya Nagatani: Multiobjective Layout Optimization of Robotic Cellular Manufacturing Systems, *Computers and Industrial Engineering*, 64(2913), pp.537–544, (2013)
- [40] Suemitsu, I., Izui, K., Yamada, T., Nishiwaki, S., Noda, A., and Nagatani, T.: A Layout Optimization Method for Robotic Cellular Manufacturing Systems Considering Task Scheduling, *CIE44 & IMSS'14 Proceedings*, pp.533–546, (2014)

- [41] D. Ishizuka, K. Izui, X. Bai, I. Suemitsu, T. Yamada, S. Nishiwaki, A. Noda, T. Nagatani, "A bi-level multiobjective optimization approach for layout design of robotic cellular manufacturing systems" *Computers & Industrial Engineering* 誌 (Special Issue on New Trends in Computers and Industrial Engineering and their Applications) , (2016)
- [42] Izui, K., Bai, X., Yamada, T., Nishiwaki, S., Noda, A., Nagatani, T., "A bi-level multiobjective optimization approach for layout design of robotic cellular manufacturing systems", *Proceedings of the 45th International Conference on Computers & Industrial Engineering*, Paper, No. 175. October 28-30, 2015, Metz France. <http://cie45.event.univ-lorraine.fr/> , (2015)
- [43] Issei SUEMITSU, Kazuhiro IZUI , Takayuki YAMADA, Shinji NISHIWAKI, Akio NODA, and Tatsuya NAGATANI: "Simultaneous optimization of layout and task schedule for robotic cellular manufacturing systems", *Computers & Industrial Engineering*, Special Issue on Contributions to Society of Information, Manufacturing and Service Systems Developments, doi:10.1016/j.cie.2016.05.036., Elsevier B.V., 2016 年発刊予定
- [44] 阪田隆司, 幸田武久, 野田哲男, 長野陽, 永谷達也: エラー解析に対するダイナミックベイジアンネットワークの応用, 第12回計測自動制御学会システムインテグレーション部門講演会予稿集, 1E1-3, pp.323-325, (2011)
- [45] Sebastian Thrun, Wolfram Burgard, Dieter Fox, 上田隆一 (訳): 「確率ロボティクス」 (*Probabilistic Robotics* の翻訳書), 毎日コミュニケーションズ (2007, 訳 2015)
- [46] 土橋宏規, 野田哲男, 横小路泰義, 長野陽, 永谷達也, 奥田晴久, 田中健一: 能動探索アルゴリズムによる組立作業用汎用ハンドのロバスト把持戦略の最適化, 第27回日本ロボット学会学術講演会予稿集 DVD-ROM, AC1D2-03, (2009)



- [47] Hiroki Dobashi, Akio Noda, Yasuyoshi Yokokohji, Hikaru Nagano, Tatsuya Nagatani, Haruhisa Okuda: Derivation of optimal robust grasping strategy under initial object pose errors. IROS 2010: pp.2096–2102, (2010)
- [48] 土橋宏規, 横小路泰義, 野田哲男, 奥田晴久: 準静的押し操作解析にもとづく把持シミュレーションと対象物体の許容初期誤差範囲の導出, 日本ロボット学会誌 vol.28, no.10, pp.1201–1212, (2010)
- [49] Hiroki Dobashi, Junichi Hiraoka, Takanori Fukao, Yasuyoshi Yokokohji, Akio Noda, Hikaru Nagano, Tatsuya Nagatani, Haruhisa Okuda, Ken-ichi Tanaka: Robust grasping strategy for assembling parts in various shapes. *Advanced Robotics* 28(15): pp.1005–1019 , (2014)
- [50] 土橋宏規, 神岡渉, 深尾隆則, 横小路泰義, 野田哲男, 長野陽, 永谷達也: 組立作業のための平行スティック指汎用ハンドによる三次元形状部品の把持の最適化, システム制御情報学会論文誌, 28/ 6, pp.258–266, (2015)
- [51] Yusuke Bota, Hajime Mizuyama, Akio Noda, Tatsuya Nagatani, Ken-ichi Tanaka: A tree-shaped motion strategy for robustly executing robotic assembly tasks, *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, ISSN 1673-565X(Print), 1862-1775(Online), Vol.11 No.12, pp.986–991, (2010)
- [52] Lei Wang, Yajie Tian and Tetsuo Sawaragi: Explanation-Based Manipulator Learning: Acquisition of Assembling Technique through Observation, *Preprints of the 17th IFAC World Congress*, CD-ROM, pp.2412–2417, (2008)
- [53] Lei Wang, Tetsuo Sawaragi, Yajie Tian: A Hierarchical Knowledge Based System for Assembly Tasks in Human-robot Cell Manufacturing, *Proc. of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, pp.127–132, (2009)
- [54] Lei Wang, Tetsuo Sawaragi, Yajie Tian and Yukio Horiguchi: Integrating CBR and EBL in an apprentice agent, *Proc. of the 2nd International Con-*

- ference on Agents and Artificial Intelligence, Vol.1 - Artificial Intelligence, pp.667–670, (2010)
- [55] Lei Wang, Tetsuo Sawaragi, Yajie Tian and Yukio Horiguchi: Acquisition of Human Expertise in Robotic Assembly, *SICE Journal of Control, Measurement, and System Integration*, 3(4), pp.299–308, (2010)
- [56] 王磊 : Studies on Interactive Knowledge Acquisition and Reuse for Teaching Industrial Robots (産業用ロボットの教示知識の対話型獲得と再利用に関する研究), 京都大学博士論文, (2010)
- [57] Lei Wang, Yajie Tian, and Tetsuo Sawaragi: Case-based automatic programming in robotic assembly production, *Industrial Robot: An International Journal*, Vol. 38, No. 1, pp.86–96, (2011)
- [58] Yukio Horiguchi, Kohei Kurono, Hiroaki Nakanishi, Tetsuo Sawaragi, Tatsuya Nagatani, Akio Noda, Ken-ichi Tanaka: Ecological interface design for teaching assembly operations to industrial robot. *IFAC HMS 2010*: pp.442–447, (2010)
- [59] 堀口由貴男, 黒野晃平, 中西弘明, 榎木哲夫, 永谷達也, 野田哲男, 田中健一: 産業用ロボット教示作業支援のための複合情報 GUI, *計測自動制御学会論文集*, Vol.47(2), pp.656–665, (2011)
- [60] 堀口由貴男 : 自動化工程の運用管理に求められる認知的作業の分析技術, *経営システム*, Vol.22, No.2, pp.72–77, (2012)
- [61] 永谷達也, 野田哲男, 田中健一: 座標系校正による産業用ロボットの位置復旧支援技術, *計測自動制御学会論文集*, Vol.50, No.1, pp.30–36, (2014)
- [62] Van Anh Ho, Tatsuya Nagatani, Akio Noda, Shinichi Hirai: What can be inferred from a tactile arrayed sensor in autonomous in-hand manipulation? *CASE 2012*: pp.461–468, (2012)

- [63] <https://ja.wikipedia.org/wiki/RT> ミドルウェア
- [64] <http://www.ros.org/>
- [65] 野田哲男, 長野陽, 堂前幸康, 永谷達也, 田中健一: 制御パラメータの調整作業を自動化するための能動型探索アルゴリズムと, その適用事例について, システム制御情報学会論文誌, 第 29 巻第 2 号, pp.65–75, (2016)
- [66] R.A.Fisher: The Design of Experiments; 9th ed. from Macmillan, ISBN 0-02-844690-9, (1935, 1971)
- [67] 山田秀: 実験計画法 方法編—基盤的方法から応答曲面法、タグチメソッド、最適計画まで, 日科技連出版社, ISBN-13: 978-4817103895 (2004)
- [68] George EP Box, K. B. Wilson: On the experimental attainment of optimum conditions, Journal of the Royal Statistical Society. Series B (Methodological), The Royal Statistical Society, 13.1, pp.1–45, (1951)
- [69] Paul J. Werbos: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, PhD thesis, Harvard University, (1974)
- [70] Ethem Alpaydin: Introduction to machine learning 2nd ed, MIT Press, (2009)
- [71] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams: Learning representations by back-propagating errors, Nature 323 (6088), pp.533-536, (1986)
- [72] Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, Andrew Ng: Building high-level features using large scale unsupervised learning, Proc. of 29th International Conference in Machine Learning, (2012)
- [73] Richard S. Sutton, Andrew G. Barto: Reinforcement Learning: An Introduction, MIT Press, (1998)

- [74] 渡辺澄夫, 萩原克幸, 赤穂昭太郎, 本村陽一, 福水健次, 岡田真人, 青柳美輝: 学習システムの理論と実現, 森北出版, pp.98–131, (2005)
- [75] B. Settles: Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, (2009)
- [76] David A. Cohn, Zoubin Ghahramani, Michael I. Jordan: Active Learning with Statistical Models, Journal of Artificial Intelligence Research 4, pp.129–145, (1996)
- [77] Dana Angluin: Queries and Concept Learning, Machine Learning 2, Kluwer Academic Publishers, pp.319–342, (1988)
- [78] 中山弘隆, 荒川雅生, 佐々木理恵: RBF ネットワークと遺伝アルゴリズムを用いた近似最適化, 計算機工学講演会論文集, 6(2), pp.681–684, 2001-05-30, (2001)
- [79] 鷺野宏治, 中山弘隆: サポートベクターマシーンによる未知目的関数最適化, 第46回システム制御情報学会研究発表講演会講演論文集, pp.134, (2002)
- [80] C. Cortes, V. Vapnik: “Support-Vector Networks,” Machine Learning, Vol. 20, pp.273–297, (1995)
- [81] Greg Schohn, David Cohn: “Less is more: Active Learning with Support Vector Machines”, ICML '00 Proceedings of the 17th International Conference on Machine Learning, pp.839–846, (2000)
- [82] Simon Tong, Daphne Koller: Support Vector Machine Active Learning with Applications to Text Classification, Journal of Machine Learning Research, pp.45–66, (2001)
- [83] Gert Cauwenberghs, Tomaso Poggio: Incremental and Decremental Support Vector Machine Learning, Advances in Neural Information Processing Systems, volume 13, (2001)

- [84] 大津裕美, 出口光一郎: 能動的物体認識のための最適カメラパラメータ選択方法, 第5回計測自動制御学会制御部門大会, pp.851–854, (2005)
- [85] S. Kullback, R.A.Leibler: On information and sufficiency, Ann. Math. Statist. 22, No1. pp.79–86, (1951)
- [86] Takamitsu Matsubara, Kotaro Shibata, Kenji Sugimoto: Active Touch Point Selection with Travel Cost in Tactile Exploration for Fast Shape Estimation of Unknown Objects, The 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM 2016) Paper ThCT6.3, (2016)
- [87] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, Vol.2, Issue 3, pp.1–27, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, (2011)
- [88] 野田哲男, 永谷達也: 能動探索アルゴリズムによる産業用ロボットの動作習熟, 第26回日本ロボット学会学術講演会, RSJ2008AC1F2-02, (2008)
- [89] 野田哲男, 永谷達也: 産業用ロボットの動作習熟における能動型探索アルゴリズム, SICE 第9回SI部門講演会, 1E3-6, pp.211–212, (2008)
- [90] 野田哲男, 岩本貴司, 佐々木一真, 玉置久: 能動型探索アルゴリズムとその放電加工条件開発への適用, 第51回システム制御情報学会研究発表講演会, 5T4-5, pp.179–180, (2007)
- [91] 野田哲男, 岩本貴司, 佐々木一真, 玉置久: 能動型探索アルゴリズムによる加工制御パラメータ調整の自動化, 電気加工学会全国大会 (2007) 講演論文集, pp.281–282, (2007)
- [92] 野田哲男, 塩澤貴弘, 長野陽: 能動型探索アルゴリズムによる加工制御パラメータ調整の自動化 第2報, 電気加工学会全国大会 (2008) 講演論文集, pp.47–48, (2008)

- [93] 時任宏, 野田哲男, 中村直幸, 瀬口正記: ある工業プロセスにおける能動型探索手法を用いた制御パラメータの最適化, 第55回システム制御情報学会研究発表講演会, ROMBUNNO.T35-3, (2011)
- [94] 森, 重政: PID オートチューニングコントローラの動向, 計測と制御, 29-8, pp.723-728, (1990)
- [95] 重政隆: セルフチューニング PID 制御の動向, システム制御情報学会誌, Vol.42, No.1, pp.35-40, (1998)
- [96] 小山正人, 長野鉄明: 規範モデルを用いた電動機の2自由度位置制御, 電気学会論文誌D, Vol.114 No.2, pp.137-143, (1994)
- [97] 松原崇充: 確率最適制御の最近の動向, システム制御情報学会誌, Vol.59, No.10, pp.369-374, (2015)
- [98] 野田哲男, 堂前幸康, 永谷達也, 奥田晴久, 田中健一: 一般形状部品の多品種供給の自動化, 日本ロボット学会論文誌, Vol.33 No.5, pp.387-394, (2015)
- [99] 日本部品供給装置工業会: 部品供給装置の導入史, [http://www.jpfn.ne.jp/index.html?http://www.jpfn.ne.jp/html/01/01\\_02.html](http://www.jpfn.ne.jp/index.html?http://www.jpfn.ne.jp/html/01/01_02.html)
- [100] Wes Iversen: Vision-guided Robotics: In Search of the Holy Grail, Automation World, February 1, 2006, <http://www.automationworld.com/networking-amp-connectivity/vision-guided-robotics-search-holy-grail>, (2006)
- [101] Kenneth Goldberg, Matthew Mason, Michael Erdmann: “Generating Stochastic Plans for a Programmable Parts Feeder”, Proceedings of the 1991 IEEE International Conference on Robotics and Automation, pp.352-359, (1991)
- [102] Brian Mirtich, Yan Zhuang, Ken Goldberg, John Craig, Rob Zanutta, Brian Carlisle, John Canny: “Estimating Pose Statistics for Robotic Part Feeders”,

- Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Vol. 2, pp.1140–1146, (1996)
- [103] Dadi Gudmundsson, Ken Goldberg: “Tuning robotic part feeder parameters to maximize throughput”, Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp.2440–2445, (1997)
- [104] Dadi Gudmundsson, Ken Goldberg: “Estimating and optimizing throughput of a robotic part feeder using queueing theory”, Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2, pp.818–824, (1998)
- [105] Robert-Paul Berretty , Ken Goldberg, Lawrence Cheung , Mark H. Overmars, Gordon Smith, A.F. van der Stappen: “Trap Design for Vibratory Bowl Feeders”, Proceedings of IEEE International Conference on Robotics and Automation, pp.2558–2563, (1999)
- [106] Bertram Drost, Markus Ulrich, Nassir Navab, Slobodan Ilic: “Model Globally, Match Locally: Efficient and Robust 3D Object Recognition ”, in Proc. of CVPR, pp.998–1005, (2010)
- [107] C. Choi, Y. Taguchi, O. Tuzel, M. Y. Liu, S. Ramalingam: “Voting-Based Pose Estimation for Robotic Assembly Using a 3D Sensor ”, Proceedings of IEEE International Conference on Robotics and Automation, pp.1724–1731, (2012)
- [108] D. Buchholz, M. Futterlieb, S. Winkelbach, F. M. Wahl, “Efficient bin-picking and grasp planning based on depth data ”, Proceedings of IEEE International Conference on Robotics and Automation, pp.3245–3250, (2013)
- [109] Jeremy Marvel, Roger Eastman, Geraldine Cheok, Kamel Saidi, Tsai Hong, Elena Messina : “Technology Readiness Levels for Randomized Bin Picking”, Performance Metrics for Intelligent Systems (PerMIS) Workshop, Special Session, Malyland, USA, Sep. 4, (2012)

- [110] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y. Kawai: “Probabilistic approach for object bin picking approximated by cylinders”, Proceedings of IEEE International Conference on Robotics and Automation, pp.3742–3747, (2013)
- [111] M. Nieuwenhuisen, D. Droschel, D. Holz, J. Stuckler, A. Berner, J. Li, R. Klein, and S. Behnke: “Mobile bin picking with an anthropomorphic service robot”, Proceedings of IEEE International Conference on Robotics and Automation, pp.2327–2334, (2013)
- [112] M. Berger, G. Bachler, S. Scherer: “Vision guided bin picking and mounting in a flexible assembly cell”, in Proc. of International Conference on Industrial Engineering Applications of Artificial Intelligence and Expert systems (IEA/AIE), vol.1821, pp.109–117, (2000)
- [113] Haruhisa Okuda, Yuichi Taguchi, Kazuhiko Sumi, Takashi Hirai: “Fast graspability evaluation on single depth maps for bin picking with general grippers”, Proceedings of IEEE International Conference on Robotics and Automation, pp.1997–2004, (2014)
- [114] K.Goldberg, B.Mirtich, Y.Zhuang, J.Craig, B.Carlisle, J.Canny: “Part Pose Statistics: Estimators and Experiments,” IEEE Transactions on Robotics and Automation, vol.15, no.5, pp.849–857, (1999)
- [115] 神岡渉, 横小路泰義, 土橋宏規, 野田哲男, 永谷達也, 長野陽: 平面上にある三次元形状部品の安定性解析, 日本機械学会ロボメカ部門講演会, 1P1-H05, (2012)
- [116] 神岡渉, 横小路泰義, 土橋宏規, 野田哲男, 永谷達也, 長野陽: 平面上にある三次元形状部品に対する安定性解析の実験的検証と把持指との接触点位置を考慮したぐらつきにくさの導出, 日本ロボット学会学術講演会, RSJ2012ACJ1-3, (2012)



- [117] 神岡渉, 横小路泰義, 土橋宏規, 野田哲男, 永谷達也, 長野陽: 平面上にある三次元形状部品と把持指との接触点位置を考慮したぐらつきにくさの解析と実験的検証, 計測自動制御学会 SI 部門講演会 SI2012, pp.1560–1563, (2012)
- [118] 神岡渉, 横小路泰義, 土橋宏規, 野田哲男, 長野陽, 永谷達也: N 本平行円柱スティック型ハンドを用いた三次元形状部品に対する安定把持状態の導出, 日本ロボット学会学術講演会, RSJ2013ACA1-08, (2012)
- [119] 神岡渉, 横小路泰義, 土橋宏規, 野田哲男, 長野陽, 永谷達也: N 本平行円柱スティック型ハンドによる三次元形状部品に対する押し操作のロバスト性評価, 計測自動制御学会 SI 部門講演会 SI2013, pp.0553–0557, (2013)
- [120] 原田研介, マニピュレーション研究把持計画を中心とした研究動向, 日本ロボット学会誌, Vol.31 No.4, pp.320–325, (2013)
- [121] 永谷達也, 野田哲男, 長野陽: ハンドとワーク間の各面接触空間を利用した把持計画の検討, 計測自動制御学会 SI 部門講演会 SI2013, pp.0545–0546, (2013)
- [122] 羽根幹人, 神岡渉, 横小路泰義, 土橋宏規, 野田哲男, 長野陽, 永谷達也: 支持面上におけるロボットハンドによる安定した把持操作の実現に向けた対象物体の安定姿勢への遷移計画, 計測自動制御学会 SI 部門講演会 SI2013, pp.0547–0552, (2013)
- [123] 羽根幹人, 土橋宏規, 横小路泰義, 藤本堅太, 永谷達也, 長野陽, 野田哲男: 平行スティック型汎用ハンドによる転がし操作を利用した対象部品の安定姿勢への遷移計画, 第 15 回計測自動制御学会システムインテグレーション部門講演会 SI2014 予稿集, pp.1675–1679, (2014)
- [124] Suemitsu, I., Izui, K., Yamada, T., Nishiwaki, S., Noda, A., Nagatani, T.: “A Layout Optimization Method for Robotic Cellular Manufacturing Systems Considering Task Scheduling”, CIE44 & IMSS’14 Proceedings, 14-16, October 2014, Istanbul, Turkey, pp.533–546, (2014)

- [125] 三菱電機 Factory Automation サイト 産業用ロボット アプリケーションパッケージ <http://www.mitsubishielectric.co.jp/fa/products/rbt/robot/pmerit/option/app/index.html>
- [126] そろそろ日本の産ロボらしい付加価値の追求と議論を! -上海国際ロボ展から現地メーカーの動向を探る, 2012.07.24, <http://www.robonable.jp/trend/2012/07/ir-0724.html>, (2012)

# 謝辞

本研究を論文にまとめるにあたり、非常に多くの方々のご指導・ご協力を賜りましたことに心より感謝申し上げます。

主指導教員 奈良先端科学技術大学院大学 小笠原司教授，副指導教員 同 杉本謙二教授，高松淳准教授，丁明助教から，数多くの丁寧かつ適切なお指導をいただきました。

同学ロボティクス研究室のスタッフ・OB・学生の皆様，同学の皆様には研究会・授業・公聴会ほか，様々な議論の機会を通じて重要な示唆をいただき，また学生生活に関するお力添えをいただきました。

本論文の基礎となる研究開発は，参考文献リストに列挙したとおり，数多くの共同研究者とのオープンイノベーションによるものです。共同研究先の各組織の教員・研究室スタッフ・学生・研究員・管理部門の皆様，国家プロジェクト各リーダー・参画団体の皆様にお礼申し上げます。

研究開発のマクロな方向性について官公庁・学協会・報道機関・顧客・関連企業の皆様と常日頃の議論を通じてご指導いただきました。

本研究の一部は“METI・NEDO 次世代ロボット知能化技術開発プロジェクト”，“NEDO 戦略的先端ロボット要素技術開発プロジェクト”の委託を受けて実施しました。

三菱電機株式会社の研究所・本社・各支社・各製作所および工場の直接部門・管理部門各位のご高配・ご尽力・ご指導のもと共同研究開発を実施しました。

妻と子供の理解と協力に感謝します。