

博士論文

数値解析を活用した  
実機械の制御パラメータ最適化における  
人と計算機の相補的關係の設計

唐澤 宏之

# 目次

目次	1
図目次	6
表目次	10
1 緒論	11
1.1 研究背景	11
1.1.1 機械とその設計手法	11
1.1.2 情報技術の進展と設計手法の変化	17
1.1.3 設計における知的作業の代替	27
1.2 研究目的	31
1.2.1 研究のターゲット	31
1.2.2 研究目的	31
1.3 本論文の構成	34
2 数値解析を活用した実機械システムの最適化	35
2.1 現象のモデル化形態の比較	35
2.1.1 モデル化の3形態の定義	35
2.1.2 3形態のモデル化の特性	37
2.1.3 3形態のモデルの関係性	41
2.2 複雑な目的関数のための最適化手法	42
2.2.1 最適化の定義	42
2.2.2 最適化手法の分類	43
2.2.3 目的関数とモデルとの違い	46
2.3 実機械システムの最適化の関連研究	46
2.4 数値解析モデルを活用した実機械システムの最適化	48
2.4.1 数値解析モデルを活用した最適化システムの利点	49

2.4.2	数値解析モデルを活用した最適化システムの課題 . . . . .	51
3	最適化システムの最適化のためのメタな最適化システムの提案 . . . . .	54
3.1	数値解析モデルを活用した最適化システムの問題 . . . . .	54
3.2	メタな最適化システムの具体例と位置づけ . . . . .	56
3.3	メタな最適化システムの定義 . . . . .	59
3.3.1	一段階メタなシステムの基本構成 . . . . .	59
3.3.2	メタな最適化システムの内部動作 . . . . .	62
3.3.3	非メタな最適化システムとメタな最適化システムとの視点の比較 . . . . .	64
3.4	メタな最適化システムの安定性や性能に関する考察 . . . . .	66
3.5	メタな最適化システムを構成する機構の提案 . . . . .	68
3.5.1	機構 1. 最適化結果の重要度に基づいた要素修正 . . . . .	69
3.5.2	機構 2. 数値解析結果の再利用 . . . . .	70
3.5.3	機構 3. 補正係数導入によるモデル修正の自動化 . . . . .	72
3.6	本章における提案の検証方法 . . . . .	73
3.6.1	具体的な実機械システム . . . . .	73
3.6.2	TM 制御パラメータ最適化システムのメタ最適化システム . . . . .	75
3.6.3	円筒研削加工の加工条件最適化システムのメタ最適化システム . . . . .	76
3.6.4	2 つの最適化問題における課題が複合する場合 . . . . .	77
4	電子制御トランスミッション制御パラメータの最適化 . . . . .	79
4.1	研究課題 . . . . .	79
4.1.1	電子制御トランスミッション . . . . .	79
4.1.2	関連研究 . . . . .	82
4.1.3	挙動模擬のための数値解析モデル . . . . .	83
4.1.4	数値解析モデルを活用した電子制御トランスミッション制御パラメータの最適化 . . . . .	83
4.1.5	本章における研究の副目的 . . . . .	86
4.2	課題達成のためのアプローチ . . . . .	86
4.2.1	扱う最適化問題の特性 . . . . .	87
4.2.2	メタな最適化システムの動作 . . . . .	87
4.3	数値解析モデル及び目的関数の構成 . . . . .	92
4.3.1	現象模擬のための数値解析モデル . . . . .	92
4.3.2	性能評価のための目的関数 . . . . .	93
4.3.3	ソフトウェア上の実装方法 . . . . .	97

4.4	計算機上の最適化実験 . . . . .	98
4.4.1	最適化手法 . . . . .	99
4.4.2	実験設定 . . . . .	101
4.4.3	単一運転条件での最適化実験 . . . . .	102
4.4.4	複数運転条件での最適化実験 . . . . .	106
4.4.5	考察 . . . . .	107
4.5	実機による評価実験 . . . . .	108
4.5.1	実験設定 . . . . .	108
4.5.2	実験結果 . . . . .	110
4.6	メタな最適化システムにおける人間の介入の様子 . . . . .	112
4.6.1	人間の介入が必要となった事例 . . . . .	112
4.6.2	メタな最適化の動作結果に関する考察 . . . . .	114
4.7	本章の結言 . . . . .	114
5	円筒研削加工条件の最適化 . . . . .	118
5.1	研究課題 . . . . .	118
5.1.1	研削加工 . . . . .	118
5.1.2	関連研究 . . . . .	122
5.1.3	数値解析モデルを活用した円筒研削加工条件最適化 . . . . .	124
5.1.4	本章における研究の副目的 . . . . .	126
5.2	課題達成のためのアプローチ . . . . .	126
5.2.1	扱う最適化問題の特性 . . . . .	127
5.2.2	メタな最適化システムの動作 . . . . .	127
5.3	高速動作する数値解析モデルの構成 . . . . .	131
5.3.1	現象模擬のための数値解析モデル (数値シミュレータ) . . . . .	131
5.3.2	ソフトウェア上での実装 . . . . .	138
5.4	計算機上の最適化実験 . . . . .	139
5.4.1	性能評価のための目的関数 . . . . .	139
5.4.2	最適化手法 . . . . .	143
5.4.3	目的関数の最適化実験 . . . . .	143
5.4.4	結果 . . . . .	144
5.4.5	考察 . . . . .	146
5.5	実機を含み自律的なモデル修正を行う最適化アルゴリズムの提案と評価 . . . . .	147
5.5.1	自律的なモデル修正を行う最適化アルゴリズムの実装 . . . . .	147
5.5.2	実機実験の設定 . . . . .	149



5.5.3	実機実験の結果 . . . . .	149
5.5.4	実機実験の考察 . . . . .	151
5.6	本章の結言 . . . . .	152
6	結言 . . . . .	155
6.1	実施内容まとめ . . . . .	155
6.2	実機械システムの最適化における成果と知見まとめ . . . . .	157
6.2.1	4 章 電子制御トランスミッションの制御パラメータ最適化 . . . . .	157
6.2.2	5 章 円筒研削加工の加工条件最適化 . . . . .	158
6.3	将来課題 . . . . .	159
A	階層型確率的最適化の提案 . . . . .	160
A.1	問題の定式化 . . . . .	160
A.2	提案手法 . . . . .	161
A.2.1	密度推定のための重要度サンプリング . . . . .	161
A.2.2	重要度に基づく重みづけを利用した VBEM 法 . . . . .	162
A.2.3	有効な下位方策選択のための上位方策 . . . . .	163
A.3	動作検証 . . . . .	164
B	探索手順転移手法 . . . . .	168
B.1	関連研究 . . . . .	168
B.1.1	人間による探索課題へのアプローチ . . . . .	168
B.1.2	機械による探索課題へのアプローチ . . . . .	168
B.1.3	知識の転移手法 . . . . .	170
B.2	Search Procedure Transfer Algorithm の提案 . . . . .	170
B.2.1	問題設定 . . . . .	170
B.2.2	タスクの定義 . . . . .	171
B.2.3	探索知識の転移手法 . . . . .	172
B.2.4	ソースタスクの設定 . . . . .	173
B.3	実装 . . . . .	174
B.3.1	タスクの生成方法 . . . . .	175
B.3.2	アルゴリズム実装 . . . . .	175
B.3.3	性能評価方法 . . . . .	176
B.4	実験 . . . . .	176
B.4.1	1 変数関数を用いた実験 . . . . .	177
B.4.2	多変数関数を用いた実験 . . . . .	178

B.4.3 複数のソースタスクを利用する実験 . . . . .	180
参考文献	182
謝辞	190
研究業績リスト	193

# 目次

1.1	設計に必要な 3 つの知識. . . . .	13
1.2	ムーアの法則と計算機の進化. (文献 [14] を元に作成) . . . . .	18
1.3	行列積計算アルゴリズムの性能向上 [16]. . . . .	18
1.4	人工知能アルゴリズムの動作に必要な計算性能 [18]. . . . .	20
1.5	初期の電子計算機 ENIAC [22]. . . . .	21
1.6	HondaJet のエンジン翼上配置 [23]. . . . .	21
1.7	バネのモデル化の 2 手法. . . . .	22
1.8	1D-CAE のコンセプト [25]. . . . .	22
1.9	AlphaGoZero の性能向上過程 [34]. . . . .	25
1.10	文献 [37] におけるロボットアームによる大規模データ収集. . . . .	26
1.11	Domain randomization [27] のコンセプト. . . . .	26
1.12	オンライン部品製造プラットフォーム meviy [40]. . . . .	27
1.13	過去の計算結果に基づいた粒子法数値シミュレーションの高速化 [46]. . . . .	30
1.14	本論文の構成. . . . .	34
2.1	モデルの定義とモデルの 3 形態. . . . .	36
2.2	3 つのモデル化形態とそれぞれの具体例. . . . .	37
2.3	ロジスティック写像における初期値 $a$ の変化による挙動の変化. . . . .	39
2.4	関数形状の例と関連する用語. . . . .	43
2.5	Skip-connection の持つ誤差関数の多峰性の低減効果 [54]. . . . .	44
2.6	クロスエントロピー法による最適化. . . . .	46
2.7	数値解析モデルを活用した最適化システム. . . . .	49
2.8	数値解析モデルと目的関数に関わるデータ. . . . .	51
3.1	球入れ対戦競技の例. . . . .	56
3.2	非メタな最適化システムとメタな最適化システムの比較. . . . .	59
3.3	最適化システム構築のための 3 行為. . . . .	60

3.4	メタ最適化システムの動作 (目的関数が固定されていない場合).	62
3.5	メタ最適化システムの動作 (目的関数が固定されている場合).	62
3.6	最適化システムとその要素を開発するとき視点.	64
3.7	メタ最適化システムを開発するときの視点.	65
3.8	最適化結果の重要度に基づいた要素修正.	69
3.9	要素が想定外の挙動をした際の要素修正.	70
3.10	計算の類似性に注目した状態数の削減イメージ.	71
3.11	数値解析結果の再利用による計算量低減.	71
3.12	関数形状を保持したモデル修正.	72
3.13	補正係数によるモデル修正.	73
3.14	数値解析モデルを活用した最適化が直面する課題.	74
3.15	TM 制御のメタ最適化システム.	75
3.16	円筒研削加工のメタ最適化システム.	76
4.1	トランスミッションの性能を示す車速駆動トルク曲線.	80
4.2	変速ショックの実際と理想.	81
4.3	制御方策とロボットの寸法の同時最適化 [87].	84
4.4	目的関数の効率的最適化のための多峰性の把握.	85
4.5	数値解析モデルを活用したトランスミッション制御パラメータ最適化システム.	88
4.6	TM 制御のメタ最適化システム (番号入り).	89
4.7	制御パラメータにより定められるクラッチ油圧波形の例.	93
4.8	評価指標 $v_{m,5}$ の 3 次元部分空間における可視化.	95
4.9	平均値と最小値の 2 パターンでの目的関数の学習曲線.	96
4.10	単一運転条件を評価する目的関数 $g_k(\mathbf{x})$ .	97
4.11	複数運転条件を評価する目的関数 $f(\mathbf{x})$ .	98
4.12	多峰性を持つ 2 次元トイプロBLEMにおける HSO の最適化過程.	100
4.13	単一運転条件下における最適化の学習曲線.	102
4.14	10,000 サンプルにわたる最適化の最終結果 ( $n = 5$ ).	103
4.15	各手法により最適化された加速度波形.	103
4.16	学習曲線と下位方策の中心が与える目的関数値の推移.	104
4.17	t-SNE [43] により次元圧縮したサンプルの分布とサンプルの特性.	105
4.18	異なる重み付けにおいて最適化されたパラメータの与える物理量.	106
4.19	複数運転条件下における最適化の学習曲線.	107
4.20	複数運転条件に対して最適化されたパラメータの与える油圧波形.	107

4.21	ダンプトラックの実機と座標系の設定. . . . .	109
4.22	数値解析モデルと実機間の誤差の観察. . . . .	110
4.23	出力軸角加速度とキャビン上で計測された 3 軸加速度. . . . .	110
4.24	現行パラメータと最適化されたパラメータにおける加速度比較. . . . .	111
5.1	生産加工技術の種類. (文献 [94] を元に作成) . . . . .	119
5.2	研削加工の原理. (文献 [96] を元に作成) . . . . .	120
5.3	円筒研削加工のメタ最適化システム (番号入り). . . . .	128
5.4	実機の少数データを用いた数値解析モデルの補正. . . . .	131
5.5	利用した円筒研削盤 RF32 (三菱重工業株式会社製). . . . .	132
5.6	砥粒と被削材の接触を模擬したシミュレーション. . . . .	132
5.7	砥粒と表面点群の時間変化. . . . .	132
5.8	汎用シミュレータ高速化のための多階層の接触粒子絞り込み. . . . .	134
5.9	事前計算される転写形状テンプレート. . . . .	136
5.10	テンプレートの連続適用による加工の模擬. . . . .	137
5.11	最適化を行う加工設定. . . . .	141
5.12	加工パスの時間・砥石位置系での表現. . . . .	142
5.13	計算機上での最適化システム. . . . .	144
5.14	計算機上での最適化の学習曲線. . . . .	144
5.15	最適化進行にともなう加工パスの変化. . . . .	145
5.16	数値解析モデルの修正機構を持つ最適化アルゴリズムのプロセス. . . . .	148
5.17	実機実験における目的関数値算出のプロセス. . . . .	150
5.18	加工後の直径の実測値の推移. . . . .	150
5.19	加工後の表面粗さと時間の実測値の推移. . . . .	151
5.20	加工性能を評価する目的関数の推移. . . . .	152
A.1	HSO の最適化プロセスの概念図. . . . .	161
A.2	2 つのトイプロブレムの関数形状. . . . .	165
A.3	3 つの峰を持つ 2 次元トイプロブレムでの学習曲線. . . . .	166
A.4	Ackley 関数での学習曲線. . . . .	166
A.5	3 つの峰を持つ 2 次元トイプロブレムにおける HSO の最適化過程. . . . .	167
B.1	二種類の時間軸. . . . .	169
B.2	知識の転移の概念図. . . . .	171
B.3	一変数関数のタスクとして用いる関数. . . . .	172
B.4	Search procedure transfer アルゴリズム. . . . .	173

---

B.5	Domain randomization の適用イメージ. . . . .	174
B.6	探索手順の具体例. . . . .	177
B.7	一変数関数タスクにおける各アルゴリズムの探索性能. . . . .	178
B.8	多変数関数タスクにおける各アルゴリズムの探索性能. . . . .	178
B.9	手順候補数が探索性能におよぼす効果. . . . .	179
B.10	複数のドメインを利用した実験の設定. . . . .	180
B.11	Domain randomization の適用結果. . . . .	181

# 表目次

2.1	3 形態のモデル化の特性比較表. . . . .	38
2.2	機械工学の分野における最適化を行った研究. . . . .	47
4.1	トランスミッションの制御最適化の関連研究において採用されている手 法. . . . .	82
4.2	異なる重み付けにおいて最適化されたパラメータの各要素. . . . .	105
4.3	実機評価実験における実験条件. . . . .	109
5.1	研削加工最適化の関連研究において採用されている手法. . . . .	123
5.2	研削加工最適化の関連研究における最適化の設定. . . . .	123
B.1	タスクとして生成された 2 変数関数. . . . .	175

# 第 1 章

## 緒論

本研究においては、「数値解析技術を活用した実在する機械システムの設計・制御パラメータ最適化」という問題を取り扱う。本章においては研究課題の社会的背景や提案手法の技術的背景について述べた後、研究目的について述べる。また、論文中の現在や現時点という表現は、2020 年執筆時のことを指す。

### 1.1 研究背景

本節では、以降の章・節における研究の背景及び議論の前提の共有を目的として、機械の設計に必要な考え方、計算機の登場による設計方法の変化について説明する。

#### 1.1.1 機械とその設計手法

本項では機械の設計にあたり必要な考え方、知識について述べる。

##### 機械の定義

まず本論文において扱う機械の定義を行う。ブリタニカ国際大百科事典の小項目事典 [1] には、機械とは「人間に有用な目的を達成するために、物体を組合せてつくり、これらの各部に所定の機能を与え、全体の機能を実現させたもの」と定義されている。本論文でもこの定義を採用する。人間は古来から身の回りで発生する様々な物理現象を利用した道具や機械を作ってきた。例えば、荷車は軸受けと円形の車輪を利用することにより、摩擦を低減することができ、楽に荷物を運ぶことができる。水車は川のようなありふれた水の流れを回転の駆動力とし、脱穀などの単調な作業を行うための力を取り出すことができる。ニューコメンの初期の蒸気機関は燃料を燃やした際の熱により液体の水を蒸気へ変え、蒸気が再び水に戻る時の圧力低下を利用して大きな力を出すことができる [2]。

18 世紀末の産業革命以来、機械産業は社会において重要な役割を果たしている [2]。



2019 年の日本企業の売上高 TOP100 のランキング [3] を見ても上位に移動・輸送，発電，建設，農業のための機械を作るメーカーが存在しており，機械産業は人々の生活と多くの点で結びついているといえる．本研究では昨今の機械産業が扱うような動力機械，作業機械を最適化の対象とする．本論文内で，機械という言葉は特に断りがない場合，動力機械，作業機械のことを指す．

### 設計の定義

上記のような機械を製作するためには，設計を行う必要がある．ここでは機械の設計について述べる．機械設計の教科書として広く利用されている実際の設計 [4] には設計の定義について，「まだ世の中にない新しいものを生み出すときには，欲しいものが何かを具体的に考えること，決めること，それを伝えるための情報を作ることが必要になる．これらの活動のすべてを設計であると定義する．」とある．言い換えると，設計は人間の行う知的活動であり，機械の利用する相手（ユーザ）が欲している機能を理解・定義し，その機能を実現するための機構（仕組み）を考案することである．設計においては，観測が難しい相手のニーズを推察したり，複数の選択肢から有用なものを選択したり，人と交渉したり，資料を読み書きするといった異なるタイプの知的作業を行う必要がある．また，実際の設計 [4] によると，「機械の価値は機械が持つ機能によって決定されるため，機能の設計は上流工程に位置付けられ，設計者は重要な役割を果たす．」とある．設計者が行う設計行為そのもの [5] や，設計行為の支援方法 [6][7][8] は研究の対象とされてきた．本研究も機械設計の支援方法に関する研究である．設計とその支援方法の議論のために，ここからは機械の設計に求められるもの，設計者に求められる知識について詳細に紹介していく．

### 設計に対する要求

機械の設計に対しては，社会的要請やユーザの便益に由来する様々な要求が存在する．以下は要求の具体例である．

- 時間がかかってはいけない
- 低コストに製造できないといけない
- 低コストに維持できないといけない
- 安全でないといけない
- 崩壊してはいけない
- 省スペースでないといけない
- 振動があってはいけない
- 製造できないといけない
- 環境負荷が小さくないといけない

- 機械の主な機能を発揮できないといけない (自動車の場合、直進できる、方向を変えることができる、荷物を載せることができる等)

機械の製造に入る前に、これらの要求をきちんと把握し、要求を高い水準で満たす良い設計を考えること重要である [4]。良い設計を考えることにより、製品の要求が良く満たされるようになり、市場における競争力につながるため、設計行為は製品の製造の上流に位置しており、設計のために多くの人員が割かれている。

上記の「低コストに製造できなくてはならないという要求」の中で表されるコストには、機械の材料費、製造費はもちろん含まれるが、設計時の人的コストも含まれる。人的コストをかけずに製品の品質を維持するためには、知識を効率良く蓄積することが必要である。そのためには、色々な対象に適用できる一般的な知識を設計現場において積み上げていくことが大事である。次小項からは設計者に求められる知識について述べていく。

### 設計に必要な 3 つの知識

設計者に求められる知識は、その利用のされ方や獲得のされ方から図 1.1 のように、3 つの知識に大別することができる。

1. 物理現象に関する知識
2. 機構の組み合わせ方に関する知識
3. 設計を進めるためのメタな知識

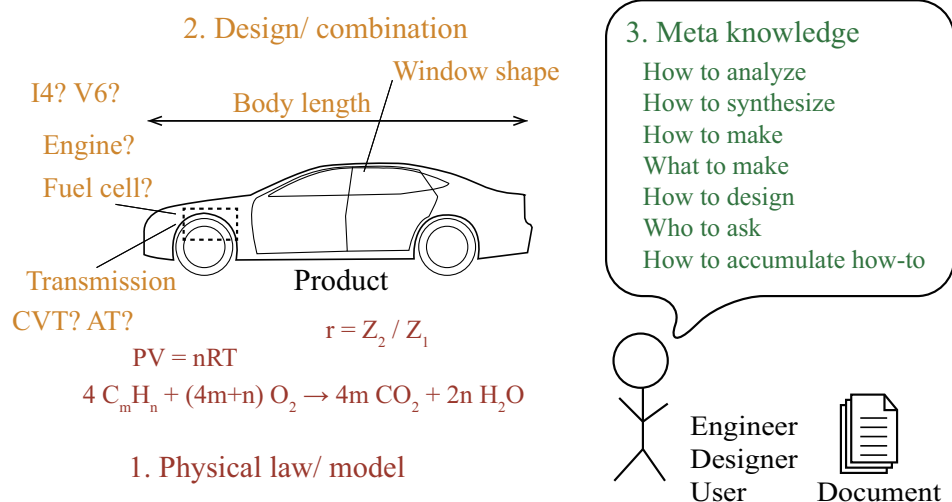


図 1.1 設計に必要な 3 つの知識.

このうち 1 の物理現象に関する知識と 2 の組み合わせ方に関する知識は、設計対象の機械に関する知識である。一方で、3 の設計を進めるためのメタな知識は、設計をする人間

や、周りの社会に関する知識である。これらの 3 つの知識について本小項において簡単に触れた後、次小項以降で説明をしていく。また、本論文においては「メタ」という接頭語を、「俯瞰」や「高次の見方」という意味で用いる。例えば、3 つ目の知識のメタな知識とは、設計行為を「俯瞰的」に見たときの、有用な法則や有用な行為に関する情報のことを指す。

1 つ目の物理現象に関する知識は機械の動作に伴う物理現象を記述したものである。本論文内では文献 [9] の定義にならい、現象を記述するものを「モデル」、もしくは「物理現象モデル」と呼ぶ。物理現象モデルの中でも一般に認知されているものは「法則」と呼ばれることもあり、等号を含んだ方程式の形で表されることも多い。

2 つめの組み合わせ方に関する知識は、機能を達成するための物理現象の組み合わせ方に関する知識である。ほとんどの設計においては相異なる複数の目標が存在する。複数の目標を同時に達成するためには多くの場合、複数の物理現象を考慮する必要がある。機構同士が組み合わさった際にどのような挙動を示すか、機能を達成するための現象の組み合わせにはどのようなものがあるかといった知識がこちらに属する。

これらの 2 つの知識を行使する活動のことを Analysis (分析) と Synthesis (統合) と呼ぶこともあり [10]、大学の機械工学の教育の場でも 2 つの知識をバランスよくつけることが必要とされている [11]。設計においては、現象を分析することにより抽象的なモデルを獲得し、現象を利用する機構を組み合わせ目的の機能を達成させる。

一方で、設計においては 3 つめの設計を進めるためのメタ (超越的) な知識も必要となる。この 3 つ目の知識には、機械を欲するユーザのニーズや、分析と統合をどのようなバランスで進めれば利益をもたらすことができるか、どのような分析を行えばユーザのニーズを素早く把握することができるか、役割分担はどのように行うべきかといった知識が属する。

特定の学術分野が具体的にどこに所属するかは一概に決めることができない。あえて分ければ、理学では 1, 3, 工学では 2, 3 が重要視され则认为られる。3 つめの知識が 2 つめの知識と一緒に語られることもあるが、本研究では、設計対象とその外部との切れ目にも注目し、2 と 3 を分けて議論する。

## 1. 物理現象に関する知識

1 つ目の物理現象に関する知識は機械の動作に伴う物理現象を記述したものである。

記述に対しては、発生している物理現象を簡潔な表現で、高い精度で近似できるものがよいとされる。一部の知識は方程式・法則という名前がついて、様々な場面で使われるようになる。一方で、現象を精度良く記述していたとしても、必要とされない知識は語り継がれることがなくなり、淘汰されていく。以下に有名な法則の例を示す。

- ニュートンの運動方程式： $F = ma$
- 運動量保存則： $\sum_{i=1}^N p_i = \text{const}$
- ケプラーの第 1 法則： $r = \frac{l}{1+\epsilon \cos \theta}$
- 静止摩擦力の式： $F \leq \mu N$
- 弾性曲線方程式： $\frac{d^2 \delta}{dx^2} = -\frac{M}{EI}$
- 片持ち梁に先端荷重をかけたときのたわみ： $\delta = \frac{Pl^3}{3EI}$
- ナビエ・ストークス方程式： $\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \nabla) \mathbf{v} = -\frac{1}{\rho} \text{grad} p + \nu \Delta \mathbf{v} + \mathbf{g}$

これらの知識は影響力や再利用性が高く、様々な場面で利用される。例えば、運動量保存則は  $F = 0$  としたニュートンの運動方程式を積分することで得られ、流体の挙動を記述するナビエ・ストークス方程式は運動量保存則を使って導かれる。

またこれらの方程式は身の回りで発生する現象を全て説明できるようにはできていない。全ての現象を式の形に落とし込めていないためか、もしくは解析の容易さや表記の簡潔さを重視して、近似が行われているからである。例えば、ベルヌーイ・オイラーの仮定は荷重をうけた梁の断面のせん断変形を 0 とするという仮定である。実際にはせん断変形がわずかに発生するが、軸方向の応力による変形に比べて、非常に小さい。この仮定をおくことで上記の弾性曲線方程式が成り立つようになり、梁のたわみが積分により求められるようになる。また、ナビエ・ストークス方程式も上記の形式(一般形)では一般解を求めることが難しいと言われており、数学界の有名な未解決問題の一つになっている。そのようなナビエ・ストークス方程式も、境界層近似、非圧縮性、定常流、一次元流れといった仮定を置くことで解を求められるようになる。

物理現象に関する知識の中で、一般的な対象、探求することで多くの人に利益が及ぶような対象は、設計者ではなく、研究者により探求が行われる。物理現象に関する知識は機械設計を進めていく上で基盤となる知識である。

## 2. 組み合わせ方に関する知識

2 つめの組み合わせ方に関する知識は、機能を達成するための物理現象の組み合わせ方に関する知識である。

設計においては、様々な要求を満たすような機構の組み合わせの探求を目指す。設計者は機械に対する複数の要求、機械で発生する複数の物理現象に目を向けないとならない。例えば、自動車は移動をするための機械であり、移動は一般的に短い時間で済ませられることが好ましいとされる。しかし、現状の自動車業界は、ひたすら最高速度を追及しているわけではない。移動時間を短くするために最高速度を上げたいならば、エンジンの出力を向上させたり、ジェットエンジンを搭載したりと様々な実現方法が考えられる。最高速度の値が伸びない背景には、複数の要求の存在がある。最高速度を上げていこうとする

と、高性能なエンジンの導入コスト、低速で走行した際の燃費が悪化、車の省スペース性を損なう、ドライバーが安全に操縦できなくなるなどいくつかの障害にぶつかる。これらの障害は、低コスト、小型、安全という他の要求に起因する。

このように物理的に形がある機械を成り立たせるためには、様々な要求を同時に考えなくてはならない。機能を満たすためにどのような機構を選んで組み合わせればよいか、機構同士の組み合わせのバランスはどうすればよいかという知識が、2 つめの組み合わせ方に関する知識である。現代社会に欠かせない機械となっている自動車は複数ある要求をうまくバランスで満たす組み合わせの元に成り立っているといえる。本論文では、組み合わせ方、組み合わせのバランスを決める値を「設計パラメータ」もしくは単に「パラメータ」と呼ぶ。機械の構成にある機構を含めるか否かのような離散的な情報も、0/1 の 2 値に対応させることにより、パラメータに含めることができる。

自動車の例では、バランスを調整した結果、車体の重さ、エンジンの出力、乗員数などの値が定まる。2 番目の組み合わせ方の知識は、問題ごとに特有な解・解決策に陥りやすい。問題への要求もしくは制約が変わることにより、自然と解決策が複雑になり、それだけ設計のための手数が増える。例えば、自動車の例でいうと、国ごとの地形や民族性、気候によって要求や制約が変わるため、国に応じた設計のカスタマイズが必要であるといわれる [12][13]。同様の理由から日本企業が製造するエレクトロニクス製品は日本市場に特化した設計になっているといわれ、この事実は設計のガラパゴス化という名称で呼ばれる [13]。

### 3. メタな知識

3 つめの知識は設計を進めるためのメタな知識である。1 つめ、2 つめの知識は、機械やその背後の現象と深く結びついているものであったが、この知識は設計をする人間や、周りの社会と深く結びついている。

3 つめの知識としては、以下のようなものがある。

- 機械を利用するユーザのニーズ
- ユーザのニーズの把握方法
- 現象の分析の方法
- 良い組み合わせを効率良く見つける方法
- 設計チームの育て方
- チームとして開発すべき製品
- 良い KPI (Key Performance Index) の設定
- 人間に対する交渉力

ここでは 4 番目の良い組み合わせを効率良く見つけるための知識の具体例として、設

計パラメータを調整する方法について紹介する。設計者は現象を記述するモデルを頭に入れた状態で、用いる現象を選択し、現象間のバランスをとるためにパラメータを決定していく。現象を記述するモデルが不透明だったり、考慮できていない現象を洗い出すため、必要に応じて、設計した機械を用いた実験を行い現象に関する知識を収集していく。そして、設計と実験を交互に繰り返しながら、現象への理解を深め、良いパラメータを見つけ出す。現象の組み合わせやパラメータ調整を上手く行い、同様のコストで新しい機能を実現したり、少ないコストで既存の機能を達成することが商品の価値につながる。この場合のコストとは機械の材料費、製造費はもちろん含まれるが、設計時の人的コストも含まれる。

このようなメタな知識は一人の設計者が全てを身につけなくてはならないわけではないが、設計をするチーム全体としてこのような知識が蓄積されていることが望ましい。列挙した中でのチームの育て方や、チームとして何を開発するべきかというのは、人間の視点、人生の視点にたって考えなくてはならない。これらは数値的合理性だけではなく、人間に対する説明可能性や人間の理解可能性を考慮しなくてはならないため、最適な行動を決めるのがしばしば困難になる。

メタな知識は設計した製品に表出しない知識でもある。製品はリバースエンジニアリングにより、機構の構成や設計パラメータが詳らかにされる。1 つめの物理現象の知識や 2 つめの組み合わせ方に関する知識はリバースエンジニアリングにより容易に見抜かれしまう。一方で、機構の製造方法や、パラメータ調整の方法、技術検討の結果は、リバースエンジニアリングからは容易には見抜かれない。メタな知識は容易には見抜かれないが、製品の競争力にはつながるため、他社に先駆けて蓄積しておくことで優位に立つことができる。加えて、メタな知識は特定の製品と結びついていないことから、一般性が高い知識であるともいえる。

### 1.1.2 情報技術の進展と設計手法の変化

計算機をはじめとした情報技術によって設計の効率は大きく向上し続けている。背後には計算機性能の指数関数的向上とそれに伴う技術革新がある。本項においては、情報技術がどう発展し、設計にどのように影響を及ぼしてきたのかについて述べる。

#### 計算機性能の指数関数的向上

ほとんどの計算機の計算は半導体回路上の回路素子で行われているため、半導体回路上の素子の数は計算機の性能に対して支配的影響力を持つ。計算機はその有用性から、人間社会の様々な活動に組み込まれ、図 1.2 のように 60 年もの間、たえず進化を続けてきた。図 1.2 左のグラフのように回路上の素子密度は 2 年におよそ 2 倍のスピードで指数関数的

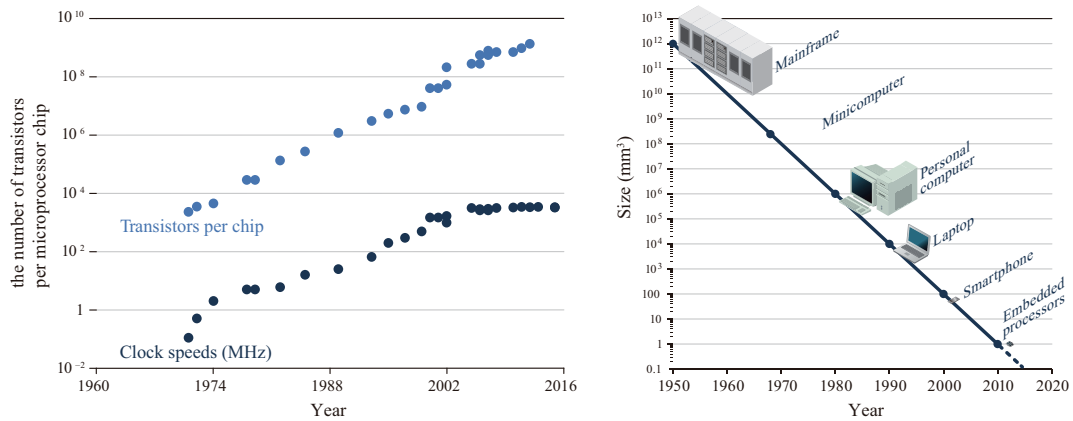


図 1.2 ムーアの法則と計算機の進化. (文献 [14] を元に作成)

に増加し続けており，代表的な回路の性能はきれいに直線に載る [15][14]. この事実は指摘者の名前をとってムーアの法則と呼ばれる [14].

半導体素子密度の向上だけではなく，効率の良いアルゴリズム，特定の種類の計算に特化した計算回路の登場も計算効率の向上に寄与している．例えば，科学計算のさまざまな場面で登場する行列計算もアルゴリズムの工夫により効率化されてきた [16].  $n \times n$  の正方行列の乗算は愚直に行うと  $O(n^3)$  の計算量が必要である．図 1.3 はこの  $n^3$  を低減させるために開発されてきたアルゴリズムの性能の推移を示す．1960 年から 2000 年の間に  $O(n^3)$  から  $O(n^{2.4})$  まで減らすことに成功している．

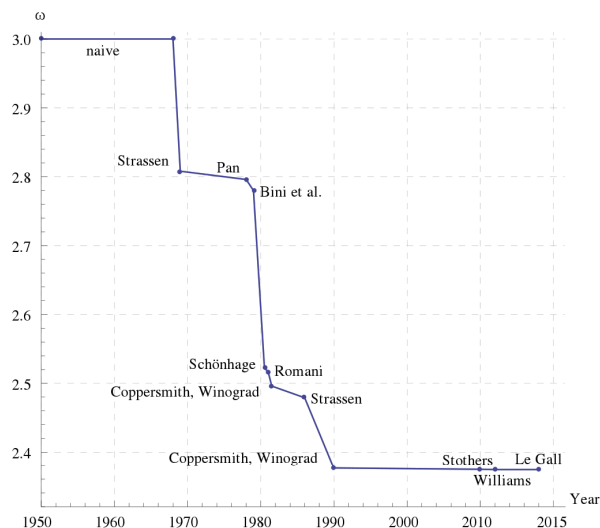


図 1.3 行列積計算アルゴリズムの性能向上 [16].

また，GPU (Graphic Processing Unit) は特定の種類の計算に特化した計算回路である．GPU は汎用的な計算回路である CPU (Central Processing Unit) とは異なり，画像処理に

特化した計算回路であり，単純だが計算量の多い 3 次元計算や行列計算を高速に行いたいときに利用される．また，CPU は汎用的な計算を行うことができるが，世の中で行われる計算の特性の偏りを利用して，CPU 自身の構成のチューニングが行われることもある．具体的には，データを効率良く処理するための命令セットが開発されたり，データの転送コストを低減するためにキャッシュメモリの種類が増加することがあった．

半導体回路の物理的制約により，素子密度の成長速度が鈍り上記のムーアの法則は成り立たなくなるといわれている．しかし，素子密度を変えずに計算機性能を高めるために，計算機の構成や半導体回路の設計をアルゴリズムによって最適化するという研究も行われており [17]，しばらく計算機性能の指数関数的な成長は続くと考えられる．

また，計算機の有用性は計算機への研究開発投資を活発にし，計算機の性能の向上に寄与することがある．図 1.4 は 1960 年から 2020 年の間の代表的なアルゴリズムの動作に必要な計算性能の増加を示した図である．2012 年頃からはムーアの法則の傾きを上回る性能の増加が起きている [18]．これは近年の人工知能技術への期待を反映して，投資が進んだ結果であると考えられる．この背景には，半導体回路の素子密度の向上が鈍化したとしても，半導体回路自体の数を増やすことにより計算能力を向上させることができるという事実がある．

このように情報技術とそれを支える計算機の性能は成長を続けている．機械設計の観点から見たとき，情報技術及び計算機の有用性には以下の二つがあると考えられる．

1. 物理現象を再現する能力
2. 自律的かつ知的な能力

ここからはこの二つ有用性について詳細に述べていく．

### 物理現象の再現能力

計算機と物理現象に関する知識とを組み合わせると，計算機上で物理現象を再現することができる．初期の電子計算機である ENIAC (Electronic Numerical Integrator and Computer, 図 1.5) は弾道計算を行うことが第一の目標として設計された [19][20]．計算機上での物理現象の再現方法としては，記号で書かれた方程式に，具体的な数値を入れていき結果を計算するものが主流である．この計算は，数値計算や数値解析，数値シミュレーションとも呼ばれる．一方で，計算機が記号で書かれた式を直接扱う自動推論，自動定理証明などの分野も存在し，これらは電子回路の設計などに利用されることもある．数値シミュレーションは創薬，気象予報，機械の設計など様々な分野 [21] において利用され，様々な面で人々の暮らしと結びついている．気象予報や，災害の被害予測は，未だに現実世界で発生していない事象を計算機上で予測するものであり，人々はこの予測をもとに対策を打つことができる．



## Two Distinct Eras of Compute Usage in Training AI Systems

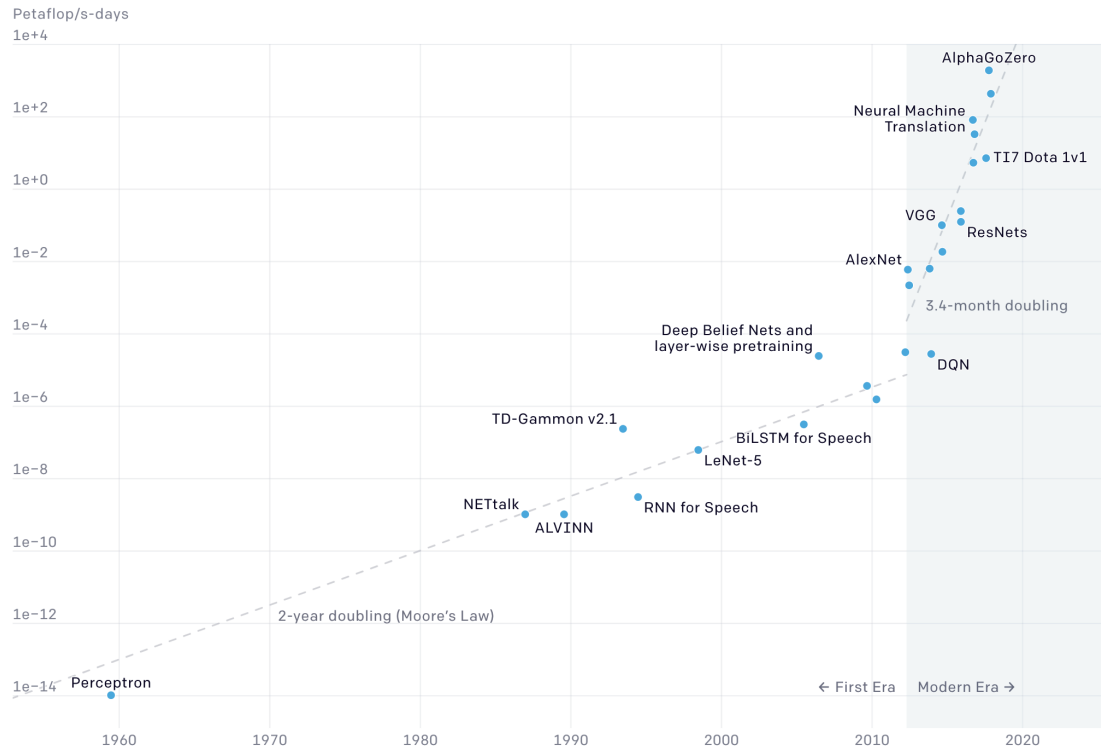


図 1.4 人工知能アルゴリズムの動作に必要な計算性能 [18].

「シミュレーション」という言葉自身は人間が脳内で行う予行演習の意味も含む。そのため、本論文内では、計算機上における物理現象の再現行為を「数値シミュレーション」と呼び、人間の行う「シミュレーション」と区別して用いる。また本論文内では、「数値シミュレーション」とほぼ同じ意味で「数値解析モデル」という言葉を用いる。本研究は、2章において後述するが、現象を再現するモデルの性質に注目している研究である。そのため、基本的には、「数値シミュレーション」と比べると世の中で利用される頻度が少ない「数値解析モデル」という言葉を用いる。一方で、論文内の説明では読みやすさのために「数値シミュレーション」という言葉を使うこともある。

数値シミュレーションの登場と発展により、数値解析学という分野も生まれた。数値解析学では現象を計算機上で高速に精度よく計算するための手法を研究する。

数値解析手法の一部は構造解析、流体解析などの名前と呼ばれ、機械設計にも利用されている。これらの数値解析手法は CAE (Computer Aided Engineering) とも呼ばれる。数値解析が実用化される前は、設計した機械構造の機能を確認するためには、記号で書かれた式を用いるか、実際に構造を製作して実機実験を行うしかなかった。そのため記号で書かれた式に載せることが難しい複雑な形状は、実機実験を行わざるを得ず、コストの面か

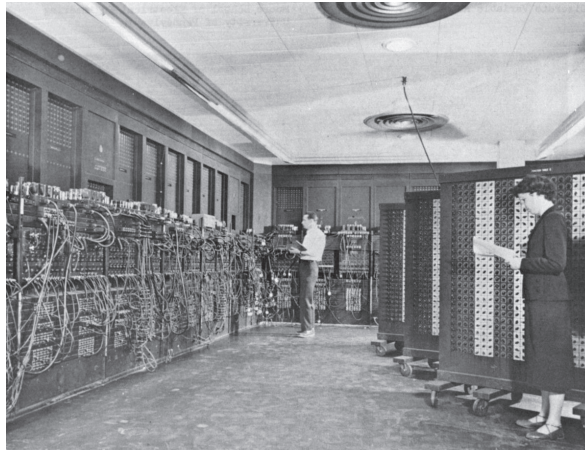


図 1.5 初期の電子計算機 ENIAC [22].

ら開発が敬遠された。

数値解析の実用化，性能向上により，計算機上で事前に実験を行えるものが増え，複雑な形状を設計できるようになった．例えば，図 1.6 のように翼上面にエンジンを設置することにより有害な抗力を減らすという画期的な機構を備えた HondaJet の設計においては，三次元オイラー解法により数値解析を行って計算機上で事前の設計検討を行っている [23][24][12]．このような，設計過程において数値シミュレーションを用いる設計方法はモデルベース設計と呼ばれることもある．

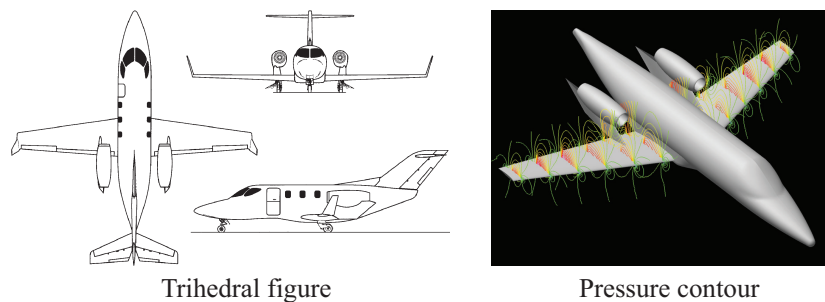


図 1.6 HondaJet のエンジン翼上配置 [23].

数値解析においては，必ずしも起こりうる全ての物理現象を記述して，計算機上で再現する必要はない．1.1.1 項の物理現象に関する知識と同じく，影響度が大きいもの，解析が容易なものが記述されて，計算機上で再現されてきた．

機械設計のための数値解析は現象の現象の再現の方法から 3D-CAE と 1D-CAE の 2 種類に分けることができる [25][9]．図 1.7 は 3D-CAE と 1D-CAE によるバネのモデル化の様子である．3D-CAE は格子間・粒子間の関係を記述した方程式をベースに，機械要素の挙動を模擬するものである．3D-CAE の具体例としては，有限要素法による機械構造にか

かる応力の計算や、粒子法による流体の速度分布の計算などがある。

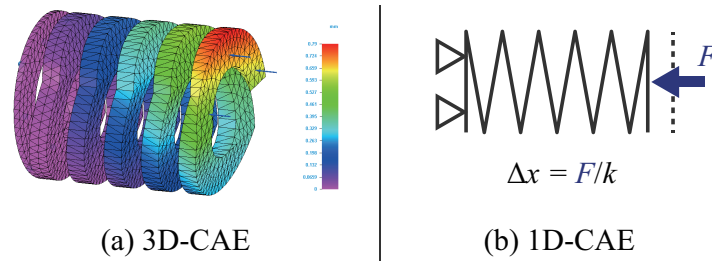


図 1.7 バネのモデル化の 2 手法.

一方で 1D-CAE は図 1.8 のように機械要素の機能を記述した方程式をベースに、機械要素を組み上げた際の機能を模擬するものである。1D-CAE は機械要素の機能が単純な式により表されることを利用している。例えば、バネは変位量が復元力に比例するというフックの法則  $F = -kx$  を満たす部品、歯車は歯数から決まる変速比  $r = \frac{Z_2}{Z_1}$  で回転を変換する部品というように、式で要素の機能を表現することができる。このように、個別の機械要素の機能は人間が扱いやすいように記号化されている。そのため人間は、力を蓄えたい機構にはバネを使う、変速をしたい機構には歯車を使うというようにパズルを解くように設計をすすめることができる。半分バネの特性を持ち、半分歯車の特性を持つような中間的な特性を持つ部品は存在しうるかもしれないが、そのような部品が登場した途端、パズルのようには解けず、設計の自由度が増大してしまい人間には設計が難しくなる。1D-CAE は人間がパズルのように組み立てた機械システムの概念実証を行うために利用される。

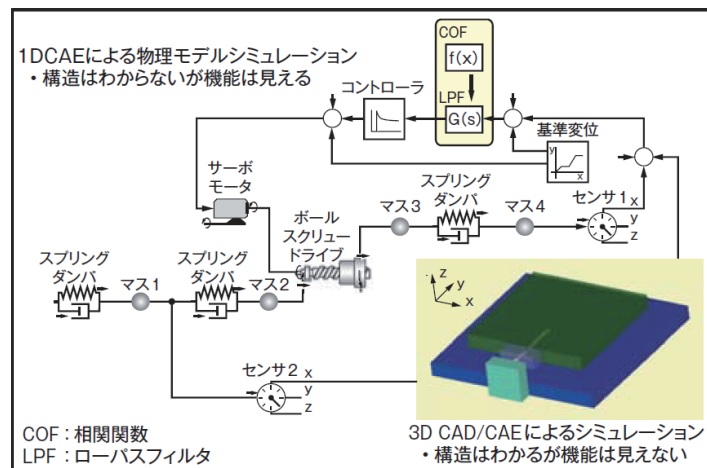


図 1.8 1D-CAE のコンセプト [25].

1D-CAE は機械要素の特性がいくつかに分かれること、そしてそれらの特性が簡単な式で表されることを利用して、少ない計算量で機械の挙動を模擬する手法である。図 1.7(a) のように、バネの弾性力を 3D-CAE の手法で模擬することも可能であるが、計算量が膨

大になりフックの法則  $F = -kx$  を再現することには向かない．3D-CAE と 1D-CAE の例のように，物理現象の切り取り方をうまく選ぶことにより，現実の機械の挙動を少ない計算量で高い精度で模擬することができる．

1 次元空間や 2 次元空間における格子間・粒子間の方程式をベースにした計算も 3D-CAE に含まれる．3 次元での計算が可能であれば 1 次元，2 次元でも計算が可能になると，及び，1D-CAE との対比をしやすいするため，格子間・粒子間の数値シミュレーションは 3D-CAE と呼ばれる．

このように計算機による数値解析には様々な形態が存在し，多くの分野で役にたっているが，自然現象の全てを再現することはできないことや，量子化や数値計算上の誤差により，模擬された現象と実際の現象には誤差が生まれる．この誤差の存在は問題とされており，気象予報において数値解析結果と現実のデータの両方を活かすためのデータ同化 [26] や，ロボットの制御則を数値シミュレーション内である程度ロバストに学習するための Domain randomization [27] などの，近似誤差があることを前提にその対策をする手法が登場している．

数値解析手法の開発者は誤差を減らすような数値解析手法の開発が求められ，設計者は数値解析に誤差が発生することを把握したうえで設計に活かしていくことが求められる．

### 自律的かつ知的な能力

計算機はプログラムにより動作する．計算機が登場してから，人間がプログラムを組むことにより，人間の知的な活動を計算機に代替させる取り組みが行われてきた．計算機上のプログラムの特徴として，繰り返し作業にも疲れを感じることがなく，安定した動作をするという点がある．また，プログラムの動作の中で想定していない挙動があったとしても，作業途中のデータが残すことが可能であり，人間による原因の追究が可能である．そのため，人間の知的活動の中でも問題設定が単純な物から代替が進んでいる．近年では機械学習技術の進化により，人間の判断のように入出力関係が複雑な関数を近似することができるようになり，代替に拍車がかかっているといわれている [28]．

人間のような知的作業を行う計算機上のプログラムやアルゴリズムは人工知能と呼ばれ，人工知能を研究する学問分野も存在している．人工知能 (Artificial Intelligence) という言葉は 1956 年のダートマス会議で定義され，現在に至るまで様々なアプローチで研究がなされている．人工知能の研究は，人間の知性の研究とも大きく関係してきた．研究者達は，既に知性があるとされる人間の知性の根源や知性の実現方法を参考に，人工知能の手法の研究を行ってきた．人工知能研究は注目する知性の根源の違いによって 2 つの分類がある [29]．人間の知性が記号を使った論理の力を根源としているとする立場は「記号主義」と呼ばれ，脳やネットワークの構造に由来する認識の力を根源とする立場は「コネクショニズム」と呼ばれる．人工知能研究はこの両方の立場が互いに競い合うことによ

て発展してきた [28]. コネクショニズムの立場の手法の一例としては, 人間などの脳のニューロンの挙動を模擬した, ニューラルネットワークがある. ニューラルネットワークは非線形な関数を複数組み合わせることにより入出力関係が複雑な関数を近似することができ, 上記の機械学習分野の発展にも寄与している.

探索及び最適化は人工知能の黎明期から現在に至るまで研究されている人工知能の代表的な分野である [30]. この分野では, 問題や目標を少ない試行回数で解決したり達成する手法の開発が目標となってきた. 探索や最適化は多くの他分野に派生しており, 機械設計分野においてもトポロジ最適化 [31] や多目的最適化などの最適化手法が生み出されている.

人工知能の歴史を通して, 探索, 最適化アルゴリズムの性能評価のために, 我々に身近なゲームが利用されてきた [30]. ゲームの中でもボードゲームはルールが単純だが, 初心者から上級者までそれぞれのレベルに応じた楽しみかたができる豊かな遊びである. また, ボードの中のプレイヤーはとりうる行動がルールによって明確に規定されており, 行動の分岐の仕方が計算機上で簡単に計算できる. 分岐の仕方を繋げたものはゲーム木と呼ばれ, 人工知能研究ではこのゲーム木をいかに効率よく探索して, 良い行動を選ぶかが争点になってきた. 問題の難しさは, ルール上ありうる状態の数と関係といわれる. ありうる状態の数はゲーム木の大きさとも呼ばれ, チェスで  $10^{123}$ , 19 路盤の囲碁では  $10^{360}$  である. これらの数は, 全部の状態を記述して比較することは不可能であるほど大きく, 探索アルゴリズムの利用が必要になる.

計算機にテーブルゲームをプレイさせるというアイデアは人工知能という言葉が定義される以前からあり, 計算可能性理論を提案したアラン・チューリング, 情報理論を提案したクロード・シャノンもチェスのプログラムに関する提言をしている. チェスのアルゴリズムは計算機性能の進化, 探索アルゴリズムの進化により 1997 年に人間のトッププロを越えた. 1997 年に人間のトッププロを破った DeepBlue [32] は 1 秒間に 2 億手の先読みを行っている. このアルゴリズムには人間による盤面の評価の方法が組み込んであり, 明らかに意味のない手や自分が不利になる手などを探索対象から除外し, 探索すべきゲーム木の広さを狭めていた. ただ, ゲーム木の大きい囲碁のようなゲームでは人間のプロレベルに至らなかった. その後, 計算機性能の進化もあり, 2015 年には盤面の評価方法を自律的に身につける機能を備えた AlphaGo [33] が囲碁のトップ棋士を破った. このアルゴリズムには探索アルゴリズムの MCTS (Monte Carlo Tree Search) とニューラルネットワークの一種である CNN (Convolutional Neural Network) が用いられている. AlphaGo は盤面を評価するために人間の棋士の対局データを利用していたが, その後に登場した AlphaGoZero [34] では対局データなしでもアルゴリズムが自習することにより, AlphaGo を上回る性能を示した. AlphaGoZero は対局データなしの状況から, 40 日間で 2900 万回の自習の対局を行い, 図 1.9 に示すように 30 日目付近で人間のデータを用いている前

世代の AlphaGo の性能を上回った。そして，AlphaGoZero の後継である AlphaZero [35] は囲碁だけではなく，チェスや将棋に対しても人間の前知識を入力することなく，各々ゲームの最強と言われていたアルゴリズムの性能を上回った。探索アルゴリズムを組んだのは人間であるが，探索アルゴリズムを変えることなく様々なゲームに対して高い性能を示したため，汎用性がある知性が登場したといわれている [35]。アルゴリズムが人間に匹敵する性能を示して以降，将棋のトップ棋士の間では，人間である棋士がいかにアルゴリズムの探索結果を参考にして成長するかが重要になると言われている [36]。

AlphaGo シリーズ [33][34][35] ではプレイヤーの置かれた状態の好ましさを今までの経験をもとに評価し，行動の判断に用いるという手法である強化学習が用いられている。強化学習アルゴリズムは人間の学習の仕組みに着想を得ているアルゴリズムであり，「問題や目標を少ない試行回数で解決したり達成する手法」という意味では探索アルゴリズムに含まれる。強化学習はボードゲームの攻略のみならず，ロボットによる物体の把持 [37] やヘリコプタの操縦 [38]，電子回路の設計 [39] など様々な対象に対して良い性能を発揮している。

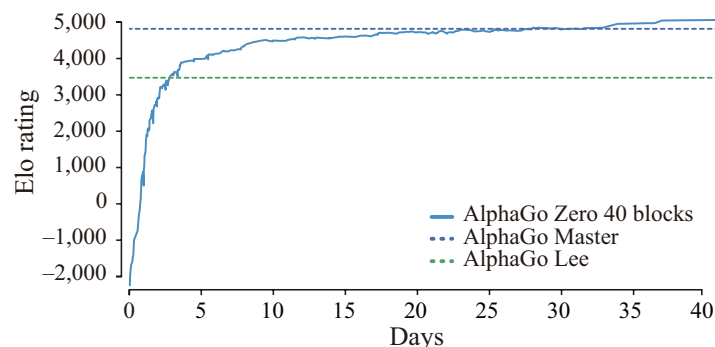


図 1.9 AlphaGoZero の性能向上過程 [34]。

文献 [37] は多自由度ロボットアームによる物体の把持の研究である。把持用のマニピュレータと物体との位置関係をカメラ画像から観測し，各時刻におけるマニピュレータの移動方向を決める，Visual servoing という手法を用いることにより，複雑に積み重なっている物体や，柔らかい不定形の物体の把持を可能としている。この論文では，ロボットに対して物体の把持方法は教えておらず，ロボットが試行錯誤の中で把持方法を獲得していく。把持方法の獲得のために，図 1.10 のように現実世界において 14 台のロボットを用意し，それらを 2 ヶ月間動かし続け 80 万試行のデータを集めている。

以上のような経緯から，探索アルゴリズムは，探索対象を人間が区切ることより人間に匹敵するもしくは人間を上回るパフォーマンスを発揮すると考えられている。区切るための境界のことをフレームと呼び，区切られた問題とは探索の目標である目的関数が定式化されており，とりうる行動が明確に決められている問題のことを指す。囲碁の例で説明す



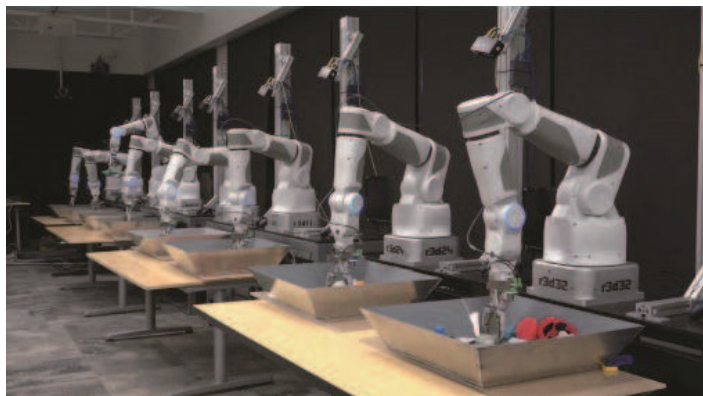


図 1.10 文献 [37] におけるロボットアームによる大規模データ収集.

ると、目的関数は勝敗であり，とりうる行動は囲碁のルール上許された打ち方である．また，探索アルゴリズムが性能を発揮するためには，囲碁 [34] や物体把持の学習 [37] の例のように，試行錯誤のための繰り返し試行ができるという要求も存在する．繰り返し試行が難しい対象の場合，性能を発揮できない可能性が高い．

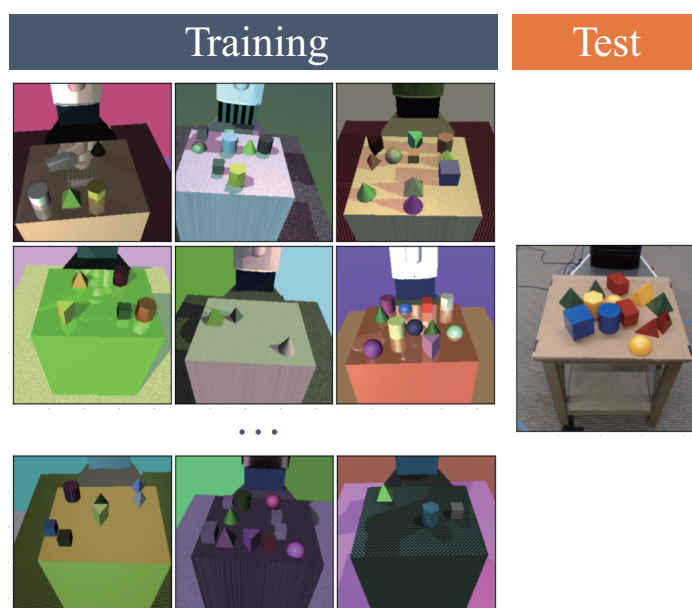


図 1.11 Domain randomization [27] のコンセプト.

前項で説明した Domain randomization (DR) は強化学習において用いられる 1 手法であり，DR を提案している文献 [27] においてはロボットの現実世界における物体把持タスクの攻略に用いられた．現実世界の環境を使って訓練すると，データ取得コストや，ロボットの故障リスクが問題となるため，計算機上の数値シミュレータ内で把持方策を学習する．その際に，現実世界の物体の色合いや影の落とし方を数値シミュレータ上で模擬する

ことは難しく、誤差が生じる。そこで、DR では現実世界の物体の色合いや影の落とし方は変動しうるが、その変動が起きたとして、行うべき把持行為は変化しないという事実を利用する。具体的には、DR では図 1.11 左のように数値シミュレータ内での色合いや影の落とし方をランダムに変動させることにより、色合いや影には意味がないということを暗にアルゴリズムに対して教え込み、現実世界 (図 1.11 右) の色合いや影の変動に頑健な把持方策を学習させた。

以上の 2 つの有用性以外に、情報を量子化し、容易に複製、伝送可能にしたことも設計における計算機の有用性と見なせる。現実世界の紙の図面は CAD (Computer Aided Design) ツール上の図面データになり管理がしやすくなり、デジタル動画により機械の動作を残し共有することが可能になり、インターネットによって設計法や部品のカatalogをあらゆる場所で参照できるようになった。これらもアナログな物理現象を計算機上のデジタルデータにより再現したものとみなすことができるため、1 つめの有用性と似ている。しかし、これらは設計される機械に内蔵されるようなものではなく、本論文の論旨とは関わりが薄いため、詳細は割愛する。

現代の設計現場においては、これらの計算機の有用性を組み合わせたサービスが登場している。例えば、図 1.12 のような機械加工製品の販売を行うミスミグループが提供する meviy (メヴィー) というサービスがある [40]。meviy を利用すると、加工部品の 3D データをインターネット上にアップロードするだけで、製造可否がアルゴリズムにより自動で判断され、価格が即時に見積もられ、最短 1 日で製品を受け取ることができる。よくある形状、材質に限定されてはいるものの、これまで人の手により行われてきた製造可否の判断、見積もりがアルゴリズムによって行われるようになった。

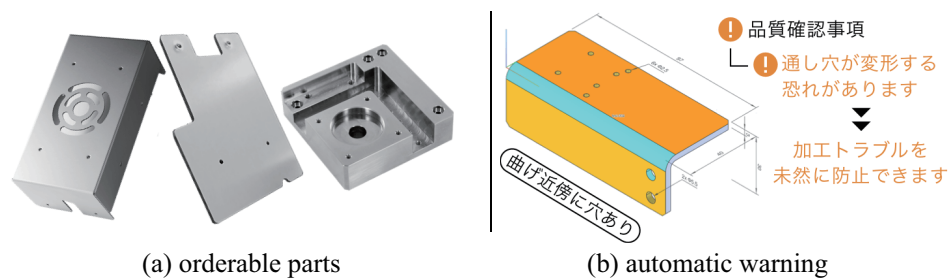


図 1.12 オンライン部品製造プラットフォーム meviy [40].

### 1.1.3 設計における知的作業の代替

上記のように、人工知能のアルゴリズムや計算環境が進化し、人間の知的活動の代替が進んでいる。本項では知的活動である設計行為の人工知能による代替可能性について考察する。設計において、人間もしくはアルゴリズムがはたしている役割について比較・議論



することは、効果の高い代替につながられるだけではなく、人間の知的能力の根源を明らかにすることにもつながると考えられるため、重要であると考えている。

まず初めに、人間の設計行為全体をアルゴリズムによって代替させることを考える。前述の通り、設計行為は知的活動であり、異なるタイプの知的作業を実施できる必要がある。設計行為には、既存の人間の知識体系を理解する能力や、設計を組み立てるための論理的思考能力が必要とされる。このような複数の異なる知的能力を持つアルゴリズムは汎用人工知能 (Artificial general intelligence) と呼ばれている [28]。汎用人工知能は現時点では実現されておらず、人工知能研究の大きな研究目標となっている。設計行為を完全に代替するような汎用人工知能が実現できない理由として以下の 3 つがあると考えられる。

- 人間の知識体系は人間が理解しやすいように最適化されており、宇宙人や AI のように異なる思考方法をとるエージェントにとっては理解が難しい。
- 人間の行っている知的タスクは報酬までが遠い。
  - 設計のようなタスクの報酬関数を設計するのは困難
  - 人間がやるような設計 (例: エンジンのターボファンの形状の改善) は自分の欲求 (例: 食欲) を満たすことに直結はしない。人間がやるような仮説検証で得られるものは知識であり、知識の価値を理解するのが難しい。
- 現実世界は複雑系であり、かつ繰り返し試行が難しい。繰り返し試行をするために、現実世界の複雑さを維持したまま計算機上の世界を作ること、方針が立てづらく難しい。

このような理由から、現時点での技術では汎用人工知能を実現することが出来ず、設計を完全自動化することは困難だと考えられる。

1.1.1 項において、設計者には 3 つの知識が必要であると述べた。3 番目のメタな知識が必要になる設計において目標とするものは、会社の利益や、人材の成長といったものも含まれ、人間にとっても定式化や数値化が難しく、条件を統制して試行回数を積むことも難しい。そのため、フレームを明確に区切った上で試行を積む必要があるような、現代の人工知能アルゴリズムが性能を発揮することは難しいと考えられる。

1 番目の知識に関する、Analysis の行為を人工知能に行わせようという試みは行われているものの [41][42]、複数の現象に関する知識を組み合わせるような論理的推論はできず、そのことから設計者や研究者の Analysis 行為の代替は遠いと考えられる。ただ、データから法則性を見抜くという Analysis の行為を行う際に、人工知能周辺分野のアルゴリズムが利用されることは多い。次元削減 [43] や関数同定 [41] という統計分野の手法はそれにあたる。例えば、文献 [43] で紹介されている t-SNE (t-distributed Stochastic Neighbor Embedding) は高次元空間のデータを、データ同士の位置関係を保持したまま、低次元に配置するという次元削減の手法である。次元削減は、データの特徴を分類するための有用

な特徴の抽出や、多次元空間におけるデータの位置関係を人間が把握しやすいように描画 [44][45] する際に用いられる。文献 [41] においては、関数同定という手法を利用して、振り子や調和振動子の運動の計測データから、記号でかけられた運動方程式を導くことに成功している。論文中的アルゴリズムは与えられた計測データをうまく近似できる式を見つけるという探索問題を解いている。このアルゴリズムでは、単振子の時の方程式立式方法をもとに、二重振り子の方程式を組むといった、人間の科学者がするような知識の流用は実現できない。また、振り子の運動において注目すべき物理量を人間が与えている点や、利用できる演算子の種類を人間が限定している点から、フレームが限られている探索問題であるといえる。近年では、文献 [42] のように、式変形が必要な数学の問題を解くアルゴリズムも登場している。論文中には言語翻訳のアルゴリズムを応用して、出題を回答に変換するアルゴリズムが提案されている。しかし、提案されている手法は厳密な数学を対象としていること、数学の中でも一部の問題に限られていることから、1.1.1 項で述べたような仮定が必要になる一般の自然現象のモデル化には適用が困難だと考えられる。

2 番目の知識に関する、Synthesis の行為の一部であれば代替の可能性があると考えられる。Synthesis の一部である設計パラメータの調整は最適化と同じ問題の構造をしている。また、数値解析技術により実機の実験が計算機上で模擬できるようになり、実験の条件設定や実験データの入力なども自動でできるようになった。そのため、実験を行って、データを集計し、良いパラメータを探索するという、設計におけるパラメータ調整は、繰り返し作業が得意な計算機上のアルゴリズムに担わせることができると考えられる。

また、アルゴリズムの探索性能も前述のように、適切なフレームに切り出すことができれば人間と同等もしくはそれ以上のパフォーマンスを発揮できると考えられる。フレームに切り出す際には設計に求められる複数の要求を把握し、目的関数として数値的に評価できる形に落とし込む必要がある。

3D-CAE は 1D-CAE に比べ解析時の要素数が多くなり、計算コストを要する。それに加え、最適化のためには、パラメータを変更して試行錯誤を行う必要があり、このことも計算コストを増大させる。3D-CAE 自体の計算コストが大きくなる問題に対しては、パターンの規則性を見抜くことができる機械学習アルゴリズムにより 3D-CAE の計算を簡略化するという研究も存在する [39][46]。文献 [46] は、粒子法による流体の挙動の数値シミュレーションを機械学習アルゴリズムを使って高速に計算するという研究である。多くの流体粒子をナビエ・ストークス方程式を満たすように時間発展させていくことは計算コストを要する。図 1.13(a) は高速化を行う前の結果であり、図 1.13(b) は高速化後の結果である。高速化により差異は出現するものの、視覚的に納得できる結果を提示することができている [46]。この結果から、文献 [46] の手法は人間の視覚を納得させるという意味においては、流体の挙動をモデル化することができているといえる。また、この高速化により、代表的な条件下において約 200 倍の高速化が達成されたと報告されている [46]。

これらの研究では、事前に通常の 3D-CAE 計算を実施し結果を記録しておき、以降はパターンマッチングにより記録しておいた結果を引き出して計算を簡略化する。繰り返し計算により計算量が増大する、3D-CAE を利用した最適化のためには、物理現象の特性をうまく見抜いて計算を簡略化することが特に有効だと考えられる。

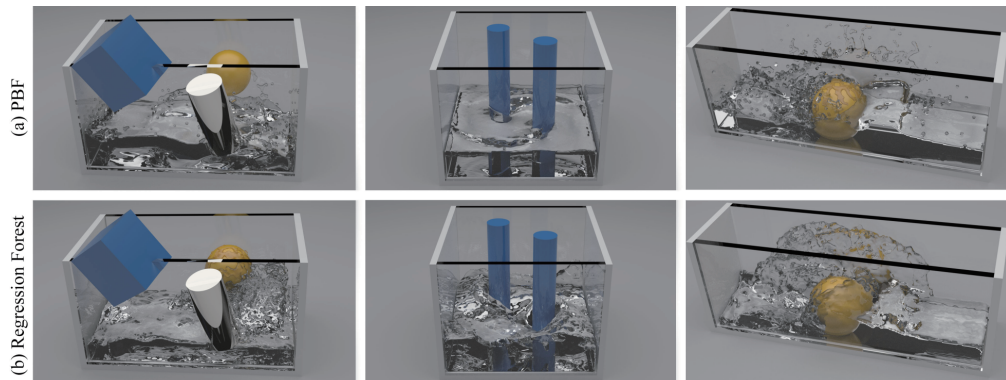


図 1.13 過去の計算結果に基づいた粒子法数値シミュレーションの高速化 [46].

上記の議論のように、本研究では、設計パラメータの調整の支援に探索/最適化アルゴリズムを利用することを考えている。その際には、人間とアルゴリズムの以下のような特性に留意して支援の仕組みを構築する必要がある。

#### ■技術者・科学者

- 設計パラメータが現象に及ぼす影響を、既にある知識から定性的に推測することができる。
- 具体例
  - － 小さいスケールでは流体抵抗などの表面力が卓越する
  - － 流線形は流体抵抗が少ない
  - － 機械が速く動くと負荷が大きくなる
- 論理的に筋が通った推論により、設計を良くするための仮説を立てる。
- 定性的な推論や可視化が困難になるため、高次元パラメータ空間における探索は苦手である。

#### ■探索/最適化アルゴリズム

- 現象の背後にある物理法則に注目せず入力と出力の間の関係のみに注目。
- 繰り返し作業を苦にしない。
- 探索時のデータ転送コストが人間が探索する場合に比べて小さい。
- 定量的に良いと思われるパラメータを見つけ出す。

- 高次元パラメータ空間における探索を低次元パラメータ空間の場合と同様に行うことができる。
- 現在の技術では、人間が行う定性的推論ができない。

加えて、省コストが要求される現場でこれらの代替を検討する際には、代替行為にもコストがかかるということに留意する必要がある。問題の難易度と技術レベルを見誤り無理のある代替をしてしまうと、かえって現場が混乱し、損をするということもありうる。一方で、代替行為に挑戦することで、携わった人間が技術的素養や、プロジェクト運営能力を身につけられたりという副次的な効果も存在しうる。設計に対するメタな知識を蓄積し、他所に見えない部分で競争力をつけられるという意味では、損とは言い切れない。

## 1.2 研究目的

### 1.2.1 研究のターゲット

以上のような議論から、数値解析手法を活用した機械システムの制御や設計のパラメータを最適化の取り組みが増えていくと考えられる。機械の寸法や部品の組み合わせ方のような機械の設計パラメータを最適化するためには、機械の動き方のみならず、機械の製造法や部品の入手性のような多岐にわたる知識を利用する必要がある。問題の範囲が広い。それに対して、制御パラメータは設計パラメータは確定のものとして、その機械の動かし方を決めるものであり、問題の範囲が限られており、議論が行いやすい。そこで本研究では研究のターゲットを「数値解析手法を活用した最適化により、現実存在する機械の制御パラメータを最適化する問題」と設定した。そのターゲットとなる問題は、人と計算機が相補的に役割を果たすことによって効率的な解決がされ则认为している。現代の機械システムの最適化において、人は、モデルやその動作環境を構築したり、設計の要求を明確化するという役割を担っており、それらの役割を計算機が担うことはできていない。また、計算機は、モデルが与えられれば物理現象を模擬することができ、目的関数が与えられれば最適化を人より効率良く行うことができる。本研究では、人と計算機が相補的に役割を果たし効率的に問題を解決していくための仕組みを設計し、提案する。

### 1.2.2 研究目的

本研究では以下のような研究目的を設定した。これらの研究目的は本論文における貢献と対応している。

1. 機械システムを最適化する既存手法・既存研究の整理
2. 最適化システムを構築・改善していく、メタな最適化システムの提案

### 3. 最適化システムを構築・改善するための手法を提案

■1. 機械システムを最適化する既存手法・既存研究の整理 これまでも、自動車、航空機、工作機械などの機械システムの設計を最適化する取り組みは行われてきた。本研究ではまずこれらの取り組みの調査を行い、取り組みの中で利用された手法や発生する課題について整理する。この目的は本論文の、1 章、2 章の内容と対応する。

■2. 最適化システムを構築・改善していく、メタな最適化システムの提案 計算機を利用して算出した最適パラメータを実際の機械システムにおいて有効に働くようにするためには、実際の機械システムとモデルとの間の誤差が小さくする必要がある。また、最適パラメータを実際の設計現場において利用することを考えると、最適パラメータの算出にかけられる時間は限られる。つまり、誤差の小さいモデルを短い時間の間で開発したうえで、パラメータの最適化を行っていくことが求められる。本研究では、モデルのような最適化システムの構成要素を効率的に開発・修正する方法に着目している。構成要素を効率的に開発・修正する行為は、「最適化システム」を最適化しているととらえることができ、本研究はこれらの行為と行為者をまとめて「メタな最適化システム」と呼ぶ。2 つ目の目的は、メタな最適化システムを定式化することである。この目的は本論文の、3 章の内容に対応する。

また、この「メタな最適化システム」が論文の主要なオリジナリティであると考えている。本研究以前にも、最適化システムを構築して、その要素を修正していくという行為は技術の現場において行われてきた。しかし、一段階メタな考え方に立って、最適化システムをどのように構築し修正していけばよいかという方法論を体系的にまとめた研究は、研究は筆者の知る限りでは存在しない。その背景としては、以下の 3 つの理由があると考えている。

1. 最適化システムの要素開発することや、最適化システムを動作させて最適パラメータを算出するだけでも十分に有用であり、成果となった
2. 最適化システムの要素開発方法や修正方法は、現場や人の持っている知識に影響されるため、一般化な方法論を取り出すことが困難
3. システムの統合という応用寄りのテーマは定量化や評価が難しい

機械設計のアナロジーを使って本提案を説明する。実際の設計 [4] のような設計の教科書においては、機械の構成要素や、構成要素の組み合わせ方だけではなく、設計するとき設計者が考えるべきことや、行うべき行動が時系列で整理されている。本提案のメタな最適化システムもこのような時系列の整理を行った結果として得られたものである。機械設計のアナロジーを利用すると、本研究は、最適化システムの設計図を提供する研究では

なく、最適化システムを組むための試行錯誤や最適化システムの製造方法についての情報を提供する研究であると考えることができる。

■3. 最適化システムを構築・改善するための手法を提案 本研究では、内部で数値解析モデルを利用しているメタな最適化システムに限定し、議論を行う。2 つ目の目的はメタな最適化システムに要求される機能や、メタな最適化システムにおいて利用できる要素の定式化であった。それに対し、3 つ目の目的は、メタな最適化システムを効率良く動作させるための、新しい機構の提案を行うことである。いくつかの機構を提案し、具体的な実機械システムの最適化問題において、機構の検証を行う。そして、その機構を用いて最適化システムの構築・改善及びパラメータの最適化を行う。この目的は本論文の、3 章、4 章、5 章の内容に対応する。3 章は機構の提案、4 章、5 章は具体的な実機械システムの最適化問題に機構を適用した際の検証内容である。

目的 3 の具体的な実機械システムの最適化問題においては、以下の 2 つの副目的を設定した。副目的は 4 章、5 章の内容を、それぞれ独立した最適化の研究として見たときの研究目的となっている。4 章、5 章の副目的は各章の研究活動の貢献に対応している。

- 具体的な問題における、研究課題を達成するための手法を提案し実行する。
- 研究課題を達成したうえで、実際の機械システムのパラメータ最適化まで行い、各問題の最適化が現代の技術の組み合わせで解決が可能か否かについて、デモンストレーションを行うことにより検証する。

### 1.3 本論文の構成

図 1.14 に本研究の構造を示す．本論文は 6 章構成となっている．図中の矢印は章間の内容のつながりを表す．

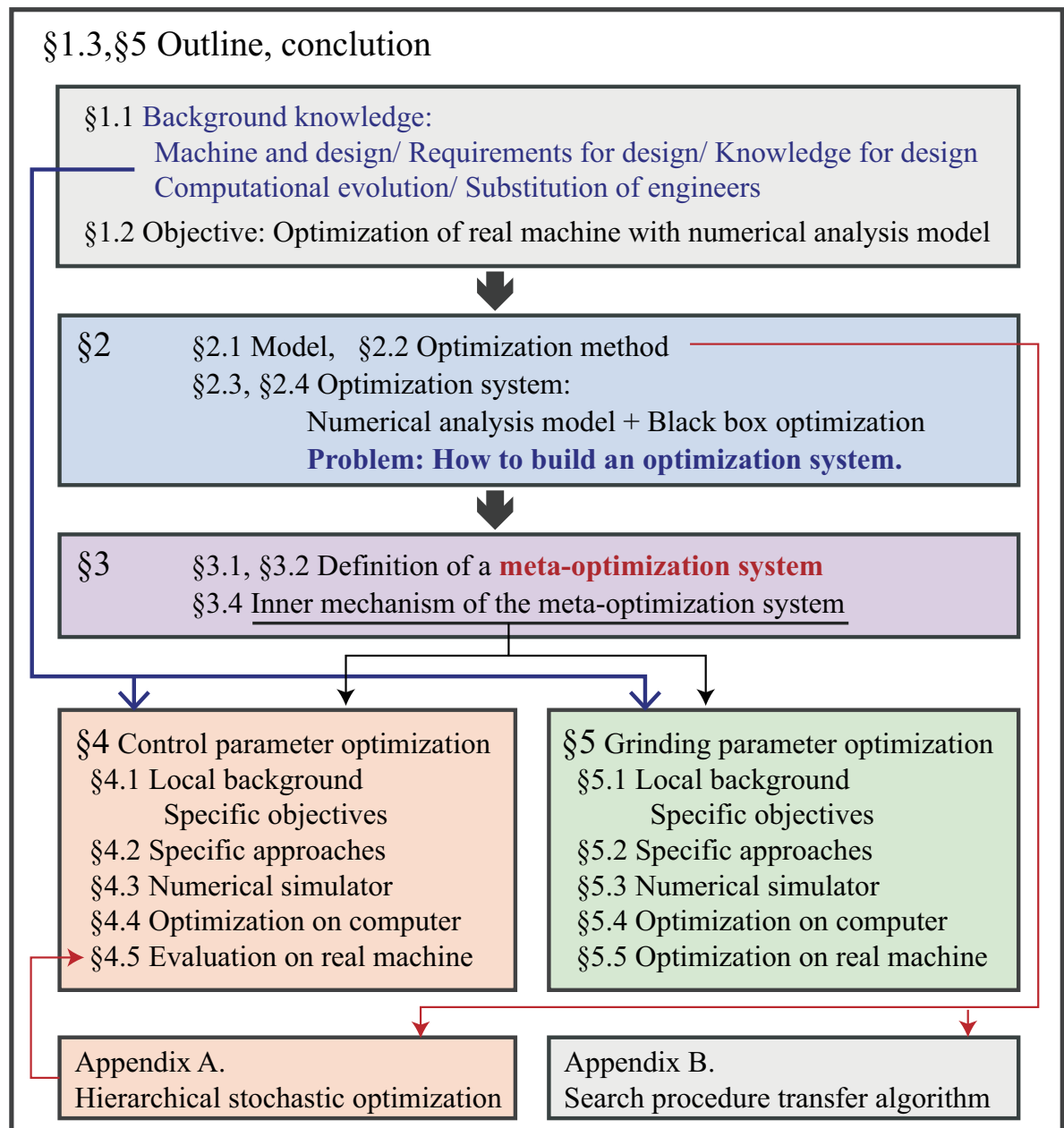


図 1.14 本論文の構成.

## 第 2 章

# 数値解析を活用した実機械システムの最適化

本研究以前にも，自動車，航空機，工作機械などの機械システムの設計を最適化する取り組みは行われてきた．本章では，既存の最適化の取り組みについて整理する．整理の観点は，最適化の仕組みである最適化システムの構成要素，既存研究，最適化システムの利点・課題である．まず，2.1 節と 2.2 節において，最適化システムにおいて現象を模擬するためのモデル，パラメータを最適化するための最適化手法について述べる．また，2.3 においては，機械システムの最適化に関する既存研究のターゲットやそれらの研究が利用している手法について整理する．そして，2.4 節においては，最適化システムの利点や技術的課題について述べる．

### 2.1 現象のモデル化形態の比較

物理現象を表すモデルには 3 つの形態が考えられる．本節ではこれらのモデル化の形態の概要，得手不得手，利用のされ方，相互関係について述べる．

#### 2.1.1 モデル化の 3 形態の定義

1.1.1 項と同じく「モデル」とは物理現象に関する知識のことを指す．モデルの表現形式を式 2.1 の形で定義する．

$$\boldsymbol{q} = m_{\boldsymbol{\theta}}(\boldsymbol{e}) \quad (2.1)$$

ここで  $\boldsymbol{e}$  は環境の状況， $\boldsymbol{q}$  を物理量とする．また，今後の議論でモデルに入力される設計・制御パラメータとして， $\boldsymbol{x}$  が出てくることがあるが，環境の状況  $\boldsymbol{e}$  を広義に捉えると  $\boldsymbol{x}$  も含まれるため，一部では区別をせずに  $\boldsymbol{e}$  のみを利用することもある． $\boldsymbol{\theta}$  はモデルの構成を表すパラメータ (変数) である．この  $\boldsymbol{\theta}$  がモデルの精度や内部で行われる計算を決定



する． $\theta$  については本章後半の 2.4 節において詳細を述べる．

このように，式 2.1 は，環境の状況をもとに現象が発生し物理量が出力される様子を表している．図 2.1 はモデルの入出力関係を表す概略図である．

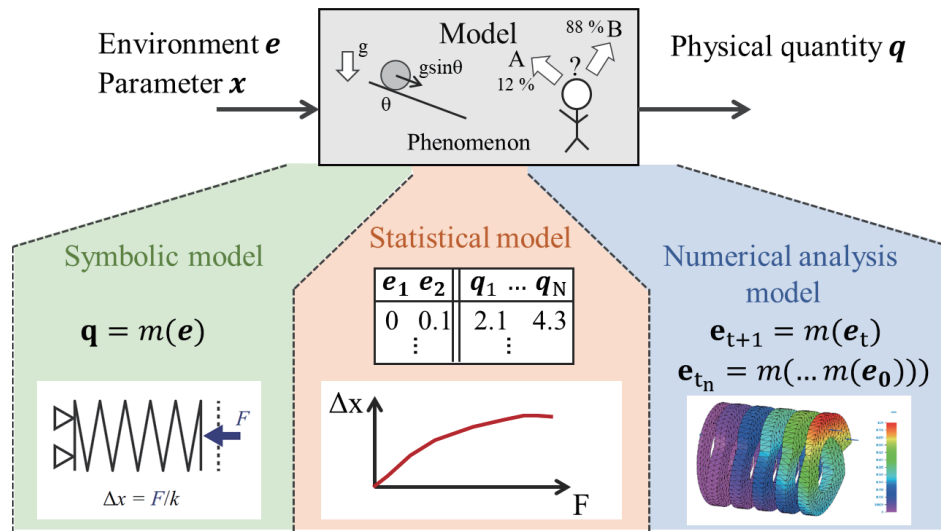


図 2.1 モデルの定義とモデルの 3 形態．

式 2.1 のモデルの定義に対して，実機の表現形式を式 2.2 の形で定義する．

$$q = M(e) \quad (2.2)$$

ここで， $M$  は実機を表す．

モデル化の形態としては以下の 3 つが存在する．

1. 記号モデル：別の記号モデルを組み合わせたり，妥当な仮定を施すことによって，方程式の形で，環境の状況  $e$  から現象の結果の物理量  $q$  を計算可能にするもの
2. 統計モデル：実験データをうまく説明するように多自由度の関数の係数を決め，環境の状況  $e$  から現象の結果の物理量  $q$  を計算可能にするもの
3. 数値解析モデル：粒子・格子・要素同士のミクロな関係を記述した式を連続的に適用しマクロな現象を再現し，環境の状況  $e$  から現象の結果の物理量  $q$  を計算可能にするもの

図 2.1 下段は基本的な機械要素であるバネを各形態のモデルで表現したときの様子である．

この分類はお互いに重複している部分はあるものの，現象を模擬するためのモデル化の営みを 3 分していると考えている．1.1.1 項において紹介した，あるモデルが他のモデルから導かれる例のように，現象自体や現象のモデルには階層性がある．そのため，モデルを外部から見たときの形態と，そのモデルの内部で利用されているモデルが別の形態にな

ることも考えられる．本論文では現象を模擬するためのモデル形態について語るとき，一番外側のモデルの形態のことを指す．

現代科学におけるモデルやモデル化行為が議論されている文献 [47] においても，モデルは3つに分類されている．文献 [47] においては現代科学においてモデルが果たす役割を説明するために，モデルを具象モデル，数理モデル，数値計算モデルという3つのモデルに分類している．それに対し，本研究では計算機により扱うことができ，機械設計に用いられるようなモデルを扱う．文献 [47] と本論文とでは対象としているモデルの範囲が異なるため，分類方法も異なっている．本研究における数値解析モデルは文献 [47] における数値計算モデルに対応する．また，本研究における記号モデル，統計モデルは文献 [47] における数理モデルに対応する．

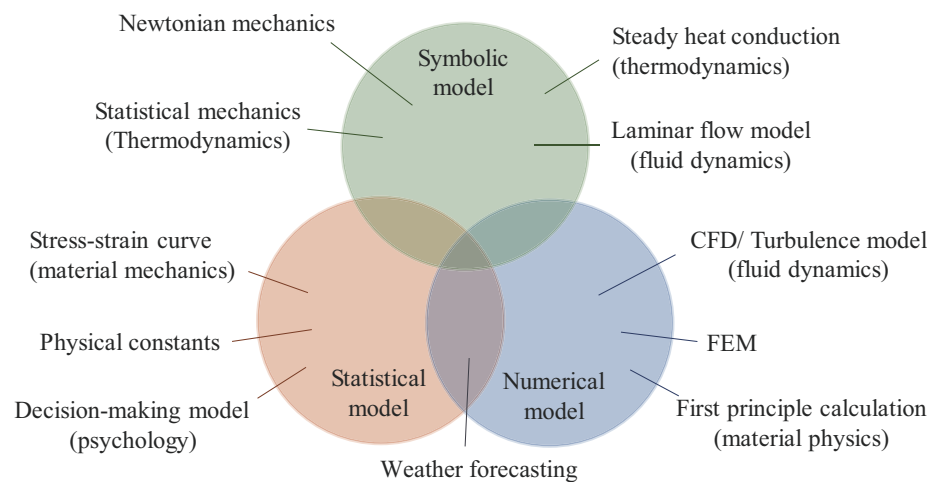


図 2.2 3つのモデル化形態とそれぞれの具体例．

図 2.2 は具体的な模擬の対象がどの形態のモデルに属するかを示している．図 2.2 は形態間で綺麗に境界が引けるわけではなく，ある対象が3形態全てで別個に模擬されることもあれば，2分類の中間的な形態で模擬されることもあることを示している．

様々な分野で異なる形態のモデル化がされていることから，3形態には明確な優劣は存在せず，3形態は特性に応じて使い分けられていると考えられる．次項ではこの特性について述べていく．

### 2.1.2 3形態のモデル化の特性

本項では設計の文脈において，各モデル化形態の持つ特性や得手不得手について述べる．モデルの構築可能性，複雑性・表現能力，再利用性，計算負荷について順に述べる．表 2.1 は以降の議論の概要を表で表したものである．表 2.1 内の丸とバツはある形態が他の形態と比べて優れた特性を持っているか否かを表している．

表 2.1 3 形態のモデル化の特性比較表.

Model type	Buildability	Expression	Transfer-ability	Computation cost
Symblic	×	×	○	○
Statistical	○	×	×	○
Numerical analysis	○	○	○	×↗

### モデルの構築可能性

モデルの構築可能性とは，現象の一部分を切り取り各形態のモデルに落とし込むことがどの程度可能かを表す．構築可能性では，模擬時の誤差の大小よりはむしろ，モデルを構築するため人間が行う必要のある手順の自明さや困難さを評価する．

設計対象が複雑化するにつれて，記号モデルの構築の可能性は下がってきた．これは現象の相互作用により全体としての現象を単一の式で表すことが難しくなったこと，設計物の物理的形狀が複雑になり数式化が難しくなったことに起因すると考えられる．

前者と関係する話題として，天体の運動における三体問題のように，支配方程式が単純でもそれらがお互いに影響しあうことによって，挙動が複雑になることがあるというものがある．三体問題における天体の挙動は複雑であり，一般解は求められていない．このように単純な支配方程式が組み合わさることにより，全体としての挙動を定式化するのが難しくなることがある．

後者の物理的形狀の複雑化について例を挙げて説明する．熱伝達 は機械設計において重要な要素であり，熱伝達経路をうまく設計することにより，熱源の過熱を防いだり，熱を効率的に交換することができる．この熱伝達を記号モデルに落とし込むときには，伝達経路の形狀を式として表す必要があり，複雑な 3 次元の形狀の模擬が難しい．他にも，ジューコフスキー翼は 2 次元の翼型を式の形で表現するものであり，完全流体の場合の翼回りの定常的な速度場を記号モデルを用いることで求めることができる [48]．しかし，現実に存在するような 3 次元翼回りの速度場や，粘性の有る流体の速度場を記号モデルにすることはできない．3 次元の速度場を計算するためにはパネル法などの数値解析的な手法を利用する必要がある [49]．

一方で，統計モデルはモデル構築のためにすべき手順は実験データをとることであり，自明である．統計モデルは計測された実験データをもとに，環境の状況  $e$  から現象の結果の物理量  $q$  を計算する関数の係数を定める．手順は自明であるものの，実験データを

集めるためには実験を行う必要があり、実験コストの大きさによってはモデルの構築が難しくなる。1.1.2 項で述べた HondaJet のような複雑な形状の設計を行う際には実機実験を行うのは困難である。

数値解析モデルは発生する現象を支配する方程式を求めることができれば、モデルの構築は容易である。支配方程式を求めるという取り組みは古くから行われており、支配方程式自体が既に存在することもあるし、存在しなかった場合も蓄積されている支配方程式を求めるための行為に対する知識 (Analysis 行為をするためのメタな知識) を利用することができる。支配方程式が複数ある場合もお互いを満たしながら計算を進めることにより、現象全体としての模擬が可能である。

### モデル (出力) の複雑性・表現能力

モデルの複雑性・表現能力とは、モデルがどの程度複雑な現象を模擬することができるかを表す。ここでの複雑さとは、モデルの入力を変化させた際の変化具合のことを表しており、複雑さが高いほど出力の変化の非線形性が大きいことをあらわす。

先述の三体問題の話題に重なる部分でもあるが、現象の方程式に非線形性があると、方程式の表記が簡単でも、現象の結果が複雑になるということがある。複雑系科学の分野で有名なロジスティック写像 (図 2.3) や多重振り子においては初期値が変わるだけで大きく挙動が変わる。数値解析モデルと相性が良く、単純な方程式を連続的に適用することによって現象を表現することができる。

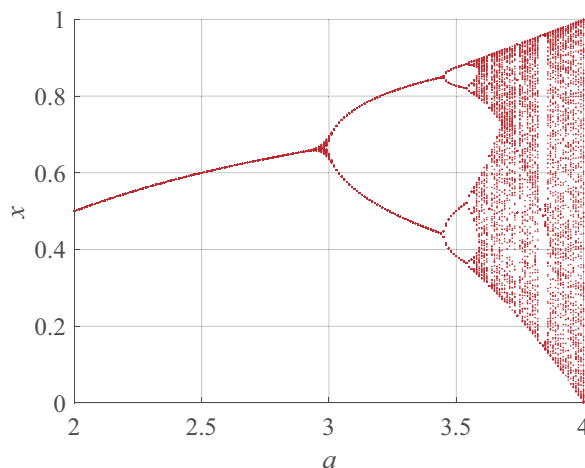


図 2.3 ロジスティック写像における初期値  $a$  の変化による挙動の変化。

一方で、統計モデルのアプローチでよく利用される応答曲面法 (RSM; Response Surface Methodology) は、複雑な関数の近似を苦手とする [50]。統計モデルの中でもニューラルネットワークは特定の条件を満たすと、万能の表現能力を示し複雑な曲面の近似も可能になる (universal approximation theorem) といわれているが、複雑な関数を近似しようとする

ると近似のための関数の自由度を高くする必要があり、自由度が高いほど係数を正しく推定するために多くのデータが必要になる。

### モデルの再利用性

モデルの再利用性とは、ある現象のモデルが、別の現象のモデルを構築するときに役に立つかを表す。

数値解析モデルは支配方程式をベースとしているため、小さい規模で確認した妥当性が大きな規模でも通用すると考えることができる。これは、構成要素の量が変わっても背後の支配方程式は変わらないためである。例えば、1D-CAEのように機械要素の機能をベースとした数値解析モデルであれば、モデルの妥当性を確かめるためにまず単純なバネマス系で数値解析モデルと実験結果が合うかを確認して、その後実験が難しいような対象に対して数値解析モデルを構築するという行為は妥当である。3D-CAEであれば、例えば、研削加工において、綿密な計測が容易な一つの砥粒がある系において実験を行い、その結果をもとに一つの砥粒の数値解析モデルを構築、そして計測が難しい多くの砥粒がある系の数値解析モデルを構築するという行為も妥当である。多くの砥粒がある系に移行して、機械の振動が問題になったり、発生した熱が問題になることがあるかもしれないが、その場合は物理現象を記述する方程式を更新すればよい。記号モデルの再利用可能性は、一度妥当な記号モデルを構築できれば、あらゆるところでモデルが使えるという事実ともつながる。ニュートンの運動方程式  $F = ma$  は非常に多くの場面において成り立つ妥当性があるといわれている法則である。例えば、図のように誰かがボールを発射する装置を開発していたとして、その装置を実際に製作し実験を行うことなく、事前に射角や速度とボールの飛距離の関係を予測することができる。このように、記号モデルには、少ない実験量や少ない記述量で、多くの場合に転用できるようなモデルを構築できるという性質がある。背後には物理法則の種類が限られている、物理法則は多くの場面で成立するという性質がある。私たちが記号モデルを転用する際には、このような物理法則の性質を暗黙のうちに利用している。

また、数値解析モデルは考慮すべき物理現象が変わらない限り転用できる。1D-CAEにおいて機構の構成を変更した別機種の挙動を模擬したい、3D-CAEにおいて解析対象の形状を少し変えて挙動を模擬したいという場合にも、支配的な方程式が変わらなければ、数値解析モデルの計算の仕組みを変更せずに済む。加えて、数値解析モデルは入力するパラメータの種類を変えても、今まで出力しようとしていなかった物理量を取り出すようにしても、支配的な方程式が変わらなければ、数値解析モデルの計算の仕組みを変更せずに済む。

一方で、統計モデルは入力と出力の形が固定されているため、入力や出力の形を変えた場合、実験データを取り直して、統計モデルを再構築する必要がある。

### 計算負荷

数値解析モデルは上記のように良い性質を持つが、方程式を連続的に適用するという性質上、他の 2 つのモデルに比べ、計算コストがかかってしまう。文献 [51] は遠心送風機の形状最適化の研究であるが、数値解析モデルによる現象の模擬は計算コストがかかるという主張のもと、統計モデルにより数値解析モデルを近似して最適化全体の計算コストを小さくする最適化システムを提案している。このように数値解析モデルは他の 2 つの形態のモデルに比べ計算コストがかかる。

また、数値解析モデルの現象模擬性能や精度は計算機性能に強く影響される。そのため、計算機性能の指数関数的上昇が起きている昨今においては、記号モデルや統計モデルに比べ、数値解析モデルの能力の向上速度が大きいと考えることができる。現時点で、数値解析で解けるなら解きたい問題があった場合、数年先には計算機性能の向上によって解けるようになっていくということが起こりうる。また同様に、数年前に数値解析で解けなかった問題が、現時点では解けるようになっていくということが起こりうる。

### 2.1.3 3 形態のモデルの関係性

ここまでは 3 形態のモデルが独立なものとして述べてきたが、お互いに関係している部分もある。例えば、観測データに基づいて統計的に仮説を立て、普遍的な法則をあらわす記号モデルを導くことがある。記号モデルの構築において 0 から 1 にする作業は統計モデルの組み立て方と同じであることがある。例えば、ケプラーの第一法則は記号モデルであるが、天文学者の蓄積してきた天体の運行に関するデータをケプラーが観察し、惑星が楕円軌道をするを仮定するとデータをうまく説明できることを発見し導かれた [52]。

また、数値解析モデルの粒子・格子・要素同士の関係性を記述する式は、記号モデルであることが多く、統計モデルであることもある。例えば、単相流体の流体解析において流体の運動を記述する方程式は記号モデルのナビエ・ストークス方程式であり、全体の挙動はナビエ・ストークス方程式を満たすように数値的に計算される [46]。別の例として、人間の交通流の数値解析において、各個人の判断を記号モデルにすることは難しいため、統計モデルにより表現するということもある。

他にも以下のようなモデル形態をまたいでいる事例が存在する。

- 現象に関する統計モデルから現象に関する記号モデルを抽出することがある。
- 統計的仮説を置きながら記号モデルを構築する統計力学のような分野も存在する。
- 現実世界での実験が不可能な対象において、数値解析モデルでの計算を行った結果を記号モデルに落とし込むこともある。

- 設計においては、初期段階において物理量のオーダーを推定するために記号モデルを利用し、その後の細部の設計のため数値解析モデルによる計算を利用することがある。

それぞれのモデル形態の得手不得手を理解し、必要に応じて組み合わせることで、実験コストや計算量を抑えながら高い現象の再現能力を実現できる。

本研究では、ここまで述べてきたような数値解析モデルの良い特性及び、能力の向上速度を鑑みて、数値解析モデルを用いて現象を模擬し、実機械システムの最適化に用いる。

## 2.2 複雑な目的関数のための最適化手法

本研究では実機械システムのパラメータの最適化をターゲットとした。本節においては世の中にある最適化手法の分類について述べ、実機械システムの最適化に適した最適化手法について述べる。

### 2.2.1 最適化の定義

最適化手法の分類に入る前に本論文における最適化の定義を述べる。最適化とは「 $d$ 次元ベクトル  $\mathbf{x}$  を入力とし、スカラー値  $y$  を出力する関数  $y = f(\mathbf{x})$  があつたとき、 $y$  を最大化するように  $\mathbf{x}$  の値を調整すること。」と定義する。研究分野や研究論文によっては  $y$  が最小化するように調整していたり、 $\mathbf{x}$  の要素数が可変となっていたりするが、どちらも簡単な換算により本論文における定義に帰着させることが可能である。

この定義における関数  $f$  を「目的関数」、 $y$  を「目的関数の値」、 $\mathbf{x}$  を「パラメータ」と呼称する。「パラメータ」は他の調整用の数値と区別するため「制御パラメータ」と呼ぶこともある。他の調整用の数値は「ハイパーパラメータ」と呼ぶ。また、パラメータ  $\mathbf{x}$  を用意して、目的関数  $f$  の出力  $y$  を計算することを「サンプリング」と呼び、 $(\mathbf{x}, y)$  の組を「サンプル」と呼ぶ。「パラメータの要素」といった場合にはベクトル  $\mathbf{x}$  の各要素  $x_i$  のことをあらわす。パラメータの要素  $x_i$  の定義域のことをパラメータの定義域と呼ぶ。最適化を行うことにより大きな  $y$  を出力させられるようになった  $\mathbf{x}$  を「解」と呼ぶ。 $d$ 次元パラメータ  $\mathbf{x}$  が張る  $d$ 次元空間のことを「探索空間」、探索空間に対して関数  $f$  の出力  $y$  を結合してもう一次元増やした  $d+1$ 次元空間における超曲面の形状を「関数形状」と呼ぶ。図2.4は  $d=2$ の時の関数形状の描画例である。 $d=2$ のとき、 $\mathbf{x}$ は2要素のベクトルで、関数形状は3次元空間上の曲面の形状のことを指す。

本論文の文脈において目的関数  $f$  は機械システムの性能を評価するものである。パラメータ  $\mathbf{x}$  は機械の構成を大きく変えずに変更できる設定値である。また、本論文内では目的関数の表記として  $f_\phi(\mathbf{q})$  も利用する。この表記において  $f$  の添字となっている  $\phi$  は



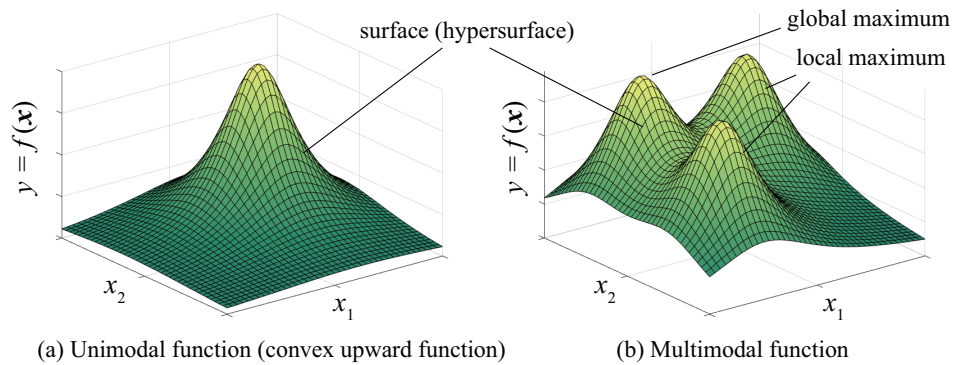


図 2.4 関数形状の例と関連する用語.

目的関数の構成を表すパラメータ (変数) であり, 入力値された値をどのように処理するかを決定する. この添字を利用した表記では, 目的関数の引数が物理量  $\mathbf{q}$  である. 本論文においては  $\mathbf{q}$  は  $\mathbf{m}_\theta$  や  $M$  の出力であり,  $\mathbf{x}$  の関数であるため,  $f_\phi$  も  $\mathbf{x}$  の関数となる. この表記は, 主に目的関数とモデルを分離して議論する際に利用する. 添字の無い  $f$  を使う表記では, 引数は  $\mathbf{x}$  である.

最適化手法に対しては, 少ないサンプル数で全体最適に近い解を見つけることが要求される. 「全体最適解」とは  $\mathbf{x}$  の定義域の中で最も  $y$  を大きくする  $\mathbf{x}$  のことである. 全体最適もしくは全体最適ではないが十分に  $y$  の値が大きい解のことを「実用的な解」と呼ぶ. また, 全体最適解ではないものの関数の極大値である解のことを「局所最適解」と呼ぶ. そして, 図 2.4(b) のように目的関数が局所最適解を複数持っている様子のことを「多峰性 (multimodality)」と呼ぶ. 一方で図 2.4(a) のように目的関数が局所最適解を持っていない様子のことを「単峰性 (unimodality)」と呼ぶ. 単峰性を持つ関数は凸関数 (本論文の定義の場合は上に凸な関数) とも呼ばれる.

### 2.2.2 最適化手法の分類

「少ないサンプル数で全体最適に近い解を見つける」という最適化手法への要求を満たすために, いくつかのアプローチが存在する. 本項においてはアプローチごとに手法进行分类し, 本研究のような機械システムの最適化に適した手法について述べる.

最適化手法に対しては, 「ノーフリーランチ定理 [53]」という, 万能な手法は存在しないという定理が成り立つと言われている. 「ノーフリーランチ定理」によると, ある最適化問題に対して, 他の手法に比べて良い性能を示す手法は, その問題における偏りを利用している. そして, 偏り方が違う問題に対しては良い性能を発揮できないという趣旨の主張がされている. 言い換えると, 最適化手法は目的関数の性質をうまく利用することで他の手法より良い性能を発揮しようとしている. 最適化手法の開発者は世の中の問題の偏り



を仮定して、その仮定に基づき最適化手法を作る。この仮定の置き方がうまいと、最適化手法が幅広い問題で通用し、より良い解を見つけられるようになる。最適化手法に組み込まれている目的関数の性質に関する仮定や事前知識は「ヒューリスティックス」と呼ばれる。

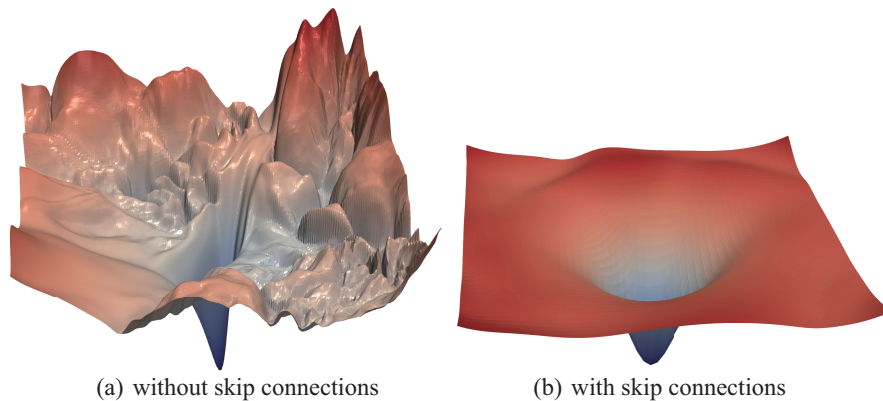


図 2.5 Skip-connection の持つ誤差関数の多峰性の低減効果 [54].

偏りの利用方法の 3 つのアプローチとそれぞれに属する手法を概説する。まず、1 つめのアプローチは最適化される目的関数の形状を工夫する手法である。例えば、凸化は目的関数が下に凸な関数であることを何らかの方法によって証明することで、局所最適解の存在を否定し、関数の勾配の情報にしたがって改善し続けることにより全体最適解にたどり着けるようにする。また、信頼領域法は凸化と似ているが、目的関数を局所領域 (信頼領域) 内で二次関数と近似することにより、最適解を高速に求める手法である。統計モデルのための近似器として使われるニューラルネットワークは、データと近似関数の誤差を減らすために誤差関数の係数を調整する必要がある。図 2.5(b) に示すネットワークの構造 (関数の内部構造) に工夫をした Resnet というニューラルネットワークの誤差関数 (目的関数) は、図 2.5(a) の工夫のないネットワークに比べて誤差関数の多峰性が小さく [54]、局所最適解にとどまってしまうことが少ない。

2 つめのアプローチは関数形状に関して得られている情報を明確に利用する手法である。これらの手法は機械学習分野における転移学習 [55] やメタ学習の一部分の手法を指す。転移学習とは解きたい問題の類似の問題で得られた知識を転用する手法全般のことを指す。また、メタ学習は学習する過程を学習するという手法全般のことを指し、転移学習と重なる部分もある。最適化においては、最適化をしたい問題を解く前に類似の問題を解いて関数形状に関する知識をためるアルゴリズムや、類似の問題を解いて最適化のサンプリングの取得方法を最適化するというアルゴリズムが転移学習やメタ学習に当てはまる。著者らも最適化したい対象を最適化する前に、最適化のサンプリング手順を別の問題で洗練させて、最適化したい対象の最適化に用いる SPT アルゴリズムを提案している [56]。

しかし、転移学習やメタ学習の手法を利用するためには、事前に類似の問題を最適化する必要があったり、問題の類似性を人間が判断して用意する必要があるなどの条件があり、研究課題として取り組んだ際、議論が複雑になってしまう。転移学習、メタ学習の詳しい内容と SPT アルゴリズムの詳細については本論文の付録 B において述べる。

3 つめのアプローチは目的関数に工夫を施さず、関数形状に関して得られている情報を明確には利用しない手法である。1 つめと 2 つめは両方とも強力なアプローチであるが、必要な条件や前もって得ておくべき情報があり、利用できる場面が限定される。本研究は数値解析モデルを活用した実機械システム最適化の可否を検証する研究であり、最適化の定量的性能の検証は最優先事項ではないため、問題の満たすべき条件や問題に関する必要な事前情報が少ない 3 つめのアプローチを取り入れる。

3 つめのアプローチは目的関数の入出力以外の情報を利用するか否かでさらに 2 つに分類できる。分類により、ホワイトボックス最適化 (WBO) とブラックボックス最適化 (BBO) の 2 種類に大別される。WBO は目的関数の導関数などを利用することにより高い最適化性能を発揮する。しかし、機械における非線形な熱の伝達やスイッチングのような離散的な動作や、現象同士の相互作用は記号モデルにすることが難しく、導関数を求めることも難しい。よって WBO は本研究がターゲットとするような制御パラメータ最適化での利用は限られる。一方で BBO [57] は目的関数の入出力のみを使った手法であり、目的関数の数式が無くても利用可能である。BBO では、数値解析モデル側で出力可能でさえあれば、複数の物理量や機能要求を同時に考慮可能であるため、機械設計における制御パラメータ最適化に適している。本研究では数値解析モデルを用いた最適化のために、目的関数に関する強い仮定や数式化が必要ないブラックボックス最適化手法 (BBO) を利用する。

ブラックボックス最適化には、いくつかのよく知られたアプローチがある。クロスエントロピー法 (CEM; Cross-entropy Method) [58] は、2 つのステップで構成される単純な確率的最適化手法である。クロスエントロピー法では最初に、サンプリングのための確率分布からパラメータが出力され、目的関数により各パラメータの目的関数値が計算される。その後、確率分布は高い目的関数値を持つエリートサンプルのみを利用して再計算され、次のサンプリングに使用される。確率分布には正規分布が用いられ、エリートサンプルの平均値と共分散行列を使用して定義される。クロスエントロピー法では、図 2.6 に示すように、確率分布に従ったサンプリングと確率分布の再計算が繰り返し行われ、目的関数の値が大きくなるパラメータを多く出力するような確率分布の獲得を目指す。CMA-ES (Covariance Matrix Adaptation Evolution Strategy)[59] は、クロスエントロピー法よりも高度な方法により次の確率分布を計算する。CMA-ES はクロスエントロピー法では存在しなかったエリートのサンプルに付けられた重みづけや、探索と活用バランスを決定する係数の自動調整の機能を持つ。

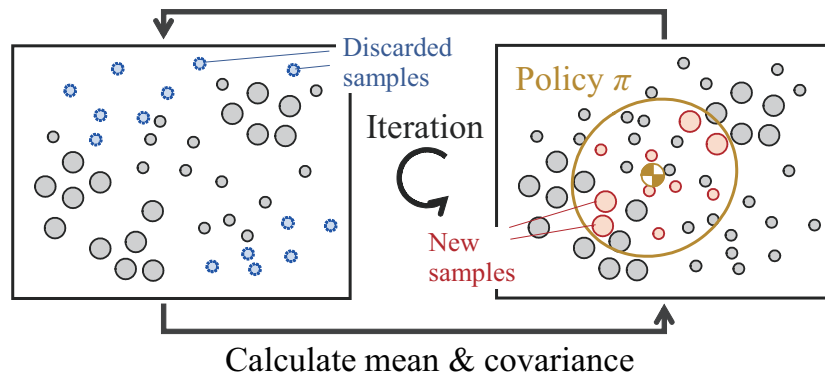


図 2.6 クロスエントロピー法による最適化.

### 2.2.3 目的関数とモデルとの違い

モデルという言葉は関数一般やシステム構成を指すこともある多義的な言葉である．例えば，文献によっては最適化対象の目的関数がモデルと呼ばれることもある．以降の議論での混乱を防ぐため，本項において本論文内における定義を行う．

本論文においては，物理現象を模擬したものを「モデル」，設計や機械の性能を評価する物を「目的関数」とする．また，本論文においては，モデルは数値ベクトルを入力として数値ベクトルを出力する関数であり，目的関数は数値ベクトルを入力としてスカラ値を出力する関数である．具体例を元に説明すると，車体の設計の例では，剛性，重さ，製作時間などの物理量が算出され，設計を評価するための目的関数の算出に用いられる．この場合，モデルは，車体の設計パラメータを入力として受け取り，物理量のベクトルを出力する．目的関数は，物理量の中で評価に用いられるものを入力として，スカラの評価値を出力する．

## 2.3 実機械システムの最適化の関連研究

本節では機械工学の分野における最適化研究の大まかな動向について述べる．文献 [60][61][62][63][64][65][66][67][68][69][70][50][71][72][73][51][74] は機械工学の分野における最適化の研究である．表 2.2 はこれらの論文を最適化対象，年代，モデル化の形態，最適化手法で整理したものである．表 2.2 において各研究は，最適化対象の属する分野順，分野の中においては年代順に並んでいる．

表 2.2 機械工学の分野における最適化を行った研究.

Author	Field	Target	Year	Optimizer type	Optimization algorithm	Model type	Numer of parameters	Evaluation in real	Journal
Y. Wakasa	General	Control	2013	BBO	CMA-ES	Numerical	?	×	SICE
T. Kobayashi, et al.	Aerospace	Structure	2005	BBO	MC	Num. + Stat.	3	×	JSME
M. Nakane, et al.	Aerospace	Shape	2006	BBO	GA	Numerical	?	×	JSASS
H. Morino, et al.	Aerospace	Shape/Structure	2005	BBO	GA	Num. + Stat.	109	×	MHI Technical Review
Y. Mizukami, et al.	Machine tools	Control	1995	BBO	GA	Num./ Sym.	13 (discrete)	×	SICE
M. Hao, et al.	Machine tools	Control	1997	BBO	GA	Num./ Sym.	1400 (discrete)	×	JSME
H. Miyagawa, et al.	Machine tools	Control	2001	BBO	GA	Numerical	10	○	JSME
Y. Nakagawa	Machine tools	Control	2006	WBO	Quasi Newton	Symbolic	4	×	JSPE
K. Yamazaki, et al.	Machine tools	Structure	2010	WBO	SQP	Num. + Stat.	3	×	JSME
A. Todoroki, et al.	Material	Structure	2000	BBO	GA	Num. + Stat.	4	×	JSME
T. Yamamoto, et al.	Material	Structure	2009	Topology optimization		Numerical	2520	×	JSME
I. Baba, et al.	Automobile	Control	2013	BBO	SPSA	Numerical	17	×	SICE
A. Oyama, et al.	Automobile	Structure	2019	BBO	GA (Cheetah)	Numerical	222	×	JSME
M. Kitamura, et al.	Marine	Structure	2007	BBO	GA	Num./ Sym.	87		JASNAOE
N. Okamoto, et al.	Marine	Shape	2016	WBO + BBO	SQP + SBO	Numerical	12	○	JASNAOE
K. Sugimura	Fluid machinery	Shape	2006	BBO	SA	Num. + Stat.	27	×	Turbomachinery

CMA-ES: Covariance matrix adaptation evolutionary strategy  
 SQP: Sequential quadratic programming  
 SPSA: Simultaneous perturbation stochastic approximation  
 MC: Monte-carlo simulation  
 GA: Genetic algorithm  
 SICE: The society of instrument and control engineers  
 JSME: The Japan society of mechanical engineers  
 JSASS: The Japan society for aeronautical and space sciences  
 JSPE: The Japan society for precision engineering  
 JASNAOE: The Japan society of naval architects and ocean engineers

表 2.2 の年代とモデル化の形態を見ると年代が新しくなるにつれて、計算機の計算能力を生かして数値解析を利用した最適化が行われるようになっていく。また、機械工学に係る、航空宇宙、工作機械、材料、自動車、船舶、流体機械のように幅広い分野において最適化の研究が行われていることがわかる。

表 2.2 によると多くの研究において BBO を利用して最適化が行われている。これは、機械設計の最適化においては目的関数の導関数などの情報を得ることが難しいためだと考えられる。BBO の中では特に遺伝的アルゴリズム (GA; Genetic Algorithm) が利用されることが多い。

このように、機械システムの最適化の関連研究においては数値解析モデルとブラックボックス最適化を組み合わせるパラメータの最適化が行われている。

文献 [69] においては、自動車向けの吸音材の吸音特性を最適化するためにトポロジ最適化が用いられている。トポロジ最適化では、最適化対象の材質を数多くのセルに分割し、セル内に材料が含まれるか否かを調整する問題を取り扱う。この場合セル内の材料の有無が最適化対象であり、最適化における見かけのパラメータ数は多くなる。トポロジ最適化では、パラメータ間の性質の均質性を利用することで、最適化における実際のパラメータの数を減らす工夫が行われている。トポロジ最適化においては有限要素法を用いた数値解析モデルが用いられるため、目的関数の微分値を求めることができず、BBO を用いて最適化が行われる。もしくはいくつかのサンプルを用いて近似した統計的な式と、WBO とを組み合わせる最適化が行われる。

## 2.4 数値解析モデルを活用した実機械システムの最適化

このように本研究は数値解析モデルとブラックボックス最適化手法を利用して、機械システムの挙動を再現し設計・制御パラメータを最適化する仕組みに注目する。この仕組みのことを「数値解析モデルを活用した最適化システム」と呼称する。本節では数値解析モデルを活用した最適化システムの利点や研究課題について述べる。

図 2.7 は数値解析モデルを活用した最適化システムを表している。図 2.7 において、 $\mathbf{x}$  は最適化対象のパラメータ、 $\mathbf{x}^*$  は最適化されたパラメータ、 $\mathbf{e}$  は機械の運転条件などの機械内部からは変更できないパラメータ、 $y$  は性能の評価値を表す。また、図 2.7 において点線から上側は計算機上での最適化の動作を表している。計算機上での最適化では、数値解析モデルがあるパラメータ  $\mathbf{x}$  を受け取り、物理量  $\mathbf{q}_v$  を出力する。次に、目的関数が物理量  $\mathbf{q}_v$  を受け取り、性能評価値  $y$  を出力する。そして、ブラックボックス最適化手法は  $\mathbf{x}$  と  $y$  との組をサンプリングし、良い  $y$  を出力する  $\mathbf{x}$  の法則性を統計的に求める。システムにおいては、最適化された結果である  $\mathbf{x}^*$  は実機でテストされ、数値解析モデルと同様に物理量  $\mathbf{q}_r$  を出力する。

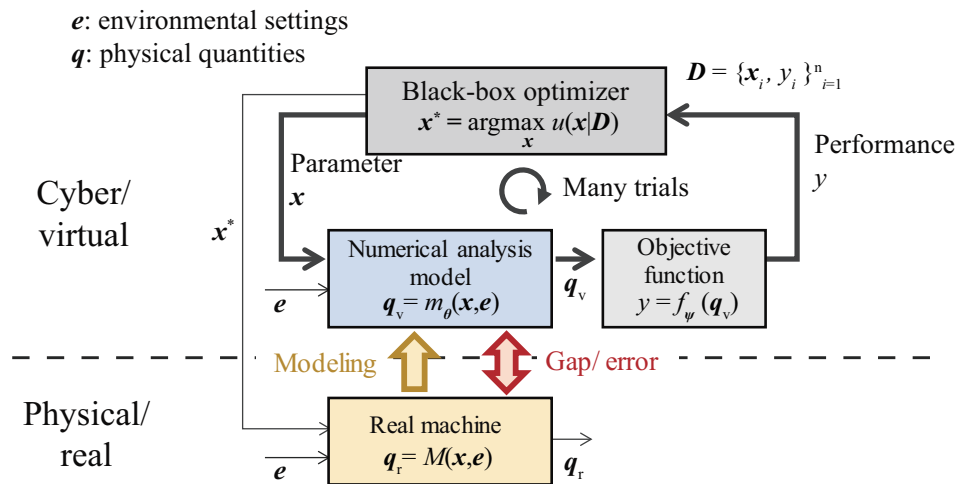


図 2.7 数値解析モデルを活用した最適化システム。

1章で述べたとおり、機械システムの設計や制御パラメータの調整においては複数の性能目標が存在する。目的関数は複数の性能目標を表したものとなる。これら複数の目標は機械のユーザの様々な要求を基にしており、それぞれの優先度やバランスは自明でないことが多い。本研究においては優先度やバランスを設計者が調整しやすくするため、目標の達成度を表す指標  $\phi$  と、指標の重み  $w$  により目的関数を構成するようにしている。ここで、 $\phi$  は関数列であり、 $w$  は数値ベクトルである。指標と重みとを分けることにより、指標を決めてしまえば、重みの調整のみ行えばよくなり、全体的な調整の難易度を低減することができる。

### 2.4.1 数値解析モデルを活用した最適化システムの利点

数値解析モデルを活用した最適化システムには、以下のような4つの利点が存在すると考えられる。

1. モデルに対して最適化することにより、実機実験の際のリスクや、コストを低減させることができる
2. 最適化により、人間に比べ性能の良いパラメータを見つけることができる
3. 最適化を通して、人間がパラメータ設定を一般化するために利用できる手がかりを見つけることができる
4. 中間出力として物理量を取り出せるため、人間の数値解析モデルや目的関数の開発が容易

1つめの利点は、2.1節で述べたようなモデルを利用することによる利点である。1章でも述べた通り、計算機上のモデルにより現象を模擬することで、実機実験にかかるコスト

を低減することができる。ここでのコストには、実機を運転することにかかる設備的・人的コスト、実験準備のコスト、摩耗など実験条件を変えてしまう要素への対処コストがある。最適化アルゴリズムによる最適化では、人手による探索に比べて探索範囲を広くとれる半面、最適化の過程において、機械が故障してしまうような安全ではないパラメータが選ばれることがある。実機で故障が起これば、最適化の進行が止まるだけでなく、修理のための追加人員や材料のコストがかかる。このような実機の実験コストが高くなる問題や故障の問題は、モデルを利用することにより回避することができる。

2 つ目の利点は、1.1.3 項の議論によると人間の取り組むパラメータ調整問題をうまくフレームに切り出すことができれば達成でき、良い性能のパラメータを見つけられると考えられる。この利点は、最適化アルゴリズムの方が人間に比べて、多次元空間での探索性能が高いこと、最適化アルゴリズムでは情報通信のオーバーヘッドが少ないこと起因すると考えられる。人間による手動パラメータ調整においては、モデルの計算結果を UI 上で観察し、背後の現象を分析し、仮説を立てて次のパラメータを決定し、UI 上で入力するという手続きが必要になり、1 つのパラメータの試行のためのオーバーヘッドが多い。また、この 1 つ目の利点は 2.3 節のようにいくつかの最適化の関連研究が存在することからも示唆される。

3 つ目の利点は、アルゴリズムが見つけた人間のものとは異なる解をもとに、人間が解を一般化して知識とすることができるという利点である。同一の製品種類の別の機種においても同様な傾向を持つパラメータだと性能が良いとか、背後ではどのような現象が起きているかなど、人間が最適化の過程や結果を分析して一般的な知識に昇華させることができる。これは、1.1.2 項で述べた、アルゴリズムの打ち方を参考に訓練を行った将棋のプロ棋士の棋力が向上することと似ている。また、将棋においてはゲーム自体を変更することはできないが、機械設計においては新しいパラメータ (新しい使い方) の発見によって、機構の改善点が見つかることもある。

4 つ目の利点は数値解析モデル、目的関数の開発に関する利点である。数値解析モデルも他の 2 形態のモデルと同じく、実機とモデルとの間に誤差が存在する。ただ、数値解析モデルのアプローチでは性能にかかわるもの以外にも様々な物理量が出力されるため、様々な面から実機の実験結果と照合することができ、誤差の原因や修正すべき箇所を突き止めやすい。さらに、統計モデルのように関数近似のためにデータをとるアプローチに比べ、数値解析モデルのように根底にある物理現象を合わせにいくアプローチの方が、少ない回数でモデルと実機が合うようになると考えられる。前者は探索空間の広さに応じたデータ数の取得が必要であるが、後者は物理現象の種類に応じた修正回数で済むためである。

数値解析モデルは物理現象を表し、目的関数は発生している物理現象が人間や設計現場にとってどれだけ好ましいかを表す。この両者を明確に分けることによって、開発が行い



やすくなると考えられる。数値解析モデルは物理現象を表しているため、物理現象が大きく変わらない限り数値解析モデルを使いまわすことができる。例えば、制御対象の機械の運転条件が変わった場合や、機械の機種が変わった場合では、数値解析モデルを使いまわすことができると考えられる。また、目的関数の役割として、数値解析モデルで模擬することが難しいものを一部模擬することがある。数値解析モデルの形に落とし込めていない現象や、模擬すると計算コストがかかってしまう現象を模擬することを目的関数が担う。このように、数値解析モデルと目的関数を分け、その間の物理量も利用することにより、数値解析モデルと目的関数の開発を効率的に行うことができる。

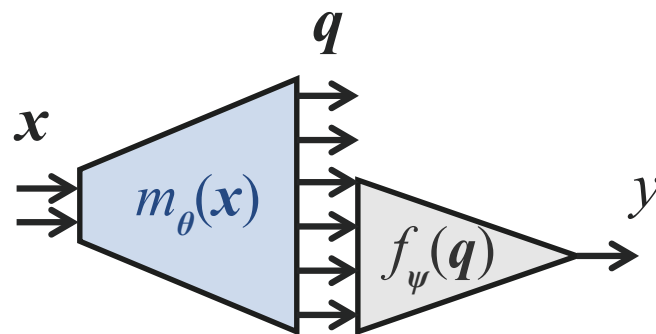


図 2.8 数値解析モデルと目的関数に関わるデータ。

図 2.8 は数値解析モデルと目的関数に関わるデータの概略図である。ここでは数値解析モデルと目的関数の分離の仕方について定性的に説明する。図 2.8 左の数値解析モデルはパラメータ  $x$  を入力として、物理量  $q$  を出力する。 $x$  に対して、数値解析モデルの中の情報が付加されるため、 $q$  は  $x$  に比べて量が多くなる。そして、物理量  $q$  の一部が図 2.8 右の目的関数に入力され、目的関数は性能の評価値  $y$  を出力する。ここでは、 $y$  はスカラー値であり、性能評価のために  $q$  の物理量を要約したものになっている。本研究においては、物理現象が模擬される部分を数値解析モデルと呼び、 $q$  の次元数を下げる要約部分を目的関数と呼ぶ。 $q$  には多くの情報が含まれているため、要約して  $y$  を計算することや、数値解析モデルの妥当性を判定するという用途に利用できる。

## 2.4.2 数値解析モデルを活用した最適化システムの課題

一方で、数値解析モデルを活用した最適化システムには、以下のような課題が存在する。いずれの課題も、数値解析モデルを活用した最適化システムを実際の設計に利用しようとした際に生じる課題である。

1. 目的関数の定義が人間にとって自明ではないこともある
2. 数値解析モデルには誤差が含まれる



### 3. 数値解析モデルの中には計算の複雑性が大きいものがある

1つ目の課題は、目的関数の定義に関する課題である。目的関数は発生している物理現象が設計にとってどの程度好ましいかを表す関数である。設計における複数の目標の存在や、定式化されていない暗黙の目標が存在することにより、目的関数の定義は自明ではない。一方で、設計や最適化の時間的コストの観点から、複数の目標の重要度付けや、暗黙の目標の抽出及び定式化は、短い時間で行われることが好ましい。

2つ目の課題は、数値解析モデルの誤差に関する課題である。現象をモデル化し、計算機上で模擬可能な形式にする際には近似誤差が発生する。特に、複数の種類の物理現象を考慮する必要がある場合は、近似誤差が発生しやすい。近似誤差が存在すると、ほとんどの場合で、数値解析モデルと実機とで最適パラメータに差が生まれる。このパラメータの差は、ほとんどの場合、近似誤差が大きくなるにしたがって大きくなる。人間は数値解析モデルの出力と、実機実験の結果を照らし合わせて、誤差原因を分析し、数値解析モデルの修正を行うことができる。1つ目の課題と同じく、設計や最適化の時間的コストの観点から、モデルの修正は、短い時間で行われることが好ましい。

3つ目の課題は、数値解析モデルの計算時間に関する課題である。数値解析モデルにおいて、多くの要素や現象を考慮すれば精度は高くなるが、現象を模擬する際の計算量が大きくなり、多くの計算時間を要する。数値解析モデルを活用した最適化においては、パラメータ  $x$  を変更しながら、何度も現象の模擬のための計算を行う。そのため、最適化には、現象の模擬するための計算時間と、最適化のためのサンプリングの回数を掛け合わせた分の時間を要する。1つ目、2つ目の課題と同じく、設計や最適化の時間的コストの観点から、1つのパラメータあたりの数値解析モデルは高速に動作することが好ましい。

このように、実機械に対して最適化結果を利用するためには、「数値解析モデルを活用した実機械システムの最適化」を行うだけではなく、数値解析モデルや目的関数の修正も行う必要がある。数値解析モデルや目的関数はプログラミング言語や数式により有限の文字数で記述できるものであるため、それらの記述もパラメータと見なすことができる。以降の議論では、数値解析モデルの構成の記述と、目的関数の構成の記述を表すパラメータをそれぞれ  $\theta, \psi$  で表す。これらのパラメータを利用して、3つの課題は以下のように書ける。

1.  $\psi$  で表される目的関数の構成を、設計の目標を反映させた形に短時間で修正する
2.  $\theta$  で表される数値解析モデルの構成を、誤差が小さくなるように短時間で修正する
3.  $\theta$  で表される数値解析モデルの構成を、現象の模擬にかかる時間が小さくなるように修正する

以上のことをまとめると、実機械において最適化結果を利用するためには、与えられた数値解析モデルと目的関数の元でパラメータ  $\boldsymbol{x}$  を最適化するのみならず、数値解析モデルと目的関数の内部パラメータ  $\boldsymbol{\theta}, \boldsymbol{\psi}$  も最適化する必要があるといえる。次章ではこれらの研究課題に対するアプローチについて述べる。

次章に入る前に、数値解析モデルと目的関数の内部パラメータ  $\boldsymbol{\theta}, \boldsymbol{\psi}$  の利用法について整理する。これらのパラメータは次のように利用される。

- 数値解析モデル :  $\boldsymbol{q} = m_{\boldsymbol{\theta}}(\boldsymbol{x})$
- 目的関数 :  $y = f_{\boldsymbol{\psi}}(\boldsymbol{q}) = f_{\boldsymbol{\psi}}(m_{\boldsymbol{\theta}}(\boldsymbol{x}))$

また、目的関数の内部パラメータ  $\boldsymbol{\psi}$  は、目的関数を構成する指標  $\phi$  と、指標の重みづけ  $\boldsymbol{w}$  を含む。

上記の議論から、 $\boldsymbol{\theta}, \boldsymbol{\psi}$  は修正される必要がある。そのため、 $\boldsymbol{\theta}$  は数値解析モデルにとってのパラメータ (変数) であり、 $\boldsymbol{\psi}$  は目的関数にとってのパラメータ (変数) である。一方、数値解析モデルを活用した最適化システム (図 2.7) においてはこれらのパラメータは固定されて計算が行われる。一般的に、モデルや目的関数の主たる変数は、入力である環境条件  $\boldsymbol{e}$ 、制御パラメータ  $\boldsymbol{x}$ 、物理量  $\boldsymbol{q}$  である。それに対し、 $\boldsymbol{\theta}, \boldsymbol{\psi}$  は助変数となるため、本研究では  $\boldsymbol{\theta}, \boldsymbol{\psi}$  を添字記法を使って表す。

## 第 3 章

# 最適化システムの最適化のためのメタな最適化システムの提案

本章では、2 章で紹介した数値解析モデルを活用した最適化システムを構築及び修正していくためのメタな最適化システムについて述べる。まず、「メタな最適化システム」という最適化システムを構築していくための新しい視点を定義し、メタな最適化システムの内部構成について述べる。次に、メタな最適化システムの視点にたって最適化システムの持つ問題を解決するための手法を提案する。そして最後に、提案する手法を検証するための方法について述べる。

### 3.1 数値解析モデルを活用した最適化システムの問題

ここではまず、2 章で提起した、数値解析モデルを活用した最適化システムが持つ問題に再度触れる。2 章では、「実機械に対して最適化結果を利用するためには、パラメータ  $x$  のみならず、数値解析モデルと目的関数の内部パラメータ  $\theta, \psi$  も最適化する必要がある」という課題が提起された。これは、「与えられた数値解析モデルと目的関数の下でパラメータを最適化するだけではなく、目的関数や数値解析モデルの修正をする必要がある」とも言い換えられる。

目的関数の修正のためには機械の性能を評価するための基準を持っている必要があり、数値解析モデルの修正のためには現実で起こっている物理現象を分析する能力が必要である。そのため、両者の修正においてはどこかで人間が介入する必要があり、アルゴリズムによる完全な代替が難しい。言い換えると、実機械で良い性能を発揮するようなパラメータを最適化により求めるには、人間とのアルゴリズムとの協調の必要があるといえる。本論文では、数値解析モデルを活用した最適化システムの構成要素である、数値解析モデルや目的関数、最適化手法を繋ぎ合わせて、計算機上での最適化が行えるようにすることを

最適化システムの構築と呼称する。そして、数値解析モデル、目的関数の両者を修正し、設計者にとって好ましい状態にすることを、最適化システムの改善と呼称する。

最適化システムを構築・改善し、実機械で良い性能を発揮するようなパラメータを求めするために人間が行っている作業は、以下のように多岐にわたる。

- 現実の現象の観察を基に、数値解析モデルを開発、修正
- 数値解析モデルを開発する際に、取りうる実装案の精度や安定性と計算量を試算して、都合の良い実装を選択
- 数値解析モデルでは考慮できないような製品の価値に関する部分を目的関数として定義
- 他の人間が蓄積してきた自然言語の形の既存知識を利用
- 現実の現象を観察するために実機実験の準備を行い、実機実験を実施

2.3 節で紹介した最適化システムの研究では、最適化システムを構成する要素の説明や最適化結果の議論が中心となっており、目的関数や数値解析モデルに誤差があった際にどのように修正していけばよいかの議論はされてこなかった。また、最適化アルゴリズムの出力する最適化結果に対して、人間の知識をどう提供して最適化システム自身を修正・改善させていくのかについては、体系的にまとめていない。ただ、方法について体系的にまとめられてはいないものの、研究論文として表出してこない現場レベルにおいて、モデルの修正や目的関数の修正は行われている。体系化されていないため、このような修正行為は属人化していたり、最適化の題材に特化していたりすると考えられる。加えて、体系的にまとめられていないと、最適化を行うにあたって何を目標せば良いかがわからず、無駄な工数がかかったり、行動を考えるコストがかかることが発生しうる。体系化が行われていない理由としては、体系化や一般化には複数種類の最適化問題を経験が必要であるが、複数種類の最適化問題を経験するのは難しいということがあると考えられる。

本研究においては、修正・改善の作業を体系的にまとめるために、最適化システムより一段階広いメタな最適化システムという考え方を提唱し、数値解析モデルや目的関数の開発・修正方法を整理する。メタな最適化システムとして考えることにより、最適化システムを構築・改善する行為を効率的に行う方策を考えることができる。最適化システムが効率よく構築・改善されるということは、実機に利用できる最適なパラメータが効率よく求められるということにつながる。また、本研究においては、メタな最適化システムの考え方の提唱に加えて、メタシステムを効果的に動作させるための仕組みを提案する。

### 3.2 メタな最適化システムの具体例と位置づけ

メタな最適化システムの定義の説明を行う前に，具体例を用いてメタな最適化システムの指し示す対象や，メタな最適化システムについて考えることについての意義を説明する．

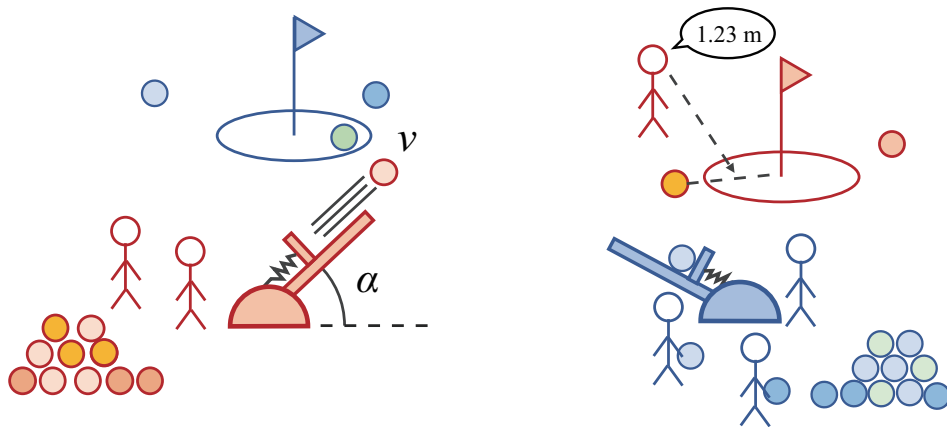


図 3.1 球入れ対戦競技の例.

ここでは，図 3.1 で示す 2 チームで争う競技を考えてみる．この競技においては，各チームに用意された発射台からボールを発射し，相手側のゴールに 10 球を先に入れられたチームが勝利となる．それぞれのチームは 3 人のメンバーから構成されている．一度に発射できるボールの数は 1 球で，チームで利用できる球の数の制限はない．発射台は初速  $v$  と射角  $\alpha$  を調整することができ，ボールの色やボールの材質を変更することもできる．

このようなルールのもと，この競技で勝つための方法として以下のようなものが考えられる．

1. 相手より早く 10 球を入れるためには，ボールの跳び方の法則を素早く掴むことが重要であり，そのためにはボールを発射してみて，発射条件と発射の結果を記録する必要があるであろう．
2. 跳び方の法則が掴めたら，それに基づいて発射条件を修正することにより，相手側のゴールにボールを入れられる確率を上げられそうである．
3. 掴んでいた法則と実際の発射結果とが異なった場合，法則を修正することで，発射条件修正の確実性が上がりそうである．
4. 法則を知っているメンバーがいたらそのメンバーに法則を聞くことで，試射の回数を削減できるかもしれない．
5. 球を発射台に素早くセットすることが有効そうである．

#### 6. 初期に素早く作戦を立てることも有効そうである。

このような競技において勝つために実行される行為が本研究におけるメタな最適化に相当する。箇条書きで書かれた勝つための方法は人間の持つメタな最適化に関する知識である。

箇条書きの1つ目におけるボールの跳び方の法則は2章で述べたモデルに対応しており、モデルを構築するためには実験が必要である。モデル構築のための実験の進め方として、実験計画法などが知られている。ここでのモデルは2章で述べたどの形態も考えられる。精度の高いモデルを構築するためには、発射の結果の高精度な計測が必要であるが、高精度な計測にもコストがかかる。例えば、メンバーの一人をボールの着地点付近に配置して、距離を測ってもらうと高精度な計測ができるが、情報伝達や計測にかかる時間コストは、目視の場合と比べて大きくなる。

箇条書きの2つ目で行っているのがモデルを利用した最適化である。重力のみ考慮した記号モデルだったら、手計算でも妥当な発射条件を求められるかもしれない。精度を上げるために、空気抵抗やボールの変形、気流の変化を考慮する場合、記号モデルでは表現が難しく、数値解析モデルが必要になる。また、重力のみを利用した記号モデルを利用して最適な条件を求められるというのは数学の知識を前提としている。チームのメンバーに数学的思考ができる人がいなかった場合や、競技が行われているのが数学が発展していない世界であった場合は、手計算で妥当な条件を求めるのは難しくなる。数値解析モデルを利用した最適化が競技の勝利につながるか否かは、モデルを構築するのにかかる時間や、手持ちの計算資源がどの程度使えるかによって変わってくる。計算資源が利用できない場合は、なんらかの方法で計算資源を調達する必要がある、追加のコストが必要となる。

人間がこのような競技を行うとき、挟叉修正という手法を使うことがある。挟叉修正は、目標の奥に着地した場合、次の発射では目標の手前に着地するようにフィードバックを行い、条件を修正するというものである。挟叉修正においても、条件の修正方法は条件と結果の間の関係のモデルに基づいて行われる。例えば、初速を遅くしたら手前に着地するようになるというのは、初速と到達距離の間のモデルに基づいた知識である。また、ボールの色は跳び方に関係なさそうと見抜いて、ボールの色の条件(パラメータ)を変えないというのは、モデル化とモデルを利用した最適化に含まれる。

箇条書きの3つ目はモデルの修正である。モデルに誤差があるという観点に立って、モデルを修正していくことも重要である。強化学習における  $\epsilon$ -greedy 法や最適化におけるバンディッドアルゴリズムのように、モデルに誤差があることを前提としてモデルを構築していく手法があるが、それらの手法と関係している。

箇条書きの4つ目はモデルに関する知識の転用に関するものである。この問題(競技)の場合、重力や空気抵抗に関する知識を持っていると、問題を解く効率をあげることがで

きる。モデルが一般化されて転用できるようになっていると、他の問題を解く際に有利である。複数の問題を解く場合、序盤の問題において転用可能性のある知識を作成しておけば、終盤の問題では問題を効率的に解くことができ、転用可能性のある知識を作成しなかった場合に比べて、問題1つあたりにかかる時間を短くできる。

箇条書きの5つ目は実験コストに関係するものである。今回の競技のルールからすると、最低でも10回はボールをセットする必要がある。ボールのセットの方法について、最初の数回で時間をかけて試行錯誤をすることで、トータルとしてセットの合計時間を短くできるかもしれない。

箇条書きの中には含まれていないが、モデル化の方法や条件決定方法は考慮せずに、条件をランダムに変えてボールを発射するという作戦も考えられるかもしれない。ゴールが広く、勝利条件の球数が1球のように少ない場合は、ランダム戦略の有効性が上がる。また、競技としては非現実的かもしれないが、ボールを1球打つのに1日かかる場合もあるかもしれない。この時は、観測にかかるコストや思考実験にかかるコストは相対的に小さくなる。このように、モデル化や最適化などの行為の必要性は問題の前提条件が変わることにより変動する。

2章で触れたような数値解析モデルを活用した最適化システムは箇条書きの2つ目に対応する。この競技の例でみたように、人間は数値解析モデルを活用した最適化の外側においても効率の勝負をすることが可能である。手に入る計算資源の量や、ボールがどれくらい変形するかのような物理的制約は最適化の外側においてのみ考慮することができる。逆に最適化の内側では、数値解析モデルを使わないという選択をしたり、高精度な計測をするように変更するといったことはできない。さらに人間は、どのようにモデル化をすればよいか、どのように数値解析モデルを使った最適化していけばよいかという、一段階外側のメタなモデルを扱うことができる。

この競技の例を使って関連研究の整理をしたい。繰り返しになるが、2章で述べたような、ホワイトボックス最適化手法やブラックボックス最適化手法とモデルとを組み合わせた最適化は、箇条書きの2番目の内容に対応する。与えられたモデルの中で最適化を行うことができたとしても、モデルの精度が悪かったり、最適化にかかる時間が長かった場合、勝利につながる確率は低くなる。

近年の機械学習手法との比較も行う。データドリブンやAIベースと呼ばれる手法においてはモデルとして統計モデルが利用され、人間のような論理を使ったモデル化は行われない。また解くべき問題に応じて、記号モデルと数値解析モデル、統計モデルとを使い分けるということも行われていない。World models[75]という強化学習手法は、脳内にモデルを持つことを人間の知性の本質ととらえ、モデルを使った先読みのような知的行動機能をエージェントに持たせている。ただ、モデルの形態や複雑さも決め打ちとなっており、人間が行うような柔軟なモデル化は出来ていない。Sim-to-real[76]と呼ばれる一連の研究

では、人間がエージェントに対して事前に洗練された数値解析モデルを与えている。エージェントは計算機上において数値解析モデルを使って訓練を行い、行動方策を学習する。解くべき問題としては、3次元ロボットの運動のような問題が設定されることが多く、そのような問題では既に数値解析モデルが洗練されているため、エージェントの環境を計算機上から現実に変えても妥当な行動をする。Sim-to-realの研究においては、エージェントは与えられた数値解析モデルの修正は行わない。

### 3.3 メタな最適化システムの定義

本節においては、2章で紹介した最適化システムに比べて、一段階外側の最適化システムであるメタな最適化システムの定義を行う。まず、メタな最適化システムの基本的な構成とシステム内部の動作について述べる。そして、メタな最適化システムの位置づけを行うために、世の中で行われている取り組みとメタな最適化システムとの比較を行う。以降の文章においては、システムの区別をつけやすくするため、2章で紹介したようなメタではない最適化システムのことを非メタな最適化システムとも呼ぶ。本研究においては、メタな最適化システムの中でも、モデルとして数値解析モデルを利用したシステムについて議論を行う。

#### 3.3.1 一段階メタなシステムの基本構成

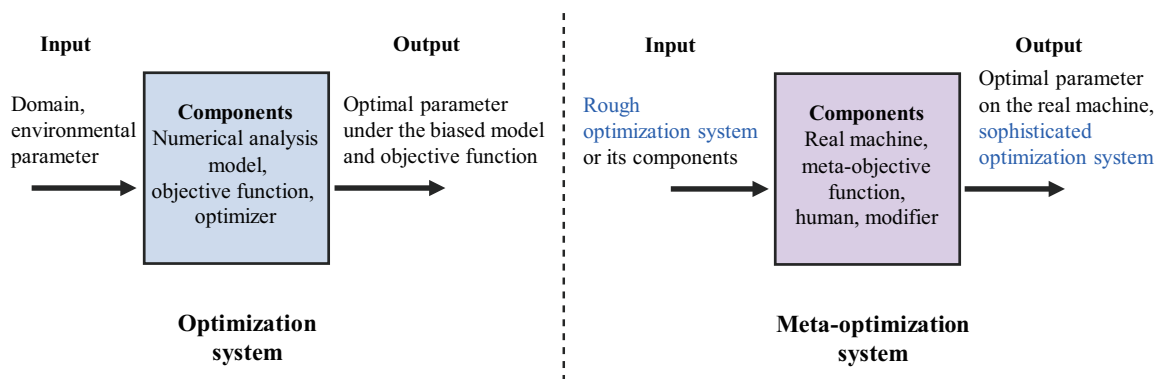


図 3.2 非メタな最適化システムとメタな最適化システムの比較。

図 3.2 では非メタな最適化システムとメタな最適化システムを比較している。ここでは、両システムの構成要素と入出力を取り上げて、両者の比較を行う。

■**非メタな最適化システム** 非メタな最適化システムは、数値解析モデル、目的関数、最適化手法から構成される。システムへの入力、パラメータの定義域や機械の動作条件である。また、システムからの出力は、与えられた数値解析モデルや目的関数の下で最適化



されたパラメータである。非メタな最適化システムの内部では、図 2.7 のような最適化の手続きが行われ、パラメータが最適化される。

■メタな最適化システム メタな最適化システムは、実機、メタな目的関数、人間（設計者）、修正アルゴリズムから構成される。システムへの入力は、非メタな最適化システムやその構成要素である数値解析モデルや目的関数である。また、システムからの出力は、実機に対して最適化されたパラメータと、非メタな最適化システムである。システムの入出力のどちらにも、非メタな最適化システムが存在するが、これはメタな最適化を通して、非メタな最適化システムが最適化されることを表している。メタな最適化の内部では、非メタな最適化システムが動作するだけでなく、人間や修正アルゴリズムが非メタな最適化システムの構成要素を修正する。

非メタ、メタの両システムとも動作のために計算資源や時間、人間の介入が必要であり、それらがシステムの動作コストとなる。

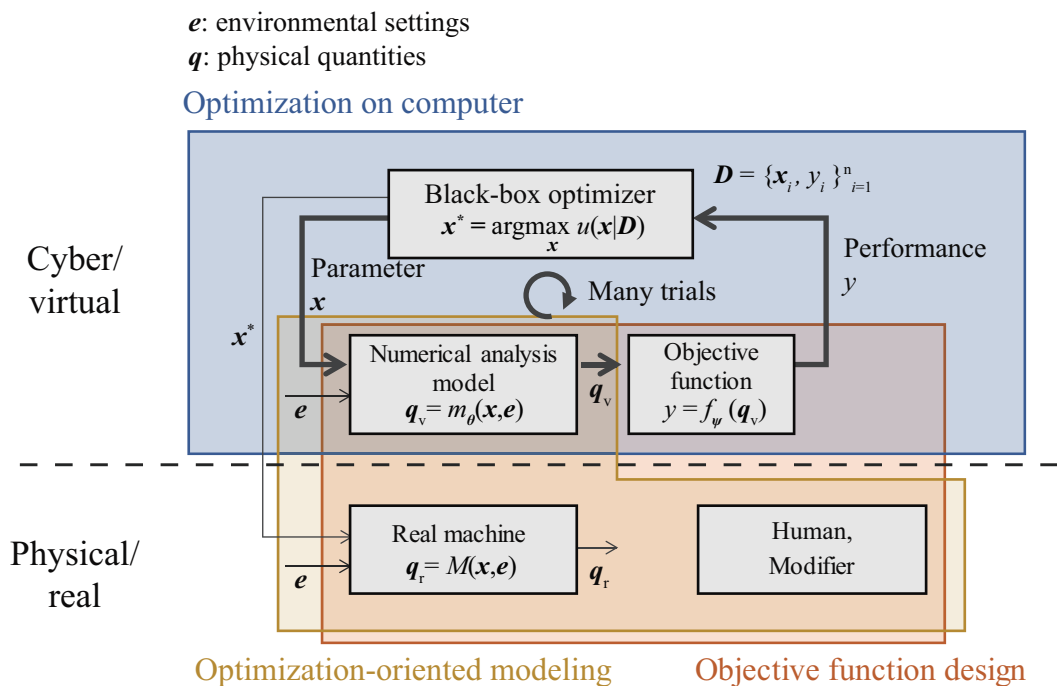


図 3.3 最適化システム構築のための 3 行為。

図 3.3 は数値解析モデルを活用した最適化システムに関する要素と、それらを修正するためにメタな最適化システムの内部で行われる動作を表した図である。関係する要素は図 2.7 とほぼ同じであるが、最適化システムを構成する 3 要素と実機だけではなく、人間と修正アルゴリズムが加わっている。メタな最適化システムの内部で行われる動作は大きく分けて 3 つあり、それぞれ、計算機上の最適化、最適化志向のモデリング、目的関数の設計である。

メタな最適化システムにはメタな目的関数があると考えることができ、メタな目的関数は実機械に対して有効なパラメータを取り出すという目標が達成されたときに大きくなるように設定される。そのため、メタな目的関数は以下のような、複数の目標から成り立っていると考えることができる。

- 目的関数  $f_\psi$  の値を大きくする
- 数値解析モデル  $m_\theta$  と実機  $M$  との間の誤差を減らす
- 目的関数  $f_\psi$  の定義を設計の目標を反映したものにする

2.2 節で紹介した最適化手法は、少ないサンプルで目的関数を最適化することが求められたように、メタな最適化システムは、これらの目標の達成を少ないコストで行うことが求められる。コストに関係する要素には以下のようなものがある。

- 実機実験に必要な装置、資源
- 計算資源
- 時間
  - 計算機上の最適化にかかる時間
  - 準備も含めて実機実験にかかる時間
  - 人間が構成要素を組み上げて最適化システムを構築するための時間
  - 人間が現象を分析することにかかる時間
  - 人間のデータ入出力にかかる時間
  - 人間が数値解析モデルの修正を行う時間
  - 関係する人間同士での情報のやり取りのための時間
- 人件費

アルゴリズムによるメタな最適化の完全な自動化が行えたならば、人間に関係する時間や、人件費は必要なくなる。しかし、現時点の技術では、メタな最適化の完全な自動化は困難であることから、人間に関係するコストを減らすことを考える必要がある。

文献 [20] においては、精度の高い数値シミュレータそれ自身がノウハウや知識であるとされている。精度の高い数値シミュレータを実現するためには、現象それ自身や、現象をどのように計算機上で模擬するべきかを知る必要がある。そして、現象をうまくとらえた、精度の高い数値シミュレータは他の機種に転用した際も利用することができる。そのため、上記の目標にある数値解析モデルの誤差を減らすことは、メタな最適化の結果として得られる知識の量を増やすことにもつながる。

また、最適化システムを図 2.7 のようにモジュール化して一度構築してしまえば、二度目以降の構築では構築のための時間を削減することができる。

メタな最適化システムを構築するための部品としては、以下のようなものがある。

- 数値解析モデル
- 目的関数
- ブラックボックス最適化手法
- 実機
- 設計知識
- 設計者
- 修正アルゴリズム

ここでは、最適化システムを構成する部品の、数値解析、目的関数、ブラックボックス最適化手法だけではなく、実機や設計のための知識、人間の設計者や修正のためのアルゴリズムを利用することができる。これらの前提を踏まえて、メタな最適化システムの構成を考える。

### 3.3.2 メタな最適化システムの内部動作

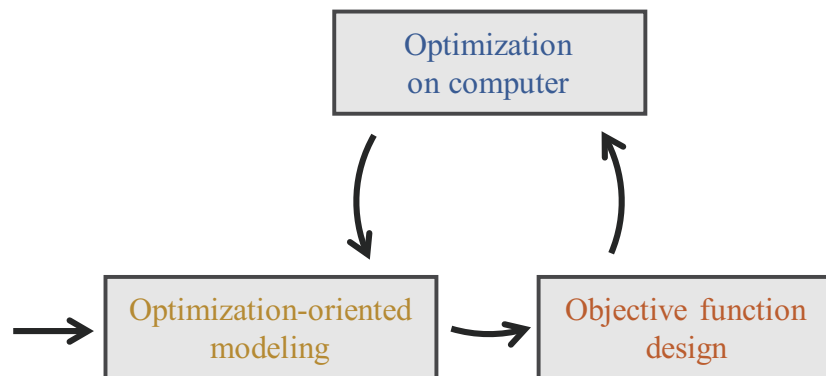


図 3.4 メタ最適化システムの動作 (目的関数が固定されていない場合).

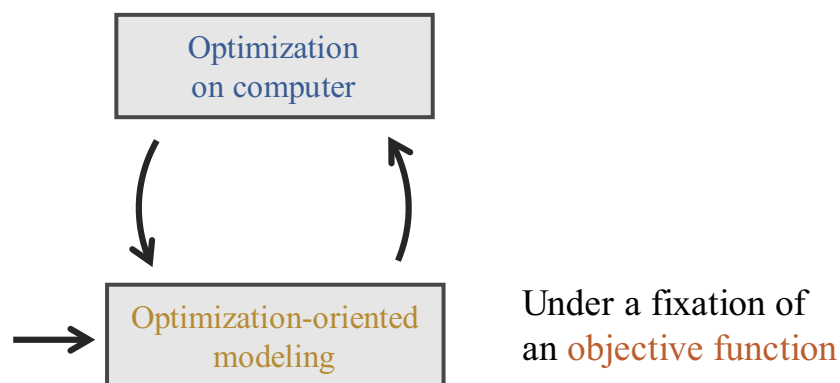


図 3.5 メタ最適化システムの動作 (目的関数が固定されている場合).

図 3.4 と図 3.5 は提案するメタな最適化システムの動作フローの概略図である。これらの動作フローは、計算機上の最適化、最適化志向のモデリング、目的関数の設計の3つの動作から構成される。

図 3.4 は数値解析モデルと目的関数の両方の改善が必要な場合の動作フローであり、図 3.5 は目的関数が確定している場合の動作フローである。どちらのフローにおいても、数値解析モデルと目的関数の開発・修正と、計算機上での最適化とは交互に行われる。

ここからは、3つの動作の内容について説明していく。説明の中では、各動作に関する要素、自動化が可能か、コストの決定要因についても記す。

■**計算機上の最適化** 計算機上の最適化においては、人間の手を介さずに、固定された数値解析モデルと目的関数の下で、パラメータの最適化が行われる。2.4 節においても述べたように、最適化の結果として、最適化されたパラメータだけではなく中間出力の各種物理量も取り出すことができる。計算機上の最適化は、人間の手を介さずに自動的行われるプロセスであり、数値解析モデルの計算時間と最適化手法のサンプル効率が時間コストを決定する。

- 関係する構成要素：数値解析モデル、目的関数、最適化手法
- 自動か手動か：自動
- コスト決定要因：数値解析モデル計算時間、最適化手法のサンプル効率

■**最適化志向のモデリング** 最適化志向のモデリングにおいては、物理現象の背後の支配方程式を明らかにしたり、支配方程式を数値計算可能な形式に落とし込むことが行われる。モデリング(数値解析モデルの開発・修正)は主に人間が行う。

以降の内容は次の「目的関数の設計」にも当てはまる内容である。最適化志向のモデリングは、一般的な精度を追求するモデリングとは異なり、メタな最適化の目標を達成するために行なわれる動作である。数値解析モデルを利用した最適化においては物理量を取り出されるため、その物理量を観察しながら多角的に要素の開発・修正を行うことができる。人間は定性的な推論が得意であるが、数値計算速度は遅いという特性を持つ。そのため、モデルや目的関数の構成に関する仮説を形成し、モデルや目的関数を利用して仮説検証実験を行い、その結果を受けてモデルや目的関数を構成していくという進め方が効率的であると考えられる。モデルや目的関数がない、生身の人間のみの場合は、仮説を形成しても、検証実験が行えない。その意味でも、人間と、計算機上のアルゴリズムとの協調が必要である。人間は、数値解析モデルが出力する物理量を中間出力も含めて考慮することができ、物理量の間にも論理的矛盾があった場合にはその検知ができ、修正につなげることができる。また、計算機上においてモデルを構築する際に、出力される物理量を観察しながら、手順で必要ない計算を省きモデルの計算量を低減させるようなこともできる。

- 関係する構成要素：数値解析モデル，実機，人間，修正アルゴリズム
- 自動か手動か：手動
- コスト決定要因：実機実験，現象分析，モデル修正

■目的関数の設計 目的関数の設計においては，設計における目標の定式化や，設計に複数の目標の定義が行われる．目的関数の設計・修正は主に人間が行う．

- 関係する要素：数値解析，目的関数，実機，人間，修正アルゴリズム
- 自動か手動か：手動
- コスト決定要因：実機実験，現象分析，目的関数修正

### 3.3.3 非メタな最適化システムとメタな最適化システムとの視点の比較

本項では，非メタな最適化システムとメタな最適化システムとの視点の比較を行う．システムの設計に携わる人間がどこに注目しているかを比較し，両システムの差分，研究対象の差を示す．

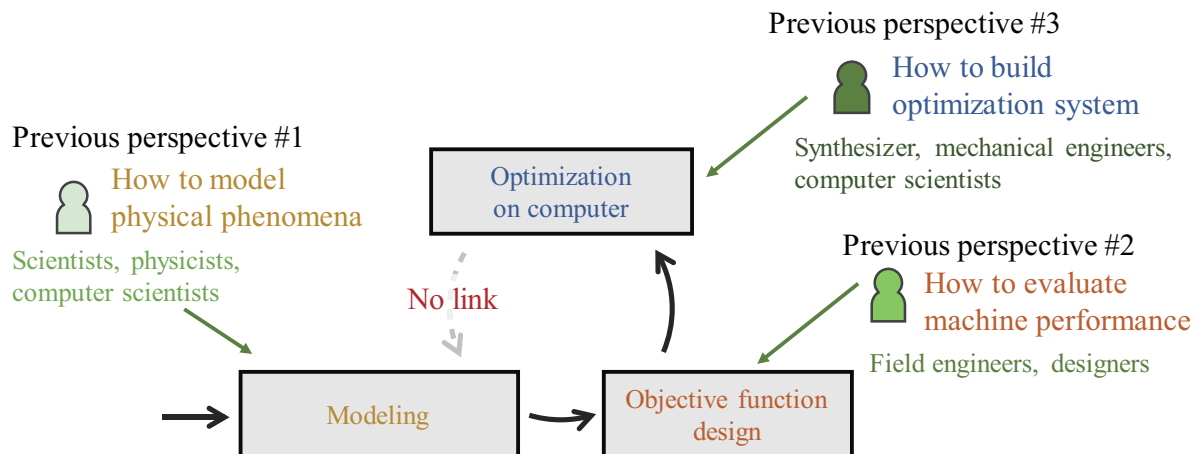


図 3.6 最適化システムとその要素を開発するとき視点.

図 3.6，図 3.7 はそれぞれ，最適化システムを開発するときの視点と，メタな最適化システムを開発するときの視点である．

図 3.6 は最適化システムやその要素を開発するときの複数の視点を表しており，複数の視点に基づいた研究や開発が行われている．

モデリングは，科学者や物理学者，計算機科学者が中心となって行われる．機械システムに関係する物理現象は機械工学の範疇で研究されることがあるが，その中では物理学の手法が使われていたり，研究内容が応用物理学会等の物理系の学会に投稿されることもあ

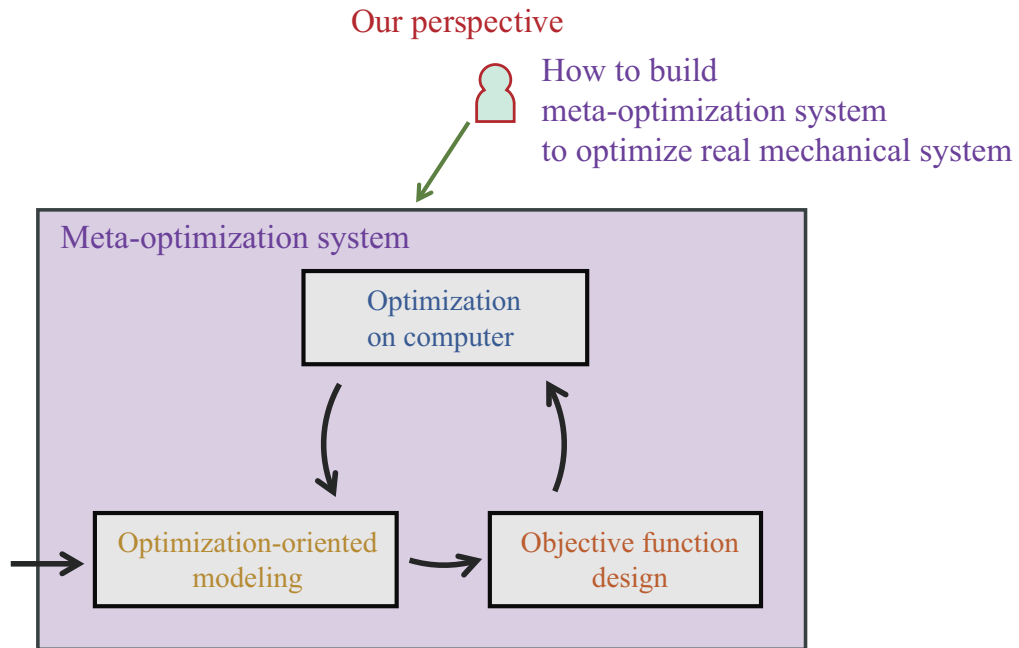


図 3.7 メタ最適化システムを開発するときの視点.

る．そのため機械システムに関係するモデリングを行う研究者は物理学者の中に含まれると見なしている．モデリングの際には物理現象の背後の支配方程式を明らかにしたり，支配方程式を数値計算可能な形式に落とし込むことに注意が払われる．

目的関数の設計は，機械システムの設計目標を知っている，設計現場の技術者や設計者が中心となって行われる．目的関数の設計の際には，目標の定式化や，複数の目標の重みづけに注意が払われる．

最適化システムの構築や計算機上の最適化は，統合 (Synthesize) を得意とする工学者，機械工学者，計算機科学者が中心となって行われる．最適化システムの組み方や計算機上の最適化の際には，モデルの精度と実行時間のトレードオフや，最適化手法の効率に注意が払われる．

機械システムの最適化の中でも，2.3 節で紹介したトポロジ最適化の一部は，既に設計支援のツールとして製品化されている．製品化モデリングの原理や，最適化システムはほとんど確定されているため，設計現場において目的関数を設定すれば，トポロジの最適化が行えるようになっている．

図 3.7 は本研究の視点である，メタ最適化システムを開発するときの視点を表している．著者はメタ最適化システムをどのように組めば，実機に利用できる最適なパラメータを効率的に求めることができるかに注意を払っている．紫色の枠内のメタ最適化システムの要素や要素間のつながりが本論文において議論される内容となる．

### 3.4 メタ最適化システムの安定性や性能に関する考察

本節ではメタ最適化システムの安定性や性能について、著者の持つ定性的な考えを述べる。メタ最適化システムの形として整理できる問題の範囲は広いが、その広さや柔軟性と引き換えに厳密な議論は難しい。定性的な考えでも、メタ最適化システムの全体像の理解を促進したり、考え方の断片が後の研究の助けになる可能性があると考えたため、この説で紹介する。

まず、メタ最適化によって求まる最適化システムの理想形を考える。この理想形はメタパラメータ  $p$  の最適解のことである。理想形では、モデルが 100% の精度を出しており、モデルや最適化に時間がかからないと考えられる。このような理想形がもし存在するのであれば、メタ最適化システムの自明な最適解である。しかし、このような理想形は、モデルがラプラスの悪魔のように実機のすべての物理的状态を把握して物理現象も理解している必要があり、不確定性原理から存在が否定されることが考えられる。そのため、最適化システムが改善されていったときの最終到達点は自明ではない。

メタ最適化がうまく行かない場合を考えよう。ある  $p$  を設定すると、その  $p$  の下で非メタ最適化が自動的に走る。また実機実験は人手が必要である場合もあるが、実機実験の人は  $p$  の設定とは無関係である。そのようなときに  $p$  の修正を何も考えずランダムに行うと、メタ最適化はうまく行かないと考えられる。 $p$  の修正を行うためには、非メタ最適化システムを回した結果を解釈できる程度の知性が最低限必要である。ランダムな修正では、猿がタイプライタの鍵盤を無限にたたき続ければいつしか文学作品が出来上がるという「無限の猿定理」のように、妥当なパラメータに行きつく確率がパラメータの数が増えるほど指数関数的に小さくなる。

メタ最適化システムはメタパラメータ  $p$  を最適化するものであるが、その最適化に際して、メタパラメータ  $p$  やそれを構成する  $\theta$  や  $\psi$  が不安定になることは少ないと考えられる。不安定な状態や発散する状態とは、数値解析モデルを構成するパラメータ  $\theta$  が最適な  $\theta_*$  に向かわず、計算時間やモデルの誤差が小さくなっていかない状態と言い換えることができるだろう。十分に知識があり、論理的思考ができる人間は、 $\theta$  の修正によってモデルの誤差や計算時間が大きくなってしまった場合に、修正前の  $\theta$  に戻ることができる。そのため、十分に知識があり論理的思考ができる人間が  $\theta$  の修正を行う場合、計算時間やモデルの誤差が大きくなってしまわないと考えられる。一方で、修正に有用な知識が十分でない場合や、モデルの誤差や計算時間の評価が正しくできていない場合は、モデルの構成  $\theta$  をどのように修正していけばよいかわからず堂々巡りになったり、 $\theta$  が発散することも起こりうる。また、 $\theta$  を正しく記録していないと、元の状態に戻すことができない。モデルを構成するパラメータ  $\theta$  が巨大だった場合、過去の  $\theta$  を保存しておくこ

ともコストになり、保存がされないことも起こりうる。

制御工学などで登場するコントローラは、コントローラの出力を自由に設定することができるが、システム出力を自在に設定することは出来ない。一方でメタ最適化システムにおけるコントローラにあたる人間はシステム出力である  $p$  を自由に設定できることが多いと考えられる。そのため、 $\theta$  や  $\psi$  の修正に問題があった場合、修正前の状態に戻すということが行いやすい。自由に  $p$  を設定できない場合というのは、例えば、修正者が複数人いてそれぞれで思惑が違う時や、特許のような社会的事情により既存の実装とは違うものを出力することを求められていた時に生じる。

メタパラメータ  $p$  を必ず改善する方法ではなく、改善方向に行く可能性が高い方法はあることができる。これはブラックボックス最適化におけるヒューリスティクスに相当する。以下に列挙するような点を踏まえることで改善方向に行く可能性を高くすることができると思われる。

1. その方向に修正することの効果を論理的に予測できたり、説明できる
2. その方向の修正は別の問題において効果を出した実績がある

1 点目における論理は、記号論理学のような厳格な論理ではなく、研究者などが仮説を立てる際の論理に近い。修正方向は無数にあるが、論理的説明ができるという制約を加えると修正方向を絞ることができ、修正をした際に改善方向に行く可能性を上げることができる。一方で、ハイパーパラメータを調整して計測値に合うようにするなどという論理的に説明がしづらい修正は、一旦は改善するかもしれないが、外挿性が悪かったり、転用可能性に乏しかったりする。2 点目について、例えば学術論文では、従来のモデルの問題を指摘し、それに対し修正を施した提案モデルとその効果を記述することがある。このような知識を利用してモデルの修正を行うことで、改善方向に行く可能性を上げることができる。良い科学論文は多くの人に通じる論理で書かれているため、多くの人が利用することができ科学の発展を大いに支えている。

本論文で扱っている制御パラメータの話から逸れてしまうが、制御パラメータより広い機械システムの設計パラメータを考えると、メタ最適化が不安定になっていない例が多数存在する。ここ 100 年間の飛行機の設計パラメータの改善と、飛行機の挙動を模擬するモデルの改善の過程はその一例である。この 100 年間で飛行機の手軽さや安全性といった性能評価値は向上した。またそれと合わせるように飛行を模擬するための手法も進化していると考えられる。主に利用されるモデルは、単純形状の翼や単純な燃焼を模擬していた記号モデルから、複雑形状や複雑な燃焼を模擬できる数値解析モデルに変化し、模擬の性能も向上している。また、数値解析モデル自身も成長している。与えられる計算資源が同じだとしても、現時点での数値解析モデルは 30 年前のものと比べて良い性能を示すだろう。このようなモデルの着実な進化は  $\theta$  が着実に改善されているということを示してい



る。現時点での最新の数値解析モデルに行きつくまで、学会等で様々なモデルが提案されている。その中には論理的には妥当性があったが、実用性に乏しく採用されなかったモデルもあるだろう。このようにメタな最適化では、常に最適解に向かって修正が進むというわけではなく、局所最適解に向かって修正されることもある。

### 3.5 メタな最適化システムを構成する機構の提案

メタな最適化システムにおける新たな機構を提案する。機構とは特定の機能を達成するための仕組みのことである。メタな最適化システムを構成している、モデルや人間などの要素と働きを組み合わせたものを機構と呼ぶ。

3.3 節で述べた通り、本研究は数値解析モデルを活用したメタ最適化システムを議論の中心に置いている。本節において提案する機構も、数値解析モデルの利用を前提としたものになっている。数値解析モデルの利用を前提にしない、メタな最適化を効率的に回す機構はいくつか考えられるが、範囲が広がってしまい十分な議論ができなくなるため本研究では具体例を簡潔に述べるのみとする。数値解析モデルの利用を前提としない機構の具体例としては以下のようなものが考えられる。

1. 統一されたフォーマットでモデルに関する知識を残す
2. 最適化対象のパラメータを絞る

■1. 統一されたフォーマットでモデルに関する知識を残す モデルに関する知識を残す際に、統一されたフォーマットで残すことにより、モデルを理解する効率が上がる。ここでの理解とは、あるモデルが何を表しているか、どのような理論に基づいているか、どのような利用法があるか、どのような仮定を置いているか、などのモデルに関する知識を知のことを指す。科学論文のフォーマットもこの統一的なフォーマットの一つである。科学論文のフォーマットでは、背景・提案・実験・考察・結論という構成が統一されていたり、英語がデファクトスタンダードとして利用されていたりする。科学論文を読む際は、フォーマットが統一されていることにより、論文内で紹介されているモデルへの理解を効率的に行うことができる。

■2. 最適化対象のパラメータを絞る 最適化対象のパラメータを絞ることにより、探索空間の広さを削減することができ、探索空間内の最適解を見つけやすくなる。絞る前の最適解と絞った後の最適解は必ず一致しない。絞る際には、制御対象の現象に大きく関係するパラメータを探索対象から外さないことが重要である。パラメータが現象に関係しているかを知るためには、要因分析や感度分析等の多変量解析手法が有用である。

ここからは、数値解析モデルの利用を前提とした機構の提案について述べる。本研究に

おいては3つの機構を提案する。

### 3.5.1 機構 1. 最適化結果の重要度に基づいた要素修正

1つ目の機構は、最適化結果の重要度に基づいた要素の修正機構である。ここでの要素とは数値解析モデルと目的関数のことを指す。この機構では、最適値から遠いパラメータにおいて要素の修正を省くことにより、要素の修正にかかる実験コストやデータ分析のコストを低減する。

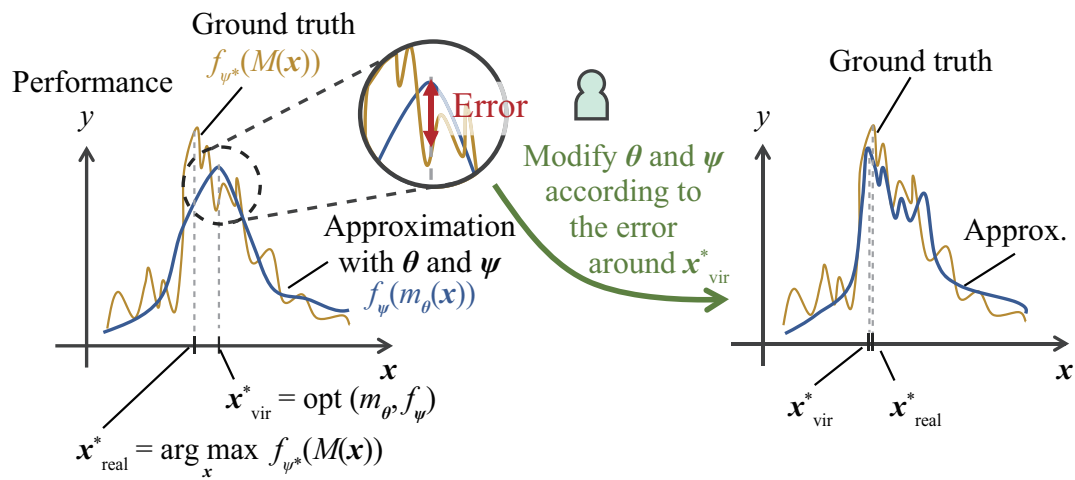


図 3.8 最適化結果の重要度に基づいた要素修正。

動作としてはまず、図 3.8 左のように数値解析モデルと目的関数の定義を固定した状態で最適化を行い、最適なパラメータを算出する。図 3.8 において、黄色の線が実機における性能を示し、青色の線は数値解析モデルと目的関数により近似された性能を示す。次に、その最適化された最適パラメータを数値解析モデルと目的関数や実機に入力し、出力される物理量や目的関数の値を観察する。そして、観察結果に基づきモデルや目的関数の定義を修正する。この、最適化、観察、修正のサイクルを繰り返すのが最適化結果の重要度に基づいた要素修正機構である。重要度は、一つのパラメータに対して定義されるものであり、目的関数の値が大きくなるようなパラメータにおいて高くなる。この機構により、目的関数の値(性能)を高くする効果のある物理現象に関して、要素の修正が優先的に行われるため、図 3.8 右のように目的関数の値が高くなるようなパラメータ付近での要素の精度が高くなる。人間が最適化システムの支援なしでモデル修正や目的関数の修正を行う場合、試すことのできるパラメータの数には限りがあるため、目的関数の値が大きくなるようなパラメータの付近では精度が低いままということが起こりうる。

一方で、要素の定義が完全であれば目的関数の値が大きくなるが、要素が完全でないため値が小さくなっているパラメータがあった場合、そのような部分の修正は取り組まれに

くく、パラメータの本来の有用性も気づかれない。

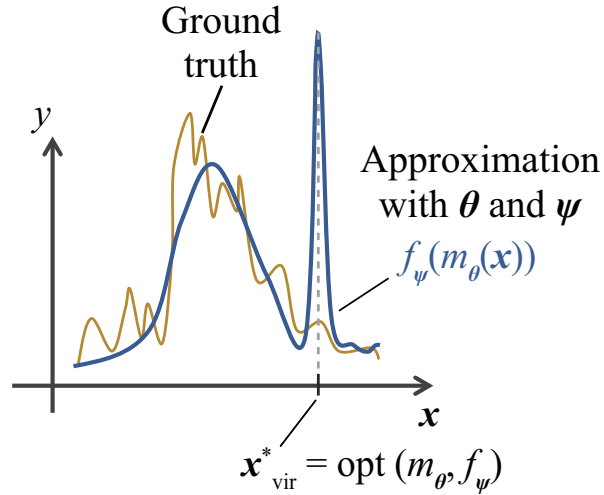


図 3.9 要素が想定外の挙動をした際の要素修正。

また、この機構においては、図 3.9 に示すように数値解析モデルがあるパラメータにおいてバグのような想定外の挙動をした結果、そのパラメータが最適値となるといった場合も検知が可能である。

### 3.5.2 機構 2. 数値解析結果の再利用

2 つ目の機構は、数値解析結果の再利用機構である。この機構では、一度計算した数値解析の結果を再利用することにより最適化全体にかかる時間を低減する。

最適化システムを利用して最適化を行うにあたり、同様のパラメータで同様の数値解析(計算)が繰り返し行われるということが発生しうる。繰り返し行われるのが本質的に同様の計算であれば、一度は愚直に計算を行い、それ以降の同様の計算を行う際には愚直に計算した結果を再利用可能である。このように最適化において、愚直な計算の結果を記録しておいて、同様のパラメータにおいて計算結果を再利用するのが、数値解析結果の再利用機構である。

人間が機構 2 を利用し、既にある数値解析モデルの計算量を削減を行うことにより、少ない計算量で最適化が行えるようになり、非メタな最適化、ひいてはメタな最適化における時間コストが低減される。

図 3.10 は計算の類似性に注目した有効パラメータ数の削減イメージである。図 3.10 の 2 つのグラフは  $x$  の張るパラメータ空間を表しており、青い点が最適化において取得されるサンプルである。この機構では、図 3.10(b) のように類似の計算結果となるパラメータを領域  $r_i$  によりくくり、この領域の中に入ったパラメータの数値解析の結果を記録して再

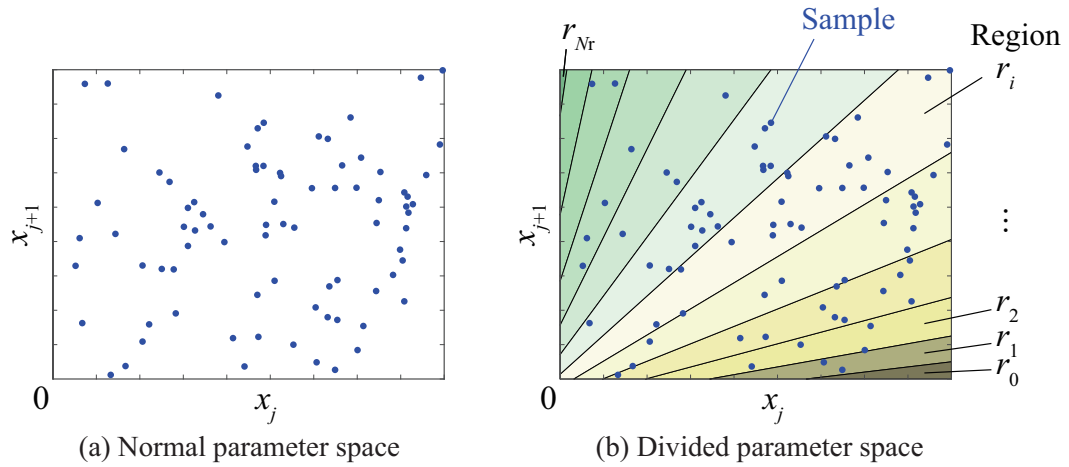


図 3.10 計算の類似性に注目した状態数の削減イメージ。

利用する．図 3.10 においては線形な境界により領域を分割しているが，分割の方法は数値解析の対象によって異なる．

この機構により，愚直な計算を行う回数が低減され，トータルの計算量を低減することができる．図 3.11 は計算量の削減イメージを示すものである．最適化におけるサンプル数を  $N$ ，類似な状態の種類数を  $N_r$ ，確率的な振舞いをさせるための類似な計算の繰り返し数を  $k$  であるとして，再利用により， $(N - kN_r)$  回分の愚直な計算を削減することができる．ただし，類似の状態の種類数  $N_r$  や類似の計算の繰り返し数  $k$  は，最適化におけるサンプル数  $N$  に比べて小さいものとしている．例えば， $N$  が 10000， $k$  が 5， $N_r$  が 50 で再利用の計算量が，愚直な計算の計算量と比べて無視できるとすると，計算量は  $\frac{1}{40}$  となり，1 ヶ月かかっていた計算が 1 日弱で完了できるようになる．

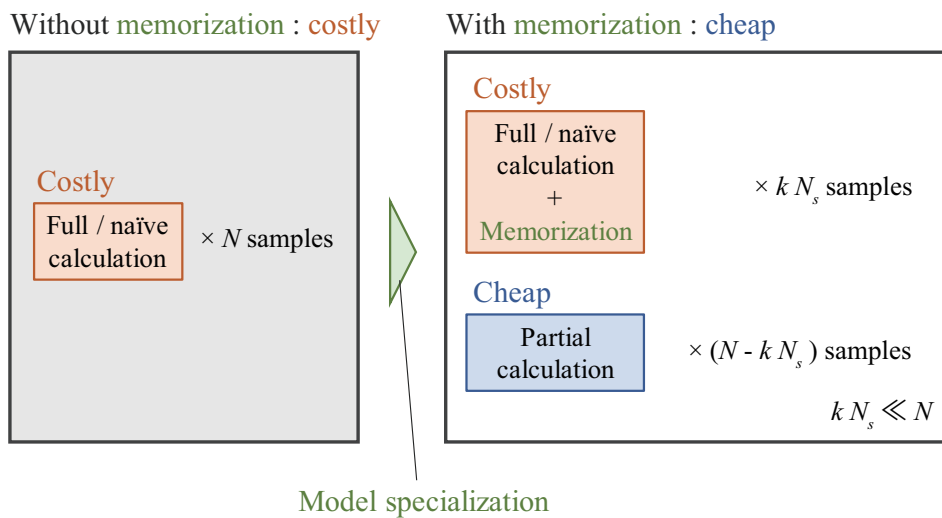


図 3.11 数値解析結果の再利用による計算量低減。

一方で、再利用に対応したモデルの構築にもコストがかかるという問題が存在する。この問題は、数値解析モデルを再利用可能なように対応させる際に追加で必要な実装のコストの問題である。再利用に対応した数値解析モデルの有用性が確かめられていくことで、構築を容易にする方法論やライブラリが開発されていくと考えている。また、本論文は構築のための方法論を一部提供するものでもある。

### 3.5.3 機構 3. 補正係数導入によるモデル修正の自動化

3つ目の機構は、補正係数導入によるモデル修正の自動化機構である。この機構では、数値解析モデルの修正を自動化し、人間を介した修正のオーバーヘッドを削減する。

この機構では、図 3.12(a), (b) のように実機と数値解析モデルとの間で生じている誤差を、モデルの補正により図 3.12(c) のように低減させる。それにより、最適パラメータ付近での数値解析モデルの誤差を低減し、最適化システムにより求められた最適パラメータが実機においても利用できるようにする。

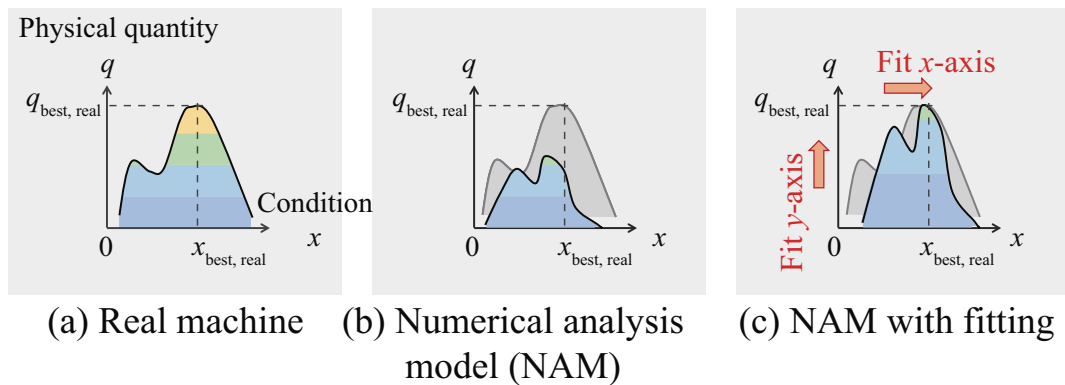


図 3.12 関数形状を保持したモデル修正.

モデルの補正は、図 3.13 のように数値解析モデルの内部パラメータ  $k_{\text{model}}$  と、入出力を補正する係数  $k_{\text{in}}$ ,  $k_{\text{out}}$  によって行われる。内部パラメータや補正係数は数値解析モデルと実機から出力された物理量を利用して決定する。本機構は、実機のデータを使って数値解析モデル自身のシステム同定を行っていることに相当する。内部パラメータや補正係数によるモデルの修正には人間の高度な判断が不要なため、修正のオーバーヘッドを減らすことができる。

一方で、補正では数値解析モデルの計算方法を修正できないという問題や、補正係数の過学習の問題が存在する。前者は、内部パラメータや係数の調整では、数値計算のアルゴリズム自体は変更できないということに起因する問題である。これを避けるためには、人間がモデルを、係数を調整すればよい段階まで作りこんでおく必要がある。後者の問題は得られる実機のデータ数に対して、係数の種類が多い場合に起こる問題である。機械学習

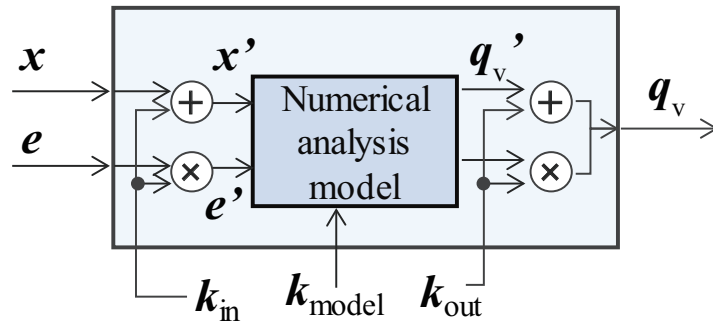


図 3.13 補正係数によるモデル修正.

における過学習のようにデータ数に対して係数の数が多い場合は、データに対する誤差を減らすことができるが、関数の形の分散が大きくなってしまう。実機実験のコストを増大させないためには、係数の数を抑える必要がある。

### 3.6 本章における提案の検証方法

本節では、前節において提案したメタな最適化システムの機構の検証方法について述べる。具体的な実機械システムを題材にし、メタな最適化システムの手順・機構に従って、最適化システムを組み上げ、最適化が行えるかを確認し、検証とする。本節では、具体的な実機械システムの紹介と、その機械システムにとっての最適化システムを構築するためのメタな最適化システムについて紹介する。本章に続く4章、5章においては、具体的な機械システムの最適化システムの構築結果と最適化結果について紹介する。

#### 3.6.1 具体的な実機械システム

図 3.14 は明確な目的関数の有無、数値解析に必要な計算コストという2軸の張る空間に最適化問題を配置したグラフである。本研究では、図 3.14 の中の丸で囲まれている2つの最適化問題に注目した。それぞれの問題は以下のような別々の技術的課題を持つ。本項においては注目した2つの最適化問題の性質について述べる。

■機械製品の性能のように目的関数が自明でない問題 1つ目の最適化問題は、機械製品の性能のように目的関数が自明でない問題である。問題の性質・課題は以下の通りである。

- 性質：計算量が小さい 1D-CAE
- 課題：複数の目標が存在しており目的関数が定式化されていない
- 課題：人間の乗り心地のように定量化しづらい指標がある

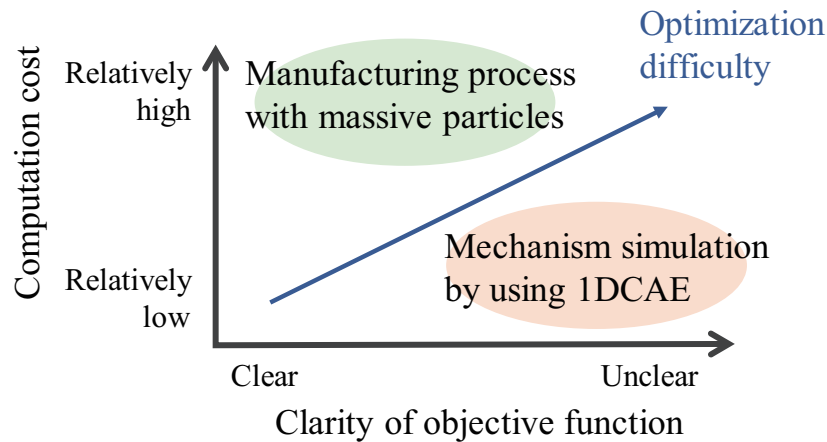


図 3.14 数値解析モデルを活用した最適化が直面する課題.

- 課題：数値解析モデルと現実との間に誤差がある

部品を組み合わせた機械システムの動作に関しては，1D-CAE により動作の模擬が行われることが多い．1D-CAE は 3D-CAE に比べて登場する要素の数が少なく，計算量が大きくなりづらい．

具体的な機械システムとしては，ダンプトラックの電子制御トランスミッションを取り扱う．電子制御トランスミッション (以降 TM) の制御パラメータを最適化するシステムの構築を行う．

■多くの要素が動的にふるまい計算量が大きくなる問題 2つ目の最適化問題は，多くの要素が動的にふるまい，その結果として計算量が大きくなる問題である．問題の性質・課題は以下の通りである．

- 性質：計算量が大きい 3D-CAE
- 性質：目的関数は既知
- 課題：反復計算が容易な高速なシミュレータが登場していない
- 課題：数値解析モデルと現実との間に誤差がある

加工現象のように材料の除去を伴う現象は，時間軸方向の動的なふるまいを見る必要がある．そのため，要素数と時間ステップ数の積の分だけ計算が必要となり，計算量が大きい．一方で，加工は生産性が定義しやすい対象であり，目的関数の定義は容易である．

具体的な機械システムとしては，円筒研削加工機を取り扱う．円筒研削加工の加工条件を最適化するシステムの構築を行う．



### 3.6.2 TM 制御パラメータ最適化システムのメタ最適化システム

TM 制御パラメータ最適化には，目的関数が自明ではないという問題，モデルと実機との誤差の問題が存在する．一方で，TM の動作を模擬する 1D-CAE は計算を必要とする要素数が 3D-CAE に比べると少なく，計算量が比較的に小さい．目的関数の定義も含めて最適化をする必要があるため，メタ最適化システムの動作としては，図 3.4 のような動作になる．

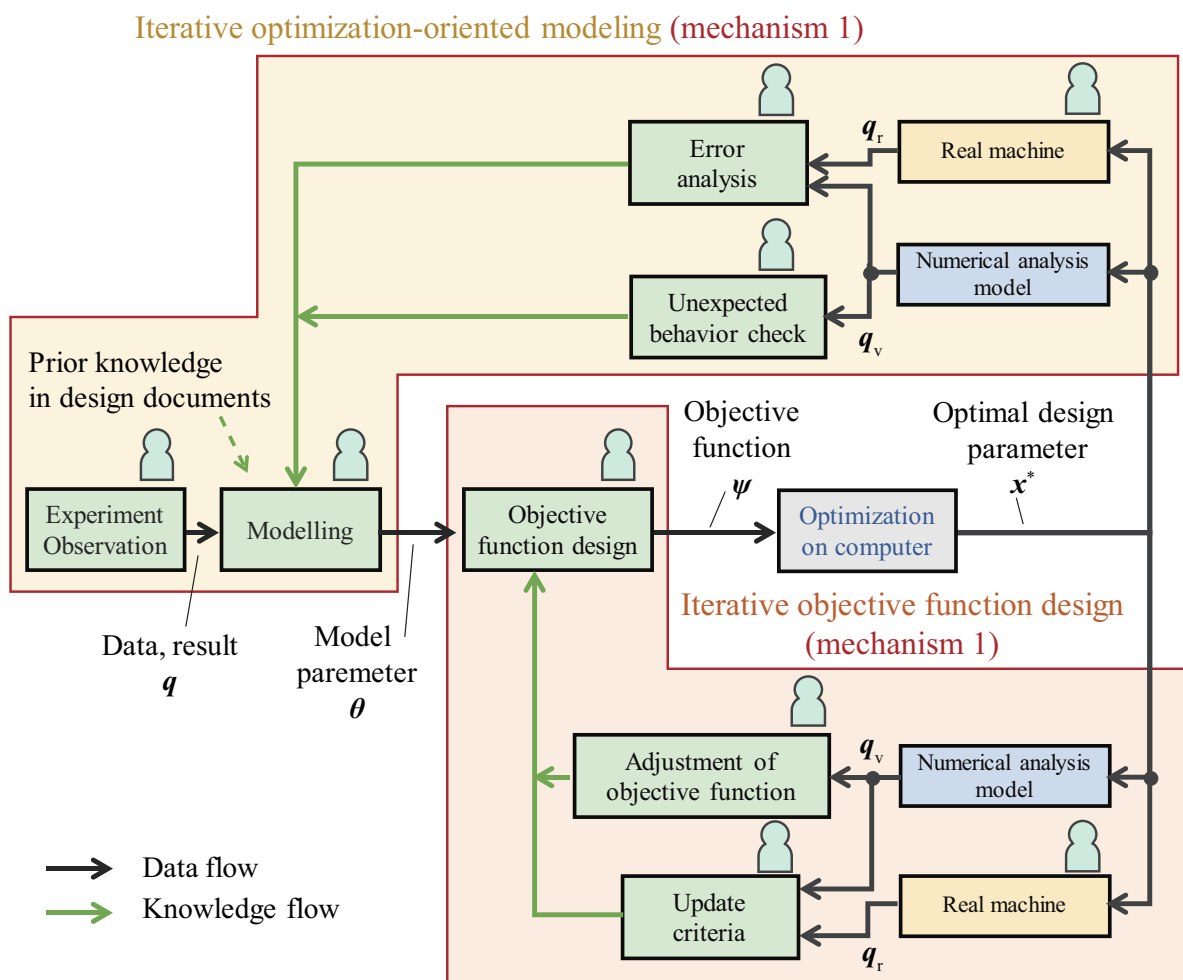


図 3.15 TM 制御のメタ最適化システム．

以上のような最適化問題の性質から，最適化システムを構築するにあたり，機構 1 の最適化結果の重要度に基づいた要素修正機構をメタ最適化システムに取り入れる．機構 1 を取り入れて，図 3.4 の動作を詳細化したのが図 3.15 である．図 3.15 のようなメタ最適化システムにより，最適化システムの構築を行い，実機を含めた最適化を行う．機構 1 では，最適化されたパラメータを要素 (モデル・目的関数) や実機に入力して，誤差の発生箇



所を検出して、要素を修正し、最適化を行うという一連の動作を繰り返す。これにより、最適パラメータを求めながら、最適パラメータ周辺における要素の精度を向上させていく。結果として、実機実験の回数や人間の判断の回数を減らすことができる。モデルの内部パラメータ  $\theta$  と目的関数の内部パラメータ  $\psi$  とともに、 $f_\psi(m_\theta(x))$  が大きくなるようなパラメータ  $x_{vir}^*$  付近で重点的に修正すれば良いという共通点があるため、どちらも機構1により修正することができる。図3.15においては描画の都合により、数値解析モデルと実機が2箇所が存在しているが、これらは同一ものを指している。モデル・目的関数の定義の最適化の中でも、人間の暗黙知を取り出す作業の部分は実機を含めなくても実施が可能である。

### 3.6.3 円筒研削加工の加工条件最適化システムのメタ最適化システム

円筒研削加工の加工条件の最適化には、モデルの計算量の問題、モデルと実機との誤差の問題が存在する。研削加工においては、表面粗さや加工時間などの確立された指標が存在し、目的関数の不明さは小さい。そのため、メタ最適化システムの動作としては、図3.5のような動作になる。

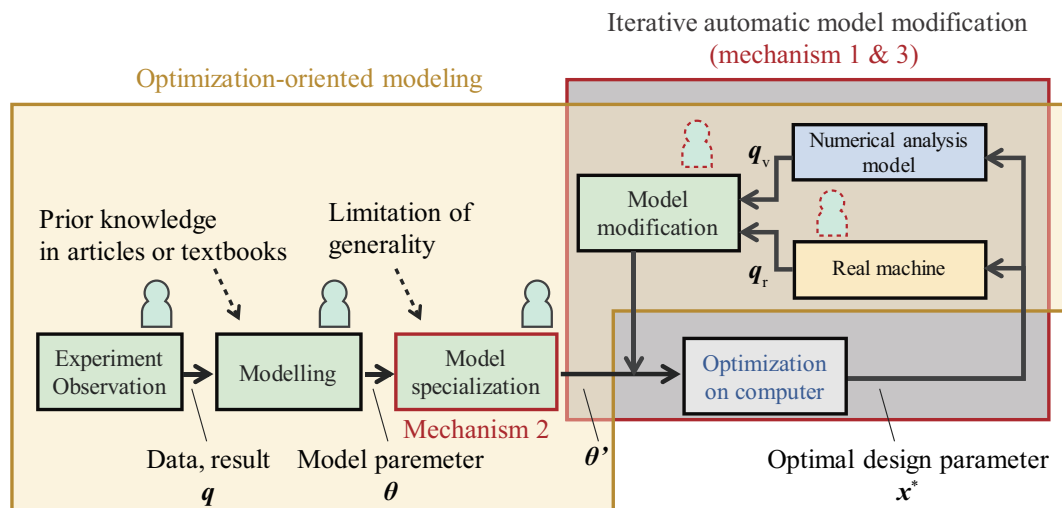


図3.16 円筒研削加工のメタ最適化システム。

以上のような最適化問題の性質から、最適化システムを構築するにあたり、機構1から3の3つの機構を取り入れる。これらの機構を取り入れて、図3.5の動作を詳細化したのが図3.16である。図3.16のようなメタ最適化システムにより、最適化システムの構築を行い、実機を含めた最適化を行う。機構2の数値解析結果の再利用機構により、モデルの汎用性に一部制限がかかるものの、計算量の削減を行うことができる。機構2においては、円筒研削加工の加工の幾何的な性質に注目し、再利用機構を含んだモデルを開発し、計算

時間を削減する。図 3.16 右側の灰色の四角で塗られた領域は、補正係数による自動的なモデル修正を反復的に行う仕組みである。この仕組みにおいては、機構 3 の補正係数導入によるモデル修正の自動化機構により、人手によるモデル修正にかかるオーバーヘッドを削減し、メタな最適化システム全体での時間効率を高める。機構 3 の機能として、計算機による最適化の結果求められたパラメータを利用して、モデルの修正を行う。また、この仕組みは機構 1 の最適化結果の重要度に基づいた要素修正機構も取り入れている。機構 1 の機能として、最適化されたパラメータを実機実験に利用して補正係数の計算を行い、数値解析モデルを修正し、最適化を行うという一連の動作を繰り返す。これにより、最適パラメータを求めながら、最適パラメータ周辺における数値解析モデルの精度を向上させていく。

### 3.6.4 2つの最適化問題における課題が複合する場合

3.6.1 項において紹介した 2 つの最適化問題における課題が複合するような最適化問題も存在する。そのような問題は、対処しなくてはならない課題が多い分、解決にコストはかかるものの、問題の分解により 3.6.2 項、3.6.3 項のような機構を利用することで効率的に解決することができると考えられる。本論文においては、そのような問題の具体的なケーススタディは取り扱わないが、解決の方針について述べる。

具体的な最適化問題としては以下のような問題が考えられる。

1. 新しいドラムを搭載した洗濯機の制御の最適化
2. 新開発エアコンのエアフロー制御の最適化
3. 流動性のあるビーズクッションを持つ椅子の設計の最適化

多くの機械システムにおいては、設計やパラメータ調整の難易度を下げるため問題が分割されており、上記のような課題が複合することは少ない。ただ、今までになかったような新製品を開発する際は、問題が複合することがある。

ここでは、1 つめの具体例である「新しいドラムを搭載した洗濯機の制御の最適化」の対応方針について述べる。洗濯機の制御においては、以下のような複数の性能目標が存在する。

- 衣服の汚れと洗剤の成分の接触回数を増加
- 衣服の繊維へ加わるダメージを低減
- 消費電力を低減
- 利用する水の量を低減
- 機外に伝わる振動を低減

- 洗濯時間を低減

この問題においては新しいドラムを搭載した洗濯機を扱っているため、今までの制御パラメータ調整の目的関数が使えず、上記のような性能目標の重みづけを新たにやり直す必要がある。また、以前に洗濯機を開発したときと比べて、洗濯機ユーザの部屋の防音性や、選択される衣類の質、クリーニング店の手軽さなどの環境条件が変わっている可能性があり、このようなことも重みづけをやり直す必要性を生じさせる。このように、「新しいドラムを搭載した洗濯機の制御の最適化問題」は「機械製品の性能のように目的関数が自明でない問題」と同じく、目的関数が自明でない。

また、数値解析によって上記の性能目標の達成度を求めるためには、液体挙動や不定形な物体の挙動、機体の振動、消費電力を考慮する必要がある。このように、「新しいドラムを搭載した洗濯機の制御の最適化問題」における数値解析は「多くの要素が動的にふるまい計算量が大きくなる問題」と同じく、多くの計算量を必要とする。

これらの対処方針としてはまず、機構2をはじめとする計算量低減のための工夫を行い、数値解析モデルの計算量を削減する。計算量削減のための工夫には、設計空間を分割という方法がある。例えば、30分間に及ぶ洗濯の全部の数値解析を行うのではなく、洗い、脱水、すすぎ等の工程を事前に分割し、それぞれの工程の中で最適化するというものである。計算量の削減から取り組むのは、メタな最適システムの機構1を回すための時間を短くする効果が高いためである。計算量の削減をせずに、機構1を回そうとした場合を考えよう。数値解析モデルで1パラメータを計算するのに1日かかっており、最適化を収束させるのに10,000サンプル必要だとしたら、機構1を回すのに10,000日かかってしまう。そうすると、目的関数や数値解析の修正を行う前に、製品の開発期間が終わってしまう。このようなことを避けるために、数値解析モデルの計算量削減を先に行う。

計算量の削減ができれば、「TM制御パラメータ最適化システムのメタ最適化システム」と同じような問題に帰着できるため、図3.15の「TM制御パラメータ最適化システムのメタ最適化システム」のフローに沿ってメタな最適化を行っていけばよい。

## 第 4 章

# 電子制御トランスミッション制御パラメータの最適化

本章では数値解析モデルを活用した電子制御トランスミッションの制御パラメータ最適化について述べる。電子制御トランスミッションの制御パラメータ最適化の研究背景について説明した後、数値解析モデルをベースとした最適化システム、及び、実機を用いた評価実験について述べる。

### 4.1 研究課題

本節では電子制御トランスミッションと制御パラメータの最適化の関連研究について述べた後、本研究において取り組む研究課題について述べる。

#### 4.1.1 電子制御トランスミッション

自動車におけるトランスミッションとはエンジン動力を効率的に利用するため、車速や負荷に応じエンジン回転を変速する機械要素である。エンジンにはパワーが最大となる回転数が存在するが、トランスミッションが歯車機構の変速比を切り替え、入力側のエンジン回転数を一定の範囲に抑えることにより、様々な車速で効果的にパワーを出力できる[77]。そのような役割から、トランスミッションはエンジンと駆動輪との間に配置される。

トランスミッションは変速の形態から AT (Automatic Transmission), MT (Manual Transmission), CVT (Continuously Variable Transmission) などが存在する。AT と MT は歯車機構が内蔵されており、噛み合う歯車の組み合わせを変更することにより離散的に変速比を切り替える形態をとっている。対して、CVT は名前にもある通り、連続的に変速比を切り替えられるような形態をとっている。CVT の連続的な変速は、歯車によるものではなく、チェーンやベルトが巻きついている 2 つのローラの径が変化することにより

実現されることが多い。AT と MT では歯車機構の切り替えを自動で行うか、ドライバーが行うかが変わってくる。AT はドライバーが歯車機構の切り替え作業を行わなくてよい分、運転が楽になるためトランスミッションの主流になっている [77]。AT では、速度が変化するタイミングを見計らって油圧アクチュエータが駆動され連結・接続を切り替えるクラッチが作動し変速が行われる。

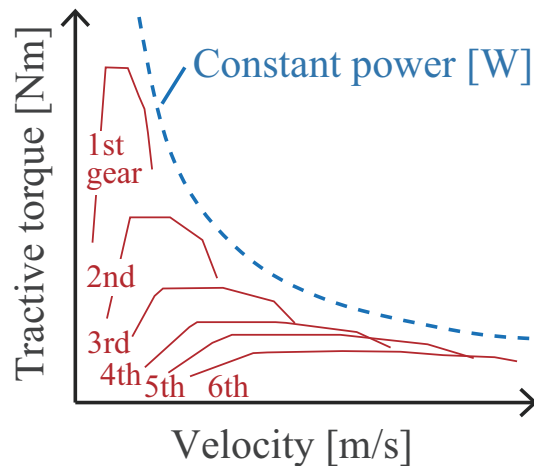


図 4.1 トランスミッションの性能を示す車速駆動トルク曲線。

図 4.1 は AT・MT による離散的なトランスミッションにとって重要な車速—トルク曲線である。図 4.1 の青色点線は線上の点においてパワーが等しい等動力線を示し、赤色線は変速の段数ごとの車速—トルク曲線を表す。ある車速における青色点線と赤色線の間のトルク差がエネルギー損失に比例する。変速を多段にすればするほど、多くの車速帯でトルク差を小さくすることができるため、トランスミッション変速の段数を増やすという多段化が進んでいる [77][78][79]。変速時に一時的にエンジンの回転数と車体速度とに不整合が生じ、ドライバーの乗り心地を悪化させる変速ショックが発生してしまう。

AT は運転の楽しさから様々な車種に取り入れられている [77]。そして、AT も機械要素であるため、車種ごとの要求や制約を受け、様々なタイプの AT が生み出されている。車はエンジンの配置場所、駆動輪の場所から FF, FR, 4WD などのタイプが存在し、それぞれ省スペース、走行の安定性、悪路の踏破性などの長所を持つ。また、エネルギー損失と変速ショックを減らすため AT の多段化が進められているが、多段化するに従いトランスミッションの部品数及び体積が増大するため、6 速、8 速の段数でバランスしている。機械に対する制約によって設計の在り方が多様化するの、1 章で述べたような機械全般や 5 章で述べる工作機械と同様である。

AT も機械であるため、1.1.1 項において述べた機械設計に対する要求を継承する。AT には機能に対する要求や、上位の機械である自動車からの要求から、変速効率が高いこと、変速ショックが小さいことが求められる。複数の機能目標、設計目標が存在する

るが，その中でも変速ショック低減の重要度が高いと言われている [77].

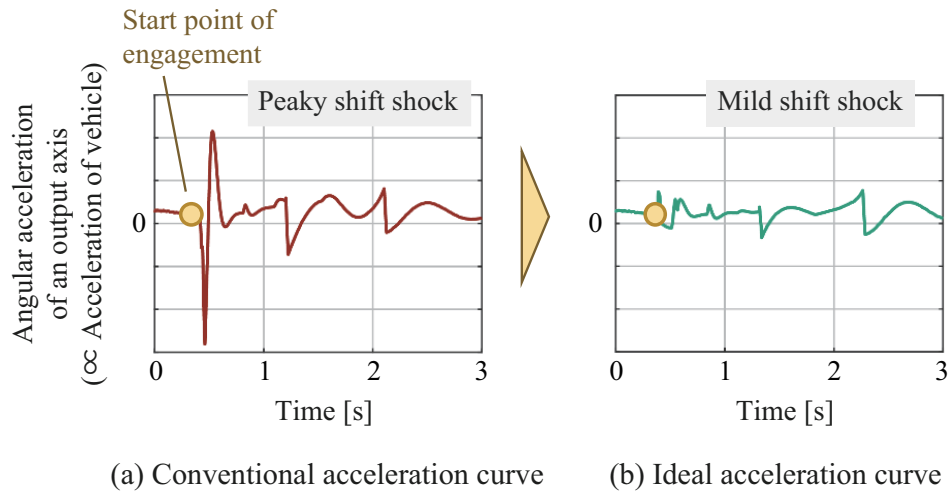


図 4.2 変速ショックの実際と理想.

変速ショックはクラッチが離れたり結合するタイミングにおいて図 4.2 のように発生する．図 4.2 のグラフは横軸が時間で，縦軸がトランスミッションの出力側の軸の角加速度である．出力側の軸である出力軸の角加速度は，車体の加速度と比例の関係にあり，車体の加速度の変動はドライバーの乗り心地や車両の積荷に影響する．設計の際には，図 4.2(a) のような加速度変動が大きい変速ではなく，図 4.2(b) のような加速度変動が小さい変速が求められる．変速ショックに対しては従来より，変速段数を増やしショックの全体量を減少させる，変速時の制御を工夫し不整合を緩和させる等の対策が行われてきた．前者はエンジンの効率向上の副次的な効果でもあるが，実施にはトランスミッション全体の設計を変更する必要がある．それに対し，後者の対策は制御機構に自由度を持たせておけば実施が可能であり手軽である．後者の一例として，油圧のライン圧やエンジンの出力を低減する制御が提案されている [80]．これらライン圧やエンジン出力などの制御パラメータの最適化は設計者が手作業で行っており，多大な時間的および人的コストを必要としているため，自動化が求められている [80][81]．これまでパラメータのテストは実験ベンチや実機にて行われてきたが，高性能な数値シミュレーションの登場や設計情報のデジタル化により計算機上での低コストなテストが可能となってきた．本研究は AT の中でも，普通自動車に比べ走行条件が大きく変動しパラメータの設定が難しい，ダンプトラックの電子制御トランスミッション (ECT; Electronically Controlled Transmission) の変速ショック低減を目標とする．大野木らの研究 [81] ではダンプトラックの 11 種類の制御パラメータを調整している．パラメータが 11 種類あるとそれぞれを 5 水準に量子化した場合でも，とりうるパラメータの組み合わせは  $5^{11} = 48,828,125$  組となり，1 パラメータのテストが 1 秒間と仮定しても全てを試すのに 565 日が必要なる．そのため，全てを試すような

愚直な探索は計算時間の点で非現実的である．このような問題に対しては，最適化手法を利用することにより少ない試行回数で良いパラメータを見つけ出すことができると考えられる．また，上記のような制御パラメータを一括に調整できれば，現状に比べて複数の指標において良い変速を実現できると考えられる．本章の取り組みにおいては，手動で行われてきたトランスミッション制御パラメータを自動で最適化するシステムの構築を目標とする．

#### 4.1.2 関連研究

本節ではトランスミッションの制御パラメータ最適化に関する関連研究を紹介する．

表 4.1 トランスミッションの制御最適化の関連研究において採用されている手法．

Authors	Optimization target	Modelling form			Number of open paremeters	Number of performance metrics	Evaluation with real machine	Optimization method	Year
		Symbolic	Statistical	Numerical analysis					
Dutta, et al.	Parameter	○	○		5	1 (Jerk)	○ (Test bench)	OC, BBO	2014
Jian, et al.	Parameter	○	○		4	1 (Pressure vibration)	○ (Test bench)	RSM+BBO	2018
Roozegar, et al.	Controller	○			(3)	2 (Rot. Sppeed & acc.)		OC	2018
Mesmer, et al.	Controller	○			(1)	1 (Pressure deviation)	○	Rule base, RL	2019
Ohnogi, et al.	Parameter			○	11	1 (Acc.)		RSM+SA	2010
Vaerenbergh, et al.	Parameter			○	4	2 (Velocity & time)	○ (Test bench)	RL	2012
Proposed	Parameter			○	11	8 (Heat, rot. speed, acc., jerk & time, )	○ (Real vehicle)	BBO	2019

OC: Optimal control

RSM: Response surface methodology

SA: Sensitivity analysis

RL: Reinforcement learning

Acc. : Acceleration

Rot. speed : Rotation speed

本研究以前にもトランスミッションの制御パラメータの最適化を行った研究 [82][83][84][85][81][86] は存在する．表 4.1 は研究の設定に関する情報を整理したものであり，各研究における現象のモデル化の形態，最適化アルゴリズム，検証実験の方法を示している．表 4.1 は，2010 年以降に限りて整理を行った表であるが，その中においても記号モデル，統計モデル，数値解析モデルを利用した研究が存在する．トランスミッションの振動や圧力変動は比較的数式化がしやすく，記号モデルを利用した解析の有用性が高いため，記号モデルを用いて最適化を行う研究が行われているのだと考えられる．表 4.1 の研究の中には，実際の機械を利用して検証を行っている研究も存在するが，テストベンチの実験が多く，実際の車体を使って実験している研究は存在しない．その背景には，実機では多くの要素が運転に関わってくるため解析が難しくなることや，実験環境の構築の

難しさがあると考えられる。

また、いずれの研究においてもトランスミッションの制御パラメータを最適化しているが、モデルや最適化手法の基礎的な研究にとどまっており、目的関数の設計法や最適化結果の実際の設計への活用法について述べられている研究は存在しない。

#### 4.1.3 挙動模擬のための数値解析モデル

複数の機械要素からなる機械システムの再現には、特定の機能を持つ機械要素をブロック線図のように組み合わせてシステム全体を再現する 1D-CAE が使われることが多い。1D-CAE では、各機械要素の機能を明確にして切り分けることにより、設計者にとっては数値解析モデルを構築しやすく、計算機にとっては少ない計算負荷で計算を行うことができるという長所があるためである。一方で、1D-CAE は機械要素の構造が影響する現象は模擬することが難しい。例えば、3次元形状が引き起こす流体損失や、機械要素間での熱伝導などは模擬することが難しい。そのような性質から、1D-CAE は概念設計、3D-CAE は詳細設計という役割分担で利用されている。このような 1D-CAE も数値解析モデルの一種であるため、物理量の時系列推移を計算し、取得することができる。1D-CAE では、トランスミッションの場合の加速度のように最終的に最適化したい値だけではなく、油圧波形や複数の軸の回転数等のような中間出力値も取得できることができるため、動作の解析が容易である。本研究では、上記のような性質を持つ 1D-CAE を活用して制御パラメータの最適化を行う。具体的に利用した 1D-CAE のソフトウェアやモデルの構成は、後の 4.3 節で述べる。

#### 4.1.4 数値解析モデルを活用した電子制御トランスミッション制御パラメータの最適化

本章においては数値解析モデルを活用した電子制御トランスミッション制御パラメータの最適化を取り扱う。本節では制御パラメータの最適化に数値解析モデルを適用することの優位性、適用に際して直面する課題について述べる。

2.2.2 項で述べた通り、最適化手法は目的関数の入出力以外の情報を利用するか否かでホワイトボックス最適化 (WBO) とブラックボックス最適化 (BBO) の 2 種類に大別できる。BBO では、数値解析モデルで出力可能でさえあれば、加速度値や発生熱などの複数要素を同時に考慮可能であるため、本章に取り組みにおいては BBO を用いて最適化を行う。

2.4.1 項で述べた通り、数値解析モデルを活用した実機械システムの最適化にはいくつかの利点がある。これらの利点をトランスミッションの制御パラメータ最適化の問題に当てはめると以下ようになる。



- 人間が一般化するための手がかりを見つけることができる
  - 新しい機構の動かしかた方を見つけられる可能性がある
  - 一般化を行った結果、機構設計を改善させるための知見が得られる可能性がある
  - アルゴリズムが出力する実用的な解を実機で試す前に、数値解析モデル上で起きている現象を確認することができる
- 物理量を取り出されるため、数値解析モデルの修正が容易
  - 内部で起きている現象がわかりやすく、数値解析モデルの修正も容易
  - 目的関数の値だけではなく性能指標の値、物理量の時系列データも残っているため目的関数の改善を行いやすい

また 1D-CAE では制御パラメータと同じように、質量やバネ定数などの機械要素の物性値を変えることもできる。機械要素の物性値 (パラメータ) も含めて最適化を行えば、最適化の仕組みを特に変えることなく、機構の設計最適化問題へ拡張できる。このような、制御パラメータだけではなく、制御対象の機械の設計パラメータも同時に最適化するという考え方は近年 (2019) の強化学習の研究においても登場している [87]。文献 [87] は数値シミュレーションの中で完結した研究ではあるものの、2 足歩行ロボットの制御方策と、寸法を同列のパラメータとして扱い、ロボットの悪路踏破性能を最適化している。図 4.3 は環境ごとに寸法が最適化されたロボットの歩行の様子を示す。図 4.3 下段のロボットは上段のロボットに比べて足が長く製造コストが高くなるが、足が長い分階段や穴のような悪路を踏破することができる。

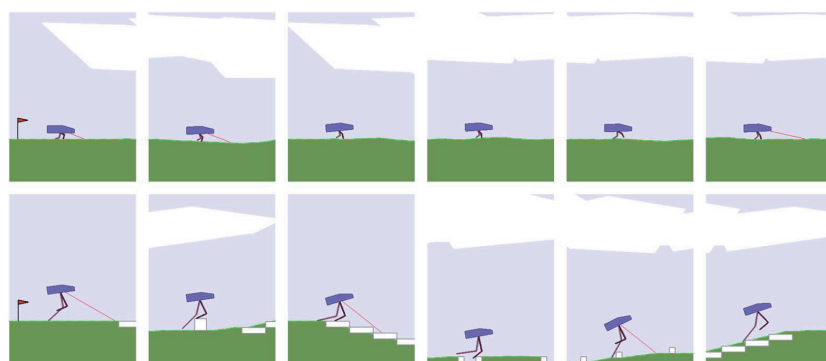


図 4.3 制御方策とロボットの寸法の同時最適化 [87].

トランスミッションは複数の機械要素が組み合わさって動作する機械であり、挙動には非線形性があると考えられる。また、トランスミッションの制御パラメータ最適化のように複数の要求が存在する場合、目的関数が図 4.4 のように多峰性を持つ可能性が高い。加えて、パラメータが固定されておらず、設計現場においてパラメータの調整の要求がある

ということは、現状のパラメータが最適ではないということがわかっているか、関数に多峰性があることを示唆していると考えられる。

目的関数が図 4.4 のように多峰性を持つ問題に対しては図 4.4 右のようにピークを把握しながら最適化を行うことで局所最適解に陥らず効率的な最適化が可能だと考えられる。強化学習の一分野の階層強化学習 (HRL; Hierarchical Reinforcement Learning) においてはこのような関数の多峰性を仮定して方策を階層化し学習効率を高めることが行われている。この多峰性の仮定は 2.2 節のノーフリーランチ定理の議論における偏りの仮定と対応する。本研究では HRL の 1 手法である HPSDE[88] を最適化向けに修正した階層型確率的最適化 (HSO; Hierarchical Stochastic Optimization) を提案して、制御パラメータ最適化に用いる。

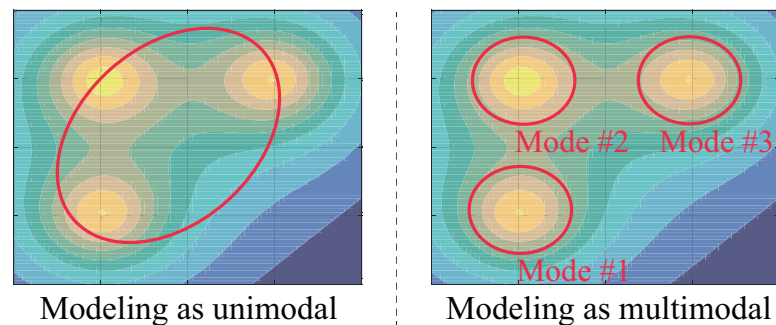


図 4.4 目的関数の効率的最適化のための多峰性の把握。

一方で、数値解析モデルを用いた最適化のアプローチには以下の二つの研究課題が存在する。

1. 複数の目標が存在しており最適化のための目的関数が定式化されていない。
2. 実機と数値解析モデルとの間に差が存在する。

課題 1 は目的関数の定式化に関するものである。先述のとおり、機械に対しては複数の要求が存在する。この中には設計現場において当たり前の暗黙知のようにになっているものもあり、要求全てを把握することが難しい。また要求は定量的だったり定性的だったり、お互いの次元が一致していなかったりと、目的関数の形への落とし込み方が自明ではない。良い設計をするには、設計現場における要求を効率よく引き出し、複数の要求を統一の尺度で扱えるような目的関数を作っていく必要がある。

課題 2 は数値解析モデルでは模擬できない実現象があるために生じる誤差に関するものである。誤差が存在すると、数値解析モデルにおける最適解と実機における最適解が異なるようになる。数値解析モデルによる最適化の効果を上げるためには、この誤差を小さくする必要がある。機械の設計の時間コストを低減するためには、誤差原因を素早く突き止

め、モデルを素早く修正できる仕組みを作ることが必要である。

本章においては、この課題1及び課題2に対し、3章において述べたメタな最適化システムの考え方を利用してアプローチする。

#### 4.1.5 本章における研究の副目的

以上の議論から、本章の電子制御トランスミッションの制御パラメータ最適化の取り組みに対して、以下のような研究の副目的を設定した。括弧内の文字は、整理のために設定した研究目的のIDを表す。

- 研究課題を解決するために手法を提案し実行する。(TM-A)
  1. トランスミッションの性能評価のための目的関数を定式化する。(TM-A-1)
  2. 実機と計算機上の数値解析モデルで同様のパラメータの出力する物理量の比較を行い誤差発生要因を明らかにする。(TM-A-2)
- 実際の機械システムのパラメータ最適化まで行い、各問題の最適化が現代の技術の組み合わせで解決が可能か否かについて、デモンストレーションを行うことにより検証する。(TM-B)
  1. 計算機上における目的関数の最適化の可否や技術的妥当性を検証する。(TM-B-1)
  2. 計算機上で最適化したパラメータを実機に適用し性能を評価する。(TM-B-2)
- 目的関数の多峰性を仮定した最適化手法である HSO を提案し、性能を評価する。(TM-C)

## 4.2 課題達成のためのアプローチ

本節では4.1節で述べた研究課題を達成するためのアプローチ(達成方針)について述べる。なお本節は、研究目的ID:TM-Aの「研究課題を解決するために手法を提案し実行する。」の提案部分である。

先の、4.1節で述べた研究課題1, 2は次のようなものであった。

1. 複数の目標が存在しており最適化のための目的関数が定式化されていない。
2. 実機と数値解析モデルとの間に差が存在する。

本章では、これらの課題へのアプローチを3章で述べたメタな最適化システムの観点から述べる。

### 4.2.1 扱う最適化問題の特性

アプローチについて述べる前に、前提条件となる、扱う最適化問題の特性について述べる。本章で扱う最適化問題において、メタな最適化システムの部品や、非メタな最適化システムの部品は以下のような特性を持つ。

- ブラックボックスな最適化アルゴリズムは数値解析モデル、目的関数、パラメータ定義域の情報の3つが与えられれば、数日の間に実用的なパラメータを求めることができる。
- 設計者は最適化されたパラメータに基づいて、探索空間に関する一般化された知見を取り出すことができる。
- 制御パラメータを設定し数値解析を行うと、目的関数のスカラ値のみならず、トランスミッションの変速動作中の各種物理量を計算することができる。
- 設計者は性能評価に関する暗黙知を知っている。
- 設計者は各種物理量の値を見て制御の良し悪しを評価できる。
- 設計者は各種物理量の値を見てモデルの挙動が意図通りかを評価できる。

本研究では、これらの特性を考慮して、TM 制御パラメータ最適化のためのメタな最適化システムを構築した。1D-CAE で数値解析が可能で、機械製品のように目的関数が自明でない問題に対しては、同様の特性が当てはまり、同様のアプローチが有効であると考えられる。

### 4.2.2 メタな最適化システムの動作

本節では、アプローチであるメタな最適化システムの動作について述べる。図4.5は非メタな最適化システムの構成を表す。メタな最適化においては、この非メタな最適化システムの構成要素である、数値解析モデルと目的関数とを洗練させながら、制御パラメータ  $x$  を最適化する。

図4.6は3章でも述べたメタな最適化システムの全体像である。これは、メタな最適化システムにおける機構1の「最適化結果の重要度に基づいた要素修正」に基づいて、非メタな最適化システムを構築・改善していくものである。図4.6において、矢印は情報の流れを表し、黒色の矢印で表されるのは要素の内部パラメータや数値解析の結果のような電子的なデータの流れである。一方、緑色の矢印は人間の持つ知識の流れであり、こちらはデータの形にするのが難しい。図4.6中の右上右下にある、青色に塗りつぶされた四角は数値解析を行うプロセスを表し、黄色に塗りつぶされた四角は実機実験を行うプロセスを

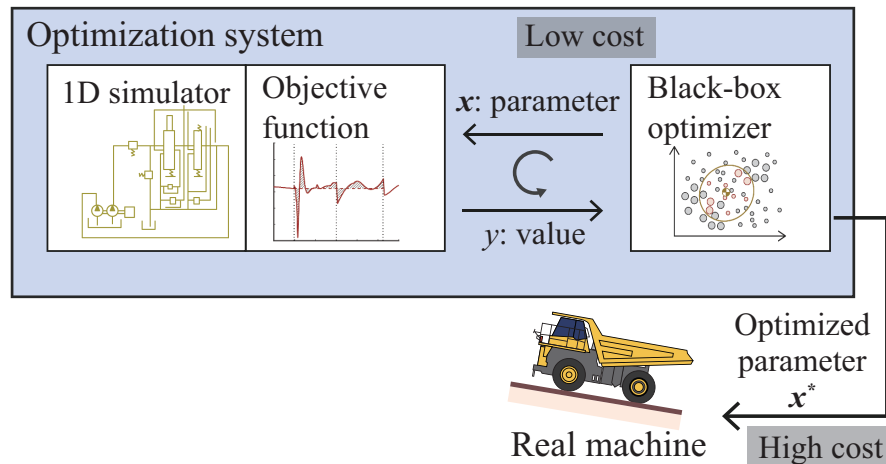


図 4.5 数値解析モデルを活用したトランスミッション制御パラメータ最適化システム。

表す。また、緑色に塗りつぶされた四角は設計者（人間）が関与したプロセスを表している。図 4.6 中の薄黄色、薄赤色の枠に示すように、メタな最適化システムにおいては、設計者が関与するプロセスを経て、モデルの修正や、目的関数の修正が行われる。

ここからは、メタな最適化システムのそれぞれのプロセスについて内容と効果を説明する。図 4.6 のように、メタな最適化は 10 個のプロセスから構成される。

■1. 実験と観察 このプロセスにおいては、対象となる機械の数値解析モデルを構築するために、機械の挙動を観察する。本章で扱うトランスミッションのような機械の設計においては、挙動が既に知られている機構を組み合わせる設計が行われることが多く、そのような場合はこのプロセスを省くことができる。逆に、設計において新しい機構を取り入れた場合は、このステップを行う必要がある。

■2. モデル化 このプロセスにおいては、プロセス 1 の観察結果や、機械の設計資料に基づき、機械の挙動を数値解析モデルに落とし込む。初回のモデル化においては、数種類のパラメータにおいて数値解析モデルを回し挙動を評価する。具体的には、数値解析モデルが発散しないか、実用的な時間で計算が可能かなどの観点からモデルを評価する。発散する場合や計算時間がかかる場合は、その原因を突き止めて、原因を解消したり、別の手法を試すことで好ましいモデルへと修正する。また、モデルのバグや誤差の知識を基に、人間（設計者）がモデルを修正する。モデルの修正の際にはモデルの中間出力と実際のデータとの一致を確認することや、ユニットテストなどの、ホワイトボックス的なやり方で修正を行う。ホワイトボックス的な修正は、1 試行から複数の知見を取り出すことができるため、修正効率が高い。

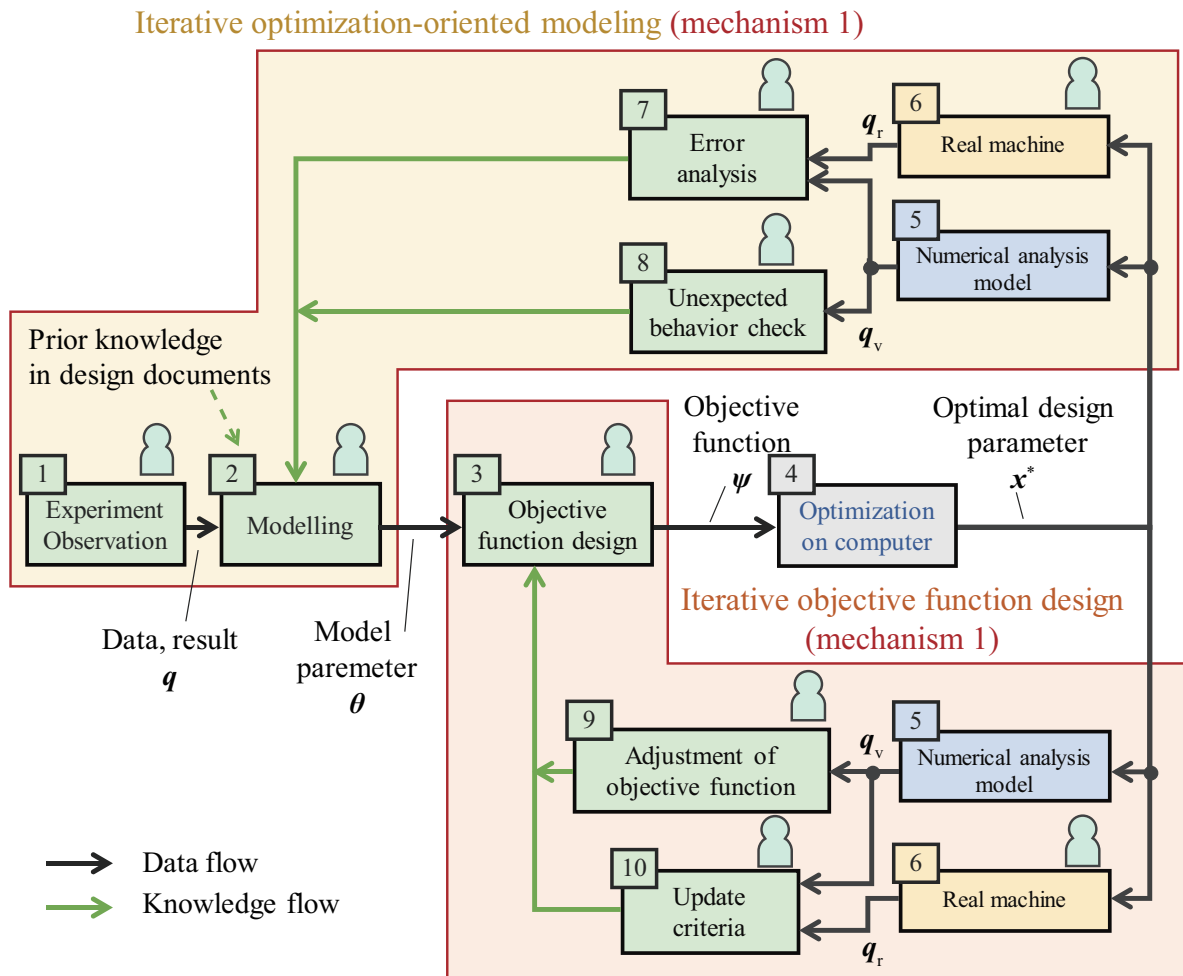


図 4.6 TM 制御のメタ最適化システム (番号入り).

■3. 目的関数の作成 このプロセスにおいては、機械の挙動を評価するための評価指標・目的関数を設計・実装する。数値解析モデルの出力する物理量を引数として、複数の評価指標が計算され、複数の評価指標をもとにスカラ値である目的関数の出力が計算される。初回の目的関数の作成においては、数種類のパラメータにおいて数値解析モデルを回し挙動を計算し、評価指標・目的関数の妥当性を評価する。設計者は、数値解析モデルで模擬された機械の動作と目的関数の値の示す好ましさが一致するかどうかを確認することで、目的関数の妥当性を評価し、妥当でなかった場合には目的関数の設計を修正する。本章の取り組みにおいては、各評価指標の値を重みづけして目的関数の値としている。また、各評価指標の目的関数に対する影響度合いを同様にするため、及び、無次元化するために、評価指標の変化幅を概算し、変化幅により評価指標を正規化する。この各評価指標の重みづけは、後のプロセスで得られた情報を基に調整される。

■4. 計算機上での最適化 このプロセスにおいては、数値解析モデルを用いた最適化を行う。このプロセスでは人間の介在がなく、数値解析モデルと目的関数を利用して、計算機上で自動的に最適化が行われる。計算機上での最適化は、人間が介在せず、実機の実験も行わないため高速に実行することができる。また、この最適化においてはブラックボックス最適化手法を利用してブラックボックス的に最適化が行われる。

■5. 最適化されたパラメータの数値解析モデルにおける評価 このプロセスにおいては、最適化されたパラメータを数値解析モデルを利用して評価する。評価の結果としては、目的関数の値だけではなく、中間出力も一緒に出力される。人間による制御パラメータの最適化が難しい問題においては、最適化されたパラメータが、プロセス2や3で人間の設計者により試された数種類のパラメータから大きく異なっていることが期待される。大きく異なっているため、プロセス2と3で試した数種類のパラメータでは発生しなかったような数値解析モデルの挙動の発生が期待される。また、試された数種類のパラメータと比べ、目的関数の値が大きくなることが期待される。

■6. 実機における評価 このプロセスにおいては、最適化されたパラメータを実機を利用して評価する。実機を利用するため、プロセス5に比べて精度は高くなるが、1試行あたりのコストが高くなる。プロセス5と同じく、目的関数の値だけではなく、中間出力も一緒に出力させることができる。ただし、プロセス5の場合とは異なり、中間出力を出力させるためには、新たにセンサを取り付ける必要性から、追加でコストがかかることもある。

■7. 誤差の解析 このプロセスにおいては、プロセス5と6で得られた数値解析モデルと実機の出力値を比較し、誤差原因に関する知見を取り出す。この知見はプロセス2に戻されモデルの修正に利用される。比較を行う出力値(物理量)の種類を増やすことにより、誤差原因の知見を取り出しやすくなる。

■8. 意図せぬ挙動の確認 このプロセスにおいては、プロセス5の数値解析モデルによるパラメータ評価結果を用いて、設計者が意図しないような数値解析モデルの挙動の有無を確認する。プロセス5においても述べたが、最適化されたパラメータは初回モデル作成時に試した数種類のパラメータと大きく異なっているため、初回の作成時には発生しなかったような意図せぬ挙動が発生することもあると考えられる。意図せぬ挙動が発生した場合、その原因を解明して、解明結果をモデル修正のための知見としてプロセス2に戻す。

■9. 目的関数の適合 このプロセスにおいては、プロセス5の数値解析モデルによるパラメータ評価結果を用いて、目的関数の設計を変更するための知見を取り出す。設計変更の知見は、評価指標の重みづけに関するもの、評価指標に関するものの2つに分けられる。重みづけと評価指標の組み合わせを複数試し、各組み合わせが設計現場の要求をどの



程度反映しているのかを，プロセス5の結果を利用して確かめる．各組み合わせの比較により，設計現場の要求を反映した組み合わせを選抜していく．

評価指標は重みに比べて自由度が高く，設計難易度が高い．一方で，評価指標は機械の性能を評価するものであればよく，機械の機能を考える際にある程度明文化されている．評価指標に関しては，この明文化されているものを利用して設計を進める．

重みづけは自由度が低く，調整（設計）難易度は低い．重みは各評価指標のバランスを決めるものであるため，評価指標の設計が定まった後に調整を行う．また，評価指標の組み合わせによっては，どんな重みづけをしたとしても，たどり着けないようなパラメータ $x$ （機械の挙動を決めるパラメータ）が存在する．重みの調整だけでは理想としている挙動にたどり着けない場合には，評価指標の設計に戻る必要がある．

評価指標の重みの調整や評価指標の実装は人間が行う必要があるが，膨大な探索空間の中で制御パラメータを最適化する繰り返し作業に比べると，方針が立てやすく，実行の負荷が小さいと考えられる．例えば，最適化されたパラメータ $x$ が，目的関数の値の上では良いが，実用するには発生熱が大きすぎるということが分かった場合，発生熱を評価する指標の重みを大きくするといった直感的な調整すれば済む．

■10. 新基準の作成 このプロセスにおいては，プロセス6で得られた実機の評価結果を用いて，目的関数において考慮できていない評価基準を見つけ出し，目的関数の設計に反映させる．プロセス9の評価指標の設計と役割がかぶる部分もあるが，プロセス10は実機の結果を用いているという部分がプロセス9と異なる．実機による評価には時間コストがかかるため，実機を使わない部分をプロセス9，使い部分をプロセス10として分離した．それにより，メタ最適化の序盤ではプロセス9を行って目的関数を低コストで修正するといった戦略を立てやすくなることを狙っている．

■システムの全体的な機能と動作 図4.6に示すように，このメタ最適化システムには，プロセス5と6の評価結果を受けて，モデルや目的関数を修正するというフィードバックループがある．このフィードバックは3章で述べた機構1の重点的な要素修正における修正に相当する．重点的に要素修正をすることにより，性能に大きく関係する部分の修正が優先的に進み効率的な要素修正が実現される．その反例として，数値解析モデルの改善を考えた際，数値解析モデルの誤差原因をつぶさに分析し，モデルの精度を上げていくことも考えられる．しかし，広いパラメータ探索空間全域で一致するように精度を向上させることはコストの適正な配分とは言えない．

メタな最適化の序盤においては，プロセス6の実機による評価にともなうフィードバックは行わず，主にプロセス5の数値解析モデルによる評価にともなうフィードバックを行う．これは，プロセス6の実機実験には実施コストと安全上のリスクがとれない，なおかつプロセス5の数値解析モデルによる評価からも得られるものが多いためである．



プロセス 2 や 3 の目的関数の改善や数値解析モデルの改善は、人の手で行う必要があり時間を要するが、プロセス 4 の計算機上での最適化やプロセス 6 の実機による評価にも同様に時間がかかるため、システムを回して行くうえで律速にはなりづらい。

以降は、トランスミッションの制御パラメータ最適化問題に対し、本節で紹介したメタな最適化を適用した結果について述べる。続く 4.3 節において数値解析モデルと目的関数の完成形について述べ、4.4 節において計算機上での最適化と数値解析モデルにおけるパラメータの評価について述べ、4.5 節において実機におけるパラメータの評価について述べる。そして、4.6 節においては、数値解析モデルと目的関数を完成形に至らせるまでのメタな最適化システムの動作について述べる。

### 4.3 数値解析モデル及び目的関数の構成

本節では現象の模擬のために利用した数値解析モデル及び、機械システムの性能の評価するための目的関数の詳細について述べる。また、追実験を容易にするため、数値解析モデルの内部の数値シミュレータや目的関数の具体的な実装方法についても述べる。本節においては、研究目標 ID:TM-A-1 の「トランスミッションの性能評価のための目的関数を定式化し形を定める。」の実践結果も述べる。

#### 4.3.1 現象模擬のための数値解析モデル

本項では電子制御トランスミッションの動作を模擬するため数値シミュレータの機能について述べる。

本研究では 1D-CAE のソフトウェアとして Siemens 社の Amesim 17 を利用した。Amesim は機械要素のシミュレーションだけではなく、電気要素のシミュレーションも可能なメカトロニクスシミュレータである。また、様々な CAE, CAD, 他の計算環境との連携が可能である。本研究のように自律的に動作する最適化システムを構築しようとした場合、数値シミュレータには最適化計算を行うためのプログラミング言語との連携が可能であることが強く求められる。本研究で利用する Amesim も Mathworks 社の Simulink と連携することができる。このような連携機能が存在しない場合は、RPA (Robotic Process Automation) などの、人間のためのインタフェースを自律的なプログラムにより扱えるようにする仕組みを導入し、自律的な最適化システムを構築する必要がある。

Amesim においては、バネ、ダンパ等の機械要素を画面上に配置し、それらを結合させることで機械全体を表現する。各機械要素はそれぞれ動的挙動が定義されており、ODE45 等の常微分方程式のソルバを利用して時間発展の結果が計算される。

### 4.3.2 性能評価のための目的関数

本項では数値解析モデルにより模擬されたトランスミッションの動作の評価方法について述べる．まずはじめに目的関数の設計過程について述べ，次に具体的に最適化に用いた最終的な目的関数について述べる．本章で行う最適化において目的関数は制御パラメータ  $\mathbf{x}$  を入力として，性能指標  $y$  を出力する． $\mathbf{x}$  の定義域はトランスミッションが暴走しないような範囲で設計者が決定した． $\mathbf{x}$  の各要素の定義域は独立であり，各要素に上限値と下限値が存在する． $\mathbf{x}$  は図 4.7 に示すようなクラッチへ印加する油圧のパターンを決定する，要素数 11 のベクトルである．

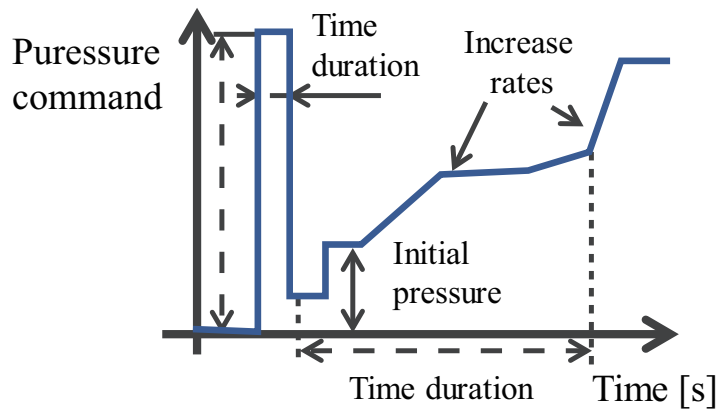


図 4.7 制御パラメータにより定められるクラッチ油圧波形の例.

#### 目的関数の具体的実装

本研究では BBO を用いるため，数値解析モデルで出力可能である限りどのような値でも性能評価に用いることができる．また，時間制限のように OK/NG の 2 値で表される制約を課すことも可能である．加速度と熱負荷のように別々の次元を持ち値のオーダーも異なる指標が登場するが，あらかじめ求めた正規化定数を用いて無次元化及び正規化を行うことで統一的に扱うことが可能となる．本研究では加速性を維持したまま，変速ショックに関係する加速度のピークを低減するように目的関数を構成した．また最適化の制約として変速時間等の制約を課した．

$$g_k(\mathbf{x}) = \exp \left\{ - \left( \sum_{i=1}^8 w_i m_i(\mathbf{x}, k) + \sum_{i=1}^3 c_i(\mathbf{x}, k) \right) \right\} \quad (4.1)$$

ダンプトラックの走行条件を固定した，単一運転条件における制御パラメータを評価する目的関数  $g_k(\mathbf{x})$  は式 (4.1) のように定義される． $g_k$  は性能指標  $m_i$  と制約条件  $c_i$  から構

成される．また各性能指標の重要度は重みベクトル  $\mathbf{w}$  によって調整する．各指標  $m_i$  は、加速度の大きさや発生する熱量を表す値  $v_i$  から計算され、 $v_i$  は数値シミュレータが出力する物理量  $\mathbf{q}$  から計算される．本研究においては式 (4.1) にあるように、8つの指標、3つの制約を設定した．

各性能指標は、それぞれの最適化のされやすさを統制するために正規化されている．正規化された性能指標  $m_i$  と制約条件  $c_i$  はそれぞれ式 4.2、式 4.3 のように定義される．

$$m_i = \frac{v_{m,i} - v_{\min,i}}{v_{\max,i} - v_{\min,i}} \quad (4.2)$$

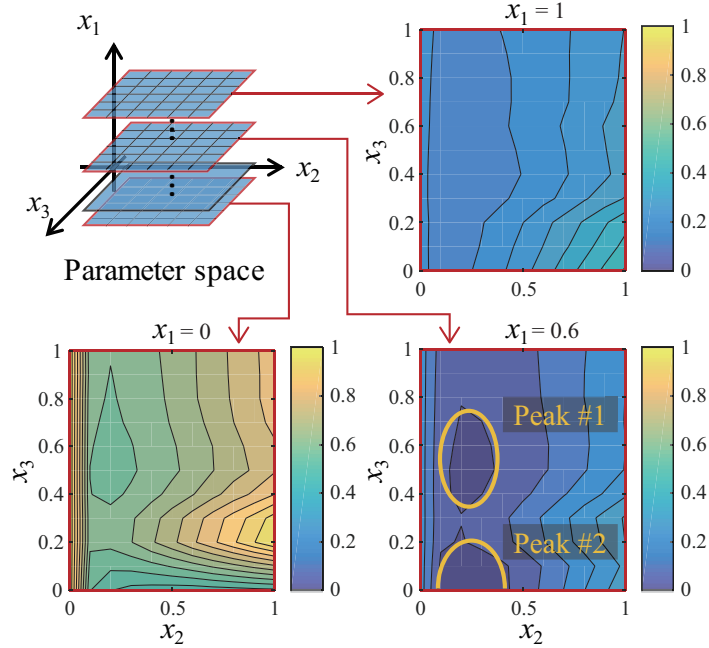
$$c_i = \begin{cases} 0 & (v_{c,i} \leq v_{\lim,i}) \\ C & (v_{c,i} > v_{\lim,i}) \end{cases} \quad (4.3)$$

ここで  $v_{m,i}$  は正規化前の指標であり、単位を持つ物理量をあらわす．また、 $v_{\min,i}$  と  $v_{\max,i}$  は各指標に対応する正規化定数であり、グリッドサーチにより求めた．グリッドサーチは本研究以前に設定されていたパラメータである現行パラメータの 11 次元のうち  $x_1$ ,  $x_2$ ,  $x_3$  の 3 次元を変数として行った．現行パラメータは人間の設計者によって調整されたものであり、表 4.2 上行のような値をとる． $v_{\lim,i}$  は制約の閾値であり、設計者が設定する定数である．

以上に挙げた性能指標はすべて、小さいほど良い性能となる．そのため新規に性能指標を追加する際には、必要に応じて負号を追加する必要がある．また、制約を満たさなかった場合は、十分大きな数  $C$  が  $c_i$  に代入され目的関数の値がほぼ 0 になる．本論文において  $C$  は 1,000 と設定した．最終的に目的関数は式 (4.1) のように、正規化された評価指標と  $m_i$  と制約条件  $c_i$  の重み付け和を用いて定義される．目的関数を指数関数の形式にした理由は、性能が極端に悪い時と制約を満たさなかった場合とを同様に扱うためである．

図 4.8 は 3 次元のパラメータ空間における指標の可視化結果である．11 次元のパラメータベクトルのうちの 3 次元を変動させ可視化を行った．図 4.8 左上のように格子点を設定し、各格子点が表すパラメータごとに評価指標を算出した．図 4.8 中のコンタ図は指標  $m_5$  の値を示している．それぞれのコンタ図では  $x_1$  が固定され、 $x_2$  と  $x_3$  が変動する．11 次元より少ない、3 次元の部分空間での可視化であるが、 $x_1, x_2 = 0.6, 0.2$  の近辺に 2 つの極小点が存在する．このことから、11 次元のパラメータ空間における目的関数は多峰性を持つことが示唆される．

次節以降の実験においては  $\mathbf{w}_1$  から  $\mathbf{w}_5$  の 5 種類の重みベクトルを利用している． $\mathbf{w}_1$  は均一な重み付け条件であり、すべての指標の重みは等しくなっている． $\mathbf{w}_2$  は加速度の目標値からのずれを低減することを優先した重み付け条件であり、 $\mathbf{w}_3$  は発生熱を低減することを優先した重みづけ条件である． $\mathbf{w}_4$  と  $\mathbf{w}_5$  は、後述する複数条件の最適化のための重み付け条件であり、重みの値は  $\mathbf{w}_1$  から  $\mathbf{w}_3$  を使った検討の結果を利用して決定した．

図 4.8 評価指標  $v_{m,5}$  の 3 次元部分空間における可視化.

#### 複数運転条件を評価する目的関数

実際のダンプトラックの走行中には道路勾配やスロットル開度などの運転条件が変動する．本研究では複数の運転条件で性能が良くなるパラメータを見つけるための取り組みも実施した．その際には，いくつかの走行条件における評価を統合した目的関数を利用し，幅広い条件の下で性能が良くなるようなパラメータの獲得を目指した．以降はその目的関数の具体的実装について述べる．

単一運転条件を評価するための目的関数  $g_k(\mathbf{x})$  を複数の運転条件に対して計算し，総合的に評価することにより，多くの運転条件における性能を反映した値を計算できると考えられる．そして，総合的に評価する方法として，目的関数  $g_k(\mathbf{x})$  の平均値と目的関数の最低値の 2 パターンを試行した．平均値と最低値，それぞれの場合の複数運転条件を考慮した目的関数は式 (4.4)，式 (4.5) のようになる．

$$f(\mathbf{x}) = \frac{1}{N_{\text{cond}}} \sum_{1 \leq k \leq N_{\text{cond}}} g_k(\mathbf{x}) \quad (4.4)$$

$$f(\mathbf{x}) = \min_{1 \leq k \leq N_{\text{cond}}} g_k(\mathbf{x}) \quad (4.5)$$

平均値と最小値の 2 パターンで最適化試行を行った際の学習曲線を図 4.9 に示す．この試行において，運転条件の数  $N_{\text{cond}}$  は 3 であり，それらの運転条件は路面の勾配が異なる．図 4.9(a) は平均値を使った目的関数，図 4.9(b-1) は最小値を使った目的関数の学習曲

線である．図 4.9(a) においては平均値を構成する 3 運転条件の目的関数値を 3 で割ったものを積み上げグラフで表示している．図 4.9(b-1) においては最小値を示した 1 運転条件の目的関数値を表示している．一方，図 4.9(b-2) においては，3 つの運転条件の目的関数値の分布を可視化するため，図 4.9(a) と同じ描画を行っている．図 4.9(b-2) のように，式 (4.5) の最小値による複数条件の合算方法の場合には，最低値の底上げをするように最適化が進行する．他の運転条件に比べ目的関数の値が悪いような運転条件が発生しづらく，多くの運転条件で平均的に良い性能を示すパラメータを見つけられると期待されるため，式 (4.5) の最小値による合算方法を採用した．

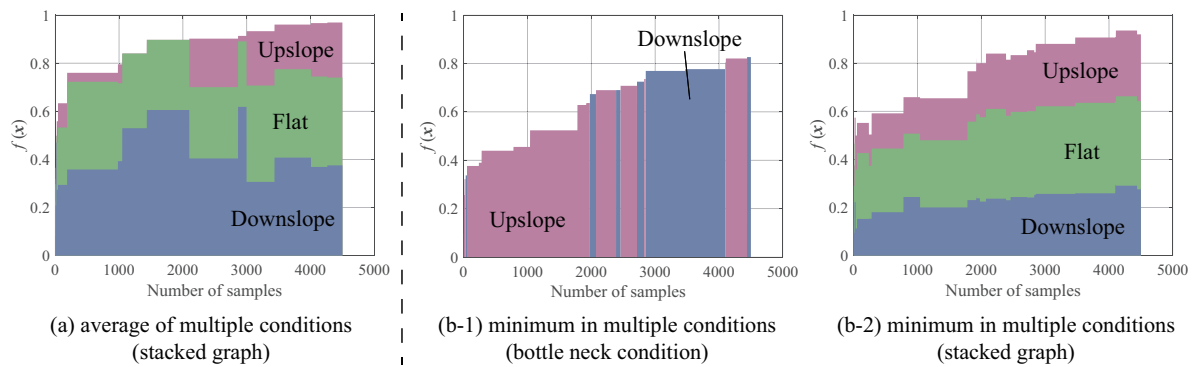


図 4.9 平均値と最小値の 2 パターンでの目的関数の学習曲線．

#### 設計した目的関数の可視化

本小項では，目的関数の部分空間における可視化結果を用いて，設計した目的関数の性質について述べる．図 4.10 は単一運転条件を評価する目的関数  $g_k(\mathbf{x})$ ，図 4.11 は複数運転条件を評価する  $f(\mathbf{x})$  の部分空間における可視化結果である．図 4.10，図 4.11 とともに目的関数のコンタ図であるため，値が大きいほど良い性能であることを示す．

図 4.10 では  $\mathbf{w}_1$  と  $\mathbf{w}_2$  の二種類の重みでの目的関数を示している．上段が  $\mathbf{w}_1$ ，下段が  $\mathbf{w}_2$  に対応する．図のそれぞれの列は変動させるパラメータの要素が異なり，左の図から順に  $(x_2, x_3)$ ， $(x_7, x_{10})$ ， $(x_9, x_{11})$  が変動している．変動の基準は図 4.8 のときと同じく，現行パラメータである．まず，コンタ図の観察から目的関数には非線形性があることがわかる．図 4.10 左下の  $\mathbf{w}_2$  の  $(x_2, x_3)$  を変動させたコンタ図では，目的関数は  $x_2$  軸方向に単調減少となっている．この部分空間では  $x_2$  軸方向に単調減少になるが，他のパラメータの要素が変わった際には必ずしも単調増加になるとは限らない．図 4.10 上段の  $\mathbf{w}_1$  と下段の  $\mathbf{w}_2$  は似た関数形状になるが，明確に異なっている部分もあり，上段左では  $x_2$  軸に対して単調減少にはならない．一方，図 4.10 中央の  $(x_7, x_{10})$  を変動させたコンタ図では，単調増加や単調減少の様子は見られず，中央付近に値が高い部分が存在する．最も黄色く目的関数の値が大きい部分が別の位置に複数存在することから，この部分空間に

おいては多峰性があることがわかる。

図 4.11 は数運転条件を評価する  $f(\mathbf{x})$  の部分空間における可視化結果である。図 4.11 の各列は異なる  $x_1$  に対応しており、それぞれのグラフでは  $(x_2, x_3)$  が変動している。図において、目的関数  $f(\mathbf{x})$  は式 (4.5) の最小値を使った合算方法のものであり、指標の重みは  $w_4$  である。この設定は次節における複数条件を考慮した最適化の時の設定と同一である。図 4.11 の左グラフ内の赤丸は現行のパラメータを表す。中央のグラフを見ると、 $x_1 = 0.6$  の部分空間において、多峰性を持つことがわかる。また、今回可視化を行った部分空間では最高値が 0.04 程度となっているが、図 4.19 に示す複数条件での最適化では、本可視化と同じ設定の目的関数を利用しているが、0.04 より大きな 0.18 という目的関数値を出力するパラメータが見つかった。

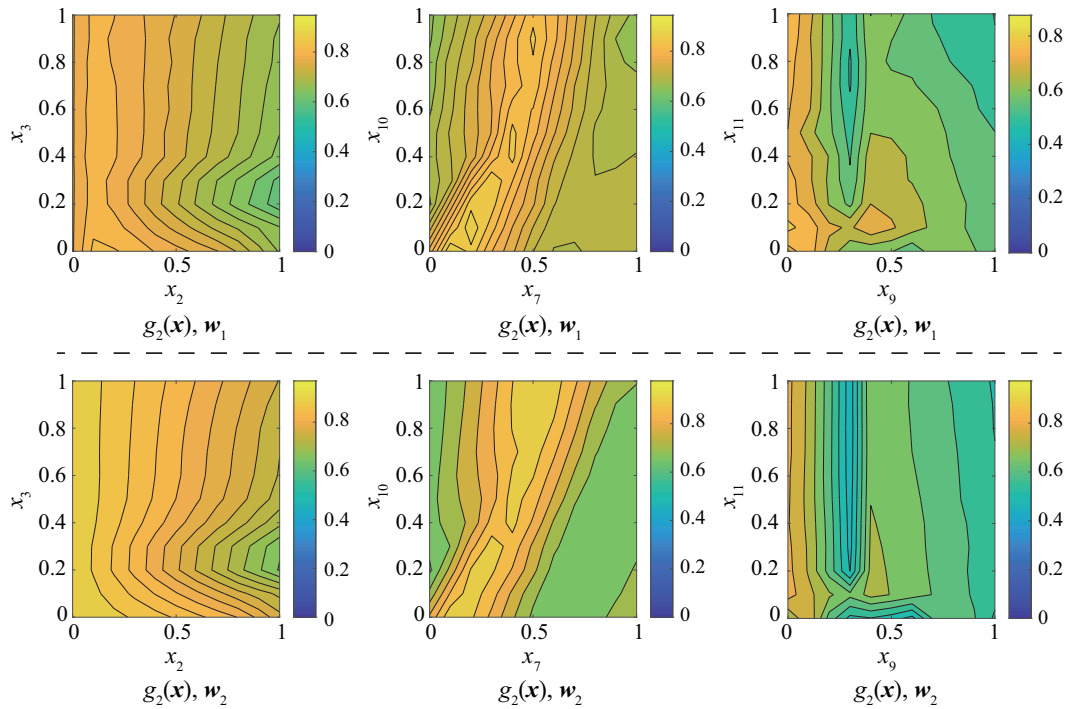
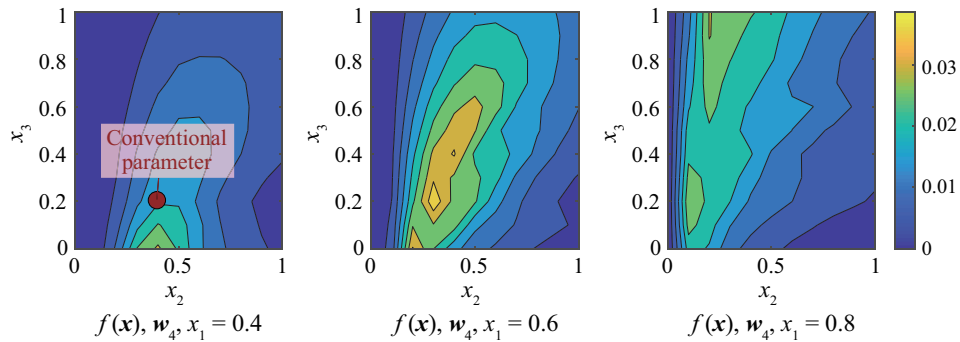


図 4.10 単一運転条件を評価する目的関数  $g_k(\mathbf{x})$ .

### 4.3.3 ソフトウェア上の実装方法

本項では計算機上での最適化システムのソフトウェア実装について述べる。Siemens 社製 Amesim と Mathworks 社製 Matlab の 2 つのソフトウェアを用いた。数値シミュレータによる挙動の模擬は Amesim が行い、目的関数  $g_k(\mathbf{x})$ ,  $f(\mathbf{x})$  や最適化のための計算は Matlab が行う。

Amesim は Matlab に付属しているツールである Simulink を介して Matlab と連携する

図 4.11 複数運転条件を評価する目的関数  $f(\mathbf{x})$ .

ことができ、Matlab 側から Amesim 上での数値シミュレーションの実行、モデルパラメータの設定、物理量の時系列データの取得が可能である。

目的関数はパラメータを受け取り第一戻り値としてスカラ値を返すという一般的な目的関数の形式になっている。また、目的関数には第二戻り値、第三戻り値が存在し、第二戻り値は実行状態を表すログメッセージ、第三戻り値は  $m_1 \sim m_8$  で表される各評価指標を格納した数値ベクトルである。目的関数として、単一運転条件を考慮したもの  $g_k(\mathbf{x})$  と複数運転条件  $f(\mathbf{x})$  を考慮したものの両者を選ぶように、目的関数を生成する関数も実装した。最適化アルゴリズムは、ハイパーパラメータと目的関数を受け取り、内部で目的関数から複数回サンプリングを行い、目的関数の最適化を行う。以上のようなインタフェースを採用することにより、目的関数をトイプロBLEMに変更したり、最適化手法を変更したりすることが容易となる。

数値シミュレーションと目的関数の計算は一台のワークステーション (Intel Core i9-7900X @3.3 GHz) を用いて行った。1 つの運転条件における 1 パラメータの目的関数の計算には 5 秒程度を要する。また、Amesim と Matlab の間のデータ受け渡しにかかる時間は数値シミュレーションに比べて極めて小さい。運転条件が固定の場合、1 日の間に約 17,000 サンプルの計算が可能である。パラメータの次元数は 11 次元であるため、グリッドサーチの場合、各々の次元が 5 水準であったとしても、48,828,125 通りの組み合わせが考えられ、全通りの計算には約 2,800 日の時間を要する。

#### 4.4 計算機上の最適化実験

本節では計算機上で行った数値解析モデルをベースとした最適化の実験及び結果について述べる。なお本節以降のグラフにおいては、製品ノウハウの機密保持のために、議論に差し支えない範囲で縦軸及び横軸の具体的数値を省略することがある。省略の際、図内での比較を行えるように、同一の図の中のグラフでは横軸の単位及び値の間隔は同一として



いる．本節の計算機上での最適化実験の目的は以下の通りである．

- 4.3.2 項で定義した ECT の目的関数の最適化が計算機上で可能であるか確かめること．(ID:TM-B-1 の検証)
- 実機実験で利用するためのパラメータを選定すること．(ID:TM-B-2 の準備)
- 著者らが提案した階層型確率的最適化アルゴリズムの性能を検証すること．(ID:TM-C の検証)
- 計算機上の最適化結果を踏まえ ECT の目的関数を修正すること．(ID:TM-A-1 の実践)

#### 4.4.1 最適化手法

本項では 4.3.2 項で定義した目的関数の最適化に用いる最適化手法について述べる．最適化においては，目的関数が式の形で得られる問題では無いため，WBO ではなく BBO の手法を用いている．

目的関数が図 4.4 のように多峰性を持つ問題に対しては図 4.4 右のようにそれぞれのピークを把握しながら最適化を行うことで局所最適解に陥らず効率的な最適化が可能だと考えられる．著者は文献 [89] において階層強化学習手法の一つである HPSDE を最適化用に修正した階層型確率的最適化 (HSO; Hierarchical Stochastic Optimization) を提案した．HSO は最適化におけるサンプル選択を行う方策  $\pi(\mathbf{x})$  を階層的にすることにより，複雑な方策を学習することを狙っている．また，HSO はその名前の通り，内部で乱数を利用するような確率的な手法であり，試行ごとに最適化結果が変化する手法である．HSO では式 (4.6) で示す期待報酬  $J(\pi)$  を最大化することを目標とする．ここで， $R(\mathbf{x})$  は報酬関数であり，本論文における目的関数と一対一で対応する．

$$J(\pi) = \int \pi(\mathbf{x})R(\mathbf{x})d\mathbf{x} \quad (4.6)$$

$$\pi(\mathbf{x}) = \sum_{o \in O} \pi(o)\pi(\mathbf{x}|o) \quad (4.7)$$

式 (4.6) に対し，式 (4.7) のように表現される階層型方策を導入する．式 (4.7) における， $\pi(o)$  は上位方策 (ゲート方策とも呼ぶ)， $\pi(\mathbf{x}|o)$  は下位方策 (オプション方策とも呼ぶ) である．式 (4.7) において， $o$  はオプションの番号であり，下位方策を表す自然数が入る． $O$  はオプション全体の集合である．複数の下位方策が目的関数の複数の峰に対応しており，1 つの上位方策はサンプリングに利用するための下位方策を選択する．HSO は式 (4.8) で表されるような期待報酬  $J(\pi)$  を最大化することを目標としながらパラメータの最適化を行う．



$$J(\pi) = \sum_{o \in O} \int \pi(o) \pi(x|o) R(x) dx \quad (4.8)$$

また HSO において下位の方策  $o$  の数は VBEM (Variational Bayesian Expectation Maximization) により自動的に調整され、目的関数の多峰性の度合いに柔軟に対応する。

HSO は図 A.5 のように最適化を進めていく。HSO の動作フローは以下のようになっている。

1. 定義域全体から均一分布によりサンプリングし、初期値依存性を減らす
2. 目的関数の値に応じてサンプルに重みづけを行う
3. 重みづけを利用して VBEM によりクラスタリングを行う。その際のクラスタ数 (オプション数) は自動調整される。
4. 上位方策が下位方策 (オプション方策) を選択して、選択された下位方策がサンプルを取得する。
5. サンプルが一定数取得されたら、下位方策を更新する。
6. 下位方策が一定回数更新されたら、3. に戻りクラスタリングを行う。

HSO の詳細構成や基本的な性能検証については本論文後半の補足 A を参照されたい。

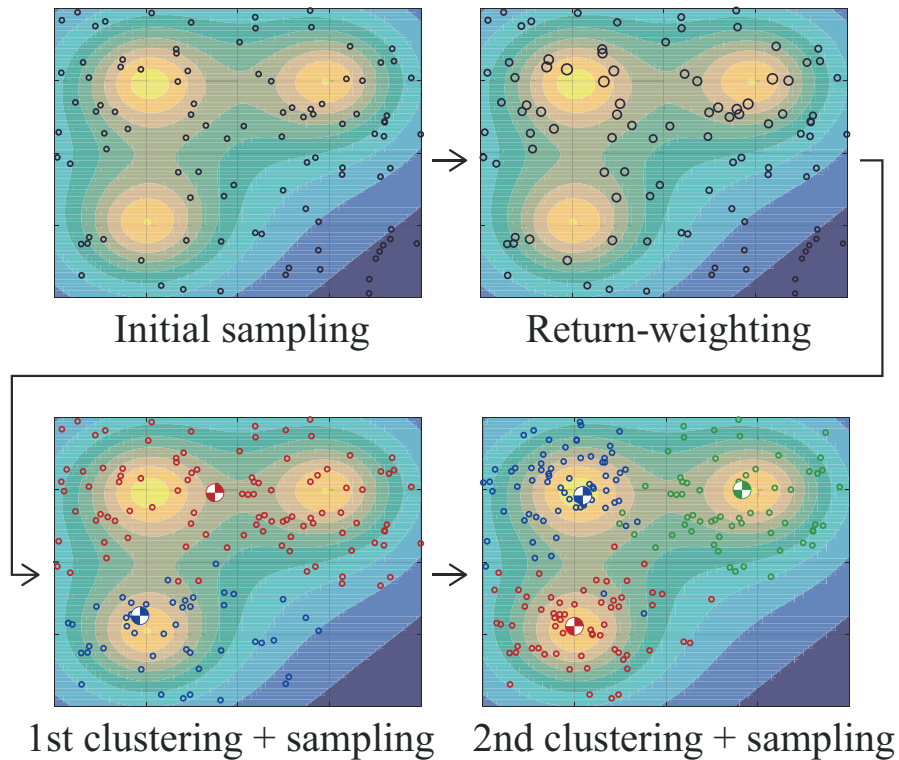


図 4.12 多峰性を持つ 2 次元トイプロブレムにおける HSO の最適化過程。

本節の実験では HSO の最適化性能を評価するために、BBO 手法である CEM (Cross-Entropy Method [58]) と CMA-ES (Covariance Matrix Adaptation Evolution Strategy [59]) を利用した。これらの2手法はシンプルなアルゴリズムであり、最適化手法や強化学習の研究において性能比較に用いられることが多い [90]。

#### 4.4.2 実験設定

本項では計算機上での最適化の実験設定について説明する。

##### 目的関数

最適化の目的関数としては 4.3.2 項の式 (4.1) 及び式 (4.5) の  $g_k(\mathbf{x})$ ,  $f(\mathbf{x})$  を利用する。最適化の目標は 11 次元のパラメータベクトル  $\mathbf{x}$  の最適化である。

##### ハイパーパラメータ

HSO のサンプリングには 3 種類のハイパーパラメータが存在する。

- $N_{\text{init}}$  : 初期の均一分布によるサンプリングのサンプル数
- $N_{\text{update}}$  : クラスタリング間の下位方策更新の回数
- $N_{\text{sample}}$  : 下位方策更新間のサンプル数

HSO のハイパーパラメータ設定を以下に示す。初期の均一分布によるサンプリングのサンプル数  $N_{\text{init}} = 1500$ 、クラスタリング間の下位方策更新の回数  $N_{\text{update}} = 6$ 、下位方策更新間のサンプル数  $N_{\text{sample}} = 100$  とした。 $N_{\text{init}}$  は、1500 より大きい場合では数では性能に差が出なかったため、1500 と設定した。 $N_{\text{sample}}$  と  $N_{\text{update}}$  は決め打ちした値である。両方の値が小さくなればなるほど、HSO は貪欲になり、局所最適に陥りやすくなると考えられる。

CEM では事前実験により良い性能を示したハイパーパラメータを利用した。CEM においても局所最適に陥ることを防ぐため、最適化の初めに均一分布からのサンプリングを行う。ハイパーパラメータは以下のように設定した。初期の均一分布によるサンプリングのサンプル数  $N_{\text{init}} = 1500$ 。方策更新間のサンプル数  $N_{\text{sample}} = 300$ 。方策更新時に残すサンプル数  $N_{\text{elite}} = 100$ 。

CMA-ES においては文献 [91] で推奨されているハイパーパラメータ設定を利用した。

### 4.4.3 単一運転条件での最適化実験

本項では，単一条件を評価する目的関数  $g_k(x)$  を利用した最適化実験の結果について述べる．実験は 2 つに分かれている．

#### 単一運転条件における最適化可能性の評価実験

本小項では，最適化手法の性能評価実験について述べる．本小項における実験においては，評価指標の重み付けを  $w_1$  の均一重みに固定している．

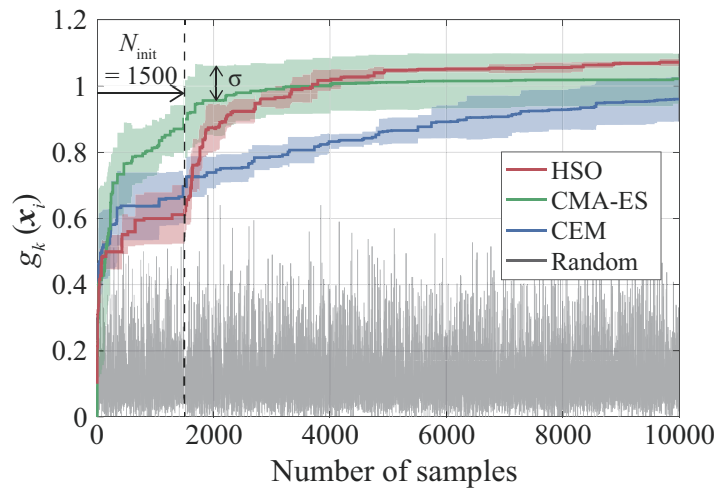


図 4.13 単一運転条件下における最適化の学習曲線．

実験ではまず，最適化手法の目的関数に対する最適化性能を比較した．図 4.13 のグラフは，最適化の学習曲線を示す．グラフにおいて実線は各々の手法の 5 試行の平均値を示し，半透明の領域は 5 試行の標準偏差を示す．HSO では他の 2 つ手法に比べて標準偏差の幅が小さく，安定した最適化性能を持っていることがわかる．HSO と CEM は，最適化の開始時に初期サンプリング方策を使用して定義域全体でサンプリングを行うため，初期の目的関数の値は CMA-ES に比べて小さい．約 4,000 サンプル付近で HSO の性能平均値が CMA-ES の平均値を超える．

図 4.14 は，各手法における 10,000 サンプル収集後の目的関数の値を表す棒グラフである．10,000 サンプル収集後の目的関数の値を比較すると，提案する HSO は CEM より有意 ( $p < 0.05$ ) に高い性能を示すが，CMA-ES に対しては有意に高い性能は示さなかった．また 3 つの最適化手法はすべて，ランダムなサンプリングよりも高い性能を示す．さらに，3 手法はすべて，人間の設計者が調整していた現行パラメータよりも高い性能を発揮するパラメータを発見している．

図 4.15 は，3 つの最適化手法によって改善された加速度波形を示す．左上以外のグラフ

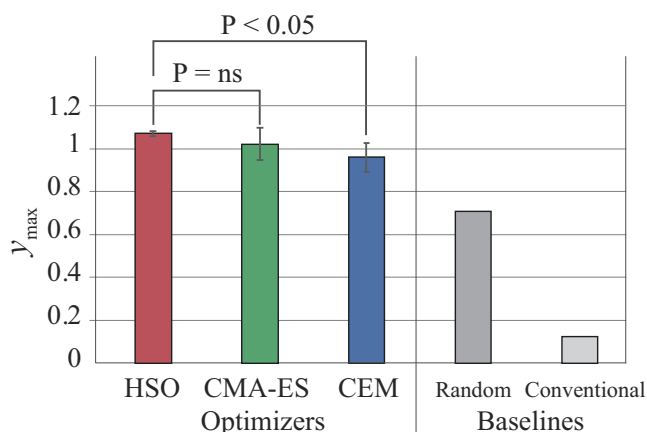
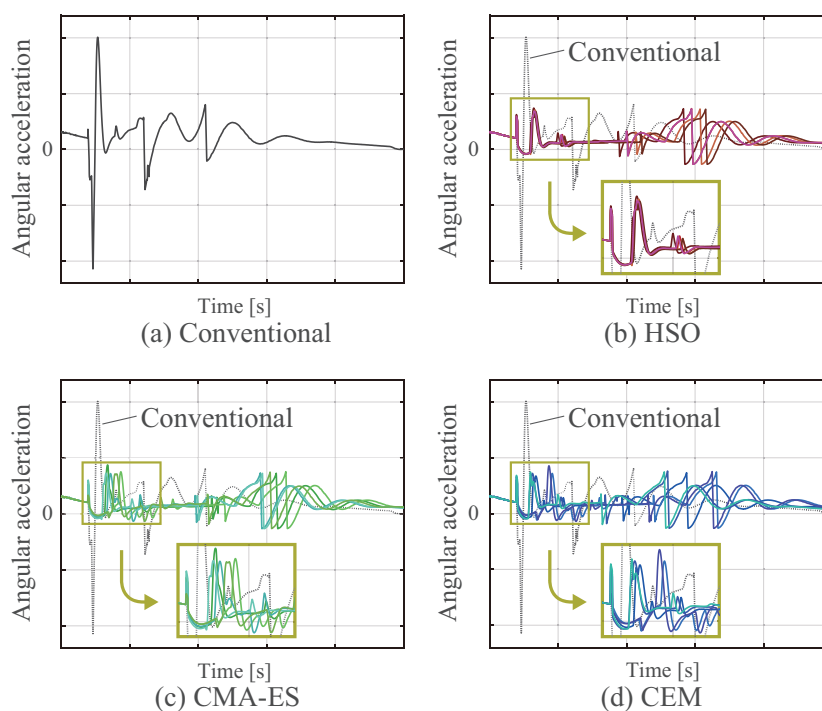
図 4.14 10,000 サンプルにわたる最適化の最終結果 ( $n = 5$ ).

図 4.15 各手法により最適化された加速度波形.

の5本の線は、5試行の最適化により得られた5つの解に対応する。最適化された波形はすべて、現行パラメータから計算された波形よりもはるかに小さいピークを持つ。HSOにより最適化された波形は互いによく一致しており、HSOは良い収束特性を持っていることがわかる。

次に、HSOの最適化過程の調査を行った。図4.16は、HSOによる最適化における各下位方策(オプション)の目的関数値の推移を示す。最適化の全行程を通じて下位の方策の数は2以上であり、HSOは常に目的関数のことを多峰性を持つ関数と見なしている。

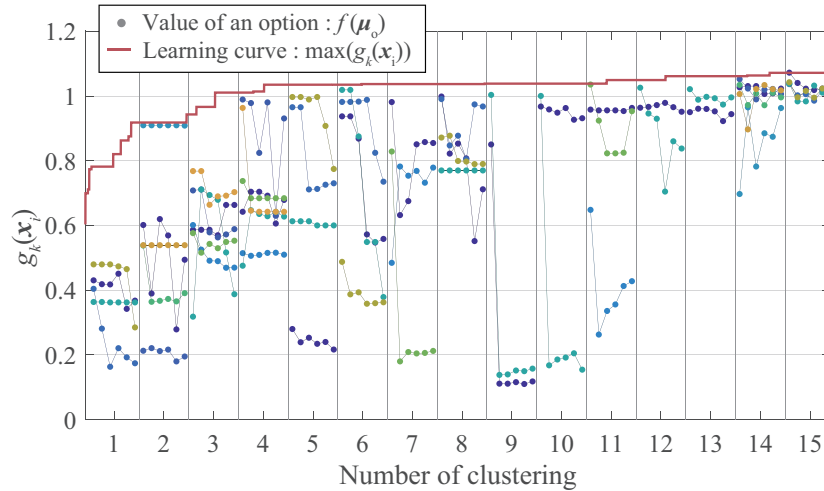


図 4.16 学習曲線と下位方策の中心が与える目的関数値の推移。

11 次元空間でのサンプルの分布を視覚化するには、t-SNE [43] を使用した。t-SNE は、高次元空間の数値ベクトルを、元の空間での位置関係を維持しながら、低次元空間に投影する次元削減の手法である。図 4.17 は、HSO が定期的に行うクラスタリングの間に取得されるサンプルの分布を示す。t-SNE の特性により、各分布図の軸は意味合いが互いに異なる。一部のサンプルのクラスは t-SNE により次元削減した 2 次元空間上においても分離されており、個別のクラスに目的関数の値が高い複数のサンプルが含まれる。このことから、HSO が目的関数の多峰性を把握していると考えられる。さらに、HSO は、11 回目のクラスタリング時に一時的に目的関数を単峰性の関数であると見なし、その後、15 回目のクラスタリング時に再び多峰性のある関数であると見なした。最初に、HSO は初期段階 (3 回目から 7 回目) で大局的な多峰性を見つけ、その中の 1 つのピークを登った後、15 回目においてピークの局所的な多峰性を見つけていると考えることができる。このような現象は、下位方策の数が単調に減少する HiREPS[92] に基づく最適化手法では発生しない。

図 4.17 中列では、下位方策の中心に対応する加速度波形を示している。最適化の初期段階 (3 回目から 7 回目) では、これらの波形は互いに異なり、各下位方策は同程度に良い目的関数の値を与える。HSO の以上のような特性により、大局的な最適化が可能になり、安定した高い最適化性能が達成されていると考えられる。

#### 単一運転条件における重みを振った実験

本小項では、評価指標の重み付けが変わったときに波形に生じる変化に関する実験について述べる。実験では、HSO が異なる重み付け条件を反映しながら波形を最適化可能かどうかを検証する。

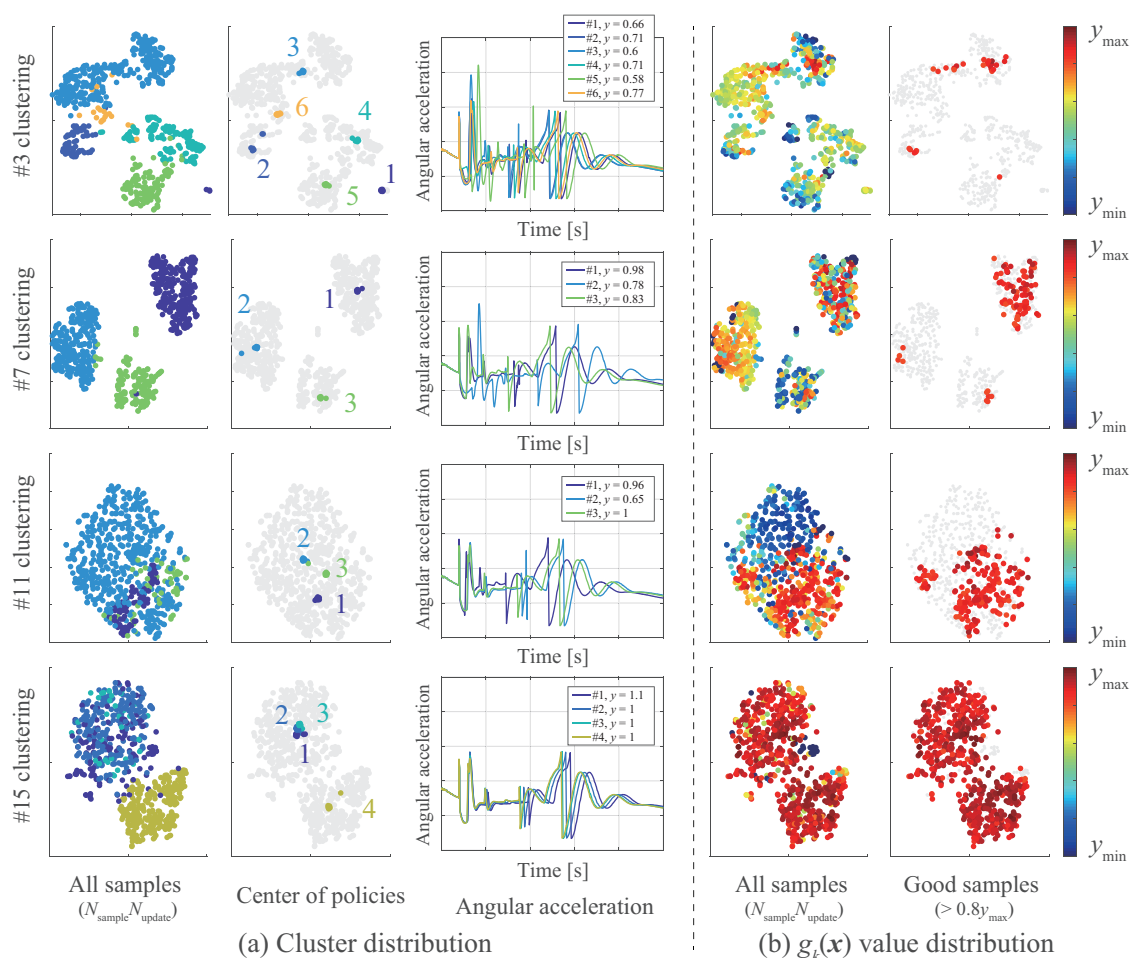


図 4.17 t-SNE [43] により次元圧縮したサンプルの分布とサンプルの特性.

利用する重み付け条件は、4.3.2 項で述べた  $w_1$ ,  $w_2$ ,  $w_3$  の 3 種類である.

表 4.2 異なる重み付けにおいて最適化されたパラメータの各要素.

Condition	Trial #	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	$g_k$
Conventional	-	0.4	0.4	0.2	0.4	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.06
$w_1$	1	0.7	0.0	0.1	0.2	0.0	0.6	0.2	0.0	0.0	0.8	0.0	1.06
	2	0.6	1.0	1.0	0.0	1.0	0.0	0.3	0.1	0.0	0.8	0.0	1.09
	3	0.6	0.4	1.0	0.0	1.0	0.5	0.3	0.0	0.1	0.8	0.0	1.09
$w_2$	1	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.2	0.5	0.0	-
$w_3$	1	0.0	1.0	0.3	1.0	1.0	0.9	0.1	0.0	0.0	0.1	0.0	-

図 4.18 は異なる重み付け条件のもとで最適化された回転速度，加速度，発生熱の波形を示す.  $w_1$  や  $w_2$  の加速度ピークを低減させる重み付けのもとでは，図 4.18(b) に示すように，加速度のピークが現行のパラメータおよび他の重み付け条件と比べて小さくなる.  $w_3$  の発生熱低減の条件のもとでは，図 4.18(c) に示すように，総熱量が小さくなる. この

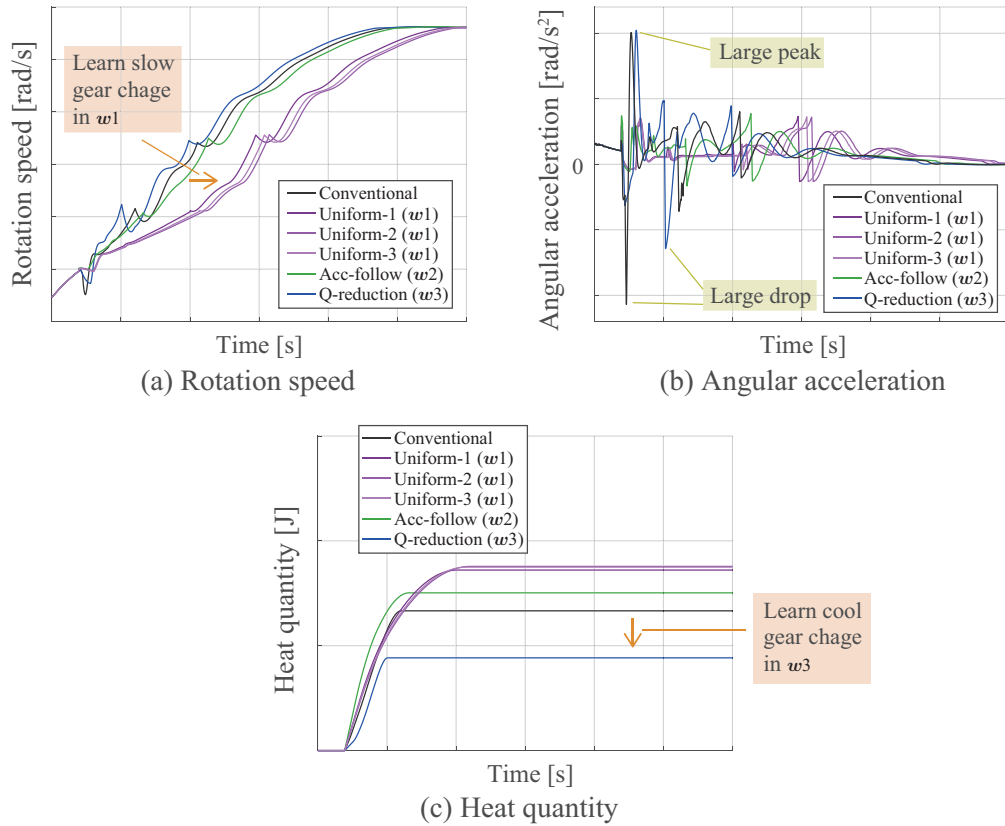


図 4.18 異なる重み付けにおいて最適化されたパラメータの与える物理量.

ように、HSO は重み付け条件を反映してパラメータ及び波形を最適化することができる。

表 4.2 は、各重み付け条件で最適化されたパラメータベクトルの各要素を示す。異なる重みのもとで最適化をした結果、パラメータの要素ごとに異なる挙動を示した。  $x_1$  や  $x_{10}$  は異なる重み付け条件のもとで異なる値に収束する。一方、  $x_8$ ,  $x_9$ ,  $x_{11}$  はどの重み付け条件のもとでも同じ値に収束する。

また、表 4.2 のように最適化されたパラメータはいずれも、人間の設計者が設定した現行のパラメータとはまったく異なる傾向となった。今回最適化されたパラメータと現行パラメータとは 3 つ以上の要素に違いがあるため、人間は探索空間をイメージすることが難しく、最適化されたパラメータに到達することは難しいと考えられる。

#### 4.4.4 複数運転条件での最適化実験

本項では、複数運転条件を考慮した目的関数  $f(x)$  を利用した最適化結果について述べる。

複数運転条件での最適化の学習曲線は図 4.19 のようになった。最適化は  $w_4$  と  $w_5$  の 2 つの重み条件で行っているが、図 4.19 では  $w_4$  での最適化結果のみを示している。図



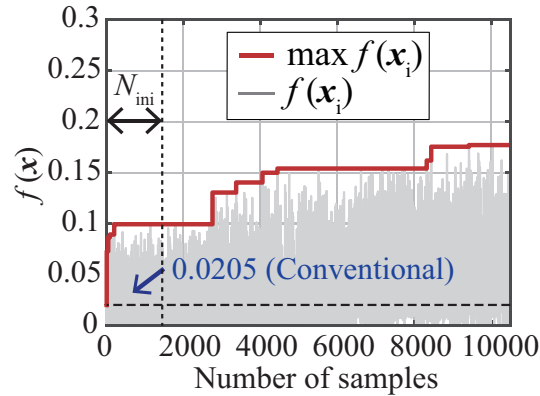


図 4.19 複数運転条件下における最適化の学習曲線。

4.19 の水平点線は現行パラメータの目的関数値を示しており，最適化により目的関数の値の上では現行を大幅に上回るパラメータを見つけ出すことに成功している。

次節の実機を利用した評価実験においては，本項の実験で最適化されたパラメータを用いる。

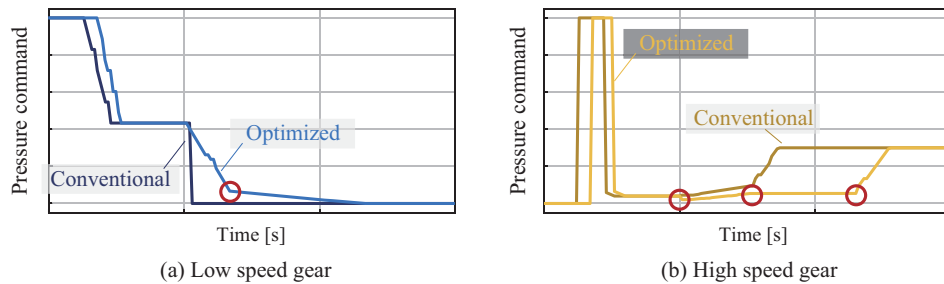


図 4.20 複数運転条件に対して最適化されたパラメータの与える油圧波形。

図 4.20 では，現行パラメータ，最適化されたパラメータが与える油圧指令パターンを比較している．図 4.20(a) は低速側のクラッチの油圧パターン，図 4.20(b) は高速側のクラッチの油圧パターンである．それぞれのグラフにおいて暗い線が現行のパラメータ、明るい線が最適化されたパラメータを示す．現行と最適を比較すると，図中赤丸のように最適ではパターンの折れ曲がりの回数が増えており，より複雑な波形になっていることが確認された。

#### 4.4.5 考察

実験結果は，図 4.13，図 4.14 に示すように，HSO が CMA-ES および CEM よりも優れていることを示す．HSO は目的関数の多峰性を考慮してパラメータを最適化するため，他の方法よりも安定した性能を実現する．本研究がターゲットとしているような機械シス



テムの最適化においては，このような安定性の高さは特に長所になる．安定性が高いと，トランスミッションの制御における運転条件のような前提条件を少し変えて最適化を行ったときに同様な解にたどりつきやすくなる．

図 4.17 に示すように，HSO の動作を視覚化すると，実験の目的関数が多峰性を持ち，HSO がそのような多峰性を把握できていることがわかる．試行回数によって CMA-ES や CEM の収束するパラメータと目的関数の値が大きく変化することも，目的関数の多峰性を示唆している．

また，図 4.17 に示すように，HSO は互いに質的に異なる複数のオプションを見つけることができるため，人間の設計者は最適化の結果から都合の良いオプションを簡単に選択できる．たとえば，2 番目に最適なオプションが目的関数の値に関しては 1 番目に最適なオプションよりも低いものの，人間の設計者が手動で調整した現行パラメータにより似ているということがありうる．そのような場合，設計者は 2 番目に最適なパラメータを使用することを好む．そして，HSO は実際にそのようなオプションを提示することができる．

さらに，HSO やその他の最適化方法では，図 4.15 に示すように，人間の設計者が調整した現行のパラメータよりも優れた物理的特性を出力するような，良いパラメータを見つけることができた．これらの最適化には 1 日もかかっておらず，手動の調整に比べて費用効率が低い．以上の議論から，提案する HSO は，設計者による手動パラメータ調整の作業を代替することができると考えられる．

## 4.5 実機による評価実験

本節では最適化されたパラメータの実機における評価実験について述べる．実機実験の目的は以下の通りである．

- 計算機上で最適化したパラメータ最適化の変速ショック低減効果を評価する．(ID:TM-B-2 の検証)
- 実機と数値解析モデルで同じパラメータの評価を行い 2 者の誤差や誤差原因を明らかにする．(ID:TM-B-1 の検証)
- 計算機上での最適化結果をドライバーの載っている実機にて評価し ECT の目的関数修正のための知見を得る．(ID:TM-A-1 の実践)

### 4.5.1 実験設定

実験には図 4.21 に示す全長 9 m，積載重量 30 t 級のダンブトラックを利用した．数値シミュレーションにおいてもダンブトラック実機と同様のトランスミッション構成や車体

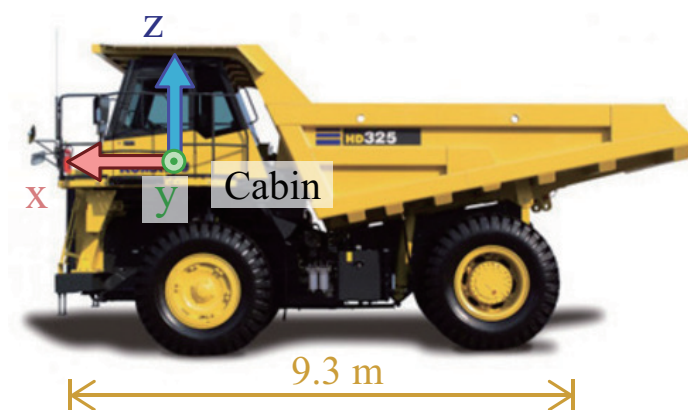


図 4.21 ダンプトラックの実機と座標系の設定.

質量を採用している．実験における走行条件は表 4.3 のように設定した．

各走行では特定の回転数まで加速した後，スロットル開度 100% のベタ踏みの状態を保つことで，計測条件を統制している．それぞれの走行において 2000 Hz の周波数で加速度を計測した．変速ショック低減の上位の目的はドライバーの乗り心地向上であるため，乗り心地に密接に関係するキャビン上での加速度を計測した．加速度計は図 4.21 のように x,y,z 軸がそれぞれ車体の進行方向，左右方向，上下方向に対応するように設置した．また計測された加速度はバターワースフィルタを用いて高周波成分をカットした．ドライバーの上下方向加速度の感度は 10 Hz を越えると減少する [93] ことから，フィルタの通過域端周波数と阻止域端周波数はそれぞれ 10 Hz, 100 Hz とした．また，運転条件とは独立で時間経過に従って加速度の平均値がドリフトする傾向があったため，加速度の絶対値は解析に利用せず走行内の変動量にのみ注目して解析を行う．キャビン上の加速度は車体による減衰，車体の剛性，重量バランスなどから影響を受けるため，数値解析モデルでは模擬が難しい．そこで，数値解析モデルと実機とを比較する際にはトランスミッションの出力軸の回転速度および角加速度を比較した．

表 4.3 実機評価実験における実験条件.

Throttle opening	100 % open
Slope	−6, 0, 6 [deg]
Number of trials	3

## 4.5.2 実験結果

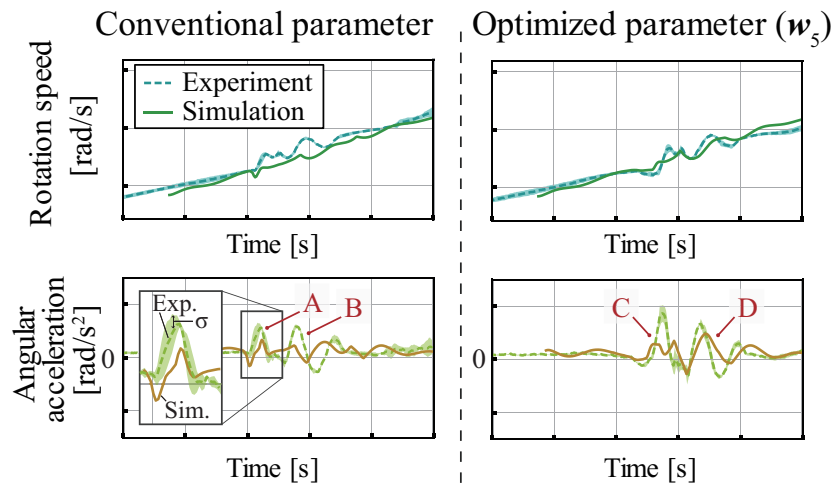


図 4.22 数値解析モデルと実機との誤差の観察.

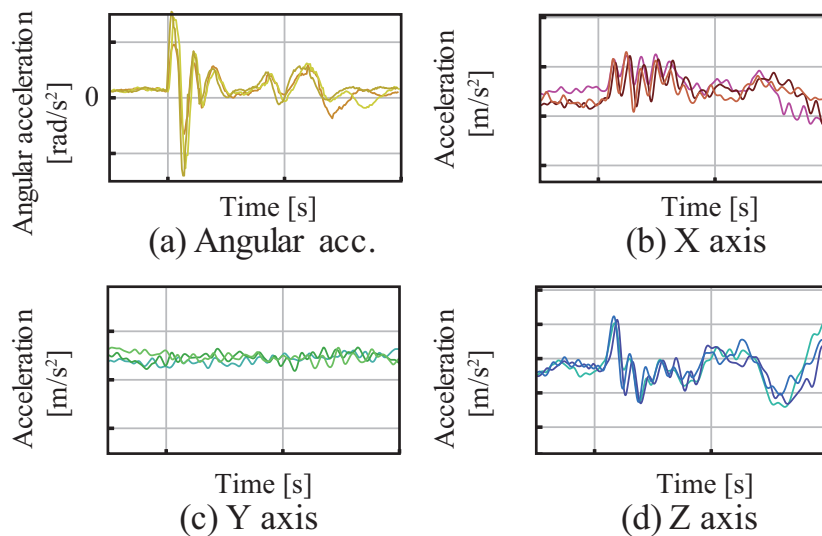


図 4.23 出力軸角加速度とキャビン上で計測された3軸加速度.

最適化されたパラメータは設計者が調整した現行パラメータから大きくかけ離れていたが、最適化時の定義域設定を慎重に実施したため実機で利用しても問題なく走行が可能であった。

図 4.22 は、勾配 0 deg の走行条件におけるシミュレーション結果と実機計測値との比較である。左列が最適化する前に用いられていた現行パラメータでの結果、右列が重みベクトル  $w_5$  の元で最適化されたパラメータでの結果を表す。上段が出力軸の回転速度、下

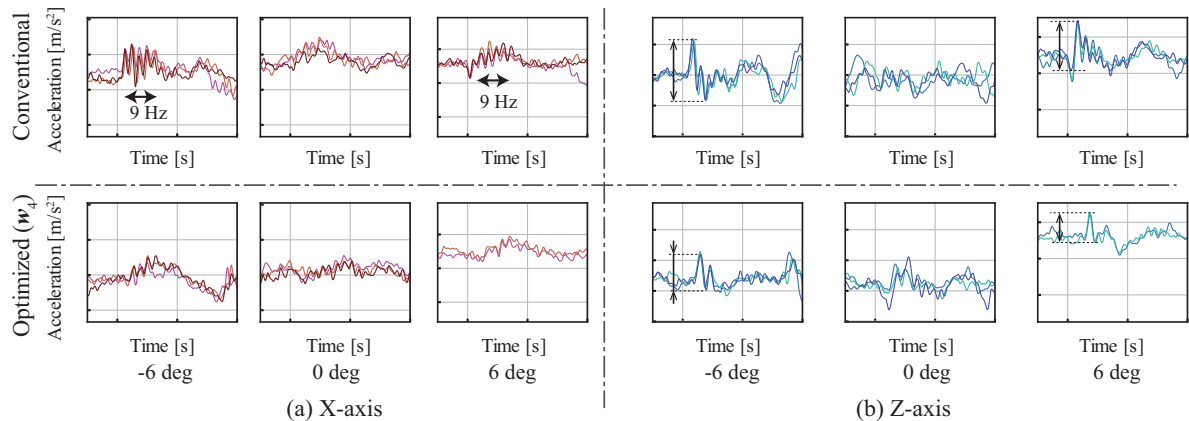


図 4.24 現行パラメータと最適化されたパラメータにおける加速度比較.

段が角加速度である．回転速度は最適パラメータでは良く一致するが，現行パラメータでは一致しない．角加速度の場合，現行パラメータでは 1 つめの大きなピーク (図中 A) で波形が一致しているが，2 つめのピーク (図中 B) では一致していない．最適パラメータでは逆に 1 つめのピーク (図中 C) の山で一致せず，2 つめのピーク (図中 D) で一致している．パラメータによって一致する箇所が異なるため，数値シミュレーションと実機との乖離には複数の原因が絡んでいる可能性がある．

図 4.23 は加速度波形の比較である．左上がトランスミッション出力軸の角加速度であり，それ以外がキャビンの加速度計の計測値である．図 4.23(b)(c)(d) のグラフは縦軸の具体的数値を省略しているが，縦軸の目盛りは同じ間隔となっている．4 つのグラフともに同条件の 3 走行の波形を表示しているが，3 波形はお互いに重なっている部分が多く，各走行には再現性があるといえる．図 4.23 によると，進行方向の x 軸だけでなく，上下方向である z 軸で加速度振幅が大きくなっていることがわかる．また，波形の類似性から角加速度の変動により生じる変速ショックは x 軸ではなく，z 軸によく現れることがわかる．変速ショックによるエネルギーが，前後方向の揺動に対してだけでなく，車体のピッチ方向の回転させることにも使われた結果だと考えられる．

最適化前後の加速度を図 4.24 に示す．上段が現行パラメータ，下段が重みベクトル  $w_4$  の元で最適化されたパラメータを表す．左列が x 軸加速度，右列が z 軸加速度を表し，左列・右列内の各列は走行コースの勾配を表す．図 4.24 の各グラフにおいても縦軸の具体的数値を省略しているが，比較を行う上段と下段のグラフで縦軸の目盛りは同じ間隔である．図 4.24 の上下を比較すると，勾配が  $-6 \text{ deg}$  及び  $6 \text{ deg}$  のとき，x 軸，z 軸ともに最適化により加速度振幅が小さくなっていることがわかる．また，x 軸加速度に関して現行パラメータでは  $9 \text{ Hz}$  程度の加速度振幅が目立っていたが，最適化によって低減された．勾配が  $0 \text{ deg}$  の条件では最適化前後で加速度振幅に大きな変化がなかった．この原因とし

て、0 deg は実機試験が行いやすいためパラメータチューニングの基準として使われていたということが考えられる。

以上のように数値シミュレーションを用いた最適化によって現行と同等あるいはそれ以上の性能を示すパラメータを見つけることができた。

## 4.6 メタな最適化システムにおける人間の介在の様子

本章では、トランスミッションの制御パラメータ最適化問題に対して、メタな最適化の観点からアプローチした。本節では4.2節で提案したメタな最適化システムにおいて人間が介入した部分の概略及び考察結果について述べる。

### 4.6.1 人間の介入が必要となった事例

いくつかのプロセス(4.2節で説明)ではシステムを動作させるために、人間の介入が必要であった。以下では、介入事例の概要として、対応するプロセス、事例の経過、実施した可視化、人間が利用した知識について述べる。なお、メタな最適化システムにおける動作の情報は製品として顕在化しないノウハウを多く含んでいるため、本節では動作を十分一般化した概略を述べるにとどめる。また、以下の事例以外にも、取り組みの担当者同士で情報交換をしたり、数値解析モデルと目的関数と最適化手法とを組みあわせるためのコーディングなど、人間の介入が必要な部分は存在した。ただ、それらの部分は本研究がターゲットとしている人間と計算機の相補関係とは直接関係がないため、本論文では紹介を行わない。

#### ■目的関数の試作段階における数値解析モデルの修正

- 対応するプロセス：「2. モデル化」，「3. 目的関数の作成」
- 事例の経過：数値シミュレーションから出力された物理量の時間刻みが一定ではなく、パラメータ  $x$  がわずかに変動するだけで指標  $m$  の値が大きく変化していた。指標の部分空間における可視化結果を行うことでこの問題が検知され、数値解析モデルが修正された。
- 実施した可視化：指標(目的関数)の部分空間における関数形状
- 人間が利用した知識：指標の関数形状に関する知識

#### ■計算機上での最適化の結果を利用した数値解析モデルの修正

- 対応するプロセス：「8. 意図せぬ挙動の検出」
- 事例の経過：計算機上での最適化により最適化されたパラメータ  $x^*$  は、数値解析

モデルの想定していない挙動を引き起こして目的関数の値を大きくするようなパラメータであった。最適化されたパラメータが与える物理量を可視化することでこの問題が検知され、数値解析モデルが修正された。

- 実施した可視化：数値解析モデルが出力する物理量の波形
- 人間が利用した知識：実機における機械の動作に関する知識、数値解析モデルの内部実装に関する知識

#### ■ 重みと最適化結果の対応関係を利用した重みの調整

- 対応するプロセス：「9. 目的関数の適合」
- 事例の経過：重み  $w$  の調整においては、まず、設計者が仮説をもって幾つかの重みのパターンを設定した上で最適化を実施した。そして、最適化結果を比較し、設計者が理想とする波形の時に、目的関数の値が高くなるように重みを調整していった。人間は指標の物理的意味を理解しているため、ある重みに関しては常に 0 でも要求性能を満たすということを早期に見抜き、それ以降はその重みを常に 0 にして、調整すべき重みの数を減らすことができた。
- 実施した可視化：数値解析モデルが出力する物理量  $q$  の波形
- 人間が利用した知識：各指標の物理的意味に関する知識、重み付けと最適化結果の対応関係に関する知識、好ましい波形に関する知識

#### ■ 重みと最適化結果の対応関係を利用した評価指標の追加

- 対応するプロセス：「9. 目的関数の適合」
- 事例の経過：重み  $w$  と最適化結果の対応関係を探っていく中で、当初設定されていた指標の元では、重みをどう組み合わせても理想的な加速波形にならないということが経験的にわかってきた。そこで、理想とする波形で目的関数の値が高くなるように新しい指標を追加した。それにより、新指標が無かった場合に比べより好ましい波形を獲得することができるようになった。
- 実施した可視化：数値解析モデルが出力する物理量  $q$  の波形、いくつかの波形と指標の値の対応関係
- 人間が利用した知識：各指標の物理的意味に関する知識、重み付けと最適化結果の対応関係に関する知識、好ましい波形に関する知識

### 4.6.2 メタな最適化の動作結果に関する考察

図 2.7 のような非メタな最適化システムは上記の事例で出てきたような知識が不要なため、人間の介入が必要なく自律的に動作するが、数値解析モデルや目的関数の定義に問題があった場合、非メタな最適化システムの内部から検出することが難しい。

4.6.1 で紹介した人間の介入事例では、全ての事例において物理量の可視化が行われた。人間による物理量の確認にも時間的コストがかかるため、主要物理量が自動的にグラフとして出力されるような仕組みや、パラメータを摂動させた際の物理量の変化量が大きい場合に警告を出すなどの仕組みを取り入れることにより、コストの低減が期待される。

また、上述のように、人間の介入事例においては様々な知識が利用された。現代の自動的なアルゴリズムにとって、指標の物理的意味や好ましい波形に関する知識を利用して、重みや指標を修正することは困難であるが、最適化結果に大きな影響を持つ重みを提示して人間の判断の材料を提供することは可能であると考えられる。また、事例の中で利用された知識の中でも、好ましい波形に関する知識や、指標の関数形状に関する知識は全てを言語化するのが難しい暗黙知である。暗黙知を人間の頭から効率的に取り出すためには、物理量の可視化結果を提示し、それが良いか悪いかを人間に判断してもらうことが有効であると考えられる。

人間の判断を待っている時間や人間が別の作業をしている時間を利用して、可視化などの判断の材料を提供していくことで、時間的コストを低減でき効率的なメタな最適化につながると考えられる。今回はメタな最適化を進めていた著者が可視化手法の開発も行っていたため、その分メタな最適化にかかる時間がかかっていたが、以降の取り組みではこの時間は削減されることが期待できる。

## 4.7 本章の結言

本章では、電子制御トランスミッションの制御パラメータ最適化について述べた。具体的には、目的関数の定式化、計算機上での最適化、最適化されたパラメータの実機における評価を行った。

本章における研究目的と、それらに対応する結果を以下に述べる。

### 本章における研究目的 (再掲)

- 研究課題を解決するために手法を提案し実行する。(TM-A)
  1. トランスミッションの性能評価のための目的関数を定式化する。(TM-A-1)
  2. 実機と計算機上の数値解析モデルで同様のパラメータの出力する物理量の比較

を行い誤差発生要因を明らかにする。(TM-A-2)

- 実際の機械システムのパラメータ最適化まで行い、各問題の最適化が現代の技術の組み合わせで解決が可能か否かについて、デモンストレーションを行うことにより検証する。(TM-B)
  1. 計算機上における目的関数の最適化の可否や技術的妥当性を検証する。(TM-B-1)
  2. 計算機上で最適化したパラメータを実機に適用し性能を評価する。(TM-B-2)
- 目的関数の多峰性を仮定した最適化手法である HSO を提案し、性能を評価する。(TM-C)

#### 取り組みの結果

目的毎に取り組みの結果について述べる。

■(TM-A-1) トランスミッションの性能評価のための目的関数を定式化する ECT の性能を評価するための目的関数の設計を行った。目的関数は複数の性能指標と、加速や発熱などの設計上の制約から構成される。計算機上の最適化、評価指標の重みを変えた実験などを通して、設計者や設計現場にある性能目標を汲み取りながら、目的関数の形に落としこんだ。目的関数の部分空間における観察の結果として、目的関数には多峰性があることが示唆された。実機における評価実験において、数値解析モデルで計算可能な出力軸角加速度は、キャビンの前後方向の加速度というよりはむしろ上下方向の加速度に大きく影響することが明らかになった。乗り心地を改善するためには、出力軸角加速度の変化の影響を受けづらい前後方向加速度を考慮できるような目的関数を設計する必要があることが示唆された。

■(TM-A-2) 実実機と計算機上の数値解析モデルで同様のパラメータの出力する物理量の比較を行い誤差発生要因を明らかにする 実機と数値解析モデルにおいて同じパラメータが出力する物理量を比較し、誤差発生要因を明らかにした。

■(TM-B-1) 計算機上における目的関数の最適化の可否や技術的妥当性を検証する 複数の評価指標があり、複数の運転条件を取るトランスミッションの制御パラメータ最適化という問題に対し、油圧系を考慮可能な数値解析モデルとブラックボックス最適化手法(BBO)とを組み合わせたシステムを構築した。実験により、構築したシステムは、1日程度の時間で実機に利用できるような最適化されたパラメータを算出することができると示された。最適化システムは、人間の設計者に比べて目的関数の値の上で良いパラメータを見つけ出すことができた。この主要な理由としては、以下の二つがあると考えられる。

1 つ目の理由は今回の最適化問題の探索空間の広さである。人間は定性的な因果のつな



がりによりパラメータ調整を行うため、探索空間の次元数が大きくなり因果のつながりが多次元に及ぶと、因果のつながりを追うことが難しくなる。一方で、最適化アルゴリズムは低次元パラメータ空間の場合と高次元パラメータ空間の場合とで質的に同じ最適化を行うことができる。11次元のように多くの次元のパラメータを同時に変更してよい場合には、アルゴリズムは人間に比べて広域において有効な探索を行うことができ、探索性能の差につながったと考えている。

2つ目の理由はパラメータ  $x$  から目的関数の値  $y$  までの因果のつながりの長さである。今回のトランスミッションの制御パラメータ最適化においては、パラメータ  $x \rightarrow$  油圧指令  $\rightarrow$  トランスミッション動作  $\rightarrow$  回転数, 発生熱  $q \rightarrow$  目的関数  $y$  というつながりがある。このようにつながりが長いと、パラメータ変更が目的関数に及ぼす影響が複雑になり、把握が難しくなる。具体的には、因果のつながりが何本も見え、それらが足し合わさって  $y$  に影響を及ぼす構図になると考えられる。このような際には、人間が得意とする因果に基づく推論は困難になり、因果に基づいていない最適化アルゴリズムにとって相対的に有利になると考えることができる。

また、本システムによる最適化試行において、最適化アルゴリズムが数値解析モデルにおいて人間の想定しないような挙動を発生させ、目的関数の値を高くすることが発生した。

■(TM-B-2) 計算機上で最適化したパラメータを実機に適用し性能を評価する システムによって最適化されたパラメータは実機でも問題なく動作し、登坂・降坂の走行条件において現行のパラメータより良い加速特性を示した。今回構築した最適化システムは、同様の問題設定となる他機種のトランスミッションや他の油圧機械のパラメータ最適化においても適用が可能だと考えられる。

■(TM-C) 目的関数の多峰性を仮定した最適化手法である HSO を提案し、性能を評価する 多峰性が考慮可能な階層強化学習の手法を参考とし、ブラックボックス最適化手法である HSO を提案した。HSO では目的関数の多峰性を把握するために、多峰性の密度分布の推定問題として問題をとらえる。密度分布の推定問題に対し、HSO はサンプリング方策を階層化し、複数の下位方策を学習する。

本章の取り組みにおいては、ECT の制御パラメータの最適化に HSO を適用することにより、HSO の性能や挙動の確認を行った。実験により、HSO は CMA-ES や CEM などの広く利用されているブラックボックス最適化手法に比べて安定した性能を示すことが分かった。また、HSO の最適化過程の観察から、HSO が目的関数の複数の峰を把握できることが示された。目的関数の多峰性が示唆されたこと、HSO の性能の安定性が高いことも考慮すると、HSO の目的関数の多峰性を把握する特性により、局所最適解に陥りづらくなったため良い性能になったと考えられる。一方で、今回の目的関数が HSO にとって

良い問題設定で有ったという可能性も排除はできない。次元数が違う問題や、目的関数の形状が定性的に異なる問題においても比較を行うことで、HSO の性能の良さを主張するための証拠を集めることができると考えている。

#### 将来課題

数値解析モデル-実機間の誤差を小さくすることで、数値解析モデルと実機の最適化の差も小さくなり、数値解析モデルを活用した最適化の効果が大きくなると期待されるため、今後は得られた知見をもとに数値解析モデルの改善を行う。

また、各最適化アルゴリズムによる探索範囲の可視化を行うことで、最適化アルゴリズムの定性的性能の議論が行えると考えている。今回の検証では HSO の探索したパラメータを次元圧縮の手法で可視化することで、HSO が多峰性を捉えているという証拠を見つけた。将来課題としてはアルゴリズム間で探索範囲を可視化・比較し、各アルゴリズムの特性を定性的に議論したい。高次元な目的関数に対するブラックボックス最適化手法のベンチマーク研究は少なく、アルゴリズムを定性的に比較する研究も少ないため、上記のような議論の貢献は大きくなると期待される。

## 第 5 章

# 円筒研削加工条件の最適化

本章では数値解析モデルを活用した円筒研削加工の加工条件最適化について述べる．円筒研削加工の加工条件最適化の研究背景について説明した後，数値解析モデルをベースとした最適化システム，及び，実機を対象とした最適化実験について述べる．

### 5.1 研究課題

本節では円筒研削加工と加工パラメータの最適化の関連研究について述べた後，本研究において取り組む研究課題について述べる．

#### 5.1.1 研削加工

工作機械は機械を作り出すための機械であり，産業革命当時から存在し，現代に至るまで発展を続けてきた [94]．蒸気機関の発明以降，工作機械と蒸気機関による動力が組み合わされることにより，高品質な部品を人の腕力や技術に頼らず安定的に製造することが可能になり，それまでの手工業的な機械の生産を代替していった [2]．部品の品質を決める指標には正確度と精度が存在する．正確度は加工結果のばらつきを表し，部品の正確度が高いと，機械設計のモジュール化や規格化が行いやすくなる．精度は加工結果が設計上の指示の順守度を表し，部品の精度が高いと，エネルギー効率や気密性などの機械の性能の向上につながる．精度や正確度が高い加工を低コストに行えることは工作機械に対する基本的要求である．加工を行うにあたり，加工される材料 (ワーク) の材質や加工後の形状によって適した加工法は異なる [94]．図 5.1 は研究開発が行われている生産加工技術の一覧図である．例えば，モース硬度の考え方に従うと，加工を行う工具は加工されるワークより硬くなくてはならない．ワークが硬いほど工具の選択肢が減るため，硬いワークを削るための工具は高価になると考えられる．(最も硬い工具に無限の入手性がある場合，この考えは破綻する．) そのような状況では硬度 6 のワークは安価な硬度 8 の工具で加工し，

硬度 8 のワークは高価な硬度 10 の工具で加工するというような役割分担が生じる。加工物の種類に応じて適した加工法が異なるため、複数の種類の工作機械が登場し、社会全体として様々な部品を低コストで製作できるようになった。

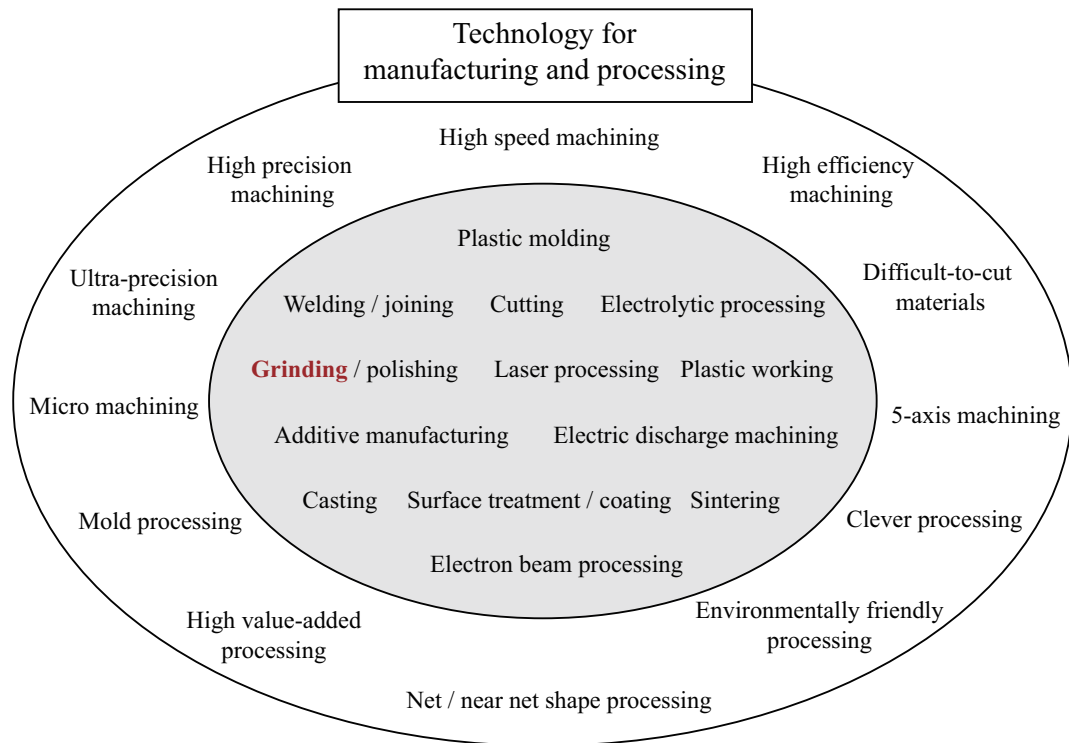


図 5.1 生産加工技術の種類。(文献 [94] を元に作成)

本研究では除去加工の一種である、研削加工に焦点を当てる。図 5.2 は代表的な研削加工である円筒研削加工の加工の原理を表す。研削加工は数ミリスケールもしくはそれより小型 [95] で高い硬度を持つ砥粒 (Abrasive grain) を連続的にワーク (Workpiece, 加工対象, 被削材) に当ててワークを削る加工法である。対して、研削加工と同様に良く用いられる切削加工では金属製の刃をもつ工具によりワークを削る。研削加工の特徴は、工具としてワークより硬度が高い砥粒を利用しているところにある。ワークの材質として代表的な鋼鉄は熱処理することにより硬度を増して耐久性を向上させることができるが、切削加工のような金属製の工具では加工しづらくなってしまふ。砥粒の材料として代表的なアルミナやダイヤモンド、CBN (Cubic Boron Nitride) は焼き入れにより硬度を増した鋼鉄より十分に硬度があり、加工を行うことができる。研削加工はこのように熱処理を施したワークに対して適用される。研削加工は硬いワークを削れる一方で、材料の除去にかかるエネルギーが切削加工に比べ大きく、材料の除去速度が小さい。そのため、鋼鉄製の機械部品は硬度が低い状態で切削加工により高速に加工が行われ、熱処理が施され、仕上げのために研削加工が行われる [78]。

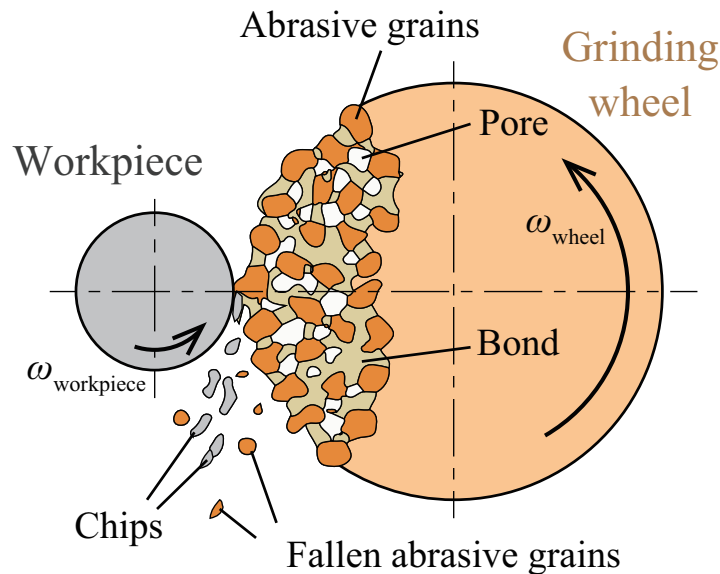


図 5.2 研削加工の原理. (文献 [96] を元に作成)

研削加工は様々な場面で利用される。研削加工された表面は硬く、寸法精度が高く、表面粗さが小さくなる [78]。そのため、研削加工は、接触や摺動が頻繁に起こる歯車やシャフトのような機械部品や、微細な形状が性能に影響を及ぼす半導体ウェハの加工に利用される。また、硬い材料を加工可能という特性を活かして、治療のために歯や骨などの生体組織を削ることに利用される。研削加工では、ワーク上に固定した座標系から見たときの砥石の走査体積の形状がワークへと転写される。研削加工ではワークが円柱形や平面形に加工されることが最も多いが、砥石の形状を工夫することで歯車 [78] やルーローの三角形のような複雑な形状も加工することが可能であり、研削加工の適用範囲は広い。本研究では機械産業を主なターゲットしているため、大量生産シャフトやベアリングのころのような金属の機械部品に対して用いられる円筒研削加工に焦点を絞る。

本研究で扱う円筒研削加工は、機械部品を作り出す 1 プロセスであるため、機械を評価するためのコストという指標を継承する。コストに関係する円筒研削加工の評価指標には次のようなものがある。

- 加工に必要な時間
- 加工に必要なエネルギー
- 加工に必要な研削油 (冷却油) の量
- 砥石 (工具) の摩耗量

また、機械の機能に影響するため、円筒研削加工により加工される表面の状態などの加工物も評価対象となる。加工物の評価指標には次のようなものがある。

- 表面応力状態
- 表面粗さ (1 次元, 2 次元)
- 形状精度 (真円度, 円筒度など)
- 寸法精度
- うねり
- 研削焼けの有無
- 表面硬度
- 表面のキズ

本章においては, これらの評価指標を合わせて生産性指標もしくは生産性と定義する. 機械加工の現場で表面粗さは 1 次元のものである  $R_a$ ,  $R_z$  が用いられることが多いが, 近年では表面の 2 次元形状も潤滑性や耐久性に影響すると言われている [97]. 今後, 製品の高性能化のために, 表面の 2 次元形状まで考慮した加工が行われるようになる可能性がある.

工作機械は様々な形状や材質のワークを加工可能なように, ある程度汎用性をもつように設計されている. そのため, 工作機械の性能を引き出すためには, 形状や材質によって工作機械の動かし方を調整する必要があるといえる. 本章においては工作機械の動かし方を決める値を制御パラメータと呼称する. 移動をつかさどる制御パラメータは工具の送り速度 (並進移動速度), 工具の回転速度, ワークの送り速度, ワークの回転速度に集約される. 円筒研削加工ではワークが一回転する間にどれくらい工具が進むかという切り込み深さ (Depth of cut) というパラメータが用いられるが, 工具の移動速度とワークの回転速度を決めることにより従属的に定まる. 送り速度や回転速度は工具とワークの相對運動のパスを決定する. 汎用的な工作機械は複数の並進方向の移動自由度を持っている, また時間方向にも自由度を持っているため, 具体的な値を持つ制御パラメータは 4 種類より多くなる. 例えば, 加工序盤に切り込み深さを大きくして, 加工終盤に切り込み深さを小さくする「粗・仕上げ加工」は, 切り込み深さを同一にした加工に比べて, 同一時間で達成できる表面精度が良くなる. このように相對速度を時間に応じて変化させることにより, 加工の生産性を向上させることができる.

制御パラメータ以外にも加工機の設計値や工具材質などの調整パラメータは存在するが, 問題が複雑になるため本研究においては機械や工具が所与で固定であるものとし, 制御パラメータのみを最適化する問題を扱う. 制御パラメータ以外のパラメータには, 砥粒の形状, 機械の剛性, 油の供給機構など多くのものが存在し, 研究が進められている. これらもパラメータであることを考えると, これらを含めて最適化が可能となれば大局的な最適解に近づくと考えられる.

研削加工されるワークの表面性状・形状決定には多くの物理現象が影響しており, 研究

が行われてきた。研究が行われてきた代表的な物理現象には次のようなものがある。

1. 発生熱による研削焼け
2. 砥石の損耗及び摩耗
3. 複数の研削モードの存在
4. 熱による変形
5. エネルギーが砥石や加工機に伝わることにより発生する共振
6. 加工の押し付け力による砥石やワーク，加工機の変形
7. 円筒及び歯車研削加工においてワーク・砥石回転数の最小公倍数が小さいことによる局所摩耗形状の転写
8. 研削痕

物理現象を利用したり，不具合を避けるために以下のような様々な工夫が存在する。カッコの内の番号は上記の物理現象に対応する。

- 粗加工と仕上げ加工を分ける (1, 2, 4)
- スパークアウト加工 (4, 5, 6)
- オシレーション加工 (7, 8)
- 回転数の最小公倍数を大きくする (7)
- トラバース加工 (7, 8)

これらは砥石やワークが既にあるものとして，その 2 者の動かし方についての工夫である。動かし方を決める数値は制御パラメータの一種であるとみなすことができる。これらの工夫を独立した工夫として切り取らず，一括に調整できれば，現状に比べて複数の指標において良い加工を実現できると考えられる。

### 5.1.2 関連研究

本項では研削加工最適化に関する関連研究，及び研削加工を模擬する数値解析モデルに関する関連研究を紹介する。

本研究以前も，研削加工を対象として加工条件を最適化する手法の研究がなされてきた。例えば，実験により組み立てた記号モデルを複数利用して最適化を行っている論文 [98][99][100]，統計的モデルを構築して最適化を行う論文 [101][102][103][104] などが存在する。表 5.1 と表 5.2 はこれらの研削加工最適化の研究を整理した表である。表 5.1 は研究の設定に関する情報を整理したものであり，各研究における現象のモデル化の形態，最適化アルゴリズム，検証実験の方法を示している。表 5.2 は最適化時の目的関数の具体的設定を整理したものであり，各研究における，加工の指標，調整対象のパラメータ

表 5.1 研削加工最適化の関連研究において採用されている手法.

Authors	Type of grinding	Modelling form			Number of open paremeters	Number of performance metrics	Preliminary experiment with real machine	Evaluation with real machine	Optimization method	Year
		Symbolic	Statistical	Numerical analysis						
Xiao, et al.	Cylindrical	○			6	1	○	○	MA	1996
Saravanan, et al.	Surface	○			4	3			GA	2002
Brinksmeier, et al.	Cylindrical	○	○		2	3	○		MA	1998
Liao, et al.	Creep feed		○		3	2		○	GA	2010
Dhavkikar, et al.	Centerless		○		4	2		○	RSM + MC	2003
Kwak, et al.	Surface		○		4	1		○	RSM	2005
Sedighi, et al.	Cylindrical		○		3	1		○	RSM	2007
Proposed	Cylindrical			○	6	2		○	CMA-ES	2019

MA: Monotonicity analysis

GA: Genetic algorithm (BBO)

RSM: Response surface methodology

MC: Monte-carlo simulation (BBO)

CMA-ES: Covariance matrix adaptation evolutionary strategy (BBO)

表 5.2 研削加工最適化の関連研究における最適化の設定.

Authors	Open parameter $x$							Performance metrics $s$								
	$N$	Wheel velocity	Workpiece velocity	Depth of cut	Timing	Dressing condition	Others	$N$	Time duration	Power consumption	Wheel life	Material removal rate	Surface roughness	Geometric dimensions	Residual stress	Surface burn
Xiao, et al.	6			2	3	1		1	1		C		C	C		C
Saravanan, et al.	4	1	1			2		3		1		1	1			C
Brinksmeier, et al.	2			1	1			3	1				1		1	
Liao, et al.	3	1	1	1				2				1	1			
Dhavkikar, et al.	4		1	1			2	2						2		
Kwak, et al.	4	1	1	1			1	1				1	C	1		
Sedighi, et al.	3	1	1	1				1					1			
Proposed	6	1	1	2	2			2	1				1	C		

を示している. 表 5.1 内の Symbolic 列の○は記号モデルを利用する研究であることを示している. Statistical 列は実験データに基づいた統計モデルを利用した研究であることを示している. 統計モデルを利用するものは最適化手法として応答曲面法 (RSM; Response Surface Methodology) を利用しているものが多い. 2.1 節の議論から応答曲面を利用する手法は非線形性や局所的な変化に弱いと考えられる. RSM を利用した研究は論文内において, 現実の機械を利用して 20 ~ 30 回程度の実験を行っている. 統計モデルを利用した手法では, 最適化対象のパラメータが増えると, モデルの構築に必要なデータ数が指数関数的に増加する. 現実の機械を利用した実験にはコストがかかるため, パラメータが多くなるに従い, 統計モデルを利用した手法は適用が難しくなる. また, 統計モデルを利用し



た手法は，データ取得実験時に変動させなかったパラメータを変動させようとする，新しいパラメータが出力に与える影響を把握するために，新たに大量のデータを追加取得する必要がある．

研削加工機に対しても，数値解析モデルによって現象をモデリングし，設計に利用するというアプローチの研究 [105][106] が登場している．文献 [105] は歯車研削加工機における加工由来の振動がワーク表面に与える影響を確かめる研究である．研究においては，運動学シミュレーションによる研削力の計算と，マルチボディシミュレーションによる工作機械の変形の両方を考慮できる数値解析モデルを構築し，現象を再現した．加えて，実際の加工機での実験も行い数値解析モデルの妥当性を確かめた．この文献によると，以前の研究では工作機械は工作機械，加工は加工でそれぞれ数値シミュレーションを行っていた．動的に加工力が変わる研削加工において，複数の物理現象間を行ったり来たりするのは難しかったと述べられている．また，文献 [106] では平面研削における個別の砥粒のモデルによる熱計算を行っている．砥粒とワークとの間の接触を模擬した，数値解析モデルを構築し各砥粒の接触により生み出される熱量を精緻に計算している．加えて，熱電対アレイ内蔵したワークを利用した実験を行い，数値解析モデルの妥当性を確認している．

これらの数値解析モデルは計算コストがかかり計算機性能を必要とする．また，これらの検証においては研究者が現実存在する加工機を用いて数値解析モデルの妥当性を確かめている．数値解析的アプローチを採用した論文が登場していることは，数値解析によって挙動を分析することが現代において新規性のある取り組みだということを示唆している．

### 5.1.3 数値解析モデルを活用した円筒研削加工条件最適化

本研究では数値解析モデルを活用した円筒研削加工条件の最適化を取り扱う．本項では加工条件最適化に数値解析モデルを適用することの優位性，適用に際して直面する課題について述べる．

まず，数値解析モデルを支える計算機性能の進化の見地から優位性を述べる．表 5.1 を見ると，1990 年代及び 2000 年代はモデルの計算負荷の小さい，記号モデルや統計モデルを利用した最適化が行われている．実験に用いる加工機の性能に適合した記号モデルや統計モデルを組み立てるためには実機における実験が必要である．20 ～ 50 年程度の時間経過では材料の組成や工具の性能は大きく変化しないことから，実機実験にかかる時間はほとんど短縮されないが，数値シミュレーションにかかる時間は指数関数的な計算機性能の向上により大幅に短縮される．ムーアの法則による半導体上の素子の集積密度が計算速度と比例すると仮定すると，20 年間で計算機性能は約 1000 倍向上する．そのため，時代が進むにつれ数値解析モデルを用いた最適化の優位性が増すと考えられる．したがって，前

章までの展開と同様に，研削加工条件最適化の問題に対しても，数値解析モデルを活用した最適化のアプローチを適用する．

次に数値解析がモデル持つ能力由来の優位性を述べる．2.1 節の議論のように，数値解析モデルでは複雑な関数を表現可能である．例えば，数値解析モデルでは，ワークと砥石の回転数の最小公倍数が小さいときに，同じ砥粒がワーク上の同一の箇所を加工して表面性状が悪くなるという現象を再現するようなモデルを組むことは容易である．一方で，統計的モデルでは十分にサンプル数を多くしないと，回転数の最小公倍数が小さいことによる表面性状の悪化の規則性を見抜くことができない．また，応答曲面による方法では，曲面の自由度を落としているため，曲面の表現能力の限界により，近似がうまくできないことも考えられる．また，2.1 節の議論のように数値解析モデルには，一度妥当なモデルを組むことができれば，加工対象や最適化すべき制御パラメータの組み合わせが変わってもモデルを転用できるという利点がある．加工物の寸法を変更したり，加工パスの自由度を増やしても，現象を再現する数値解析モデルの構成は全く変更せずに，計算機上で最適化を行うことができる．統計的モデルを使った最適化では，過去に統計的モデルを組み立てた際に変更したパラメータしか変更できない．例えば，2 個のパラメータを複数の水準に変更して取得したデータから作成した統計モデルは，3 個のパラメータを変更するような最適化問題にすぐさま転用することができない．

一方で，数値解析モデルを用いたアプローチには以下の二つの研究課題が存在する．

1. 十分な計算速度で加工を模擬可能な数値解析モデルが存在せず最適化が困難である．
2. 実機と数値解析モデルとの間に差が存在する．

課題 1 は数値解析モデルの計算速度に関するものである．研削加工の数値解析モデルにおいては 3D-CAE のように粒子・格子間の関係性を記述した式をもとに加工を模擬する．ここで粒子・格子とは砥石上の砥粒とワークの表面の形状をあらわす点群である．一般的な研削加工は砥粒がランダムな配置であることにより，多数の砥粒がワークに当たって表面が均一に削られる．そのため加工を模擬するためには多くの砥粒と多くのワーク点群を登場させなくてはならず，計算コストがかかる．最適化における反復的なサンプリングを行うため，肝心な部分の精度を下げずに済むような計算量低減手法を考案し実装する必要がある．

課題 2 は数値解析モデルでは模擬できない実現象があるために生じる誤差に関するものである．一部課題 1 とも関係する部分であるが，材料と材料の接触は正確なモデリングが難しく，多くの近似をせざるを得ないため誤差が生じる．このような誤差が存在すると，数値解析モデルにおける最適解と実機における最適解が異なるようになる．気象予報分野におけるデータ同化 [26] のように，数値解析モデルで計算できる関数形状と，実機のデー

タとを共に利用して最適化を進めていくことで、実機における実用的な解を少ない実機の加工回数で求められると考えられる。

#### 5.1.4 本章における研究の副目的

以上の議論から、本章の円筒研削加工の加工条件最適化の取り組みに対して、以下のような研究の副目的を設定した。括弧内の文字は、整理のために設定した研究目的の ID を表す。

- 研究課題を解決するための手法を提案し実行する。(CG-A)
  1. 円筒研削加工の特性に注目し簡略化を施した数値シミュレータを開発する。(CG-A-1)
  2. 実機と数値解析モデルとの差を補正しながら実機の制御パラメータを最適化する。(CG-A-2)
- 実際の機械システムのパラメータ最適化まで行い、各問題の最適化が現代の技術の組み合わせで解決が可能か否かについて、デモンストレーションを行うことにより検証する。(CG-B)
  1. 計算機上における目的関数の最適化の可否や技術的妥当性を検証する。(CG-B-1)
  2. 提案手法によって実機向けに最適化されたパラメータの性能を評価する。(CG-B-2)

## 5.2 課題達成のためのアプローチ

本節では 5.1 節で述べた研究課題を達成するためのアプローチ (達成方針) について述べる。なお本節は、研究目的 ID:CG-A の「研究課題を解決するために手法を提案し実行する。」の提案部分である。

先の、5.1 節で述べた研究課題 1, 2 は次のようなものであった。

1. 十分な計算速度で加工を模擬可能な数値解析モデルが存在せず最適化が困難である。
2. 実機と数値解析モデルとの間に差が存在する。

本章では、これらの課題へのアプローチを 3 章で述べたメタな最適化システムの観点から述べる。

### 5.2.1 扱う最適化問題の特性

アプローチについて述べる前に、前提条件となる、扱う最適化問題の特性について述べる。本章で扱う最適化問題において、メタな最適化システムの部品や、非メタな最適化システムの部品は以下のような特性を持つ。

- ブラックボックスな最適化アルゴリズムは数値解析モデル、目的関数、パラメータ定義域の情報の 3 つが与えられれば、人の手を必要とせずに最適化を行うことができる。
- 非メタな最適化における目的関数が自明であり、洗練をさせる必要がない。
- 加工条件 (制御パラメータ) を設定し数値解析を行うと、目的関数のスカラー値のみならず、表面形状の推移などの加工中の各種物理量を計算することができる。
- 研削加工を模擬するためには多くの砥粒と多くのワーク点群を登場させなくてはならず、数値解析には多くの計算コストがかかる。
- 円筒形状は幾何的に対称性を持つ。
- 円筒研削加工において除去加工が発生する部分は、ワークのごく狭い領域に限られる。
- 数値解析モデルを利用して円筒研削加工の現象模擬を行う際、異なるパラメータ間で共有できる計算結果が存在する。

本研究では、これらの特性を考慮して、研削加工条件最適化のためのメタな最適化システムを構築した。3D-CAE で数値解析が行われて、目的関数が自明である問題に対しては、同様の特性が当てはまり、同様のアプローチが有効であると考えられる。例えば、流体の数値シミュレーションを利用した、流体機械の寸法最適化は同様の問題だと考えられる。

### 5.2.2 メタな最適化システムの動作

本項では、アプローチであるメタな最適化システムの動作について述べる。メタな最適化においては、非メタな最適化システムの構成要素である数値解析モデルを洗練させながら、加工条件 (制御パラメータ)  $x$  を最適化する。この洗練のステップでは、数値解析モデルの精度だけではなく、計算時間も改善される。これは、メタな最適化において精度だけではなく、コストも目標に含まれることに起因する。数値解析モデルの修正と計算時間の低減は 1 対 1 対応ではないため、モデルの精度が悪すぎて最適化の障害になる場合は、計算時間の低減より精度の向上が優先されることもある。

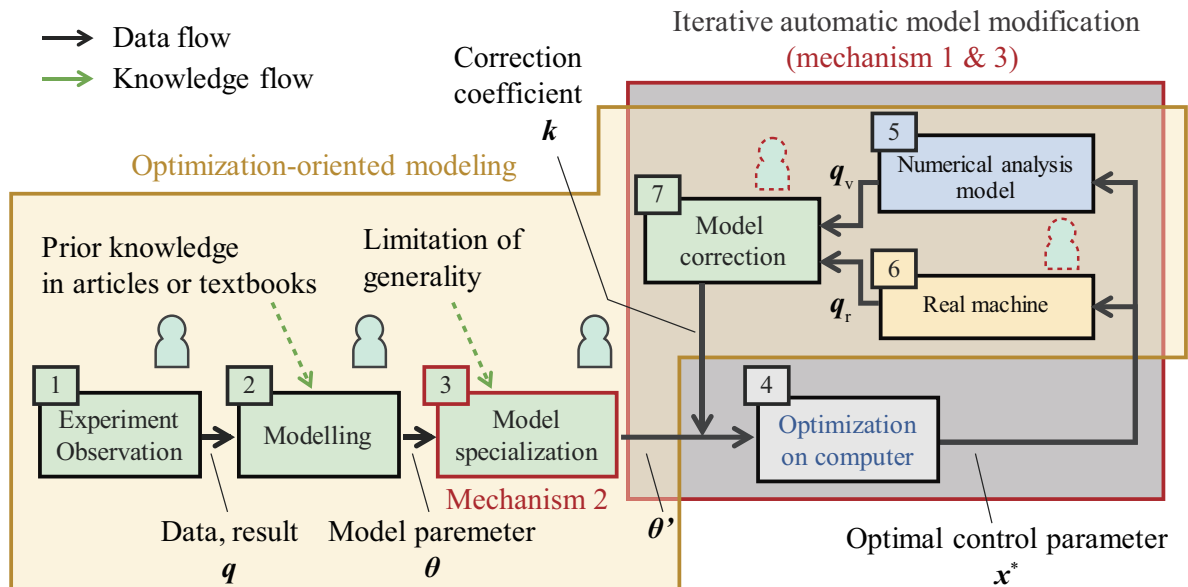


図 5.3 円筒研削加工のメタ最適化システム (番号入り)。

図 5.3 は 3 章でも述べたメタな最適化システムの全体像である。これは、メタな最適化システムにおける機構 2 の「数値解析結果の再利用機構」と機構 1 と機構 3 を組み合わせた「補正係数を利用した自動的かつ反復的なモデル修正」に基づいて、非メタな最適化システムを構築・改善していくものである。図中の矢印や四角の色が指し示すものは 4.2 節の図 4.6 と同様である。図 5.3 中の黒色の矢印は電子的なデータの流れ、緑色の矢印は人間の持つ知識の流れを表す。図 5.3 の右上の青色の四角と、黄色の四角はそれぞれ数値解析を行うプロセス、実機実験を行うプロセスを表す。また、緑色の四角は、設計者(人間)が関与したプロセスを表す。本章で扱う円筒研削加工の最適化問題においては、目的関数の定義が自明で改善を行う必要がないため、4.2 節の図 4.6 のような目的関数の改善のプロセスは存在しない。

ここからは、メタな最適化システムのそれぞれのプロセスについて内容と効果を説明する。図 5.3 のように、メタな最適化は 7 個のプロセスから構成される。

■1. 実験と観察 このプロセスにおいては、対象となる機械の数値解析モデルを構築するために、機械の挙動を観察する。本章で扱うような工作機械は、加工現象を引き起こすために、挙動が既に知られている機構を組み合わせで設計されることが多く、そのような場合はこのプロセスを省くことができる。

■2. モデル化 このプロセスにおいては、プロセス 1 の観察結果や、教科書や論文などの蓄積された知見を用いて機械で起こる現象を数値解析モデルに落とし込む。このモデル化プロセスの内部では、人間の手によって、モデルの作成、モデルの挙動評価、モデル修

正が繰り返し行われる。挙動評価においては、数種類のパラメータにおいて、作成した数値解析モデルを回し、モデルの挙動が意図したものであるかどうかを評価する。具体的には、数値解析モデルが発散しないか、実用的な時間で計算が可能かなどの観点からモデルを評価する。発散する場合や計算時間がかかる場合は、その原因を突き止めて、原因を解消したり、別の手法を試すことで好ましいモデルへと修正する。モデル修正の際には、モデルの中間出力とプロセス 1 で得た実際のデータとの一致を確認することや、ユニットテストなどの、ホワイトボックス的なやり方で修正が行われる。ホワイトボックス的な修正は、1 試行から複数の知見を取り出すことができるため、修正効率が高い。本章で取り扱う円筒研削加工の最適化においては、円筒研削加工の幾何的な特性、対称性に注目して計算を簡略化した数値解析モデル (数値シミュレータ) を構築している。

■3. モデルの特殊化 このプロセスでは、既にある数値解析モデルの一般性を制限することにより、現象模擬にかかる計算量を低減する。最適化問題によっては、プロセス 2 のモデル化の時点で計算量低減が実現されていたり、モデルの特殊化が行えない場合がある。そのような場合はこのプロセスは行われず、プロセス 2 で作成したモデルをプロセス 4 以降でも用いる。本研究が提唱しているモデルの特殊化においては、あるパラメータで計算した数値解析結果を他のパラメータに対して利用し、計算量の低減を行う。このプロセスは図 4.6 にもあるように、機構 2 の数値解析結果の再利用に対応している。本章で取り扱う円筒研削加工の最適化においては、加工条件が特定の条件を満たすと、ワーク表面の形状が同一になるという事実に基づいて、表面形状計算の再利用を行っている。1.1.3 項で紹介した文献 [46] では粒子法数値シミュレータによる流体計算を **Regression forest** という統計的手法を利用することにより簡略化している。このような簡略化もモデルの特殊化のプロセスとして利用可能であると考えられる。

■4. 計算機上での最適化 このプロセスにおいては、数値解析モデルを用いた最適化を行う。このプロセスでは人間の介在がなく、数値解析モデルと目的関数を利用して、計算機上で自動的に最適化が行われる。計算機上での最適化は、人間が介在せず、実機の実験も行わないため高速に実行することができる。また、この最適化においてはブラックボックス最適化手法を利用してブラックボックス的に最適化が行われる。

■5. 最適化されたパラメータの数値解析モデルにおける評価 このプロセスにおいては、最適化されたパラメータを数値解析モデルを利用して評価する。評価の結果としては、目的関数の値だけではなく、中間出力も一緒に出力される。人間による加工条件の最適化が難しい問題においては、最適化されたパラメータが、プロセス 2 や 3 で人間の設計者により試された数種類のパラメータから大きく異なっていることが期待される。大きく異なっているため、プロセス 2 と 3 で試した数種類のパラメータでは発生しなかったような数値

解析モデルの挙動の発生が期待される。また、試された数種類のパラメータと比べ、目的関数の値が大きくなることが期待される。

■6. 実機における評価 このプロセスにおいては、最適化されたパラメータを実機を利用して評価する。実機を利用するため、プロセス 5 に比べて精度は高くなるが、1 試行あたりのコストが高くなる。プロセス 5 と同じく、目的関数の値だけではなく、中間出力も一緒に出力させることができる。ただし、プロセス 5 の場合とは異なり、中間出力を出力させるためには、新たにセンサを取り付ける必要性から、追加でコストがかかることもある。

■7. モデルの補正係数の計算 このプロセスにおいては、プロセス 5 と 6 で得られた数値解析モデルと実機の出力値を比較し、モデルを補正するための値である補正係数  $k$  を計算する。補正係数の計算の必要性は次段落の「補正係数を利用した自動的かつ反復的なモデル修正」機構の詳細とともに述べる。

■補正係数を利用した自動的かつ反復的なモデル修正 このモデル修正機構は、3 章で紹介した機構 1 と 3 を組み合わせたものであり、プロセス 5 と 6 で得られた結果を利用してモデルを修正するものである。この機構は図 4.6 の右側にある四角の内側の 4 つのプロセス (4, 5, 6, 7) から構成されている。

数値解析モデルと実機との誤差が存在すると、数値解析モデルにおける最適解と実機における最適解が異なるようになる。この機構では、数値解析モデルを実機の少数のデータを使って補正し、モデルにおける最適解と実機における最適解が近づくようにする。加工の持つ非線形性や、加工に対する複数の要求の存在により、研削加工を評価する目的関数には図 5.4 左グラフのように非線形性や多峰性が現れると考えられる。2.1 節の議論において、数値解析モデルは複雑な関数を表現できると述べた。数値解析モデルが実機における加工の非線形性を模擬していれば、数値解析モデルの出力を利用して計算される目的関数にも非線形性や多峰性が現れると考えられる。この機構では、図 3.13 における  $k_{in}$ ,  $k_{out}$  のような補正係数によりモデルの入出力を補正し、数値解析モデルの関数形状を補正する。数値解析モデルが再現する非線形性や多峰性といった関数形状の情報を損なわないまま数値解析モデルの補正を行うことで、モデルにおける最適解と実機における最適解が近づくようにする。この機構では、最適化されたパラメータを実機実験に利用して補正係数の計算を行い、数値解析モデルを修正し、再び最適化を行うという一連の動作を繰り返す。これにより、最適パラメータを求めながら、最適パラメータ周辺における数値解析モデルの精度を向上させていく。

また、本研究が提案している数値解析モデルを活用した最適化の対象となるような最適化問題では、実機の実験コストは数値解析を行った場合のコストより高い。コストが高いため実験データをあまり多くとれない場合は、数値解析モデルの誤差の補正時の自由度を

高くしてしまうと、過学習が起こり数値解析モデルの再現した関数形状を著しく損なってしまう。過学習を防ぐためには、補正係数の種類は少なくする必要がある。

本論文における研削加工機を利用した実験では、人間が自らの手によって計測や補正係数計算を行っておりその分コストがかかっているが、機械のコネクテッド化が進むことにより、このコストは低減されると考えられる。

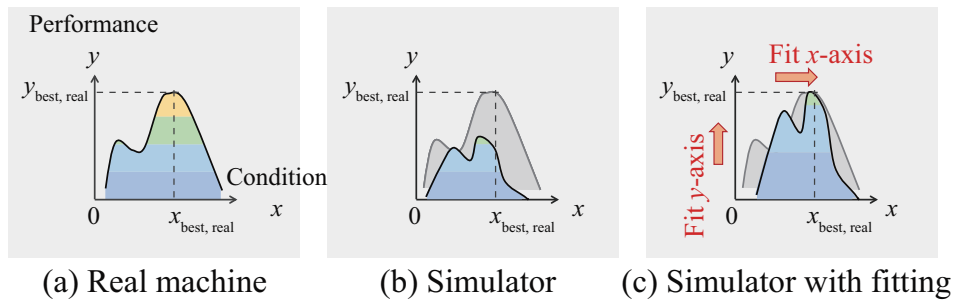


図 5.4 実機の少数データを用いた数値解析モデルの補正。

以降は、円筒研削加工の加工条件最適化問題に対し、本節で紹介したメタな最適化を適用した結果について述べる。続く 5.3 節において数値解析モデルと目的関数の完成形について述べ、5.4 節において計算機上での最適化と数値解析モデルにおけるパラメータの評価について述べ、5.5 節において実機におけるパラメータの評価について述べる。

## 5.3 高速動作する数値解析モデルの構成

本節では現象の模擬のために利用した数値解析モデルの詳細について述べる。また、追実験を容易にするため、具体的な実装方法についても述べる。

本章の検証においては、実機実験を行うために、図 5.5 に示す東京大学生産技術研究所千葉実験所にあるアンギュラ型円筒研削加工機 RF32 (三菱重工業株式会社製) を利用する。そのため、数値解析モデルで模擬する対象もアンギュラ型の円筒研削加工機と設定した。一般的な円筒研削加工機は砥石が円筒型であり、砥石の軸とワークの軸が平行になるように設計されている。一方で、アンギュラ型円筒研削加工機は、砥石が 2 つの円錐台を組み合わせた形状をしており、砥石の軸とワークの軸が交わるように設計されている。これによりアンギュラ型ではワークの円筒面と段付き部が同時に研削可能となっている。

### 5.3.1 現象模擬のための数値解析モデル (数値シミュレータ)

本項では円筒研削加工を模擬するため数値シミュレータの動作原理や、高速計算のための計算簡略化手法について述べる。研究を進めるにあたっては、置く仮定が少ない汎用的な数値シミュレータを開発し、その後汎用的な数値シミュレータから模擬できる状況を限





図 5.5 利用した円筒研削盤 RF32 (三菱重工業株式会社製).

定して高速化した数値シミュレータを開発した．汎用的な数値シミュレータを汎用シミュレータ，模擬できる状況を限定した高速な数値シミュレータを高速シミュレータと呼称する．本節ではそれらを順を追って説明する．

#### 汎用シミュレータの動作原理

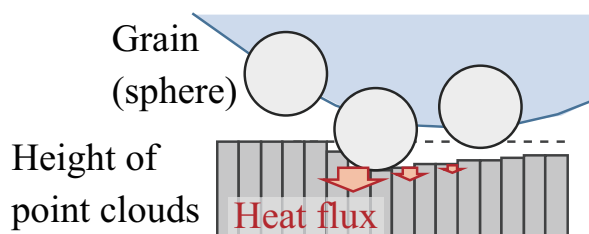


図 5.6 砥粒と被削材の接触を模擬したシミュレーション．

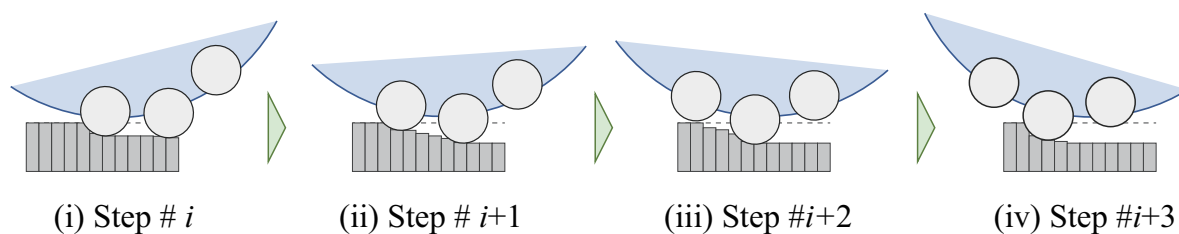


図 5.7 砥粒と表面点群の時間変化．

数値シミュレータでは図 5.6 のように砥石上に存在する砥粒とワークとの間の接触を計算する．本数値シミュレータにおいて，砥粒は球形であり，ワーク表面は点群から構成される．1 つの時間ステップの間に砥石とワークが並進移動・回転し位置関係が変化する．

各時間ステップにおいて、砥粒と点群の接触判定が行われ、図 5.6 のように接触量の分だけ点群の高さが減少する。この減少分が除去量に相当する。図 5.7 は回転する砥石上の砥粒により表面点群が加工される様子を示している。図 5.7 内の各図は各時間ステップの状態を表しており、砥粒の移動に伴い、接触が発生し、表面点群の高さが減少していることがわかる。また、引張強さと除去量に比例した発生熱が計算される。実際の研削加工に用いられる砥粒は理想的な球形状ではなく角ばった形状をしているが、本シミュレータにおいては計算の簡易化のため砥粒形状を球形状としている。先行研究においても砥粒を球形状で近似している例が存在する [106]。

点群と球砥粒との間の接触判定には高速化のために行列計算を利用している。本シミュレータを実装している技術計算言語 Matlab では行列の積和計算の実行が高速化されており、積和計算を高速に行うことができる。

接触計算においては、点群を構成する各点の 3 次元座標が行列に格納されており、球の中心座標とのずれから接触量が計算される。これらの計算を 1 行の間に行い Matlab の変数空間を経由しないことにより、データ転送にかかる時間を削減している。

数値シミュレータの各種定数は、現象の模擬精度を担保をするために、もしくは加工機の諸元をそのまま用いて次のように定めた。

- 時間ステップ  $\Delta t = 100 \mu\text{s}$  (シミュレーション精度から決定)
- 点群密度  $\sigma_{\text{pc}} = 1,000 \text{ mm}^{-2}$  (シミュレーション精度から決定)
- 砥粒直径  $r_{\text{grain}} = 0.300 \text{ mm}$  (加工機の諸元から決定)
- 砥粒存在面密度  $\sigma_{\text{grain}} = 0.1 \text{ mm}^{-2}$  (加工機の諸元から決定)
- 砥石の短直径  $r_{\text{short}} = 220$  (加工機の諸元から決定)
- 砥石の長直径  $r_{\text{long}} = 260$  (加工機の諸元から決定)
- 砥石幅  $w_{\text{wheel}} = 69.3 \text{ mm}$  (加工機の諸元から決定)
- 砥粒個数  $N_{\text{grain}} = 1,205,257$  (加工機の諸元から決定)
- 砥石軸とワーク軸のなす角  $\theta_{\text{angular}} = 30 \text{ deg}$  (加工機の諸元から決定)

#### 汎用シミュレータのための多階層の接触判定

汎用的なシミュレータでは愚直な接触判定の計算結果を損なうことなく高速化をするために、接触判定アルゴリズムの工夫を施した。本節においてはそれらの工夫について述べる。

上述のように、本研究で扱うような円筒研削加工のシミュレーションにおいては砥粒数と表面点群数、時間ステップ数が大きくなる。各砥粒と各表面点との愚直な接触判定をすると、 $O(N_{\text{grain}}N_{\text{pc}}N_{\text{step}})$  の計算量が必要になり、最適化への利用が難しくなる。また、精度維持のためには時間ステップ数  $N_{\text{step}}$  を削ることは難しく、砥粒数と表面点群

数との積である  $N_{\text{grain}}N_{\text{pc}}$  の部分の計算量を減らす必要がある．前述の積が分解でき、 $O((N_{\text{grain}} + N_{\text{pc}})N_{\text{step}})$  になったとしても大きいので、 $N_{\text{grain}}$ ,  $N_{\text{pc}}$  自身も小さくする必要がある．そのような背景のもと次のような 4 つの計算量低減手法を考案した．

1. 加工発生領域に侵入していない砥粒と点群を判定から除外．
2. ワーク上の注視領域内の点群のみ接触判定を行う．( $O(N_{\text{pc}}N_{\text{step}}) \rightarrow O(\gamma N_{\text{pc}}N_{\text{step}})$ )
3. 砥石表面をメッシュに分割しメッシュ単位で加工発生領域に入っているかを判定．( $O(N_{\text{grain}}N_{\text{step}}) \rightarrow O(N_{\text{step}})$ )
4. メッシュごとに所属する砥粒を生成．( $O(N_{\text{grain}}^2) \rightarrow O(N_{\text{grain}})$ )

ここで  $\gamma$  はワーク全体の表面積に対する注視領域の面積の割合を表す．図 5.8 に各低減手法の概略を示す．この 4 つの手法は円筒研削加工の数値シミュレーションのみならず、対象形状を持つ歯車を研削する歯車加工の数値シミュレーションにも適用可能である．また、工具の各点とワーク上の点群との接触判定を行うような、切削加工を模擬するための数値シミュレーションにも適用可能である．

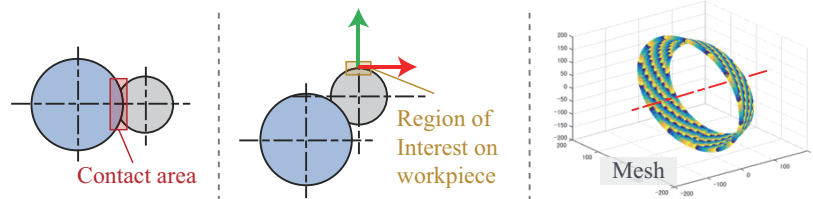


図 5.8 汎用シミュレータ高速化のための多階層の接触粒子絞り込み．

■1. 加工発生領域に侵入していない砥粒と点群を接触判定から除外 この手法は砥粒と点群の総当たりでの接触判定を避け、計算量を削減するものである．グローバルな座標系において、砥石は  $xy$  平面上を運動し、砥石とワークの中心軸は  $z$  軸に直行している．また、砥石とワークの中心軸の高さの  $z$  座標は同一である．そのため、砥石とワークの外周のほとんどの部分は接触が起こらず、図 5.8 左赤枠の部分でしか接触が発生しない．よって、この赤枠の中に入っている砥粒とワーク表面点群のみ接触判定を行うことで、計算結果を変えずに接触判定の計算量を減らすことができる．この赤枠の部分を加工発生領域と呼称する．加工発生領域の導入により、およそ砥石とワークの外周に占める加工発生領域の割合の分だけ接触判定の回数が減る．加工発生領域の導入にあたり、接触判定回数は減らすことができるものの、新たに表面点群と砥粒が加工発生領域へ侵入しているかの判定をする必要が出てくる．表面点群と砥粒の加工発生領域への侵入を愚直に判定すると、それぞれ  $O(N_{\text{pc}}N_{\text{step}})$ ,  $O(N_{\text{grain}}N_{\text{step}})$  の計算量が必要になり、これらは依然として大きい．2, 3 の実装は加工発生領域への侵入判定にかかる計算量をさらに減らすためのものであ

る。加工発生領域の高さは砥石の送り速度によって変化する。高さを十分大きな固定値としても良いが、送り速度によって適応的に変わる実装にすることにより絞り込みを効果的に行うことができる。

■2. ワーク上の注視領域内の点群のみ接触判定 この手法はワークの対称性に注目して、ワーク全周における加工結果をある一部分の加工結果に代表させるものである。円筒研削加工において、ワークは円筒状であり、周方向の各点における加工結果はおおよそ同様になる。そこで、図 5.8 中央のように、ワーク上の一部分に注目 (この部分を注視領域と呼ぶ) して、ワーク全体を代表させる。注視領域をワークの一部に限定しても、前述のような回転数の最小公倍数が小さい時に発生する砥石表面形状の転写は模擬される。注視領域は一個だけではなく複数個とることができ、複数個とることにより全体の加工を模擬することに近くなる。ある注視領域の加工結果が他の注視領域に対して影響を与えないのであれば、注視領域ごとに加工の模擬は並列的に行うことが可能である。ワーク全体の振動や砥石の損耗など、ある注視領域における加工結果が他の加工に影響することを考慮する場合、並列化が難しくなる。

■3. 砥石表面をメッシュに分割しメッシュ単位で加工発生領域への侵入判定 この手法は加工発生領域領域に侵入する砥粒を高速に絞りこむためのものである。そのために、図 5.8 右のように砥石表面の領域をメッシュに分割する。メッシュを使うことにより、砥粒一粒一粒に対する加工発生領域侵入判定を迂回することができる。実装としてはまず、メッシュが加工領域に侵入しているかを判定する。次に、侵入しているメッシュに所属する砥粒を取り出し、接触判定を行う。また、砥石の回転角を使うことにより、加工発生領域へ侵入しているメッシュの番号を  $O(1)$  で計算することができる。メッシュが粗い (メッシュの数が少ない) と、この判定時点で残る砥粒の数が多くなるため、砥粒の加工発生領域の侵入判定の計算量が大きくなる。一方でメッシュが細かい (メッシュの数が多) と、この判定時点で残る砥粒の数は小さくなり物理的な下限値に近づくが、メッシュの数が増えるため、メッシュ扱いにかかる計算量が大きくなる。このようにメッシュの粗さと計算量との間にはトレードオフが存在する。

■4. メッシュごとに所属する砥粒を生成 この手法は砥粒の所属するメッシュを求める際の処理に関するものである。1, 2, 3 の手法とは違いシミュレータの時間発展開始前に、一回行えばよい。3 を行うためには各メッシュに属する砥粒の情報を知っている必要がある。砥石上の全ての砥粒を生成してから、所属するメッシュを求めると  $O(N_{\text{grain}})$  の時間がかかる。また、長大な配列から 1 つ 1 つ砥粒の情報を取り出すと時間がかかるため、あるメッシュに所属砥粒の情報はメッシュごとに行列としてまとめられている必要がある。メッシュごとに分ける際に、ソートなど検索を利用する操作を行うと  $O(N_{\text{grain}} \log N_{\text{grain}})$

の計算量が必要になる．またこの計算量は行列計算の恩恵を受づらく，減らすことが難しい．それに対する策として，メッシュの切り方が決まった時点で，メッシュごとに砥粒を生成することで，計算量オーダーを  $O(N_{\text{grain}})$  に減らし，行列計算の恩恵も受けられるような実装にすることができる．

#### 高速シミュレータのための転写形状の事前計算

ここからは汎用的なシミュレータではなく，利用状況を限定した高速なシミュレータについて述べる．以降の議論のため，汎用シミュレータにより加工を一回模擬するために必要な時間を  $T_{\text{general}}$  とおく．

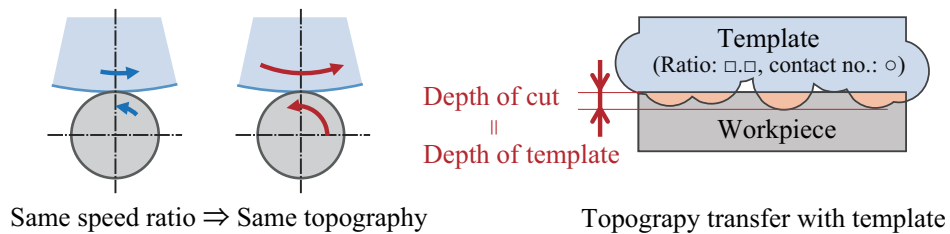


図 5.9 事前計算される転写形状テンプレート．

図 5.9 左のように砥石とワークの回転速度比が同一の時，幾何的な転写形状は同一という事実がある．ワークが一回転する間に砥石が回転する角度を  $\theta_{\text{wheel}}$ ，ワークが一回転するのにかかる時間を  $t_{\text{workpiece}}$  とすると，式 (5.1) の関係式が成り立つ．回転速度比が一定の場合，右辺が一定となるため，左辺の  $\theta_{\text{wheel}}$  も一定となる．

$$\theta_{\text{wheel}} = \omega_{\text{wheel}} * \frac{2\pi}{60} * t_{\text{workpiece}} = \frac{\omega_{\text{wheel}}}{\omega_{\text{workpiece}}} * 2\pi \quad (5.1)$$

言い換えると回転速度比が同じ場合，ワーク上の注視領域が一回転して加工発生領域から退出して再び侵入してくる間に，砥石も同じ角度だけ回転する．そのため回転速度比が同じ場合，あるワークの注視領域に対して接触する砥粒の組み合わせも同じになるため，幾何的に転写される形状が同じになる．

高速シミュレータではこの事実を使い，ある回転速度比において砥石がワークを削り取る形状を記録しておき，同じ回転速度比のシミュレーションを行うときにその記録を用いて高速に加工を再現する．記録の仕方としては，ワークが加工発生領域に侵入して退出するまでに砥粒が走査する空間の形状を記録する．この形状を記憶したものをテンプレートと呼ぶ．そのため，ワークが1周するたびにテンプレート1枚が計算される．図 5.9 右は1次元でのテンプレートの概略を表す．実際のシミュレーションでは一つ次元の多いテンプレートを利用している．図において，青色の領域が砥粒が走査した部分をあらわすテンプレートで，灰色の領域がワークをあらわす．そして，赤色の領域がテンプレートとワー

クが接触して削られる部分である。砥石の送り速度はワークが一回転している間にワークがどれくらい進むかを表しており、テンプレートを使った計算においては、送り速度はテンプレートの押し付け量と等しくなる。

図 5.10 はテンプレートを連続的に押し付けることによる加工の模擬の様子を表している。ワークが 1 周して注視領域が砥石と接触するたびに、テンプレートの押し付けが行われる。図 5.10 の縦方向は時間の経過を表しており、下に行くほど加工の終盤であることをあらわす。図中右列のワーク表面形状の推移のようにテンプレートによりワークの表面が次々と除去されていく。テンプレートの形状は基本的には各回の接触で異なるが、ワークと砥石の回転速度比が 1 の時にのみ、すべてのテンプレートの形状が等しくなる。

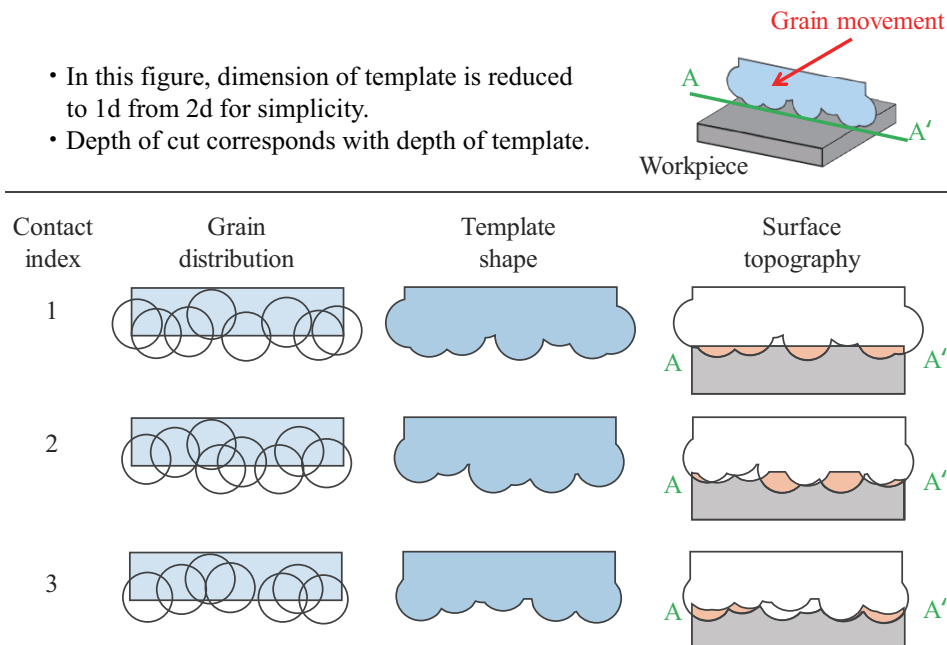


図 5.10 テンプレートの連続適用による加工の模擬。

テンプレートの事前計算による利点について述べる。テンプレートを事前に計算しておけば、どの砥粒がワークにぶつかるかという接触判定を回避できるのみならず、注視領域が加工発生領域に入っている間に加工される領域を 1 ステップで計算できるようになる。テンプレートを使うにあたり、同一速度比では砥石上の砥粒の配置を同一にせざるを得ない。砥石上の砥粒の配置を同一にすることを砥石生成の乱数シードを固定すると表現する。

現実の研削加工においても砥粒の配置はランダムに配置されている。そのため、砥石の乱数のシード値が変わっても加工結果に大きな影響を及ぼさないと考えることが妥当である。言い換えると、諸元や摩耗状態の同じ砥石を 2 種類用意して加工を行っても、表面の品質には質的影響がないと考えることが妥当である。したがって、砥石生成の乱数シード

値 (具体的には砥粒の配置) を固定してしまっても加工結果に大きな影響はないと考えられる. なお, この議論はそれぞれの砥粒の座標が綿密に設計されて決められているような砥石では成り立たない. 実際の高速シミュレータの実装では, 砥粒の偏りの影響を少なくするため一つの速度比に対して 3 種類の乱数シードのテンプレートを用意している.

テンプレートを利用しない場合は毎回汎用シミュレータで行う接触判定をしなくてはならない. 最適化において計算する加工条件数を  $N_{\text{total}}$  とすると, テンプレートを利用しなかった場合に接触判定にかかる計算量は  $N_{\text{total}} T_{\text{general}}$  となるが, テンプレートを利用した場合は  $k T_{\text{general}}$  となる. ここで  $k$  は定数で, 同一速度比に対して用意される乱数シードの数である. 前の段落のように本章の実装では  $k = 3$  としている. テンプレートを利用すれば最適化のサンプル数が増えても, 最適化にかかる時間がサンプル数に比例して増加しない.

また, 砥石とワークの回転速度比は連続的に存在するが, これを量子化することによりテンプレートの生成数を定数に抑えている. 量子化の幅を示す速度比の公比は 1.1 とした.

ここからは, テンプレートを利用しても模擬可能なものとテンプレートを利用することにより模擬できなくなるものを整理する. 模擬可能なものとして特筆すべきものは, 加工中に送り速度が変わる点, 砥粒配置のランダム性によって毎回少しずつ加工結果が変わる点という 2 点である. 送り速度の変化はテンプレートの押し付け量を変えることにより模擬が可能である. 砥石配置のランダム性による加工結果の微小な変化は, テンプレートのシード値を複数用意することによって模擬が可能である.

一方模擬できなくなるのは, 各砥粒の除去量や各砥粒の発生熱である. テンプレートを作成する際にどの砥粒がどれくらいの時間の間にワークを除去したかの情報を消してしまっており, 模擬が不可能になる.

また, 原理上模擬は可能だが計算量の都合上模擬が難しくなるものも存在する. 砥石の移動自由度が 2 自由度になると, 精度を維持するために必要なテンプレートの数が増大してしまう.

### 5.3.2 ソフトウェア上での実装

これらのシミュレータはすべて技術計算言語 Matlab 上で実装した. メッシュの分割数, 各種定数は数パターン試行して計算速度が高くなるものを選んだ.

転写形状テンプレートは最適化以前の段階で計算した. たまたま砥粒配置が偏ってしまい指標へ悪影響をおよぼすことがないように, 同じ速度比に対して 3 種類の砥粒配置を用意した ( $k = 3$ ). 回転速度比は区間  $[\frac{\omega_{\text{wheel,min}}}{\omega_{\text{workpiece,max}}}, \frac{\omega_{\text{wheel,max}}}{\omega_{\text{workpiece,min}}}]$  に収まる. ここで,  $\omega_{\text{workpiece,min}}$ ,  $\omega_{\text{workpiece,max}}$ ,  $\omega_{\text{wheel,min}}$ ,  $\omega_{\text{wheel,max}}$  はそれぞれワーク最小回転数, ワーク最



大回転数, 砥石最小回転数, 砥石最大回転数を示し, 本章においては 40, 200, 400, 1000 である. そのため回転速度比は区間  $[2, 25]$  内にあるものに限られ,  $r_0 = 1$  を初項とした公比 1.1 の等比数列の  $r_8 = 2.144$  から  $r_{33} = 23.225$  までは存在する. テンプレートは合計で  $(33 - 8 + 1) * 3 = 78$  組計算された. テンプレートの事前計算には 3 台のワークステーションを利用し, 半日程度の時間を要した.

## 5.4 計算機上の最適化実験

本節では計算機上で行った数値解析モデルをベースとした最適化の実験及び結果について述べる.

計算機上での最適化実験の目的は以下の通りである.

- 人間が実用的な解をわかっている問題において最適化を行い, 数値解析モデル及び最適化システムの妥当性を評価

人間が実用的な解をわかっている最適化問題において人間と異なる解が導かれた場合, 数値解析モデルによる実現象のモデル化に無視できない誤差があるか, 加工を評価するための目的関数が人間の評価基準を反映できていないという可能性が高い. 誤差の存在, 反映の可否を素早く洗い出すために最適化実験を行う. また, 最適化実験を行うことにより, 今回の問題において最適化にかかる計算コスト, 時間コストが判明する. 実験の結果として, 数日間最適化のための計算を行っても, 解が収束しないということも考えられうる. 本章の実験においては研削加工を数値解析モデルをベースに最適化することの妥当性も検証する.

まず, 最適化の問題設定として目的関数について述べ, 続いて利用する最適化手法, 最適化実験の結果について述べる. どんな種類のワークをどのように加工すればよいかという加工の目標も目的関数を構成するための情報に含まれる.

### 5.4.1 性能評価のための目的関数

本項では研削加工を評価するための目的関数について述べる. 具体的には, 目的関数を構成する指標や制約条件, 最適化対象のパラメータについて述べる. 制約条件の中に加工の寸法目標も含まれる.

本章の取り組みの目的は数値解析モデルにより研削加工条件の最適化が行えるか確かめることである. そのため研削加工の設定は特殊なものである必要はなく, 一般的なものでよい. 指標や制約条件, 最適化対象のパラメータの決定には, 一般的な加工に用いられるようなものを選ぶため, 表 5.2 に示す先行研究のものを参考した. 研削加工の加工条件最



適化の論文における，登場する指標と制約条件，最適化を行うパラメータを示している．

外部から与えられる加工の設定としては加工の目標寸法，加工物の材質がある．加工対象のワークははじめ直径 50 mm であり，このワークに対して 100  $\mu\text{m}$  の除去を行い，最終的に目標寸法である 49.9 mm を目指す．材料としては軸受鋼の一種である SUJ2 を焼き入れしたものを利用した．SUJ2 は主に，軸受けの鋼球やころ（ローラ）として用いられる．ころの寸法精度や表面性状はベアリングの転がり摩擦や寿命に影響してくるため，ころの製造においては焼き入れ後にも精密な加工が可能な研削加工が用いられる．

最適化対象の制御パラメータと定義域は表 5.2 に示す先行研究を参考に次の 6 つを設定した．図 5.11 は 6 つの制御パラメータを図示するものである．

- 砥石回転数  $\omega_{\text{wheel}}$  [rpm]  $\in [400, 1000]$
- ワーク回転数  $\omega_{\text{workpiece}}$  [rpm]  $\in [40, 200]$
- 中間点時刻  $t_1$  [s]  $\in [0, 10]$
- 中間点座標  $p_1$  [mm]  $\in [0, 0.2]$
- 終了点時刻  $t_2$  [s]  $\in [0, 10]$
- 終了点座標  $p_2$  [mm]  $\in [0, 0.2]$

中間点，終了点とは砥石の位置指定をあらわしており，砥石は  $t_1$  の時刻に座標  $p_1$ ， $t_2$  の時刻に座標  $p_2$  にいることを示す．加工開始時刻である  $t_0$  から  $t_1$  までの期間と， $t_1$  から  $t_2$  までの期間では砥石は等速に移動する．中間点は 1 点である必然性はなく，複数になっても良いが，粗・仕上げ加工が発現しうる最小の点数として 1 点と設定した．砥石座標  $p_1$ ， $p_2$  は図 5.11 に示すような一軸上の点を表している．砥石移動の自由度が 2 次元だと，オシレーション加工やトラバース加工が発現しうるような問題設定になるが，中間点の数と同様に，粗・仕上げ加工が発現しうる最小の自由度として図 5.11 中の  $p$  軸方向のみの移動と設定した．砥石回転数  $\omega_{\text{wheel}}$  とワーク回転数  $\omega_{\text{workpiece}}$  の定義域は実機の加工機の制約から決定した．また，最適化手法により生成されたサンプルが上記の定義域から逸脱すると目的関数の値は最低値の 0 になるようにした．

加工の評価指標としては，先行研究でも用いられる，表面粗さ，加工時間を選択した．今回，最適化には 5.3.1 項で述べた高速シミュレータを用いるため，発生熱は目的関数に含めなかった．高速シミュレータの除去計算においても，一回の注視領域と砥石との接触において除去された体積とかかった時間の計算は高速計算可能である．そのため，シミュレータの速度を大幅に損なうことなく，発生熱を大まかに計算できる．また，次節において実機を含んだ実験を行う都合上，指標は実機での計測が可能である必要がある．発生熱が関係してくる，ワーク表面の焼きの有無やワークへの入熱量，残留応力値は当研究室において計測する技術が未達であり，実機での計測が不可能だと判断した．発生熱の計算，計測が不可能でも，計算機上での最適化の可能性検証には支障が無いと考え，本章の最適

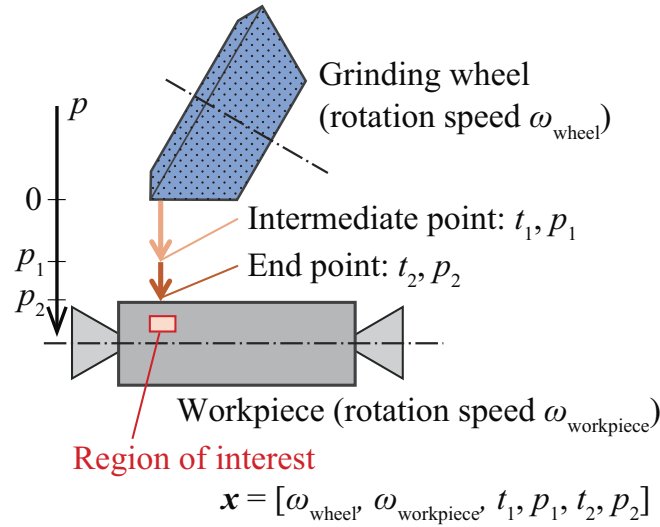


図 5.11 最適化を行う加工設定.

化実験においては発生熱を考慮しなかった.

加工の制約条件としては、以下のものを設定した.

- 中間点・終了点順序制約  $t_1 < t_2$
- 砥石送り速度上限  $\frac{p_1}{t_1} < v_{\text{lim}}, \frac{p_2 - p_1}{t_2 - t_1} < v_{\text{lim}}$
- 砥石逆走禁止  $p_1 > 0, p_2 > p_1$
- 加工の時間制限  $t_2 < t_{\text{lim}}$
- 寸法目標  $d_{\text{target}} \pm d_{\text{tolerance}}$
- 寸法公差  $d_{\text{target}}$

いずれかの制約に違反すると目的関数の値は 0 になるように実装している.  $d_{\text{target}}$  は 0.1 mm と設定した.  $v_{\text{lim}}$  は加工機に損害が加わらないように決めた.  $t_{\text{lim}}$  は 100  $\mu\text{m}$  を削るのには十分長い 10 s と設定した.

図 5.12 のグラフは  $t_1, t_2, p_1, p_2$  から構成される加工パスを時間・砥石位置系で表記したものである. このグラフをもとに最適化の目標, 課されている制約について説明する. グラフにおいて横軸が時間, 縦軸が砥石の位置を表し, 原点と丸 2 つを結ぶ折れ線がそれぞれの加工パスを表している. 3 本の点線は送り速度上限, 寸法公差, 加工の時間制限の 3 つの制約を表している. 制約に違反しないように表面粗さと加工時間を小さくする加工条件を見つけ出すことが最適化における目標である. グラフにおいて, 灰色のパスは加工前にわかる制約違反のパス, 緑色のパスは加工をしてみたらわかる制約違反のパスを表す. 点線で表現されておらず, パスを灰色にする制約としては, 中間点・終了点順序制約, 砥石逆走禁止制約がある. パスが緑色になる制約違反は本章の実験においては, 寸法公差

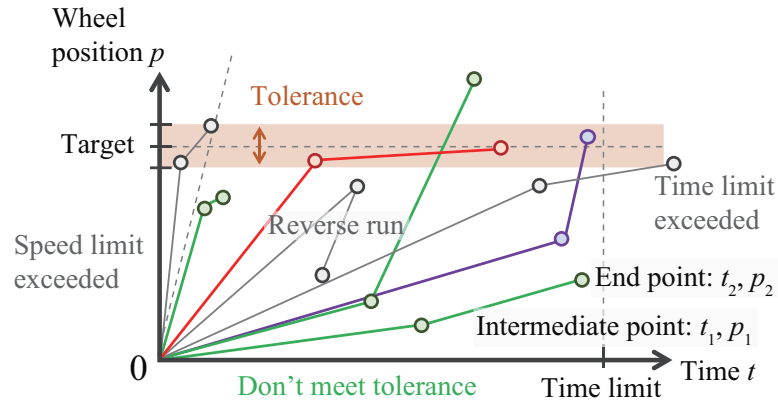


図 5.12 加工パスの時間・砥石位置系での表現.

違反のみである．紫色や赤色のパスは制約違反しなかったパスを表し，色が赤に近いほど目的関数値が高く，色が青に近いほど目的関数値が低いことを表している．

目的関数は式 5.2，式 5.3 のように定義した．式内において  $w_1$ ,  $w_2$  は評価指標の重みを示す重みベクトル  $\mathbf{w}$  の要素である． $c_i$  は制約条件が満たされなかった際に大きな定数になる項であり，制約を満たした場合は 0，制約を満たさなかった場合は 1000 になる．

$$s(\mathbf{x}) = -w_1 \frac{Ra(\mathbf{x}) - Ra_{\min}}{Ra_{\max} - Ra_{\min}} - w_2 \frac{t(\mathbf{x}) - t_{\min}}{t_{\max} - t_{\min}} - \sum_{i=1}^5 c_i(\mathbf{x}) \quad (5.2)$$

$$h(\mathbf{x}) = \exp(s) \quad (5.3)$$

評価指標はお互いに次元が異なるため，正規化を行った上で重みづけ和をとったものを目的関数としている．正規化後の上限・下限の値が特定の区間に収まるように正規化が行われることが多い．この下限から上限の幅と評価指標の重みが，評価指標の重要度付けに与える影響は同一である．例えば，区間  $[0, 1]$  に正規化した指標の重みが 3 であるときと，区間  $[0, 3]$  に正規化した指標の重みが 1 であるときには，それらの指標の変化が目的関数に与える影響は同一である．そのため，指標の重要度は，重みを変更することによって，正規化後の下限から上限の幅からは独立に制御できる．正規化後の上限・下限の値は議論のしやすさのために，表面粗さ場合は精度の基準値をもとに，加工時間の場合はパラメータの定義域をもとに決定した．具体的な値は次のとおりである． $Ra_{\max} = 1.0$ ,  $Ra_{\min} = 0.6$ ,  $t_{\max} = 10$ ,  $t_{\min} = 0$

前述のとおり，指標間の重要性は重みベクトル  $\mathbf{w}$  の調整によって変更することができる．重みベクトルは加工機が用いられる生産現場の要求によって変わってきて，具体的な値は調べるのが難しいため， $\mathbf{w} = [1, 1]$  とした． $\mathbf{w}$  の各要素のバランスによって，目的関数の形状，あるいは複雑性が変化する．ある指標 1 と，その指標 1 に負号を付けただけ

という指標 2 があつたとき，指標 1 と指標 2 を 1 対 1 の割合で重みづけ和をとると，お互いが相殺されて目的関数が恒等的に 0 になる．このように指標と重みによっては目的関数の形状が単純になるということが起こりうるが，今回採用している評価指標はお互い異なる物理量から計算されており打ち消しあうということは考えづらい．そのため，重みのバランスが変わっても目的関数の複雑性は変化しないと仮定して，以降の検証では  $w = [1, 1]$  に固定してすすめる．

式 5.3 の指数関数は，極端に加工の指標が悪い時と制約を満たさなかった時ともに目的関数値が同じ値に漸近してほしいがための実装である．本節で述べた研削加工の設定は次節の実機における研削加工の最適化においても同様である．

### 5.4.2 最適化手法

本項では，目的関数を最適化するための最適化手法について述べる．本章で目指しているものは数値解析モデルを利用した最適化システムである．目的関数やその元になる物理量は数値解析的に求められるため，式の形にはならない．そのため，本章の最適化においてもブラックボックス最適化の手法を用いる．

具体的な最適化手法としては，4.4.1 項で紹介した，進化計算に基づいた手法である CMA-ES を利用した．CMA-ES は定期的に方策を更新するが，異なる方策間でサンプルを持ち越さない簡潔な手法となっている．CMA-ES は同じく 4.4.1 項で紹介した CEM や HSO などと同様に確率的な方策を持つ．CMA-ES のハイパーパラメータは先行研究で推奨されている値を用いる．

図 5.13 に数値解析モデル，目的関数，最適化手法の関係を示す．最適化手法は内部に方策を持ち，方策に従って次に試すべき加工の制御パラメータ  $\mathbf{x}$  を提示する．次に，制御パラメータ  $\mathbf{x}$  に基づいてシミュレータが動作し，加工が模擬される．そして，加工の結果から Ra や加工時間，制約条件を満たしたかが計算され，目的関数の値  $y$  が算出される．最適化手法は  $\mathbf{x}$  と  $y$  の組から  $y$  が高くなるような法則性を見抜き，方策を更新する． $\mathbf{x}$ ,  $y$  の計算と方策更新を繰り返し行うことにより，最終的に高い  $y$  を示す  $\mathbf{x}$  を求める．CMA-ES は貪欲に最適値を目指すだけではなく，周囲への探索も行うため性能が高いと言われている．更新される方策には周囲への探索を行うことも含まれている．

### 5.4.3 目的関数の最適化実験

本項では計算機上での最適化の実験設定について説明する．

最適化の目的関数としては 5.4.1 項の式 (5.3) で示す  $f(\mathbf{x})$  を利用する．最適化対象のパラメータ  $\mathbf{x}$  は 6 次元である．最適化手法としては 5.4.2 項で紹介した CMA-ES を利用す

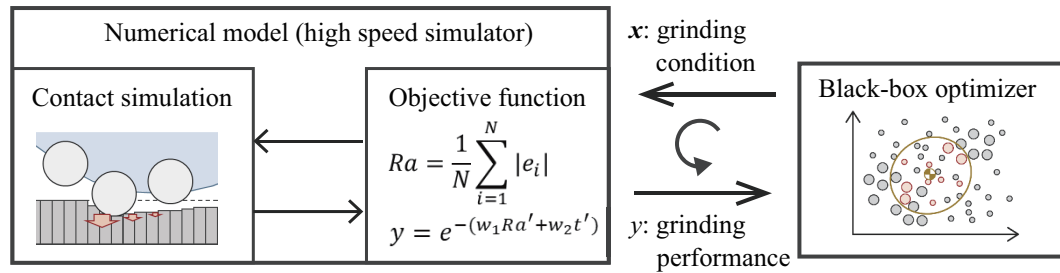


図 5.13 計算機上での最適化システム。

る。CMA-ES の方策更新の間のサンプル数  $\lambda$  は 900 と設定した。

テンプレートを用いた加工のシミュレーションや最適化のための計算は一台のワークステーション (Intel Core i9-7900X @3.3 GHz) 上で行った。また、テンプレートの事前計算には 3 台のワークステーションを利用した。加工の数値シミュレーションや最適化計算は技術計算言語 Matlab 上で実装されている。

計算機上での最適化では取得サンプル数の上限を 20,000 サンプルと設定した。20,000 サンプルの模擬には 1 時間程の時間を要する

#### 5.4.4 結果

本項では、計算機上での最適化結果を示す。

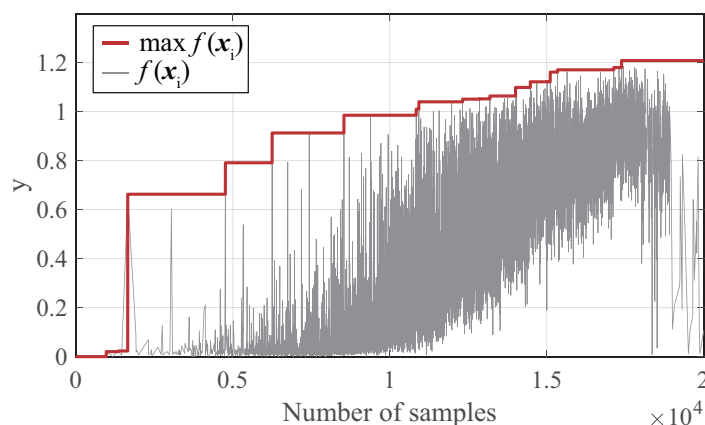


図 5.14 計算機上での最適化の学習曲線。

図 5.14 は最適化の学習曲線である。横軸が取得したサンプル数で、縦軸が目的関数の値を示す。灰色の線が各サンプルの目的関数の値を表し、赤線が各サンプルを取得するまでの最高値を結んだものである。灰色の線は制約を満たせずに 0 となったサンプルを結んでいない。灰色の線を見ると 18,000 サンプル付近で大きく性能が落ちていることがわかる。16,000 サンプル付近から目的関数の最高値も収束してきている。

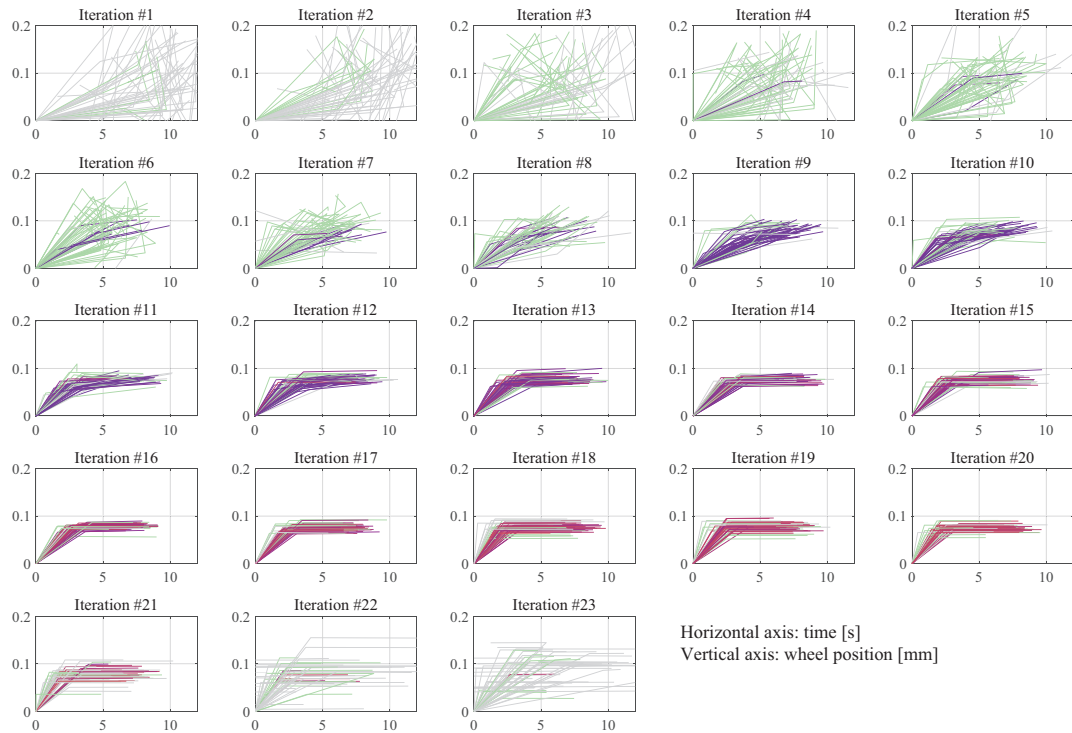


図 5.15 最適化進行にともなう加工パスの変化。

図 5.15 は最適化の過程におけるパラメータを図示したものである。パラメータの図示のために、図 5.12 と同じく、横軸時間、縦軸砥石座標のグラフを用いている。図中の各グラフは、1 回の方策更新に対応する。左上の Iteration #1 のグラフは最適化開始時から 1 回の方策更新までの間に取得されたサンプルを図示しており、その右の Iteration #2 のグラフは 1 回の方策更新から 2 回の方策更新までの間に取得されたサンプルを図示している。方策更新間のサンプル数は 900 なので、Iteration #1 のグラフには 1 個目から 900 個目のサンプルが含まれる。表示の都合上、各グラフには 900 サンプル中からランダムに選んだ 50 サンプルを示している。Iteration #1 から #5 では灰色や緑色の制約を違反したサンプルが多い。Iteration #9 のあたりから制約を満たすことができるサンプルの量が支配的になる。Iteration #9 から #20 までの間ではパスの色が紫色から赤色に変化しており、目的関数の値が大きくなっていることを示している。また、それと並行して、加工パスが送りの多い粗加工と送りの少ない仕上げ加工との組み合わせに収束していくことが見て取れる。Iteration #21 から #23 では灰色や緑色のサンプルが多くなっている。これは、図 5.14 の学習曲線における 18,000 サンプル付近での性能低下と対応している。

### 5.4.5 考察

本節では計算機上での最適化実験の結果について考察する．計算機上での最適化実験の目的は，人間が実用的な解をわかっている問題において最適化を行い，数値解析モデル及び最適化システムの妥当性を評価することであった．

まず，数値解析モデル・目的関数の妥当性に注目する．図 5.15 のように，提案する最適化システムは，粗・仕上げ加工という加工パスを自律的に学習することに成功した．この粗・仕上げの組み合わせによる加工パスは人間が既に知っている実用的な加工パスである．そのため，表面粗さと加工時間といくつかの制約条件からなる目的関数の構成は，加工現場における要求を反映したものであるといえる．

次に，数値解析モデルを利用した最適化システムの妥当性を考える．今回の計算機上での実験では最適化に要した時間は 1 時間あまりであった．その間に 20,000 サンプルの計算が行われた．この 1 時間には加工のテンプレートの事前計算の時間は含まれていないが，含んだとしても半日程度である．このことから，円筒研削加工における砥粒とワークの接触を模擬した数値解析モデルとブラックボックスな最適化手法の組み合わせにより，円筒研削加工の加工条件が，実用に耐えうる時間内で最適化可能であるといえる．また，今回の最適化においては 6 つのパラメータを最適化した．パラメータの数が増えるにしたがって探索空間は指数関数的に広がるため，7 つ以上パラメータにおいて実用に耐えうる時間内に最適化を終えるためには，シミュレーションの高速化，事前知識の注入を行う必要があると思われる．

次に，加工パスの学習のされ方に注目する．また，図 5.15 のように最適化アルゴリズムはまず制約の満たし方を覚え，そこからは連続的に良い加工パスが学習されていくことが分かった．図 5.14 の学習曲線を見ても連続的に目的関数値が向上している．そのため，今回の 6 パラメータでの最適化において目的関数は複雑な形状ではなかったという可能性が示唆される．一方，図 5.14，図 5.15 のように最適化の終盤では急にサンプルの性能が落ちた．これは目的関数が極大値周辺で急峻形状を持つ関数になっているためだと考えられる．アルゴリズムは，最適化の終盤に至るまでに極端な粗・仕上げ加工を学習した．極端な粗・仕上げ加工とは，粗加工の送り速度が制限いっぱい，仕上げ加工では送り速度がほぼ 0 となるような加工パスを示す．方策が極端な粗・仕上げ加工を学習した状態では，新たにサンプリングをすると制約を満たせないサンプルも容易に発生してしまう．このことが性能の低下を引き起こしたのだと考えられる．

## 5.5 実機を含み自律的なモデル修正を行う最適化アルゴリズムの提案と評価

本節では実機を含み自律的なモデル修正を行う最適化アルゴリズムの評価実験について述べる．実験目的は自律的なモデル修正を行う最適化アルゴリズムを評価すること，数値解析モデルを活用した実機における研削加工条件の最適化の可否を検証することである．

### 5.5.1 自律的なモデル修正を行う最適化アルゴリズムの実装

本項では自律的なモデル修正を行う最適化アルゴリズムの実装について述べる．最適化アルゴリズムの具体的な実装の説明の前にアルゴリズムの背景と要求を整理する．

最適化アルゴリズムの背景として，次のような事実が存在すると考えている．

1. 実機と数値解析モデルの間には誤差が存在する
2. 実機の工作機械の価値向上のため，最終的には実機における加工結果から計算される目的関数を最適化したい
3. 実機における実験コストやデータ取得コストは大きい．5 から 10 個程度の変更可  
能な加工条件があったときに実用的な精度で統計モデルを構築することは困難で  
ある
4. 数値解析モデルにおける実験コストやデータ取得コストは小さい
5. 実機加工において，加工条件は加工結果に対して非線形性を持つことがある
6. 数値解析モデルにおいては実機加工における上記の非線形性を模擬することがで  
きる

以上のような背景から，数値解析モデルの現象の模擬結果を利用した実機における最適化を行うことを考えている．そして，最適化アルゴリズムに対しては次のような要求機能があると考えられる．

1. 実機における加工の性能を最適化できる
2. 実機における実験やデータ取得の回数を削減できる

ここからは，提案する最適化アルゴリズムである，実機のデータを利用した数値解析モデルの修正を行う最適化アルゴリズムの詳細について述べる．提案するアルゴリズムは数値解析モデルと実機との間に誤差が発生することを前提としており，実機の加工結果を用いて数値解析モデルを修正する機構を持つ．提案アルゴリズムのフローは図 5.16 のようになっている．提案アルゴリズムでは，繰り返し計算を必要とするために計算コストのか



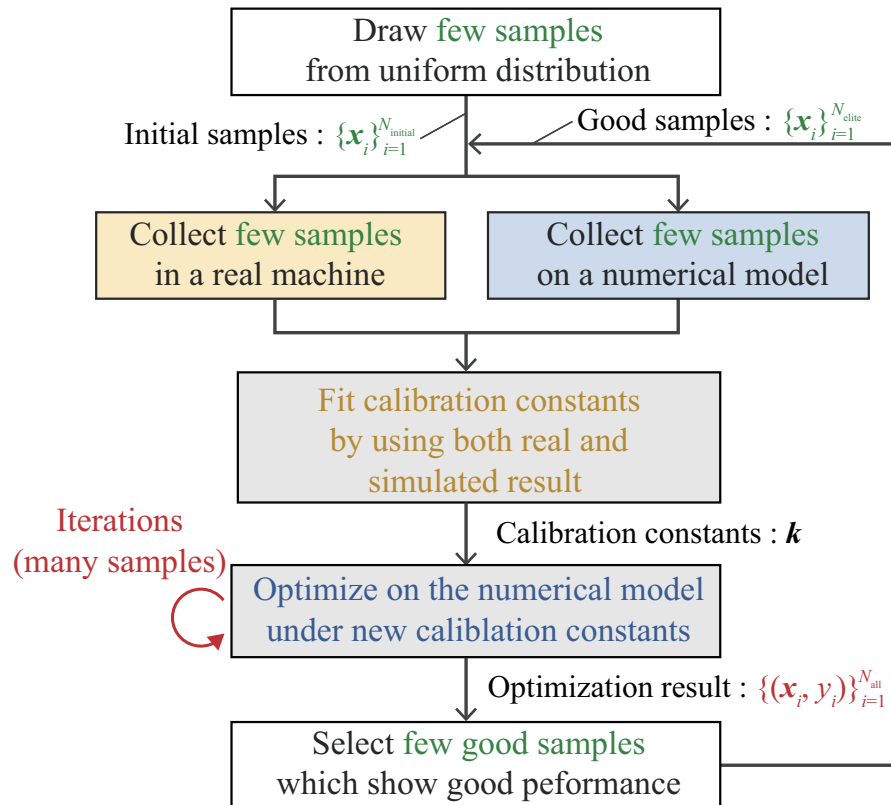


図 5.16 数値解析モデルの修正機構を持つ最適化アルゴリズムのプロセス。

かる目的関数の最適化は計算機上の数値解析モデルに対して行い、少量の実機のデータを用いて数値解析モデルを修正する。数値解析モデルの修正においては、数値解析モデルへの入出力に限定して補正が行われるようになっている。入出力に限定して補正を行うことにより、数値解析モデルの非線形性を損なわないという効果と、補正係数の計算の簡易化という効果が得られる。このアルゴリズムは補正係数を使って数値解析モデルの関数形状を補正していくものである。データ数が少ない場合に滑らかな曲面しか利用できない応答曲面法と異なり、このアルゴリズムではデータ数が少なくても加工現象が持つ非線形性を損なうことがない。

図 5.16 に示すモデル修正を含んだ最適化のフローについて説明する。図 5.16 中の色付きの四角が実機における加工と数値解析モデルにおける加工の模擬を示している。図 5.16 一段目のように、アルゴリズムではまずパラメータの定義域内から  $N_{\text{initial}}$  個のパラメータをランダムに生成する。 $N_{\text{initial}}$  は実機における加工の回数に対応するため、実験コストの低減のために小さい値に設定されることを想定している。次に、図 5.16 二段目のように、それらの少数のサンプルをもとに実機による加工と数値解析モデルによる加工の模擬を行い、加工結果を記録する。本章では加工結果は表面粗さと加工後の直径である。次に、図 5.16 三段目のように、生成されたパラメータと記録した加工結果を用いて数値解析モデル

の補正係数を計算し、数値解析モデルの補正に用いる。補正係数の計算においては、数値解析モデルの出力する表面粗さと直径が、実機における表面粗さと直径に近づくようにする。次に、図 5.16 四段目のように、実機の結果を用いて入出力が補正された数値解析モデルを用いて、計算機上での最適化を行う。これは計算機上での最適化であるため、実機のみを使った最適化に比べて小さなコストで行うことができる。そして、図 5.16 五段目のように、計算機上の目的関数で高い値を示した数サンプルを選び出して、図 5.16 二段目の加工の実行に戻る。この時の選び出すサンプルの数を  $N_{\text{elite}}$  とする。図 5.16 二段目から図 5.16 五段目の手続きを繰り返すことにより、目的関数値の高いパラメータ付近での実機と数値解析モデルとの誤差が低減されていき、数値解析モデル上での最適解が実機の最適解に近づくことを狙っている。

### 5.5.2 実機実験の設定

実機実験における実験には図 5.5 に示す円筒研削盤を利用した。実機を含んだ最適化においても、5.4 節の数値解析モデルをベースとした計算機上での最適化と同じ目的関数、制約条件を利用した。

図 5.17 は実機において加工を行い、目的関数の値を計算するまでのプロセスを示す。5.4 節の計算機上での最適化においては、表面粗さや直径はワーク点群データから計算されていたが、実機実験においては物理的な計測装置で計測する。図 5.17 中央上に示すように、表面粗さは表面粗さ測定器 (ミットヨ製, サーフテスト SJ-210), 直径はマイクロメータ (ミットヨ製, クーラントプルーフマイクロメータ MDC-50MX) を利用して計測した。表面粗さは一回の加工毎に 5 箇所で計測を行い、直径は 3 箇所で計測を行った。

提案アルゴリズムにおいて実機における加工回数に対応する値は次のように設定した。初期サンプル数  $N_{\text{initial}} = 10$ , 選出サンプル数  $N_{\text{initial}} = 5$ . 補正係数の計算の回数は 2 回とした。これらの値は研削加工の最適化についての先行研究 [102][103] の実機加工回数を目安にして決定した。また、計算機上の最適化後のサンプル選出においては、最も目的関数の値が大きくなったサンプルの目的関数値を  $y_{\text{max}}$  として、 $0.8 * y_{\text{max}}$  以上の値をとったサンプルからランダムに選出した。

### 5.5.3 実機実験の結果

図 5.18, 図 5.19, 図 5.20 はそれぞれ実機を含んだ最適化の過程における、加工寸法、物理量、目的関数の値の推移をあらわすグラフである。これらのグラフにおいては横軸はサンプルの番号を表す。黒丸で表される 10 個のサンプルは初期のサンプルを表す。薄い緑色の丸の 5 サンプルは初期の 10 サンプルを用いて補正されたモデルを利用した最適化

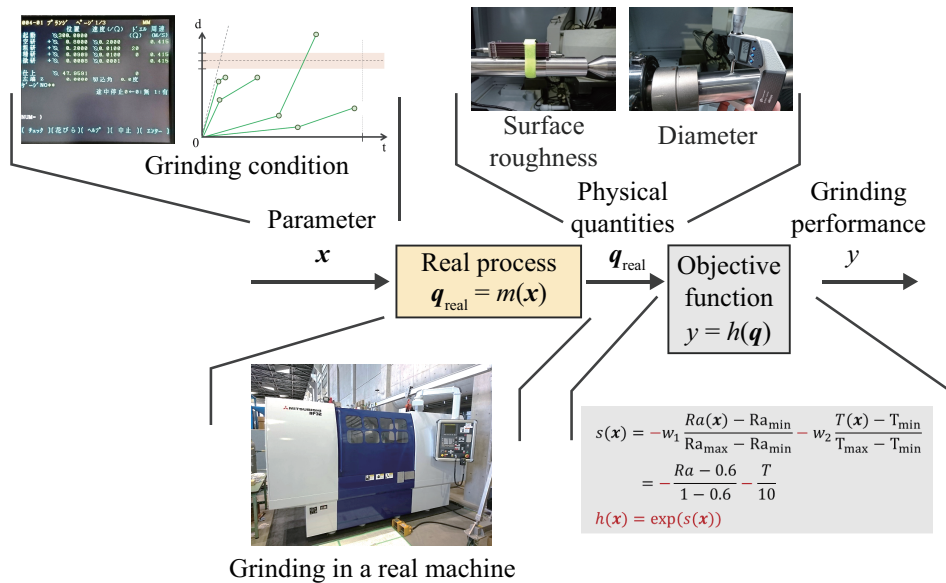


図 5.17 実機実験における目的関数値算出のプロセス。

によって求められたサンプルを表す。そして、濃い緑色の 5 サンプルは薄い緑色の 5 サンプルによって補正されたモデルを利用した最適化によって求められたサンプルを表す。濃い緑色のサンプルは算出のために 2 回モデルの補正を行っていることになる。

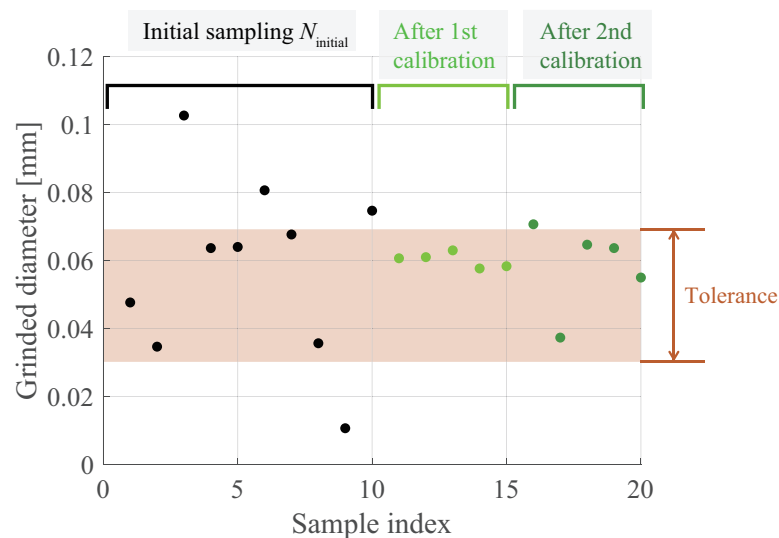


図 5.18 加工後の直径の実測値の推移。

図 5.18 に示す加工寸法は、寸法制約に関する値であり、図中の茶色枠は公差範囲を表している。茶色枠に入っていないサンプルは制約を満たしていないことを示しており、これらのサンプルの目的関数の値は最小値の 0 となる。制約を満たさないサンプルは、初期のランダムサンプルにおいては 10 サンプル中 4 サンプルあったが、モデルの補

正後は 10 サンプル中 1 サンプルに減少した。

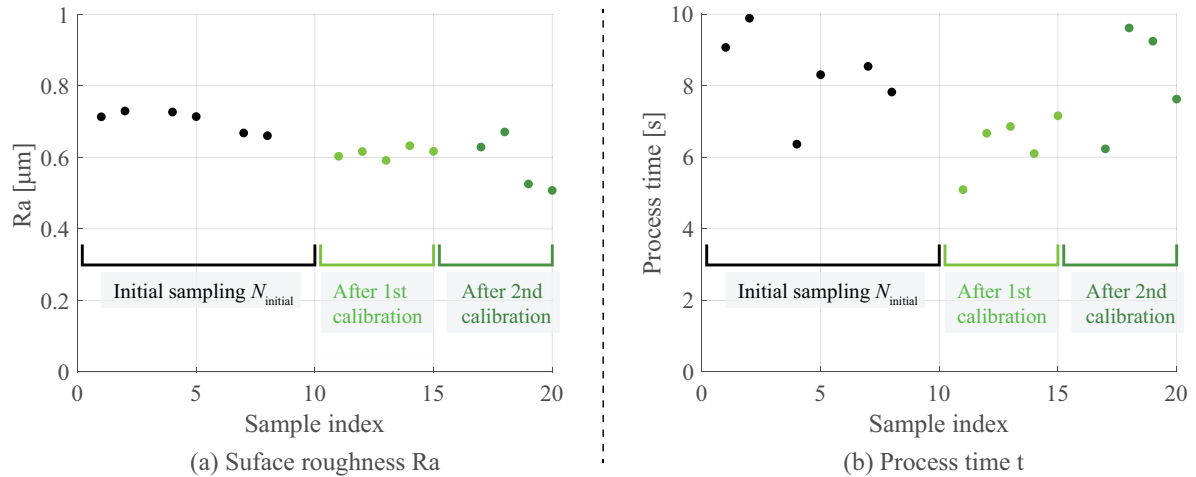


図 5.19 加工後の表面粗さと時間の実測値の推移。

図 5.19(a), (b) はそれぞれのサンプルの加工後の表面粗さ，加工にかかった時間を表している．寸法制約を満たす必要が無いのであれば，ほとんど削らずに表面粗さを良好なままにしたり，加工をスタート直後に終了し加工時間を短くするようなパラメータも許されてしまう．最適化によって制約が満たされるようになり物理量が良好になったこと図示するために，図 5.19 においては，寸法制約を満たしたサンプルのみプロットしている．図 5.19(a) のように，モデルの 1 回目の補正後のサンプルでは，初期サンプルに比べて表面粗さが小さくなった．また，2 回目の補正後のサンプルでは，1 回目補正後ののサンプルより表面粗さが小さくなるサンプルもあれば逆の傾向を示すサンプルも存在した．図 5.19(b) のように，モデルの 1 回目の補正後のサンプルでは，初期サンプルに比べて加工時間が短くなった．一方で，2 回目の補正後のサンプルでは，初期サンプルと同程度の加工時間がかかるようになった．

図 5.20 はそれぞれのサンプルの目的関数の値を表している．補正後モデルで最適化したサンプル 10 個中の 8 個において，初期サンプル 10 個の最大値を上回る値となった．このことから提案するアルゴリズムによる目的関数の値を増加させる効果は存在することがわかる．実機加工を行った 20 サンプルを通して最も良い目的関数の値となったのは，1 回目の数値解析モデルの補正の後のサンプルであった．

#### 5.5.4 実機実験の考察

本項では，自律的に数値解析モデルの補正を行い，最適値周辺での実機との誤差を低減するというアルゴリズムを提案した．提案するアルゴリズムを用いて，実機の少量のデータから数値解析モデルを修正することにより，図 5.20 のように実機における加工性能を

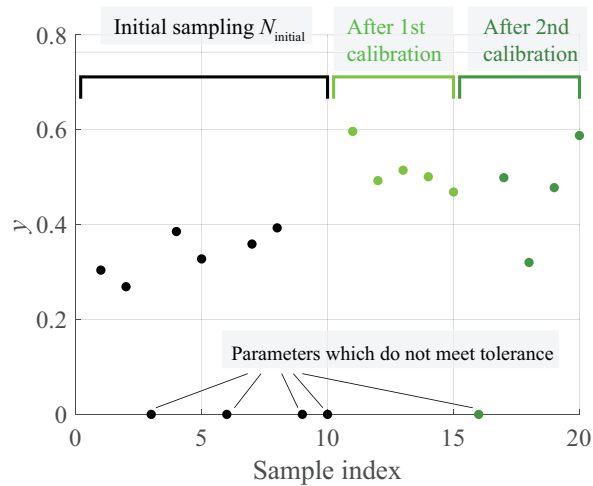


図 5.20 加工性能を評価する目的関数の推移.

向上させられるということが分かった.

図 5.19 によると, 2 回目の補正後は 1 回目の補正後に比べ, 時間をかけて丁寧に加工するようなパラメータが選ばれている. これにより, 表面粗さは改善するものの, 加工時間がかかるようになってしまい, 目的関数の値の平均は低くなっている. 2 回目の補正後に選ばれたサンプルは, 数値解析モデル上では, 加工時間が長くなっても (悪化しても), 引き換えに得られる表面粗さの改善が上回るため, 目的関数の値が大きくなり, 良いサンプルとして選ばれている. 実機においては, 時間をかけた加工を行うことにより表面粗さの改善は起こるものの, 改善量は数値解析モデルに比べて小さい. 以上のような理由から, 2 回目の補正後には, 時間をかけた加工のパラメータが選ばれ, 実機における加工時の目的関数の値の平均は小さくなったと考えられる.

## 5.6 本章の結言

本章では, 円筒研削加工の加工条件最適化について述べた. 具体的には, 計算量を削減した数値解析モデルの開発, 誤差の発生を前提とした同化による最適化手法の提案, 計算機上での最適化, 最適化されたパラメータの実機における評価を行った.

本章における研究目的と, それらに対応する結果を以下に述べる.

### 本章における研究目的 (再掲)

- 研究課題を解決するために手法を提案し実行する. (CG-A)
  1. 円筒研削加工の特性に注目し簡略化を施した数値シミュレータを開発する. (CG-A-1)
  2. 実機と数値解析モデルとの差を補正しながら実機の制御パラメータを最適化する.

る。(CG-A-2)

- 実際の機械システムのパラメータを最適化まで行い，各問題の最適化が現代の技術の組み合わせで解決が可能か否かについて，デモンストレーションを行うことにより検証する。(CG-B)
- 1. 計算機上における目的関数の最適化の可否や技術的妥当性を検証する。(CG-B-1)
- 2. 提案手法によって実機向けに最適化されたパラメータの実機における性能を評価する。(CG-B-2)

### 取り組みの結果

目的毎に取り組みの結果について述べる。

■(CG-A-1) 円筒研削加工の特性に注目し簡略化を施した数値シミュレータを開発する  
円筒研削加工における一つ一つの砥粒とワークとの接触を考慮してなおかつ低計算量で動作する数値解析モデルを開発した。開発では特性の違う汎用シミュレータと高速シミュレータの2つの数値解析モデルを開発した。汎用シミュレータでは，円周方向の一部分を区切った検査面積を設定することなどにより，計算結果を変えないまま計算時間を削減した。高速シミュレータでは，回転数比が同一な場合，幾何的転写形状が同一になるという性質を利用して，回転数比毎に転写形状テンプレートを用意し加工を模擬する。高速シミュレータではシミュレータの動作状況を限定することにより，シミュレーションの精度を維持したまま汎用シミュレータに比べ計算時間を削減できた。これらの数値解析モデルの高速化により最適化における反復的サンプリングが十分可能になった。

■(CG-A-2) 実機と数値解析モデルとの差を補正しながら実機の制御パラメータを最適化する  
数値解析モデルと実機との間の誤差を補正しながら，実機における最適化を行う自律的な最適化システムを提案し，誤差を補正しなかった場合に比べて高い性能を発揮した。

■(CG-B-1) 計算機上における目的関数の最適化の可否や技術的妥当性を検証する  
5.4 節の検証結果から，円筒研削加工における砥粒とワークの接触を模擬した数値解析モデルとブラックボックスな最適化手法の組み合わせにより，円筒研削加工の加工条件が，実用に耐えうる時間内で最適化可能であるといえる。

また，最適化アルゴリズムは，粗・仕上げ加工という加工パスを自律的に見つけ出すことに成功した。粗・仕上げ加工は人間も知っている定石のような加工法である。背景には，2つの要因が考えられる。1つ目の要因は目的関数の形が単純で，最適解の候補がほとんどなかったというものである。今回はメタな最適化システムの視点に立った最適化シ

システムの構築という目的や、実機において実用的なパラメータを算出するという目的があったため、最適化の条件は単純なものであり、加工パスの自由度も 2 と低かった。そのため、今回の問題は多峰性が少なく、最適化問題としては簡単なだったと考えることができる。この仮説の検証方法としては、部分空間での可視化を組み合わせること、貪欲な最適化手法の初期値を変えてみてたどり着く先を評価することなどが考えられる。

2 つ目の要因はモデルがうまく組めたことである。最適化結果が人間の定石通りになったのは、簡約化したモデルでも「粗加工では表面は粗くなる」という事実が成り立ったためだと考えられる。また、加工をまったくせずに目的関数の値を大きくすることなどの意図しない挙動が発生しなかったことも、モデルがうまく組めている証拠になると考えている。

■(CG-B-2) 提案手法によって実機向けに最適化されたパラメータの実機における性能を評価する CG-A-2 と同内容につき記述を割愛する。

#### 将来課題

本章の取り組みにより、以下の課題の存在が明らかになった。以下の課題に取り組むことにより、提案する手法の優位性の提示、研削加工条件の大局的最適化、数値解析モデルを用いたアプローチの優位性の向上という効果が見込める。

1. 統計モデルを用いた研削加工の実機条件最適化実験
2. 研削加工条件最適化における目的関数の非線形性・多峰性の確認
3. 衝突判定・テンプレート適用処理の複数コア，GPU 計算を用いた高速化
4. ワーク半径方向 ( $p$  軸) だけではなく，ワーク直径方向の自由度も持たせた最適化
5. 加工パスの時間方向の分割数を増やした最適化

## 第 6 章

# 結言

本研究では「数値解析手法を活用した最適化により現実存在する機械システムを最適化する問題」をターゲットとし、背後にある技術や既存の最適化アプローチについて整理した。整理の結果、最適化システムを構築・修正する方法論の研究は存在しないということを見出した。そして、最適化システムを構築・修正するための、メタな最適化という視点を定義した。加えて、具体的な実機械システムの最適化問題を題材とし、メタな最適化の視点に基づいて最適化システムの構築・修正を行い、実機械システムにおいて性能の良いパラメータを求めることができた。本章ではそれらの実施内容と成果及び知見、将来課題について概要を述べる。

### 6.1 実施内容まとめ

1 章においてはまず、昨今の機械産業の状況、機械設計の方法、情報技術の進歩について紹介し議論を行った。その議論により、今後、数値解析手法を活用した機械システムの制御や設計のパラメータを最適化の取り組みが増えていくと考えられるため、本研究の研究目的を「人間の設計者が行っている具体的な最適化問題を題材にし、背後に隠れている研究課題を明らかにする」とことと定めた。

2 章においては数値解析モデルとブラックボックス最適化手法を組み合わせた最適化システム（「数値解析モデルを活用した最適化システム」）の定義、利点、課題について述べた。2 章の議論における要点は以下の通りである。

- 物理現象のモデル化の形態には、「記号モデル」、「統計モデル」、「数値解析モデル」の 3 つが存在する。
- 数値解析モデルは、その内部において基礎的な原理に基づく計算を繰り返し行うことにより、物理現象を模擬する。
- 数値解析モデルは、出力が複雑なモデルを構築できる、再利用性が高いという利点



が存在する。

- 数値解析モデルは，計算負荷が高くなるという欠点を持つが，計算機性能の向上やアルゴリズムの工夫により，欠点は解消する傾向にある。
- 目的関数の形状に関して強い仮定を置かない最適化手法の一種として，ブラックボックス最適化がある。ブラックボックス最適化は，目的関数の定式化が必要ないことから数値解析モデルと組み合わせることに適している。
- 機械工学分野の最適化の研究においては，数値解析モデルとブラックボックス最適化とが組み合わせて利用されることが多い。
- 実機械に対して最適化結果を利用するためには，与えられた数値解析モデルと目的関数の元でパラメータ  $x$  を最適化するのみならず，数値解析モデルと目的関数の内部パラメータ  $\theta, \psi$  も最適化する必要がある

3 章においては，2 章で紹介した数値解析モデルを活用した最適化システムを構築及び修正していくためのメタな最適化システムについて述べた。具体的には，「メタな最適化システム」という最適化システムを構築していくための新しい視点を定義し，メタな最適化システムの視点にたって最適化システムの持つ問題を解決するための機構を提案した。3 章の議論における要点は以下の通りである。

- 最適化システムを構築・改善し，実機械で良い性能を発揮するようなパラメータを求めるために人間が行っている作業は，多岐にわたる。
- 実機械で良い性能を発揮するようなパラメータを最適化により求めるには，人間とのアルゴリズムとの協調の必要がある。
- メタな最適化システムは，実機，メタな目的関数，人間（設計者），修正アルゴリズムから構成される。
- 提案するメタな最適化システムの主要な動作は，計算機上の最適化，最適化志向のモデリング，目的関数の設計の 3 つに集約される
- 機構 1. 最適化結果の重要度に基づいた要素修正：最適値から遠いパラメータにおける要素の修正を省き，要素の修正にかかる実験コストやデータ分析のコストを低減する。
- 機構 2. 数値解析結果の再利用：一度計算した数値解析の結果を再利用し，最適化全体にかかる時間を低減する。
- 機構 3. 補正係数導入によるモデル修正の自動化：数値解析モデルの修正を自動化し，人間を介した修正のオーバーヘッドを低減する。

4 章においては，機械システムの最適化の具体例の一つ目として，電子制御トランスミッション (ECT) の制御パラメータの最適化について述べた。

- ECT では十分な実行速度を持つ数値解析モデルは存在するものの、最適化は人手によって行われており、設計者の設計指針を反映した目的関数は存在しなかった。(4.1 節)
- 設計者の設計指針を反映させた目的関数を素早く作り出すための目的関数の設計指針を示し、設計解や設計の途中過程について述べた。(4.2 節, 4.3 節)
- 計算機上で自律的に動作する制御パラメータ最適化システムを構築し、最適化実験を行った。(4.4 節)
- 目的関数の多峰性を明に仮定したブラックボックス最適化手法である階層型確率的最適化 (HSO) を提案し、ECT の性能を評価する目的関数の最適化に利用した。(補足 A, 4.4 節)
- 計算機上での最適化により求められた実用的なパラメータの実機における性能評価を行った。評価の中で数値シミュレータと実機との誤差に注目し、数値シミュレータの改善点について議論した。(4.5 節)

5 章においては、機械システムの最適化の具体例の二つ目として、円筒研削加工の加工条件の最適化について述べた。

- 円筒研削加工では目的関数の定義は従来研究において議論されていたが、最適化計算を行うために十分な計算速度で加工を模擬することが可能な数値シミュレータが存在せず、本研究において提唱している数値解析モデルを活用した最適化が計算コスト的に困難であった。(5.1 節)
- 円筒研削加工の幾何的な特性に注目して計算を簡略化した数値シミュレータを開発した。(5.3 節)
- 計算機上で自律的に動作する加工条件 (制御パラメータ) 最適化システムを構築し、最適化実験を行った。(5.4 節)
- 加えて、数値解析モデルと実機の間誤差を補正しながら、実機における最適化を行う自律的な最適化システムを提案し、実機を利用した評価実験を行った。(5.5 節)

## 6.2 実機械システムの最適化における成果と知見まとめ

### 6.2.1 4 章 電子制御トランスミッションの制御パラメータ最適化

1. 1D-CAE をベースとした数値解析モデルとブラックボックス最適化の組み合わせにより ECT の制御パラメータが最適化可能であると示した。
  - 設計検証に用いられる 1D-CAE の数値解析モデルとブラックボックス最適化手法を組み合わせ最適化されたパラメータは実機の ECT においても既存の

パラメータを上回る性能を発揮した。

- 複数の異なる目標が存在する ECT の目的関数の設計のため、現場の設計者を含んだシステムを反復的に回すことにより目的関数が洗練されていくということを示した。
2. 複数の次元の異なる性能目標が存在する ECT においては、性能に関する目的関数が多峰性を持つことが示唆された。証拠としては以下のような事実及び実験結果が存在する。
    - 設計者がパラメータの数を絞れておらず、パラメータ同士の相互作用があることが示唆されている。
    - 単一なガウス分布をサンプリング方策とする最適化手法である CEM や CMA-ES を利用した場合、最適化の結果が安定しない。
    - HSO が最適化の過程において異なる峰を捉えている。
    - 部分空間に限定した目的関数の可視化結果は多峰性を持つ。
  3. HSO は目的関数の多峰性を陽に想定して最適化を行うため、ECT の目的関数最適化の際には、多峰性を想定しない最適化手法に比べて安定して良い解を見つけることができることが分かった。
  4. 計算機上のモデルと実機での挙動の差異の観察から、数値シミュレータの修正に関する以下の知見が取り出された。
    - 最適化に利用した数値シミュレータは車体の 3 次元的な挙動を模擬できない
  5. 最適化アルゴリズムが人間の意図していないパラメータを見つけてしまう場合の具体例を示した。具体的には、最適化アルゴリズムが以下のようなパラメータを発見した。
    - 評価指標の設定不良により、目的関数の値としては高いが実機に搭載するには好ましくないパラメータ。
    - 数値シミュレーションにおいて設計者の意図せぬ動作を引き起こすパラメータ。

## 6.2.2 5 章 円筒研削加工の加工条件最適化

1. 砥粒と被削材の衝突を模擬した数値解析モデルとブラックボックス最適化の組み合わせにより円筒研削加工条件が最適化可能であると示した。
  - 加工パスの経由点及び終点の時間と位置、砥石と被削材の回転数を同時に最適化する問題においては、アルゴリズムに与える前提知識なしでも粗加工と仕上げ加工を順番に行うという良く知られた加工法を学習できるということが分

かった。

- 数値解析モデルと実機との誤差を補正しながら，実機における最適化を行う自律的な最適化システムを提案し，誤差を補正しなかった場合に比べて高い最適化性能を発揮した。
2. 円筒研削加工において幾何的な特性に注目することにより数値シミュレーションの精度を維持したまま計算時間を削減できることを示した。
- 円周方向の一部分を区切った検査面積を設定することで数値シミュレーションの精度を維持したまま計算時間が削減できることを示した
  - 回転数比が同一な場合，幾何的転写形状が同一になるという性質を利用して，回転数比毎にテンプレートを用意することで，数値シミュレーションの精度を維持したまま計算時間を削減できることを示した。

## 6.3 将来課題

本論文における議論から，下記の将来課題が存在すると考えている。

- 全般
  - － 単純な現象の 3 形態によるモデル化を通して各形態の特性を定量的に調査
- 4 章 電子制御トランスミッションの制御パラメータ最適化
  - － パラメータ定義域付近の挙動を安定化させた HSO の開発
  - － 各最適化アルゴリズムによる探索範囲の可視化と定性的性能の議論
- 5 章 円筒研削加工の加工条件最適化
  - － 統計モデルを用いた研削加工の実機条件最適化実験
  - － 研削加工条件最適化における目的関数の非線形性・多峰性の確認
  - － 衝突判定・テンプレート適用処理の複数コア，GPU 計算を用いた高速化
  - － 研削加工の数値解析モデルの内部実装として統計モデルを使うことにより発生熱を低計算コストで模擬

## 付録 A

# 階層型確率的最適化の提案

本補足においては、著者らが提案している階層型確率的最適化 (HSO; Hierarchical stochastic optimization) の定義、性能評価結果について述べる。HSO はブラックボックス最適化手法の一種とみなすことができる。多峰性を持つ目的関数の最適化の問題に対処するために、階層強化学習の手法である HPSDE (Hierarchical Policy Search with return-weighted Density Estimation)[88] を用いる。HPSDE は標準的な強化学習の問題において状態  $s$  が与えられた状態で、行動  $a$  の確率密度を学習することを目標とするが、HSO は最も良い目的関数値を与える最適なパラメータを取得することを目標とする。

### A.1 問題の定式化

以降の議論では、強化学習の文脈で使用する表記法を使用する。パラメータ  $x$  はシステムへの入力であり、 $R(x)$  はシステムの目的関数である。 $R(x)$  をパラメータ  $x$  の報酬とも呼ぶ。以降は、確率密度の形で表される、パラメータ  $x$  のサンプル方策  $\pi(x)$  について考える。方策をベースとした強化学習の問題と同様に、式 (Eq. A.1) で与えられる方策  $\pi$  の期待報酬を最大化する問題として、パラメータ  $x$  を最適化する問題の定式化する。

$$J(\pi) = \int \pi(x)R(x)dx \quad (\text{A.1})$$

提案手法では、多峰性の有る目的関数を扱うことを目指しているため、式 (A.2) のような階層的な方策について考える。

$$\pi(x) = \sum_{o \in O} \pi(o)\pi(x|o), \quad (\text{A.2})$$

ここで、 $O$  は  $o$  の集合である。式 (A.2) において、 $o$  はオプションを表す潜在変数であり、目的関数のそれぞれの峰に対応することを企図している。式 (A.2) の  $\pi(x|o)$  を下位方策もしくはオプション方策と呼ぶ。

式 (A.1) および式 (A.2) を用いて，期待される報酬は式 (A.3) のように書ける．

$$J(\pi) = \sum_{o \in \mathcal{O}} \int \pi(o) \pi(\mathbf{x}|o) R(\mathbf{x}) d\mathbf{x} \quad (\text{A.3})$$

$o$  が指定または事前定義されている場合， $J(\pi)$  を最大化する問題は，標準的な確率的最適化問題または単一の方策を使用した強化学習の問題と同等のものになる．この問題の難点は，アルゴリズムが真の密度分布  $\pi(o)$  を事前に知らずに  $\pi(\mathbf{x})$  を学習する必要があることである．つまり， $\pi(o)$  と  $\pi(\mathbf{x}|o)$  を同時に学習する必要がある．

## A.2 提案手法

HSO は，目的関数に基づいて密度  $\pi^*(\mathbf{x})$  によってサンプリングされる  $\mathbf{x}$  の確率密度を推定することにより， $o$  の分布を推定する．本節では， $\pi(o)$  と  $\pi(\mathbf{x}|o)$  を同時に学習する 3 つの方法について説明する．図 A.1 は，HSO の最適化プロセスを示す．

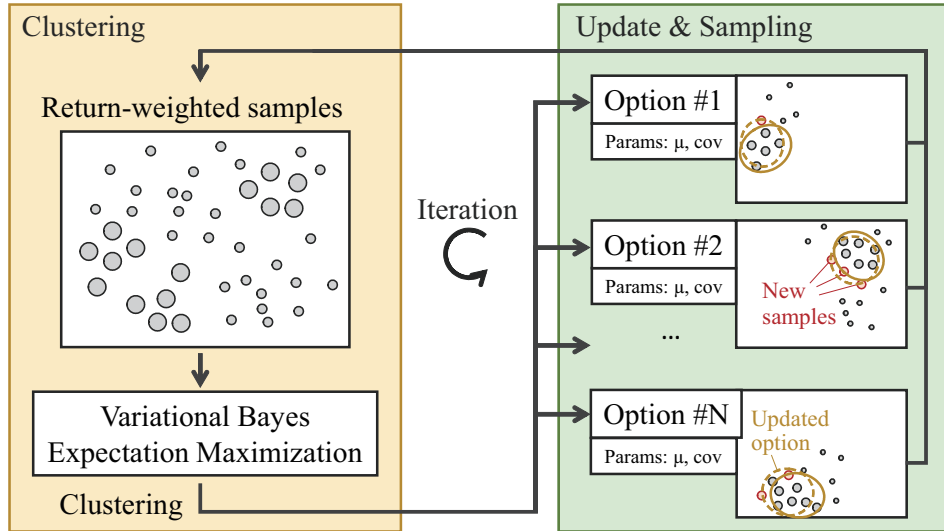


図 A.1 HSO の最適化プロセスの概念図．

### A.2.1 密度推定のための重要度サンプリング

まず，式 (A.4) で与えられるパラメータ  $\mathbf{x}$  の密度を考える．

$$\pi^*(\mathbf{x}) = \frac{f(R)}{Z}, \quad (\text{A.4})$$

ここで， $Z$  は分配関数であり， $f(R)$  は報酬  $R$  に関して単調増加関数である．この分布に従うと，報酬値が高い  $\mathbf{x}$  が高い確率で選ばれる．したがって， $R(\mathbf{x})$  のモードを見つける

ことは、理想方策  $\pi^*(\mathbf{x})$  のモードを見つけることと同等である。このように、目的関数の複数のモードを見つける問題を、 $\pi^*(\mathbf{x})$  におけるマルチモーダル分布を推定する問題に帰着させることができる。重要度サンプリングを使用して、理想方策  $\pi^*(\mathbf{x})$  によって生成されたサンプルの確率密度分布を推定する。  $N$  個のサンプル  $\{\mathbf{x}_i\}_{i=1}^N$  が方策  $\pi_{\text{old}}$  から選ばれたとすると、各サンプルの重要度は式 (A.5) のように設定される。

$$W(\mathbf{x}_i) = \frac{\pi^*(\mathbf{x}_i)}{\pi_{\text{old}}(\mathbf{x}_i)} = \frac{f(R(\mathbf{x}_i))}{Z\pi_{\text{old}}(\mathbf{x}_i)} \quad (\text{A.5})$$

重要度の重みを正規化すると、式 (A.5) にあった分配関数は打ち消され、式 (A.6) が得られる。

$$\tilde{W}(\mathbf{x}_i) = \frac{W(\mathbf{x}_i)}{\sum_{j=1}^N W(\mathbf{x}_j)} = \frac{\frac{f(R(\mathbf{x}_i))}{Z\pi_{\text{old}}(\mathbf{x}_i)}}{\sum_{j=1}^N \frac{f(R(\mathbf{x}_j))}{Z\pi_{\text{old}}(\mathbf{x}_j)}}. \quad (\text{A.6})$$

式 (A.6) のように正規化された重みを使用して、方策  $\pi^*(\mathbf{x})$  によって生成されたサンプルの密度を推定できる。この重要度に応じた重みづけは HRL に関する以前の研究 [88, 45] で提案されているものであり、この重要に応じた重みづけを本研究で提案する HSO にも適用する。HSO においても、先行研究と同じく [88],  $f(R)$  として  $\exp(R)$  を用いる。

### A.2.2 重要度に基づく重みづけを利用した VBEM 法

現在の方策  $\pi_{\text{old}}(\mathbf{x})$  から抽出したサンプルと重要度の重みを使用して密度  $\pi^*(\mathbf{x})$  を推定することにより、多峰性の持つ分布の潜在変数  $o$  を推定する方法について述べる。この問題は、カルバック・ライブラー情報量  $D_{KL}(\pi^*(\mathbf{x})||\pi(\mathbf{x}))$  を最小化する問題として定式化できる。

$$D_{KL}(\pi^*(\mathbf{x})||\pi(\mathbf{x})) = \int \pi^*(\mathbf{x}) \log \frac{\pi^*(\mathbf{x})}{\sum_{o \in \mathcal{O}} \pi(o)\pi(\mathbf{x}|o)} d\mathbf{x}. \quad (\text{A.7})$$

$D_{KL}(\pi^*(\mathbf{x})||\pi(\mathbf{x}))$  の最小化は、重み付き対数尤度の最大化によって達成される [88]。最大化問題を効率的に解くために、下位方策は、ガウス分布  $\pi(\mathbf{x}|o) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  によって与えられるという仮定を置く。このような仮定の下では、潜在変数  $o$  の推定は混合ガウス分布のクラスタリング問題と同等である。この論文では、変分ベジアン EM アルゴリズム (VBEM; Variational Bayesian Expectation Maximization)[107] を使用してクラスタリングを行う。VBEM には、最尤法に基づく EM アルゴリズムと比較してクラスター数を自動的に調整できるという利点があり、ユーザがクラスター数を調整する必要がない。

### A.2.3 有効な下位方策選択のための上位方策

下位方策  $\pi(\mathbf{x}|o)$  の学習後, 上位方策  $\pi(o)$  は以降の最適化を効率的に行えるように  $o$  を選択する必要がある. HSO においては, 上位方策として, 探索 (exploration) と活用 (Exploitation) を効果的に割り当てることができる UCB (Upper Confidence Boundary) 方策 [108] を使用する. UCB 方策では探索と活用のバランスを加味したサンプルの価値を出力するユーティリティ関数という関数が存在し, ユーティリティ関数の出力が最大になるサンプルを選択する. HSO においては, 式 (A.8) に示すように, ガウス過程回帰 [109] を使用して目的関数を近似し, UCB 方策のユーティリティ関数を計算する.

報酬関数は, 以下のようにガウス過程回帰を用いて近似される.

$$R(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (\text{A.8})$$

ガウス過程回帰において, パラメータ同士の類似性を示すカーネルには, 以下のような二乗指数カーネルを使用する.

$$k(\mathbf{x}_i, \mathbf{x}_j) = s^2 \exp\left(\frac{-\|\mathbf{x}_j - \mathbf{x}_i\|^2}{2l}\right) \quad (\text{A.9})$$

ここで,  $s$  と  $l$  はハイパーパラメータであり, 最適化によって取得される [109]. データセット  $\mathcal{D} = \{(\mathbf{x}_i, R(\mathbf{x}_i))\}_{i=1}^n$  が与えられると, テスト入力  $\mathbf{x}'$  の報酬の予測分布は, ガウス分布を利用して以下ようになる.

$$p(R|\mathbf{x}') \sim \mathcal{N}(\mu(\mathbf{x}'), \Sigma(\mathbf{x}')), \quad (\text{A.10})$$

ここで, 期待報酬の平均と分散は以下のように推定される.

$$\mu_R(\mathbf{x}') = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{R}_{1:n}, \quad (\text{A.11})$$

$$\sigma_R(\mathbf{x}') = k(\mathbf{x}', \mathbf{x}') - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}, \quad (\text{A.12})$$

ここで,  $\mathbf{R}_{1:n} = [R(\mathbf{x}_1), \dots, R(\mathbf{x}_n)]^T$  であり,  $\mathbf{k} = [k(\mathbf{x}_1, \mathbf{x}'), \dots, k(\mathbf{x}_n, \mathbf{x}')]^T$ ,  $\tilde{\mathbf{k}} = [k(\mathbf{x}', \mathbf{x}_1), \dots, k(\mathbf{x}', \mathbf{x}_n)]^T$  であり,  $\mathbf{K}$  は以下のように与えられる.

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}. \quad (\text{A.13})$$

下位方策の番号  $o$  が選択されると, パラメータ  $\mathbf{x}$  は  $\pi(\mathbf{x}|o)$  から選択される.  $o$  が与えられた場合の期待報酬を計算するには, 以下のように周辺分布を計算する必要がある.

$$p(R|o) = \int \pi(\mathbf{x}|o) p(R|\mathbf{x}) d\mathbf{x}. \quad (\text{A.14})$$



**Algorithm 1** Hierarchical Stochastic Optimization (HSO)**Input:** the maximum number of the clusters  $O_{\max}$ 

Initialize the option policies, e.g. random policy

Collect the samples  $\mathcal{D} = \{(\mathbf{x}_i, R_i)\}_{i=1}^N$  by performing the initial policy**repeat**    Compute the importance of each sample  $\tilde{W}(\mathbf{x}_i)$     Estimate  $p(o|\mathbf{x}_i)$  through density estimation using the importance weight  $\tilde{W}$     Assign the sample  $\mathbf{x}_i$  to option  $o_i^* = \arg \max p(o|\mathbf{x}_i)$  for  $i = 1, \dots, N$     **for each**  $o$  **do**        Train the  $o$ th policy using a policy search method    **end for**

Train the GP model to approximate the return function

Select the option  $o^* = \arg \max \mathbb{E}[R|\pi(\mathbf{x}|o), \mathbf{x}, o]$ Draw samples by following  $\pi(\mathbf{x}|o^*)$  and calculate the returnsRecord the data  $\mathcal{D}_{\text{new}} = \{(\mathbf{x}_j, R_j)\}_{j=1}^M$ Store the data  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{new}}$ **until** the maximum of the returns converged

$p(R|\mathbf{x}')$  をガウス分布で近似すると、周辺分布  $p(R|o)$  も以下のようにガウス分布で近似することができる。

$$p(R|o) \sim \mathcal{N}(\mu_o, \sigma_o), \quad (\text{A.15})$$

ここで、各下位方策の平均  $\mu_o$  と分散  $\sigma_o$  は、 $\pi(\mathbf{x}|o)$  に従って選択されたサンプルの平均値と分散値により近似される。上記の結果を使用して、UCB 方策 [110] に従ってオプション変数  $o$  が選択される。

$$o^* = \arg \max_{o \in \mathcal{O}} (\mu_o + \beta \sigma_o) \quad (\text{A.16})$$

式 (A.16) のように、上位方策は  $o$  を選択する。これにより、平均と分散の加重和が最大化される。式 (A.16) において  $\beta$  は 1 とした。

HSO の疑似コードは Algorithm 1 のようになる。

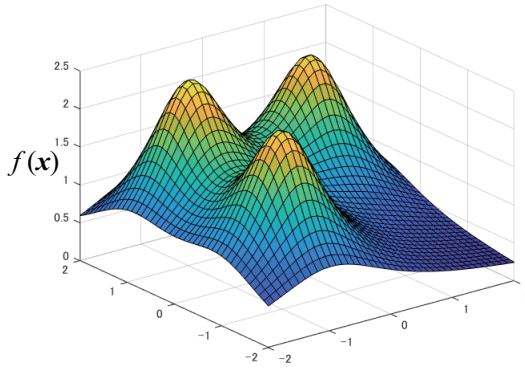
### A.3 動作検証

本節では、トイプロブレム (検証用問題) を用いて HSO の性能を評価する。トイプロブレムとして式 (A.17) と式 (A.18) で表される、入力パラメータが 2 次元で多峰性を持つ 2 つの目的関数を用いる。

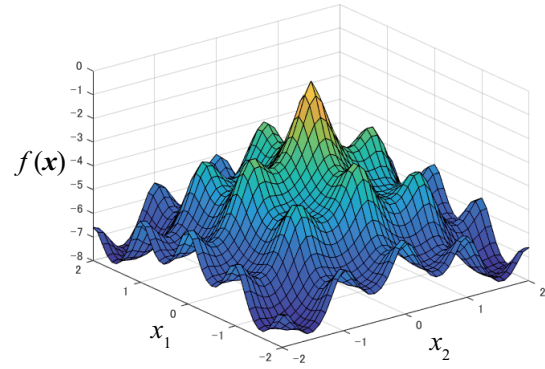
$$f(x_1, x_2) = \sum_{i=1}^3 \frac{1}{(x_1 + a_i)^2 + (x_2 + b_i)^2 + d}, \quad (\text{A.17})$$

ここで,  $(a_1, a_2, a_3) = (-1, 1, 1)$ ,  $(b_1, b_2, b_3) = (-1, 1, -1)$  であり,  $d = 0.5$  である.

$$\begin{aligned} f(x_1, x_2) = & 20 \exp\left[-0.2 \sqrt{0.5(x_1^2 + x_2^2)}\right] \\ & + \exp[0.5(\cos(2\pi x_1) + \cos(2\pi x_2))] \\ & - e - 20 \end{aligned} \quad (\text{A.18})$$



(a) Tri-modal function



(b) Ackley function

図 A.2 2つのトイプロブレムの関数形状.

1 つ目の目的関数は、図 A.2(a) に示すように、3 つの極大値を持つように定義されている。2 つ目の目的関数は、Ackley 関数という最適化手法のベンチマークに用いられる関数であり、図 A.2(b) に示すように、数多くの極大値を持つ。HSO の性能を評価するために、4.4 節と同様に、よく知られているブラックボックス最適化手法である CEM と CMA-ES を利用して比較を行った。これらの手法は強化学習の研究で利用されたり比較されることが多く [90], CMA-ES は現時点において最先端のブラックボックス最適化手法と見なされている [111]。図 A.3, A.4 のグラフは、最適化の学習曲線を示す。これらのグラフは、HSO がトイプロブレムで CEM および CMA-ES と同等の最適化性能を発揮することを示す。また、図 A.3 のように、多峰性が明確な 1 つ目のトイプロブレムでは HSO の性能が他の 2 手法に比べて高くなった。

図 A.5 は、HSO を 1 つ目の目的関数に適用した結果を示す。最適化の初期段階においては、図 A.5 の左上に示すように、サンプルは均一分布の方策から取得される。この初期サンプリングにより、局所解に陥ることが防止される。次に、サンプルが重要度に応じて

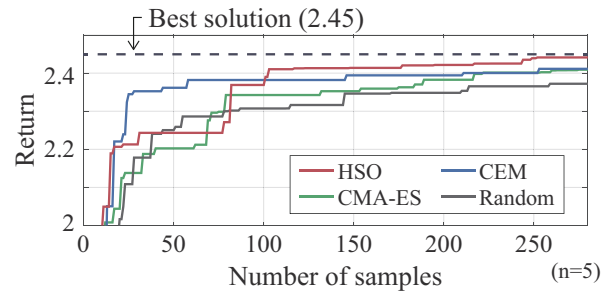


図 A.3 3つの峰を持つ2次元トイプロブレムでの学習曲線。

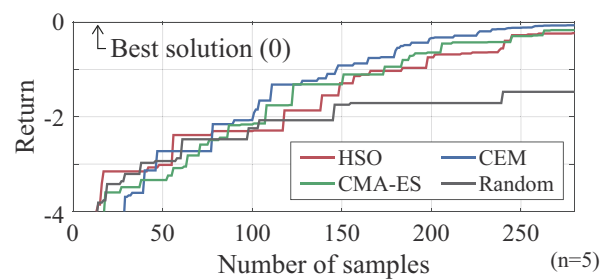


図 A.4 Ackley 関数での学習曲線。

重みづけされる。これにより、**HSO** は目的関数の多峰性を見つけることができる。重み付けが完了すると、下の段に示すように、サンプルは **VBEM** アルゴリズムを使用してクラスタリングされる。クラスタリングされた各クラスはオプションとも呼ばれる。その後、クラス毎の下位方策に従ってサンプリングが行われ、その結果に応じて下位方策が数回更新される。上位方策と下位方策を改善するために、クラスタリングとサンプリングが交互に行われる。1つ目のトイプロブレムにおいては、**HSO** は目的関数の3つの峰を把握することに成功した。

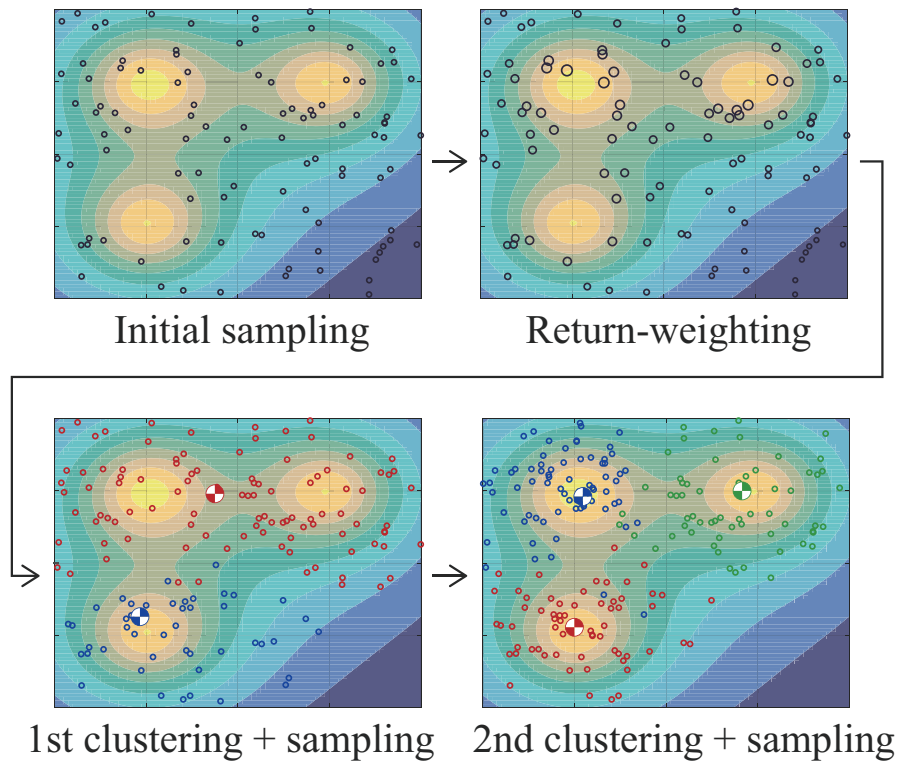


図 A.5 3つの峰を持つ2次元トイ問題における HSO の最適化過程.

## 付録 B

# 探索手順転移手法

### B.1 関連研究

#### B.1.1 人間による探索課題へのアプローチ

少ない手順で多くの情報を得る手法群は実験計画法と呼ばれており、様々な場面で用いられている。以下に示す 2 手法 [112] は人間が行う実験計画法の代表的なものである。

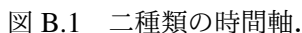
- タグチメソッド：直交表を用いて感度を解析する。
- 一時一事法：多くの変数により現象が決まる系において、1 度に 1 つのみ変数を変更して各変数の感度解析や、各変数と現象との間の詳細なメカニズムを考察する。

自然界の現象はエネルギー保存則や遺伝の法則などいくつかの基本的な法則に従って発生する。人間は様々な計測から背後の法則を理解したり、法則と法則をつなぎ合わせて新たな法則を生み出したりすることが出来るため、上記のような実験計画法により、要素還元的に法則を理解し一定の成功を挙げてきた。現状、人間が計算機上のアルゴリズムに対して持つ優位な特性として、メタな思考がある。メタな思考とは、法則を探し出す手順に対しても何らかの法則を見つけ出すといったような、与えられた問題の枠にとらわれない思考である。人間はこの思考法によって、次々と新しい発明を行うことが出来た。

#### B.1.2 機械による探索課題へのアプローチ

実験の手続きが複雑ではなく、計算機の性能を活かしたい探索課題に対しては、自動的に探索を行うアルゴリズムが開発されてきた。ただ、人間の手を介さず自動的に実験を行うといえども、グリッドサーチのような愚直な探索方法ではコストに見合わないような探索課題が多く存在している。例えば、時間や金銭的成本がかかる化学物質の合成パラメータ探索や、顧客に対して有効な広告提示方法の探索などがそれにあたる。

BOA の代表的な実装として、ガウス過程回帰 (Gaussian Process Regression: GPR)[113] を用いたものがある (GPR-BOA)[114]. ガウス過程回帰とはカーネル法を用いることで、複雑なモデルの近似が可能な回帰手法であり、近似に用いる関数群を選ぶステップを内部に持つことによって近似式の滑らかさを担保可能な手法である. また、通常の回帰手法と異なり、推定値のみならず、推定値の分散も同時に求まる. この性質を利用して、BOA の解候補の選択時に分散の大きな点を優先的に訪れることにより効果的な関数近似を行うことが出来る. 一方で、推定値が高い点を優先的に訪れることにより獲得報酬を高くすることも出来る.



GPR-BOA のような BOA の手法は一定の成果を上げているものの、ある探索課題に取り組んだ際に得られた知見を別の探索に用いることができていない．図 B.1 に示すように、ある一つの探索課題によって得られた知識を別の探索課題に用いることで効果的な探索が行える可能性がある．卑近な例で例えると、人間は機械の故障箇所を同定する時にまず点検すべき箇所の知識を持っており、機械のマニュアルにはそのような知識が掲載されている．従来の BOA の手法では、このような探索の定石に関する知識を利用することが出来ない．

### B.1.3 知識の転移手法

データ駆動的アプローチにおいては，なるべく抽象的な知識を抽出し，複数の課題に共通な知識を取り出す転移学習やメタ学習といった分野の研究が盛んである [115]．抽象的な知識は多くの課題において成立するため，各課題を解く際の事前知識として用いることができ，学習コストの低減につながる．先述のような，シミュレーションにおいてマニピュレータの把持方法を学び現実世界に転用する手法 [27] や，画像認識に共通して有用な特徴量抽出方法をローカルな画像認識課題に転用する手法 [116] などが存在する．

これらの手法の目標は，与えられた入力（観測値，画像）に対して適した出力（次の動作，クラスラベル）を返すような関数を近似することである．一方で，このような関数近似のためのデータ収集手法に関して，メタ的に知識を抽出してタスクの枠を超えて利用するという研究は少ない．

以上のような背景から，本論文では探索課題において効率的な探索手法に関する知識を抽出し，類似の探索課題に用いる手法の開発を目指す．実験においてはデータの用意のしやすさを考慮し仮想のタスクを設定し，提案手法のふるまいの考察を行う．

## B.2 Search Procedure Transfer Algorithm の提案

### B.2.1 問題設定

本論文では，探索課題において，関数近似誤差を小さく，探索時の平均獲得報酬を大きくすることを目標とする．一般的な最適化問題においては後者の獲得報酬最大化を主要な目標とし，関数近似のための探索というのは獲得報酬最大化の一手段に過ぎない．本論文では以下の 2 つの理由から関数近似誤差最小化も獲得報酬最大化と並ぶ目標と設定している．

1 つ目の理由は，関数全体の形状を知っておくことによる評価関数変動時の対応の柔軟性にある．例えば，機械加工においては，表面精度や 1 ワークあたりの加工速度，工具への負荷など複数の評価基準のもと評価基準の合計値が最も高くなる点が最適加工条件とされる．しかし，加工の需要が下がりワークの加工速度に対する優先度が下がるようなことが起こると，評価基準の合計が最大値となる最適加工条件は変動する．この際に，全体の関数形状を知っておけば最適加工条件更新の対応が容易となる．

2 つ目は，人間によるアルゴリズムの動作チェックが容易になるというものである．データ駆動的アプローチが社会で用いられるようになるにつれて，人間とは異なる基準で判断を行うこともあるアルゴリズムの判断 [117] を信用してよいのかという問題が盛んに議論されるようになってきた．関数近似を行っていれば，計算機上の近似関数を用いるこ

とで，人間が関数への入力を変えてみてシミュレーションを行うことが出来る．このシミュレーションにより，過学習が発生していないか，人間の経験則に反した予測をしていないかということを人間が直感的に確認することができる．

### B.2.2 タスクの定義

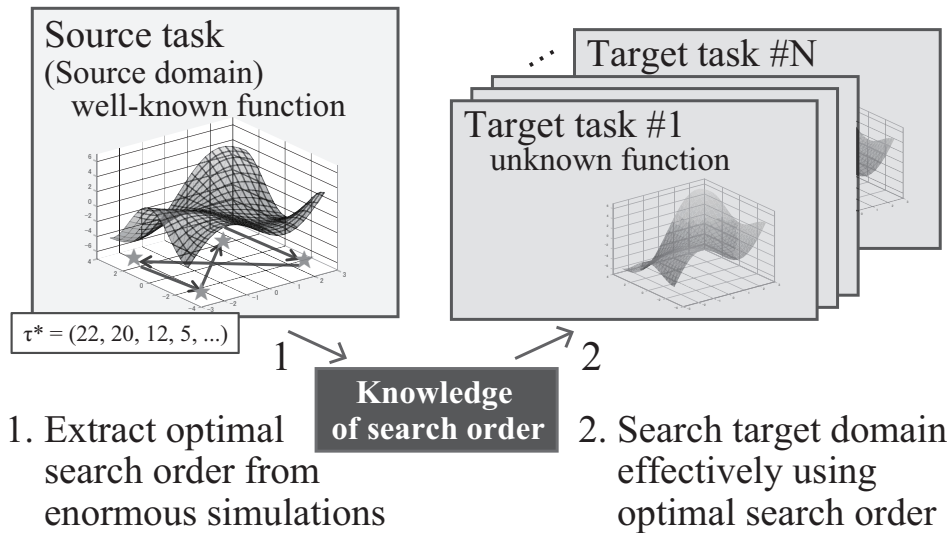


図 B.2 知識の転移の概念図．

本論文では，アルゴリズムが解くタスクとして関数の探索課題を設定している．図 B.2 のようにソースタスクを既知の関数，ターゲットタスクを形状が未知の関数とする．本手法の利用想定として，人間がタスクの類似性を判断しやすい問題を想定しており，具体的な適用先としては，回路設計 [39] におけるパラメータチューニング，液体流路の設計最適化など，人間がある程度背後の物理現象を理解しているが，高速なシミュレーションが難しいような問題を想定している．また，本手法はターゲットタスクのデータ取得コストが高く，ソースタスクのデータ取得コストが低いときに効果を発揮することを期待した手法である．例えば，目標が調整パラメータが 10 個ある機械のパラメータ最適化であり，人間が各パラメータの大まかなふるまいがわかっているときに用いることを想定している．そのような場合，ターゲットタスクは目標としている機械の評価関数の探索であり，ソースタスクには人間が持っている各パラメータの感度や影響度合いの知識を関数化したものを設定する．本手法は Fig. 2 のように関数形が既知で高速な計算が可能なソースタスクの探索を行い，探索に関する知識を抽出し，ターゲットタスクの探索を効率化することを目指している．図 B.3 は本論文の中で用いるソースタスク，ターゲットタスクの具体的な例を表している．図 B.3 では，どのタスク（関数）も 1 入力 1 出力の正弦波であるが，そ



れぞれの振幅や位相が異なる。

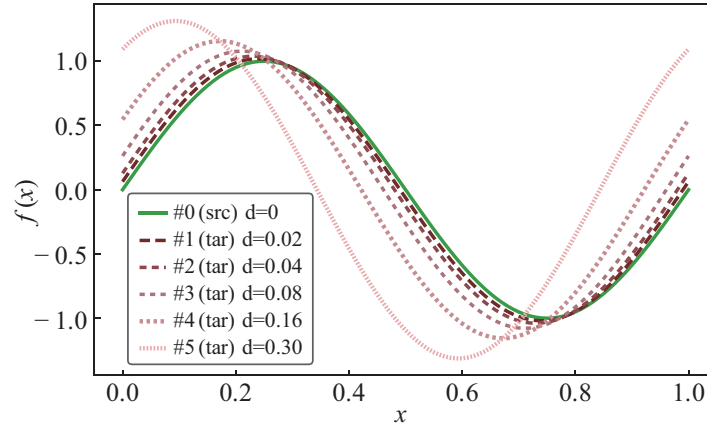


図 B.3 一変数関数のタスクとして用いる関数。

### B.2.3 探索知識の転移手法

データ取得コストが大きいタスクにおいて効率的な探索を行うために、類似する既知のタスクから探索手順を転移し、解きたいタスクに用いる手法、Search Procedure Transfer Algorithm を提案する。以降、当手法を SPT と呼称する。

B.4 に SPT アルゴリズムの流れを示す。アルゴリズムは 3 つの大きなステップにわけられる。1 つ目は探索手順生成ステップであり、何らかの指針に基づき訪れる探索点の手順を生成し、生成した探索手順と既知であるソースタスクを用いてシミュレーションを行う。本論文の実装では関数の定義域の範囲で乱数を用いて生成する。次のステップ 2 では、探索を行った結果と評価関数を元に、各手順の性能を評価する。評価関数には関数近似誤差  $e$  と平均獲得報酬  $r$  の二つの指標を用いる。どちらの指標の優先度を高くするかはユーザーが設定する。どちらかの指標の重みを 0 とすることも可能であるが、手法の性質を考えると  $e$  に対する重みを残しておくことにより、転移した手順に沿って探索した後の追加のサンプリングで用いることのできる情報が豊かになることが期待される。以上のステップでは 2 種類のハイパーパラメータ  $N_{\text{sample}}$ ,  $N_{\text{cand}}$  が存在する。 $N_{\text{sample}}$  は探訪する点の数であり、ソースタスク、ターゲットタスクの両方で共通である。 $N_{\text{cand}}$  はソースタスクで評価される探索手順候補の数であり、 $N_{\text{cand}}$  が大きくなるにつれて手順の中の最高性能が高くなることが期待される。最後のステップ 3 では性能が高いと判定された探索手順を転移させ、ターゲットタスクを探索する。本論文の実装においては  $N_{\text{cand}}$  個の手順の中から最も性能が高い手順を転移している。また、 $N_{\text{domain}}$  はターゲットタスクの個数を表す。本論文の実験においては 5 を用いた。

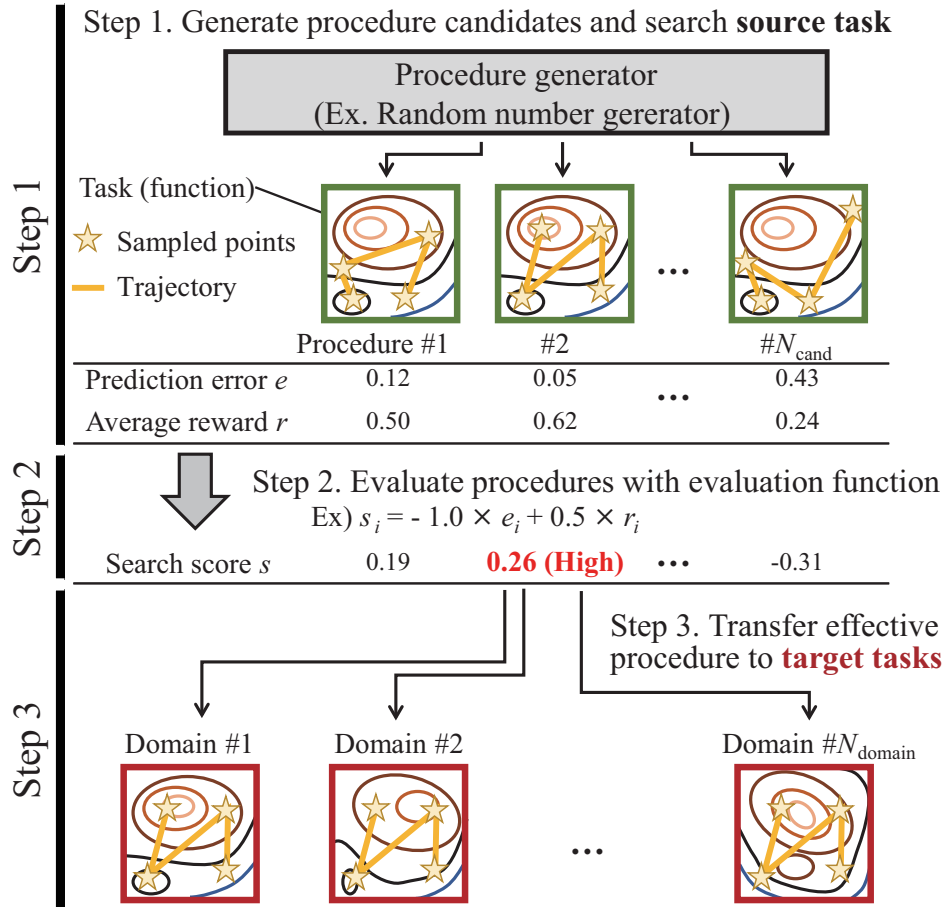


図 B.4 Search procedure transfer アルゴリズム。

今回のように探索に関するメタな知識を抽出することなく、ソースタスクの知識を利用してターゲットタスクの探索を効率化することも考えられる。例えば、ソースタスクの関数の値に応じてターゲットタスクの探索優先度を決める方法があるが、ハイパーパラメータを増やさず関数近似と最適化を両立した探索を行うことが難しい。

#### B.2.4 ソースタスクの設定

SPT アルゴリズムの実利用では、ソースタスクはターゲットタスクとなるべく似るように設定するのが良い。しかし、SPT アルゴリズムが用いられるような多くの問題ではターゲットタスクの関数形ははっきりとはわかっていないため、似るように設定することには困難が伴う。その困難への対策として、Domain randomization[27] の考え方を導入する。Domain randomization は多自由度マニピュレータによる物体の把持をシミュレーション内で訓練させる際の工夫である。その背景として、カメラにより観測される現実世界の物体の色合いや影は変動しうるが、その変動によって把持行為の本質はほぼ変化しないとい

う事実がある． Domain randomization では色合いや影の変動は本質的に意味がないという事実をアルゴリズムに対し暗に教え込むために，シミュレータ内の物体の色や影の落とし方をランダムに変えて学習を進める．

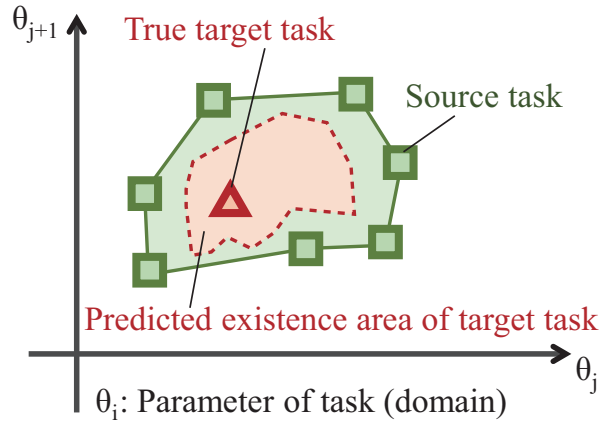


図 B.5 Domain randomization の適用イメージ．

本論文において SPT アルゴリズムの応用例として提案する， Domain randomization の概念を図 B.5 に示す．ターゲットタスクの具体的な関数形はわからないが，大まかにある特定の範囲（図 B.5 の破線内の部分）に入っているという知識は得られていると仮定する．機械の例で示すと，あるパラメータを増加させたときに性能が向上することが知られているが，線形な比例関係にあるのか，指数関数のような非線形な関係があるのかわかっていない場合などである．提案する手法では，その特定の範囲を取り囲むように複数のソースタスクを設定し，複数のソースタスクにおいて有効な手順を抽出する．上記の例では，あるパラメータ  $x_i$  が性能  $a$  に与える影響を  $a = bx_1 + c$ ,  $a = bx_1^2 + c$ ,  $a = be^{x_1} + c$  などとモデル化し，複数のソースタスクと設定することを想定している．そのようなソースタスクの設定により，ソースタスクによって囲まれるターゲットタスクの存在領域全体を通して有効な手順が抽出されることが期待される．つまり，人間が試行錯誤により経験則的に獲得した物理モデルが多少のあいまいさを持っていたとしても，そのあいまいさに従ってソースタスクを設定することにより，一定の探索性能を示す手順の算出が可能になると期待できる．一方，先述のソースタスクの関数の値に応じてターゲットタスクの探索優先度を決める方法では， Domain randomization のような複数のタスクの情報を使うことが難しい．

### B.3 実装

本節では実験に用いる各タスクの生成方法や SPT アルゴリズムをはじめとした実験に用いるアルゴリズムの実装，探索性能の評価基準について述べる．

### B.3.1 タスクの生成方法

多変数入力を受け付け、なめらかで多峰性のある関数を生成するために係数  $a$  と関数セット  $S_f = \{(ax)^2, 2^{ax}, \sin(ax), \cos(ax)\}$  の中から基本関数を変数の数だけ選択し、それらの総乗をとったものをソースタスクとした。ターゲットタスクはソースタスクの係数ベクトルを変形して生成した。なお、ここでの変形とはベクトルの要素に対してある実数 (本論文では 1.1) をべき乗した値を掛け合わせるものであり、べき指数は平均 0, 分散 1 の正規乱数発生器の出力を用いる。本論文では、実験のために 1 つのソースタスクに対して 5 つのターゲットタスクを生成した。また、論文内においてはソースタスクとターゲットタスクを合わせてタスクセットと呼ぶ。関数生成にあたり、関数同士の類似性を示す値である距離  $d$  という値を定義した。関数同士の距離  $d$  は 2 つの関数値の差を  $\mathbf{x}$  の定義域全体において足し合わせたものであり、式 (B.1) のように定義される。5 つ目のターゲットタスク (以降 #5 と表記) とソースタスクとの距離が既定の値になるように、係数ベクトルの変形量であるべき指数を調整し #5 を生成し、#1 ~ #4 の各タスクに関しては # の数が 1 小さくなるにつれて  $d$  が半分になるように係数ベクトルを変形して生成する。2 変数関数、既定の  $d = 0.16$  において生成された関数の例を B.1 に示す。また、図 B.3 は本節の生成方法とは係数の与え方が異なるものの、1 変数関数のタスクセットを図示したものである。

$$d = \int_D |g(\mathbf{x}) - f(\mathbf{x})| d\mathbf{x} \quad (\text{B.1})$$

$D$  : domain of a function,  $f, g$  : functions

表 B.1 タスクとして生成された 2 変数関数.

Domain #0 (source)	: $\sin(4.37x_0) \times (1.74x_1)^2$
Domain #1 (target)	: $\sin(4.31x_0) \times (1.76x_1)^2$
Domain #2 (target)	: $\sin(4.24x_0) \times (1.77x_1)^2$
Domain #3 (target)	: $\sin(4.11x_0) \times (1.81x_1)^2$
Domain #4 (target)	: $\sin(3.86x_0) \times (1.87x_1)^2$
Domain #5 (target)	: $\sin(3.41x_0) \times (2.01x_1)^2$

### B.3.2 アルゴリズム実装

SPT アルゴリズムの手順生成器には乱数発生器を用いた。SPT アルゴリズムでは #0 をソースタスク、#1 ~ #5 をターゲットタスクと設定した。比較するアルゴリズムとして、

SPT-best, SPT-worst, GPR-BOA の報酬優先, GPR-BOA 近似優先, ランダム探索の 5 つを設定した. SPT-best と SPT-worst はそれぞれソースタスクで最も性能が高かった手順と低かった手順を転移するものである. 報酬優先 (BOA-exploit) はベイズ的に探索を行う際に, 次の点で得られる報酬の期待値が最も高くなるように探索点を選ぶ方法, 近似優先 (BOA-explore) は近似関数の分散が大きい点を探索点として選ぶ方法である. BOA においては報酬優先と近似優先をバランスさせることも考えられるが, 今回の実験設定においては付録 A に示すように BOA の性能に大きな影響は与えなかったため, 比較においては単純な報酬優先, 近似優先のみを取り上げる. どのアルゴリズムにおいても関数近似には GPR を用いており, GPR のカーネルには放射基底関数 (RBF) を用いている. GPR には RBF の場合のみならずカーネルの設定値がハイパーパラメータとして存在する. ある問題において高い性能を示したアルゴリズムはほかの問題を解かせると低い性能を示し, 万能な最適化アルゴリズムは存在しない (No Free Lunch Theorem[53]) と言われており, 最適化アルゴリズムの開発時は, パラメータが手法の性能に及ぼす感度の考察が必要とされる. 一方で, SPT アルゴリズムは, アルゴリズムの利用想定を限定しており, ある単一のタスクを解くために他のタスクで得た知識を転用するという戦略をとっているため, 一般的な最適化手法とは用いる情報が異なる. SPT アルゴリズムにおいて, GPR のパラメータはタスク間で転用される情報に対して質的な影響を及ぼさないため, 本研究においては GPR のパラメータの変動が最終的な性能に及ぼす影響の考察は割愛する.

### B.3.3 性能評価方法

探索手順の性能評価及び, 探索の性能評価には図 B.4 中段および, 式 (B.2) に示すような近似誤差  $e$ , 平均報酬  $r$  の線形和を用いた. この性能評価指標を Search score と呼ぶ. 本論文では式 (B.2) 中の  $w_1, w_2$  にはそれぞれ  $-1, 0.5$  を用いた.

$$\text{Search score } s = w_1 e + w_2 r \quad (\text{B.2})$$

誤差や報酬に関してはタスク毎の比較を行いやすくするため, 算出の際に関数の値域が  $[0, 1]$  になるように値を正規化している. 近似誤差の算出では, 本章で述べた 2 つの関数同士の誤差算出関数式 (B.1) を用いて, 実際の関数と近似された関数の距離を求め, 近似誤差とした.

## B.4 実験

本章では実施した 3 つの実験の設定および結果について述べる. 実験の最も基本的な目標は探索手順の転移により探索が効率化されることの確認である. 1 つ目の実験は図 B.3 に示す 1 変数関数のタスクセットを用いた実験であり, 探索評価基準の妥当性や手順転移の

有効性を示す．2 つ目の実験は B.3.1 項で述べた生成方法によって生成した多変数関数を用いた実験であり，提案手法が用いられるような探索課題における手法の有効性を示す．3 つ目の実験は複数のソースタスクを用いた実験であり，Domain randomization の応用のための予備実験である．

#### B.4.1 1 変数関数を用いた実験

1 つ目の実験では，図 B.3 における #0 をソースタスクとして有効な探索手順の抽出を行い，探索手順を #1 ~ #5 のターゲットタスクの探索へ転用した．なお，ハイパーパラメータ 2 種，実験試行回数は次のように設定した．探索点数  $N_{\text{sample}} = 3$ ，手順候補数  $N_{\text{cand}} = 100$ ，試行数  $N_{\text{trial}} = 10$ ．

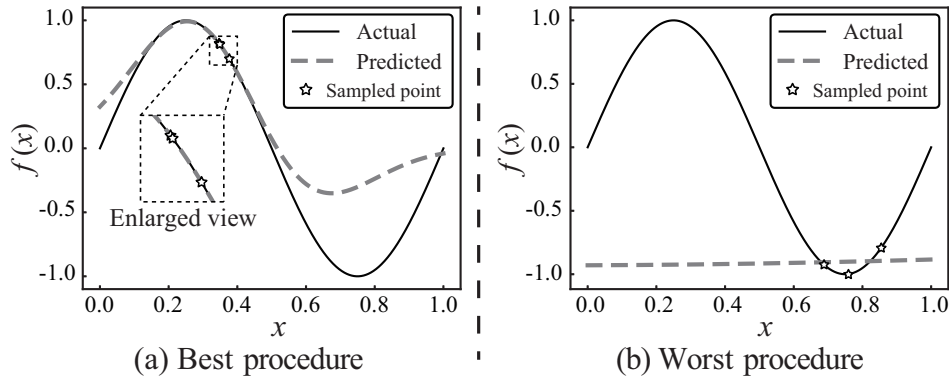


図 B.6 探索手順の具体例．

図 B.6 はソースタスクにおいて最も性能が高いと判定された探索手順 (図 B.6(a))，最も性能が低いと判定された探索手順 (図 B.6(b)) を示す．(a) は関数値の最高値に近い 3 点を探訪しているのに対し，(b) は最低値に近い 3 点を探訪している．また，(b) に比べ (a) は関数近似 (図 B.6 中破線) が上手くいっていることがわかる．

図 B.7 のグラフはターゲットタスクにて探索を行った際の各アルゴリズムの性能を示す．横軸がソースタスクとの距離  $d$  を表し，縦軸が近似誤差と平均報酬の線形和の探索性能を表す．図中の線は試行の平均値，パターンが描かれた領域は分散の大きさを表す．

ソースタスクからの距離が近い 4 つのタスクでは SPT-best が最高性能を示した．一番遠い  $d = 0.30$  のタスクでは BOA-exploit が最高性能を上げている．一方，SPT-worst はどの条件においても最も性能が悪く，想定通りの動作をしている．また，ソースタスクから近いタスク #1, #2 では SPT-best の性能の分散が小さく，手順の転移の性能が安定していると言える．以上のことから 1 次元の正弦波の探索課題において，関数同士の距離が  $d = 0.16$  以下の条件では探索手順の転移の有効性があると言える．



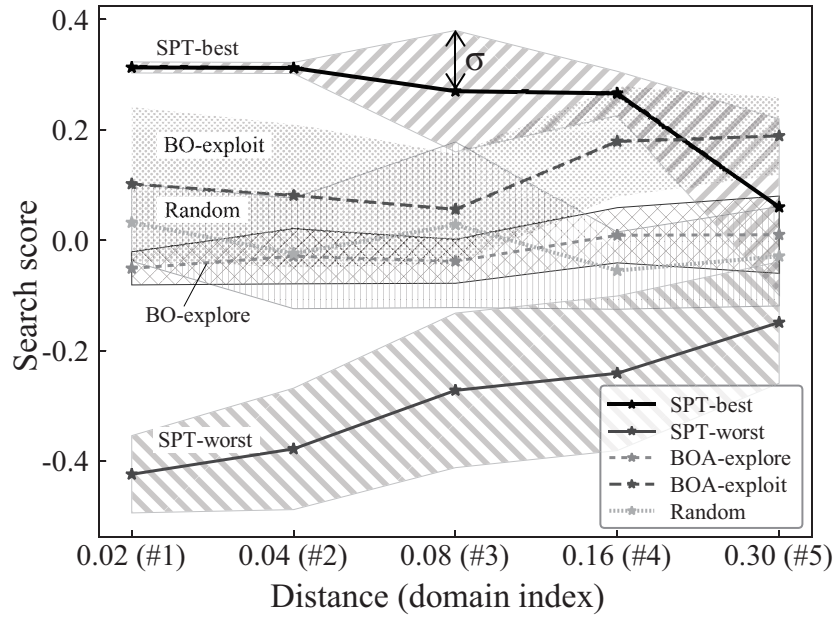


図 B.7 一変数関数タスクにおける各アルゴリズムの探索性能.

#### B.4.2 多変数関数を用いた実験

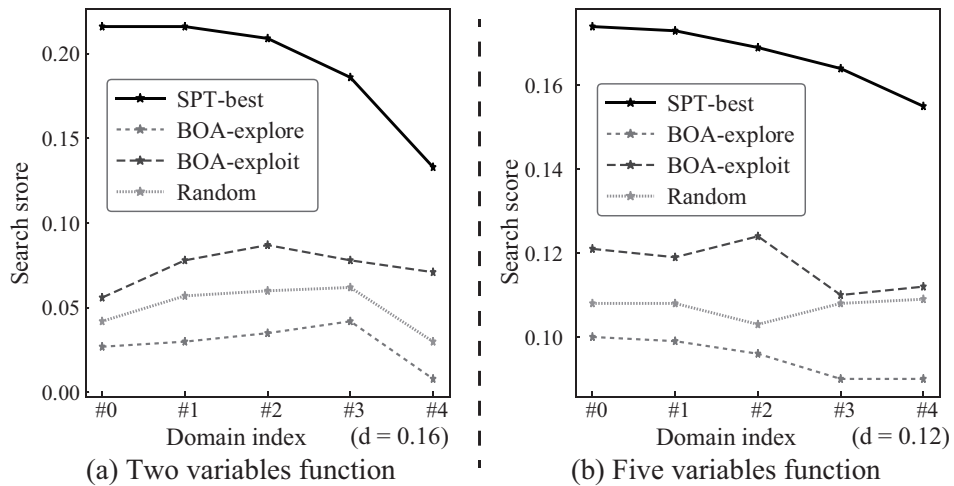


図 B.8 多変数関数タスクにおける各アルゴリズムの探索性能.

2 つ目の実験では B.3.1 項で述べた生成方法によって生成された 2 変数関数, 5 変数関数のタスクセットを用いて前節のような実験を行った. 5 変数関数の場合も B.1 と同様に B.3.1 にて示した関数セットの中から変数の数と同じ 5 個の関数を選び 5 変数関数を構成した. ソースタスクと #5 のターゲットタスクとの距離は 2 変数では 0.16, 5 変数で

は 0.12 と設定した．その他のハイパーパラメータ 2 種，実験試行回数 2 種は次のように設定した．探索点数は 2 変数と 5 変数で， $N_{\text{sample}} = 3, 5$ ，手順候補数  $N_{\text{cand}} = 100$ ，タスクセットの数  $N_{\text{set}} = 10$ ，セットごとの試行数  $N_{\text{trial}} = 10$ ．

図 B.8 のグラフにターゲットタスクにて探索を行った結果を示す．図 B.7 と同じく，横軸はソースタスクからの距離，縦軸は探索性能を表す．1 変数関数の場合と同様に SPT-best が最も高い探索性能を示し，探索性能はソースタスクとの距離が遠いタスクであればあるほど低下した．2 次元，5 次元の探索課題において，関数同士の距離が  $d = 0.16$ ， $d = 0.12$  以下の条件では探索手順の転移の有効性があると言える．

図 B.9 は生成する手順候補数  $N_{\text{cand}}$  と転移後の探索性能の関係を示す．2 変数関数を利用し，先ほどの実験条件の  $N_{\text{cand}}$  のみを変動させた．図 B.9 に示す性能は，前述の実験において SPT-best に次いで性能が高くなった BOA-exploit の性能を用いてタスク毎に正規化を施している．図 B.9 の点線右側が各  $N_{\text{cand}}$  における性能を示しており，生成する手順の候補数  $N_{\text{cand}}$  を増やすことで，探索性能を向上させることができるといえる．

一方，関数の変数の数や探索点数  $N_{\text{sample}}$  が増えるに従い，探索手順に関する探索空間の大きさが指数的に増加することから，多変数における手順の生成には，生成した手順と探索性能の対応から手順の良し悪しを決める要素を抽出するなどの対策が必要になると考えられる．

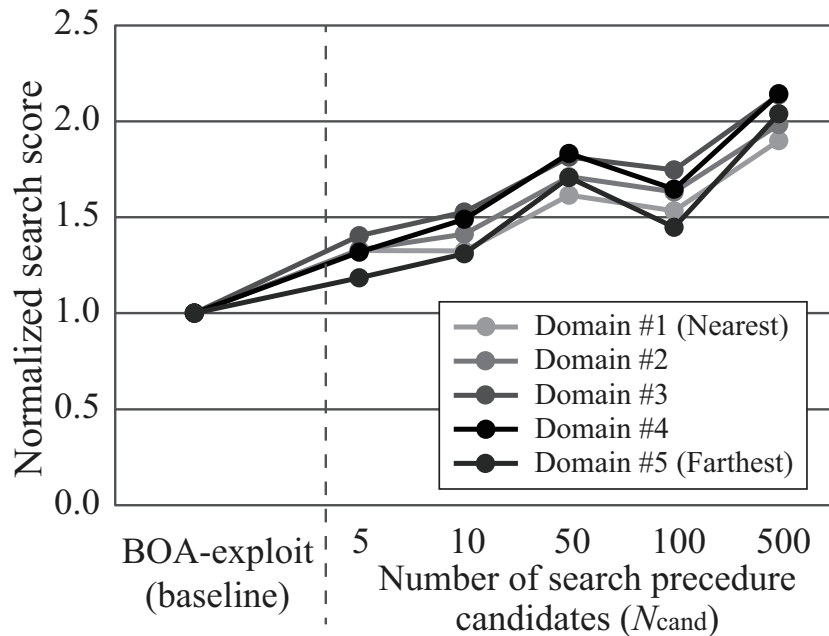


図 B.9 手順候補数が探索性能におよぼす効果．



### B.4.3 複数のソースタスクを利用する実験

Domain randomization (図 B.5) の応用では探索したいターゲットタスクの大体の予測をつけ、ターゲットタスクの周囲に複数のソースタスクを配置することにより、ターゲットタスクにおいて有効な探索手順の抽出を狙う。この実験では複数のソースタスクを利用した際に、ソースタスク間の領域全体に対して有効な手順を抽出できるかを確認する。

図 B.10 は比較を行う 2 手法を表したものである。図 B.10 上 SPT-single が 1 つのソースタスクを用いるもの、図 B.10 下 SPT-multi が 2 つのソースタスクを用いるものである。図 B.10 中の横軸は係数ベクトルの変形量を表しており、四角と三角はそれぞれソースタスクとターゲットタスクを表す。また、B.3.1 項にあるように、ターゲットタスク #1 から番号が 1 増えるにつれて距離  $d$  が倍増していく。実験には 2 変数関数を用い、各種パラメータは次のように設定した。  $d = 0.16$ ,  $N_{\text{sample}} = 3$ ,  $N_{\text{cand}} = 100$ ,  $N_{\text{set}} = 10$ ,  $N_{\text{trial}} = 5$ .

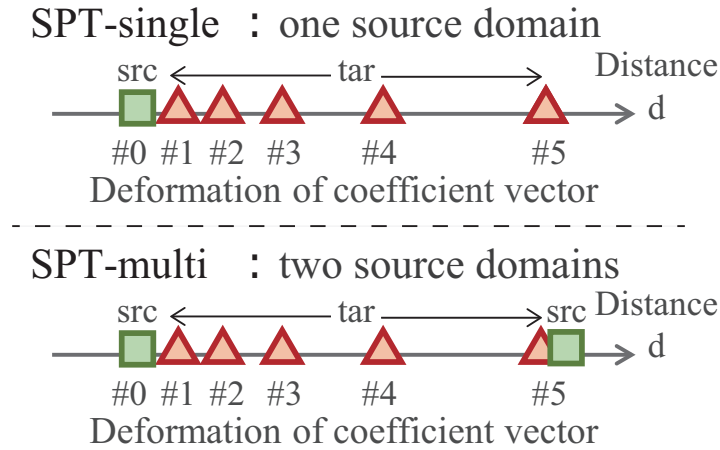


図 B.10 複数のドメインを利用した実験の設定.

図 B.11 に結果を示す。複数ソースタスクを利用した場合の SPT-multi (図中実線) は、全体を通して BOA-exploit や SPT-single の最低スコアよりも高いスコアになっており、複数のターゲットタスク間で安定した成績を示している。SPT-multi はソースタスクとして #0, #5 を用いているため、その間で安定してスコアが高いが、#0 に近い #1 ~ #3 では #0 のみをソースタスクとした SPT-single の性能が優っている。複数のソースタスクを設定し、ターゲットタスクを挟みこむことで、間のターゲットタスクに対して安定して高い性能を示す手順を抽出できているといえる。

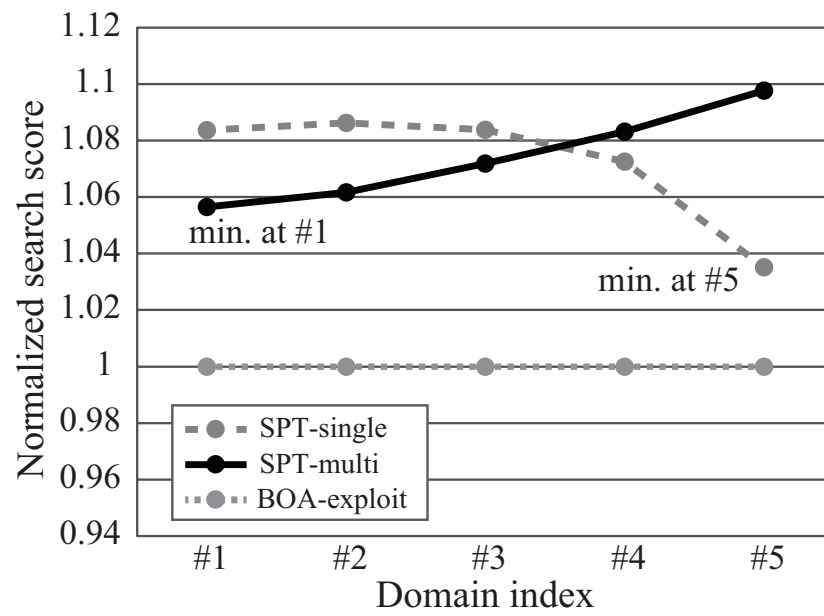


図 B.11 Domain randomization の適用結果.

## 参考文献

- [1] ブリタニカ国際大百科事典 小項目事典収録キーワード一覧- コトバンク.  
<https://kotobank.jp/dictionary/britannica/>, (参照 2020 年 3 月 1 日) .
- [2] S. Winchester, 梶山あゆみ. 精密への果てなき道：シリンダーからナノメートル EUV チップへ. 早川書房, 2019.
- [3] 売上高ランキング - 日経電子版. <https://www.nikkei.com/markets/ranking/>, (参照 2020 年 1 月 19 日) .
- [4] 畑村洋太郎, 実際の設計研究会. 実際の設計 改訂新版 機械設計の考え方と方法. 日刊工業新聞社, 2014.
- [5] 吉川弘之. 一般設計過程. 精密機械, Vol. 47, No. 4, pp. 405–410, 1981.
- [6] 間瀬久雄, 絹川博之, 森井洋, 中尾政之, 畑村洋太郎. 思考過程の思考展開図表現に基づく機械設計支援システム. 人工知能学会論文誌, Vol. 17, No. 1, pp. 94–103, 2002.
- [7] 村上存, 富山哲男. 設計と人工知能：設計シンポジウムの 20 年の歩みから (論文特集「設計知識の管理と高度利用」). 人工知能学会誌, Vol. 17, No. 1, pp. 84–90, 2002.
- [8] 野崎直行, 今野栄一, 佐藤満, 坂入慎, 澁谷利行, 金澤裕治, S. Georgescu. 製品設計における人工知能技術の応用 (特集ものづくり) – (ものづくりを支えるテクノロジー). *Fujitsu*, Vol. 67, No. 3, pp. 58–65, 2016.
- [9] 畑村洋太郎, 実際の設計研究会. 続・実際の設計 改訂新版機械設計に必要な知識とモデル. 日刊工業新聞社, 2017.
- [10] 松居吉哉. Analysis と Synthesis. 光学, Vol. 12, No. 6, p. 413, 1983.
- [11] 研究 - 東京大学工学部機械工学科 (機械 A). <http://www2.mech.t.u-tokyo.ac.jp/kikaiA/study/index.html>, (参照 2020 年 1 月 13 日) .
- [12] 日経ものづくり 2012 年 4 月号 ブレーク前夜 HondaJet 金型基地「沖縄」. 日経 BP 社, 2012.
- [13] 宮崎智彦. ガラパゴス化する日本の製造業:産業構造を破壊するアジア企業の脅威. 東洋経済新報社, 2008.
- [14] M. M. Waldrop. The chips are down for moore's law. *Nature News*, Vol. 530, No. 7589, p. 144, 2016.

- 
- [15] G. E. Moore, et al. Cramming more components onto integrated circuits, 1965.
  - [16] Matrix multiplication - Wikipedia. [https://en.wikipedia.org/wiki/Matrix\\_multiplication](https://en.wikipedia.org/wiki/Matrix_multiplication), (参照 2020 年 1 月 19 日) .
  - [17] J. Dean. The Deep Learning Revolution and Its Implications for Computer Architecture and Chip Design. 2019.
  - [18] D. Amodei, D. Hernandez, G. Sastry, J. Clark, G. Brockman, and I. Sutskever. AI and Compute. <https://en.wikipedia.org/wiki/ENIAC>, (参照 2020 年 2 月 3 日) , 2018.
  - [19] 山田宏尚, 大坪克俊. シミュレーションと数値計算の基礎. 共立出版, 2018.
  - [20] スーパーコンピューティング技術産業応用協議会. 「第 3 の科学」 : コンピュータ・シミュレーションが拓く産業の明日. アドバンスソフト, 2009.
  - [21] 課題実施機関 (重点・萌芽的) - 理化学研究所計算科学研究センター (R-CCS). <https://www.r-ccs.riken.jp/jp/post-k/pi/organizations>, (参照 2020 年 1 月 19 日) .
  - [22] ENIAC - Wikipedia. <https://en.wikipedia.org/wiki/ENIAC>, (参照 2020 年 2 月 16 日) .
  - [23] M. Fujino. Design and development of the hondajet. *Journal of aircraft*, Vol. 42, No. 3, pp. 755–764, 2005.
  - [24] 跡部隆, 蕪木英雄, 細川巖. 3 次元オイラー方程式の解の発散について (流れの非線形性と乱流の統計性質). 1998.
  - [25] 大富浩一, 羽藤武宏. 1DCAE によるものづくりの革新 (特集機械システム設計のためのシミュレーション技術). 東芝レビュー, Vol. 67, No. 7, pp. 7–10, 2012.
  - [26] 淡路敏之, 蒲地政文, 池田元美, 石川洋一. データ同化 : 観測・実験とモデルを融合するイノベーション. 京都大学学術出版会, 2009.
  - [27] J. Tobin, R. Fong, et al. Domain randomization for transferring deep neural networks from simulation to the real world. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2017.
  - [28] 鳥海不二夫. 強い AI・弱い AI : 研究者に聞く人工知能の実像. 丸善出版, 2017.
  - [29] 神畠敏弘, 鹿島久嗣. レクチャーシリーズ : 「人工知能の今」 [第 5 回] 機械学習分野の俯瞰と展望. 人工知能, Vol. 34, No. 6, pp. 905–915, 2019.
  - [30] 新田克己, 小野功. レクチャーシリーズ : 「人工知能の今」 [第 2 回] 問題解決, 探索, 最適化の基礎と展開. 人工知能, Vol. 34, No. 3, pp. 370–380, 2019.
  - [31] T. Kizaki, T. Fujii, M. Iwama, M. Shiraishi, N. Sugita, and S. Ahn. Design of a CFRP-elastomer composite with high stiffness and damping capability. *CIRP Annals*, Vol. 67, No. 1, pp. 413–418, 2018.
  - [32] M. Campbell, A. J. Hoane, and F. Hsu. Deep Blue. *Artificial Intelligence*, Vol. 134, No. 1-2, pp. 57–83, 2002.
  - [33] D. Silver, A. Huang, C. J. Maddison, et al. Mastering the game of Go with deep neural

- networks and tree search. *nature*, Vol. 529, No. 7587, p. 484, 2016.
- [34] D. Silver, J. Schrittwieser, et al. Mastering the game of Go without human knowledge. *Nature*, Vol. 550, No. 7676, pp. 354–359, 2017.
- [35] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, Vol. 362, No. 6419, pp. 1140–1144, 2018.
- [36] 羽生善治九段が語る AI で進化する将棋界。ディープラーニングで棋譜記録が変わる - Business Insider Japan. <https://www.businessinsider.jp/post-194875>, (参照 2020 年 1 月 21 日) .
- [37] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, Vol. 37, No. 4-5, pp. 421–436, 2018.
- [38] P. Abbeel, A. Coates, and A. Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, Vol. 29, No. 13, pp. 1608–1639, 2010.
- [39] Z. Wang, S. Rosa, L. Xie, B. Yang, S. Wang, et al. DEFO-NET: Learning Body Deformation Using Generative Adversarial Networks. In *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2440–2447. IEEE, 2018.
- [40] meviy (メヴィー) - 3DCAD データアップロードで即時見積、最短 1 日出荷 - ミスミ. <https://meviy.misumi-ec.com/>, (参照 2020 年 2 月 4 日) .
- [41] M. Schmidt and H. Lipson. Distilling Free-Form Natural Laws from Experimental Data. *Science*, Vol. 324, No. 5923, pp. 81–85, 2009.
- [42] G. Lample and F. Charton. Deep Learning for Symbolic Mathematics. 2019.
- [43] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, Vol. 9, pp. 2579–2605, 2008.
- [44] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, et al. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, pp. 529–533, 2015.
- [45] T. Osa, V. Tangkaratt, and M. Sugiyama. Hierarchical Reinforcement Learning via Advantage-Weighted Information Maximization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.
- [46] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross. Data-driven fluid simulations using regression forests. *ACM Trans. Graph.*, Vol. 34, No. 6, 2015.
- [47] M. Weisberg, 松王政浩. 科学とモデル：シミュレーションの哲学入門. 名古屋大学出版会, 2017.
- [48] 牧野光雄. 航空力学の基礎 (第 2 版). 産業図書, 1989.

- [49] 中村佳朗. CFD の歴史的レビューと展望 (第 28 回数値流体力学シンポジウム特集). 日本流体力学学会誌, Vol. 34, No. 2, pp. 59–65, 2015.
- [50] 大山聖, 小平剛央, 立川智章, 渡辺毅, 鈿持寛正. 多目的進化アルゴリズムと FEM 構造解析を用いた複数車種の車両構造同時設計最適化. 日本機械学会論文集, Vol. 85, No. 879, 2019.
- [51] 杉村和之. 遠心送風機の空力形状最適設計と知識マイニング. ターボ機械, Vol. 34, No. 6, pp. 357–365, 2006.
- [52] 大関真之. 人間と機械の協業の未来 - 専門家の知識を機械学習に統合する - ZDNet Japan. <https://japan.zdnet.com/article/35082453/4/>, (参照 2020 年 1 月 30 日) .
- [53] D. H. Wolpert, W. G. Macready, et al. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, Vol. 1, No. 1, pp. 67–82, 1997.
- [54] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pp. 6389–6399, 2018.
- [55] 神瀧敏弘. 転移学習. 人工知能学会誌, Vol. 25, No. 4, pp. 572–580, 2010.
- [56] 唐澤宏之, 福井類, 割澤伸一. 類似タスクにおける経験に基づいた探索アルゴリズム. 計測自動制御学会論文集, Vol. 55, No. 11, pp. 709–716, 2019.
- [57] S. Freek and O. Sigaud. Policy Improvement Methods: Between Black-Box Optimization and Episodic Reinforcement Learning. *Journées Francophones Planification, Décision, et Apprentissage pour la conduite de systèmes*, 2012.
- [58] P. de Boer, D. Kroese, et al. A Tutorial on the Cross-Entropy Method. *Annals of operations research*, Vol. 134, No. 1, pp. 19–67, 2005.
- [59] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, Vol. 9, No. 2, pp. 159–195, 2001.
- [60] 若佐裕治. データ駆動型制御における非線形補償と最適化手法. 計測と制御, Vol. 52, No. 10, pp. 872–877, 2013.
- [61] 小林孝, 野村武秀, 神藤正樹, 矢尾彰, 大串哲朗, 後藤明広. 応答曲面法とモンテカルロ法を用いたロバスト最適化設計手法. 日本機械学会論文集 B 編, Vol. 71, No. 703, pp. 914–921, 2005.
- [62] 中根昌克, 今野友和, 池谷弦, 石川芳男, 船木一幸, 都木恭一郎. ソフト最適化手法による MPD スラスタのノズル形状最適化. 日本航空宇宙学会論文集, Vol. 54, No. 631, pp. 360–366, 2006.
- [63] 森野裕行, 竹中啓三, 畑中圭太. 環境適応型高性能小型航空機設計における多分野統合最適化技術 (MDO) の適用. 三菱重工技報, Vol. 42, No. 5, pp. 216–219, 2005.
- [64] 水上裕登, 松下裕一, 花木義麿, 大西昇, 杉江昇. 旋削加工における遺伝的アルゴリ

- ズムを用いた同時加工順序の決定法. 計測自動制御学会論文集, Vol. 31, No. 5, pp. 591–597, 1995.
- [65] M. Hao, 水垣善夫, 吉川浩一. 遺伝的アルゴリズムを用いた工具経路生成法の一手法. 日本機械学会論文集 C 編, Vol. 63, No. 608, pp. 1385–1390, 1997.
- [66] 宮河秀和, 山本暁洋, 二見茂, 住本正, 後藤聡, 中村征俊. 工作機械の輪郭制御における遺伝的アルゴリズムを用いた最適ゲインチューニング. 精密工学会誌, Vol. 67, No. 4, pp. 591–596, 2001.
- [67] 中川泰忠. 両面研磨における摩擦距離のウエハ面内分布均一化のための加工条件最適化. 精密工学会誌論文集, Vol. 72, No. 5, pp. 641–646, 2006.
- [68] 轟章, 石川哲也. GA を用いた複合材料の積層構成最適化における応答曲面の実験計画法. 日本機械学会論文集 A 編, Vol. 66, No. 645, pp. 978–985, 2000.
- [69] 山本崇史, 丸山新一, 西脇眞二, 吉村允孝. 閉空間における音圧応答最小化を目的とした吸音材のトポロジー最適化 (機械力学, 計測, 自動制御). 日本機械学会論文集 C 編, Vol. 75, No. 754, pp. 1647–1655, 2009.
- [70] 馬場一郎, 東俊一, 杉江俊治. ハイブリッド自動車の燃費最適化制御器設計. 計測自動制御学会論文集, Vol. 49, No. 9, pp. 887–894, 2013.
- [71] 山崎光悦, 大塚貴康, 韓晶, 長谷川貴志, 西山貞雄. プレス加工によるアルミ缶蓋の成形過程最適化に関する研究. 日本機械学会論文集 C 編, Vol. 76, No. 771, pp. 2839–2846, 2010.
- [72] 北村充, 濱田邦裕, 鈴木博喜, 矢野佳苗, 徳岡慶輔, 大槻康明. 船体中央断面最適設計の実用化と高速最適化システムの開発に関する研究. 日本船舶海洋工学会論文集, Vol. 6, pp. 141–149, 2007.
- [73] 岡本直也, 鈴木和夫, 日野孝則, 増田聖始. BHP 最小化を目的とした船尾船型最適化手法 (第 1 報). 日本船舶海洋工学会論文集, Vol. 23, pp. 19–26, 2016.
- [74] 光石衛, 奥村努, 杉田直彦, 長尾高明, 畑村洋太郎. ニューラル・ネットワークと遺伝的アルゴリズムとを用いた高精度マシニングセンタの開発. 日本機械学会論文集 C 編, Vol. 61, No. 591, pp. 4517–4522, 1995.
- [75] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.
- [76] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018.
- [77] 守本佳郎. AT の変速機構及び制御入門. グランプリ出版, 2011.
- [78] 黒河周平, 西村幸久. 日本機械学会講習会教材 歯車加工の基礎と応用-自動車生産を支える歯車加工の最新技術まで-. No. 16-169. 日本機械学会, 2016.
- [79] *Motor fan illustrated*. モーターファン別冊, No. 124. 三栄書房, 2017.

- 
- [80] 射場本正彦, 黒岩弘, 佐藤一彦, 箕輪利通. スムーズな走行を指向する変速機制御. 日立評論, Vol. 77, No. 2, pp. 27–32, 1995.
- [81] 大野木博章, 久世隆ほか. トランスミッション動力伝達のシミュレーションによる最適化. 品質工学, Vol. 18, No. 2, pp. 88–94, 2010.
- [82] A. Dutta, Y. Zhong, et al. Model-based and model-free learning strategies for wet clutch control. *Mechatronics*, Vol. 24, No. 8, pp. 1008–1020, 2014.
- [83] H. Jian, W. Wei, et al. Optimization of a pressure control valve for high power automatic transmission considering stability. *Mechanical Systems and Signal Processing*, Vol. 101, pp. 182–196, 2018.
- [84] M. Roozegar and J. Angeles. Gear-shifting in a novel modular multi-speed transmission for electric vehicles using linear quadratic integral control. *Mechanism and Machine Theory*, Vol. 128, pp. 359–367, 2018.
- [85] F. Mesmer, T. Szabo, and K. Graichen. Feedforward control of a hydraulic clutch actuation path. In *Proceedings of European Control Conference (ECC)*, pp. 620–626, 2019.
- [86] K. V. Vaerenbergh, A. Rodriguez, M. Gagliolo, P. Vrancx ほか. Improving wet clutch engagement with reinforcement learning. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2012.
- [87] D. Ha. Reinforcement learning for improving agent design. *Artificial Life*, Vol. 25, No. 4, pp. 352–365, 2019.
- [88] T. Osa and M. Sugiyama. Hierarchical policy search via return-weighted density estimation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3860–3867, 2018.
- [89] H. Karasawa, T. Kanemaki, K. Oomae, R. Fukui, M. Nakao, and T. Osa. Hierarchical stochastic optimization with application to parameter tuning for electronically controlled transmissions. *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, pp. 628–635, 2020.
- [90] F. Stulp and O. Sigaud. Path Integral Policy Improvement with Covariance Matrix Adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.
- [91] N. Hansen and A. Auger. Principled Design of Continuous Stochastic Search: From Theory to Practice. In *Theory and principled methods for the design of metaheuristics*, pp. 145–180. Springer, 2014.
- [92] C. Daniel, G. Neumann, et al. Hierarchical Relative Entropy Policy Search. *Journal of Machine Learning Research*, Vol. 17, No. 1, pp. 3190–3239, 2016.



- [93] 谷藤克也. 乗り心地管理のための振動解析システムの開発: 第1報, 振動解析システムの概要. 日本機械学会論文集C編, Vol. 52, No. 481, pp. 2405–2408, 1986.
- [94] 帯川利之, 笹原弘之, 齊藤卓志, 谷泰弘, 平田敦, 吉野雅彦, 池野順一, 大竹尚登, 国枝正典, 長藤圭介, 新野俊樹. はじめての生産加工学. 講談社, 2016.
- [95] 砥粒 - 株式会社ノリタケカンパニーリミテド.  
<https://www.noritake.co.jp/products/support/detail/25/>, (参照 2020 年 2 月 9 日).
- [96] 海野邦昭. 絵とき「研削加工」基礎のきそ. 日刊工業新聞社, 2006.
- [97] F. Hashimoto, H. Yamaguchi, P. Krajnik, K. Wegener, R. Chaudhari, et al. Abrasive fine-finishing technology. *CIRP Annals*, Vol. 65, No. 2, pp. 597–620, 2016.
- [98] G. Xiao and S. Malkin. On-Line Optimization for Internal Plunge Grinding. *CIRP Annals*, Vol. 45, No. 1, pp. 287–292, 1996.
- [99] E. Brinksmeier, H. K. Tönshoff, C. Czenkusch, and C. Heinzl. Modelling and optimization of grinding processes. *Journal of Intelligent Manufacturing*, Vol. 9, No. 4, pp. 303–314, 1998.
- [100] R. Saravanan, P. Asokan, and M. Sachidanandam. A multi-objective genetic algorithm (GA) approach for optimization of surface grinding operations. *International Journal of Machine Tools and Manufacture*, Vol. 42, No. 12, pp. 1327–1334, 2002.
- [101] M. Sedighi and D. Afshari. Creep feed grinding optimization by an integrated GA-NN system. *Journal of Intelligent Manufacturing*, Vol. 21, No. 6, pp. 657–663, 2010.
- [102] M. N. Dhavlikar, M. S. Kulkarni, and V. Mariappan. Combined Taguchi and dual response method for optimization of a centerless grinding operation. *Journal of Materials Processing Technology*, Vol. 132, No. 1-3, pp. 90–94, 2003.
- [103] J. Kwak. Application of Taguchi and response surface methodologies for geometric error in surface grinding process. *International Journal of Machine Tools and Manufacture*, Vol. 45, No. 3, pp. 327–334, 2005.
- [104] N. Alagumurthi, K. Palaniradja, and V. Soundararajan. Optimization of Grinding Process Through Design of Experiment (DOE)—A Comparative Study. *Materials and Manufacturing Processes*, Vol. 21, No. 1, pp. 19–21, 2006.
- [105] C. Dietz, K. Wegener, and W. Thyssen. Continuous generating grinding: Machine tool optimisation by coupled manufacturing simulation. *Journal of Manufacturing Processes*, Vol. 23, pp. 211–221, 2016.
- [106] H. N. Li and D. Axinte. On a stochastically grain-discretised model for 2D/3D temperature mapping prediction in grinding. *International Journal of Machine Tools and Manufacture*, Vol. 116, pp. 60–76, 2017.

- 
- [107] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.
  - [108] N. Srinivas, A. Krause, et al. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.
  - [109] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.
  - [110] P. Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, Vol. 3, pp. 397–422, 2002.
  - [111] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv preprint arXiv:1703.03864*, 2017.
  - [112] E. Alpaydin. *Introduction to machine learning*. MIT press, 2009.
  - [113] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
  - [114] E. Contal, V. Perchet, and N. Vayatis. Gaussian process optimization with mutual information. In *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML)*, 2014.
  - [115] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, Vol. 22, No. 10, pp. 1345–1359, 2009.
  - [116] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, Vol. 313, No. 5786, pp. 504–507, 2006.
  - [117] C. Szegedy, W. Zaremba, et al. Intriguing properties of neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

## 謝辞

本研究を進めるに当たり，たくさんのご支援とご協力を頂きました．このようなご支援・ご協力なしでは本論文は成り立たなかったと思っております．感謝の意を表し，この場にて謝辞を述べさせていただきます．

■福井准教授 指導教員の福井先生からは博士課程にチャレンジするという貴重なチャンスをいただきました．博士課程へのチャレンジは不確定な部分も多くあるものですが，私の場合はかなり身につけられたことが多く，実りある時間となりました．博士課程終盤においては，調子の浮き沈み激しい私に対しても粘り強い指導をいただきました．このほかにも，福井先生からは学部時代から実に多くのことを教えていただきました．その中には一生涯残るような重要な教えも少なくありません．どうもありがとうございました．

■割澤教授 割澤先生からは研究室の長として様々な援助をいただきました．私の研究に対しても，博士研究の進め方や博士論文に書き方についていくつかの重要なアドバイスをいただきました．割澤研究室では，幅広い研究テーマを扱える土壌が培われており，そのことが自分の研究の考察を深めたり，技術者として技術を見る目を養うのにつながったと考えています．ありがとうございました．

■論文審査の先生方 本論文を審査いただいている，岡田先生，杉田先生，陳先生，福井先生，割澤先生には様々な視点から，ご指導頂きました．本論文の3章は，先生方のご指摘がなくては書き上げることができなかったと考えております．個々の事例を一般化し，一般化したものを別の研究の中で位置づけて語るという研究活動の基本を，博士課程の終盤でおさらいすることができたのはとても良い経験になりました．ありがとうございました．

■伴助教 伴先生とは共同で研究をしていた時期もあり，様々なことを学ばせていただきました．同じテーマをフラットな立場で扱うことができていたので，研究の各場面における行動方を効率良く学べたと思っています．そのほかにも，伴先生とは事ある毎に議論をしたり，体を動かしたりしていた記憶があります．これらはとても良い刺激になっていました．ありがとうございました．

■研究室の秘書様方 割澤研究室の秘書の久米さん、庵さん、本木さんには様々なサポートをいただきました。今では、事務だけではなく、日々の何気ない雑談によってもサポートされていたのだと思っています。ありがとうございました。

■中尾教授 中尾先生には研究室が変わった後も、様々な面でお世話になりました。中尾研究室で学部、修士課程の時に教わった考え方が、博士課程においても大いに役に立ちました。本論文との直接的な関わりとしては、まとめ方について議論をしていただいたり、共同研究の場を用意いただいたというものがあります。ありがとうございました。

■九州工業大学 長准教授 長先生からは、1年という短い間でしたが、強化学習周りの技術だけではなく、論文執筆法、研究の評価の仕方など様々なことを教えていただきました。また、最適化を本研究の柱とできたのは、長先生との議論の中で最適化手法の種類や限界について教わっていたからだと思います。ありがとうございました。

■木崎特任助教 木崎先生には、研究の方向性の相談や加工実験の準備などで度々お世話になりました。本論文とのかかわりとしては、5章において登場した研削加工の数値シミュレーションの開発には沢山の支援を頂きました。この他にも、自ら手を動かして様々な種類の研究を遂行していく姿にはとても勇気づけられていました。ありがとうございました。

■共同研究先の皆様 共同研究先の三菱重工工作機械と小松製作所の皆さまからは本研究を進めるにあたり様々なご協力をいただきました。共同研究を通して、大学内ではなかなか得られない、設計現場の声や、実際の設計の進め方という知識を得ることができました。また、実際の現場で機械を触ったり動かしたりすることは、机上における議論にはない説得力があり、研究を進める上での大きなモチベーションとなりました。ありがとうございました。

■実際の設計研究会の皆様 実際の設計研究会の皆様には、研究会の場を通して様々なことを教わりました。本研究との関わりとしては、設計における要求や設計法を設計するメタな考え方について教わりました。研究会において本研究について議論したわけではありませんが、実際研のコアとなる考え方が私の頭の中にじわりじわりと浸透し、実際研の考え方を踏まえたような本論文が出来上がったのだと思っています。ありがとうございました。

■割澤研 2018 年修士卒 根本君 研究室の後輩であった根本君には研究室時代からお世話になっています。根本君とは様々な議論をかわしており、その内容が本論文のあらゆる箇所に活かされています。また、自らの業務で忙しい中、本論文へのフィードバックをいた

だきました。ありがとうございました。

■研究室の学生の皆様 研究室の学生の皆様とは研究の議論であったり，課外活動であったり様々な面でお世話になりました。本研究も皆様の存在なしでは為しえなかったと考えています。ありがとうございました。これからも居心地の良い研究室を創ってってください。

最後に本研究をサポートしてくださった全ての方，友人，家族に御礼を申し上げます。ありがとうございました。

令和2年8月 唐澤 宏之

## 研究業績リスト

## 研究業績リスト（2020 年 8 月 6 日時点）

新領域創成科学研究科 人間環境学専攻

割澤・福井・米谷研究室

47-177616 唐澤 宏之

### 論文誌(英文)

1. **Hiroyuki Karasawa**, Tomohiro Kanemaki, Kei Oomae, Rui Fukui, Masayuki Nakao, Takayuki Osa, Hierarchical Stochastic Optimization with Application to Parameter Tuning for Electronically Controlled Transmissions, IEEE Robotics and Automation Letters, Vol.5, No.2, pp.628-635, 2020.

### 論文誌(和文)

1. **唐澤 宏之**, 福井 類, 割澤 伸一, 類似タスクにおける経験に基づいた探索アルゴリズム —人工データを用いた探索性能評価—, 計測自動制御学会論文集, Vol.55, No.11, pp.709-716, 2019.

### 国際学会(査読付)

1. **Hiroyuki Karasawa**, Takuro Okubo, Rui Fukui, Masayuki Nakao and Yuichi Kodama, Wall-Contacting Drive for Underground Automatic Transport Vehicle, International Symposium on Experimental Robotics (ISER), 2016.
2. Yuki Ban, **Hiroyuki Karasawa**, Rui Fukui and Shin'ichi Warisawa, Relaxushion: Controlling the Rhythm of Breathing for Relaxation by Overwriting Somatic Sensation, SIGGRAPH Asia 2018, 2018.
3. **Hiroyuki Karasawa**, Takehito Yoshida, Rui Fukui, Toru Kizaki, Kohei Fujii and Shin'ichi Warisawa, Analysis of Chip Shape Distribution Using Image Processing Technology to Estimate Wearing Condition of Gear Grinding Wheel, The 52nd CIRP Conference on Manufacturing Systems (CIRP CMS), 2019.
4. **Hiroyuki Karasawa**, Rui Fukui, Masahiko Watanabe and Shin'ichi Warisawa, Simultaneous Recognition of Hand Shape and Two-Axis Wrist Bending Using Wearable Wrist Contour Measuring Device, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2019.
5. **Hiroyuki Karasawa**, Tomohiro Kanemaki, Kei Oomae, Rui Fukui, Masayuki Nakao, Takayuki Osa, Hierarchical Stochastic Optimization and its Application in Parameter Tuning for Electronically Controlled Transmissions, The International Conference on Robotics and Automation (ICRA), 2020.

### 国内学会(査読付)

1. **唐澤 宏之**, 福井 類, 割澤 伸一, 類似タスクにおける有効探索手順の転移による探索の効率化, 第 24 回ロボティクスシンポジア, Mar., 2019.
2. **唐澤 宏之**, 金牧 知宏, 大前 圭, 中尾 政之, 長 隆之, 福井 類, 階層型確率的最適化によるトランスミッション制御パラメータの学習と実機による評価, 第 25 回ロボティクスシンポジア, Mar., 2020.

3. 福井 類, 山之城 晃士, 立山 尚樹, 唐澤 宏之, 沖芝 俊祐, 割澤 伸一, 装着型手首凹凸計測デバイスによる握りこみ強さ推定の試み — 握りこみ強さの変化による手首凹凸形状への影響分析 —, 第 25 回ロボティクスシンポジア, Mar., 2020.
4. 吉田 健人, 福井 類, 唐澤 宏之, 藤井 浩平, 割澤 伸一, 円筒研削砥石の摩耗状態推定のための微小切屑の画像取得と幾何特徴抽出, 第 25 回ロボティクスシンポジア, Mar., 2020.

#### 国内学会(査読無)

1. 福井 類, 唐澤 宏之, 早川 直樹, 渡邊 匡彦, 中尾 政之, 装着型手首凹凸計測デバイスによる手形状と手首屈曲状態の同時認識, 日本機械学会 ロボティクス・メカトロニクス講演会 2015, May, 2015.
2. 唐澤 宏之, 福井 類, 木崎 通, アイヤル クリティカ, 中尾 政之, 画像処理技術を用いた歯車研削表面品質の高速検査の試み, 日本機械学会 ロボティクス・メカトロニクス講演会 2017, May, 2017.
3. 福井 類, 沖芝 俊祐, 唐澤 宏之, 割澤 伸一, ウェアラブルディスプレイのための手首凹凸計測による手指の動的運動認識, 日本機械学会 ロボティクス・メカトロニクス講演会 2017, May, 2017.
4. 福井 類, 塔下 大嗣, 大久保 拓郎, 唐澤 宏之, 割澤 伸一, 坑内自動運搬車両のための壁面接触を利用した高速走行法 — ロングレンジ距離センサの追加による壁面障害回避性能の向上 —, 日本機械学会 ロボティクス・メカトロニクス講演会 2017, May, 2017.
5. 唐澤 宏之, 福井 類, 木崎 通, 藤井 浩平, 割澤 伸一, 中尾 政之, 歯車研削砥石の摩耗状態推定のための磁気凝集現象を用いた切屑形状推定, 精密工学会 2017 年度秋季大会, Sep., 2017.
6. 唐澤 宏之, 金牧 知宏, 大前 圭, 福井 類, 中尾 政之, 長 隆之, 階層型確率的最適化を用いたトランスミッションの制御パラメータの学習 — シミュレータによる条件候補の選定 —, 日本機械学会 ロボティクス・メカトロニクス講演会 2019, June, 2019.
7. 福岡 勇児, 福井 類, 唐澤 宏之, 吉田 健人, 大石 浩史, 三品 俊二, 中嶋 梓, 大型工作機械におけるワーク重量推定と動力学解析による制御パラメータ調整法, 精密工学会 2019 年度春季大会, Mar., 2020.

#### 執筆

1. 実際の設計選書 続・実際の設計 [改訂新版] 機械設計に必要な知識とモデル 編著 畑村洋太郎

#### 特許

1. シート材搬送装置、およびシート材搬送方法, 特開 2017-193407, 2016 年 4 月 20 日出願, 第 3 発明者
2. 研削盤, 特開 2019-42857, 2017 年 8 月 31 日出願, 第 4 発明者

#### 受賞等

1. SIGGRAPH Asia 2018 Emerging Technologies Best Demo Award voted by Committee



以上

博士論文

令和2年8月6日 提出

唐澤 宏之