

データ駆動型スマート塗装システムの構築と実装

(Construction and Implementation of a Data-Driven Smart
Painting System)

広島大学 先進理工系科学研究科

先進理工系科学専攻 電気システム制御プログラム

D201640 山本 翔平

2024年3月

目次

第1章	序論	1
1.1	研究の背景	1
1.2	本論文の目的と各章の構成	7
第2章	Just-in-Time 法に基づく動作条件決定機構の構築	11
2.1	緒言	11
2.2	Just-in-Time 法に基づく動作条件決定機構	12
2.2.1	システムの記述	13
2.2.2	初期データベースの構築	14
2.2.3	近傍データの抽出	14
2.2.4	分類境界の作成	15
2.2.5	有効な良好データの決定	17
2.2.6	動作条件の計算	17
2.3	数値シミュレーション	18
2.3.1	対象システム	18
2.3.2	シミュレーション結果	20
2.4	結言	28
第3章	データベース駆動型塗装品質予測器の構築	29
3.1	緒言	29
3.2	データベース駆動型塗装品質予測器	29
3.2.1	システムの記述	30

3.2.2	初期データベースの構築	30
3.2.3	近傍データの抽出	31
3.2.4	予測値の計算	32
3.2.5	修正量の更新	33
3.3	数値シミュレーション	34
3.3.1	対象システム	34
3.3.2	シミュレーション結果	35
3.4	結言	41
第4章	データ駆動型スマート塗装システムの構築と実装	42
4.1	緒言	42
4.2	データ駆動型スマート塗装システム	42
4.2.1	近傍データの抽出	44
4.2.2	動作条件の計算	46
4.2.3	予測値の計算	47
4.2.4	近傍データの修正	48
4.2.5	修正量の更新	48
4.3	数値シミュレーション	49
4.3.1	対象システム	49
4.3.2	シミュレーション結果	50
4.4	実装と評価検証	65
4.4.1	機器構成	66
4.4.2	実装準備	68
4.4.3	実装条件	69
4.4.4	実装結果	71

4.5	結言	83
第5章	結論	84
	付録	88
	参考文献	93
	引用既発表論文	96
	A. 学術雑誌等掲載論文	96
	B. 国際学会講演論文	96
	その他既発表論文	97
	C. 国際学会講演論文	97
	謝辞	98

第1章 序論

1.1 研究背景

日本の全産業の就業者数は、新型コロナウイルスの感染拡大による影響もあり、Fig. 1.1に示すように2019年から2020年にかけて減少したものの、2022年には6,723万人と増加している[1]。一方、全産業に占める製造業の就業者の割合は、2002年には19.0%であったが、2006年以降は低下傾向で推移しており、2022年には15.5%となっている。さらに、日本の製造業の就業者数は、2002年には1,202万人であったが、2002年から2012年までは減



Fig. 1.1: Number of employed people in all industries [1]

少傾向にあり、以降はほぼ横ばいで推移しており、2022年には1,044万人にまで減少している。つまり日本の製造業において、新規入職者数よりも退職者の数が多いことを意味する。全産業に占める製造業の就業者の割合が低下傾向で推移していることから、今後も新規入職者数よりも退職者の数が多くなることが想定される。

日本の製造業における若年就業者数は、Fig. 1.2に示すように2002年には384万人であったが、2002年から2012年までは減少傾向にあり、以降はほぼ横ばいで推移しており、2022年には255万人にまで減少している。一方、製造業における高齢就業者数は、Fig. 1.3に示すように2002年には58万人であったが、2002年から2018年までは増加傾向にあり、以降はほぼ横ばいで推移しており、2022年は90万人にまで増加している。

上述の理由から、今後も少子高齢化による製造業の労働力不足は続くことが想定される。

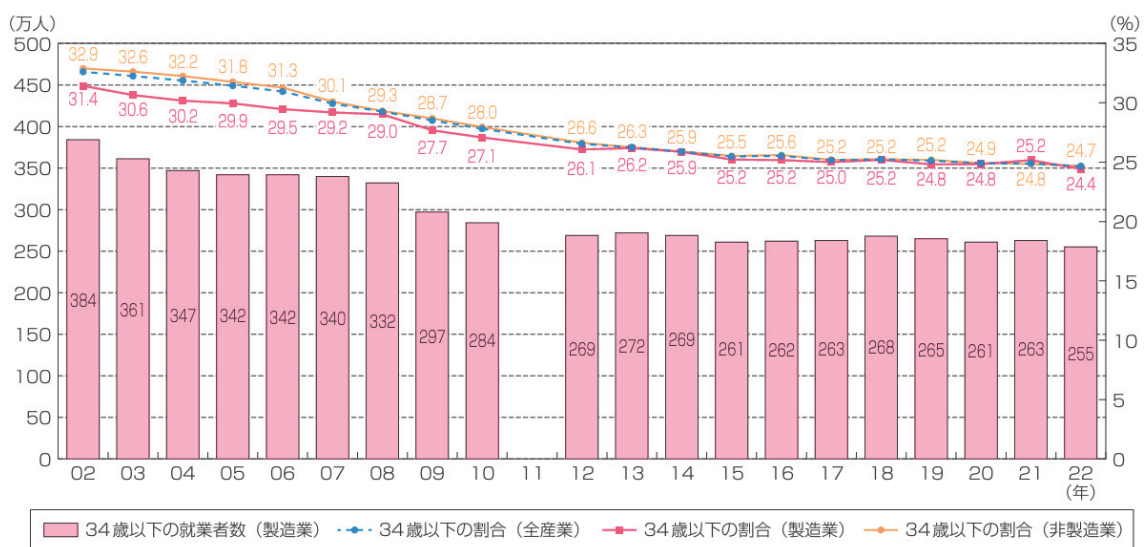


Fig. 1.2: Number of young workers in manufacturing industry [1]

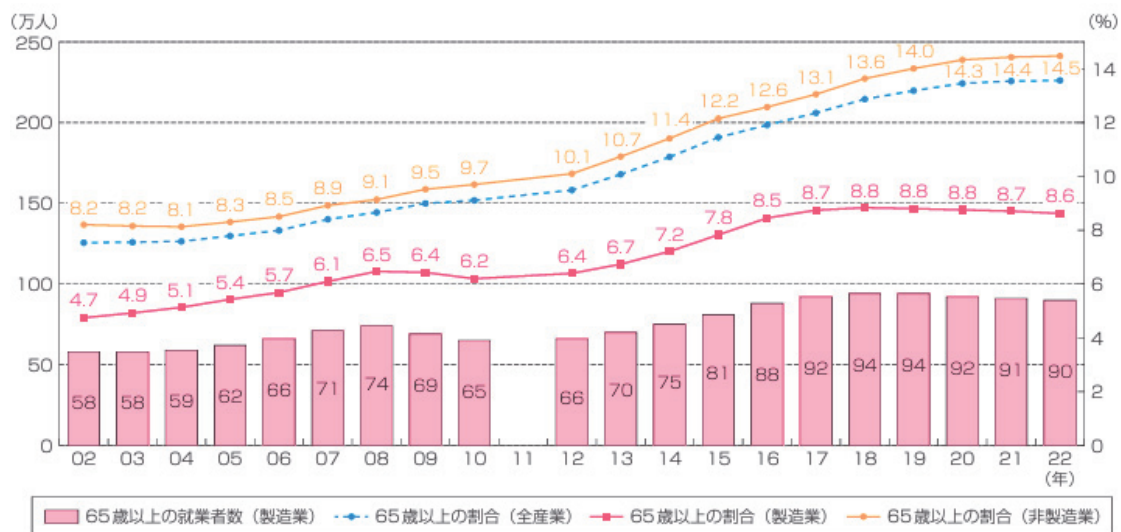


Fig. 1.3: Number of elderly workers in manufacturing industry [1]

この課題に対し経済産業省は、ものづくりのスマート化の方向性やレベル、スマート化を進める上での成功のポイントを整理したスマートファクトリーロードマップ[2]を発表した。その中で、技能の継承による人材不足への対応を目的とした取り組みでは、熟練者のノウハウをデータベースに蓄積し、それを分析し機械に学習させることで、熟練者を代替できると示されている。

(5) 「人材不足・育成への対応」に向けたロードマップ (1/1)

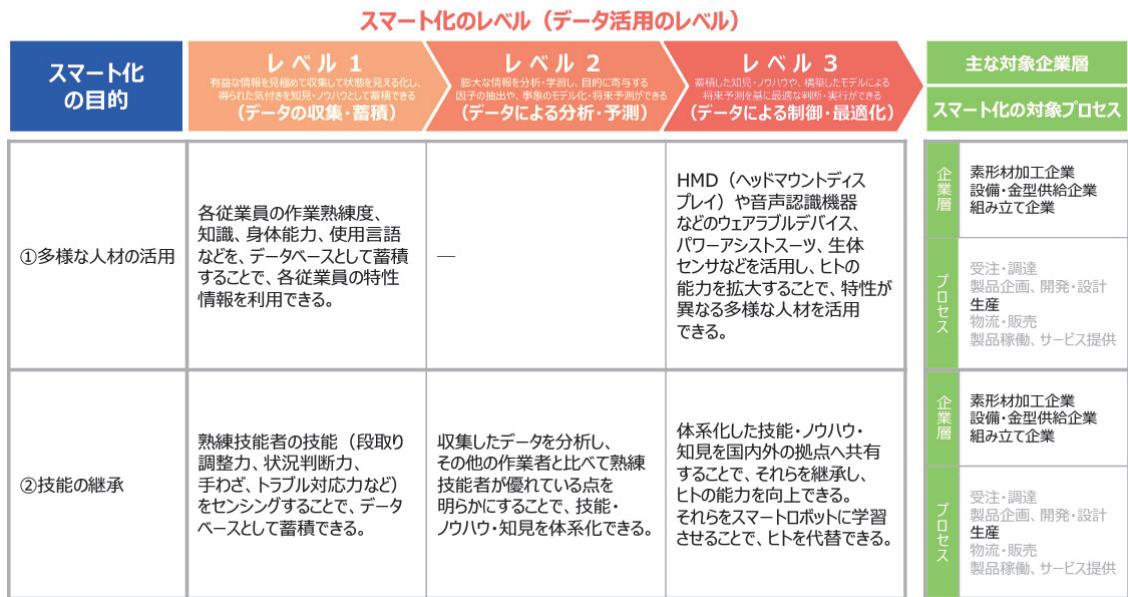


Fig. 1.4: Roadmap of the smart factory [2]

近年、塗装分野においても自動化が進められてきている。塗装の目的は、対象物の防錆性や対候性を保持することであり、塗装機で塗布された塗料により、対象物に塗膜を形成することで実現する。塗装工程には塗装作業、強制乾燥、品質検査などがある。例えば、少子高齢化による労働力不足の対策のため、Fig. 1.5 に示す塗装ロボットが導入されている。塗装ロボットは、産業用ロボットの先端に固定された塗装機を用いて塗料を吐出し、被塗装物へ塗装する。この塗装ロボットにより、これまで作業者が行なってきた塗装作業を自動化することができる。

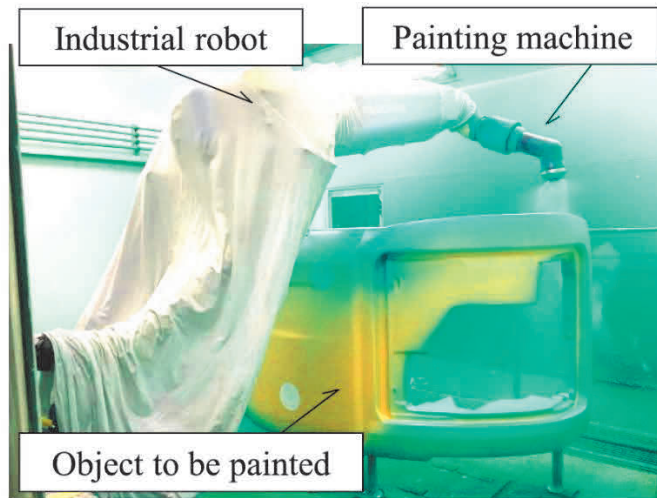


Fig. 1.5: Painting robot

一方、塗料には有機溶剤が含まれることが多く、センサを導入することが容易でないため、塗装分野では他の分野と比較して AI（人工知能）の導入が進んでいない状況である[3].

塗膜の光沢が求められる塗装では、回転霧化式静電塗装と呼ばれる塗装方式がとられる[4]. Fig. 1.6 にベルカップ回転霧化式静電塗装機の構成を示す. ベルカップ回転霧化式静電塗装機は、空気モータ、ベルカップ型の霧化頭、パターン調整用空気の噴射ノズル、および塗料ノズルによって構成されている. 塗料ノズルから霧化頭内に供給された塗料は、遠心力によって霧化頭の内壁を薄膜となって流動し、その先端近傍で微粒化される. その後塗粒は、被塗装物に向かって飛行し、塗着し塗膜を形成する.

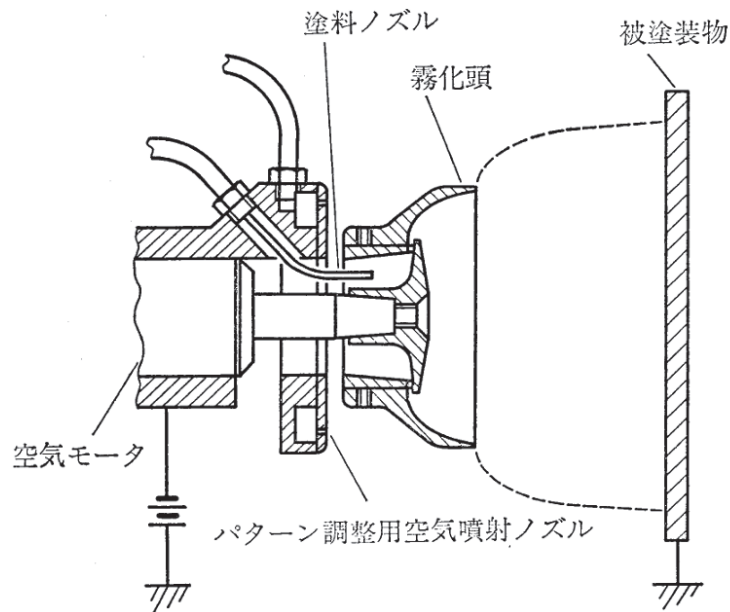


Fig. 1.6: Electrostatic paint machine

ベルカップ回転霧化式静電塗装方式では、塗粒径の増加に伴い未硬化塗膜の粘度が低下する。さらに、未硬化塗膜は被塗装面に到達した塗粒の合体によって形成される。塗膜の仕上がり品質（塗装品質）は未硬化塗膜の粘度、塗粒径に大きく影響する[5]。

ベルカップ回転式静電霧化塗装方式における塗装不具合は、タレや Fig. 1.7 に示すゆず肌 [6] など様々な種類がある。塗装不具合の要因としては、塗装ブース内の温度・湿度などの環境条件、ベルカップ回転数、吐出量などの動作条件があり、各要因が複雑に関連している。環境条件が調整できないような状況では、環境条件に応じて動作条件を適切に調整する必要がある。

上述の理由より、塗装ロボットの動作条件の調整には、未だに作業者が介在している。しかし、動作条件の調整は難しいことから、塗装不具合が発生する状況である。

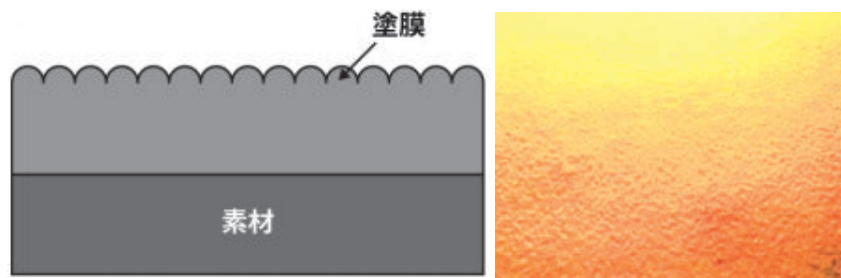


Fig. 1.7: Orange peel

塗装品質を維持するためには、環境条件および塗装品質に応じて動作条件を調整する必要がある。しかし、塗装品質が良好・不良のように定性的な値をとる場合、動作条件を適切に調整するための手法が確立されていない。この問題を解決するためには、

- ①環境条件に応じて動作条件を適応的に決定する機構
 - ②環境条件および動作条件から塗装品質を予測する機構
- を構築する必要があると考えられる。

1.2 本研究の目的と各章の構成

本研究では、塗装ロボットの動作条件を自動的に調整することを目的とし、Fig. 1.8 に示す①と②を備えたデータ駆動型スマート塗装システムを構築する。

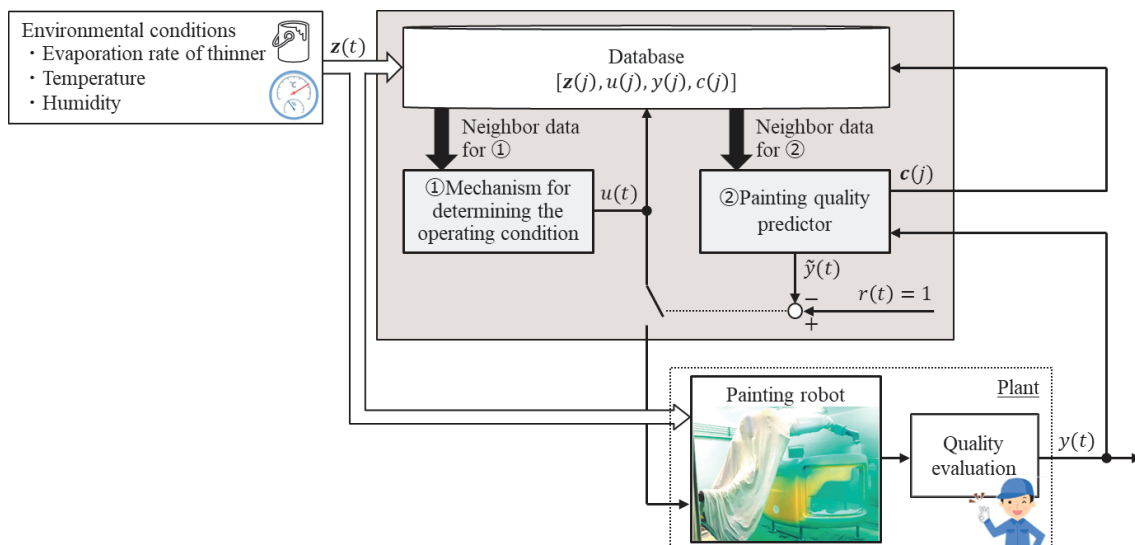


Fig. 1.8: Data-Driven Smart Painting System

本システムは，データベース，①動作条件決定機構(Mechanism for determining the operating condition)および②塗装品質予測器(Painting quality predictor)から構成される．データベースには，予め過去の塗装データを格納しておく．

①では，塗装直前の環境条件に基づき適切な動作条件を決定する．②では，環境条件および動作条件から塗装品質を予測する．この予測値は良好(+1)もしくは不良(-1)で表され，予測値が不良(-1)を示す場合には①の動作条件を修正する．この作業を繰り返し，予測値が良好(+1)となる際の動作条件を，設備の入力として与える．最後に，設備から得られた塗装品質の実測値を基に，②を介してデータベースを更新することで，②の正解率の向上につながり，①により適切な動作条件が決定されることになる．なお，この予測器はソフトセンサ[7][8]の役割があり，環境条件および動作条件から塗装品質を予測することができる．

ソフトセンサは，周辺の運転データから計測の困難な物理量を予測するものであり，近年ではJust-in-Time (JIT) 法[9]を用いた方法[10][11]が注目されている．JIT法はデータベースを有しており，現時刻のシステムの状態(クエリ)に対して，そのクエリと類似するデ

ータをデータベースから選択し、それらを用いて局所的にモデルを構築し、値を出力する方法である。

JIT法を用いて局所モデルを構築し値を出力することは、①においても有効であると考えており、①にも JIT法を用いることとする。

本研究の構成は、以下に示す5章から構成される。

第1章では、塗装分野の自動化に関する取り組みについて述べる。さらに、本研究の主題であるデータ駆動型スマート塗装システムが備える機能を明確にする。

第2章では、不良データを活用した Just-in-Time 法に基づく動作条件の決定機構の設計法を提案する。本手法は以下の手順によって構成される。まず、データベースに予め良好データおよび不良データを格納する。次に、クエリ（要求点）に近い良好データおよび不良データを用い、サポートベクトルマシンに基づき良好・不良の分類境界を作成し、クエリにおいて有効な良好データを決定する。その良好データを用いて重み付き局所線形平均法により動作条件を決定する。これにより、変化する環境条件に対して、適応的に動作条件を決定することが可能となると考える。最後に、塗装工程を模擬した数値シミュレーションにより提案法の有効性を評価する。

第3章では、データベース駆動型アプローチに基づく塗装品質予測器の設計法について提案する。提案法では、環境条件および動作条件をクエリとし、クエリ付近の良好データおよび不良データを用い、重み付き局所線形平均法により予測値（連続値）を決定する。この予測値（連続値）をしきい値により2値化することで、良好・不良の予測値を得る。さらに正解率の向上のため、データベースの要素として修正量を導入し、これを学習を通して更新する。結果として修正量が前述のしきい値を適応的に変更していることについて言及する。最後に、数値シミュレーションにより提案法の有効性を評価する。

第4章では、上述の動作条件決定機構と塗装品質予測器を統合した、データ駆動型スマート塗装システムを構築する。具体的には、動作条件決定機構により決定された動作条件

を基に，塗装品質予測器により出来上がりの塗装品質を予測する．予測値が-1(不良)を示す場合には，近傍データを修正し，再度動作条件決定機構により動作条件を決定する．さらに，塗装工程を模擬した数値シミュレーションにより，提案法の有効性を検証する．最後に，実際に量産で使用されている塗装ロボットに本システムを適用し，その有用性を評価する．

第5章では，本研究を総括すると共に，今後の研究展望について述べる．

第 2 章 Just-in-Time 法に基づく動作条件決定機構の構築

2.1 緒言

塗装品質を維持するためには、環境条件および塗装品質に応じて動作条件を適切に調整する必要がある。しかし、塗装品質が良好・不良のような定性的な値をとる場合、動作条件を適切に調整するための手法が確立されていない。環境条件に応じて出力を計算する手法として、前述の JIT 法を用いる手法が考えられる。しかし製造業では、運転データの総数は少なく、クエリと一致するような情報ベクトルをもつデータベースのデータは存在しないと考えられる。また、製造業の設備の入出力関係は非線形となることが多く[12]、良好データを利用する JIT 法に基づく動作条件の決定では、製品品質が良好になるとは限らない。

本研究では、不良データを活用した JIT 法に基づく動作条件決定機構の設計法を提案する。提案法では、データベースに予め良好データだけでなく不良データも格納する。そして、決定される動作条件がデータベースの不良データから遠ざかるよう、データベース内の良好データおよび不良データを用いて、動作条件を決定する。まず、クエリ付近の良好データおよび不良データを用い、サポートベクトルマシン[13][14] (Support Vector Machine. 以下 SVM) により良好および不良の分類境界を作成し、クエリに対し有効な良好データを決定する。さらに、その良好データを用い重み付き局所線形平均法 (Linearly Weighted Average. 以下 LWA) により動作条件を決定する。最後に、塗装工程を模擬した数値シミュレーションにより提案法の有効性を検証する。

2.2 Just-in-Time 法に基づく動作条件決定機構

Fig. 2.1 に本研究で提案する JIT 法に基づく動作条件決定機構の概要図を示す。データベースに格納されたデータのうち、 $\mathbf{z}(j) = [z_1(j), \dots, z_n(j)]$ は、塗装ブース内の温度・湿度などの環境条件、 n はその次数、 $u(j)$ は動作条件（ベルカップ回転数）、 $y(j)$ は塗装品質であり+1（良好）または-1（不良）の値を取ることにする。ここで、 $y(j)$ が1または-1となるデータの集合をそれぞれ、+1または-1のクラスに属するとする。また、 $\mathbf{z}(t) = [z_1(t), \dots, z_n(t)]$ は、塗装ブース内の温度・湿度などの環境条件、 $u(t)$ は動作条件（ベルカップ回転数）であり、データベースを用いて計算する。 $y(t)$ は設備からの出力（塗装品質）であり、+1（良好）または-1（不良）の値を取ることにする。Fig. 2.1の動作条件決定機構では、 $y(t)$ が+1となるように $u(t)$ の値を決定する必要がある。

Fig. 2.2 に本研究における提案法の概略図を示す。なお、多次元空間を示すことができないため、Fig. 2.2では、 $z_1 - u$ 平面上で動作条件 $u(t)$ が決定される様子を示す。(a)クエリが入力されたとき、(b)特定の指標を基にデータベースから近傍データを抽出する。次に、(c)近傍データの良好および不良の情報を基に分類境界を作成し、クエリにおいて品質が良好と分類される領域を抽出する。そして、(d)近傍データのうち、その範囲に収まる良好データを有効な良好データとし、LWAにより動作条件 $u(t)$ を計算する。JIT法では、出力は計算の対象となるデータの内側に存在するため、有効な良好データを用いた出力は、クエリにおいて品質が良好と分類される領域に収まる。これが本提案法の特徴である。

前述の分類境界は、SVMを用いて近傍データを線形分離し作成する。SVMでは、マージ

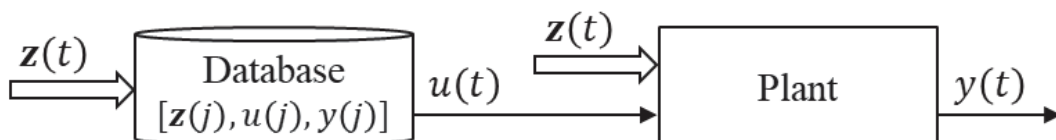


Fig. 2.1: Mechanism for determining the operating condition.

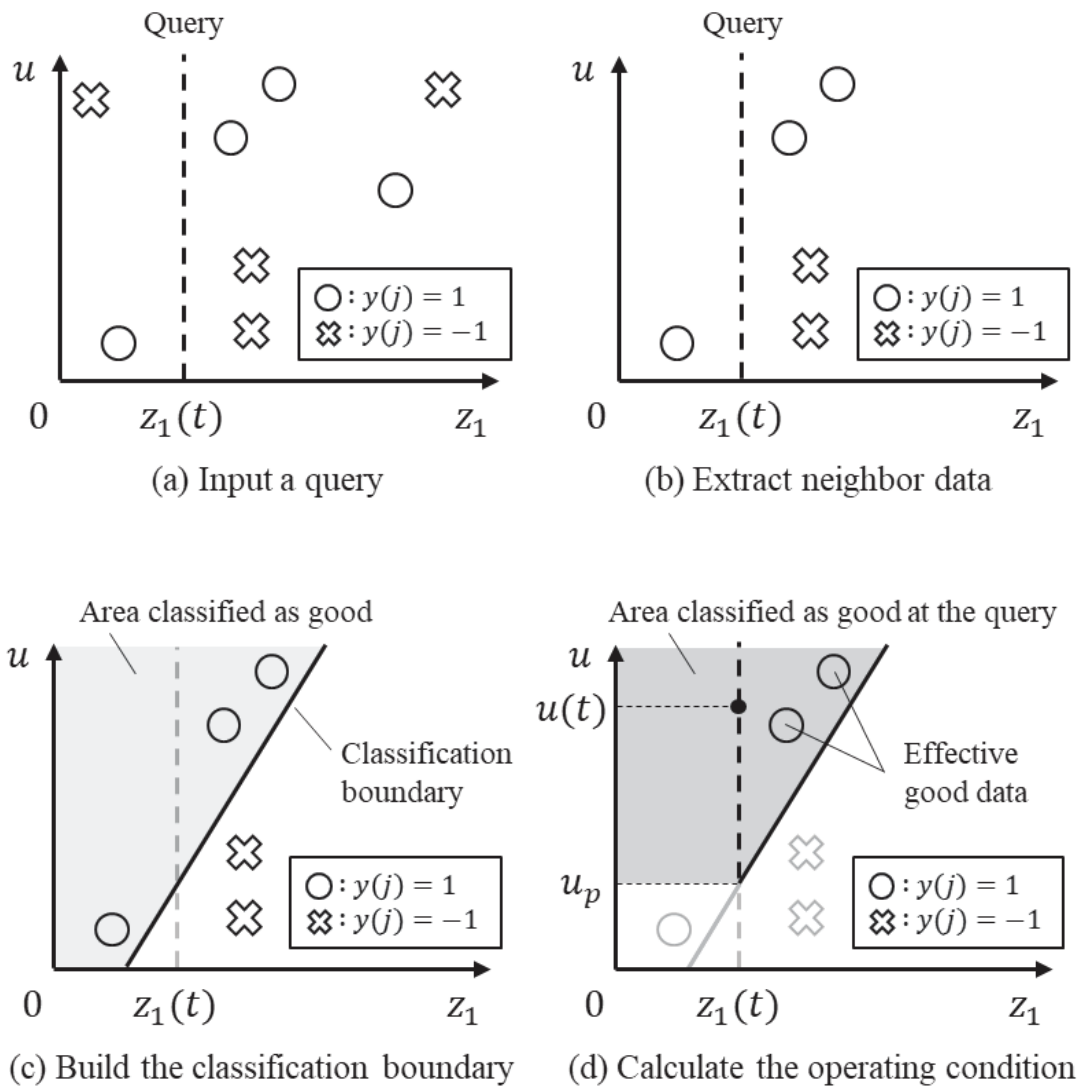


Fig. 2.2: Schematic of the proposed method.

ンという概念により，誤差関数が小さくなるように分類境界が作成される．マージンとは，分類境界と近傍データ間の最短距離を意味し，SVMによりマージンは最大になるよう計算される．本研究では，良好データおよび不良データが混在する状況を想定し，ソフトマージンへの緩和を行い，誤分類を許容することとする．

〈2.2.1〉 システムの記述 対象とする設備からの出力は離散時間非線形システムとし，次式のように記述される．

$$y(t) = f(\boldsymbol{\psi}_f(t)) \quad (2.1)$$

ここで、 $f(\cdot)$ は非線形関数、 $\boldsymbol{\psi}_f(t)$ は時刻 t における環境条件 $\mathbf{z}(t) = [z_1(t), \dots, z_n(t)]$ および動作条件 $u(t)$ とし、次式で定義する。

$$\boldsymbol{\psi}_f(t) := [\mathbf{z}(t), u(t)] \quad (2.2)$$

〈2.2.2〉 初期データベースの構築 JIT 法に基づき、情報ベクトル $\boldsymbol{\phi}(j)$ により構成される次式のデータセットにより、初期データベースを構築する。

$$\boldsymbol{\Phi}(j) := [\boldsymbol{\phi}(j), u(j), y(j)] \quad (2.3)$$

$$(ただし j = 1, 2, \dots, N_I)$$

$$\boldsymbol{\phi}(j) := [z_1(j), \dots, z_n(j)] \quad (2.4)$$

ここで、 $N_I, y(j)$ はそれぞれ、データベースのデータの数、塗装品質を示す目的変数である。

〈2.2.3〉 近傍データの抽出 現在のシステムの状態 $\mathbf{q}(t)$ をクエリと呼び、次式で定義する。

$$\mathbf{q}(t) := [z_1(t), \dots, z_n(t)] \quad (2.5)$$

近傍データの基準として、時刻 t におけるクエリ $\mathbf{q}(t)$ とデータベース内の j 番目のデータ $\boldsymbol{\phi}(j)$ との類似度[15][16][17] $S(\mathbf{q}(t), \boldsymbol{\phi}(j))$ を計算する。

$$S(\mathbf{q}(t), \boldsymbol{\phi}(j)) = \prod_{l=1}^n \frac{1}{\sqrt{2\pi}h_l} \exp\left\{-\frac{(q_l(t) - \phi_l(j))^2}{2h_l^2}\right\} \quad (2.6)$$

ここで、 $q_l(t)$ は時刻 t におけるクエリの第 l 番目の要素を表す。 $\phi_l(j)$ はデータベース内の第 j 番目の情報ベクトル $\boldsymbol{\phi}(j)$ の第 l 番目の要素を表す。 h_l は情報ベクトルに対する各クラスのバンド幅であり、次式に示すプラグ・イン法 (Plug-In Method)[18]により計算される。

$$h_l = \frac{1.06\sigma_l}{N^{\frac{1}{5}}} \quad (2.7)$$

ここで、各クラスの標準偏差 σ_l は次式のように定義される。

$$\sigma_l = \sqrt{\frac{1}{N} \sum_{m=1}^N (\phi_l(m) - \mu_l)^2} \quad (2.8)$$

ここで、 N は各クラスのデータ数、 μ_l は各クラスの ϕ_l の平均値とする。(2.6)式において、類似度が最大となるのは、クエリと同じ情報ベクトルがデータベースに存在する場合であり、類似度は次式のように表記できる。

$$S(\mathbf{q}(t), \mathbf{q}(t)) = \prod_{l=1}^n \frac{1}{\sqrt{2\pi h_l}} \quad (2.9)$$

したがって、本研究では、下記の類似度に基づく条件式を満たすデータ $\phi(j)$ を近傍データとして選択する。

$$\begin{aligned} S(\mathbf{q}(t), \phi(j)) &\geq T_{th} \cdot S(\mathbf{q}(t), \mathbf{q}(t)) \\ &= T_{th} \cdot \prod_{l=1}^n \frac{1}{\sqrt{2\pi h_l}} \end{aligned} \quad (2.10)$$

ここで、 T_{th} は類似度のしきい値であり、その範囲は $0 \leq T_{th} \leq 1$ である。ここで、各クラスの近傍データの数は、(2.2)式の $\psi_f(t)$ の要素数である $(n+1)$ 件以上選択されることとする。各クラスの近傍データの数が $(n+1)$ 件未満の場合は、そのクラスの中で類似度の高い上位 $(n+1)$ 件のデータを、そのクラスの近傍データとする。近傍データとして選択されたデータの数を $k(t)$ とする。

〈2.2.4〉 分類境界の作成 多次元空間 $\boldsymbol{\psi} = [z_1, \dots, z_n, u]$ において、SVM により作成される分類境界の式を次に示す。

$$\mathbf{0} = \mathbf{w}\boldsymbol{\psi}^T + b \quad (2.11)$$

ここで、 $\mathbf{w} = [w_1, \dots, w_{n+1}]$ および b は分類境界のハイパーパラメータであり、この値は近傍データにより誤差関数が小さくなるよう計算される。この分類境界により良好と分類される領域では、次式を満たす。

$$0 \leq \mathbf{w}\boldsymbol{\psi}^T + b \quad (2.12)$$

SVM において近傍データを分類する場合、以下のサポートベクトルマシンの最適化問題を解く必要がある。

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in [k(t)]} \xi_i$$

$$\text{s.t. } -(y_i(\mathbf{w}\boldsymbol{\psi}_i^T + b) - 1 + \xi_i) \leq 0, \quad i \in [k(t)] \quad (2.13)$$

$$-\xi_i \leq 0, \quad i \in [k(t)]$$

C は正則化係数、 ξ_i はスラック関数であり、第 1 項はマージンの逆数、第 2 項は誤分類によるペナルティの総和を意味し、 C は各項の重みを調整する役割がある。(2.13)式は制約条件を含んでおり、最急降下法のような数値解析法を適用することはできない。

この制約付き最適化問題に対し、本研究ではラグランジュの未定乗数法を適用する。ラグランジュの未定乗数法は制約付き最適化問題を解くための手法であり、元の関数に対し、制約条件に係数を掛けた項を新たに追加し、その関数の最適化問題を解く形となる。

(2.13)式の関数にラグランジュの未定乗数法を適用すると、次式のようになる。

$$L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in [k(t)]} \xi_i - \sum_{i \in [k(t)]} \alpha_i (y_i(\mathbf{w}\boldsymbol{\psi}_i^T + b) - 1 + \xi_i) - \sum_{i \in [k(t)]} \mu_i \xi_i \quad (2.14)$$

ここで、 $L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu})$ はラグランジュ関数、 $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{k(t)})^T$ 、 $\boldsymbol{\mu} = (\mu_1, \dots, \mu_{k(t)})^T$ は双対変数と呼ばれる。(2.14)式を双対変数について最大化すると、次式が得られる。

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \left(-\frac{1}{2} \sum_{i, j \in [k(t)]} \alpha_i \alpha_j y_i y_j \boldsymbol{\psi}_i \boldsymbol{\psi}_j^T + \sum_{i \in [k(t)]} \alpha_i \right) \quad (2.15)$$

$$\text{s.t. } \sum_{i \in [k(t)]} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i \in [k(t)]$$

$L(\boldsymbol{\alpha})$ が最大となる $\boldsymbol{\alpha}$ を求めるため、次式に示す最急降下法により α_i の更新を行う。

$$\begin{aligned}
\alpha_i &\leftarrow \alpha_i + \eta \frac{\partial L(\boldsymbol{\alpha})}{\partial \alpha_i} \\
&= \alpha_i + \eta \frac{\partial}{\partial \alpha_i} \left(-\frac{1}{2} \sum_{i,j \in [k(t)]} \alpha_i \alpha_j y_i y_j \boldsymbol{\psi}_i \boldsymbol{\psi}_j^T + \sum_{i \in [k(t)]} \alpha_i \right) \\
&= \alpha_i + \eta \left(1 - y_i \boldsymbol{\psi}_i \sum_{j \in [k(t)]} \alpha_j y_j \boldsymbol{\psi}_j^T \right)
\end{aligned} \tag{2.16}$$

この更新は、(2.15)式が特定の収束条件を満たすまで行う。ここで、 η は学習係数である。(2.15)式が収束したときの α_i を $\tilde{\alpha}_i$ とする。 $0 < \tilde{\alpha}_i < C$ を満たす $\tilde{\alpha}_i$ はサポートベクターと呼ばれる。サポートベクターの数を $S(t)$ とすると、分類境界のハイパーパラメータは次式で表される。

$$\mathbf{w} = \sum_{i \in [S(t)]} \tilde{\alpha}_i y_i \boldsymbol{\psi}_i \tag{2.17}$$

$$b = \frac{1}{S(t)} (y_i - \mathbf{w} \boldsymbol{\psi}_i^T) \tag{2.18}$$

〈2.2.5〉 有効な良好データの決定 クエリ $\mathbf{q}(t)$ と分類境界 $0 = \mathbf{w} \boldsymbol{\psi}^T + b$ との交点における動作条件 u を u_p とすると、次式が成立する。

$$0 = \mathbf{w} [\mathbf{q}(t), u_p]^T + b \tag{2.19}$$

したがって、以下の式が成り立つ。

$$u_p = -\frac{1}{w_{n+1}} ([w_1, \dots, w_n] \mathbf{q}(t)^T + b) \tag{2.20}$$

クエリにおいて品質が良好と分類される領域は、分類境界のハイパーパラメータ w_{n+1} および(2.12)式により次式で表すこととする。

$$\begin{cases} 0 \leq \mathbf{w} \boldsymbol{\psi}^T + b \text{ かつ } u \geq u_p & (w_{n+1} \geq 0) \\ 0 \leq \mathbf{w} \boldsymbol{\psi}^T + b \text{ かつ } u < u_p & (w_{n+1} < 0) \end{cases} \tag{2.21}$$

近傍データのうち、(2.21)式を満たす良好データを有効な良好データと定義し、そのデータの数を $g(t)$ とする。

〈2.2.6〉 動作条件の計算 データベースからの出力 $u(t)$ は、前述の有効な良好データ

を用い、LWAにより計算する.

$$u(t) = \sum_{i=1}^{g(t)} v_i u(i), \sum_{i=1}^{g(t)} v_i = 1 \quad (2.22)$$

ここで、 v_i は有効な良好データの第*i*番目の情報ベクトル $\phi(i)$ に対する重みであり、次式で表される.

$$v_i = \frac{S(\mathbf{q}(t), \phi(i))}{\sum_{i=1}^{g(t)} S(\mathbf{q}(t), \phi(i))} \quad (2.23)$$

以下に、これまでに述べた手順をアルゴリズムとして示す.

【不良データを活用した JIT 法に基づく動作条件決定機構のアルゴリズム】

1. 初期データベースの構築
2. 検証データからクエリ $\mathbf{q}(t)$ を抽出し、(2.10)式を満たす近傍データ $\phi(j), (j = 1, 2, \dots, k(t))$ を、データベースより抽出
3. 近傍データの情報を基に、(2.15)式が収束するまで(2.16)式により α_i を更新し、収束したときの値 $\tilde{\alpha}_i$ を用い、(2.17)式および(2.18)式により分類境界を作成
4. 近傍データのうち、(2.21)式を満たす良好データを有効な良好データ $\phi(i), (i = 1, 2, \dots, g(t))$ として定義
5. (2.22)式よりデータベースからの出力 $u(t)$ を計算

次章では、数値シミュレーションを通して本提案法の有効性を検証する.

2.3 数値シミュレーション

〈2.3.1〉 対象システム 数値シミュレーションでは、塗装工程を模擬した 4 入力 1 出力系のシステムを取り扱う. (2.2)式の $\psi_f(t)$ は、塗装ブース内の温度 $z_1(t)$ [°C]、塗装ブース内の湿度 $z_2(t)$ [%]、シンナーの蒸発速度 $z_3(t)$ [$10^{-1}\%/min$]、ベルカップ回転数 $u(t)$ [krpm]を用い、次式で定義する.

$$\boldsymbol{\psi}_f(t) := [\mathbf{z}(t), u(t)] \quad (2.24)$$

ただし, $\mathbf{z}(t) = [z_1(t), z_2(t), z_3(t)]$ である. また(2.4)式の情報ベクトル $\boldsymbol{\phi}(j)$ および(2.5)式のクエリ $\mathbf{q}(t)$ は, それぞれ次式で定義する.

$$\boldsymbol{\phi}(j) := [z_1(j), z_2(j), z_3(j)] \quad (2.25)$$

$$\mathbf{q}(t) := [z_1(t), z_2(t), z_3(t)] \quad (2.26)$$

出力 $y(t)$ の値は, 塗装品質が良好な状態を 1, 塗装品質が不良な状態を -1 とする.

実際の運転データの数はいくつか少なく, さらに良好データに比べて不良データの数が少ないため, 何らかの仮想設備を真の設備とみなし, 提案法の有効性を検証する必要がある. そのため本研究では, 次式で表される SVM による分類モデルを仮想設備とする.

$$f(\boldsymbol{\psi}_f(t)) = \sum_{i \in [S(t)]} \alpha_i y_i K(\boldsymbol{\psi}_i, \boldsymbol{\psi}_f(t)) + b$$

$$K(\boldsymbol{\psi}_i, \boldsymbol{\psi}_f(t)) = \exp(-\gamma \|\boldsymbol{\psi}_i - \boldsymbol{\psi}_f(t)\|^2)$$

$$y(t) = \begin{cases} -1, & f(\boldsymbol{\psi}_f(t)) \geq 0 \\ 1, & f(\boldsymbol{\psi}_f(t)) < 0 \end{cases} \quad (2.27)$$

$K(\boldsymbol{\psi}_i, \boldsymbol{\psi}_f(t))$ はカーネル関数であり, 本研究では RBF カーネルを用いる. 各変数の値は, $S(t) = 90$, $b = -0.99343$, $\gamma = 0.1$ である. 各 i における $\alpha_i y_i$, $\boldsymbol{\psi}_i = [z_{1i}, z_{2i}, z_{3i}, u_i]$ の値は Appendix 1 に示す. また, この分類モデルによる分類領域を Fig.2.3 に示す. なお Fig. 2.3 は可視化のため, z_2, z_3 を特定の値に固定したときの, $z_1 - u$ 平面上の分類領域を表す. Fig.2.3 より, この仮想設備の入出力関係は非線形であることが分かる.

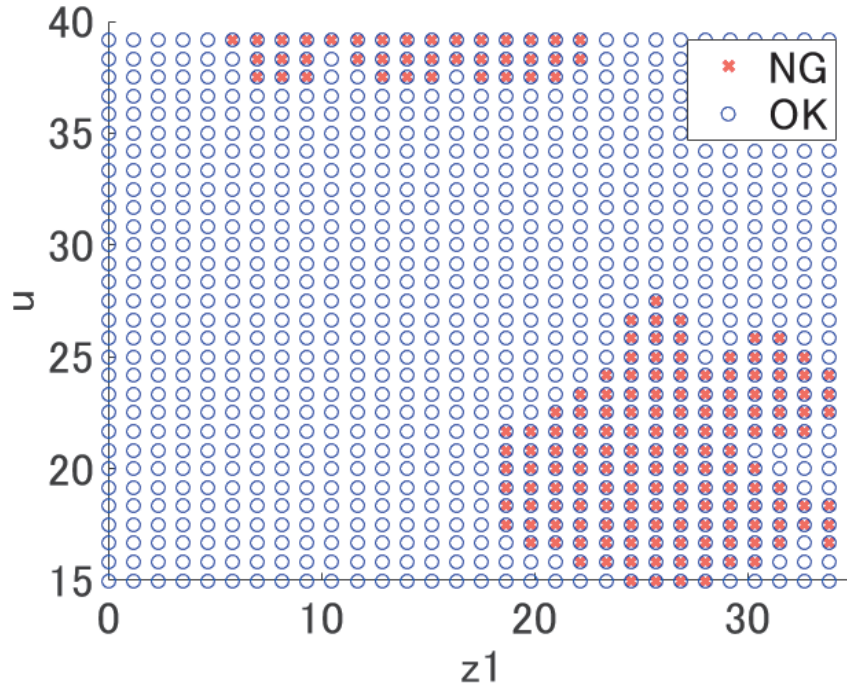


Fig. 2.3: Classification area of $y(t)$.

本数値シミュレーションでは，環境条件および動作条件の変化を考慮して， $z_1(t), z_2(t), z_3(t), u(t)$ はそれぞれ， $[0, 35], [30, 75], [15, 35], [15, 40]$ の範囲で等間隔に 30 件ずつ与える．そのため，仮想的な運転データの総数は $30^4 = 810000$ 件である．このデータセットから，初期データセットおよび検証用データセットをランダムに抽出する．そのため，各データセットに含まれるデータは，時間的にランダムになっていることに注意されたい．またデータを抽出する際，各クラスの件数割合は同じとする．

〈2.3.2〉 シミュレーション結果 まず，歩留まりを検証するために，検証データの出力 $y(t)$ の値を基に，次式で示される歩留まり F を導入する．

$$F := \frac{N_M}{N_V} \quad (2.28)$$

ここで， N_M は検証データの出力 $y(t)$ が 1（良好）となる検証データの数， N_V は検証データの数である．

提案法の有効性を検証するため，全ての良好データを使用した JIT 法（従来法），および

提案法を用いて数値シミュレーションを行う。

本研究では、(2.15)式のラグランジュ関数 $L(\boldsymbol{\alpha})$ が 3 回以上続けて減少する、もしくは繰り返し計算の回数が 4000 回となるとき、 $L(\boldsymbol{\alpha})$ が収束したと判定し繰り返し計算が終了することとする。

シミュレーションの設定値を Table 2.1 に示す。提案法の汎用性を確認するため、データセットからの抽出の仕方が異なる初期データセットおよび検証用データセットを 5 種類用意する。数値シミュレーションによる歩留まりは、5 種類のシミュレーション結果の平均値とする。

Fig. 2.4 に、繰り返し計算による $L(\boldsymbol{\alpha})$ の推移を示す。Fig. 2.4 より、 $L(\boldsymbol{\alpha})$ は繰り返し計算の実施に伴い区分的に増加することが分かる。このデータセットの例では、117 回の繰り返し計算で $L(\boldsymbol{\alpha})$ が収束したものと判定された。

Table 2.2 に、各手法における歩留まりの平均値を示す。提案法では、従来法と比較して歩留まりが向上することが確認できる。

ここで、提案法により検証データの出力 $\mathbf{y}(t)$ が良好となった理由について考察する。Fig. 2.5 に示すように以下の値を定義する。

Table 2.1: Fixed parameters in simulation.

Number of initial data	$N_I = 400$
Number of training data	$N_T = 400$
Number of test data	$N_V = 200$
Threshold of similarity	$T_{th} = 0.216$
Regularization coefficient	$C = 1$
Initial value of $\boldsymbol{\alpha}$	$\boldsymbol{\alpha} = \mathbf{0}$
Learning rate	$\eta = 0.0002$

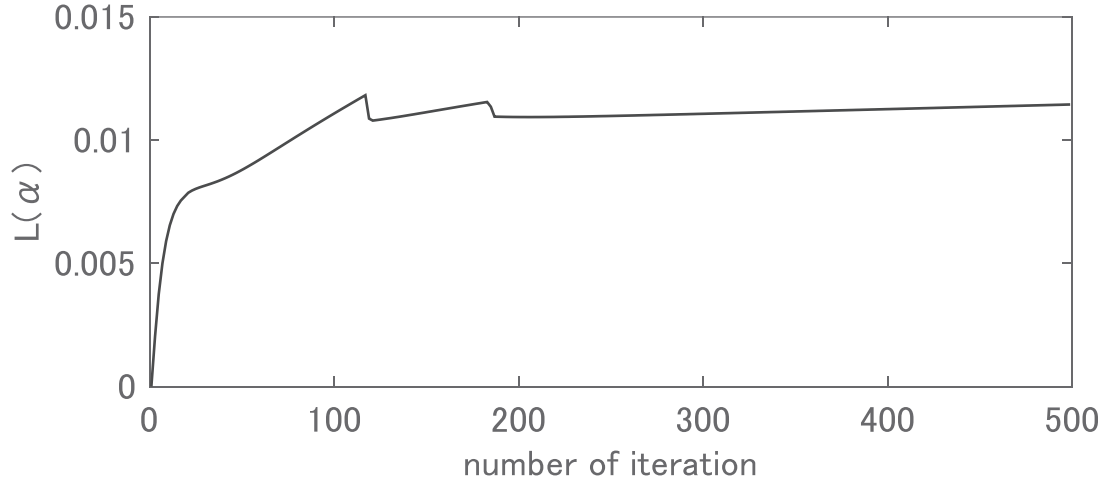


Fig. 2.4: Trajectories of $L(\alpha)$.

Table 2.2: The simulation result of the average of the yield.

Conventional method (JIT)	0.928
Proposed method	0.985

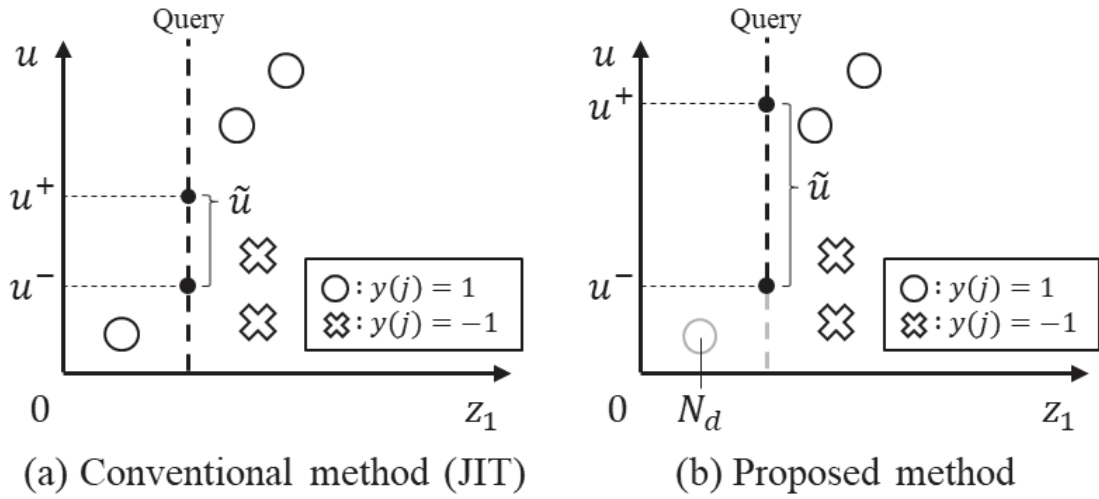


Fig. 2.5: The Manhattan distance \tilde{u} from u^+ to u^- .

u^- : 不良データを用いた LWA による出力. 不良データの中心

u^+ : 各手法による出力

\tilde{u} : u^+ から u^- までのマンハッタン距離. $\tilde{u} = |u^+ - u^-|$

N_d : 従来法(JIT 法)では選択され, 提案法では削除された良好データの数

提案法により $y(t)$ が改善された一例における各値について, Table 2.3 に示す. Table 2.3 より, u^- は従来法と提案法では同じであるものの, \tilde{u} は提案法の方が大きく, それだけ, 不良データの中心から遠ざかっていることが分かる. このことは, N_d だけ不良データの中心に近い良好データが削除されたためと考えられる.

次に Table 2.4 のように, 初期データの数, 初期データの良好数と不良数との比率 (良好

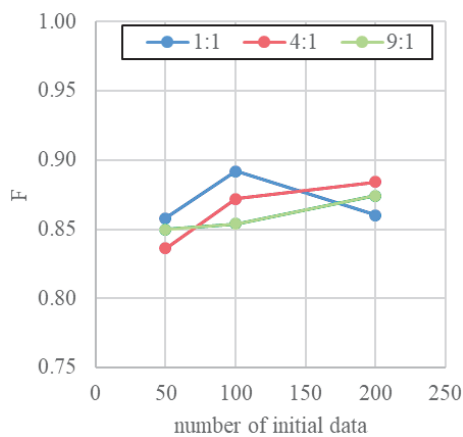
Table 2.3: The average of u^- , u^+ , \tilde{u} and N_d .

	u^-	u^+	\tilde{u}	N_d
Conventional method (JIT)	21.31	21.59	0.28	0
Proposed method	21.31	25.86	4.55	3.09

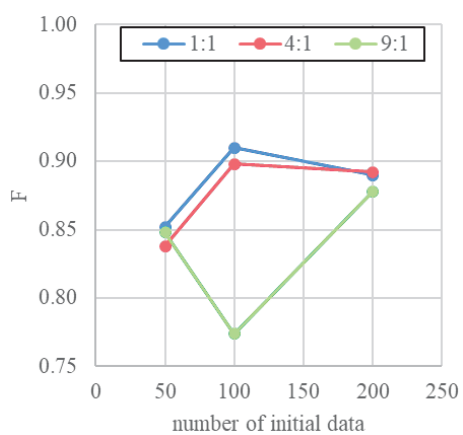
Table 2.4: Fixed parameters in simulation.

Number of initial data	$N_I = [50, 100, 200]$
Number of test data	$N_V = 200$
Class ratio of initial data	[9: 1, 4: 1, 1: 1]
Threshold of similarity	$T_{th} = 0.216$
Regularization coefficient	$C = 1$
Initial value of α	$\alpha = \mathbf{0}$
Learning rate	$\eta = 0.0002$

数：不良数. 以下，良好数と不良数のクラス比) を変化させた場合の，JIT法および提案法による歩留まりの変化を Fig. 2.6 に示す. また，JIT法および提案法による歩留まりの平均値を Table. 2.5 に示す. Fig. 2.6 および Table. 2.5 より，初期データの数 が 100，良好数と不良数のクラス比が 9:1 のとき，JIT法による歩留まりよりも提案法による歩留まりの方が低いことがわかる. これは，前述の「分類境界の構築」をおこなう際，少数派のクラスが重視されず正しく分類境界が構築されなかったことにより，有効な良好データが正しく選択



(a) Conventional method (JIT)



(b) Proposed method

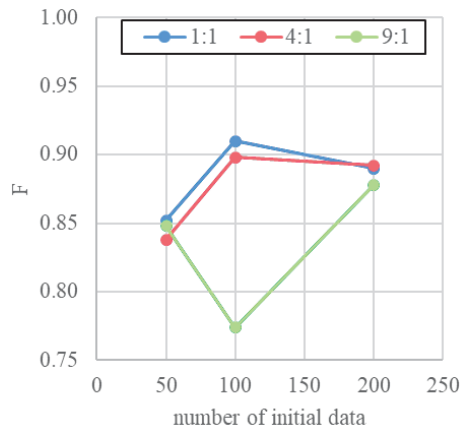
Fig. 2.6: The simulation result of the yield.

Table 2.5: The simulation result of the average of the yield.

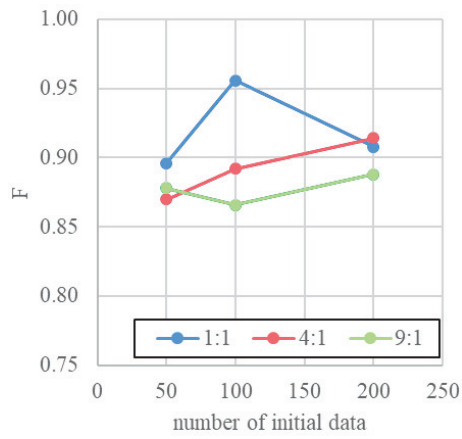
Conventional method (JIT)	0.864
Proposed method	0.864

されなかったためと考えられる.

この問題に対し, 数値解析法として最急降下法の代わりに DCDM 法[19]を用いる. DCDM 法は, SVM の線形分離に特化したアルゴリズムであり, SVM により局所的に線形分離をおこなう本研究に対しても有効であると考え. Fig. 2.7 に, 数値解析法として最急降下法および DCDM 法を使用した場合の歩留まりの変化を示す. ただし, シミュレーションの設定値は Table 2.4 と同じとする. Fig. 2.7 より, 提案法 (最急降下法) による歩留まりの低い箇所が, 提案法 (DCDM 法) による歩留まりでは改善されることがわかる. また, 提案法 (最急降下法) および提案法 (DCDM 法) による歩留まりの平均を Table 2.6 に示す. Table 2.6 より, 提案法 (最急降下法) による歩留まりよりも提案法 (DCDM 法) による歩留まりの方が高いことが分かる.



(a) Proposed method (Gradient descent method)



(b) Proposed method (DCDM method)

Fig. 2.7: The simulation result of the yield.

Table 2.6: The simulation result of the average of the yield.

Proposed method (Gradient descent method)	0.864
Proposed method (DCDM method)	0.896

本数値シミュレーションに用いたコンピュータのスペックを Table 2.7 に示す. Table 2.8 に, 最急降下法および DCDM 法による SVM の最適化にかかる繰り返し計算数および 1 度の繰り返し計算にかかる計算負荷の比較表を示す. Table 2.8 より, 最急降下法を用いるよりも, DCDM 法を用いる方が最適化にかかる繰り返し計算数が少なく, かつ計算時間も短いことが分かる.

Table 2.7. Computer spec.

Processor	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
System type	64-bit operating system
Memory	16.0GB
OS	Windows 11 Home

Table 2.8. Number of iterations and calculation time.

	Number of iterations [times]		Calculation time [s/times]	
	Average	Standard deviation	Average	Standard deviation
Gradient descent method	852.4	562.4	0.68	0.49
DCDM method	726.3	433.7	0.07	0.05

2.4 結言

本研究では、歩留まりの向上のため、不良データを活用した JIT 法に基づく動作条件決定機構の設計法を提案し、数値シミュレーションによりその有効性を検証した。具体的には、クエリ付近の良好データおよび不良データを用い、SVM により良好・不良の分類境界を作成し、クエリにおいて品質が良好と分類される動作条件の範囲を求め、有効な良好データを決定する。その後、その良好データを用いて LWA により動作条件を決定する。数値例を通して、提案法の歩留まりの向上が確認できた。とくに提案法では、新たな動作条件が不良データの中心から遠ざかり、これにより塗装品質が良好となる可能性を示した。

また、SVM により分類境界を作成する際の数値解析手法として、線形 SVM に特化した DCDM 法を用いることで、繰り返し計算の回数や計算時間を抑えることができ、さらに歩留まりの向上が確認できた。

第 3 章 データベース駆動型塗装品質

予測器の構築

3.1 緒言

塗装品質は作業者により視覚的に検査しており、ロボットによる塗装作業から塗装品質の検査までに時間遅延がある。この時間遅延による影響を低減するため、塗装直前に塗装品質の要因から塗装品質を予測できれば、この課題が解決できると考えられる。

本章では、データベース駆動型アプローチに基づく塗装品質予測器の設計法を提案し、数値例によりその有効性を検証する。提案法では、環境条件および動作条件をクエリとし、クエリ付近の良好データおよび不良データを用い、重み付き局所線形平均法により予測値（連続値）を決定する。この予測値（連続値）をしきい値により 2 値化することで、良好・不良の予測値を得る。さらに正解率の向上のため、データベースの要素として修正量を導入し、これを学習を通して更新する。結果として修正量が前述のしきい値を適応的に変更していることについて言及する。

3.2 データベース駆動型塗装品質予測器

Fig. 3.1 に本研究で提案するデータベース駆動型塗装品質予測器の概要を示す。 $\mathbf{z}(t) = [z_1(t), \dots, z_n(t)]$ は、塗装ブース内の温度・湿度などの環境条件、 n はその次数、 $u(t)$ は動作条件（ベルカップ回転数）である。 $y(t)$ は設備からの出力（塗装品質）であり、1（良好）または-1（不良）の値を取るとする。 $\hat{y}(t)$ はデータベースを用いて計算する、塗装品質の予測値（連続値）である。また $\hat{y}(t)$ をしきい値処理により、1または-1に2値化

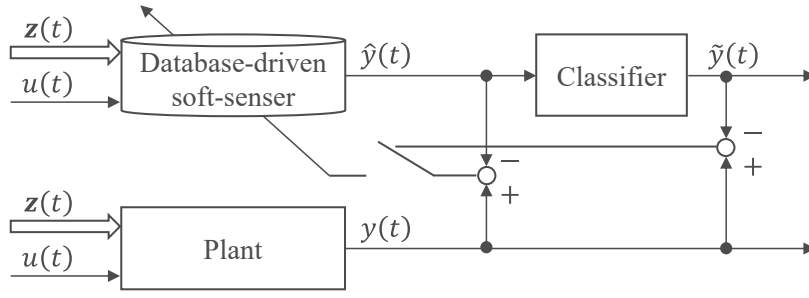


Fig. 3.1: Schematic of the proposed database-driven quality predictor.

した値が $\tilde{y}(t)$ である。ここで、 $y(t)$ もしくは $\tilde{y}(t)$ が 1 または -1 となるデータの集合をそれぞれ、1 または -1 のクラスに属するとする。Fig. 3.1 の予測器では、 $y(t)$ と $\tilde{y}(t)$ とが異なるときに、 $y(t)$ と $\hat{y}(t)$ との誤差が小さくなるよう、データベース内の修正量を更新する。

〈3.2.1〉 システムの記述 対象とする設備からの出力は、第 2 章と同様に離散時間非線形システムとし、次式のように記述される。

$$y(t) = f(\boldsymbol{\psi}_f(t)) \quad (3.1)$$

ここで、 $f(\cdot)$ は非線形関数、 $\boldsymbol{\psi}_f(t)$ は時刻 t における環境条件 $\mathbf{z}(t) = [z_1(t), \dots, z_n(t)]$ および動作条件 $u(t)$ とし、次式で定義する。

$$\boldsymbol{\psi}_f(t) := [\mathbf{z}(t), u(t)] \quad (3.2)$$

〈3.2.2〉 初期データベースの構築 データベース駆動型アプローチに基づき、情報ベクトルに加え、予測値（目的変量）に対する修正量 $c(j)$ [19][20]により構成される次式のデータセットにより、初期データベースを構築する。

$$\boldsymbol{\Phi}(j) := [\boldsymbol{\phi}'(j), y(j), c(j)] \quad (3.3)$$

$$(\text{ただし } j = 1, 2, \dots, N_I)$$

$$\boldsymbol{\phi}'(j) := [z_1(j), \dots, z_n(j), u(j)] \quad (3.4)$$

ここで、 $N_I, y(j)$ はそれぞれ、データベースのデータ数、塗装品質を示す目的変数である。なお、JIT 法では修正量 $c(j)$ は含まれない。本手法では、この $c(j)$ の学習を通して、正解率の向上を図るとともに、初期データベースの $y(j)$ を書き換えることなく真値を保存す

ることができるという特徴を有している。ただし、 $c(j)$ の初期値は 0 とする。

〈3.2.3〉 近傍データの抽出 現在のシステムの状態 $\mathbf{q}'(t)$ をクエリと呼び、次式で定義する。

$$\mathbf{q}'(t) := [z_1(t), \dots, z_n(t), u(t)] \quad (3.5)$$

近傍データの基準として、時刻 t におけるクエリ $\mathbf{q}'(t)$ とデータベース内の j 番目のデータ $\boldsymbol{\phi}'(j)$ との類似度 $S(\mathbf{q}'(t), \boldsymbol{\phi}'(j))$ を計算する。

$$S(\mathbf{q}'(t), \boldsymbol{\phi}'(j)) = \prod_{l=1}^{n+1} \frac{1}{\sqrt{2\pi}h_l} \exp\left\{-\frac{(q'_l(t) - \phi'_l(j))^2}{2h_l^2}\right\} \quad (3.6)$$

ここで、 $q'_l(t)$ は時刻 t におけるクエリの第 l 番目の要素を表す。 $\phi'_l(j)$ はデータベース内の第 j 番目の情報ベクトル $\boldsymbol{\phi}'(j)$ の第 l 番目の要素を表す。 h_l は情報ベクトルに対する各クラスのパンド幅であり、次式に示すプラグ・イン法により計算される。

$$h_l = \frac{1.06\sigma_l}{N^{\frac{1}{5}}} \quad (3.7)$$

ここで、各クラスの標準偏差 σ_l は次式で定義される。

$$\sigma_l = \sqrt{\frac{1}{N} \sum_{m=1}^N (\phi'_l(m) - \mu_l)^2} \quad (3.8)$$

ここで、 N は各クラスのデータ数、 μ_l は各クラスの ϕ_l の平均値である。近傍データは、次の類似度に基づく条件式を満たすデータ $\boldsymbol{\phi}'(j)$ である。

$$S(\mathbf{q}'(t), \boldsymbol{\phi}'(j)) \geq T'_{th} \cdot \prod_{l=1}^{n+1} \frac{1}{\sqrt{2\pi}h_l} \quad (3.9)$$

ここで、 T'_{th} は類似度のしきい値であり、その範囲は $0 \leq T'_{th} \leq 1$ である。(3.9)式において、類似度が最大となるのは、クエリと同じ情報ベクトルがデータベースに存在する場合であり、類似度は次式のように表記できる。

$$S(\mathbf{q}'(t), \mathbf{q}'(t)) = \prod_{l=1}^{n+1} \frac{1}{\sqrt{2\pi h_l}} \quad (3.10)$$

また、近傍データの無いクラスが存在する場合、そのクラスの中で類似度が最大となるデータ1件を近傍データとする。抽出された近傍データの数を $k(t)$ とする。

〈3.2.4〉 予測値の計算 データベースからの出力 $\hat{y}(t)$ は、修正量 $c(j)$ を含めて LWA により計算する。

$$\hat{y}(t) = \sum_{i=1}^{k(t)} w_i (y(i) + c(i)), \quad \sum_{i=1}^{k(t)} w_i = 1 \quad (3.11)$$

ここで、 w_i はデータベースより選択された、第 i 番目の情報ベクトル $\boldsymbol{\phi}'(i)$ に対する重みであり、次式で与える。

$$w_i = \frac{S(\mathbf{q}'(t), \boldsymbol{\phi}'(i))}{\sum_{i=1}^{k(t)} S(\mathbf{q}'(t), \boldsymbol{\phi}'(i))} \quad (3.12)$$

また、 $\hat{y}(t)$ は連続値であるため、これを2値化する出力 $\tilde{y}(t)$ をしきい値 T_y を用いて次のように定義する。

$$\tilde{y}(t) := \begin{cases} 1 & (\hat{y}(t) \geq T_y) \\ -1 & (\hat{y}(t) < T_y) \end{cases} \quad (3.13)$$

ところで、(3.11)式は次のように表記できる。

$$\hat{y}(t) = \sum_{i=1}^{k(t)} w_i y(i) + \sum_{i=1}^{k(t)} w_i c(i) \quad (3.14)$$

(3.14)式に(3.13)式を代入すると、次式を得ることができる。

$$\sum_{i=1}^{k(t)} w_i y(i) \geq T_y - \sum_{i=1}^{k(t)} w_i c(i) \quad (3.15)$$

一方、前述の通り、JIT 法では $c(i)$ が存在しないので、JIT 法により出力を2値化する場合、(3.15)式に対応した次式を得ることができる。

$$\sum_{i=1}^{k(t)} w_i y(i) \geq T_y \quad (3.16)$$

(3.15)式と(3.16)式とを比較すると、(3.15)式の右辺は 2 値化の際の新たなしきい値とみなすことができる。これは、データベースの修正量 $c(i)$ の学習を通して、局所的に適切なしきい値が更新されると解釈できる。つまり、この予測器はしきい値を適応的に変更する仕組みを有しており、これが本手法の特徴となっている。

〈3.2.5〉 **修正量の更新** クエリの出力 $y(t)$ と出力 $\hat{y}(t)$ が異なる場合、近傍データに含まれる修正量 $c(i)$ を更新する。更新には、次式の最急降下法を用いる。

$$c(i) \leftarrow c(i) - \eta \frac{\partial J(t)}{\partial c(i)} \quad (3.17)$$

ここで、 η は修正量の学習係数である。また $J(t)$ は 2 乗誤差関数であり、次式のように定義される。

$$J(t) = \frac{1}{2} (y(t) - \hat{y}(t))^2 \quad (3.18)$$

(3.18)式より、

$$\begin{aligned} \frac{\partial J(t)}{\partial \hat{y}(t)} &= \frac{\partial J(t)}{\partial \hat{y}(t)} \left\{ \frac{1}{2} (y(t) - \hat{y}(t))^2 \right\} \\ &= \hat{y}(t) - y(t) \end{aligned} \quad (3.19)$$

(3.14)式より、

$$\begin{aligned} \frac{\partial \hat{y}(t)}{\partial c(i)} &= \frac{\partial}{\partial c(i)} \left(\sum_{i=1}^{k(t)} w_i y(i) + \sum_{i=1}^{k(t)} w_i c(i) \right) \\ &= w_i \end{aligned} \quad (3.20)$$

である。(3.17)式は微分連鎖則により、次式として書ける。

$$\begin{aligned} c(i) \leftarrow c(i) - \eta \frac{\partial J(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial c(i)} \\ = c(i) - \eta w_i (\hat{y}(t) - y(t)) \end{aligned} \quad (3.21)$$

一方、クエリの出力 $y(t)$ と出力 $\hat{y}(t)$ が等しい場合、近傍データの修正量 $c(i)$ は更新しない。また、このときの 2 乗誤差関数 $J(t)$ は 0 とする。全ての学習用データを用いて修正量

の更新を行った後、次式により 2 乗誤差関数の総和 E を計算する.

$$E = \sum_{t=1}^{N_T} J(t) \quad (3.22)$$

ここで、 N_T は学習用データの数である. (3.22)式が予め設定した終了条件を満たすまで、同じ学習用データを用いて修正量の更新を行う.

以下に、これまでに述べた手順をアルゴリズムとして示す.

【データベース駆動型塗装品質予測器のアルゴリズム】

1. 初期データベースの構築
 2. 教師データからクエリ $q'(t)$ を抽出し、(3.9)式の条件式を満たす近傍データ $\phi'(i), (i = 1, 2, \dots, k(t))$ を、データベースより抽出
 3. (3.11)式と(3.13)式から、それぞれ $\hat{y}(t)$ と $\tilde{y}(t)$ を計算
 4. クエリの出力 $y(t)$ と出力 $\tilde{y}(t)$ が異なる場合、(3.21)式により近傍データの修正量 $c(i)$ を更新し、(3.18)式の 2 乗誤差関数 $J(t)$ を計算
 5. 2 乗誤差関数の総和 E が、予め設定した終了条件を満たさない場合、2 へ戻る
- 次節では、数値シミュレーションを通して本提案法の有効性を検証する.

3.3 数値シミュレーション

〈3.3.1〉 対象システム 数値シミュレーションでは、塗装工程を模擬した 4 入力 1 出力系のシステムを取り扱う. $\psi_f(t)$ は、塗装ブース内の温度 $z_1(t)$ [°C], 塗装ブース内の湿度 $z_2(t)$ [%], シンナーの蒸発速度 $z_3(t)$ [$10^{-1}\%/min$], ベルカップ回転数 $u(t)$ [krpm]を用い、次式のように定義する.

$$\psi_f(t) := [z(t), u(t)] \quad (3.24)$$

ただし、 $z(t) = [z_1(t), z_2(t), z_3(t)]$ である. また情報ベクトル $\phi(j)$ およびクエリ $q(t)$ は、それぞれ次式のように定義する.

$$\boldsymbol{\phi}(j) := [z_1(j), z_2(j), z_3(j)] \quad (3.25)$$

$$\boldsymbol{q}(t) := [z_1(t), z_2(t), z_3(t)] \quad (3.26)$$

出力 $\boldsymbol{y}(t)$ の値は、塗装品質が良好な状態を 1，塗装品質が不良な状態を -1 とする。

実際の運転データの数少なく、さらに良好データに比べて不良データの数少ないため、何らかの仮想設備を真の設備とみなし、提案法の有効性を検証する必要がある。そのため本章においても、2章の(2.27)式と同じ分類モデルを仮想設備とする。

本数値例において、環境条件や動作条件の変化を考慮し、 $z_1(t), z_2(t), z_3(t), u(t)$ はそれぞれ、 $[0, 35], [30, 75], [15, 35], [15, 40]$ の範囲で等間隔に 30 件ずつ与える。そのため、仮想的な運転データの総数は $30^4 = 810000$ 件である。クラスが 1 となるデータの数 806068 件、クラスが -1 となるデータの数 3932 件である。このデータセットから、被学習用データセット、学習用データセットおよび検証用データセットをランダムに抽出する。そのため、各データセットに含まれるデータは、時間的にランダムになっていることに注意されたい。またデータを抽出する際、各クラスの件数割合は同じとする。

〈3.3.2〉 シミュレーション結果 まず、正解率を検証するために、検証データの出力 $\boldsymbol{y}(t)$ と出力 $\hat{\boldsymbol{y}}(t)$ の値を基に、次式で示される正解率 F を導入する。

$$F := \frac{N_M}{N_V} \quad (3.27)$$

ここで、 N_M は検証データの出力 $\boldsymbol{y}(t)$ と出力 $\hat{\boldsymbol{y}}(t)$ が一致する検証データの数、 N_V は検証データの数である。

提案法の有効性を検証するため、ニューラルネットワーク法、(3.16)式による 2 値化処理を施した JIT 法、および提案法を用いて数値シミュレーションを行う。

本研究では、(3.22)式で与えられる 2 乗誤差関数の総和 E が 0.1 未満となる、もしくは繰り返し学習の回数が 1000 回となるとき、 E が収束したと判定し繰り返し学習が終了することとする。シミュレーションの設定値を Table 3.1 に示す。提案法の汎用性を評価するため、抽出の仕方の異なるデータセットを 5 種類ほど用意する。数値シミュレーションによる正

解率は、5種類のシミュレーション結果の平均値とする。

Fig. 3.2 に、繰り返し学習による E の推移を示す。Fig. 3.2 より、 E は繰り返し学習に伴い減少し、最終的には収束していることが分かる。このデータセットの例では、55 回の繰り返し学習で E が 0 に収束したと判定された。

Fig. 3.3 に、学習用データセット、および(3.15)式のしきい値 $T_y - \sum_{i=1}^{k(t)} w_i c(i)$ の変化を示す。(a) JIT 法では、学習用データの真値が 1 の場合に、誤分類が発生していることが分かる。(b) 提案法における 2 回の繰り返し学習後は、その誤分類を削減するように、しきい値が小さくなる方向に修正されていることが分かる。ここでしきい値は、学習用データの真

Table 3.1: Fixed parameters in simulation.

Number of initial data	$N_I = 400$
Number of training data	$N_T = 400$
Number of test data	$N_V = 200$
Threshold of similarity	$T_{th} = 0.2^4$
Threshold of output	$T_y = 0$
Learning rate	$\eta = 0.1$

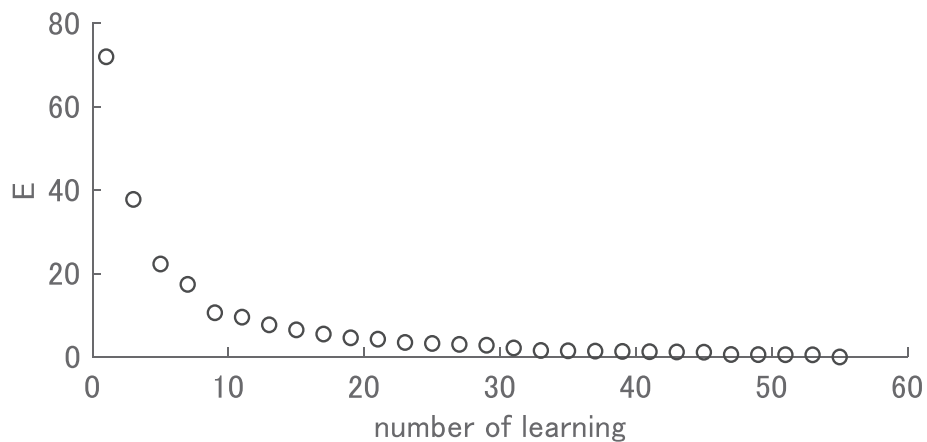
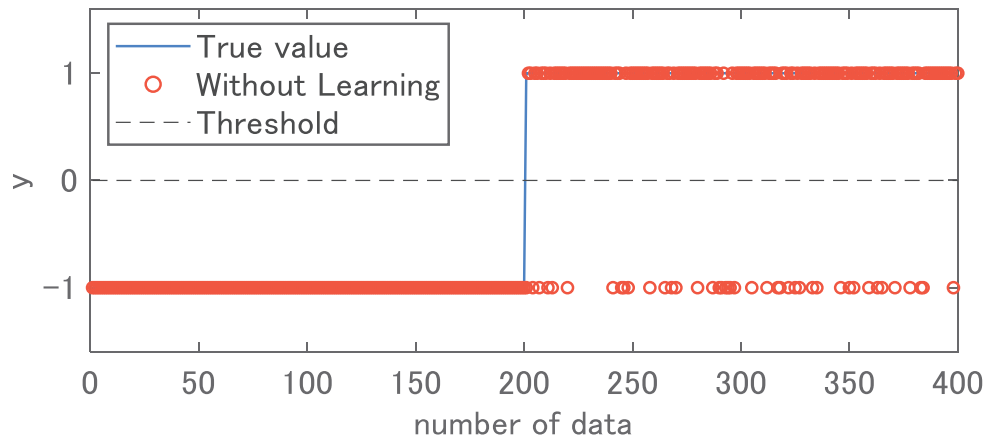
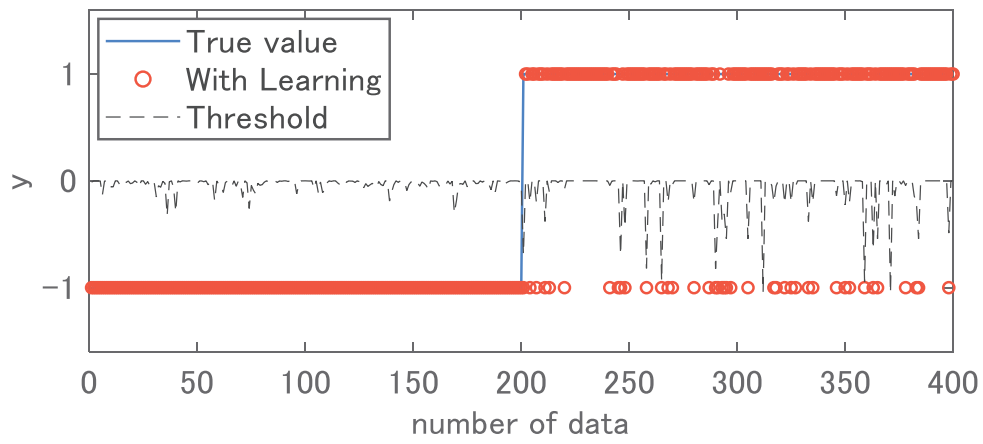


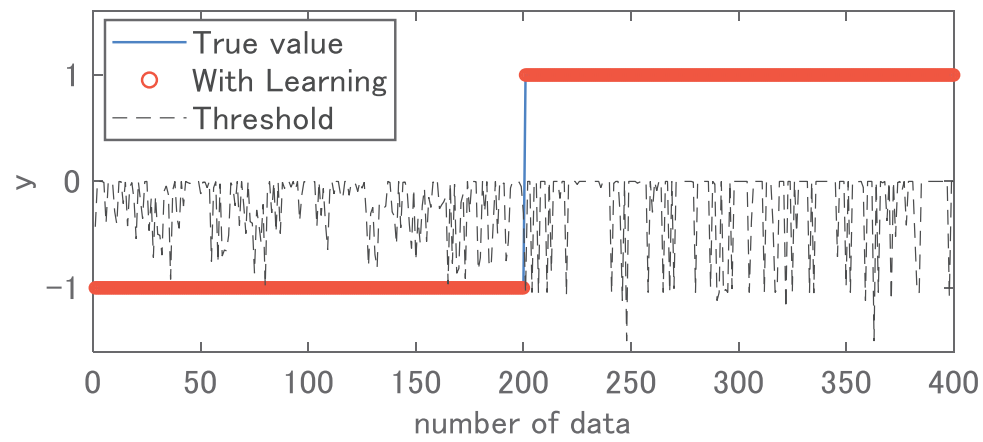
Fig. 3.2: Trajectories of the criterion E .



(a) JIT with binarization



(b) Proposed method (2 time iterations)

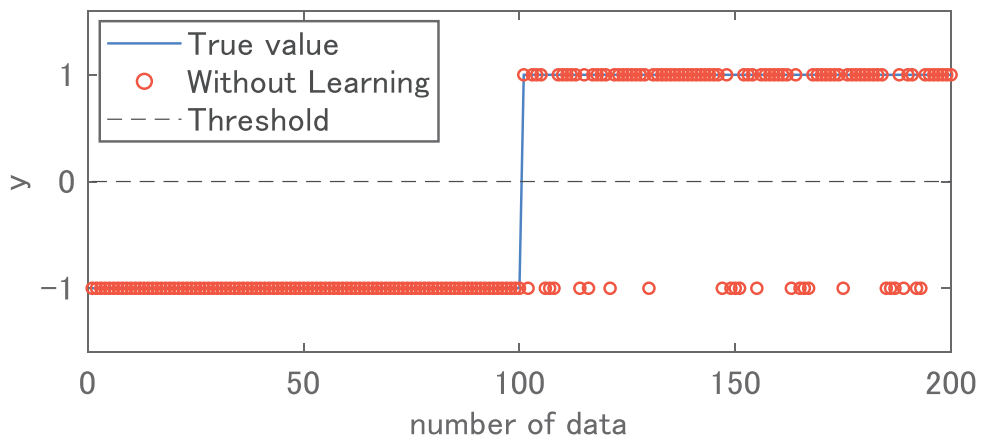


(c) Proposed method (55 time iterations)

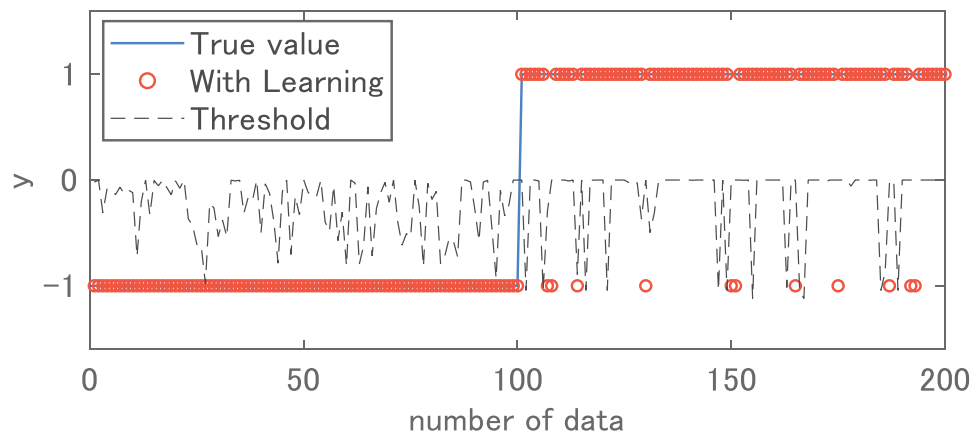
Fig. 3.3: The simulation result with the training data.

値が -1 の場合でも変化している。これは、正の値に修正量を更新されたデータベース内のデータが、真値が -1 である学習用データの近傍データとして選択され、(3.15)式によりその修正量の影響が表れたためである。また、(c) 提案法における 55 回の繰り返し学習後は、誤分類がないことが分かる。さらに、真値が -1 の場合よりも真値が 1 の場合の方が、しきい値が変化することが分かる。

Fig. 3.4 に、検証用データを用いた数値シミュレーション結果を示す。(b) 提案法では、(a) JIT 法の場合に発生していた誤分類が軽減されることが分かる。



(a) JIT with binarization



(b) Proposed method

Fig. 3.4: The simulation result with the test data.

Table 3.2 に、各予測手法における正解率を示す。Table 3.2 より、データベース駆動型アプローチによる予測では、従来法と比較し、正解率が向上することが確認できる。

また Table 3.3 のように初期データの数、学習用データの数を変化させた場合の、JIT 法および提案法による正解率の変化をそれぞれ、Fig. 3.5, Fig. 3.6 に示す。また正解率の平均をそれぞれ、Table 3.4, Table 3.5 に示す。Fig. 3.5 および Fig. 3.6 より、初期データの数および学習用データの数に関係なく、JIT 法による正解率よりも提案法による正解率の方が高いことが分かる。Table 3.4, Table 3.5 より、JIT 法による正解率よりも提案法による正解率の方が高いことが分かる。

Table 3.2: The average of the accuracy.

Neural network	0.58
JIT with binarization	0.90
Proposed method	0.93

Table 3.3: Fixed parameters in simulation.

Number of initial data	$N_I = [200, 400, 600, 800]$
Number of training data	$N_T = [500, 1000, 1500, 2000]$
Number of test data	$N_V = 200$
Threshold of similarity	$T_{th} = 0.2^4$
Threshold of output	$T_y = 0$
Learning rate	$\eta = 0.1$

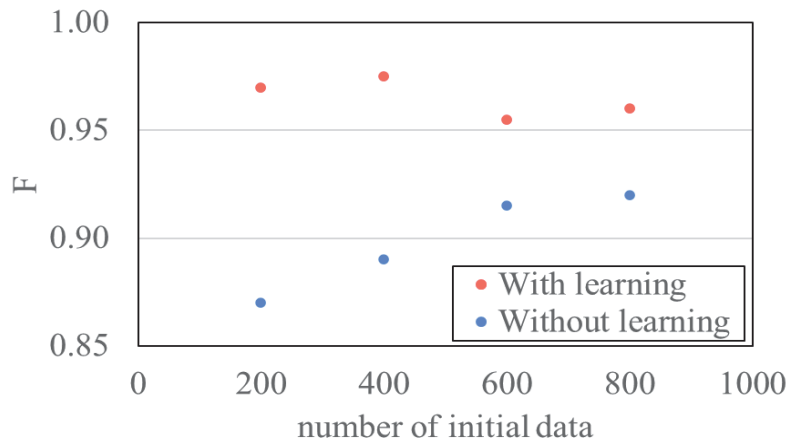


Fig. 3.5: The simulation result with the training data.

Table 3.4: The average of the accuracy.

JIT with binarization	0.90
Proposed method	0.97

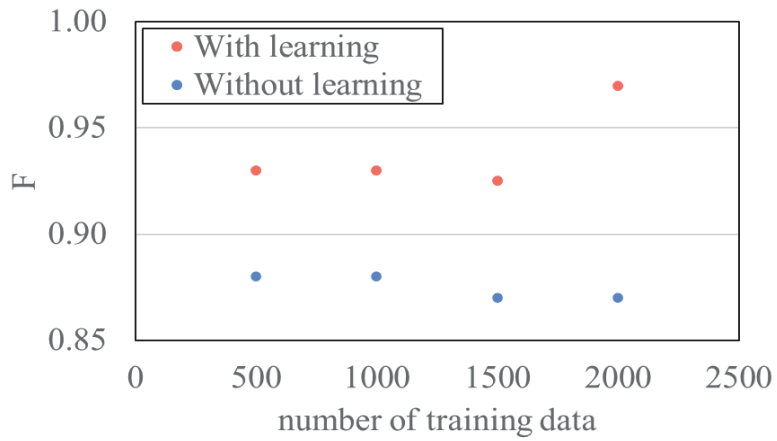


Fig. 3.6: The simulation result with the training data.

Table 3.5: The average of the accuracy.

JIT with binarization	0.88
Proposed method	0.94

3.4 結言

本章では，ロボットによる塗装作業から塗装品質の検査までの時間遅延による影響を低減するため，データベース駆動型塗装品質予測器の設計法について提案した．本手法は，塗装品質の正解率向上のため，データベース駆動型アプローチに基づき，2 値化の際のしきい値を適応的に変更する仕組みを有する塗装品質予測器を提案した．さらに，数値シミュレーションによりその有効性を検証した．その結果，従来法よりも提案法の正解率が高く，さらに修正量の学習によりしきい値を適応的に変更することで，誤分類が削減されることを示した．

第 4 章 データ駆動型スマート塗装システム の構築と実装

4.1 緒言

第 2 章では、環境条件に応じて動作条件を適応的に決定するため、JIT 法に基づく動作条件決定機構を構築した。また第 3 章では、環境条件と動作条件から塗装品質を予測するため、データベース駆動型アプローチに基づく塗装品質予測器を構築した。本章では、この動作条件決定機構と塗装品質予測器を統合した、データ駆動型スマート塗装システムを構築する。

4.2 データ駆動型スマート塗装システム

Fig. 4.1 に第 1 章で示したデータ駆動型スマート塗装システムの概要図を再掲する。このシステムは、データベース、①動作条件決定機構および②塗装品質予測器から構成される。データベースには、予め過去の塗装データを格納しておく。

データベースに格納されたデータのうち、 $\mathbf{z}(j) = [z_1(j), \dots, z_n(j)]$ は塗装ブース内の温度・湿度などの環境条件、 n はその次数、 $u(j)$ は動作条件（ベルカップ回転数）、 $y(j)$ は設備からの出力（塗装品質）であり 1（良好）または -1（不良）の値を取るとする。 $\mathbf{z}(j) = [z_1(j), \dots, z_n(j)]$ は、塗装ブース内の温度・湿度などの環境条件、 n はその次数である。

①の動作条件決定機構では、 $\mathbf{z}(t)$ を基にデータベースから近傍データを抽出し、その近傍データを基に動作条件 $u(t)$ を決定する。②の塗装品質予測器では、 $\mathbf{z}(t)$ および①で決定した $u(t)$ を基にデータベースから近傍データを抽出し、その近傍データを基に塗装品質の

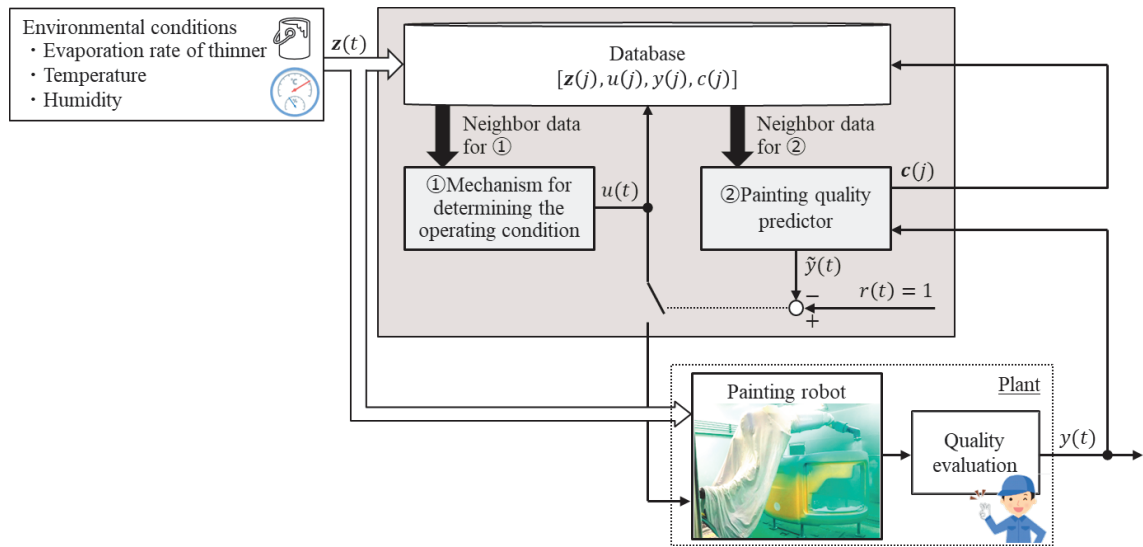


Fig. 4.1: Schematic of the data-driven smart painting process system.

予測値 $\hat{y}(t)$ を出力する。この $\hat{y}(t)$ は、良好(+1)もしくは不良(-1)で表され、 $\hat{y}(t)$ が不良(-1)を示す場合には、①で選択した近傍データを抽出し直すことで $u(t)$ を修正する。この作業を繰り返し、 $\hat{y}(t)$ が良好(+1)となる際の $u(t)$ を、設備の入力として与える。最後に、設備から得られた塗装品質の実測値 $y(t)$ を基に、②を介してデータベースの修正量 $c(j)$ を更新することで、②の正解率の向上につながり、①により適切な動作条件が決定されることになる。

Fig. 4.2 に本研究における提案法の概略図を示す。なお、多次元空間を描画できないため、Fig.4.2 では、 $z_1 - u$ 平面上で動作条件 $u(t)$ が決定される様子を示す。(a)クエリが入力されたとき、データベースより良好データおよび不良データを近傍データとして抽出する。その後、(b)①動作条件決定機構により良好および不良の分類境界を作成し、クエリにおいて出力が良好となるための良好データを有効とし、このデータを用いて $u(t)$ を計算する。その後、クエリおよび $u(t)$ の情報を基に、(c)②塗装品質予測器により予測値（離散値） $\hat{y}(t)$ を計算する。ここで $\hat{y}(t)$ が良好(+1)の場合、設備へこの $u(t)$ を入力する。 $\hat{y}(t)$ が不良(-1)の場合、次の $u(t)$ の候補を選択するため、(d)近傍データのうちクエリに最も近い

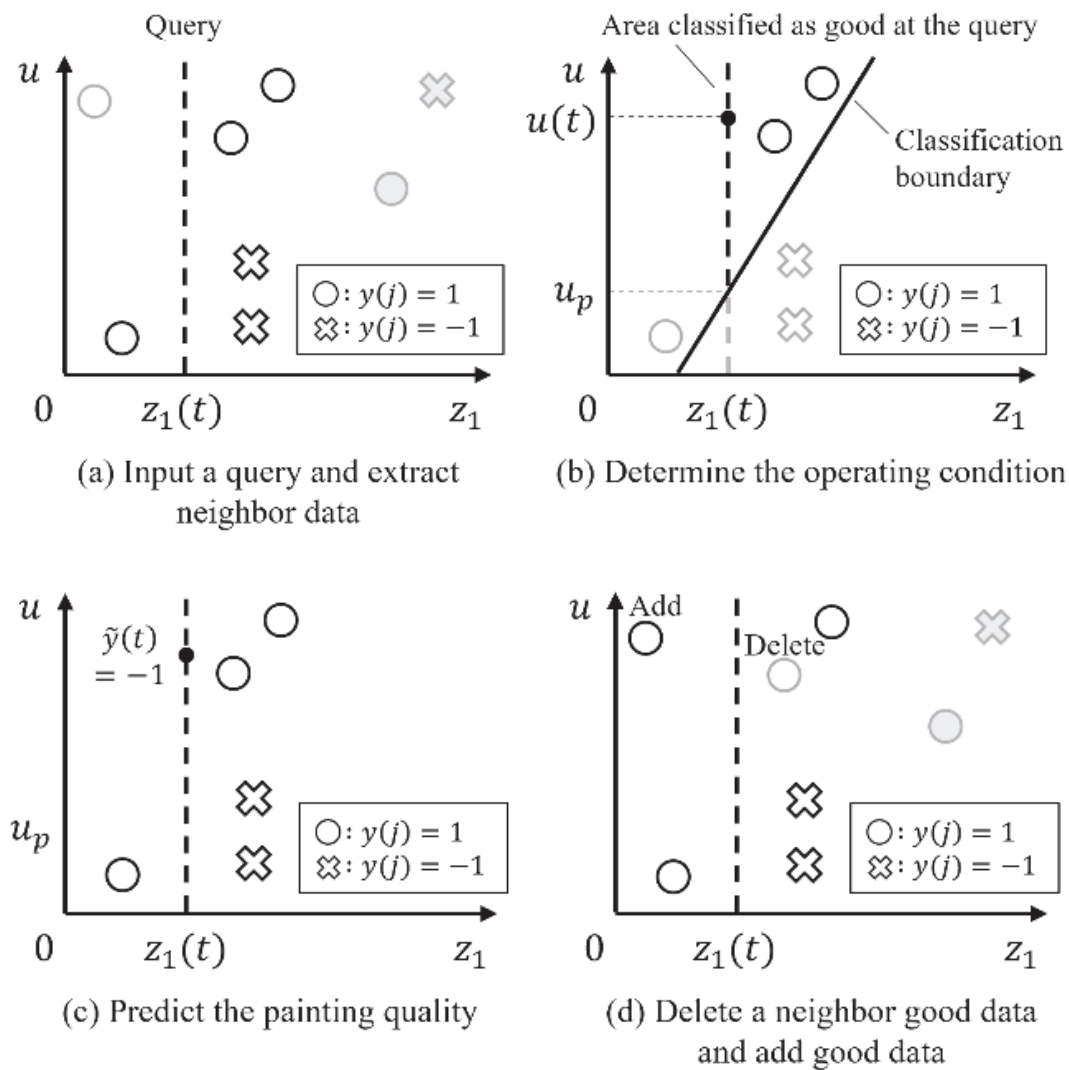


Fig. 4.2: Schematic of the proposed method.

良好データを削除する。また、近傍データがなくならないよう、データベースのデータのうち、クエリに最も近い良好データを近傍データとして追加する。その後、(b),(c),(d)を特定の回数だけ繰り返す。

〈4.2.1〉 近傍データの抽出 対象とする設備からの出力は、第 2 章および第 3 章と同

様に次式の離散時間非線形システムで記述される。

$$y(t) = f(\boldsymbol{\psi}_f(t)) \quad (4.1)$$

ここで、 $f(\cdot)$ は非線形関数、 $\boldsymbol{\psi}_f(t)$ は時刻 t における環境条件 $\mathbf{z}(t)$ 、動作条件 $u(t)$ を用い、次式で定義する。

$$\boldsymbol{\psi}_f(t) := [\mathbf{z}(t), u(t)] \quad (4.2)$$

JIT 法に基づき、情報ベクトル $\boldsymbol{\phi}(j)$ により構成される次式のデータセットにより、初期データベースを構築する。

$$\boldsymbol{\Phi}(j) := [\boldsymbol{\phi}(j), u(j), y(j), c(j)] \quad (4.3)$$

$$(ただし j = 1, 2, \dots, N_I)$$

$$\boldsymbol{\phi}(j) := [z_1(j), \dots, z_n(j)] \quad (4.4)$$

$$\bar{\boldsymbol{\phi}}(j) := [z_1(j), \dots, z_n(j), u(j)] \quad (4.5)$$

ここで、 $N_I, y(j)$ はそれぞれ、データベースのデータ数、塗装品質を示す目的変数である。

現在のシステムの状態 $\mathbf{q}(t)$ をクエリと呼び、次式で定義する。

$$\mathbf{q}(t) := [z_1(t), \dots, z_n(t)] \quad (4.6)$$

近傍データの基準として、時刻 t におけるクエリ $\mathbf{q}(t)$ とデータベース内の j 番目のデータ $\boldsymbol{\phi}(j)$ との類似度 $S(\mathbf{q}(t), \boldsymbol{\phi}(j))$ を計算する。

$$S(\mathbf{q}(t), \boldsymbol{\phi}(j)) = \prod_{l=1}^n \frac{1}{\sqrt{2\pi}h_l} \exp\left\{-\frac{(q_l(t) - \phi_l(j))^2}{2h_l^2}\right\} \quad (4.7)$$

ここで、 $q_l(t)$ は時刻 t におけるクエリの第 l 番目の要素を表す。 $\phi_l(j)$ はデータベース内の第 j 番目の情報ベクトル $\boldsymbol{\phi}(j)$ の第 l 番目の要素を表す。 h_l は情報ベクトルに対する各クラスのパンド幅であり、次式に示すプラグ・イン法により計算される。

$$h_l = \frac{1.06\sigma_l}{N^{\frac{1}{5}}} \quad (4.8)$$

ここで、各クラスの標準偏差 σ_l は次のように定義される。

$$\sigma_l = \sqrt{\frac{1}{N} \sum_{m=1}^N (\phi_l(m) - \mu_l)^2} \quad (4.9)$$

このとき、 N は各クラスのデータ数、 μ_l は各クラスの ϕ_l の平均値である。(4.7)式において、類似度が最大となるのは、クエリと同じ情報ベクトルがデータベースに存在する場合であり、類似度は次式として表記できる。

$$S(\mathbf{q}(t), \mathbf{q}(t)) = \prod_{l=1}^n \frac{1}{\sqrt{2\pi}h_l} \quad (4.10)$$

したがって、本研究では、次の類似度に基づく条件式を満たすデータ $\phi(j)$ を近傍データとして選択する。

$$\begin{aligned} S(\mathbf{q}(t), \phi(j)) &\geq T_1 \cdot S(\mathbf{q}(t), \mathbf{q}(t)) \\ &= T_1 \cdot \prod_{l=1}^n \frac{1}{\sqrt{2\pi}h_l} \end{aligned} \quad (4.11)$$

ここで、 T_1 は類似度のしきい値であり、その範囲は $0 \leq T_1 \leq 1$ である。ここで、各クラスの近傍データの数は、(4.2)式の $\psi_f(t)$ の要素数である $(n+1)$ 件以上選択されることとする。各クラスの近傍データの数が $(n+1)$ 件未満の場合、そのクラスの中で類似度の高い上位 $(n+1)$ 件のデータを、そのクラスの近傍データとする。選択された近傍データの数を $k(t)$ とする。

〈4.2.2〉 **動作条件の計算** 選択した近傍データを基に、「2.2.5 有効な良好データの定義」より有効な良好データを決定する。この有効な良好データの数を $g(t)$ とする。データベースからの出力 $u(t)$ は、有効な良好データを用い、LWAにより計算する。

$$u(t) = \sum_{i=1}^{g(t)} v_i u(i), \quad \sum_{i=1}^{g(t)} v_i = 1 \quad (4.12)$$

ここで、 v_i はデータベースより選択された、第 i 番目の情報ベクトル $\phi(i)$ に対する重みであり、次式で与える。

$$v_i = \frac{S(\mathbf{q}(t), \boldsymbol{\phi}(j))}{\sum_{i=1}^{g(t)} S(\mathbf{q}(t), \boldsymbol{\phi}(j))} \quad (4.13)$$

〈4.2.3〉 予測値の計算 ②の塗装品質予測器に対するクエリ $\bar{\mathbf{q}}(t)$ は以下で表される.

$$\bar{\mathbf{q}}(t) := [z_1(t), \dots, z_n(t), u(t)] \quad (4.14)$$

ここで, $z_1(t), \dots, z_n(t)$ は(4.6)式のクエリ $\mathbf{q}(t)$, $u(t)$ は(4.12)式の動作条件である.

クエリ $\bar{\mathbf{q}}(t)$ とデータベース内の j 番目のデータ $\bar{\boldsymbol{\phi}}(j)$ との類似度 $S(\bar{\mathbf{q}}(t), \bar{\boldsymbol{\phi}}(j))$ は以下で表される.

$$S(\bar{\mathbf{q}}(t), \bar{\boldsymbol{\phi}}(j)) = \prod_{l=1}^{n+1} \frac{1}{\sqrt{2\pi}h_l} \exp \left\{ -\frac{(\bar{q}_l(t) - \bar{\phi}_l(j))^2}{2h_l^2} \right\} \quad (4.15)$$

バンド幅 h_l および各クラスの標準偏差 σ_l は, それぞれ(7)式, (8)式と同じである. 近傍データは, 次の類似度に基づく条件式を満たすデータ $\bar{\boldsymbol{\phi}}(j)$ である.

$$S(\bar{\mathbf{q}}(t), \bar{\boldsymbol{\phi}}(j)) = T_2 \prod_{l=1}^{n+1} \frac{1}{\sqrt{2\pi}h_l} \quad (4.16)$$

ここで, T_2 は類似度のしきい値である. また近傍データの数を $k(t)$ とする.

データベースからの出力 $\hat{y}(t)$ は, 近傍データを用い, LWA により計算する.

$$\hat{y}(t) = \sum_{i=1}^{k(t)} \bar{v}_i (y(i) + c(i)), \quad \sum_{i=1}^{k(t)} \bar{v}_i = 1 \quad (4.17)$$

ここで, \bar{v}_i はデータベースより選択された, 第 i 番目の情報ベクトル $\bar{\boldsymbol{\phi}}(i)$ に対する重みであり, 次式で与える.

$$\bar{v}_i = \frac{S(\bar{\mathbf{q}}(t), \bar{\boldsymbol{\phi}}(j))}{\sum_{i=1}^{k(t)} S(\bar{\mathbf{q}}(t), \bar{\boldsymbol{\phi}}(j))} \quad (4.18)$$

また, $\hat{y}(t)$ は連続値であるため, これを2値化する出力 $\tilde{y}(t)$ を, しきい値 T_y を用いて次のように定義する.

$$\tilde{y}(t) := \begin{cases} 1 & (\hat{y}(t) \geq T_y) \\ -1 & (\hat{y}(t) < T_y) \end{cases} \quad (4.19)$$

〈4.2.4〉 近傍データの修正

(4.19)式の $\tilde{y}(t)$ が 1 (良好) を示す場合, (4.12)式の $u(t)$ を Fig. 4.1 に示す設備の動作条件として入力する.

(4.19)式の $\tilde{y}(t)$ が -1 (不良) を示す場合, ①の動作条件決定機構により次の動作条件を決定するため, 以下の処理により近傍データの修正, 動作条件の修正をおこなう.

1. 有効な良好データのうち, (4.11)式の類似度 $S(\mathbf{q}(t), \boldsymbol{\phi}(j))$ が最大となるデータ 1 件を削除
2. (4.11)式の条件式を満たさないデータのうち, (4.11)式の類似度 $S(\mathbf{q}(t), \boldsymbol{\phi}(j))$ が最大となるデータ 1 件を, ①の動作条件決定機構に対する近傍データに追加
3. 新たな近傍データで「4.2.2 動作条件の計算」, 「4.2.3 予測値の計算」を再度おこない, (4.19)式の $\tilde{y}(t)$ が 1 (良好) を示すまで繰り返す.

〈4.2.5〉 修正量の更新 $\tilde{y}(t)$ と設備からの出力 $y(t)$ が異なる場合, 以下の最急降下法

により近傍データの修正量 $c(i)$ を更新する.

$$c(i) \leftarrow c(i) - \eta \frac{\partial J(t)}{\partial c(i)} \quad (4.20)$$

ここで, η は修正量の学習係数である. また $J(t)$ は 2 乗誤差関数であり, 次式のように定義する.

$$J(t) = \frac{1}{2} (y(t) - \hat{y}(t))^2 \quad (4.21)$$

(4.21)式は微分連鎖則に基づき, 次式のように表記できる.

$$\begin{aligned} c(i) &\leftarrow c(i) - \eta \frac{\partial J(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial c(i)} \\ &= c(i) - \eta \bar{v}_i (\hat{y}(t) - y(t)) \end{aligned} \quad (4.22)$$

一方, $\tilde{y}(t)$ と設備からの出力 $y(t)$ が同じ場合, 修正量 $c(i)$ は更新せず, $J(t)$ は 0 とする. 全ての学習用データを用いて修正量の更新を行ったあと, 次式により $J(t)$ の総和 E を計算

する.

$$E = \sum_{t=1}^{N_T} J(t) \quad (4.23)$$

ここで、 N_T は学習用データの数である。 E が予め設定した学習終了条件を満たすまで、同じ学習用データを用いて修正量の更新を行う。

以下に、これまでに述べた手順をアルゴリズムとして示す。

【データ駆動型スマート塗装システムのアルゴリズム】

1. 初期データベースの構築
2. 検証データからクエリ $\mathbf{q}(t)$ を抽出し、(4.11)式の条件式を満たす近傍データ $\boldsymbol{\phi}(j), (j = 1, 2, \dots, k(t))$ を、データベースより抽出
3. 近傍データから有効な良好データを決定し、(4.12)式により動作条件 $\mathbf{u}(t)$ を計算
4. クエリ $\mathbf{q}(t)$ および動作条件 $\mathbf{u}(t)$ より、予測値 $\hat{y}(t)$ を計算
5. $\hat{y}(t)$ が -1 (不良) の場合、有効な良好データのうち (4.11)式の類似度 $S(\mathbf{q}(t), \boldsymbol{\phi}(j))$ が最大となるデータ 1 件を削除し、(4.11)式の条件式を満たさないデータのうち(4.11)式の類似度 $S(\mathbf{q}(t), \boldsymbol{\phi}(j))$ が最大となるデータ 1 件を、近傍データに追加
6. $\hat{y}(t)$ が 1 (良好) となるまで、3, 4 を繰り返す

次章では、数値シミュレーションを通して本提案法の有効性を検証する。

4.3 数値シミュレーション

〈4.3.1〉 対象システム 数値シミュレーションでは、塗装工程を模擬した 4 入力 1 出力系のシステムを取り扱う。(4.2)式の $\boldsymbol{\psi}_f(t)$ は、塗装ブース内温度 $z_1(t)$ [°C]、塗装ブース内湿度 $z_2(t)$ [%]、シンナーの蒸発速度 $z_3(t)$ [$10^{-1}\%/min$]、ベルカップ回転数 $u(t)$ [krpm] を用い、次式のように定義する。

$$\boldsymbol{\psi}_f(t) := [\mathbf{z}(t), u(t)] \quad (4.24)$$

ただし、 $\mathbf{z}(t) = [z_1(t), z_2(t), z_3(t)]$ である。また(4.4)式の情報ベクトル $\boldsymbol{\phi}(j)$ および(4.5)式の情報ベクトル $\bar{\boldsymbol{\phi}}(j)$ 、(4.6)式のクエリ $\mathbf{q}(t)$ および(4.14)式のクエリ $\bar{\mathbf{q}}(t)$ は、それぞれ次式のように定義する。

$$\boldsymbol{\phi}(j) := [z_1(j), z_2(j), z_3(j)] \quad (4.25)$$

$$\bar{\boldsymbol{\phi}}(j) := [z_1(j), z_2(j), z_3(j), u(j)] \quad (4.26)$$

$$\mathbf{q}(t) := [z_1(t), z_2(t), z_3(t)] \quad (4.27)$$

$$\bar{\mathbf{q}}(t) := [z_1(t), z_2(t), z_3(t), u(t)] \quad (4.28)$$

出力 $y(t)$ の値は、塗装品質が良好な状態を 1、塗装品質が不良な状態を -1 とする。

実際の運転データの数は少なく、さらに良好データに比べて不良データの数が少ないため、第 2 章および第 3 章で用いた SVM による分類モデルを仮想設備とし、提案法の有効性を検証する。本数値例において、環境条件や動作条件の変化を考慮し、 $z_1(t), z_2(t), u(t)$ はそれぞれ、 $[20, 30]$, $[30, 75]$, $[15, 40]$ の範囲で等間隔に 30 件ずつ与える。そのため、仮想的な運転データの総数は $30^3 = 27000$ 件である。なお、 $z_3(t)$ は一定の値をとることとする。このデータセットから、初期データセット、学習用データセットおよび検証用データセットをランダムに抽出する。そのため、各データセットに含まれるデータは、時間的にランダムになっていることに注意されたい。

〈4.3.2〉 シミュレーション結果 まず、歩留まりを検証するために、次式で示される歩留まり F を導入する。

$$F := \frac{N_M}{N_V} \quad (4.29)$$

ここで、 N_M は検証データの出力 $y(t)$ が 1 (良好) となる検証データの数、 N_V は検証データの数である。

提案法の有効性を検証するため、初期データベースによる動作条件の決定 (従来法) および修正量のオフライン学習による動作条件の決定 (提案法) により数値シミュレーショ

ンをおこなう.

本研究では, (4.23)式の 2 乗誤差関数の総和 E が 1.5 以下となる, もしくは繰り返し計算の回数が 5000 回となる時, 繰り返し計算が終了することとする.

シミュレーションの設定値を Table 4.1 に示す. 提案法の汎用性を検証するため, 抽出の仕方の異なるデータセットを 5 種類ほど用意する. 数値シミュレーションによる歩留まりは, 5 種類のシミュレーション結果の平均値とする.

Fig. 4.3 に, 繰り返し学習による E の推移を示す. E は繰り返し学習に伴い減少し, 最終的には収束していることが分かる. このデータセットの例では, 152 回の繰り返し学習で

Table 4.1: Fixed parameters in simulation.

Number of initial data	$N_I = 100$
Number of training data	$N_T = 100$
Number of test data	$N_V = 200$
Threshold of similarity	$T_1 = 0.216$
Threshold of similarity	$T_2 = 0.0016$
Threshold of output	$T_y = 0$
Learning rate	$\eta = 0.2$

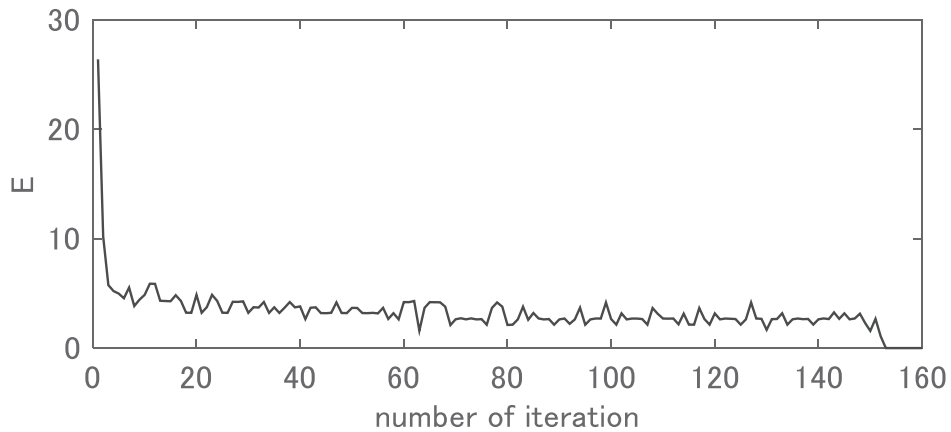


Fig. 4.3: Trajectories of E .

E が 1.5 以下となった.

Fig. 4.4 に, 従来法の検証結果として, 設備からの出力 $y(t)$ およびしきい値 $T_y - \sum_{i=1}^{k(t)} \bar{v}_i c(i)$ の変化を示す. 不良発生の要因として, 設備の出力が-1(不良)となるようなクエリ $\bar{q}(t)$ の場合でも, 塗装品質予測器の正解率が 1(良好)と予測し, 近傍データの修正がおこなわれなかったためと考えられる.

Fig. 4.5 に, 提案法の検証結果を示す. 従来法で誤分類が発生していた箇所では, とくにしきい値が変化しており, これにより塗装品質予測器の正解率が向上している. オフライン学習により修正量 $c(i)$ を学習することが, 歩留まりの改善に有効であることが分かった.

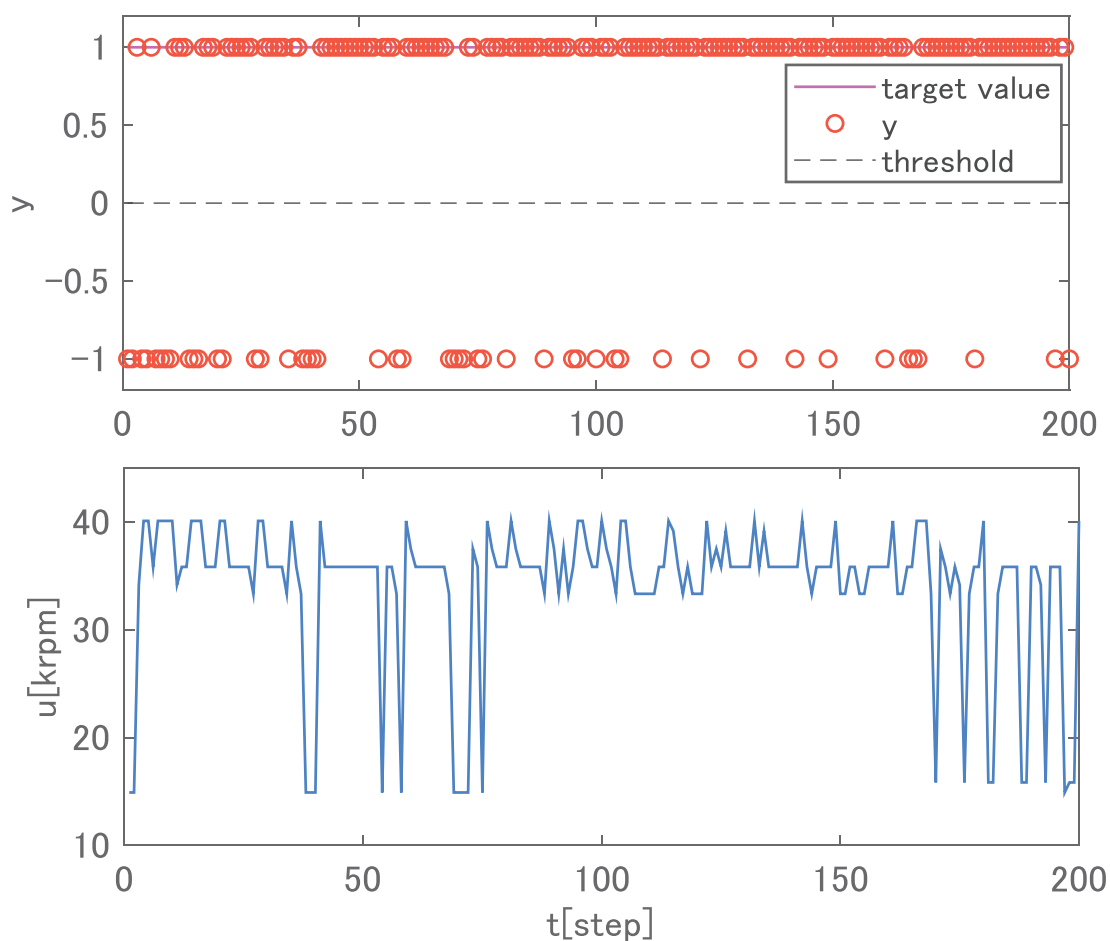


Fig. 4.4: The simulation result of conventional method.

Table 4.2 に、各手法における歩留まりを示す。提案法では、従来法と比較し歩留まりの向上が確認できる。

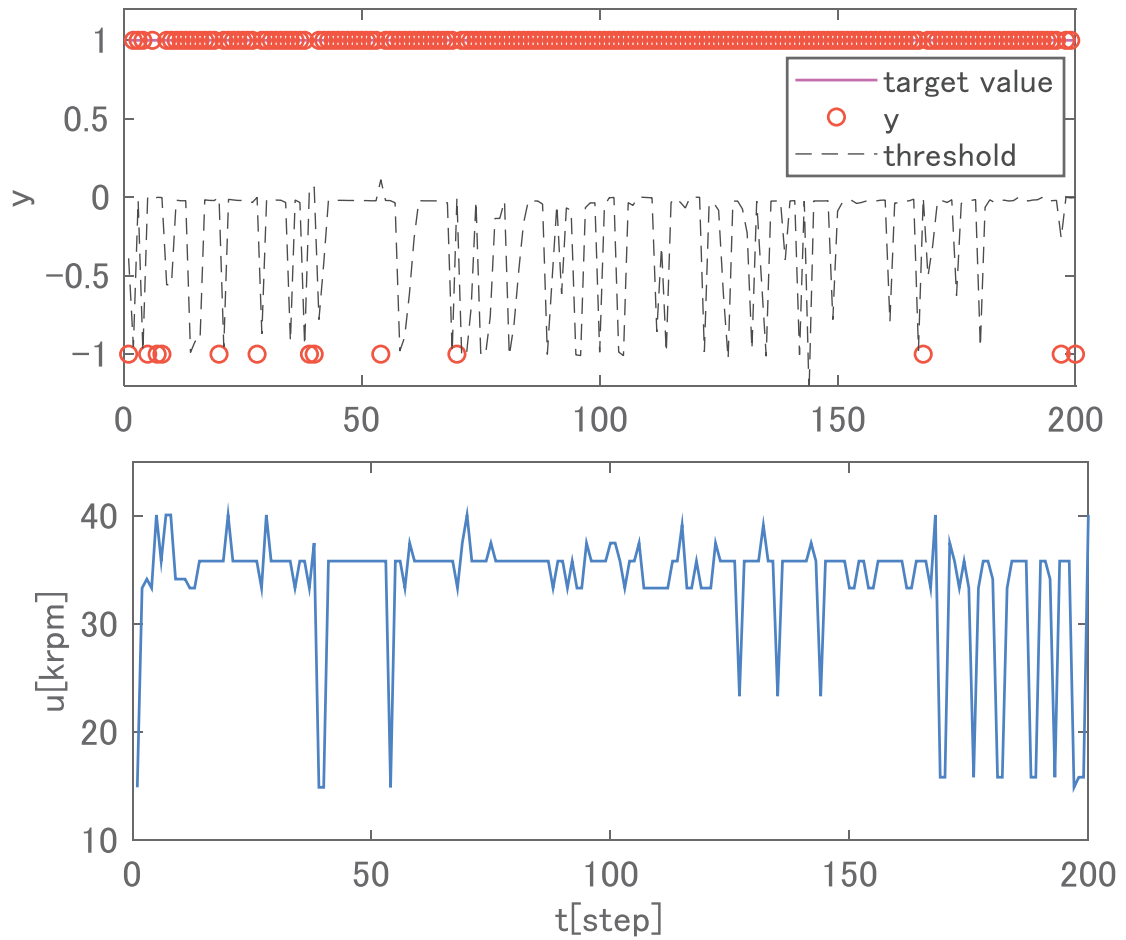


Fig. 4.5: The simulation result of proposed method.

Table 4.2: The simulation result of the average of the yield.

Conventional method	0.952
Proposed method	0.987

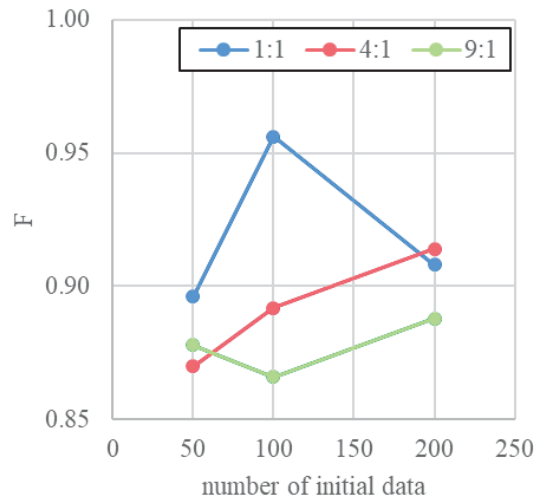
次に Table 4.3 のように、初期データの数、初期データの良好数と不良数との比率（良好数：不良数。以下、良好数と不良数のクラス比）を変化させた場合の歩留まりの変化を Fig. 4.6 に示す。なお、 $z_1(t), z_2(t), z_3(t), u(t)$ はそれぞれ、 $[0, 35], [30, 75], [15, 35], [15, 40]$ の範囲で変化することとしている。

ここで、近傍データの修正による効果を確認するため、比較対象は(a) 動作条件決定機構による動作条件の決定、(b) 近傍データの修正あり（学習なし）とする。また歩留まりの平均値を Table. 4.4 に示す。

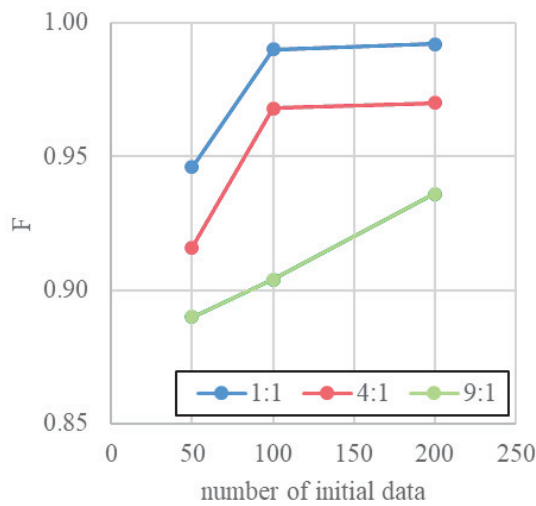
Fig. 4.6 および Table. 4.4 より、(a) 動作条件決定機構による動作条件の決定による歩留まりよりも、(b) 近傍データの修正あり（学習なし）による歩留まりの方が高いことが分かる。この結果から、近傍データを修正することが歩留まりの向上につながる事がわかる。

Table 4.3: Fixed parameters in simulation.

Number of initial data	$N_I = [50, 100, 200]$
Number of test data	$N_V = 200$
Class ratio of initial data	$[9: 1, 4: 1, 1: 1]$
Threshold of similarity	$T_1 = 0.216$
Threshold of similarity	$T_2 = 0.0016$
Threshold of output	$T_y = 0$
Learning rate	$\eta = 0.2$



(a) Determination of operating condition by mechanism for determining the operating condition



(b) Modify neighborhood data (Without learning)

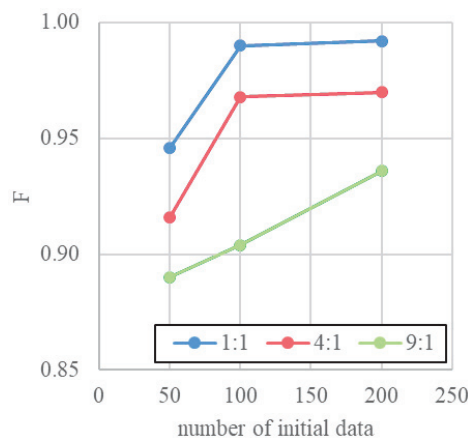
Fig. 4.6: The simulation result of the yield.

Table 4.4: The simulation result of the average of the yield.

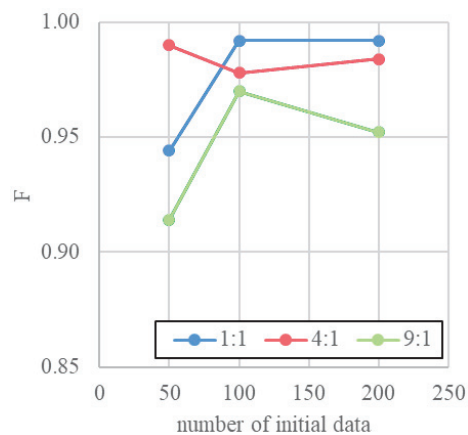
(a) Determination of operating condition by mechanism for determining the operating condition	0.896
(b) Modify neighborhood data (Without learning)	0.946

次に、オフライン学習による効果を確認するため、比較対象を(a) 近傍データの修正あり (学習なし), (b) 近傍データの修正あり (オフライン学習あり) とし, Table 4.3 と同じ条件で数値シミュレーションをおこなった結果を Fig. 4.7 に示す. また歩留まりの平均値を Table. 4.5 に示す.

Fig. 4.7 および Table. 4.5 より, (a) 近傍データの修正あり (学習なし) による歩留まりよりも, (b) 近傍データの修正あり (オフライン学習あり) による歩留まりの方が高いことが分かる. この結果から, オフライン学習により歩留まりが向上することがわかる.



(a) Modify neighborhood data (Without learning)



(b) Modify neighborhood data (With offline learning)

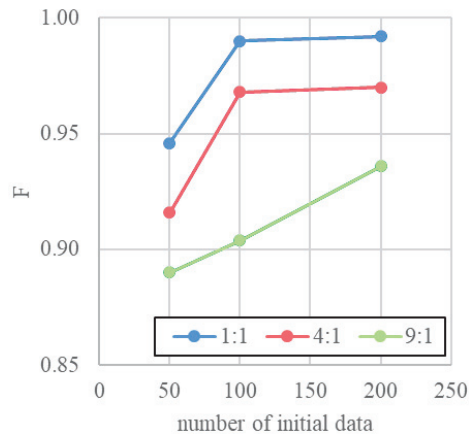
Fig. 4.7: The simulation result of the yield.

Table 4.5: The simulation result of the average of the yield.

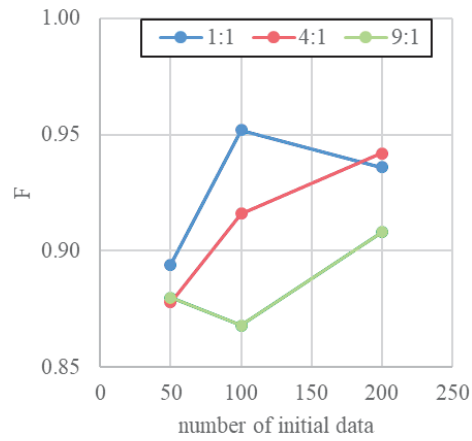
(a) Modify neighborhood data (Without learning)	0.946
(b) Modify neighborhood data (With offline learning)	0.966

次に、オンライン学習による効果を確認するため、比較対象を(a) 近傍データの修正あり（学習なし）、(b) 近傍データの修正あり（オンライン学習あり）、(c) 近傍データの修正あり（検証データの追加による学習）とし、Table 4.3 と同じ条件で数値シミュレーションをおこなった結果を Fig. 4.8 に示す。また歩留まりの平均値を Table. 4.6 に示す。なお、(c)の検証データの追加による学習とは、検証データ 1 件の検証が終わるたびに、その検証データをデータベースへ追加することを意味する。

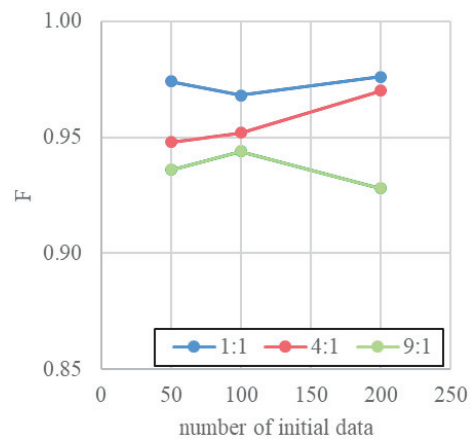
Fig. 4.8 および Table. 4.6 より、(b) 近傍データの修正あり（オンライン学習あり）および (c) 近傍データの修正あり（検証データの追加による学習）による歩留まりよりも、(a) 近傍データの修正あり（学習なし）による歩留まりの方が高い。これは、検証データ 1 件ごとの環境条件の変化が大きく、オンライン学習または検証データの追加による学習により、過学習が発生したためだと考えられる。



(a) Modify neighborhood data (Without learning)



(b) Modify neighborhood data (With online learning)



(c) Modify neighborhood data (With adding test data to the database)

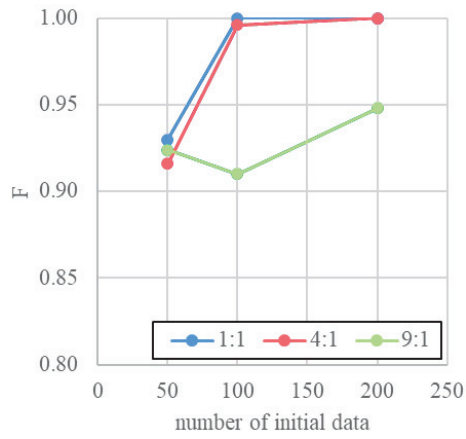
Fig. 4.8: The simulation result of the yield.

Table 4.6: The simulation result of the average of the yield.

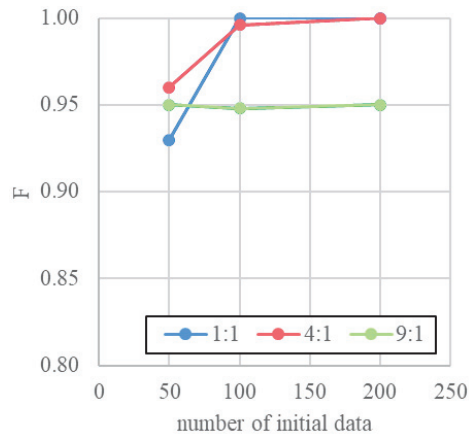
(a) Modify neighborhood data (Without learning)	0.946
(b) Modify neighborhood data (With online learning)	0.909
(c) Modify neighborhood data (With adding test data to the database)	0.955

次に、実環境を想定した環境条件により数値シミュレーションをおこなう。具体的には、 $z_1(t), z_2(t), u(t)$ はそれぞれ、 $[20, 30], [30, 75], [15, 40]$ の範囲で変化し、 $z_3(t)$ は一定の値をとることとする。他のパラメータは Table 4.3 と同等とする。比較対象を、(a) 近傍データの修正あり（学習なし）、(b) 近傍データの修正あり（オンライン学習あり）、(c) 近傍データの修正あり（検証データの追加による学習）とし、数値シミュレーションをおこなった結果を Fig. 4.9 に示す。また歩留まりの平均値を Table. 4.7 に示す。

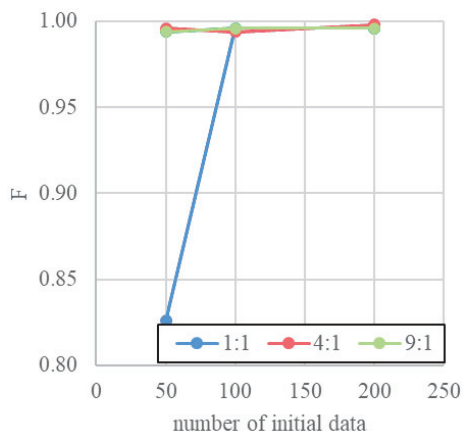
Fig. 4.9 および Table. 4.7 より、(a) 近傍データの修正あり（学習なし）による歩留まりよりも、(b) 近傍データの修正あり（オンライン学習あり）および(c) 近傍データの修正あり（検証データの追加による学習）による歩留まりの方が高い。これは上述の過学習が解消されたためだと考える。ただし、(c) 近傍データの修正あり（検証データの追加による学習）では、初期データの数が 50、初期データ的良好数と不良数のクラス比が 1:1 のときに歩留まりが低い。これは、動作条件決定機構により近傍データを線形分離する際、正則化係数 C が大きいために正しく分離されず、有効な良好データが決定できなかったためだと考えられる。



(a) Modify neighborhood data (Without learning)



(b) Modify neighborhood data (With online learning)



(c) Modify neighborhood data (With adding test data to the database)

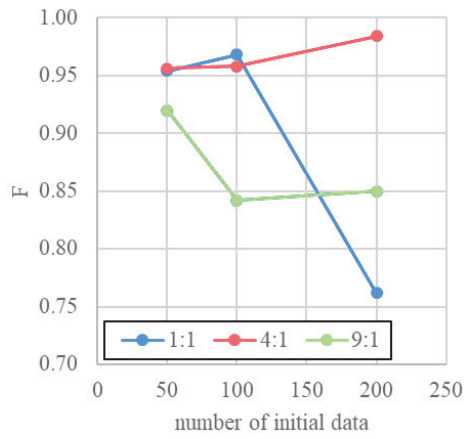
Fig. 4.9: The simulation result of the yield.

Table 4.7: The simulation result of the average of the yield.

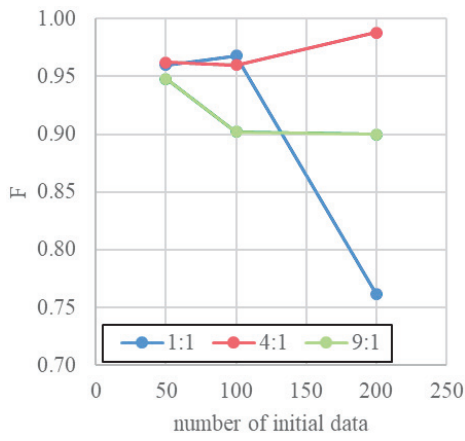
(a) Modify neighborhood data (Without learning)	0.958
(b) Modify neighborhood data (With online learning)	0.979
(c) Modify neighborhood data (With adding test data to the database)	0.971

最後に、正則化係数 C を変更した場合の数値シミュレーションをおこなう。具体的には、 C を 1 から 0.01 へと変更する。他のパラメータは Table 4.3 と同等とする。比較対象を、(a) 近傍データの修正あり（学習なし）、(b) 近傍データの修正あり（オンライン学習あり）、(c) 近傍データの修正あり（検証データの追加による学習）とし、数値シミュレーションをおこなった結果を Fig. 4.10 に示す。また歩留まりの平均値を Table. 4.8 に示す。

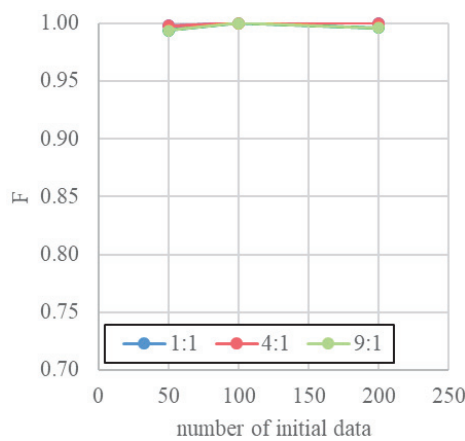
Fig. 4.10 および Table. 4.8 より、(a) 近傍データの修正あり（学習なし）による歩留まりよりも、(b) 近傍データの修正あり（オンライン学習あり）および(c) 近傍データの修正あり（検証データの追加による学習）による歩留まりの方が高い。これは上述の過学習が解消されたためだと考える。



(a) Modify neighborhood data (Without learning)



(b) Modify neighborhood data (With online learning)



(c) Modify neighborhood data (With adding test data to the database)

Fig. 4.10: The simulation result of the yield.

Table 4.8: The simulation result of the average of the yield.

(a) Modify neighborhood data (Without learning)	0.910
(b) Modify neighborhood data (With online learning)	0.928
(c) Modify neighborhood data (With adding test data to the database)	0.999

本節では、JIT法の考え方にに基づき、良好データおよび不良データを用いたデータ駆動型スマート塗装システムを提案し、数値シミュレーションを通してその有効性を検証した。

具体的には、動作条件決定機構により決定された動作条件を基に、塗装品質予測器により出来上がりの塗装品質を予測する。予測値が-1(不良)を示す場合には、近傍データを修正し、再度動作条件決定により動作条件を決定する。数値シミュレーションを通して、提案法における歩留まりの向上が確認できた。とくに提案法では、塗装品質予測器の正解率向上により、塗装品質が良好となる可能性を示した。

また実環境を想定し、オンライン学習の方法やシミュレーションの設定値を変更して数値シミュレーションを実施した。最終的に結果が良くなったときの設定値を Table 4.9 に示す。ただし、SVMの正則化係数など、実際のデータの分布により調整が必要なパラメータは、試行錯誤的に決定する必要がある。

Table 4.9: Fixed parameters in simulation.

Learning method	Add test data to the database
Number of initial data	$N_I = [50, 100, 200]$
Number of test data	$N_V = 200$
Class ratio of initial data	[9: 1, 4: 1, 1: 1]
Threshold of similarity	$T_1 = 0.216$
Threshold of similarity	$T_2 = 0.0016$
Threshold of output	$T_y = 0$
Regularization coefficient	$C = 0.01$
Learning rate	$\eta = 0.2$

4.4 実装と評価検証

本システムの実装対象は、コベルコ建機株式会社の油圧ショベルのカウンタウェイト塗装工程において、Fig. 4.11 に示す量産で使用されている塗装ロボットとする。カウンタウェイトは油圧ショベルの後方に取り付けられる部品であり、土を掘削する際、土の重さにより油圧ショベル本体が前方に転倒しないよう、自身の重さによりバランスを取る役割を担っている。カウンタウェイトは人の目線の高さに取り付けられることもあり、その塗装に求められる品質基準が厳しい。

コベルコ建機株式会社では、塗装ロボットによる塗装作業がおこなわれた後、通常乾燥、強制乾燥、品質検査がおこなわれる。品質検査では、人の目視により前述の塗装不具合の有無が判断される。

塗装不具合の種類として、1度対応すれば以降塗装不具合が発生しないもの、都度対応が必要なものに大別される。前者は被塗装物の局所に発生するようなタレやゆず肌があり、塗装ロボットの軌跡を修正することで解決する。後者は被塗装物に大域的に発生するようなタレやゆず肌であり、ベルカップ回転数の設定値を修正することで解決する。

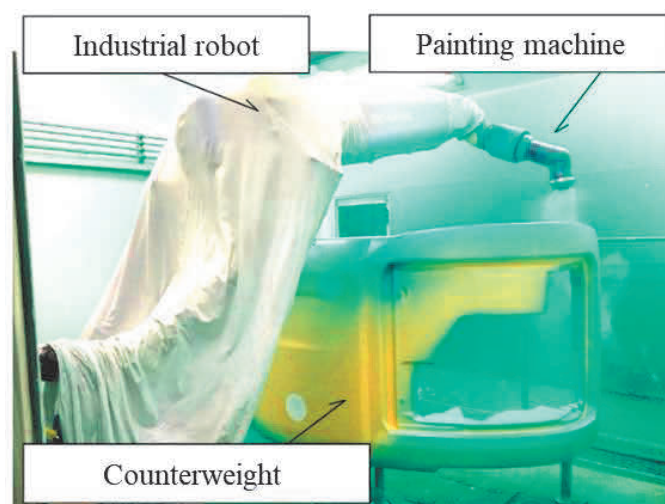


Fig. 4.11: Painting robot and counterweight.

〈4.4.1〉 機器構成

実装に用いた塗装ロボットを Fig. 4.12 に示す。塗装ロボットは塗装ブースの中に設置されており、塗装ブース外の制御盤と繋がっている。この制御盤に、イーサネットケーブルを用いて PC と接続する。PC には温湿度計を接続する。この PC 内には、データベース (Microsoft Excel)、制御盤との通信ソフトウェア (Microsoft Excel)、①動作条件決定機構および②塗装品質予測器の実行プログラム (python) が存在している。シンナーの蒸発速度は、制御盤との通信ソフトウェアへ直接入力する。①動作条件決定機構および②塗装品質予測器の実行プログラム (python) には、温湿度計からの現在の温度および湿度の情報の取得、制御盤との通信ソフトウェア (Microsoft Excel) のシンナーの蒸発速度の読み込み、データベース (Microsoft Excel) の読み込みの機能が存在する。

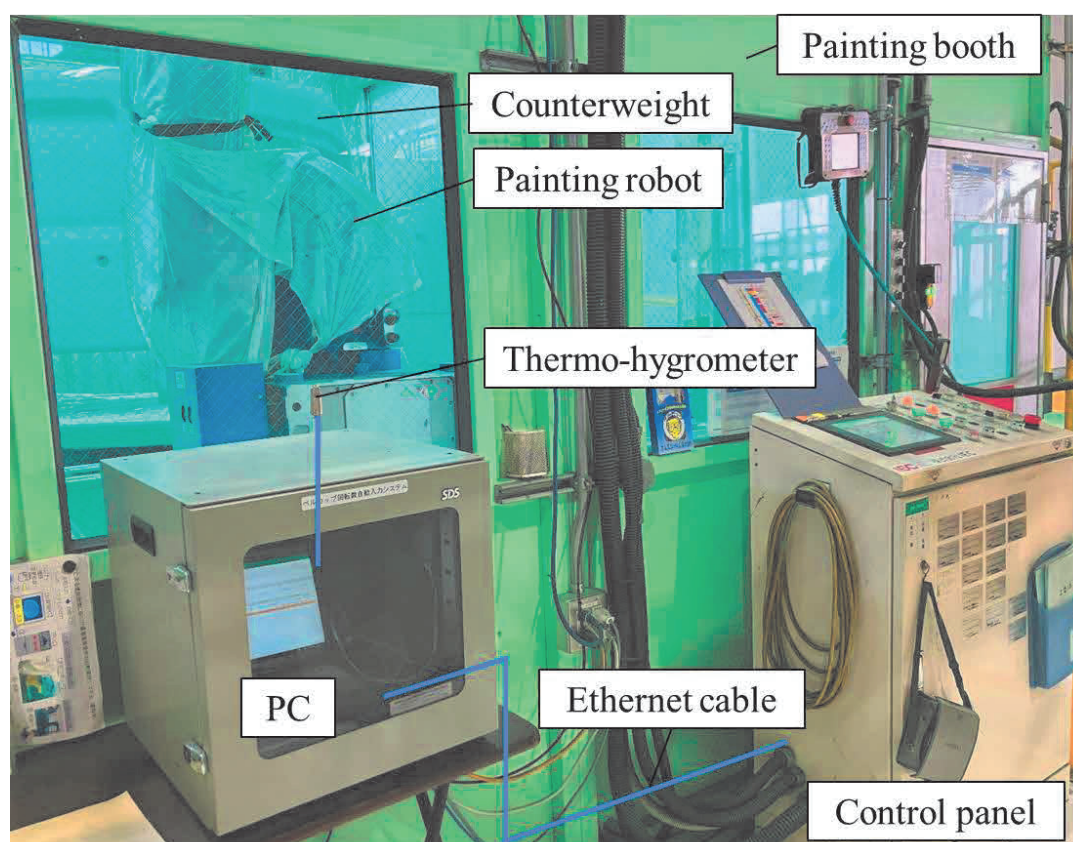


Fig. 4.12: Painting robot and counterweight.

実装におけるデータ駆動型スマート塗装システムのフローを以下に示す。

【データ駆動型スマート塗装システムのフロー】

1. 塗装ブースへのカウンタウェイトの搬送指示を制御盤が受け、制御盤が PC にその信号を送信する
2. PC 内の①動作条件決定機構および塗装品質予測器の実行プログラムを起動し、温湿度計から現在の温度および湿度の情報の取得し、シンナーの蒸発速度およびデータベースの読み込みをおこない、ベルカップ回転数を計算する
3. 2 で取得された温度の情報を基に Table 4.9 よりベルカップ回転数の基準値を決定する。2 で計算されたベルカップ回転数が、この基準値から $\pm 2.5\text{krpm}$ の範囲に収まる場合はそのベルカップ回転数を使用し、範囲外の場合は基準値を使用する
4. PC から制御盤にベルカップ回転数が送信され、制御盤内のベルカップ回転数が更新される
5. 2 で取得された温度、湿度、シンナーの蒸発速度、3 で決定されたベルカップ回転数がデータベースに格納される
6. 更新されたベルカップ回転数で塗装ロボットによる塗装作業が実施される
7. 翌日の操業開始前に、前日に塗装したカウンタウェイトの品質評価結果がデータベースに格納される

Table 4.9: Reference value of bell cup rotation speed.

Temperature z_1 [-]	Rotation speed of the bell cup u [-]
$30 \leq z_1$	20
$25 \leq z_1 < 30$	20
$20 \leq z_1 < 25$	25
$15 \leq z_1 < 20$	25
$7 \leq z_1 < 15$	30

〈4.4.2〉 実装準備

初期データベースを構築するため、2019年5月22日から2019年8月23日に取得した181件の操作データを用いる。初期データの数はいくつかあり、良好数と不良数のクラス比は166:15である。各要素 $z_1(j), z_2(j), z_3(j), u(j)$ の範囲はそれぞれ、 $[21.3, 34.2], [30, 75], [17.16, 19.8], [15, 27]$ である。Fig. 4.13 に初期データベースの要素ごとの相関図をクラス別に示す。

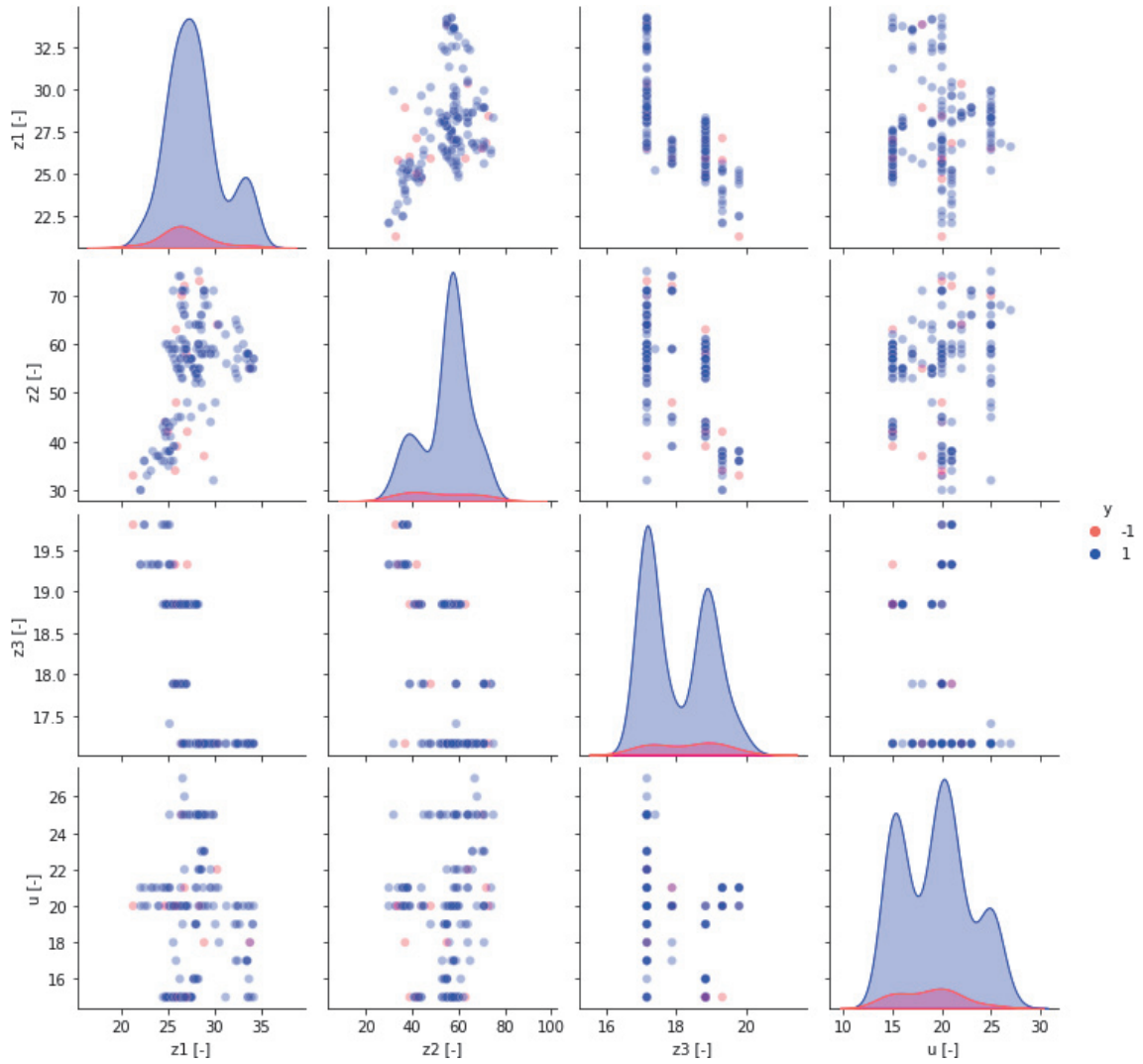


Fig. 4.13: Pair plot of initial data.

〈4.4.3〉 実装条件

実装条件を 4.3 節の数値シミュレーションのパラメータと比較し、Table 4.10 に示す。各設定値は基本的に数値シミュレーションのときと同等だが、①動作条件決定機構の SVM による分類境界を構築する際の正則化係数 C は、試行錯誤的に決定した。

Table 4.10: Fixed parameters in simulation.

	Implementation	Numerical simulation
Sampling time	$T_S = 10[\text{min}]$	$T_S = 10[\text{min}]$
Learning method	Add test data to the database	Add test data to the database
Number of initial data	$N_I = 181$	$N_I = [50, 100, 200]$
Class ratio of initial data	166: 15	[9: 1, 4: 1, 1: 1]
Range of initial data	$z_1 = [21.3, 34.2]$ $z_2 = [30.0, 75.0]$ $z_3 = [17.1, 19.8]$ $u = [15.0, 27.0]$	$z_1 = [0, 35.0]$ $z_2 = [30.0, 75.0]$ $z_3 = [15.0, 35.0]$ $u = [15.0, 40.0]$
Number of test data	$N_V = 1795$	$N_V = 200$
Range of test data	$z_1 = [9.1, 36.2]$ $z_2 = [20.8, 70.0]$ $z_3 = [5.4, 15.1]$	$z_1 = [0, 35.0]$ $z_2 = [30.0, 75.0]$ $z_3 = [15.0, 35.0]$
Threshold of similarity	$T_1 = 0.216$	$T_1 = 0.216$
Regularization coefficient	$C = 1$	$C = 0.01$
Threshold of similarity	$T_2 = 0.0016$	$T_2 = 0.0016$
Threshold of output	$T_y = 0$	$T_y = 0$

〈4.4.4〉 実装結果

実装結果として、本システム導入後のロボット塗装の1日の歩留まりの推移を Fig. 4.14 に、1日の歩留まりの平均値を Table 4.11 に示す。期間は2023年8月25日から2023年12月8日までである。比較対象として、2022年の作業者による塗装の1日の歩留まり、ロボット塗装の1日の歩留まりを示す。期間は2022年8月25日から2022年12月12日までである。2022年のロボット塗装の歩留まりにおいて、2022年10月4日から2022年10月16日までのデータのみ存在する。本システム導入後のロボット塗装の1日の歩留まりにおいて、2023年10月19日から2023年11月2日までのデータは、塗装ロボットの設備そのもののトラブルにより存在しない。Table 4.11 より、1日の歩留まりの平均値は、本システム導入後のロボット塗装が最も高いことが分かる。また Fig. 4.14 より、2022年の作業者による塗装の1日の歩留まりが減少していた時期においても、本システム導入後のロボット塗装の1日の歩留まりは減少傾向にないことが分かる。

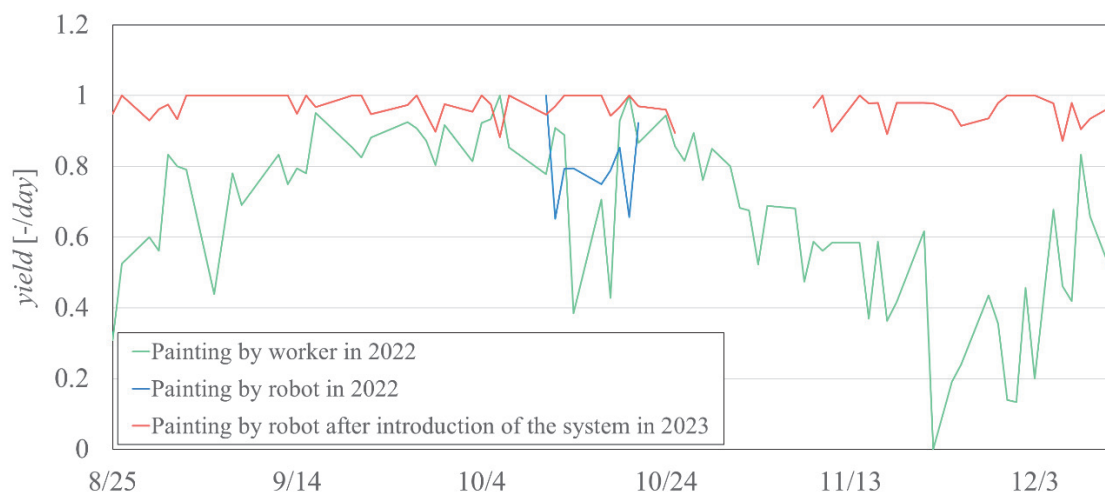


Fig. 4.14: Yield after installing this system.

Table 4.11: Average of the yield.

Painting by worker in 2022	0.66
Painting by robot in 2022	0.80
Painting by robot after installing this system	0.97

また本システム導入後のロボット塗装において、検証データの環境条件および動作条件の各要素 $z_1(t), z_2(t), z_3(t), u(t)$ の推移を Fig. 4.15 に示す。なお、Fig. 4.15 の点線は初期データベースの Table 4.10 の範囲を意味する。Fig. 4.15 より、環境条件および動作条件 $z_1(t), z_2(t), u(t)$ は連続的に変化し、環境条件 $z_3(t)$ は離散的に変化していることが分かる。また検証データの環境条件において、 $z_1(t)$ は11月10日まではデータベースの内挿、11月13日以降はデータベースの外挿、 $z_2(t)$ はデータベースの内挿、 $z_3(t)$ はデータベースの外挿となることが分かる。Fig. 4.15 において、検証データの温度 $z_1(t)$ に応じて以下のように分け、それぞれについて考察する。

1. 検証データの温度が初期データベースの温度の内挿となる（8月25日～9月28日）
2. 検証データの温度の変動が激しい（9月29日～10月18日）
3. 検証データの温度が初期データベースの温度の外挿となる（11月7日～11月14日9時25分）
4. 検証データの温度が初期データベースの温度の外挿となり、Table 4.9 のベルカップ回転数の基準値が変更となる（11月14日9時35分～12月8日）

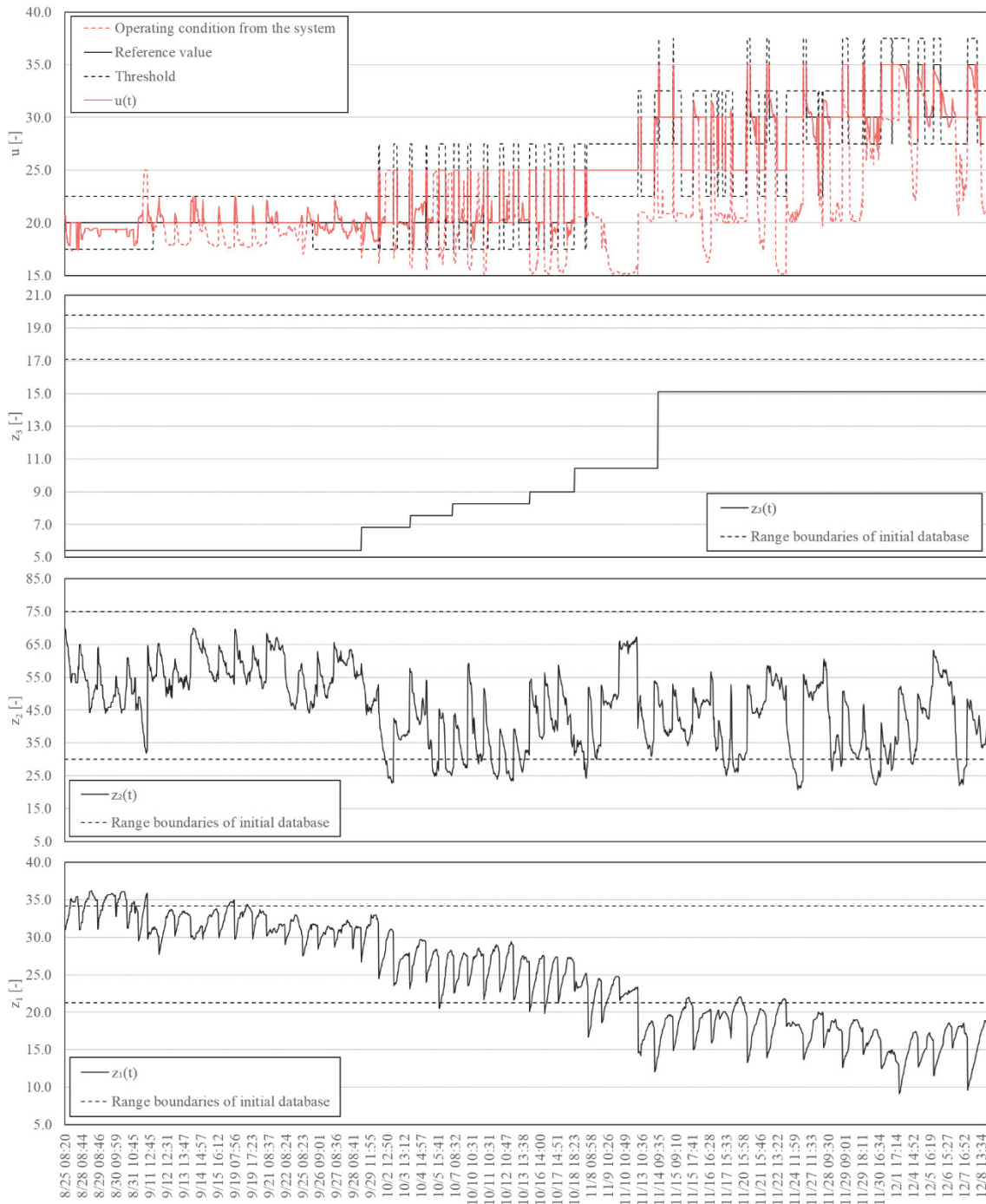


Fig. 4.15: Environmental conditions and operating condition of test data.

1. 検証データの温度が初期データベースの温度の内挿となる（8月25日～9月28日）

検証データの環境条件および動作条件の各要素 $z_1(t), z_2(t), z_3(t), u(t)$ の値を Fig. 4.16 に示す。Fig. 4.16 より、9月11日から9月25日では基準値からのしきい値を厳しくしたため $u(t)$ はしきい値を超えているが、その他の期間では $u(t)$ はしきい値内に収まっている。

ここで、環境条件に対応した動作条件の出力の様子、データベースへの検証データの追加による学習の効果を確認するため、Fig. 4.16 から8月25日および8月28日のデータを抜粋したグラフを Fig. 4.17 に示す。Fig. 4.17 より、温度 $z_1(t)$ の上昇に応じてベルカップ回転数 $u(t)$ が下降しており、熟練者の感覚と一致している。次に、8月25日の11時53分から15時05分までベルカップ回転数が一定の値を取っている。これは、【データ駆動型スマート塗装システムのフロー】の3より、本システムからのベルカップ回転数が基準値からのしきい値を超えた影響によるものである。また、8月25日10時15分のベルカップ回転数は17.8[-]であったが、同じ温度である翌日8月28日の9時45分のベルカップ回転数は18.9[-]である。これは、8月25日の検証データがデータベースへ追加されたことによると考えられる。8月25日の11時53分から15時05分では本システムからのベルカップ回転数が基準値からのしきい値を超えていたが、8月28日の10時48分から17時37分では、同様の学習により本システムからのベルカップ回転数が基準値からのしきい値を超えないようになったと考えられる。

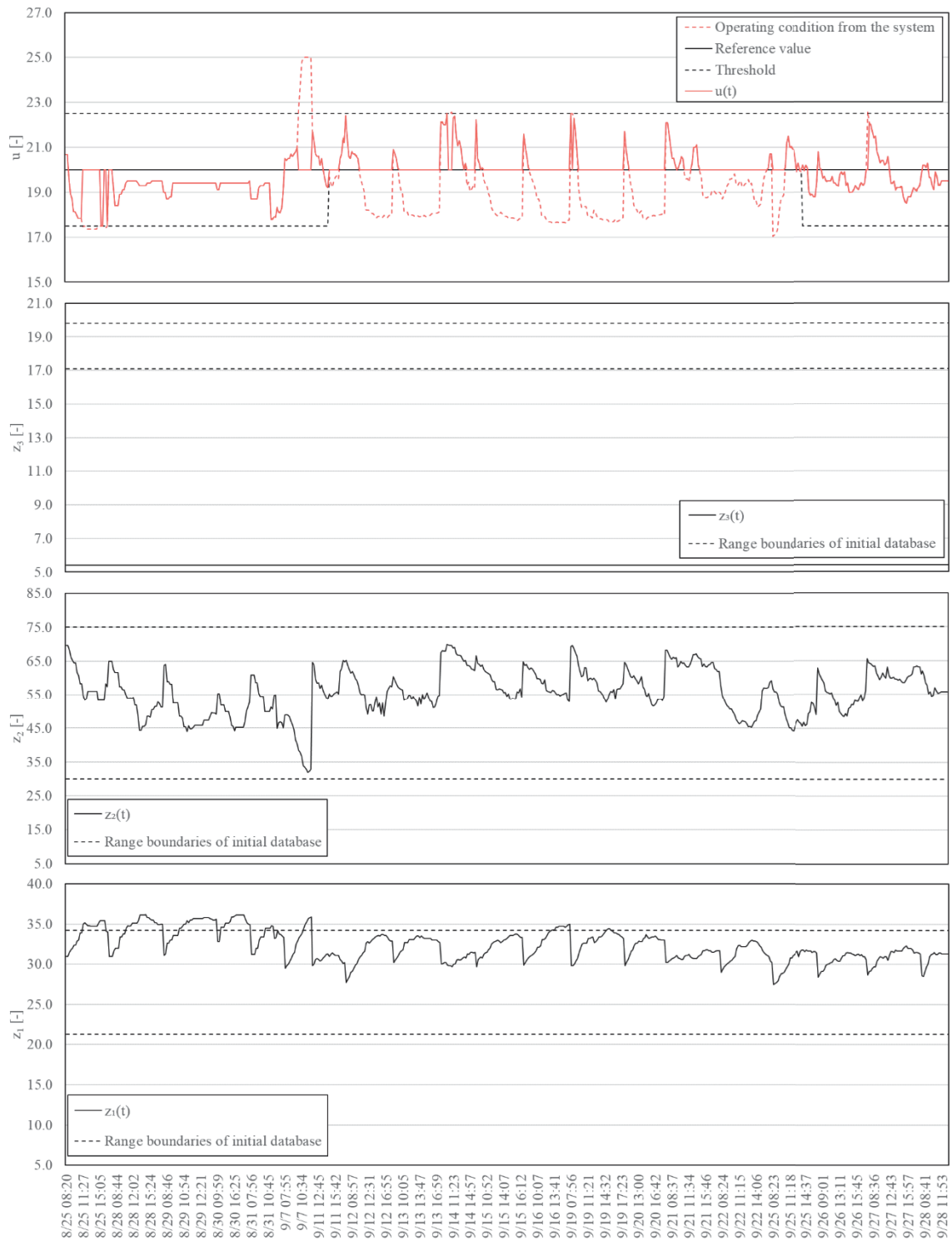


Fig. 4.16: Environmental conditions and operating condition of test data.

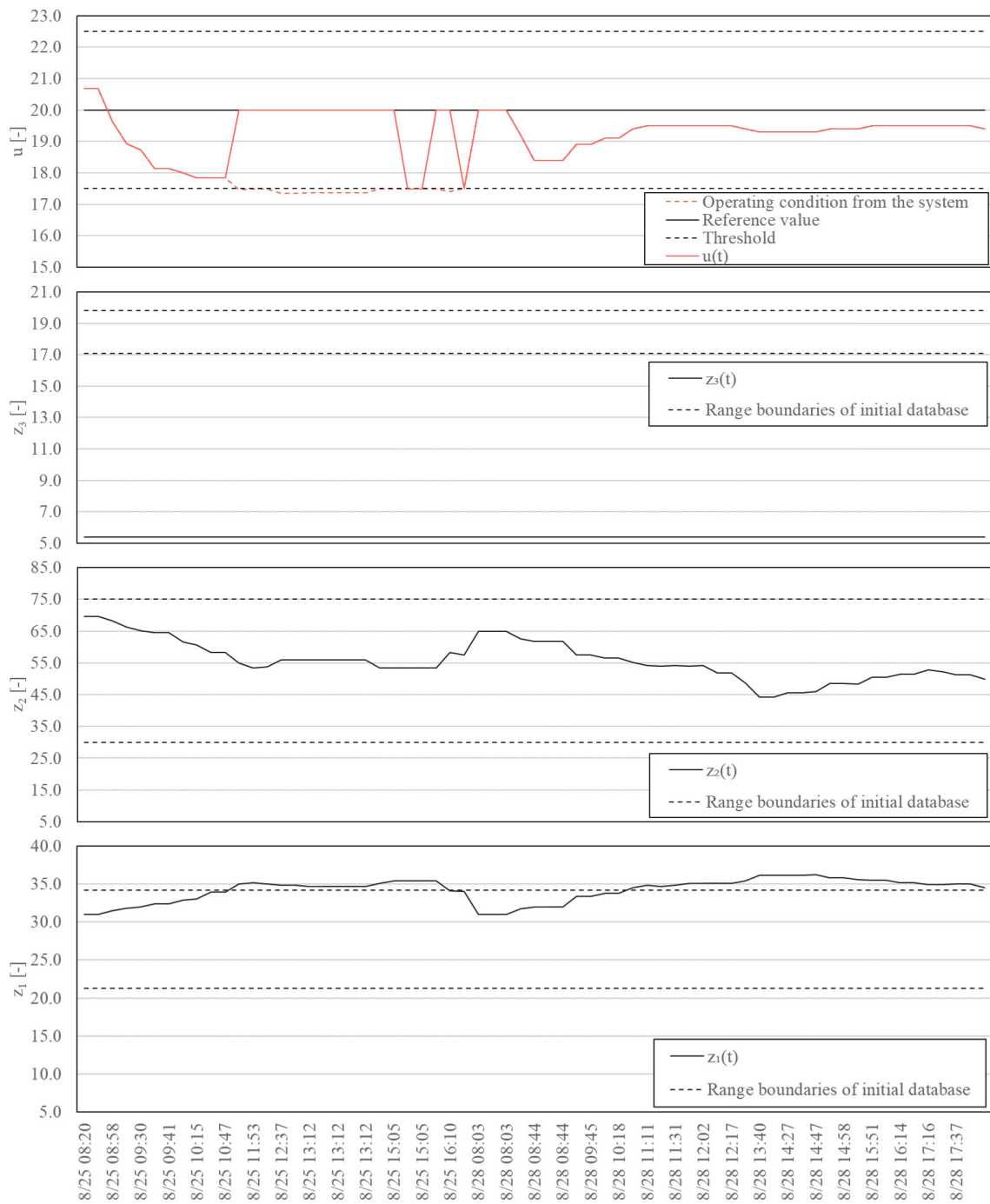


Fig. 4.17: Environmental conditions and operating condition of test data.

2. 検証データの温度の変動が激しい (9月29日～10月18日)

検証データの環境条件および動作条件の各要素 $z_1(t), z_2(t), z_3(t), u(t)$ の値を Fig. 4.18 に示す。Fig. 4.18 より、ベルカップ回転数の基準値が 20.0[-] のときは、本システムからのベルカップ回転数は基準値からのしきい値内に収まるが、ベルカップ回転数の基準値が 25.0[-] のときはしきい値を超えることが分かる。

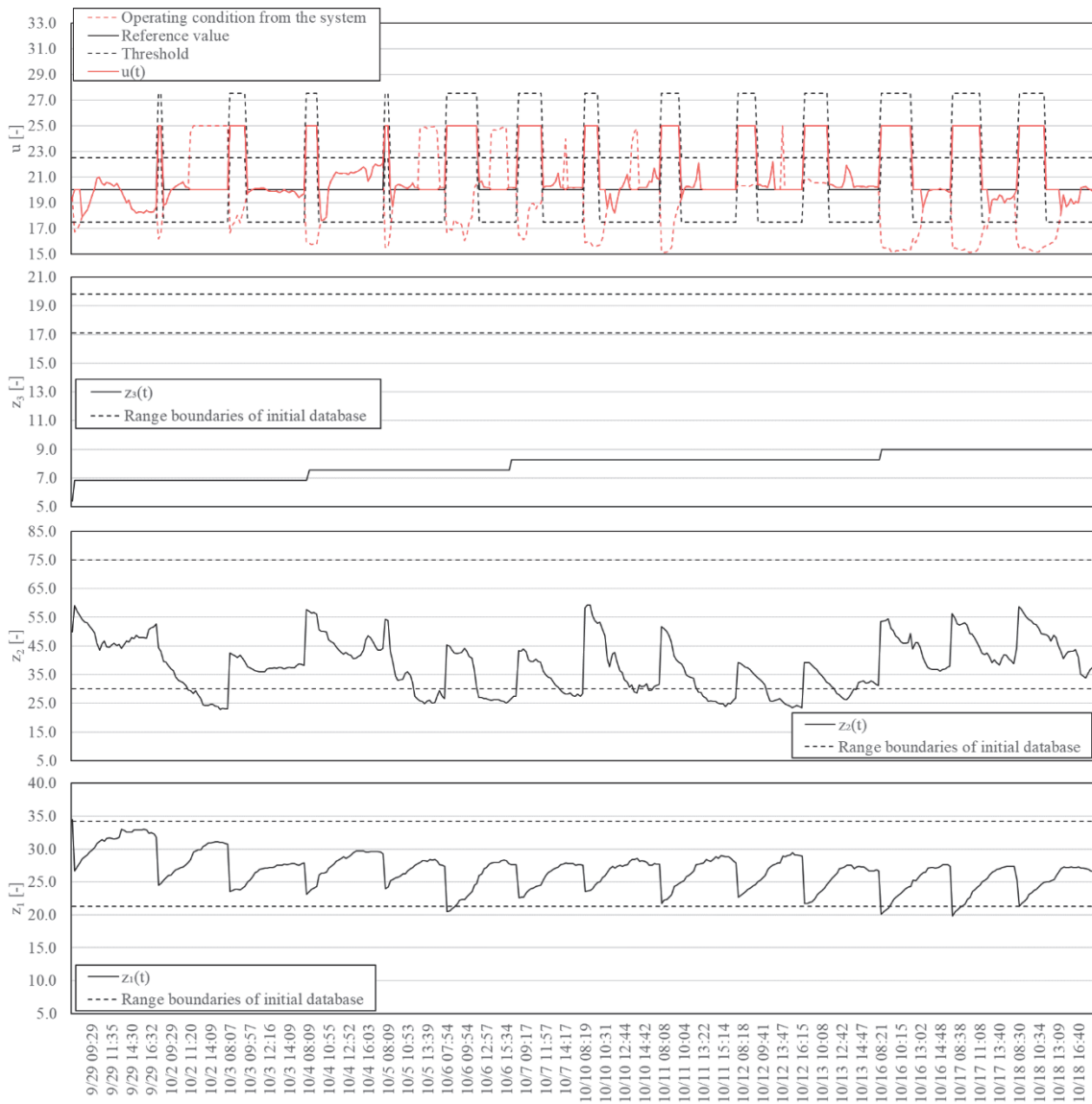


Fig. 4.18: Environmental conditions and operating condition of test data.

3. 検証データの温度が初期データベースの温度の外挿となる（11月7日～11月14日9時25分）

検証データの環境条件および動作条件の各要素 $z_1(t), z_2(t), z_3(t), u(t)$ の値を Fig. 4.19 に示す。Fig. 4.19 より、本システムからのベルカップ回転数は基準値からのしきい値を超えることが分かる。

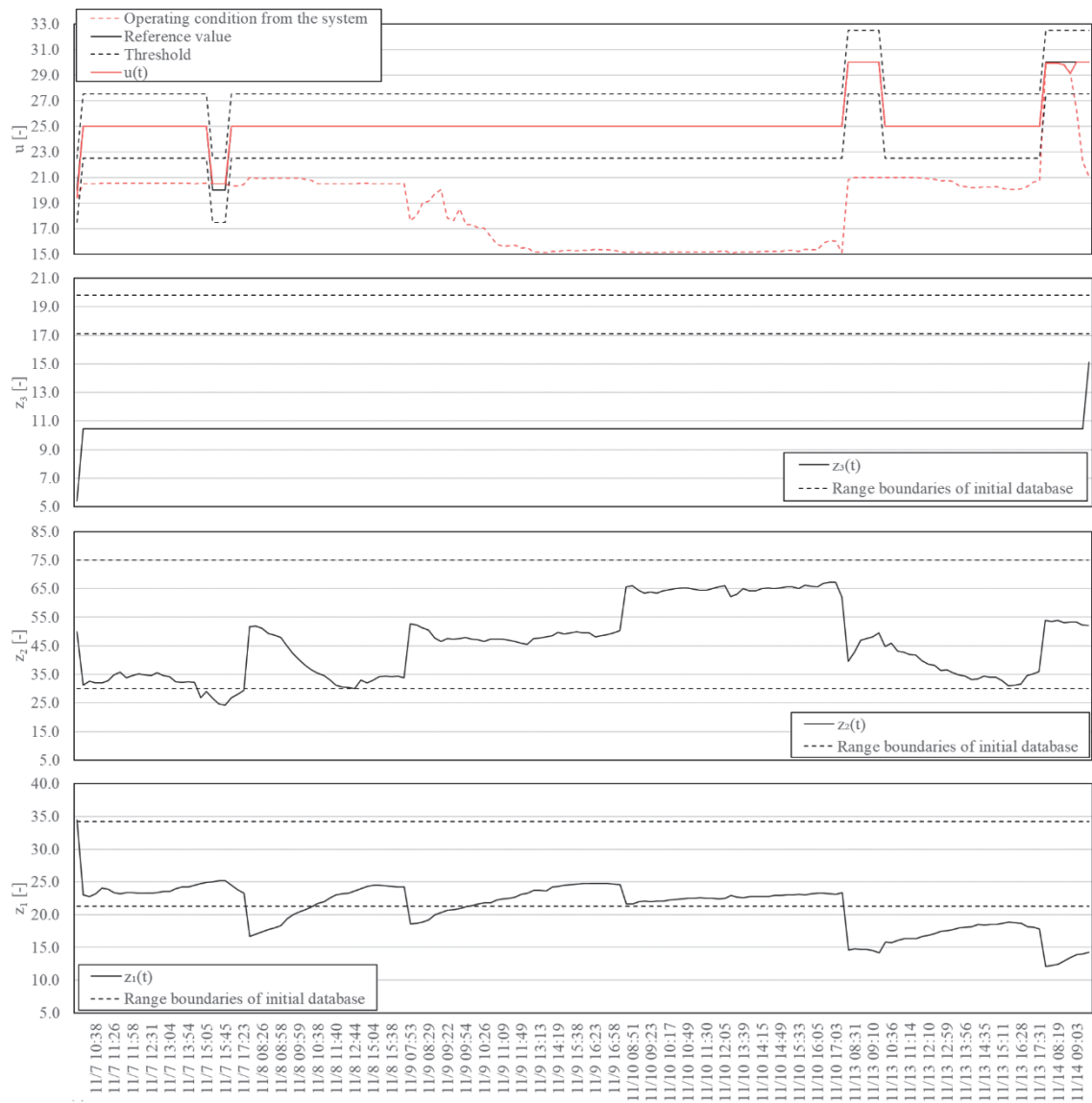


Fig. 4.19: Environmental conditions and operating condition of test data.

4. 検証データの温度が初期データベースの温度の外挿となり，Table 4.9 のベルカップ回転数の基準値が変更となる（11月14日9時35分～12月8日）

実装評価中，製造現場の要望により Table 4.9 のベルカップ回転数の基準値を Table 4.12 のように変更した．具体的には， $15 \leq z_1 < 20$ ， $7 \leq z_1 < 15$ のときのベルカップ回転数の値がそれぞれ，25 から 30 へ，30 から 35 へと変更となっている．

検証データの環境条件および動作条件の各要素 $z_1(t), z_2(t), z_3(t), u(t)$ の値を Fig. 4.20 に示す．Fig. 4.20 より，本システムからのベルカップ回転数は基準値からのしきい値を超えることが分かる．

Table 4.12: Reference value of bell cup rotation speed.

Temperature z_1 [-]	Rotation speed of the bell cup u [-]
$30 \leq z_1$	20
$25 \leq z_1 < 30$	20
$20 \leq z_1 < 25$	25
$15 \leq z_1 < 20$	30
$7 \leq z_1 < 15$	35

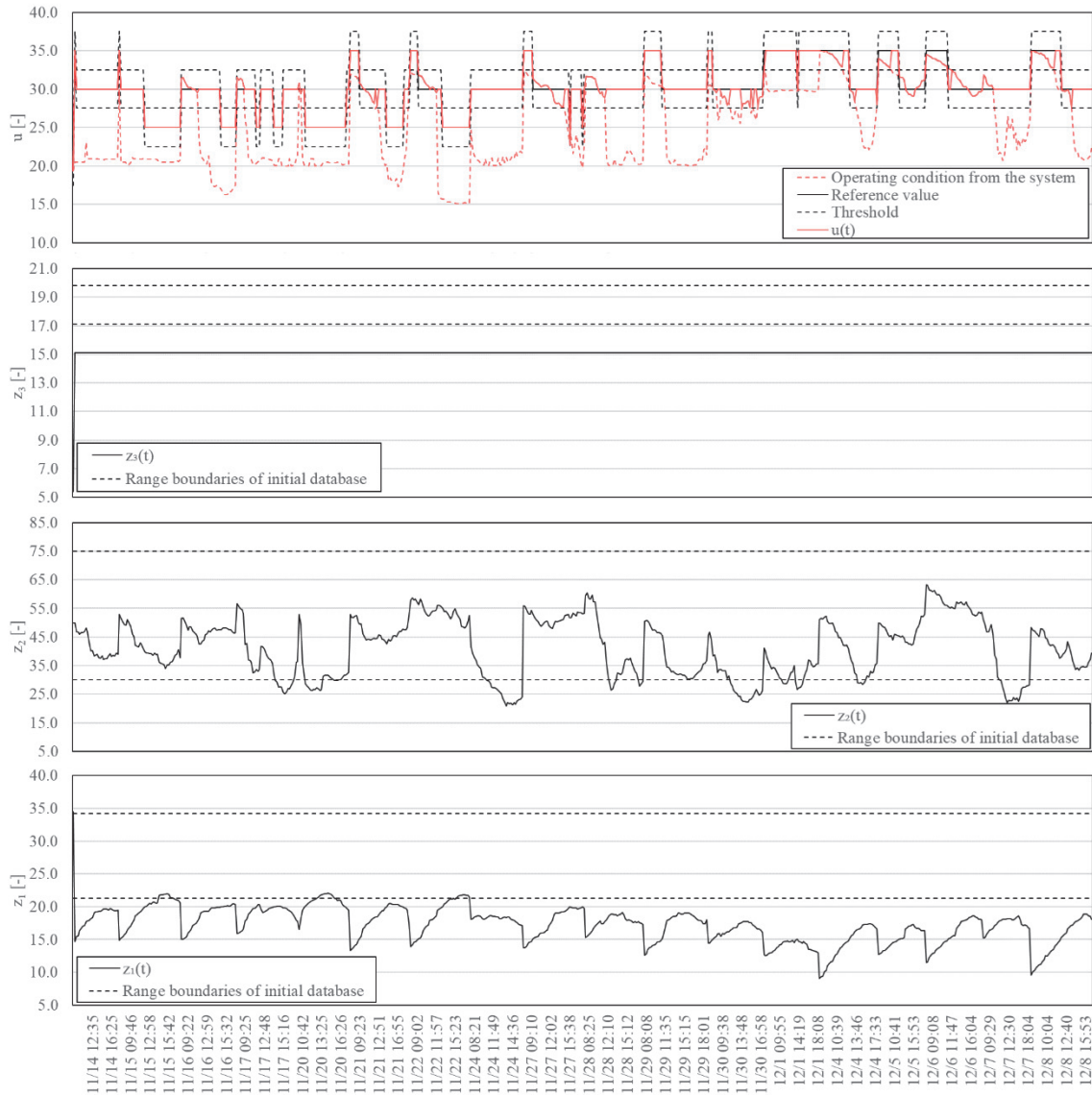


Fig. 4.20: Environmental conditions and operating condition of test data.

4.5 結言

本研究では、JIT法の考え方にに基づき、良好データおよび不良データを用いたデータ駆動型スマート塗装システムを提案した。具体的には、動作条件決定機構により決定された動作条件を基に、塗装品質予測器により出来上がりの塗装品質を予測する。予測値が-1(不良)を示す場合には、近傍データを修正し、再度動作条件決定により動作条件を決定する。

数値シミュレーションを通して、提案法における歩留まりの向上が確認できた。とくに提案法では、塗装品質予測器の予測精度の向上により、塗装品質が良好となる可能性を示した。また実環境を想定し、オンライン学習の方法やシミュレーションの設定値を変更して数値シミュレーションを実施し、適切なパラメータの値を得ることができた。

実装では、本システム適用によるロボット塗装の歩留まりが、昨年度の作業による塗装の歩留まりおよび昨年度のロボット塗装による歩留まりよりも高くなることが分かった。また、環境条件に応じて動作条件が連続的に変化する様子が再現され、定性的に熟練者の感覚と一致するものであった。また検証データを初期データベースへ追加することにより、前日では本システムからのベルカップ回転数が基準値からのしきい値を超えたような温度帯でも、ベルカップ回転数が基準値からのしきい値を超えないように修正されることが分かった。

第 5 章 結論

本研究では、塗装ロボットにおいて、環境条件および塗装品質に応じて適切な動作条件を決定するため、

①環境条件に応じて動作条件を適応的に決定する機構

②環境条件および動作条件から塗装品質を予測する機構

を備えるデータ駆動型スマート塗装システムを構築した。

第 1 章では、塗装分野の自動化に関する取り組みについて述べ、データ駆動型スマート塗装システムが備える機能を明確にした。

第 2 章では、歩留まりの向上のため、不良データを活用した JIT 法に基づく動作条件決定機構の設計法を提案し、数値シミュレーションによりその有効性を検証した。具体的には、クエリ付近の良好データおよび不良データを用い、SVM により良好・不良の分類境界を作成し、クエリにおいて品質が良好と分類される動作条件の範囲を求め、有効な良好データを決定する。その後、その良好データを用いて LWA により動作条件を決定する。数値シミュレーションを通して、提案法の歩留まりの向上が確認できた。とくに提案法では、新たな動作条件が不良データの中心から遠ざかり、これにより塗装品質が良好となる可能性を示した。

また、SVM により分類境界を作成する際の数値解析手法として、線形 SVM に特化した DCDM 法を用いることで、繰り返し計算の回数や計算時間を抑えることができ、さらに歩留まりの向上が確認できた。

第 3 章では、ロボットによる塗装作業から塗装品質の検査までの時間遅延による影響を低減するため、データベース駆動型塗装品質予測器の設計法を提案した。提案法は、塗装品

質の正解率向上のため、データベース駆動型アプローチに基づき、2 値化の際のしきい値を適応的に変更する仕組みを有する塗装品質予測器を提案した。さらに、数値シミュレーションによりその有効性を検証した。その結果、従来法よりも提案法の正解率が高く、さらに修正量の学習によりしきい値を適応的に変更することで、誤分類が削減されることを示した。

第4章では、上述の動作条件決定機構と塗装品質予測器を統合した、データ駆動型スマート塗装システムを構築した。具体的には、動作条件決定機構により決定された動作条件を基に、塗装品質予測器により出来上がりの塗装品質を予測する。予測値が-1(不良)を示す場合には、近傍データを修正し、再度動作条件決定により動作条件を決定する。

数値シミュレーションを通して、提案法における歩留まりの向上が確認できた。とくに提案法では、塗装品質予測器の正解率の向上により、塗装品質が良好となる可能性を示した。また実環境を想定し、オンライン学習の方法やシミュレーションの設定値を変更して数値シミュレーションを実施し、適切なパラメータの値を得ることができた。

最後に、実際に量産で使用されている塗装ロボットに提案システムを実装し、その有用性を評価した。実装では、本システム適用によるロボット塗装の歩留まりが、昨年度の作業者による塗装の歩留まりおよび昨年度のロボット塗装による歩留まりよりも高くなることが分かった。また、環境条件に応じて動作条件が連続的に変化する様子が再現され、定性的に熟練者の感覚と一致するものであった。また検証データを初期データベースへ追加することにより、前日では本システムからのベルカップ回転数が基準値からのしきい値を超えたような温度帯でも、ベルカップ回転数が基準値からのしきい値を超えないように修正されることが分かった。

今後の展望として、本研究のデータ駆動型スマート塗装システムに対し、今後取り組む予定の項目を示す。

1. 品質評価結果の入力デジタル化
2. 品質評価の自動化
3. ②塗装品質予測器によるデータベースの修正量の活用によるデータ圧縮
4. データベースへの機種情報の格納

1. 品質評価結果の入力デジタル化

現在の生産ラインでは、紙に記載された品質評価結果を1日分まとめて入力しているため、その日の塗装不具合の情報をリアルタイムに反映することができない。これにより、その日の塗装不具合を回避するための動作条件が決定できないと考えられる。この対策として、例えばタブレット入力により1台ごとに塗装不具合の情報をリアルタイムにフィードバックできれば課題を解決できると考える。

2. 品質評価の自動化

現在の生産ラインでは、塗装不具合の有無は作業者が目視で判定しており、さらにその結果も良好か不良かの2値で判定しているため、作業者や作業者の体調などにより塗装不具合の有無の判定結果がばらつき、これが原因で適切な動作条件が出力されないことが想定される。この対策として、品質評価装置の導入により、ロボットが塗装不具合の有無を判定することで、判定結果のばらつきを抑え、適切な動作条件の出力につながると考える。さらにロボットから定量的な値を出力することで、塗装不具合に対するデータ分析を詳細に行うことができると考える。

3. ②塗装品質予測器によるデータベースの修正量の活用によるデータ圧縮

本研究では、検査結果の学習方法として、データベースへ検証データを追加することとしている。現在データベースのデータの削除のルールを決定していないため、データベースのデータ数が多くなっている。これにより、本システムにより動作条件の計算をおこなうにあたり、特に近傍データの抽出をおこなう際に計算負荷が高くなると考えられる。

本研究の第 3 章で述べたように修正量を活用することで、データベースのデータ数を増やすことなく、②塗装品質予測器の予測精度を保つことができる。ただ検査結果の学習方法としては、データベースへ検証データを追加することが良いと第 4 章で示している。今後の学習方法として、「生産ラインが稼働しているときは検証データをデータベースへ追加し、その日の生産ラインの稼働が終了したあとはその日の検証データを用いてデータベースの修正量を学習する」ことで、本システムによる歩留まりを保ちつつ、データベースのデータ数の抑制ができると考える。

4. データベースへの機種情報の格納

本研究で使用したカウンタウェイトは、機種によりその形状が異なり、塗装不具合の有無に影響することが考えられる。そのため、将来的に機種ごとに適切なベルカップ回転数を設定することで塗装不具合の削減につながると考えられる。

付録

1. (2.27)式の各 i における $\alpha_i y_i$, $\boldsymbol{\psi}_i = [z_{1i}, z_{2i}, z_{3i}, u_i]$ の値

$$f(\boldsymbol{\psi}_f(t)) = \sum_{i \in [S(t)]} \alpha_i y_i K(\boldsymbol{\psi}_i, \boldsymbol{\psi}_f(t)) + b$$

$$K(\boldsymbol{\psi}_i, \boldsymbol{\psi}_f(t)) = \exp(-\gamma \|\boldsymbol{\psi}_i - \boldsymbol{\psi}_f(t)\|^2)$$

$$y(t) = \begin{cases} -1, & f(\boldsymbol{\psi}_f(t)) \geq 0 \\ 1, & f(\boldsymbol{\psi}_f(t)) < 0 \end{cases} \quad (2.27)$$

ただし, $S(t) = 90$, $b = -0.99343$, $\gamma = 0.1$ である.

i	$\alpha_i y_i$	z_{1i}	z_{2i}	z_{3i}	z_{4i}
1	-1.446390000	24.8	38	33	21
2	-8.858240000	26.7	71	29.8	20
3	-1.244920000	16.8	43	27	40
4	-0.005014490	21.8	35	21.6	30
5	-2.574690000	25.6	44	29.8	20
6	-0.019598700	27.8	53	16.2	25
7	-0.006567040	16.1	38	21.6	30
8	-10.231600000	26.4	68	28.6	25
9	-13.421200000	23.2	34	32.2	21
10	-0.000703105	20.7	37	21.6	30

i	$\alpha_i \psi_i$	z_{1i}	z_{2i}	z_{3i}	z_{4i}
11	-2.660970000	23.4	38	32.2	21
12	-0.040551400	24.5	64	16.2	20
13	-0.004438100	23.2	67	18	25
14	-0.006452200	22.4	89	18	25
15	-4.307290000	25.6	39	29.8	20
16	-4.935690000	26.4	74	29.8	21
17	-0.004315990	8.5	51	64.8	30
18	-14.663200000	33.8	56	28.6	18
19	-0.007272660	27.1	48	28.6	25
20	-0.010383500	29.9	32	28.6	25
21	-0.006765080	13.2	56	27	30
22	-10.957900000	30.4	64	28.6	21
23	-8.074780000	33.7	55	28.6	16
24	-0.814254000	22.1	30	32.2	20
25	-27.461100000	24.1	37	32.2	20
26	-0.206499000	28.6	66	28.6	23
27	-3.677650000	10.4	54	27	40
28	-0.006878280	23.2	40	19.8	25
29	-24.878300000	25.3	36	32.2	20
30	-0.005947640	12	47	64.8	30
31	-0.126256000	34.1	55	28.6	19
32	-23.135000000	26.6	71	28.6	25

i	$\alpha_i \psi_i$	z_{1i}	z_{2i}	z_{3i}	z_{4i}
33	-0.000927442	4.3	50	64.8	30
34	-0.116045000	26.5	53	31.4	15
35	-0.006938380	17.9	45	25.2	30
36	-2.612690000	28.6	63	28.6	22
37	-0.006198740	23.4	76	18	23.4
38	-1.836330000	28.9	70	28.6	23
39	-0.027412800	29.5	52	16.2	20
40	-0.000636896	28	54	31.4	19
41	-0.005876720	15	50	27	30
42	-1.904310000	26.6	61	31.4	19
43	-0.335003000	28.3	64	28.6	25
44	-0.065062900	29.9	59	28.6	25
45	-0.003786430	28.9	50	17.4	15.8
46	-0.945716000	3.2	32	64.8	30
47	-0.002081900	20.5	37	21.6	30
48	-0.018157300	24.2	58	16.8	25
49	-1.295110000	17.2	45	27	40
50	-6.185030000	26.3	61	31.4	16
51	-0.005842900	3.5	49	64.8	30
52	-0.006071200	18.1	44	25.2	20
53	-0.002359560	10.5	51	64.8	30
54	-8.531170000	24.1	58	16.8	20

i	$\alpha_i y_i$	z_{1i}	z_{2i}	z_{3i}	z_{4i}
55	-27.461100000	15.1	42	27	40
56	-0.005604050	23	71	18	25
57	-5.787450000	25.1	35	32.2	20
58	-5.556840000	32.5	53	28.6	17
59	-27.461100000	24.6	36	33	21
60	-15.721500000	24.4	36	33	21
61	-0.006676110	22.1	62	21.6	30
62	-0.005418250	12	27	64.8	30
63	-0.005493070	30	46	16.2	20
64	-3.899960000	16.8	42	27	40
65	-0.251883000	24.8	41	31.4	15
66	-0.012979400	10.3	32	64.8	30
67	-0.035089100	28	52	28.6	25
68	-0.455153000	28	54	31.4	19
69	-20.456100000	34.1	55	28.6	19
70	-0.007870670	24.8	44	31.4	15
71	-0.065814300	24.8	44	31.4	15
72	1.643400000	25.9	63	31.4	15
73	4.804130000	9.1	53	27	40
74	1.344610000	20.2	47	25.2	40
75	2.167760000	26.5	51	29.8	19
76	51.896500000	24.6	35	33	21

i	$\alpha_i \psi_i$	z_{1i}	z_{2i}	z_{3i}	z_{4i}
77	2.691810000	28.9	37	28.6	18
78	32.946100000	15.1	42	27	40
79	0.979432000	20.1	43	25.2	40
80	5.324050000	21.3	33	33	20
81	41.976100000	33.8	55	28.6	18
82	65.705100000	24.5	37	32.2	20
83	2.368510000	3.7	35	64.8	30
84	7.353950000	26.5	62	31.4	17
85	13.515600000	26.8	72	29.8	21
86	9.585210000	25	58	16.2	20
87	14.022900000	30.3	64	28.6	22
88	3.975020000	24.7	44	31.4	20
89	29.627100000	26.5	70	28.6	25
90	2.982460000	18.2	44	25.2	40

参考文献

- [1] 経済産業省：「2023年版ものづくり白書（全体版）」,
<https://www.meti.go.jp/report/whitepaper/mono/2023/pdf/all.pdf>, (2023年11月19日)
- [2] 経済産業省：「スマートファクトリーロードマップ」,
https://www.chubu.meti.go.jp/b21jisedai/report/smart_factory_roadmap/roadmap.pdf, (2022年11月21日)
- [3] 平野克己：「塗装用ロボットの現状と今後の動向」, 塗装技術, Vol.57, No.6, pp.57-61 (2018)
- [4] 舘 和幸：「塗料の微粒化と塗膜の仕上がり品質」, 色材協会誌, Vol.60, No.10, pp.527-535 (1987)
- [5] T. C. Patton : "Paint Flow and Pigment Dispersion", p. 112, Interscience Publisher (1964)
- [6] オーウェル株式会社：「塗装トラブルと対策【ゆず膚】」
<https://www.owell.co.jp/ya06>, (2023年11月19日)
- [7] T. Mejdell and S. Skogestad: "Estimation of distillation compositions from multiple temperature measurements using partial-least-squares regression", Industrial & Engineering Chemistry Research, Vol. 30, pp. 2543-2555 (1991)
- [8] M. Ohshima and M. Tanigaki: "Quality control of polymer production processes", Journal of Process Control, Vol. 10, No. 2-3, pp. 135-148 (2000)
- [9] A. Stenman, F. Gustafsson and L. Ljung: "Just in time models for dynamical systems", Proceedings of 35th IEEE Conference on Decision and Control, pp. 1115-1120 (1996)

- [10]K. Fujiwara, M. Kano and S. Hasebe: “Correlation-based Just-In-Time Modeling for Softsensor Design”, Transactions of the Society of Instrument and Control Engineers, Vol. 44, pp. 317-324 (2008) (in Japanese)
藤原幸一・加納学・長谷部伸治：「相関型 Just-In-Time モデリングによるソフトセンサの設計」, 計測自動制御学会論文集, Vol. 44, No. 4, pp. 317-324 (2008)
- [11]H. Shigemori, M. Kano and S. Hasebe: “Optimum quality design system for steel products through locally weighted regression model”, Journal of Process Control, Vol. 21, No. 2, pp. 293-301 (2011)
- [12]中林暁男, 和田英彦：「第3次人工知能ブームを機に見る製造業におけるデータ解析技術の変遷」, 横河技報, Vol. 59, No.1 (2016)
- [13]竹内一郎・鳥山昌幸：「サポートベクトルマシン」, 講談社 (2018)
- [14]C.M.ビショップ(元田ほか(監訳))：「パターン認識と機械学習 下 ベイズ理論による統計的予測」, 丸善出版 (2008)
- [15]赤穂昭太郎：「カーネル多変量解析」, 岩波書店(2008)
- [16]T. Ide: Introduction to Anomaly Detection using Machine Learning, Coronasha (2015)
井手剛：「入門 機械学習による異常検知」, コロナ社 (2015)
- [17]T. Kinoshita and T. Yamamoto: “Design of a database-driven control system based on the similarity”, IEEJ Transactions on Electronics, Information and Systems, Vol.140, No.3, pp.312-319 (2019) (in Japanese)
木下拓矢・山本透：「類似度に基づくデータベース駆動型制御系の一設計」, 電気学会論文誌 C(電子・情報・システム部門誌), Vol.140, No.3, pp.312-319 (2019)
- [18]C. R. Loader: “Bandwidth selection: classical or plug-in?”, The Annals of Statistics, Vol.27, No.2, pp.415-438 (1999)
- [19]C. J. Hsieh, K. W. Chang, C.J. Lin, S. S. Keerthi, and S. Sundararajan: “A Dual Coordinate

Descent Method for Large-scale Linear SVM”, Proceedings of the 25th international conference on machine learning, pp. 408-415 (2007)

[20]H. Kano, T. Kinoshita and T. Yamamoto: “Design of a Database-Driven Soft Sensor”, Proceedings of the SICE Annual Conference, pp. 1141-1144 (2021)

[21]T. Kinoshita, H. Kano and T. Yamamoto: “Design of a control system using a database-driven soft-sensor”, 2021 IEE-Japan Industry Applications Society Conference, pp. 49-52 (2021) (in Japanese)

木下拓矢・加納晴香・山本透：「データベース駆動型ソフトセンサを用いたフィードバック制御系の一設計」, 2021年電気学会産業応用部門大会, pp. 49-52 (2021)

引用既発表論文

A. 学術雑誌等掲載論文

- [1] 山本翔平, 木下拓矢, 脇谷伸, 山本透, 岡保伸, 山下俊治, "塗装プロセスにおけるデータベース駆動型塗装品質予測器の一設計", 電気学会論文誌 C (電子・情報・システム部門誌), 第 143 巻, 第 3 号, pp.216-221, 2023
- [2] 山本翔平, 山本透, 松田浩孝, "塗装プロセスにおける Just-in-Time 法に基づく動作条件決定機構の一設計", 電気学会論文誌 C (電子・情報・システム部門誌), 第 143 巻, 第 7 号, pp.680-685, 2023

B. 国際学会講演論文

- [1] Shohei Yamamoto, Takuya Kinoshita, Shin Wakitani, Toru Yamamoto, Hiroataka Matsuda, Yasunobu Oka, Susumu Matsunaga, Toshiharu Yamashita, Design of a Database Driven Quality Predictor for Painting Systems, 2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP), Vancouver, BC, Canada pp.194-197, 2022

その他既発表論文

C. 国際学会講演論文

- [1] Shohei Yamamoto, Takuya Kinoshita, Shin Wakitani, Toru Yamamoto, Hirotaka Matsuda, Yasunobu Oka, Susumu Matsunaga, Toshiharu Yamashita, Design of a Database Driven Quality Predictor for Painting Systems, 2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP), Vancouver, BC, Canada pp.194-197, 2022

謝辞

本論文の作成にあたり、多くの方々からご支援とご協力をいただきました。まず初めに、指導教員である広島大学先進理工系科学研究科の山本透教授に心より深い感謝を申し上げます。私がコベルコ建機株式会社から広島大学に出向している時から、専門的なアドバイスや温かいお言葉をいただきました。研究テーマやその方針が決まっていない状態からのスタートでしたが、山本透教授に親身に接して頂き、さらに深くご指導ご鞭撻いただいたため、研究テーマが決定でき、無事研究を完了することができました。

当初の指導教員である大阪公立大学情報学研究科の大野修一教授、広島大学デジタルものづくり教育研究センターの大松繁特任教授、ならびに広島大学先進理工系科学研究科の脇谷伸准教授、木下拓矢准教授にも心より感謝を申し上げます。研究の遂行にあたり、基本的な知識や専門的な知識を丁寧にご教示いただき、研究の基礎を築くことができました。

さらに、広島大学システム制御論研究室のメンバーや、広島大学デジタルものづくり教育研究センターのメンバー、コベルコ建機株式会社のメンバーにも心より感謝を申し上げます。学術的な視点、企業的な視点から有益な意見をいただき、学術的でありながらも企業の課題にマッチした成果とすることができました。

さらに、データ提供や実装機器に関しては、出向元であるコベルコ建機株式会社の多大な協力を得ました。実装機器として量産設備を使用することを快諾いただき、実装評価が円滑に進みました。

最後に、この論文が完成するまで支えてくださった私の家族や友人、同僚の皆さまに深く感謝いたします。皆様のサポートがあつてこそ、私の研究が円滑に進み成果を上げることができました。

心より感謝いたします。

コベルコ建機株式会社

山本 翔平