

論文 / 著書情報
Article / Book Information

題目(和文)	構造化ニューラルネットワークによる化学プロセス制御
Title(English)	
著者(和文)	大庭武泰
Author(English)	takehiro ohba
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第4127号, 授与年月日:1999年3月26日, 学位の種別:課程博士, 審査員:石田 愈
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:甲第4127号, Conferred date:1999/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

学位論文

構造化ニューラルネットワークによる

化学プロセス制御

1999年1月

東京工業大学 総合理工学研究科

化学環境工学専攻

大庭 武泰

謝辞

本研究をまとめるにあたり、懇切丁寧で的を得た御指導を賜りました東京工業大学資源化学研究所教授 石田 愈先生に感謝いたします。また東京工業大学資源化学研究所助教授 鈴木 孝弘先生の堅実な御指導に感謝いたします。さらに、金紅光助手、山本昌志助手をはじめとする石田・鈴木研究室の皆様にもお礼を申し上げます。

本研究を行うにあたっては日本学術振興会から特別研究員としての支援に感謝いたします。また本研究は文部省科学研究費補助金による研究成果であります。

目次

1.	本研究の目的と既往の研究	1
1.1.	本研究の目的	1
1.2.	論文の構成	2
1.3.	既往の研究	5
1.3.1.	ニューラルネットワーク	5
1.3.2.	ニューラルネットコントローラ	7
1.3.3.	初期 PENN 法	11
2.	PENN コントローラの提案	12
2.1.	PENN コントローラの制御及び学習方法	12
2.1.1.	制御方法	12
2.1.2.	経験学習	15
2.1.3.	ポリシー学習	16
2.2.	PENN コントローラの特徴	18
2.2.1.	速度型 PI コントローラとの比較	19
2.2.2.	PENN コントローラの実出力	20
2.2.3.	時間遅れ対応パラメータ k の選択	22
2.3.	第 2 章の結論	24
3.	構造化 PENN コントローラの提案	24
3.1.	構造化 PENN の構造	24
3.2.	学習方法	26
3.2.1.	経験学習	26
3.2.2.	ポリシー学習	26
3.3.	制御手順	29
3.4.	ノイズへの対応	29
3.5.	PENN コントローラへの遅れの影響	31
3.5.1.	SISO プロセス	31

3.5.2.MIMO プロセス	34
3.6. 学習結果のネットワークからのプロセスゲインの推定	39
3.7. 行列式を利用したネットワークの状態把握	46
3.8. プロセス特性変化への対応	47
3.9. 第3章の結論	50
4. 結晶成長プロセスへの構造化 PENN コントローラの応用	51
4.1. 結晶成長プロセスとそのモデル	51
4.2. 制御方法と結果	54
4.3. ノイズの考慮	60
4.4. ゲインの推定	61
4.5. 第4章の結論	62
5. 重油加熱炉プロセスへの構造化 PENN コントローラの応用	63
5.1. 重油加熱炉プロセスとそのモデル	63
5.2. 制御方法と結果	69
5.3. ノイズの考慮	75
5.4. ゲインの推定	79
5.5. 行列式によるネットワークの状態把握	80
5.6. 特性変化を伴うプロセス	80
5.6.1.燃料組成の切替	81
5.6.2.燃料組成の周期的変化	89
5.6.3.空気温度の周期的変化	91
5.7. 第5章の結論	94
6. 結論	95
NOMENCLATURE	96
REFERENCES	98

1. 本研究の目的と既往の研究

1.1. 本研究の目的

化学プロセスを制御するにはその特徴を把握する必要がある。まず化学プロセスは非線形特性を有し、多入力多出力系であるという特徴を持っている。また、運転中にプロセスの特性が変化することがあり、運転員が介入する必要がある。さらに、プロセス内部の情報を得るには高価な測定器が必要であったり、そもそも測定できない場合があり、または多くの情報を同時に得るためには多数の測定器を導入が必要で、コスト的に不可能となる場合がある。そのために制御に利用できる情報が限定されてくる。また、化学反応自体が複雑で、しかも反応物の供給や生成物の移動といった拡散などの物質移動や熱移動の影響が加わり、数式によるモデル化が難しく、十分な精度が期待できないことが多い。

すなわち化学プロセスを制御するコントローラには、非線形処理能力、多変数処理能力、プロセス特性の把握機能が求められる。これらの能力を併せ持つ一つのツールがニューラルネットワークである。

ニューラルネットワーク（人工神経網）とは生物の脳細胞で行われる情報処理を模倣したものである。ニューラルネットワークの特徴は非線形処理、並列分散処理、学習、推定で言い表される。非線形処理を可能にしているのは、ネットワークの演算の際に非線形関数を使用することによる。並列分散処理は、ニューラルネットが複数のニューロンユニットとそれらを結合する重みで形成されていることで行われる。複数の入出力を同時に演算可能で、複数の結合重みによって分散された形で学習が行われる。学習機能は与えた学習データを満足するよう結合重みを学習アルゴリズムによって自動調節することである。そして推定は、すでに学習済みのデータとは異なる入力を与えられた場合に非線形関数での内挿及び外挿処理によって行われる。

このニューラルネットワークを化学プロセスの制御に用いる場合には、非線形学習機能を利用してプロセスの特性を自動的に把握し、同時に複数の入出力を扱うことが可能となる。しかし従来のニューロコントローラには欠点がある。学習した結

果は複数の結合重みとして分散して表されるので、なにを学習したのかがわからない。また推定機能のうちの外挿計算によって思いも寄らぬ値が出力される可能性がある。さらにニューロコントローラは測定値の誤差をも学習してしまい不安定な状態になるおそれがある。そのためニューロコントローラを適用するのは危険であると認識されて実用化されにくい状況にある。

そこで本論文の目的を次のように設定した。学習の完了したニューラルネットワークから化学プロセスの特性を簡単な演算によって取り出すことができるコントローラを提案する。そして、このコントローラを測定値にノイズが含まれる場合に適用できるようにノイズ対策を提案する。さらに、プロセス内のループ毎に遅れ特性が異なる対象に適用する。最後に、プロセス特性が経時的に変化する対象への適用も検討する。

1.2. 論文の構成

本論文の構成を図 1-1 に示す。第 2 章では SISO ニューロコントローラである PENN(Policy-and Experience-driven Neural Network)コントローラを提案し、PENN 法の基本的な説明を行う。また、PENN コントローラと PID コントローラとの関係を示すことでネットワーク重みの意味づけを行う。

第 3 章では MIMO プロセス用のニューロコントローラである構造化 PENN コントローラを提案する。第 2 章で提案した SISO プロセス用の PENN を操作量と制御量の組み合わせの数だけ接続して 1 つのネットワークを構成する。そしてプロセス出力にノイズが含まれる場合の対策方法、制御で得られるネットワークからの学習内容の取り出し方法、およびプロセス内のループ毎に遅れが異なる場合への対応を説明する。また、3 入出力の線型プロセスを例にとって学習内容の取り出しを提案する。幅広い制御対象範囲に対応するために、ノイズ対策とプロセス特性の変化へ対応する学習アルゴリズムを提案する。

第 4 章では構造化 PENN コントローラを、5 入力 5 出力の結晶成長プロセスに応用する。ノイズ対策を適用し、提案した対策が有効であることを示す。重油加熱炉はループ毎に遅れが異なるプロセスであるが、構造化 PENN コントローラにより良好に制御することを示す。

第 5 章では、構造化 PENN コントローラを重油加熱炉制御に適用する。このプロセスは、制御量が出口重油温度と排気ガス中の酸素濃度であるため変数対毎の応答が異なるプロセスである。第 3 章で提案した遅れに対応したネットワーク構造を用い、ノイズ対策およびネットワークの状態把握法を適用する。また、制御対象の特性が時間とともに変化する場合について、構造化 PENN コントローラの適用法とネットワーク重みの変化について考察する。重油加熱炉の燃料が切り替わる場合と、供給空気温度が周期的に変化する場合についてネットワークの変化を把握し、プロセス特性の変化に伴ってネットワーク重みが変わることによって制御が行われていることを確認する。燃料が切り替わる場合には切替の情報をうまく利用することで良好に制御を行う。

第 6 章では本研究の成果として可能になった主要事項をまとめた。

第 1 章：本研究の目的と既往の研究
第 2 章：PENN コントローラの提案
第 3 章：構造化 PENN コントローラの提案
第 4 章：構造化 PENN コントローラの結晶プロセスへの応用
第 5 章：構造化 PENN コントローラの重油加熱炉への応用
第 6 章：結論

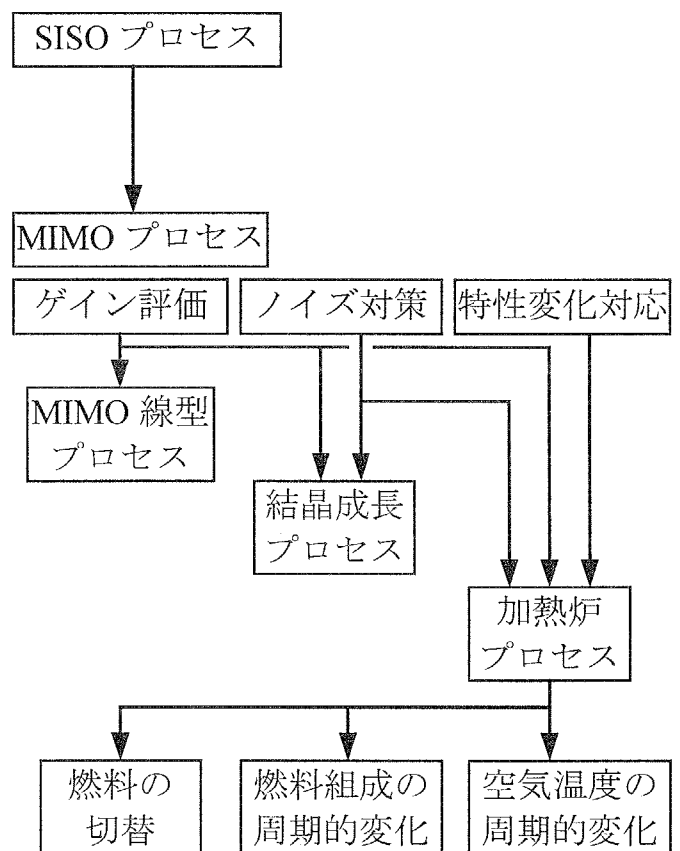


図 1-1 本論文の構成

1.3. 既往の研究

既往の研究として、ニューラルネットワーク、ニューロコントローラ、そして PENN 法を説明する。

1.3.1. ニューラルネットワーク

ニューラルネットワーク(NN)は1980年代になってラメルハートらによる研究で層状ネットにおけるバックプロパゲーション学習法が再認識され、その非線型関数近似能力を利用した研究が各方面でなされている。例として画像(文字)認識、音声認識、制御といった研究が行われている。また、活性化関数を基底関数にした RBF ネットワークや各ユニットを相互結合させたホップフィールドネット等が提案されているが、入力に対する写像を実現する方法といえる。ニューラルネットワークの特徴としては、(1)非線型関数の近似、(2)学習による最適化、(3)多変数並列処理、が挙げられるがこれらはいずれもプロセス制御には有効な特徴である。

この研究では、階層型ネットワークでバックプロパゲーション学習法による NN を用いる。図 1-2 にその構造を、以下に出力及び学習方法を示す。

・出力方法

入力データは入力層と呼ばれる、演算を行わないユニットに入力され(式 1-1)、中間層のユニットへ重みを付けられ送られる。中間層のユニット内では重み付けされたデータを加え、活性化関数と呼ばれる関数へ入力し、得られたデータを出力層に送る(式 1-2)。出力層でもまた、中間層のユニットと同じ演算が行われる。

$$O_i = I_i \quad (1-1)$$

$$S_j = \sum_i O_i w_{ji} + \theta_j \quad O_j = f(S_j) \quad (1-2)$$

θ はしきい値と呼ばれ、入力へのバイアスとして使用される。しきい値は常に 1 を出力するユニットからの結合重みと考えることができ、他の重みと同様の学習アルゴリズムを適用する。

活性化関数には一般に非線形関数で微分可能な関数が採用される。今回はシグモイド関数(式 1-3)を採用したが学習対象にあわせて線形関数を使用することもある。

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (1-3)$$

・学習方法

学習させるデータとは、入力データとその望ましい出力データ（教師データ）の組である。教師付き学習の一般化デルタルールは、入力データによって出力されたデータと教師データとの差が評価され、その評価値に比例して結合の重みが更新される。評価された誤差に基づいて出力層から入力層の方向へ学習が行われる（誤差逆伝播法）。重みの最適化法はここに示した1次偏微分に基づく方法（最急降下法）の他に前回の更新量を考慮するモーメント法や2次偏微分に基づく方法が提案されている。一般的に2次の最適化法の方が1回あたりの収束は速いが計算量が増大する。本研究で使用するニューラルネットワークは最急降下法のみを使用した。

[出力層と中間層間の重みの更新]

$$E_k = \frac{(T_k - O_k)^2}{2} \quad (1-4)$$

$$\frac{\partial E_k}{\partial w_{kj}} = \frac{\partial E_k}{\partial O_k} \cdot \frac{\partial O_k}{\partial w_{kj}} = \frac{\partial E_k}{\partial O_k} \cdot \frac{\partial O_k}{\partial S_k} \cdot \frac{\partial S_k}{\partial w_{kj}} \quad (1-5)$$

$$\Delta w_{kj} = -\alpha \frac{\partial E_k}{\partial w_{kj}} \quad (1-6)$$

[中間層と入力層間の重みの更新]

$$\begin{aligned} \frac{\partial E_k}{\partial w_{ji}} &= \sum_k \left(\frac{\partial E_k}{\partial O_j} \right) \cdot \frac{\partial O_j}{\partial w_{ji}} = \sum_k \left(\frac{\partial E_k}{\partial S_k} \cdot \frac{\partial S_k}{\partial O_j} \right) \cdot \frac{\partial O_j}{\partial w_{ji}} \\ &= \sum_k \left(\frac{\partial E_k}{\partial O_k} \cdot \frac{\partial O_k}{\partial S_k} \cdot \frac{\partial S_k}{\partial O_j} \right) \cdot \frac{\partial O_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial w_{ji}} \end{aligned} \quad (1-7)$$

$$\Delta w_{ji} = -\beta \frac{\partial E_k}{\partial w_{ji}} \quad (1-8)$$

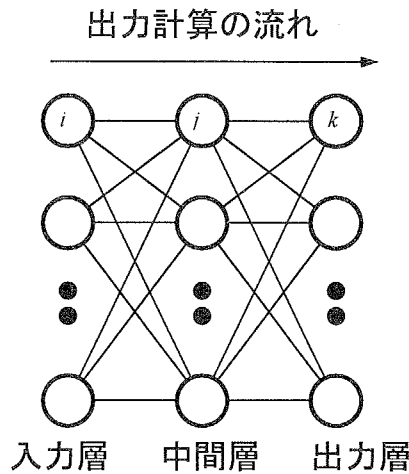


図 1-2 階層型ネットワーク

また、学習誤差である入力データによって出力されたデータと教師データとの差を評価する際に、Hampel らは二次関数ではなくハイパボリックタンジェントを用いた関数を使用した。学習データセットの中で他とは大きく外れたデータを除外することで関数近似での安定性の向上を図った。

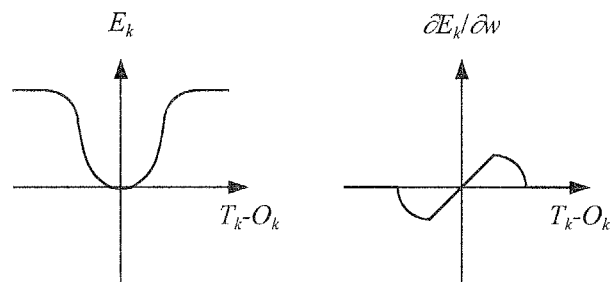


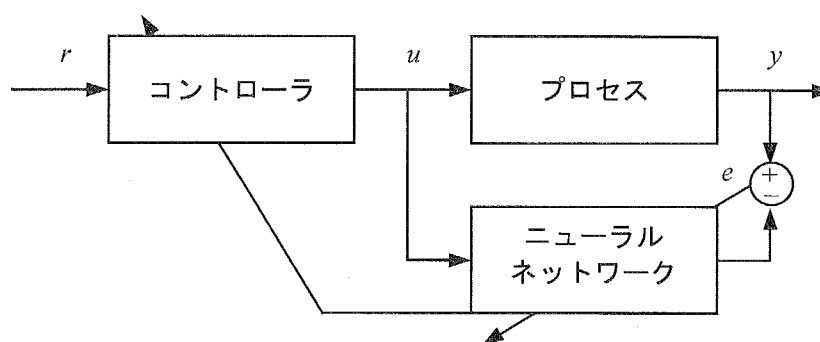
図 1-3 Hampel の tanh estimator

図 1-3 の関数を用いると、評価誤差の大きいデータは学習しないことになり、大きく外れた学習データを除外することができる。

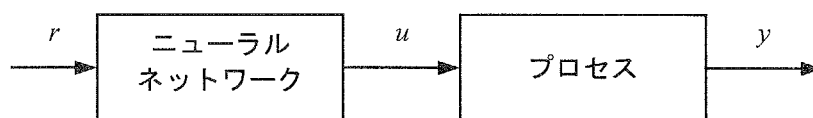
1.3.2. ニューラルネットワークコントローラ

ニューラルネットワークを制御に用いる場合、その使い方を大きく分類するとニューラルネットワークをプロセスの同定器として用いるか、そのままコントローラとして用いるかに分けられる。同定器として用いる間接型制御系（図 1-4a）（並列型制御系ともいわれる）ではネットワークにプロセスの順モデルを学習させ、その結果で他のコントローラをチューニングする方法である。本論文で使用するニューラルネットワークでは、ネットワーク自身をコントローラとして用いて逆モデルを学

習させる、直接型制御系（図 1-4b）または直列型制御系を利用する。



(a) 間接型制御系

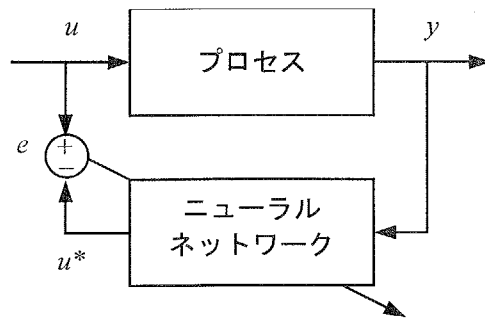


(b) 直接型制御系

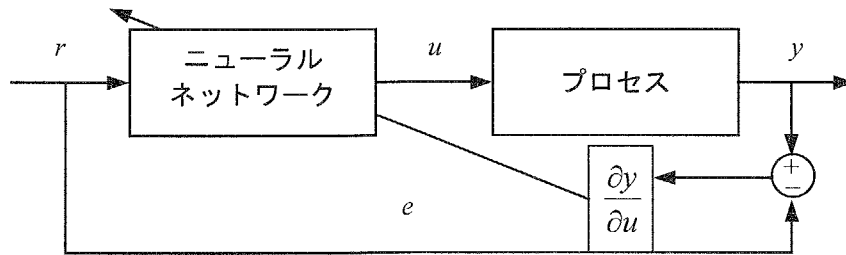
図 1-4 ニューロコントローラの制御系

間接型制御系では、基本的に制御は既存の方法で行われるため制御に対する信頼性が高いといわれる。直接型制御系ではニューラルネットワークの不安定さが直接コントローラに影響を及ぼすためネットワーク自身の安定性に気を配る必要があるが、逆にニューラルネットワークの持つ利点を直接応用できるといえる。

Psaltis, Sideris, Yamamura は逆モデルを学習させる方法として直接法と制御偏差法を提案した（図 1-5）。直接法においてニューラルネットワークはプロセス入力（操作量）に基づいた誤差で学習が行われ、制御偏差法ではプロセス出力（制御量）に基づいた誤差で学習が行われる。直接法では操作量がニューラルネットワークの出力値となっているので、操作量の誤差をそのまま学習に用いることができる。しかし、制御偏差法では制御量の偏差をそのままデルタルールの誤差逆伝播法に適用することはできない。何らかの方法で $\frac{\partial y}{\partial u}$ を得て制御量の誤差を操作量の誤差に変換する必要がある。 $\frac{\partial y}{\partial u}$ を得る方法として、プロセス入力を小さく変更してその出力変化を測定する方法を提案している。



(a) 直接法



(b) 制御偏差法

図 1-5 逆モデルの学習アルゴリズム

徐、辻および金子は間接型の制御系を利用したコントローラを提案した。このコントローラのユニークな点は、モデル化誤差を含むプロセスモデルとニューラルネットワークが並列に配置される点である。ニューラルネットワークの出力はフィードバックコントローラの出力を補正する信号であり、モデルの出力とニューラルネットワークの出力がモデルに入力されて得られる出力の誤差を用いてネットワークの学習を行う。この構成をとることで、モデル化誤差を含むプラントの同定と制御を同時に行うことができる。

山田、藪田は直接型制御系で、ニューロコントローラ出力と従来のコントローラ出力の和をコントローラ出力とした構成を用いて適応型フィードフォワードフィードバックコントローラを離散時間系にて設計した。学習初期では従来のコントローラ出力を用いることで安定性を保証し、その後学習が進むにつれてニューラルネットワークの出力で制御を行う。この方法はニューロコントローラの初期学習に注意を払ったものである。

Bhat と McAvoy はニューラルネットワークを用いて化学プロセスのダイナミックモデルを作成し、コントローラとして利用した。連続攪拌反応器での pH の制御を行うために、様々な大きさの階層型ニューラルネットにプロセスの入出力を学習さ

せ、将来の pH の予測を出力できるようにした。予測性能がよいネットワークを選び、制御に利用した。

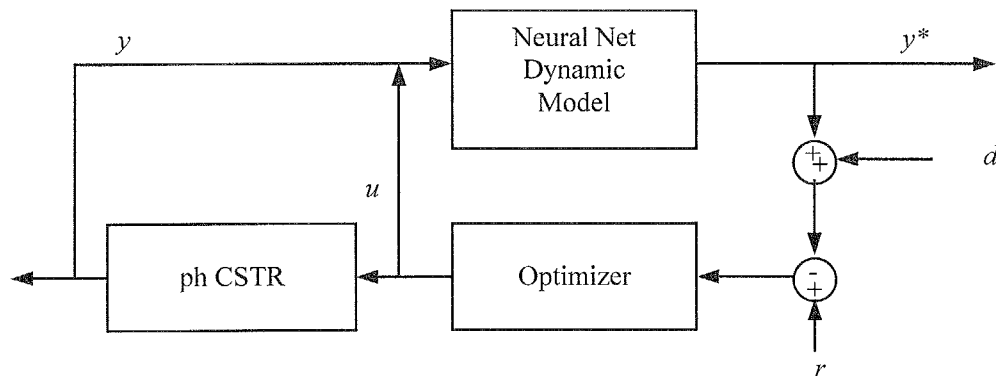


図 1-6 ニューロモデルによる制御

Willis らはプロセスの測定値を定期的に続けてサンプリングできない場合について、ニューロモデルを用いて推定し制御を行った。ペニシリンの生産プロセスにおいて、バイオマスの量を推定するネットワークを過去のデータを学習させることで構築した。バイオマスの測定には時間がかかるため PI コントローラへの入力が遅れてしまうところをニューロモデルの出力を利用した。

また、ネットワークで学習された情報を得るために Mavrovouniotis と Chang (1992) は階級ネットワークを提案した。プロセスモデルを得るために様々な入力データを用いるとき、これらの入力データ種間での関係を得て取捨選択する。このとき入力データ種をいくつかの組に分けて入力ユニットを構成し、学習後に得られた結合重み及びしきい値を直接比較することでデータ種の貢献度を評価した。

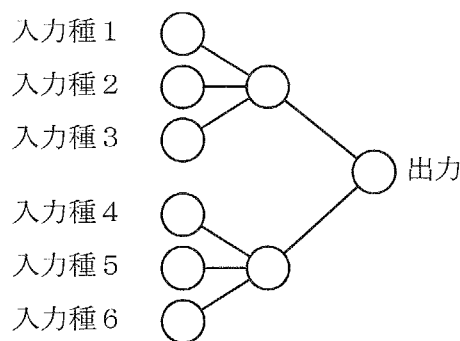


図 1-7 階級ネットワーク

1.3.3. 初期 PENN 法

ニューラルネットワークによる制御では、過去の制御結果を学習してプロセスモデルを生成する。しかし、制御結果は設定値周辺の挙動を示すのみであるため、学習によって得られるNNモデルによるコントローラは学習範囲外での動作が保証できない。さらに、制御結果に含まれる測定誤差をも学習してしまい、実プロセスとかけ離れたモデルが生成されることがある。そこで、NNに与える学習データとして制御結果（ローカルデータ）だけではなく、制御変数空間を覆う大まかな制御方法をIF-THENルールで示した制御ポリシー（グローバルデータ）を採用した。制御ポリシーを制御結果と同時に学習させ、制御結果に含まれる誤差の学習を抑制することでロバスト性に寄与する。これら2種類の学習データを用いる方法をPENN(Policy-and Experience-driven Neural Network)法と呼ぶ。また、コントローラにファジー的な要素を組み込むことが可能になる。

SISO プロセスに対して PENN を利用したニューロコントローラが、1991年に石田愈教授によって PENN 法とともに発表されている。図に示すように、NN への入力は設定値と制御量との偏差を用いた3つであり、それぞれ制御量偏差の時間的差分値2つと制御量偏差である。この3入力の PENN は非線型な対象の定値制御では十分良好な制御を行うことができる。

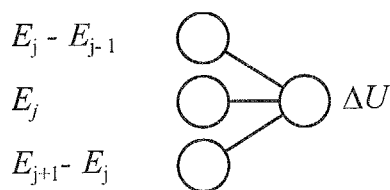


図 1-8 3入力 PENN コントローラ

2. PENN コントローラの提案

この章では、操作量と制御量が各々1つであるプロセス (SISO プロセス) に対応したニューロコントローラを提案する。このコントローラは経験と制御ポリシーという2種類の学習データを使用する PENN 法を採用する。経験データは対象プロセスの過去の運転結果または制御実行中に刻々と得られる制御結果であり、制御ポリシーはプロセスについての既知の知見をルールで示したものである。経験データを決められた個数の重みで効率的に学習することでプロセス特性の局所的な学習を効果的に行うことが可能となり、制御ポリシーの学習で大局的な制御方針を学習できる。

ニューラルネットの入力変数には、制御量の過去と現在の差と、将来望ましい目標値と現在の差、出力変数には操作量の変更量のみを使用する。このため、一般的な制御で使用される、目標値と現在の値の偏差を利用しない特徴を持つ。また、将来望ましい目標値を利用するため、目標値の変更に対して前もって制御動作を開始でき、無駄がなく遅れのない制御が期待できるという予見制御である。

また、このコントローラで使用する変数と速度型 PI コントローラで使用する変数と比較し、ニューロコントローラの重みの役割を把握する。

2.1. PENN コントローラの制御及び学習方法

2.1.1. 制御方法

図 2-1 に示す制御系で使用するネットワーク構造は図 2-1 に示すように、入力ユニットが2つで出力ユニットが1つという非常に小さいものである。入力ユニットでは演算を行わず、入った情報は次のユニットへ結合重み ω_m , ω_p により重み付けされて送られる。出力ユニットではシグモイド関数 (式 1-3) により非線形演算が行われる。また、しきい値は使用しない。ネットワークの入力は制御量に関する差、出力は操作量の差であり速度型制御である。

入力変数は現在時刻 j の制御量 y_j を使用して ΔY_m と ΔY_p を次のように定義する。

$$\begin{aligned}
Y_j &= \frac{y_j - y_{\min}}{y_{\max} - y_{\min}}, R_{j+k} = \frac{r_{j+k} - y_{\min}}{y_{\max} - y_{\min}} \\
\Delta Y_m &= Y_j - Y_{j-1} \\
\Delta Y_p &= R_{j+k} - Y_j
\end{aligned}
\tag{2-1}$$

添え字 $j-1$ は現在時刻 j から 1 制御間隔前の時刻を示し、 $j+k$ は k 間隔後の時刻を示している。したがって入力変数 ΔY_m は過去からの制御量の変化を表し、フィードバック制御の要素である。そして入力変数 ΔY_p は将来 (k 制御間隔後) の目標値 R_{j+k} への変化を表すことになり、予見制御を実現する要素である。添え字 p は現在時刻から時間軸に沿って正の方向 (plus) を、 m は負の方向 (minus) を意味している。パラメータ k の値の指定については 2-2 に説明する。

出力変数は操作量の変更量であり、現在時刻で指定する操作量 u_j と 1 制御間隔前で指定した操作量 u_{j-1} の差である。操作量変更量の最大値 Δu_{\max} は制御対象プロセスによって異なる定数である。

$$\Delta U = (u_j - u_{j-1}) / \Delta u_{\max}
\tag{2-2}$$

制御時に目標値 R は時刻 $j+k$ での目標値を用い、入力変数 ΔY_p は目標値 R_{j+k} と現在の値 Y_j との差となる。そしてニューラルネットワークの演算により出力変数 ΔU が算出され、新しい操作量 u_j を次のようにする。

$$u_j = u_{j-1} + \Delta U \cdot \Delta u_{\max}
\tag{2-3}$$

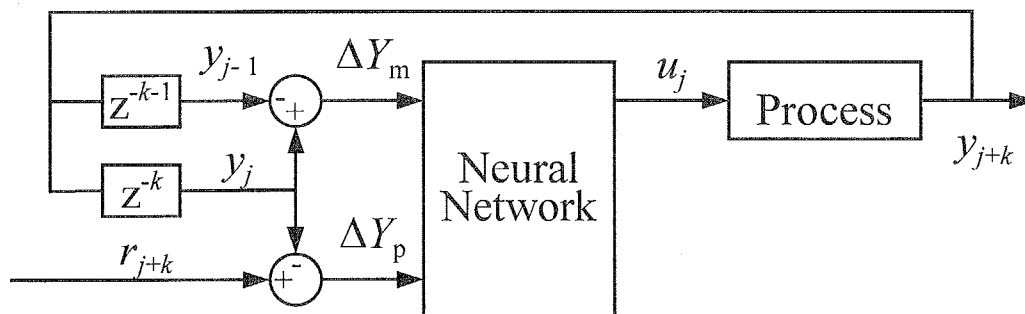


図 2-1 制御方法

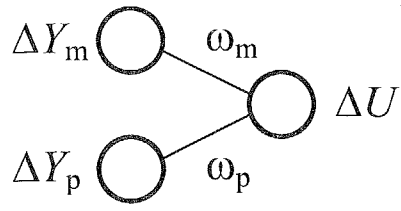


図 2-2 ネットワーク構造

2.1.2. 経験学習

経験学習は図 2-3 で示す直接法で行う。現在時刻の制御量 Y_j は $k-1$ 制御間隔前に行われた操作量変更の結果を反映したものである。ニューラルネットワークの学習データを次のようにする。

$$\begin{aligned}\Delta Y_p &= Y_j - Y_{j-k} \\ \Delta Y_m &= Y_{j-k} - Y_{j-k-1} \\ \Delta U &= (u_{j-k} - u_{j-k-1}) / \Delta u_{\max}\end{aligned}\tag{2-4}$$

図 2-4 に具体的な手順を図示した。まず制御時にネットワークの出力演算によって ΔU が得られ、プロセスへの入力となる。すると k ステップ経過後のプロセス応答が目標値であった時点のデータとして得られる。この入出力関係をネットワークの学習データにすることでプロセス応答を直接学習することが可能であり、局所的な逆モデルを得ることができる。この経験学習を過去の運転結果の学習と制御中のオンライン学習で用いる。

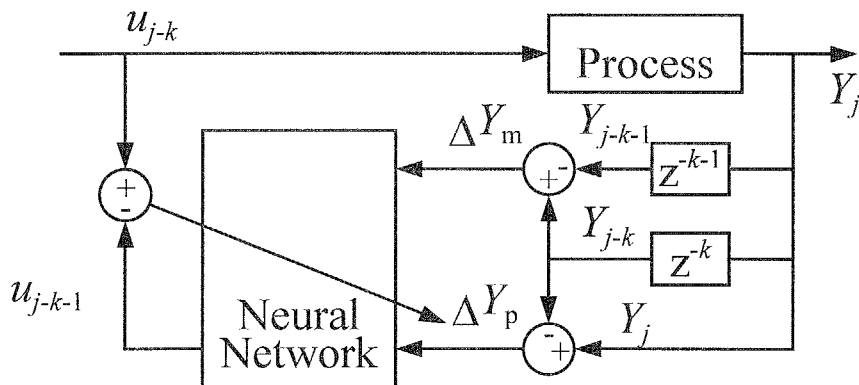


図 2-3 直接法による経験学習

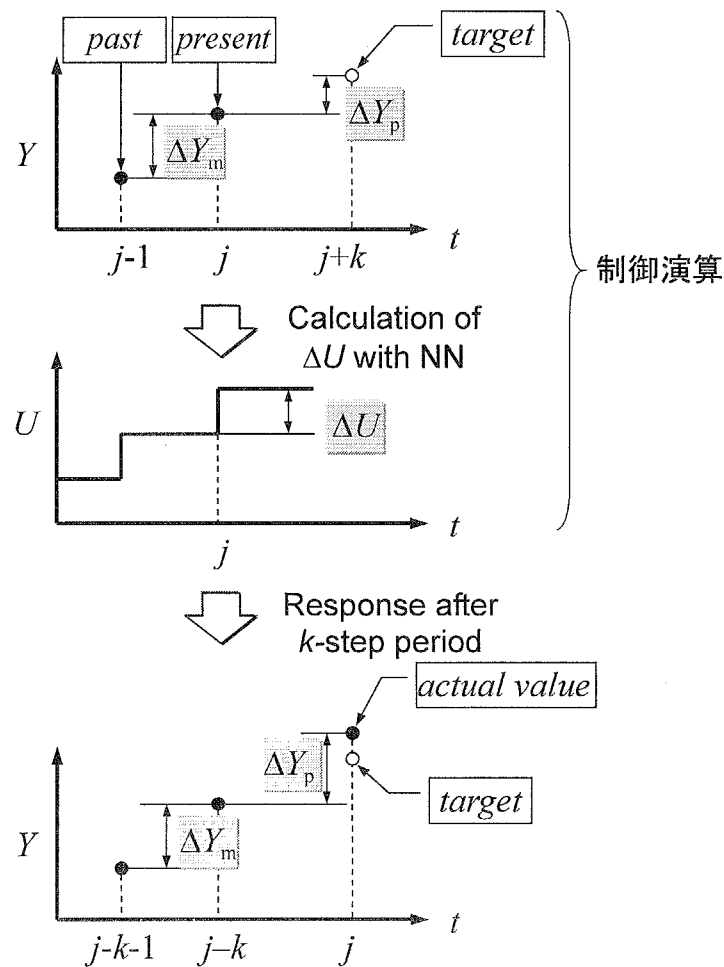


図 2-4 経験学習の手順

また、PENN コントローラは決められた大きさのネットワークを使用するので、対象プロセスを全変数域にわたってモデル化するには結合重みの個数が不足することも考えられる。しかし、上述したオンライン学習により、その時点でのプロセス特性を即座に学習に反映することで局所的なモデル化が可能であり、非線形プロセスへ適用することができる。

2.1.3. ポリシー学習

先に述べた直接法による経験学習では、数多くそして広範囲にわたった経験データがある場合は精度の高い逆モデルが得られる。しかし、一般的に実際の運転状況では操作領域と目標設定値が限られており、化学プロセスにおいても同様である。また得られる経験データは常にノイズを含んでいる。そこで PENN コントローラでは経験データだけでなく大局的な制御ポリシーを導入した。制御の対象としている

プロセスについて、操作量と制御量の関係の大まかな特性があらかじめわかっている場合が多い。この特性を IF-THEN ルールにより記述する。これらのデータで用いる値は PENN で扱う入出力値の最大または最小値である。そのためニューラルネットワークが苦手とする外挿計算を制御の際に行わないので信頼性の向上が見込める。

以下に示す制御ポリシーは操作量を増加させると制御量が増加し、操作量を減少させると制御量が減少するというプロセス特性を IF-THEN ルールで表したものである。

- 1 : IF $\Delta Y_m = 0$ AND $\Delta Y_p = 0$ THEN $\Delta U = 0$
- 2 : IF $\Delta Y_m = -1$ AND $\Delta Y_p = 0$ THEN $\Delta U = +1$
- 3 : IF $\Delta Y_m = +1$ AND $\Delta Y_p = 0$ THEN $\Delta U = -1$
- 4 : IF $\Delta Y_m = 0$ AND $\Delta Y_p = -1$ THEN $\Delta U = -1$
- 5 : IF $\Delta Y_m = 0$ AND $\Delta Y_p = +1$ THEN $\Delta U = +1$

ルール 1 はプロセスの定常状態を示しているが、敷居値を用いない場合は学習させる必要がない。ルール 2 は制御量が最大値から下降して最低値になり ($\Delta Y_m = 0 - 1$)、その値を維持する場合 ($\Delta Y_p = 0 - 0$) には操作量を最大幅増加 ($\Delta U = +1$) させる場合を示している。図 2-5 に PENN で使用する変数を 3 次元座標で示し、制御ポリシーの位置を表示した。各変数は $[-1, +1]$ に規格化されているので、それぞれのポリシーは PENN で扱う変数域では一番外側に存在することになる。そのため経験データが示す局所的データをすべてカバーすることになる。

これら 2 種類の学習データ (経験データと制御ポリシー) は同じニューラルネットワークに適用される。経験データと制御ポリシーに矛盾が生じている場合はこれらのデータの学習回数によって学習結果が異なってくる。ノイズを多く含んだ経験データの場合にはより多くの制御ポリシー学習を行うことでノイズの影響を抑えることができる。

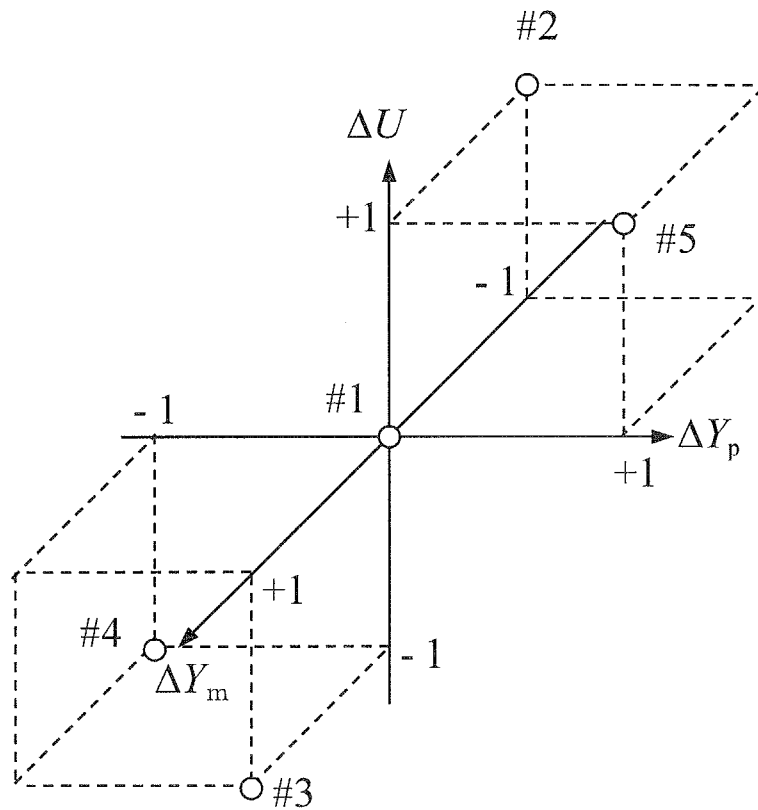


図 2-5 変数空間における制御ポリシーの位置

2.2. PENN コントローラの特性

PENN コントローラで使用する変数について、速度型 PI コントローラとの比較でその役割について把握し、重要な変数である k の指定の指針を示す。

2.2.1. 速度型 PI コントローラとの比較

古くから用いられている速度型 PI コントローラと比較することで PENN コントローラの特性を把握する。

速度型の PI コントローラは次の式で表される。

$$\Delta U = K_p \left\{ (E_j - E_{j-1}) + \frac{T_s}{T_i} E_j \right\} \quad (2-5)$$

速度型制御では、まず操作量の微分量を求め、その後積分（加算）して操作量を得る。そのため、式 2-5 での偏差の変化量 $(E_j - E_{j-1})$ の加算値が偏差となり位置型制御での比例項になり、偏差 E_j の加算値が位置型制御の積分項となる。

目標値が時間に対して変わらない定値制御の場合、目標値 R は定数である。

$$R_j = R_{j-1} = R_{j+k} \quad (2-6)$$

すると PENN の入力変数は

$$\begin{aligned} \Delta Y_m &= Y_j - Y_{j-1} \\ &= (R_j - Y_j) - (R_{j-1} - Y_{j-1}) \\ &= E_j - E_{j-1} \end{aligned} \quad (2-7)$$

と

$$\begin{aligned} \Delta Y_p &= R_{j+k} - Y_j \\ &= R_j - Y_j \\ &= E_j \end{aligned} \quad (2-8)$$

である。

よって、PENN の出力変数（操作量変更量）はシグモイド関数を f とすると、

$$\Delta U = f((E_j - E_{j-1})\omega_m + E_j\omega_p) \quad (2-9)$$

となり、式 2-5 と式 2-9 を比較すると、速度型 PI コントローラでの比例項には重み ω_m が、積分項には重み ω_p が対応していることがわかる。

相違点は、PENN ではシグモイド関数を用いているため、非線形演算が行われ、同時に出力 ΔU に (-1,+1) の制限があることと、重みが学習によって自動的にチューニングされることである。

2.2.2. PENN コントローラの出力

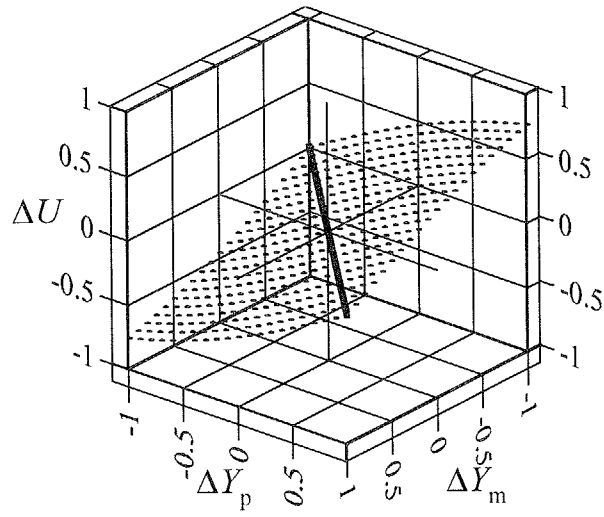
PENN が学習した結果得られる重み ω_m , ω_p と 3 つの変数 ΔY_m , ΔY_p , ΔU で示される PENN コントローラの変数について説明する。

操作量を増加させると制御量が増加し、操作量を減少させると制御量が減少するという先に示した制御ポリシーを学習させた場合、重み ω_m が負の値で ω_p が正の値となる。このときネットワークからは図 2-6(a)で示す出力曲面が得られる。図中の太い実線は曲面と面 $\Delta U = 0$ との交線である。

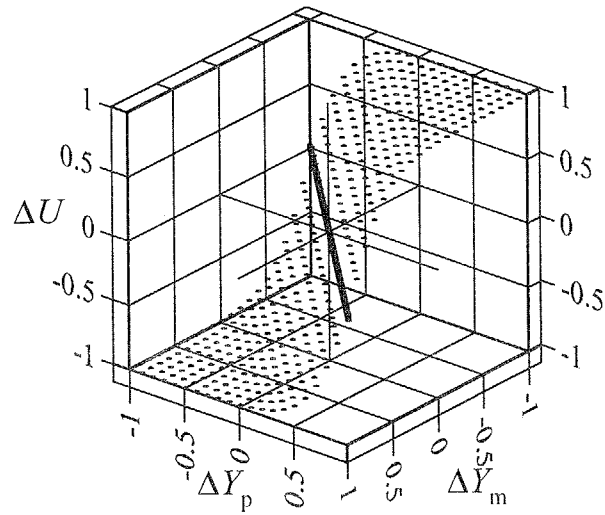
重みの絶対値が大きい場合、曲面の傾きは急になり図 2-6(b)で示すような曲面となる。このときのコントローラの出力は小さい変化に対しても大きな値となる。

また、2つの重みの大きさが異なる場合は図 2-6(c)のようになる。図の場合では

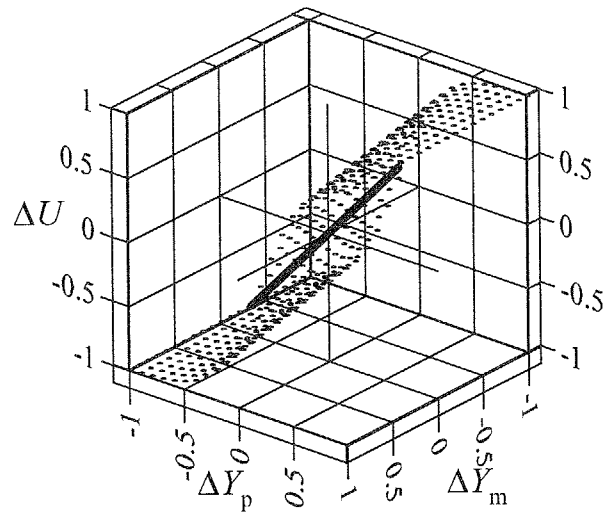
絶対値では ω_p のほうが大きいので、PID コントローラでの積分項がより重視されている状況である。



(a) $\omega_m = -1.0, \omega_p = +1.0$



(b) $\omega_m = -5.0, \omega_p = +5.0$



(c) $\omega_m = -1.0, \omega_p = +5.0$

図 2-6 PENN の出力図

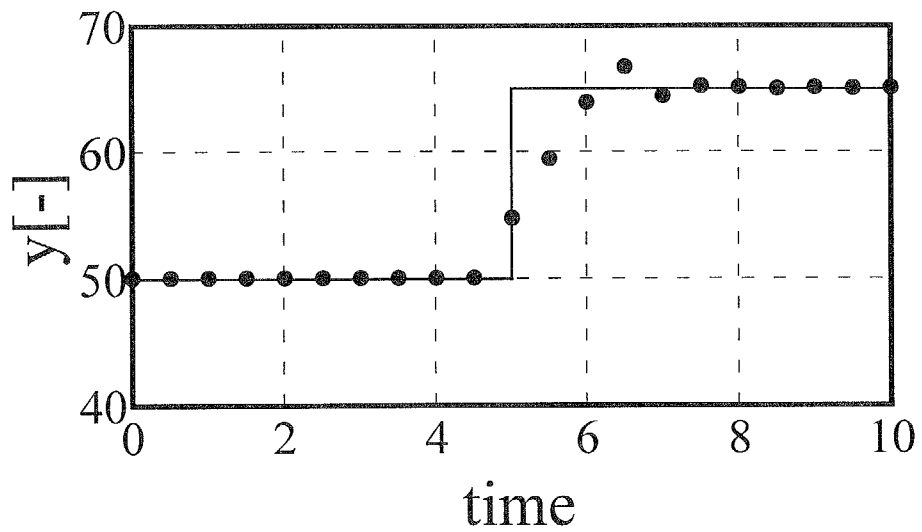
2.2.3. 時間遅れ対応パラメータ k の選択

PENN コントローラにおいてパラメータ k は重要な役割を持つ。パラメータ k は先に示した PENN 入力変数のうち、 ΔY_p の定義式で制御量の将来の値として k 制御間隔後の目標値を用いるため、遅れに対応したパラメータである。 k の値の決定には次に挙げる要因を考慮する。

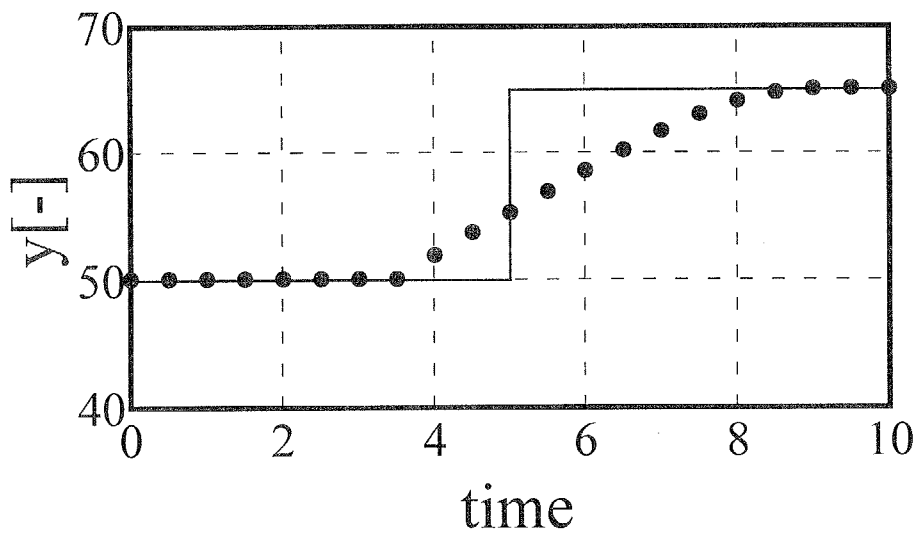
まず、プロセス応答が遅れを伴う場合にはその分 k の値を増やす必要がある。この議論は第 3 章で行う。また、操作量の変更制限値 Δu_{\max} が小さいために設定値変更に追従できない場合では (図 2-7)、パラメータ k は制御の目的に応じて指定する必要がある。設定値変更時刻よりも前に制御量を変えて良い場合は k の値を増加させ、設定値変更時刻より後でのみ制御量を変えてよい場合は k の値を小さくすることになる。図 2-9 は一次遅れモデルの伝達関数 G で表した SISO プロセスにおいて、ステップ状の目標値 (図中の実線) を与えた場合の結果である。

$$G = \frac{K}{1 - Ls} \quad (2-9)$$

$K=5.0, L=0.2$ とし、定常値は $u=50.0$ で $y=50.0$ である。時間 5.0 で目標値を 50 から 65 に変更した。



(a) $k=1$ の場合



(b) $k=3$ の場合

図 2-7 パラメータ k の値による制御結果の違い

2.3. 第2章の結論

制御ポリシーと経験データを学習に用いるニューロコントローラである PENN コントローラを提案した。

- 1) 過去と将来においての制御量の時間的変化量と操作量の変更量のみを用いた非常に小さいネットワークを使用する。
- 2) プロセス応答を直接経験データとして学習でき、同時に制御ポリシーを学習させることでニューラルネットワークの内挿機能のみを使うことになる。
- 3) 速度型 PI コントローラと PENN コントローラが使用するパラメータとの関連を示した。
- 4) 時間遅れ対応パラメータ k はプロセスの遅れ特性と目標値設定を考慮して決定する。

3. 構造化 PENN コントローラの提案

化学プロセスの制御では、非線形特性と共に多変数を取り扱う場合が多い。そこでこの章では第2章で提案した SISO プロセス用 PENN コントローラを複数個組み合わせさせたネットワークを構成し、MIMO プロセス用 PENN コントローラを提案する。提案する構造化 PENN コントローラは、PENN 法の特徴である直接的なプロセス応答の学習と大まかなポリシーの学習を利用しつつ、制御変数間の干渉特性を学習できるものである。SISO 用 PENN コントローラを操作量と制御量の組み合わせの数だけ接続したネットワークを用いることで、変数間関係を学習する結合重みが限定され、その結果制御後に得られる学習済みのネットワークからプロセスの干渉の特性をゲイン定数として把握する。また、構造化 PENN コントローラの適応範囲を拡大するために、変数対毎に遅れ特性が異なるプロセスへ対応したネットワーク構造と、プロセスから得られるデータに含まれるノイズへの対策方法を提案する。そしてプロセス特性が変化した場合のネットワークへの影響を考察し、対応策を提案する。

3.1. 構造化 PENN の構造

操作量の数が M で制御量の数が N であるプロセスに対しての構造化 PENN を図 3-1 に示す。全体では入力、中間、出力層の3層構造である。入力層と中間層は第2章で提案した SISO-PENN を制御量と操作量の各組に対して割り当てたもので、中間層と出力層間の重みは SISO-PENN の干渉を学習するために付加されたものである。その結果、ネットワークの大きさは、入力ユニット数が制御量の種類 N の2倍であり、中間ユニット数は制御量と操作量のすべての組み合わせ数 ($N \times M$) で、出力ユニット数は操作量の種類 M となる。

通常のニューラルネットワークでは、すべてのユニットが重みによって結合されており、学習対象を学習できるような中間層のユニット数を選択する。これに対して提案するネットワーク構造の特徴は、制御量 r と操作量 q の組は図 3-2 に示す部分ネットワークの3つの重み ($v_{r,m}$, $v_{r,p}$, $\omega_{q,r}$) を使用するため、ユニットの個数が決定されており、各組ごとに重みを解析することが可能であることである。

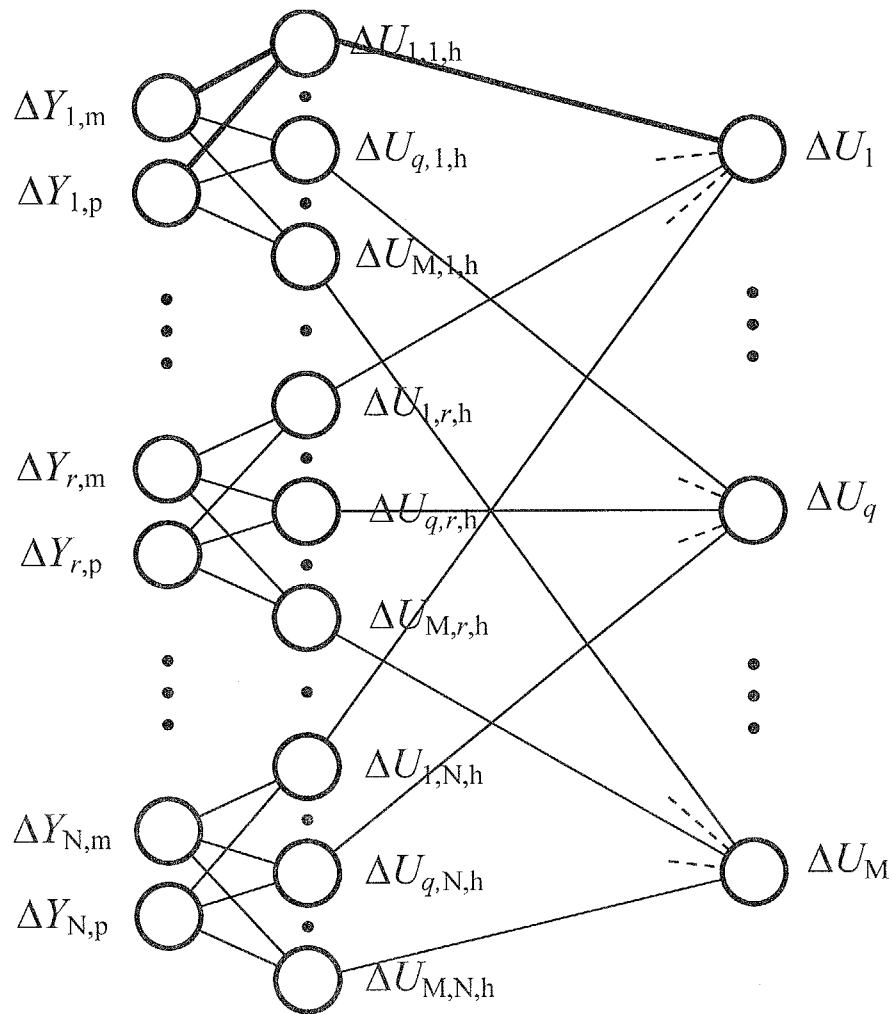


図 3-1 構造化 PENN

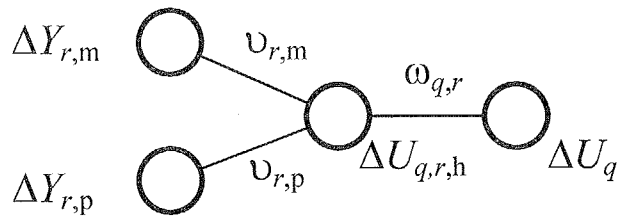


図 3-2 部分ネットワーク

3.2. 学習方法

提案した構造では、操作量と制御量の組に対して PENN を使用しているため、第2章で示した2種類の学習法、経験学習とポリシー学習を利用できる。

3.2.1. 経験学習

SISO プロセスの経験学習では教師データセットとして操作量の変更量 (ΔU) と制御量の時間的変化のデータ (ΔY_m と ΔY_p) を使用した。MIMO プロセスでの経験データも同様に、すべての操作量の変更量と制御量の変化量を1教師データセットとして学習に用いる。具体的には入力変数には $(\Delta Y_{1,m}, \Delta Y_{1,p}, \dots, \Delta Y_{r,m}, \Delta Y_{r,p}, \dots, \Delta Y_{N,m}, \Delta Y_{N,p})$ とし、教師信号 (望ましい出力) として $(\Delta U_1, \dots, \Delta U_r, \dots, \Delta U_M)$ を1データセットとして学習させる。つまり、制御変数の組に対して個別に PENN を割り当てているが、経験学習では同時にすべての結合重みの更新を行う。これに対して、次に説明するポリシー学習では、構造化 PENN の特徴を生かして個別の SISO-PENN に対して学習を行う。

3.2.2. ポリシー学習

経験学習に対してポリシー学習では、図 3-2 が表す各制御変数の組について別々のポリシーを学習させることが可能である。しかし、ポリシー学習によって中間層と出力層間の重みが更新されると、プロセスの持つ干渉特性の学習を妨げることになるため適用方法に注意を払う必要がある。

まず、操作量と制御量の間明らかに強い相関が存在し、その方向が分かっている場合には中間層と出力層間の重みを含んだ3つの重み (図 3-2) に対し、以下の1から5番のポリシーデータを学習させる。この1から5番のポリシーは正の相関 (操作量を増加させると制御量が増加する) でのデータであり、負の相関を学習させる場合には ΔU_q の符号を逆にする。

$$1: \text{IF } \Delta Y_{r,m} = 0 \quad \text{AND } \Delta Y_{r,p} = 0 \quad \text{THEN } \Delta U_q = 0$$

$$2: \text{IF } \Delta Y_{r,m} = -1 \quad \text{AND } \Delta Y_{r,p} = 0 \quad \text{THEN } \Delta U_q = +1$$

- 3 : IF $\Delta Y_{r,m} = +1$ AND $\Delta Y_{r,p} = 0$ THEN $\Delta U_q = -1$
 4 : IF $\Delta Y_{r,m} = 0$ AND $\Delta Y_{r,p} = -1$ THEN $\Delta U_q = -1$
 5 : IF $\Delta Y_{r,m} = 0$ AND $\Delta Y_{r,p} = +1$ THEN $\Delta U_q = +1$

次に、操作量と制御量の間的相关についての方向のみがわかっている場合には、図 3-2 の部分ネットワークの入力ユニットと中間ユニット間の重み ($v_{r,m}, v_{r,p}$) に対してポリシーを適用する。具体的にはポリシーデータの教師信号を出力ユニットの出力 ΔU_q ではなく中間ユニットの出力である $\Delta U_{q,r,h}$ に与える。入力と中間ユニット間の重みにのみポリシーを適用すると、中間ユニットと出力ユニット間の重みは経験学習によって更新され、結果として重みの符号と制御変数組の相関の方向を示すことになる。この学習により、ノイズを学習してしまう場合の影響（重みが 0 になる）を受けにくくなるというポリシー学習の効果を中間ユニットと出力ユニット間の重みについて期待できる。

最後に、相関の強さも方向も未知である場合には次の 6 番と 7 番のポリシーを入力ユニットと中間ユニットの重みに適用する。

- 6 : IF $\Delta Y_{r,m} = -1$ AND $\Delta Y_{r,p} = +1$ THEN $\Delta U_{q,r,h} = +1$
 7 : IF $\Delta Y_{r,m} = +1$ AND $\Delta Y_{r,p} = -1$ THEN $\Delta U_{q,r,h} = -1$

これら 2 つのポリシーデータを PENN の変数空間で示すと図 3-3 となる。これらの点を学習した場合でも経験学習で得られる相関の方向には影響を及ぼさない。これらのデータを学習させるねらいも入力ユニットと中間ユニットの間の重みをノイズの影響を受けにくくすることにある。

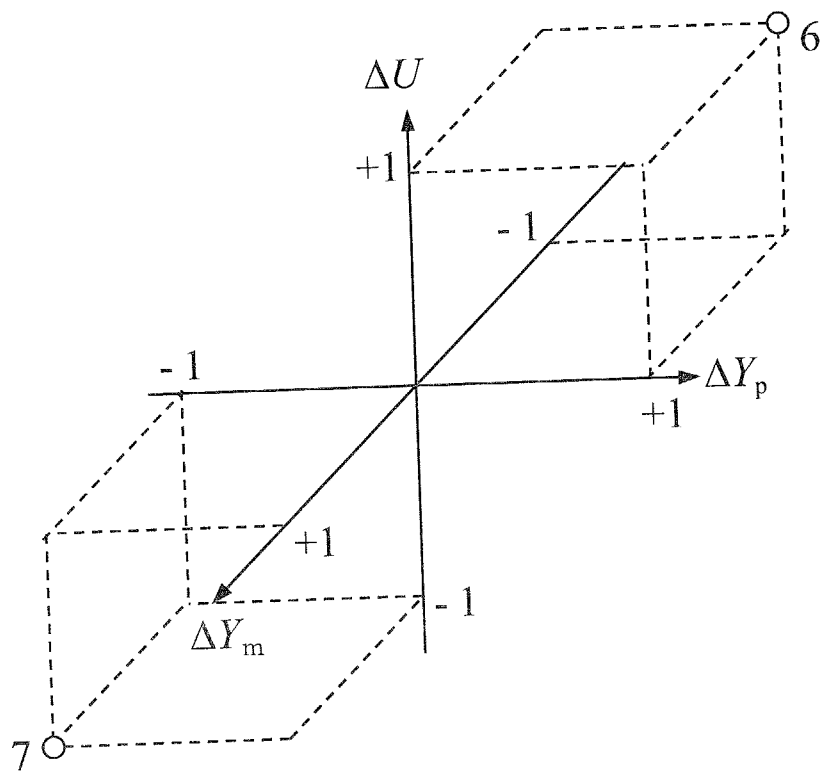


図 3-3 制御ポリシー

3.3. 制御手順

構造化 PENN コントローラによる制御を行う際の手順を、制御前の初期化と制御後の学習に分けて説明する。

・制御前の初期化

[ネットワーク重みの初期化]

すべての重みをある幅を持った乱数により初期化する。その後中間層と出力層の間の重みを0に初期化する。これは初期段階で構造化 PENN が干渉特性を学習していない状態にするためである。

[制御前ポリシー学習]

各変数組に対してポリシー学習を行う。ポリシーは完全に学習できるので重みの更新量が0になるまで学習を行えるが、制御前のポリシー学習は経験学習の前に重みの符号を整える役割なので厳密に行う必要はない。

[過去のプロセスデータの学習]

対象プロセスの運転結果が利用できる場合は、その運転結果を経験データとして学習させる。学習の終了は重みの更新量がある値を下回るまで行うこととする。運転結果が極端に偏っている場合は過度に学習を行わない方がよい。

・制御後の学習

制御を開始した後、オンラインでの経験学習とポリシー学習を行う。経験学習とポリシー学習の回数は対象プロセスのノイズの量によって変える必要がある。ノイズの影響が大きい場合はポリシー学習の回数を増加させる。

3.4. ノイズへの対応

現実のプロセスでは、プロセスから得られるデータに含まれるノイズを考慮する必要がある。ノイズがある場合ではプロセス内の干渉を把握することが困難になる。そこで、工夫を加えて制御を行う方法を提案する。ノイズの影響を低減する方法として、まずデータ処理法の平滑化が考えられるが、ここではニューラルネットワークを改良することで対応した。ノイズを含んだデータを学習しないという方針で、学習関数に工夫を加える。

ニューラルネットワークの学習法である誤差逆伝播法で用いる誤差評価関数は、通常、式 3-8 で示すように教師信号と出力信号との差の 2 乗を用いて評価し、その評価値に比例した値で重みを更新している。

$$\begin{aligned} \frac{\partial E_h}{\partial \omega_{hg}} &= \frac{\partial E_h}{\partial O_h} \cdot \frac{\partial O_h}{\partial \omega_{hg}} \\ &= \frac{\partial \left((T_h - O_h)^2 / 2 \right)}{\partial O_h} \cdot \frac{\partial O_h}{\partial \omega_{hg}} = -(T_h - O_h) \cdot \frac{\partial O_h}{\partial \omega_{hg}} \end{aligned} \quad (3-1)$$

E_h , O_h , T_h , ω_{hg} はそれぞれ評価値, 出力値, 教師信号, 更新対象の重み係数である。

経験データのうち、大きさの比較的小さいノイズを含んだものだけを学習から除外するために、誤差評価関数に不感帯を導入した (図 3-4)。不感帯を導入することにより、ネットワークがすでに学習している情報と近い経験データは学習されなくなる。よってネットワークが正しいプロセスの逆モデルを学習していれば、ノイズを含んだ経験データを除外することが可能となる。そのため、結合重みの初期状態と初期に与えられる学習データの影響を受ける制御開始時の学習状態が保持される傾向になるので、制御前学習が重要となる。制御前学習ではできる限り多くの経験データを学習させるほうが、ネットワーク重みが制御変数域に対して偏りにくくなる。対象プロセスのシミュレータが利用できる場合には最初にその出力を学習させることも考えられる。不感帯の幅は制御前の経験学習での学習誤差の分布を考慮して決定する。

既往のノイズ対策手法としては、測定データを平滑化する目的でさまざまな手法が提案されている。本研究で提案したノイズ対策はニューラルネットワークの学習機能に対して実装するので、既往のデータ処理法の併用が可能でその効果を阻害しない。

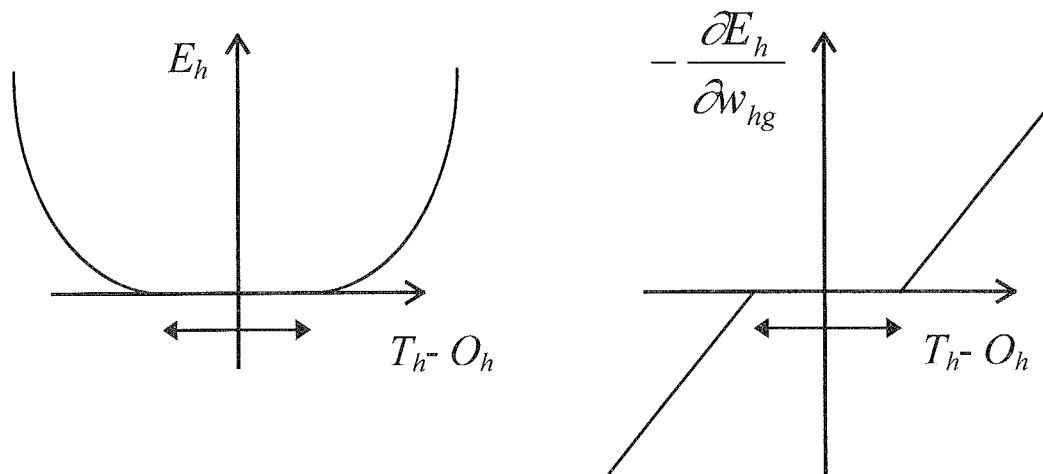


図 3-4 誤差評価関数に不感帯を設ける

3.5. PENN コントローラへの遅れの影響

実際の化学プラントでは長い応答のループを持つプロセスが存在し、さらに多変数プロセス内でも応答の異なるループが混在していることが多い。ここでは PENN コントローラを遅れの伴うプロセスに応用した場合の PENN の特性と応用結果について考察する。

PENN コントローラでは遅れに対応するときには、入力変数の一つである、望ましい制御量変化 ΔY_p を指定するパラメータ k を適切に選択する必要がある。パラメータ k とは k サンプル間隔後の目標値と現在の制御量との差が ΔY_p と指定するものである。したがって、応答が遅い場合には十分な制御量変化が見込まれるように k の値を大きくとり、速い場合には小さくとる。纯粹無駄時間がある場合にはその期間以降での目標値を採用する。

3.5.1. SISO プロセス

まず SISO 線型プロセスに PENN コントローラを適用した。プロセスは 1 次遅れの伝達関数を使用し、ゲイン定数を 5 に固定して時定数およびパラメータ k を変えることで PENN の結合重みの解釈を行った。プロセス定数を表 3-1 に示し、制御結果を図 3-5 に示す。

表 3-2 に示すゲイン定数評価結果より、時定数が 0 で k の値が 1 の場合は実際のゲイン定数とよく一致しているが、時定数が増えてプロセス応答が遅くなると評価

ゲイン値は低く評価される。これは $k=1$ すなわち 1 サンプル後の制御量での評価のため応答が遅いということは 1 サンプル後ではまだ制御量変化が小さい段階であることによる。また、同じ時定数の場合では k の値が大きくなると実際よりも大きく評価されることが見て取れる。これは $k=2$ の場合、2 サンプル間隔後の制御量を用いるためその分制御量変化が大きくなることによる。

表 3-1 SISO 線型プロセスの各種定数

操作量初期定常値	50.0
操作量変更量最大値	20.0
制御量初期定常値	50.0
制御量最低値	0.0
制御量最大値	100.0
制御間隔	0.5

表 3-2 時定数の違いと k の値による評価ゲイン定数

時定数 l	$k=1$	$k=2$	$k=3$
0.0	5.18	6.71	10.48
0.1	4.31	7.61	9.90
0.2	3.61	7.37	9.22
0.5	2.25	5.99	7.02
1.0	1.46	5.70	6.45
2.0	0.90	2.26	4.49
5.0	0.40	0.99	1.93

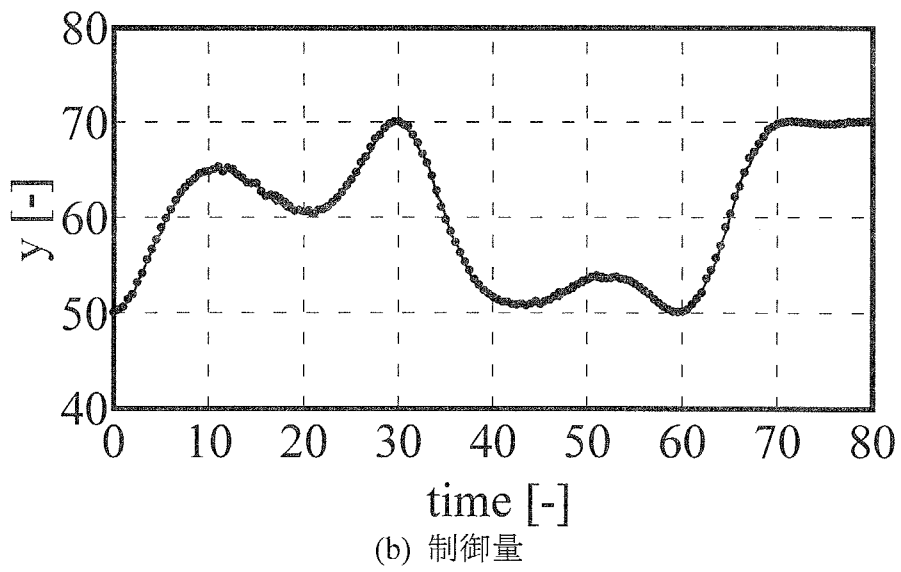
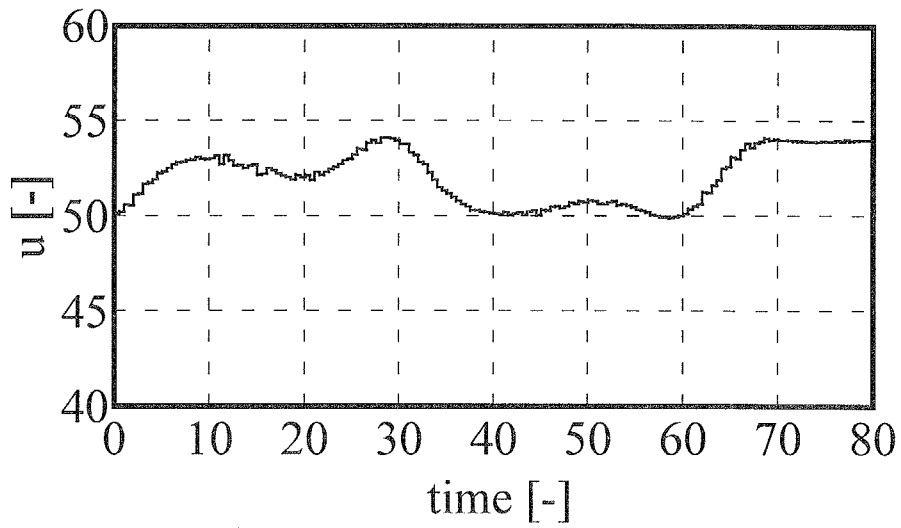


図 3-5 SIS0 線型プロセスの制御結果

3.5.2. MIMO プロセス

MIMO プロセスで制御量と操作量の組み合わせごとに遅れの特性が異なる場合を想定し、構造化 PENN の特性を把握する。対象プロセスとして2入力2出力線型プロセスを用い、時定数を変化させることでパラメータ k と評価ゲイン定数について考察する。

プロセスのゲイン定数を $g_{11}=5.0$, $g_{12}=1.0$, $g_{21}=1.0$, $g_{22}=5.0$ とし、時定数はすべて0とする。この場合、図 3-6 の制御結果のように、操作量 1 が制御量 1 を、操作量 2 が制御量 2 を制御することになる。ここで時定数の変え方を主ループの l_{11} と副ループの l_{21} とし、それぞれ場合について構造化 PENN で算出される評価ゲイン定数を計算した。

主ループを変更した場合の結果を表 3-3 に示す。時定数を変えない場合は実際のゲイン定数と近い値として評価されているが、時定数 l_{11} を増加させた場合対応するゲイン定数 g_{11} が減少する。これは SISO プロセスの場合と同様の結果である。しかし、その他の評価ゲインも干渉のため値が変化している。

また、時定数を増加させていくと制御が不安定な場合そして制御不能の場合がでてくる。制御が不安定な場合とは、片方の制御量のみが目標値に沿って制御される場合であり、試行毎に異なった制御結果が得られる。表の一番右の列には、構造化 PENN から評価ゲインを算出する際の、プロセス定数を考慮する前の評価ゲイン行列の行列式を示している。行列式が 0 に近いか極端に大きな値をとる場合すなわち行列が正則ではない場合はネットワーク重みが不安定になっており、その結果制御に失敗すると考えられる。この結果から、構造化 PENN が対象プロセスを制御できない場合を行列式により判断できる場合があるといえる。制御開始前の運転データの学習段階ですでにネットワークが不安定な場合は制御を行わない、オンライン学習で不安定になった場合は他の制御方式に切り替えるか学習前の重みに戻して対処するという使用方法が考えられる。

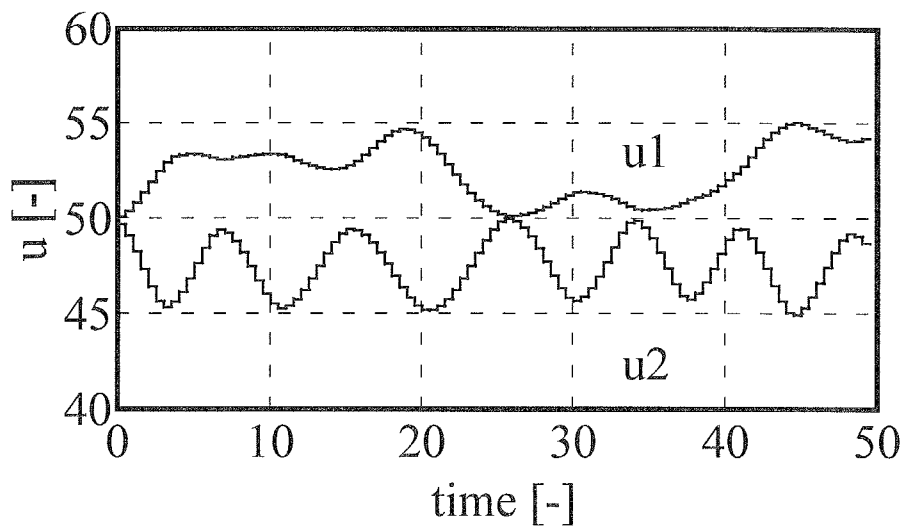
副ループの時定数 l_{21} を変えた場合は対応する評価ゲイン定数 g_{21} が減少するだけで、他の値にはあまり影響を及ぼさない結果が得られた。

表 3-3 主ループの時定数を変えた場合

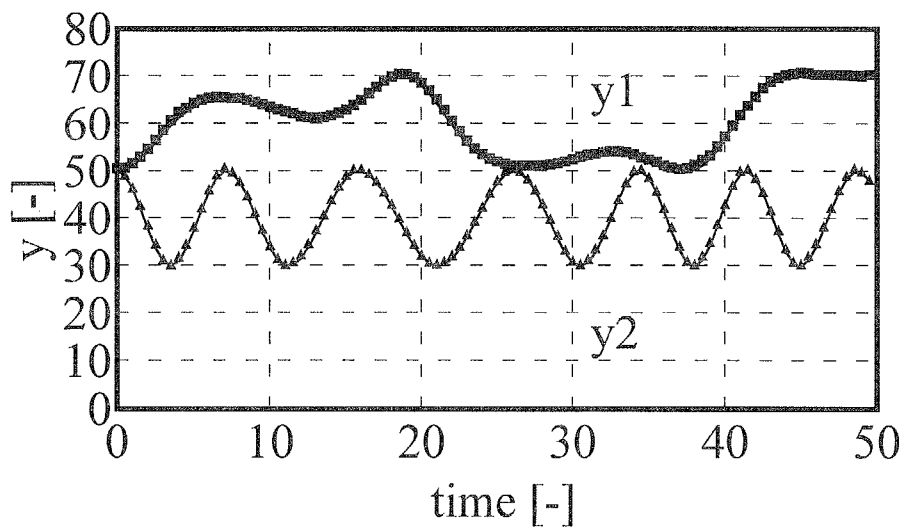
時定数		u_1-y_1	u_1-y_2	u_2-y_1	u_2-y_2	行列式
u_1-y_1		5.00	1.00	1.00	5.00	
制御成功	0.0	4.27	1.19	1.48	3.73	0.57
↑	0.5	2.08	0.79	0.77	3.51	0.27
↑	1.5	1.03	0.74	0.68	3.51	0.12
↑	2.5	0.68	0.67	0.72	3.51	0.08
↑	4.0	0.53	0.65	0.71	3.42	0.05
不安定	4.5	0.52	0.59	0.67	3.28	0.05
↑	4.8	0.48	-0.01	0.55	2.36	0.05
制御不能	4.9	-5286.08	-132.29	-36.97	2.38	-698.93
↑	5.0	1.66E+13	-173.28	8.36E+10	2.75	2.41E+12
↑	15.0	-136.59	221.60	0.97	2.96	-24.73

表 3-4 副ループの時定数を変えた場合

時定数		u_1-y_1	u_1-y_2	u_2-y_1	u_2-y_2	行列式
u_2-y_1						
制御成功	0.0	4.27	1.19	1.48	3.73	0.57
↑	0.5	4.08	1.15	0.93	3.67	0.56
↑	1.5	3.99	1.13	0.49	3.64	0.56
↑	2.5	3.84	1.11	0.02	3.62	0.55
↑	5.0	3.88	1.12	0.11	3.66	0.56
↑	15.0	3.90	1.13	0.15	3.62	0.56



(a) 操作量



(b) 制御量

図 3-6 2 入力 2 出力線型プロセスの制御結果

時定数を大きくした場合に制御が不能となったが、パラメータ k を変更することで制御可能となることがある。時定数とパラメータ k との関係を様々な組み合わせについて調べた。主ループ以外の時定数を 0.2 として主ループの時定数 l_{11} を大きくし、対応する部分ネットワークのパラメータ k_{11} のみを変えた場合についての結果を表 3-5 に示す。その他の k の値は 1 である。ループごとに k の値を変える場合のネットワーク構造は図 3-7 となる。表 3-5 の値は各制御量の目標値からの差を合計したものであり、制御に成功した組み合わせでは斜字体で記してある。

表 3-5 主ループの時定数とパラメータ k の組み合わせによる制御成績

k	l_{11}	y_1					y_2				
		3.0	6.0	8.0	10.0	12.0	3.0	6.0	8.0	10.0	12.0
1.0		<i>17.0</i>	<i>20.5</i>	127.4	29.8	1194.8	<i>33.0</i>	<i>45.0</i>	4225.6	450.5	109.7
2.0		<i>34.1</i>	<i>37.5</i>	<i>55.3</i>	<i>87.7</i>	29.4	<i>45.0</i>	<i>63.3</i>	<i>84.6</i>	<i>125.7</i>	2245.6
3.0		179.9	<i>45.7</i>	818.7	1075.5	<i>87.8</i>	180.6	<i>52.0</i>	44.7	224.3	<i>79.0</i>
4.0		2139.5	2554.9	212.6	996.4	868.0	455.8	24.9	5810.3	608.3	197.3
5.0		2054.4	---	---	---	1130.7	255.6	---	---	---	74.0

この結果から、パラメータ k の値が 1 よりも 2 のほうが大きい時定数においても制御が成功するが、4 以上の場合は時定数に関わらず不安定であるといえる。すなわち遅れを含むプロセスに構造化 PENN を用いる場合パラメータ k の大きさは 3 までとするのがよい。ただし、遅れが小さい場合（時定数が 3 未満）にはパラメータ k によらず制御は成功した。また、副ループの時定数を変えた場合はいずれの組み合わせでも制御は成功した。

これまでの結果より、パラメータ k の選択は第 2 章で説明した操作量変更量の最大値 (Δu_{\max}) と目標値変更設定の考慮とともに、本章で説明した遅れの特性を考慮する必要がある。また、プロセスが非常に長い無駄時間を伴う場合には他の方法を利用して無駄時間を補償する必要がある。

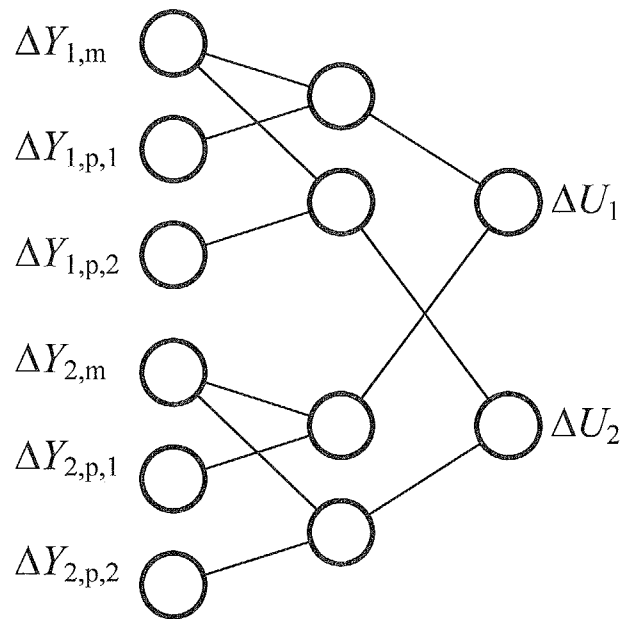


図 3-7 ループごとにパラメータ k が異なる構造化 PENN コントローラ

3.6. 学習結果のネットワークからのプロセスゲインの推定

通常用いられるニューラルネットワークは入出力データを学習させるために、中間層のユニット数を選択するので、学習は容易にできる。しかし、学習結果は分散された複数の重みの値として表現されるため、ネットワークから学習した内容を取り出すことは困難であった。そして、どの重みが学習の際に重要な役割を担っているのかがわからず、ネットワーク重みが正常な状態にあるのかどうかの判断が下せないため信頼性が十分確保されているとはいえない。これに対して構造化 PENN では学習に用いる重みが限定されており、部分ネットワーク毎に重みの解釈を試みる事が可能である。ここでは入力と出力の数が等しい冗長自由度のない系を想定する。

構造化 PENN の重みを解析することにより学習したプロセスの干渉特性を評価する方法を、3入力3出力の線型プロセスによって具体的に説明する。あらかじめ干渉特性がわかっているプロセスを制御した後に、ネットワークの重みから干渉特性を評価し、実際の特徴との比較を行う。ここで以下の点に留意した。

図 3-8 に示す 1 次遅れの伝達関数を 9 つ組み合わせることで図 3-9 のプロセスを作成した。遅れの時定数はすべて等しく 0.0 とし、ゲイン定数のみを図のように変えることで干渉特性を表した。このゲイン定数を、ネットワークの重みから推定する。

制御間隔を 0.5 とし、プロセスの定数値 $u_{q,\min}$, $u_{q,\max}$, $\Delta u_{q,\max}$, $y_{r,\min}$, $y_{r,\max}$ をそれぞれ 0.0, 100.0, 20.0, 0.0, 100.0 とした。3 つの制御量の初期値 $y_{r,\text{initial}}$ を 50.0 とし、このときの操作量を 50.0 とした。1 次遅れの時定数を小さくしたので PENN で用いる入力パラメータ ΔY_p を算出する際の k の値を 1 とした。目標値は次の式とした。

$$\begin{aligned} y_1 &= y_{1,\text{initial}} + 10.0 - 10.0 \times \sin\{36.0(t/50.0) + 0.5\pi\} \\ y_2 &= y_{2,\text{initial}} + 10.0 - 10.0 \times \sin[33.0\{(t-1.0)/50.0 + 0.5\pi\}] \\ y_3 &= y_{3,\text{initial}} + 10.0 - 10.0 \times \sin[30.0\{(t-3.0)/50.0 + 0.5\pi\}] \end{aligned} \quad (3-2)$$

使用した制御前学習用経験データを図 3-10 に示す。このデータには 98 組の経験データが含まれており、操作量をランダムに変更することで得たものである。

使用した構造化 PENN コントローラは入力ユニットが 6 つで中間ユニット 9 つ、出力ユニットが 3 つで構成されていて、結合重みの個数は入力層と中間層間に 18、

中間層と出力層間に 9 である。制御前には結晶成長プロセスの場合と同様に重みの初期化を行い、制御前の経験データは 3 回試行分学習を行った。

制御結果は図 3-11 のようになり、すべての制御量が目標値に沿って良好に制御された。

次に、制御後に得られたネットワークから干渉特性を得る方法を説明する。この場合の干渉特性とはゲイン定数のことである。ネットワークから対象プロセスのゲイン定数を評価には 2 段階の手順を踏む。

(1) 構造化 PENN を PENN 行列 \mathbf{P} で表現する。行列 \mathbf{P} の要素 $p_{r,q}$ は SISO-PENN (制御量と操作量の関係を学習したもの) に対してそれぞれ得られるものである。

(2) ゲイン定数を、PENN 行列 \mathbf{P} を用いて計算する。

まず、PENN 行列 \mathbf{P} を求める。中間ユニットの出力値 $\Delta U_{q,r,h}$ は入力値 $\Delta Y_{r,m}$ and $\Delta Y_{r,p}$ によって次のように得られる。(敷居値は使用していない)

$$\Delta U_{q,r,h} = f(\Delta Y_{r,p}, \Delta Y_{r,m}) = \frac{1 - e^{-(\Delta Y_{r,p} v_{r,p} + \Delta Y_{r,m} v_{r,m})}}{1 + e^{-(\Delta Y_{r,p} v_{r,p} + \Delta Y_{r,m} v_{r,m})}} \quad (3-3)$$

同様に、部分ネットワークの出力値 ΔU_q が次のように計算される。

$$\Delta U_q = g(\Delta U_{q,r,h}) = \frac{1 - e^{-\Delta U_{q,r,h} \omega_{q,r}}}{1 + e^{-\Delta U_{q,r,h} \omega_{q,r}}} \quad (3-4)$$

$\Delta Y_{r,m}$ が 0 のときに、 $\Delta Y_{r,p}$ の ΔU_q への影響の大きさを PENN パラメータ $p_{r,q}$ と呼ぶ。

$$p_{r,q} = \left. \frac{\partial \Delta U_q}{\partial \Delta Y_{r,p}} \right|_{\Delta Y_{r,m}=0} = \frac{v_{r,p} \omega_{q,r}}{4} \quad (3-5)$$

このように、PENN パラメータは制御量の変化量と操作量の変更量の関係を線型近似したものになり、操作量の変更量は次のように表すことができる。

$$\Delta U_q \approx \sum_r p_{r,q} \Delta Y_{r,p} \quad (3-6)$$

例題プロセスである 3 入力 3 出力プロセスでは PENN パラメータは 9 つあり、そのうちの $p_{1,1}, p_{1,2}, p_{1,3}$ は入力値 $\Delta Y_{1,p}, \Delta Y_{2,p}, \Delta Y_{3,p}$ に 1, 0, 0 を与えた場合に得られる。同様に $p_{2,1}, p_{2,2}, p_{2,3}$ は 0, 1, 0, を、 $p_{3,1}, p_{3,2}, p_{3,3}$ は 0, 0, 1 を入力することで得られる。

次に、得られた PENN 行列からゲイン定数を算出する。ゲイン定数 $g_{q,r}$ はプロセ

スの定義より次のように表せる。

$$\Delta Y_{r,p} = \sum_q g_{q,r} \Delta U_q \quad (3-7)$$

式 3-5 と式 3-6 から、ゲイン定数 $g_{q,r}$ は $g'_{q,r}$ として次の連立方程式により評価できる。

$$\begin{aligned} \Delta Y_{1,p} = 1 &= g'_{1,1} p_{1,1} + g'_{2,1} p_{1,2} + g'_{3,1} p_{1,3} \\ \Delta Y_{2,p} = 0 &= g'_{1,2} p_{1,1} + g'_{2,2} p_{1,2} + g'_{3,2} p_{1,3} \\ \Delta Y_{3,p} = 0 &= g'_{1,3} p_{1,1} + g'_{2,3} p_{1,2} + g'_{3,3} p_{1,3} \\ \Delta Y_{1,p} = 0 &= g'_{1,1} p_{2,1} + g'_{1,2} p_{2,2} + g'_{1,3} p_{2,3} \\ \Delta Y_{2,p} = 1 &= g'_{1,2} p_{2,1} + g'_{2,2} p_{2,2} + g'_{3,2} p_{2,3} \\ \Delta Y_{3,p} = 0 &= g'_{1,3} p_{2,1} + g'_{2,3} p_{2,2} + g'_{3,3} p_{2,3} \\ \Delta Y_{1,p} = 0 &= g'_{1,1} p_{3,1} + g'_{2,1} p_{3,2} + g'_{3,1} p_{3,3} \\ \Delta Y_{2,p} = 0 &= g'_{1,2} p_{3,1} + g'_{2,2} p_{3,2} + g'_{3,2} p_{3,3} \\ \Delta Y_{3,p} = 1 &= g'_{1,3} p_{3,1} + g'_{2,3} p_{3,2} + g'_{3,3} p_{3,3} \end{aligned} \quad (3-8)$$

さらに、プロセスごとに異なる定数である操作量の最大変更量および制御量の上下限値を考慮すると、

$$\Delta Y_r = \frac{y_{r,\max} - y_{r,\min}}{\Delta u_{q,\max}} g_{q,r} \Delta U_q \quad (3-9)$$

となる。

その結果、例題プロセスのゲイン定数は次のように評価された。

$$G' = \begin{bmatrix} -2.419 & -4.934 & +2.852 \\ +2.990 & -2.067 & +1.502 \\ -0.771 & +1.170 & -3.720 \end{bmatrix}$$

実際のゲイン定数は

$$G = \begin{bmatrix} -3.000 & -6.000 & +3.000 \\ +3.000 & -3.000 & +2.000 \\ -1.000 & +1.000 & -4.000 \end{bmatrix}$$

である。

G' と G を比較すると、まず、各ゲイン定数の符号が一致していることがわかる。そして、操作量1に対する制御量1のゲイン定数-3.000がネットワークからの評価では-2.419、操作量2に対する制御量1のゲイン定数+2.000に対しては+1.502という結果となっている。ネットワークから評価されたゲイン定数と実際のゲイン定数の差は最大で18%以下であり、干渉特性を十分把握していると言える。

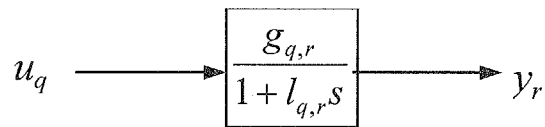


図 3-8 伝達関数

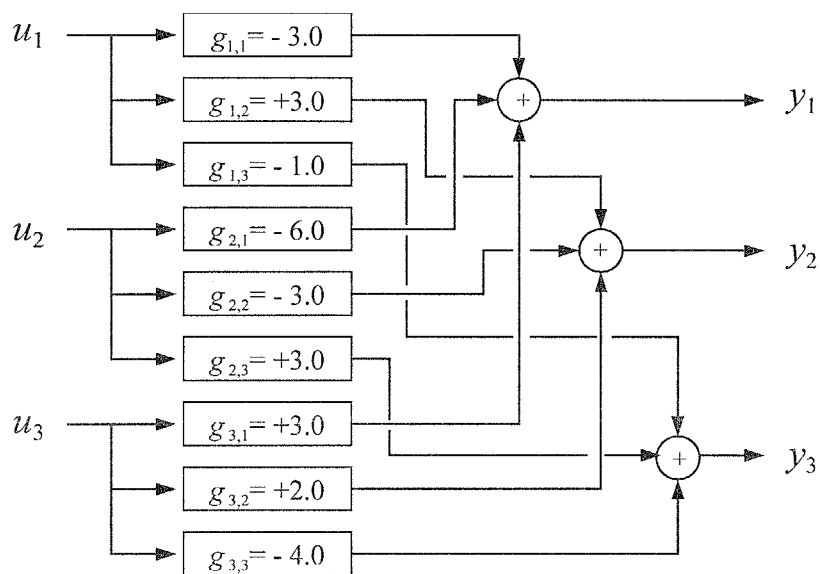
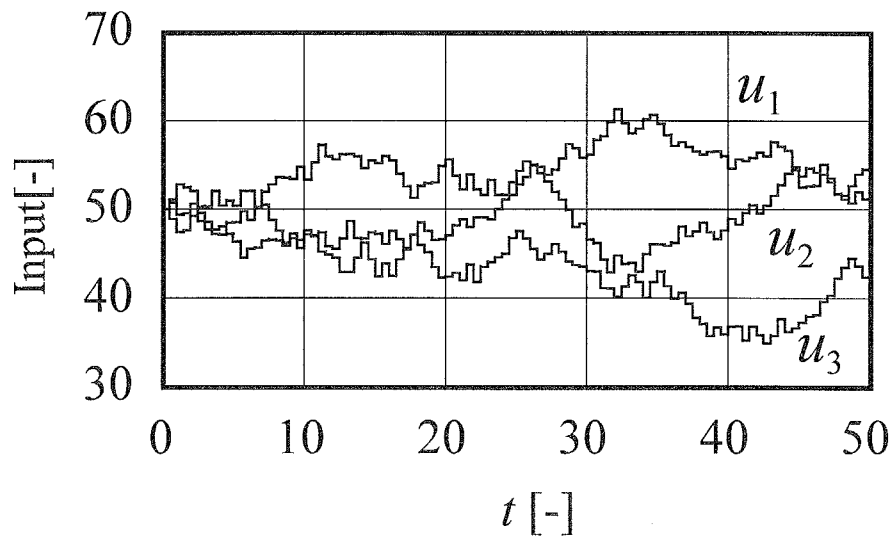


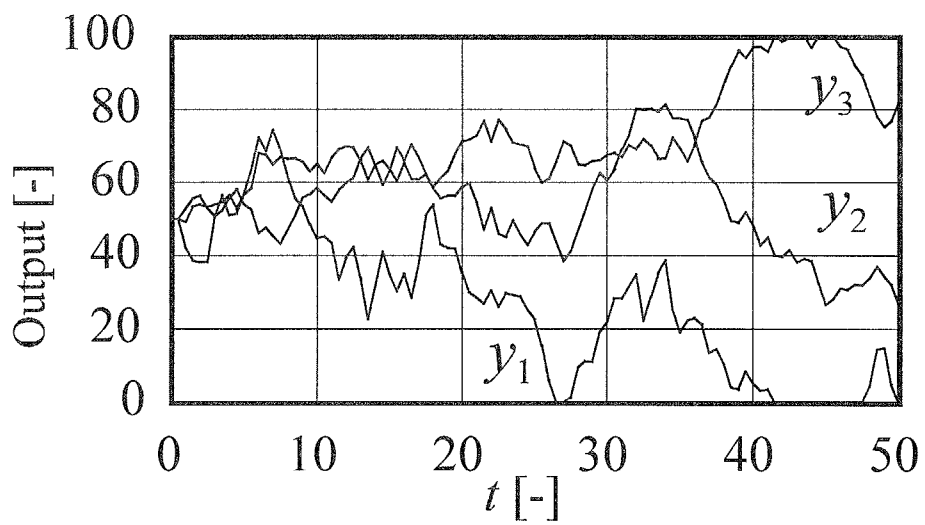
図 3-9 3 入力 3 出力線型プロセス

表 3-6 3 入力 3 出力線型プロセスの各種定数

操作量初期値	50.0
操作量変更量最大値	20.0
制御量初期定常値	50.0
制御量最低値	0.0
制御量最大値	100.0
制御間隔	0.5

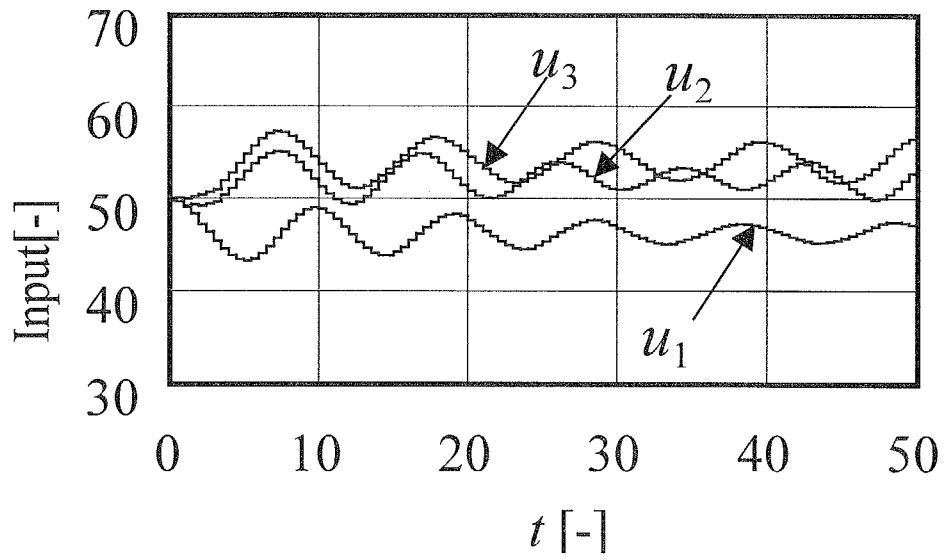


(a) 操作量

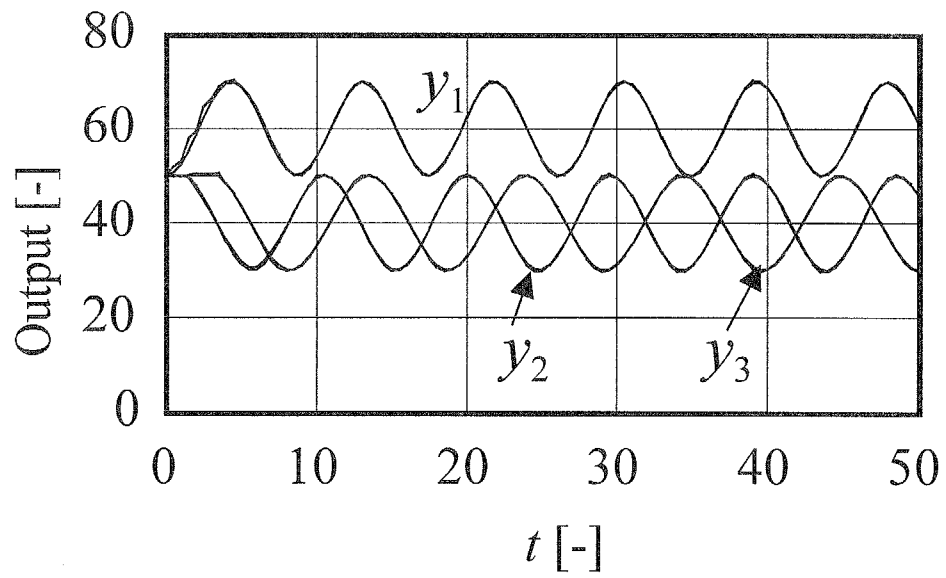


(b) 制御量

図 3-10 制御前学習データ



(a) 操作量



(b) 制御量

図 3-11 制御結果

3.7. 行列式を用いたコントローラの状態把握

既存のニューロコントローラは、学習した内容が把握できないために、信頼性に問題がある。これに対して構造化 PENN コントローラは、前節で提案した学習した内容の把握機能を利用することで自己診断が可能である。

前節で説明した、プロセスの干渉特性をゲイン定数としてネットワークから算出する方法では、構造化 PENN に含まれる SISO-PENN を PENN パラメータで代表し、その逆行列を計算している。ここではこの PENN パラメータの行列 \mathbf{P} に注目し、構造化 PENN の状態を把握する。

PENN 行列 \mathbf{P} が正則ではない場合、逆行列が計算できないので、プロセスの干渉特性の評価が行えない。そして、PENN 行列の逆行列が計算不可能であるときは、構造化 PENN が不安定もしくは実現不可能な干渉特性を学習していると言え、プロセス制御が正常に行われていないことになる。

行列の行列式が 0 である時は、その行列が正則ではないといえるので、これを構造化 PENN の状態を示す指標とする。構造化 PENN から得られた PENN 行列 \mathbf{P} の行列式 $|\mathbf{P}|$ が 0 か 0 に近い場合は構造化 PENN の状態が不安定であるとする。

ただし、構造化 PENN の状態が不安定ではないと判断された場合でも、制御の安定性が保証されたわけではないことに注意する必要がある。行列式はコントローラの状態を示すものであるため、構造化 PENN が安定と判断されても、そのネットワーク重みが正確にプロセス特性を反映したものであるかどうかは不明であるため、制御の保証にはならない。

利用法としては、制御時にこの PENN 行列の行列式を確認し、不安定な方向に向かっているのかどうかを判断し、完全に不安定になる前に警告を発することが考えられる。このような自己診断機能は、既存のニューラルネットワークには見られない特徴である。

3.8. プロセス特性変化への対応

PENN はオンライン学習によってプロセス特性を効果的に学習し、即座にネットワーク重みに反映させることができる。実際のプロセスは、その特性が常に固定さ

れているものと、時間が経過すると変化するものがある。プロセス特性が徐々に変化する場合には、局所的なオンライン学習によって特性の学習が可能ではあるが、短時間に大きな変化がある場合にはプロセス特性を学習することが困難になる。ここでは SISO プロセスにてプロセス特性が大きく変化する状況を作り出し、変化時の運転データを学習した際にネットワークに与える影響を考察し、対応策を提案する。

SISO プロセスとして 1 次遅れの伝達関数を用い、パラメータを表 3-7 に示した。1 回の試行を時間 20 として 7 試行目まではプロセス特性を変化させずに制御およびオンライン学習を行い、その後時間 5.5 以降では、コントローラからの操作量に強制的に値 20 を加えた。これは操作量にバイアスがかかる状況である。コントローラはこのバイアス量を利用できないとする。ネットワークは構造化 PENN の部分ネットワークである図 3-2 を使用した。

その結果 (図 3-12)、オンライン学習を繰り返して行くと次第にネットワークの重みが減少してほぼ 0 になった。入力層と中間層間の重み v_m, v_p には制御ポリシーの学習が行われているため重みは 0 にはならないが、中間層と出力層間の重み w が 0 になった。

プロセス特性が変化しない状況では、制御量変化は操作量の変更によって起きるが、プロセス特性が変化するとその影響を受けて制御量も変化する。このときの運転データを学習させると、ニューロコントローラは操作量と制御量の誤った関係を学習することになる。プロセス特性変化の原因 (ここではバイアス量) をニューラルネットワークの入力として用いている場合は、制御量変化と特性変化の原因との関係を学習によって考慮できるが、このニューロコントローラでは操作量変化以外の情報 (この例では操作量にバイアス量 20 が加わる) を利用できない状況である。

プロセス特性変化が PENN コントローラに与える影響を軽減する策として、オンライン学習で用いる経験データが、それまでに PENN が獲得した情報と大きく異なる場合はその経験データを学習しないという次のアルゴリズムを採用した。

これから学習しようとする経験データの入力パラメータ $\Delta Y_m, \Delta Y_p$ を PENN に入力してネットワーク出力 ΔU を得て、経験データの教師信号である出力パラメータ ΔU と符号を比較する。符号が異なっている場合はその経験データを破棄する。

図 3-13 で具体的に説明する。制御時に低い目標値 $\Delta Y_{p,target}$ に対して PENN コント

ローラが操作量変化 ΔU を算出したとする。学習時にはこの値が教師信号となる。大きな特性変化が原因となった制御量変化は高い値 $\Delta Y_{p,observed}$ として表れるが、この値をネットワークに入力して出力を計算すると、高い目標値に対する出力として教師信号とは異なった符号の出力値 ΔU が出力される。このような符号の異なる経験データを除外することで大きな特性変化の影響を回避することが可能となる。

大きな特性変化による制御量変動の抑制としては平滑化手法による対応が考えられるが、平滑化では特性変化の際の学習データを除外することは困難である。また、大きなノイズが測定値に入った場合にもこの学習アルゴリズムが有効である。さらに、計算時間への影響についても最小限であり、通常の学習計算には出力計算を含んでいるので符号比較の追加のみで済む。

提案したアルゴリズムを採用した結果図 3-14 のようにプロセス特性の変化の影響を抑制することができた。ネットワークの結合重みは0にはならず、特性が変化した直後は目標値から外れているが、すぐに追従している。

表 3-7 SISO プロセスで使用した条件

操作量初期値	50.0
操作量変更量最大値	50.0
制御量初期定常値	50.0
制御量最低値	0.0
制御量最大値	100.0
制御間隔	0.5
ゲイン定数	5.0
時定数	0.2

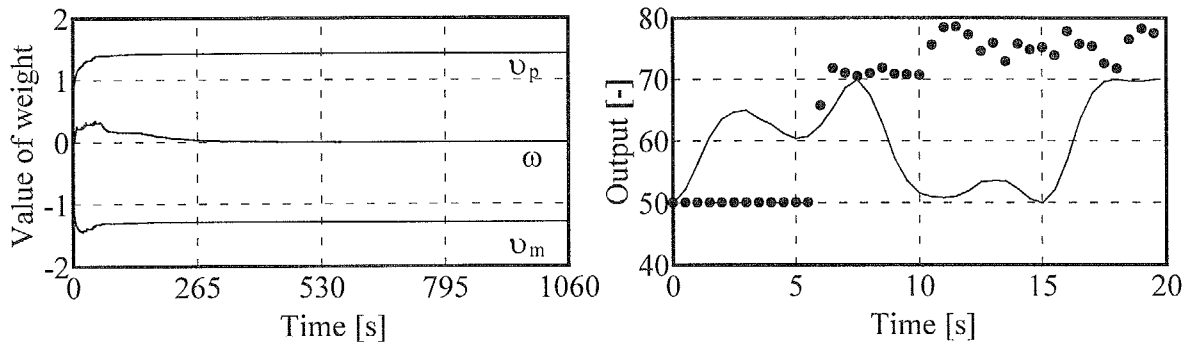


図 3-12 対応策を施していない場合

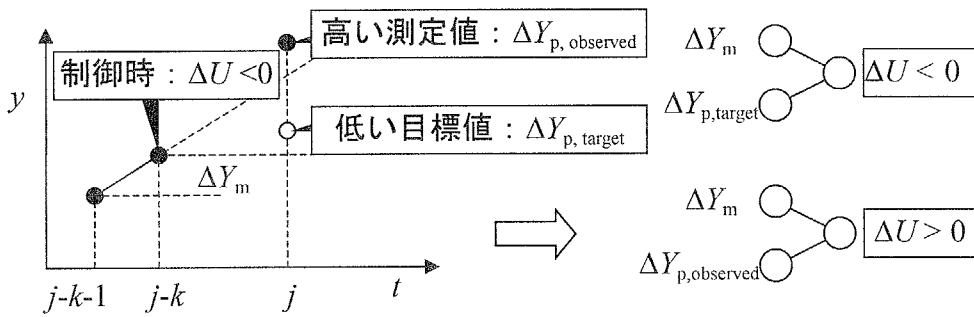
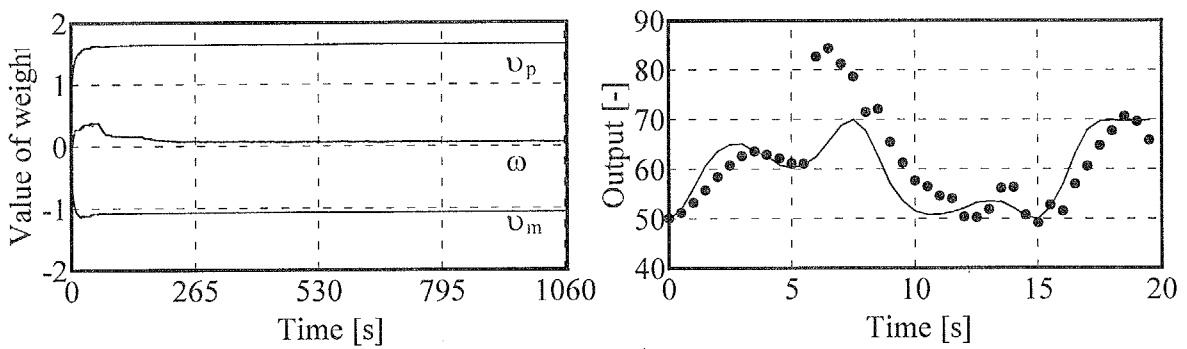


図 3-13 特性変化への対応策



(b) 対応策を施した場合

図 3-14 重みの経時的変化と最終的な制御結果

3.9. 第3章の結論

MIMO プロセス用コントローラである構造化 PENN コントローラを提案した。

- ・ SISO 用の PENN コントローラをプロセスの制御量と操作量の組み合わせの数だけ結合し、1対1対応のネットワーク構造とした。
- ・ 対象とする制御量と操作量の組み合わせ間に強い関係がある場合は入力層と中間層の結合重みと、中間層と出力層間の重みの両方を学習に用いる。
- ・ また、関係が弱いまたは未知である場合は入力層と中間層の結合重みのみを用いる。

学習済みのネットワークから干渉特性を取り出す方法を提案した。

- ・ 変数間の干渉特性を、ゲイン定数として評価する方法を提案した。
- ・ また、ゲイン定数の評価の際に得られる行列が正則かどうかを判断することでネットワークが不安定かどうかを判断できる。

構造化 PENN の状態を把握する指標を提案した。

- ・ ネットワーク重みから得られる行列が正則でないときは構造化 PENN が不安定であるとするものである。
- ・ オンライン学習によってネットワークが不安定な方向に学習が進んでいるかどうかを確認することが可能である。

変数対毎に遅れの特徴が異なるプロセスに対応するネットワーク構造を提案した

- ・ 変数対毎にそれぞれ異なる目標値の時間的な位置を指定できる構造である。

ニューラルネットワークで使用する誤差評価関数に不感帯を導入し、

- ・ 比較的小さなノイズが含まれている場合に対応した。

プロセス特性変化時の経験データの悪影響を防ぐための対応策を提案した。

- ・ 特性変化時の経験データを除外する学習アルゴリズムを提案した。
- ・ 非常に少ない演算の追加のみで済み、かつ効果的な方法であることを確認した。
- ・ 同時に、大きなノイズが混入したデータを除外する方法でもある。

4. 構造化 PENN コントローラの結晶成長プロセスへの応用

第3章で提案した構造化 PENN コントローラの制御応用例として、結晶成長プロセスを取り上げる。結晶成長プロセスは5入力5出力プロセスで制御量が温度で操作量が電気ヒーター出力であり、すべての変数対間の遅れ特性が同じプロセスである。このプロセスに対して、ネットワーク重みからのゲイン定数の評価とノイズ対策の有効性を評価する。

4.1. 結晶成長プロセスのシミュレータ

アントラセンの結晶成長プロセスを図 4-1 に示す。このプロセスは伊藤と加藤(1994)により実験装置及びシミュレータが作成されたもので、干渉を持つ他変数プロセスである。図に示すように、5の電気ヒーターが試料管内の温度分布を制御する。アントラセンの温度は5つの熱電対によって測定され、その位置は試料中央部でヒーターと同じ高さにあるものとした。試料管は直径7mmで高さ36mmである。まずアントラセンが融点以上に加熱され、その後底の方から徐々に冷却されて結晶が成長する。このときの望ましい温度分布は図の右側にあるグラフで示すもので、ある温度勾配 G を持つ温度プロファイルが速度 R で移動するように制御を行うことが目的である。コンピューターによるシミュレーションを行い、シミュレータでは試料管を高さ方向に50等分し、それぞれの部分において熱収支を計算した。

$$\frac{\partial x(z,t)}{\partial t} = \frac{\partial^2 x(z,t)}{\partial z^2} + u(z,t) - v(z,t) \quad (0 \leq z \leq 1) \quad (4-1)$$

$$\text{with } x(z,0) = x_0(z) \text{ and } \left. \frac{\partial x}{\partial z} \right|_{z=0} = \left. \frac{\partial x}{\partial z} \right|_{z=1} = 0 \quad (4-2)$$

x, z, t, u, v はそれぞれ 温度, 高さ位置, 時間, 電気ヒーターからの無次元化した熱量, 熱損失である。これらの方程式を刻み時間 6s で積分してシミュレータを作成した。このシミュレータは線型であるが、伊藤と加藤によって実験装置とよく一致していることが確認されている。

制御間隔を 30s とし、遅れ時間が 30s 以下の小さいものであると想定した。初期値として温度は 491.15K、ヒーター出力を 4.79 J/m²•s とした。一回の試行について 2100s かかるものとした。

また、このプロセスでは5カ所の温度の間に干渉がある。電気ヒーターからの熱は同じ位置にある温度だけでなくその周りの温度をも変えるためである。

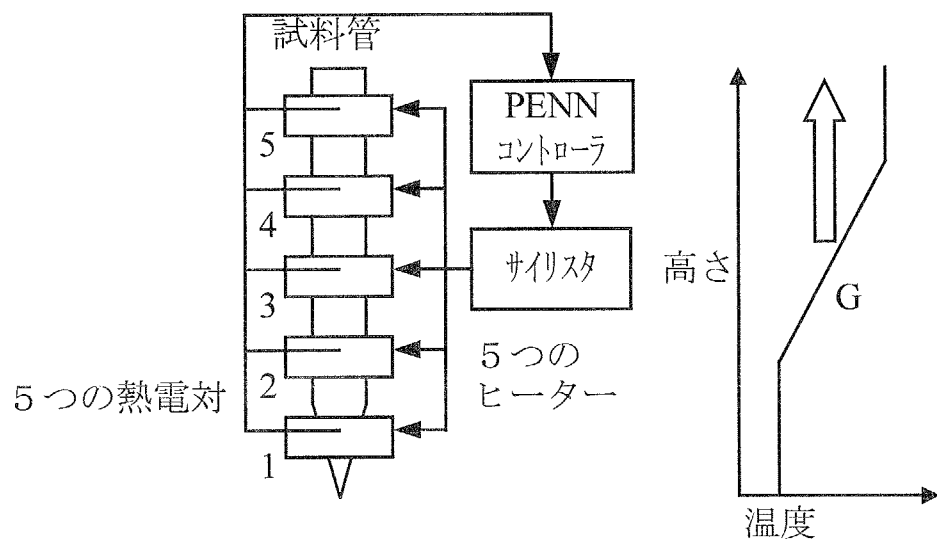


図 4-1 結晶成長プロセス

4.2. 制御方法と結果

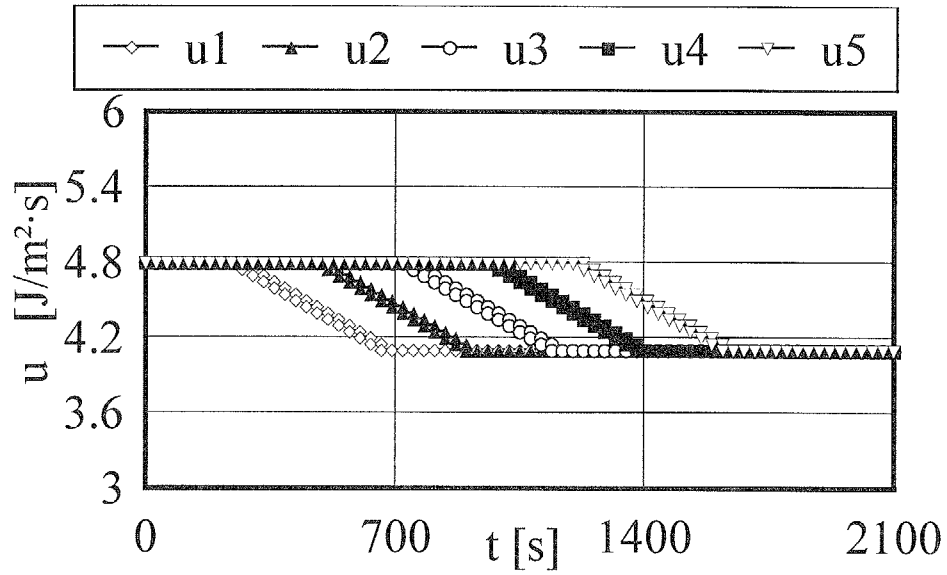
結晶成長プロセスは5入力5出力のプロセスであるため、構造化 PENN は入力ユニットが10、中間ユニットが25、出力ユニットが5となる。また、重みの個数は入力層と中間層間が50で中間層と出力層間が25である。また、PENN の入力変数のうちの1つである ΔY_{rp} の算出ではすべての温度について $k=1$ すなわち次のサンプリング間隔後の値との差を採用した。これは装置が小さく、制御間隔である30s以内でヒーター操作の結果が温度変化として十分に現れるからである。

ヒーターの出力を上げると温度が上昇することから、操作量と制御量との関係は正の相関を持つと言える。またヒーターはその同じ高さ位置にある熱電対に対して強く影響を及ぼすことから、これらの変数の組に対して制御ポリシーとしては正の相関を表すポリシーを適用した。他の組に対しては入力層と中間層の結合重みに対してのみ正の相関を表すポリシーを適用した。制御前のポリシー学習では2番から5番のポリシーデータを各1回学習させた。

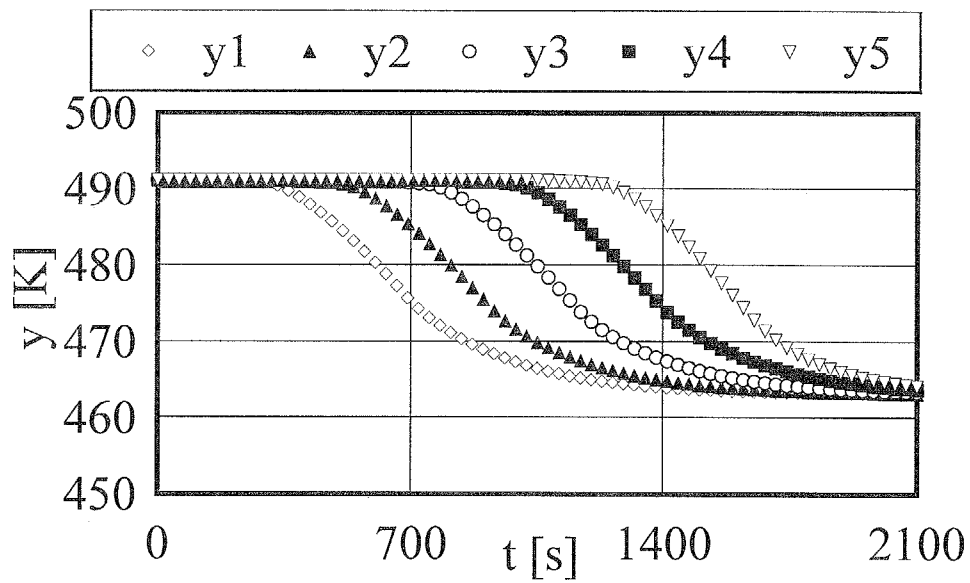
制御を開始する前に学習させる運転データとして図4-2のデータを使用した。このデータはヒーター出力を直線的に減少させた後一定に保つ操作を各ヒーターに対して時間をずらして行って得られたものである。この操作の結果、各温度測定点での温度はなだらかに下降した後一定になる結果が得られた。1試行2100sで制御間隔が30sなので経験データとしては $2100/30 - 1 = 69$ 含まれることになる。

経験学習の際には、少ないデータを有効に利用するために同じデータを再利用した。図4-3に示すように、学習する経験データの7点前までさかのぼってそれらを3回学習させたため、同じデータを $8 \times 3 = 24$ 回学習させたことになる。制御開始後のオンライン学習でも同様に7点さかのぼり、それらを各点毎に3回学習させた。この経験学習を同じ運転データについて3試行分を行った。

制御中のポリシー学習は制御間隔毎に2から5番のポリシーを1回学習させた。



(a) ヒーターからの熱



(b) 測定された温度変化

図 4-2 制御前学習用の運転データ

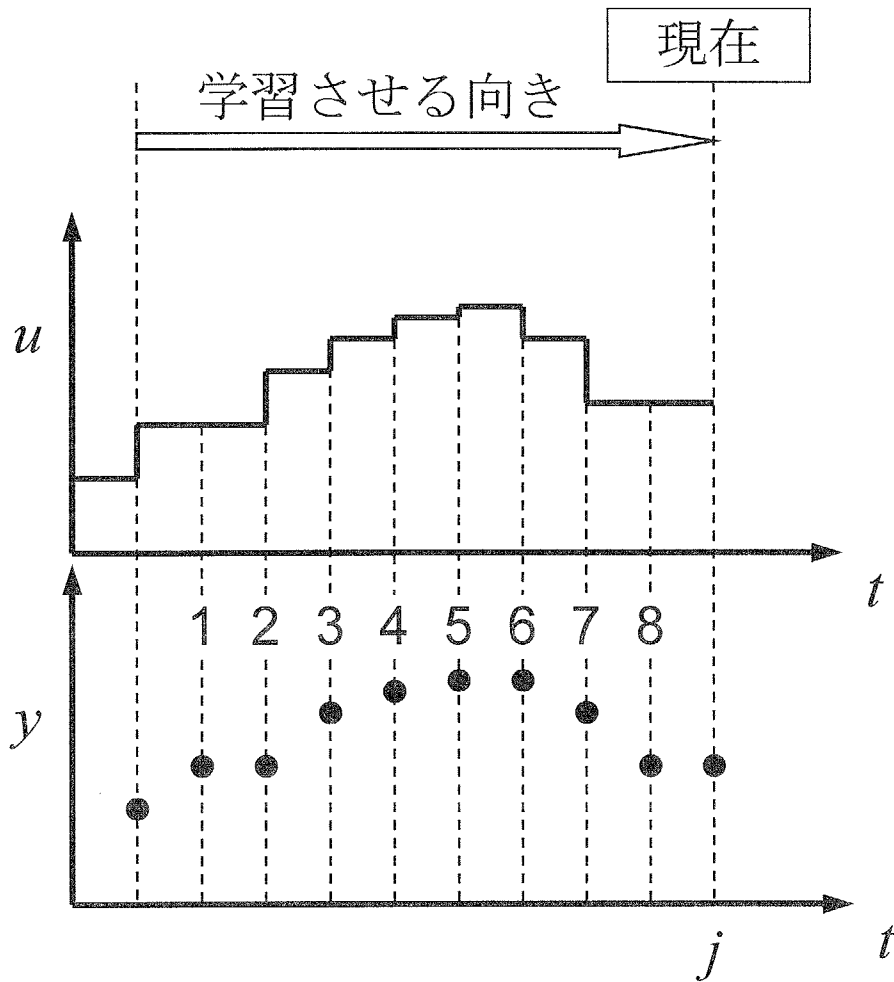


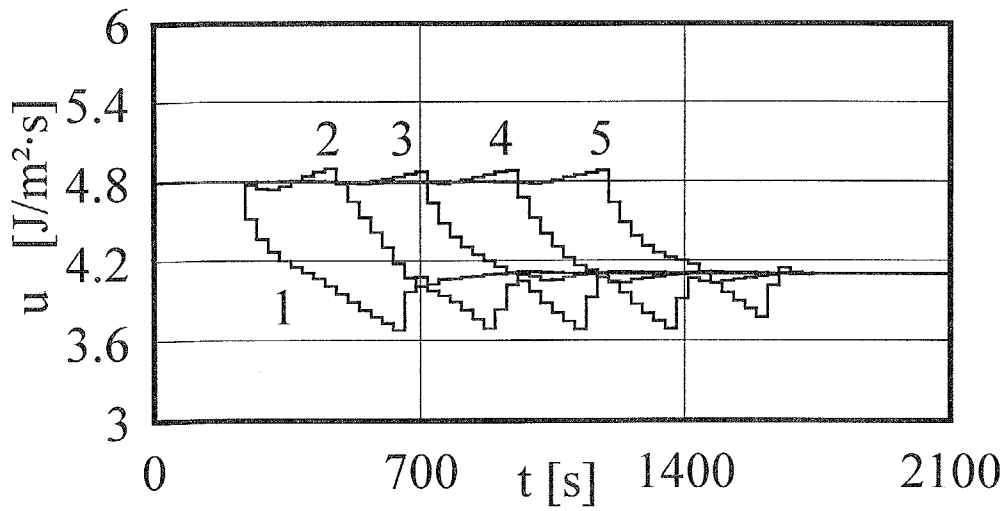
図 4-3 経験データの再利用

制御開始 1 試行目の制御結果を図 4-4 に示す。一番底にあるヒーター及び熱電対から順番に 1 から 5 の番号が割り振られている。温度を示すグラフ (b) の実線で示されている目標温度に沿って制御を行い、点が測定された温度である。

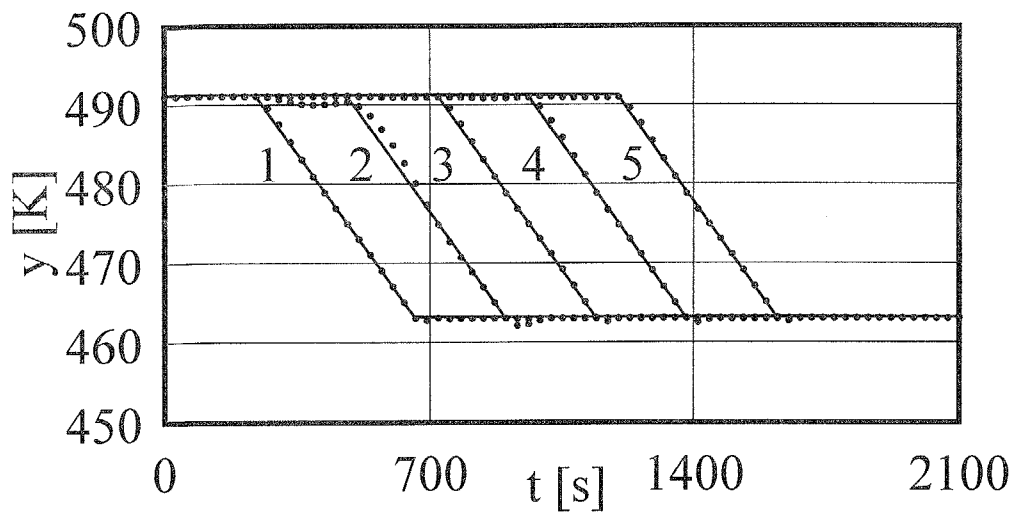
2 試行目の制御結果 (図 4-5) と比較すると、温度測定点 2 について 350s 付近で目標からのずれが確認できる。これは、この段階ではまだ構造化 PENN がプロセスの干渉特性を学習していないといえる。2 試行目の結果では目標からのずれはほぼ無くなり、これ以降の結果も同様であった。また、図 4-5 において、ヒーター 1 の出力が下げられたときに、熱電対 2 の温度が下がってしまわないようにヒーター 2 の出力が上げられている。このことからネットワークが干渉特性を学習していることがわかる。

このときの構造化 PENN 結合重み (75 個) の推移は図 4-5(c)に示すようにほとんど変化していない。これは結晶成長プロセスが線型プロセスであるため、構造化 PENN がオンライン学習で重みを更新せずに制御を行っているといえる。

この結果より、構造化 PENN は結晶成長プロセスの特性を少ない試行回数で学習することができたといえる。

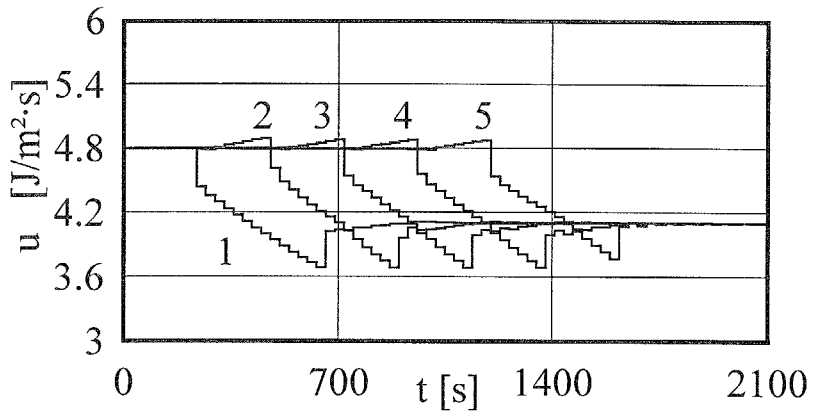


(a) ヒーター出力

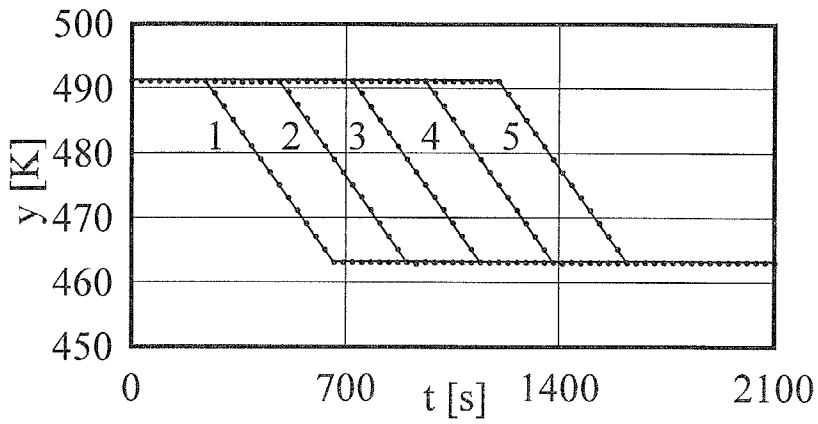


(b) 温度

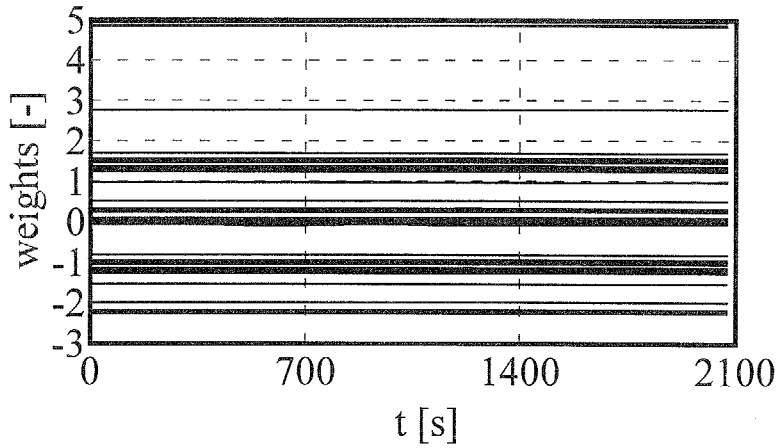
図 4-4 1 試行目の制御結果



(a) ヒーター出力



(b) 温度



(c) ネットワーク結合重み

図 4-5 2 試行目の制御結果

4.3. ノイズの考慮

観測ノイズを想定した場合の制御結果を示す。想定したノイズは $\pm 1\text{K}$ の幅を持つホワイトノイズで、シミュレータの出力温度に加えることでシミュレートした。制御前学習での運転データにもノイズを加えている。図 4-6 は2回目の試行結果である。この結果から、構造化 PENN がノイズをも学習してしまい、温度が目標値から外れてしまっている。このときポリシーデータは制御間隔毎に学習させている。

第3章で提案したノイズ対策を利用した結果が図 4-7 である。提案した方法は有効に機能し、良好な制御結果を得ることができた。

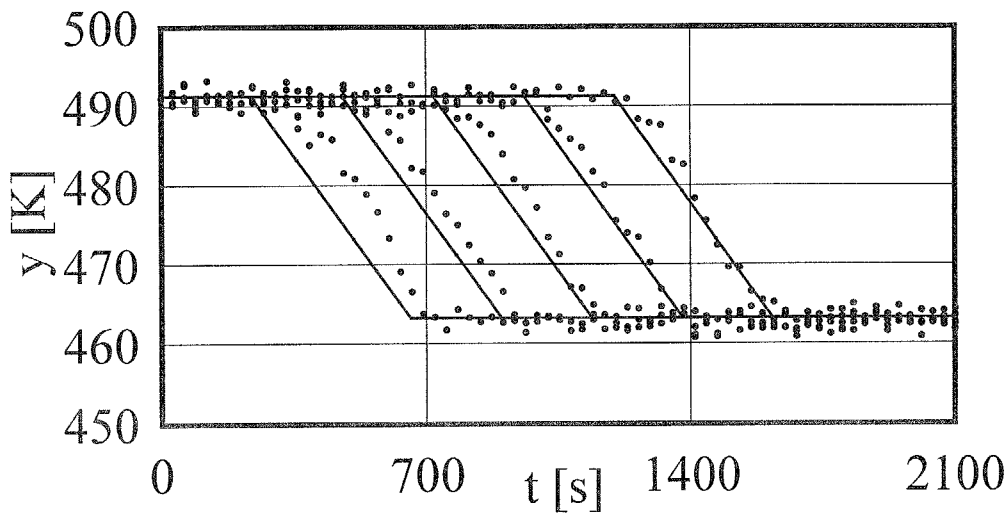


図 4-6 温度測定時にノイズが入った場合

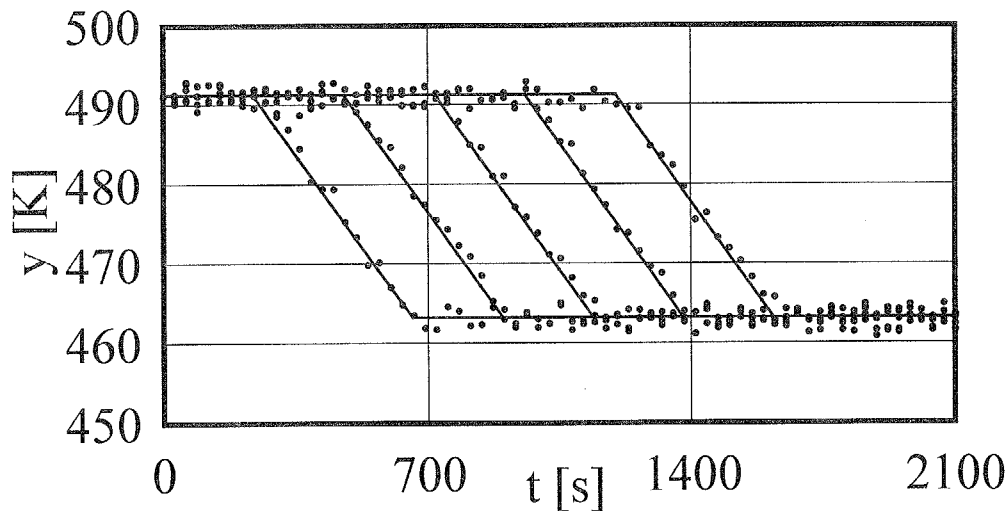


図 4-7 ノイズ対策後の制御結果

4.4. ゲインの推定

結晶成長プロセスの制御で得られた構造化 PENN の重みに対して、第3章で提案したゲイン定数の評価法を適用し、表 4-1 の結果を得た。ヒーター1に対して温度1のゲイン定数が 4.862、ヒーター2に対して温度2が 4.871 というように、プロセスは同じ位置にある電気ヒーターと温度との間の強い関係が大きな値のゲイン定数として現れており、次に大きい干渉としてはヒーター4が温度5に及ぼす影響が-0.113 と大きいものとなっていることがわかる。この結果はポリシー学習を行った場合でも行わなかった場合でも同様に得られた。

表 4-1 結晶成長プロセスでの評価ゲイン定数

ヒーター u \ 温度 y	1	2	3	4	5
1	4.862	-0.070	0.038	-0.025	0.006
2	0.061	4.871	-0.102	0.076	-0.045
3	0.013	0.042	4.848	-0.112	0.062
4	0.006	0.034	0.025	4.814	-0.113
5	-0.031	0.010	-0.074	-0.069	4.729

4.5. 第4章の結論

制御の応用例として結晶成長プロセスの制御を行った。

- ・遅れがないプロセスとして結晶成長プロセスの制御を行い、多変数プロセスの制御が可能であることを示した。

線型プロセスに対しては構造化 PENN はプロセスを学習できたため、オンライン学習での重みの更新が行われずに良好な制御結果が得られた。

- ・観測ノイズを想定した場合には、提案したノイズ対策法が有効に機能して制御が可能となった。

- ・学習済みのネットワーク重みから、結晶成長プロセスの干渉特性を得た。

5. 構造化 PENN コントローラの重油加熱炉への応用

構造化 PENN コントローラの応用として、重油加熱炉の制御を行った。重油加熱炉は、操作量が燃料ガス及び空気供給量で、制御量が出口酸素濃度と出口重油温度という、それぞれ異なった種類の変数で変数対毎に遅れ特性が異なっている。また、放射伝熱による加熱が行われており、非線形プロセスである。

このプロセスに対しては、結晶成長プロセスと同様に、ゲイン定数の評価とノイズ対策の評価を行う。さらに、プロセス特性が制御中に変化する場合は想定し、第3章で提案した特性変化への対応策の有効性を評価する。また、行列式を利用したネットワークの安定性の評価も同時に行う。

5.1. 重油加熱炉のシミュレータ

制御対象である重油加熱炉を図 5-1 に示す。被加熱流体（重油: $n\text{-C}_{10}\text{H}_{22}$ ）は上部熱交換器に入り、熱せられた後に下部出口より流出する。装置下部では燃料ガスと空気が供給され、バーナーにより装置壁と管を加熱し、生じた高温ガスが上部熱交換器にて使用される。つまり装置下部では輻射による加熱、上部では対流による加熱が行われる。重油の沸騰は起こらないものとする。

制御の目標は燃料及び空気量を適正にしつつ重油の出口温度を設定温度にすることである。そのため制御量として被加熱流体の出口温度と排気ガスの酸素濃度、操作量として燃料供給量、空気供給量を用いる。排気ガスの酸素濃度は、空気供給量を適正にするためである。

シミュレータは加熱炉を図 5-1 に示すように大きく 4 つの部分に分けて作成した。熱交換器部分 4 の形状は縦横 6 m 高さ 2 m とし、0.0529 m の輸送管が 200 m 通っている。管内の重油はプラグフローで沸騰はしないものとする。

3 つに分けた放射伝熱部分 (1~3) を説明する。それぞれ縦 10m 横 10m、下から順に高さ 4m, 7m, 2m である。重油管は四方の壁に配置する。一番下の部分は炎からの放射伝熱を考慮し領域を小さくすることで温度が高い部分になるようにした。各部分のシミュレーションは図 5-2 のように簡略化して行い、壁を内側表面とその外側部分に分けて考える。内側表面部分でガス及び炎からの放射熱を受け取り、残りの部分には伝導伝熱が行われるものとした。表面部分は壁厚み 0.2 m の 10% とし、外側

部分から装置外部への熱損失を考える。

各部分の温度を完全混合槽モデルとして以下の連立微分方程式で記述し、ルンゲクッタ法により解の計算を行った。内側と外側の壁温度についてそれぞれ3つ、重油温度について4つ、熱交換器内のガス温度についての11元連立微分方程式である。

Inside Wall

$$V_w C_{p,wall} \frac{dT_{w1}}{dt} = F_g H_g (T_{w,in} - T_w) - C_{H,w} (T_w - T_{wo}) A_w - C_{H,oil} (T_w^4 - T_{oil}^4) A_{oil} + H_{comb} \quad (5-1)$$

Outside Wall

$$V_{wo1} C_{p,wall} \frac{d\left(\frac{T_{wo} + T_w}{2}\right)}{dt} = C_{H,w} (T_w - T_{wo}) A_{w1} - C_{H,ext} (T_{env}^4 - T_{wo}^4) \quad (5-2)$$

Oil

$$V_{oil} C_{p,oil} \frac{dT_{oil}}{dt} = F_{oil} H_{oil} (T_{oil,in} - T_{oil}) + C_{H,oil} (T_w^4 - T_{oil}^4) A_{oil} \quad (5-3)$$

Heat Exchange

$$V_g C_{p,gas} \frac{dT_g}{dt} = F_g H_g (T_{w3} - T_g) - U (T_g - T_{o4}) A_{HE} \quad (5-4)$$

$$V_{o4} C_{p,oil} \frac{dT_{o4}}{dt} = F_o H_o (T_{o4} - T_g) + U (T_g - T_{o4}) A_{HE} \quad (5-5)$$

温度に関する4次式となっており、非線形プロセスである。

燃料ガスは組成一定とし、常に完全燃焼していると仮定する。排気ガスの酸素濃度については装置を3分割し、各部を排気ガス成分（酸素、窒素、二酸化炭素、水）について完全混合槽モデルで記述しルンゲクッタ法により計算を行った（12元連立微分方程式）。

シミュレーションで使用した条件を表5-1にまとめた。

表 5-1 シミュレーション詳細

パイプ径	直径 0.0529m			
パイプ長	部分 4 200 m	部分 3 1200×4 m	部分 2 2100×4 m	部分 1 300×4 m
燃料ガス	C ₂ H ₆			
重油供給量	40.3 mol・s ⁻¹			
エネルギー利用割合	燃焼熱 100%	重油加熱 91.5%	排気ガス 5.9%	熱損失 2.6%
ルンゲクッタ刻み	0.01 s			
制御間隔	30.0 s			
燃料ガス初期値	10.0 mol・s ⁻¹			
供給空気量初期値	114.3mol・s ⁻¹			
重油温度初期値	791.258K			
出口酸素濃度初期値	0.0321843			
重油温度最低値	298.15K			
最高値	900.0K			
酸素濃度最低値	0.0			
最高値	0.21			
燃料ガス最大量	30.0 mol・s ⁻¹			
変更最大量	10.0 mol・s ⁻¹			
供給空気最大量	342.9mol・s ⁻¹			
変更最大量	114.3mol・s ⁻¹			

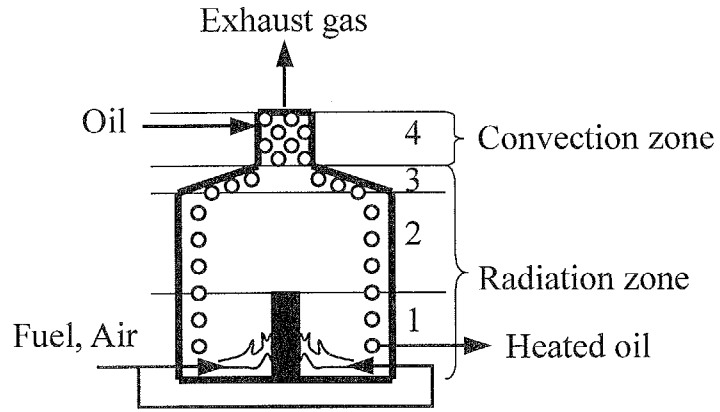


図 5-1 加熱炉装置図

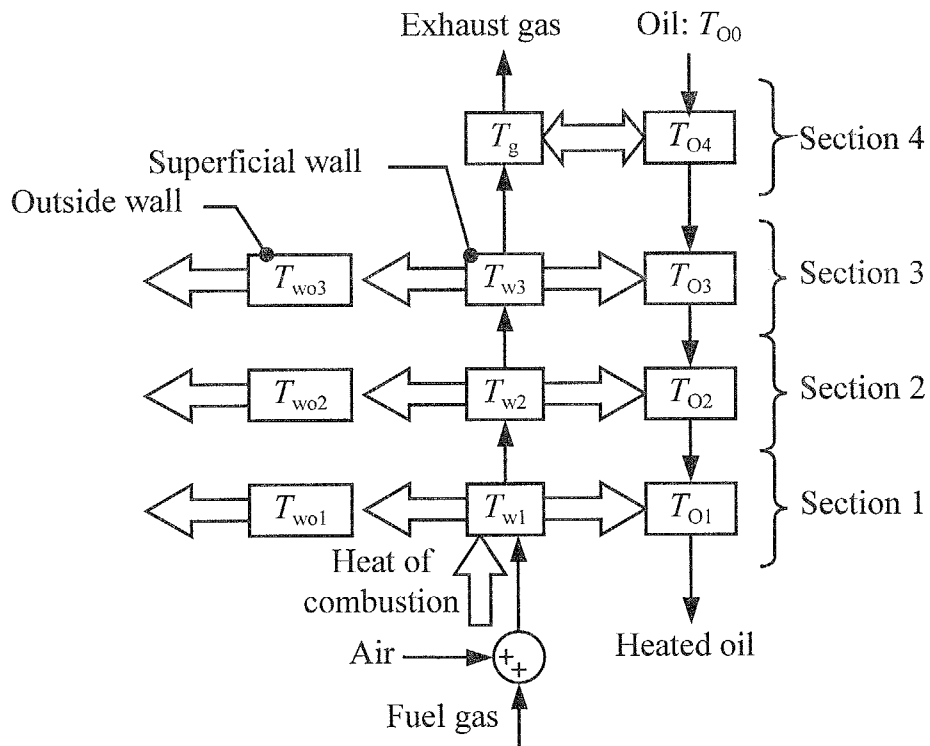
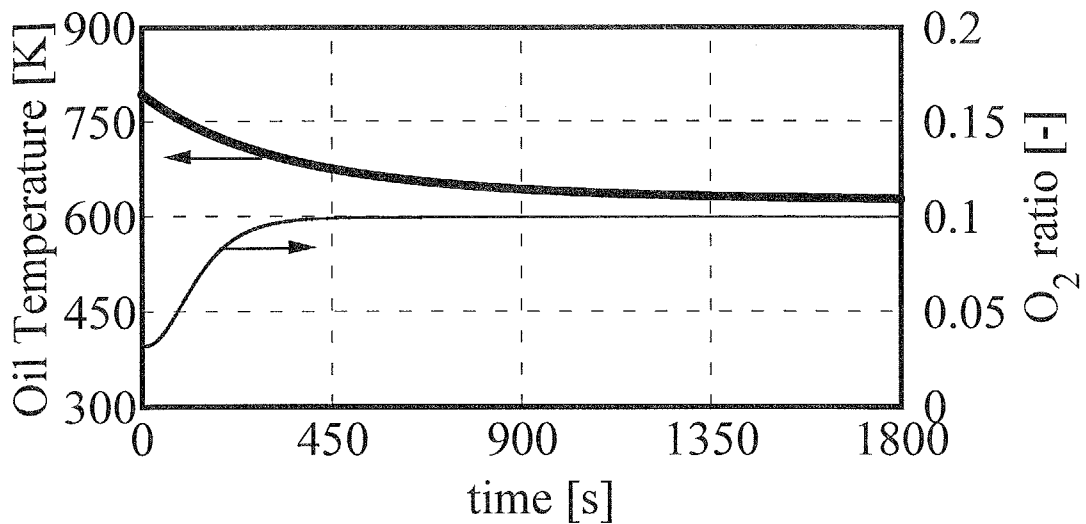
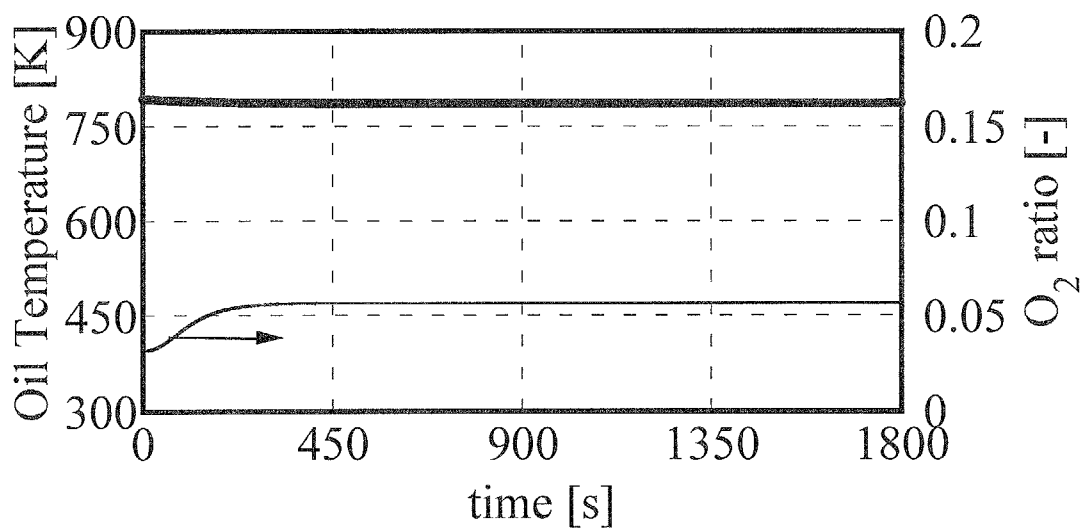


図 5-2 シミュレータの詳細図

定常状態から燃料の供給量をステップ状に変更した場合の温度および酸素濃度の変化を図 5-3 に示す。温度の応答は酸素濃度の応答に比較して長い遅れを伴っていることがわかる。また、燃料供給量が各制御量に与える影響と、空気量を与える影響にも違いがある変数間干渉を持ったプロセスである。シミュレータ式では燃料ガスと温度の積の項、および温度の 4 次項があるため、このプロセスが非線型プロセスであるといえる。



(a) 燃料供給量を $10 \text{ mol}\cdot\text{s}^{-1}$ から $6 \text{ mol}\cdot\text{s}^{-1}$ に変更



(b) 空気供給量を $114 \text{ mol}\cdot\text{s}^{-1}$ から $134 \text{ mol}\cdot\text{s}^{-1}$ に変更

図 5-3 重油加熱炉のステップ応答

5.2. 制御方法と制御結果

この重油加熱炉は2入力2出力プロセスであり、ループごとに遅れの特性が異なるため、使用した構造化ネットワークは図 3-10 に示す構造を持つ。ステップ応答結果では、重油温度の応答は1次遅れプロセスに近いが、酸素濃度の応答は無駄時間+時間遅れの特性を持つプロセスのものである。PENN の遅れ対応パラメータ k の値は k サンプル間隔後の目標値の値を採用することを指定するので、その時点で十分な制御量の変化が得られる必要がある。無駄時間がある場合、小さい k の値では十分大きな制御量変化が得られない。よってこのプロセスでは酸素濃度の k の値として 3 を使用した。また、1次遅れ応答を示す重油温度に対しては第3章での議論から、3以下の値を用いればよい。

燃料供給量を増加させると重油温度は上がり酸素濃度は低下する。また空気供給量を増加させると重油温度は下がり酸素濃度は上昇する。よって制御ポリシーとしては「燃料供給量-重油温度」には正の相関のポリシー、「燃料供給量-酸素濃度」には負の相関のポリシー、「酸素供給量-重油温度」には負の相関のポリシー、「酸素供給量-酸素濃度」には正の相関のポリシーを適用した。結晶成長プロセスと異なり、加熱炉プロセスでは特に強い関係を持つ操作量と制御量の組み合わせが未知であるため、中間ユニットと出力ユニット間の重みを更新する制御ポリシーは使用しない。制御ポリシーの学習は制御間隔ごとに1回ずつ行った。

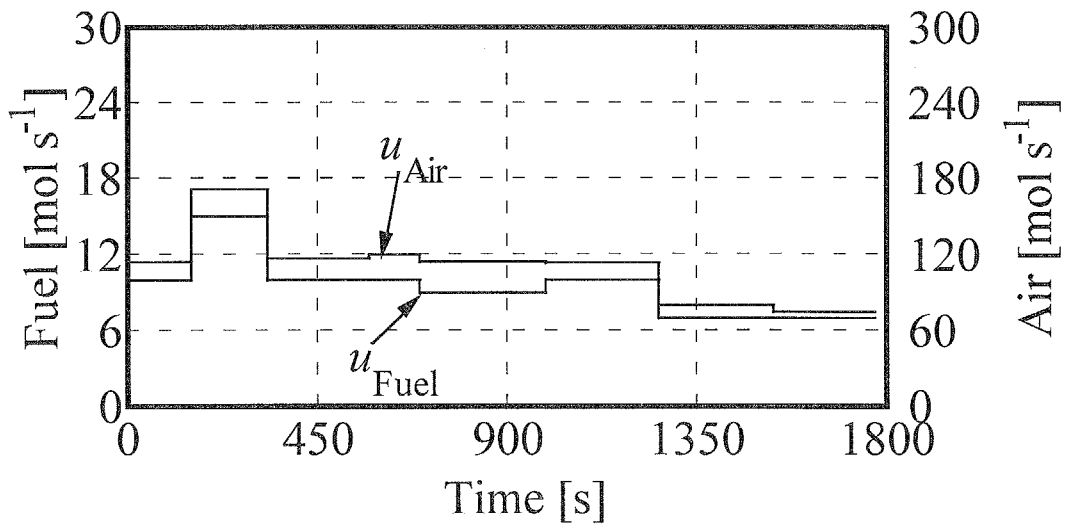
制御前学習用の運転データとして図 5-4 のデータを用いた。経験学習の際には6点前までさかのぼり、それらを3回学習させたため、同じデータを $6 \times 3 = 18$ 回学習させたことになる。制御開始後のオンライン学習でも同様に6点さかのぼり、それらを各点毎に3回学習させた。この制御前経験学習を同じ運転データについて3試行分を行った。

目標値は次のように設定した。酸素濃度の目標値は制御が行われているかどうかを調べる目的で正弦波を与えており、過剰空気量 10%での酸素濃度を初期値にしている。

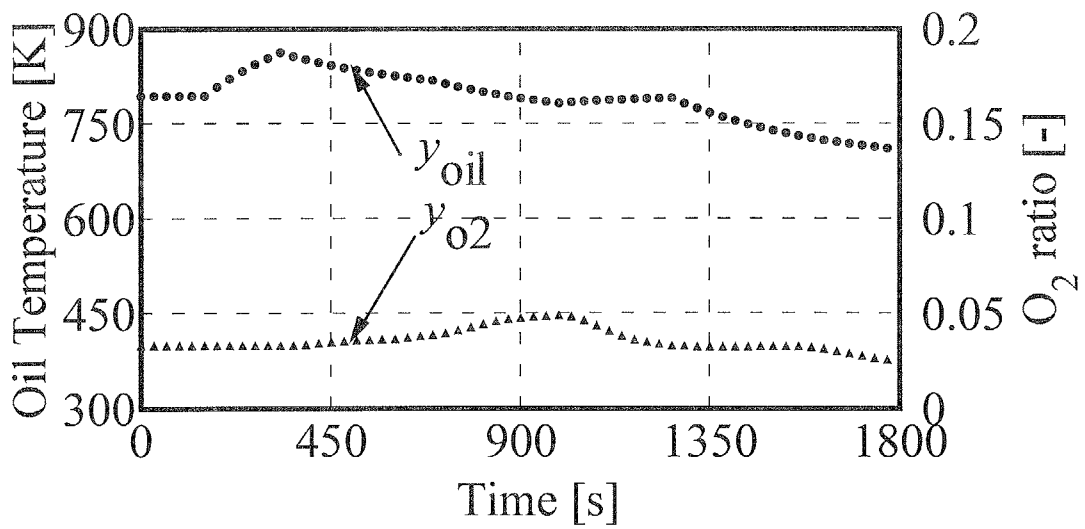
$$\begin{aligned} r_{\text{oil}} &= y_{\text{oil,initial}} + 60.0 \times (t-150.0)/180.0 && (t > 150.0) \\ r_{\text{oil}} &= y_{\text{oil,initial}} + 60.0 && (t > 150.0+180.0) \\ r_{\text{oil}} &= y_{\text{oil,initial}} + 60.0 - 40.0 \times (t-900.0)/180.0 && (t > 150.0+180.0) \end{aligned}$$

$$r_{\text{oil}} = y_{\text{oil,initial}} + 60.0 - 40.0 \quad (t > 900.0 + 180.0)$$
$$r_{\text{O}_2} = y_{\text{O}_2, \text{initial}} + y_{\text{O}_2} \times 0.3 + y_{\text{O}_2} \times 0.3 \times \cos(2\pi(t / 1800.0))$$

時間 t は実際の時間を 1800s で割った余りとした。すなわち重油温度は 1800s 毎に設定値変更がされることになる。



(a) 操作量

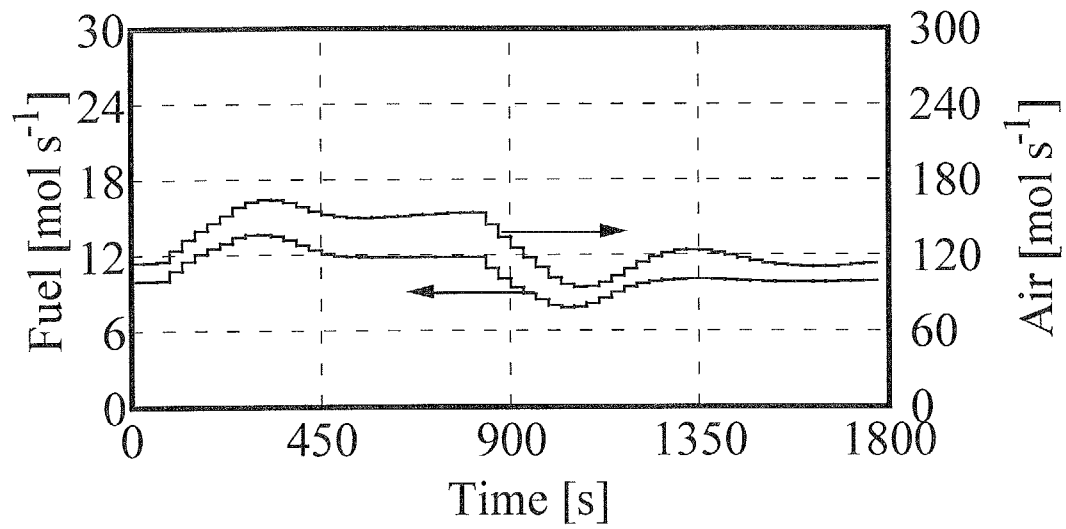


(a) 制御量

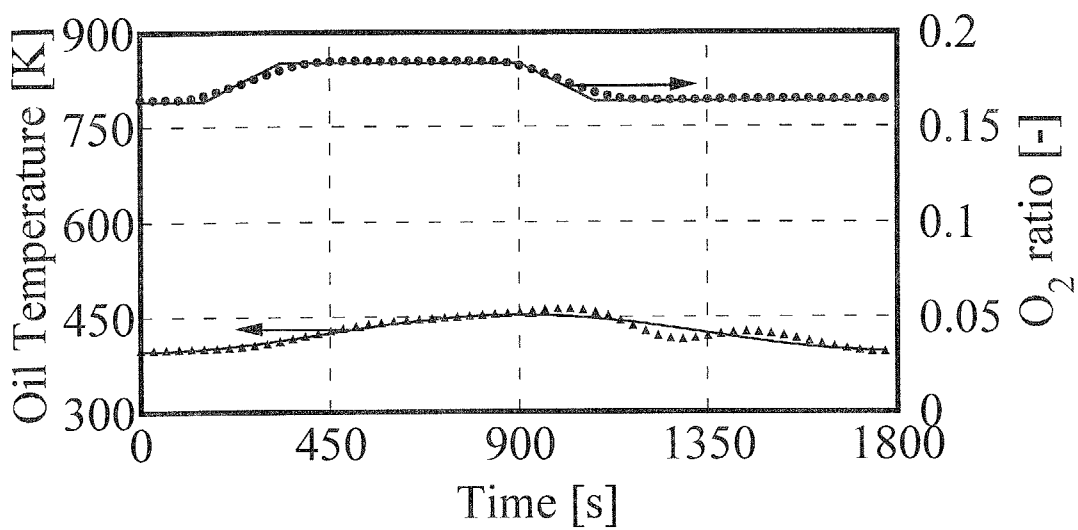
図 5-4 制御前学習用データ

図 5-5 に $k_{oil} = 3$, $k_{O_2} = 3$ での制御結果の経過を示す。開始直後は酸素濃度について目標値からのずれが見られるが大きく外れてはいない。これは制御前学習によって NN がプロセス特性を学習していることを示している。1800s 以降ではほぼ同じ制御結果となり、時間が経過するとともに制御成績は向上し 100 時間後の結果は図 5-6 のようになりそれ以降は制御成績には変化が見られなかった。このように、PENN での制御では制御前学習だけでなく制御中にも学習をしていくことで制御成績を上げることが可能であると言える。

また、図 5-6(c)には評価ゲインの推移を示した。これより構造化 PENN コントローラがオンライン学習によって結合重みを変更しつつ良好な制御を行っていることがわかる。

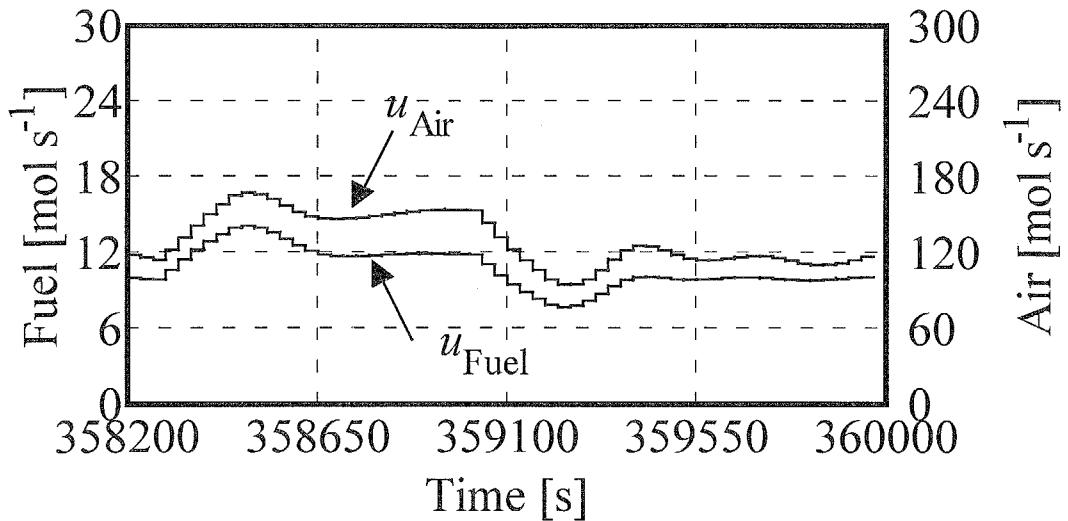


(a) 操作量

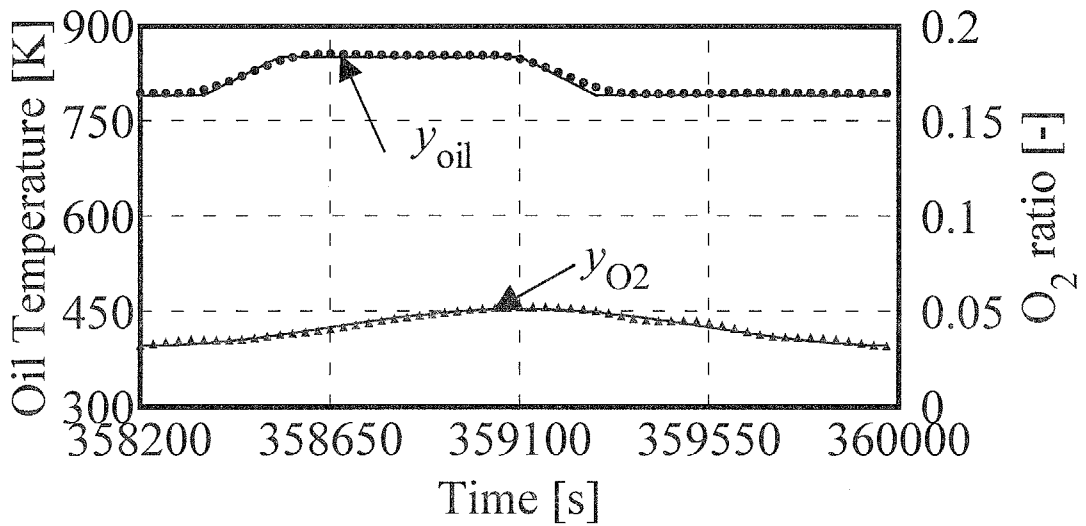


(b) 制御量

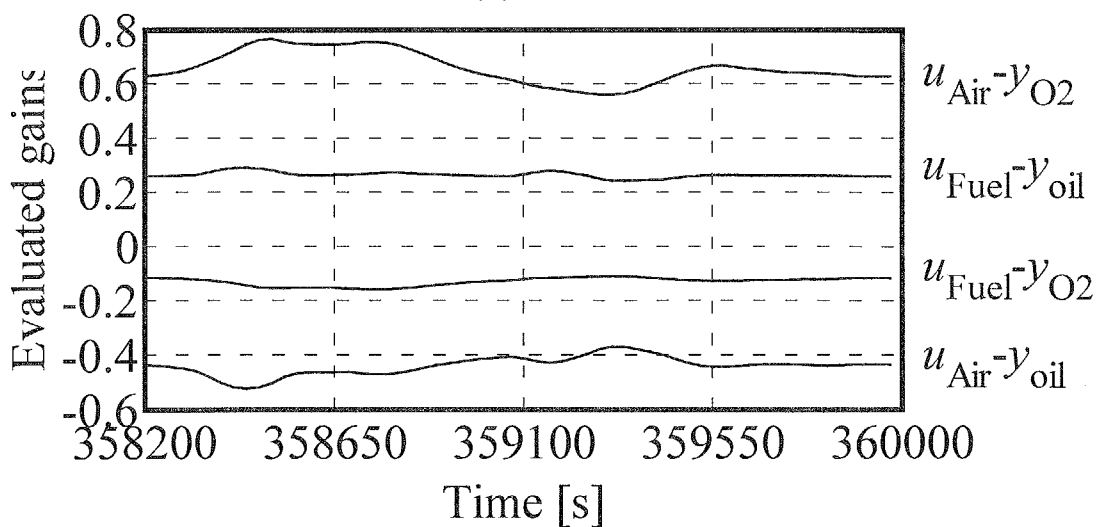
图 5-5 加熟炉制御結果



(a) 操作量



(b) 制御量



(c) 評価ゲイン定数

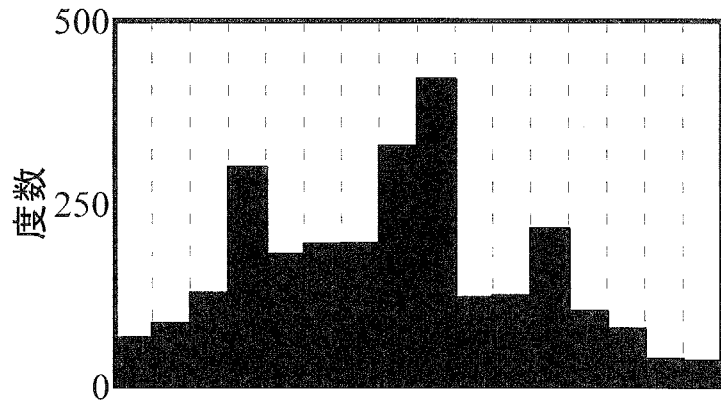
図 5-6 100 時間経過後の制御結果

5.3. ノイズの考慮

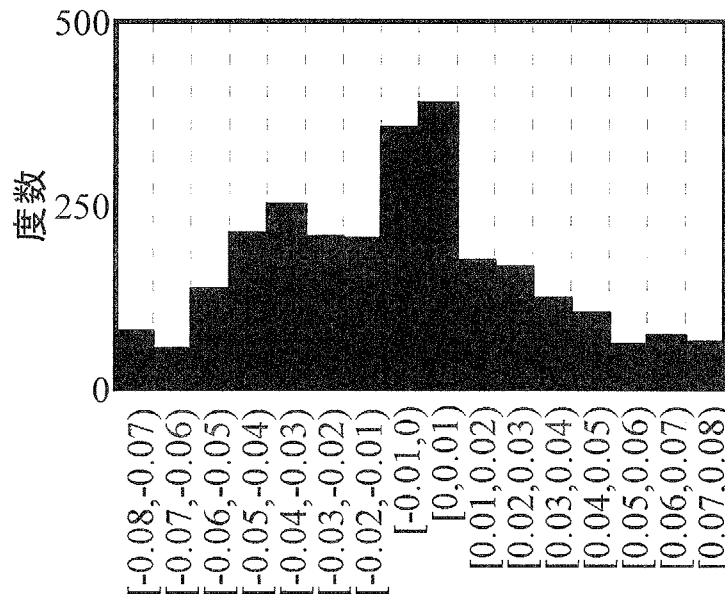
実際のプロセスに近い状況での制御を行うため、観測ノイズとシステムノイズを想定した場合について制御を行った。観測ノイズとしては重油温度には $\pm 5K$ 、酸素濃度については $\pm 0.25\%$ の幅の乱数をシミュレータ出力に加えた。システムノイズは、シミュレータ式の放射伝熱係数に数%の幅の変動を持たせた。ノイズ対策として第3章で説明した、学習時の誤差評価関数へ不感帯を導入する方法を用いた。不感帯の幅を決定するときには制御前経験データ学習における学習誤差の分布(図 5-7)を考慮した。ノイズを含む誤差範囲の頻度は多くなっていると考えられるため不感帯の幅を $-0.02 \sim +0.02$ とした。

まず観測ノイズのみを考慮する。ノイズ対策を施さない場合と施した場合の制御結果を図 5-8 に示す。ノイズ対策を施さない場合(a)は構造化 PENN がノイズの影響を受け制御中に酸素濃度が 0 になってしまい制御を中止した。学習での誤差評価関数に不感帯を導入した場合(b)、ノイズの影響は低減され、制御が不安定になるまでの時間が延びた。そこで追加のノイズ対策として、現在値を含む過去3点の制御量の平均値を現在の制御量とした移動平均法を用いた結果(d)のように満足のゆく制御結果が得られた。なお、移動平均法のみでは(c)のように制御不能になった。

不感帯と移動平均を導入した場合の、評価ゲイン定数の推移を図 5-9 に示す。それぞれのゲイン定数はなめらかに変化しており、ノイズの影響をそのまま学習していないと言え、ノイズ対策が有効に機能していると判断できる。

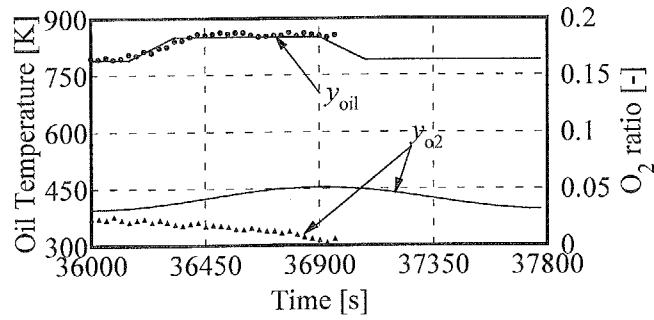


(a) 操作量：燃料

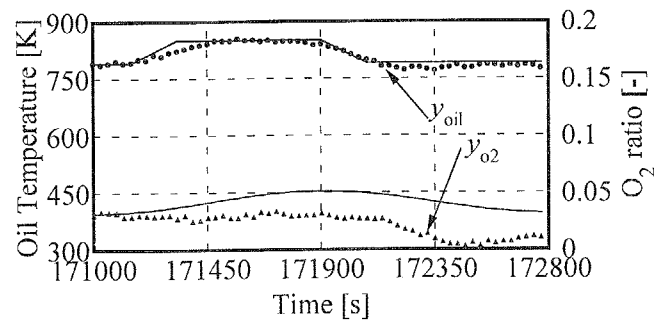


評価誤差
(b) 操作量：空気

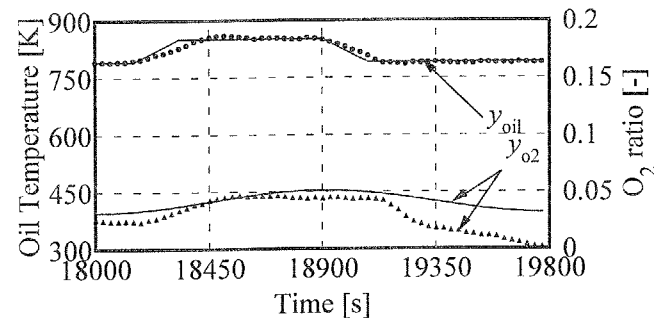
図 5-7 学習時の評価誤差の分布



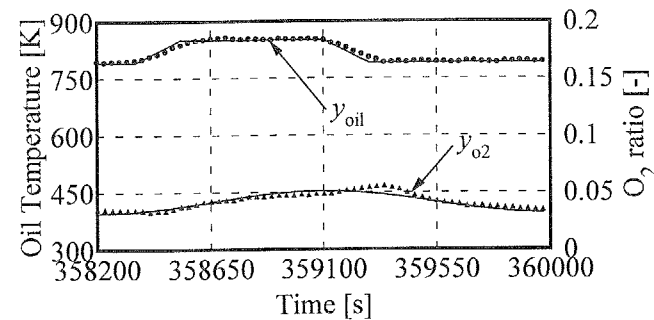
(a) ノイズ対策なし



(b) 不感帯の導入のみ



(c) 移動平均のみ



(d) 不感帯と移動平均

図 5-8 ノイズがある場合の制御結果

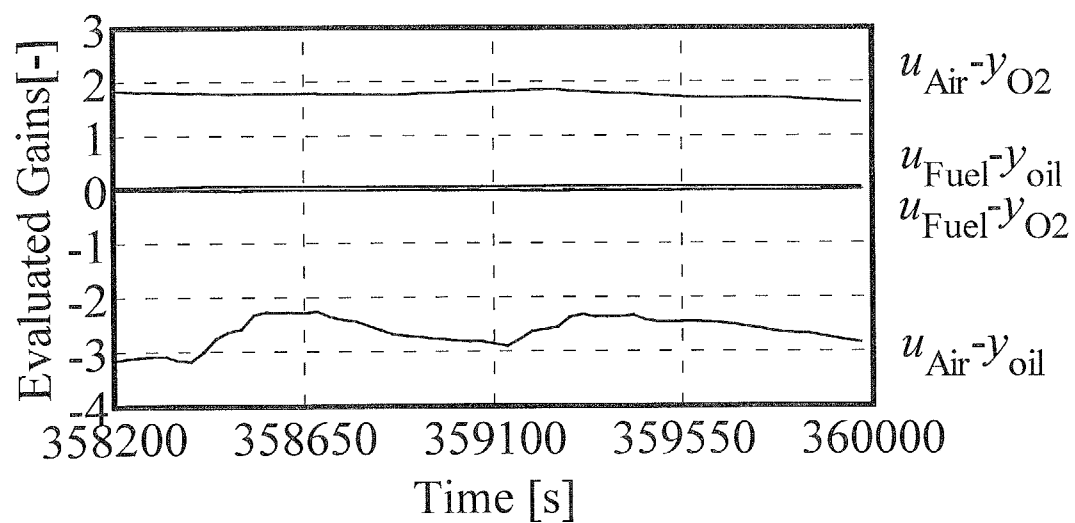


図 5-9 不感帯および移動平均導入時の評価ゲイン定数の推移

次に、システムノイズを考慮した。放射伝熱係数に与えるシステムノイズ幅と出口重油温度との関係を調べた結果、表 5-2 のようになった。プロセスが定常状態にあるときに放射伝熱係数の 1 % から 10 % までの変化を与えたときの 1 制御間隔後の温度変化を示している。

表 5-2 システムノイズが出口重油温度に与える影響

変動割合[%]	温度変化量[K]	変動割合[%]	温度変化量[K]
+1	+1.7	-1	-1.74
+3	+5.2	-3	-5.5
+5	+8.5	-5	-9.0
+10	+16.2	-10	-18.0

システムノイズ単独では±3%の幅まではノイズ対策と移動平均の併用で制御可能であった。システムノイズと観測ノイズを両方考慮した場合システムノイズとしては1%、観測ノイズは±5Kまでは制御可能であった。

5.4. ゲインの推定

表 5-3 と表 5-4 に 100 時間経過後の構造化 PENN より評価した、プロセス定数を考慮しない場合と考慮した場合のゲイン定数を示す。プロセス定数（操作量の最大変更量、制御量の最小値最大値）を考慮すると、この例ではガス 1 モル、温度 1 K、酸素濃度 1 % 単位のゲイン定数になる。

表 5-3 加熱炉プロセスの評価ゲイン定数

プロセス定数を考慮しないゲイン		
	y_{oil}	y_{o_2}
u_{Fuel}	0.262	-0.115
u_{Air}	-0.433	0.630

表 5-4 加熱炉プロセスの評価ゲイン定数

プロセス定数を考慮したゲイン		
	y_{oil}	y_{o_2}
u_{Fuel}	15.789	-2.405×10^{-03}
u_{Air}	-2.280	1.158×10^{-03}

プロセス定数を考慮しない場合では、[燃料-重油温度]のゲイン定数の方が[空気-重油温度]より絶対値が小さくなっている。しかし、プロセス定数を考慮した場合では逆に燃料が重油温度の与える影響のほうが大きくなっている。

5.5. 行列式によるネットワークの状態把握

不感帯のみを導入した場合の、構造化 PENN から得られる評価ゲイン行列の行列式の推移を図 5-10 に示す。この場合の制御結果 (図 5-8(b)) では時間 172470s で制御に失敗し、酸素濃度が 0 に非常に近くなかったが、行列式の推移を見るとそれ以前に時間 120000s 以降 0 付近の値になっており PENN の状態が不安定になっていることがわかる。このように、構造化 PENN はネットワークの状態が行列式によって把握できる特徴を持っていると言える。

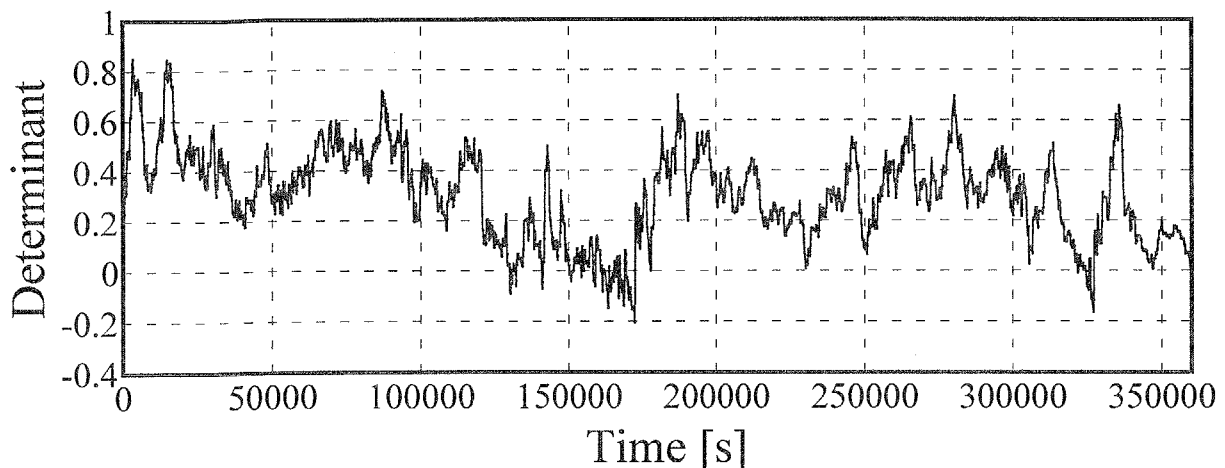


図 5-10 行列式の推移

5.6. 特性変化を伴うプロセス

化学プロセスの運転ではしばしばコントローラだけでは制御が追いつかない場面が起き、運転員が介入する必要がでてくる。プロセスが定常状態に近い状況ではコントローラのみで制御できる場合が多いが、原料の組成が切り替わるなどでプロセスの特性が大きく変わってしまうと既存の PID コントローラなどではパラメータを変更する必要がある。ニューロコントローラではプロセス特性の変化を自動的に

学習して制御を行うことが期待される。ここでは、プロセス特性の変化として重油加熱炉で燃料ガスの切り替えおよび供給空気温度の変化について、構造化 PENN の適用法を検討した。

5.6.1. 燃料組成の切替

燃料ガスとして、他の複数のプロセスから排出される排ガスと、純粋な燃料を併用していると想定した。そのため、他プロセスの運転状況によって導入する排ガスが切り替わるものとし、排ガスだけでは不足する熱量を純粋な燃料で補う。切り替わり方は短い時間に大きく変化する場合を想定した。

燃料ガスの持つ発熱量が燃料の切り替えによって大きくなると、コントローラの指令とは関係なく重油温度が上昇する。コントローラの指令が温度上昇を意図した場合では問題ないが、そうでない場合はオンライン学習によって操作量と制御量の全く逆の関係を学習することになる。

排ガスの組成として、比較的高カロリー、中程度および低カロリーの3種類を用意し、900sごとに「高カロリー $6\text{ mol}\cdot\text{s}^{-1}$ 」－「排ガスなし」－「中程度 $6\text{ mol}\cdot\text{s}^{-1}$ 」－「低カロリー $6\text{ mol}\cdot\text{s}^{-1}$ 」－と順次切り替える。

Table 5-4 排ガスの組成と発熱量

排ガス	組成	発熱量 $\text{J}\cdot\text{mol}^{-1}$
中程度	CH_4 50% N_2 50%	445335
低カロリー	CH_4 20% N_2 80%	171834
高カロリー	CH_4 80% N_2 20%	712536

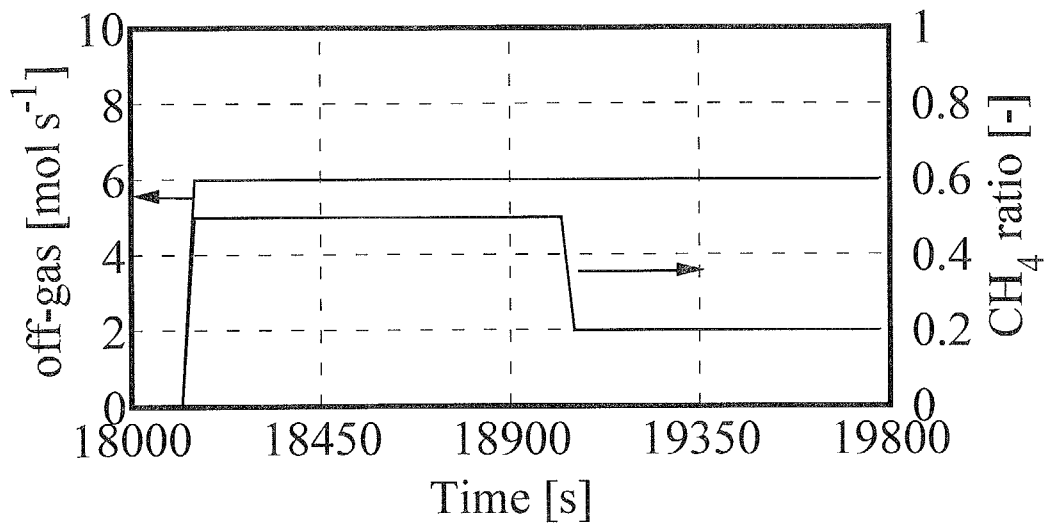
制御を開始して最初の 18000s は切り替えを行わずに制御を行った。最初の切り替えは 18150s で行い、このときの結果を図 5-11 に示す。18150s にて高カロリーの排ガスが導入され、重油温度が上昇し酸素濃度が減少したことが読みとれる。その 900s 後には排ガスの供給が無くなったために温度の下降と酸素濃度の上昇がみられる。

制御を続けていくと、図 5-12 に示すように制御不能に陥った。構造化 PENN で得られる行列式の値も 0 となりネットワークが不安定になっていることがわかる。

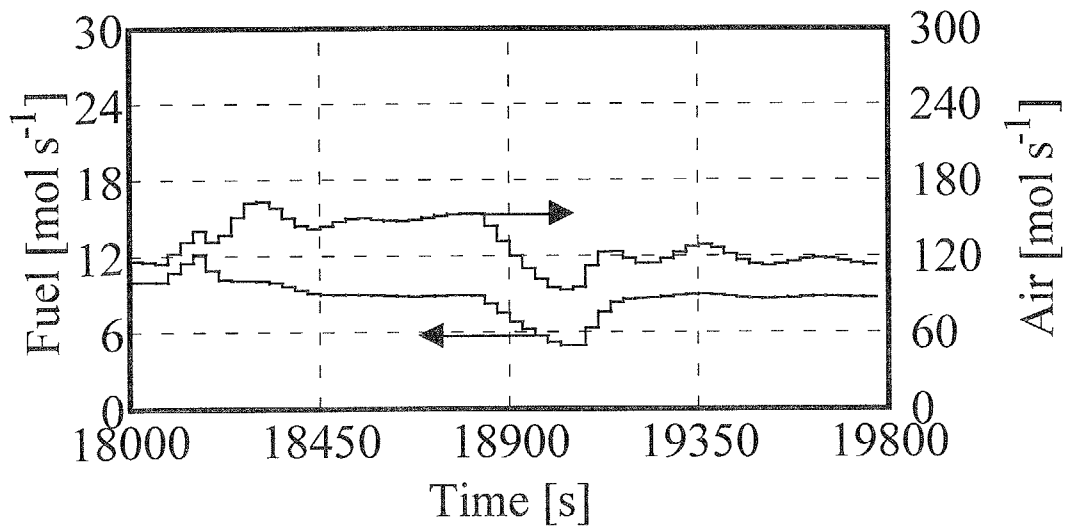
先に述べた対策を施した場合は図 5-13 のようになったが、その効果としては施

さない場合と比較して制御不能になるまでの時間が伸びたといえる。この燃料の切り替えはあらかじめ知らされておらず、しかもその変化も大きいものであるため先の対応策だけでは不十分であることがわかった。そこで他に切り替えに関して利用できる情報を利用することにした。

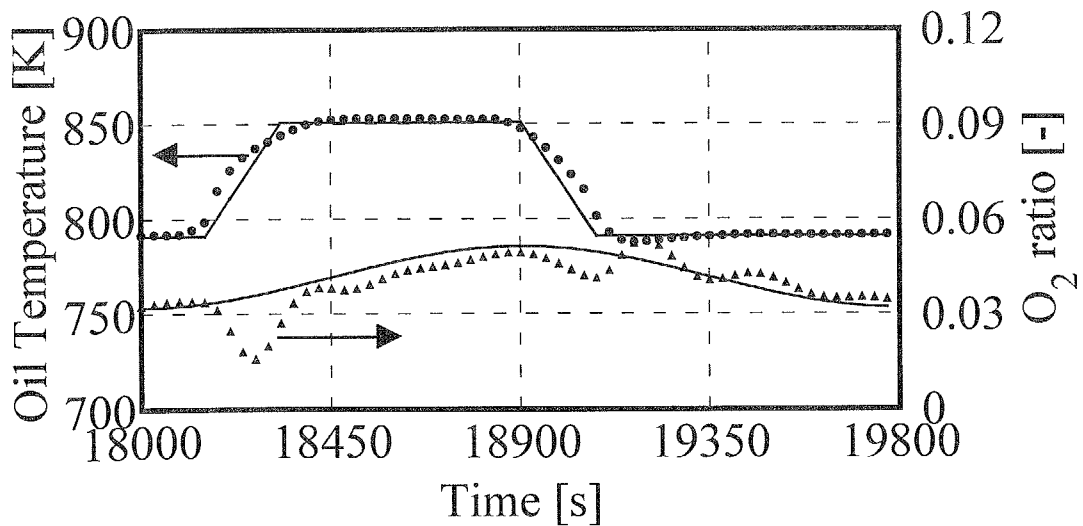
切り替えに関する情報としては、そのタイミング、導入される排ガス量、排ガスの持つ熱量がある。ここでは切り替えは突然起きるものではなく、数分前には把握できるとする。排ガス量とその熱量は、他プロセスの運転状況からある程度の予測ができるものとする。



(a) 排ガスの量と組成



(b) 操作量



(c) 制御量

図 5-11 最初の切り替え時での制御結果

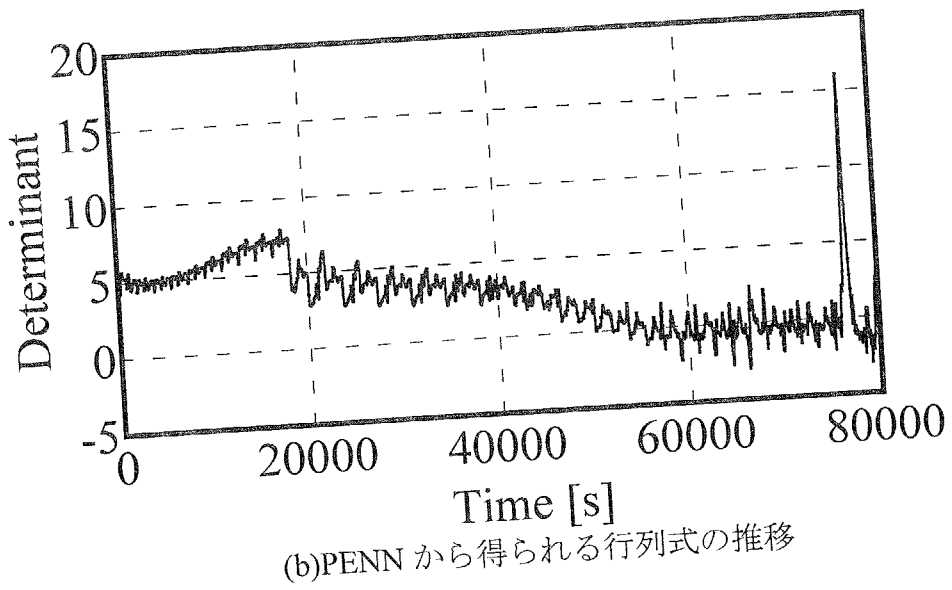
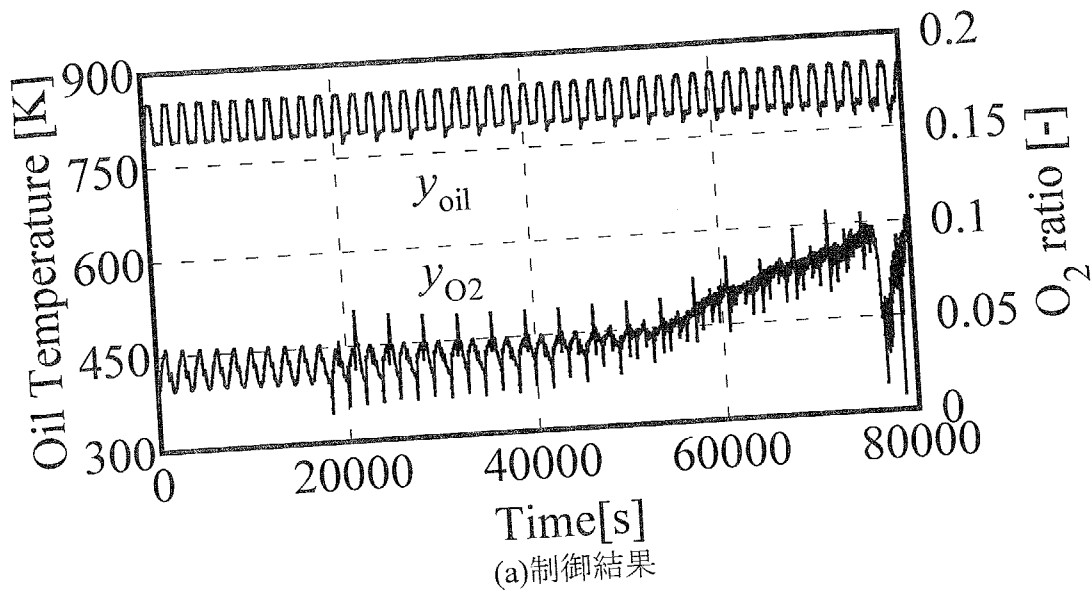


図 5-12 制御結果

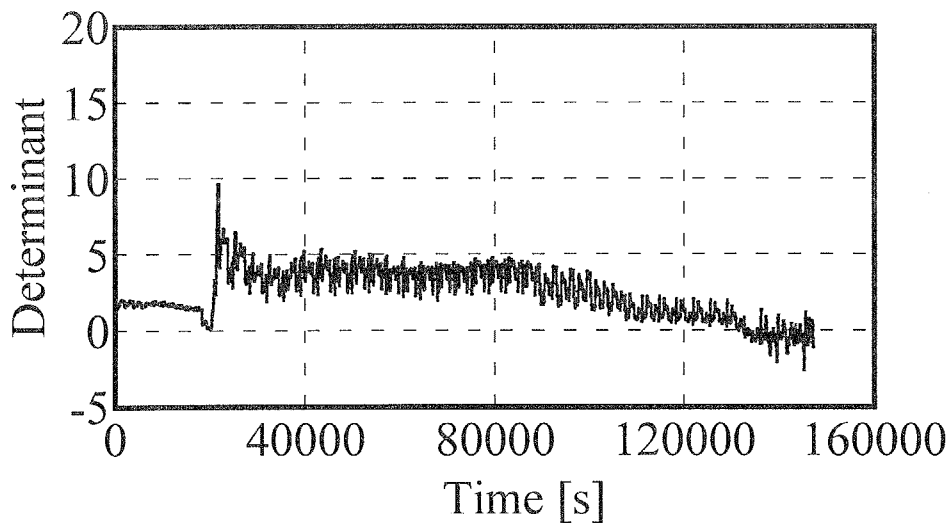
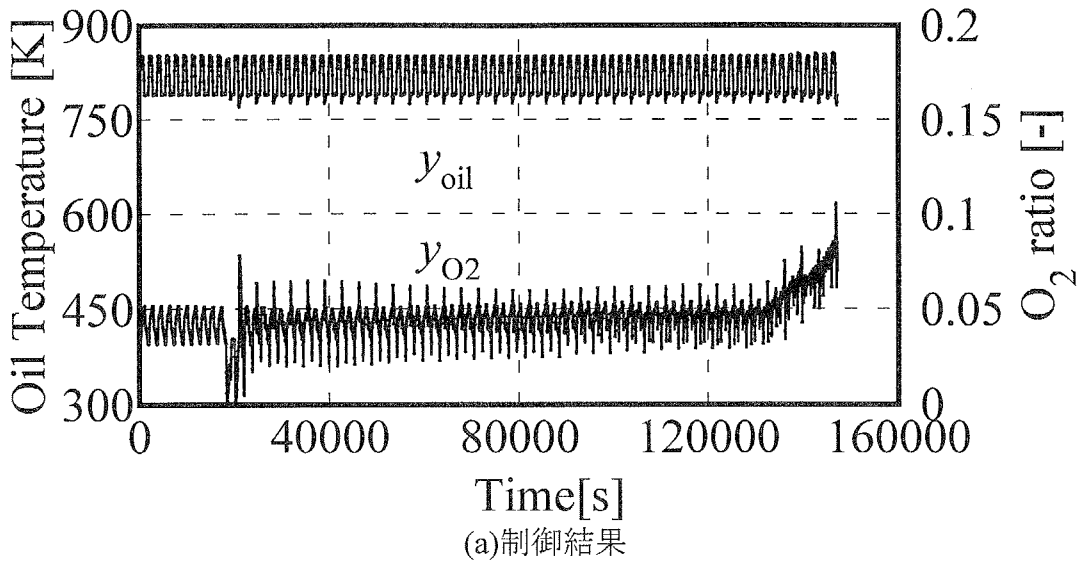
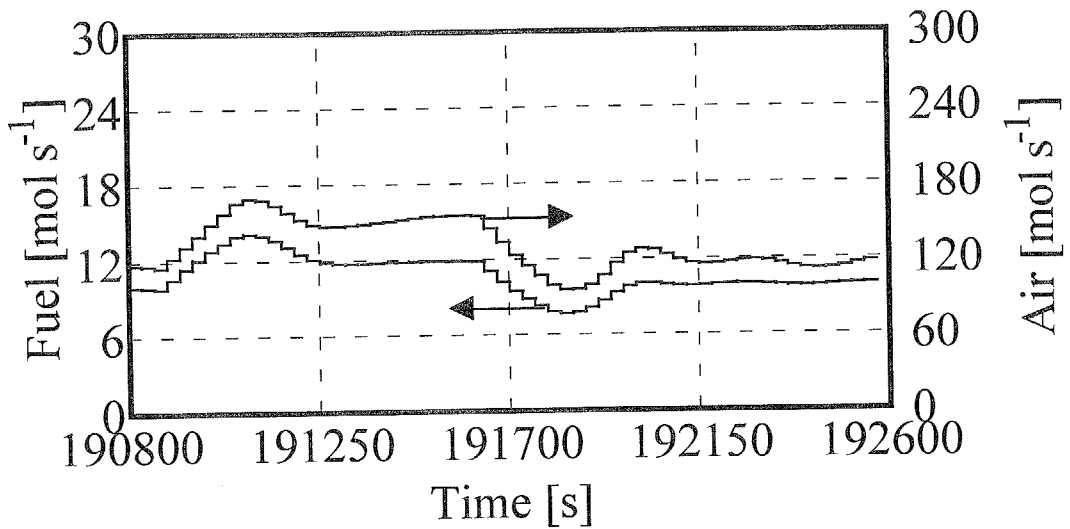
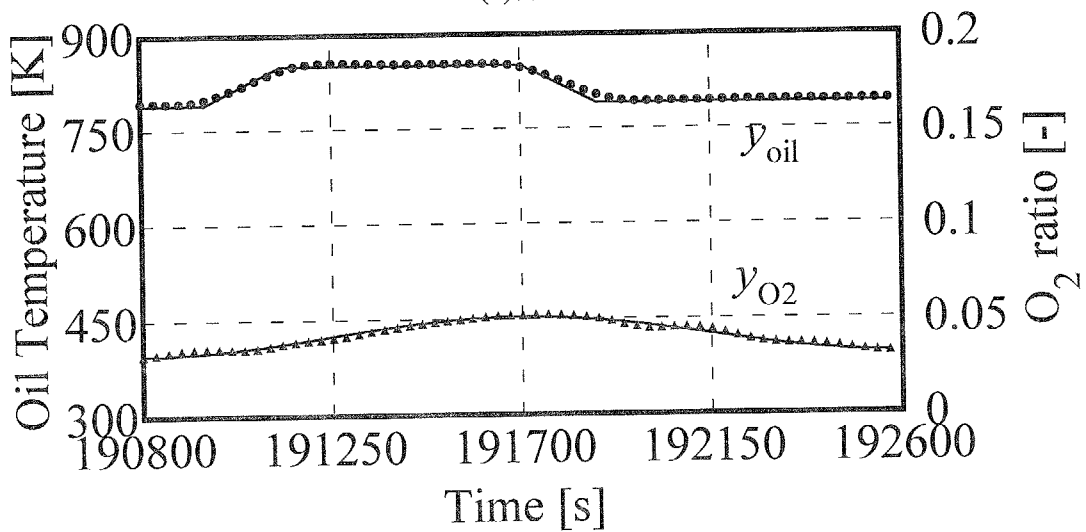


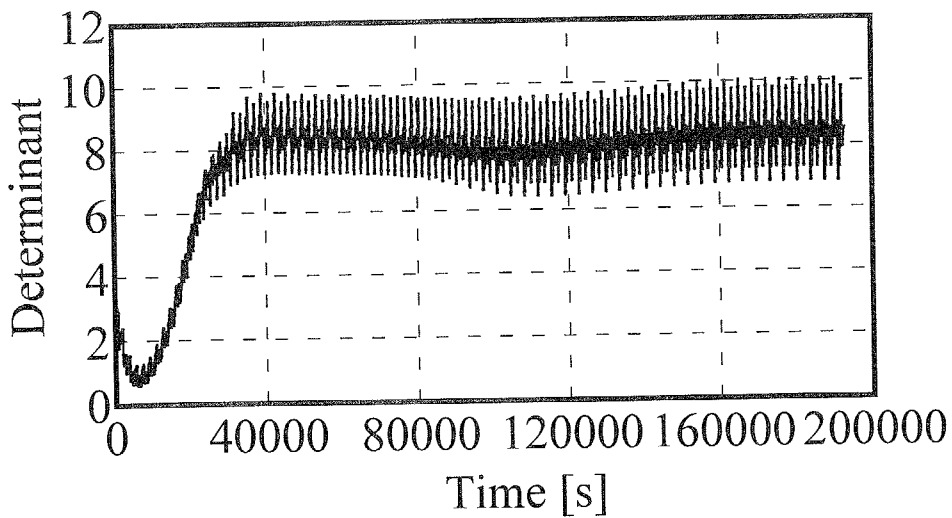
図 5-13 対策を施したときの制御結果



(a)操作量



(b)制御量



(c) 行列式の推移

図 5-14 排ガスの熱量を考慮した制御結果

これらの情報を利用すると、切り替え時に導入される排ガスの持つ総発熱量が計算できる。するとこの排ガスの発熱量に見合った分だけ純粋な燃料ガスを減らすことで重油温度への切り替えの影響をうち消すことができる。しかし排ガスと燃料ガスの酸素消費量が異なるので酸素濃度については切り替えの影響が残る。

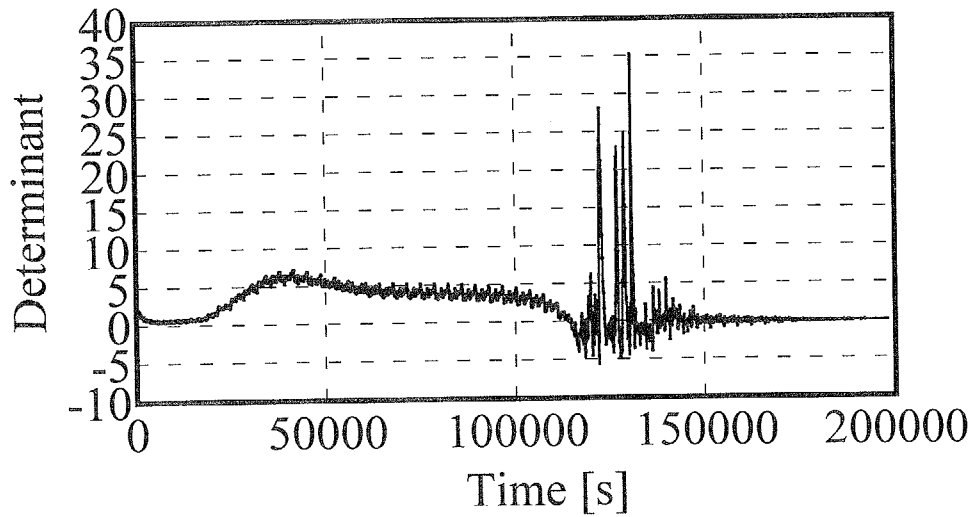
まず排ガス量とその熱量が正確に把握できた場合の制御結果を示す（図 5-14）。切り替えの影響はほとんどみられず、良好な制御が行われた。つぎに、排ガスの熱量が正確に把握できず、見込み違いが生じている場合について検討した。見込み違いとして正確な熱量の±30%までの差を検討した。その結果を表 5-5 にまとめた。

表 5-5 排ガス発熱量の見込み違いによる制御結果

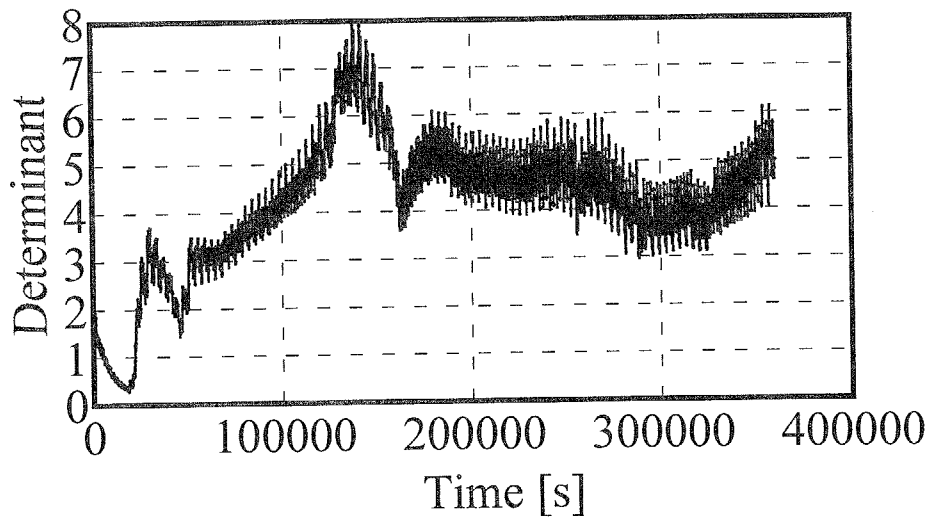
見込み違い	制御の成否	見込み違い	成否
+10%	成功	-15%	成功
+15%	成功	-20%	成功
+20%	失敗	-30%	失敗
+30%	失敗		

また、第 3 章で提案した切り替え対策を併用した場合、発熱量見込み違い-30%の場合で制御成績を向上することができた（図 5-15）。

これらの結果から、排ガスの発熱量の見込み違いが 15%程度であれば切り替えが起きた場合でも制御を行うことが可能であることが分かった。



(a) 切り替え対策を併用しない場合。



(b) 切り替え対策を併用した場合。

図 5-15 切り替え対策の併用効果

5.6.2. 燃料組成の周期的変化

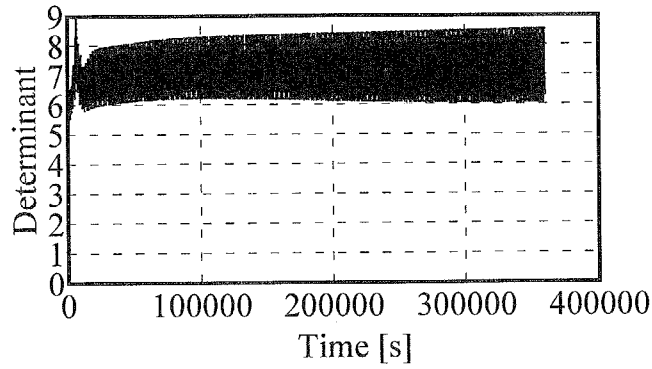
次に、排ガスの選択による切り替えのように短時間で大きくプロセス特性が変化するのではなく、時間が経つにつれて導入した排ガスの組成が変化する場合について検討した。

排ガスは $6\text{mol}\cdot\text{s}^{-1}$ だけ導入し、その排ガスのメタン組成が周期的に変化する場合を想定した。基準組成はメタン 0.5 で残りを窒素とし、メタン組成が振幅 0.1 または 0.2、周期を 1800s,900s,600s,450s で変化する組み合わせで行った。

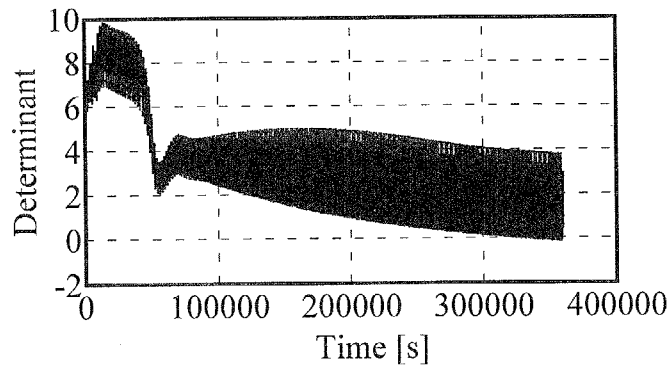
表 5-6 燃料組成の変化と制御結果

振幅	周期	結果	振幅	周期	結果
0.1	1800s	成功	0.2	1800s	成功
0.1	900s	成功	0.2	900s	成功
0.1	600s	成功	0.2	600s	失敗
0.1	450s	失敗			

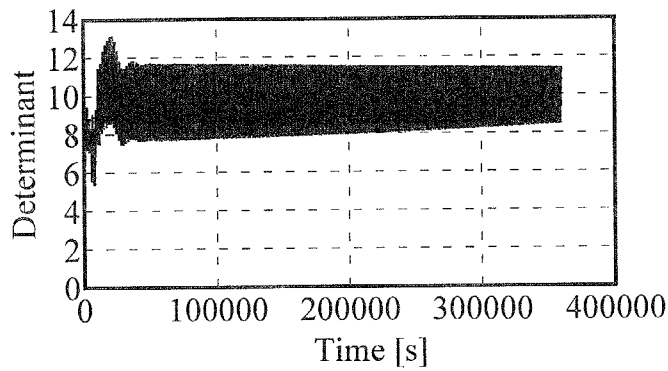
周期を小さくするに従って制御成績が悪い結果が得られた。また振幅が大きい場合は周期の下限值が高くなっている傾向が見られる。制御に成功した場合と失敗した場合の構造化 PENN から得られる行列式を図 5-16 に示した。その結果、制御に失敗した場合は行列式の値が 0 付近になりネットワークの状態が不安定になっていることが確認できた。



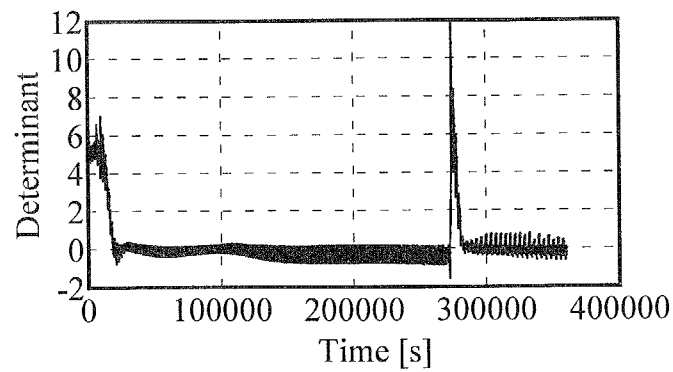
(a) 振幅 0.1 周期 600s



(b) 振幅 0.1 周期 450s



(c) 振幅 0.2 周期 900s



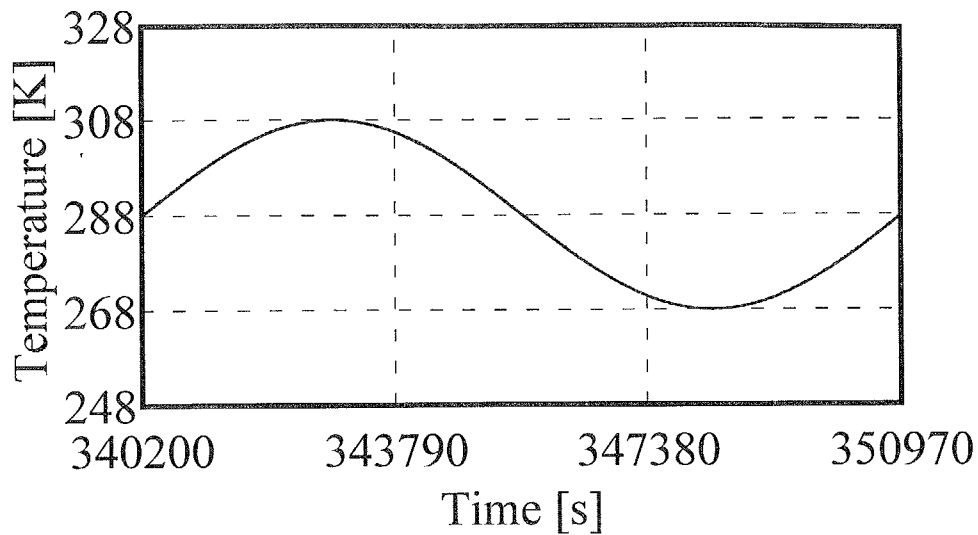
(d) 振幅 0.2 周期 600s

図 5-16 排ガス組成の周期的変化

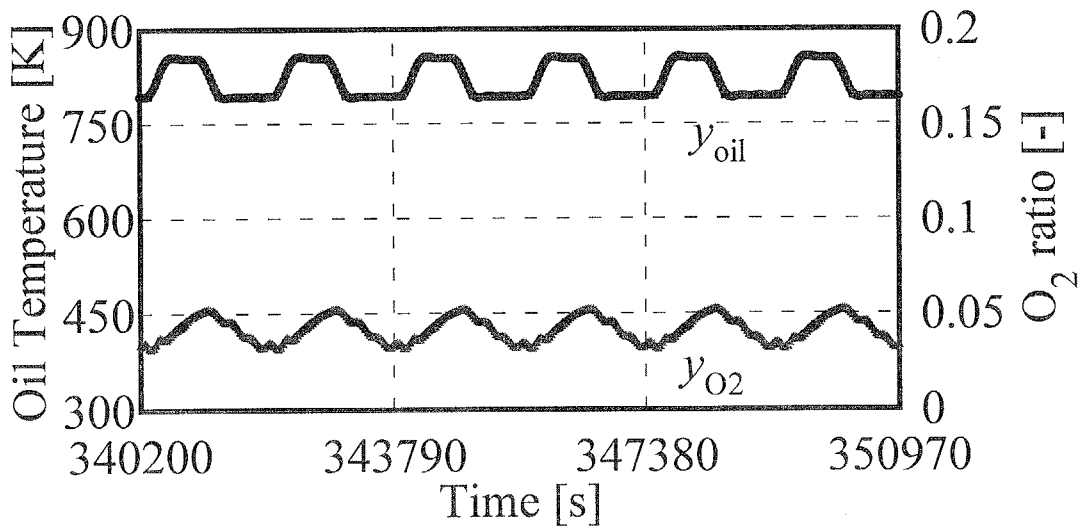
5.6.3. 空気温度の周期的変化

加熱炉の運転では供給する空気温度の影響も考慮に入れる必要がある。朝昼の温度差を想定して空気温度を-5 から+35°Cの間で周期的に変化させた。周期を3時間と6時間について制御を行ったが、両方の場合で良好に制御された。周期3時間の場合の結果を図5-17に示した。また、空気温度変化がPENNに及ぼした影響をみるため、評価ゲイン定数を算出した。図5-17(c)ではゲインが目標値設定により変化しているため影響が確認しづらい。そこで1800s毎の平均値を算出して図5-18に示した。1800s毎に平均したのは目標値が1800s周期で設定されているためである。この図より、空気温度変化にほぼ同期して評価ゲイン定数が推移していることから、構造化PENNは重みを変更して空気温度変化に対応していることがわかる。

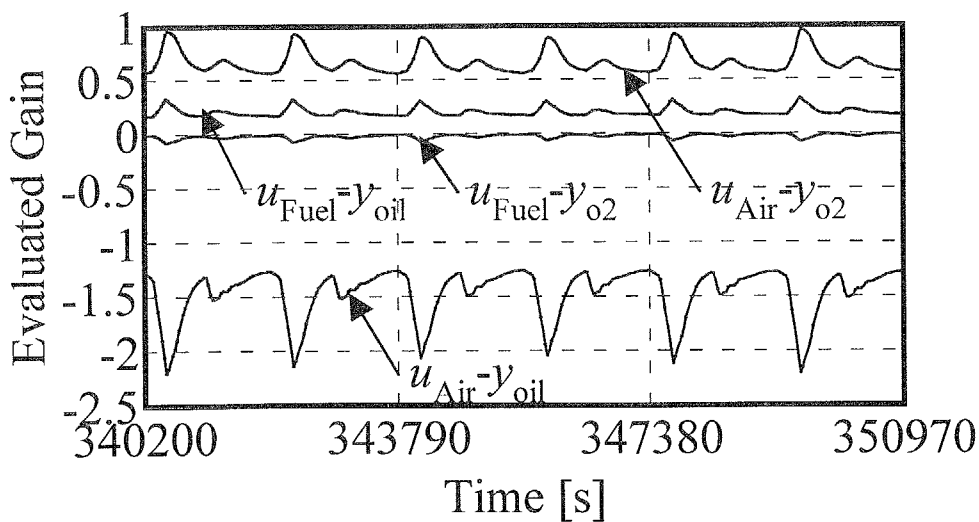
以上の結果から、構造化PENNコントローラはプロセスの特性変化が短時間に急激に起きる場合はそのまま用いることを避ける必要があるが、特性変化に関する情報がある程度の精度で得られる場合はそれを利用することで制御を行うことが可能である。また特性変化が緩やかに起きる場合はオンライン学習による重みの調節によって対応できることを確認した。



(a) 供給空気温度



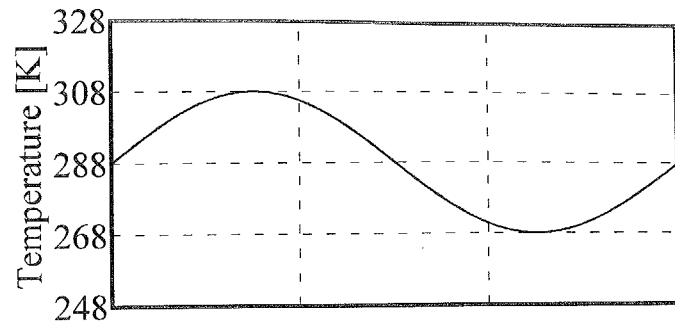
(b) 操作量



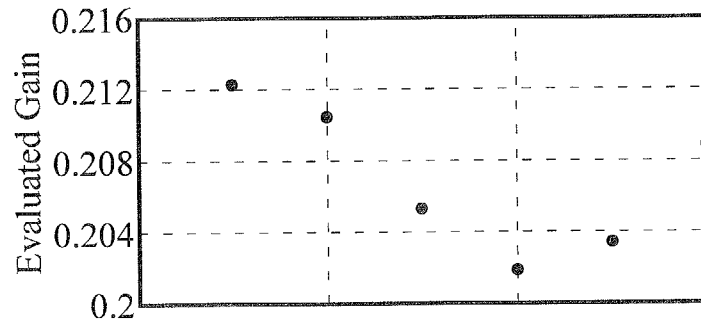
(c) 評価ゲイン定数

図 5-17 供給空気温度変化の影響

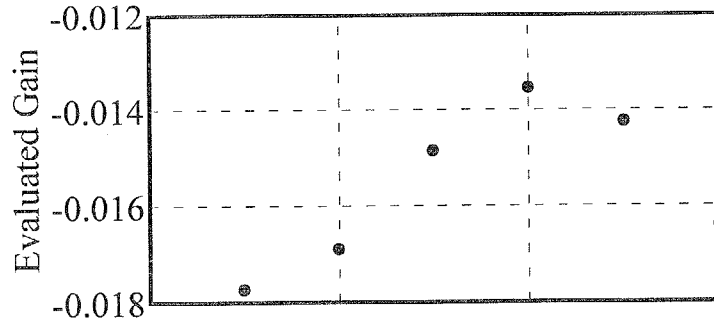
(a) 空気温度



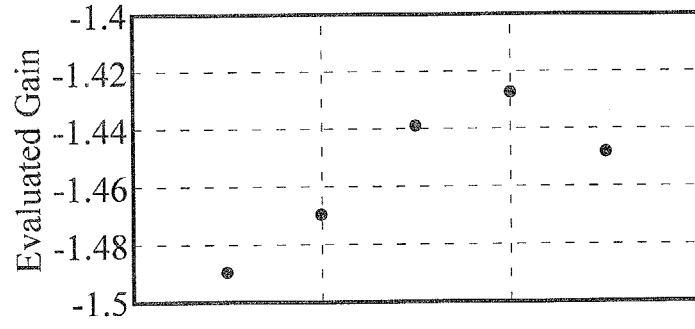
(b) 燃料-重油温度



(c) 燃料-酸素濃度



(d) 空気量-重油温度



(e) 空気量-酸素濃度

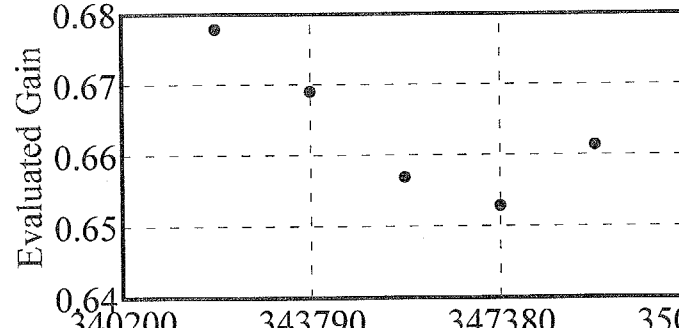


図 5-18 空気温度の変化と評価ゲイン定数の推移

5.7. 第5章の結論

- ・変数間の組み合わせ毎に、遅れ特性が異なるプロセスである重油加熱炉の制御を行った。

非線形プロセスに対してはオンライン学習による重みの更新で良好に制御していることを確認した。

システムノイズと観測ノイズを考慮した結果、放射伝熱係数へのシステムノイズ幅1%と観測ノイズ幅±5Kまでの制御が可能であることを確認した。

- ・プロセスの干渉特性をゲイン定数として把握した。
- ・プロセスの特性変化として、重油加熱炉の燃焼ガスが量と組成について切り替わる場合について適用した結果次の結論を得た。
 - ・切替後の排ガス発熱量の見込み値の分だけ燃料ガスを絞ることで制御を行うことが可能となった。
 - ・発熱量の見込み違いの限界は15%であることがわかった。
- ・燃料組成及び供給空気温度が周期的に変化する場合について検討した。
 - ・周期的変化の周期が短い場合は制御不能になることを確認した。
 - ・周期的変化に伴ってネットワーク重みが変わり制御が行われることを確認した。
- ・第3章で提案した、ネットワークの状態把握法を適用して、ノイズの影響、特性変化の影響を受けて構造化 PENN の状態が不安定になることを確認することができた。

6. 結論

本研究は化学プロセスの制御を目的として、次の事項を可能にした。

- SISO プロセスに対しては、制御量の差に関するパラメータである ΔY_m （過去からの変化量）と ΔY_p （将来望ましい変化量）と操作量の変更量である ΔU のみを用いた非常に小さいネットワークを用いることで、学習速度が向上した。
- MIMO プロセスについては SISO 用の PENN の特性を生かしながらそのネットワークの組み合わせとして構造化ネットワークを提案した。
- ノイズがある場合に、学習関数の誤差評価関数に不感帯を導入し、制御を可能にした。
- オンラインでネットワークを解析することでゲイン定数としてプロセスの干渉特性を把握できた。
- 解析で得られる行列式でネットワークの状態をオンラインで自己診断することが可能となった。
- プロセス特性が経時的に変化する場合、特性変化によって及ぼされる影響を取り除く方法を提案し、制御を可能にした。

NOTATION

E = error from the set point

O = normalized outputted data[-]

R = set point[-]

T = normalized teaching data[-]

U = normalized manipulated variable[-]

Y = normalized controlled variable[-]

ΔY_m = increment of the controlled variable in a previous time step

ΔY_p = increment of the controlled variable in the next k steps

g = gain parameter[-]

g' = evaluated gain parameter[-]

l = lag time constant[-]

p = PENN parameter[-]

r = setpoint

t = time[s]

u = manipulated variable[J/m²·s], dimensionless heat supply[-]

v = dimensionless heat loss[-]

x = dimensionless temperature[-]

y = controlled variable[K], dimensionless temperature[-]

z = position in the ampoule tube[-]

Greek letters

θ = threshold

υ = weight of neural network between input and hidden layer[-]

ω = weight of neural network between hidden and output layer[-]

Subscripts

h = index showing hidden layer

j = index showing present time

k = k steps after j

m = time interval before present time

p = time interval after present time

q = index for the manipulated variable

r = index for the controlled variable

REFERENCES

- An, P.E., M Brown, C.J. Harris and S. Chen, "Comparative Aspects of Neural Network Algorithms for On-line Modelling of Dynamic Processes," *Journal of Systems and Control Engineering*, 207, 223-241(1993)
- Aoyama, Atsushi, Francis J. Doyle III and Vankat Venkatasubramanian, "A Fuzzy Neural-network Approach for Nonlinear Process Control," *Engineering. Application on Artificial Intelligence*, 8, 483-498(1995)
- Baldi, Pierre and Amir F. Atiya, "How Delays Affect Neural Dynamics and Learning," *IEEE Transactions of Neural Networks*, 5, 612-621(1994)
- Bhat N. and T.J. McAvoy, "Use of Neural Nets For Dynamic Modeling and Control of Chemical Process Systems," *Computers Chemical Engineering*, 14, 573-583(1990)
- Bhat, Naveen V., Peter A. Minderman, Jr., Thomas McAvoy and Nam Sum Wang, "Modeling Chemical Process Systems via Neural Computation," *IEEE Control Systems Magazine*, April, 24-31(1990)
- Broomhead, D.S. and David Lowe, "Multivariable Functional Interpolation and Adaptive Networks," *Complex Systems*, 2, 321-355(1988)
- Bulsari, A., "Multivariable Control of a Linear System using Feed-Forward Neural Networks," *Second IEEE Conference on Control Applications*, September, 13--16(1993)
- Chessari, C.J., G.W. Barton and J.A. Romagnoli, "A Strategy for the Nonlinear Control of Affine Systems Using Multiple Neural Networks," *Proceedings of the American Control Conference*, 1470-1473(1993)
- Chen, S. and S. A. Billings, "Neural Networks for Nonlinear Dynamic System Modeling and Identification," *Int. J. Control*, 56, 319-346(1992)
- Chovan, Tibor, Thierry Catfolis and Kurt Meert, "Neural Network Architecture for Process Control Based on the RTRL Algorithm," *AIChE Journal*, 493-502(1996)
- Falb, Peter L. and William A. Wolovich, "Decoupling in the Design and Synthesis of Multivariable Control Systems," *IEEE Transactions on Automatic Control*, AC-12, 651-659(1967)
- Fuchs, Avraham, and Daniel. E. Lewin, "Modeling, Simulation and Control of a Crude Oil Preheating Furnace," *Chemical Engineering Science*, 4, 661-679(1993)
- Gues, Allon, James L. Elibert and Moshe Kam, "Neural Network Architecture for Control," *IEEE Control Systems Magazine*, April, 22-25(1998)
- Hampel, F. R., P.J. Rousseeuw, and E. Ronchetti, "The Change of variance Curve and Optimal Redescending M-estimators," *Journal of American Statist. Assoc.*, 76, 643-648(1981)

- Hoskins, J.C. and D.M. Himmelblau, "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering," *Computer Chemical Engineering*, 12, 881-890(1988)
- Ishida, M., "Control by a Policy-and-Experience-driven Neural Network," *J. of Chem. Eng. of Japan*, 25, 108-111 (1992)
- Ishida, M. and T. Ohba, "Control by a New Policy- and Experience-driven Neural Network to Follow a Desired Trajectory," *J. of Chem. Eng. of Japan*, 27, 137-138(1994)
- Ishida, M. and J. Zhan, "Neural Model-Predictive Control of Distributed Parameter Crystal Growth Process," *AIChE J.*, 41, 2333-2335(1995)
- Ito, Y., and N. Katoh, "A new Bridgeman Process for Single Crystal Growth Based on Control of Distributed Parameter System," *Kagaku Kougaku Ronbunshu*, 20, 288 (1994)
- Khalid, Marzuki, Rubiyah yusof and Sigeru Omatu, "Neural Networks for Self-Learning Process Control Systems," *Journal of Artificial Neural Networks*, 1, 23-49(1994)
- Kravaris, Costas and Masoud Soroush, "Synthesis of Multivariable Nonlinear Controllers by Input/Output Linearization," *AIChE journal*, 36, 249-263(1990)
- Lee, T.H., Q.G. Wang and K.K.Tan, "Robust Smith-Predictor Controller for Uncertain Delay Systems," *AIChE Journal*, 42, 1033-1040(1996)
- Mavrovouniotis, M. L., and S. Chang, "Hierarchical Neural Networks," *Computers chemical Engineering*, 16, 347-369 (1992)
- Morris, A. J., G. A. Montague, and M. J. Willis, "Artificial Neural Networks: Studies in Process Modeling and Control," *Trans. IChemE*, 72, 1-19 (1994)
- Nabhan, Tarek M. and Albert Y. Zomaya, "Toward Generating Neural Network Structures for Function Approximation," *Neural Networks*, 7, 89-99(1994)
- Nakanishim Hiroaki, Takehisa Khoda and Koich Inoue, "A Design Method of Porical State Feed-Back Control Systems by use of neural Network," *T.SICE*, 9, 882-889(1997)
- Narendra, K. S., and Snehasis Mukhopadhyay, "Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks," *Neural Networks*, 7, 737-752 (1994)
- Ohba, Takehiro and Masaru Ishida, "Structured PENN Controller for MIMO Process," *Journal of Chemical Engineering of Japan*, 31, 515-550(1998)
- Ohba, Takehiro and Masaru Ishida, "Application of a Neural Network to Evaluation of Interactions in a MIMO Process," *AIChE Journal*, 44, 2018-2024(1998)
- Padilla, Salvador and Jesys Alvarez, "Control of Continuous Copolymerization Reactors," *AIChE Journal*, 43, 448-463(1997)

- Pham, D.T. and S.J. Oh, "Adaptive control of a robot using neural networks," *Robotica*, 12, 553-561(1994)
- Psaltis, D., A. Sideris and A. Yamamura, "A Multilayered Neural Network Controller," *IEEE Control Systems Magazine*, April, 17-21(1988)
- Rivera, Daniel E., Manfred Morari and Sigurd Skogestad, "Internal Model Control. 4. PID Controller Design," *Industrial Engineering Chemical Process Des. Dev.*, 25, 252-265(1986)
- Shibata, Katsunari, "A Neural-Network to get Correlated Information among Multiple Inputs," *Proceedings of 1993 International Joint Conference on Neural Networks*, 2532-2535(1993)
- Soroush, Masoud and Costas Kravaris, "Discrete-Time Nonlinear Feedback Control of Multivariable Processes," *AIChE Journal*, 42, 187-203(1996)
- Ungar, L.H., B.A. Powell and S.N. Kamens, "Adaptive Networks for Fault Diagnosis and Process Control," *Computers Chemical Engineering*, 14, 561-572(1990)
- Xu, Bing Hong, Toshio Tsuji and Makoto Kaneko, "Identification and Control for a Class of Plant with Uncertainties Using Neural Network," *T.SICE*, 32, 510-516(1996)
- Yamada, Yakayuki and Tetsuro Yabuta, "Some Remarks on Characteristics of Direct Neural-Controller with Regard to Adaptive Control," *T.SICE*, 7, 784-791(1991)