

2次元フィルタおよびプロセスシステムに対する
適応アルゴリズムの開発に関する研究

(STUDIES ON DEVELOPMENT OF ADAPTIVE
ALGORITHMS FOR TWO-DIMENSIONAL FILTERS AND
PROCESS SYSTEMS)

内容梗概

本論文は、筆者が広島大学大学院工学研究科 博士課程後期 複雑システム工学専攻において行った、2次元フィルタおよびプロセスシステムに対する適応アルゴリズムの開発に関する研究成果を5つの章にまとめたものである。

第1章では、デジタル信号処理の分野における2次元適応 FIR フィルタとその適応アルゴリズム、およびプロセス制御の分野で広く用いられている適応制御、PID 制御系とその調整アルゴリズムについて概説し、その歴史背景、当該分野における本研究の目的を明確にしている。そして、5章からなる本論文の概要を詳細に述べている。

第2章では、直交射影アルゴリズムを用いた2次元適応 FIR フィルタの設計について提案している。従来の2次元 LMS アルゴリズム、2次元正規化アルゴリズムについて詳述し、図式解釈を通してこれらの問題点を指摘する。2次元直交射影アルゴリズムが提案され、その収束条件が与えられている。一般的には垂直あるいは水平の場合に対して信号の性質が異なる場合が多いため、水平・垂直の2方向から処理を行う直交射影アルゴリズムおよび水平・垂直の2方向の入力信号を直交化させるアルゴリズムについても同様に考察する。最後に幾つかの応用例を示し、これらの手法の有効性を示している。

第3章では、制御性能評価と制御系設計を融合するモデル予測型 PID 制御系の設計について提案している。定常状態における制御性能に関して、代表的な PID 制御則との比較により GPC 制御則の有効性について検証を行なっている。制御系従事者にとって直感的に理解できるパラメータを用いて GPC-PID 制御則の設計パラメータの調整を行うことにより、プラント従事者にとって極めて有効で有益な PID パラメータ調整法を確立している。また、定常状態における制御性能を評価することで性能が劣化した PID 制御器のパラメータを設計パラメータの再調整により適応させるアルゴリズムを構築している。コンピュータシミュレーションにより、従来法であるセルフチューニングコントローラと比較することで、定常状態における制御性能の改善やシステム同定回数的大幅な削減などの観点から本手法の有効性を検証している。

第4章では、制御性能評価機能を有する CMAC-PID 制御系の設計について提案している。CMAC および CMAC-PID 制御系について詳述し、CMAC-PID 制御系に含まれる荷

重値を第3章で述べた方法により更新させることで、所望の定常特性が得られるように適応させるアルゴリズムについて提案されている。これにより、1つのCMACで所望の過渡性能および定常性能が得られるコントローラが構築できる。最後にコンピュータシミュレーションにより、第3章の方法と比較することで、非線形システムにおいて本手法の有効性を検証している。

第5章では、本研究における成果を総括し、今後検討すべき課題等について述べている。

目次

第1章	緒論	1
1.1	研究の背景	1
1.1.1	2次元フィルタの現状	2
1.1.2	プロセスシステムの現状	3
1.2	本研究の目的と構成	5
第2章	直交射影アルゴリズムによる2次元適応 FIR フィルタの設計	9
2.1	緒言	9
2.2	2次元 LMS および NLMS 適応 FIR フィルタ	10
2.2.1	2次元ウィナーフィルタ	10
2.2.2	2次元 LMS アルゴリズム	13
2.2.3	2次元正規化 LMS アルゴリズム	16
2.3	2次元直交射影適応 FIR フィルタの設計	19
2.3.1	2次元直交射影アルゴリズムの構成	19
2.3.2	収束性の証明	21
2.3.3	アルゴリズムの一般化	22
2.4	2方向処理による2次元 LMS および NLMS 適応 FIR フィルタ	23
2.4.1	2方向処理による2次元 LMS アルゴリズム	23
2.4.2	2方向処理による2次元 LMS アルゴリズムの収束性	25
2.4.3	2方向処理による2次元正規化 LMS アルゴリズム	26
2.5	2方向処理による2次元直交射影適応 FIR フィルタの設計	27
2.5.1	2方向処理による2次元直交射影アルゴリズムの構成	27
2.5.2	2方向処理による2次元直交射影アルゴリズムの収束性の証明	28

2.5.3	アルゴリズムの一般化	31
2.6	2方向の入力を直交化させる2次元直交射影適応 FIR フィルタの設計 . . .	33
2.6.1	アルゴリズムの構成	33
2.6.2	アルゴリズムの一般化	34
2.7	数値例	36
2.7.1	2次元直交射影適応 FIR フィルタの適用例	36
2.7.2	2方向処理による2次元直交射影適応 FIR フィルタの適用例	43
2.7.3	2方向の入力を直交化させる2次元直交射影適応 FIR フィルタの適用例	46
2.8	結言	49
第3章	モデル予測機能を有するパフォーマンス駆動型 PID 制御系の設計	51
3.1	緒言	51
3.2	モデル予測型 PID 制御系	52
3.2.1	制御対象モデルの記述	52
3.2.2	PID コントローラ	53
3.2.3	GPC 則による PID パラメータの調整法	54
3.3	パフォーマンス駆動型 PID 制御系の設計	56
3.3.1	定常状態における制御性能	56
3.3.2	GPC-PID 制御則の制御性能	57
3.3.3	重み係数 λ の設定方法	59
3.3.4	性能評価指数	61
3.3.5	システム同定	63
3.3.6	パフォーマンス駆動型 PID コントローラ	64
3.4	数値例	64
3.4.1	時変システム	65
3.4.2	非線形システム	77
3.5	結言	82
第4章	パフォーマンス駆動型 CMAC-PID 制御系の設計	83
4.1	緒言	83

4.2	CMAC-PID 制御系	85
4.2.1	CMAC の概要	85
4.2.2	PID 制御への適用	86
4.3	パフォーマンス駆動型 CMAC-PID 制御系の設計	88
4.3.1	概要	88
4.3.2	モデル予測型 PID 制御	88
4.3.3	制御性能評価	90
4.3.4	システム同定	91
4.3.5	荷重表の更新方法	92
4.3.6	パフォーマンス駆動型 CMAC-PID コントローラ	92
4.4	数値例	93
4.5	結言	105
第5章 結 論		107
謝 辞		109
参 考 文 献		111
引用既発表文献		119

第1章

緒論

1.1 研究の背景

近年、マイクロプロセッサの発展に伴い、デジタルによる制御装置が広く用いられるようになり、デジタル信号を処理する制御理論や技術 [1,2] の果たす役割はますます重要なものとなっている。多くの制御対象は連続時間システムであるが、実際にはそれらから得られる信号を離散時間・離散値へ変換し、処理を行っている。こうした背景には、デジタル制御は、高精度の制御が実現できる、製造工程の簡易化が図られる、温度変動などによる特性変化がない、デジタル演算のため雑音などの特性変化に強い、などのさまざまな利点が挙げられることに起因する。デジタル制御は一般に計測制御の分野で発展した理論であるが、デジタルに変換した信号を解析・処理を行うことから、情報通信分野、おもにデジタル信号処理と密接な関係があり [3]、これらの分野を融合した研究・開発は今後もさらに続くものと思われる。

しかし、一般のデジタル制御では、設計の際に要求される制御装置を固定させている場合が多い。このままでは、処理対象信号などの特性の変化によってシステムの特性を決定するパラメータ値を変化させることができず、制御装置に要求される特性を完全に決定することができない場合には十分に対処できない。そのため、システム特性の先験的情報が完全に既知でない場合には、信号処理の過程で制御装置がある基準のもとに最適となるようにパラメータ値を逐次修正する機能を備えた制御装置が必要になる。このように信号処理過程で必要に応じて制御装置の特性を変化させる機能を備えた制御のことを適応制御 [4,5] と呼ぶ。

適応制御の構成は、制御装置（コントローラまたはフィルタ）およびその制御装置のパラ

メータを調整する適応機構（適応アルゴリズム）の2つから構成されている。本研究では、デジタル信号処理およびプロセス制御の2つの分野の観点から、それぞれ2次元デジタルフィルタおよびプロセスシステムで用いられている適応アルゴリズムの開発に焦点を当てる。上述のそれぞれの分野における現状について以下に詳述する。

1.1.1 2次元フィルタの現状

近年、デジタル信号処理は、音声などの1次元信号の域にとどまらず、静止画、動画、コンピュータ断層撮影写真、空間的に配置された幾つかのセンサアレイによって同時に受信された時間波形など、平面的、立体的、超立体的な広がりをもった多次元離散空間信号を処理する目的で、情報通信、計測制御、医療、地震、海洋など幅広い分野で利用されている [6]。多次元デジタル信号処理が発展した理由として、実データの多くが本来は多次元であること、CCD 画像センサ、X 線や NMR コンピュータ断層写真撮影センサ、地震用アレイ、音響用アレイ等、多次元センサアレイが昨今広く利用できるようになってきたことなどが挙げられる。特に、適応フィルタの果たす役割は、デジタル信号処理の対象となる多くの分野で欠かせない存在であり、通信や画像処理を初めとする様々な信号処理の応用において有用なものになっている。

適応フィルタは元来、伝送路の歪みを補償する自動等化器、未知の伝送路を介して戻ってくるエコーや雑音の消去、音声の分析・認識における特徴パラメータの抽出等に見られるように、1次元信号を処理する目的で発展してきた [7-10]。しかし、最近では画像などの多次元信号を処理する目的で多次元適応フィルタ [11-15] が研究されており、画像強調、ノイズの低減化、データ圧縮などへの応用が期待されている。

適応フィルタは、フィルタ部分とアルゴリズム部分の2つに分けて考えることができる。フィルタ部分において最も一般的な回路構成は、零点のみが調整可能な適応 FIR フィルタと、極と零点の両方が調整可能な適応 IIR フィルタがある。適応 IIR フィルタ [16-19] は、所望の特性を低次のフィルタで達成できる場合が多いが、多次元システムにおける安定性の理論が十分に確立されていないため安定性の保証が非常に難しく、また帰還部分が存在するため丸め誤差などの影響を受けやすくなりフィルタ係数の精度の点では不利となる。これに対して適応 FIR フィルタは、安定判別を必要とせず、多次元システムにおける理論の解析が行いやすいなどの利点がある。しかし、これで所望の特性を得るには、高次のフィルタがしばしば必要になる。

また、アルゴリズム部分において、LMS(Least Mean Square) アルゴリズムと RLS(Recursive Least Squares) アルゴリズムが広く用いられている [20]。LMS アルゴリズムは、計算量が少なく、構造が簡単なため、非常に多くの研究 [21–26] がなされており、ハードウェアとして実現されてきているが、収束速度が遅く、入力信号の性質を受けやすいなどの問題点がある。これに類似したアルゴリズムとして正規化 LMS(Normalized LMS:NLMS) アルゴリズム [27] があり、これら 2つのアルゴリズムは実用段階に入った適応信号処理を支える代表的な適応アルゴリズムといえる。一方、RLS アルゴリズムは、その起源が制御工学におけるシステム同定の分野にあり、非常に優れた収束特性を示す。しかし、このアルゴリズムは計算量が非常に多いため、特に 2次元信号のような大量のデータを実時間で処理するには向いていないといえる。そのため、高速アルゴリズム、シストリックアレーを用いた手法などの研究 [28–30] も報告されているが、多次元信号を処理するアルゴリズムとしてハードウェアの実現化は多大なコストを要するなどの問題がある。

また、正規化 LMS アルゴリズムを一般化したアルゴリズムとして、直交射影アルゴリズムがある [31–35]。このアルゴリズムは、有色な入力信号において LMS アルゴリズムより収束速度が速く、RLS アルゴリズムより計算量が少ないという利点がある。この意味において、直交射影アルゴリズムは、LMS アルゴリズムと RLS アルゴリズムの中間に位置し、多次元データをリアルタイムで高精度に処理を行うアルゴリズムとして適しているといえる。

1.1.2 プロセスシステムの現状

化学、石油精製、鉄鋼などに代表されるプロセス産業界において、原材料から各種製品を製作する過程で様々な制御装置が用いられている [36]。たとえば、石油精製プラントでは、蒸留棟の温度制御、圧力制御、液量制御、流量制御など、目的の異なる数多くの制御装置から構成されている。このように複雑な構造を有するプロセス制御系では、操業に際して製品品質・安全性・経済性・環境保全などの項目から要求される条件をすべて満たさなければならない。しかし、原料の特性、冷却水の特性などの動作条件や外気温などの環境条件の変動により、プラントに最適な稼働条件は時々刻々と変化する。そのため、プラントの稼働状況を逐次監視し、常に最適な状態を維持し続ける必要がある。

この問題を解決する手法のひとつとしてセルフチューニングコントロール (Self-Tuning Control:STC) [37] がある。これは、適応制御の一種であり、評価規範の最小化に基づく

離散時間確率系制御理論とシステム同定理論を融合させたシステムで構成される。その評価規範の種類により、最小分散制御 (Minimum Variance Control:MVC) [38], 一般化最小分散制御 (Generalized Minimum Variance Control:GMVC) [39, 40], 一般化予測制御 (Generalized Predictive Control:GPC) [41–45] などの方法が報告されている。しかし、プロセスシステムの多くは非線形性や不確かさが含まれているため、上述の制御則では十分な性能を発揮できないことが分かっている。そこで、知的情報処理を用いた制御系の設計に関して数多くの研究が行われており、ニューラルネットワーク (Neural Network:NN) を用いた制御 [46–53], 小脳演算モデル (Cerebellar Model Articulation Controller:CMAC) を用いた制御 [54–59] などが報告されている。

また、プロセスプラントに用いられる制御装置として PID 制御が広く占めている [60]。PID 制御は、比例・積分・微分の3要素から構成されており、各制御の役割が直感的にわかりやすく、実装化が容易であるという特徴から、プロセス産業界において80%以上の制御装置で使用されている。そのPIDパラメータを調節する方法は古くから多くの研究が行われており、Ziegler & Nichols(ZN)法 [61], Chien, Hrones & Reswick(CHR)法 [62], 部分的モデルマッチング法 [63, 64] などが代表的なPIDパラメータ調節法としてよく知られている。また、PID制御系の高度化に関する研究も数多く行われており、セルフチューニングGMVC-PID制御 [65–67], セルフチューニングGPC-PID制御 [68–71], ニューラルネットワークPID制御 [72, 73], 小脳演算モデルを用いたPID制御 [74], 生物進化論的PID制御 [75, 76], データベース型PID制御 [77–79] などが報告されている。

一方、最近では、定常状態でのプロセスの入出力データを用いてプロセスを監視し、異常検出や制御系の評価を行う研究が、制御性能評価 (Control Performance Assessment:CPA) または制御性能監視 (Control Performance Monitoring:CPM) として盛んに行われている。これは、プラントの停止やシステムへの信号付加によるシステム同定を極力行うことなく、システムの稼動データより制御性能の良さを評価する方法として、プロセス産業界から注目されている研究分野である。これらの研究は、1989年に発表されたHarrisの論文 [80] に端を発し、最小分散制御則に基づいて制御系の評価・診断を行う方法について研究が報告されている。さらに、現在の制御量の分散と達成可能な制御量の分散の比として表される制御性能評価指数 [81–83] について提案があり、多変数プロセス制御を対象とした制御性能評価に関する研究 [84, 85] も行われている。また、CPAに関する多くのレビュー [86–89] も報告されている。

そして、高品質維持だけではなく、プラント制御の安定化および最適化運転による運用コストの低減の観点から制御性能評価を行う研究 [90,91] が行われるようになった。上述のように、プロセスプラントの特性は環境の変化や製品の変更などの原因で時間と共に変化するため、プロセス制御系の持つ性能を十分に発揮できず、所望の制御性能が得られない場合が起こる。現状では、STC を用いて、システム同定の結果からシステムパラメータの推定を行い、制御性能が十分に発揮されるように制御装置を逐次調整を行っている。システム同定によるパラメータ推定は、計算量の増大やパラメータ推定の信頼性に問題がある。実際の現場では、制御性能評価の観点で制御装置を必要と時のみ適応させる方法の確立が望まれている。そこで、制御性能評価と制御系設計を複合的に融合させる研究 [92-94] が新しく立ち上げられており、さらに発展させていく必要がある。

1.2 本研究の目的と構成

1.1 節で述べた背景をもとに、本論文では、2次元フィルタおよびプロセスシステムで用いられる適応アルゴリズムに関し、以下の項目について検討を行っている。

1. デジタル信号処理分野において、2次元データをリアルタイムで高精度に処理を行う2次元 FIR フィルタの適応アルゴリズムを開発すること。
2. プロセス制御分野において、プロセス産業界で広く用いられている PID 制御系を制御性能評価に基づいて適応するアルゴリズムを開発すること。

本論文は5つの章から構成されており、その具体的な内容を以下に詳述する。

第2章では、2次元データの特性を考慮して高精度に処理を行う2次元直交射影適応フィルタについて提案している。従来の2次元 LMS アルゴリズム、2次元 NLMS アルゴリズムは、更新式が簡単であるため処理速度が速いという特長があるが、有色性を含むデータでは収束速度が著しく劣化する問題があった。そこで、1次元直交射影アルゴリズム [32] を2次元へ拡張させた2次元直交射影アルゴリズムについて提案している。これにより、計算量は多くなるものの上述の問題を解決している。また、提案した手法の収束性について証明し、収束するための条件を明らかにした。しかし、2次元データの場合は、一般的には垂直あるいは水平の場合に対して信号の性質が異なる場合が多い。そこで、2次元データの自由度を考慮し、水平・垂直の2方向から処理を行う直交射影アルゴリズムについて同様に提案し、収束するための条件を導いている。さらに、水平方向および垂直方向の入力信号を直交

化させる方法について提案している。これにより、2次元データの特徴を最大限に活用することができ、推定精度の高いフィルタが実現できる。最後に、提案した3つのアルゴリズムをそれぞれ2次元システム同定に適用し、従来法と比較することによって本手法の有効性を検証している。

第3章では、制御性能評価と制御系設計を融合するモデル予測型 PID 制御系の設計について提案している。プロセス制御の分野では、GPC 則 [41] による制御系が広く用いられており、GPC 則に基づいて PID パラメータを調整する方法 [44] が報告されている。しかし、設計者が設定を行う GPC パラメータの調整により制御系の性能に大きく影響を及ぼすことが分かっており、その設定方法は試行錯誤的に安全なパラメータを求めているのが現状である。そこで、制御性能評価に基づいてシステムパラメータの推定を行い、設計者が所望の制御性能が得られるよう設計パラメータを適応させるセルフチューニング GPC-PID 制御系の設計について提案している。定常状態における制御性能に関して、代表的な PID 制御則との比較により GPC 則の有効性について検証を行っている。制御系従事者にとって直感的に理解できるパラメータを用いて GPC-PID 制御則の設計パラメータの調整を行うことにより、プラント従事者にとって極めて有効で有益な PID パラメータ調整法を確立している。また、定常状態における制御性能を評価することで、性能が劣化した PID 制御器のパラメータを設計パラメータの再調整により適応させるアルゴリズムを構築している。コンピュータシミュレーションにより、従来法であるセルフチューニングコントローラと比較することで、定常状態における制御性能の改善やシステム同定回数の大幅な削減などの観点から、本手法の有効性を検証している。

第4章では、非線形プラントシステムを対象とした制御性能評価機能を有する PID 制御系の設計について提案している。第3章では、おもに定常状態における制御性能に着目しており、非線形システムにおける過渡状態の制御性能（立上がり時間、行過ぎ量、整定時間など）については検討されていない。プロセスプラントのほとんどは非線形性を有しており、線形制御則を用いた制御系では過渡特性に必要な条件を満足しないことが分かっている。この問題への対応として、小脳演算モデルを用いた PID 制御系 [74] (以降 CMAC-PID 制御系と呼ぶ) が提案されているが、その評価規範から所望の定常特性を得る PID パラメータが得られないという問題がある。そこで、第3章による手法を適用し、CMAC-PID 制御系に含まれる荷重値を更新させることで、所望の定常特性が得られるように CMAC-PID 制御系を適応させるアルゴリズムについて提案している。これにより、非線形システムにおい

て所望の過渡特性および定常特性が得られる制御系が構築できる。最後にコンピュータシミュレーションにより、従来法であるセルフチューニングコントローラと比較することで、過渡特性および定常特性の観点から本手法の有効性を検証している。

第5章では、本研究における成果を総括し、今後検討すべき課題等について述べている。

第2章

直交射影アルゴリズムによる2次元 適応 FIR フィルタの設計

2.1 緒言

デジタル信号処理の分野において1次元適応フィルタは重要な役割を演じているが、最近では2次元適応フィルタも多くの注目が集まっている [8,9]。これは、膨大なデータをリアルタイムで取り扱う必要性から、マルチメディアデータの処理において広く用いられている。2次元適応フィルタで扱われる信号の性質を考えると、画像などの非定常性を含む有色入力信号に対して速い収束特性が必要とされている。1次元の場合には、このような信号に対する収束特性の改善において、直交射影アルゴリズム [31,32] とその高速アルゴリズム [34,35] が有用であることが知られている。このアルゴリズムは RLS アルゴリズムよりも計算量は少なく、NLMS アルゴリズムよりも収束速度が速いという特徴を持つ。2次元データの処理量は膨大であるため、2次元適応フィルタにおいて計算量は非常に重要な要素である。

本章では、1次元直交射影アルゴリズムを2次元に拡張した直交射影適応 FIR フィルタの設計と2次元データの特徴を活かした2次元直交射影アルゴリズムの開発について論ずる。従来の2次元 LMS アルゴリズム [15] と比べ計算量は増加するが、収束速度が入力信号の性質を受けにくいなどの性質を持っている。また上述では、スキャン方向を一方向にして2次元データを処理するアルゴリズムを論じているが、一般的には垂直あるいは水平の場合に対して信号の性質が異なる場合が多い。そこで、水平・垂直の2方向から処理を行う直交

射影アルゴリズムについても同様に考察する。これにより、射影ベクトルの位置を自由に選ぶことができ、収束特性の改善が期待できる。

本章の構成を以下に示す。2.2節で2次元ウィナーフィルタ、2次元LMSアルゴリズム、2次元正規化LMSアルゴリズムについてそれぞれ論じ、図式解釈を通してこれらの問題点を指摘する。2.3節で2次元直交射影アルゴリズムを提案し、収束性について証明している。そして、2.4節で2方向処理による2次元LMSアルゴリズム、2次元正規化LMSアルゴリズムについてそれぞれ論じる。2.5節において2.3節で提案した2次元直交射影アルゴリズムを2方向処理によるアルゴリズムへ拡張し、収束性について証明する。そして、2.6節では、水平・垂直両方向の入力信号を直交化させるアルゴリズムを考察する。最後に、コンピュータシミュレーションにより、提案したアルゴリズムを2次元システム同定に応用した例を示し、妥当性を検証する。

2.2 2次元LMSおよびNLMS適応FIRフィルタ

2.2.1 2次元ウィナーフィルタ

基本的な2次元ウィナーフィルタが図2.1に示されている。ここで、 $D = \{d(i, j) \mid 0 \leq i \leq M-1, 0 \leq j \leq N-1\}$ は $M \times N$ 目標画像、 $U = \{u(i, j) \mid 0 \leq i \leq M-1, 0 \leq j \leq N-1\}$ は $M \times N$ 入力画像、 $d(i, j)$ と $u(i, j)$ はそれぞれの画素で、次の2次元因果性FIRフィルタが使用される。

$$y(i, j) = \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w(p, q) u(i-p, j-q) \quad (2.1)$$

2次元ウィナーフィルタの $m \times n$ 係数マトリクス W を、

$$W = \begin{bmatrix} w(0,0) & w(0,1) & \cdots & w(0,n-1) \\ w(1,0) & w(1,1) & \cdots & w(1,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ w(m-1,0) & w(m-1,1) & \cdots & w(m-1,n-1) \end{bmatrix} \quad (2.2)$$

とおく。点 (i, j) における誤差信号が、

$$e(i, j) = d(i, j) - \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w(p, q) u(i-p, j-q) \quad (2.3)$$

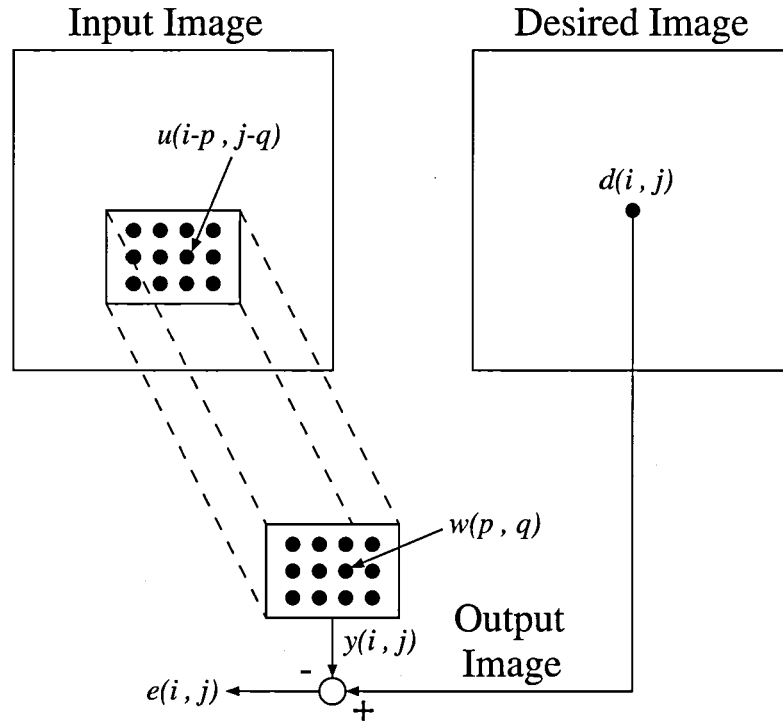


図 2.1 2次元ウィナーフィルタ

で与えられる。このとき、2乗誤差は、

$$\begin{aligned}
 e^2(i, j) &= d^2(i, j) - 2 \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w(p, q) d(i, j) u(i-p, j-q) \\
 &\quad + \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} \sum_{r=0}^{m-1} \sum_{s=0}^{n-1} w(p, q) w(r, s) u(i-p, j-q) u(i-r, j-s) \quad (2.4)
 \end{aligned}$$

となる。

2次元ウィナーフィルタでは、次の平均2乗誤差

$$\begin{aligned}
 J &= E[e^2(i, j)] \\
 &= E[d^2(i, j)] - 2 \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w(p, q) P(p, q) \\
 &\quad + \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} \sum_{r=0}^{m-1} \sum_{s=0}^{n-1} w(p, q) w(r, s) R(r-p, s-q) \quad (2.5)
 \end{aligned}$$

を最小にする $m \times n$ 重みマトリクス \mathbf{W} を見つけることが問題になる。ただし、

$$P(p, q) = E[d(i, j) u(i-p, j-q)]$$

$$R(r-p, s-q) = E[u(i-p, j-q)u(i-r, j-s)]$$

は目標画像 $d(i, j)$ と入力画像 $u(i, j)$ の相互相関関数および $u(i, j)$ の自己相関関数をそれぞれ表している。式 (2.5) は次のように変形できる。

$$J = E[d^2(i, j)] - 2 \sum_{p=0}^{m-1} \mathbf{p}_p^T \mathbf{w}_p + \sum_{p=0}^{m-1} \sum_{r=0}^{m-1} \mathbf{w}_p^T \mathbf{R}_{r-p} \mathbf{w}_r \quad (2.6)$$

ただし、

$$\begin{aligned} \mathbf{w}_p &= \begin{bmatrix} w(p, 0) & w(p, 1) & \dots & w(p, n-1) \end{bmatrix}^T \\ \mathbf{p}_p &= \begin{bmatrix} P(p, 0) & P(p, 1) & \dots & P(p, n-1) \end{bmatrix}^T \\ \mathbf{R}_{r-p} &= \begin{bmatrix} R(r-p, 0) & R(r-p, 1) & \dots & R(r-p, n-1) \\ R(r-p, -1) & R(r-p, 0) & \dots & R(r-p, n-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(r-p, 1-n) & R(r-p, 2-n) & \dots & R(r-p, 0) \end{bmatrix} \end{aligned}$$

式 (2.6) を \mathbf{w}_p で微分し $\mathbf{0}$ とおけば、

$$\frac{\partial J}{\partial \mathbf{w}_p} = -2\mathbf{p}_p + 2 \sum_{r=0}^{m-1} \mathbf{R}_{r-p} \mathbf{w}_r = \mathbf{0} \quad (2.7)$$

となる。これより、定常状態での2次元ウィナーホッフ方程式

$$\begin{aligned} P(p, q) &= \sum_{r=0}^{m-1} \sum_{s=0}^{n-1} w^0(r, s) R(r-p, s-q) \\ p &= 0, 1, \dots, m-1; \quad q = 0, 1, \dots, n-1 \end{aligned} \quad (2.8)$$

が得られる。ここで、 \mathbf{W}^0 を最適ウィナーフィルタの $m \times n$ 係数マトリクスとして定義し、 $w^0(r, s)$ はその第 rs 要素を表すこととすると、式 (2.8) のマトリクス表現は、

$$\begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1 & \dots & \mathbf{R}_{m-1} \\ \mathbf{R}_{-1} & \mathbf{R}_0 & \dots & \mathbf{R}_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{1-m} & \mathbf{R}_{2-m} & \dots & \mathbf{R}_0 \end{bmatrix} (\mathbf{r}\mathbf{s}\mathbf{W}^0)^T = (\mathbf{r}\mathbf{s}\mathbf{P})^T \quad (2.9)$$

で表される。ただし、

$$\mathbf{P} = \begin{bmatrix} P(0, 0) & P(0, 1) & \dots & P(0, n-1) \\ P(1, 0) & P(1, 1) & \dots & P(1, n-1) \\ \vdots & \vdots & \ddots & \vdots \\ P(m-1, 0) & P(m-1, 1) & \dots & P(m-1, n-1) \end{bmatrix}$$

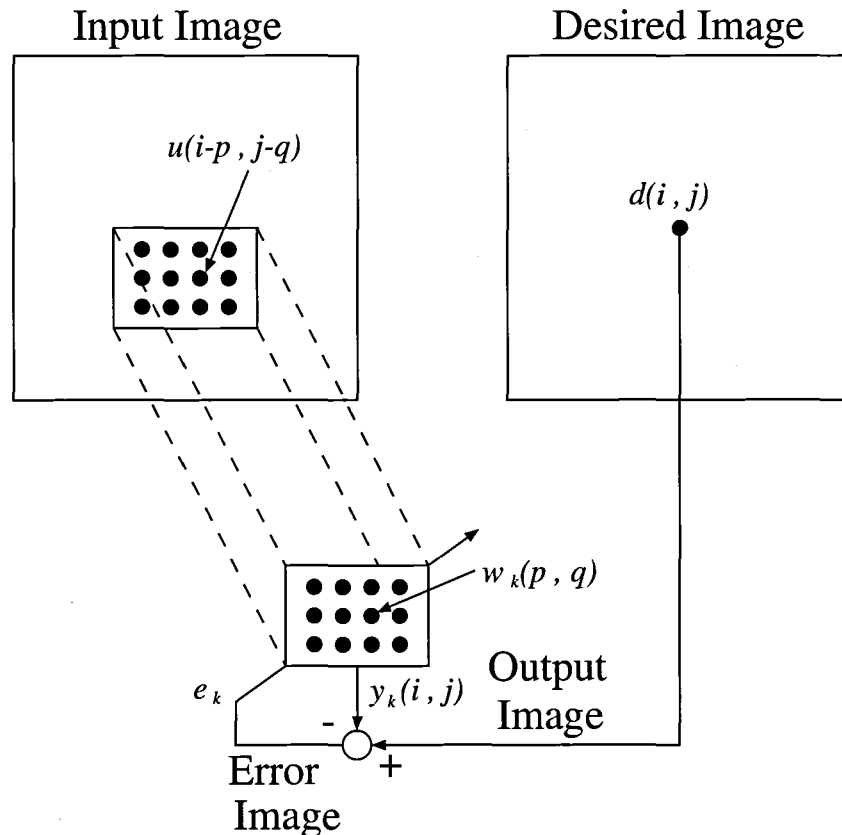


図 2.2 2次元 FIR 適応フィルタ

であり, $\text{rs}P$ はマトリクス P の行展開を表す。最適な $m \times n$ 係数マトリクス W^0 を求めるためには, 式 (2.9) を解くことが必要である。しかし, 自己相関関数や相互相関関数を正確に計算するためには無限個のデータが必要であり, 物理的な実現可能性も含めて完全な解を得ることはできない。この近似解を得るための一つの方法は, 1次元の場合に広く使用されている LMS(Least Mean Square) アルゴリズムを適用することである [11]。

2.2.2 2次元 LMS アルゴリズム

2次元適応 FIR フィルタのブロック線図を図 2.2 に示す。ここで, 次の特性可変な 2次元因果性 FIR フィルタ

$$y_k(i, j) = \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w_k(p, q) u(i-p, j-q) \quad (2.10)$$

が使用されている。ただし、 $k = Mj + i$ ($0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$) である。時刻 k における入力データ窓 U_k は、

$$U_k = \begin{bmatrix} u(i, j) & u(i, j - 1) & \cdots & u(i, j - n + 1) \\ u(i - 1, j) & u(i - 1, j - 1) & \cdots & u(i - 1, j - n + 1) \\ \vdots & \vdots & \ddots & \vdots \\ u(i - m + 1, j) & u(i - m + 1, j - 1) & \cdots & u(i - m + 1, j - n + 1) \end{bmatrix} \quad (2.11)$$

で定義される。 U_k の要素を $u_k(p, q) = u(i - p, j - q)$ で表す。ただし、 $0 \leq p \leq m - 1$, $0 \leq q \leq n - 1$ である。時刻 k における $m \times n$ 係数マトリクス W_k は、

$$W_k = \begin{bmatrix} w_k(0, 0) & w_k(0, 1) & \cdots & w_k(0, n - 1) \\ w_k(1, 0) & w_k(1, 1) & \cdots & w_k(1, n - 1) \\ \vdots & \vdots & \ddots & \vdots \\ w_k(m - 1, 0) & w_k(m - 1, 1) & \cdots & w_k(m - 1, n - 1) \end{bmatrix} \quad (2.12)$$

で表される。このとき、時刻 k における誤差信号は、

$$e_k = d(i, j) - \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w_k(p, q) u(i - p, j - q) \quad (2.13)$$

となる。ここでは、式 (2.9) を満たす最適な係数マトリクス W^0 の解を求めるために、係数マトリクス W_k の更新に最急降下アルゴリズムを使用する。このとき、時刻 k における係数マトリクス W_k の更新式は、次式で与えられる。

$$W_{k+1} = W_k - \mu G_k \quad (2.14)$$

ただし、

$$G_k = \begin{bmatrix} \frac{\partial E[e_k^2]}{\partial w_k(0, 0)} & \cdots & \frac{\partial E[e_k^2]}{\partial w_k(0, n - 1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial E[e_k^2]}{\partial w_k(m - 1, 0)} & \cdots & \frac{\partial E[e_k^2]}{\partial w_k(m - 1, n - 1)} \end{bmatrix} \quad (2.15)$$

であり、 W_k と W_{k+1} は更新前と更新後の係数マトリクスをそれぞれ表し、 μ は収束速度を制御するスカラー量で、ステップサイズパラメータと呼ばれている。式 (2.15) で G_k を直接計算することは困難であるため、 $E[e_k^2]$ の代わりに e_k^2 を用いて通常計算される。この

とき、式 (2.15) は、式 (2.11) および式 (2.13) を用いると、

$$\begin{aligned}
 \mathbf{G}_k &= \begin{bmatrix} \frac{\partial e_k^2}{\partial w_k(0,0)} & \cdots & \frac{\partial e_k^2}{\partial w_k(0,n-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_k^2}{\partial w_k(m-1,0)} & \cdots & \frac{\partial e_k^2}{\partial w_k(m-1,n-1)} \end{bmatrix} \\
 &= 2e_k \begin{bmatrix} \frac{\partial e_k}{\partial w_k(0,0)} & \cdots & \frac{\partial e_k}{\partial w_k(0,n-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_k}{\partial w_k(m-1,0)} & \cdots & \frac{\partial e_k}{\partial w_k(m-1,n-1)} \end{bmatrix} \\
 &= -2e_k \mathbf{U}_k
 \end{aligned} \tag{2.16}$$

となる。よって、式 (2.14) は、

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k \mathbf{U}_k \tag{2.17a}$$

または

$$w_{k+1}(p, q) = w_k(p, q) + 2\mu e_k u(i-p, j-q) \tag{2.17b}$$

に書き換えられる。式 (2.17) は 2次元 LMS 適応 FIR フィルタの係数マトリクスに対する更新アルゴリズムを与えるもので、行列演算、平均操作、微分操作等を一切必要とせずに実現できる。

以下に、LMS 法による具体的なアルゴリズムを示す。

[2次元 LMS アルゴリズム]

1° 初期条件を設定する。

$$\mathbf{W}_0 = \text{任意の初期値} \tag{2.18}$$

2° $i = 0, j = 0, k = 1$ とする。

3° 時刻 k における適応フィルタの出力 $y_k(i, j)$ を求める。

$$y_k(i, j) = (\text{rs}\mathbf{W}_k)(\text{rs}\mathbf{U}_k)^T \tag{2.19}$$

4° 誤差信号 e_k を計算する。

$$e_k = d(i, j) - y_k(i, j) \tag{2.20}$$

5° 係数マトリクス \mathbf{W}_k を更新する。

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k \mathbf{U}_k \tag{2.21}$$

6° $k = k + 1$ とする。

7° $i = M - 1$ まで 3°~6° を繰り返す。

8° $j = N - 1$ まで 3°~7° を繰り返す。

このアルゴリズムの収束は、初期値 W_0 の選択に依存せず、最適な μ を選ぶことによって任意の初期値 W_0 に対して常に保証される。しかし、非定常環境下では、収束速度、定常偏差の間にトレード・オフが存在するため、応用の目的に応じて μ の値を選ぶ必要がある。

2.2.3 2次元正規化 LMS アルゴリズム

2.2.2 節で述べた 2次元 LMS アルゴリズムは、係数マトリクスの更新における情報量が少ないため、入力信号が定数倍に変化するなどの場合、ステップサイズパラメータを調整しなければならないという問題があった。このような問題を解消するため、2次元正規化 LMS アルゴリズムにおいては、適応フィルタの係数マトリクス W_k を次のような手続きにより更新させる。

係数マトリクス W_k の各要素 $w_k(p, q)$ を、対応する入力マトリクス U_k の各要素 $u_k(p, q)$ の大きさに比例して修正する。係数修正マトリクスの大きさは、この入力に対して正しい出力値が出るように、式 (2.13) で表わされる誤差信号に比例して定める。すなわち、時刻 k において入力マトリクス U_k が与えられたとき、係数マトリクス W_k を ΔW_k だけ加えて、それを時刻 $k + 1$ における係数マトリクス W_{k+1} とする。すなわち、次式のよう修正する。

$$\begin{aligned} \Delta W_k &\equiv W_{k+1} - W_k \\ &= (d(i, j) - y_k(i, j)) \frac{U_k}{\|U_k\|^2} \end{aligned} \quad (2.22)$$

$$\|U_k\|^2 \equiv \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} u(i-p, j-q)^2$$

このとき、

$$\begin{aligned} (\text{rs}W_{k+1})(\text{rs}U_k)^T &= (\text{rs}W_k)(\text{rs}U_k)^T + (\text{rs}\Delta W_k)(\text{rs}U_k)^T \\ &= y_k(i, j) + (d(i, j) - y_k(i, j)) \\ &= d(i, j) \end{aligned} \quad (2.23)$$

となるから、時刻 k において修正を完了した係数マトリクス \mathbf{W}_{k+1} は、時刻 k における入力マトリクス \mathbf{U}_k に対して、正しい応答を与えることを示している。

このアルゴリズムは、式 (2.22) のように、入力マトリクスのノルム $\|\mathbf{U}_k\|^2$ で正規化しているため、このような名称で呼ばれている。これにより、LMS アルゴリズムにおいて、入力マトリクス \mathbf{U}_k が大きくなった場合に生じる雑音の増幅を解決している。

係数ベクトルの修正を行う具体的なアルゴリズムは、次のようになる。

[2次元正規化 LMS アルゴリズム]

1° 初期条件を設定する。

$$\mathbf{W}_0 = \text{任意の初期値} \quad (2.24)$$

2° $i = 0, j = 0, k = 1$ とする。

3° 時刻 k における適応フィルタの出力 $y_k(i, j)$ を求める。

$$y_k(i, j) = (\text{rs}\mathbf{W}_k)(\text{rs}\mathbf{U}_k)^T \quad (2.25)$$

4° 誤差信号 e_k を計算する。

$$e_k = d(i, j) - y_k(i, j) \quad (2.26)$$

5° 係数マトリクス \mathbf{W}_k を更新する。

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu \frac{e_k}{\|\mathbf{U}_k\|^2} \mathbf{U}_k \quad (2.27)$$

6° $k = k + 1$ とする。

7° $i = M - 1$ まで 3°~6° を繰り返す。

8° $j = N - 1$ まで 3°~7° を繰り返す。

このアルゴリズムの幾何学的表示を図 2.3 に示している [27]。ここで、 $\mathbf{w}_k = (\text{rs}\mathbf{W}_k)^T$ 、 $\mathbf{w}^0 = (\text{rs}\mathbf{W}^0)^T$ 、 $\mathbf{u}_k = (\text{rs}\mathbf{U}_k)^T$ であり、 $\Pi_k \equiv \{\mathbf{w} \mid \mathbf{w} \in \mathbf{R}^{mn}, \mathbf{w}^T \mathbf{u}_k = d(i, j)\}$ は、入力ベクトル \mathbf{u}_k に対して出力が $d(i, j)$ に等しくなるような係数ベクトルの全体であり、 mn 次元ユークリッド空間中の超平面をなす。図 2.3 に示すように $\mu = 1$ のとき、 \mathbf{w}_{k+1} は \mathbf{w}_k から Π_k に下ろした垂線の足である。 $\mu \leq 0$ あるいは $\mu \geq 2$ ならば、 $\|\mathbf{w}_{k+1} - \mathbf{w}^0\| \geq \|\mathbf{w}_k - \mathbf{w}^0\|$ が成立する。したがって、係数ベクトルが \mathbf{w}^0 に収束するためには、 $0 < \mu < 2$ でなければならない。このとき、 $\|\mathbf{w}_{k+1} - \mathbf{w}^0\| \leq \|\mathbf{w}_k - \mathbf{w}^0\|$ が成立するので、収束は単調である。また、入力 $\{u_k(p, q)\}$ を定数倍しても収束速度は変化しない。LMS アルゴリズムにおいて

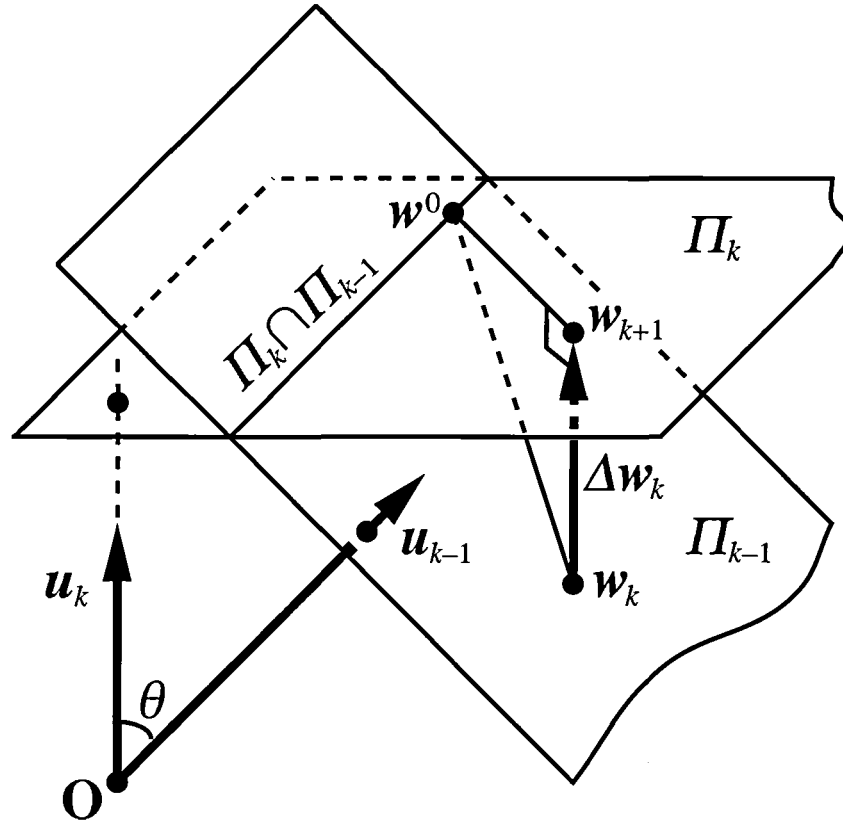


図 2.3 2次元正規化 LMS アルゴリズムの幾何学的表示 ($\mu = 1.0$, $mn = 3$)

は、係数ベクトルの収束の単調性が保証されず、また、収束速度は入力 $\{u_k(p, q)\}$ の振幅を変えるだけで変化してしまう。この意味においては、正規化 LMS アルゴリズムの方が LMS アルゴリズムより好ましい性質を持っていることが分かる。

しかし、図 2.3 から明らかなように、係数ベクトルの収束速度には、 Π_k と Π_{k-1} とのなす角が大きく関係する。すなわち、 Π_k と Π_{k-1} とのなす角が 0 または π に近づくと、

$$\frac{\|w_{k+1} - w^0\|}{\|w_k - w^0\|} \rightarrow 1 \quad (2.28)$$

となり、収束速度が低下する。 Π_k と Π_{k-1} とのなす角を θ とすると、これは u_k と u_{k-1} とのなす角でもあり、

$$\cos \theta = \frac{u_k^T u_{k-1}}{\|u_k\| \|u_{k-1}\|} \quad (2.29)$$

となる。式 (2.29) の右辺は入力信号 $\{u_k\}$ の 1 次の標本自己相関係数にほかならない。したがって、信号の 1 次の自己相関係数の絶対値が 1 に近づくにつれて、収束速度が低下することになる。この現象は、これらのアルゴリズムにおける係数ベクトルの修正方向が u_k

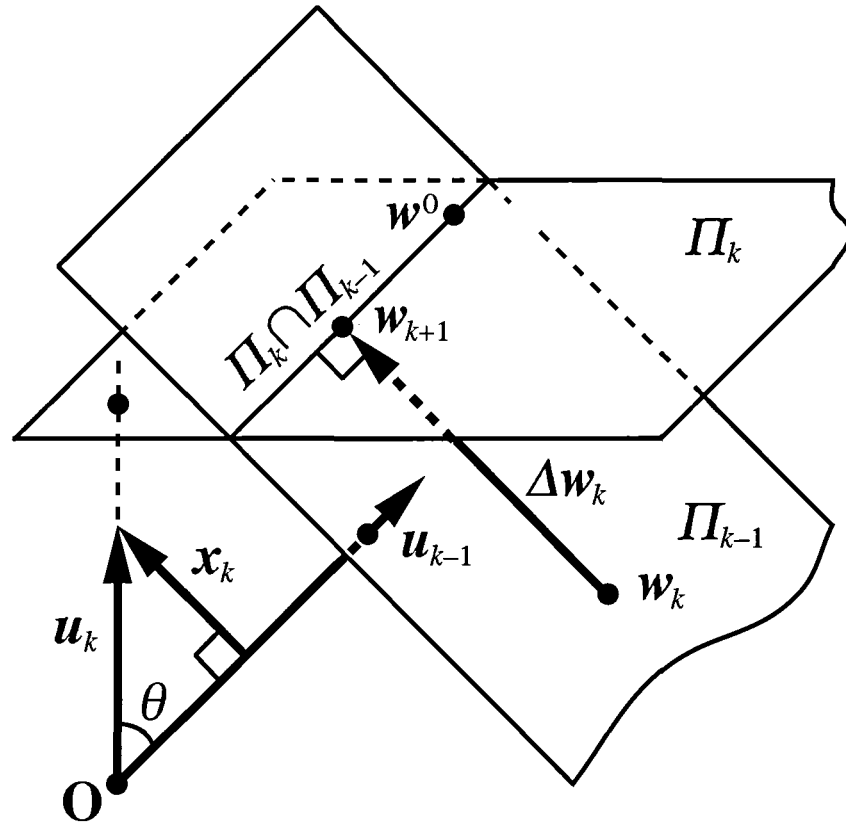


図 2.4 2次元直交射影アルゴリズムの幾何学的表示 ($\mu = 1.0, mn = 3$)

の方向に限られていることが原因であり、これを解決するためには係数ベクトルの修正方向を再検討する必要がある。

2.3 2次元直交射影適応 FIR フィルタの設計

2.3.1 2次元直交射影アルゴリズムの構成

図 2.3 から明らかなように、 u_k と u_{k-1} のなす角によらず一定の収束速度を保つためには、 w_k から Π_k ではなく、 $\Pi_k \cap \Pi_{k-1}$ に垂線を下ろせばよい。このアルゴリズムにおける幾何学的表示を図 2.4 に示す [32]。ここでも、便宜のため $w_k = (rsW_k)^T$ 、 $w^0 = (rsW^0)^T$ 、 $u_k = (rsU_k)^T$ を用いている。このアルゴリズムは、次のように具体化される。

[2次元直交射影アルゴリズム]

1° 初期条件を設定する。

$$W_0 = \text{任意の初期値} \quad (2.30)$$

2° $i = 0, j = 0, k = 1$ とする。

3° \mathbf{u}_k に直交なベクトル \mathbf{x}_k を求める。

$$\mathbf{x}_k = \mathbf{u}_k - \frac{\mathbf{u}_{k-1}^T \mathbf{u}_k}{\|\mathbf{u}_{k-1}\|^2} \mathbf{u}_{k-1} \quad (2.31)$$

ここで, $\|\mathbf{u}_{k-1}\|^2 = \mathbf{u}_{k-1}^T \mathbf{u}_{k-1}$ はベクトルノルムである。

4° 時刻 k における適応フィルタの出力 $y_k(i, j)$ を求める。

$$y_k(i, j) = \mathbf{w}_k^T \mathbf{u}_k \quad (2.32)$$

5° 誤差信号 e_k を計算する。

$$e_k = d(i, j) - y_k(i, j) \quad (2.33)$$

6° 係数ベクトルの補正項を求める。

$$\Delta \mathbf{w}_k = \frac{e_k}{\mathbf{x}_k^T \mathbf{u}_k} \mathbf{x}_k \quad (2.34)$$

7° 係数ベクトル \mathbf{w}_k を更新する。

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \Delta \mathbf{w}_k \quad (2.35)$$

8° $k = k + 1$ とする。

9° $i = M - 1$ まで 3°~8° を繰り返す。

10° $j = N - 1$ まで 3°~9° を繰り返す。

Π を \mathbf{R}^{mn} のアフィン部分空間とすると、 \mathbf{R}^{mn} の点を Π 上に直交射影する写像を \mathbf{P}_Π と書くことにする。 $\mu = 1$ のときの2次元正規化 LMS アルゴリズムは、

$$\mathbf{w}_{k+1} = \mathbf{P}_{\Pi_k}(\mathbf{w}_k) \quad (2.36)$$

によって記述され、2次元直交射影アルゴリズムでは、

$$\mathbf{w}_{k+1} = \mathbf{P}_{\Pi_k \cap \Pi_{k-1}}(\mathbf{w}_k) \quad (2.37)$$

によって、係数ベクトルを修正するアルゴリズムということになる。

2.3.2 収束性の証明

ここで、2次元直交射影アルゴリズムの収束の証明をする。

$\mu = 1$ のときの係数ベクトル更新式を、

$$\tilde{\mathbf{w}}_{k+1} = \mathbf{w}_k + \Delta \mathbf{w}_k \quad (2.38)$$

とすれば、

$$\tilde{\mathbf{w}}_{k+1} = \mathbf{P}_{\Pi_k \cap \Pi_{k-1}}(\mathbf{w}_k) \quad (2.39)$$

であり、 $\mathbf{w}^0 \in \Pi_k \cap \Pi_{k-1}$ が成立するので、 $\tilde{\mathbf{w}}_{k+1} - \mathbf{w}^0$ と $\Delta \mathbf{w}_k$ は直交する。したがって、ピタゴラスの定理より、

$$\|\mathbf{w}_k - \mathbf{w}^0\|^2 = \|\Delta \mathbf{w}_k\|^2 + \|\tilde{\mathbf{w}}_{k+1} - \mathbf{w}^0\|^2 \quad (2.40)$$

となる。また図 2.4 より、

$$\|\mathbf{w}_{k+1} - \mathbf{w}^0\|^2 = \|\tilde{\mathbf{w}}_{k+1} - \mathbf{w}_{k+1}\|^2 + \|\tilde{\mathbf{w}}_{k+1} - \mathbf{w}^0\|^2 \quad (2.41)$$

である。ここで、

$$\begin{aligned} \tilde{\mathbf{w}}_{k+1} - \mathbf{w}_{k+1} &= (\mathbf{w}_k - \Delta \mathbf{w}_k) - (\mathbf{w}_k - \mu \Delta \mathbf{w}_k) \\ &= (\mu - 1) \Delta \mathbf{w}_k \end{aligned} \quad (2.42)$$

であるから、式 (2.42) を式 (2.41) に代入すると、

$$\|\mathbf{w}_{k+1} - \mathbf{w}^0\|^2 = (\mu - 1)^2 \|\Delta \mathbf{w}_k\|^2 + \|\tilde{\mathbf{w}}_{k+1} - \mathbf{w}^0\|^2 \quad (2.43)$$

となる。よって、式 (2.40) と式 (2.43) より、次式を得る。

$$\begin{aligned} \|\mathbf{w}_{k+1} - \mathbf{w}^0\|^2 - \|\mathbf{w}_k - \mathbf{w}^0\|^2 &= \{(\mu - 1)^2 - 1\} \|\Delta \mathbf{w}_k\|^2 \\ &= \mu(\mu - 2) \|\Delta \mathbf{w}_k\|^2 \end{aligned} \quad (2.44)$$

式 (2.44) より、 $0 < \mu < 2$ ならば右辺は負となるため、 $\|\mathbf{w}_{k+1} - \mathbf{w}^0\| \leq \|\mathbf{w}_k - \mathbf{w}^0\|$ が成り立つ。よって、係数ベクトルが \mathbf{w}^0 に収束するためには、 $0 < \mu < 2$ が必要であり、また μ がこの範囲にあるとき、係数ベクトルは決して \mathbf{w}^0 から遠ざかることがない。すなわち、収束は単調であることが分かる。

2.3.3 アルゴリズムの一般化

以上のアルゴリズムを拡張して、次式

$$\mathbf{w}_{k+1} = \mathbf{P}_{\Pi_k \cap \Pi_{k-1} \cap \dots \cap \Pi_{k-r+1}}(\mathbf{w}_k) \quad (2.45)$$

によって係数を修正するアルゴリズムを考える。ここで、 r は直交射影アルゴリズムの次数である。

係数修正ベクトル $\Delta \mathbf{w}_k$ は、次式を満たすように決定される。

$$\begin{cases} d_k &= \mathbf{u}_k^T [\mathbf{w}_k + \Delta \mathbf{w}_k] \\ d_{k-1} &= \mathbf{u}_{k-1}^T [\mathbf{w}_k + \Delta \mathbf{w}_k] \\ &\vdots \\ d_{k-r+1} &= \mathbf{u}_{k-r+1}^T [\mathbf{w}_k + \Delta \mathbf{w}_k] \end{cases} \quad (2.46)$$

ここで、 $d_k = \mathbf{u}_k^T \mathbf{w}^0$ である。式 (2.46) を、マトリクス表示すると、次式のようにになる。

$$\mathbf{d}_k = \tilde{\mathbf{U}}_k^T [\mathbf{w}_k + \Delta \mathbf{w}_k] \quad (2.47)$$

または、等価的に次式のようにになる。

$$\mathbf{e}_k = \tilde{\mathbf{U}}_k^T \Delta \mathbf{w}_k = \mathbf{d}_k - \tilde{\mathbf{U}}_k^T \mathbf{w}_k \quad (2.48)$$

ここで、

$$\begin{aligned} \mathbf{d}_k &= [d_k, d_{k-1}, \dots, d_{k-r+1}]^T \\ \tilde{\mathbf{U}}_k &= [\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-r+1}] \end{aligned}$$

であり、 \mathbf{e}_k は誤差ベクトルと呼ばれている。式 (2.47) において、次数 r が $m \times n$ より小さいとき、係数修正ベクトル $\Delta \mathbf{w}_k$ の解は無数に存在する。しかし、 $\Delta \mathbf{w}_k$ のノルムを最小にするような解をとるとすると、係数修正ベクトルは唯一に求められ、次式のようにになる。

$$\begin{aligned} \Delta \mathbf{w}_k &= (\tilde{\mathbf{U}}_k^T)^+ \mathbf{e}_k \\ &= \tilde{\mathbf{U}}_k [\tilde{\mathbf{U}}_k^T \tilde{\mathbf{U}}_k]^{-1} \mathbf{e}_k \end{aligned} \quad (2.49)$$

ここで、 $(\tilde{\mathbf{U}}_k^T)^+$ は、 $\tilde{\mathbf{U}}_k^T$ の擬似逆マトリクスである [95]。式 (2.49) は、逆マトリクスを含むため、 r が増えるにつれ計算量は r^3 に比例して増加する。

本節で用いた次数 r を使用すると、正規化 LMS アルゴリズムは $r = 1$ のときの直交射影アルゴリズム、2.3.1 節にて提案したアルゴリズムは $r = 2$ のときの直交射影アルゴリズムということになる。

2.4 2方向処理による2次元 LMS および NLMS 適応 FIR フィルタ

2.4.1 2方向処理による2次元 LMS アルゴリズム

図 2.2 に示された2次元適応 FIR フィルタのブロック図において、特性可変な2次元因果性 FIR フィルタが、

$$y(i, j) = \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w_{ij}(p, q) u(i-p, j-q) \quad (2.50)$$

で与えられるものとする。ただし、 $0 \leq i \leq M-1$ 、 $0 \leq j \leq N-1$ である。点 (i, j) における入力データ窓 U_{ij} は、

$$U_{ij} = \begin{bmatrix} u(i, j) & u(i, j-1) & \cdots & u(i, j-n+1) \\ u(i-1, j) & u(i-1, j-1) & \cdots & u(i-1, j-n+1) \\ \vdots & \vdots & \ddots & \vdots \\ u(i-m+1, j) & u(i-m+1, j-1) & \cdots & u(i-m+1, j-n+1) \end{bmatrix} \quad (2.51)$$

で定義される。 U_{ij} の要素を $u_{ij}(p, q) = u(i-p, j-q)$ で表す。ただし、 $0 \leq p \leq m-1$ 、 $0 \leq q \leq n-1$ である。点 (i, j) における $m \times n$ 重み行列 W_{ij} は、

$$W_{ij} = \begin{bmatrix} w_{ij}(0, 0) & w_{ij}(0, 1) & \cdots & w_{ij}(0, n-1) \\ w_{ij}(1, 0) & w_{ij}(1, 1) & \cdots & w_{ij}(1, n-1) \\ \vdots & \vdots & \ddots & \vdots \\ w_{ij}(m-1, 0) & w_{ij}(m-1, 1) & \cdots & w_{ij}(m-1, n-1) \end{bmatrix} \quad (2.52)$$

で表される。このとき、点 (i, j) における誤差信号は、

$$e_{ij} = d(i, j) - \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w_{ij}(p, q) u(i-p, j-q) \quad (2.53)$$

となる。ここでは、式 (2.9) を満たす最適な重み行列 W^0 の近似解を求めるために、重み行列 W_{ij} の更新に (i, j) 平面の水平、垂直の2方向を考慮した拡張型最急降下アルゴリズム

ムを使用する。このとき、点 (i, j) における重み行列 W_{ij} の更新式は、次式で与えられる。

$$W_{i+1,j+1} = f_h W_{i,j+1} + f_v W_{i+1,j} - \mu_h G_{i,j+1} - \mu_v G_{i+1,j} \quad (2.54)$$

ただし、

$$G_{ij} = \begin{bmatrix} \frac{\partial E[e_{ij}^2]}{\partial w_{ij}(0,0)} & \cdots & \frac{\partial E[e_{ij}^2]}{\partial w_{ij}(0,n-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial E[e_{ij}^2]}{\partial w_{ij}(m-1,0)} & \cdots & \frac{\partial E[e_{ij}^2]}{\partial w_{ij}(m-1,n-1)} \end{bmatrix} \quad (2.55)$$

$$f_h + f_v = 1, \quad f_h, f_v \geq 0$$

であり、 $W_{i,j+1}$ 、 $W_{i+1,j}$ と $W_{i+1,j+1}$ は更新前と更新後の重み行列をそれぞれ表し、 μ_h と μ_v はそれぞれ水平、垂直方向の収束速度を制御するステップサイズパラメータである。式 (2.54) で $G_{i,j+1}$ および $G_{i+1,j}$ を直接計算することは困難であるため、 $E[e_{i,j+1}^2]$ と $E[e_{i+1,j}^2]$ の代わりに $e_{i,j+1}^2$ と $e_{i+1,j}^2$ をそれぞれ用いて計算すれば、式 (2.54) は、

$$W_{i+1,j+1} = f_h W_{i,j+1} + f_v W_{i+1,j} + 2\mu_h e_{i,j+1} U_{i,j+1} + 2\mu_v e_{i+1,j} U_{i+1,j} \quad (2.56a)$$

または

$$w_{i+1,j+1}(p,q) = f_h w_{i,j+1}(p,q) + f_v w_{i+1,j}(p,q) + 2\mu_h e_{i,j+1} u(i-p, j+1-q) + 2\mu_v e_{i+1,j} u(i+1-p, j-q) \quad (2.56b)$$

に書き換えられる。式 (2.56) は (i, j) 平面の2方向を考慮した2次元 LMS 適応 FIR フィルタの重み行列の更新アルゴリズムである。これは、行列演算、平均操作、微分操作等は一切必要とせず実現できる。アルゴリズムの収束性は、初期値 W_{00} の選択に依存せず、ステップサイズパラメータ μ_h 、 μ_v の適当な選択によって任意の初期値 W_{00} に対してその収束が常に保証される。なお、 W_{i0} ($i = 1, 2, \dots, M-1$) および W_{0j} ($j = 1, 2, \dots, N-1$) は式 (2.56) の特別な場合として、

$$\begin{aligned} W_{i+1,0} &= W_{i0} + 2\mu_h e_{i0} U_{i0} \\ W_{0,j+1} &= W_{0j} + 2\mu_v e_{0j} U_{0j} \end{aligned} \quad (2.57)$$

によって計算される。

2.4.2 2方向処理による2次元LMSアルゴリズムの収束性

以下では画像が定常な空間的統計量をもつと仮定して、アルゴリズムの収束性について考察する [12]。式 (2.56) を変形すれば、

$$\begin{aligned}
(\text{rs}\mathbf{W}_{i+1,j+1})^T &= [f_h \mathbf{I}_{mn} - 2\mu_h (\text{rs}\mathbf{U}_{i,j+1})^T (\text{rs}\mathbf{U}_{i,j+1})] (\text{rs}\mathbf{W}_{i,j+1})^T \\
&\quad + [f_v \mathbf{I}_{mn} - 2\mu_v (\text{rs}\mathbf{U}_{i+1,j})^T (\text{rs}\mathbf{U}_{i+1,j})] (\text{rs}\mathbf{W}_{i+1,j})^T \\
&\quad + 2\mu_h d(i, j+1) (\text{rs}\mathbf{U}_{i,j+1})^T \\
&\quad + 2\mu_v d(i+1, j) (\text{rs}\mathbf{U}_{i+1,j})^T
\end{aligned} \tag{2.58}$$

となる。ただし、 $\text{rs}\mathbf{W}$ は行列 \mathbf{W} の行展開を表す。式 (2.58) の両辺について期待値をとり、式 (2.5) から式 (2.9) を参照すれば、

$$\begin{aligned}
\mathbb{E}[(\text{rs}\mathbf{W}_{i+1,j+1})^T] &= (f_h \mathbf{I}_{mn} - 2\mu_h \mathbf{R}) \mathbb{E}[(\text{rs}\mathbf{W}_{i,j+1})^T] \\
&\quad + (f_v \mathbf{I}_{mn} - 2\mu_v \mathbf{R}) \mathbb{E}[(\text{rs}\mathbf{W}_{i+1,j})^T] \\
&\quad + 2(\mu_h + \mu_v) \mathbf{R} (\text{rs}\mathbf{W}^0)^T
\end{aligned} \tag{2.59}$$

が得られる。ただし、 $\mathbb{E}[\cdot]$ は期待値を表す。式 (2.59) より、次式が導ける。

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\varepsilon}_{i+1,j+1}] &= (f_h \mathbf{I}_{mn} - 2\mu_h \mathbf{R}) \mathbb{E}[\boldsymbol{\varepsilon}_{i,j+1}] \\
&\quad + (f_v \mathbf{I}_{mn} - 2\mu_v \mathbf{R}) \mathbb{E}[\boldsymbol{\varepsilon}_{i+1,j}]
\end{aligned} \tag{2.60}$$

ただし、

$$\begin{aligned}
\boldsymbol{\varepsilon}_{ij} &= (\text{rs}\mathbf{W}_{i,j})^T - (\text{rs}\mathbf{W}^0)^T \\
\mathbf{R} &= \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1 & \cdots & \mathbf{R}_{m-1} \\ \mathbf{R}_{-1} & \mathbf{R}_0 & \cdots & \mathbf{R}_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{1-m} & \mathbf{R}_{2-m} & \cdots & \mathbf{R}_0 \end{bmatrix}
\end{aligned}$$

$i \rightarrow \infty, j \rightarrow \infty$ のとき $\mathbb{E}[\boldsymbol{\varepsilon}_{ij}] \rightarrow \mathbf{0}$ であれば、 $\mathbb{E}[\mathbf{W}_{ij}] \rightarrow \mathbf{W}^0$ となる。これは、式 (2.60) で表される局所状態空間モデルが漸近安定であることを示している。式 (2.60) は、フォーナシニ・マーケシニモデルの一種であり、これが漸近安定であるための必要十分条件は、領域

$$\bar{U}_2 = \left\{ (z_1^{-1}, z_2^{-1}) \mid |z_1^{-1}| \leq 1, |z_2^{-1}| \leq 1 \right\} \tag{2.61}$$

において、

$$\begin{aligned}
&\det[\mathbf{I}_{mn} - z_1^{-1}(f_h \mathbf{I}_{mn} - 2\mu_h \mathbf{R}) - z_2^{-1}(f_v \mathbf{I}_{mn} - 2\mu_v \mathbf{R})] \\
&= \det[(1 - f_h z_1^{-1} - f_v z_2^{-1}) \mathbf{I}_{mn} + 2(\mu_h z_1^{-1} + \mu_v z_2^{-1}) \mathbf{R}] \neq 0
\end{aligned} \tag{2.62}$$

が成立することである。いま, $mn \times mn$ 対称行列 R を固有値・固有ベクトル分解すれば,

$$R = Q\Lambda Q^T \quad (2.63)$$

となる。ただし,

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{mn})$$

ここで, $\lambda_k (k = 1, 2, \dots, mn)$ は R の固有値, Q は R の固有ベクトルからなる直交行列である。式 (2.62) に式 (2.63) を代入すれば,

$$\begin{aligned} \det \left[(1 - f_h z_1^{-1} - f_v z_2^{-1}) I_{mn} + 2(\mu_h z_1^{-1} + \mu_v z_2^{-1}) \Lambda \right] \\ = \prod_{i=1}^{mn} (1 - (f_h - 2\mu_h \lambda_i) z_1^{-1} - (f_v - 2\mu_v \lambda_i) z_2^{-1}) \neq 0 \end{aligned} \quad (2.64)$$

となる。これは,

$$1 - (f_h - 2\mu_h \lambda_i) z_1^{-1} - (f_v - 2\mu_v \lambda_i) z_2^{-1} \neq 0 \quad (2.65)$$

$$i = 1, 2, \dots, mn$$

に等価である。式 (2.65) が成立するための必要十分条件は,

$$|f_h - 2\mu_h \lambda_i| + |f_v - 2\mu_v \lambda_i| < 1 \quad (2.66)$$

$$i = 1, 2, \dots, mn$$

によって与えられる。収束ファクタ μ_h, μ_v が式 (2.66) を満たすとき, 定常な空間的統計量をもった画像に対して, 最適な重み行列 W^0 への収束性が常に保証される。さらに, $f_h = f_v = 0.5$ かつ $\mu_h = \mu_v = \mu$ のとき, (2.66) の条件は,

$$0 < \mu < \frac{1}{2\lambda_{\max}} \leq \frac{1}{2\lambda_i} \quad (2.67)$$

に書き換えられる。ただし, λ_{\max} は R の最大固有値を表す。

2.4.3 2方向処理による2次元正規化 LMS アルゴリズム

2.2.3 節で述べた2次元正規化 LMS アルゴリズムにおいても, 水平, 垂直方向を考慮したアルゴリズムに拡張することができる。すなわち, 2.4.1 節で述べた LMS アルゴリズムでの係数更新式において, 入力マトリクス $U_{i,j+1}$ および $U_{i+1,j}$ のノルムに比例して, 係

数を修正すればよい。具体的な係数更新式は、次式のようになる。

$$\begin{aligned} \mathbf{W}_{i+1,j+1} &= f_h \mathbf{W}_{i,j+1} + f_v \mathbf{W}_{i+1,j} \\ &\quad + \mu_h e_{i,j+1} \frac{\mathbf{U}_{i,j+1}}{\|\mathbf{U}_{i,j+1}\|^2} + \mu_v e_{i+1,j} \frac{\mathbf{U}_{i+1,j}}{\|\mathbf{U}_{i+1,j}\|^2} \\ \|\mathbf{U}_{ij}\|^2 &\equiv \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} u(i-p, j-q)^2 \end{aligned} \quad (2.68a)$$

または

$$\begin{aligned} w_{i+1,j+1}(p, q) &= f_h w_{i,j+1}(p, q) + f_v w_{i+1,j}(p, q) \\ &\quad + \mu_h e_{i,j+1} \frac{u(i-p, j+1-q)}{\sum_{p=0}^{m-1} \sum_{q=0}^{n-1} u(i-p, j+1-q)^2} \\ &\quad + \mu_v e_{i+1,j} \frac{u(i+1-p, j-q)}{\sum_{p=0}^{m-1} \sum_{q=0}^{n-1} u(i+1-p, j-q)^2} \end{aligned} \quad (2.68b)$$

よって、式 (2.53) と式 (2.68) で係数を更新している。これは、行列演算を必要とし、LMS アルゴリズムに比べて計算量は多少増加する。しかし、2.2.3 節で述べたように、収束速度は入力 の性質に左右されにくくなる。

2.5 2方向処理による2次元直交射影適応 FIR フィルタの設計

2.5.1 2方向処理による2次元直交射影アルゴリズムの構成

2.3 節で述べた2次元直交射影アルゴリズムにおいても、水平・垂直方向を考慮した2方向処理によるアルゴリズムに拡張することができる。

具体的な係数更新式について以下に説明する。まず、水平方向における入力 $\mathbf{u}_{i,j+1}$ に直交なベクトル $\mathbf{x}_{i,j+1}^h$ 、及び垂直方向における入力 $\mathbf{u}_{i+1,j}$ に直交なベクトル $\mathbf{x}_{i+1,j}^v$ をそれぞれ求める。

$$\mathbf{x}_{i,j+1}^h = \mathbf{u}_{i,j+1} - \frac{\mathbf{u}_{i-1,j+1}^T \mathbf{u}_{i,j+1}}{\|\mathbf{u}_{i-1,j+1}\|^2} \mathbf{u}_{i-1,j+1} \quad (2.69)$$

$$\mathbf{x}_{i+1,j}^v = \mathbf{u}_{i+1,j} - \frac{\mathbf{u}_{i+1,j-1}^T \mathbf{u}_{i+1,j}}{\|\mathbf{u}_{i+1,j-1}\|^2} \mathbf{u}_{i+1,j-1} \quad (2.70)$$

次に、式 (2.53) で求めた誤差信号を用いて、係数を更新する。

$$\begin{aligned} \mathbf{w}_{i+1,j+1} &= f_h \mathbf{w}_{i,j+1} + f_v \mathbf{w}_{i+1,j} \\ &\quad + \mu_h \frac{e_{i,j+1}}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \mathbf{x}_{i,j+1}^h + \mu_v \frac{e_{i+1,j}}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \mathbf{x}_{i+1,j}^v \end{aligned} \quad (2.71)$$

ただし, $\mathbf{u}_{i,j} = (\text{rs}U_{i,j})^T$, $\mathbf{w}_{i,j} = (\text{rs}W_{i,j})^T$ である。

2.5.2 2方向処理による2次元直交射影アルゴリズムの収束性の証明

以下では, 入力定常かつ独立であると仮定して, アルゴリズムの収束性を考察する。式 (2.53) を用いて式 (2.71) を変形すれば,

$$\begin{aligned}
\mathbf{w}_{i+1,j+1} &= f_h \mathbf{w}_{i,j+1} + f_v \mathbf{w}_{i+1,j} \\
&\quad + \mu_h \frac{d(i,j+1) - \mathbf{w}_{i,j+1}^T \mathbf{u}_{i,j+1}}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \mathbf{x}_{i,j+1}^h \\
&\quad + \mu_v \frac{d(i+1,j) - \mathbf{w}_{i+1,j}^T \mathbf{u}_{i+1,j}}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \mathbf{x}_{i+1,j}^v \\
&= \left\{ f_h \mathbf{I}_{mn} - \mu_h \frac{[\mathbf{x}_{i,j+1}^h \mathbf{u}_{i,j+1}^T]}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \right\} \mathbf{w}_{i+1,j} \\
&\quad + \left\{ f_v \mathbf{I}_{mn} - \mu_v \frac{[\mathbf{x}_{i+1,j}^v \mathbf{u}_{i+1,j}^T]}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \right\} \mathbf{w}_{i,j+1} \tag{2.72} \\
&\quad + \mu_h \frac{d(i,j+1)}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \mathbf{x}_{i,j+1}^h + \mu_v \frac{d(i+1,j)}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \mathbf{x}_{i+1,j}^v
\end{aligned}$$

となる。ここで, $(\mathbf{w}_{i,j}^T \mathbf{u}_{i,j}) \mathbf{x}_{i,j} = [\mathbf{x}_{i,j} \mathbf{u}_{i,j}^T] \mathbf{w}_{i,j}$ となる変換を用いている。式 (2.72) において, $d(i,j+1) = (\mathbf{w}^0)^T \mathbf{u}_{i,j+1}$, $d(i+1,j) = (\mathbf{w}^0)^T \mathbf{u}_{i+1,j}$ を代入して, 両辺について期待値をとると,

$$\begin{aligned}
\mathbf{E}[\mathbf{w}_{i+1,j+1}] &= \left\{ f_h \mathbf{I}_{mn} - \mu_h \mathbf{E} \left[\frac{[\mathbf{x}_{i,j+1}^h \mathbf{u}_{i,j+1}^T]}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \right] \right\} \mathbf{E}[\mathbf{w}_{i+1,j}] \\
&\quad + \left\{ f_v \mathbf{I}_{mn} - \mu_v \mathbf{E} \left[\frac{[\mathbf{x}_{i+1,j}^v \mathbf{u}_{i+1,j}^T]}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \right] \right\} \mathbf{E}[\mathbf{w}_{i,j}] \\
&\quad + \mu_h \mathbf{E} \left[\frac{[\mathbf{x}_{i,j+1}^h \mathbf{u}_{i,j+1}^T]}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \right] \mathbf{E}[\mathbf{w}^0] + \mu_v \mathbf{E} \left[\frac{[\mathbf{x}_{i+1,j}^v \mathbf{u}_{i+1,j}^T]}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \right] \mathbf{E}[\mathbf{w}^0] \tag{2.73}
\end{aligned}$$

となる。ここで,

$$\tilde{\mathbf{R}}^h = \mathbf{E} \left[\frac{[\mathbf{x}_{i,j+1}^h \mathbf{u}_{i,j+1}^T]}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \right] \tag{2.74}$$

$$\tilde{\mathbf{R}}^v = \mathbf{E} \left[\frac{\begin{bmatrix} \mathbf{x}_{i+1,j}^v \mathbf{u}_{i+1,j}^T \end{bmatrix}}{\mathbf{u}_{i+1,j}^v \mathbf{x}_{i+1,j}^v} \right] \quad (2.75)$$

とおく。式 (2.74), 式 (2.75) に式 (2.69) を代入すると, スカラー部 (分母) は次式

$$\begin{aligned} \mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h &= \mathbf{u}_{i,j+1}^T \left\{ \mathbf{u}_{i,j+1} - \frac{\mathbf{u}_{i-1,j+1}^T \mathbf{u}_{i,j+1}}{\|\mathbf{u}_{i-1,j+1}\|^2} \mathbf{u}_{i-1,j+1} \right\} \\ &= \mathbf{u}_{i,j+1}^T \mathbf{u}_{i,j+1} - \frac{\mathbf{u}_{i-1,j+1}^T \mathbf{u}_{i,j+1}}{\|\mathbf{u}_{i-1,j+1}\|^2} \mathbf{u}_{i,j+1}^T \mathbf{u}_{i-1,j+1} \end{aligned} \quad (2.76)$$

$$\begin{aligned} \mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^h &= \mathbf{u}_{i+1,j}^T \left\{ \mathbf{u}_{i+1,j} - \frac{\mathbf{u}_{i+1,j-1}^T \mathbf{u}_{i+1,j}}{\|\mathbf{u}_{i+1,j-1}\|^2} \mathbf{u}_{i+1,j-1} \right\} \\ &= \mathbf{u}_{i+1,j}^T \mathbf{u}_{i+1,j} - \frac{\mathbf{u}_{i+1,j-1}^T \mathbf{u}_{i+1,j}}{\|\mathbf{u}_{i+1,j-1}\|^2} \mathbf{u}_{i+1,j}^T \mathbf{u}_{i+1,j-1} \end{aligned} \quad (2.77)$$

となり, マトリクス部 (分子) は次式

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{i,j+1}^h \mathbf{u}_{i,j+1}^T \end{bmatrix} &= \left\{ \mathbf{u}_{i,j+1} - \frac{\mathbf{u}_{i-1,j+1}^T \mathbf{u}_{i,j+1}}{\|\mathbf{u}_{i-1,j+1}\|^2} \mathbf{u}_{i-1,j+1} \right\} \mathbf{u}_{i,j+1}^T \\ &= \begin{bmatrix} \mathbf{u}_{i,j+1} \mathbf{u}_{i,j+1}^T \end{bmatrix} - \frac{\mathbf{u}_{i-1,j+1}^T \mathbf{u}_{i,j+1}}{\|\mathbf{u}_{i-1,j+1}\|^2} \begin{bmatrix} \mathbf{u}_{i-1,j+1} \mathbf{u}_{i,j+1}^T \end{bmatrix} \end{aligned} \quad (2.78)$$

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{i+1,j}^h \mathbf{u}_{i+1,j}^T \end{bmatrix} &= \left\{ \mathbf{u}_{i+1,j} - \frac{\mathbf{u}_{i+1,j-1}^T \mathbf{u}_{i+1,j}}{\|\mathbf{u}_{i+1,j-1}\|^2} \mathbf{u}_{i+1,j-1} \right\} \mathbf{u}_{i+1,j}^T \\ &= \begin{bmatrix} \mathbf{u}_{i+1,j} \mathbf{u}_{i+1,j}^T \end{bmatrix} - \frac{\mathbf{u}_{i+1,j-1}^T \mathbf{u}_{i+1,j}}{\|\mathbf{u}_{i+1,j-1}\|^2} \begin{bmatrix} \mathbf{u}_{i+1,j-1} \mathbf{u}_{i+1,j}^T \end{bmatrix} \end{aligned} \quad (2.79)$$

となる。ここで, 仮定より入力信号は独立であるので, 式 (2.78), 式 (2.79) の第2項の期待値をとったものは無視することができる。よって, 次式のように書き換えられる。

$$\begin{aligned} \tilde{\mathbf{R}}^h &\simeq \mathbf{E} \left[\frac{\begin{bmatrix} \mathbf{u}_{i,j+1} \mathbf{u}_{i,j+1}^T \end{bmatrix}}{\mathbf{u}_{i,j+1}^T \mathbf{u}_{i,j+1}} \right] \\ &\simeq \frac{\mathbf{E} \left[\begin{bmatrix} \mathbf{u}_{i,j+1} \mathbf{u}_{i,j+1}^T \end{bmatrix} \right]}{\mathbf{E} \left[\|\mathbf{u}_{i,j+1}\|^2 \right]} \end{aligned} \quad (2.80)$$

$$\begin{aligned} \tilde{\mathbf{R}}^v &\simeq \mathbf{E} \left[\frac{\begin{bmatrix} \mathbf{u}_{i+1,j} \mathbf{u}_{i+1,j}^T \end{bmatrix}}{\mathbf{u}_{i+1,j}^T \mathbf{u}_{i+1,j}} \right] \\ &\simeq \frac{\mathbf{E} \left[\begin{bmatrix} \mathbf{u}_{i+1,j} \mathbf{u}_{i+1,j}^T \end{bmatrix} \right]}{\mathbf{E} \left[\|\mathbf{u}_{i+1,j}\|^2 \right]} \end{aligned} \quad (2.81)$$

式(2.80), 式(2.81)での分子の値は, 入力自己相関関数を表わしており, 両式とも次式

$$\begin{aligned}\tilde{\mathbf{R}} &= \tilde{\mathbf{R}}^h = \tilde{\mathbf{R}}^v \\ &= \frac{1}{\mathbb{E}[\|\mathbf{u}_{i,j}\|^2]} \mathbf{R}\end{aligned}\quad (2.82)$$

のように書き表わされる。よって, 式(2.75)は, 次のように書き換えられる。

$$\begin{aligned}\mathbb{E}[\mathbf{w}_{i+1,j+1}] &= \left\{ f_h \mathbf{I}_{mn} - \mu_h \tilde{\mathbf{R}} \right\} \mathbb{E}[\mathbf{w}_{i,j+1}] + \left\{ f_v \mathbf{I}_{mn} - \mu_v \tilde{\mathbf{R}} \right\} \mathbb{E}[\mathbf{w}_{i+1,j}] \\ &\quad + (\mu_h + \mu_v) \tilde{\mathbf{R}} \mathbb{E}[\mathbf{w}^0]\end{aligned}\quad (2.83)$$

ここで, $\varepsilon_{i,j} = \mathbf{w}_{i,j} - \mathbf{w}^0$ とおくと, 式(2.83)は, 次式のように変形できる。

$$\begin{aligned}\mathbb{E}[\varepsilon_{i+1,j+1}] &= \left(f_h \mathbf{I}_{mn} - \mu_h \tilde{\mathbf{R}} \right) \mathbb{E}[\varepsilon_{i,j+1}] \\ &\quad + \left(f_v \mathbf{I}_{mn} - \mu_v \tilde{\mathbf{R}} \right) \mathbb{E}[\varepsilon_{i+1,j}]\end{aligned}\quad (2.84)$$

式(2.84)において, $i \rightarrow \infty, j \rightarrow \infty$ のとき $\mathbb{E}[\varepsilon_{ij}] \rightarrow \mathbf{0}$ であれば, $\mathbb{E}[\mathbf{w}_{ij}] \rightarrow \mathbf{w}^0$ となる。これが漸近安定であるための必要十分条件は, 式(2.61)で表わされる領域において,

$$\begin{aligned}\det[\mathbf{I}_{mn} - z_1^{-1}(f_h \mathbf{I}_{mn} - \mu_h \tilde{\mathbf{R}}) - z_2^{-1}(f_v \mathbf{I}_{mn} - \mu_v \tilde{\mathbf{R}})] \\ = \det[(1 - f_h z_1^{-1} - f_v z_2^{-1}) \mathbf{I}_{mn} + (\mu_h z_1^{-1} + \mu_v z_2^{-1}) \tilde{\mathbf{R}}] \neq 0\end{aligned}\quad (2.85)$$

が成立することである。いま, $mn \times mn$ 対称行列 \mathbf{R} を固有値・固有ベクトル分解すれば,

$$\mathbf{R} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \quad (2.86)$$

となる。ただし,

$$\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{mn}) \quad (2.87)$$

ここで, $\lambda_k (k = 1, 2, \dots, mn)$ は \mathbf{R} の固有値, \mathbf{Q} は \mathbf{R} の固有ベクトルからなる直交行列である。 \mathbf{R} の固有値 $\lambda_k (k = 1, 2, \dots, mn)$ は, 次式を満たす。

$$\begin{aligned}\sum_{k=1}^{mn} \lambda_k &= \text{tr} \mathbf{\Lambda} \\ &= \det[\mathbf{u}_{i,j} \mathbf{u}_{i,j}^T] \\ &= \|\mathbf{u}_{i,j}\|^2\end{aligned}\quad (2.88)$$

式 (2.88) より, $\tilde{\mathbf{R}}$ は入力ベクトルのノルム $\|\mathbf{u}_{i,j}\|^2$ を含んでいるため,

$$\begin{aligned}
 \tilde{\mathbf{R}} &= \frac{\mathbf{R}}{\mathbb{E}[\|\mathbf{u}_{i,j}\|^2]} \\
 &= \mathbf{Q} \frac{\mathbf{\Lambda}}{\mathbb{E}[\|\mathbf{u}_{i,j}\|^2]} \mathbf{Q}^T \\
 &\simeq \mathbf{Q} \frac{\|\mathbf{u}_{i,j}\|^2 \mathbf{I}_{mn}}{\mathbb{E}[\|\mathbf{u}_{i,j}\|^2]} \mathbf{Q}^T \\
 &\simeq \mathbf{Q} \mathbf{I}_{mn} \mathbf{Q}^T \\
 &= \mathbf{I}_{mn}
 \end{aligned} \tag{2.89}$$

となる。よって, 式 (2.85) は, 次式となる。

$$\begin{aligned}
 \det \left[(1 - f_h z_1^{-1} - f_v z_2^{-1}) \mathbf{I}_{mn} + (\mu_h z_1^{-1} + \mu_v z_2^{-1}) \mathbf{I}_{mn} \right] \\
 = 1 - (f_h - \mu_h) z_1^{-1} - (f_v - \mu_v) z_2^{-1} \neq 0
 \end{aligned} \tag{2.90}$$

式 (2.90) が成立するための必要十分条件は,

$$|f_h - \mu_h| + |f_v - \mu_v| < 1 \tag{2.91}$$

によって与えられる。ステップサイズパラメータ μ_h, μ_v が式 (2.91) を満たすとき, 定常な空間的統計量をもった画像に対して, 最適な重みベクトル \mathbf{w}^0 への収束性が常に保証される。さらに, $f_h = f_v = 0.5$ かつ $\mu_h = \mu_v = \mu$ のとき, 式 (2.91) の条件は,

$$0 < \mu < 1 \tag{2.92}$$

に書き換えられる。

2.5.3 アルゴリズムの一般化

Π を \mathbf{R}^{mn} のアフィン部分空間とするとき, \mathbf{R}^{mn} の点を Π 上に直交射影する写像を \mathbf{P}_Π と書くことにすると, 水平, 垂直方向を考慮した2次元直交射影アルゴリズムは,

$$\mathbf{w}_{i+1,j+1} = f_h \mathbf{P}_{\Pi_{i,j+1} \cap \Pi_{i-1,j+1}}(\mathbf{w}_{i,j+1}) + f_v \mathbf{P}_{\Pi_{i+1,j} \cap \Pi_{i+1,j-1}}(\mathbf{w}_{i+1,j}) \tag{2.93}$$

ということになる。以上のアルゴリズムを拡張して,

$$\begin{aligned}
 \mathbf{w}_{i+1,j+1} &= f_h \mathbf{P}_{\Pi_{i,j+1} \cap \Pi_{i-1,j+1} \cap \dots \cap \Pi_{i-r+1,j+1}}(\mathbf{w}_{i,j+1}) \\
 &\quad + f_v \mathbf{P}_{\Pi_{i+1,j} \cap \Pi_{i+1,j-1} \cap \dots \cap \Pi_{i+1,j-s+1}}(\mathbf{w}_{i+1,j})
 \end{aligned} \tag{2.94}$$

によって係数を修正するアルゴリズムを考える。ここで、 r および s は、それぞれ直交射影アルゴリズムの水平、垂直方向の次数である。

係数修正ベクトル $\Delta \mathbf{w}_{i,j+1}$ 、 $\Delta \mathbf{w}_{i+1,j}$ は、次式を満たすように決定される。

$$\begin{cases} d_{i,j+1} &= \mathbf{u}_{i,j+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ d_{i-1,j+1} &= \mathbf{u}_{i-1,j+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ &\vdots \\ d_{i-r+1,j+1} &= \mathbf{u}_{i-r+1,j+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \end{cases} \quad (2.95)$$

$$\begin{cases} d_{i+1,j} &= \mathbf{u}_{i+1,j}^T [\mathbf{w}_{i+1,j} + \Delta \mathbf{w}_{i+1,j}] \\ d_{i+1,j-1} &= \mathbf{u}_{i+1,j-1}^T [\mathbf{w}_{i+1,j} + \Delta \mathbf{w}_{i+1,j}] \\ &\vdots \\ d_{i+1,j-s+1} &= \mathbf{u}_{i+1,j-s+1}^T [\mathbf{w}_{i+1,j} + \Delta \mathbf{w}_{i+1,j}] \end{cases} \quad (2.96)$$

マトリクス表示すると、次式のようにになる。

$$\mathbf{d}_{i,j+1}^h = \left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \quad (2.97)$$

$$\mathbf{d}_{i+1,j}^v = \left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T [\mathbf{w}_{i+1,j} + \Delta \mathbf{w}_{i+1,j}] \quad (2.98)$$

または、等価的に次式のようにになる。

$$\mathbf{e}_{i,j+1}^h = \left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T \Delta \mathbf{w}_{i,j+1} = \mathbf{d}_{i,j+1}^h - \left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T \mathbf{w}_{i,j+1} \quad (2.99)$$

$$\mathbf{e}_{i+1,j}^v = \left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T \Delta \mathbf{w}_{i+1,j} = \mathbf{d}_{i+1,j}^v - \left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T \mathbf{w}_{i+1,j} \quad (2.100)$$

ここで、

$$\begin{aligned} \mathbf{d}_{i,j+1}^h &= [d_{i,j+1}, d_{i-1,j+1}, \dots, d_{i-r+1,j+1}]^T \\ \mathbf{d}_{i+1,j}^v &= [d_{i+1,j}, d_{i+1,j-1}, \dots, d_{i+1,j-s+1}]^T \\ \tilde{\mathbf{U}}_{i,j+1}^h &= [\mathbf{u}_{i,j+1}, \mathbf{u}_{i-1,j+1}, \dots, \mathbf{u}_{i-r+1,j+1}] \\ \tilde{\mathbf{U}}_{i+1,j}^v &= [\mathbf{u}_{i+1,j}, \mathbf{u}_{i+1,j-1}, \dots, \mathbf{u}_{i+1,j-s+1}] \end{aligned}$$

であり、 $\mathbf{e}_{i,j+1}^h$ 、 $\mathbf{e}_{i+1,j}^v$ は誤差ベクトルと呼ばれている。式 (2.99)、式 (2.100) において、次数 r, s が $m \times n$ より小さいとき、係数修正ベクトル $\Delta \mathbf{w}_{i,j+1}$ 、 $\Delta \mathbf{w}_{i+1,j}$ の解は無数に存在する。しかし、 $\Delta \mathbf{w}_{i,j+1}$ 、 $\Delta \mathbf{w}_{i+1,j}$ のノルムを最小にするような解をとるとすると、係

数修正ベクトルは唯一に求められ、次式のようになる。

$$\begin{aligned}\Delta \mathbf{w}_{i,j+1} &= \left(\left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T \right)^+ \mathbf{e}_{i,j+1}^h \\ &= \tilde{\mathbf{U}}_{i,j+1}^h \left[\left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T \tilde{\mathbf{U}}_{i,j+1}^h \right]^{-1} \mathbf{e}_{i,j+1}^h\end{aligned}\quad (2.101)$$

$$\begin{aligned}\Delta \mathbf{w}_{i+1,j} &= \left(\left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T \right)^+ \mathbf{e}_{i+1,j}^v \\ &= \tilde{\mathbf{U}}_{i+1,j}^v \left[\left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T \tilde{\mathbf{U}}_{i+1,j}^v \right]^{-1} \mathbf{e}_{i+1,j}^v\end{aligned}\quad (2.102)$$

ここで、 $\left(\left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T \right)^+$ 、 $\left(\left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T \right)^+$ は、それぞれ $\left(\tilde{\mathbf{U}}_{i,j+1}^h \right)^T$ 、 $\left(\tilde{\mathbf{U}}_{i+1,j}^v \right)^T$ の擬似逆マトリクスである。式 (2.101) および式 (2.102) は、逆マトリクスを含むため、 r, s が増えるにつれ計算量はそれぞれ r^3, s^3 に比例して増加する。

本節で用いた次数 r, s を使用すると、正規化 LMS アルゴリズムは $r = s = 1$ のときの直交射影アルゴリズムと、2.5.1 節のアルゴリズムは $r = s = 2$ のときの直交射影アルゴリズムということになる。

2.6 2方向の入力を直交化させる2次元直交射影適応 FIR フィルタの設計

2.6.1 アルゴリズムの構成

本節では、水平、垂直方向の入力を直交化させるアルゴリズムを考える。2.3 節では、係数マトリクスを $\Pi_k \cap \Pi_{k-1} \cap \cdots \cap \Pi_{k-r+1}$ すなわち、 $\Pi_{i,j} \cap \Pi_{i-1,j} \cap \cdots \cap \Pi_{i-r+1,j}$ で表わされる超平面において直交化させるアルゴリズムを論じたが、2次元データにおいて、必ずしも直交化させるために用いる入力ベクトルをスキャン方向に限定することはない。例えば、 $\mathbf{u}_{i,j}, \mathbf{u}_{i+1,j-1}, \mathbf{u}_{i,j-1}$ によって形成される超平面を用いて、係数ベクトル $\mathbf{w}_{i,j}$ を更新させてもよいということになる。

2次元データは、1次元のときに比べ自由度が大きいいため、このようなアルゴリズムは2.3.1 節で述べたアルゴリズムよりも好ましい方法であるといえる。以下では、係数ベクトル $\mathbf{w}_{i+1,j+1}$ を求めるために、 $(i+1, j+1)$ において水平方向へ1つ遅延させた入力ベクトル $\mathbf{u}_{i,j+1}$ と、垂直方向へ1つ遅延させた入力ベクトル $\mathbf{u}_{i+1,j}$ を用いて、係数ベクトルを更

新するアルゴリズムを考察する。これを式で表わすと次式のようにになる。

$$\mathbf{w}_{i+1,j+1} = f_h \mathbf{P}_{\Pi_{i,j+1} \cap \Pi_{i+1,j}}(\mathbf{w}_{i,j+1}) + f_v \mathbf{P}_{\Pi_{i+1,j} \cap \Pi_{i,j+1}}(\mathbf{w}_{i+1,j}) \quad (2.103)$$

式(2.103)は、2.5.1節で提案したアルゴリズムの変形であるが、このように、水平・垂直方向の入力を直交化させることでよりよく2次元データを推測できると思われる。式(2.103)における具体的なアルゴリズムは、以下のようにになる。

$$\mathbf{x}_{i,j+1}^h = \mathbf{u}_{i,j+1} - \frac{\mathbf{u}_{i+1,j}^T \mathbf{u}_{i,j+1}}{\|\mathbf{u}_{i+1,j}\|^2} \mathbf{u}_{i+1,j} \quad (2.104)$$

$$\mathbf{x}_{i+1,j}^v = \mathbf{u}_{i+1,j} - \frac{\mathbf{u}_{i,j+1}^T \mathbf{u}_{i+1,j}}{\|\mathbf{u}_{i,j+1}\|^2} \mathbf{u}_{i,j+1} \quad (2.105)$$

$$e_{i,j} = d(i,j) - \mathbf{w}_{i,j}^T \mathbf{u}_{i,j} \quad (2.106)$$

$$\begin{aligned} \mathbf{w}_{i+1,j+1} &= f_h \mathbf{w}_{i,j+1} + f_v \mathbf{w}_{i+1,j} \\ &\quad + \mu_h \frac{e_{i,j+1}}{\mathbf{u}_{i,j+1}^T \mathbf{x}_{i,j+1}^h} \mathbf{x}_{i,j+1}^h + \mu_v \frac{e_{i+1,j}}{\mathbf{u}_{i+1,j}^T \mathbf{x}_{i+1,j}^v} \mathbf{x}_{i+1,j}^v \end{aligned} \quad (2.107)$$

2.6.2 アルゴリズムの一般化

以上のアルゴリズムを拡張して、

$$\begin{aligned} \mathbf{w}_{i+1,j+1} &= f_h \mathbf{P}_{\Pi_{i,j+1} \cap \Pi_{i-1,j+1} \cap \dots \cap \Pi_{i-r_1+1,j+1} \cap \Pi_{i+1,j} \cap \Pi_{i+1,j-1} \cap \dots \cap \Pi_{i+1,j-r_2+1}}(\mathbf{w}_{i,j+1}) \\ &\quad + f_v \mathbf{P}_{\Pi_{i+1,j} \cap \Pi_{i+1,j-1} \cap \dots \cap \Pi_{i+1,j-s_1+1} \cap \Pi_{i,j+1} \cap \Pi_{i-1,j+1} \cap \dots \cap \Pi_{i+s_2+1,j+1}}(\mathbf{w}_{i+1,j}) \end{aligned} \quad (2.108)$$

によって係数を修正するアルゴリズムを考える。ここで、 r_1, r_2 および s_1, s_2 は、それぞれ直交射影アルゴリズムの水平、垂直方向の次数である。

係数修正ベクトル $\Delta \mathbf{w}_{i,j+1}$, $\Delta \mathbf{w}_{i+1,j}$ は、次式を満たすように決定される。

$$\left\{ \begin{array}{l} d(i,j+1) = \mathbf{u}_{i,j+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ d(i-1,j+1) = \mathbf{u}_{i-1,j+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ \vdots \\ d(i-r_1+1,j+1) = \mathbf{u}_{i-r_1+1,j+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ d(i+1,j) = \mathbf{u}_{i+1,j}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ d(i+1,j-1) = \mathbf{u}_{i+1,j-1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \\ \vdots \\ d(i+1,j-r_2+1) = \mathbf{u}_{i+1,j-r_2+1}^T [\mathbf{w}_{i,j+1} + \Delta \mathbf{w}_{i,j+1}] \end{array} \right. \quad (2.109)$$

$$\left\{ \begin{array}{l} d(i+1, j) = \mathbf{u}_{i+1, j}^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \\ d(i+1, j-1) = \mathbf{u}_{i+1, j-1}^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \\ \vdots \\ d(i+1, j-s_1+1) = \mathbf{u}_{i+1, j-s_1+1}^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \\ d(i, j+1) = \mathbf{u}_{i, j+1}^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \\ d(i-1, j+1) = \mathbf{u}_{i-1, j+1}^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \\ \vdots \\ d(i-s_2+1, j+1) = \mathbf{u}_{i-s_2+1, j+1}^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \end{array} \right. \quad (2.110)$$

マトリクス表示すると、次式のようになる。

$$\mathbf{d}_{i, j+1}^h = \left(\tilde{\mathbf{U}}_{i, j+1}^h \right)^T [\mathbf{w}_{i, j+1} + \Delta \mathbf{w}_{i, j+1}] \quad (2.111)$$

$$\mathbf{d}_{i+1, j}^v = \left(\tilde{\mathbf{U}}_{i+1, j}^v \right)^T [\mathbf{w}_{i+1, j} + \Delta \mathbf{w}_{i+1, j}] \quad (2.112)$$

または、等価的に次式のようになる。

$$\mathbf{e}_{i, j+1}^h = \left(\tilde{\mathbf{U}}_{i, j+1}^h \right)^T \Delta \mathbf{w}_{i, j+1} = \mathbf{d}_{i, j+1}^h - \left(\tilde{\mathbf{U}}_{i, j+1}^h \right)^T \mathbf{w}_{i, j+1} \quad (2.113)$$

$$\mathbf{e}_{i+1, j}^v = \left(\tilde{\mathbf{U}}_{i+1, j}^v \right)^T \Delta \mathbf{w}_{i+1, j} = \mathbf{d}_{i+1, j}^v - \left(\tilde{\mathbf{U}}_{i+1, j}^v \right)^T \mathbf{w}_{i+1, j} \quad (2.114)$$

ここで、

$$\mathbf{d}_{i, j+1}^h = [d(i, j+1), d(i-1, j+1), \dots, d(i-r_1+1, j+1), \\ d(i+1, j), d(i+1, j-1), \dots, d(i+1, j-r_2+1)]^T$$

$$\mathbf{d}_{i+1, j}^v = [d(i+1, j), d(i+1, j-1), \dots, d(i+1, j-s_1+1), \\ d(i, j+1), d(i-1, j+1), \dots, d(i-s_2+1, j+1)]^T$$

$$\tilde{\mathbf{U}}_{i, j+1}^h = [\mathbf{u}_{i, j+1}, \mathbf{u}_{i-1, j+1}, \dots, \mathbf{u}_{i-r_1+1, j+1}, \\ \mathbf{u}_{i+1, j}, \mathbf{u}_{i+1, j-1}, \dots, \mathbf{u}_{i+1, j-r_2+1}]$$

$$\tilde{\mathbf{U}}_{i+1, j}^v = [\mathbf{u}_{i+1, j}, \mathbf{u}_{i+1, j-1}, \dots, \mathbf{u}_{i+1, j-s_1+1}, \\ \mathbf{u}_{i, j+1}, \mathbf{u}_{i-1, j+1}, \dots, \mathbf{u}_{i-s_2+1, j+1}]$$

であり、 $\mathbf{e}_{i, j+1}^h$, $\mathbf{e}_{i+1, j}^v$ は誤差ベクトルと呼ばれている。式 (2.113), 式 (2.114) において、次数 $(r_1 + r_2)$, $(s_1 + s_2)$ が $m \times n$ より小さいとき、係数修正ベクトル $\Delta \mathbf{w}_{i, j+1}$, $\Delta \mathbf{w}_{i+1, j}$ の解は無数に存在する。しかし、 $\Delta \mathbf{w}_{i, j+1}$, $\Delta \mathbf{w}_{i+1, j}$ のノルムを最小にするような解をと

るとすると、係数修正ベクトルは唯一に求められ、次式のようになる。

$$\begin{aligned}\Delta w_{i,j+1} &= \left(\left(\tilde{U}_{i,j+1}^h \right)^T \right)^+ e_{i,j+1}^h \\ &= \tilde{U}_{i,j+1}^h \left[\left(\tilde{U}_{i,j+1}^h \right)^T \tilde{U}_{i,j+1}^h \right]^{-1} e_{i,j+1}^h\end{aligned}\quad (2.115)$$

$$\begin{aligned}\Delta w_{i+1,j} &= \left(\left(\tilde{U}_{i+1,j}^v \right)^T \right)^+ e_{i+1,j}^v \\ &= \tilde{U}_{i+1,j}^v \left[\left(\tilde{U}_{i+1,j}^v \right)^T \tilde{U}_{i+1,j}^v \right]^{-1} e_{i+1,j}^v\end{aligned}\quad (2.116)$$

ここで、 $\left(\left(\tilde{U}_{i,j+1}^h \right)^T \right)^+$ 、 $\left(\left(\tilde{U}_{i+1,j}^v \right)^T \right)^+$ は、それぞれ $\left(\tilde{U}_{i,j+1}^h \right)^T$ 、 $\left(\tilde{U}_{i+1,j}^v \right)^T$ の擬似逆マトリクスである。式 (2.115) および式 (2.116) は、逆マトリクスを含むため、 $(r_1 + r_2)$ 、 $(s_1 + s_2)$ が増えるにつれ計算量はそれぞれ $(r_1 + r_2)^3$ 、 $(s_1 + s_2)^3$ に比例して増加する。

2.7 数値例

2.7.1 2次元直交射影適応FIRフィルタの適用例

ここでは、2.3節で提案した手法を、非定常環境下での2次元システム同定に適応した例について考える。システム同定のブロック図を図2.5に示す。図2.5において、 $u(i, j)$ はフィルタへの入力信号、 $d(i, j)$ は所望の信号を表わしている。

入力信号として、平均0.00、分散89.30の白色雑音を、次式で表わされる低域通過フィルタに通して得られた有色雑音を用いている。

$$\begin{aligned}r(i, j) &= 0.2r(i, j-1) + 0.2r(i-1, j) + 0.3u(i, j-1) + 0.3u(i-1, j) \\ x(i, j) &= 0.8r(i, j) + 0.2u(i, j)\end{aligned}\quad (2.117)$$

ここで、この有色信号の平均は0.00、分散は15.63である。

対象となる未知のシステムとしては、次式

$$d(i, j) = (\text{rs}\mathbf{W}^0)(\text{rs}\mathbf{X}(i, j))^T \quad (2.118)$$

$$\mathbf{X}(i, j) = \begin{bmatrix} x(i, j) & x(i, j-1) & x(i, j-2) \\ x(i-1, j) & x(i-1, j-1) & x(i-1, j-2) \\ x(i-2, j) & x(i-2, j-1) & x(i-2, j-2) \end{bmatrix}$$

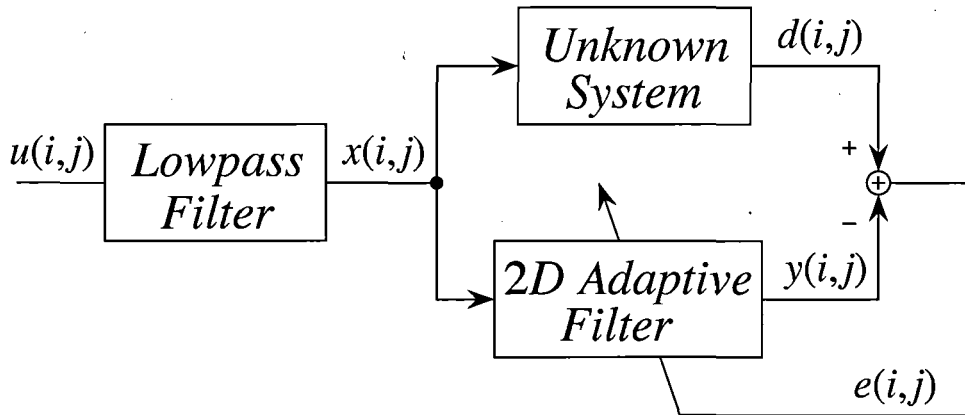


図 2.5 システム同定のブロック図

$$W^0 = \begin{bmatrix} 0.00 & -1.20 & 0.36 \\ 0.40 & -0.28 & -0.168 \\ -0.04 & -0.056 & 0.0256 \end{bmatrix}$$

を用いている。また、2次元データの大きさは、 $M = N = 100$ ，係数マトリクスの大きさは、 $m = n = 3$ とし、係数の初期値は、

$$W_0 = \begin{bmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 \end{bmatrix}$$

としている。以上の条件により、シミュレーションを行った。白色雑音を用いたときの収束状況を図 2.6 に示している。図 2.6 における評価関数 $err(i)$ は、次のように計算している。

$$err(i) = \frac{1}{2(100-i)} \sum_{j=i}^{M-1} \{e(i,j)^2 + e(j,i)^2\}, \quad (0 \leq i \leq 99) \quad (2.119)$$

ここで、各アルゴリズムで用いたステップサイズパラメータは以下のように設定した。

- LMS アルゴリズム : 5.00×10^{-3}
- 正規化 LMS アルゴリズム : 1.00
- 直交射影アルゴリズム : 1.00
- 直交射影アルゴリズム ($r = 3$) : 1.00

また、有色雑音を用いたときの収束状況を図 2.7 に示している。ここで、各アルゴリズムで

用いたステップサイズパラメータは以下のように設定した。

$$\begin{aligned} \text{LMS アルゴリズム} &: 2.00 \times 10^{-3} \\ \text{正規化 LMS アルゴリズム} &: 1.00 \\ \text{直交射影アルゴリズム} &: 1.00 \\ \text{直交射影アルゴリズム } (r = 3) &: 1.00 \end{aligned}$$

白色雑音を用いた場合、直交射影アルゴリズムのほうが若干速く収束しているが、各アルゴリズムとも $i = 10$ 以内に収束している。しかし、有色雑音を用いた場合、LMS アルゴリズムは $i = 23$ 、正規化 LMS アルゴリズムは $i = 14$ で収束しており、収束速度は著しく劣化する。しかし、直交射影アルゴリズムでは、その影響をあまり受けず、 $i = 10$ で収束している。また、次数 $r = 3$ の直交射影アルゴリズムでは、 $i = 8$ で収束しており、有色雑音の影響をほとんど受けていないことが分かる。

次に非定常環境下におけるシミュレーションを行う。システムへの入力 $u(i, j)$ の平均を水平数 i と等しくなるように変化させる。ただし、その他の設定値は変更しないものとする。このときの白色雑音を用いたときの収束状況を図 2.8 に示す。ここで、各アルゴリズムで用いたステップサイズパラメータは以下のように設定した。

$$\begin{aligned} \text{LMS アルゴリズム} &: 2.00 \times 10^{-5} \\ \text{正規化 LMS アルゴリズム} &: 1.00 \\ \text{直交射影アルゴリズム} &: 1.00 \\ \text{直交射影アルゴリズム } (r = 3) &: 1.00 \end{aligned}$$

また、有色雑音を用いたときの収束状況を図 2.9 に示す。ここで、各アルゴリズムで用いたステップサイズパラメータは以下のように設定した。

$$\begin{aligned} \text{LMS アルゴリズム} &: 1.00 \times 10^{-5} \\ \text{正規化 LMS アルゴリズム} &: 1.00 \\ \text{直交射影アルゴリズム} &: 1.00 \\ \text{直交射影アルゴリズム } (r = 3) &: 1.00 \end{aligned}$$

これらの結果より、提案手法は他の手法より早く収束していることがわかる。

これらの結果は、アルゴリズムの幾何学的考察から予想された通りであり、本手法の有効性が示された。

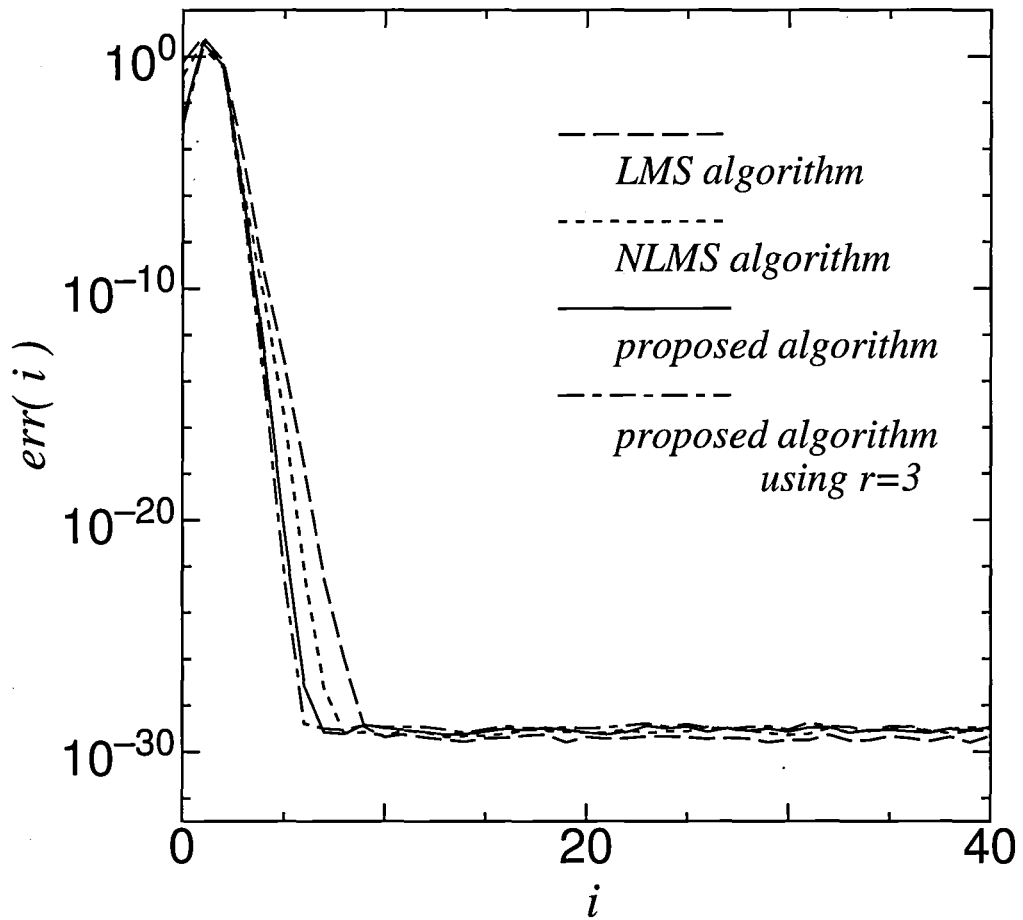


図 2.6 白色入力信号のときの各適応アルゴリズムの収束状況

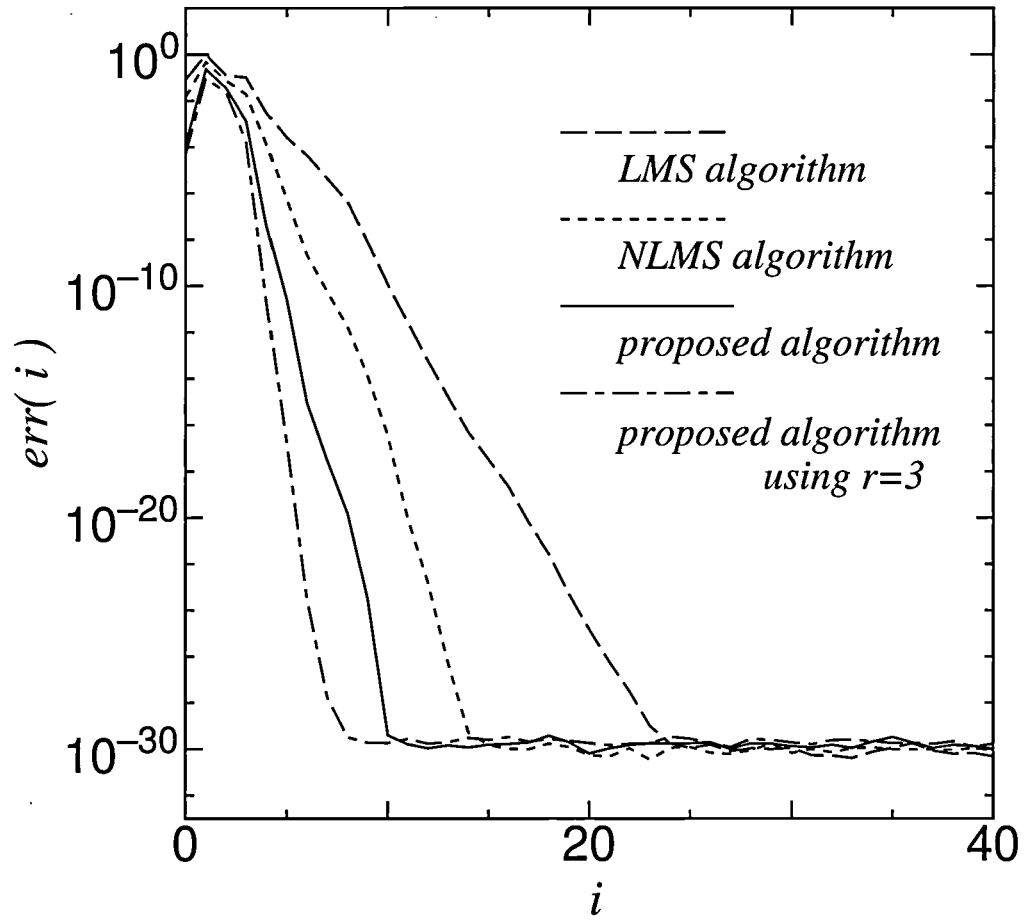


図 2.7 有色入力信号のときの各適応アルゴリズムの収束状況

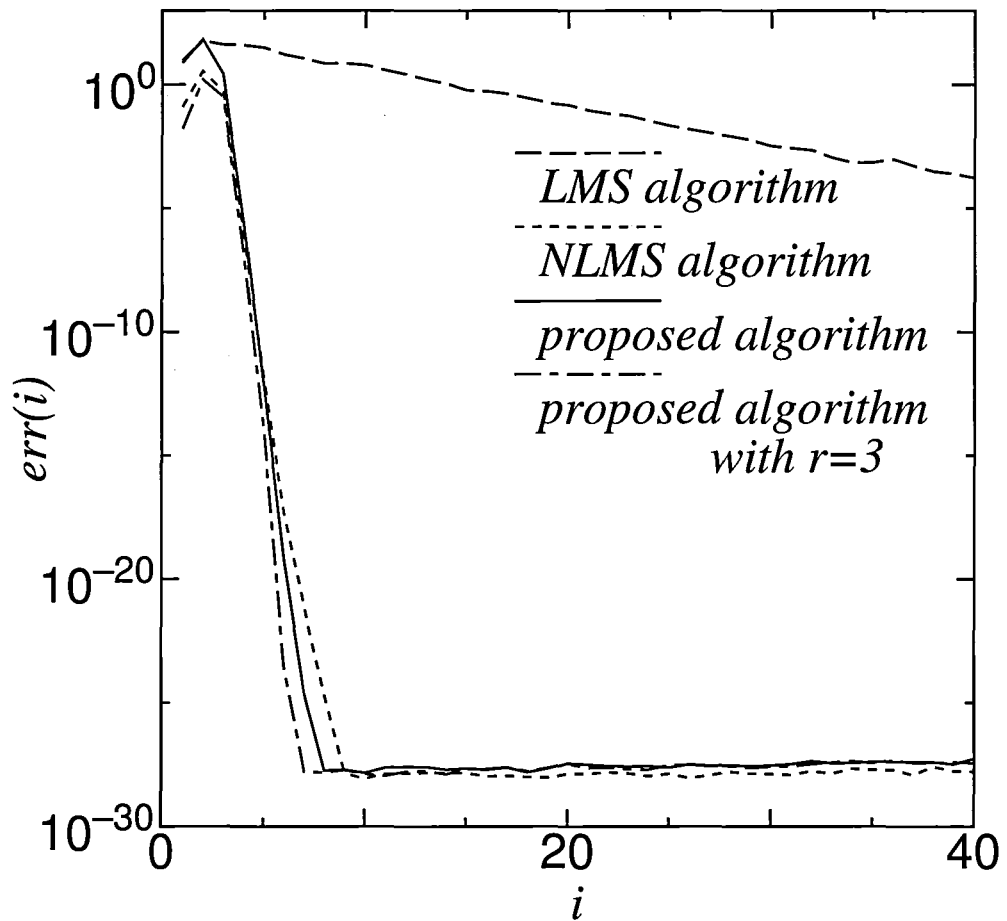


図 2.8 非定常環境下における白色入力信号のときの各適応アルゴリズムの収束状況

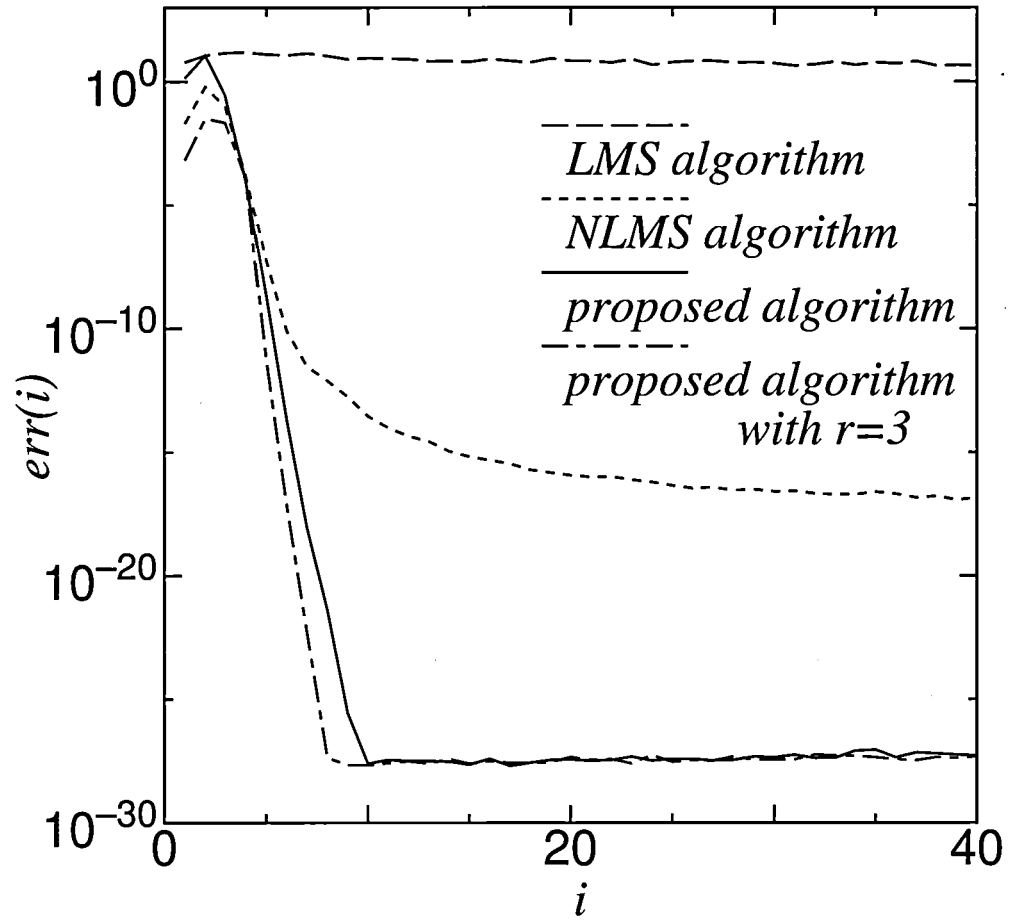


図 2.9 非定常環境下における有色入力信号のときの各適応アルゴリズムの収束状況

2.7.2 2方向処理による2次元直交射影適応 FIR フィルタの適用例

ここでは、2.5 節で提案した手法を、前節と同様の2次元システム同定に適応した例について考える。

白色雑音を用いたときの収束状況について図 2.10 に示している。ここで用いたステップサイズパラメータは $\mu_h = \mu_v = \mu$ とし、各アルゴリズムにおいて以下のように設定した。

LMS アルゴリズム	:	1.10×10^{-3}
正規化 LMS アルゴリズム	:	1.00
直交射影アルゴリズム	:	0.75
$r = 3, s = 3$ のときの直交射影アルゴリズム	:	0.85

また、有色雑音を用いたときの収束状況を図 2.11 に示している。ここで用いたステップサイズパラメータは、各アルゴリズムにおいて以下のように設定した。

LMS アルゴリズム	:	4.50×10^{-3}
正規化 LMS アルゴリズム	:	1.10
直交射影アルゴリズム	:	1.05
$r = 3, s = 3$ のときの直交射影アルゴリズム	:	1.00

白色雑音を用いた場合、LMS アルゴリズムおよび正規化 LMS アルゴリズムでは、ほぼ同様の収束速度を示しているが、直交射影アルゴリズムは、これらのアルゴリズムよりも速く収束しているのが分かる。また、次数を $r = 3, s = 3$ へあげると更に速く収束している。次に、有色雑音を用いた場合、各アルゴリズムともその影響を受けているが、直交射影アルゴリズムは、LMS アルゴリズム、正規化 LMS アルゴリズムと比べて、速く収束していることが分かる。また、次数を $r = 3, s = 3$ へあげると更に速く収束し、 $i = 100$ 以内で収束している。以上の結果から、2.5 節で提案した2次元直交射影アルゴリズムが、有色入力環境下での収束速度の点において有効であることが示された。

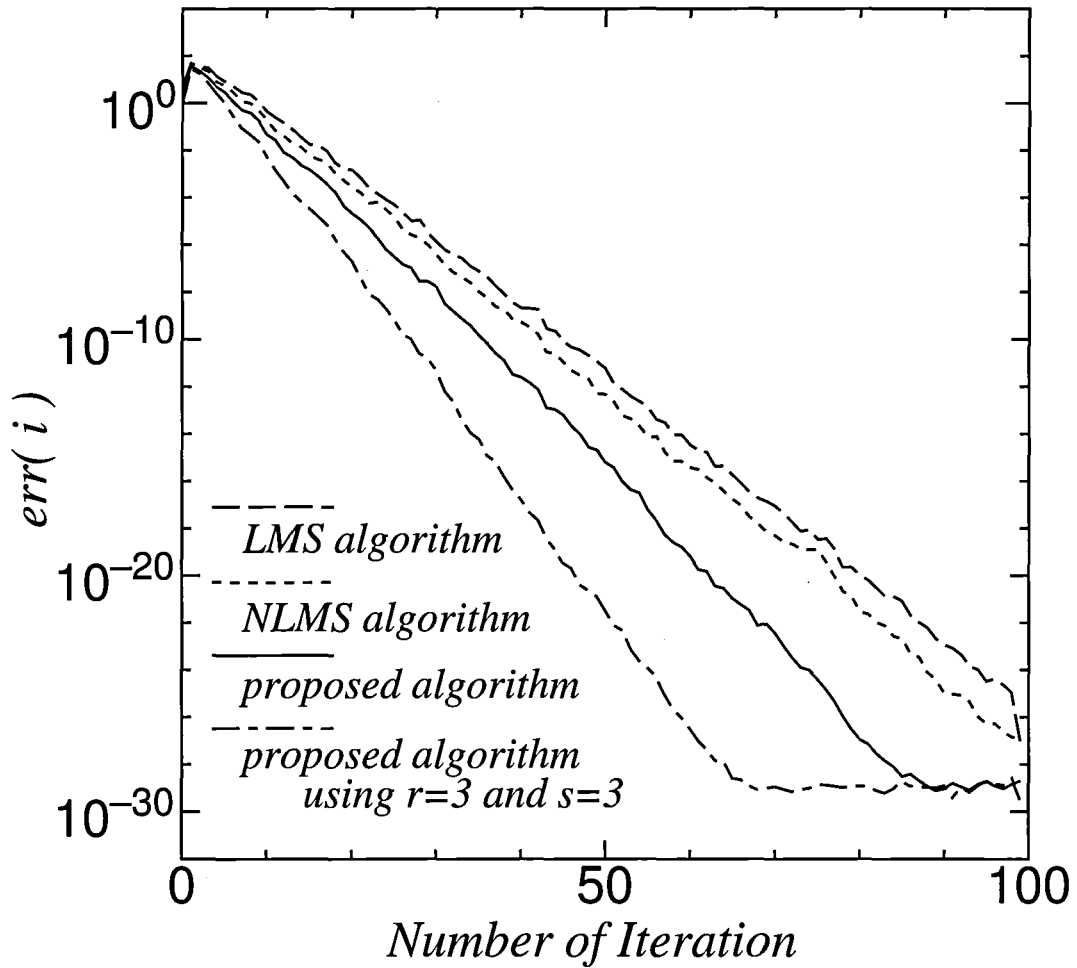


図 2.10 白色入力信号のときの各適応アルゴリズムの収束状況

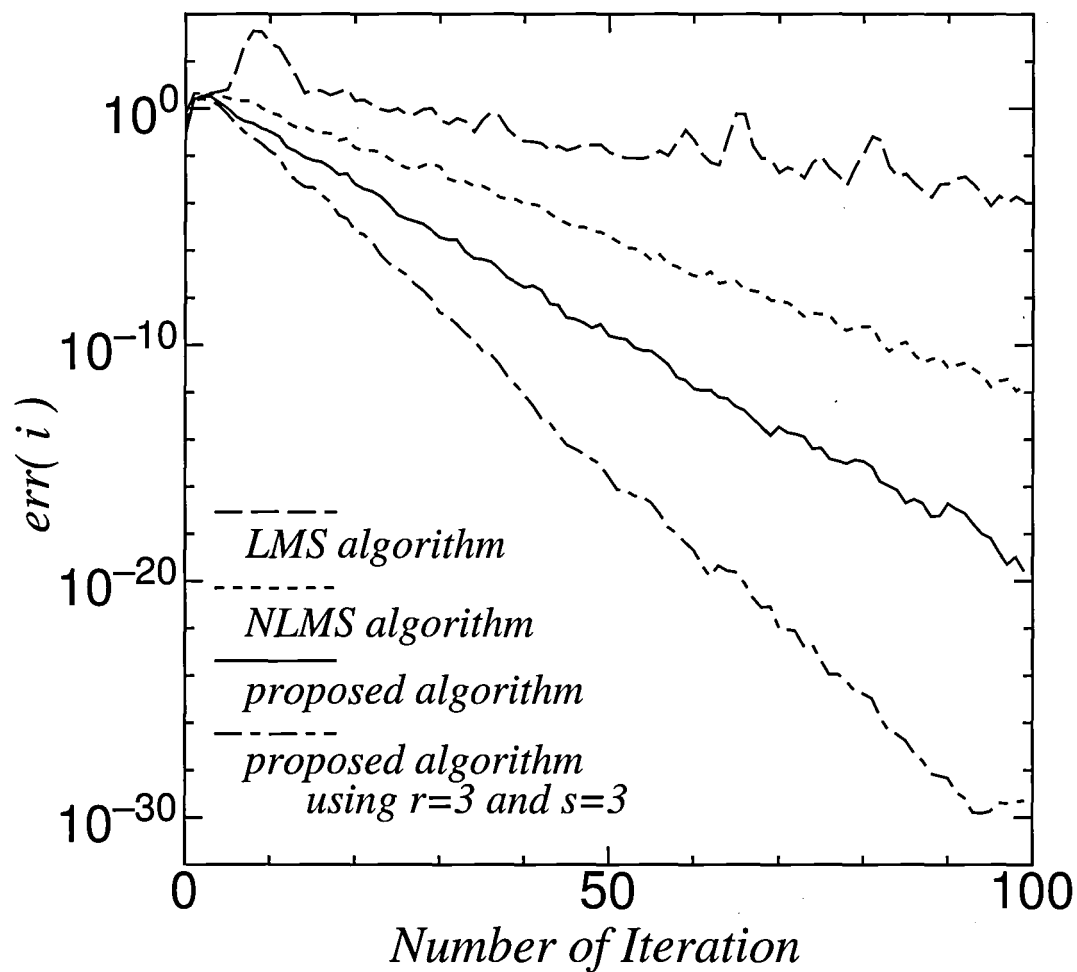


図 2.11 有色入力信号のときの各適応アルゴリズムの収束状況

2.7.3 2方向の入力を直交化させる2次元直交射影適応 FIR フィルタの適用例

また、2.6節で提案した手法で前節と同様のシミュレーションを行った。白色雑音を用いたときの収束状況を図 2.12 に示している。ここで用いたステップサイズパラメータは $\mu_h = \mu_v = \mu$ とし、各アルゴリズムにおいて以下のように設定した。

$$\text{Algorithm1}(r_1 = s_1 = 1, r_2 = s_2 = 1) : 0.85$$

$$\text{Algorithm2}(r_1 = s_1 = 1, r_2 = s_2 = 2) : 0.75$$

$$\text{Algorithm3}(r_1 = s_1 = 2, r_2 = s_2 = 1) : 0.75$$

$$\text{Algorithm4}(r_1 = s_1 = 2, r_2 = s_2 = 2) : 0.75$$

また、有色雑音を用いたときの収束状況を図 2.13 に示している。ここで用いたステップサイズパラメータは、各アルゴリズムにおいて以下のように設定した。

$$\text{Algorithm1}(r_1 = s_1 = 1, r_2 = s_2 = 1) : 1.05$$

$$\text{Algorithm2}(r_1 = s_1 = 1, r_2 = s_2 = 2) : 0.80$$

$$\text{Algorithm3}(r_1 = s_1 = 2, r_2 = s_2 = 1) : 1.00$$

$$\text{Algorithm4}(r_1 = s_1 = 2, r_2 = s_2 = 2) : 0.75$$

次数に着目すると、*Algorithm1* は次数 $r = s = 2$ の場合の直交射影アルゴリズムに相当し、*Algorithm2* と *Algorithm3* は次数 $r = s = 3$ の場合の直交射影アルゴリズムに相当する。白色雑音を用いた場合、*Algorithm1* は $i = 85$ で収束しており、次数 $r = s = 2$ の場合の直交射影アルゴリズムとほぼ同等の速度で収束している。このアルゴリズムにおいても、次数を上げることにより収束速度は速くなっている。また、有色雑音を用いた場合、各アルゴリズムともその影響を受けているが、次数を上げることによりその影響は小さくなっていることが分かる。以上により、水平、垂直の2方向の入力信号を直交化させても良好な結果が得られ、本手法の有効性が示された。

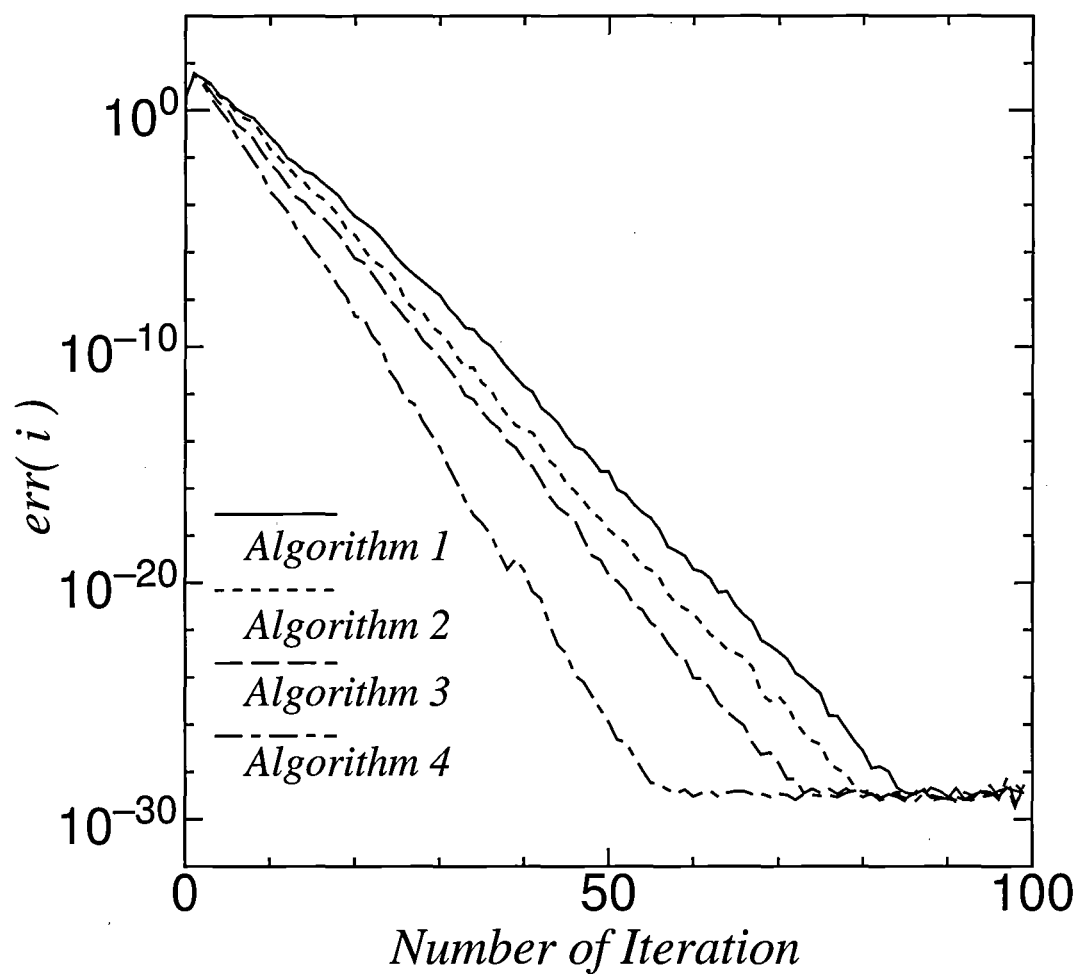


図 2.12 白色入力信号のときの各適応アルゴリズムの収束状況

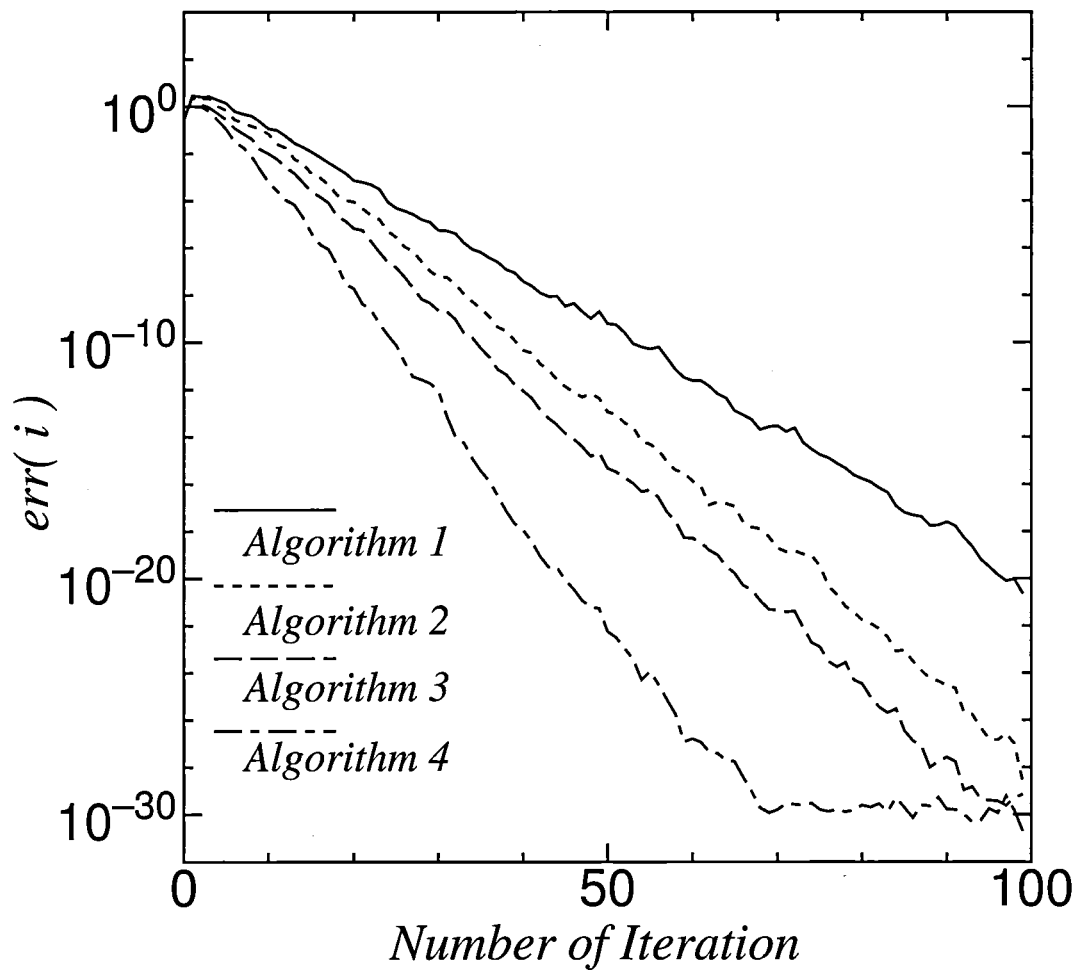


図 2.13 有色入力信号のときの各適応アルゴリズムの収束状況

2.8 結言

本章では、直交射影アルゴリズムを用いた2次元適応 FIR フィルタについて論じた。2.2 節では、従来法について説明し、図式解釈から問題点を指摘した。2.3 節では、係数マトリクスをシュミットの正規直交化法を用いて修正し、入力信号の性質に左右されにくい2次元直交射影アルゴリズムについて提案した。このアルゴリズムにより、計算量は増加するが、有色な入力信号において収束速度を速くすることができる。また、このアルゴリズムの収束性について証明し、最適解へ単調に収束することを示した。さらに、このアルゴリズムを一般化し、2次元正規化 LMS アルゴリズムとの関連性について述べた。2.5 節では、水平および垂直方向処理を考慮した2次元直交射影アルゴリズムについて提案した。画像などの2次元データを処理する場合、一般的には水平あるいは垂直の場合に対して信号の性質が異なる場合が多く、両方向から処理することにより、2次元データを正しく推定できる可能性が高いと考えられる。そして、このアルゴリズムを一般化させ、2次元正規化 LMS アルゴリズムとの関係を述べた。2.6 節では、水平、垂直方向の入力を直交化させるアルゴリズムを提案し、アルゴリズムの一般化について述べた。2.7 節では、コンピュータシミュレーションにより本手法のアルゴリズムの妥当性を検証した。本章で提案したアルゴリズムを、非定常環境下における2次元システム同定に適用し、従来法と比較することによって本手法の有効性について示した。

第3章

モデル予測機能を有するパフォーマンス駆動型 PID 制御系の設計

3.1 緒言

化学プロセスに代表されるプロセス産業界では、プラント自動制御および安定したプラント制御の手法として、取扱いの容易さや物理的意味の明確さなどの観点から PID 制御法が広く利用されている [37,60]。近年では、コントローラの高性能化やシステムの特性変動などの影響により、モデル予測制御法に代表されるような高度な制御技術が適用されている。そのひとつに一般化予測制御 (Generalized Predictive Control:GPC) [41] がプロセス制御の分野において広く用いられており、GPC 制御則に基づいて PID パラメータを調整する方法 [44] やセルフチューニング PID 制御系 [68] など数多く研究が行われている。しかし、設計者が設定を行う GPC 制御パラメータの調整により制御系の性能に大きく影響を及ぼすことが分かっており、その設定方法に関する研究 [45,71] が行われているが、試行錯誤的に安全なパラメータを求めているのが現状である。

一方、高品質維持と運用コストの低減の観点から制御性能評価に関する研究が盛んに行われている [81,90,91]。これは、プラントの停止やシステムへの信号付加によるシステム同定を極力行うことなく、システムの稼働データより制御性能の良さを評価する方法について提案されている。環境の変化や製品の変更などの原因でプラントの特性が時間と共に変化するため、常に所望の制御性能が得られない場合が起こる。そのときには、システム同定によりシステムパラメータの推定を行い、制御性能が十分に発揮されるように適切な PID パラ

メータに再調整する必要がある。システム同定によるパラメータ推定は計算量の増大やパラメータ推定の信頼性に問題があり、実際の現場では制御性能評価の観点で PID パラメータを調整する方法の確立が望まれている。この問題に対して、文献 [93] では、セルフチューニング制御の観点から、モデリング性能を評価し、その評価が劣化した場合にシステム同定と PID パラメータの調整を行う方法が考察されている。この方法によると制御性能よりもモデリング性能の評価に重点が置かれており、所望の制御性能を維持することを考えた場合、直感的ではなく、現場オペレータにもあまり馴染まないものとなっている。

そこで本章では、設計者が所望の制御性能が得られる設計パラメータの設計方法、および制御性能評価に基づいてシステムパラメータの推定を行うセルフチューニング GPC-PID 制御系の一設計法を提案する。まず、モデル予測型 PID 制御系による PID パラメータ調節法（以降、GPC-PID という。）について説明する。次に定常状態における制御性能の一つとして、制御誤差分散と制御入力の差分の分散（以降、制御差分入力分散という。）を取り上げ、代表的な PID パラメータ調整則との比較を通して、GPC-PID 制御則の優位性を示すとともに、制御性能評価の観点から GPC-PID 制御則における設計パラメータを適応させる方法について考察する。さらに、制御性能評価指数について定義し、所望の制御性能が得られない場合にプラントのシステムパラメータ同定を含め PID パラメータを調整する方法、いわゆるパフォーマンス駆動型 PID 制御系の設計方法を提案する。最後にシミュレーションを行い、本手法の有効性について検証する。

3.2 モデル予測型 PID 制御系

3.2.1 制御対象モデルの記述

プロセス制御系を対象とする場合、「一次遅れ+むだ時間」系として記述することが多い。ここでも、対象とする制御システムを次式のように与えられるものとする。

$$G(s) = \frac{K}{1+Ts} e^{-Ls} \quad (3.1)$$

ここで、プラントシステムパラメータ K , T , L はそれぞれプラントシステムのゲイン、時定数、むだ時間を表している。

次に、むだ時間を一次遅れ系で近似した次式で表される設計用モデルを考える。

$$G'(s) = \frac{K}{(1+Ts)(1+Ls)} \quad (3.2)$$

式 (3.2) をサンプリング間隔 $T_s[\text{sec}]$ で離散化した CARIMA (Controlled Auto-Regressive and Integrated Moving Average) モデルと呼ばれる離散時間伝達関数モデルを考える。

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \frac{\chi(k)}{\Delta} \quad (3.3)$$

ここで,

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} \quad (3.4)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} \quad (3.5)$$

であり, $u(k)$ は制御入力信号, $y(k)$ はシステム出力信号, $\chi(k)$ は平均 0, 分散 σ_χ^2 のガウス性白色雑音を表す。また, z^{-1} は時間遅延演算子で $z^{-1}u(k) = u(k-1)$ を, Δ は差分演算子で $\Delta := 1 - z^{-1}$ をそれぞれ意味する。

セルフチューニングコントローラでは, プラントへの制御差分入力 $\Delta u(k)$ とシステム出力 $y(k)$ よりシステムパラメータ同定器を通してシステムパラメータ K, T, L が得られ, 式 (3.3) を用いて PID パラメータが求められる。

3.2.2 PID コントローラ

次式で与えられる PID 制御則を考える。

$$\Delta u(k) = K_P \left(\frac{T_s}{T_I} + \Delta + \frac{T_D}{T_s} \Delta^2 \right) e(k) \quad (3.6)$$

ここで, PID コントローラのパラメータ K_P, T_I, T_D はそれぞれ比例ゲイン, 積分時間, 微分時間を表している。また, $e(k)$ は制御誤差信号で, ステップ状で与えられる目標値 $w(k)$ を用いて次式で表される。

$$e(k) = w(k) - y(k) \quad (3.7)$$

以下の考察を簡単にするため, 式 (3.6) を次式として書き換える。

$$L(z^{-1})y(k) + \Delta u(k) - L(z^{-1})w(k) = 0 \quad (3.8)$$

ただし,

$$L(z^{-1}) = K_P \left\{ \left(1 + \frac{T_s}{T_I} + \frac{T_D}{T_s} \right) - \left(1 + \frac{2T_D}{T_s} \right) z^{-1} + \frac{T_D}{T_s} z^{-2} \right\} \quad (3.9)$$

である。PID 制御系の制御性能は、式 (3.6) または式 (3.8) で与えられる PID パラメータ K_P, T_I, T_D に大きく影響する。ここでは、モデル予測制御のひとつである GPC 制御則により PID パラメータを調整する [44]。

3.2.3 GPC 則による PID パラメータの調整法

次の評価関数を最小化することを考える [41]。

$$J = E \left[\sum_{j=N_1}^{N_2} \{y(k+j) - w(k)\}^2 + \sum_{j=1}^{N_u} \lambda(j) \{\Delta u(k+j-1)\}^2 \right] \quad (3.10)$$

ここで、 $N_1 \sim N_2$ は予測区間、 N_u は制御区間、 $\lambda(j)$ は制御差分入力信号に対する重み係数、 $E[\cdot]$ は期待値を表している。ここでは、簡単のため $N_1 = 1$, $N_2 = N_u = N$, $\lambda(1) = \lambda(2) = \dots = \lambda$ とする。式 (3.10) の評価関数を最小にする制御則は次式のように得られる。

$$\sum_{j=1}^N \{p_j F_j(z^{-1})\} y(k) + \left\{ 1 + z^{-1} \sum_{j=1}^N p_j s_j \right\} \Delta u(k) - \sum_{j=1}^N p_j w(k) = 0 \quad (3.11)$$

ここで、 $F_j(z^{-1})$ と s_j は、次の Diophantine 方程式を解くことで得られる。

$$1 = \Delta A(z^{-1}) E_j(z^{-1}) + z^{-j} F_j(z^{-1}) \quad (3.12)$$

$$E_j(z^{-1}) B(z^{-1}) = R_j(z^{-1}) + z^{-j} s_j \quad (3.13)$$

ここで、

$$E_j(z^{-1}) = 1 + e_1 z^{-1} + \dots + e_{j-1} z^{-(j-1)} \quad (3.14)$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1} z^{-1} + f_{j,2} z^{-2} \quad (3.15)$$

$$R_j(z^{-1}) = r_0 + r_1 z^{-1} + \dots + r_{j-1} z^{-(j-1)} \quad (3.16)$$

である。また、 p_j は次式から求められる。

$$[p_1, p_2, \dots, p_N] = [1, 0, \dots, 0] (\mathbf{R}^T \mathbf{R} + \lambda \mathbf{I})^{-1} \mathbf{R}^T \quad (3.17)$$

ここで、行列 \mathbf{R} は式 (3.16) の係数からなる行列で、次式で表される。

$$\mathbf{R} = \begin{bmatrix} r_0 & & & \\ r_1 & r_0 & & \mathbf{0} \\ \vdots & & \ddots & \\ r_{N-1} & r_{N-2} & \cdots & r_0 \end{bmatrix} \quad (3.18)$$

いま、式 (3.11) 左辺第 2 項の $\Delta u(k)$ の係数多項式は z^{-1} の 1 次式であるが、定常状態を重視して定常ゲインに置き換えた次式を考える。

$$\nu = 1 + \sum_{j=1}^N p_j s_j \quad (3.19)$$

このとき、式 (3.11) に対して次式を得る。

$$\tilde{F}(z^{-1})y(k) + \Delta u(k) - \tilde{F}(1)w(k) = 0 \quad (3.20)$$

ただし、

$$\begin{aligned} \tilde{F}(z^{-1}) &= \frac{1}{\nu} \sum_{j=1}^N p_j F_j(z^{-1}) \\ &= \tilde{f}_0 + \tilde{f}_1 z^{-1} + \tilde{f}_2 z^{-2} \end{aligned} \quad (3.21)$$

である。式 (3.8) と式 (3.20) を比較することにより、近似的に GPC 制御則と PID 制御則が等価となる。

$$K_P = -(\tilde{f}_1 + 2\tilde{f}_2) \quad (3.22)$$

$$T_I = -\frac{\tilde{f}_1 + 2\tilde{f}_2}{\tilde{f}_0 + \tilde{f}_1 + \tilde{f}_2} \cdot T_s \quad (3.23)$$

$$T_D = -\frac{\tilde{f}_2}{\tilde{f}_1 + 2\tilde{f}_2} \cdot T_s \quad (3.24)$$

以上により、GPC 制御則に基づいて PID コントローラパラメータを調整することができる。

GPC 則に含まれる予測区間 N と重み係数 λ は、設計者が与える設計パラメータである。予測区間 N は、実用的には時定数 T とむだ時間 L の和に相当するステップ数あたりに設定するのが良いとされている [44]。重み係数 λ を設定する方法については、特性多項式の零点を安定にするように解析的に求める方法 [45]、ロバスト安定性の観点から調整する方法 [71] などが考察されているが、試行錯誤を完全に排除していない。次節にて、制御性能評価の観点から重み係数 λ を決定する方法について考察する。

3.3 パフォーマンス駆動型 PID 制御系の設計

3.3.1 定常状態における制御性能

化学プロセスや石油精製プロセスにおいては、過渡状態における即応性よりも定常状態における安定性を非常に重視している。また、昨今の環境問題や省エネルギー化による運用コスト削減についても同時に求められている。そのため、定常特性の評価として制御誤差 $e(k)$ と制御差分入力 $\Delta u(k)$ の両方を考慮し稼働している制御系の良し悪しを評価することは非常に有効な手段となる。

まず、式 (3.1) に対応した次の離散時間詳細モデルを考える。

$$\alpha(z^{-1})y(k) = z^{-(d+1)}\beta(z^{-1})u(k) + \xi(k)/\Delta \quad (3.25)$$

ここで、

$$\alpha(z^{-1}) = 1 + \alpha_1 z^{-1} \quad (3.26)$$

$$\beta(z^{-1}) = \beta_0 + \beta_1 z^{-1} \quad (3.27)$$

であり、 d は離散時間系におけるむだ時間を表し、 $\xi(k)$ は平均 0、分散 σ_ξ^2 のガウス性白色雑音とする。

式 (3.8) と式 (3.25) から、 $w(k) = 0$ とすると次式が得られる。

$$e(k) = -\frac{1}{T(z^{-1})}\xi(k) \quad (3.28)$$

$$\Delta u(k) = -\frac{L(z^{-1})}{T(z^{-1})}\xi(k) \quad (3.29)$$

ただし、 $T(z^{-1})$ は次式で定義している。

$$T(z^{-1}) = \Delta\alpha(z^{-1}) + z^{-(d+1)}\beta(z^{-1})L(z^{-1}) \quad (3.30)$$

このとき、制御誤差 $e(k)$ と制御差分入力 $\Delta u(k)$ の分散は、 H_2 ノルムを用いて次式で計算できる [93]。

$$\hat{\sigma}_e^2 = E[\{e(k)\}^2] = \left\| -\frac{1}{T(z^{-1})} \right\|_2^2 \cdot \sigma_\xi^2 \quad (3.31)$$

$$\hat{\sigma}_u^2 = E[\{\Delta u(k)\}^2] = \left\| -\frac{L(z^{-1})}{T(z^{-1})} \right\|_2^2 \cdot \sigma_\xi^2 \quad (3.32)$$

以上により、システムパラメータ、PID パラメータおよび σ_ξ^2 により、定常状態における制御誤差分散、制御差分入力分散を推定できる。

3.3.2 GPC-PID 制御則の制御性能

式 (3.10) から分かるように、制御パラメータ λ を変化させたとき、評価関数 J を最小にする制御誤差分散と制御差分入力分散の組合せが存在する。縦軸を定常状態での制御誤差分散 $\hat{\sigma}_e^2$ 、横軸を定常状態での制御差分入力分散 $\hat{\sigma}_u^2$ とすると、 λ をパラメータとした制御性能曲線が得られる。

GPC-PID 制御則の制御性能曲線を図 3.1 に示す。ここで、ゲイン $K = 1.0$ 、時定数 $T = 10.0$ 、むだ時間 $L = 5.0$ 、サンプリング時間 $T_s = 1.0[\text{sec}]$ 、白色雑音の分散 $\sigma_\xi^2 = 10^{-5}$ とした。また、GPC 制御則の設定パラメータである予測区間 N を $N = T + L$ とし、重み係数 λ を 0 から 1000 まで増加させ、それぞれに対応した PID コントローラパラメータを式 (3.22)~(3.24) を用いて求めた。そして、 λ に対応した制御誤差分散 $\hat{\sigma}_e^2$ 、制御差分入力分散 $\hat{\sigma}_u^2$ を式 (3.31) および式 (3.32) を用いてそれぞれ算出した。

図 3.1 からわかるように、制御誤差分散 $\hat{\sigma}_e^2$ と制御差分入力分散 $\hat{\sigma}_u^2$ の間には trade-off の関係があり、この曲線は trade-off 曲線 [90,91] と呼ばれている。ただし、制御誤差分散 $\hat{\sigma}_e^2$ を最小とする λ が存在し、さらに λ を減少させると $\hat{\sigma}_e^2$ は増大し発散してしまう。これは、コントローラとして PID 制御を用いたため、コントローラの構造を簡略化したことなどに起因する。すなわち、GPC-PID 制御による定常状態における制御性能および安定性は制御重み λ に大きく影響する。

比較のため、よく知られている代表的な PID 制御則でも上述と同様の条件のもとで $\hat{\sigma}_e^2$ および $\hat{\sigma}_u^2$ を求める。

ZN : ZN 法 [61]

$$K_P = 1.2 \frac{T}{KL}, \quad T_I = 2.0L, \quad T_D = 0.5L$$

CHR1 : CHR 法 [62] (目標値応答, オーバーシュートなし)

$$K_P = 0.6 \frac{T}{KL}, \quad T_I = T, \quad T_D = 0.5L$$

CHR2 : CHR 法 (目標値応答, オーバーシュート 20%)

$$K_P = 0.95 \frac{T}{KL}, \quad T_I = 1.36T, \quad T_D = 0.47L$$

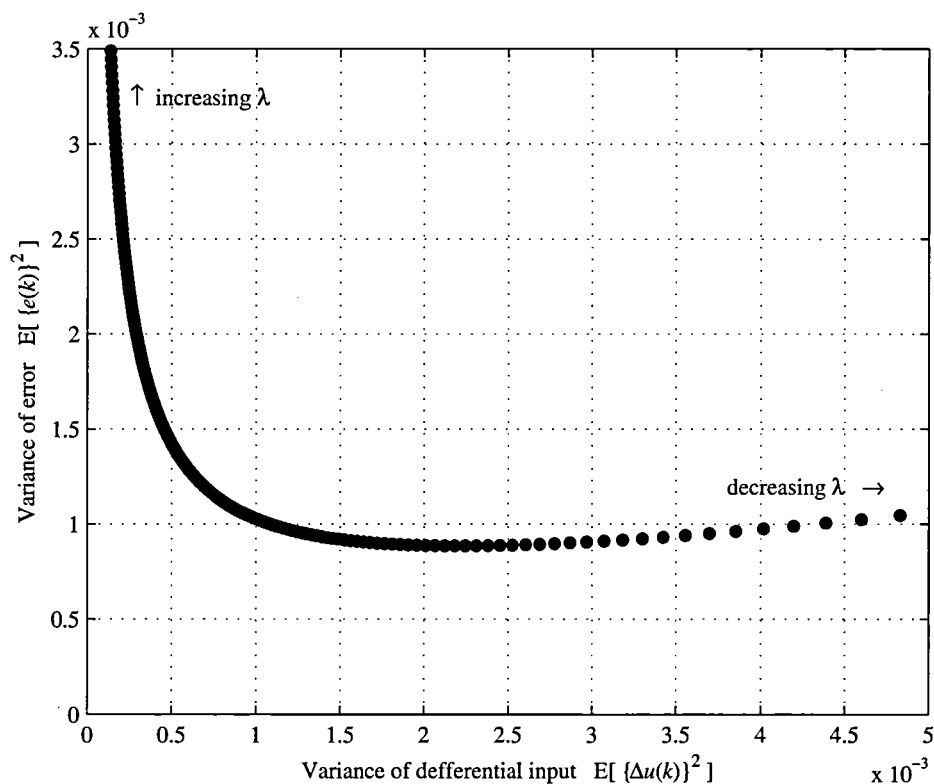


図 3.1 Trade-off curve of GPC-PID control law at $K = 1.0$, $T = 10.0$, $L = 5.0$

CHR3 : CHR 法 (外乱応答, オーバershootなし)

$$K_P = 0.95 \frac{T}{KL}, \quad T_I = 2.38L, \quad T_D = 0.42L$$

CHR4 : CHR 法 (外乱応答, オーバershoot 20%)

$$K_P = 1.2 \frac{T}{KL}, \quad T_I = 2.0L, \quad T_D = 0.417L$$

PM : むだ時間を除いた部分的モデルマッチング法 [63, 64]

部分的モデルマッチング法における参照モデルを次式のように記述する。

$$G_P(s) = \frac{e^{-Ls}}{1 + \sigma s + \alpha_2 \sigma^2 s^2 + \alpha_3 \sigma^3 s^3 + \dots} \quad (3.33)$$

ここで, σ は参照モデルの応答時間に相当するパラメータを表し, $\{\alpha_2, \alpha_3, \alpha_4, \dots\} = \{0.5, 0.15, 0.03, \dots\}$ を用いた。システムプラントと PID 制御器を系に含む閉ループ伝達関数を式 (3.33) と等しいとおくことで PID パラメータが得られる。

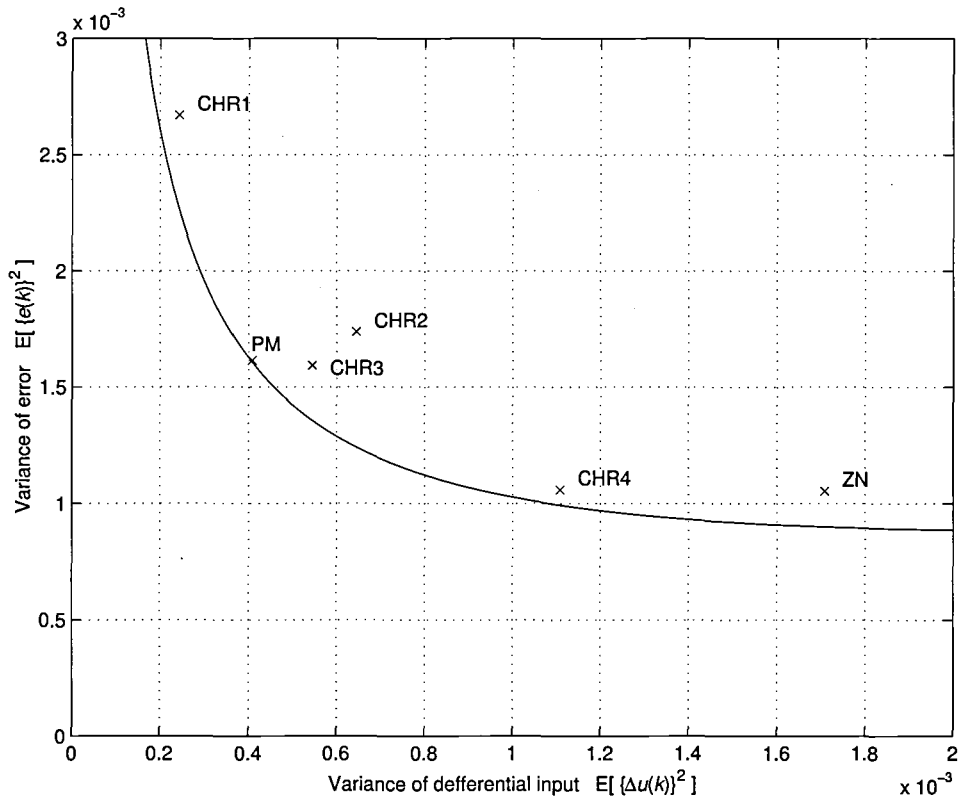


図 3.2 Comparison of GPC trade-off curve with several control laws at $K = 1.0$, $T = 10.0$, $L = 5.0$

これら代表的な制御則の定常状態における制御性能を GPC-PID 制御則の trade-off 曲線上にプロットした図を図 3.2 に示す。図 3.2 より、GPC 則による trade-off 曲線の上側もしくは曲線に沿うようにプロットされている。この結果から、GPC 則の重み係数を適切な値に設定することで、代表的な制御則をすべて包括することができる事が分かる。言い換えれば、 λ を適切な値へ調整することにより、プラント従事者の望む制御性能が得られることが分かる。

3.3.3 重み係数 λ の設定方法

定常状態での特徴を考慮すると、特にプロセス制御には追従性とロバスト性（安定性）の両面のバランスがとれた制御性能が得られることが望ましい。また、製品品質と運用コストなどの関係から、プラント従事者にとって好ましいバランスはその時の状況に応じて変化する。そこで、trade-off 曲線上の望ましい制御性能が得られるような λ を設定する方法につ

いて考察する。

ZN 法により調整した PID パラメータによるシステムの過渡応答波形は非常に振動的で、CHR1 法はオーバーシュートがほとんどない過渡応答波形が得られることが知られている。これらの特徴は GPC 制御則でも同様であり、 λ が小さいときの過渡応答波形は振動的であり、大きくなるにつれその振動は抑えられる。つまり、時間応答波形における最大行過ぎ量を指定することで、 λ を調整することができることを意味する。

そこで、閉ループ系におけるシステムのステップ応答出力波形 $y(t)$ の最大行過ぎ量 M_0 [%] に基づいて λ を調整することを考える。最大行過ぎ量 $M_0 = 8, 20, 40, 60, 80, 100$ [%] となる λ をそれぞれ過渡応答波形から求め、GPC-PID trade-off 曲線上にプロットした図を図 3.3 に示す。この M_0 を大きく設定すれば追従性に優れた λ の値が得られ、 M_0 を小さく設定すれば安定性に重点を置いた λ が得られることがわかる。これらの性質は、実用的なシステムパラメータ値 (K, T, L) において同様の傾向が見られることが実験的に示されている [96]。 M_0 から λ を設定するアルゴリズムを以下に示す。

[λ 設定アルゴリズム]

- 1° 後で述べるシステム同定結果から、ゲイン K 、時定数 T 、むだ時間 L を算出する。
- 2° 所望の最大行過ぎ量 M_0 、ならびに $x(0)$ と δ_x を設定する。
- 3° λ の初期値を、 $\lambda = 10^{x(0)}$ として設定する。
- 4° 1° で算出した K, T および L を真値と見なして、式 (3.22)~(3.24) により PID パラメータを求める。
- 5° 4° で算出した PID パラメータによりシミュレーションを行い、その応答波形から最大行過ぎ量 M'_0 を求める。
- 6° (i) $M_0 \cong M'_0$ ならば、その λ を採用する。(ii) $M_0 > M'_0$ ならば、 $\lambda = 10^{x(i)}$ として、4° へ戻る。ただし、 $x(i) \leftarrow x(i-1) - \delta_x$ 。(iii) $M_0 < M'_0$ ならば、 $\lambda = 10^{x(i)}$ として、4° へ戻る。ただし、 $x(i) \leftarrow x(i-1) + \delta_x$ 。

上記アルゴリズムによる λ の調整は試行錯誤的に求められているが、 M_0 と λ は 1 対 1 の関係にあるため、比較的簡単に求められる。本来は定常状態での制御指標を用いて PID パラメータを設計すべきであるが、定常状態での分散の評価と過渡状態での最大行過ぎ量の間には相関があり、 M_0 に基づいた PID パラメータ調整の方が制御系を設計する上で非常に容易である。また、 M_0 は PID パラメータと同様に制御従事者にとって馴染み深く取扱い

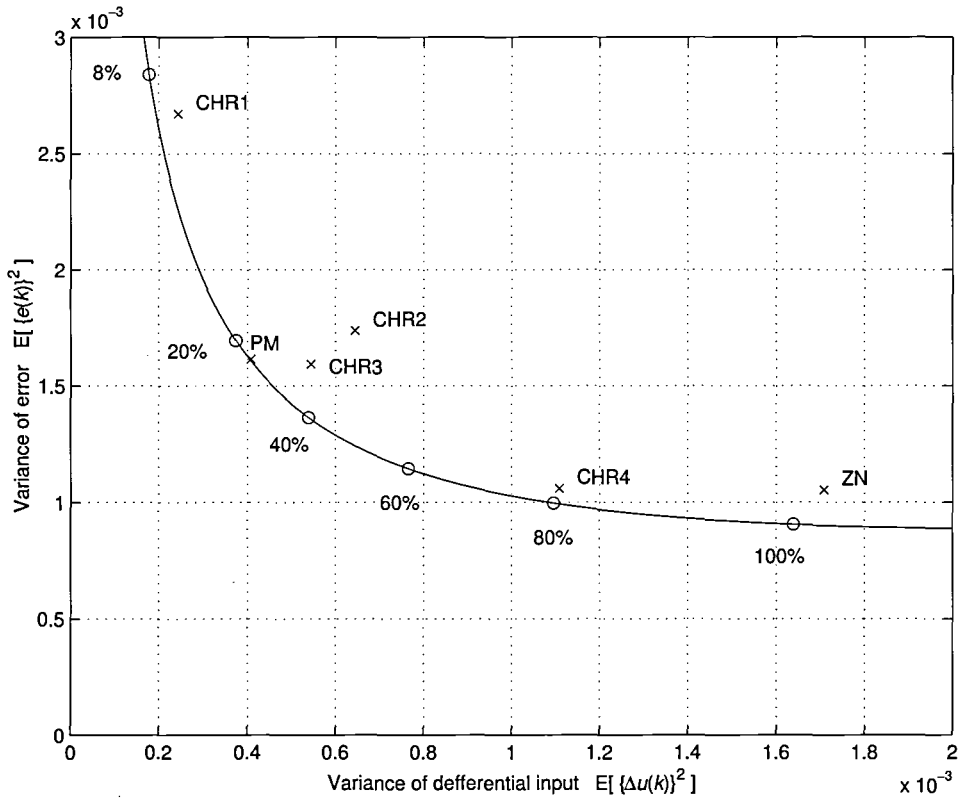


図 3.3 Relation between M_0 and λ at GPC trade-off curve at $K = 1.0, T = 10.0, L = 5.0$

やすいなどの利点から、制御性能の評価は定常状態での分散を用い、設計は過渡状態での最大行過ぎ量を用いることとする。

目標値 $w(k)$ に変更が生じると当然オーバーシュート $M_0[\%]$ の応答波形が結果として得られることとなる。特に化学プラントではオーバーシュートは 20[%] 以下で運用されており、 $M_0 > 20[\%]$ に相当する定常状態での制御性能に設定する場合は、過渡状態と定常状態とで λ を使い分けるかもしくは I-PD 制御器などのオーバーシュートを抑える構造を使用する必要がある。

3.3.4 性能評価指数

制御性能を評価する指数に関して数多く提案されている [81]。プロセスのシステムパラメータが既知またはよい精度で推定されている場合は、PID パラメータ値を用いることで制御誤差、制御差分入力の分散 $\hat{\sigma}_e^2, \hat{\sigma}_u^2$ が推定される。これらは式 (3.31), (3.32) から得られた計算値であり、実際の操業データから得られる制御誤差、制御差分入力の分散は、十分な

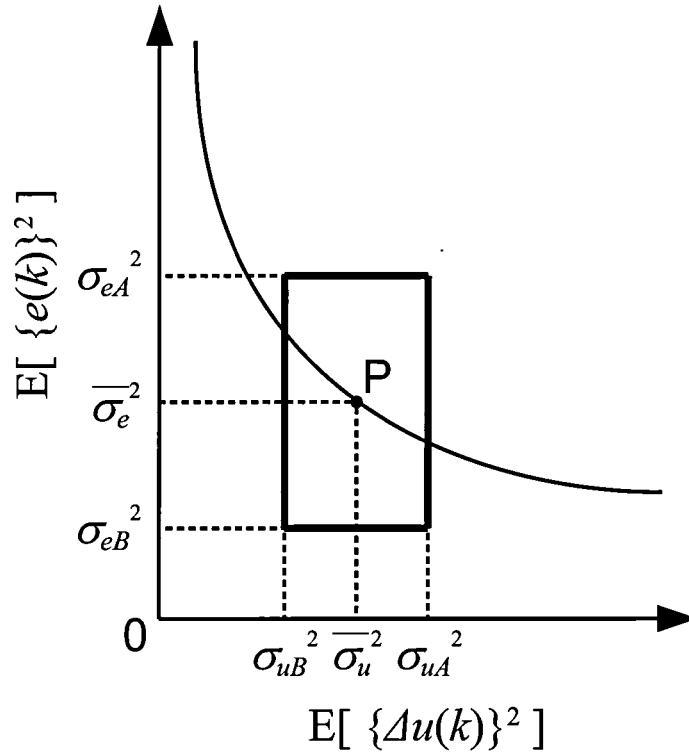


図 3.4 Region of control performance assessment index

データ量ではこれよりも若干大きくなることが知られている。

図 3.4 に概要図を示す。現在の時間 k における制御性能として P 点に位置しているとす。ある時刻 k で得られる制御誤差，制御差分入力の分散はそれぞれ以下のように求められる。

$$\sigma_e^2(k) = \frac{1}{M_e} \sum_{i=1}^{M_e} e^2(k-i+1) \quad (3.34)$$

$$\sigma_u^2(k) = \frac{1}{M_u} \sum_{i=1}^{M_u} \{\Delta u(k-i+1)\}^2 \quad (3.35)$$

ここで， M_e ， M_u は平均誤差分散区間，平均差分入力分散区間をそれぞれ表している。これらは，点 P の周りに分布し，ある一定の範囲内に収まる。つまり， $\sigma_{eB}^2 < \sigma_e^2(k) < \sigma_{eA}^2$ ， $\sigma_{uB}^2 < \sigma_u^2(k) < \sigma_{uA}^2$ を満足する領域が存在することを意味する。もし， $\sigma_e^2(k)$ ， $\sigma_u^2(k)$ がこの領域内に入らなかった場合は，プラントパラメータに変更があったことを意味するため，システム同定を行う必要がある。

この考えをもとに、以下の制御性能評価指数を定義する [81]。

$$\eta_e(k) = \frac{\hat{\sigma}_e^2}{\sigma_e^2(k)} \quad (3.36)$$

$$\eta_u(k) = \frac{\hat{\sigma}_u^2}{\sigma_u^2(k)} \quad (3.37)$$

ここで、 $\eta_e(k)$ 、 $\eta_u(k)$ は、1 に近いほど制御性能がよく、0 または無限大へ近づくにつれ制御性能が悪くなる。これらの値が設計者が設定する範囲内

$$\eta_{e,\min} < \eta_e(k) < \eta_{e,\max} \quad (3.38)$$

$$\eta_{u,\min} < \eta_u(k) < \eta_{u,\max} \quad (3.39)$$

にあれば、現在の制御性能は良いと判断する。サンプル長さ M_e 、 M_u が小さい場合は、 $\eta_e(k)$ 、 $\eta_u(k)$ は雑音の統計的性質によりばらつきが大きくなり制御性能の良し悪しを判断することが難しくなる。逆にサンプル長さを大きくすると $\eta_e(k)$ 、 $\eta_u(k)$ は 1 に近い値になるが、プラントシステムに変更があったとしても指数に表れるまで時間がかかる。サンプル長さとしきい値に関しては、適度な値に設定しておかなければならない [81]。

3.3.5 システム同定

式 (3.26) および式 (3.27) の係数 α_1 、 β_0 、 β_1 の推定値 $\hat{\alpha}_1(k)$ 、 $\hat{\beta}_0(k)$ 、 $\hat{\beta}_1(k)$ は、次の重み付き最小二乗法により求めることができる。

$$\hat{\theta}(k) = (\Phi^T V \Phi)^{-1} \Phi^T V y \quad (3.40)$$

ここで、

$$\hat{\theta}(k) = [\hat{\alpha}_1(k), \hat{\beta}_0(k), \hat{\beta}_1(k)]^T \quad (3.41)$$

$$\Phi = [\psi(k-1), \psi(k-2), \dots, \psi(k-M_s)]^T \quad (3.42)$$

$$\psi(k-i) = [-\Delta y(k-i), \Delta u(k-\hat{d}-i), \Delta u(k-\hat{d}-i-1)]^T \quad (i=1, 2, \dots, M_s) \quad (3.43)$$

$$y = [\Delta y(k), \Delta y(k-1), \dots, \Delta y(k-M_s+1)]^T \quad (3.44)$$

$$V = \text{diag}(v_1, v_2, \dots, v_{M_s}) \quad (3.45)$$

であり、 v_i ($i=1, 2, \dots, M_s$) は重み係数を、 M_s はシステム同定に用いられるデータ長さをそれぞれ表している。また、むだ時間 d の推定値 \hat{d} は、あらかじめ想定される範囲内

$(d_{\min} \leq \hat{d} \leq d_{\max})$ において推定誤差

$$\varepsilon(k) = \Delta y(k) - \hat{\theta}(k)^T \psi(k-1) \quad (3.46)$$

が最も小さくなる値として求める。

このとき、システムパラメータの推定値 \hat{K} , \hat{T} , \hat{L} は、次式により計算できる [97]。

$$\hat{K}(k) = \frac{\hat{\beta}_0(k) + \hat{\beta}_1(k)}{1 + \hat{\alpha}_1(k)} \quad (3.47)$$

$$\hat{T}(k) = -\frac{1}{\log_e(-\hat{\alpha}_1(k))} \cdot T_s \quad (3.48)$$

$$\hat{L}(k) = \left\{ \frac{\hat{\beta}_1(k)}{\hat{\beta}_0(k) + \hat{\beta}_1(k)} + \hat{d} \right\} \cdot T_s \quad (3.49)$$

3.3.6 パフォーマンス駆動型 PID コントローラ

これまでに考察した手続きをまとめ、以下のようにアルゴリズムを構築する。また、提案した手法による概略図を図 3.5 に示す。

[アルゴリズム]

- 1° 重み付き最小二乗法 (式 (3.40)) を用いてシステムパラメータを推定する。
- 2° 3.3.3 節に示す設定アルゴリズムにより λ を求める。
- 3° GPC-PID 制御則 (式 (3.22)~(3.24)) を用いて PID パラメータを調整する。
- 4° $k = k + 1$ とする。
- 5° 式 (3.31),(3.32) を用いて、制御誤差および制御差分入力の分散の推定値 $\hat{\sigma}_e^2$, $\hat{\sigma}_u^2$ を計算する。ここで、 σ_ε^2 は式 (3.46) の分散 σ_e^2 を代わりに用いる。
- 6° $\eta_e(k)$, $\eta_u(k)$ (式 (3.36),(3.37)) を求め、式 (3.38),(3.39) を満足する場合は 4° へ、そうでない場合は 1° へ戻る。

なお、システムのノイズの分散が大きくなった場合も、制御性能評価指数が指定の領域を超え、システム同定と PID パラメータの調整が一度行われるが、システム同定の結果、 σ_e が変更され、制御性能評価指数が指定の範囲内に入ることが期待できる。

3.4 数値例

本論文で提案したアルゴリズムの有効性を、2つのシミュレーションにより検証する。

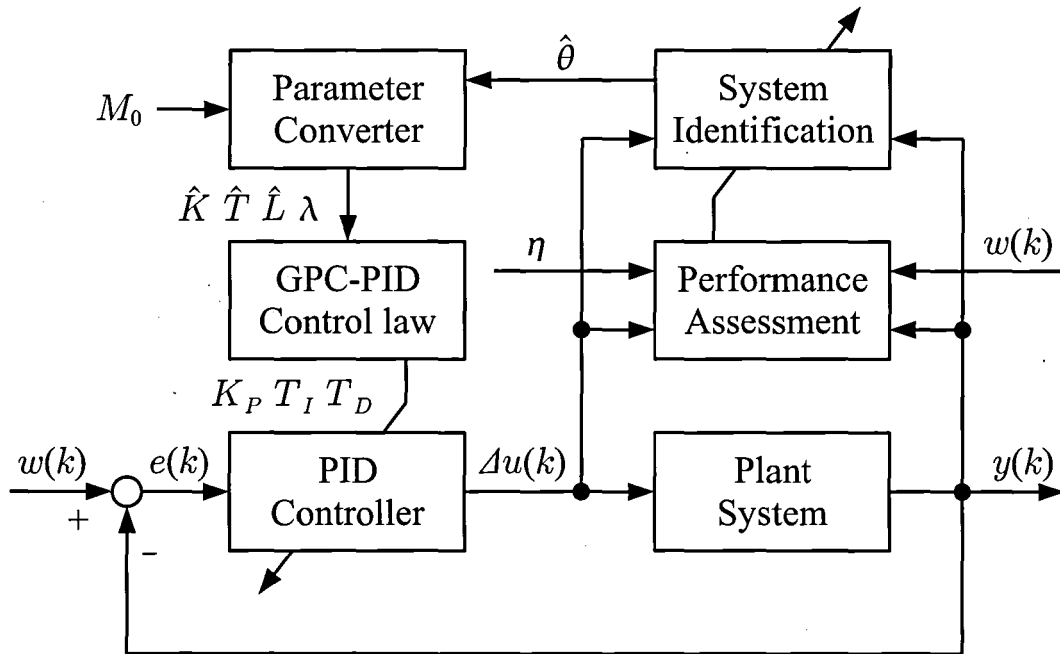


図 3.5 Schematic structure of the proposed PID controller

3.4.1 時変システム

制御対象が式 (3.1) で与えられるとする。ここで、 K, T, L は、図 3.6 のように時間とともに変化するものとし、サンプリング時間を $T_s = 1.0[\text{sec}]$ 、モデル化誤差 $\xi(k)$ は平均 0、分散 10^{-5} とする。また、目標値は $w(t) = 1.0, x(0) = 5.0, \delta_x = 0.005$ とする。

この制御対象に対し、最初の 2000[step] においてはシステムが既知であると仮定し、1~1000[step] では ZN 法、1001[step] からは提案した方法により制御を行なった。GPC パラメータである重み λ は、最大行過ぎ量が $M_0 = 40[\%]$ となる過渡応答波形が得られるように設定した。また、制御性能評価指標におけるパラメータを $M_e = M_u = 1000, \eta_{e,\min} = \eta_{u,\min} = 0.5, \eta_{e,\max} = \eta_{u,\max} = 1.5$ とし、システム同定におけるパラメータを $M_s = 500, V$ を単位行列とした。50[step] ごとに制御性能評価指数 $\eta_e(k), \eta_u(k)$ を求め制御性能の評価を行なった。評価指数が設定領域外となりシステム同定を行なった場合、その直後から得られる稼働データより評価指標を求める必要があり、十分なデータ量が得られるまで評価指標にばらつきが生じ、制御性能を正しく評価できないことがある。そこで、システム同定を行なってから評価指数が安定するまでの 500[step] の間は無駄なシステムパラ

メータ推定を行なわないこととした。

上記のパラメータによりシミュレーションを行なった。誤差 $e(k)$, 差分入力 $\Delta u(k)$ の時間推移を図 3.7 に, 制御性能評価指数 $\eta_e(k)$, $\eta_u(k)$ の時間推移を図 3.8 に, PID パラメータと λ の時間推移を図 3.9 に, システムパラメータ同定による推定値の時間推移を図 3.10 にそれぞれ示す。図 3.7 において 2000[step] までの 1000[step] 前後の波形を見ていくと, 提案した方法により制御誤差分散が若干増加しているが制御差分入力分散がかなり抑えられており, 設定した制御性能が得られていることがわかる。

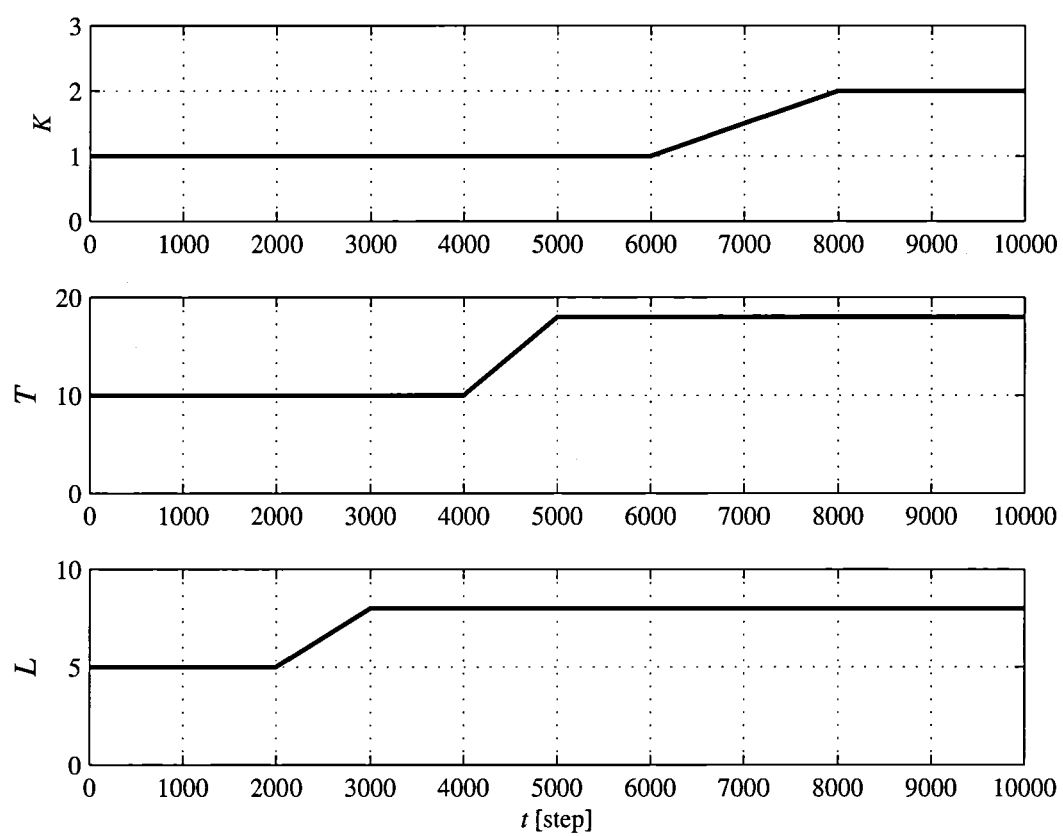
2000[step] 以降では, 制御性能評価指数がしきい値を超えた 3400[step], 4650[step], 5600[step], 6650[step], 7700[step] においてシステム同定を行なっている。詳しく見ていくと, システムのむだ時間 L が変化したことにより, 2500[step] あたりから $\eta_e(k)$, $\eta_u(k)$ とともに下がり始めて, 結果的に $\eta_e(k)$ がしきい値を超えたためシステム同定を行なっている。次にシステムの時定数 T が変化したことにより $\eta_e(k)$ のみが減少し, しきい値を超えていることがわかる。さらに, システムのゲイン K が変化したことにより, 今度は $\eta_u(k)$ のみが反応ししきい値を超えている。つまり, システムの変動により, $\eta_e(k)$ かもしれない $\eta_u(k)$ のどちらかしか反応しないときがあり, $\eta_e(k)$, $\eta_u(k)$ を用いて制御性能評価を行なうことは非常に有効であることがわかる。また, システム同定によりシステムパラメータを正しく推定できていないときがあり, 制御性能指数が 1 からややはずれていることで確認できる。

比較のため, セルフチューニング GPC-PID 制御系により同様のシミュレーションも行なった。 $\lambda = 30$ と固定し, 1000[step] 以降はステップ毎にシステム同定を行なった。誤差 $e(k)$, 差分入力 $\Delta u(k)$ の時間推移を図 3.11 に, 制御性能指数 $\eta_e(k)$, $\eta_u(k)$ の時間推移を図 3.12 にそれぞれ示す。システム変動前では, 比較的大きな λ を設定したため, 制御誤差の分散が大きくなり追従性が悪い。また, システムの変動に伴い, λ が適切に調整されないため, ゲイン K が大きくなるにつれ制御系が不安定になる。これにより, λ の調整手法の有効性が確かめられた。

一方, 文献 [93] のモデリング性能評価に基づいたパフォーマンス・アダプティブ PID 制御による制御結果との比較について考察する。ただし, 文献 [93] での設計パラメータを $\gamma = 3.0$ とした。先と同様に, 誤差 $e(k)$, 差分入力 $\Delta u(k)$ の時間推移を図 3.13 に, 制御性能指数 $\eta_e(k)$, $\eta_u(k)$ の時間推移を図 3.14 にそれぞれ示す。モデリング性能評価に基づいて基づいて調整する場合, 本論文での提案手法に比べ, システム同定の回数が多くなっている。制御性能においても, 提案手法の方が優れている。これは, 提案手法が直接, 制御性能

を評価して PID パラメータを調整していることによるものと考えられる。

計算コストや制御系への負荷を考慮すると、PID パラメータを逐次調整することはできるだけ避けたい。そのような観点からみれば、本手法ではシステム同定回数を極力抑えているため、非常に有効であることがいえる。

図 3.6 Trajectories of system parameters (K, T, L)

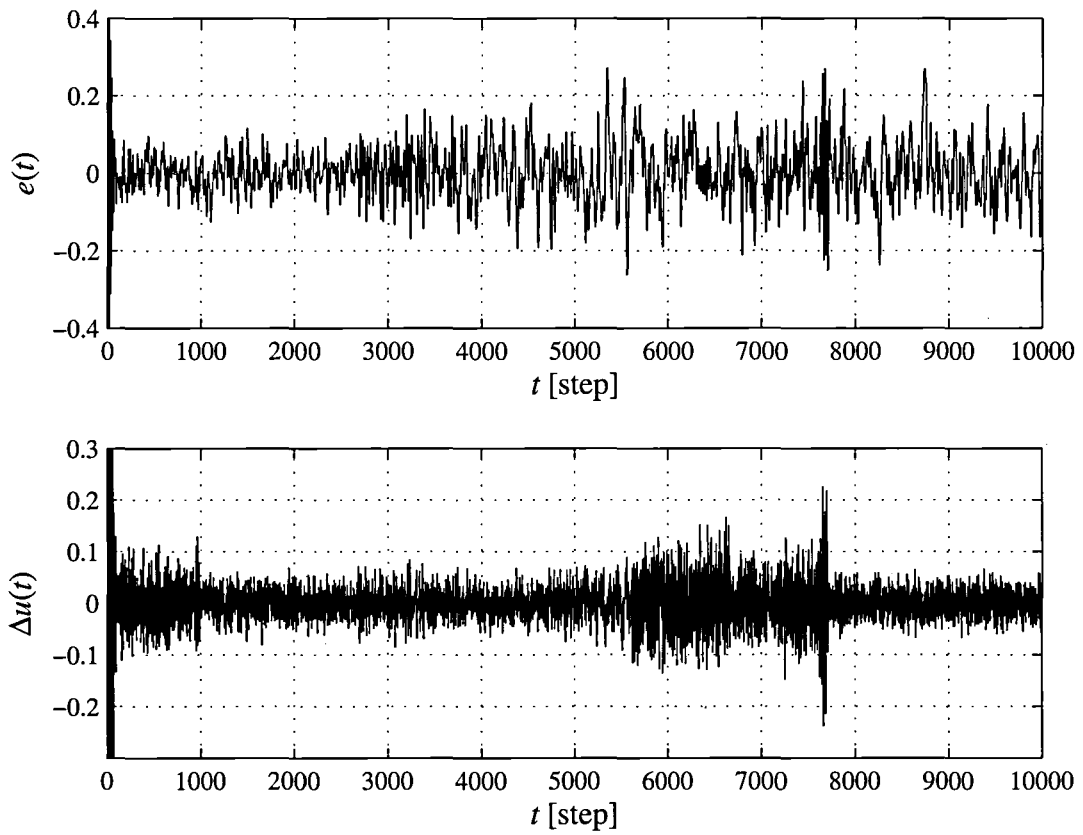


图 3.7 Control result using the proposed control scheme

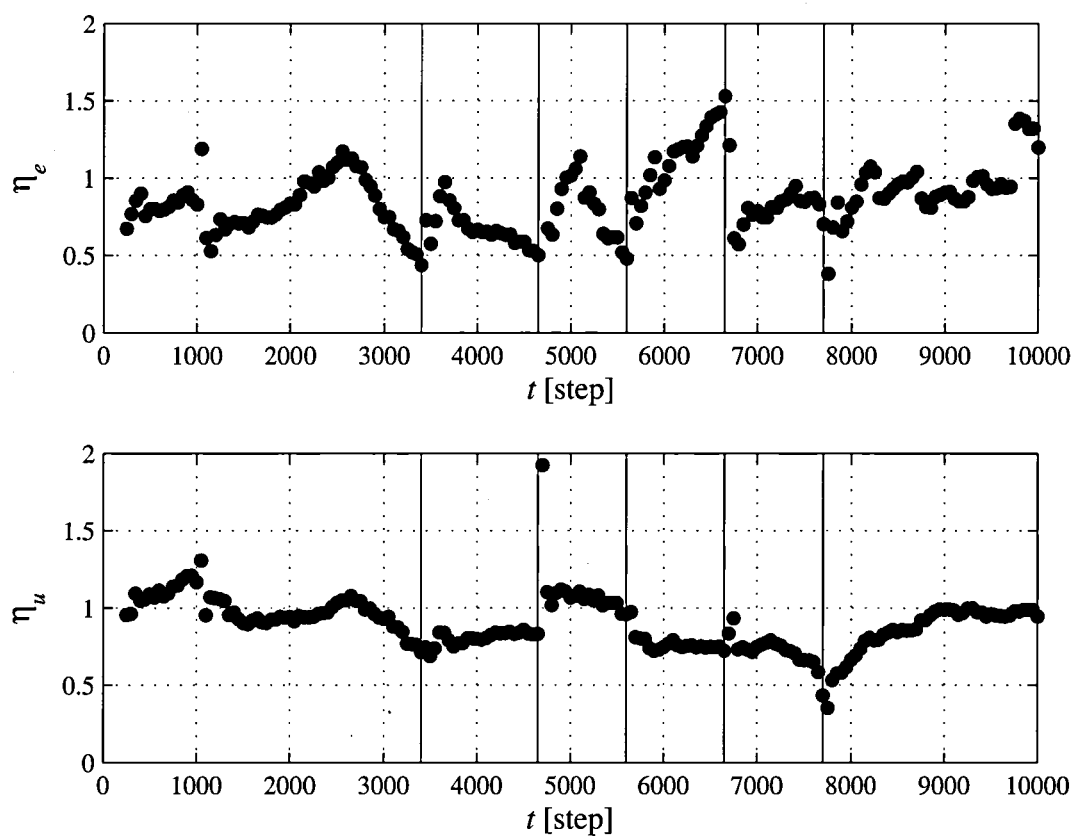


図 3.8 Trajectories of control performance assessment indices $\eta_e(k)$ (top) and $\eta_u(k)$ (bottom)

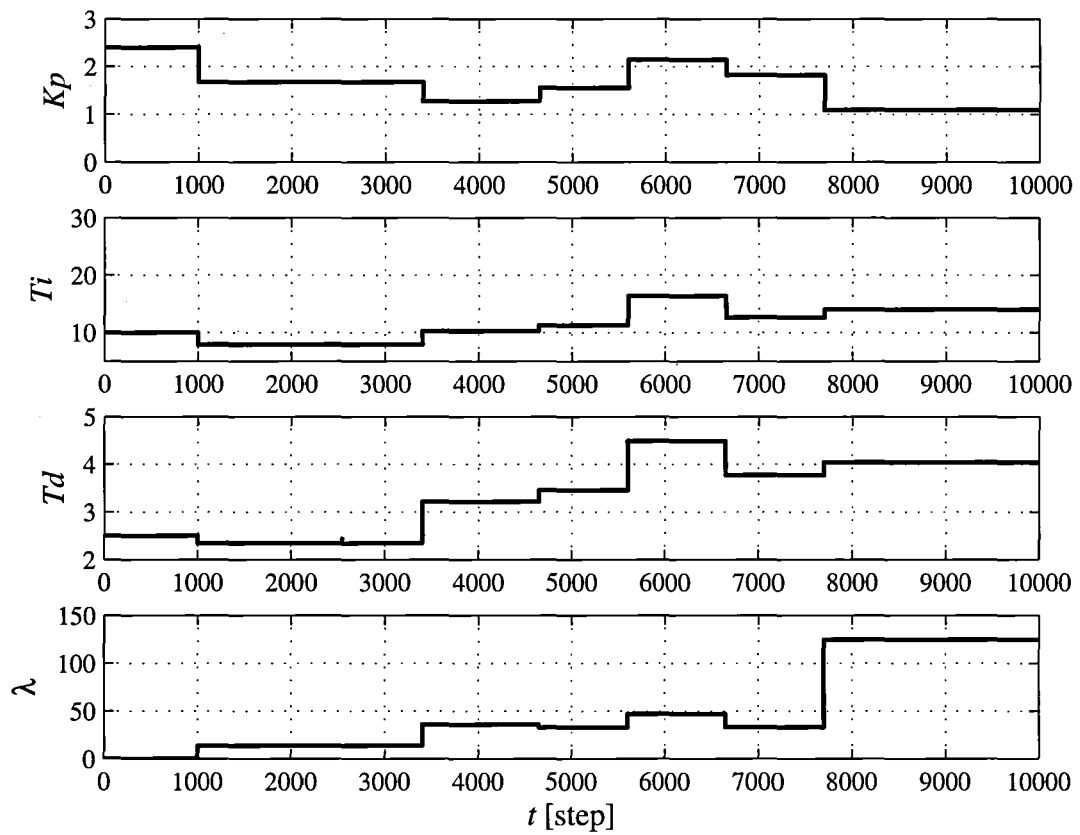


图 3.9 Trajectories of PID parameters (K_p , T_i , T_d) and GPC Weighting factor (λ)

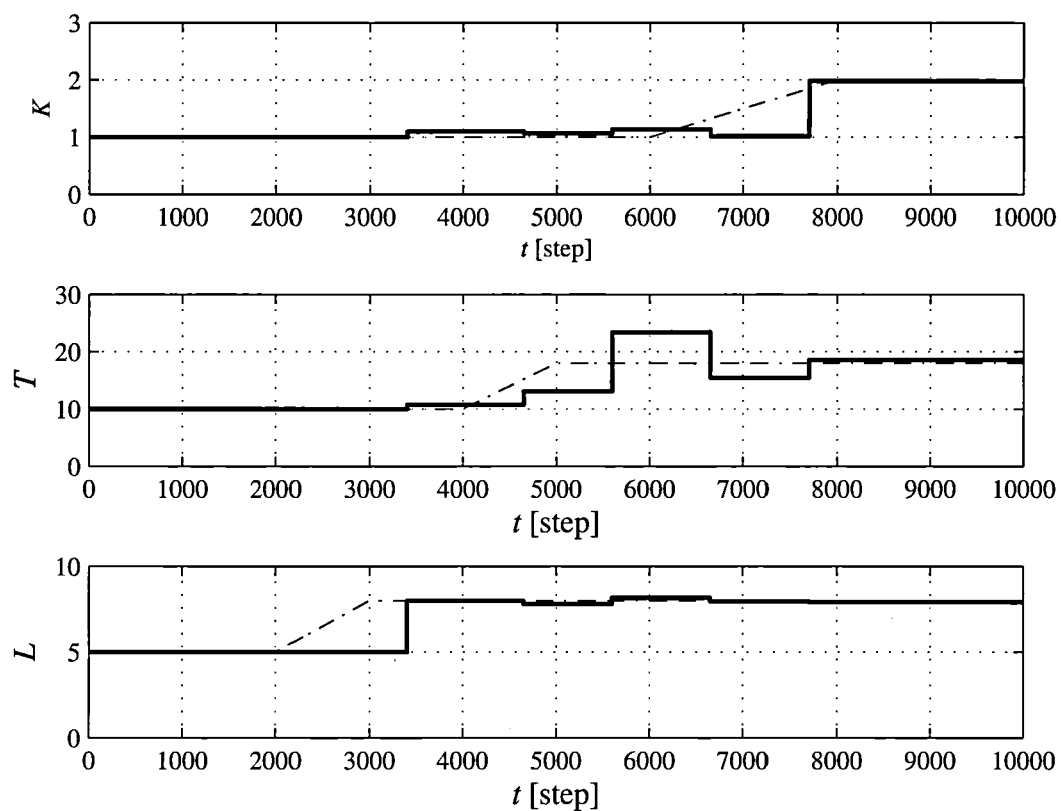


図 3.10 Trajectories of system parameter estimates ($\hat{K}(k)$, $\hat{T}(k)$, $\hat{L}(k)$)

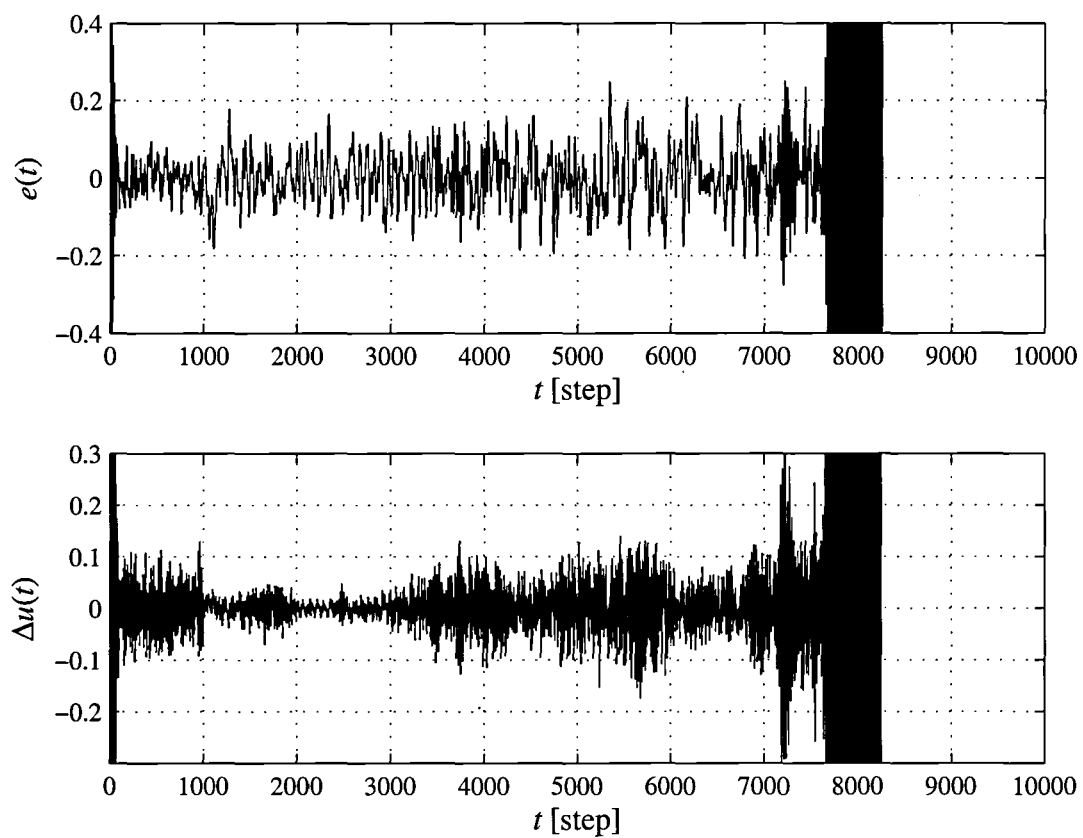


图 3.11 控制结果使用自调 GPC-PID 控制器

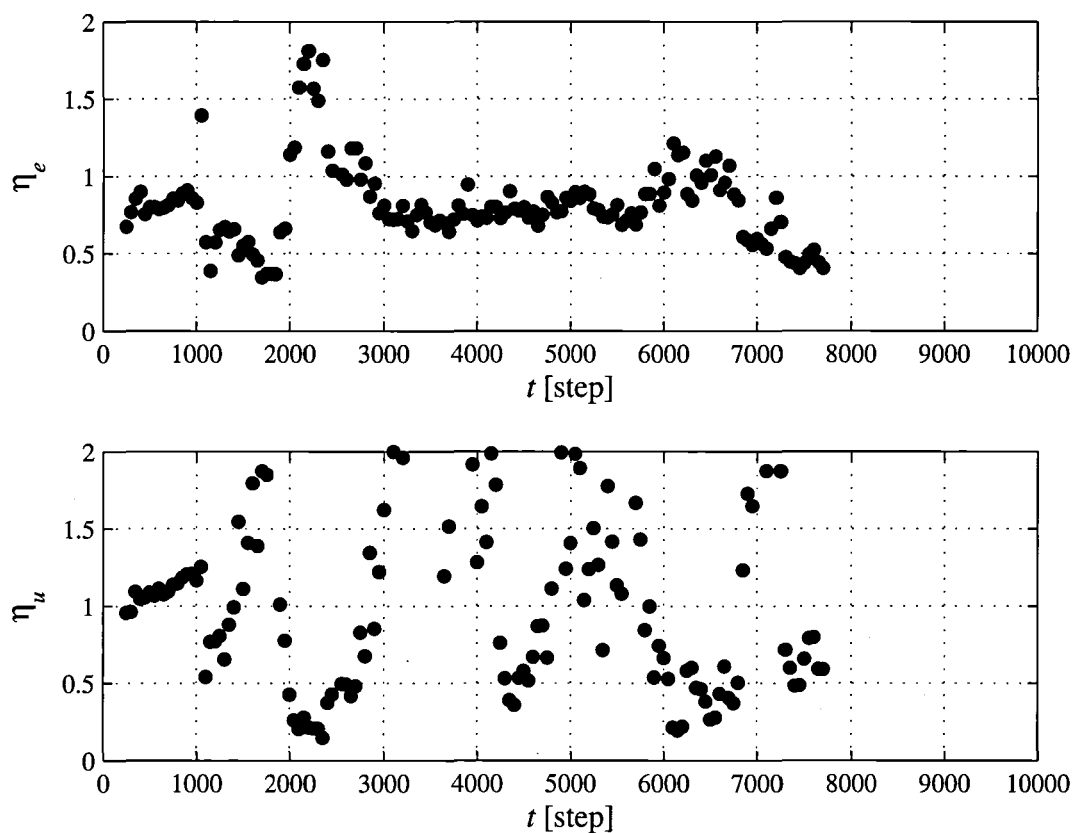


図 3.12 Trajectories of control performance assessment indices η_e (top) and η_u (bottom) using the self-tuning GPC-PID controller

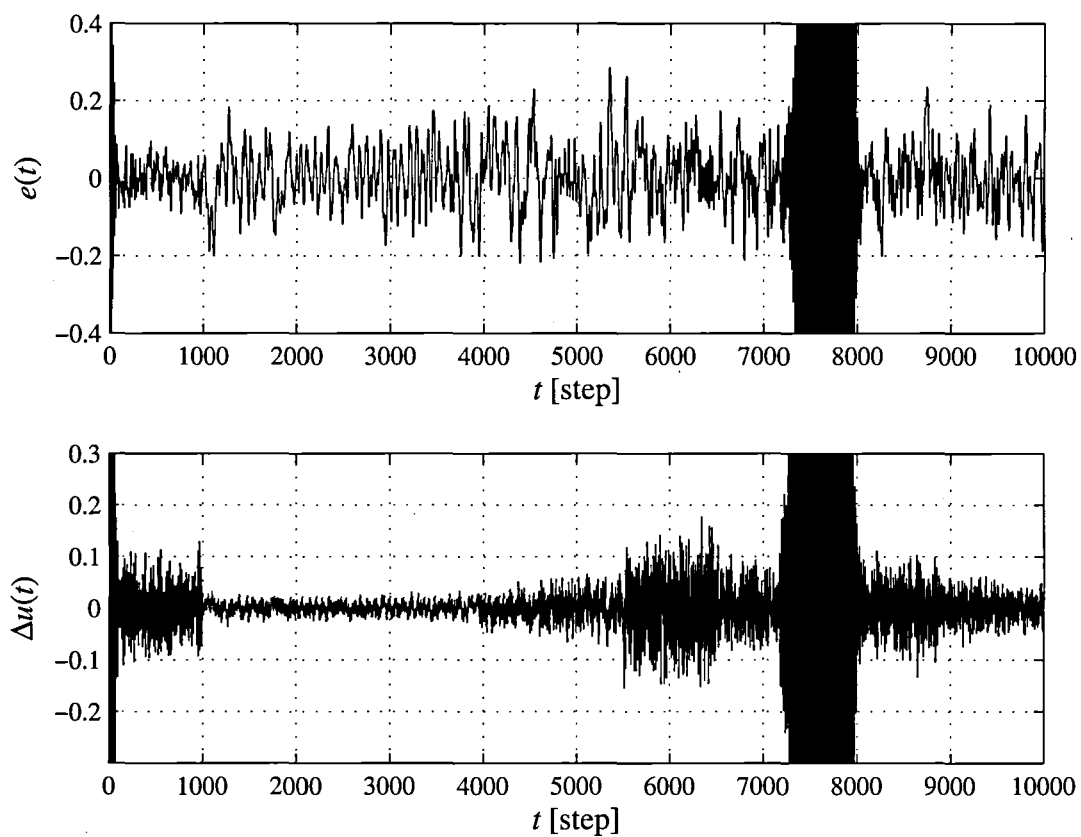


图 3.13 控制结果使用性能自适应 PID 控制器

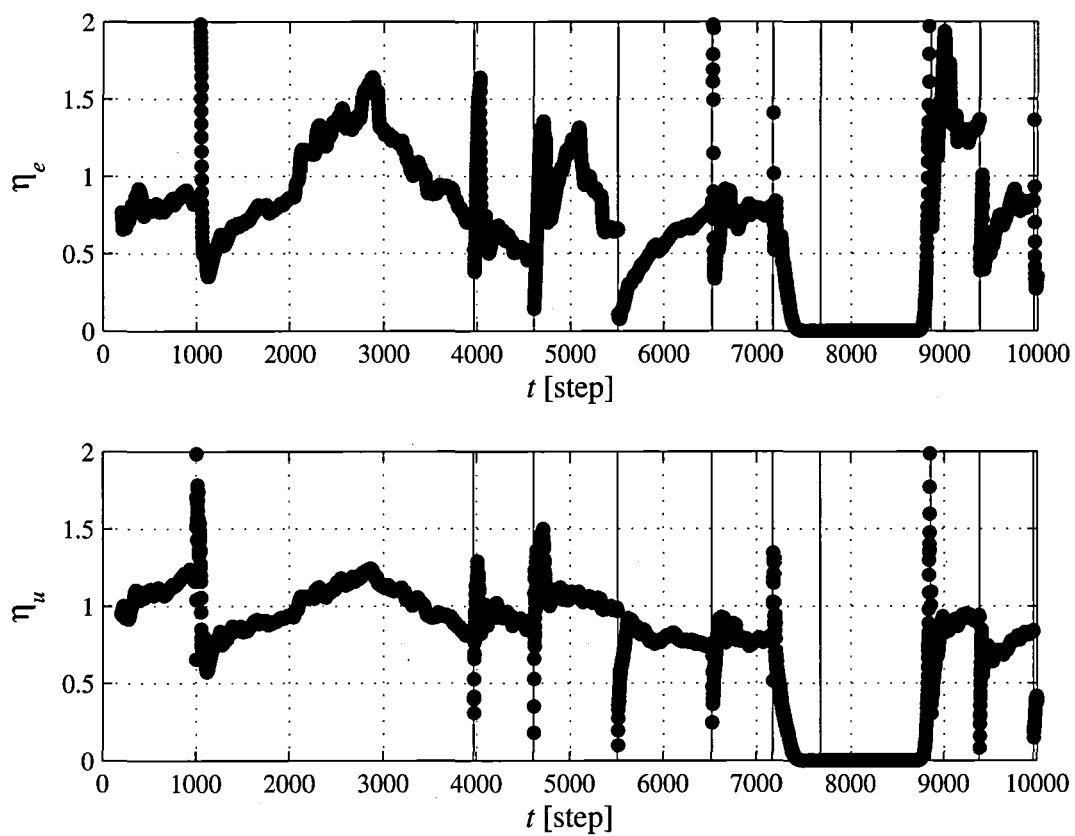


図 3.14 Trajectories of control performance assessment indices η_e (top) and η_u (bottom) using performance-adaptive PID controller

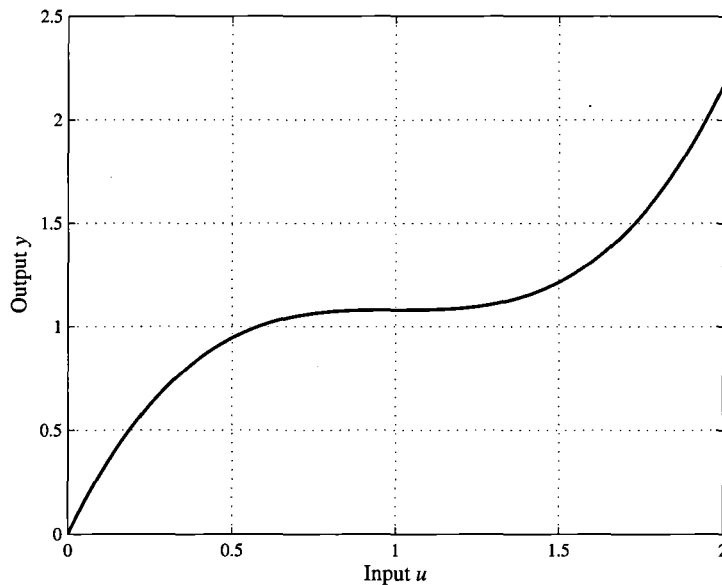


図 3.15 Static property of Hammerstein model

3.4.2 非線形システム

次式で表わされる Hammerstein model [98] を考える。

$$y(k) = 0.6y(k-1) - 0.1y(k-2) + 1.2x(k-d-1) - 0.1x(k-d-2) + \frac{\xi(t)}{\Delta} \quad (3.50)$$

$$x(k) = 1.5u(k) - 1.5u^2(k) + 0.5u^3(k) \quad (3.51)$$

このモデルの静特性を図 3.15 に示す。 $u = 1$ の周りで非線形性を有していることがわかる。目標値 $w(k)$ は次のように設定した。

$$w(k) = \begin{cases} 1 & (0 \leq k < 1000) \\ 0.7 & (1000 \leq k < 2000) \\ 1.5 & (2000 \leq k < 3000) \\ 2.0 & (3000 \leq k < 4000) \end{cases} \quad (3.52)$$

ユーザ設定パラメータは $M_0 = 10[\%]$, $M_e = M_u = 300$, $M_s = 200$ とし、残りは前例と同様に設定した。ただし、目標値が変更した直後は制御性能を正しく評価するためのデータが少ないため、システム同定を行わない無反応区間 300[step] を設けている。

提案手法による結果を図 3.16 に、制御性能評価指数の時間推移を図 3.17 にそれぞれ示す。図 3.17 より、システム同定は 300[step], 1300[step] と 3300[step] に行われており、 $w(k) = 1.5$ のときには行なわれていないことがわかる。これは、 $w(k) = 1.5$ のときのシステム特性が $w(k) = 0.7$ のときとほぼ同等であるためであり、制御性能評価指数も設定範囲内に収まっている。

比較のためセルフチューニング GPC-PID 制御系により同様のシミュレーションを行った。その結果を図 3.18 に示す。ここで、 $\lambda = 30$ と固定した。 λ が適切に調整されないため $w(k) = 2.0$ で発散しているのがわかる。本数値例によっても、提案手法の有効性が検証された。

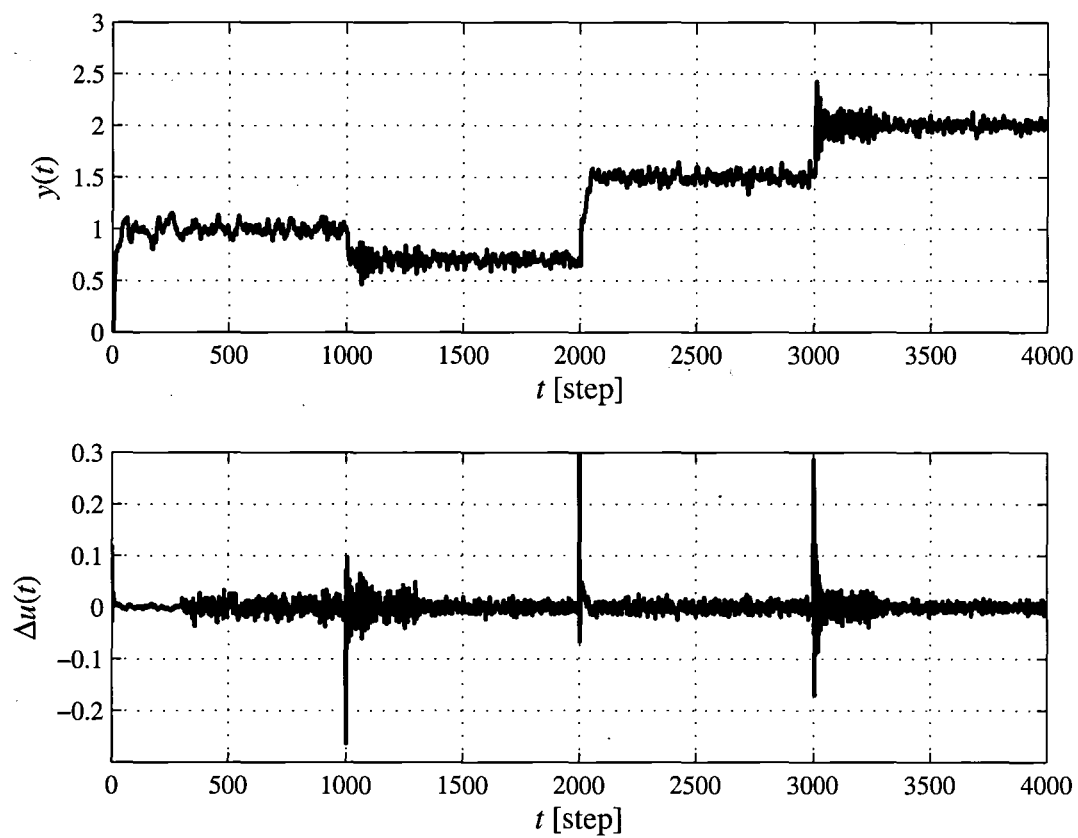


图 3.16 Control results using the proposed controller

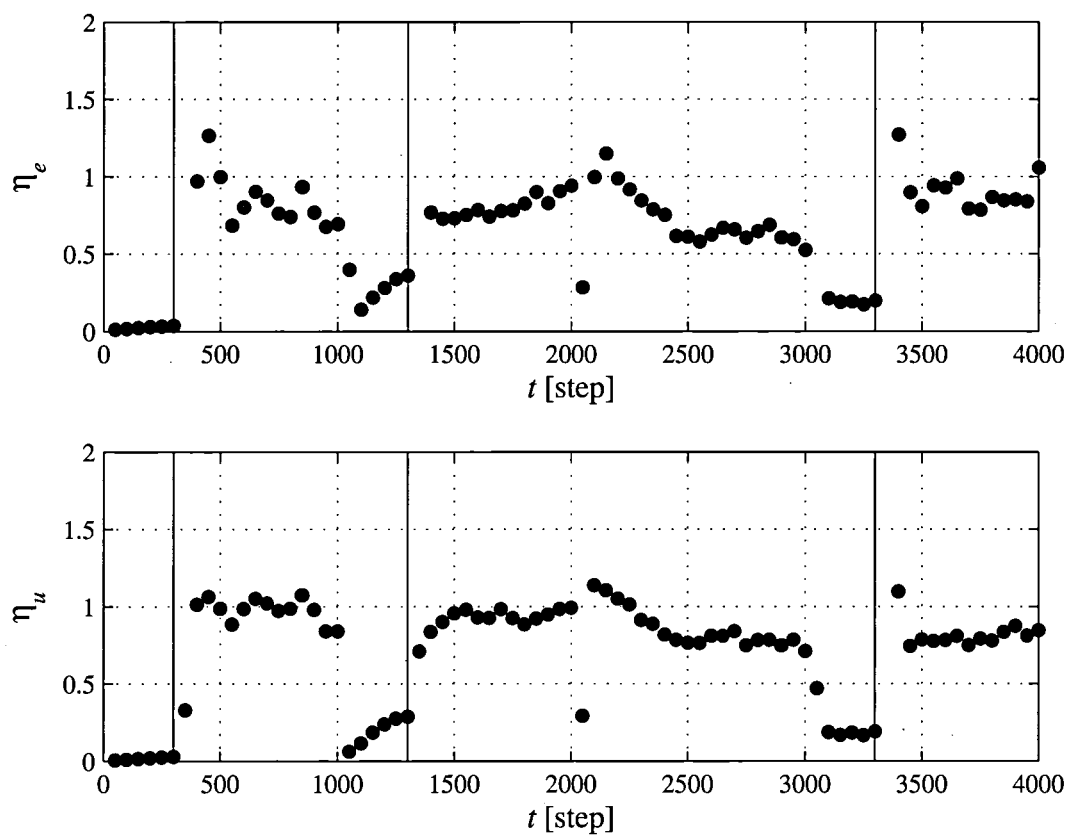


図 3.17 Trajectories of control performance assessment indices: $\eta_e(k)$ (top) and $\eta_u(k)$ (bottom) using proposed algorithm

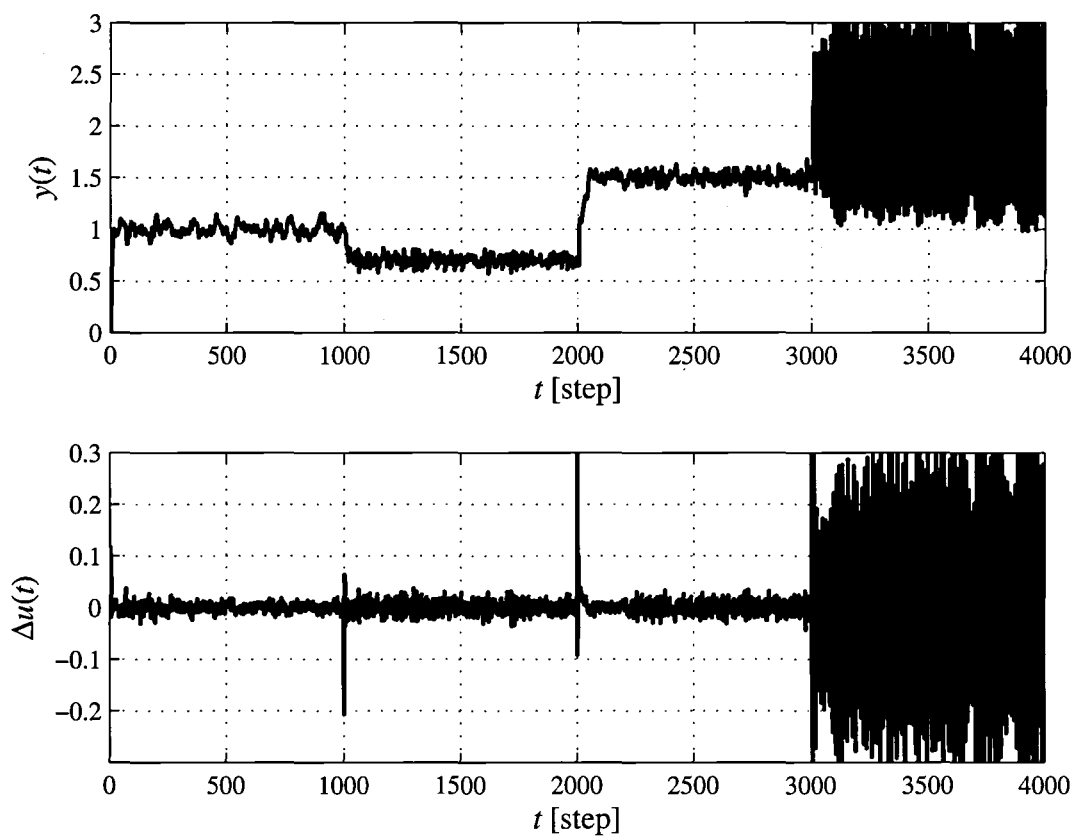


图 3.18 控制结果使用自调谐 GPC-PID 控制器

3.5 結言

本章では、制御性能評価によりセルフチューニング機構が駆動し、GPC-PID 制御則により PID パラメータを調整する制御法について提案した。GPC 制御則における定常状態での制御性能について示し、代表的な PID パラメータ調整則と比較することで GPC 制御則の有効性について説明し、設計パラメータである重み係数の設定法について提案した。逐次的に制御性能を評価する指標について示し、その指標を基にシステム同定、PID パラメータ調整、GPC 設定パラメータ λ の調整が行われるアルゴリズムについて考察した。最後にシミュレーションを行い、本手法の有効性について検証した。提案した手法は、直感的に理解できるパラメータを全般的に使用しているため、プラント従事者にとって極めて有効で有益な PID パラメータ調整法であるといえる。ただし、システム変動を検知するまである程度の時間を要するため、 M_e , M_u の設定方法も含めて今後解決していく必要がある。

第4章

パフォーマンス駆動型 CMAC-PID 制御系の設計

4.1 緒言

近年、石油精製プロセスを代表とする化学プロセス産業界では、今なお PID 制御法が主に用いられている [60]。この理由として、構造が簡単であること、比例・積分・微分といった PID パラメータの持つ物理的意味が明確であることなどが考えられる。制御系の性能を左右する PID パラメータの調整法についてはさまざまな手法 [61–63, 65] が提案されているが、プロセス制御ではシステムが非線形性を有しており、またその特性が時間とともに変動するため、適切な PID パラメータへ調整することは困難を要する。

この問題への対応として、脳内の神経伝達構造をモデル化したニューラルネットワーク (Neural Network: NN) を用いた制御法 [37, 46–53] が数多く提案されている。NN は非線形システムを精度よく制御できる有効な手法であるが、十分な学習を行なうために多くのデータと時間を必要とする。また、人の小脳皮質内における情報処理メカニズムを数学的にモデル化した小脳演算モデル (Cerebellar Model Articulation Controller: CMAC) が Albus によって提案されている [54, 55]。CMAC は、ある入力点に対応する荷重値をその近傍のいくつかの入力点で共有する構造により非線形関数を効率よく学習させることができる。これにより NN と比べ学習回数を大幅に低減することができ、また構造が簡単で設計パラメータが少ないなど利点がある。これらの特長を活かし、ロボット制御 [58] やプロセス制御 [57]、PID パラメータを調整する手法 [74] (以降 CMAC-PID 制御法と呼ぶ) などへ適用されて

いる。

一方、定常状態での稼働データからシステムの制御性能を評価・監視する研究 [80–91] が近年盛んに行なわれている。これは、時々刻々と変化するプラントシステムを常に最適な状態に制御することを目的としており、運転コストの低減や製品品質の向上が期待されている。この問題に対して、パフォーマンス・アダプティブ PID 制御系 [93] およびパフォーマンス駆動型 PID 制御系 [99] について提案されており、線形システムにおいて有効性を検証している。しかし、それらは定常特性に主眼をおき議論がなされており、過渡状態における制御性能（立上がり時間、行過ぎ量、整定時間など）を含めた考察は行なわれていない。また、線形システムを対象としているため、そのまま非線形システムに適用することが困難である。

そこで本章では、制御性能評価に基づいた CMAC-PID 制御系の設計について提案する。まず所望の過渡特性が得られるよう CMAC を学習する。つぎに、定常特性に着目し CMAC を学習する。具体的には、所望の定常特性が得られていない場合、入出力データに基づいてシステム同定を行ない、一般化予測制御 (GPC) との関連に基づいた PID チューニングにより PID パラメータを求め、これを CMAC に反映させる。本手法の特徴は、以下の通りである。

1. 所望の過渡特性と定常特性が得られる PID コントローラを、1つの CMAC で構成することができる。
2. CMAC を用いたことで、非線形システムに対しても適用可能である。

本章の構成を以下に示す。まず、CMAC および CMAC-PID 制御系について説明する。つぎに、定常状態での制御性能評価について詳述し、所望の定常特性が得られる PID パラメータの導出法について示し、CMAC-PID 制御系へ適用する方法について提案する。最後に、本手法を非線形システムに適用した数値例を示し、その有用性を検証する。

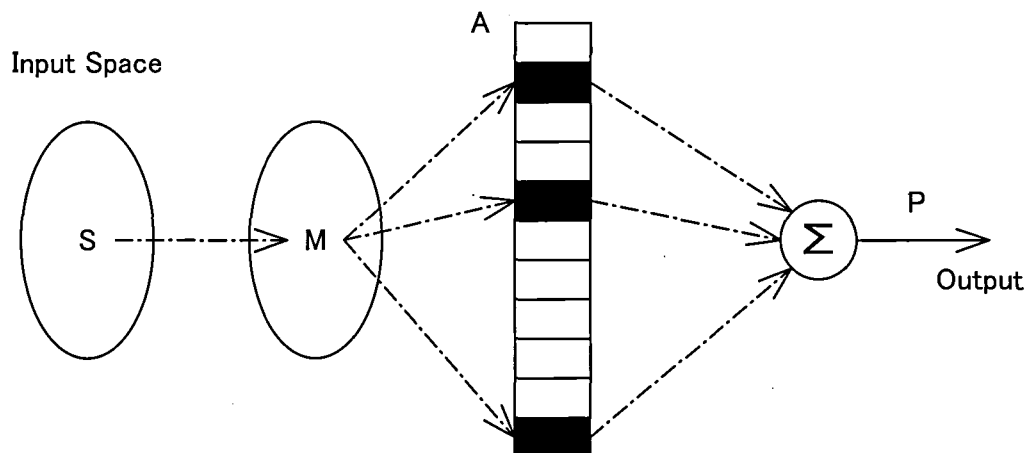


図 4.1 Schematic figure of CMAC.

4.2 CMAC-PID 制御系

4.2.1 CMAC の概要

図 4.1 に CMAC の概念図を示す。CMAC は、つぎのような一連の写像関係で表わされる [54]。

$$S \rightarrow M \rightarrow A \rightarrow P \quad (4.1)$$

ここで、 S は入力ベクトル、 M は S をコード化するための中間値、 A は M に基づいて参照する荷重表、 P は CMAC の出力値である。CMAC は、入力空間内の入力値をラベル集合に変換し ($S \rightarrow M$)、そのラベルに対応した荷重表を参照し ($M \rightarrow A$)、その参照した荷重の総和を出力値とする ($A \rightarrow P$)。このとき、目標とする値と実際の出力値との差を用いて荷重表を修正する。

CMAC の具体的な例として、1 入力 1 出力、荷重表数 3 の場合を図 4.2 に示す。実際の入力値を量子化することで $s = 7$ が得られたとする。それに対応したラベル集合 $m^* = \{C, H, K\}$ へ変換され、このラベル集合の各要素に対応した荷重値から荷重 $a^* = \{6, 4, 8\}$ を参照し、これらの総和 $\sum a^* = 18$ を出力値とする。このとき、教師信号 20 と出力値との差 2 を荷重表の数 3 で割った値、すなわち $2/3$ を参照した荷重値それぞれに加えることで荷重表を修正する。

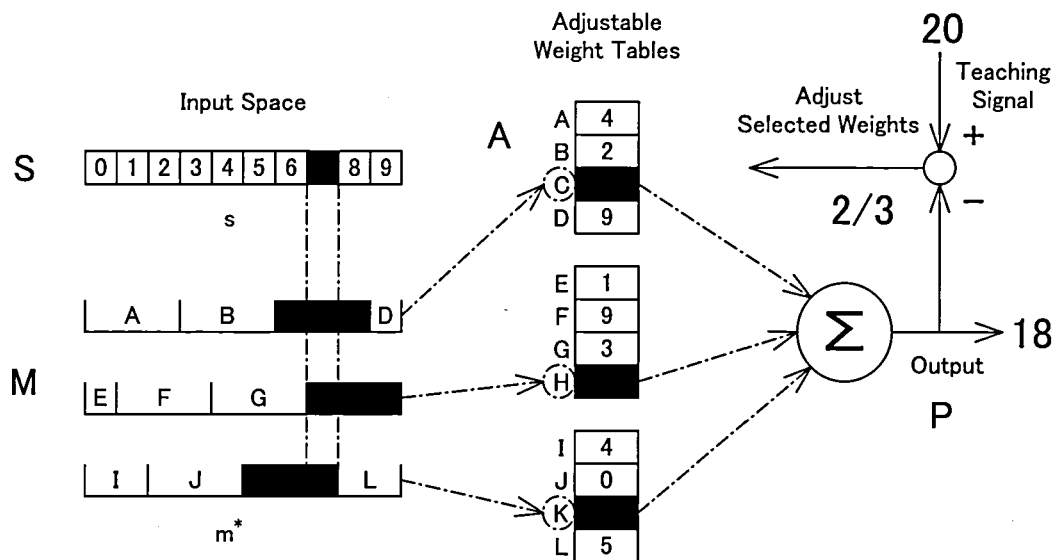


図 4.2 Example of the CMAC model with one input, one output and three weight tables.

入力空間内のある2つの入力値が近い場合（例えば，図 4.2 の $s = 7$ における $s = 6$ または $s = 8$ ），一部重複したラベル集合が得られる。この間隔が離れるにつれてラベルの重複度は減少し，ある一定間隔以上離れるとすべて異なったラベル集合を参照することとなる。このような汎化機能により，CMAC は非線形性の強いモデルを少ない学習回数で近似することができる。また，CMAC では出力計算および荷重の修正において荷重表の一部のみを参照するため，逆伝搬法による階層型 NN に比べて速い時間で出力計算および学習が可能となる。

4.2.2 PID 制御への適用

本節では，CMAC を PID 制御に適用する方法 [74] について説明する。図 4.3 に CMAC-PID 制御系の概略図を示す。次式で与えられる PID 制御則を考える。

$$\Delta u(t) = K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t) \quad (4.2)$$

ここで， $u(t)$ は制御入力， $y(t)$ は制御量， $e(t)$ は制御誤差である。 t は離散時間を表わし，サンプリング時間 T_s [sec] としている。 Δ は差分演算子を表わし， $\Delta = 1 - z^{-1}$ で定義される。 z^{-1} は時間遅延演算子を表わし， $z^{-1}u(t) = u(t-1)$ を意味する。CMAC の出力となるのは PID パラメータ K_P, K_I, K_D であり，3つの CMAC を用意する。また，CMAC へ

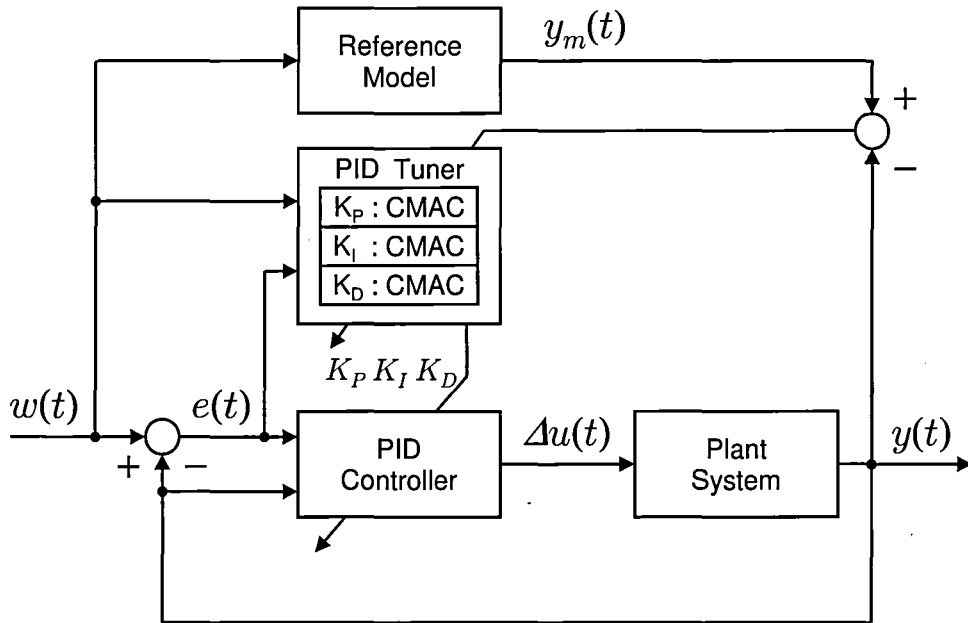


図 4.3 Block diagram of the conventional CMAC-PID controller.

の入力信号は、ステップ状で与えられる目標値 $w(t)$ 、制御誤差 $e(t)$ 、 $e(t-1)$ の 3 変量とする。ある時刻 t において、次式で表わされる評価規範を考える。

$$J(t) = \frac{1}{2}\varepsilon^2(t) = \frac{1}{2}\{y_m(t) - y(t)\}^2 \quad (4.3)$$

この評価規範を最小とする PID パラメータを出力するよう、次式に示す勾配に基づいて学習を行なう。

$$\frac{\partial J(t)}{\partial K_P} = \varepsilon(t)\Delta y(t) \frac{\partial y(t)}{\partial u(t-1)} \quad (4.4)$$

$$\frac{\partial J(t)}{\partial K_I} = -\varepsilon(t)e(t) \frac{\partial y(t)}{\partial u(t-1)} \quad (4.5)$$

$$\frac{\partial J(t)}{\partial K_D} = \varepsilon(t)\Delta^2 y(t) \frac{\partial y(t)}{\partial u(t-1)} \quad (4.6)$$

ここで、 $\partial y(t)/\partial u(t-1)$ はシステムヤコビアンである。これにより、CMAC の荷重表は次式により更新することで学習を行なう。

$$W_{P,h}^{\text{new}}(t) = W_{P,h}^{\text{old}}(t) - g_P \frac{\partial J}{\partial K_P} \frac{1}{N_w} \quad (4.7)$$

$$W_{I,h}^{\text{new}}(t) = W_{I,h}^{\text{old}}(t) - g_I \frac{\partial J}{\partial K_I} \frac{1}{N_w} \quad (4.8)$$

$$W_{D,h}^{\text{new}}(t) = W_{D,h}^{\text{old}}(t) - g_D \frac{\partial J}{\partial K_D} \frac{1}{N_w} \quad (4.9)$$

ここで、 W^{old} は参照した荷重値であり、 W^{new} はそれを更新した荷重値を表わしている。また、 $h = 1, 2, \dots, N_w$ で、 N_w は参照する荷重表の総数を表わす。さらに、 g_P, g_I, g_D は荷重修正の割合を決定する学習係数である。

CMAC-PID 制御では、学習を行なう前に適切な初期荷重表を作成する必要がある。オーバーシュートがない過制動な応答波形が得られる比較的低感度の PID パラメータを初期値として選べばよいことが知られている。しかし、式 (4.3) からわかるように、誤差を最小とするように荷重表が更新され、制御入力の変化量については考慮されていない。また、式 (4.4)–(4.6) からわかるように、定常状態では各勾配量は小さくなり、適切な荷重表を得るためには多くの学習回数を必要とする。しかし、学習回数の増加により過渡状態において不安定な挙動を示し、過学習という結果を生じることが知られている。すなわち、少ない学習回数で得られた荷重表では、所望の定常特性が得られないことを意味する。

4.3 パフォーマンス駆動型 CMAC-PID 制御系の設計

4.3.1 概要

これまでに、制御性能評価機能を有する PID 制御系の設計 [93, 99] について提案がされている。しかし、これらの研究ではそれらは定常特性にのみ焦点を当て、制御性能評価に基づいて PID パラメータを求める方法について議論されている。そこで、過渡状態において優れた性能改善が期待できる従来の CMAC-PID コントローラに、定常状態における制御性能評価機能を付加した新しい CMAC-PID コントローラの構成について考察する。

4.3.2 モデル予測型 PID 制御

対象とする制御システムの記述モデル（詳細モデル）を「一次遅れ+むだ時間」系として与える。

$$G(s) = \frac{K}{1 + Ts} e^{-Ls} \quad (4.10)$$

ここで、システムパラメータ K, T, L はそれぞれシステムゲイン、時定数、むだ時間を表わしている。さらに、離散時間伝達関数モデルへ変換した次式を考える。

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) \quad (4.11)$$

$$\left. \begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} \end{aligned} \right\} \quad (4.12)$$

つぎに、式 (4.2) で与えられる PID 制御則を考える。以下の考察を簡単にするため、式 (4.2) を次式として書き換える。

$$L(z^{-1})y(t) - L(1)w(t) + \Delta u(t) = 0 \quad (4.13)$$

$$\begin{aligned} L(z^{-1}) &= K_P + K_I + K_D \\ &\quad - (K_P + 2K_D)z^{-1} + K_D z^{-2} \end{aligned} \quad (4.14)$$

ここでは、GPC 則に基づいて PID パラメータ (K_P, K_I, K_D) を調整する [44]。GPC 則では、つぎの評価関数を最小化することを考える。

$$J_g(t) = \mathbf{E} \left[\sum_{j=1}^N \{y(t+j) - w(t)\}^2 + \lambda \sum_{j=1}^N \{\Delta u(t+j-1)\}^2 \right] \quad (4.15)$$

ここで、 N は予測区間、 λ はシステム入力の増分に対する重み係数、 $\mathbf{E}[\cdot]$ は期待値を表わしている。式 (4.15) を最小にする制御則は、次式のように得られる。

$$\sum_{j=1}^N \{p_j F_j(z^{-1})\} y(t) - \sum_{j=1}^N p_j w(t) + \left\{ 1 + z^{-1} \sum_{j=1}^N p_j s_j \right\} \Delta u(t) = 0 \quad (4.16)$$

ここで、 $F_j(z^{-1}), p_j, s_j$ は、つぎの Diophantine 方程式を解くことで得られる。

$$1 = \Delta A(z^{-1}) E_j(z^{-1}) + z^{-j} F_j(z^{-1}) \quad (4.17)$$

$$E_j(z^{-1}) B(z^{-1}) = R_j(z^{-1}) + z^{-j} s_j \quad (4.18)$$

ここで、

$$E_j(z^{-1}) = 1 + e_1 z^{-1} + \cdots + e_{j-1} z^{-(j-1)} \quad (4.19)$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1} z^{-1} + f_{j,2} z^{-2} \quad (4.20)$$

$$R_j(z^{-1}) = r_0 + r_1 z^{-1} + \cdots + r_{j-1} z^{-(j-1)} \quad (4.21)$$

$$[p_1, p_2, \cdots, p_N] = [1, 0, \cdots, 0] (\mathbf{R}^T \mathbf{R} + \lambda \mathbf{I})^{-1} \mathbf{R}^T \quad (4.22)$$

$$\mathbf{R} = \begin{bmatrix} r_0 & & & \\ r_1 & r_0 & & \mathbf{0} \\ \vdots & & \ddots & \\ r_{N-1} & r_{N-2} & \cdots & r_0 \end{bmatrix} \quad (4.23)$$

である。いま、式 (4.16) 左辺第 3 項を定常ゲインに置き換え、

$$\tilde{F}(z^{-1})y(t) - \tilde{F}(1)w(t) + \Delta u(t) = 0 \quad (4.24)$$

$$\tilde{F}(z^{-1}) = \tilde{f}_0 + \tilde{f}_1 z^{-1} + \tilde{f}_2 z^{-2}$$

を得る。式 (4.13) と式 (4.24) を比較することにより、近似的に GPC 則と PID 制御則が等価となる。

$$K_P = -(\tilde{f}_1 + 2\tilde{f}_2) \quad (4.25)$$

$$K_I = \tilde{f}_0 + \tilde{f}_1 + \tilde{f}_2 \quad (4.26)$$

$$K_D = \tilde{f}_2 \quad (4.27)$$

以上により、GPC 則に基づいて PID パラメータを調整することができる。GPC 則に含まれる予測区間 N と重み係数 λ は、設計者が与える設計パラメータである。予測区間 N は、実用的には時定数 T とむだ時間 L の和に相当するステップ数あたりに設定するのが良いとされている [44]。また、第 3 章において、定常状態における制御性能の観点から、閉ループ系におけるステップ応答出力波形の最大行過ぎ量 $M_0[\%]$ に基づいて λ を調整する方法 [99] について提案を行なった。本研究でもこの方法により λ を決定する。

4.3.3 制御性能評価

まず、式 (3.1) の詳細モデルに対応した、つぎの離散時間モデルを考える。

$$\alpha(z^{-1})y(t) = z^{-(d+1)}\beta(z^{-1})u(t) + \xi(t)/\Delta \quad (4.28)$$

$$\left. \begin{aligned} \alpha(z^{-1}) &= 1 + \alpha_1 z^{-1} \\ \beta(z^{-1}) &= \beta_0 + \beta_1 z^{-1} \end{aligned} \right\} \quad (4.29)$$

ここで、 d は離散時間におけるむだ時間を表わし、 $\xi(t)$ は平均 0、分散 σ_ξ^2 のガウス性白色雑音とする。

このとき、定常状態での制御誤差 $e(t)$ と制御差分入力 $\Delta u(t)$ の分散は、 H_2 ノルムを用いて次式で計算できる [93]。

$$\hat{\sigma}_e^2 = E[\{e(t)\}^2] = \left\| -\frac{1}{T(z^{-1})} \right\|_2^2 \cdot \sigma_\xi^2 \quad (4.30)$$

$$\hat{\sigma}_u^2 = E[\{\Delta u(t)\}^2] = \left\| -\frac{L(z^{-1})}{T(z^{-1})} \right\|_2^2 \cdot \sigma_\xi^2 \quad (4.31)$$

ただし、 $T(z^{-1})$ は次式で定義している。

$$T(z^{-1}) = \Delta\alpha(z^{-1}) + z^{-(d+1)}\beta(z^{-1})L(z^{-1}) \quad (4.32)$$

プロセスのシステムパラメータが既知またはよい精度で推定されている場合、制御誤差および制御差分入力の分散は、式 (4.30) および式 (4.31) から推定できる。

また、時間 t において、実際の操業データから得られる制御誤差の分散 $\sigma_e^2(t)$ 、制御差分入力の分散 $\sigma_u^2(t)$ は、それぞれデータ長 M_e, M_u として制御誤差 $e(t)$ 、制御差分入力 $\Delta u(t)$ の二乗平均で求められる。これらは、それぞれ $\hat{\sigma}_e^2, \hat{\sigma}_u^2$ の周りに分布し、ある一定の範囲内に収まる。もし、 $\sigma_e^2(t), \sigma_u^2(t)$ がその範囲内に入らなかった場合は、所望の定常特性が得られていないことを意味し、システム同定によりシステムパラメータおよび PID パラメータを再調整する必要がある。このような考え方にに基づき、以下の制御性能評価指数を定義する [81]。

$$\eta_e(t) = \frac{\hat{\sigma}_e^2}{\sigma_e^2(t)}, \quad \eta_u(t) = \frac{\hat{\sigma}_u^2}{\sigma_u^2(t)} \quad (4.33)$$

ここで、 $\eta_e(t), \eta_u(t)$ は 1 に近いほど制御性能がよく、0 または無限大へ近づくにつれ制御性能が悪くなる。これらの値が設計者が設定する範囲内

$$\eta_e^{\min} \leq \eta_e(t) \leq \eta_e^{\max}, \quad \eta_u^{\min} \leq \eta_u(t) \leq \eta_u^{\max} \quad (4.34)$$

にあれば、現在の制御性能は良いと判断する。

4.3.4 システム同定

式 (4.29) の係数 $\alpha_1, \beta_0, \beta_1$ の推定値 $\hat{\alpha}_1(t), \hat{\beta}_0(t), \hat{\beta}_1(t)$ は、つぎの逐次最小二乗法により求めることができる。

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\Gamma(t-1)\psi(t-1)}{\rho + \psi^T(t-1)\Gamma(t-1)\psi(t-1)} \cdot \varepsilon_p(t) \quad (4.35)$$

$$\Gamma(t) = \frac{1}{\rho}\Gamma(t-1) - \frac{\Gamma(t-1)\psi(t-1)\psi^T(t-1)\Gamma(t-1)}{1 + \psi^T(t-1)\Gamma(t-1)\psi(t-1)} \quad (4.36)$$

$$\varepsilon_p(t) = \Delta y(t) - \hat{\theta}(t-1)^T \psi(t-1) \quad (4.37)$$

$$\hat{\theta}(t) = [\hat{\alpha}_1(t), \hat{\beta}_0(t), \hat{\beta}_1(t)]^T \quad (4.38)$$

$$\psi(t-1) = [-\Delta y(t-1), \Delta u(t-d-1), \Delta u(t-d-2)]^T \quad (4.39)$$

ここで、 ρ は忘却係数 ($0 < \rho \leq 1$) を、 $\varepsilon_p(t)$ は予測誤差を表わしている。このとき、システムパラメータの推定値 \hat{K} , \hat{T} , \hat{L} は、次式により計算できる [97]。

$$\hat{K}(t) = \frac{\hat{\beta}_0(t) + \hat{\beta}_1(t)}{1 + \hat{\alpha}_1(t)} \quad (4.40)$$

$$\hat{T}(t) = -\frac{T_s}{\log_e(-\hat{\alpha}_1(t))} \quad (4.41)$$

$$\hat{L}(t) = \left\{ \frac{\hat{\beta}_1(t)}{\hat{\beta}_0(t) + \hat{\beta}_1(t)} + d \right\} \cdot T_s \quad (4.42)$$

4.3.5 荷重表の更新方法

CMAC-PID コントローラの荷重表には、PID パラメータが記録されている。そこで、定常状態で望ましい性能が得られる PID パラメータ ($\hat{K}_P, \hat{K}_I, \hat{K}_D$) が求められたとき、荷重表において現在の目標値 $w(t)$ および $e(t) = 0$, $e(t-1) = 0$ に相当する荷重値を次式により変更する。

$$W_{P,h} = \hat{K}_P / N_w, \quad (h = 1, 2, \dots, N_w) \quad (4.43)$$

$$W_{I,h} = \hat{K}_I / N_w, \quad (h = 1, 2, \dots, N_w) \quad (4.44)$$

$$W_{D,h} = \hat{K}_D / N_w, \quad (h = 1, 2, \dots, N_w) \quad (4.45)$$

ここで、 $W_{P,h}, W_{I,h}, W_{D,h}$ は、過渡応答波形を学習させた後に得られる荷重値を表わしている。前述したとおり、CMAC は入力値に応じて参照する荷重値が異なるため、式 (4.43)–(4.45) で荷重値を書き換えることで、過渡特性に影響を及ぼすことなく、定常特性が改善されるコントローラが構成できる。つまり、1つの CMAC-PID コントローラで過渡状態と定常状態の両面において所望の制御特性が得られることを意味する。

4.3.6 パフォーマンス駆動型 CMAC-PID コントローラ

図 4.4 に本手法の概略図を示す。これまでに考察した手続きをまとめ、以下のようにアルゴリズムを構築する。

[アルゴリズム]

- 1° 荷重表を初期化する。
- 2° 所望の過渡特性が得られるよう、CMAC-PID コントローラの荷重表を学習する。

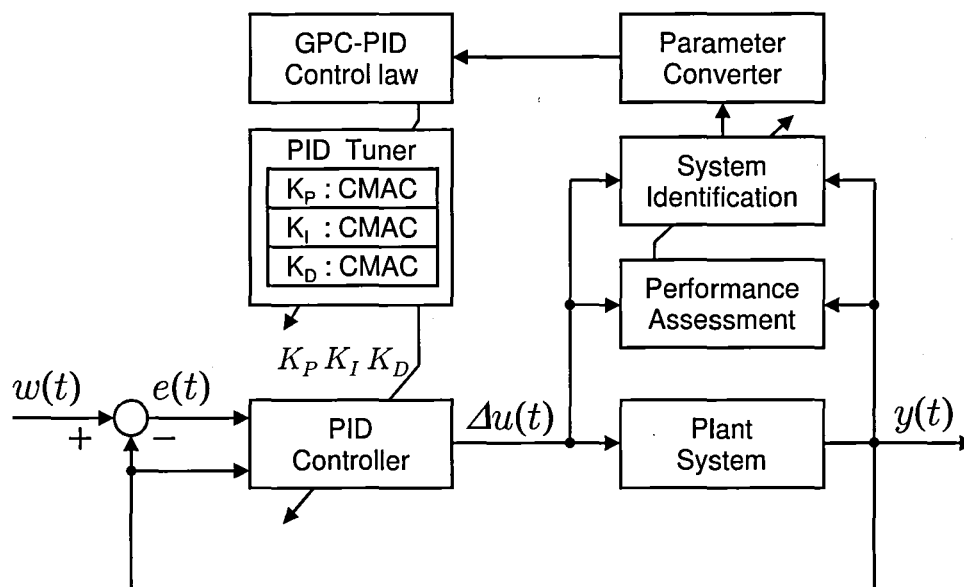


図 4.4 Block diagram of the proposed CMAC-PID controller.

3° 所望の定常特性が得られていない場合，システム同定を行ない，制御性能評価に基づいた PID パラメータを求め，式 (4.43)–(4.45) により CMAC-PID コントローラの荷重表を学習する。

4.4 数値例

本論文で提案した手法の有効性をシミュレーションにより検証する。制御対象として，以下の式で表わされる Bilinear モデル [100] を考える。

$$\begin{aligned}
 y(t) = & 0.4y(t-1) - 0.09y(t-2) + 0.3u(t-d-1) \\
 & - 0.1u(t-d-2) + 0.1y(t-1)u(t-d-1) \\
 & + 0.05y(t-2)u(t-d-2) + \frac{\xi(t)}{\Delta}
 \end{aligned} \tag{4.46}$$

ここで， $\xi(t)$ は平均 0，分散 0.02^2 のガウス性白色雑音を用いた。図 4.5 に示すこのモデルの静特性から，非線形性を有することは明らかである。また，目標値 $w(t)$ は以下のように設定した。

$$w(t) = \begin{cases} 2.0 & (0 \leq t < 1000) \\ 4.0 & (1000 \leq t < 2000) \\ 6.0 & (2000 \leq t \leq 3000) \end{cases} \tag{4.47}$$

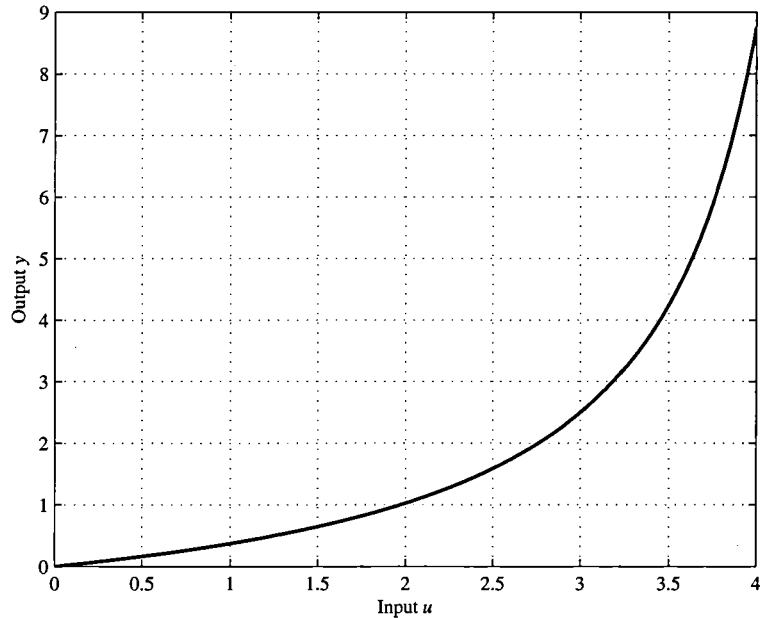


図 4.5 Static property of the bilinear system.

さらに、参照モデルとして次式を用いた。

$$y_m(t) = 0.8y_m(t-1) + 0.2w(t-d-1) \quad (4.48)$$

また、サンプリング時間を $T_s = 1.0[\text{sec}]$ とし、むだ時間は既知であると仮定し $d = 3$ とした。CMAC の設計パラメータである量子化幅を 0.25, 荷重表の数を $N_w = 4$, 学習係数を $g_P = 10^{-2}$, $g_I = 10^{-5}$, $g_D = 4 \times 10^{-2}$ のようにそれぞれ設定した。初期 PID パラメータは、4.5 の $u = 4$ におけるシステムパラメータをシステム同定により導出し、CHR 法により求められた値を若干変更し、 $K_P = 0.41$, $K_I = 0.014$, $K_D = 0.43$ と設定した。GPC 則で用いられる設計パラメータについて、 λ を $M_0 = 15[\%]$ となるように設定した。システム同定に使用される忘却係数を $\rho = 1.00$, データ長を 400 とした。CPA に用いられる設計パラメータでは、 $M_e = M_u = 300$, $\eta_e^{\min} = \eta_u^{\min} = 0.5$, $\eta_e^{\max} = \eta_u^{\max} = 1.5$ と設定し、50[step] ごとに制御性能評価指数 $\eta_e(t)$, $\eta_u(t)$ を求め制御性能の評価を行なった。目標値に変更があった場合、評価指数が安定する 500[step] が過ぎるまでシステム同定を行なわないようにした。

上記のパラメータによりシミュレーションを行なった。まず所望の過渡応答特性が得られるよう荷重表の学習を行なった。初期状態による CMAC-PID 制御系の応答波形および制

御差分入力信号の時間推移を図 4.6 に、12 回の学習を行なった後の応答波形および制御差分入力信号の時間推移を図 4.7 に、そのときの PID ゲインの時間推移を図 4.8 にそれぞれ示す。図 4.6 および図 4.7 から、過渡特性における改善は認められるが、目標値 $w(t) = 6.0$ において定常状態での制御誤差および制御差分入力の分散が大きくなっており、定常性能が著しく劣化していることがわかる。これは、図 4.8 からわかるように、定常状態での PID パラメータが適切に更新されないために起こる。また、 K_P および K_I の定常状態におけるパラメータ値は、初期条件で与えたパラメータとほぼ同じ値となっており、少ない学習回数では適切に更新できないことがわかる。

つぎに、所望の定常特性が得られるよう提案した手法を適用した。その結果を図 4.9 に、そのときの PID ゲインの時間推移を図 4.10 に、制御性能評価指数の時間推移を図 4.11 にそれぞれ示す。これらより、500[step],1500[step],2500[step] において制御性能評価機能が働き、システム同定が行なわれ PID パラメータが再調整されていることがわかる。その後は制御性能評価指数 $\eta_e(t)$, $\eta_u(t)$ はそれぞれ 1 付近に収束していることから、所望の制御性能が得られていることがわかる。また、図 4.9 からわかるように、各目標値において制御性能評価機能が動作した後は、制御誤差の分散が低く抑えられている。また、区間 $t = 2500\text{--}3000[\text{step}]$ では、制御誤差分散のみならず制御差分入力分散についても改善されていることがわかる。

さらに、本手法により作成された荷重表を用いて、再度同様のシミュレーションを行なった。その結果を図 4.12 に、そのときの PID ゲインの時間推移を図 4.13 にそれぞれ示す。以上の結果から、1 つの CMAC-PID コントローラで、過渡特性を維持したまま、定常特性を改善させることができることが確かめられた。

比較のため、文献 [99] によるパフォーマンス駆動型 GPC-PID 制御系により同様のシミュレーションを行なった。その結果を図 4.14 に示す。目標値が変化することでシステムの変動に追従できず、不安定な動作を引き起こしていることがわかる。これらの結果から、過渡状態における速応性、定常状態における追従性の両面において本手法の有効性が確かめられた。

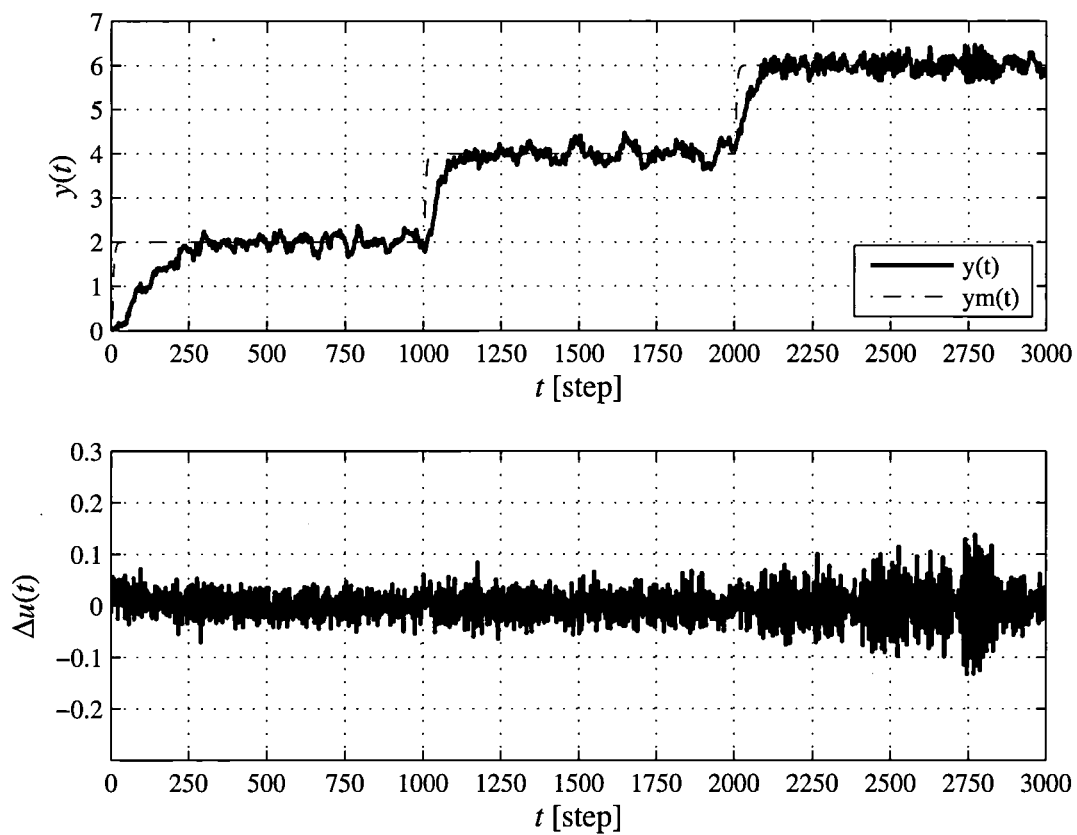


図 4.6 Control result using the CMAC-PID controller with the initial settings.

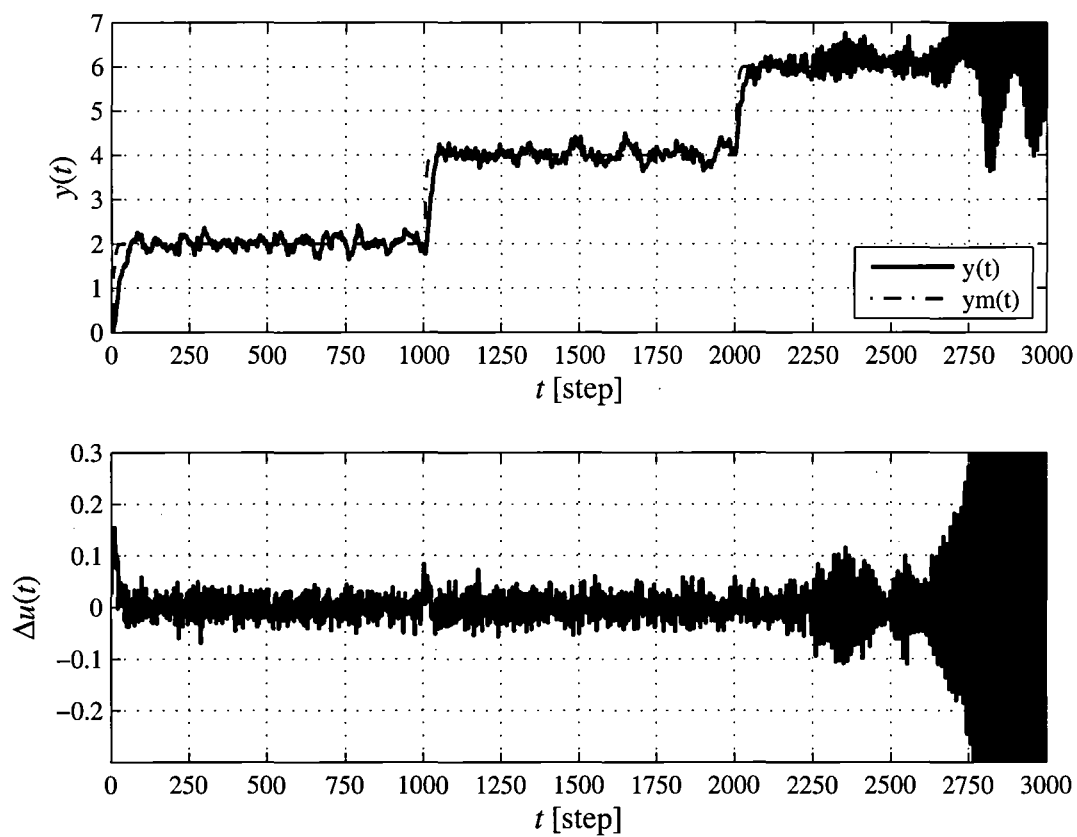


图 4.7 控制结果使用 CMAC-PID 控制器在学习瞬态状态后。

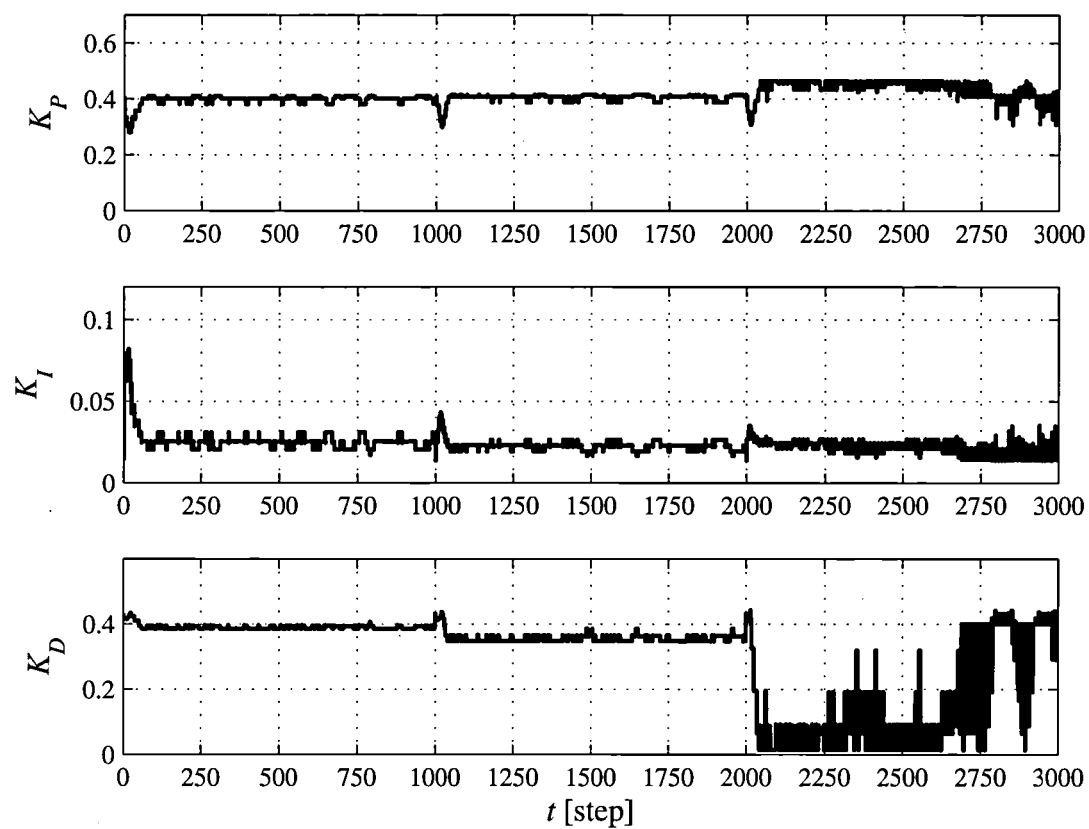


図 4.8 Trajectories of PID parameters corresponding to Fig.4.7.

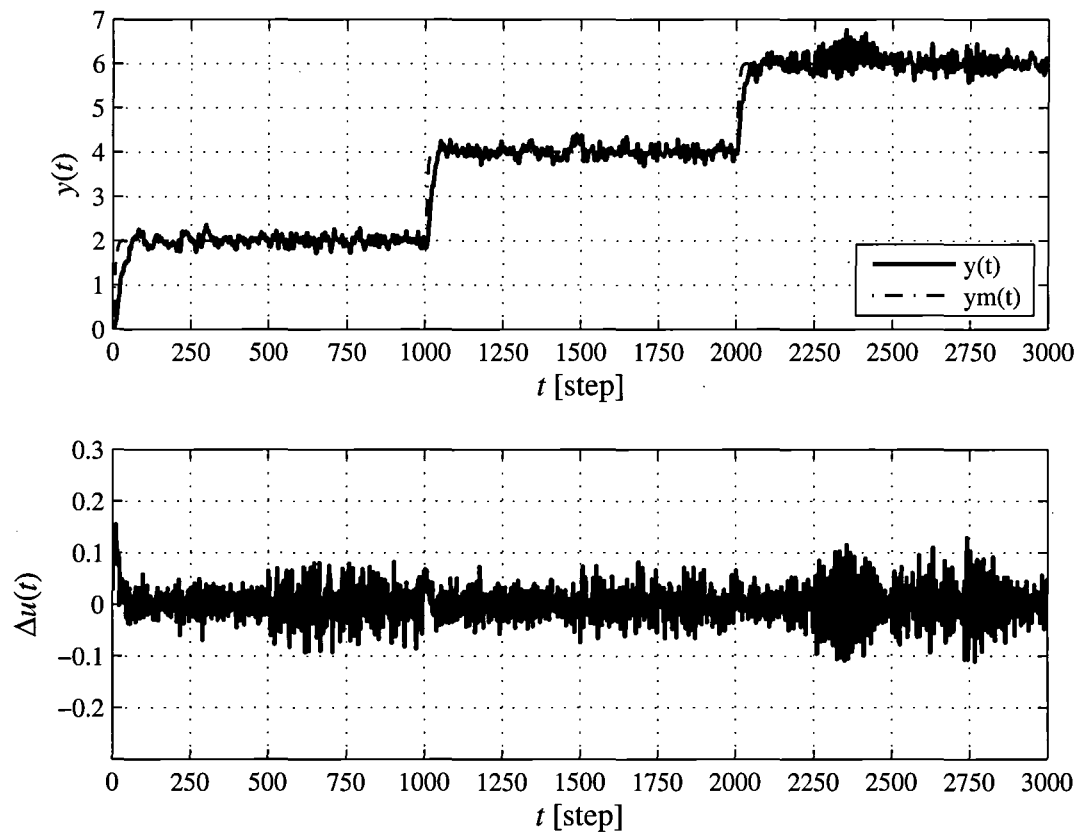


图 4.9 控制结果使用 CMAC-PID 控制器在学习稳态后。

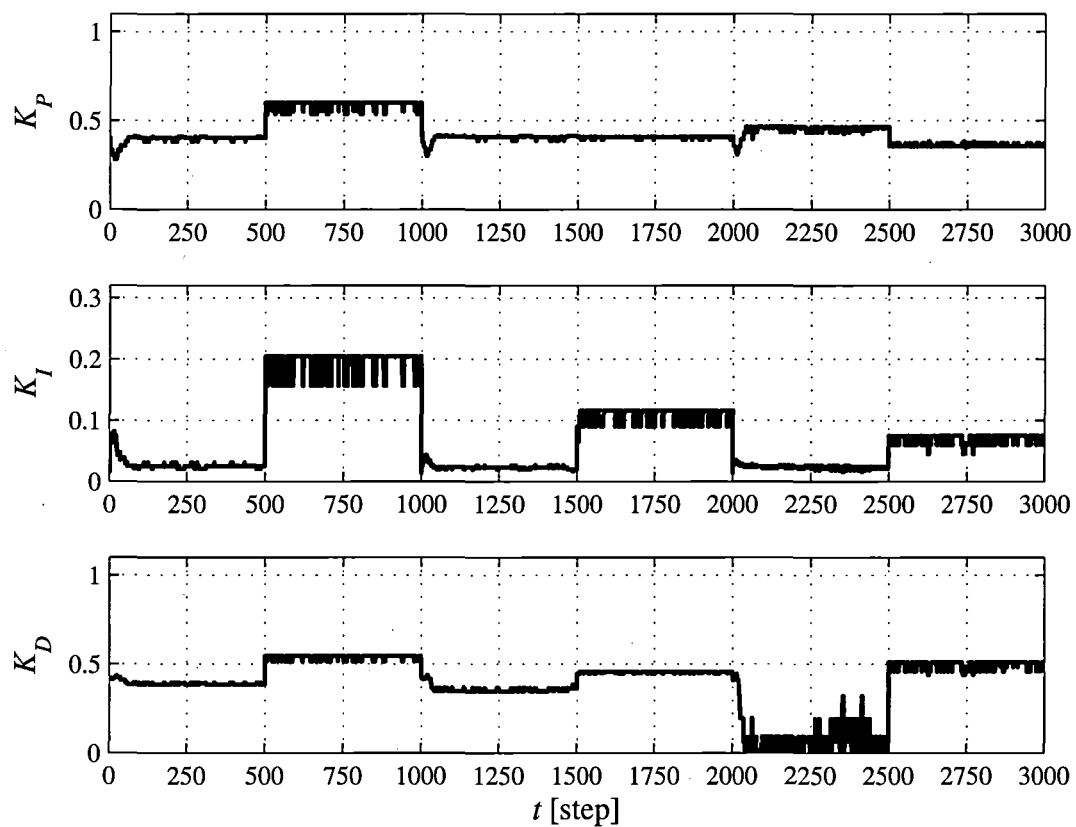


図 4.10 Trajectories of PID parameters corresponding to Fig.4.9.

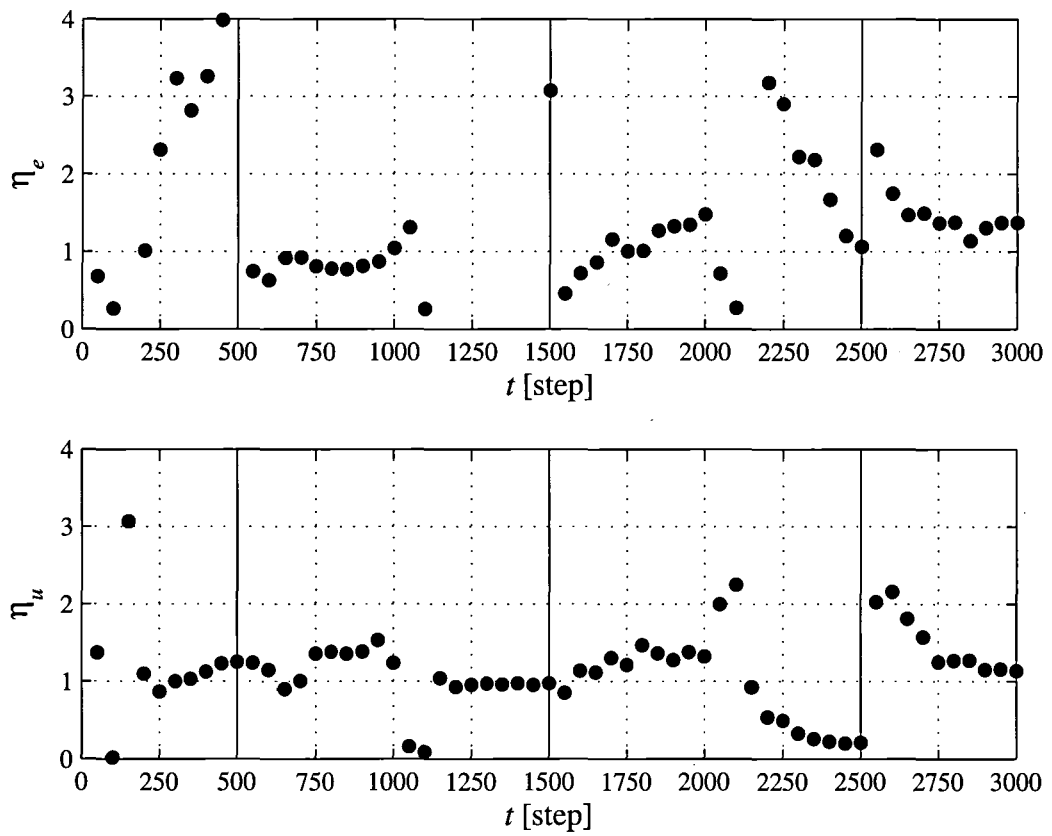


图 4.11 Trajectories of CPA indices corresponding to Fig.4.9.

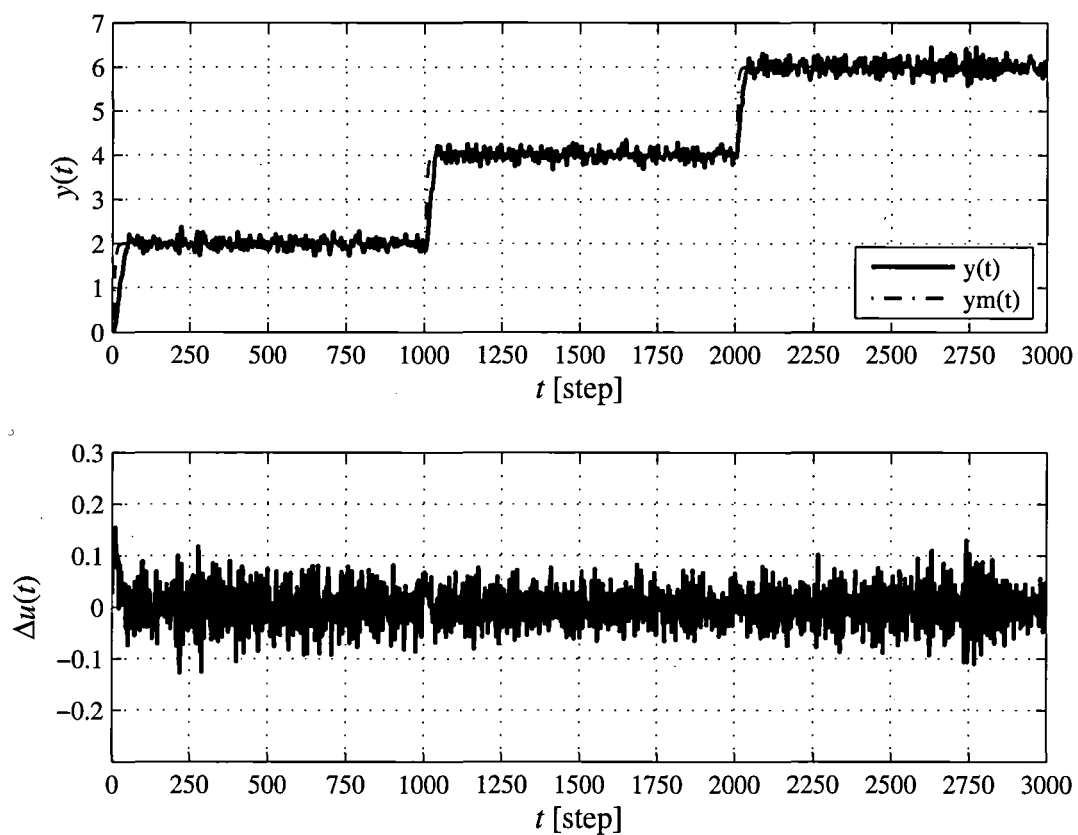


図 4.12 Control result using the weight tables after the proposed scheme.

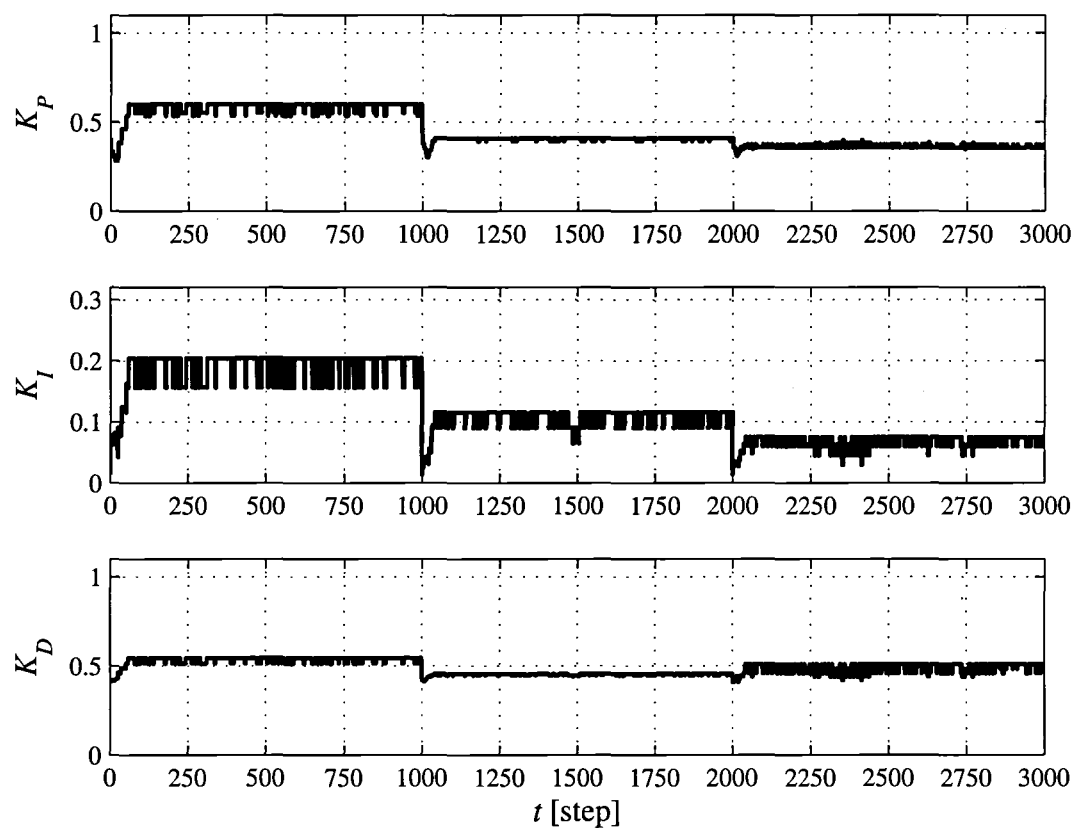


图 4.13 Trajectories of PID parameters corresponding to Fig.4.12.

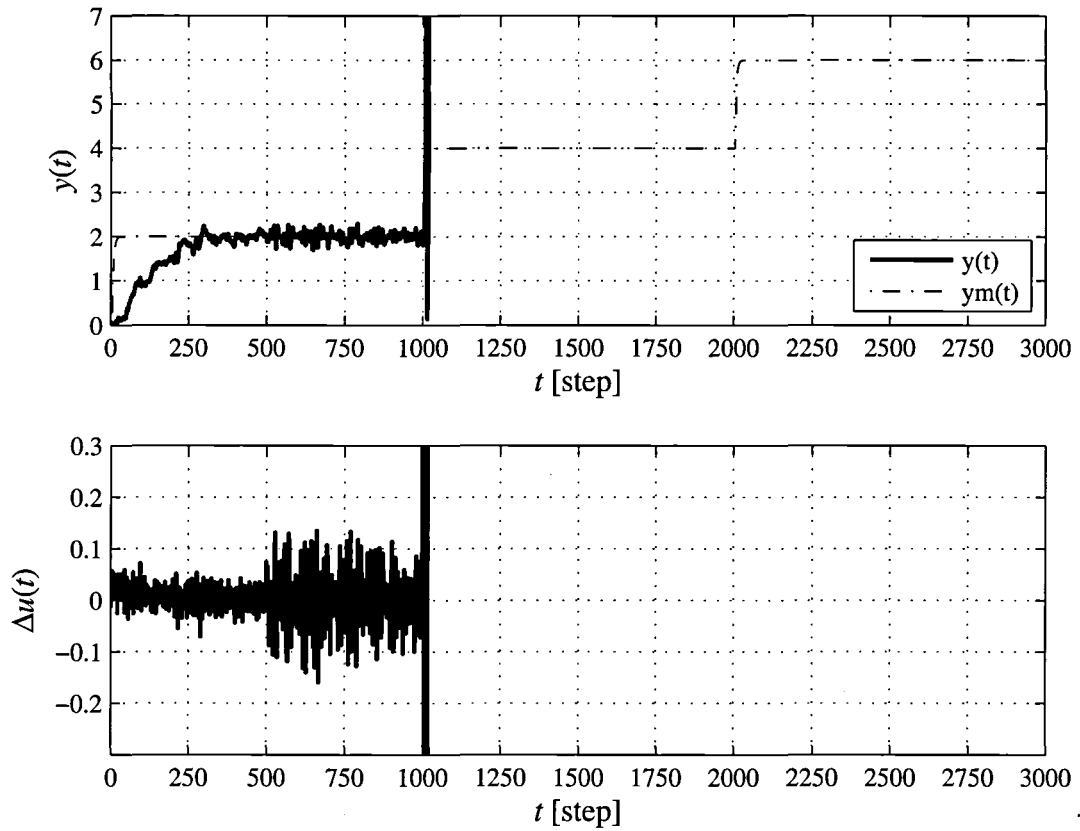


図 4.14 Control result using the performance-driven GPC-PID controller.

4.5 結言

本章では、制御性能評価機能を有する CMAC-PID 制御系の設計について提案した。本手法によると、(1) 所望の過渡特性と定常特性を得る PID コントローラが一つの CMAC で構成することができ、また、(2) 非線形システムに対しても適用することができる。特に、(1) の特徴は、熱プロセスのような即応性と安定性の双方が重視されるプロセスに有効であると考えられる。今後は、本手法を射出成形プロセスなどの熱プロセスに適用し、本手法の有用性を検証する予定である。

第5章

結 論

本論文は、2次元フィルタおよびプロセスシステムで用いられる適応アルゴリズムの設計に関する研究成果をまとめたものである。本研究を終えるにあたって、全体を通して総括を行い、今後の課題について言及する。

第2章では、2次元直交射影アルゴリズムを用いた適応 FIR フィルタの設計について論じた。2次元へ拡張した直交射影アルゴリズムが示され、その収束条件を導いた。2次元データの自由度を考慮し、水平・垂直の2方向から処理を行う直交射影アルゴリズムについて考察した。さらに、水平方向および垂直方向の入力信号を直交化させる方法を示した。これにより、2次元データの特徴を活用したフィルタを構築でき、推定精度の高いフィルタが実現できることが期待される。最後に、提案した3つのアルゴリズムをそれぞれ2次元システム同定に適用し、従来法と比較することによって本手法の有効性を検証した。今後は、高速処理向き構造やハードウェア実装化に向けた検討が必要である。

第3章では、制御性能評価と制御系設計を融合するモデル予測型 PID 制御系の設計について論じた。定常状態における制御性能に関して、代表的な PID 制御則との比較により GPC 制御則の有効性について検証した。制御系従事者にとって直感的に理解できるパラメータを用いて GPC-PID 制御則の設計パラメータの調整を行うことにより、プラント従事者にとって極めて有効で有益な PID パラメータ調整法を確立した。また、定常状態における制御性能を評価することで、性能が劣化した PID 制御器のパラメータ調整を設計パラメータの再調整により行う制御系を構築した。コンピュータシミュレーションにより、従来法であるセルフチューニングコントローラと比較することで、定常状態における制御性能の改善やシステム同定回数的大幅な削減などの観点から本手法の有効性を検証した。今後の課

題として、システム変動を検知するまである程度の時間を要するため、 M_e , M_u の設定方法について解決していく必要がある。また、生物進化論的 (GA) アルゴリズムを適用し、制御性能評価に基づいて GPC 設定パラメータを調整することで、より性能に優れた PID コントローラが構築できると考えられる。

第4章では、非線形プラントシステムを対象とした制御性能評価機能を有する PID 制御系の設計について論じた。第3章による手法を適用し、CMAC-PID 制御系に含まれる荷重値を更新させることで、所望の定常特性が得られるように CMAC-PID 制御系を適応させるアルゴリズムについて考察した。これにより、非線形システムにおいて所望の過渡特性および定常特性が得られる制御系が構築できる。最後にコンピュータシミュレーションにより、従来法であるセルフチューニングコントローラと比較することで、過渡特性および定常特性の観点から本手法の有効性を検証した。CMAC の特徴から過渡特性が特別に優れているとは言えず、今後の課題として、プラントが時間により変化した場合の対応を考慮した CMAC-PID 制御系の設計について、今後さらに発展させていく必要がある。また、他の知的情報処理技術による定常特性を明らかにし、制御性能評価機能を有する新しい制御系を構築することが挙げられ、さらに考察を重ねていく必要がある。

謝 辞

本論文は、筆者が広島大学大学院工学研究科 博士課程後期 複雑システム工学専攻に入学し、広島大学大学院工学研究科 複雑システム工学専攻の雛元孝夫教授の御指導のもとに作成したものである。懇切な御指導ならびに御鞭撻を賜りながら本研究を遂行することができた。本稿を終えるにあたり、甚大なる感謝を捧げます。

本研究を遂行するにあたり、広島大学大学院教育学研究科 科学文化教育学専攻の山本透教授には、熱心な御教示、丁寧な御指導、並びに有益な御助言を賜りました。ここに深甚の謝意を捧げます。

本研究をまとめるにあたり、広島大学大学院工学研究科複雑システム工学専攻の西崎一郎教授、並びに機械システム工学専攻の佐伯正美教授には、有益な御教示と御配慮を賜りました。ここに厚く敬意を呈します。

そして本論文の内容について、筆者が学生のころからひとかたならぬ御忠告と貴重な御助言を頂いた関西大学工学部 電子情報システム工学科の棟安実治教授には、深く感謝申し上げます。

さらに、筆者が赴任している津山工業高等専門学校で日頃から多岐にわたり御指導、御助力を頂いている、電気電子工学科の下西二郎教授には、心から感謝の意を表します。

これまでの研究過程において、津山工業高等専門学校 元校長 大森晋爾博士、前校長 阿部武治博士、校長 稲葉英男博士には多大な配慮を賜り、厚く御礼申し上げます。

日頃から研究進行に際し、種々の面でお世話になった広島大学大学院工学研究科複雑システム工学専攻複雑システム制御論研究室並びに教育学研究科科学文化教育学専攻技術・情報系コース研究室の諸氏に深謝します。

最後に本研究を進めるにあたり、終始暖かい御支援と御配慮を賜りました津山工業高等専門学校電気電子工学科 伊藤國雄教授、田邊茂教授、植月唯夫教授そして原田寛治教授をはじめ、教職員の皆様ならびに学生諸氏にあらためて感謝申し上げます。

参考文献

- [1] 荒木光彦. デジタル制御理論入門. システム制御情報ライブラリー, No. 4. 朝倉書店, 1991.
- [2] 古田勝久 (編集). デジタル制御. コンピュートロール, No. 27. コロナ社, 1989.
- [3] 木村英紀. デジタル信号処理と制御. 昭晃堂, 1984.
- [4] K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, New York, 1989.
- [5] 北森俊行, 新誠一 (編集). 適応制御. コンピュートロール, No. 32. コロナ社, 1990.
- [6] 樋口龍雄 (編集). 多次元デジタル信号処理. コンピュートロール, No. 30. コロナ社, 1990.
- [7] S. Haykin. *Introduction to Adaptive Filters*. Macmillan Publishing Company, New York, 1984.
- [8] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1985.
- [9] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
- [10] 武部幹. 適応フィルタ入門. 現代工学社, 1987.
- [11] M. M. Hadhoud and D. W. Thomas. The Two-Dimensional Adaptive LMS (TDLMS) Algorithm. *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 5, pp. 485–494, 1988.
- [12] M. Ohki and S. Hashiguchi. Two-Dimensional LMS Adaptive Filters. *IEEE Trans. on Consumer Electronics*, Vol. 37, No. 1, pp. 66–73, Feb. 1991.
- [13] W. B. Mikhael and S. M. Ghosh. Two-Dimensional Variable Step-Size Sequential Adaptive Gradient Algorithms with Applications. *IEEE Trans. on Circuits and*

- Systems*, Vol. 38, No. 12, pp. 1577–1580, 1991.
- [14] W. K. Jenkins, J. C. Strait, and R. P. Faust. Two-Dimensional Adaptive Digital Filters with Reduced Computational Complexity. *Computers Elect. Eng.*, Vol. 18, No. 3/4, pp. 227–246, 1992.
- [15] A. C. Tan and S. T. Chen. Two-Dimensional Adaptive LMS IIR Filter. In *Proc. IEEE International Symposium on Circuits and Systems*, Vol. 3, pp. 299–302, Chicago, Illinois, May 1993.
- [16] 棟安実治, 雛元孝夫. LMS法を用いた2次元適応状態空間フィルタ. 電気情報通信学会論文誌, Vol. J76-A, No. 4, pp. 598–604, 1993.
- [17] J. J. Shynk. Adaptive IIR Filtering. *IEEE Acoust. Speech Signal Processing Magazine*, pp. 4–21, April 1989.
- [18] C. R. Johnson Jr. Adaptive IIR filtering; Current results and open issues. *IEEE Trans. on information theory*, Vol. IT-30, No. 2, pp. 237–250, March 1984.
- [19] D. A. Jhons, W. M. Snelgrove, and A. S. Sedra. SHARF: An algorithm for adaptive IIR digital filters. *IEEE trans. on acoust. Speech and Signal Processing*, Vol. ASSP-28, pp. 428–440, August 1980.
- [20] 久保田一. 適応アルゴリズム [I]~[IV]. 電子情報通信学会誌, Vol. 74, No. 5-8, 1991.
- [21] P. L. Feintuch. An Adaptive Recursive LMS Filter. *Proceedings of the IEEE*, Vol. 64, pp. 1622–1624, Nov. 1976.
- [22] S. Karni and G. Zeng. A New Convergence Factor for Adaptive Filters. *IEEE Trans. on Circuits and Systems*, Vol. 36, No. 7, pp. 1011–1012, 1989.
- [23] R. H. Kwong and E. W. Johnston. A Variable Step Size LMS Algorithm. *IEEE Trans. on Signal Processing*, Vol. 40, No. 7, pp. 1633–1642, 1992.
- [24] G. Long, F. Ling, and J. G. Proakis. The LMS Algorithm with Delayed Coefficient Adaptation. *IEEE trans. on acoust. Speech and Signal Processing*, Vol. 37, No. 9, pp. 1397–1405, 1989.
- [25] M. D. Meyer and D. P. Agawal. Vectorization of DLMS Transversal Adaptive Filter. *IEEE Trans. on Signal Processing*, Vol. 42, No. 11, pp. 3237–3240, 1994.
- [26] N. R. Shanbhag and K. K. Parhi. *Pipelined Adaptive Digital Filters*. Kluwer Academic Publishers, Boston, 1993.

-
- [27] 野田淳彦, 南雲仁一. システムの学習的同定法. 計測と制御, Vol. 7, No. 9, pp. 597–605, 1968.
- [28] A. M. Sequeira and C. W. Therrien. A New 2-D Fast RLS Algorithm. In *IEEE Proc. ICASSP'90*, pp. 1401–1404, Albuquerque, 1990.
- [29] 棟安実治, 上本栄治, 雛元孝夫. 1次元RLSアルゴリズムに基づく2次元適応フィルタ. 電子情報通信学会論文誌, Vol. J78-A, No. 10, pp. 1298–1305, 1995.
- [30] 酒井英昭. 最近の適応アルゴリズムの動向, RLS法を中心として. 日本音響学会誌, Vol. 48, No. 7, pp. 493–500, 1992.
- [31] 雛元孝夫, 前川禎男. 拡張された学習的同定法. 電気学会論文誌C, Vol. 95, No. 10, pp. 227–234, 1975.
- [32] 尾関和彦, 梅田哲夫. アフィン部分空間への直交射影を用いた適応フィルタ・アルゴリズムとその諸性質. 電子情報通信学会論文誌, Vol. J67-A, No. 2, pp. 126–132, 1984.
- [33] 藤井哲郎, 島田正治. 多チャンネル適応デジタルフィルタ. 電子情報通信学会論文誌, Vol. J69-A, No. 10, pp. 1226–1233, 1986.
- [34] S. L. Gay and S. Tavathia. The fast projection algorithm. In *Proc. ICASSP*, Vol. 2, pp. 945–948, 1995.
- [35] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima. A Fast Projection Algorithm for Adaptive Filtering. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E78-A, No. 10, pp. 1355–1361, 1995.
- [36] 橋本伊織, 長谷部伸治, 加納学. プロセス制御工学. 朝倉書店, 2002.
- [37] 大松繁, 山本透. セルフチューニングコントロール. 計測自動制御学会学術図書, 1996.
- [38] K. J. Åström, U. Borisson, L. Ljung, and B. Wittenmark. Theory and Applications of Self-Tuning Regulators. *Automatica*, Vol. 13, No. 5, pp. 457–476, 1977.
- [39] D. W. Clarke and P. J. Gawthrop. Self-tuning controller. *Proc. IEE*, Vol. 122D, pp. 929–934, 1975.
- [40] D. W. Clarke and P. J. Gawthrop. Self-tuning control. *Proc. IEE*, Vol. 126D, No. 6, pp. 633–640, 1979.
- [41] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized Predictive Control – Part I. The Basic Algorithm. *Automatica*, Vol. 23, No. 2, pp. 137–148, 1987.
- [42] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized Predictive Control – Part

- II. Extensions and Interpretations. *Automatica*, Vol. 23, No. 2, pp. 149–160, 1987.
- [43] D. W. Clarke and C. Mohtadi. Properties of Generalized Predictive Control. *Automatica*, Vol. 25, No. 6, pp. 859–875, 1989.
- [44] 増田士朗, 山本透, 大嶋正裕. モデル予測制御 III 一般化予測制御 (GPC) とその周辺. システム/制御/情報, Vol. 46, No. 9, pp. 578–584, 2002.
- [45] 森泰親, 秋元博之, 中内靖, 入道真. 一般化予測制御の応答特性改善のための一手法. 計測自動制御学会論文集, Vol. 31, No. 12, pp. 2024–2026, 1995.
- [46] K. S. Narendra and K. Parthasarathy. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Trans. on Neural Networks*, Vol. 11, No. 4, p. 27, 1990.
- [47] S. Haykin. *Neural Networks*. Macmillian College Publishing, New York, 1994.
- [48] 辻純一, 大森浩充, 佐野昭. ニューラルネットを併用した適応制御. 計測自動制御学会論文集, Vol. 30, No. 3, pp. 295–302, 1994.
- [49] 平嶋洋一, 飯国洋二, 足立紀彦. ニューラルネットワークを用いた適応制御系の一構成法. システム制御情報学会論文誌, Vol. 7, No. 8, pp. 281–286, 1994.
- [50] 徐炳鴻, 辻敏夫, 金子真. ニューラルネットを利用したモデル化誤差を含むプラントの同定と制御. 計測自動制御学会論文集, Vol. 32, No. 4, pp. 510–516, 1996.
- [51] 美口純一, 呉漢生, 水口孝一. ニューラルネットを用いたモデル規範型適応制御の一構成法. 計測自動制御学会論文集, Vol. 33, No. 6, pp. 477–482, 1997.
- [52] 沖俊任, 山本透, 兼田雅弘, 清水顯. ニューラルネットワークを用いた非線形システムに対する一般化最小分散制御系の一設計. 電気学会論文誌, Vol. 119C, No. 2, pp. 218–224, 1999.
- [53] 大西義浩, 山本透, 大松繁. ニューラルネットワークを用いた非線形システムの学習制御. 電気学会論文誌, Vol. 121C, No. 11, pp. 1683–1689, 2001.
- [54] J. S. Albus. A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). *Trans. ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 97, No. 9, pp. 220–227, 1975.
- [55] J. S. Albus. Data Storage in the Cerebellar Model Articulation Controller (CMAC). *Trans. ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 97, No. 9, pp. 228–233, 1975.

- [56] 伊賀清, 飯国洋二. 階層型メモリをもつ CMAC の設計法. 計測自動制御学会論文集, Vol. 28, No. 10, pp. 1224–1230, 1992.
- [57] T. Yamamoto and M. Kaneda. Intelligent Controller Using CMACs with Self-Organized Structure and Its Application for a Process System. *IEICE Trans. on Fundamentals*, Vol. E82-A, No. 5, pp. 856–860, 1999.
- [58] 藤沢正一郎, 黒住亮太, 大西諒, 川田和男, 山本透. CMAC を用いたオンライン学習による全方向移動ロボットの経路追従制御. 電気学会論文誌, Vol. 122D, No. 9, pp. 910–917, 2002.
- [59] 高尾健司, 加納寛子, 山本透, 雛元孝夫. CMAC と階層型ニューラルネットワークを併用したインテリジェント制御系の一設計. 計測自動制御学会論文集, Vol. 40, No. 1, pp. 62–69, 2004.
- [60] 須田信英. PID 制御. 朝倉出版, 1992.
- [61] J. G. Ziegler and N. B. Nicols. Optimum Setting for Automatic Controllers. *Trans. ASME*, Vol. 64, No. 2, pp. 759–768, 1942.
- [62] K. L. Chien, J. A. Hrons, and J. B. Reswick. On the Automatic Control of Generalized Passive Systems. *Trans. ASME*, Vol. 74, No. 1, pp. 175–185, 1952.
- [63] 北森俊行. 制御対象の部分的知識に基づく制御系の設計法. 計測自動制御学会論文集, Vol. 15, No. 4, pp. 549–555, 1979.
- [64] 北森俊行, 竹内信明, 西崎晶. むだ時間をループ外に追い出した PID および I-PD 制御系の設計法. 第 35 回計測自動制御学会学術講演会予稿集, pp. 107–108, 1996.
- [65] 山本透, 兼田雅弘. 一般化最小分散制御則に基づくセルフチューニング PID 制御器の一設計. システム制御情報学会論文誌, Vol. 11, No. 1, pp. 1–9, 1998.
- [66] 山本透, S.L.Shah. 多重ループ型セルフチューニング PID 制御系の一設計. システム制御情報学会論文誌, Vol. 11, No. 4, pp. 163–171, 1998.
- [67] 山本透. セルフチューニング方式の新展開—一般化最小分散制御と PID 制御—. 計測と制御, Vol. 40, No. 10, pp. 704–710, 2001.
- [68] M. Asano and T. Yamamoto. A Design of Self-Tuning Predictive PID Controllers. *IEICE Trans. Fundamentals*, Vol. E84-A, No. 7, pp. 1779–1783, 2001.
- [69] 片山優, 山本透, 間田泰弘, 藤井憲三. 多重ループ型ロバスト PID 制御系の一設計. 電気学会産業計測制御研究会資料, ☒ C-01, No. 51, pp. 7–12, 2001.

- [70] M. Katayama, T. Yamamoto, and Y. Mada. A Design of Multiloop Predictive Self-Tuning PID Controllers. *Asian Journal of Control*, Vol. 4, No. 4, pp. 472–481, 2002.
- [71] 片山優, 山本透, 間田泰弘. 多重ループ型ロバスト PID 制御系の実用的設計. 電気学会論文誌, Vol. 123C, No. 2, pp. 330–337, 2003.
- [72] 青山武郎, 大松繁. ニューラルネットワークによるセルフチューニング PID 制御系の設計. 電気学会論文誌, Vol. 116C, No. 11, pp. 1197–1201, 1996.
- [73] 山本透, 沖俊任, 兼田雅弘. ニューラルネットワークを併用したセルフチューニング PID 制御系の一設計. 計測自動制御学会論文集, Vol. 34, No. 7, pp. 682–688, 1998.
- [74] 黒住亮太, 山本透. 小脳演算モデルを用いたインテリジェント PID 制御系の一設計. 電気学会論文誌, Vol. 125C, No. 4, pp. 607–615, 2005.
- [75] 山本透, 満倉靖恵, 兼田雅弘. 遺伝的アルゴリズムを用いた PID 制御器の一設計. 計測自動制御学会論文集, Vol. 35, No. 4, pp. 531–537, 1999.
- [76] 満倉靖恵, 山本透, 兼田雅弘. GA を併用したセルフチューニング PID 制御系の一設計. 計測自動制御学会論文集, Vol. 36, No. 1, pp. 75–81, 2000.
- [77] 太田純, 山本茂. PID 制御器のデータベース駆動型チューニング. 計測自動制御学会論文集, Vol. 40, No. 6, pp. 664–669, 2004.
- [78] 高尾健司, 山本透, 雛元孝夫. Memory-Based 型 PID コントローラの設計. 計測自動制御学会論文集, Vol. 40, No. 9, pp. 898–905, 2004.
- [79] 高尾健司, 山本透, 雛元孝夫. Memory-Based 型システム同定による一般化予測制御系の一設計. 電気学会論文誌, Vol. 125C, No. 3, pp. 442–449, 2005.
- [80] T. J. Harris. Assessment of Control Loop Performance. *Canadian Journal of Chemical Engineering*, Vol. 67, No. 5, pp. 856–861, 1989.
- [81] L. Desborough and T. Harris. Performance Assessment Measures for Univariate Feedback Control. *Canadian Journal of Chemical Engineering*, Vol. 70, No. 6, pp. 1186–1197, 1992.
- [82] L. Desborough and T. Harris. Performance Assessment Measures for Univariate Feedforward/Feedback Control. *Canadian Journal of Chemical Engineering*, Vol. 71, No. 4, pp. 605–616, 1993.
- [83] C. B. Lynch and G. A. Dumont. Control loop performance monitoring. *IEEE*

-
- Trans. Control Systems Technology*, Vol. 4, No. 2, pp. 185–192, 1996.
- [84] B. Huang, S. L. Shah, and K. E. Kwok. Good, bad or optimal? Performance Assessment of Multivariable Processes. *Automatica*, Vol. 33, No. 6, pp. 1175–1183, 1997.
- [85] A. McNabb and S. J. Qin. Projection based MIMO control performance monitoring: I-covariance monitoring in state space. *Journal of Process Control*, Vol. 13, No. 8, pp. 739–757, 2003.
- [86] S. J. Qin. Control performance monitoring - a review and assessment. *Computers & Chemical Engineering*, Vol. 23, No. 2, pp. 173–186, 1998.
- [87] T. J. Harris, C. T. Seppala, and L. D. Desborough. A Review of Performance Monitoring and Assessment Techniques for Univariate and Multivariate Control Systems. *Journal of Process Control*, Vol. 9, No. 1, pp. 1–17, 1999.
- [88] 加納学, 山下善之. プロセス制御系の制御性能評価と監視. 計測と制御, Vol. 44, No. 2, pp. 125–129, 2005.
- [89] M. Jelali. An overview of control performance assessment technology and industrial applications. *Control Engineering Practice*, Vol. 14, No. 5, pp. 441–466, 2006.
- [90] B. Huang and S. L. Shah. *Performance Assessment of Control Loops : Theory and Applications*. Springer-Verlag, London, 1999.
- [91] B. Huang. A pragmatic approach towards assessment of control loop performance. *Int. J. Adapt. Control Signal Process.*, Vol. 17, No. 7-9, pp. 589–608, 2003.
- [92] 高尾健司, 大西義浩, 山本透, 雛元孝夫. 制御性能評価に基づくパフォーマンス・アダプティブ PID コントローラ的设计. 計測自動制御学会論文集, Vol. 43, No. 2, pp. 110–117, 2007.
- [93] 山本透. モデリング性能評価に基づくパフォーマンス・アダプティブ PID 制御系的设计. 電気学会論文誌, Vol. 127C, No. 12, pp. 2101–2108, 2007.
- [94] T. Kono, T. Yamamoto, and T. Hinamoto. Design of a Data-Driven Performance-Adaptive PID Controller. In *Proc. Second International Conference on Innovative Computing, Information and Control*, p. 430, 2007.
- [95] 児玉慎三, 須田信英. システム制御のためのマトリクス理論. 計測自動制御学会学術図

- 書, S53.
- [96] 八木秀幸, 波多野崇, 下西二郎, 山本透, 雛元孝夫. モデル予測型 PID 制御系の制御性能に関する一考察. 計測自動制御学会第 7 回制御部門大会講演論文集, No. 72-2-5, 東京, 2007.
- [97] R. Wood, S. Shah, and R. Miller. Adaptive Predictive Control Employing On-Line Time Delay Estimation. *Advances in Instrumentation and Control*, pp. 857–864, 1995.
- [98] L. Zi-Qiang. On identification of the controlled plants described by the Hammerstein system. *IEEE Trans. Automatic Control*, Vol. AC-39, pp. 569–573, 1994.
- [99] 八木秀幸, 下西二郎, 山本透, 雛元孝夫. モデル予測機能を有するパフォーマンス駆動型 PID 制御系の設計. 計測自動制御学会論文集, Vol. 44, No. 10, pp. 809–818, 2008.
- [100] R. R. Mohler and W. J. Kalodziej. An overview of stochastic bilinear control process. *IEEE Trans. Systems Man Cybernetics*, Vol. SMC-10, No. 12, pp. 913–918, 1980.

引用既発表文献

学術雑誌等掲載論文

1. Mitsuji Muneyasu, Hideyuki Yagi and Takao Hinamoto. "A Realization of 2D Adaptive Filters using Affine Projection Algorithm". *Journal of the Franklin Institute*, Vol. 335B, No. 7, pp. 1185-1193, 1998.
2. 八木秀幸, 下西二郎, 山本透, 雛元孝夫. "モデル予測機能を有するパフォーマンス駆動型 PID 制御系の一設計". 計測自動制御学会論文集, Vol. 44, No. 10, pp. 809-818, 2008.
3. Hideyuki Yagi, Jiro Shimonishi, Toru Yamamoto and Takao Hinamoto. "Design of Performance-Adaptive PID Controller with a Model Prediction Structure", *International Journal of Innovative Computing, Information and Control* (in press).
4. 八木秀幸, 下西二郎, 山本透, 雛元孝夫. "CMAC を用いたパフォーマンス駆動型 PID 制御系の設計". 電気学会論文誌 C (照会后判定).

国際会議講演論文

1. Mitsuji Muneyasu, Hideyuki Yagi and Takao Hinamoto. "A Realization of 2D Adaptive Filters using Affine Projection Algorithm". *IEEE International Conference on Communication Systems and IEEE International Workshop on Intelligent Signal Processing & Communication Systems*, Vol. 2, pp. 1068-1072, Singapore, November, 1996.
2. Mitsuji Muneyasu, Hideyuki Yagi and Takao Hinamoto. "A Realization of Cascade Adaptive IIR Filters based on the RLS Algorithm". *International Technical Conference on Circuits/Systems, Computers and Communications*, Vol. 1,

- pp. 299-302, Naha, Okinawa, July, 1997.
3. Mitsuji Muneyasu, Hideyuki Yagi and Takao Hinamoto. "A Novel Type of 2-D Adaptive Filters Using Affine Projection Algorithm". *International Technical Conference on Circuits/Systems, Computers and Communications*, Vol. 1, pp. 283-286, Naha, Okinawa, July, 1997.
 4. Hideyuki Yagi, Fumio Nishizaki, Mitsuji Muneyasu, Jiro Shimonishi and Takao Hinamoto. "An Adaptive FIR Filter Based on Delayed NLMS Algorithm and Its Analysis". *International Technical Conference on Circuits/Systems, Computers and Communications*, Vol. 2, pp. 1023-1026, Sado, Niigata, 1999.
 5. Hideyuki Yagi, Jiro Shimonishi, Toru Yamamoto and Takao Hinamoto. "Design of a Performance-Driven Model Predictive PID Controller". *IEEE International Symposium on Industrial Electronics*, pp. 1243-1248, Cambridge UK, July, 2008.
 6. Hideyuki Yagi, Jiro Shimonishi, Toru Yamamoto and Takao Hinamoto. "Design of a Performance-Driven CMAC-PID Controller". *European Control Conference*, 2009 (投稿中) .