

博士学位論文

氏名（本籍）	森田 昌一（東京都）
学位の種類	博士（工学）
学位記番号	博甲第 173 号
学位授与年月日	令和 4 年 3 月 31 日
学位授与の要件	学位規則第 4 条第 1 項
学位論文題目	ニューラルネットワーク制御の オンライン・オフライン融合型学習法を用いた サーボシステムの位置決め制御

論文審査委員	主査	森下 明平
	副査	黄 慶九
	〃	福岡 豊
	〃	斎藤 秀俊
	〃	濱根 洋人
	〃	劉 康志（千葉大学）
	〃	

ニューラルネットワーク制御の
オンライン・オフライン融合型学習法を用いた
サーボシステムの位置決め制御
Position Control of Servo System Using Online-Offline
Integrated Learning Method of Neural Network Control

2022年2月17日

工学院大学大学院 工学研究科 電気・電子工学専攻

森田 昌一

目次

第1章 序論	6
1.1 研究の背景.....	6
1.2 研究の目的.....	7
1.3 論文の構成.....	8
第2章 サーボシステムの基礎知識	9
2.1 はしがき.....	9
2.2 永久磁石同期電動機 (PMSM) ^{[36] [37]}	9
2.2.1 表面磁石型同期電動機 (SPMSM) の構造 ^[37]	10
2.2.2 埋込磁石型同期電動機 (IPMSM) の構造 ^[37]	10
2.3 サーボモータの制御 ^[37]	11
2.3.1 表面磁石型同期電動機 (SPMSM) の制御 ^[37]	15
2.3.2 埋込磁石型同期電動機 (IPMSM) の制御 ^[37]	15
2.4 サーボシステムの構成 ^{[38] - [42]}	20
2.4.1 PID 制御を用いたフィードバック制御 ^{[38] [39]}	20
2.4.2 極配置法を用いた PID ゲインの設計 ^{[38] [40]}	23
2.4.3 現代制御理論によるサーボシステムの表現 ^{[41] [42]}	27
2.5 あとがき.....	30
第3章 ニューラルネットワーク制御の基礎知識	31
3.1 はしがき.....	31
3.2 ニューラルネットワークの仕組み.....	31
3.2.1 ニューラルネットワークの構造.....	32
3.2.2 誤差逆伝播を用いた最急降下法.....	33
3.3 ニューラルネットワーク制御の仕組み.....	37
3.3.1 ニューロコントロールと適応制御 ^[5]	38
3.3.2 フィードバック誤差学習による閉ループシステムの学習制御 ^[33]	43
3.4 あとがき.....	49

第4章 動的モデルを用いた開発環境	50
4.1 はしがき.....	50
4.2 サーボシステムの動的モデルと構成.....	50
4.3 シミュレーションのモデル.....	52
4.4 あとがき.....	52
第5章 オンライン・オフライン融合型学習法	53
5.1 はしがき.....	53
5.2 学習法のアルゴリズム.....	53
5.2.1 オンライン学習法のアルゴリズム.....	53
5.2.2 オフライン学習法のアルゴリズム.....	54
5.2.3 オンライン・オフライン融合型学習法のアルゴリズム.....	55
5.3 学習法のまとめ.....	57
5.4 結果と考察.....	57
5.4.1 オンライン学習法を用いた学習済重みの獲得.....	58
5.4.2 負荷イナーシャ変更前後の応答性とロバスト性.....	59
5.4.3 目標軌道の周波数変更前後の応答性とロバスト性.....	64
5.4.4 応答性とロバスト性に関する考察.....	65
5.5 あとがき.....	67
第6章 学習率と回数閾値の更新則	68
6.1 はしがき.....	68
6.2 学習率と回数閾値の新しい更新則の提案.....	68
6.3 学習率の増減に関わるサーボシステムの特性.....	68
6.3.1 学習率の増減に関わる定常特性.....	69
6.3.2 学習率の増減に関わる過渡特性.....	71
6.4 最小二乗法に基づく学習率、回数閾値の定式化および更新則の導出.....	73
6.4.1 学習率、回数閾値と誤差最大値との定式化.....	73
6.4.2 学習率、回数閾値と整定時間との定式化.....	75
6.4.3 学習率と回数閾値の更新則の導出.....	77

6.5	学習率、回数閾値の更新則に基づくサーボシステムの解析結果.....	79
6.5.1	解析結果.....	79
6.5.2	サーボシステムの定常特性に関する考察.....	81
6.5.3	サーボシステムの過渡特性に関する考察.....	81
6.6	あとがき.....	82
第7章 カスケードニューラルネットワーク制御.....		83
7.1	はしがき.....	83
7.2	オンライン・オフライン融合型学習法と併用する新しい制御法の提案.....	83
7.3	カスケードニューラルネットワーク制御の理論.....	84
7.3.1	カスケードニューラルネットワーク制御の構成.....	84
7.3.2	カスケードニューラルネットワーク制御のモデル.....	87
7.4	結果と考察.....	91
7.4.1	負荷イナーシャ変更前後の応答性とロバスト性.....	91
7.4.2	電機子巻線抵抗の変更前後の応答性とロバスト性.....	94
7.4.3	周期外乱印加後の応答性とロバスト性.....	95
7.4.4	応答性とロバスト性に関する考察.....	96
7.5	あとがき.....	97
第8章 結論.....		98
謝辞.....		99
参考文献.....		100

第1章 序論

1.1 研究の背景

ニューラルネットワーク制御は、適応制御や非線形制御の1種である [1] - [6]。加えて、ニューラルネットワーク制御は、学習を通して制御対象の動特性を獲得できるため、ロボットや数値制御工作機械などのサーボシステムの高精度な位置決め制御に期待される [7] - [17]。

今まで、ニューラルネットワーク制御は2つの学習法があり、これは、オンライン学習とオフライン学習である。本研究でのサーボモータの位置決め制御のシミュレーション結果と考察から分かったこととして、オンライン学習法は、フィードバック誤差に基づいて学習を常に行われ、負荷変動と外乱に対応でき、ロバスト性が優れている。ただし、整定時間が長く、実時間制御に向いていない。オフライン学習法は、十分なオンライン学習で獲得したニューラルネットワークをフィードフォワード補償器として使い、膨大な計算量が無くなったため、制御系の応答性が速い。しかし、フィードバック誤差に基づき常に学習を行わないため、負荷変動と外乱に対応できない。そのため、本研究ではオンライン学習法のロバスト性が優れている利点とオフライン学習法の応答性が良い利点を融合させ、オンライン・オフライン融合型学習法を発想してきた。

次に、この学習法のパラメータである学習率とサンプリングタイム内の繰り返しループ回数の閾値（回数閾値）について、学習率、回数閾値とサーボシステムの応答性、ロバスト性との関係がまだ明らかではない。また、提案した学習法の学習率と回数閾値のチューニングにおいて、手間がかからないことおよび学習の効率を向上するための検討が必要である。

ニューラルネットワークの学習率の更新則についての研究では、神田らが Proportional-Integral-Differential (PID) 制御と予測を使って学習率を決定する方法を提案していた [18] [19]。また、Cater は誤差評価関数が大きく、その勾配が小さい場合、学習率を増加させる Heuristic Learning Algorithm (HLA) アルゴリズムを提案していた [20]。Vogl らは誤差評価関数の変化によって学習率を増減させる方法を提案していた [21] [22]。さらに、Jacobs 法に基づき評価関数の曲面の谷で発生する重みの振動を抑制する Kick Out 法がある [23] - [25]。加えて、Jacobs 法の勾配項のみを持つ重みの更新則 Delta-Bar-Delta (DBD) もある [26] [27]。これらの更新則は、学習率増減の微分を平滑化するような方法であるため、重み増減の振動を抑制することができる。しかし、学習率の増減について学習の効率と学習の極小値との関係がまだ定式化されていないため、サーボシステムの応答性、ロバスト性との関係を結ぶことは難しい。

そして、提案したオンライン・オフライン融合型学習法に基づき、負荷変動や外乱による制御対象の影響をさらに抑制し、位置決め精度を一層高めるため、制御対象への制御入力とフィードバック誤差との逆モデルを新たなニューラルネットワーク補償器の学習で獲得するような新しいニューラルネットワーク制御の検討が必要である。

1.2 研究の目的

本研究では、サーボシステムの位置決め制御に対して、負荷変動と外乱の対応、実時間制御の実現を達成するため、フィードバック誤差が閾値を超えなければオフライン学習を行いフィードバック誤差が閾値を超えればサンプリングタイム内で少ない繰り返しループのオンライン学習を行うようなオンライン・オフライン融合型学習法を提案する [28] - [30]。

具体的には、まず、十分なオンライン学習で獲得したニューラルネットワークの重みを基準重みとしてロードし、ニューラルネットワーク制御はオフライン学習法で実施される。次に、フィードバック誤差が閾値を超えなければ、オフライン学習法が行われる。フィードバック誤差が閾値を超えれば、サンプリングタイム内で少ない繰り返しループのオンライン学習法が行われる。フィードバック誤差が閾値まで収束したら、オフライン学習法に戻る。これらの手順によって、提案法はオフライン学習法の良い応答性とオンライン学習法の優れているロバスト性を融合させられ、負荷変動と外乱の対応および実時間制御が可能であるニューラルネットワーク制御を実現できる。さらに、フィードバック誤差学習 [31] - [34] に基づくニューラルネットワーク制御の構成を利用して、DC サーボモータの位置決め制御のシミュレーションを行う。このシミュレーションでは、DC サーボモータの負荷イナーシャの変更前後に対して、ニューラルネットワーク制御にオンライン学習法、オフライン学習法、提案法をそれぞれ適用して、シミュレーションの結果と考察から提案法の有効性を明らかにする。

加えて、本研究では、サーボシステムの位置決め制御に対して、学習率、回数閾値とサーボシステムの応答性、ロバスト性との関係をシミュレーションとその解析データから明らかにする。そして、最小二乗法によってそれらの関係の定式化を行い、セルフチューニングが行えるような学習率と回数閾値の更新則を提案する [35]。

具体的には、まず、サーボモータの位置決め制御の動的モデルを用いた解析環境を構築して、負荷イナーシャ変更前後において、オンライン・オフライン融合型学習法を行い、学習率、回数閾値とサーボシステムの応答性、ロバスト性との関係をシミュレーションとその解析データから明らかにする。そして、学習率、回数閾値、フィードバック誤差、整定時間との関係の定式化を最小二乗法で行い、セルフチューニングが行えるような学習率と回数閾値の更新則を提案する。最後に、フィードバック誤差の指標値と整定時間の指標値を設けて、提案された学習率と回数閾値の更新則により学習率と回数閾値を求めて、負荷イナーシャ変更前後のシミュレーションとその結果から更新則の有効性を示す。

さらに、本研究では、オンライン・オフライン融合型学習法を用いたニューラルネットワーク補償器と並列に別のニューラルネットワーク補償器を接続させるようなカスケード型ニューラルネットワーク制御を提案する。具体的には、新しいニューラルネットワークの入力層にフィードバック誤差を入力させて、オンライン学習法を行い、そのニューラルネットワーク補償器の出力と単体のニューラルネットワーク補償器の出力を制御対象に加算する。これらの手順によって、負荷変動や外乱による影響が打ち消され、補償器の効果を総合的に向上させられる。そして、DC サーボモータの位置決め制御のシミュレーション結果と考察から提案法の有効性を示す。

1.3 論文の構成

本論文の構成は、以下の通りである。

第1章では、ニューラルネットワーク制御の新しい学習法について、提案したその背景と目的を述べた。次に、提案した学習法のパラメータ設定の課題点およびその解決法を述べた。その次に、単体のニューラルネットワーク制御を活用した新しいニューラルネットワーク制御について、提案したその背景と目的を述べた。

第2章では、永久磁石同期電動機（PMSM：Permanent Magnet Synchronous Motor）の種類にて、表面磁石型同期電動機（SPMSM：Surface Permanent Magnet Synchronous Motor）と埋込磁石型同期電動機（IPMSM：Interior Permanent Magnet Synchronous Motor）を例とするサーボモータの構造とそれらの制御法を説明する。また、サーボシステムの構成も説明する。

第3章では、ニューラルネットワークとはどのような構成であるか、どのようなアルゴリズムが使われているか、学習のアルゴリズムの1種である誤差逆伝播を用いた最急降下法を説明する。また、ニューラルネットワーク制御の先行研究も説明する。

第4章では、制御対象であるサーボモータの位置決め制御の構成を説明して、その制御系の動的モデルを示す。加えて、サーボモータの位置決め制御のシミュレーションを行うため、MATLAB/Simulinkにより作成した制御系の解析モデルを示して、そのモデルを用いたシミュレーションの具体的な解析方法を説明する。

第5章では、まず、従来のオンライン学習法およびオフライン学習法のアルゴリズムについてそれぞれの利点と欠点を説明して、オンライン学習法の利点およびオフライン学習法の利点を融合させたオンライン・オフライン融合型学習法のアルゴリズムについて説明する。次に、前章で説明したサーボモータの位置決め制御のシミュレーションを行い、PID制御とオンライン学習法をそれぞれ用いたシミュレーションの結果についての比較、オンライン学習法、オフライン学習法、提案したオンライン・オフライン融合型学習法のシミュレーション結果について比較を行う。最後に、それらの結果から提案した学習法で達成できたことを述べる。

第6章では、提案したオンライン・オフライン融合型学習法のパラメータ設定の課題に対してセルフチューニングが行えるような学習率と回数閾値の更新則を提案する。また、その更新則を導出するまでの手順を説明して、導出された更新則に基づいて求められた学習率と回数閾値をオンライン・オフライン融合型学習法に代入する。そして、位置決め制御のシミュレーションを行い、その結果から提案した更新則の有効性を述べる。

第7章では、提案したオンライン・オフライン融合型学習法を用いたニューラルネットワーク補償器ともう1つ別のニューラルネットワーク補償器を併用したカスケード型ニューラルネットワーク制御について、その構成や理論を説明する。そして、位置決め制御のシミュレーションを行い、その結果から提案した制御法の有効性を述べる。

第8章では、本論文の結論を述べる。

第2章 サーボシステムの基礎知識

2.1 はしがき

本章では、第3章で述べる制御対象とニューラルネットワークとの間の関係を明らかにする前段階として、まず、永久磁石同期電動機（PMSM）の種類を説明する。種類は、回転子の表面に永久磁石を張り付ける表面磁石型同期電動機（SPMSM）と回転子の内部に永久磁石を埋め込む埋込磁石型同期電動機（IPMSM）の2種類であり、それらの構造を2.2節で説明する。次に、それらの制御方法を2.3節で説明する。最後に、フィードバック制御を用いた全体のサーボシステムの構成について、PID制御のゲイン設定や現代制御理論を用いた表現を2.4節で説明する。

2.2 永久磁石同期電動機（PMSM）^{[36] [37]}

本節では、赤津（2012）および武田、松井、森本、本田（2001）に基づき説明する。永久磁石を用いた電動機には、直流（DC：Direct Current）を電源にする直流モータと交流（AC：Alternating Current）を電源にする交流モータがある。DCモータでは、フレミング左手の法則によって電機子巻線で発生する電磁力、図2.1に示すようなブラシと整流子との接触によりモータの回転が連続して行われる。ACモータでは、磁界が回転することでモータの回転が連続して行われる。回転磁界形モータは単相交流モータと三相交流モータに分類され、それらのモータの中で、単相誘導モータと三相誘導モータ、単相同期モータと三相同期モータに分類される。そして、パルス波の入力によってモータの回転が行われるブラシレスモータとステッピングモータがある。加えて、ブラシレスモータでは、矩形波入力で駆動するブラシレスDC（BLDC）モータと正弦波入力で駆動するブラシレスAC（BLAC）モータがある。さらに、ブラシレスモータでは、固定子を外側に回転軸を備えた回転子を内側に備えたインナーロータ形のモータと固定子を内側に回転軸を備えた回転子を外側に備えたアウターロータ形のモータと呼ばれている。インナーロータ形とアウターロータ形は、固定子と回転子との間にギャップがあり、そのギャップの磁力線の方向が回転軸に対して垂直な方向になる。ここで、PMSMのロータの形状は図2.2に示される。

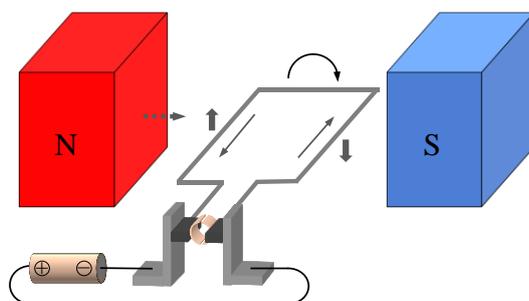


図2.1 DCモータの整流子とブラシ^[36]

赤津観：史上最強カラー図解 最新版 モータ技術のすべてがわかる本，ナツメ社，図B1-2-1・図B1-2-5（2012）を参考に作成

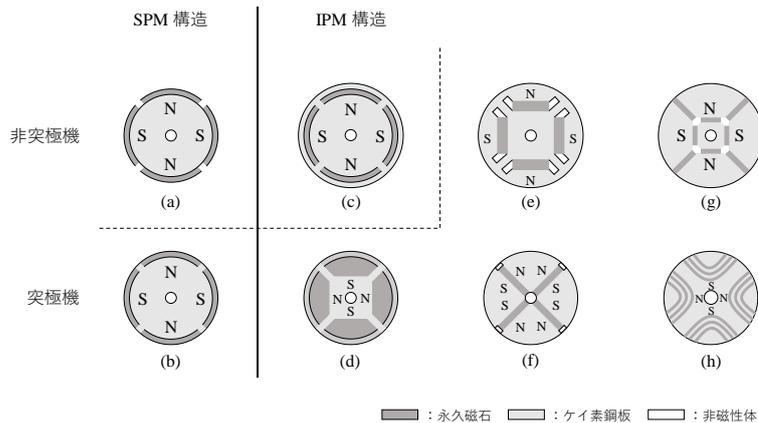


図 2.2 PMSM の代表的なロータ構造 [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 1.3 (2001) を一部改変

2.2.1 表面磁石型同期電動機 (SPMSM) の構造 [37]

図 2.2(a) と図 2.2(b) のように回転子表面に永久磁石を配置する PMSM は SPMSM に分類される。ここで、永久磁石の N 極方向に関するインダクタンスは d 軸 (Direct-axis) インダクタンスと呼ばれ、d 軸より電気角で 90 度進んだ方向に関するインダクタンスは q 軸 (Quadrature-axis) インダクタンスと呼ばれる。図 2.2(a) の SPMSM では磁気抵抗が回転子の位置に無関係になるため、d 軸インダクタンスと q 軸インダクタンスの値が等しい関係になる。図 2.2(b) は永久磁石がロータの表面に露出しているため SPMSM の部類であるが、磁石両端の鋼板がロータの表面に出ているため突極形に分類される。突極形は、図 2.2(b) 以外にも (d) から (h) が該当する。SPMSM の特徴は、インダクタンスの値が小さいため、高速回転時の端子電圧の上昇と磁気飽和による影響を受けにくいという利点がある。一方で、高速回転時に遠心力による永久磁石の飛散を防止するための保護管が必要になるという欠点や永久磁石の磁束と電機子巻線電流間に作用する電磁力により発生するマグネットトルク (Magnetic Torque) しか利用できないという欠点がある。

2.2.2 埋込磁石型同期電動機 (IPMSM) の構造 [37]

図 2.2(c) から図 2.2(h) のように回転子内部に永久磁石を埋め込む PMSM は IPMSM に分類される。ここで、図 2.2(c) は電流が少しでも大きくなると表面の薄い鋼板部分が磁気飽和を起こし非突極形 (円筒形) になる。図 2.2(e) の IPMSM では、d 軸方向の磁束が磁気抵抗によって通りにくく q 軸方向の磁束が通りやすいため、q 軸インダクタンスの値が d 軸インダクタンスよりも大きいという関係になる。この関係は突極性に対する逆突極性と呼ばれる。IPMSM の特徴は、永久磁石が内部に埋め込まれているため、保護管が不要という利点やマグネットトルクと併用して回転磁界と回転子の突極との吸引力だけによって生ずるリラクタンストルク (Reluctance Torque) も利用できるという利点がある。一方で、q 軸インダクタンスの値が大きいため、高速回転時の端子電圧の上昇と磁気飽和による影響を受けやすいという欠点がある。

2.3 サーボモータの制御 [37]

前節では、SPMSM と IPMSM の構造を説明した。SPMSM と IPMSM の電流やトルクなどを制御する方法としては、ベクトル制御と呼ばれる方法がある。本節では、武田、松井、森本、本田 (2001) に基づいて、そのベクトル制御の理論を説明する。ベクトル制御には、 $i_d = 0$ 制御、最大トルク/電流制御、最大トルク/磁束制御、弱め磁束制御、力率 1 制御、最大効率制御がある。IPMSM で $i_d = 0$ 制御を行う場合は、マグネットトルクのみが利用できて、BLDC モータの等価回路が扱える。まず、PMSM は図 2.3 に示されるような等価回路で表現できる。

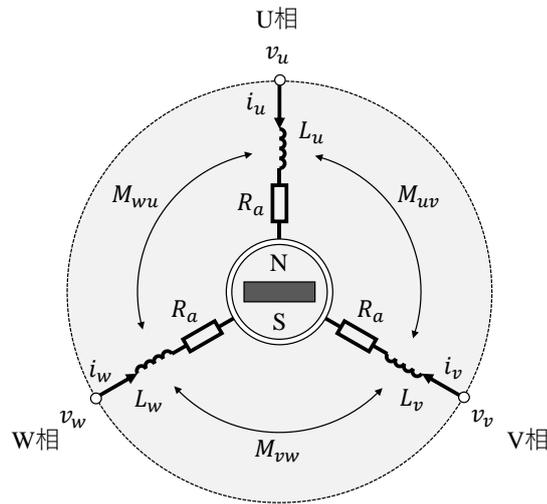


図 2.3 PMSM の三相座標モデル [37]

武田洋次，松井信行，森本茂雄，本田幸夫：埋込磁石同期モータの設計と制御，オーム社，図 1.5 (2001) を一部改変

このモデルの自己インダクタンス L_u, L_v, L_w と相互インダクタンス M_{uv}, M_{vw}, M_{wu} は式(2.1)と式(2.2)にそれぞれ示され、永久磁石の電機子鎖交磁束 $\Psi_{fu}, \Psi_{fv}, \Psi_{fw}$ は式(2.3)に示される。また、 l_a は 1 相分の漏れインダクタンス、 L_a は 1 相分のインダクタンスの平均値、 L_{as} は 1 相分のインダクタンスの振幅値、 θ_{re} は電気角、 Ψ_f は 1 相分の電機子鎖交磁束の最大値を示す。

$$\begin{aligned} L_u &= l_a + L_a - L_{as} \cos(2\theta_{re}) \\ L_v &= l_a + L_a - L_{as} \cos\left(2\theta_{re} + \frac{2}{3}\pi\right) \\ L_w &= l_a + L_a - L_{as} \cos\left(2\theta_{re} - \frac{2}{3}\pi\right) \end{aligned} \quad (2.1)$$

$$\begin{aligned} M_{uv} &= -\frac{1}{2}L_a - L_{as} \cos\left(2\theta_{re} - \frac{2}{3}\pi\right) \\ M_{vw} &= -\frac{1}{2}L_a - L_{as} \cos(2\theta_{re}) \\ M_{wu} &= -\frac{1}{2}L_a - L_{as} \cos\left(2\theta_{re} + \frac{2}{3}\pi\right) \end{aligned} \quad (2.2)$$

$$\begin{aligned}
\Psi_{fu} &= \Psi_f \cos(\theta_{re}) \\
\Psi_{fv} &= \Psi_f \cos\left(\theta_{re} - \frac{2}{3}\pi\right) \\
\Psi_{fw} &= \Psi_f \cos\left(\theta_{re} + \frac{2}{3}\pi\right)
\end{aligned} \tag{2.3}$$

次に、PMSMの三相電圧方程式は、式(2.1)から式(2.3)を用いて式(2.4)に示される。なお、 v_u, v_v, v_w は電機子電圧、 i_u, i_v, i_w は電機子電流、 R_a は電機子巻線抵抗、 p は時間微分の演算子を示す。

$$v_{uvw} = \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} = \begin{bmatrix} R_a + pL_u & pM_{uv} & pM_{wu} \\ pM_{uv} & R_a + pL_v & pM_{vw} \\ pM_{vu} & pM_{vw} & R_a + pL_w \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + p \begin{bmatrix} \Psi_{fu} \\ \Psi_{fv} \\ \Psi_{fw} \end{bmatrix} \tag{2.4}$$

ここで、三相座標モデルから二相座標モデルへ変換する。それは、式(2.5)と式(2.6)に示される。また、二相座標モデルは図 2.4 に示される。三相座標モデルから二相座標モデルへの変換は、 $\alpha - \beta$ 変換（クラーク変換）と呼ばれ、式(2.6)に示される変換行列 $C_{\alpha\beta}$ で変換が行われる。

$$v_{\alpha\beta} = \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = C_{\alpha\beta} \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} \tag{2.5}$$

$$i_{\alpha\beta} = \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = C_{\alpha\beta} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} \tag{2.6}$$

$$C_{\alpha\beta} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{2.7}$$

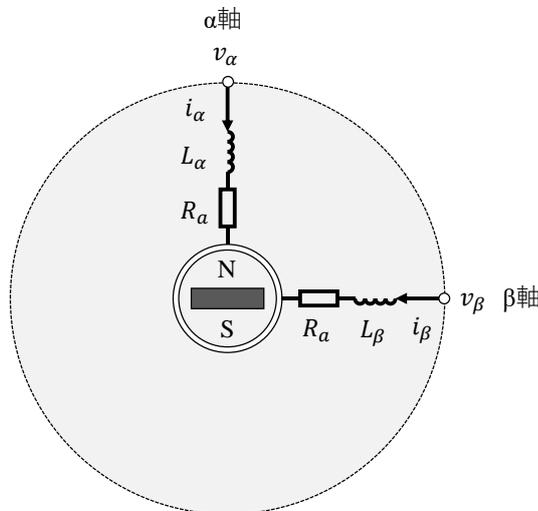


図 2.4 PMSM の二相座標モデル [37]

武田洋次，松井信行，森本茂雄，本田幸夫：埋込磁石同期モータの設計と制御，オーム社，図 1.6（2001）を参考に作成

式(2.7)において、変換行列 $C_{\alpha\beta}$ の係数 $\sqrt{2/3}$ を係数 K に置き換えて、式(2.8)の計算を行うと $K = \sqrt{2/3}$ を導出することができる。なお、 T は転置行列を示し、 I は単位行列を示す。また、係数 K は $2/3$ に置き換えることもできる。

$$C_{\alpha\beta}C_{\alpha\beta}^T = I \Leftrightarrow K^2 \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}^T = K^2 \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.8)$$

そして、二相座標モデルから回転座標モデルへ変換する。それは、式(2.9)と式(2.10)に示される。また、回転座標モデルは図 2.5 に示される。二相座標モデルから回転座標モデルへの変換は、 $d-q$ 変換（パーク変換）と呼ばれ、式(2.11)に示される変換行列 C_{dq} で変換が行われる。

$$v_{dq} = \begin{bmatrix} v_d \\ v_q \end{bmatrix} = C_{dq} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \quad (2.9)$$

$$i_{dq} = \begin{bmatrix} i_d \\ i_q \end{bmatrix} = C_{dq} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (2.10)$$

$$C_{dq} = \begin{bmatrix} \cos(\theta_{re}) & \sin(\theta_{re}) \\ -\sin(\theta_{re}) & \cos(\theta_{re}) \end{bmatrix} \quad (2.11)$$

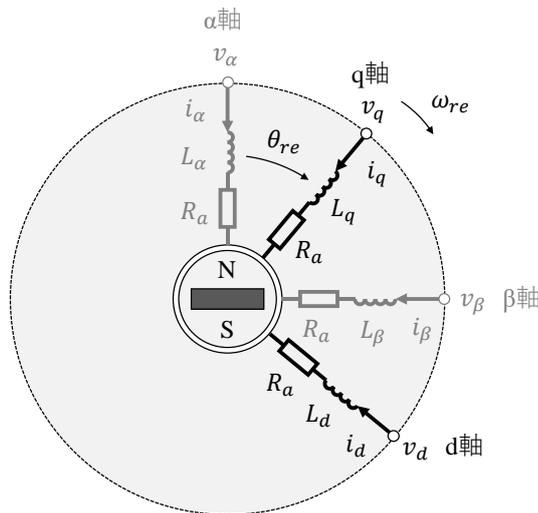


図 2.5 PMSM の回転座標モデル [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 1.6 (2001) を参考に作成

変換行列 $C_{\alpha\beta}$ と変換行列 C_{dq} を用いて式(2.12)に示される三相座標モデルから回転座標モデルへ直接変換する変換行列 C を作成する。

$$C = C_{dq}C_{\alpha\beta} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta_{re}) & \cos\left(\theta_{re} - \frac{2}{3}\pi\right) & \cos\left(\theta_{re} + \frac{2}{3}\pi\right) \\ -\sin(\theta_{re}) & -\sin\left(\theta_{re} - \frac{2}{3}\pi\right) & -\sin\left(\theta_{re} + \frac{2}{3}\pi\right) \end{bmatrix} \quad (2.12)$$

この変換行列 C を用いて PMSM の三相電圧方程式を $d-q$ 座標軸に変換した電圧方程式は、式(2.13)に示される。また、 d 軸インダクタンス L_d は式(2.14)、 q 軸インダクタンス L_q は、式(2.15)、電機子鎖交磁束 Ψ_a は式(2.16)に示される。なお、 ω_{re} は電気角速度を示す。

$$v_{dq} = \begin{bmatrix} v_d \\ v_q \end{bmatrix} = C \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} = C \begin{bmatrix} R_a + pL_u & pM_{uv} & pM_{wu} \\ pM_{uv} & R_a + pL_v & pM_{vw} \\ pM_{vu} & pM_{vw} & R_a + pL_w \end{bmatrix} C^T C \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + C p \begin{bmatrix} \Psi_{fu} \\ \Psi_{fv} \\ \Psi_{fw} \end{bmatrix} \quad (2.13)$$

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_a + pL_d & -\omega_{re}L_q \\ \omega_{re}L_d & R_a + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{re}\Psi_a \end{bmatrix}$$

$$L_d = l_a + \frac{3}{2}(L_a - L_{as}) \quad (2.14)$$

$$L_q = l_a + \frac{3}{2}(L_a + L_{as}) \quad (2.15)$$

$$\Psi_a = \sqrt{3/2} \Psi_f \quad (2.16)$$

したがって、PMSM を定常運転させた時の電圧や電流などの関係は、図 2.6 に示される。

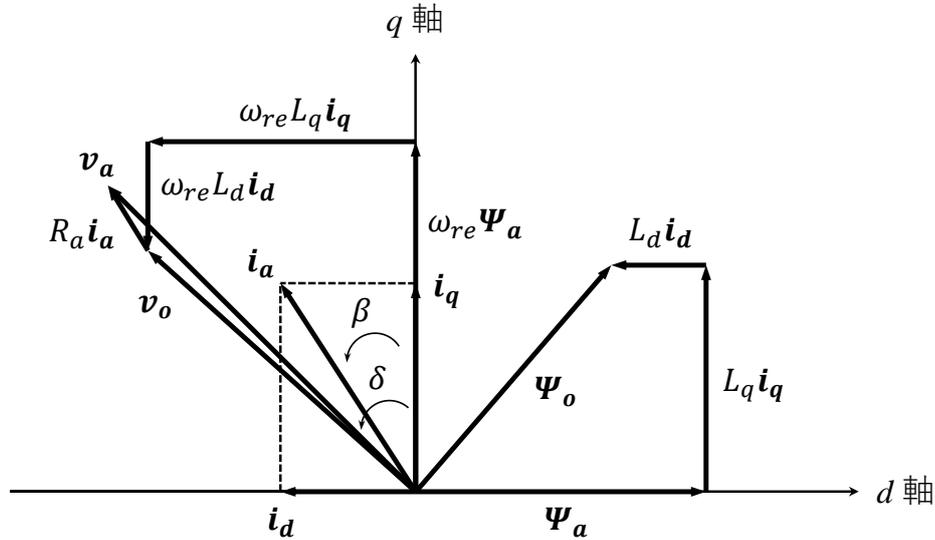


図 2.6 PMSM の基本ベクトル図 [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 1.7 (2001) を一部改変

PMSM のトルク T は、図 2.6 より電流ベクトル i_a と電機子鎖交磁束ベクトル Ψ_o との外積により求められ、磁極のペア数 (極対数) P_n を用いて式(2.17)のように示される。極対数 P_n は、モータ軸の回転角度 (機械角) θ_{rm} と電気角 θ_{re} を用いて式(2.18)のように示される。なお、 T_m は PMSM のマグネットトルクと呼ばれ、 T_r は PMSM のリラクタンストルクと呼ばれる。

$$T = P_n \{ \Psi_a i_q + (L_d - L_q) i_d i_q \} = P_n \Psi_a i_q + P_n (L_d - L_q) i_d i_q = T_m + T_r \quad (2.17)$$

$$P_n = \theta_{re} / \theta_{rm} \quad (\because P_n = 1 \text{ ならば } \theta_{re} = \theta_{rm}) \quad (2.18)$$

2.3.1 表面磁石型同期電動機 (SPMSM) の制御 [37]

SPMSM では、式(2.17)で示したマグネットトルク T_m のみを利用できる。よって、d 軸の電流は $i_d = 0$ となるように制御を行う必要がある。この制御法は、 $i_d = 0$ 制御と呼ばれるベクトル制御の 1 種である。この制御法では、トルク発生に関与する d 軸電流を常に 0 にするため、同一トルクに対する電流は最小となり効率的である。一方で、IPMSM でも $i_d = 0$ 制御を行うことは可能である。しかし、IPMSM の特徴であるリラクタンストルク T_r が利用できなくなるため、IPMSM にとって $i_d = 0$ 制御は必ずしも適切な制御方法とはならない。また、 $L_d = L_q$ ならば $i_d = 0$ 制御と同じになる。

2.3.2 埋込磁石型同期電動機 (IPMSM) の制御 [37]

電流ベクトル \mathbf{i}_a と電流ベクトル \mathbf{i}_q との間には、電流位相 β (進み位相角) がある。この電流位相には、同一電流に対して発生トルクを最大にできる電流位相が存在する。これは、電機子電流に対して最も効率的にトルクを発生させる条件になる。このような状態になるように電流ベクトルを制御する方法は、最大トルク/電流制御と呼ばれている。最適な電流位相を求めるためには、式(2.17)を電流ベクトルの大きさ I_a と電流位相 β を用いて式(2.19)に置き換える。

$$T = P_n \{ \Psi_a i_q + (L_d - L_q) i_d i_q \} = P_n \left\{ \Psi_a I_a \cos(\beta) + \frac{1}{2} (L_q - L_d) I_a^2 \sin(2\beta) \right\} \quad (2.19)$$

そして、式(2.19)を電流位相 β で偏微分して 0 と置き、電流位相 β について解く。それによって電流位相 β は式(2.20)のように求められる。

$$\frac{\partial T}{\partial \beta} = P_n \{ -2(L_q - L_d) I_a^2 \sin^2(\beta) - \Psi_a I_a \sin(\beta) + (L_q - L_d) I_a^2 \} = 0$$

$$\beta = \sin^{-1} \left(\frac{-\Psi_a + \sqrt{\Psi_a^2 + 8(L_q - L_d) I_a^2}}{4(L_q - L_d) I_a} \right) \quad (2.20)$$

また、図 2.6 より d 軸電流と q 軸電流との関係は、式(2.22)に示される。

$$i_d = \frac{\Psi_a}{2(L_q - L_d)} - \sqrt{\frac{\Psi_a^2}{4(L_q - L_d) I_a} + i_q^2} \quad (2.21)$$

最大トルク/電流制御を行うと、電流ベクトルは負荷状態に応じて最大トルク/電流制御曲線上を移動する。同一トルクが発生した場合は、電流が最小になることから銅損が最小になり、電流の上限値を考慮する場合は最大の発生トルクが得られる。最大トルク/電流制御時の i_d, i_q の特性を表した電流ベクトルの軌跡は、図 2.7 に示される。

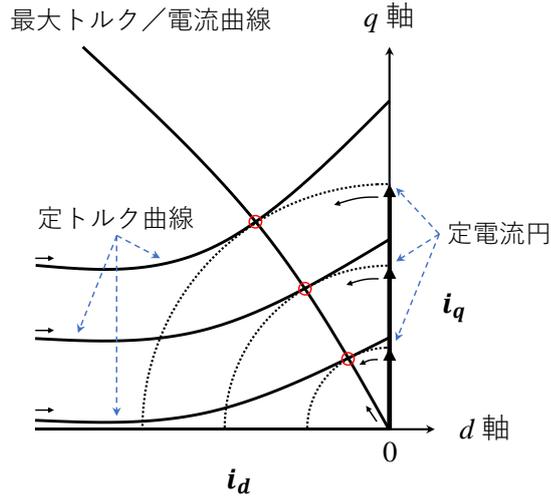


図 2.7 最大トルク／電流制御時の電流ベクトル軌跡 [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 2.8 (2001) を参考に作成

次に、電流位相に対する電流と鎖交磁束の特性において、同一トルク発生時に鎖交磁束が最小になる条件があるとする。これは、鎖交磁束に対して最も効率的にトルクを発生させる条件になり、鉄損が最小になる条件にもなる。このような状態になるように制御する方法は、最大トルク／磁束制御と呼ばれている。ここで、鎖交磁束は図 2.6 より式(2.22)と式(2.23)に示される。

$$\begin{bmatrix} \Psi_{od} \\ \Psi_{oq} \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \Psi_a \\ 0 \end{bmatrix} \quad (2.22)$$

$$\Psi_o = \sqrt{\Psi_{od}^2 + \Psi_{oq}^2} = \sqrt{(L_d i_d + \Psi_a)^2 + (L_q i_q)^2} \quad (2.23)$$

式(2.24)のように、式(2.23)を用いて式(2.17)の q 軸電流 i_q を消去し、d 軸電流 i_d で偏微分したものを 0 と置く。結果的に i_d と i_q は、それぞれ式(2.25)と式(2.26)のように求められる。最大トルク／磁束制御時の電流ベクトル軌跡は、図 2.8 に示される。

$$\frac{\partial T}{\partial i_d} = P_n \{ \Psi_a + (L_d - L_q) i_d \} \frac{\sqrt{\Psi_o^2 - (L_d i_d + \Psi_a)^2}}{L_q} = 0 \quad (2.24)$$

$$i_d = -\frac{1}{L_d} \left(\Psi_a + \frac{-L_q \Psi_a + \sqrt{(L_q \Psi_a)^2 + 8(L_q - L_d)^2 \Psi_o^2}}{4(L_q - L_d)} \right) \quad (2.25)$$

$$i_q = \pm \frac{1}{L_q} \sqrt{\Psi_o^2 - \left(\frac{-L_q \Psi_a + \sqrt{(L_q \Psi_a)^2 + 8(L_q - L_d)^2 \Psi_o^2}}{4(L_q - L_d)} \right)^2} \quad (2.26)$$

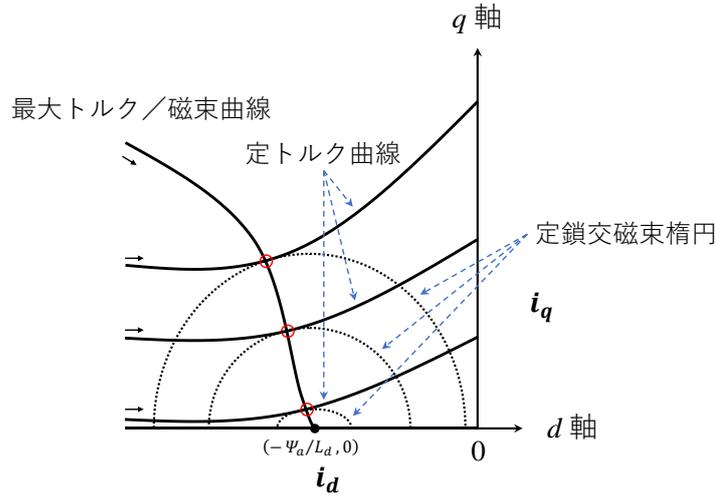


図 2.8 最大トルク／磁束制御時の電流ベクトル軌跡 [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 2.9 (2001) を参考に作成

その次に、d 軸電流 i_d が逆向きである、すなわち、 i_d が負であると反作用により減磁効果が生じる。この減磁効果により d 軸方向の磁束を減少させることができる。これは、等価的な弱め界磁制御になり、弱め磁束制御と呼ばれている。ここで、モータの端子電圧を誘起電圧 V_o と見なして、電気角速度 ω_{re} と電機子鎖交磁束ベクトルの大きさ Ψ_o を用いると、式(2.27)に示される。また、その誘起電圧 V_o を制限値 V_{om} に保たせると式(2.28)のように示される。

$$V_o = \omega_{re} \Psi_o = \omega_{re} \sqrt{(L_d i_d + \Psi_a)^2 + (L_q i_q)^2} \quad (2.27)$$

$$(L_d i_d + \Psi_a)^2 + (L_q i_q)^2 = \left(\frac{V_{om}}{\omega_{re}} \right)^2 \quad (2.28)$$

式(2.28)は 2 次元平面上で楕円になり、 i_d と i_q との関係は式(2.29)のように示される。ただし、 $|i_q| \leq V_{om}/(\omega_{re} L_q)$ であり、 i_d が $i_d = -\Psi_a/L_d$ 以上ならば分子の第 2 項は正である。

$$i_d = \frac{-\Psi_a \pm \sqrt{(V_{om}/\omega_{re})^2 + (L_q i_q)^2}}{L_d} \quad (2.29)$$

弱め磁束制御を行って、速度が一定の式(2.28)の楕円（定誘起電圧楕円）と最大トルク／誘起電圧曲線との交点で得られるトルクは、正の領域で最大または負の領域で最大になる。弱め磁束制御時の電流ベクトル軌跡は、図 2.9 に示される。

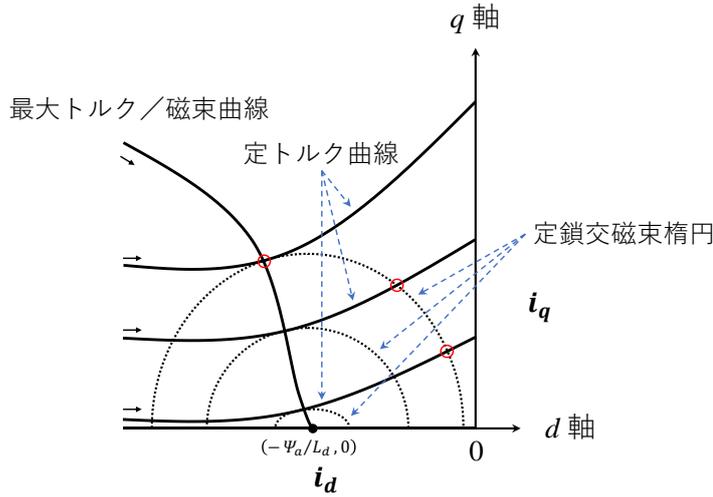


図 2.9 弱め磁束制御時の電流ベクトル軌跡 [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 2.10 (2001) を参考に作成

その次に、電機子電圧ベクトル v_a と電流ベクトル i_q との間には、電圧位相 δ (進み位相角) がある。電流位相 β と電圧位相 δ との関係が $\beta = \delta$ になる、すなわち、 $i_d/i_q = v_d/v_q$ を満たせば力率を 1 にできる。このような状態になるように制御する方法は、力率 1 制御と呼ばれている。ここで、 $d-q$ 座標軸に変換した電圧方程式を以下の式のように分ける。

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_a & 0 \\ 0 & R_a \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} v_{od} \\ v_{oq} \end{bmatrix} + p \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad (2.30)$$

$$\begin{bmatrix} v_{od} \\ v_{oq} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_{re} L_q \\ \omega_{re} L_d & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{re} \Psi_a \end{bmatrix} \quad (2.31)$$

定常状態時では、式(2.30)の第 3 項は 0 になるため、式(2.32)のように置き換えられる。

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_a & 0 \\ 0 & R_a \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 & -\omega_{re} L_q \\ \omega_{re} L_d & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{re} \Psi_a \end{bmatrix} \quad (2.32)$$

i_d と i_q との関係は、式(2.32)から式(2.33)のように楕円で示される。また、 i_d は式(2.34)に示される。ただし、 $|i_q| \leq \Psi_a / 2\sqrt{L_d L_q}$ である。

$$\left(i_d + \frac{\Psi_a}{2L_d} \right)^2 + \left(\sqrt{\frac{L_q}{L_d}} i_q \right)^2 = \left(\frac{\Psi_a}{2L_d} \right)^2 \quad (2.33)$$

$$i_d = \frac{-\Psi_a \pm \sqrt{\Psi_a^2 - 4L_d L_q i_q^2}}{2L_d} \quad (2.34)$$

最後に、トルクおよび速度が一定である時、電流ベクトルを制御することで銅損と鉄損の値はそれぞれ変動するため、それらの損失を最小にして効率を最大にすることができる。このような状態になるように制御する方法は、最大効率制御と呼ばれている。ここで、損失が考慮されたモータモデルにおける損失 W_{loss} は、 i_d と等価銅損抵抗 R_c の電流との差 i_{od} 、電気角速度 ω_{re} 、トルク T を用いて表現できる。任意のトルクおよび速度において損失 W_{loss} が最小になるためには、式(2.35)のように損失 W_{loss} を i_{od} で偏微分したものを 0 と置く。

$$\frac{\partial W_{\text{loss}}(i_{od}, \omega_{re}, T)}{\partial i_{od}} = 0 \quad (2.35)$$

式(2.35)により式(2.36)の関係が導出される。 f_1, f_2, K_ω は、式(2.37)から式(2.39)に示される。

$$f_1(i_{od})f_2(i_{od}) = K_\omega T^2 \quad (2.36)$$

$$f_1(i_{od}) = P_n^2 \{R_a R_c^2 i_{od} + \omega_{re}^2 L_d (R_a + R_c) (\Psi_a + L_d i_{od})\} \quad (2.37)$$

$$f_2(i_{od}) = \{\Psi_a + (L_d - L_q) i_{od}\}^3 \quad (2.38)$$

$$K_\omega = \{R_a R_c^2 + (\omega_{re} L_d)^2 (R_a + R_c) (L_d - L_q)\} \quad (2.39)$$

式(2.36)を満たす最適な電流 i_{od} が得られると i_{oq} は式(2.40)のように示される。ただし、式(2.36)の計算は困難であるため、トルクと速度に関する近似関数など用いて i_{od} を計算する。 i_d と i_q は以下に示される。また、最大効率制御時の電流ベクトル軌跡は、図 2.10 に示される。

$$i_{oq} = \frac{T/P_n}{\Psi_a + (L_d - L_q) i_{od}} \quad (2.40)$$

$$i_d = i_{od} - \frac{\omega_{re} L_q i_{oq}}{R_c} \quad (2.41)$$

$$i_q = i_{oq} + \frac{\omega_{re} (\Psi_a + L_d i_{od})}{R_c} \quad (2.42)$$

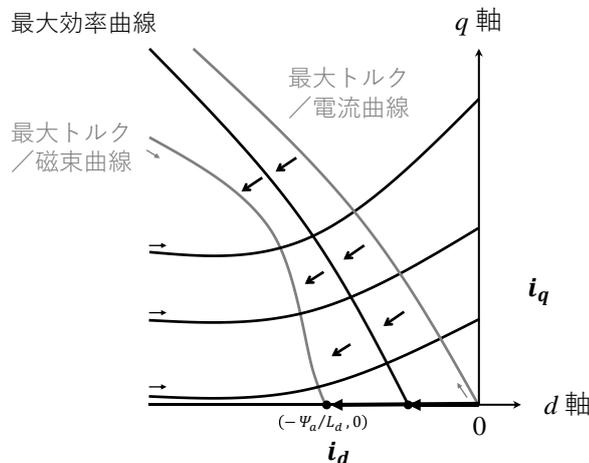


図 2.10 最大効率制御の電流ベクトル軌跡 [37]

武田洋次, 松井信行, 森本茂雄, 本田幸夫: 埋込磁石同期モータの設計と制御, オーム社, 図 2.11 (2001) を参考に作成

2.4 サーボシステムの構成 [38] - [42]

前節では、サーボモータに関する構造や制御法を説明した。本節では、サーボモータを制御対象として、目標位置や目標速度が設定された場合にどのような方法で制御対象を追従させるように制御するかを説明して、サーボシステムの構成も説明する。

制御には、1つの入力に対して1つの出力を行う古典制御 (Classical Control) と呼ばれている制御の方法と複数の入力に対して複数の出力を行う現代制御 (Modern Control) と呼ばれている制御の方法がある。上記で説明した古典制御には、洗濯機のようにボタンのオンとオフによってあらかじめ組み込まれているプログラムを順番に実行するシーケンス制御 (Sequential Control)、水槽用ヒータなど外気温に基づいて目標の水温を達成するための操作量を決定しておきヒータのパワーを調節するフィードフォワード制御 (Feedforward Control)、水槽用ヒータなど温度センサで検知された現在の水温と目標の水温を比較してその誤差をなくすようにヒータの制御量を調節するフィードバック制御 (Feedback Control) がある。また、現代制御は、古典制御のように伝達関数と呼ばれている表現の方法を用いて入力と出力との特性を表すことに対して、状態空間と呼ばれている表現の方法を用いてシステム内部の変数の変化まで観測することができる。

古典制御におけるフィードバック制御には、時間に関係なく目標値が一定である定値制御、時間の経過で目標値が変化する追値制御、任意に目標値が時間変化する追従制御がある。追従制御では、回転位置や回転速度を制御するサーボ機構がある。フィードバック制御において、目標値と出力値との誤差をなくすために PID 制御と呼ばれる代表的な制御方法が用いられている。PID 制御では、フィードバック誤差を定数倍して誤差を修正する P 制御、P 制御で取り除けない誤差を積分で修正する I 制御、フィードバック誤差の振動を微分で抑制して目標値への追従性を向上させる D 制御の3つの制御器から構成されている。

2.4.1 PID 制御を用いたフィードバック制御 [38] [39]

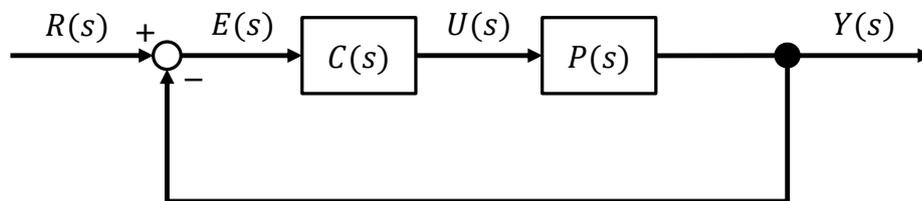


図 2.11 基本的なフィードバック制御系のブロック線図 [38]

横山修一、濱根洋人、小野垣仁：基礎と実践 制御工学入門，コロナ社，図 4.1 (2009) を一部改変

本項では、横山、濱根、小野垣 (2009) および小坂 (2013) に基づき説明する。まず、フィードバック制御系の基本形は図 2.11 に示される。なお、この図では時間関数 $f(t)$ から複素関数 $F(s)$ にラプラス変換したものとして、 $P(s)$ は制御対象、 $C(s)$ は制御器、 $R(s)$ は目標値、 $Y(s)$ は制御量、 $E(s)$ は制御偏差、 $U(s)$ は操作量を示す。

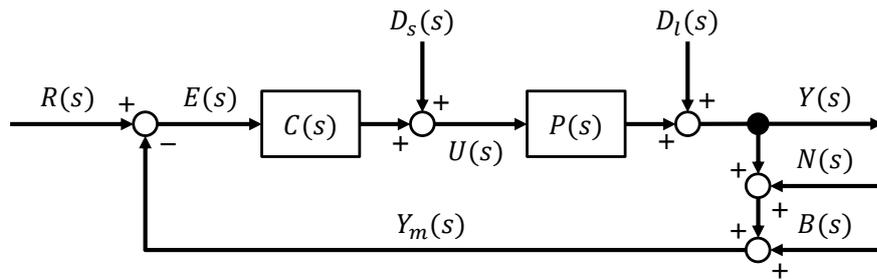


図 2.12 一般的なフィードバック制御系のブロック線図 [38]

横山修一，濱根洋人，小野垣仁：基礎と実践 制御工学入門，コロナ社，図 8.6 (2009) を一部改変

次に、外乱のあるフィードバック制御系のブロック線図は、図 2.12 に示される。ここで、 $D_s(s)$ はシステム外乱（入力外乱）、 $D_l(s)$ は負荷外乱（出力外乱）、 $N(s)$ はセンサへのノイズや AD 変換器の量子化誤差、 $B(s)$ は電圧降下などの要因で生じるバイアス（オフセット）を示す。

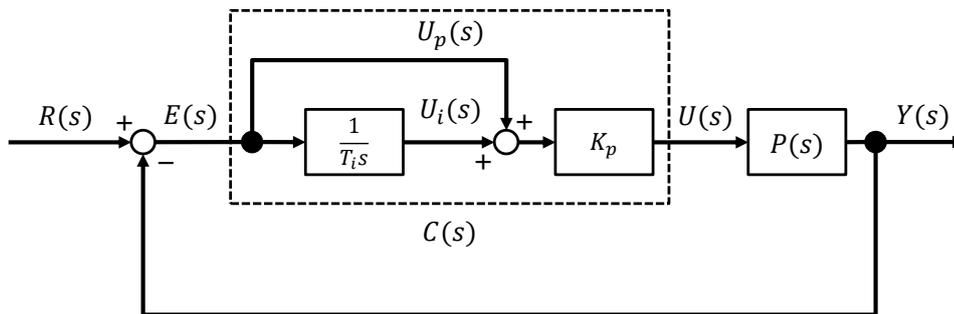


図 2.13 PI 制御系のブロック線図 [39]

小坂学：mbed マイコンによるモーター制御設計法，科学情報出版株式会社，図 1.31 (2013) を一部改変

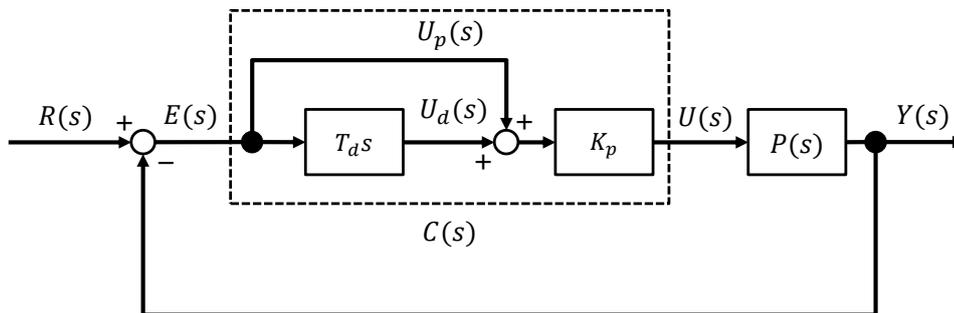


図 2.14 PD 制御系のブロック線図 [39]

小坂学：mbed マイコンによるモーター制御設計法，科学情報出版株式会社，図 1.34 (2013) を一部改変

最後に、図 2.11 と図 2.12 に示したフィードバック制御について、それらの図の中の伝達関数 $C(s)$ に PID 制御が用いられる。PID 制御の種類としては、図 2.13 に示されるような P 制御と I 制御を組み合わせた PI 制御、図 2.14 に示されるような P 制御と D 制御を組み合わせた PD 制御、そして、PD 制御と PI 制御を組み合わせた PID 制御がある。さらに、PD 制御と PI 制御を組み合わせた PID 制御の中でも、図 2.15 に示されるような標準の PID 制御、図 2.16 に示されるような I-PD 制御、図 2.17 に示されるような PI-D 制御がある。

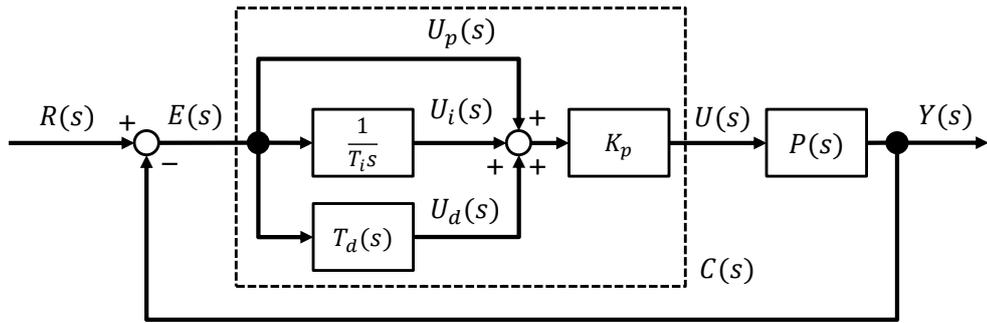


図 2.15 標準 PID 制御系のブロック線図 [38] [39]

横山修一，濱根洋人，小野垣仁：基礎と実践 制御工学入門，コロナ社，図 8.18 (2009) および，小坂学：mbed マイコンによるモーター制御設計法，科学情報出版株式会社，図 1.37 (2013) を一部改変

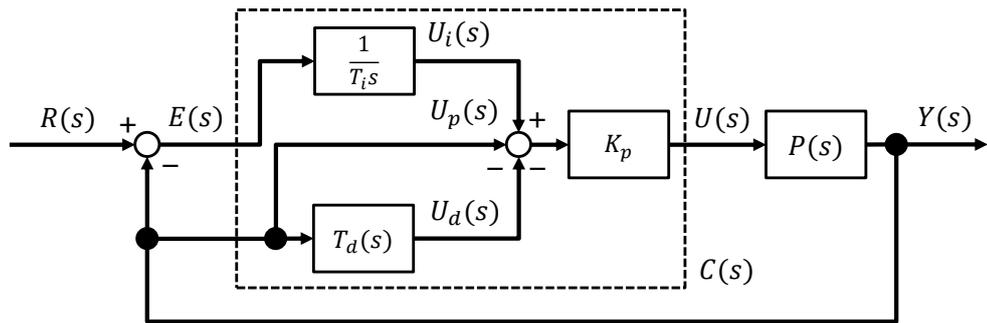


図 2.16 I-PD 制御系のブロック線図 [39]

小坂学：mbed マイコンによるモーター制御設計法，科学情報出版株式会社，図 1.39 (2013) を一部改変

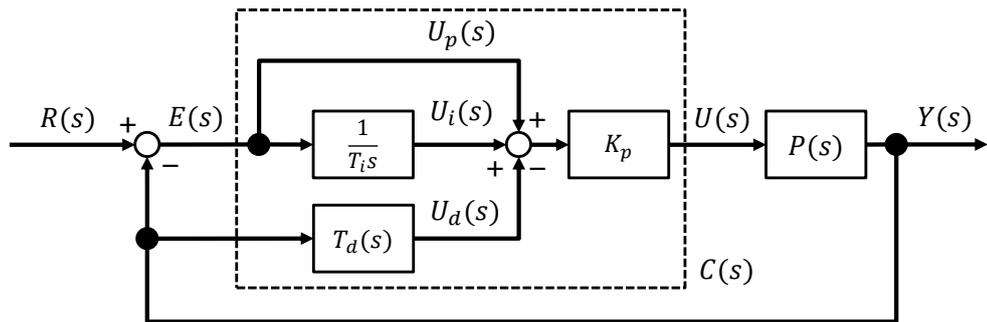


図 2.17 PI-D 制御系のブロック線図 [38] [39]

横山修一，濱根洋人，小野垣仁：基礎と実践 制御工学入門，コロナ社，図 8.18 (2009) および，小坂学：mbed マイコンによるモーター制御設計法，科学情報出版株式会社，図 1.39 (2013) を一部改変

PI 制御は、P 制御を行った際のフィードバック誤差の残り（定常偏差）を I 制御により 0 にするような制御である。ただし、この制御をロボットに用いる場合は、バックラッシュを取り除こうと小刻みに振動する現象が生じるため、I 制御を不完全積分の形にして動作させる。そして、PD 制御は、P 制御の動作を速めるような制御である。ただし、定常偏差が解決しないという問題がある。また、高周波ノイズが入力信号に含まれている場合は、そのノイズを増幅してチャタリングを発生させるという問題がある。定常誤差については、I 制御を加えて PID 制御の形にする。高周波ノイズについては、D 制御を不完全微分の形にして動作させる。

I-PD 制御は、図 2.16 のように、偏差に対しては I 制御を行い、制御量に対しては P 制御および D 制御を行うという制御である。標準の PID 制御では、制御の初期で大きな偏差が生じた場合にその偏差がそのまま PID 制御に加わるため、オーバーシュートしやすくなる。I-PD 制御の形にすることで、大きな偏差が生じた場合にその偏差が I 制御しか加わらないため、オーバーシュートしにくくなる。PI-D 制御は、図 2.17 のように、偏差に対しては P 制御および I 制御を行い、制御量に対しては D 制御を行うという制御である。この制御法についても大きな偏差が生じた場合にその偏差が P 制御および I 制御しか加わらないため、標準の PID 制御と比較してオーバーシュートしにくい。したがって、I-PD 制御は、比例微分先行型 PID 制御とも呼ばれて、PI-D 制御は、微分先行型 PID 制御とも呼ばれている。

ここで、図 2.13 の PI 制御器から図 2.17 の PI-D 制御器までのそれぞれの伝達関数 $C(s)$ は、式(2.43)から式(2.47)に示される。なお、 K_p は P 制御の修正量を調節するゲイン、 K_i は I 制御の修正量を調節するゲイン、 T_i は I 制御の積分時間、 K_d は D 制御の修正量を調節するゲイン、 T_d は D 制御の微分時間を示す。

$$C_{pi}(s) = K_p + \frac{K_i}{s} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (2.43)$$

$$C_{pd}(s) = K_p + K_d s = K_p (1 + T_d s) \quad (2.44)$$

$$C_{pid}(s) = K_p + \frac{K_i}{s} + K_d s = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2.45)$$

$$C_{i-pd}(s) = \frac{K_i}{s} - (K_p + K_d s) = K_p \left(\frac{1}{T_i s} - 1 - T_d s \right) \quad (2.46)$$

$$C_{pi-d}(s) = K_p + \frac{K_i}{s} - K_d s = K_p \left(1 + \frac{1}{T_i s} - T_d s \right) \quad (2.47)$$

2.4.2 極配置法を用いた PID ゲインの設計 ^{[38] [40]}

本項では、横山、濱根、小野垣 (2009) に基づいて説明する。PID 制御を行う際は、P 制御、I 制御、D 制御それぞれのゲインを決定する必要があり、これらのゲインの値によってシステムの応答が変化する。ゲインの決定方法には、ステップ応答法 (Step Response Method) と限界感度法 (Ziegler and Nichols Frequency Response Method)、極配置法 (Pole Placement Method) がある。

ステップ応答法は、制御対象の閉ループのステップ応答波形から PID 制御のゲインを求める方法である。制御対象には入力値が決まると出力値が決まる性質 (定位性) と入力値が決まっても出力値が一意に決まらない性質 (無定位性) がある。制御対象が定位性である場合はむだ時間 L を含めた 1 次遅れ系に近似して、制御対象が無定位性である場合はむだ時間 L を含めた積分系に近似する。そして、CHR の調整則 (Chien, Hrones and Reswch Method) と呼ばれている代表的な PID 制御のゲイン導出式によって PID ゲインの値を求める。この調整則では、オーバーシュートなしとオーバーシュートあり (振幅減衰比 20%) でゲイン導出式の係数が異なる。

限界感度法は、制御対象の閉ループのステップ応答波形において持続的な振動（持続振動）が生じた場合の波形から PID 制御のゲインを求める方法である。この方法では、P 制御のみ動作させて P 制御のゲイン K_p の値を小さな値から徐々に値を増加させる。 K_p の値を増加させるとある値で応答波形に持続的な振動が生じることから持続振動が生じた際の周期とゲイン導出式によって PID ゲインの値を求める。一方で、極配置法は、制御対象の伝達関数 $P(s)$ と制御器の伝達関数 $C(s)$ による閉ループの応答が望ましい応答となるよう極を配置する方法である。その極とは、伝達関数の特性方程式の解である。この方法では、ステップ応答法や限界感度法と異なり実験的に PID ゲインを求める必要はない。

例えば、ここに 2 次遅れ系の伝達関数 $P(s)$ があって標準 PID 制御器 $C(s)$ を用いる場合は、以下に示される式(2.50)から式(2.58)の手順に沿って計算することで PID ゲインの値が求められる。まず、制御対象と制御器の伝達関数は、式(2.48)と式(2.49)にそれぞれ示される。

$$P(s) = \frac{A}{s(Bs + 1)} \quad (2.48)$$

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2.49)$$

次に、制御対象と制御器による閉ループの伝達関数 $G_{cl}(s)$ は、式(2.50)に示される。

$$\begin{aligned} G_{cl}(s) &= \frac{P(s)C(s)}{1 + P(s)C(s)} \\ &= \frac{\frac{A}{s(Bs + 1)} \cdot K_p \left(1 + \frac{1}{T_i s} + T_d s \right)}{1 + \frac{A}{s(Bs + 1)} \cdot K_p \left(1 + \frac{1}{T_i s} + T_d s \right)} \\ &= \frac{\frac{AK_p \left(1 + \frac{1}{T_i s} + T_d s \right)}{s(Bs + 1)}}{\frac{s(Bs + 1) + AK_p \left(1 + \frac{1}{T_i s} + T_d s \right)}{s(Bs + 1)}} \\ &= \frac{AK_p \left(1 + \frac{1}{T_i s} + T_d s \right)}{s(Bs + 1) + AK_p \left(1 + \frac{1}{T_i s} + T_d s \right)} \\ &= \frac{\frac{S}{B} \cdot AK_p \left(1 + \frac{1}{T_i s} + T_d s \right)}{\frac{S}{B} \cdot s(Bs + 1) + \frac{S}{B} \cdot AK_p \left(1 + \frac{1}{T_i s} + T_d s \right)} \\ &= \frac{\frac{AK_p T_d}{B} s^2 + \frac{AK_p}{B} s + \frac{AK_p}{B T_i}}{s^3 + \left(\frac{1}{B} + \frac{AK_p T_d}{B} \right) s^2 + \frac{AK_p}{B} s + \frac{AK_p}{B T_i}} \end{aligned} \quad (2.50)$$

式(2.50)の特性方程式は、式(2.51)のように s の 3 次式で示される。

$$s^3 + \left(\frac{1}{B} + \frac{AK_p T_d}{B}\right)s^2 + \frac{AK_p}{B}s + \frac{AK_p}{BT_i} = 0 \quad (2.51)$$

固有周波数を ω_n 、減衰係数を ζ とする一般的な 3 次系の閉ループの伝達関数において、その特性方程式は式(2.52)に示される。なお、 α は ω_n の係数を示す。

$$\begin{aligned} & (s + \alpha\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2) \\ & = s^3 + (2\zeta\omega_n + \alpha\omega_n)s^2 + (2\zeta\alpha\omega_n^2 + \omega_n^2)s + \alpha\omega_n^3 = 0 \end{aligned} \quad (2.52)$$

式(2.51)と式(2.52)の係数を比較すると、以下のように示される。

$$\left(\frac{1}{B} + \frac{AK_p T_d}{B}\right) = 2\zeta\omega_n + \alpha\omega_n \quad (2.53)$$

$$\frac{AK_p}{B} = 2\zeta\alpha\omega_n^2 + \omega_n^2 \quad (2.54)$$

$$\frac{AK_p}{BT_i} = \alpha\omega_n^3 \quad (2.55)$$

K_p, T_i, T_d について、式(2.53)から式(2.55)をそれぞれ整理すると以下に示される。

$$K_p = \frac{B\omega_n^2(2\zeta\alpha + 1)}{A} \quad (2.56)$$

$$T_i = \frac{2\zeta\alpha + 1}{\alpha\omega_n} \quad (\because K_i = K_p/T_i) \quad (2.57)$$

$$T_d = \frac{B\omega_n(2\zeta + \alpha) - 1}{B\omega_n^2(2\zeta\alpha + 1)} \quad (\because K_d = K_p T_d) \quad (2.58)$$

ここで、式(2.52)の解は以下のように示される。

$$s_1 = -\alpha\omega_n \quad (2.59)$$

$$s_2 = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \quad (2.60)$$

式(2.60)の解において減衰係数が $0 < \zeta < 1$ のとき、式(2.61)のように置き換える。

$$s_2 = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2} = x \pm jy \quad (2.61)$$

式(2.61)より式(2.62)と式(2.63)が導出される。

$$\omega_n = -x/\zeta \quad (2.62)$$

$$\omega_n^2 = \frac{y^2}{(1 - \zeta^2)} \quad (2.63)$$

減衰係数 ζ は、式(2.62)と式(2.63)より式(2.64)のように示すことができる。また、固有周波数 ω_n は、式(2.63)と式(2.64)より式(2.65)のように示すことができる。

$$\zeta = \pm \frac{x}{\sqrt{x^2 + y^2}} \quad (2.64)$$

$$\omega_n = \sqrt{x^2 + y^2} \quad (\omega_n > 0) \quad (2.65)$$

さらに、式(2.60)の解において減衰係数が $0 < \zeta < 1$ 以外のとき、式(2.66)または式(2.67)のように示すことができる。

$$w_n = -x \quad (\zeta = 1) \quad (2.66)$$

$$w_n = \sqrt{x^2 - y^2} \quad (\zeta < 1) \quad (2.67)$$

極配置法のまとめとして、式(2.59)と式(2.60)に示される閉ループの伝達関数の特性方程式の解を決める、すなわち、係数 α と固有周波数 ω_n および減衰係数 ζ を決めることによって、式(2.53)から式(2.55)に示される PID ゲインの値が理論的に求められるということである。

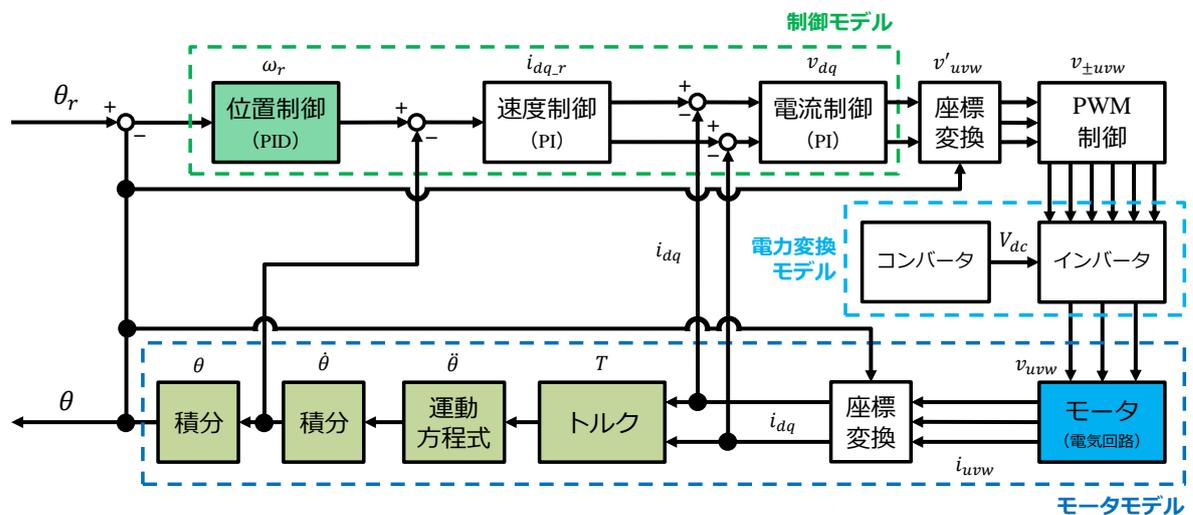


図 2.18 サーボシステムのブロック図 [40]

森本茂雄, 真田雅之: 省エネモーターの原理と設計法, 科学情報出版株式会社, 図 4.17 (2013) を参考に作成

ここで、BLAC モーターの制御システムをブロック図として、図 2.18 に示す。このシステムでは、モーターモデル、電力変換モデル、制御モデルの 3 つのモデルに分類できる。まず、モーターモデルはインバータにより供給される三相交流電圧からエンコーダによるモーターの回転位置の読み取りまでを座標変換やエネルギー変換などで示し、制御モデルはモーターの回転角、速度、電流をマイコンによる PID 制御で示している。また、インバータで三相交流電圧を生成させるための PWM (Pulse Width Modulation) 制御がある。その次に、電力変換モデルは電源により供給される三相交流電圧から直流に変換するコンバータと直流を三相交流電圧に変換するインバータで示している。そして、 $i_d = 0$ 制御を行う場合は L_d と L_q の値に依らず、トルクが i_q との比例になるため、図 2.18 で塗りつぶされている箇所のブロック図が BLDC モーターの制御システムと見なせる。

2.4.3 現代制御理論によるサーボシステムの表現 [41] [42]

2.4.1 項および 2.4.2 項では、フィードバック制御のシステムについて、伝達関数を用いて説明した。本項では、佐藤、下本、熊澤 (2012) および浜田、松本、高橋 (1997) に基づき説明する。現代制御では、線形微分方程式で与えられたシステムの数学モデルを状態空間表現で記述する。この状態空間表現では、状態方程式と出力方程式に分けて、制御対象の内部状態を表現する。

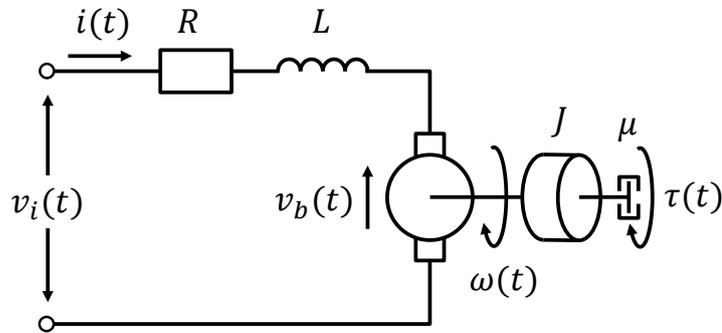


図 2.19 DC モータの等価回路 [41]

佐藤和也, 下本陽一, 熊澤典良: はじめての現代制御, 講談社, 図 2.1 (2012) を一部改変

2.3.1 項で述べた通り PMSM で $i_d = 0$ 制御を行うと $d-q$ 座標軸に変換した二相電圧の式 (2.13) において、 q 軸に関する巻線抵抗の項とインダクタンスに関する項が残り、永久磁石によるトルクのみが利用できる。すなわち、図 2.19 で示されるような DC モータの等価回路が扱える。この回路の抵抗を R 、インダクタンスを L 、端子電圧を $v_i(t)$ 、電流を $i(t)$ とする。また、このモータの逆起電力を $v_b(t)$ 、逆起電力の定数を K_b 、電機子巻線の回転角速度を $\omega(t)$ 、電機子巻線の回転角を $\theta(t)$ とすると、この回路の微分方程式は式(2.68)に示される。

$$L \frac{di(t)}{dt} + Ri(t) = v_i(t) - v_b(t) = v_i(t) - K_b \omega(t) \quad (2.68)$$

さらに、電機子巻線の慣性モーメントを J 、電機子巻線へのトルクを $\tau(t)$ 、トルク定数を K_τ 、ブラシによる粘性摩擦抵抗を μ とすると、運動方程式は式(2.69)に示される。

$$J \frac{d\omega(t)}{dt} + \mu \omega(t) = \tau(t) = K_\tau i(t) \quad (2.69)$$

ここで、状態変数ベクトルと呼ばれているシステム内部の状態を表すようなベクトルを導入する。加えて、入力関係式を $v_i(t) = u(t)$ 、出力関係式を $\omega(t) = y(t)$ とそれぞれ定義する。入力関係式は状態方程式に用いられて、出力関係式は出力方程式に用いられる。そして、図 2.20 の状態変数ベクトル $\mathbf{x}(t)$ は、式(2.70)に示される。

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} \quad (2.70)$$

式(2.68)、式(2.69)、式(2.70)によって状態空間表現は、次のように示される。

$$\frac{d}{dt} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_t}{J} & -\frac{\mu}{J} \end{bmatrix} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \quad (2.71)$$

$$y(t) = [0 \quad 1] \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} \quad (2.72)$$

式(2.71)および式(2.72)について、以下のように置き換える。

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \quad (2.73)$$

$$y(t) = \mathbf{C}\mathbf{x}(t) \quad (2.74)$$

\mathbf{A} はシステムの係数行列、 \mathbf{B} は入力ベクトル、 \mathbf{C} は出力ベクトルと呼ばれている。現代制御では、式(2.68)および式(2.69)のような線形微分方程式を状態変数ベクトルによって式(2.73)および式(2.74)のような形で表現する。図 2.19 で示されるような DC モータの場合では、電機子巻線における回転角速度および電流の変化の様子が分かる。古典制御ではブロック線図によってシステムが表現されることに対して、現代制御では状態変数線図と呼ばれている図によってシステムが表現される。ここで、式(2.73)および式(2.74)の状態変数線図は、以下に示される。

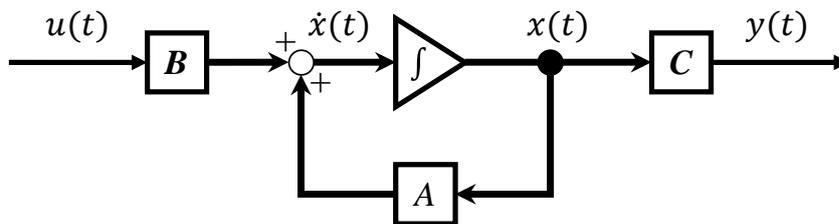


図 2.20 DC モータの状態変数線図 [41]

佐藤和也，下本陽一，熊澤典良：はじめての現代制御，講談社，図 5.4 (2012) を一部改変

そして、この DC モータにフィードバック制御を適用して、安定した制御を行う。ここで、制御対象の状態変数 $\mathbf{x}(t)$ が検出できるものとして、新しい入力 $v(t)$ を与える。入力を含めた全体の構成は、図 2.21 のような状態変数線図で示される。このような構成は、状態フィードバックと呼ばれている。加えて、図 2.21 中の \mathbf{K} は、任意の閉ループ系の極を設定できるような状態フィードバック係数ベクトルと呼ばれている。

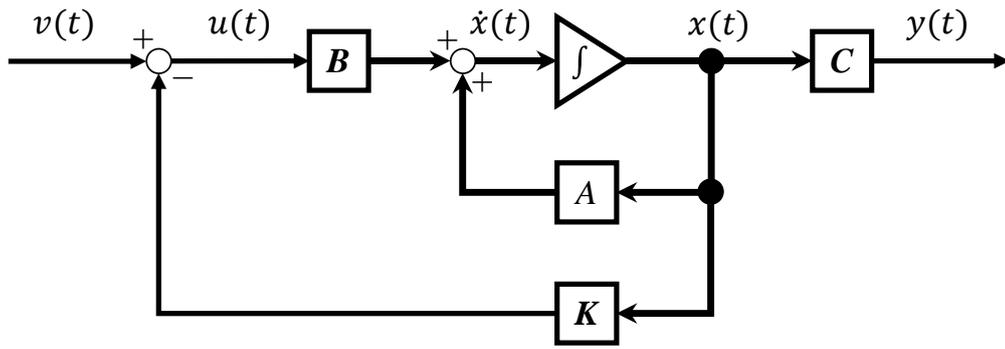


図 2.21 状態フィードバック [41] [42]

佐藤和也, 下本陽一, 熊澤典良: はじめての現代制御, 講談社, 図 9.2 (2012) および,
 浜田望, 松本直樹, 高橋徹: 現代制御理論入門, コロナ社, 図 8.6 (1997) を参考に作成

よって、状態フィードバック係数ベクトル \mathbf{K} を含めた状態空間表現は、以下に示される。

$$\frac{d\mathbf{x}(t)}{dt} = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) + \mathbf{B}v(t) \quad (2.75)$$

$$y(t) = \mathbf{C}\mathbf{x}(t) \quad (2.76)$$

この状態フィードバック係数ベクトル \mathbf{K} を求めるためには、制御対象の状態フィードバックと制御対象を変換したシステム $\mathbf{z}(t)$ の状態フィードバックからそれぞれの係数を比較して、もとの状態変数に戻す必要がある。ここで、 $\mathbf{z}(t)$ の状態変数を $\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c$ として $\mathbf{x}(t)$ および $\mathbf{z}(t)$ の関係を結ぶための変換行列を \mathbf{P} とすると、 $\mathbf{z}(t)$ は次のように示される。

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{A}_c\mathbf{z}(t) + \mathbf{B}_c u(t) = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{z}(t) + \mathbf{P}^{-1}\mathbf{B}u(t) \quad (2.77)$$

$$y(t) = \mathbf{C}_c\mathbf{z}(t) = \mathbf{C}\mathbf{P} \quad (2.78)$$

$\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c$ は可制御標準形と呼ばれ、式(2.79)と式(2.80)のように示される。ここで、可制御とはシステムを初期状態から有限時間内に所望の値に収束させられるかということである。加えて、可制御標準形で扱う a_n, \dots, a_1 は、制御対象の特性多項式 $D(s) = \det(s\mathbf{I} - \mathbf{A})$ の係数であり、 $b_n \dots b_1$ は、制御対象の伝達関数における分子の係数である。

$$\mathbf{A}_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & \cdots & -a_1 \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (2.79)$$

$$\mathbf{C}_c = [b_n \ b_{n-1} \ \cdots \ b_1] \quad (2.80)$$

可制御標準形への変換行列 \mathbf{P} は、可制御行列 \mathbf{V}_c を用いて式(2.81)のように求める。変換行列 \mathbf{P} を求めた後、所望の特性多項式 $D_d(s) = s^n + d_1s^{n-1} + \dots + d_n$ の係数から可制御標準形に基づく状態フィードバック係数ベクトル $\hat{\mathbf{K}}_c$ を求める。それは、式(2.82)によって求められる。

$$\mathbf{P} = \mathbf{V}_c \boldsymbol{\alpha} = [\mathbf{B}, \mathbf{A}\mathbf{B}, \dots, \mathbf{A}^{n-1}\mathbf{B}] \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ a_1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.81)$$

$$\hat{\mathbf{K}}_c = \mathbf{d} - \mathbf{a} = [d_n, d_{n-1}, \dots, d_1] - [a_n, a_{n-1}, \dots, a_1] \quad (2.82)$$

$\mathbf{z}(t) = \mathbf{P}^{-1}\mathbf{x}(t)$ の関係を用いると、もとの制御対象の閉ループ系は次のように示される。

$$\frac{d\mathbf{x}(t)}{dt} = \{\mathbf{A} - \mathbf{B}(\hat{\mathbf{K}}_c\mathbf{P}^{-1})\}\mathbf{x}(t) + \mathbf{B}v(t) \quad (2.83)$$

$$y(t) = \mathbf{C}\mathbf{x}(t) \quad (2.84)$$

したがって、状態フィードバック係数ベクトル \mathbf{K} は、以下のように求められる。

$$\mathbf{K} = \hat{\mathbf{K}}_c\mathbf{P}^{-1} \quad (2.85)$$

2.5 あとがき

本章では、制御対象とニューラルネットワークとの間の関係を明らかにする前段階として、まず、永久磁石同期電動機 (PMSM) の種類^{[36][37]}を説明した。種類は、回転子の表面に永久磁石を張り付ける表面磁石型同期電動機 (SPMSM) と回転子の内部に永久磁石を埋め込む埋込磁石型同期電動機 (IPMSM) の2種類ある。そして、SPMSM と IPMSM それぞれの構造を2.2節で説明した。次に、SPMSM と IPMSM それぞれの制御方法^[37]を2.3節で説明した。SPMSM では、 $i_d = 0$ 制御と呼ばれるベクトル制御があり、同一トルクに対する電流が最小となる。また、IPMSM では、最大トルク/電流制御、最大トルク/磁束制御、弱め磁束制御、力率1制御、最大効率制御について、電流のベクトル軌跡を用いながら制御の理論を説明した。最後に、フィードバック制御を用いた全体のサーボシステムの構成^{[38]-[42]}について、古典制御と現代制御との違いを説明して、PID 制御のゲイン設定の方法や状態変数を用いた制御対象の表現を2.4節で説明した。

第3章 ニューラルネットワーク制御の基礎知識

3.1 はしがき

本章では、まず、ニューラルネットワークとはどのような構成になっているか、ニューラルネットワークにはどのような学習のアルゴリズムがあるか、それらの仕組みを3.2節で説明する。次に、ニューラルネットワーク制御にはどのような種類があるか、ニューロコントロールと適応制御、フィードバック誤差学習による閉ループシステムの学習制御を例として、先行研究の内容を3.3節で説明する。これらの説明により、制御対象とニューラルネットワークとの間にはどのような関係があるか明らかになる。

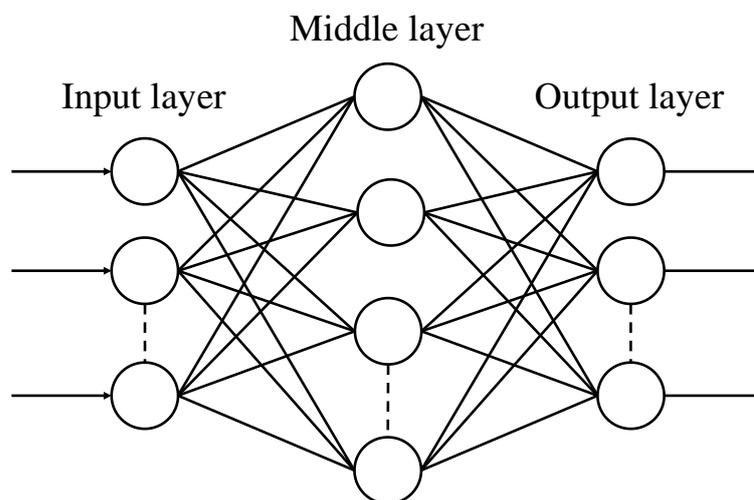
3.2 ニューラルネットワークの仕組み

ニューラルネットワーク (Neural Network) とは、人間の神経細胞を模倣した数学モデルであり、そのモデルを利用して膨大で様々なデータから自ら傾向や法則性を掴み学習させる。そして、学習によって獲得したデータから解析を行ったり、次に起こる現象を予測したりすることがニューラルネットワークで実現する目標である。ニューラルネットワークは、1940年代から研究され始めて1958年には単純パーセプトロンと呼ばれるパターン認識法 (Pattern Recognition) [43] が考案された。この方法は、複数の入力から一つのみ出力を行う関数において、入力信号に重みと呼ばれる値を乗算し、その総和の値が閾値を超えたら1を出力、1未満なら0を出力するという方法である。ただし、この方法では線形分離が可能である場合のみに限る方法であったため、1969年にパーセプトロンを用いて分離が不可能な例が示されて限界が提示された。しかし、パーセプトロンを重ねて多層のパーセプトロンを構成することにより線形分離が不可能であったものも分離が可能となり、研究者に再び注目されるようになった。

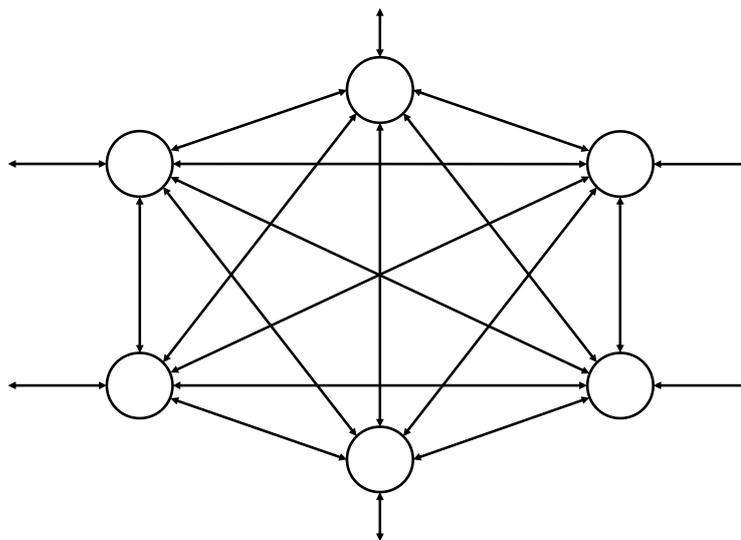
さらに、1986年に誤差逆伝播法 (Backpropagation) [43] [44] と呼ばれる多層パーセプトロンを改善したアルゴリズムが考案されて、研究者に注目された。この方法は、入力層と出力層の間に中間層 (隠れ層) を設けることで学習の効率を向上させるという方法である。ただし、この方法では大量の学習データが必要となり当時はデータが不足していた。しかし、1970年から1980年代に集積回路などの登場で計算機が発達して、学習のデータも十分に集まったため、再びその方法が注目された。1989年には、畳み込みニューラルネットワーク (Convolutional Neural Network) [44] と呼ばれるアルゴリズムが提案された。この方法は、これまでの全結合のニューラルネットワークで無視されるデータの形状に関する情報を活用して、学習の効率を高める方法であり、これまでの全結合のニューラルネットワークに畳み込み層とプーリング層と呼ばれる特殊な層を導入した新しい方法である。その後、2000年代から学習に必要な膨大なデータを容易に集めることが可能となって、また、コンピュータがより高性能になったこともあり、2000年代後半からディープラーニング (Deep Learning) [44] を用いた研究が活発になった。

3.2.1 ニューラルネットワークの構造

ニューラルネットワークの構造には、入力層 (Input Layer)、中間層 (Middle Layer)、出力層 (Output Layer) を設けてそれぞれの階層の中にニューロンを配置する階層型ニューラルネットワークと任意の 2 つのニューロンとの間に双方向の結合を持たせる相互結合型ニューラルネットワークの 2 種類の構造がある [5] [6] [14]。階層型ニューラルネットワークと相互結合型ニューラルネットワークの構造は、図 3.1(a) と 3.1(b) にそれぞれ示される。



(a) 階層型ニューラルネットワーク



(b) 相互結合型ニューラルネットワーク

図 3.1 ニューラルネットワークの構造 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 3 図（1992）より引用

ニューラルネットワークでは、ニューロンだけではなくニューロンとニューロンの結びつきを指すシナプス結合があり、その結合の強さを示す重み (Weight) と呼ばれるパラメータがニューロンに設けられている。また、ニューラルネットワークの階層について、中間層を 2 層以上で構成して学習させているものは、ディープラーニングと呼ばれている。

本研究では、図 3.2 に示すような階層型ニューラルネットワークを構成した。この図において $x(i)$ 、 $y(j)$ 、 $z(k)$ それぞれは、入力層、中間層、出力層のニューロンを示す関数であり、 $v(j,k)$ は中間層から出力層の重み、 $w(i,j)$ は入力層から中間層の重みである。 i 、 j 、 k それぞれは、入力層、中間層、出力層のニューロンの番号である。

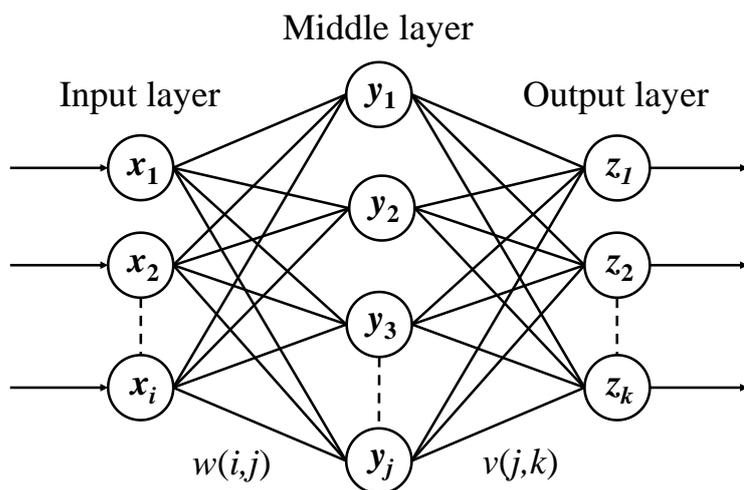
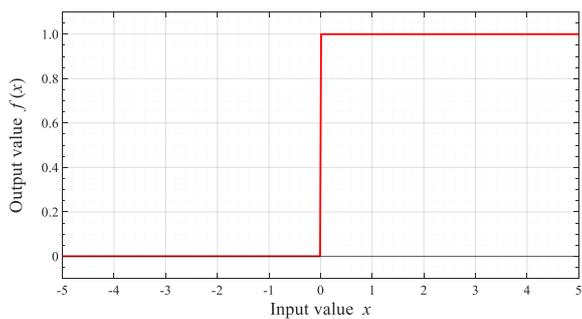


図 3.2 本研究で使うニューラルネットワークの構成

3.2.2 誤差逆伝播を用いた最急降下法

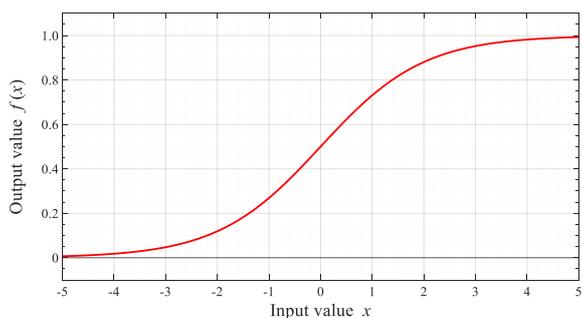
図 3.1(a) で示した階層型ニューラルネットワークにおいて、入力層から出力層に向かってニューロンに関する順伝播^{[14][44]}と呼ばれる計算が行われる。順伝播の計算では、線形変換と非線形変換が行われる。線形変換では、ニューロンに送られる入力信号と重みを乗算したものをニューロンの出力信号として次の階層のニューロンに送り、送られてきた出力信号の総和が計算される。非線形変換では、送られてきた出力信号の総和を用いて入力層、中間層、出力層のニューロンを示す関数の出力が計算される。

この計算が行われる際に、ステップ関数、シグモイド関数、双曲線正接関数、恒等関数などの活性化関数 (Activation Function) が用いられる。ステップ関数は 0 か 1 の値が出力される関数である。これは、パーセプトロンの計算で用いられる関数である。また、シグモイド関数は 0 から 1 の連続した値が出力される関数、双曲線正接関数は -1 から 1 の連続した値が出力される関数、恒等関数は入力された値がそのまま出力される関数である。上記で説明した活性化関数の定義を図 3.3 の式(3.1)から式(3.4)にそれぞれ示す。



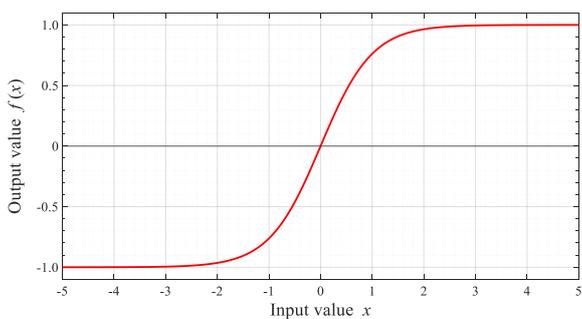
$$f(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (3.1)$$

(a) ステップ関数



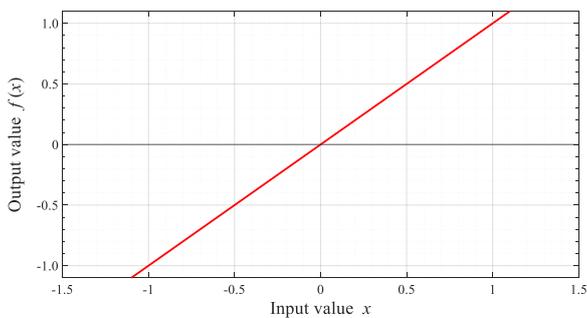
$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

(b) シグモイド関数



$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$

(c) 双曲線正接関数



$$f(x) = x \quad (3.4)$$

(d) 恒等関数

図 3.3 活性化関数の定義

一方で、出力層から入力層に向かってニューロンに関する逆伝播^{[14][44]}と呼ばれる計算が行われる。逆伝播での計算では、順伝播で得られた信号に対してニューロンの微分が乗算されて、次のニューロンに送られる。微分の計算では、連鎖律と呼ばれる合成関数の微分が用いられている。誤差逆伝播とは、入力データに対する理想的な出力の信号を指す教師信号と実際に得られた出力信号との間に生じる誤差を用いてニューラルネットワークを学習させる方法であり、順伝播と逆伝播が行われる。また、教師信号と出力信号の誤差はニューラルネットワークの誤差であり、この誤差を最小にするために最急降下法と呼ばれる方法が用いられる。誤差逆伝播に基づいた制御の研究として、フィードバック誤差学習法^{[31]-[34]}を用いた研究がある。

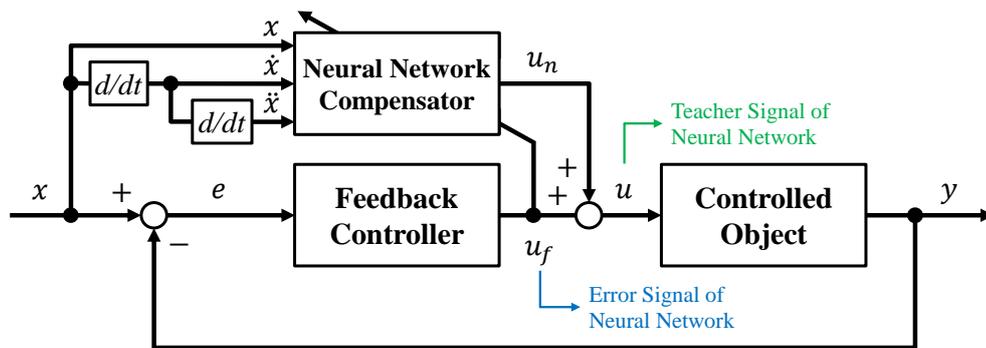


図 3.4 ニューラルネットワークを用いたサーボシステムの構成^[33]

五味裕章, 川人光男: フィードバック誤差学習による閉ループシステムの学習制御, システム制御情報学会論文誌, 4 巻, 1 号, Fig.1・Fig.2 (1991) を参考に作成

本研究では、フィードバック誤差学習法に基づき、図 3.4 に示されるニューラルネットワーク制御を用いたサーボシステムを構成した。図 3.4 の中にある変数の説明は、以下の通りである。

- x : 目標値、また、フィードバック制御系の入力でもある。
- e : 追従偏差、また、フィードバック誤差でもある。
- u_f : フィードバック制御器の出力、また、ニューラルネットワークの誤差信号でもある。
- u_n : ニューラルネットワークの出力である。
- u : 制御対象への操作量、またニューラルネットワークの教師信号でもある。
- y : 制御量、また、フィードバック制御系の出力でもある。
- x : 目標値、また、フィードバック制御系の入力でもある。

この構成の制御器は、フィードバック制御器にニューラルネットワークから構成されるフィードフォワード制御器と並列に接続したものである。このニューラルネットワークでは、制御対象の逆動的モデルを学習で獲得するため、制御対象への入力 u は教師信号として扱われる。これによって、教師信号 u と出力信号 u_n の差はニューラルネットワークの誤差であり、フィードバック制御器の出力 u_f と等しい。そのため、ニューラルネットワークは、学習を通じて出力 u_n が教師信号 u と一致させるように働き、制御対象の逆動的モデルを獲得して、フィードフォワード制御器としてフィードバック制御器を補償する。

$$E_n = \frac{1}{2} \sum_{k=1}^a (u_n(k) - u(k))^2 = \frac{1}{2} \sum_{k=1}^a (-u_f(k))^2 \quad (3.5)$$

$$\frac{\partial E_n}{\partial u_n(k)} = u_n(k) - u(k) = -u_f(k) \quad (3.6)$$

$$\frac{\partial u_n(k)}{\partial v(j,k)} = \frac{-2 \cdot (-y(j)) \cdot e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} = \frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} \cdot y(j) \quad (3.7)$$

$$\frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} = \frac{1}{2} \left(1 - \left(\frac{1 - e^{-\sum_{j=1}^b v(j,k)y(j)}}{1 + e^{-\sum_{j=1}^b v(j,k)y(j)}}\right)^2\right) = \frac{1}{2} (1 - (u_n(k))^2) \quad (3.8)$$

$$\frac{\partial u_n(k)}{\partial y(j)} = \frac{-2 \cdot (-v(j,k)) \cdot e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} = \frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} \cdot v(j,k) \quad (3.9)$$

$$\frac{\partial y(j)}{\partial w(i,j)} = \frac{-2 \cdot (-x(i)) \cdot e^{-\sum_{i=1}^c w(i,j)x(i)}}{\left(1 + e^{-\sum_{i=1}^c w(i,j)x(i)}\right)^2} = \frac{2e^{-\sum_{i=1}^c w(i,j)x(i)}}{\left(1 + e^{-\sum_{i=1}^c w(i,j)x(i)}\right)^2} \cdot x(i) \quad (3.10)$$

$$\frac{2e^{-\sum_{i=1}^c w(i,j)x(i)}}{\left(1 + e^{-\sum_{i=1}^c w(i,j)x(i)}\right)^2} = \frac{1}{2} \left(1 - \left(\frac{1 - e^{-\sum_{i=1}^c w(i,j)x(i)}}{1 + e^{-\sum_{i=1}^c w(i,j)x(i)}}\right)^2\right) = \frac{1}{2} (1 - (y(j))^2) \quad (3.11)$$

$$\begin{aligned} \frac{\partial E_n}{\partial v(j,k)} &= \frac{\partial E_n}{\partial u_n(k)} \cdot \frac{\partial u_n(k)}{\partial v(j,k)} \\ &= \frac{1}{2} (-u_f(k)) \cdot (1 - (u_n(k))^2) \cdot y(j) = \delta_1 \cdot y(j) \end{aligned} \quad (3.12)$$

$$\begin{aligned} \frac{\partial E_n}{\partial w(i,j)} &= \frac{\partial E_n}{\partial u_n(k)} \cdot \frac{\partial u_n(k)}{\partial y(j)} \cdot \frac{\partial y(j)}{\partial w(i,j)} \\ &= (-u_f(k)) \cdot \frac{1}{2} (1 - (u_n(k))^2) \cdot v(j,k) \cdot \frac{1}{2} (1 - (y(j))^2) \cdot x(i) \\ &= \delta_1 \cdot v(j,k) \cdot \frac{1}{2} (1 - (y(j))^2) \cdot x(i) = \delta_2 \cdot x(i) \end{aligned} \quad (3.13)$$

ここで、式(3.5)は誤差信号 E_n を表し、式(3.6)は E_n におけるニューラルネットワークの出力 $u_n(k)$ の偏微分である。 a 、 b 、 c それぞれは、入力層、中間層、出力層のニューロンの数である。式(3.7)は $u_n(k)$ における $v(j,k)$ の偏微分であり、式(3.8)は式(3.7)の右辺にある $y(j)$ の係数を $u_n(k)$ で表している。式(3.9)は $u_n(k)$ における $y(j)$ の偏微分である。式(3.10)は $y(j)$ における $w(i,j)$ の偏微分であり、式(3.11)は式(3.10)の右辺にある $x(i)$ の係数を $y(j)$ で表している。そして、式(3.12)と式(3.13)は式(3.6)から式(3.11)を通して導出される。

本研究では、ニューラルネットワークの誤差が最小になるように、入力層と中間層の重みと中間層と出力層の重みを更新する。具体的に、ニューラルネットワークの教師信号とする制御対象への操作量 u がニューラルネットワークの出力 u_n との誤差 E_n を式(3.6)から式(3.13)に示すような最急降下法で最小化することによって、出力層から中間層へ、また、中間層から入力層への順で重み $v(j,k)$ 、 $w(i,j)$ を変化させて、最終的に正しい出力 u_n が得られるようにする。

3.3 ニューラルネットワーク制御の仕組み

前節では、ニューラルネットワークの仕組みについて、ニューラルネットワークの構造と誤差逆伝播を用いた最急降下法を説明した。誤差逆伝播を用いた最急降下法では、フィードバック誤差学習法に基づき、本研究で構築したニューラルネットワーク制御を用いたサーボシステムについて、その構成とアルゴリズムを説明した。本節では、ニューラルネットワーク制御に関する研究の例として、ニューロコントロールと適応制御とフィードバック誤差学習による閉ループシステムの学習制御を挙げる。

ニューロコントロールと適応制御^{[5][6]}では、著者の大松が、従来の適応制御に関する理論の問題についていくつか指摘している。これらの問題に対して、大松は、ニューラルネットワークが持っている機能を利用することでこれらの問題を解決することができると述べている。大松はまず、ニューラルネットワークを構成しているニューロンのモデルについて定義を示している。次に、ニューラルネットワークで学習を行う際に用いられる誤差逆伝播のアルゴリズムについて6つのステップで手順を示している。最後に、ニューラルネットワークを用いてどのように適応制御を構成すれば良いかを、直列型のニューロコントローラ、並列型のニューロコントローラ、セルフチューニング型のニューロコントローラというように3つの学習機構に大きく分類して、それぞれの学習機能を述べている。

フィードバック誤差学習による閉ループシステムの学習制御^{[32][33]}では、著者の五味と川人が従来の適応制御の問題点と神経回路モデルを用いた学習の問題点について指摘している。特に、神経回路モデルを用いた学習では、教師あり学習についての問題点を指摘している。この問題に対して、著者の五味と川人は、フィードバック誤差学習法を提案した。この学習法は、神経回路モデルに制御対象の逆システムを学習させて前向きな制御を行うというような方法であり、逆動特性モデル学習の理論と非線形レギュレータ学習の理論について解説している。また、フィードバック誤差学習法を用いて学習を行うことにより、制御対象の非線形性が学習後に補償されて参照モデルで規定の応答に従うようになると述べている。規定の応答に従うとは、すなわち、シナプス荷重が最適値に収束して制御入力と神経回路モデルによる非線形フィードバックコントローラとの誤差が0になることである。五味と川人は、確率微分方程式の平均化とリアプノフの方法を用いてその証明を述べている。

3.3.1 ニューロコントロールと適応制御 [5]

この論文の著者である大松は、これまでに提案されてきた適応制御に関する理論の問題点として、構造変化に対する汎化能力、非線形システムに対する理論展開の困難さ、未知雑音に対する学習能力の劣化というような問題点を指摘している。これらの問題点に対して、ニューラルネットワークが持っている並列処理、分散記憶、学習機能などを利用することによって解決できると大松は主張し、代表的な適応制御がニューラルネットワークを用いてどのように構成することが可能であるかを述べている。

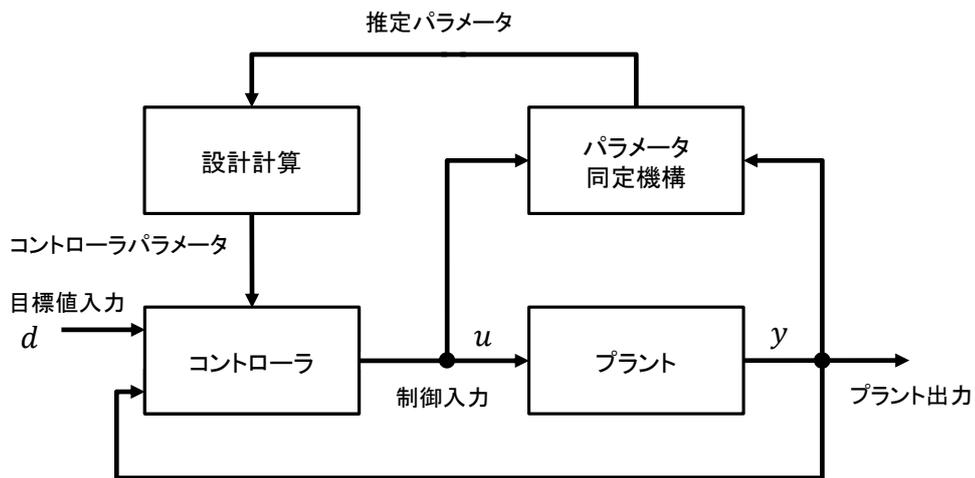


図 3.5 適応制御系のブロック線図 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 1 図（1992）より引用

大松は、図 3.5 のような適応制御系のブロック線図を示した。また、従来の適用制御の問題を次のように述べている。制御系の設計は、この図のようにプラントの出力 y を目標値 d に可能な限り近づける制御入力 u を生成するように、コントローラを設計する。例えば、塩素生成の化学プロセスの場合では、反応炉の圧力、水蒸気量、温度などがプラント出力になり、電解槽の電流と反応炉の入り口のバルブ開度が目標値に近づけさせるための制御入力になる。ここで、対象とするプラントの数学モデルが正確に与えられている場合には、制御入力を制御系の特性解析やコンピュータのシミュレーションなどに基づいて決定できる。しかし、実際のプラントでは、正確な数学モデルが入手できない場合や数学モデルを記述することができない場合がある。そのような場合、プラントの入出力データに基づき、プラントの数学モデルまたはそれに含まれる未知パラメータを推定する必要がある。このようにプラントの動特性を推定するプロセスのことをシステム同定と呼ぶ。そのシステム同定で得られたパラメータの推定値を真値とみなし、数学モデルの出力が目標値になるように制御入力を決定する方法が適応制御の設計手法である。ただし、適応制御の設計手法において、むだ時間などのパラメータや外乱の特性が既知であり、プラントが線形システムで表現できるというような条件がある。従来の適応制御では、これらの条件を満たさないと適用することが困難であった。

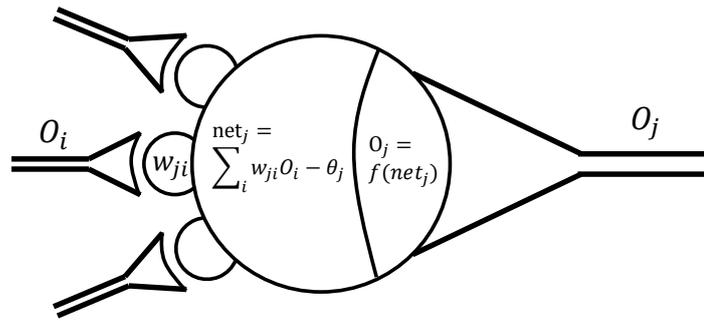


図 3.6 ニューロンの数学モデル [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 2 図（1992）より引用

大松は、図 3.6 のようにニューロンの数学モデルを示した。また、学習のアルゴリズムを次のように述べている。図 3.6 において、細胞体 j への n 個の入力 O_i ($i = 1, 2, \dots, n$) は、シナプス結合によって重み付けられている。 w_{ji} はニューロン i からニューロン j への結合係数、 θ_j は閾値を示す。また、 net_j はニューロン j の電位であり、 $O_j = f(net_j)$ の関係でパルスを出力する。関数 f は、ステップ関数やシグモイド関数などが用いられる。ニューラルネットワークの構造は、階層型ニューラルネットワークとする。階層型ニューラルネットワークでは、入力層の信号に対して出力層で得られる信号が望ましいものとなるように、ニューロン間の結合係数を系統的に求める誤差逆伝播法が提案されている。中間層の層数を 1 とし、その結合係数を w_{ji} 、出力層の結合係数を w_{kj} とする。そして、結合係数の n 回の増分を $\Delta w_{ji}(n)$ 、望ましい出力ベクトルの k 番目の値を t_k とする。シグモイド関数を利用する場合、慣性係数 α と学習係数 η をある正の定数とすると、誤差逆伝播法のアルゴリズムは以下の通りである。ただし、出力 O_k が 0 から 1 の範囲外の値になる場合は O_k のスケールを変換する。

ステップ 1： $n=1$ とおく。

ステップ 2： $\delta_k = (t_k - O_k)O_k(1 - O_k)$

ステップ 3： $\Delta w_{kj}(n+1) = \eta \delta_k O_j + \alpha \Delta w_{kj}(n)$

ステップ 4： $\delta_j = O_j(1 - O_j) \sum_k \delta_k w_{kj}$

ステップ 5： $\Delta w_{ji}(n+1) = \eta \delta_j O_i + \alpha \Delta w_{ji}(n)$

ステップ 6： $n \rightarrow n+1$ としてステップ 2 に行く。

大松は、図 3.7 のように従来の制御系のブロック線図を示した。また、その構成について次のように述べている。フィードフォワードコントローラでは、プラントの出力の目標値が与えられると、その目標値に対応した制御入力決定される。フィードバックコントローラでは、制御入力に対するプラント出力と目標値を比較して、その偏差が減少するように制御入力を変更する。ここで、プラントの数学モデルや動特性が未知であるシステムを制御する場合、人間では適当な試験信号を入力して、それらの出力を観測する。そして、試行錯誤の実験を繰り返すことで望ましい制御入力をプラントの入出力のデータから知見を獲得して、最終的に目標値へ近づけさせるような制御入力を生成する。大松は、このようなプロセスに基づいて、ニューロコントローラを構成すると図 3.8 (a) から 3.8 (c) のように表現できると述べている。

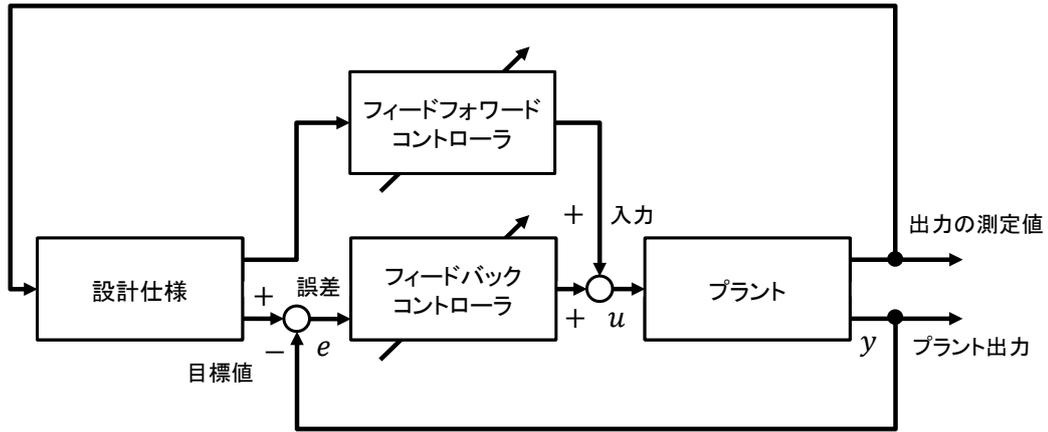


図 3.7 制御系のブロック線図 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 4 図（1992）より引用

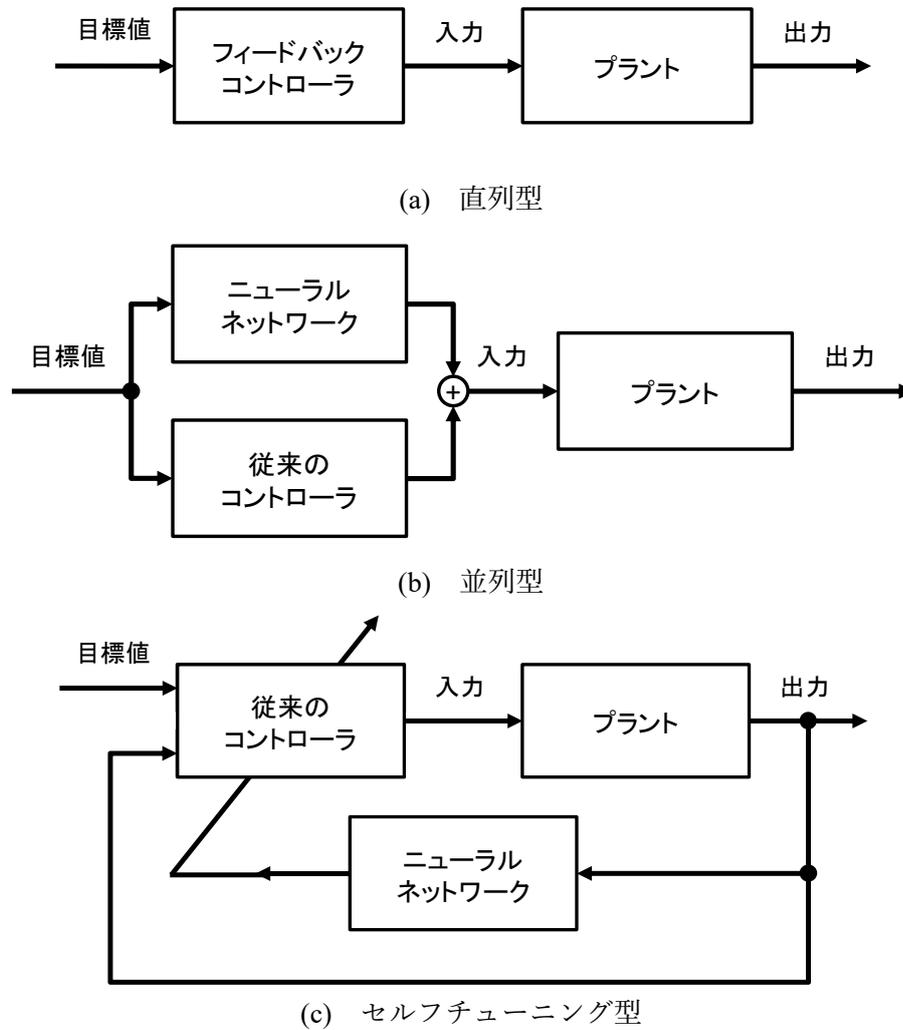
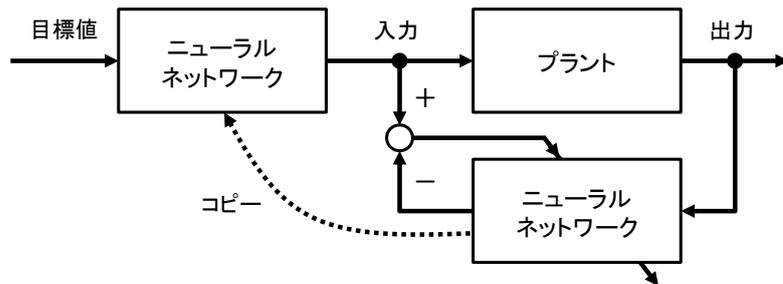
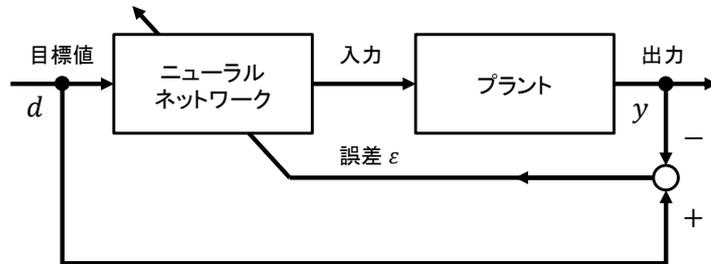


図 3.8 ニューロコントローラの代表的なタイプ [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 5 図（1992）より引用



(a) 一般化学習機構



(b) 特殊化学習機構

図 3.9 直列型学習機構 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36巻，12号，第6図（1992）より引用

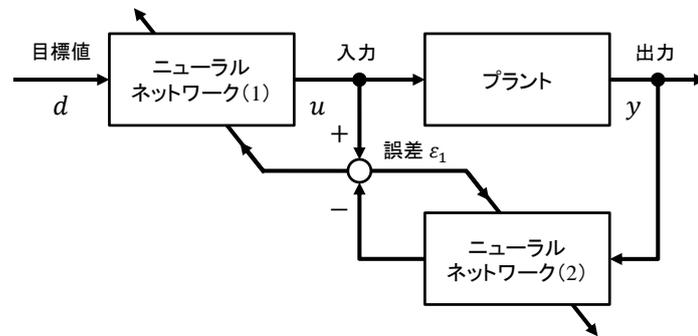


図 3.10 間接的な学習機構 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36巻，12号，第7図（1992）より引用

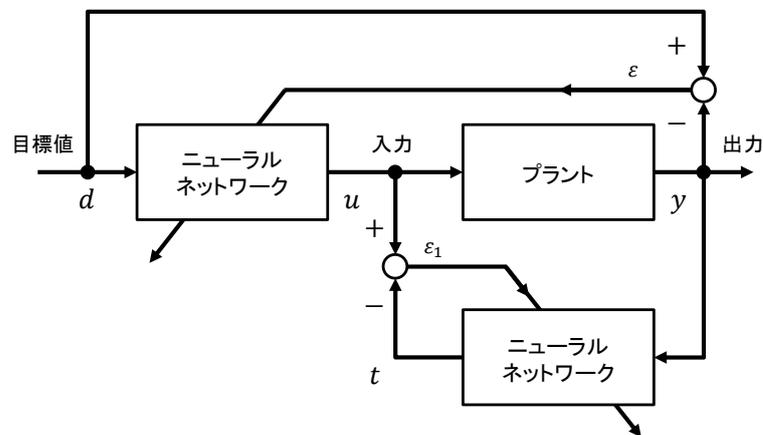


図 3.11 混合学習機構 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36巻，12号，第8図（1992）より引用

まず、図 3.8(a) の構成では、制御性能の劣化を招く可能性があるため、直列型のコントローラを実現する場合は、図 3.9(a) または 3.9(b) のような構成にする。図 3.9(a) は、プラントの逆システムをニューラルネットワークで構成し、それをフィードフォワードコントローラとして利用する。すなわち、図 3.10 の構成である。ニューラルネットワーク 1 は目標値 d からプラント入力 u への逆システムを示し、ニューラルネットワーク 2 はプラント出力 y からプラント入力 u への逆システムを示している。この構成では、プラント出力 y が目標値 d に近くなるようなプラントの制御入力 u をあらかじめ決定することはできないため、図 3.9(b) の構成で解決できるが、多くの学習回数が必要であり、ニューラルネットワークの結合係数 w_{ji} の初期値の選び方も考える必要がある。そこで、図 3.11 の構成が考案された。この構成では、誤差 ε_1 が十分に小さくなるまでニューラルネットワークを学習させてフィードフォワードコントローラへコピーする。また、結合係数 w_{ji} の初期値は図 3.9 で示したニューラルネットワークの結合係数 w_{ji} とする。次に、並列型のコントローラを実現する場合は、図 3.12 のような構成にする。この構成では、図 3.7 で示したフィードフォワードコントローラをニューラルネットワークで構成し、フィードバックコントローラに PID 制御を適用する。そして、目標値とプラント出力との誤差を最小とするようニューラルネットワークを学習させる。最後に、セルフチューニング型のコントローラを実現する場合は図 3.13 のような構成にする。この構成では、PID ゲインをニューラルネットワークにより調整する方法であり、人間によるパラメータ調整をニューラルネットワークで代用する。

大松は、適応制御の構成と誤差逆伝播のアルゴリズムを示し、ニューラルネットワークを用いた適応制御の構成を 3 つの学習機構に分類して、それぞれの学習機能をまとめた。

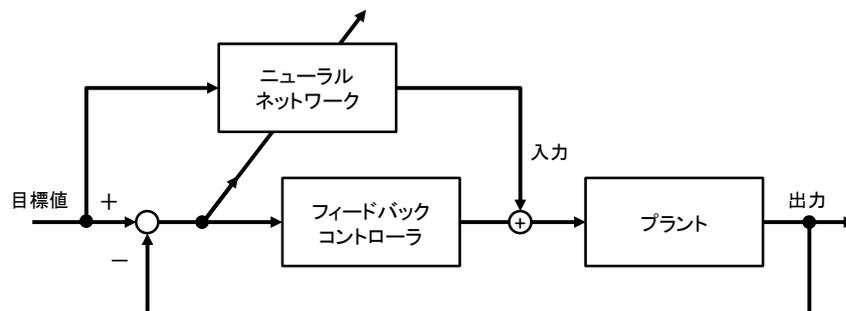


図 3.12 フィードバック学習機構 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 9 図（1992）より引用

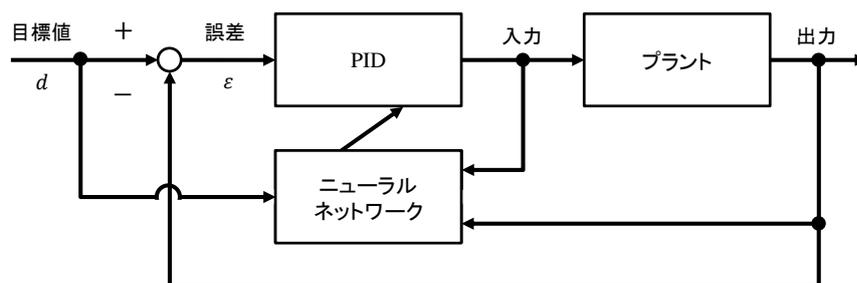


図 3.13 セルフチューニング PID 学習機構 [5]

大松繁：ニューロコントロールと適応制御，システム制御情報学会誌，36 巻，12 号，第 10 図（1992）より引用

3.3.2 フィードバック誤差学習による閉ループシステムの学習制御 [33]

この論文の著者である五味と川人は、神経回路モデルを非線形フィードバックコントローラに利用するときの学習法を提案した。この学習法では、制御対象の非線形性が学習後に補償されて参照モデルで規定の応答に従うようになると述べている。また、五味と川人は、この学習の枠組みにおける収束性を確率微分方程式の平均化とリアプノフの方法で証明した。さらに、動物の小脳における姿勢や歩行の学習制御との関連も議論している。

五味と川人は、適応制御の理論について次のように述べている。フィードバックを必要とする系において、従来は、適応制御理論や神経回路モデルによる方法などが提案されていた。適応制御理論は、線形化した制御対象にて多く利用されており、非線形が強い制御対象の場合には、Dubowsky らや Slotine らの理論を用いる。しかし、Dubowsky らの理論では、非線形性を持った制御対象を高速で動かす場合に適応速度が追いつかないという問題がある。また、Slotine らの理論では、アンプ、アクチュエータ、伝達機構など推測が困難な未知の非線形要素が制御対象に含まれる場合に対応できないという問題がある。

五味と川人は、神経回路モデルを用いた学習問題でよく利用されている教師あり学習について次のように述べている。神経回路モデルを用いてロボットやプラントの制御を行う場合は、正解となる教師信号が必要になるが、その教師信号が得られない場合が多い。この場合は、制御対象の動特性を考慮せずにある時点での制御対象の出力の誤差からその制御対象への入力に誤差を導出することが難しい。この問題に対して、Barto らや Jordan らが対策を提案した。Barto らの提案では、コントローラの過去の動作履歴を記憶して、ある時点での失敗（誤差）から記憶している過去の動作に対し修正を加えるという提案であった。この方法では、学習によってどのくらいの制御性能が向上するかが定量的に判りづらく、収束性も明らかではない。Jordan らの提案では、制御対象からの出力の誤差に基づいて制御対象への入力に誤差（コントローラの出力の誤差）を制御対象の順モデルから逆伝播して計算させるという提案であった。この方法では、制御対象の順モデルが必要であり、誤差を逆伝播させる必要もある。

五味と川人は、随意運動制御系において、フィードバック信号を神経回路モデルの学習のための誤差信号として用いて、神経回路モデルに制御対象の逆システムを学習させて前向きな制御を行う方法を提案してきた。この方法がフィードバック誤差学習法である。そして、この学習方法を用いて、神経回路モデルで構成した適応フィードバックコントローラを学習させる方法が提案されたが、目標位置が一定という条件のもとでの提案であった。五味と川人は、その方法を拡張して目標軌道を与えた場合にも同様の学習が行えることを理論的に示した。また、拡張した方法の枠組みを動物の姿勢保持や歩行などのフィードバックを必要とする制御に対する小脳の適応機構のモデルとして議論を行った。加えて、五味と川人は、提案した学習方法について、ロボットや一般の自動機械、プラントなどの適応学習制御にも応用が可能であると述べている。さらに、実際に利用する場合は、加速度の検出やニューラルネットワークコントローラの種類およびその効果的な学習などを解決する必要があると述べている。

五味と川人は、制御対象の非線形性を補償する適応フィードフォワードコントローラ（CFC：Conventional Feedforward Controller）の理論を次のように述べている。まず、制御対象への入力を τ 、CFC からの出力を τ_c 、非線形フィードバックコントローラ（NNFC：Neural Network Model for Adaptive Nonlinear Feedback Controller）からの出力を τ_n として、式(3.14)から式(3.16)で表す。

$$f(\ddot{\theta}, \dot{\theta}, \theta) = \tau \quad (3.14)$$

$$\tau_c = K_2(\ddot{\theta}_d - \ddot{\theta}) + K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta) \quad (3.15)$$

$$\tau_n = \Phi(\ddot{\theta}, \dot{\theta}, \theta, w) \quad (3.16)$$

ここで、関数 f は未知の非線形性を含む制御対象の動特性、 $\ddot{\theta}, \dot{\theta}, \theta$ は制御対象の加速度、速度、位置のベクトル、 $\ddot{\theta}_d, \dot{\theta}_d, \theta_d$ は指令値のベクトル、 K_2, K_1, K_0 は加速度、速度、位置の各誤差のフィードバックゲインベクトル、 w は NNFC の内部の結合の重み（シナプス荷重）を代表した変数である。また、関数 Φ は NNFC を構成する神経回路モデルの構造に依存している。このモデルでは、シナプス荷重 w を最適値 \hat{w} に取ることによって目的とする f を任意の精度で実現できるものとする。すなわち、式(3.17)を満たすものとする。

$$\Phi(\ddot{\theta}, \dot{\theta}, \theta, \hat{w}) = f(\ddot{\theta}, \dot{\theta}, \theta) \quad (3.17)$$

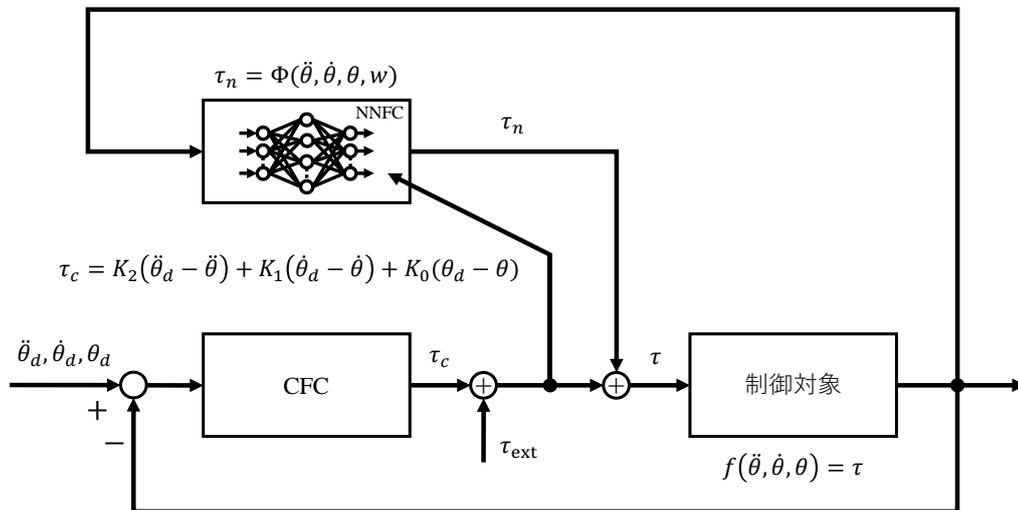


図 3.14 逆動特性モデル学習 [33]

五味裕章，川人光男：フィードバック誤差学習による閉ループシステムの学習制御，システム制御情報学会論文誌，4巻，1号，Fig.1 (1991) より引用

五味と川人は、図 3.14 のような制御系の構成を示した。また、この系の全体の動特性を次のように述べている。外乱を τ_{ext} とおくと制御対象の逆動特性から計算される制御入力 τ は式(3.18)で示され、 τ と NNFC の出力 τ_n の差を τ_{imag} とおくと式(3.19)で示される。式(3.19)は、NNFC が対象の完全な逆モデルを与えられないために生じる誤差トルクとみなすことができる。

$$\tau = \tau_n + \tau_c + \tau_{ext} \quad (3.18)$$

$$\tau_{imag} = \tau - \tau_n = \tau_c + \tau_{ext} \quad (3.19)$$

さらに、CFC として与えられた規範モデルで規定される仮想的な動的システムに対する入力 τ_{ext} に余分な擾乱 $-\tau_{\text{imag}}$ が加わっているかのように表現でき、式(3.20)で示される。

$$K_2(\ddot{\theta} - \ddot{\theta}_d) + K_1(\dot{\theta} - \dot{\theta}_d) + K_0(\theta - \theta_d) = \tau_{\text{ext}} - \tau_{\text{imag}} \quad (3.20)$$

図 3.14 に示したフィードバック誤差学習を行う場合、学習の速度係数を代表させて正定行列 η で表すと、その学習則は式(3.21)で示される。

$$\frac{dw}{dt} = \eta \left(\frac{\partial \Phi}{\partial w} \right)^T (\tau_c + \tau_{\text{ext}}) = \eta \left(\frac{\partial \Phi}{\partial w} \right)^T \tau_{\text{imag}} \quad (3.21)$$

式(3.21)の $\partial \Phi / \partial w$ は NNFC の出力に対する w の依存度を表した項であり、多層パーセプトロンを用いる場合はその項が式(3.22)のように表現される。多層パーセプトロンの階層を m 、階層 k の第 q ユニットへの入力の総和を i_q^k 、出力を O_p^k として、 $k-1$ 層の第 p ユニットから k 層の第 q ユニットへのシナプス荷重を $w_{p,q}^{k-1,k}$ 、各ユニットの入出力関係を h_k とする。

$$\begin{aligned} O_q^k &= h_k(i_q^k) \\ i_q^k &= \sum_p w_{p,q}^{k-1,k} O_p^{k-1} \end{aligned} \quad (3.22)$$

そして、最終段のシナプス荷重 $w_{p,q}^{m-1,m}$ の変化則は、式(3.23)で示せる。

$$\begin{aligned} \frac{dw_{p,q}^{m-1,m}}{dt} &= \eta^{m-1,m} \left(\frac{\partial \Phi}{\partial w_{p,q}^{m-1,m}} \right)^T \tau_{q \text{ imag}} \\ &= \eta^{m-1,m} \left(\frac{\partial i_q^m}{\partial w_{p,q}^{m-1,m}} \right) \left(\frac{\partial h_m}{\partial i_q^m} \right) \tau_{q \text{ imag}} \end{aligned} \quad (3.23)$$

$(\partial h_m / \partial i_q^m) \tau_{q \text{ imag}} = d_q^m$ とすると、 k 層のシナプス荷重 $w_{p,q}^{k-1,k}$ の変化則は式(3.24)で示せる。

$$\begin{aligned} \frac{dw_{p,q}^{k-1,k}}{dt} &= \eta^{k-1,k} \left(\frac{\partial \Phi}{\partial w_{p,q}^{k-1,k}} \right)^T \tau_{\text{imag}} \\ &= \eta^{k-1,k} \left(\frac{\partial i_q^k}{\partial w_{p,q}^{k-1,k}} \right) d_q^k \end{aligned} \quad (3.24)$$

$$\text{ただし} \quad d_q^k = \left(\frac{\partial h_k}{\partial i_q^k} \right) \sum_r \left(\frac{\partial i_r^{k+1}}{\partial h_k} d_r^{k+1} \right) \quad (3.25)$$

したがって、式(3.21)の $\tau_c + \tau_{\text{ext}}$ は、各重み w に対して割り振られるように逆伝播するため、誤差逆伝播法に相当する。五味と川人は、この学習則で $m=2$ かつ h として線形関数を用いたとき Widrow-Hoff の学習則となり $\tau_c + \tau_{\text{ext}}$ が誤差信号の役割を果たすと述べている。また、五味と川人は、次の3つの条件のもとでシナプス荷重 w が最適値 \hat{w} に収束して τ_{imag} が0になることを証明した。

その証明は以下の通りである。

- 条件 1 : 学習速度係数 η は非常に小さな正定値に設定する。
- 条件 2 : 外乱 τ_{ext} および軌道指令の加速度項 $\ddot{\theta}_d$ は 1 回微分可能な強混合強定常過程とする。
- 条件 3 : CFC は学習前、学習中、学習後において制御対象からの出力が漸近安定になるように設計する。

式(3.19)と式(3.21)は、次の確率微分方程式で置き換えられる。

$$f\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) - \Phi\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega), \omega(t, \omega)\right) \quad (3.26)$$

$$= K_2\left(\ddot{\theta}_d(t, \omega) - \ddot{\theta}(t, \omega)\right) + K_1\left(\dot{\theta}_d(t, \omega) - \dot{\theta}(t, \omega)\right) + K_0\left(\theta_d(t, \omega) - \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega)$$

$$\frac{d\omega(t, \omega)}{dt} = \eta \left(\frac{\partial \Phi\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega), \omega(t, \omega)\right)}{\partial \omega} \right)^T \left(\tau_c\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega) \right) \quad (3.27)$$

ω は確率空間での 1 サンプル点である。式(3.27)の解は、式(3.28)に示される平均化方程式の解によって近似されることから、式(3.28)の解 M の収束性に関して議論する。

$$\frac{dM}{dt} = \eta E \left[\left(\frac{\partial \Phi\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega), \omega(t, \omega)\right)}{\partial \omega} \right)^T \left(\tau_c\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega) \right) \right]_{\omega=M} \quad (3.28)$$

まず、式(3.29)に示す関数 L を用いて、平均化方程式のリアプノフ関数の候補として式(3.30)に示される関数 V を考える。その関数 V の時間微分は、式(3.31)で表される。

$$L = \frac{1}{2} \left(\tau_c\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega) \right)^T \left(\tau_c\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega) \right) \quad (3.29)$$

$$V(M, t) = E \left[L\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega), \tau_{\text{ext}}(t, \omega)\right) \right]_{\omega=M} \quad (3.30)$$

$$= \left[L\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega), \ddot{\theta}_d(t, \omega), \dot{\theta}_d(t, \omega), \theta_d(t, \omega), \tau_{\text{ext}}(t, \omega)\right) \right]_{\omega=M}$$

$$\begin{aligned} \frac{dV(M, t)}{dt} &= E \left[\left(\tau_c\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega) \right)^T \right. \\ &\quad \left. \times \frac{d}{dt} \left(\tau_c\left(\ddot{\theta}(t, \omega), \dot{\theta}(t, \omega), \theta(t, \omega)\right) + \tau_{\text{ext}}(t, \omega) \right) \right]_{\omega=M} \end{aligned} \quad (3.31)$$

式(3.31)を簡単化すると、式(3.32)のように表させる。

$$\frac{dV(M, t)}{dt} = E \left[(\tau_c + \tau_{\text{ext}})^T \left\{ \frac{\partial \tau_c}{\partial \ddot{\theta}} \frac{d\ddot{\theta}}{dt} + \frac{\partial \tau_c}{\partial \dot{\theta}} \frac{d\dot{\theta}}{dt} + \frac{\partial \tau_c}{\partial \theta} \frac{d\theta}{dt} + \frac{d\tau_{\text{ext}}}{dt} \right\} \right]_{w=M} \quad (3.32)$$

また、 $\ddot{\theta}$ は式(3.33)で表させる。

$$\ddot{\theta}(t, \omega) = h\{\dot{\theta}(t, \omega), \theta(t, \omega), \ddot{\theta}_d(t, \omega), \dot{\theta}_d(t, \omega), \theta_d(t, \omega), \tau_{\text{ext}}(t, \omega)\} \quad (3.33)$$

よって、式(3.32)は式(3.34)で表させる。

$$\begin{aligned} \frac{dV}{dt} = E \left[(\tau_c + \tau_{\text{ext}})^T \left\{ -K_2 \left[\frac{\partial \ddot{\theta}}{\partial w} \frac{dw}{dt} + \frac{\partial \ddot{\theta}}{\partial \dot{\theta}} \frac{d\dot{\theta}}{dt} + \frac{\partial \ddot{\theta}}{\partial \theta} \frac{d\theta}{dt} + \frac{\partial \ddot{\theta}}{\partial \ddot{\theta}_d} \frac{d\ddot{\theta}_d}{dt} + \frac{\partial \ddot{\theta}}{\partial \dot{\theta}_d} \frac{d\dot{\theta}_d}{dt} \right. \right. \right. \\ \left. \left. \left. + \frac{\partial \ddot{\theta}}{\partial \theta_d} \frac{d\theta_d}{dt} + \frac{\partial \ddot{\theta}}{\partial \tau_{\text{ext}}} \frac{d\tau_{\text{ext}}}{dt} \right] - K_1 \frac{d\dot{\theta}}{dt} - K_0 \frac{d\theta}{dt} + \frac{d\tau_{\text{ext}}}{dt} \right\} \right]_{w=M} \quad (3.34) \end{aligned}$$

ここで、 w と θ に着目して2つの部分に分けると、式(3.35)で表させる。

$$\begin{aligned} \frac{dV}{dt} = E \left[(\tau_c + \tau_{\text{ext}})^T \left\{ -K_2 \frac{\partial \ddot{\theta}}{\partial w} \right\} \frac{dw}{dt} \right]_{w=M} \\ + E \left[(\tau_c + \tau_{\text{ext}})^T \left\{ -K_2 \left(\frac{\partial \ddot{\theta}}{\partial \dot{\theta}} \frac{d\dot{\theta}}{dt} + \frac{\partial \ddot{\theta}}{\partial \theta} \frac{d\theta}{dt} + \frac{\partial \ddot{\theta}}{\partial \ddot{\theta}_d} \frac{d\ddot{\theta}_d}{dt} + \frac{\partial \ddot{\theta}}{\partial \dot{\theta}_d} \frac{d\dot{\theta}_d}{dt} + \frac{\partial \ddot{\theta}}{\partial \theta_d} \frac{d\theta_d}{dt} + \frac{\partial \ddot{\theta}}{\partial \tau_{\text{ext}}} \frac{d\tau_{\text{ext}}}{dt} \right) \right. \right. \\ \left. \left. - K_1 \frac{d\dot{\theta}}{dt} - K_0 \frac{d\theta}{dt} + \frac{d\tau_{\text{ext}}}{dt} \right\} \right]_{w=M} \quad (3.35) \end{aligned}$$

式(3.35)の右辺第2項は、 $w = M$ に固定した時の関数 L の時間平均を微分したものに等しいが、条件3により安定性が補償されているため、時間に依存せず一定になる。そして、その微分は0になることから式(3.35)は式(3.36)で表される。

$$\frac{dV}{dt} = E \left[(\tau_c + \tau_{\text{ext}})^T \left\{ -K_2 \frac{\partial \ddot{\theta}}{\partial w} \right\} \frac{dw}{dt} \right]_{w=M} \quad (3.36)$$

式(3.36)の $\partial \ddot{\theta} / \partial w$ を計算するために、式(3.26)を用いて式(3.33)を w で偏微分する。

$$\frac{\partial f}{\partial \ddot{\theta}} \frac{\partial \ddot{\theta}}{\partial w} - \frac{\partial \Phi}{\partial \ddot{\theta}} \frac{\partial \ddot{\theta}}{\partial w} - \frac{\partial \Phi}{\partial w} = -K_2 \frac{\partial \ddot{\theta}}{\partial w} \Leftrightarrow \frac{\partial \ddot{\theta}}{\partial w} = \left(\frac{\partial f}{\partial \ddot{\theta}} - \frac{\partial \Phi}{\partial \ddot{\theta}} + K_2 \right)^{-1} \frac{\partial \Phi}{\partial w} \quad (3.37)$$

式(3.27)を式(3.26)に代入すると式(3.38)が得られる。

$$\begin{aligned} \frac{dV}{dt} = E \left[(\tau_c + \tau_{\text{ext}})^T \left\{ -K_2 \left(\frac{\partial f}{\partial \ddot{\theta}} - \frac{\partial \Phi}{\partial \ddot{\theta}} + K_2 \right)^{-1} \frac{\partial \Phi}{\partial w} \right\} \frac{dw}{dt} \right]_{w=M} \\ = E \left[-(\tau_c + \tau_{\text{ext}})^T K_2 \left(\frac{\partial f}{\partial \ddot{\theta}} - \frac{\partial \Phi}{\partial \ddot{\theta}} + K_2 \right)^{-1} \frac{\partial \Phi}{\partial w} \eta \left(\frac{\partial \Phi}{\partial w} \right)^T (\tau_c + \tau_{\text{ext}}) \right]_{w=M} \quad (3.38) \end{aligned}$$

条件3を考えると、次の行列は準正定であることがわかる。

$$K_2 \left(\frac{\partial f}{\partial \ddot{\theta}} - \frac{\partial \Phi}{\partial \ddot{\theta}} + K_2 \right)^{-1} \frac{\partial \Phi}{\partial w} \eta \left(\frac{\partial \Phi}{\partial w} \right)^T \quad (3.39)$$

よって、 $dV/dt \leq 0$ が導かれる。符号は、 $\tau_{\text{imag}} \neq 0$ の状態で $\partial \Phi / \partial w \neq 0$ すなわち、ローカル ミニマムに陥らなければ $\tau_{\text{imag}} = 0$ の時だけ成立する。そして、平均化方程式の解は実際の方程式の解を近似するため $w \rightarrow \hat{w}$ が得られる。また、これはフィードバック誤差学習により NNFC が制御対象の逆動特性モデルに形成されたと言える。以上が、五味と川人による逆モデル学習での証明であり、非線形レギュレータ学習の理論についても、次のように述べている。

$$R(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) = \tau \quad (3.40)$$

まず、制御対象の動特性を加速度に依存する項とその他の項に分けて式(3.40)のように表す。 R, N は未知の非線形項を含む制御対象の動特性によって決まる項であり、多くのロボットなどの動特性では式(3.40)のように表される。CFC は式(3.15)のように設定して、NNFC はフィードバック信号 $\theta, \dot{\theta}$ とフィードフォワード信号 $\ddot{\theta}_d, \dot{\theta}_d, \theta_d$ を入力とする。そして、シナプス荷重 w を含めて、式(3.41)のように表す。

$$\tau_n = \Phi(\ddot{\theta}_d, \dot{\theta}_d, \theta_d, \dot{\theta}, \theta, w) \quad (3.41)$$

五味と川人は、図 3.15 のような式(3.40)と式(3.41)を用いた制御系の構成を示した。また、この系の全体の動特性を次のように述べている。

$$\begin{aligned} \tau &= \tau_c + \tau_n \\ \Leftrightarrow K_2(\ddot{\theta}_d - \ddot{\theta}) + K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta) + \Phi - R(\theta)\ddot{\theta} - N(\theta, \dot{\theta}) &= 0 \end{aligned} \quad (3.42)$$

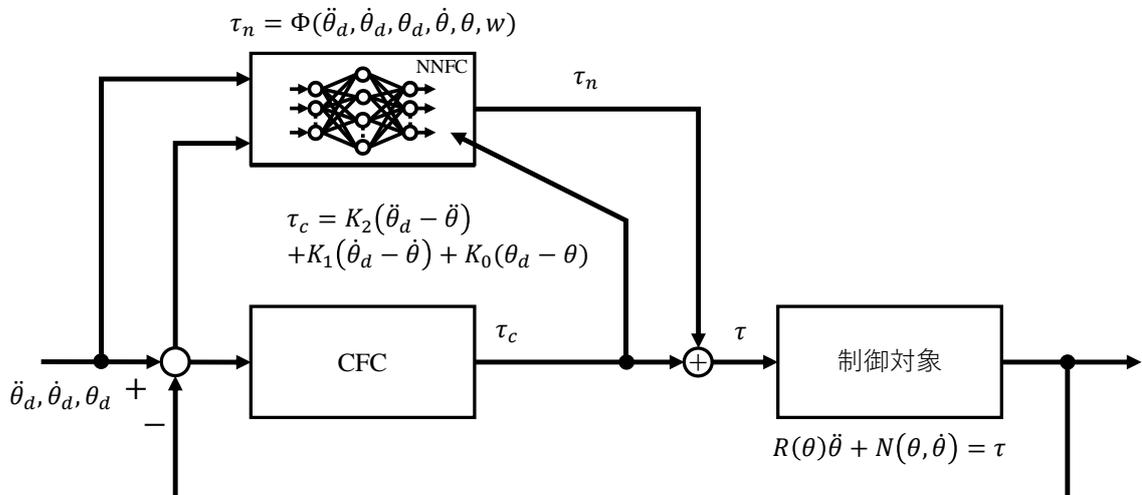


図 3.15 非線形レギュレータ学習 [33]

五味裕章, 川人光男: フィードバック誤差学習による閉ループシステムの学習制御, システム制御情報学会論文誌, 4 巻, 1 号, Fig.2 (1991) より引用

ここで、NNFC で実現する関数 Φ が式(3.43)のような場合を考える。

$$\Phi_d = N(\theta, \dot{\theta}) + R(\theta)\ddot{\theta}_d + R(\theta)K_2^{-1}\left(K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta)\right) \quad (3.43)$$

そのような場合、式(3.42)は式(3.44)で表される。

$$(R(\theta) + K_2)(\ddot{\theta}_d - \ddot{\theta}) + K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta) + R(\theta)K_2^{-1}\left(K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta)\right) = 0 \quad (3.44)$$

そして、単位行列 I を用いて以下の式のように表される。

$$(I + R(\theta)K_2^{-1})\left(K_2(\ddot{\theta}_d - \ddot{\theta}) + K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta)\right) = 0 \quad (3.45)$$

式(3.45)で $\xi = \theta_d - \theta$ とおき、 $I + R(\theta)K_2^{-1} \neq 0$ であることを用いると式(3.46)で表される。

$$K_2\ddot{\xi} + K_1\dot{\xi} + K_0\xi = 0 \quad (3.46)$$

これは、目標値 θ_d と実際の出力 θ の誤差の自由応答は設定した CFC の出力が 0 となるように動くという意味になる。五味と川人は、CFC からの出力を NNFC の学習のための誤差信号として学習を進めた時は、式(3.46)の応答に一致することを次のように述べている。

$$P = \Phi - \left[N(\theta, \dot{\theta}) + R(\theta)\ddot{\theta}_d + R(\theta)K_2^{-1}\left(K_1(\dot{\theta}_d - \dot{\theta}) + K_0(\theta_d - \theta)\right) \right] = -(I + R(\theta)K_2^{-1})\tau_c \quad (3.47)$$

関数 P を式(3.47)として、フィードバック誤差学習を行うと、NNCF の学習則は式(3.48)で表される。また、式(3.42)と式(3.48)の確率微分方程式と平均化方程式に対するリアプノフ関数の候補 J を式(3.49)とすると、逆モデル学習で述べた証明と同様の手順で $dV/dt \leq 0$ が導かれる。

$$\frac{dw}{dt} = \eta \left(\frac{\partial \Phi}{\partial w} \right)^T \tau_c \quad (3.48)$$

$$J = E\eta \left[\frac{1}{2} P^T P \right]_{w=M} \quad (3.49)$$

五味と川人は、神経回路モデルをフィードバックコントローラに利用するときの学習法を提案し、この学習法の収束性を確率微分方程式の平均化とリアプノフの方法を用いて証明した。

3.4 あとがき

本章では、まず、ニューラルネットワークの仕組みについて、その構造と誤差逆伝播のアルゴリズムを説明した。次に、ニューラルネットワーク制御の仕組みについて、先行研究のニューロコントロールと適応制御 [5] [6] の内容および先行研究のフィードバック誤差学習による閉ループシステムの学習制御 [32] [33] の内容を説明した。これらの説明によって、制御対象とニューラルネットワークとの間にはどのような関係があるか明らかにされた。

第4章 動的モデルを用いた開発環境

4.1 はしがき

本章では、まず、制御対象のサーボモータの位置決め制御について、その構成とその制御系の動的モデルを 4.2 節で示す。次に、サーボモータの位置決め制御のシミュレーションを行うために作成した制御系の解析モデルを 4.3 節で示して、最後に、その解析モデルを用いたシミュレーションの具体的な解析方法も説明する。

4.2 サーボシステムの動的モデルと構成

図 4.1 のように、制御対象のサーボモータの位置決め制御の構成は、DC サーボモータ、カップリング、負荷とする DC モータから直列で接続される。ここで、図 3.4 のニューラルネットワーク制御を用いたサーボシステムの構成で示した x 、 \dot{x} 、 \ddot{x} を θ_r 、 $\dot{\theta}_r$ 、 $\ddot{\theta}_r$ に置き換えると、ニューラルネットワーク制御を用いたサーボ制御の構成は、図 4.2 のように示される。

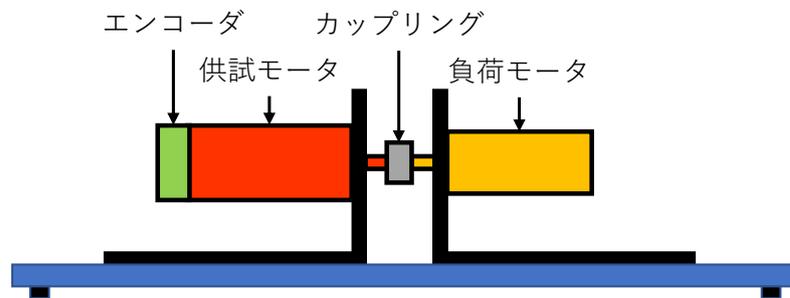


図 4.1 DC サーボモータの位置決め制御の構成

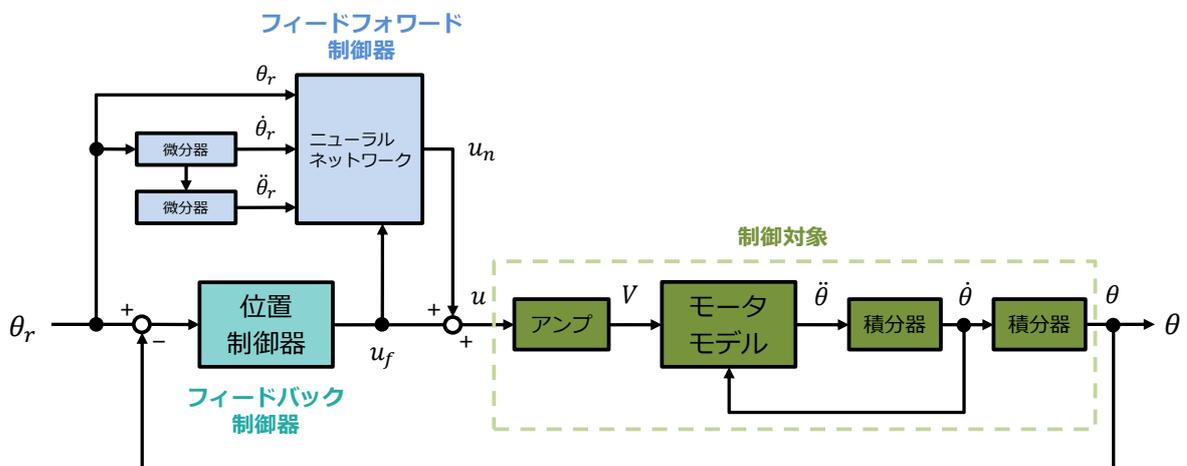


図 4.2 ニューラルネットワーク制御を用いたサーボ制御の構成

図 4.2 にあるモータの動的モデルは、式(4.1)で示す電氣的エネルギーと機械的エネルギーの変換に基づいた電機子電圧 V と回転位置 θ との数学的な関係である。また、制御器はフィードバック制御器にニューラルネットワークから構成されるフィードフォワード制御器と並列に接続したものである。加えて、電機子電圧 V と回転角速度 $\dot{\theta}$ の関係は、トルク定数 K_t が逆起電力定数 K_e と等しい関係を用いて、式(4.2)で示される。そして、回転角加速度は、式(4.1)と式(4.2)の連立によって求められ、式(4.3)に示される。ここで、 R は相間の端子抵抗、 I は電機子電流、 J はモータの慣性イナーシャ J_M と負荷イナーシャ J_L を合成したイナーシャである。

$$\tau = K_t I = J \frac{d^2\theta}{dt^2} = J\ddot{\theta} = (J_M + J_L)\ddot{\theta} \quad (4.1)$$

$$V = RI + K_e \frac{d\theta}{dt} = RI + K_e \dot{\theta} \quad (4.2)$$

$$\frac{d^2\theta}{dt^2} = \frac{K_t}{JR} \left(V - K_e \frac{d\theta}{dt} \right) = \frac{K_t}{JR} (V - K_e \dot{\theta}) \quad (4.3)$$

本研究のシミュレーションにおいて、DC サーボモータはマクソンの EC-max 22 を扱った。そのパラメータは表 4.1 に示される。また、シミュレーションの開発環境は表 4.2 に示される。

表 4.1 ブラシレス DC モータのパラメータ (EC-max22)

記号	名称	値
V_n	定格電圧	24 V
I_n	定格電流	0.657 A
N_n	定格回転	8250 rpm
T_n	定格トルク	10.8 mNm
R	電機子抵抗	12.4 Ω
K_t	トルク定数	18.1 mNm/A
J_M	モータイナーシャ	2.25×10^{-7} kgm ²

表 4.2 シミュレーションの開発環境

OS	Windows 10 Home (64bit)
CPU	Intel Core i7-7700
GPU	NVIDIA GeForce GTX1050
Memory	32GB (16GB + 16GB)
Software	MathWorks MATLAB R2020a
Toolbox	MATLAB 9.8, Simulink 10.1

4.3 シミュレーションのモデル

図 4.3 に MATLAB/Simulink で構成したシミュレーションモデルを示す。このモデルにおいて、解析するためのサンプリングタイムは 1 ms であり、微積分の解析法は 4 次ルンゲクッタ法である。 θ_r のブロックは、目標位置の生成器であり、回転角 ± 90 度の振幅、1 Hz の周波数、0 deg の位相差の正弦波である角度軌道を生じる。フィードバック制御器は PID 制御を用いる。PID 制御のゲインは極配置法で算出した値で設定される。増幅器 K はモータモデルの前に接続されて、 θ_r 、 $\dot{\theta}_r$ 、 $\ddot{\theta}_r$ はニューラルネットワーク補償器のブロックに接続される。また、モータの出力である角度 θ と角速度 $\dot{\theta}$ は、角加速度 $\ddot{\theta}$ の積分によって求められる。

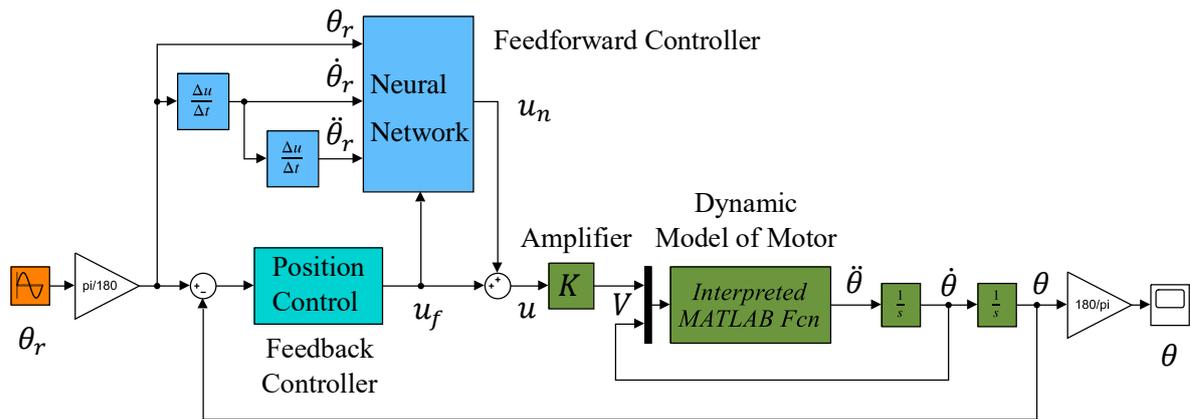


図 4.3 ニューラルネットワーク制御を用いたサーボシステムのシミュレーションモデル

このシミュレーションモデルを用いて、サーボモータの位置決め制御のシミュレーションを行う。ここで、増幅器 K の設定については、 $K = 1$ 、すなわち、制御対象への操作量 u に 1 倍したものを電機子電圧 V とする。また、負荷イナーシャ J_L やニューラルネットワークの学習率と慣性係数などの設定については、次章 5.4 節の 5.4.2 項で説明する。

4.4 あとがき

本章では、まず、制御対象のサーボモータの位置決め制御について、その構成とその制御系の動的モデルを示した。次に、サーボモータの位置決め制御のシミュレーションを行うために作成した制御系の解析モデルを示した。最後に、その解析モデルを用いたシミュレーションの具体的な解析について、サンプリングタイムは 1ms と設定して、微積分の解析法は 4 次ルンゲクッタ法を適用した。また、目標位置の生成器は、回転角 ± 90 度の振幅、1 Hz の周波数、0 deg の位相差の正弦波というように設定して、フィードバック制御器は PID 制御を用いた。PID 制御のゲインは、第 2 章で説明した極配置法に基づいて算出した値を設定した。

従来のオンライン学習法のフローチャートを図 5.1 に示す。 $\Delta v(j,k)(N)$ は $v(j,k)(N)$ の差分、 $\Delta w(i,j)(N)$ は $w(i,j)(N)$ の差分、 N は重み $v(j,k)(N)$ 、 $w(i,j)(N)$ を計算する回数、 η は学習率、 α は η の修正量を調整するための慣性係数である。まず、 $\Delta v(j,k)(N)$ 、 $v(j,k)(N)$ 、 $w(i,j)(N)$ 、 $y(j)$ 、 $z(k)$ はゼロの値で初期化されて、 $\Delta w(i,j)(N)$ はランダムな値で初期化される。次に、位置、速度、加速度の指令値 x 、 \dot{x} 、 \ddot{x} はニューラルネットワークの入力とする。誤差信号 E_n はフィードバック制御器の出力 $u_f(k)$ から取得される。さらに、式(5.1)から式(5.6)によって、 $\Delta v(j,k)(N)$ 、 $v(j,k)(N)$ 、 $\Delta w(i,j)(N)$ 、 $w(i,j)(N)$ 、 $y(j)$ 、 $z(k)$ が計算される。最後に、 $z(k)$ はニューラルネットワークの出力 $u_n(k)$ とする。

$$\Delta v(j,k)(N) = -\eta \frac{\partial E_n}{\partial v(j,k)} + \alpha \Delta v(j,k)(N-1) \quad (5.1)$$

$$v(j,k)(N) = v(j,k)(N-1) + \Delta v(j,k)(N) \quad (5.2)$$

$$\Delta w(i,j)(N) = -\eta \frac{\partial E_n}{\partial w(i,j)} + \alpha \Delta w(i,j)(N-1) \quad (5.3)$$

$$w(i,j)(N) = w(i,j)(N-1) + \Delta w(i,j)(N) \quad (5.4)$$

$$y(j) = f_y(w(i,j)(N), x(i)) = \frac{2}{1 + e^{-\sum_{i=1}^c (w(i,j)(N))x(i)}} - 1 \quad (5.5)$$

$$u_n(k) = z(k) = f_z(v(j,k)(N), y(j)) = \frac{2}{1 + e^{-\sum_{j=1}^b (v(j,k)(N))y(j)}} - 1 \quad (5.6)$$

5.2.2 オフライン学習法のアルゴリズム

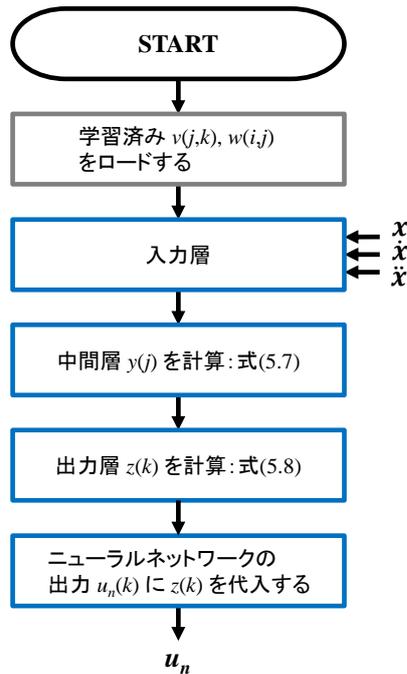


図 5.2 オフライン学習法のフローチャート

従来のオフライン学習法のフローチャートを図 5.2 に示す。まず、十分なオンライン学習で獲得した重み $w(i, j)$ 、 $v(j, k)$ はニューラルネットワークにロードされる。次に、位置、速度、加速度の指令値 x 、 \dot{x} 、 \ddot{x} はニューラルネットワークの入力とする。また、式(5.7)と式(5.8)によって、 $y(j)$ と $z(k)$ を計算する。最後に、 $z(k)$ はニューラルネットワークの出力 $u_n(k)$ とする。この学習法では、重み $w(i, j)$ 、 $v(j, k)$ は更新されない。

$$y(j) = f_y(w(i, j), x(i)) = \frac{2}{1 + e^{-\sum_{i=1}^c w(i, j)x(i)}} - 1 \quad (5.7)$$

$$u_n(k) = z(k) = f_z(v(j, k), y(j)) = \frac{2}{1 + e^{-\sum_{j=1}^b v(j, k)y(j)}} - 1 \quad (5.8)$$

5.2.3 オンライン・オフライン融合型学習法のアルゴリズム

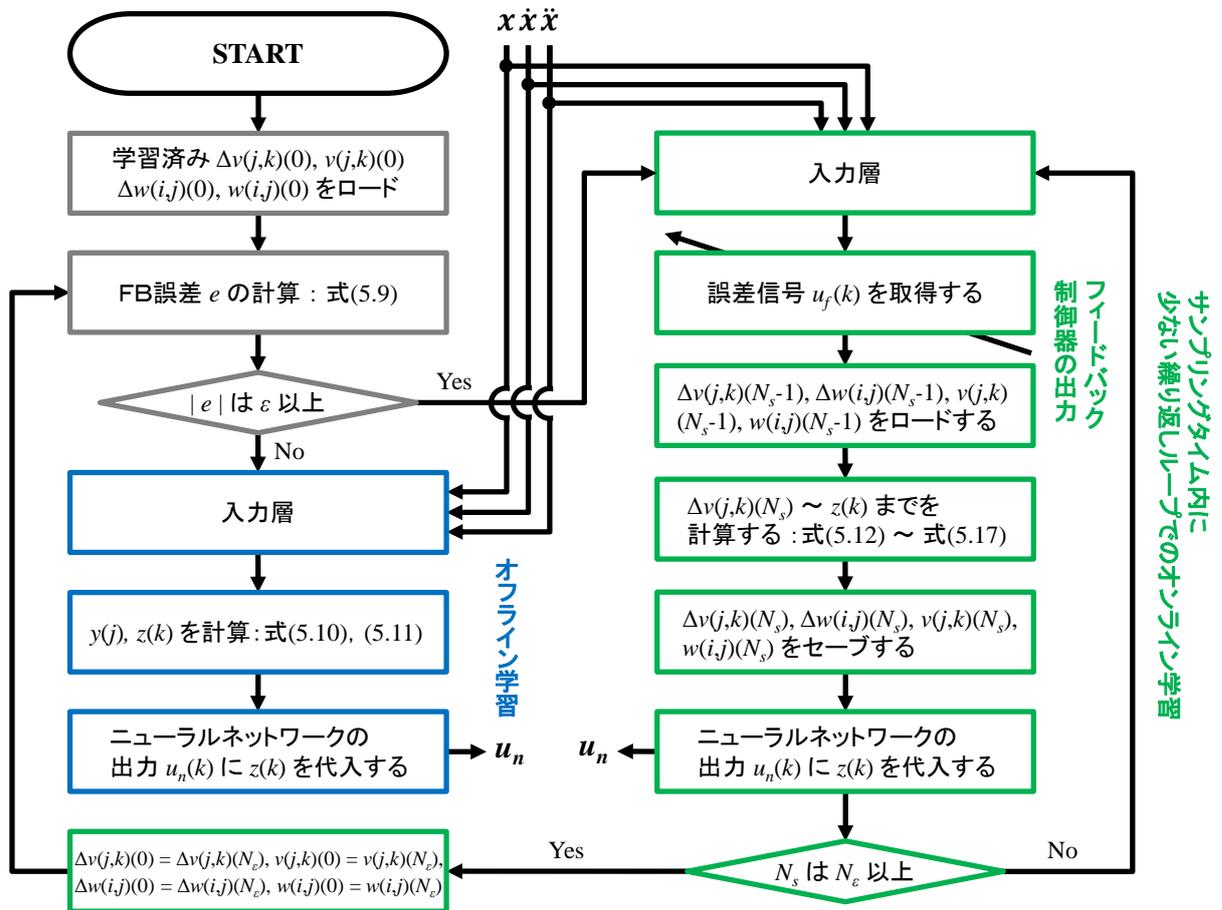


図 5.3 オンライン・オフライン融合型学習法のフローチャート

まず、十分なオンライン学習で獲得した重みはニューラルネットワークにロードされて、 $\Delta v(j, k)(0)$ 、 $v(j, k)(0)$ 、 $\Delta w(i, j)(0)$ 、 $w(i, j)(0)$ とする。但し、本論文のサーボモータの位置決め

制御のシミュレーション結果によって、重み $v(j, k)$ はゼロの値で初期化すると、より良い結果が得られるため、本研究では重み $v(j, k)$ はゼロの値で初期化する。また、位置、速度、加速度の指令値 x 、 \dot{x} 、 \ddot{x} はニューラルネットワークの入力である。加えて、誤差信号 E_n はフィードバック制御器の出力 $u_f(k)$ から取得される。

次に、フィードバック誤差 e は式(5.9)で計算される。計算した誤差 e が閾値 ε を超えなければ、オフライン学習法は行われて、 $y(j)$ と $z(k)$ は式(5.10)と式(5.11)で計算される。そして、 $z(k)$ はニューラルネットワークの出力 $u_n(k)$ とする。もしフィードバック誤差 e が閾値 ε を超えれば、サンプリングタイム内において少ない繰り返しループでオンライン学習を行う。各ループで、ニューラルネットワークの入力は位置、速度、加速度の指令値 x 、 \dot{x} 、 \ddot{x} で与えられる。加えて、誤差信号 E_n はフィードバック制御器の出力 $u_f(k)$ から取得される。ニューラルネットワークに $\Delta v(j, k)(N_s - 1)$ 、 $v(j, k)(N_s - 1)$ 、 $\Delta w(i, j)(N_s - 1)$ 、 $w(i, j)(N_s - 1)$ はロードされて、式(5.12)から式(5.17)によって $\Delta v(j, k)(N_s)$ 、 $v(j, k)(N_s)$ 、 $\Delta w(i, j)(N_s)$ 、 $w(i, j)(N_s)$ 、 $y(j)$ 、 $z(k)$ は計算される。 $z(k)$ の値は $u_n(k)$ に与える。

その次に、もしサンプリングタイム内の繰り返しループの回数 N_s が 閾値 N_ε になっていなければ、これらの計算は繰り返し行われる。もしサンプリングタイム内の繰り返しループの回数 N_s が閾値 N_ε になれば、オンライン学習法を終了させて、計算がオフライン学習法に戻る。その時に、獲得した重み $\Delta v(j, k)(N_\varepsilon)$ 、 $v(j, k)(N_\varepsilon)$ 、 $\Delta w(i, j)(N_\varepsilon)$ 、 $w(i, j)(N_\varepsilon)$ はオフライン学習法の学習済重み $\Delta v(j, k)(0)$ 、 $v(j, k)(0)$ 、 $\Delta w(i, j)(0)$ 、 $w(i, j)(0)$ とする。

$$e = x - y \quad (5.9)$$

$$y(j) = f_y(w(i, j)(N_s), x(i)) = \frac{2}{1 + e^{-\sum_{i=1}^c (w(i, j)(N_s))x(i)}} - 1 \quad (5.10)$$

$$u_n(k) = z(k) = f_z(v(j, k)(N_s), y(j)) = \frac{2}{1 + e^{-\sum_{j=1}^b (v(j, k)(N_s))y(j)}} - 1 \quad (5.11)$$

$$\Delta v(j, k)(N_s) = -\eta \frac{\partial E_n}{\partial v(j, k)} + \alpha \Delta v(j, k)(N_s - 1) \quad (5.12)$$

$$v(j, k)(N_s) = v(j, k)(N_s - 1) + \Delta v(j, k)(N_s) \quad (5.13)$$

$$\Delta w(i, j)(N_s) = -\eta \frac{\partial E_n}{\partial w(i, j)} + \alpha \Delta w(i, j)(N_s - 1) \quad (5.14)$$

$$w(i, j)(N_s) = w(i, j)(N_s - 1) + \Delta w(i, j)(N_s) \quad (5.15)$$

$$y(j) = f_y(w(i, j)(N_s), x(i)) = \frac{2}{1 + e^{-\sum_{i=1}^c (w(i, j)(N_s))x(i)}} - 1 \quad (5.16)$$

$$u_n(k) = z(k) = f_z(v(j, k)(N_s), y(j)) = \frac{2}{1 + e^{-\sum_{j=1}^b (v(j, k)(N_s))y(j)}} - 1 \quad (5.17)$$

このような学習法は、オフライン学習法の良い応答性とオンライン学習法の優れているロバスト性を融合させられて、負荷変動と外乱の対応および実時間制御が可能であるニューラルネットワーク制御を実現できる。

5.3 学習法のまとめ

表 5.1 学習法の比較

	オンライン学習	オフライン学習	オンライン・オフライン融合型学習
応答性	悪い	良い	良い
ロバスト性	良い	悪い	良い

表 5.1 に 3 つの学習法の比較を示す。3 つの学習法の比較は、以下の通りにまとめられる。

1. オンライン学習法では、負荷変動や外乱が発生すると、それらの変化が式(3.5)に示されるニューラルネットワークの誤差信号 E_n へ反映されるため、負荷変動や外乱に対応することができて、ロバスト性が良い。ただし、サンプリングタイム内で 1 回しか式(5.1)から式(5.6)の計算がされないため、整定時間が長くて、応答性が悪い。
2. オフライン学習法では、式(5.7)と式(5.8)しか計算されないため、整定時間が短くて、応答性が良い。ただし、負荷変動や外乱が学習に反映されないため、ロバスト性が悪い。
3. 提案法では、フィードバック誤差が閾値を超えなければ、オフライン学習法と同じく、式(5.10)と式(5.11)の少ない計算が実行されるため、応答性が良い。また、フィードバック誤差が閾値を超えれば、式(5.12)から式(5.17)に示されるように、サンプリングタイム内で少ない繰り返しループのオンライン学習法を行うため、負荷変動や外乱に対応できて、ロバスト性も良い。

5.4 結果と考察

本節では、位置決め制御のシミュレーション結果を 5.4.1 項から 5.4.3 項に示す。5.4.1 項ではオンライン学習法を用いて、十分なオンライン学習によって学習済重みを獲得する。5.4.2 項では負荷イナーシャ変更前後において、オンライン学習法、オフライン学習法、提案法を用いた場合の位置決め制御の応答性とロバスト性に関するシミュレーション結果を示す。5.4.3 項では、目標軌道の周波数が変更された場合のシミュレーション結果を示す。そして、5.4.4 項では、シミュレーションから分かった結果と考察をまとめる。

5.4.1 オンライン学習法を用いた学習済重みの獲得

図 5.4 と図 5.5 は、PID 制御とオンライン学習法 ($\eta = 0.004$ 、 $\alpha = 0.001$) をそれぞれ用いた場合の回転角度 θ の結果およびフィードバック誤差 e を示している。負荷イナーシャの値は $J_L = J_M = 2.25 \times 10^{-7} \text{kgm}^2$ と設定した。そして、40 秒以降には学習効果がそれ以上増加しなかったため、40 秒の時点で獲得した重みを学習済重みと決めた。

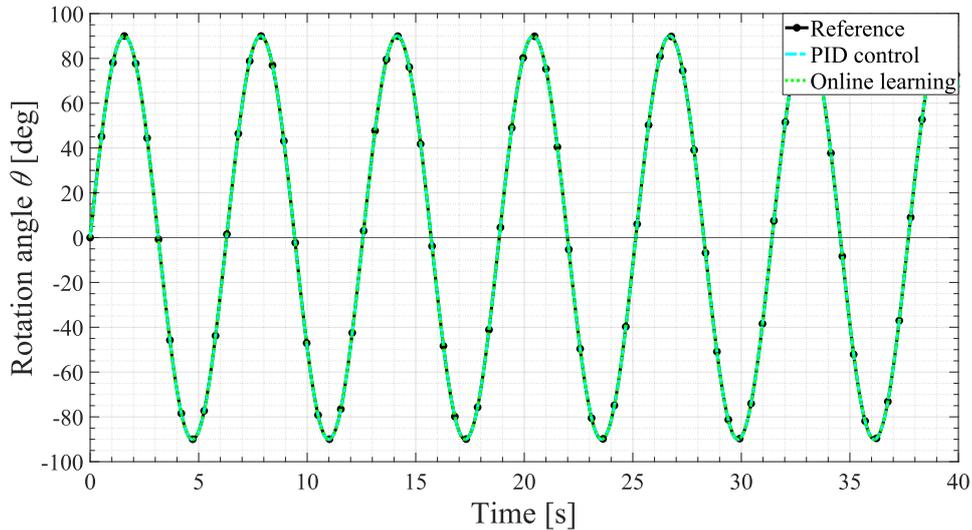


図 5.4 PID 制御とオンライン学習法の応答

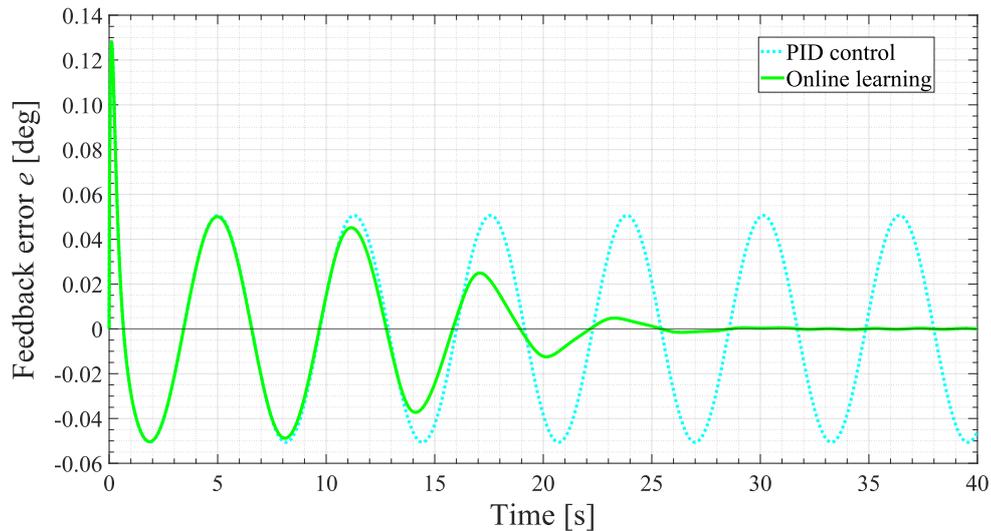


図 5.5 PID 制御とオンライン学習法それぞれのフィードバック誤差

5.4.2 負荷イナーシャ変更前後の応答性とロバスト性

図 5.6 は、負荷イナーシャ変更前後のモータの目標と応答 θ を示している。オンライン学習法のパラメータは $\eta = 0.004$ 、 $\alpha = 0.001$ であり、提案法のパラメータは $\eta = 0.004$ 、 $\alpha = 0.001$ 、 $N_e = 10$ である。また、シミュレーション開始から 15 秒後に負荷イナーシャ値の大きさをステップ状として $J_L = J_M = 2.25 \times 10^{-7} \text{ kgm}^2$ から $J_L = 10 J_M = 2.25 \times 10^{-6} \text{ kgm}^2$ へ増大させる。

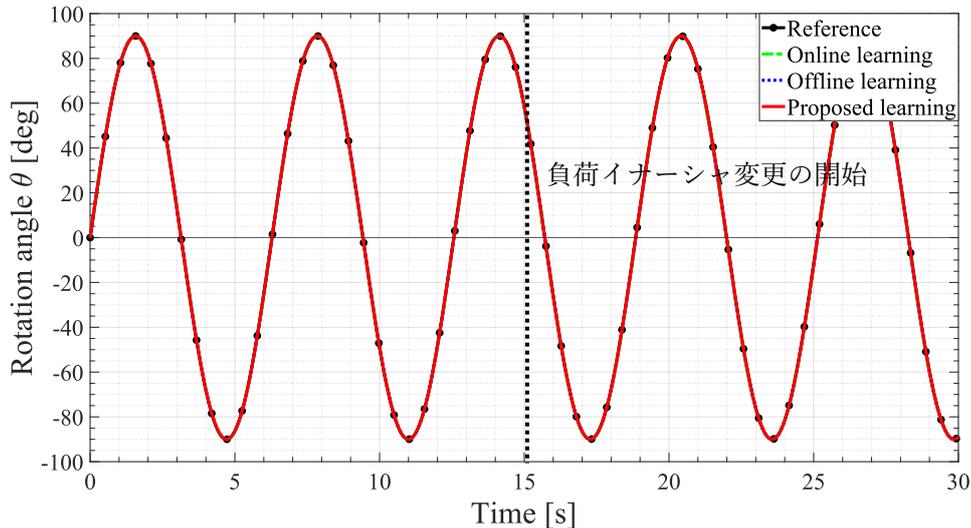
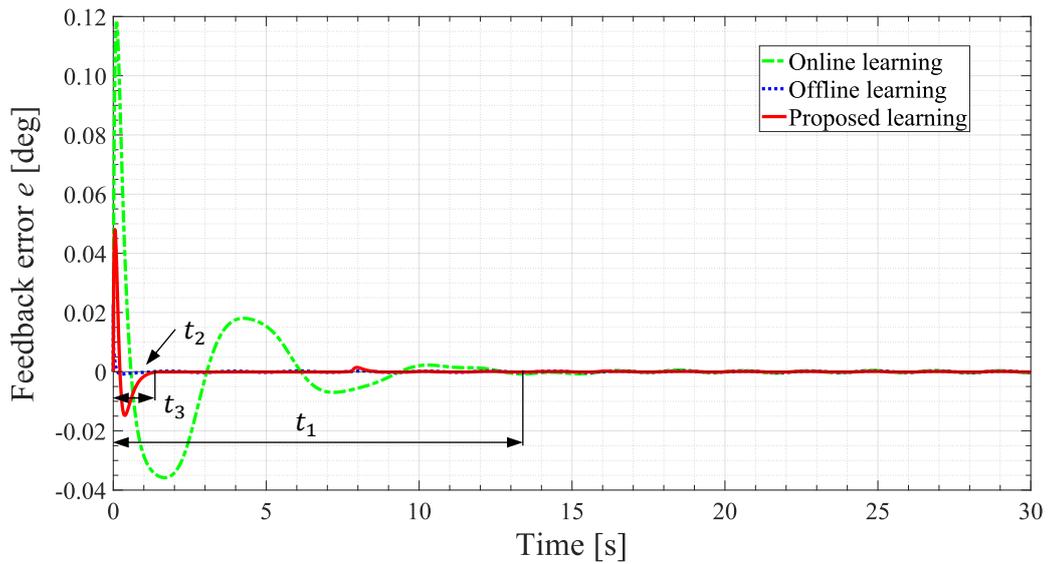
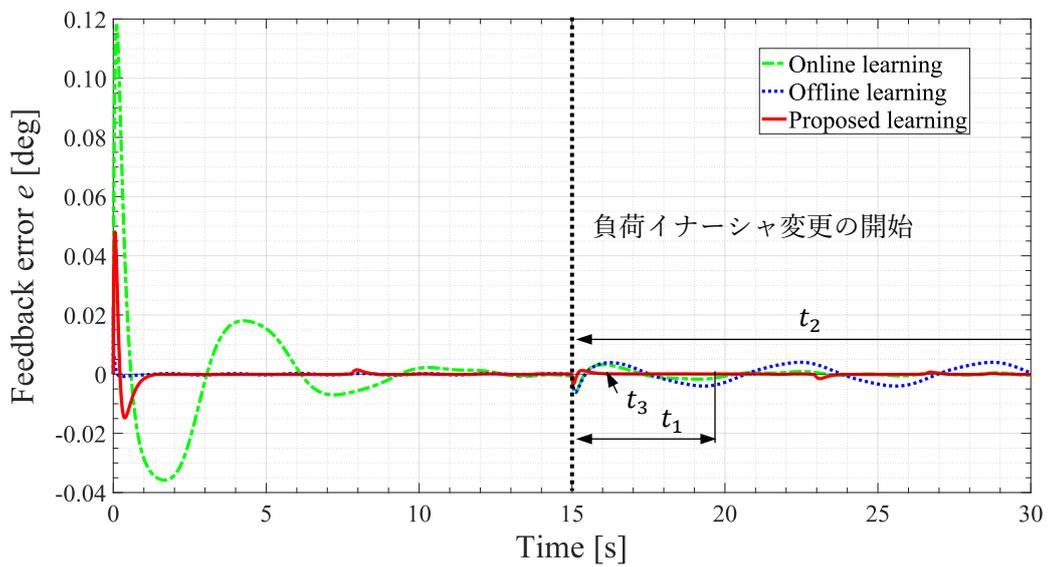


図 5.6 オンライン学習法、オフライン学習法、オンライン・オフライン融合型学習法それぞれの目標と応答（負荷イナーシャの変更前後）

図 5.7 (a) と 5.7 (b) それぞれは、負荷イナーシャ変更前後のフィードバック誤差 e の結果を示している。図 5.8 (a) と 5.8 (b) それぞれは、負荷イナーシャ変更前後のニューラルネットワーク補償器の出力 u_n の結果を示している。図 5.9 (a) は、図 5.8 (a) の 0 秒から 14 秒までの拡大図であり、図 5.9 (b) は、図 5.8 (b) の 14 秒から 20 秒までの拡大図である。また、図 5.10 (a) と 5.10 (b) それぞれは、負荷イナーシャ変更前後のフィードバック制御器の出力 u_f である。式(3.5)によって、図 5.10 はニューラルネットワーク補償器の誤差信号 E_n の結果とみなせる。

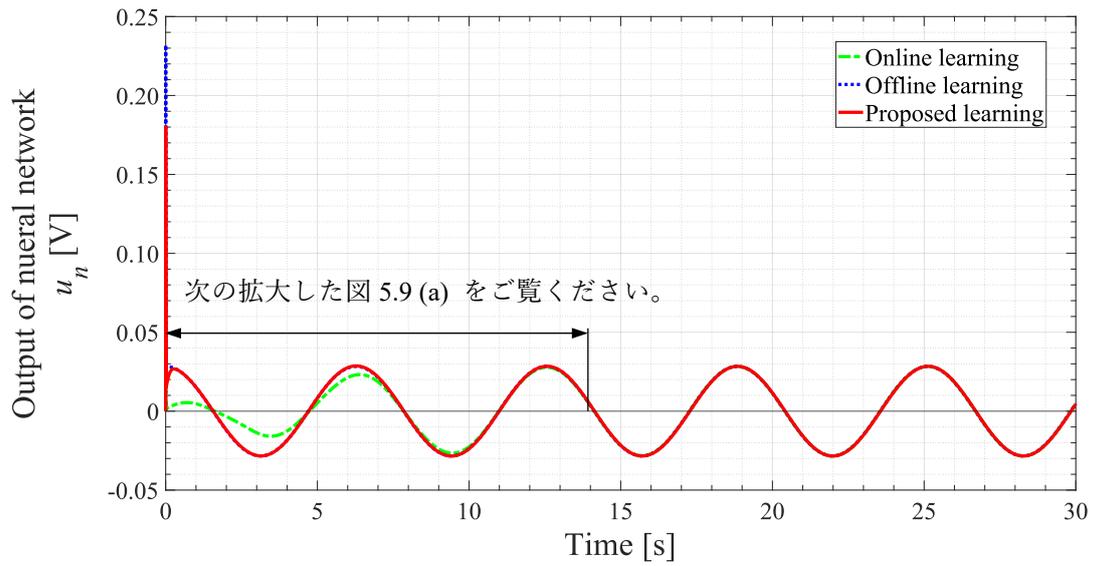


(a) 負荷イナーシャ変更前

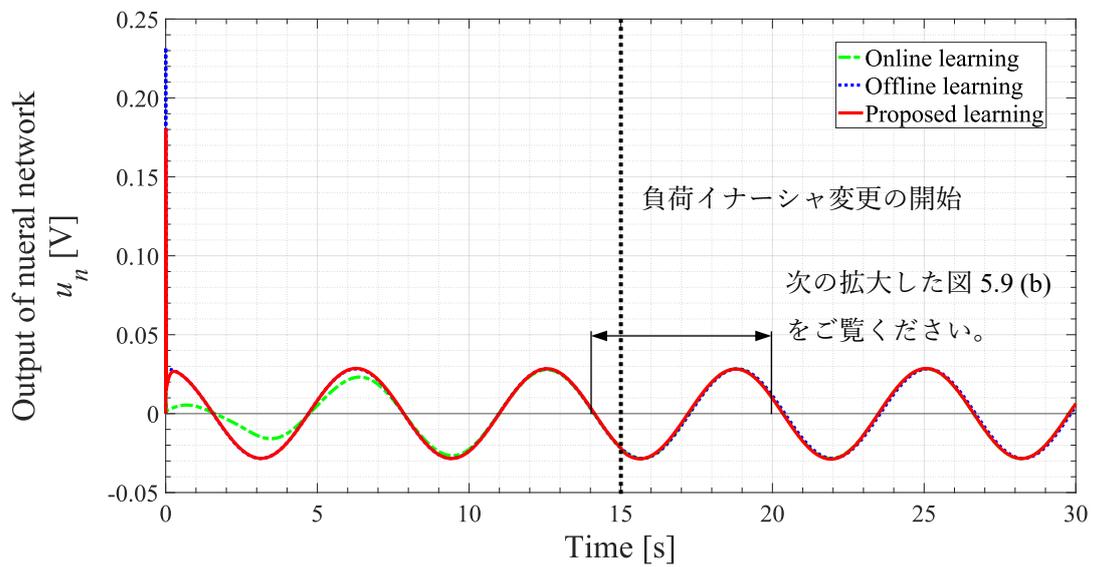


(b) 負荷イナーシャ変更後

図 5.7 オンライン学習法、オフライン学習法、オンライン・オフライン融合型学習法それぞれのフィードバック誤差（負荷イナーシャの変更前と変更後）

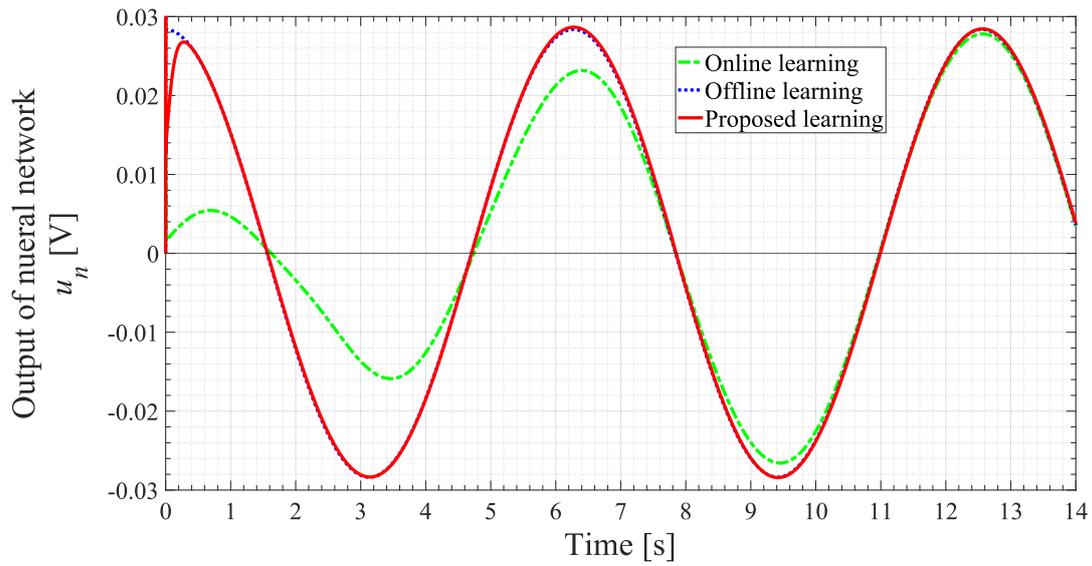


(a) 負荷イナーシャ変更前

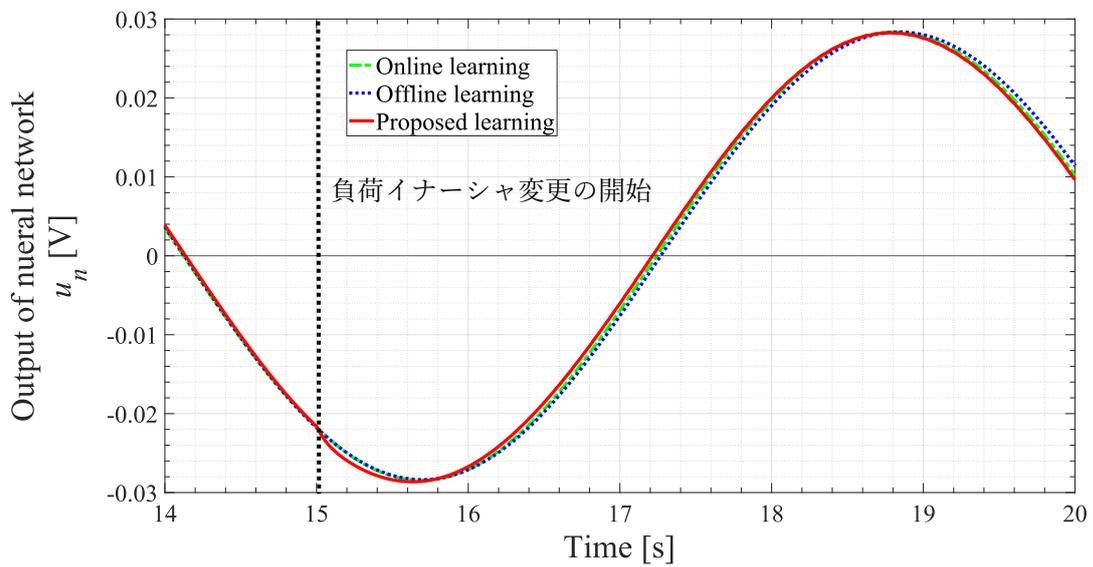


(b) 負荷イナーシャ変更後

図 5.8 オンライン学習法、オフライン学習法、オンライン・オフライン融合型学習法それぞれのニューラルネットワーク出力（負荷イナーシャの変更前と変更後）

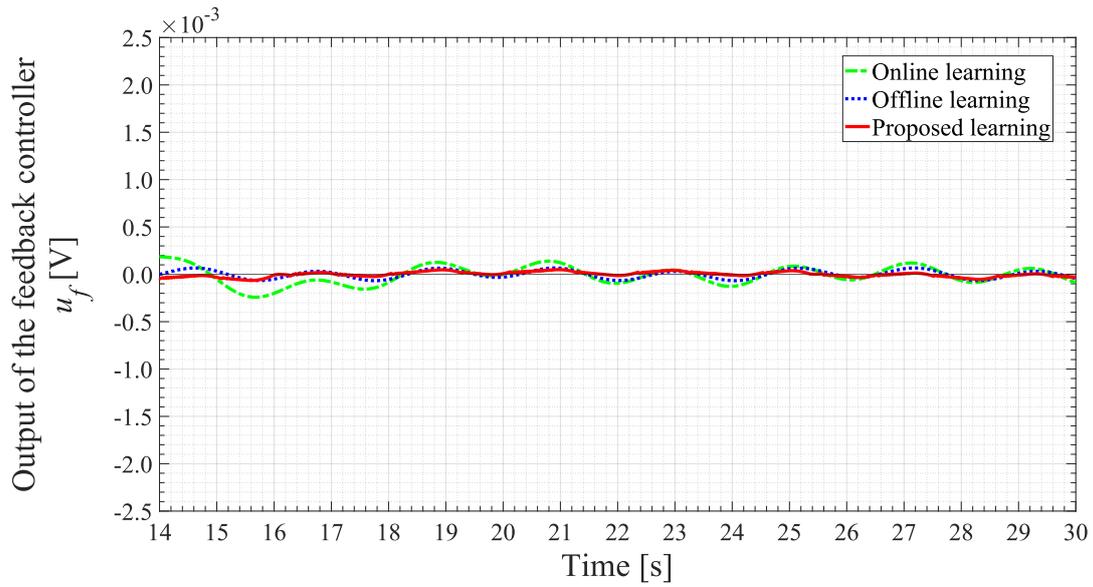


(a) 負荷イナーシャ変更前

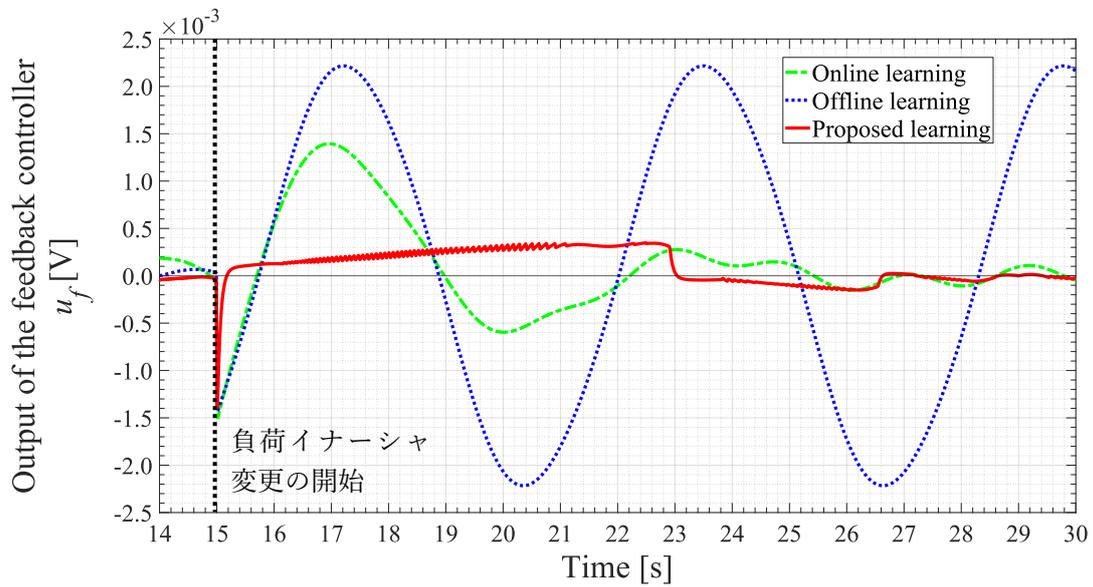


(b) 負荷イナーシャ変更後

図 5.9 オンライン学習法、オフライン学習法、オンライン・オフライン融合型学習法それぞれのニューラルネットワーク出力 (図 5.8 の拡大図)



(a) 負荷イナーシャ変更前



(b) 負荷イナーシャ変更後

図 5.10 オンライン学習法、オフライン学習法、オンライン・オフライン融合型学習法それぞれのフィードバック制御器の出力（負荷イナーシャの変更前と変更後）

5.4.3 目標軌道の周波数変更前後の応答性とロバスト性

ここで、目標軌道 θ_r の周波数を 1.0 Hz から 0.5 Hz、3.0 Hz に変更したフィードバック誤差 e の結果は、図 5.11 と図 5.12 にそれぞれ示される。その場合に、提案法の学習に関連するパラメータは $\eta = 0.005$ 、 $\alpha = 0.001$ 、 $N_\varepsilon = 3$ である。

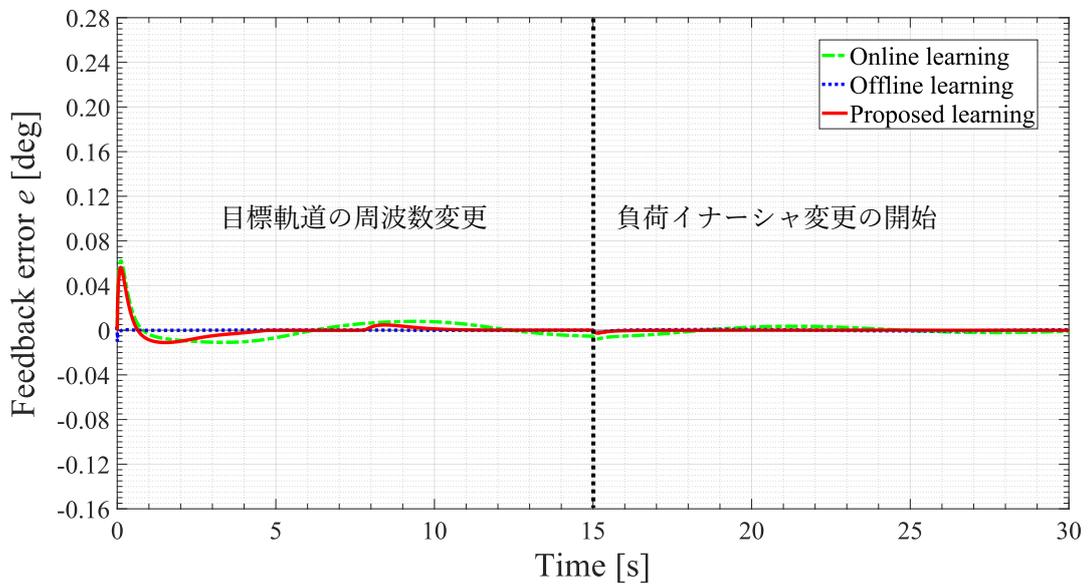


図 5.11 目標軌道の周波数を 0.5 Hz に変更した場合のそれぞれのフィードバック誤差 (負荷イナーシャの変更前後)

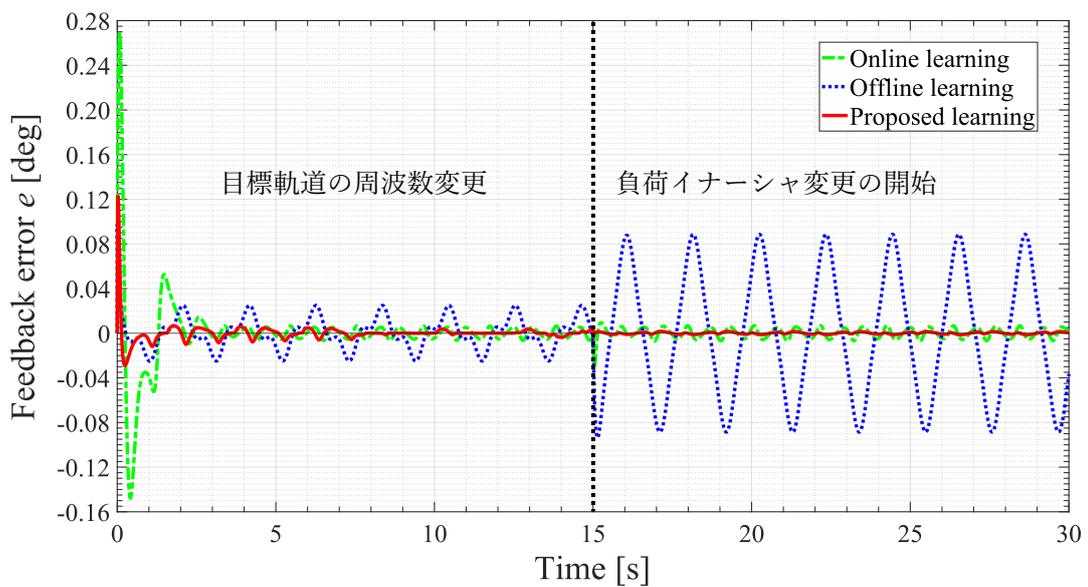


図 5.12 目標軌道の周波数を 3.0 Hz に変更した場合のそれぞれのフィードバック誤差 (負荷イナーシャの変更前後)

5.4.4 応答性とロバスト性に関する考察

(A) 負荷イナーシャ変更前の結果と考察は、以下の通りである。

1. 整定時間と定常偏差

オンライン学習法の整定時間は、図 5.7 (a) の t_1 であり、13.38 秒であった。

その時の定常偏差の最大値は $\pm 0.0008 \text{ deg}$ であった。

オフライン学習法の整定時間は、図 5.7 (a) の t_2 であり、1.05 秒であった。

その時の定常偏差の最大値は $\pm 0.0002 \text{ deg}$ であった。

提案法の整定時間は、図 5.7 (a) の t_3 であり、1.41 秒であった。

その時の定常偏差の最大値は $\pm 0.0001 \text{ deg}$ であった。

整定時間について、提案法はオフライン学習法よりおよそ 1.3 倍長くなったが、オンライン学習法よりおよそ 1/9.5 短くなった。すなわち、応答性について、提案法はオンライン学習法よりおよそ 9.5 倍速くなった。よって、提案法はニューラルネットワーク制御の実時間制御を実現可能であることが示された。定常偏差についても、提案法がオンライン学習法とオフライン学習法より小さいことは確認された。

2. ニューラルネットワークの出力

図 5.8 (a) はニューラルネットワークの出力 u_n を示し、図 5.9 (a) は図 5.8 (a) の拡大図である。提案法の出力値は、オンライン学習法より大きな出力値を得て、オフライン学習法とほぼ同じ大きさの出力値を得た。これは、過渡状態で提案法の学習効果が現れていることを意味する。

3. ニューラルネットワークの誤差信号

図 5.10 (a) はニューラルネットワークの誤差信号 E_n 、すなわち、フィードバック制御器の出力 u_f を示している。提案法の誤差信号は、オンライン学習法とオフライン学習法よりも常に小さかった。これは、過渡状態でも定常状態でも提案法の学習効果が現れていることを意味する。

(B) 負荷イナーシャ変更後の結果と考察は、以下の通りである。

1. 整定時間と定常偏差

オンライン学習法の整定時間は、図 5.7 (b) の t_1 であり、4.84 秒であった。

その時の定常偏差の最大値は $\pm 0.0010 \text{ deg}$ であった。

オフライン学習法の整定時間は、定常偏差の波形が持続的に振動したため、図 5.7 (b) の t_2 のマークが示すように 15 秒以上になった。

その時の定常偏差の最大値は $\pm 0.0040 \text{ deg}$ であった。

提案法の整定時間は、図 5.7 (b) の $t_3 = 1.16$ 秒であった。その時の定常偏差の最大値は ± 0.0001 deg であった。

整定時間について、提案法はオフライン学習法よりおよそ 1/13、オンライン学習法よりおよそ 1/4.2 短くなった。すなわち、応答性について、提案法はオフライン学習法よりおよそ 13 倍、オンライン学習法よりおよそ 4.2 倍速くなった。

定常偏差について、オフライン学習法は提案法より定常偏差の最大値がおよそ 40 倍となった。これは、オフライン学習法が負荷イナーシャ変更に対応できず、ロバスト性が欠けていることを意味する。オンライン学習法の定常偏差の最大値はオフライン学習法より 1/4 であるが、整定時間は提案法より 4.2 倍長くなった。これは、オンライン学習法が実時間制御に向いていないことを意味する。

負荷の急激な変更後の応答性について、提案法はオフライン学習法よりはるかに速くなり、オンライン学習法よりおよそ 4.2 倍速くなった。提案法の定常偏差の最大値は、オフライン学習法の 1/40、オンライン学習法の 1/10 に小さくなった。したがって、提案法がロバスト性が高い上に実時間制御に向いていることが分かった。

2. ニューラルネットワークの出力

図 5.8 (b) はニューラルネットワークの出力 u_n を示し、また、図 5.9 (b) は図 5.8 (b) の拡大図である。提案法は、15.0 から 15.6 秒の間にオンライン学習法とオフライン学習法より大きな出力値を得た。これは、過渡状態で提案法の学習効果が現れていることを意味する。

3. ニューラルネットワークの誤差信号

図 5.10 (b) はニューラルネットワークの誤差信号 E_n 、すなわち、フィードバック制御器の出力 u_f を示している。図 5.10 (b) に示した誤差信号 E_n について、提案法はオンライン学習法とオフライン学習法より小さかった。これは、過渡状態でも定常状態でも提案法の学習効果が現れていることを意味する。

(C) 目標軌道の周波数変更後の結果と考察は、以下の通りである。

1. 目標軌道の周波数を半分に変更した場合

図 5.11 は、目標軌道の周波数が半分に変更された場合のモータ回転角度の偏差を示している。負荷イナーシャ変更の前後における応答性と定常偏差について、図 5.11 から提案法がオフライン学習法より劣っていたが、オンライン学習法よりは非常に優れていたことが分かった。この結果について、周波数が低く変更されるとモータの加速度が小さくなり、角度の偏差が定常に近い状態であったため、オフライン学習法でも対応できたと考えられる。これより、オフライン学習法には目標軌道の周波数が低く変更された場合に良いロバスト性を持つことが分かった。また、提案法にも目標軌道の周波数が低く変更された場合に良いロバスト性を持つことが分かった。

2. 目標軌道の周波数を3倍に変更した場合

図 5.12 は、目標軌道の周波数が3倍に変更された場合のモータ回転角度の偏差を示している。負荷イナーシャ変更の前後における応答性と定常偏差について、図 5.12 から提案法がオンライン学習法とオフライン学習法より非常に優れていたことが分かった。また、オフライン学習法には負荷イナーシャを変更した時点で持続的に振動したため、目標軌道の周波数を高く変更した場合にオフライン学習法がロバスト性が欠けていることが分かった。

以上の結果と考察から、5.3 節に示した従来のオンライン学習法、オフライン学習法と提案法との比較表 5.1 が証明された。提案法は、オンライン学習法の実時間制御に向いていないという欠点とオフライン学習法のロバスト性が欠けているという欠点を解決した。すなわち、提案法はオフライン学習法の応答性が良いという利点とオンライン学習法のロバスト性が優れているという利点を融合させることができた。

5.5 あとがき

本章では、オンライン学習法およびオフライン学習法のアルゴリズムと提案したオンライン・オフライン融合型学習法のアルゴリズムを説明した。

オンライン学習法は、負荷変動や外乱が発生すると、それらの変化がニューラルネットワークの誤差信号へ反映されるため、負荷変動や外乱に対応できて、ロバスト性が良いという利点がある。ただし、サンプリングタイム内で1回しか重みに関する計算がされないため、整定時間が長くて、応答性が悪いという欠点がある。オフライン学習法は、出力に関する計算しかされないため、整定時間が短くて、応答性が良いという利点がある。ただし、負荷変動や外乱が学習に反映されないため、ロバスト性が悪いという欠点がある。提案法は、フィードバック誤差が閾値を超えなければ、オフライン学習法と同じく、出力に関するの少ない計算が実行されるため、応答性が良い。また、フィードバック誤差が閾値を超えれば、サンプリングタイム内で少ない繰り返しループのオンライン学習法を行うため、負荷変動や外乱に対応できてロバスト性も良い。

そして、上記で述べた3つの学習法の比較を証明するために、オンライン学習法、オフライン学習法、提案したオンライン・オフライン融合型学習法をそれぞれ用いた場合のシミュレーション結果を示した。このシミュレーションでは、オンライン学習法を用いた学習済重みの獲得についてのシミュレーションを行って、負荷イナーシャ変更前後の応答性とロバスト性についてのシミュレーションを行った。負荷イナーシャ変更前後のシミュレーションでは、負荷イナーシャ値の大きさをステップ状に変更させた。さらに、目標軌道の周波数変更前後の応答性とロバスト性についてのシミュレーションも行った。

これらのシミュレーション結果とその考察により、提案法はオンライン学習法の欠点とオフライン学習法の欠点を解決した。すなわち、提案法はオフライン学習法の利点とオンライン学習法の利点を融合させることができた。

第6章 学習率と回数閾値の更新則

6.1 はしがき

本章では、提案したオンライン・オフライン融合型学習法のパラメータに関する課題点と解決するための方法を6.2節で述べて、オンライン・オフライン融合型学習法を用いた学習率の増減に関わるサーボシステムの特性を6.3節に示す。そして、その特性の中から適切なパラメータの組み合わせを抽出して、最小二乗法で定式化を行い、新しい更新則を導出する。この一連の流れを6.4節で説明する。さらに、その更新則に基づき求められた値をオンライン・オフライン融合型学習法に代入して、サーボモータの位置決め制御のシミュレーションを行い、シミュレーションの結果と考察を6.5節で述べる。

6.2 学習率と回数閾値の新しい更新則の提案

前章では、サーボシステムの位置決め制御に対して、負荷変動と外乱の対応、実時間制御の実現を達成するため、フィードバック誤差が閾値を超えなければオフライン学習を行い、フィードバック誤差が閾値を超えればサンプリングタイム内で少ない繰り返しループのオンライン学習を行うようなオンライン・オフライン融合型学習法を提案した。

この提案法のパラメータである学習率 η 、サンプリングタイム内の繰り返しループの回数閾値 N_e とサーボシステムの応答性、ロバスト性との関係がまだ明らかではない。そのため、学習率 η と回数閾値 N_e のチューニングにおいて、手間がかかり、学習の効率が良いとは言えない。また、今までのニューラルネットワークの学習率の更新則についての研究では、学習率の増減について学習の効率と学習の極小値との関係がまだ定式化されていないため、サーボシステムの応答性、ロバスト性との関係を結ぶことが難しい [18] - [27]。

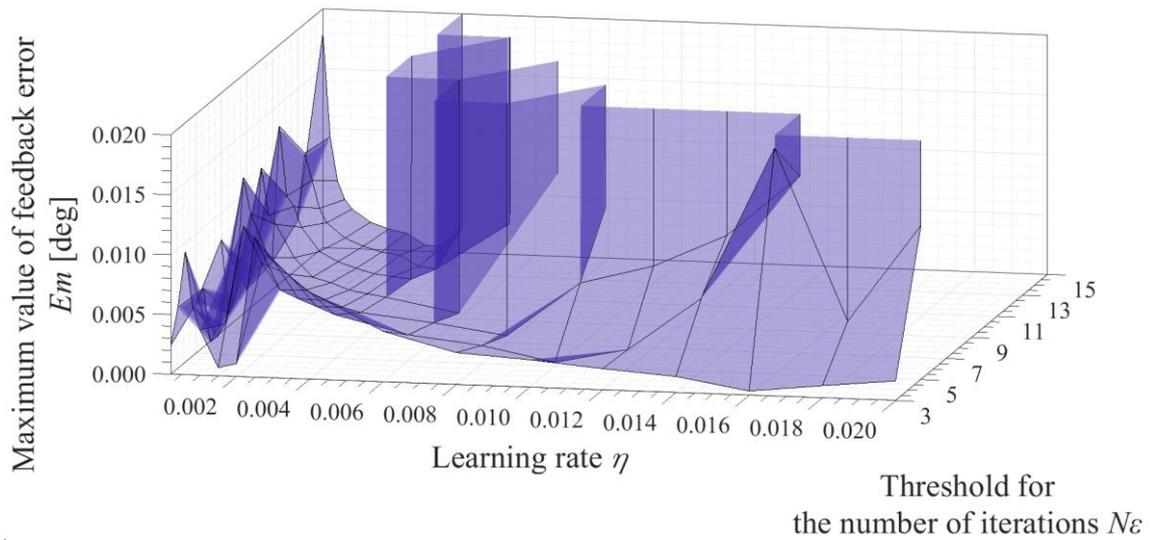
以上の背景に基づき、本研究では、サーボシステムの位置決め制御に対して、学習率 η 、回数閾値 N_e とサーボシステムの応答性、ロバスト性との関係をシミュレーションとその解析データから明らかにする。さらに、最小二乗法によってそれらの関係の定式化を行い、セルフチューニングが行えるような学習率 η と回数閾値 N_e の更新則を提案する。

6.3 学習率の増減に関わるサーボシステムの特性

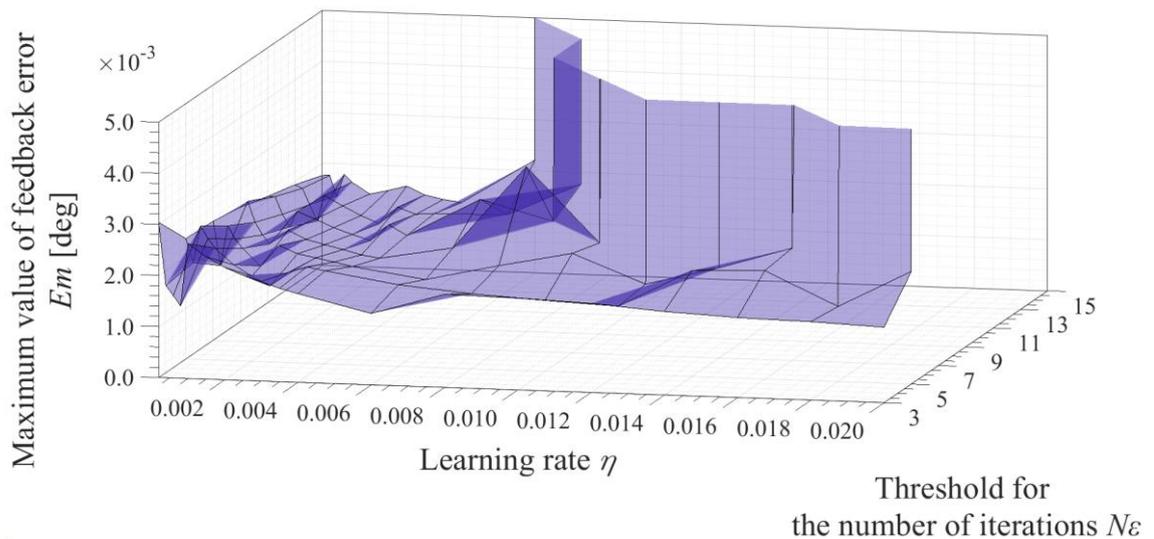
本節では、オンライン・オフライン融合型学習法を用いた学習率の増減に関わるサーボシステムの特性は、6.3.1項と6.3.2項に示される。6.3.1項では、負荷イナーシャ変更前後のフィードバック誤差の最大値 E_m 、学習率 η 、回数閾値 N_e の3つのパラメータに関する定常特性を図6.1に示す。6.3.2項では、負荷イナーシャ変更前後の整定時間 T_s 、学習率 η 、回数閾値 N_e の3つのパラメータに関する過渡特性を図6.2に示す。

6.3.1 学習率の増減に関わる定常特性

オンライン・オフライン融合型学習法のパラメータについて、まず、 $\eta = 0.004$ 、 $\alpha = 0.001$ で得られた時の重みを学習済重みとした。次に、学習率は $\eta = 0.0002$ から $\eta = 0.02$ 、回数閾値は $N_e = 3$ から $N_e = 15$ 、慣性係数は $\alpha = 0.001$ と設定した。また、負荷イナーシャ値は $J_L = J_M = 2.25 \times 10^{-7} \text{kgm}^2$ と設定した。図 6.1(a) と 6.1(b) それぞれは、負荷イナーシャを変更する前後でのフィードバック誤差の最大値、学習率、回数閾値のシミュレーション結果を示している。加えて、それらの値を表 6.1 と表 6.2 に示す。以下では、学習率のデータの数を N と定義する。



(a) 負荷イナーシャ変更前



(b) 負荷イナーシャ変更後

図 6.1 オンライン・オフライン融合型学習法を用いたフィードバック誤差の最大値、学習率、回数閾値（負荷イナーシャの変更前と変更後）

表 6.1 フィードバック誤差の最大値 E_m (負荷イナーシャ変更前)

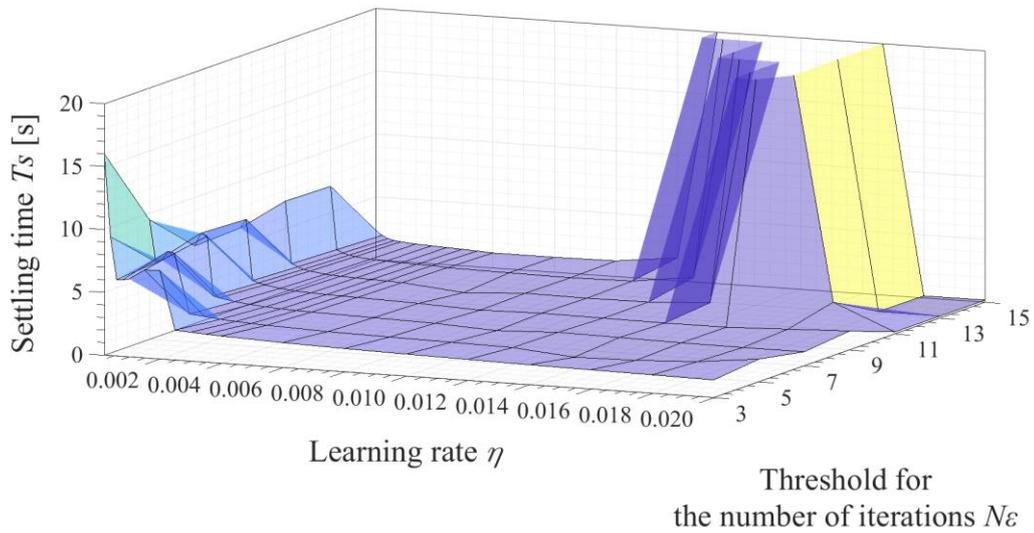
データ の数 N	学習率 η	回数閾値 N_ε						
		3	5	7	9	11	13	15
1	0.0002	0.0023	0.0040	0.0077	0.0018	0.0022	0.0039	0.0179
2	0.0004	0.0055	0.0054	0.0015	0.0035	0.0137	0.0096	0.0093
3	0.0006	0.0102	0.0009	0.0031	0.0119	0.0078	0.0074	0.0056
4	0.0008	0.0057	0.0014	0.0129	0.0080	0.0070	0.0051	0.0041
5	0.0010	0.0037	0.0025	0.0099	0.0073	0.0053	0.0039	0.0033
6	0.0015	0.0006	0.0107	0.0063	0.0046	0.0033	0.0027	0.0022
7	0.0020	0.0010	0.0077	0.0050	0.0030	0.0025	0.0020	0.0017
8	0.0025	0.0116	0.0055	0.0037	0.0026	0.0020	0.0016	0.0014
9	0.0030	0.0098	0.0048	0.0029	0.0020	0.0016	0.0014	0.0005
10	0.0035	0.0084	0.0042	0.0024	0.0018	0.0014	0.0005	0.0004
11	0.0040	0.0072	0.0036	0.0020	0.0015	0.0013	0.0004	0.0021
12	0.0060	0.0042	0.0021	0.0014	0.0004	4.9431	0.0044	4.6882
13	0.0080	0.0026	0.0015	0.0005	5.5826	0.0106	4.9578	6.8082
14	0.0100	0.0023	0.0003	0.0051	0.0097	6.5867	6.6505	6.9485
15	0.0120	0.0017	0.0016	0.0068	6.4762	6.7786	6.9239	6.9364
16	0.0140	0.0013	0.0061	0.0096	6.3216	6.9894	7.0780	6.9886
17	0.0160	0.0003	0.0189	0.0147	6.9937	6.9735	7.1675	7.1116
18	0.0180	0.0010	0.0046	6.6503	7.0457	6.9121	7.0811	7.0688
19	0.0200	0.0016	0.0128	6.8744	6.9920	7.0130	6.9704	7.0120

表 6.2 フィードバック誤差の最大値 E_m (負荷イナーシャ変更後)

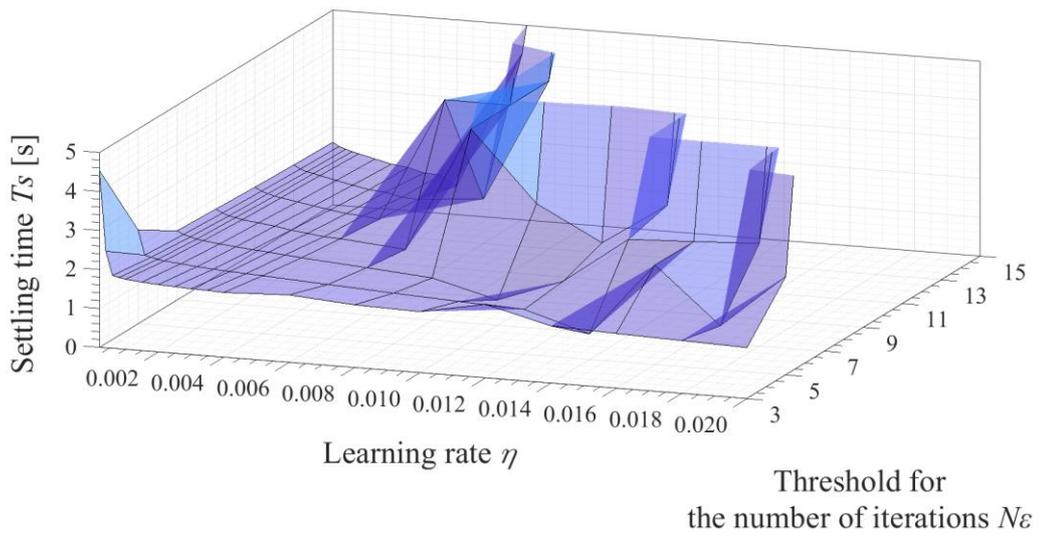
データ の数 N	学習率 η	回数閾値 N_ε						
		3	5	7	9	11	13	15
1	0.0002	0.0030	0.0023	0.0023	0.0022	0.0020	0.0019	0.0017
2	0.0004	0.0018	0.0018	0.0014	0.0023	0.0020	0.0018	0.0017
3	0.0006	0.0016	0.0026	0.0022	0.0019	0.0017	0.0015	0.0013
4	0.0008	0.0014	0.0023	0.0019	0.0017	0.0015	0.0012	0.0018
5	0.0010	0.0026	0.0021	0.0018	0.0015	0.0018	0.0017	0.0016
6	0.0015	0.0023	0.0018	0.0014	0.0018	0.0016	0.0015	0.0014
7	0.0020	0.0022	0.0016	0.0019	0.0016	0.0015	0.0013	0.0015
8	0.0025	0.0020	0.0014	0.0017	0.0015	0.0013	0.0012	0.0016
9	0.0030	0.0019	0.0017	0.0016	0.0014	0.0012	0.0015	0.0014
10	0.0035	0.0018	0.0018	0.0016	0.0013	0.0016	0.0014	0.0013
11	0.0040	0.0017	0.0018	0.0015	0.0013	0.0014	0.0014	0.0013
12	0.0060	0.0014	0.0015	0.0013	0.0015	0.0021	0.0015	0.0022
13	0.0080	0.0018	0.0014	0.0016	0.0032	0.0017	0.0021	3.3284
14	0.0100	0.0018	0.0013	0.0019	0.0017	0.6822	2.3544	2.2298
15	0.0120	0.0017	0.0013	0.0013	0.1770	3.6661	2.2298	2.2299
16	0.0140	0.0016	0.0020	0.0017	0.4300	2.2298	2.2298	2.2299
17	0.0160	0.0015	0.0021	0.0021	3.1378	2.2298	2.2299	2.2299
18	0.0180	0.0015	0.0014	0.0935	2.2298	2.2298	2.2299	2.2300
19	0.0200	0.0014	0.0022	3.4260	2.2297	2.2298	2.2300	2.2299

6.3.2 学習率の増減に関わる過渡特性

オンライン・オフライン融合型学習法のパラメータについては、前項と同じように設定した。図 6.2 (a) と 6.2 (b) それぞれは、負荷イナーシャを変更する前後での整定時間、学習率 η 、回数閾値 N_ϵ のシミュレーション結果を示している。また、それらの値を表 6.3 と表 6.4 に示す。



(a) 負荷イナーシャ変更前



(b) 負荷イナーシャ変更後

図 6.2 オンライン・オフライン融合型学習法を用いた整定時間、学習率、回数閾値 (負荷イナーシャの変更前と変更後)

表 6.3 整定時間 T_s (負荷イナーシャ変更前)

データ の数 N	学習率 η	回数閾値 N_ε						
		3	5	7	9	11	13	15
1	0.0002	16.00	9.503	6.158	6.320	7.194	7.105	2.352
2	0.0004	9.453	6.199	7.214	7.039	2.254	2.109	1.975
3	0.0006	6.089	6.357	7.010	2.215	2.040	1.883	1.755
4	0.0008	6.100	7.108	2.279	2.065	1.883	1.738	1.640
5	0.0010	6.205	6.986	2.166	1.947	1.770	1.650	1.572
6	0.0015	7.000	2.241	1.959	1.745	1.614	1.534	1.479
7	0.0020	7.054	2.107	1.818	1.635	1.534	1.470	1.422
8	0.0025	2.398	1.992	1.717	1.567	1.482	1.423	1.373
9	0.0030	2.310	1.903	1.648	1.512	1.441	1.380	1.322
10	0.0035	2.238	1.830	1.597	1.482	1.405	1.377	1.267
11	0.0040	2.177	1.769	1.557	1.449	1.369	1.290	1.198
12	0.0060	1.996	1.612	1.445	1.326	0.914	1.226	1.196
13	0.0080	1.868	1.517	1.346	1.129	1.207	1.179	1.252
14	0.0100	1.767	1.439	1.107	1.229	1.157	1.255	1.836
15	0.0120	1.684	1.364	1.313	1.175	1.194	1.893	30.00
16	0.0140	1.611	0.940	1.247	1.127	1.477	30.00	30.00
17	0.0160	1.545	1.031	1.161	1.138	30.00	30.00	30.00
18	0.0180	1.482	1.269	1.129	1.322	2.198	0.194	0.172
19	0.0200	1.415	1.815	1.131	1.513	0.180	0.176	0.185

表 6.4 整定時間 T_s (負荷イナーシャ変更後)

データ の数 N	学習率 η	回数閾値 N_ε						
		3	5	7	9	11	13	15
1	0.0002	4.540	2.400	1.770	1.720	1.670	1.630	1.600
2	0.0004	2.480	1.770	1.690	1.630	1.580	1.540	1.510
3	0.0006	1.870	1.730	1.640	1.570	1.530	1.480	1.450
4	0.0008	1.830	1.690	1.600	1.540	1.480	1.450	1.400
5	0.0010	1.800	1.660	1.570	1.500	1.450	1.410	1.360
6	0.0015	1.760	1.620	1.520	1.450	1.380	1.320	1.260
7	0.0020	1.730	1.580	1.480	1.400	1.310	1.230	1.160
8	0.0025	1.710	1.550	1.440	1.330	1.240	1.150	1.110
9	0.0030	1.690	1.520	1.410	1.280	1.180	1.120	1.000
10	0.0035	1.680	1.500	1.370	1.230	1.150	1.020	0.980
11	0.0040	1.660	1.480	1.330	1.210	1.100	1.000	0.740
12	0.0060	1.730	1.390	1.210	1.070	4.300	1.140	4.290
13	0.0080	1.610	1.300	1.080	4.290	4.290	4.300	15.00
14	0.0100	1.570	1.220	0.650	2.510	15.00	15.00	15.00
15	0.0120	1.890	1.150	1.260	1.639	15.00	15.00	15.00
16	0.0140	1.470	0.650	2.470	2.735	15.00	15.00	15.00
17	0.0160	1.410	2.557	2.545	15.00	15.00	15.00	15.00
18	0.0180	1.360	1.130	2.633	15.00	15.00	15.00	15.00
19	0.0200	1.310	2.530	15.00	15.00	15.00	15.00	15.00

6.4 最小二乗法に基づく学習率、回数閾値の定式化および更新則の導出

本節では、前節のサーボシステムの特徴から最小の誤差になるパラメータの組み合わせ、もしくは、最短の収束時間になるパラメータの組み合わせを抽出して、それらのパラメータの組み合わせに最小二乗法を適用する。上記について、フィードバック誤差の最大値 E_m 、学習率 η 、回数閾値 N_ε の定式化を 6.4.1 項に説明して、整定時間 T_s 、学習率 η 、回数閾値 N_ε の定式化を 6.4.2 項に説明する。さらに、学習率と回数閾値の更新則の導出を 6.4.3 項に説明する。

6.4.1 学習率、回数閾値と誤差最大値との定式化

学習率、回数閾値と誤差最大値の定式化について説明する。表 6.2 と表 6.3 のそれぞれの学習率 η と回数閾値 N_ε に対して、シミュレーションの解析で得られたフィードバック誤差の最大値を $E_m(i)$ として、学習率 η と回数閾値 N_ε に関わるフィードバック誤差の最大値の推定関数を $f_i(\eta, N_\varepsilon)$ と定義する。また、推定関数には多項式曲線を用いる。そして、 $E_m(i)$ と $f_i(\eta, N_\varepsilon)$ との差の二乗和を E_{em} と定義して、式(6.1)に示す。

$$E_{em} = \sum_{i=1}^N [E_m(i) - f_i(\eta, N_\varepsilon)]^2 = \sum_{i=1}^N [E_m(i) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 \quad (6.1)$$

パラメータの値 a 、 b 、 c 、 d 、 e は、式(6.2)から式(6.6)で求められる。

$$\frac{\partial E_{em}}{\partial a} = \frac{\partial}{\partial a} \sum_{i=1}^N [E_m(i) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.2)$$

$$\frac{\partial E_{em}}{\partial b} = \frac{\partial}{\partial b} \sum_{i=1}^N [E_m(i) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.3)$$

$$\frac{\partial E_{em}}{\partial c} = \frac{\partial}{\partial c} \sum_{i=1}^N [E_m(i) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.4)$$

$$\frac{\partial E_{em}}{\partial d} = \frac{\partial}{\partial d} \sum_{i=1}^N [E_m(i) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.5)$$

$$\frac{\partial E_{em}}{\partial e} = \frac{\partial}{\partial e} \sum_{i=1}^N [E_m(i) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.6)$$

また、式(6.2)から式(6.6)の整理された式は、式(6.7)から式(6.11)に示される。

$$a \sum_{i=1}^N \eta^4 + b \sum_{i=1}^N \eta^2 N_\varepsilon^2 + c \sum_{i=1}^N \eta^3 + d \sum_{i=1}^N \eta^2 N_\varepsilon + e \sum_{i=1}^N \eta^2 = \sum_{i=1}^N \eta^2 E_m(i) \quad (6.7)$$

$$a \sum_{i=1}^N \eta^2 N_\varepsilon^2 + b \sum_{i=1}^N N_\varepsilon^4 + c \sum_{i=1}^N \eta N_\varepsilon^2 + d \sum_{i=1}^N N_\varepsilon^3 + e \sum_{i=1}^N N_\varepsilon^2 = \sum_{i=1}^N N_\varepsilon^2 E_m(i) \quad (6.8)$$

$$a \sum_{i=1}^N \eta^3 + b \sum_{i=1}^N \eta N_\varepsilon^2 + c \sum_{i=1}^N \eta^2 + d \sum_{i=1}^N \eta N_\varepsilon + e \sum_{i=1}^N \eta = \sum_{i=1}^N \eta E_m(i) \quad (6.9)$$

$$a \sum_{i=1}^N \eta^2 N_\varepsilon + b \sum_{i=1}^N N_\varepsilon^3 + c \sum_{i=1}^N \eta N_\varepsilon + d \sum_{i=1}^N N_\varepsilon^2 + e \sum_{i=1}^N N_\varepsilon = \sum_{i=1}^N N_\varepsilon E_m(i) \quad (6.10)$$

$$a \sum_{i=1}^N \eta^2 + b \sum_{i=1}^N N_\varepsilon^2 + c \sum_{i=1}^N \eta + d \sum_{i=1}^N N_\varepsilon + Ne = \sum_{i=1}^N E_m(i) \quad (6.11)$$

加えて、求められたパラメータ a 、 b 、 c 、 d 、 e は、式(6.12)に示される。

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \sum_{i=1}^N \begin{pmatrix} \eta^4 & \eta^2 N_\varepsilon^2 & \eta^3 & \eta^2 N_\varepsilon & \eta^2 \\ \eta^2 N_\varepsilon^2 & N_\varepsilon^4 & \eta N_\varepsilon^2 & N_\varepsilon^3 & N_\varepsilon^2 \\ \eta^3 & \eta N_\varepsilon^2 & \eta^2 & \eta N_\varepsilon & \eta \\ \eta^2 N_\varepsilon & N_\varepsilon^3 & \eta N_\varepsilon & N_\varepsilon^2 & N_\varepsilon \\ \eta^2 & N_\varepsilon^2 & \eta & N_\varepsilon & N \end{pmatrix}^{-1} \sum_{i=1}^N \begin{pmatrix} \eta^2 E_m(i) \\ N_\varepsilon^2 E_m(i) \\ \eta E_m(i) \\ N_\varepsilon E_m(i) \\ E_m(i) \end{pmatrix} \quad (6.12)$$

表 6.1 と表 6.2 それぞれでフィードバック誤差が最小値になる学習率 η と回数閾値 N_ε の組み合わせに最小二乗法を適用する。それによって、負荷イナーシャ変更前後の近似曲線 \tilde{E}_{mb} と \tilde{E}_{ma} 、すなわち、式(6.13)と式(6.14)が得られる。そして、それらの曲線は、図 6.3 に示される。

$$\tilde{E}_{mb} = 10.2\eta^2 - (8.8 \times 10^{-6})N_\varepsilon^2 - 0.2\eta + (1.4 \times 10^{-4})N_\varepsilon + (1.1 \times 10^{-3}) \quad (6.13)$$

$$\tilde{E}_{ma} = 1.4\eta^2 + (7.6 \times 10^{-6})N_\varepsilon^2 - (4.5 \times 10^{-2})\eta - (1.7 \times 10^{-4})N_\varepsilon + (2.4 \times 10^{-3}) \quad (6.14)$$

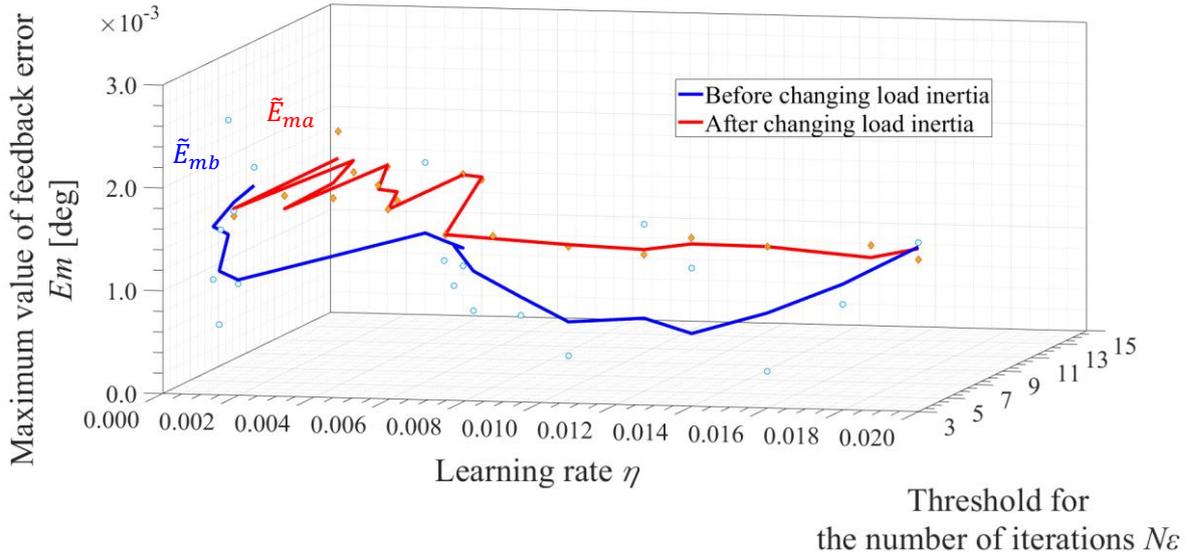


図 6.3 学習率、回数閾値と誤差最大値のデータに基づく近似曲線

負荷イナーシャ変更前の近似曲線 \tilde{E}_{mb} と負荷イナーシャ変更後の近似曲線 \tilde{E}_{ma} では、学習率 η を増加させると全く異なる曲線を描く傾向がある。また、 \tilde{E}_{mb} と \tilde{E}_{ma} は、回数閾値 N_ε を増加させても全く異なる曲線を描く傾向がある。

6.4.2 学習率、回数閾値と整定時間との定式化

学習率、回数閾値と整定時間の定式化について説明する。表 6.3 と表 6.4 のそれぞれの学習率 η と回数閾値 N_ε に対して、シミュレーションの解析で得られた整定時間を $T_s(j)$ として、学習率 η と回数閾値 N_ε に関わる整定時間の推定関数を $f_j(\eta, N_\varepsilon)$ と定義する。また、推定関数には多項式曲線を用いる。そして、 $T_s(j)$ と $f_j(\eta, N_\varepsilon)$ との差の二乗和を E_{ts} と定義して、式(6.15)に示す。

$$E_{ts} = \sum_{j=1}^N [T_s(j) - f_j(\eta, N_\varepsilon)]^2 = \sum_{j=1}^N [T_s(j) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 \quad (6.15)$$

パラメータの値 a 、 b 、 c 、 d 、 e は、式(6.16)から式(6.20)で求められる。

$$\frac{\partial E_{ts}}{\partial a} = \frac{\partial}{\partial a} \sum_{j=1}^N [T_s(j) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.16)$$

$$\frac{\partial E_{ts}}{\partial b} = \frac{\partial}{\partial b} \sum_{j=1}^N [T_s(j) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.17)$$

$$\frac{\partial E_{ts}}{\partial c} = \frac{\partial}{\partial c} \sum_{j=1}^N [T_s(j) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.18)$$

$$\frac{\partial E_{ts}}{\partial d} = \frac{\partial}{\partial d} \sum_{j=1}^N [T_s(j) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.19)$$

$$\frac{\partial E_{ts}}{\partial e} = \frac{\partial}{\partial e} \sum_{j=1}^N [T_s(j) - (a\eta^2 + bN_\varepsilon^2 + c\eta + dN_\varepsilon + e)]^2 = 0 \quad (6.20)$$

また、式(6.16)から式(6.20)の整理された式は、式(6.21)から式(6.25)に示される。

$$a \sum_{j=1}^N \eta^4 + b \sum_{j=1}^N \eta^2 N_\varepsilon^2 + c \sum_{j=1}^N \eta^3 + d \sum_{j=1}^N \eta^2 N_\varepsilon + e \sum_{j=1}^N \eta^2 = \sum_{j=1}^N \eta^2 T_s(j) \quad (6.21)$$

$$a \sum_{j=1}^N \eta^2 N_\varepsilon^2 + b \sum_{j=1}^N N_\varepsilon^4 + c \sum_{j=1}^N \eta N_\varepsilon^2 + d \sum_{j=1}^N N_\varepsilon^3 + e \sum_{j=1}^N N_\varepsilon^2 = \sum_{j=1}^N N_\varepsilon^2 T_s(j) \quad (6.22)$$

$$a \sum_{j=1}^N \eta^3 + b \sum_{j=1}^N \eta N_\varepsilon^2 + c \sum_{j=1}^N \eta^2 + d \sum_{j=1}^N \eta N_\varepsilon + e \sum_{j=1}^N \eta = \sum_{j=1}^N \eta T_s(j) \quad (6.23)$$

$$a \sum_{j=1}^N \eta^2 N_\varepsilon + b \sum_{j=1}^N N_\varepsilon^3 + c \sum_{j=1}^N \eta N_\varepsilon + d \sum_{j=1}^N N_\varepsilon^2 + e \sum_{j=1}^N N_\varepsilon = \sum_{j=1}^N N_\varepsilon T_s(j) \quad (6.24)$$

$$a \sum_{j=1}^N \eta^2 + b \sum_{j=1}^N N_\varepsilon^2 + c \sum_{j=1}^N \eta + d \sum_{j=1}^N N_\varepsilon + Ne = \sum_{j=1}^N T_s(j) \quad (6.25)$$

加えて、求められたパラメータ a 、 b 、 c 、 d 、 e は式(6.26)に示される。

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \sum_{j=1}^N \begin{pmatrix} \eta^4 & \eta^2 N_\varepsilon^2 & \eta^3 & \eta^2 N_\varepsilon & \eta^2 \\ \eta^2 N_\varepsilon^2 & N_\varepsilon^4 & \eta N_\varepsilon^2 & N_\varepsilon^3 & N_\varepsilon^2 \\ \eta^3 & \eta N_\varepsilon^2 & \eta^2 & \eta N_\varepsilon & \eta \\ \eta^2 N_\varepsilon & N_\varepsilon^3 & \eta N_\varepsilon & N_\varepsilon^2 & N_\varepsilon \\ \eta^2 & N_\varepsilon^2 & \eta & N_\varepsilon & N \end{pmatrix}^{-1} \sum_{j=1}^N \begin{pmatrix} \eta^2 T_s(j) \\ N_\varepsilon^2 T_s(j) \\ \eta T_s(j) \\ N_\varepsilon T_s(j) \\ T_s(j) \end{pmatrix} \quad (6.26)$$

表 6.3 と表 6.4 それぞれで最短の収束時間になる学習率 η と回数閾値 N_ε の組み合わせに最小二乗法を適用する。それによって、負荷イナーシャ変更前後の近似曲線 \tilde{T}_{sb} と \tilde{T}_{sa} 、すなわち、式(6.27)と式(6.28)が得られる。そして、それらの曲線は図 6.4 に示される。

$$\tilde{T}_{sb} = (6.5 \times 10^3)\eta^2 + (3.2 \times 10^{-4})N_\varepsilon^2 - 220.8\eta - (9.9 \times 10^{-2})N_\varepsilon + 3.3 \quad (6.27)$$

$$\tilde{T}_{sa} = (7.3 \times 10^3)\eta^2 + (6.9 \times 10^{-3})N_\varepsilon^2 - 232.8\eta - 0.3N_\varepsilon + 3.8 \quad (6.28)$$

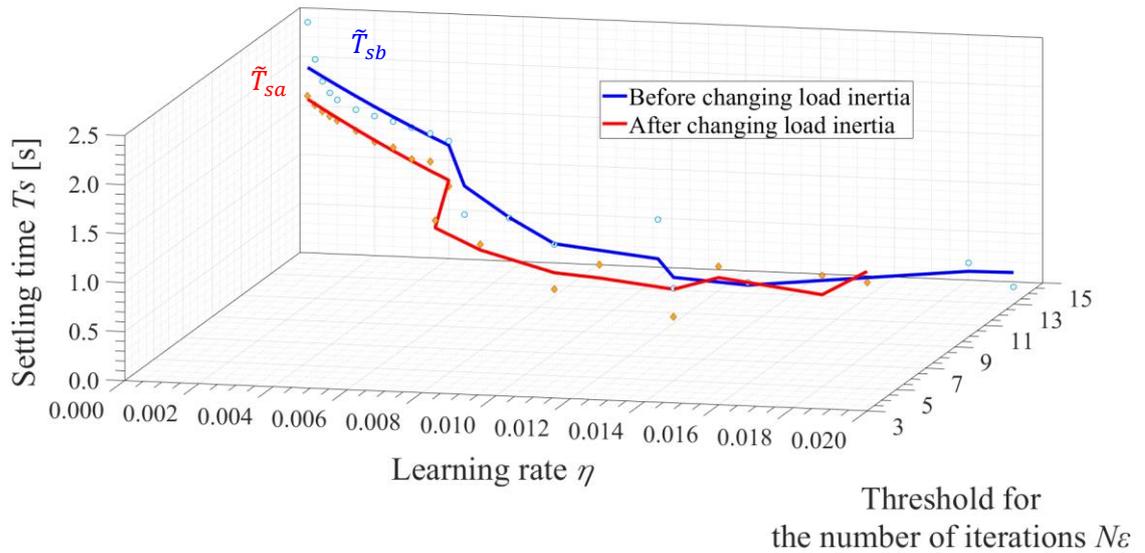


図 6.4 学習率、回数閾値と整定時間のデータに基づく近似曲線

負荷イナーシャ変更前の近似曲線 \tilde{T}_{sb} と負荷イナーシャ変更後の近似曲線 \tilde{T}_{sa} では、学習率 η と回数閾値 N_ε を増加させると、学習率の一定値まで同じよう曲線を描く傾向がある。また、 \tilde{T}_{sb} と \tilde{T}_{sa} は、学習率 η と回数閾値 N_ε の一定値を超えると異なる曲線を描く傾向がある。

6.4.3 学習率と回数閾値の更新則の導出

更新則を導出するためには、学習率を算出する必要がある。その学習率を算出するためには、指標値のフィードバック誤差の最大値 E_{mr} と整定時間 T_{sr} を設けて、回数閾値 \tilde{N}_ε を算出する。回数閾値 \tilde{N}_ε を算出するためには、近似曲線 \tilde{E}_{mb} と \tilde{T}_{sb} の連立方程式もしくは近似曲線 \tilde{E}_{ma} と \tilde{T}_{sa} の連立方程式を解く。 \tilde{E}_{mb} または \tilde{E}_{ma} それぞれのパラメータを a_1, b_1, c_1, d_1, e_1 と定義し、 \tilde{T}_{sb} または \tilde{T}_{sa} それぞれのパラメータを a_2, b_2, c_2, d_2, e_2 と定義する。そして、回数閾値 \tilde{N}_ε を算出するための式の係数を A, B, C と定義すると、式(6.29)と式(6.30)に示される。

$$\tilde{N}_\varepsilon = A\tilde{\eta}^2 + B\tilde{\eta} + C \quad (6.29)$$

$$A = -\frac{a_1 b_2 - a_2 b_1}{d_1 b_2 - d_2 b_1}$$

$$B = -\frac{c_1 b_2 - c_2 b_1}{d_1 b_2 - d_2 b_1} \quad (6.30)$$

$$C = \frac{b_2 E_{mr} - b_1 T_{sr} - (e_1 b_2 - e_2 b_1)}{d_1 b_2 - d_2 b_1}$$

\tilde{E}_{mb} または \tilde{E}_{ma} に \tilde{N}_ε を代入して、展開された式にフェラーリの解法^{[45][46]}を用いる。展開された式の係数は D, E, F, G, H と定義されて、式(6.31)と式(6.32)に示される。

$$D\eta^4 + E\eta^3 + F\eta^2 + G\eta + H = 0 \quad (6.31)$$

$$D = b_1 A^2$$

$$E = 2b_1 AB$$

$$F = a_1 + b_1(B^2 + 2AC) + d_1 A \quad (6.32)$$

$$G = 2b_1 BC + c_1 + d_1 B$$

$$H = b_1 C^2 + d_1 C + e_1 - \tilde{E}_{mb}$$

式(6.31) に係数 z を用いた式 $\tilde{\eta} = z - E/4D$ を代入して、展開された4次乗の式の係数は p, q, r と定義されて、式(6.33)と式(6.34)に示される。

$$z^4 + pz^2 + qz + r = 0 \quad (6.33)$$

$$p = -\frac{3E^2}{8D^2} + \frac{F}{D}$$

$$q = \frac{E^3}{8D^3} - \frac{EF}{2D^2} + \frac{G}{D} \quad (6.34)$$

$$r = -\frac{3E^4}{256D^4} + \frac{E^2 F}{16D^3} - \frac{EG}{4D^2} + \frac{H}{D}$$

式(6.33)の z の 4 次式を左辺として、係数 t を用いた式 $tz^2 + t^2/4$ を両辺に加える。そして、その式の平方完成は、式(6.35)に示される。

$$\left(z^2 + \frac{t}{2}\right)^2 = (t-p)\left\{z - \frac{q}{2(t-p)}\right\}^2 - \frac{q^2}{4(t-p)} + \frac{t^2}{4} - r \quad (6.35)$$

式(6.35)の右辺の $-q^2/4(t-p) + t^2/4 - r = 0$ を成り立たせるため、 $-q^2/4(t-p) + t^2/4 - r = 0$ を t について整理すると、式(6.36)に示される。

$$t^3 - pt^2 - 4rt + 4pr - q^2 = 0 \quad (6.36)$$

式(6.36)の解の 1 つを t_0 と定義して、式(6.35)は式(6.37)に変形される。

$$\left(z^2 + \frac{t_0}{2}\right)^2 = (t_0-p)\left\{z - \frac{q}{2(t_0-p)}\right\}^2 \quad (6.37)$$

式(6.37)の両辺の平方根を取り除いて、解の公式を用いて z の解を求める。そして、 $\tilde{\eta} = z - E/4D$ に z の解を代入すると、学習率 $\tilde{\eta}$ は以下の 4 つの解で定式化される。ただし、4 つの解の中に、 $0 < \tilde{\eta} < 1$ を満たさない解、複素数の解が含まれる場合は、それらを使わない。

$$\begin{aligned} \tilde{\eta}_{1,2} &= -\frac{E}{4D} + \frac{1}{2}\left(-\sqrt{t_0-p} \pm \sqrt{-t_0-p + \frac{2q}{\sqrt{t_0-p}}}\right) \\ \tilde{\eta}_{3,4} &= -\frac{E}{4D} + \frac{1}{2}\left(\sqrt{t_0-p} \pm \sqrt{-t_0-p - \frac{2q}{\sqrt{t_0-p}}}\right) \end{aligned} \quad (6.38)$$

また、学習率 $\tilde{\eta}$ の解を式(6.29)に代入すると、回数閾値 \tilde{N}_ε は式(6.39)で定式化される。ただし、負の数、複素数の解、大きな数値である場合は、それらを使わない。

$$\begin{aligned} \tilde{N}_{\varepsilon 1, \varepsilon 2} &= A\tilde{\eta}_{1,2}^2 + B\tilde{\eta}_{1,2} + C \\ \tilde{N}_{\varepsilon 3, \varepsilon 4} &= A\tilde{\eta}_{3,4}^2 + B\tilde{\eta}_{3,4} + C \end{aligned} \quad (6.39)$$

以上の手順によって、学習率と回数閾値の更新則は導出された。

6.5 学習率、回数閾値の更新則に基づくサーボシステムの解析結果

本節では、まず、学習率 η と回数閾値 N_ε の更新則に基づいて求められた回数閾値 \tilde{N}_ε と学習率 $\tilde{\eta}$ をオンライン・オフライン融合型学習法に代入する。次に、サーボモータの位置決め制御に対して、負荷イナーシャ変更前後のシミュレーションを行う。その結果を 6.5.1 項に示す。なお、負荷イナーシャ値は $J_L = J_M = 2.25 \times 10^{-7} \text{kgm}^2$ と設定する。最後に、シミュレーションで得られた結果の値と指標値との一致について、6.5.2 項と 6.5.3 項に述べる。

6.5.1 解析結果

指標値を $E_{mr} = 0.002 \text{deg}$ 、 $T_{sr} = 1.5 \text{秒}$ と決め、その指標値に対して、更新則で求められた負荷イナーシャ変更前後の学習率と回数閾値は $\tilde{\eta} = 0.022$ 、 $\tilde{N}_\varepsilon = 2$ と $\tilde{\eta} = 0.00029$ 、 $\tilde{N}_\varepsilon = 20$ であった。これらのパラメータの組み合わせに基づいたシミュレーションの結果を示す。図 6.5 は、負荷イナーシャ変更前後のフィードバック誤差 e の結果を示している。図 6.6 は、負荷イナーシャ変更前後のニューラルネットワーク補償器の出力 u_n の結果を示している。そして、図 6.7 は、負荷イナーシャ変更前後のフィードバック制御器の出力 u_f の結果を示している。

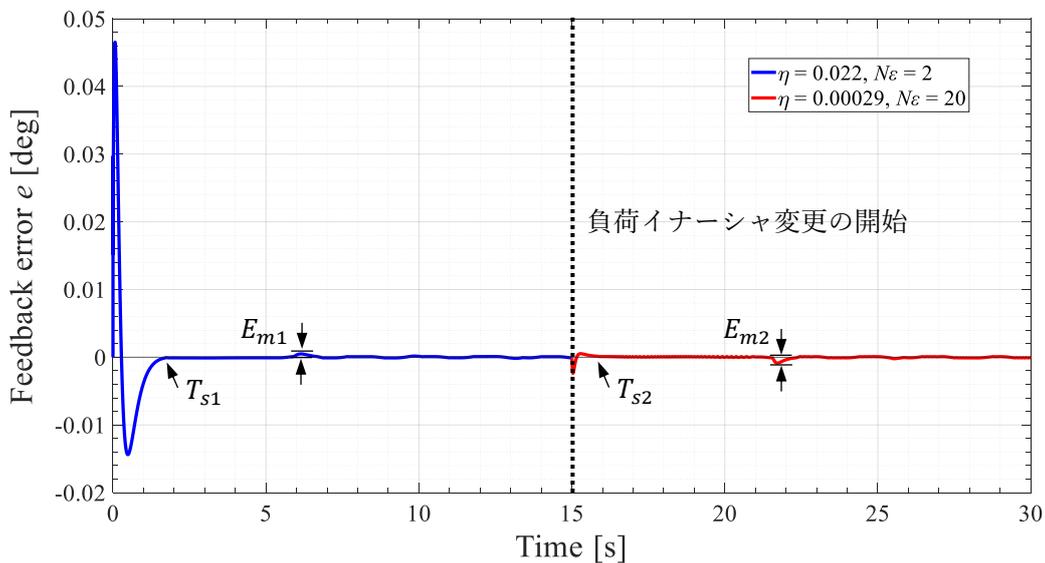


図 6.5 学習率と回数閾値の更新則に基づくオンライン・オフライン融合型学習法のフィードバック誤差（負荷イナーシャ変更前後）

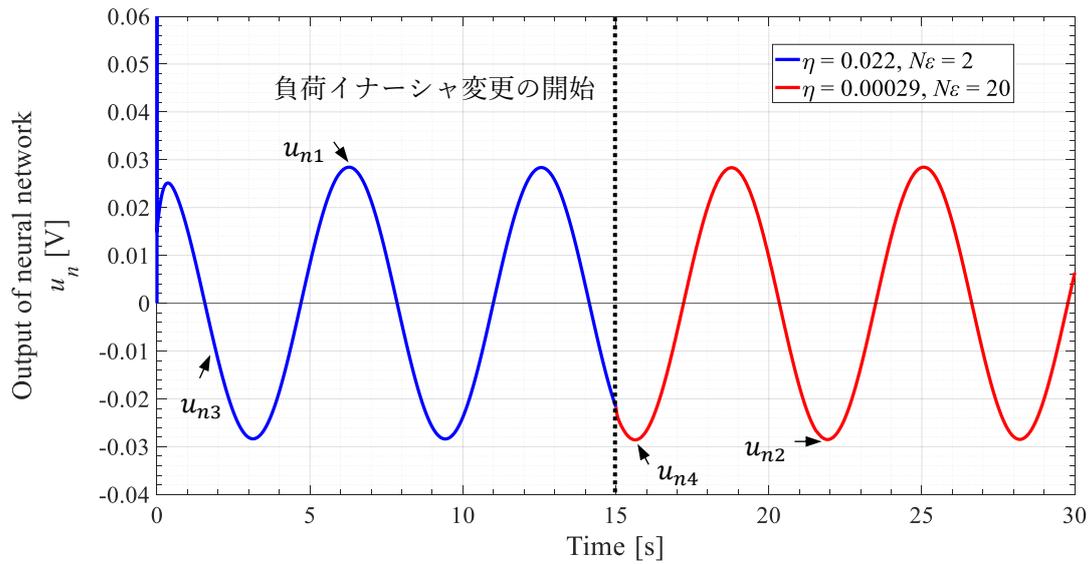


図 6.6 学習率と回数閾値の更新則に基づくオンライン・オフライン融合型学習法のニューラルネットワーク出力（負荷イナーシャ変更前後）

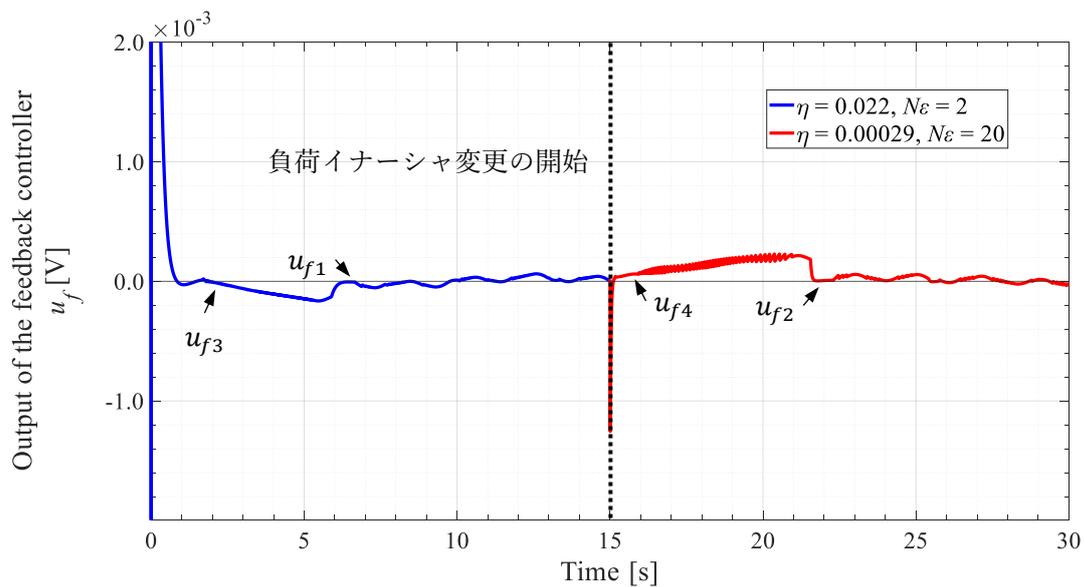


図 6.7 学習率と回数閾値の更新則に基づくオンライン・オフライン融合型学習法のフィードバック制御器の出力（負荷イナーシャ変更前後）

6.5.2 サーボシステムの定常特性に関する考察

まず、負荷イナーシャ変更前の結果を考察する。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での定常誤差の最大値は、図 6.5 の E_{m1} であり、0.00046 deg であった。その時の時間は、6.125 秒であった。また、その時のニューラルネットワークの出力は、図 6.6 の u_{n1} であり、その時のフィードバック制御器の出力は、図 6.7 の u_{f1} であった。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での定常誤差の最大値 E_{m1} は、指標値 E_{mr} 0.002 deg のおよそ 1/4.3 である。よって、シミュレーションの解析結果から、定常誤差の最大値 E_{m1} が指標値 E_{mr} の範囲に収められた。

次に、負荷イナーシャ変更後の結果を考察する。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での定常誤差の最大値は、図 6.5 の E_{m2} であり、0.00086 deg であった。その時の時間は、21.73 秒であった。また、その時のニューラルネットワークの出力は、図 6.6 の u_{n2} であり、その時のフィードバック制御器の出力は、図 6.7 の u_{f2} である。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での定常誤差の最大値 E_{m2} は、フィードバック誤差の指標値 E_{mr} 0.002 deg のおよそ 1/2.3 である。よって、シミュレーションの解析結果から定常誤差の最大値 E_{m2} が指標値 E_{mr} の範囲に収められた。

6.5.3 サーボシステムの過渡特性に関する考察

まず、負荷イナーシャ変更前の結果を考察する。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での整定時間は図 6.5 の T_{s1} であり、1.692 秒であった。その時のフィードバック誤差の値は、閾値の範囲内で収まった。また、その時のニューラルネットワークの出力は、図 6.6 の u_{n3} であり、その時のフィードバック制御器の出力は、図 6.7 の u_{f3} である。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での整定時間 T_{s1} は、整定時間の指標値 T_{sr} 1.5 秒のおよそ 1.1 倍であるが、整定時間の指標値に近い値である。よって、シミュレーションの解析結果から、整定時間 T_{s1} が指標値 T_{sr} の範囲に収められた。

次に、負荷イナーシャ変更後の結果を考察する。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での整定時間は図 6.5 の T_{s2} であり、0.920 秒であった。その時のフィードバック誤差の値は、閾値の範囲内であった。また、その時のニューラルネットワークの出力は、図 6.6 の u_{n4} であり、負荷イナーシャ変更前より少し大きい出力値を得た。その時のフィードバック制御器の出力は、図 6.7 の u_{f4} である。求められた学習率 $\tilde{\eta}$ と回数閾値 \tilde{N}_ε での整定時間 T_{s2} は、整定時間の指標値 T_{sr} 1.5 秒のおよそ 1/1.6 である。よって、シミュレーションの解析結果から整定時間 T_{s2} が指標値 T_{sr} の範囲に収められた。

以上の結果と考察によって、学習率 η 、回数閾値 N_ϵ とサーボシステムの応答性、ロバスト性との関係が明らかになった。また、学習率 η や回数閾値 N_ϵ やフィードバック誤差や整定時間との関係の定式化から学習率 η と回数閾値 N_ϵ の更新則を導出した。さらに、提案した学習率 η と回数閾値 N_ϵ の更新則は、セルフチューニングが行えるような学習率 η と回数閾値 N_ϵ の更新則として相応しいと判断できる。

6.6 あとがき

本章では、オンライン・オフライン融合型学習法のパラメータにおける課題点と解決法を述べた。パラメータにおける課題点とは、学習率、サンプリングタイム内の繰り返しの回数閾値とサーボシステムの応答性、ロバスト性との関係がまだ明らかでないことである。加えて、今までのニューラルネットワークの学習率の更新則についての研究では、学習率の増減について学習の効率と学習の極小値との関係がまだ定式化されていないため、サーボシステムの応答性、ロバスト性との関係を結ぶことが難しい。

これらの問題点の解決法として、本研究では、サーボシステムの位置決め制御に対して、学習率、回数閾値とサーボシステムの応答性、ロバスト性との関係をシミュレーションとその解析データから明らかにした。さらに、最小二乗法によってそれらの関係の定式化を行って、セルフチューニングが行えるような学習率と回数閾値の更新則を提案した。

具体的には、まず、学習率の増減に関わるサーボシステムの特性を示した。この特性によって、学習率、回数閾値とサーボシステムの応答性、ロバスト性との関係をシミュレーションとその解析データから明らかにできた。次に、この特性の中から適切なパラメータの組み合わせを抽出して、それらのパラメータの組み合わせに最小二乗法を適用した。最後に、最小二乗法によって学習率、回数閾値と誤差最大値のデータに基づく近似曲線を求めて、学習率、回数閾値と整定時間のデータに基づく近似曲線も求めた。そして、フェラーリの解法^{[45][46]}を用いて、目的である学習率、回数閾値の更新則の導出を行った。

さらに、学習率と回数閾値の更新則に基づいて求められた値をオンライン・オフライン融合型学習法に代入して、サーボモータの位置決め制御のシミュレーションを行った。このシミュレーションでは、フィードバック誤差の指標値と整定時間の指標値を決めて、負荷イナーシャ変更前後のシミュレーションを行った。その結果として、定常誤差の最大値と整定時間はそれぞれの指標値の範囲に収められた。したがって、提案した学習率と回数閾値の更新則は、セルフチューニングが行えるような学習率と回数閾値の更新則として相応しいと判断した。

第7章 カスケードニューラルネットワーク制御

7.1 はしがき

本章では、第5章で提案したオンライン・オフライン融合型学習法を用いたニューラルネットワーク補償器にもう1つ別のニューラルネットワーク補償器を接続した新しいニューラルネットワーク制御を提案して、その背景と目的を7.2節で説明する。また、新しいニューラルネットワーク制御の理論として、制御の構成と学習法のアルゴリズムを7.3節で説明する。そして、オンライン・オフライン融合型学習法を用いた単体のニューラルネットワーク補償器と新しいニューラルネットワーク補償器のそれぞれのシミュレーション結果を7.4節で示して考察を述べる。

7.2 オンライン・オフライン融合型学習法と併用する新しい制御法の提案

第5章では、オンライン学習法とオフライン学習法を融合させたオンライン・オフライン融合型学習法を提案した。また、提案法をサーボモータの位置決め制御のシミュレーションに適用して、提案法が負荷イナーシャの変更と目標軌道の周波数の変更に対応できることを示した。そして、第6章では、オンライン・オフライン融合型学習法の学習率、サンプリングタイム内の繰り返しループの回数閾値とサーボシステムの応答性、ロバスト性との関係を明らかにさせて、最小二乗法でそれらの関係の定式化を行い、セルフチューニングが行えるような学習率と回数閾値の更新則を提案して、サーボモータの位置決め制御のシミュレーション結果により提案した更新則の有効性を述べた。

本章では、単体のニューラルネットワーク補償器にもう1つ接続した新しいニューラルネットワーク制御法を提案する。第5章および第6章では、オンライン・オフライン融合型学習法を用いてフィードバック制御器の補償を行った。この時、ニューラルネットワーク補償器は1つであった。補償器の効果によって、負荷イナーシャの変更と目標軌道の周波数変更に対応できた。なお、フィードバック誤差の閾値（許容範囲値）を変更すれば、その閾値に収まるようにニューラルネットワーク補償器が働いて、フィードバック制御器を補償することができる。

ここで、負荷変動や外乱による制御対象の影響をさらに抑制し、位置決め精度を高めたい場合は補償器をどのように構成すれば良いか考える。本研究では、サーボシステムの位置決め制御に対して、新しいニューラルネットワーク補償器を単体のニューラルネットワーク補償器と並列に接続する。もう1つのニューラルネットワーク補償器による学習の効果が加わることでフィードバック制御器の更なる補償が行われ、位置決め精度がより高まる。本研究では、このような制御法をカスケードニューラルネットワーク制御と定義する [47] [48]。

7.3 カスケードニューラルネットワーク制御の理論

本節では、カスケードニューラルネットワーク制御の理論について、まず、カスケードニューラルネットワーク制御とは、どのような構成であるかを 7.3.1 項に説明する。その構成としては、2 種類の構成が考えられる。それらの構成の共通点は、1 つ目のニューラルネットワーク補償器の誤差信号および出力を追加したニューラルネットワーク補償器の誤差信号に用いる点である。そして、相違点は、追加したニューラルネットワーク補償器の誤差信号に 1 サンプルタイム前のニューラルネットワーク補償器の出力を戻すか否かという点である。次に、カスケードニューラルネットワーク制御のモデルとシミュレーションのモデルを 7.3.2 項に示す。

7.3.1 カスケードニューラルネットワーク制御の構成

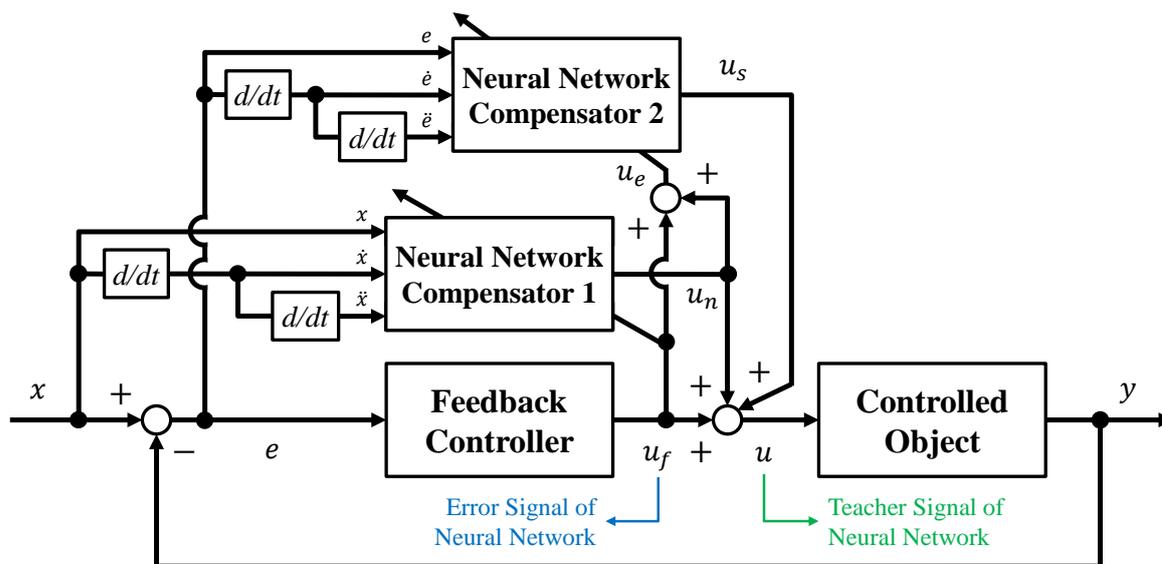


図 7.1 ポジティブカスケードニューラルネットワーク制御の構成

図 7.1 に示されるカスケードニューラルネットワーク制御の構成では、フィードバック制御器の出力、すなわち、下段のニューラルネットワーク補償器の誤差信号 u_f と下段のニューラルネットワーク補償器の出力信号 u_n を加え合わせたものが上段のニューラルネットワーク補償器の誤差信号 u_e である。また、下段の補償器の誤差信号 u_f に下段の補償器の出力信号 u_n と上段の補償器の出力信号 u_s を加え合わせたものが制御対象への入力、すなわち、ニューラルネットワークの教師信号 u である。

上段のニューラルネットワークは、教師信号 u とフィードバック誤差 e との逆モデルを獲得して、負荷変動や外乱で生じた誤差を打ち消す。従って、学習を通じて出力信号 u_s が教師信号 u と一致させるように働き、フィードバック制御器を補償する。そのため、ニューラルネットワークの誤差が最小になるように、入力層と中間層との間の重み、中間層と出力層との間の重みを更新させる。その手順は、次の式(7.1)から式(7.9)に示される。

$$\begin{aligned}
E_s &= \frac{1}{2} \sum_{k=1}^a (u_s(k) - u(k))^2 = \frac{1}{2} \sum_{k=1}^a \left(u_s(k) - (u_f(k) + u_n(k) + u_s(k)) \right)^2 \\
&= \frac{1}{2} \sum_{k=1}^a - (u_f(k) - u_n(k))^2 = \frac{1}{2} \sum_{k=1}^a (-u_e(k))^2
\end{aligned} \tag{7.1}$$

$$\frac{\partial E_s}{\partial u_s(k)} = u_s(k) - u(k) = -u_e(k) \tag{7.2}$$

$$\frac{\partial u_s(k)}{\partial v(j,k)} = \frac{-2 \cdot (-y(j)) \cdot e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} = \frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} \cdot y(j) \tag{7.3}$$

$$\frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} = \frac{1}{2} \left(1 - \left(\frac{1 - e^{-\sum_{j=1}^b v(j,k)y(j)}}{1 + e^{-\sum_{j=1}^b v(j,k)y(j)}} \right)^2 \right) = \frac{1}{2} \left(1 - (u_s(k))^2 \right) \tag{7.4}$$

$$\frac{\partial u_s(k)}{\partial y(j)} = \frac{-2 \cdot (-v(j,k)) \cdot e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} = \frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{\left(1 + e^{-\sum_{j=1}^b v(j,k)y(j)}\right)^2} \cdot v(j,k) \tag{7.5}$$

$$\frac{\partial y(j)}{\partial w(i,j)} = \frac{-2 \cdot (-x(i)) \cdot e^{-\sum_{i=1}^c w(i,j)x(i)}}{\left(1 + e^{-\sum_{i=1}^c w(i,j)x(i)}\right)^2} = \frac{2e^{-\sum_{i=1}^c w(i,j)x(i)}}{\left(1 + e^{-\sum_{i=1}^c w(i,j)x(i)}\right)^2} \cdot x(i) \tag{7.6}$$

$$\frac{2e^{-\sum_{i=1}^c w(i,j)x(i)}}{\left(1 + e^{-\sum_{i=1}^c w(i,j)x(i)}\right)^2} = \frac{1}{2} \left(1 - \left(\frac{1 - e^{-\sum_{i=1}^c w(i,j)x(i)}}{1 + e^{-\sum_{i=1}^c w(i,j)x(i)}} \right)^2 \right) = \frac{1}{2} \left(1 - (y(j))^2 \right) \tag{7.7}$$

$$\begin{aligned}
\frac{\partial E_s}{\partial v(j,k)} &= \frac{\partial E_s}{\partial u_s(k)} \cdot \frac{\partial u_s(k)}{\partial v(j,k)} \\
&= \frac{1}{2} (-u_e(k)) \cdot \left(1 - (u_s(k))^2 \right) \cdot y(j) = \delta_1 \cdot y(j)
\end{aligned} \tag{7.8}$$

$$\begin{aligned}
\frac{\partial E_s}{\partial w(i,j)} &= \frac{\partial E_s}{\partial u_s(k)} \cdot \frac{\partial u_s(k)}{\partial y(j)} \cdot \frac{\partial y(j)}{\partial w(i,j)} \\
&= (-u_e(k)) \cdot \frac{1}{2} \left(1 - (u_s(k))^2 \right) \cdot v(j,k) \cdot \frac{1}{2} \left(1 - (y(j))^2 \right) \cdot x(i) \\
&= \delta_1 \cdot v(j,k) \cdot \frac{1}{2} \left(1 - (y(j))^2 \right) \cdot x(i) = \delta_2 \cdot x(i)
\end{aligned} \tag{7.9}$$

ここで、式(7.1)は誤差信号 E_s を表し、式(7.2)は E_s におけるニューラルネットワークの出力 $u_s(k)$ の偏微分である。 a 、 b 、 c それぞれは、入力層、中間層、出力層のニューロンの数である。式(7.3)は $u_s(k)$ における $v(j,k)$ の偏微分であり、式(7.4)は式(7.3)の右辺にある $y(j)$ の係数を $u_n(k)$ で表している。式(7.5)は $u_n(k)$ における $y(j)$ の偏微分である。式(7.6)は $y(j)$ における $w(i,j)$ の偏微分であり、式(7.7)は式(7.6)の右辺にある $x(i)$ の係数を $y(j)$ で表している。そして、式(7.8)と式(7.9)は式(7.2)から式(7.7)を通して導出される。

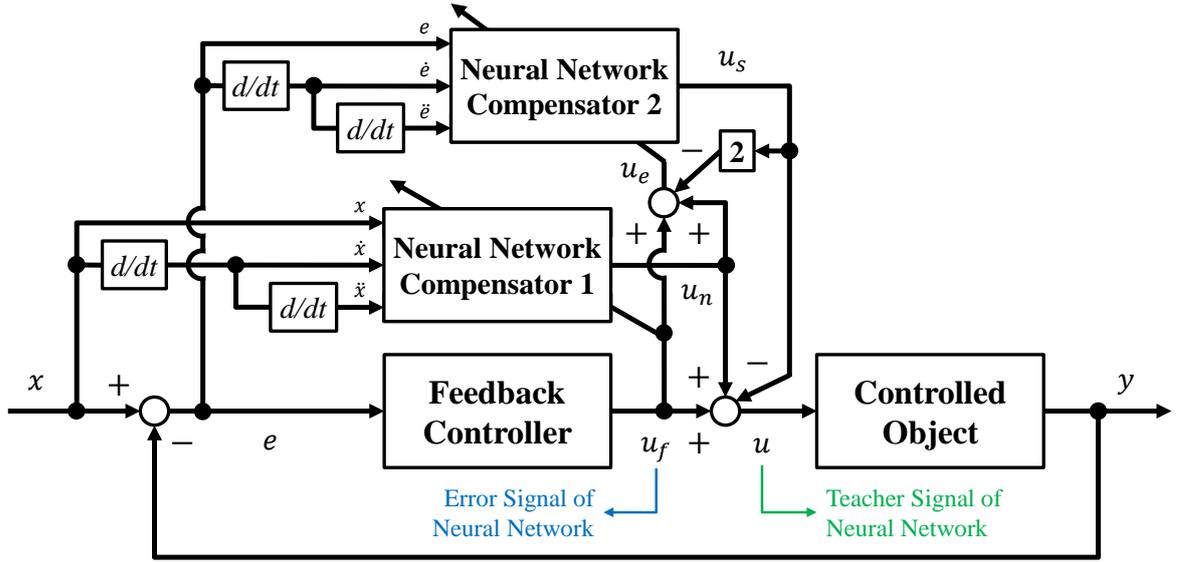


図 7.2 ネガティブカスケードニューラルネットワーク制御の構成

図 7.2 に示されるカスケードニューラルネットワーク制御では、下段の補償器の誤差信号 u_f と下段の補償器の出力信号 u_n に加えて、上段の補償器の出力信号 u_s の 2 倍を加え合わせたものが上段の補償器の誤差信号 u_e である。この構成の場合では、出力信号 u_s を取得するために 1 サンプルタイム前の出力信号 $u_s(k-1)$ を取得する必要があるため、 u_s を擬似的な信号として自身に入力する。また、下段の補償器の誤差信号 u_f に下段の補償器の出力 u_n と上段の補償器の出力 u_s を加え合わせたものが教師信号 u になる。

上段のニューラルネットワークは、教師信号 u とフィードバック誤差 e との逆モデルを獲得して、負荷変動や外乱で生じた誤差を打ち消す。従って、学習を通じて出力信号 u_s が教師信号 u と一致させるように働き、フィードバック制御器を補償する。そのため、ニューラルネットワークの誤差が最小になるように、入力層と中間層との間の重み、中間層と出力層との間の重みを更新させる。その手順は、次の式(7.10)から式(7.18)に示される。

$$\begin{aligned}
 E_s &= \frac{1}{2} \sum_{k=1}^a (u_s(k) - u(k))^2 = \frac{1}{2} \sum_{k=1}^a (u_s(k) - (u_f(k) + u_n(k) - u_s(k)))^2 \\
 &= -\frac{1}{2} \sum_{k=1}^a (u_f(k) + u_n(k) - 2u_s(k))^2 \approx -\frac{1}{2} \sum_{k=1}^a (u_f(k) + u_n(k) - 2u_s(k-1))^2 \quad (7.10) \\
 &= -\frac{1}{2} \sum_{k=1}^a u_e(k)
 \end{aligned}$$

$$\frac{\partial E_s}{\partial u_s(k)} = u_s(k) - u(k) = -u_e(k) \quad (7.11)$$

$$\frac{\partial u_s(k)}{\partial v(j,k)} = \frac{-2 \cdot (-y(j)) \cdot e^{-\sum_{j=1}^b v(j,k)y(j)}}{(1 + e^{-\sum_{j=1}^b v(j,k)y(j)})^2} = \frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{(1 + e^{-\sum_{j=1}^b v(j,k)y(j)})^2} \cdot y(j) \quad (7.12)$$

$$\frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{(1 + e^{-\sum_{j=1}^b v(j,k)y(j)})^2} = \frac{1}{2} \left(1 - \left(\frac{1 - e^{-\sum_{j=1}^b v(j,k)y(j)}}{1 + e^{-\sum_{j=1}^b v(j,k)y(j)}} \right)^2 \right) = \frac{1}{2} (1 - (u_s(k))^2) \quad (7.13)$$

$$\frac{\partial u_s(k)}{\partial y(j)} = \frac{-2 \cdot (-v(j,k)) \cdot e^{-\sum_{j=1}^b v(j,k)y(j)}}{(1 + e^{-\sum_{j=1}^b v(j,k)y(j)})^2} = \frac{2e^{-\sum_{j=1}^b v(j,k)y(j)}}{(1 + e^{-\sum_{j=1}^b v(j,k)y(j)})^2} \cdot v(j,k) \quad (7.14)$$

$$\frac{\partial y(j)}{\partial w(i,j)} = \frac{-2 \cdot (-x(i)) \cdot e^{-\sum_{i=1}^c w(i,j)x(i)}}{(1 + e^{-\sum_{i=1}^c w(i,j)x(i)})^2} = \frac{2e^{-\sum_{i=1}^c w(i,j)x(i)}}{(1 + e^{-\sum_{i=1}^c w(i,j)x(i)})^2} \cdot x(i) \quad (7.15)$$

$$\frac{2e^{-\sum_{i=1}^c w(i,j)x(i)}}{(1 + e^{-\sum_{i=1}^c w(i,j)x(i)})^2} = \frac{1}{2} \left(1 - \left(\frac{1 - e^{-\sum_{i=1}^c w(i,j)x(i)}}{1 + e^{-\sum_{i=1}^c w(i,j)x(i)}} \right)^2 \right) = \frac{1}{2} (1 - (y(j))^2) \quad (7.16)$$

$$\begin{aligned} \frac{\partial E_s}{\partial v(j,k)} &= \frac{\partial E_s}{\partial u_s(k)} \cdot \frac{\partial u_s(k)}{\partial v(j,k)} \\ &= \frac{1}{2} (-u_e(k)) \cdot (1 - (u_s(k))^2) \cdot y(j) = \delta_3 \cdot y(j) \end{aligned} \quad (7.17)$$

$$\begin{aligned} \frac{\partial E_s}{\partial w(i,j)} &= \frac{\partial E_s}{\partial u_s(k)} \cdot \frac{\partial u_s(k)}{\partial y(j)} \cdot \frac{\partial y(j)}{\partial w(i,j)} \\ &= (-u_e(k)) \cdot \frac{1}{2} (1 - (u_s(k))^2) \cdot v(j,k) \cdot \frac{1}{2} (1 - (y(j))^2) \cdot x(i) \\ &= \delta_3 \cdot v(j,k) \cdot \frac{1}{2} (1 - (y(j))^2) \cdot x(i) = \delta_4 \cdot x(i) \end{aligned} \quad (7.18)$$

ここで、式(7.10)は誤差信号 E_s を表し、式(7.11)は E_s におけるニューラルネットワークの出力 $u_s(k)$ の偏微分である。なお、 a 、 b 、 c それぞれは、入力層、中間層、出力層のニューロンの数である。式(7.12)は $u_s(k)$ における $v(j,k)$ の偏微分であり、式(7.13)は式(7.12)の右辺にある $y(j)$ の係数を $u_s(k)$ で表している。式(7.14)は $u_s(k)$ における $y(j)$ の偏微分である。式(7.15)は $y(j)$ における $w(i,j)$ の偏微分であり、式(7.16)は式(7.15)の右辺にある $x(i)$ の係数を $y(j)$ で表している。そして、式(7.17)と式(7.18)は式(7.11)から式(7.16)を通して導出される。

7.3.2 カスケードニューラルネットワーク制御のモデル

次の図 7.3 にあるモータの動的モデルは、第 4 章で説明したように電気的エネルギーと機械的エネルギーの変換に基づいた電機子電圧 V と回転位置 θ との数学的な関係である。また、上段の補償器で構成されるニューラルネットワークの入力層では、 e, \dot{e}, \ddot{e} を接続する。電機子電圧 V と回転角速度 $\dot{\theta}$ の関係は、トルク定数 K_t が逆起電力定数 K_e と等しい関係を用いる。そして、回転角加速度は、トルクの方程式と電圧の方程式の連立によって求められる。次の図 7.4 にあるモータの動的モデルについても図 7.3 と同様のモデルを用いる。

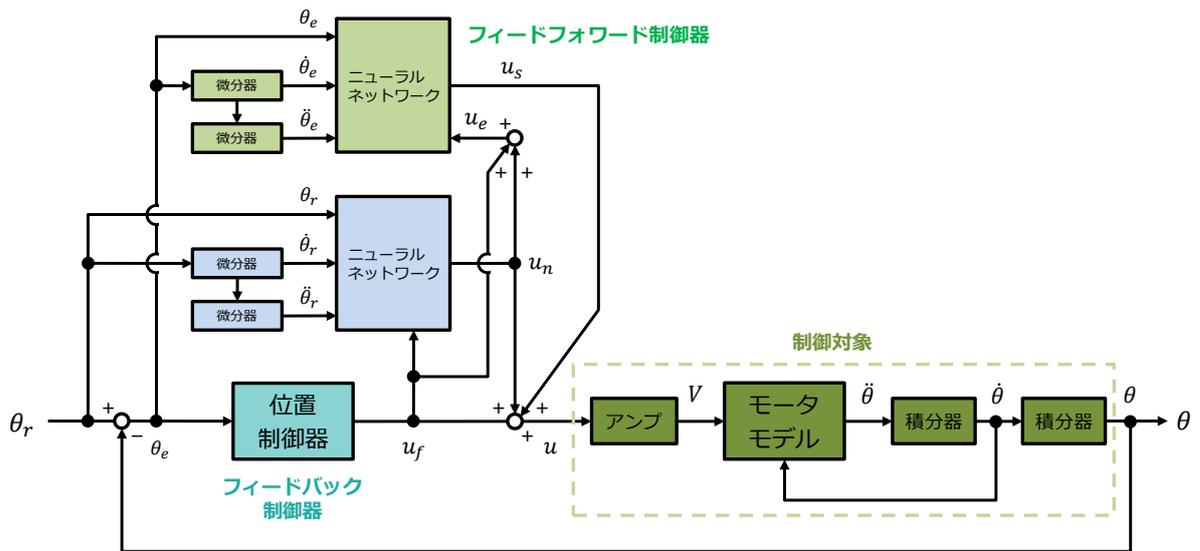


図 7.3 ポジティブカスケードニューラルネットワーク制御を用いたサーボ制御の構成

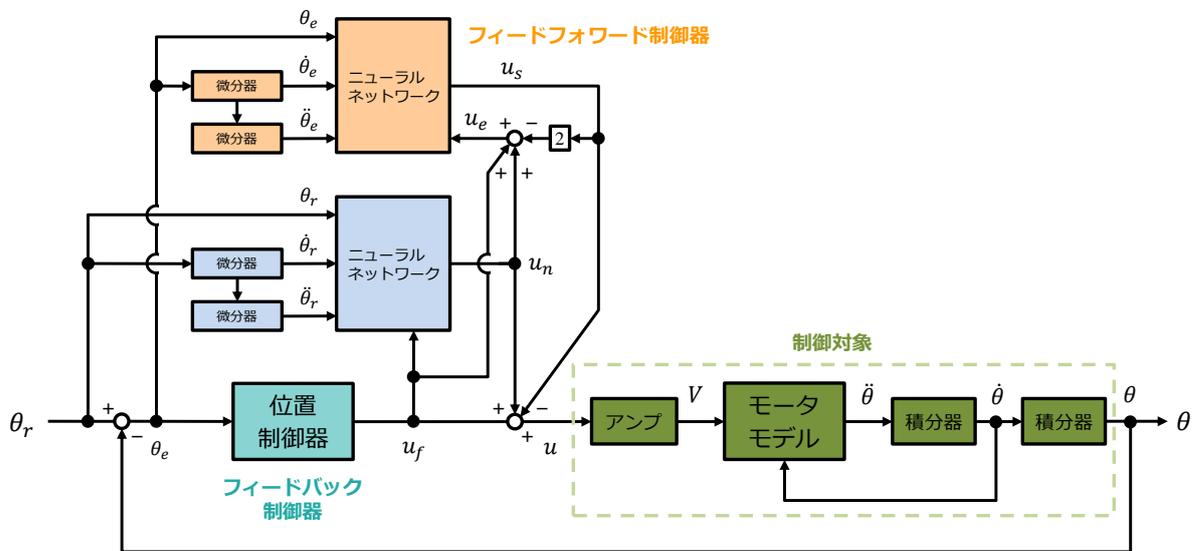


図 7.4 ネガティブカスケードニューラルネットワーク制御を用いたサーボ制御の構成

図 7.3 にある上段の補償器の誤差信号と図 7.4 にある上段の補償器の誤差信号を比較してみる。図 7.3 にあるニューラルネットワークの誤差信号 u_e は E_s と等しい関係であり、図 7.4 にあるニューラルネットワーク補償器の誤差信号 u_e は E_s と等しい関係である。図 7.3 の誤差信号 u_e は、フィードバック制御器の出力、すなわち、下段の補償器の誤差信号 u_f に下段の補償器の出力 u_n を加えたものである。一方で、図 7.4 の誤差信号 u_e は、フィードバック制御器の出力、すなわち、下段の補償器の誤差信号 u_f に下段補償器の出力 u_n と上段の補償器で得られた出力 u_s を戻して 2 倍したものを加えている。以上を参考に、カスケードニューラルネットワーク制御の構成をシミュレーションのモデルで表すと、図 7.5 と図 7.6 にそれぞれ示される。

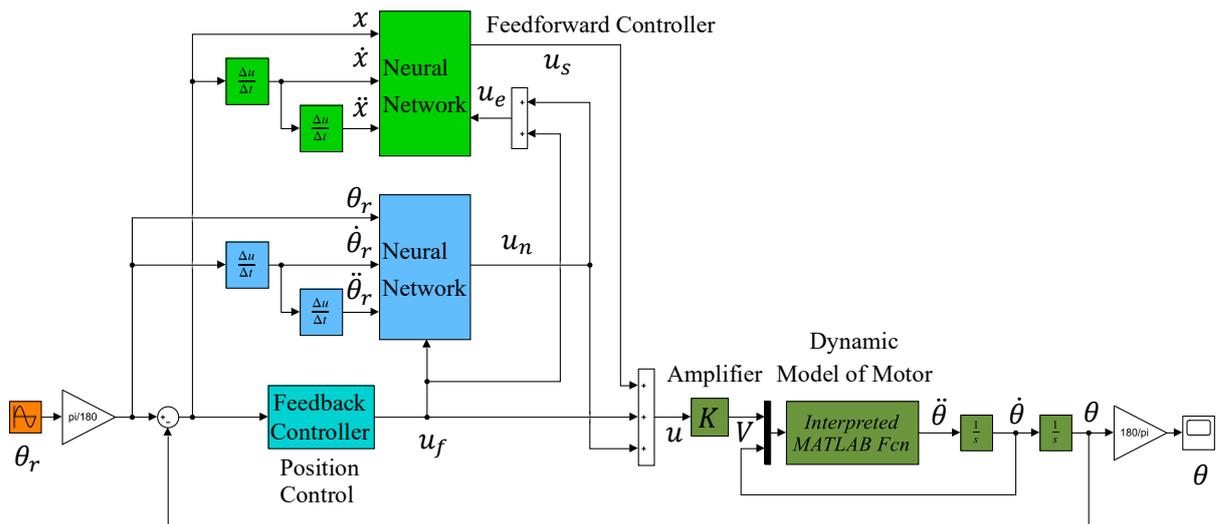


図 7.5 ポジティブカスケードニューラルネットワーク制御を用いたサーボシステムのシミュレーションモデル

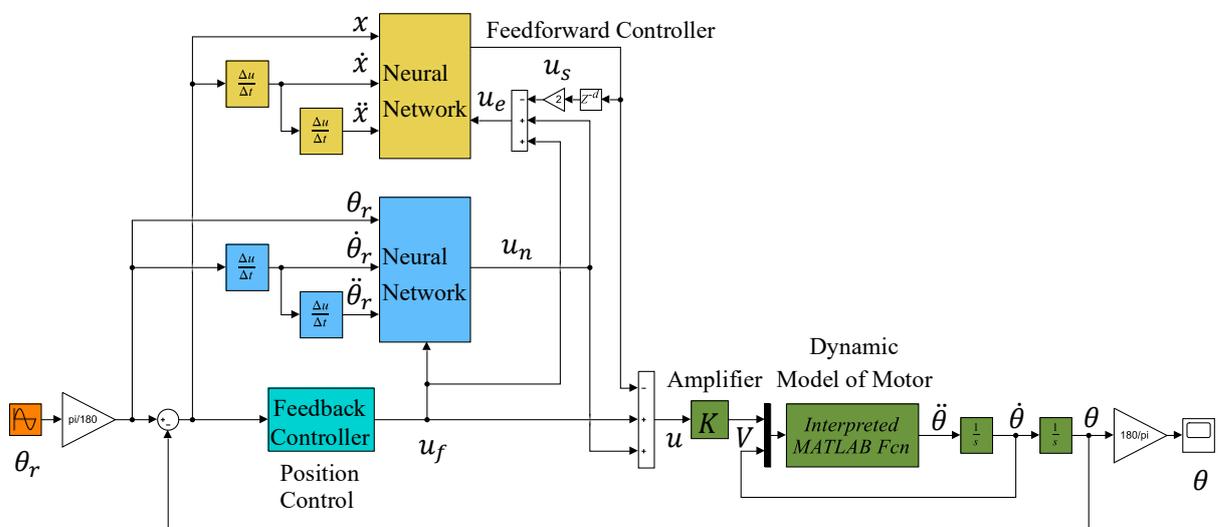


図 7.6 ネガティブカスケードニューラルネットワーク制御を用いたサーボシステムのシミュレーションモデル

図 7.5 と図 7.6 それぞれのモデルにおいて、解析するためのサンプリング時間は 1 ms であり、微積分の解析法は 4 次ルンゲクッタ法である。 θ_r のブロックは、目標位置の生成器であり、回転角 ± 90 度の振幅、1 Hz の周波数、0 deg の位相差の正弦波である角度軌道を生成する。また、図 7.6 にある z^{-d} のブロックは、入力信号を遅延するためのブロックである。MATLAB/Simulink のシミュレーションにおいて、上段のニューラルネットワーク補償器のブロックから出力された信号 u_s を戻すと代数ループになる。そのため、 z^{-d} のブロックに u_s を接続する。

次に、カスケードニューラルネットワーク制御の学習法のフローチャートについて、図 7.7 と図 7.8 にそれぞれ示される。

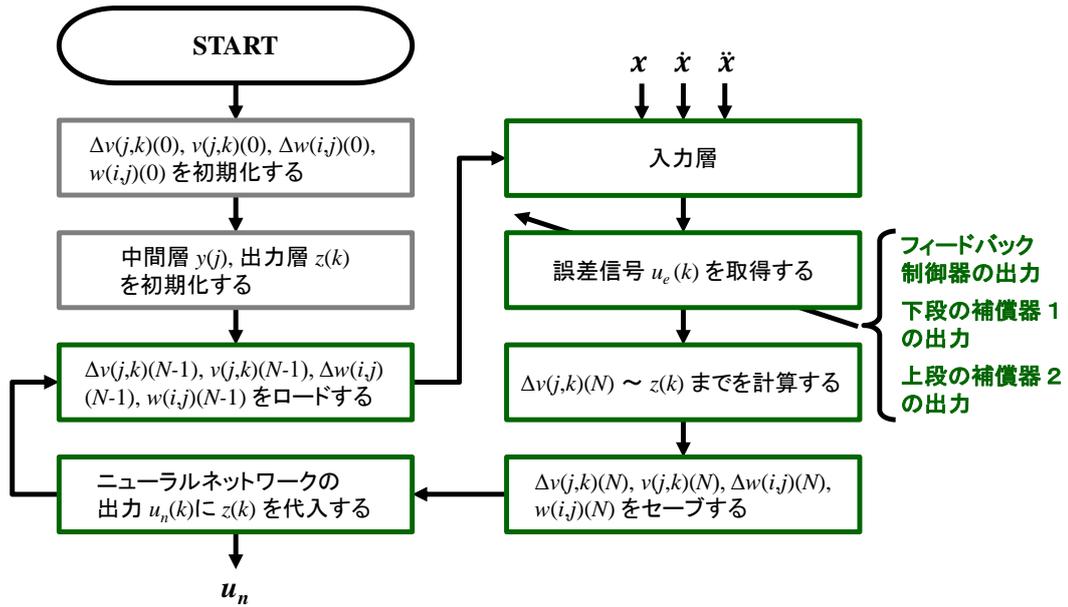


図 7.7 ポジティブカスケードニューラルネットワーク制御のフローチャート

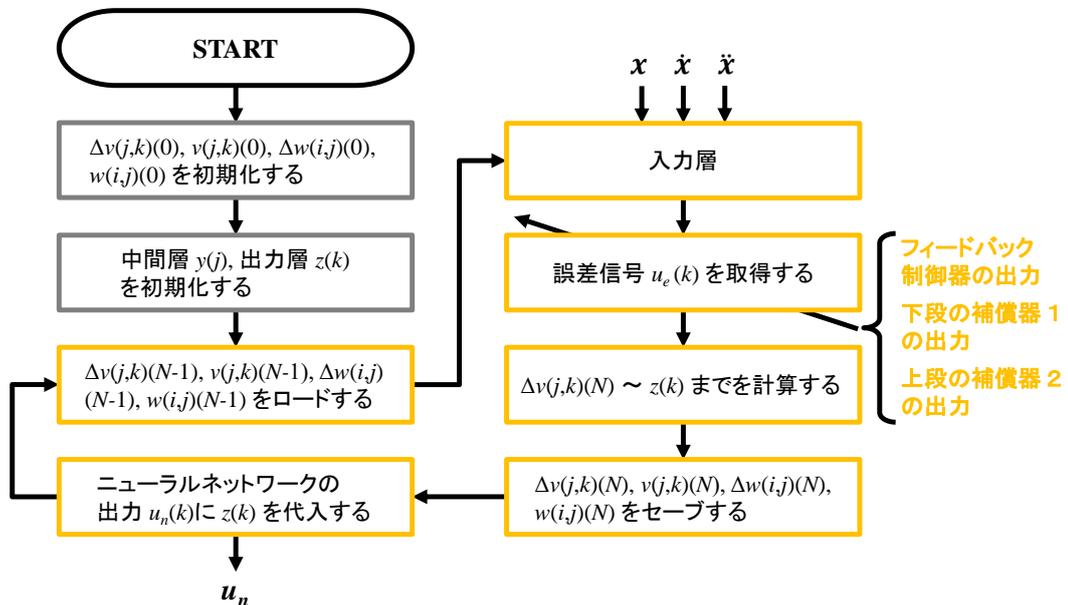


図 7.8 ネガティブカスケードニューラルネットワーク制御のフローチャート

上段のニューラルネットワーク補償器では、オンライン学習法のフローチャートを利用することによって学習の効果を確認する。まず、 $\Delta v(j,k)(N)$ 、 $v(j,k)(N)$ 、 $w(i,j)(N)$ 、 $y(j)$ 、 $z(k)$ はゼロの値で初期化されて、 $\Delta w(i,j)(N)$ はランダムな値で初期化される。次に、位置、速度、加速度の指令値 x 、 \dot{x} 、 \ddot{x} はニューラルネットワークの入力として、誤差信号 E_s は $u_e(k)$ から取得される。さらに、 $\Delta v(j,k)(N)$ 、 $v(j,k)(N)$ 、 $\Delta w(i,j)(N)$ 、 $w(i,j)(N)$ 、 $y(j)$ 、 $z(k)$ が計算される。最後に、 $z(k)$ はニューラルネットワークの出力 $u_n(k)$ とする。

7.4 結果と考察

本節では、位置決め制御のシミュレーションの結果を 7.4.1 項に示して、結果の考察を 7.4.2 項に述べる。7.4.1 項では、負荷イナーシャ変更前後において、オンライン・オフライン融合型学習法を用いたニューラルネットワーク制御とカスケードニューラルネットワーク制御をそれぞれ用いた場合の位置決め制御の応答性とロバスト性に関する結果が示される。

7.4.1 負荷イナーシャ変更前後の応答性とロバスト性

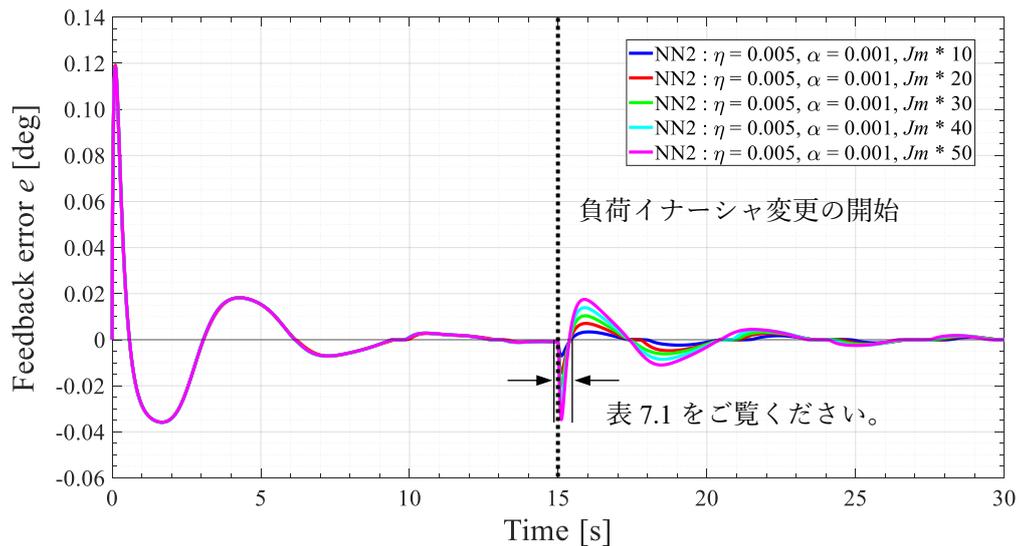


図 7.9 ポジティブカスケードニューラルネットワーク制御におけるフィードバック誤差

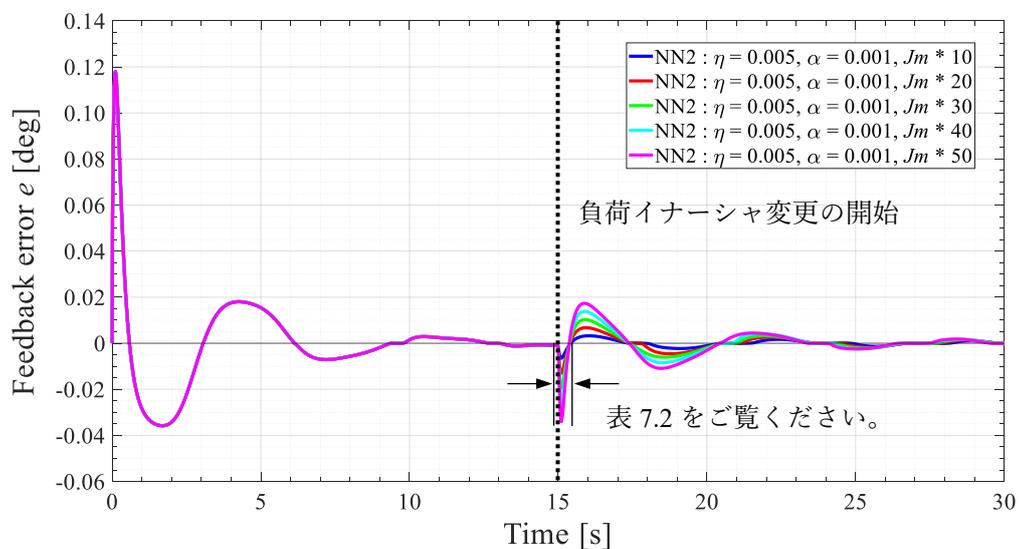


図 7.10 ネガティブカスケードニューラルネットワーク制御におけるフィードバック誤差

オンライン・オフライン融合型学習法のパラメータは $\eta = 0.004$ 、 $\alpha = 0.001$ 、 $N_\varepsilon = 1$ であり、提案法のパラメータは $\eta = 0.004$ 、 $\alpha = 0.001$ である。また、シミュレーション開始から 15 秒後に負荷イナーシャ値の大きさをステップ状に $J_L = 10 J_M = 2.25 \times 10^{-6} \text{ kgm}^2$ から $J_L = 50 J_M = 11.25 \times 10^{-6} \text{ kgm}^2$ まで 10 倍ずつ増大させる。図 7.9 と図 7.10 は、負荷イナーシャ変更前後のフィードバック誤差 e の結果を示している。図 7.11 と図 7.12 は、提案法の上段のニューラルネットワーク補償器の出力 u_s の結果を示している。そして、オンライン・オフライン融合型学習法と提案法において、負荷イナーシャ変更直後のフィードバック誤差の振幅値および上段の補償器の出力値は、表 7.1 と表 7.2 にそれぞれ示される。

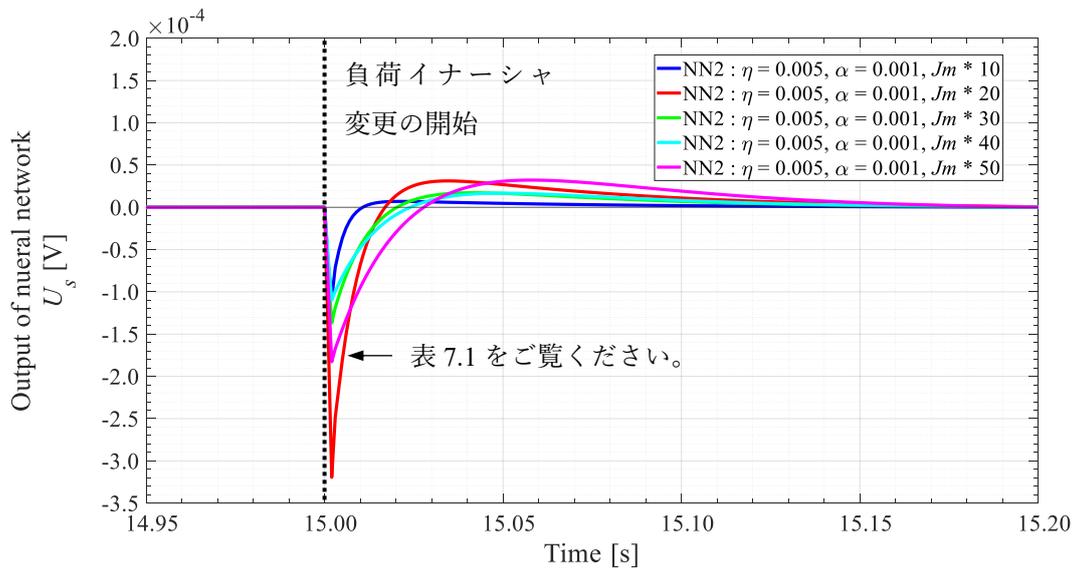


図 7.11 ポジティブカスケードニューラルネットワーク制御における上段の補償器の出力

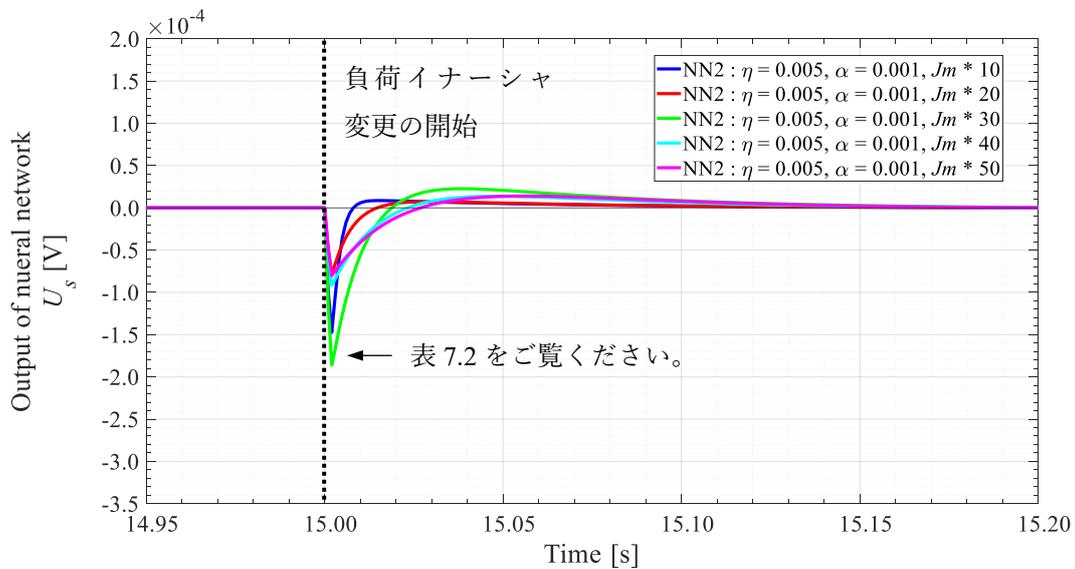


図 7.12 ネガティブカスケードニューラルネットワーク制御における上段の補償器の出力

表 7.1 単独制御とポジティブカスケードニューラルネットワーク制御の効果比較
(負荷イナーシャ変更直後の振幅最大値)

補償器の構成	単独	ポジティブカスケード		評価
	NN1 (融合型)	NN1 (融合型)	NN2 (オンライン)	
負荷慣性 J_L	フィードバック誤差 e			向上率
$10 J_M \text{ kgm}^2$	$- 6.696 \times 10^{-3} \text{ deg}$	$- 7.032 \times 10^{-3} \text{ deg}$		$- 5.02 \%$
$20 J_M \text{ kgm}^2$	$- 13.140 \times 10^{-3} \text{ deg}$	$- 14.390 \times 10^{-3} \text{ deg}$		$- 9.52 \%$
$30 J_M \text{ kgm}^2$	$- 19.850 \times 10^{-3} \text{ deg}$	$- 20.210 \times 10^{-3} \text{ deg}$		$- 1.81 \%$
$40 J_M \text{ kgm}^2$	$- 26.870 \times 10^{-3} \text{ deg}$	$- 27.150 \times 10^{-3} \text{ deg}$		$- 1.04 \%$
$50 J_M \text{ kgm}^2$	$- 34.170 \times 10^{-3} \text{ deg}$	$- 34.730 \times 10^{-3} \text{ deg}$		$- 1.64 \%$
負荷慣性 J_L	補償器の出力 u_n		補償器の出力 u_s	向上率
$10 J_M \text{ kgm}^2$	$- 28.270 \times 10^{-3} \text{ v}$		$- 1.080 \times 10^{-4} \text{ v}$	$+ 0.38 \%$
$20 J_M \text{ kgm}^2$	$- 28.560 \times 10^{-3} \text{ v}$		$- 3.200 \times 10^{-4} \text{ v}$	$+ 1.12 \%$
$30 J_M \text{ kgm}^2$	$- 28.850 \times 10^{-3} \text{ v}$		$- 1.370 \times 10^{-4} \text{ v}$	$+ 0.47 \%$
$40 J_M \text{ kgm}^2$	$- 29.150 \times 10^{-3} \text{ v}$		$- 1.100 \times 10^{-4} \text{ v}$	$+ 0.38 \%$
$50 J_M \text{ kgm}^2$	$- 29.450 \times 10^{-3} \text{ v}$		$- 1.830 \times 10^{-4} \text{ v}$	$+ 0.62 \%$

表 7.2 単独制御とネガティブカスケードニューラルネットワーク制御の効果比較
(負荷イナーシャ変更直後の振幅最大値)

補償器の構成	単独	ネガティブカスケード		評価
	NN1 (融合型)	NN1 (融合型)	NN2 (オンライン)	
負荷慣性 J_L	フィードバック誤差 e			向上率
$10 J_M \text{ kgm}^2$	$- 6.696 \times 10^{-3} \text{ deg}$	$- 6.523 \times 10^{-3} \text{ deg}$		$+ 2.58 \%$
$20 J_M \text{ kgm}^2$	$- 13.140 \times 10^{-3} \text{ deg}$	$- 13.020 \times 10^{-3} \text{ deg}$		$+ 0.91 \%$
$30 J_M \text{ kgm}^2$	$- 19.850 \times 10^{-3} \text{ deg}$	$- 19.620 \times 10^{-3} \text{ deg}$		$+ 1.16 \%$
$40 J_M \text{ kgm}^2$	$- 26.870 \times 10^{-3} \text{ deg}$	$- 26.720 \times 10^{-3} \text{ deg}$		$+ 0.56 \%$
$50 J_M \text{ kgm}^2$	$- 34.170 \times 10^{-3} \text{ deg}$	$- 34.010 \times 10^{-3} \text{ deg}$		$+ 0.47 \%$
負荷慣性 J_L	補償器の出力 u_n		補償器の出力 u_s	向上率
$10 J_M \text{ kgm}^2$	$- 28.270 \times 10^{-3} \text{ v}$		$- 1.470 \times 10^{-4} \text{ v}$	$+ 0.52 \%$
$20 J_M \text{ kgm}^2$	$- 28.560 \times 10^{-3} \text{ v}$		$- 0.790 \times 10^{-4} \text{ v}$	$+ 0.27 \%$
$30 J_M \text{ kgm}^2$	$- 28.850 \times 10^{-3} \text{ v}$		$- 1.860 \times 10^{-4} \text{ v}$	$+ 0.64 \%$
$40 J_M \text{ kgm}^2$	$- 29.150 \times 10^{-3} \text{ v}$		$- 0.920 \times 10^{-4} \text{ v}$	$+ 0.32 \%$
$50 J_M \text{ kgm}^2$	$- 29.450 \times 10^{-3} \text{ v}$		$- 0.800 \times 10^{-4} \text{ v}$	$+ 0.27 \%$

7.4.2 電機子巻線抵抗の変更前後の応答性とロバスト性

モータの電機子巻線抵抗は、モータの周囲温度によって抵抗値が変化する。第4章の表4.1で示したブラスレスDCモータの電機子巻線抵抗は、周囲温度が25°Cでの抵抗値である。ここで、周囲温度を T 、周囲温度25°Cでの電機子巻線抵抗を R_{25} 、銅の抵抗温度係数を α 、周囲温度 T での電機子巻線抵抗を R_T と定義すると、 R_T は式(7.1)に示される。

$$R_T = R_{25}[1 + \alpha(T - 25)] = R_{25}[1 + 0.0039(T - 25)] \quad (7.1)$$

また、時刻 t での巻線温度を T_w 、雰囲気温度を T_b 、熱平衡状態での巻線温度を T_e 、熱時定数を τ_h とする。モータを駆動して巻線温度が T_b から T_e に到達した時、 T_w は式(7.2)に示され、モータを停止して巻線温度が T_e から T_b に到達した時、 T_w は式(7.3)に示される。

$$T_w = (T_e - T_b) \left(1 - e^{-\frac{t}{\tau_h}} \right) + T_b \quad (7.2)$$

$$T_w = (T_e - T_b) e^{-\frac{t}{\tau_h}} + T_b \quad (7.3)$$

表4.1で示したブラスレスDCモータの熱時定数は $\tau_h = 1.69$ sであり、巻線温度が $T_b = 25^\circ\text{C}$ から $T_e = 30^\circ\text{C}$ 、 $T_b = 25^\circ\text{C}$ から $T_e = 60^\circ\text{C}$ に到達した時のシミュレーション結果は図7.13と図7.14にそれぞれ示される。図7.13は、上段のニューラルネットワーク補償器の出力を示している。図7.14は、図7.13の14.95秒から15.20秒までの拡大図である。

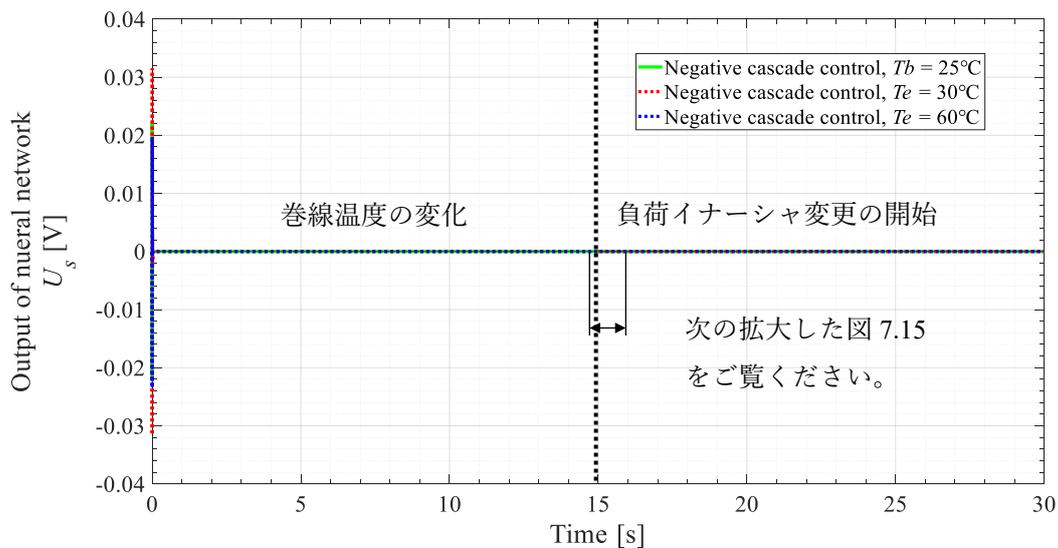


図7.13 巻線温度30°Cまたは60°C到達時における上段のニューラルネットワーク補償器の出力 (負荷イナーシャ変更前後)

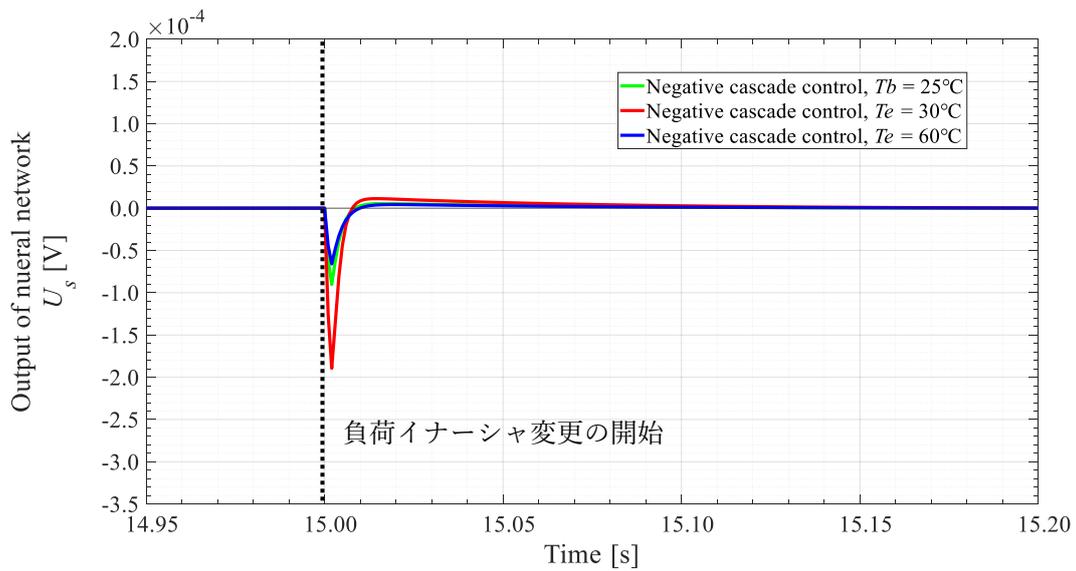


図 7.14 巻線温度 30°C または 60°C 到達時における上段のニューラルネットワーク補償器の出力 (図 7.13 の拡大図)

7.4.3 周期外乱印加後の応答性とロバスト性

制御対象 u に 24 Hz の周期外乱 $D(t) = 0.01 \sin(150.8 t)$ を印加した場合のシミュレーション結果を図 7.15 から図 7.17 にそれぞれ示される。図 7.15 は、負荷イナーシャ変更前後のフィードバック誤差を示している。図 7.16 は、上段のニューラルネットワーク補償器の出力を示している。図 7.17 は、図 7.16 の 14.95 秒から 15.20 秒までの拡大図である。

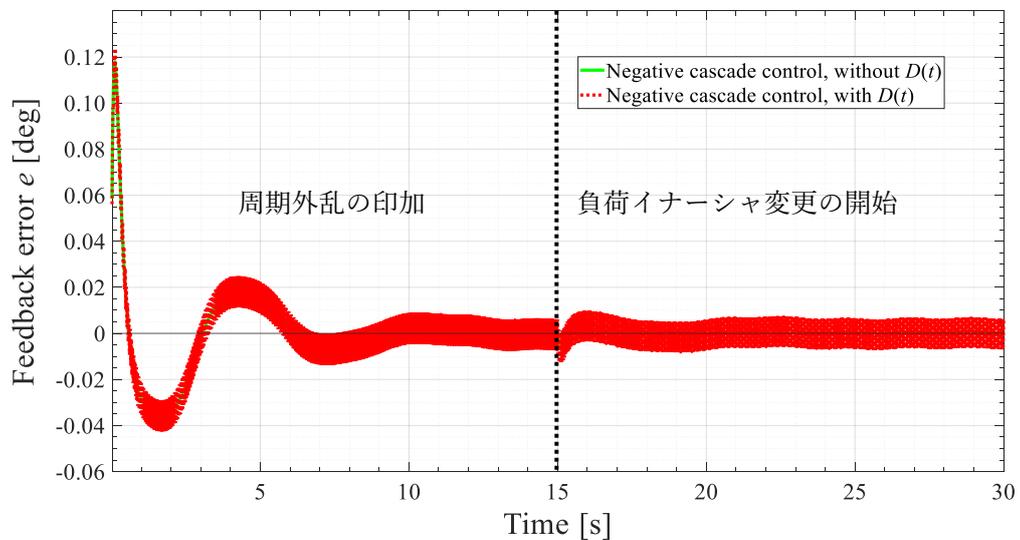


図 7.15 周期外乱 24 Hz の印加有無におけるフィードバック誤差 (負荷イナーシャ変更前後)

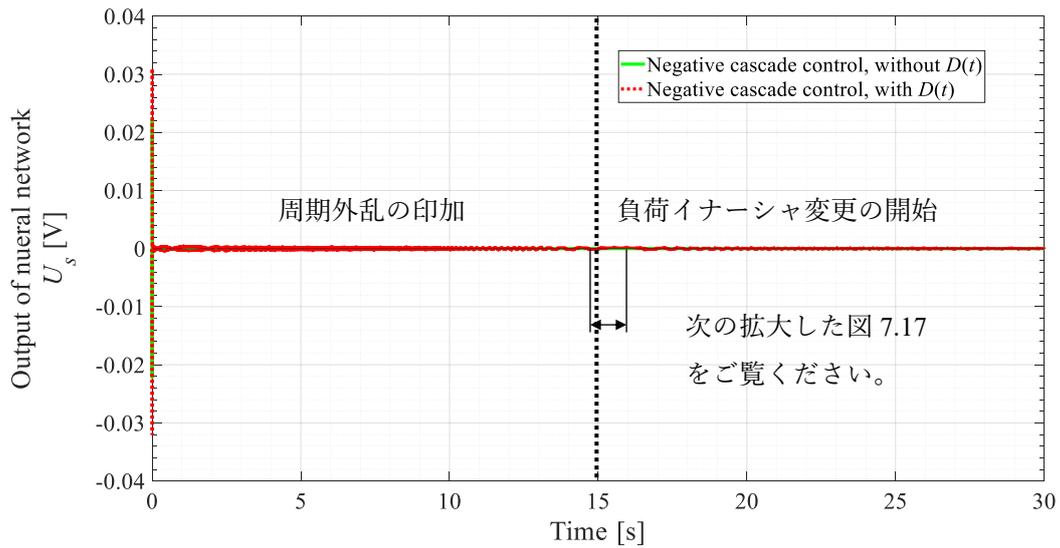


図 7.16 周期外乱 24 Hz の印加有無における上段のニューラルネットワーク補償器の出力 (負荷イナーシャ変更前後)

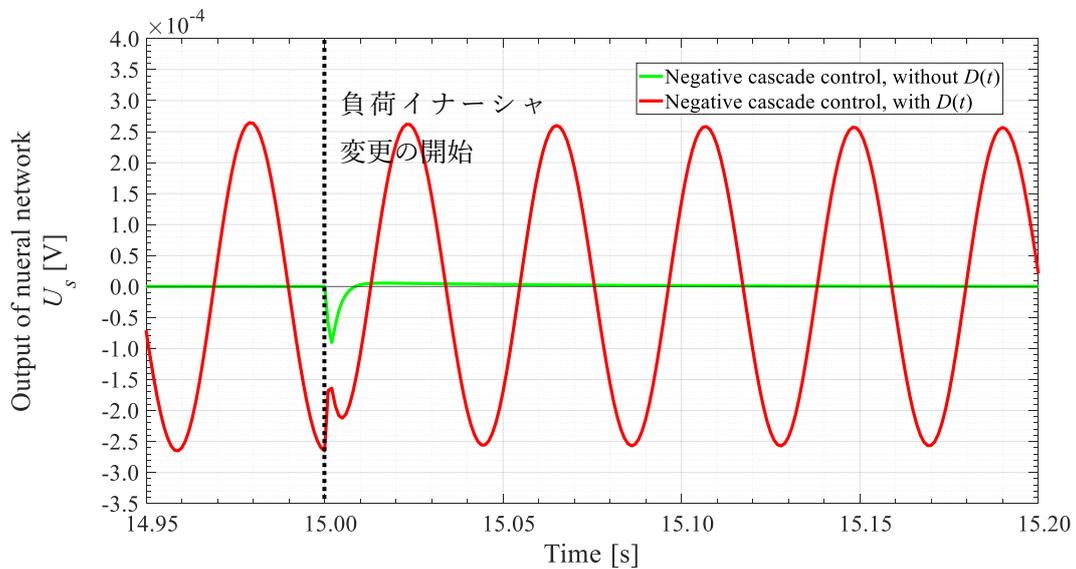


図 7.17 周期外乱 24 Hz の印加有無における上段のニューラルネットワーク補償器の出力 (図 7.16 の拡大図)

7.4.4 応答性とロバスト性に関する考察

図 7.9 から図 7.12 および表 7.1 と表 7.2 より、下段のニューラルネットワーク補償器の出力 u_n とポジティブカスケードニューラルネットワーク制御における上段の補償器の出力 u_s を比較すると、モータの慣性イナーシャの 10 倍から 50 倍までの負荷では、補償器全体の出力の向上率が 0.38%、1.12%、0.47%、0.38%、0.62% であることが確認された。そして、下段の補償器のみの

フィードバック誤差 e を比較すると、モータの慣性イナーシャの 10 倍から 50 倍までの負荷では、補償器全体の出力の向上率が -5.02% 、 -9.52% 、 -1.81% 、 -1.04% 、 -1.64% であることが確認された。すなわち、ポジティブカスケードニューラルネットワーク制御では、ニューラルネットワーク補償器全体の出力のみを改善させることが分かった。

また、下段のニューラルネットワーク補償器の出力 u_n とネガティブカスケードニューラルネットワーク制御における補償器の出力 u_s を比較すると、モータの慣性イナーシャの 10 倍から 50 倍までの負荷では、補償器全体の出力の向上率が 0.52% 、 0.27% 、 0.64% 、 0.32% 、 0.27% であることが確認された。さらに、下段の補償器のみのフィードバック誤差 e を比較すると、モータの慣性イナーシャの 10 倍から 50 倍までの負荷では、補償器全体の出力の向上率が 2.58% 、 0.91% 、 1.16% 、 0.56% 、 0.47% であることが確認された。すなわち、ネガティブカスケードニューラルネットワーク制御では、ニューラルネットワーク補償器全体の出力とフィードバック誤差の両方を改善させることが分かった。

さらに、ネガティブカスケードニューラルネットワーク制御では、電機子巻線抵抗 R の温度が 30°C に到達した状態で負荷イナーシャ値 J_L を変更させると、基準となる巻線温度 25°C の場合よりも上段の補償器の出力がおよそ 3 倍も増加していることが確認された。また、 24 Hz の周期外乱を制御対象 u に印加した状態で負荷イナーシャ値 J_L を変更させると、負荷イナーシャ変更後でも $\pm 0.012\text{ deg}$ の振動が発散せず維持されていることが分かった。

従って、単体のニューラルネットワーク制御の構成よりもニューラルネットワーク制御をもう 1 つ接続したカスケードニューラルネットワーク制御の構成にすることで、フィードバック制御器を更に補償できることが分かった。そして、カスケードニューラルネットワーク制御の構成について、ネガティブなカスケードニューラルネットワーク制御の構成にすることで、ニューラルネットワーク補償器の出力、すなわち、ニューラルネットワークの学習の効果が向上されることを確認した。

7.5 あとがき

本章では、提案したオンライン・オフライン融合型学習法に基づいて、負荷変動や外乱による制御対象の影響をさらに抑制して、位置決め精度を高めるために、制御対象への制御入力とフィードバック誤差との逆モデルを新たなニューラルネットワーク補償器の学習で獲得するようなカスケードニューラルネットワーク制御を提案した。そして、サーボシステムの位置決め制御に対して、サーボモータの位置決め制御のシミュレーションを行った。このシミュレーションでは、負荷イナーシャ値の変更、電機子巻線の温度変化、 24 Hz の周期外乱の印加を行った。その結果として、ネガティブなカスケードニューラルネットワーク補償器を接続することでフィードバック制御器の更なる補償が行われ、位置決め精度を高めることができた。従って、シミュレーションの結果と考察から提案法の有効性を示した。

第8章 結論

本研究では、サーボシステムの位置決め制御に対して、負荷変動と外乱の対応および実時間制御の実現を達成するため、オンライン学習法のロバスト性が優れている利点と今までのオフライン学習法の応答性が良い利点を融合させたニューラルネットワーク制御のオンライン・オフライン融合型学習法を提案した。

具体的には、まず、十分なオンライン学習で獲得したニューラルネットワークの重みを基準重みとしてロードして、ニューラルネットワーク制御はオフライン学習法で実施される。次に、フィードバック誤差が閾値を超えなければ、オフライン学習法が行われる。そして、フィードバック誤差が閾値を超えれば、サンプリングタイム内で少ない繰り返しループのオンライン学習法が行われる。さらに、サーボモータのための負荷イナーシャ変更前後の位置決め制御をシミュレーションして、その結果から提案法の有効性を明らかにさせた。

また、オンライン・オフライン融合型学習法の学習率、回数閾値とサーボシステムの応答性、ロバスト性との関係を明らかにして、学習率、サンプリングタイム内の少ない繰り返しループの回数閾値、フィードバック誤差、整定時間との関係を最小二乗法によって定式化した。加えて、それらの関係から学習率と回数閾値の更新則を導出した。そして、シミュレーションの結果から提案した学習率と回数閾値の更新則は、セルフチューニングが行えるような学習率と回数閾値の更新則として相応しいと判断した。

さらに、負荷変動や外乱による制御対象の影響をさらに抑制し、位置決め精度を高めるため、制御対象への制御入力とフィードバック誤差との逆モデルを新たなニューラルネットワーク補償器の学習で獲得するようなカスケードニューラルネットワーク制御を提案した。

具体的には、新しく追加したニューラルネットワーク補償器を上段のニューラルネットワーク補償器として、下段のオンライン・オフライン融合型学習法を用いたニューラルネットワーク補償器と共にフィードバック制御器を補償する。追加したニューラルネットワーク補償器は、オンライン学習法のアルゴリズムを利用して、ニューラルネットワークを学習させた。また、負荷イナーシャ値の変更や電機子巻線の温度変化、周期外乱の印加を行い、位置決め制御のシミュレーションを行った。その結果、1サンプリングタイム前のニューラルネットワーク補償器の出力信号を用いたネガティブカスケードニューラルネットワーク制御の構成にすることで、サーボモータの位置決め精度を向上させることができた。従って、提案法であるカスケードニューラルネットワーク制御の有効性を示せた。

以上の3つの提案とシミュレーションの結果により、サーボシステムの位置決め制御に対して、オンライン・オフライン融合型学習法を用いた負荷変動と外乱の対応および実時間制御、セルフチューニング、位置決め精度を高めるようなサーボモータの位置決め制御を実現させた。

謝辞

本論文題目であるニューラルネットワークのオンライン・オフライン融合型学習法を用いたサーボシステム位置決め制御に関する研究を行うに当たり、忙しいにもかかわらず丁寧にご指導とご助言をして頂きました指導教員の 森下明平 教授と学部4年から博士課程修了まで多くのご指導とご助言をして頂きました 黄慶九 准教授には、深く感謝申し上げます。特に、新型コロナウイルスの感染防止により本学に入構することが厳しくなった時期があり、講義、学部生への研究指導、大学院生への研究指導、会議など大変であったと存じます。その中、本研究および本論文のご指摘頂き大変ありがとうございました。

また、審査会にご参加頂きました 福岡豊 教授、斎藤秀俊 教授、濱根洋人 教授、千葉大学の 劉康志 教授にもご教授頂き誠に感謝致します。併せて、電磁力応用システム研究室の 森村 暢夫 先輩および 淀川 翔太 先輩には、同期モータに関する知識や非接触給電に関する知識などご教授頂き感謝致します。加えて、制御システム研究室の学部生、修士生、卒業生の皆様とも有意義な時間を過ごすことができました。そして、コロナ禍でも学生生活を支え続けて頂いた家族には、感謝の意を表します。

さらに、DC サーボモータの位置決め制御の構成を 黄慶九 准教授と考案する際、参考にさせていただきましたリアルテック有限会社 鈴木啓之 様とライズ・シー株式会社技術部 荒永昭博 様にもお礼申し上げます。鈴木様は、ブラシレスモータの実験装置の製作依頼の受諾や実験装置に関する疑問点、不具合などのサポートで大変お世話になりました。荒永様は、鈴木様からの委託で実験装置の販売を行っており、購入の手続きから配送まで大変お世話になりました。さらに、研究成果を発表する場を提供して頂きました 一般社団法人電気学会 様やその他の学会様、ご助言をして頂きました聴講者の皆様にもお礼申し上げます。

最後に、本研究を進めるに当たってお世話になりました指導教員から学会の聴講者の皆様まで全ての方々に、改めましてお礼申し上げ謝辞と致します。

参考文献

- [1] 前川佐理, 田中亜実, 「多層ニューラルネットワークを用いた PMSM の低速域センサレス制御の高性能化」, 電気学会論文誌D (産業応用部門誌), 141 巻, 10 号, pp.749-762, 2021.
- [2] 張斌, 古川将司, 林憲玉, 「ニューラルネットワークを利用した 4 ロータフライングロボットの適応 PID 制御に関する研究」, 電気学会論文誌C(電子・情報・システム部門誌), 141 巻, 3 号, pp.464-470, 2021.
- [3] Lina Sheng, Wenxu Yan, Liang Geng and Dezhi Xu, Adaptive Composite Neural Network Control for Variable Speed Wind Turbines, Part B: Applications, *ICIC Express Letters*, vol.8, no.9, pp.1339-1346, 2017.
- [4] 齋藤陽平, 羅志偉, 渡部慶二, 村松鋭一, 藤井省三, 「未知の動的環境におけるロボットの 2 自由度適応力追従制御」, システム制御情報学会論文誌, 18 巻, 6 号, pp.203-212, 2005.
- [5] 大松繁, 「ニューロコントロールと適応制御」, システム制御情報学会誌, 36 巻, 12 号, pp.769-775, 1992.
- [6] 大松繁, 「ニューラルネットワークと適応制御」, 北森俊行, 新誠一 (編), 「適応制御」, コンピュートロール, 32 号, コロナ社, pp.73-79, 1990.
- [7] Mengxiang Zhuang and Qixin Zhu, The Internal Mode Fractional-Order PID Control Based on Neural Network for the Temperature of Air-Conditioned Rooms, *International Journal of Innovative Computing, Information and Control*, vol.17, no.3, pp.1019-1028, 2021.
- [8] Bingran Li, Hui Zhang, Peiqing Ye, Jinsong Wang, Trajectory smoothing method using reinforcement learning for computer numerical control machine tools, *Robotics and Computer-Integrated Manufacturing*, vol.61, no.101847, 2020.
- [9] Qixin Zhu, Ye Lu, Jinsong Zhou, Xuemei Niu, Ou Xie, Yehu Shen, Fuzhou Niu and Yonghong Zhu, Two-Degree-of-Freedom Compound Control Based on RBF Neural Network for Air Conditioning Temperature Control System, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.445-456, 2020.

- [10] Oleksandr Veligorskyi, Roustiam Chakirov, Maksym Khomenko, and Yuriy Vagapov, Artificial Neural Network Motor Control for Full-Electric Injection Moulding Machine, *IEEE International Conference on Industrial Technology*, pp.60-65, 2019.
- [11] 進士裕介, ルーベンピンゾン, 平田豊, 「汎用小脳ニューラルネットワークモデルによる実機ロボット適応制御」, 人工知能学会第二種研究会資料, 2016 巻, AGI-002 号, pp.1-4, 2016.
- [12] José Luis Calvo-Rolle, Oscar Fontenla-Romero, Beatriz Pérez-Sánchez, Bertha Guijarro-Berdiñas, Adaptive Inverse Control Using an Online Learning Algorithm for Neural Networks, *Informatica*, vol. 25, no.3, pp.401-414, 2014.
- [13] Hashmia, A.L., Dakheel, S.H., Speed Control of Separately Exited DC Motor Using Artificial Neural Network, *Journal of Engineering and Development*, vol.16, no.4, pp.349-362, 2012.
- [14] Qingjiu Huang, Kenzo Nonami, Humanitarian Mine Detecting Six-Legged Walking Robot and Hybrid Neuro Walking Control with Position/Force Control, *Mechatronics*, vol.13, no.8-9, pp.773-790, 2003.
- [15] 陳亮, 堤正臣, 葛東方, アントニオモラン, 「自律制御システムによる CNC 工作機械の送り駆動精度の安定化手法 (ニューラルネットワークを使用した手法)」, 日本機械学会論文集 (C 編) , 61 巻, 591 号, pp.4504-4509, 1995.
- [16] 陳亮, 堤正臣, 「ニューラルネットワークによる CNC 工作機械の位置決め誤差補正に関する研究 (多次元位置決め誤差の補正)」, 日本機械学会論文集 (C 編) , 61 巻, 583 号, pp.1224-1229, 1995.
- [17] 漆原史朗, 鎌野琢也, 鈴木茂行, 原田寛信, 「ニューラルネットワークを用いたリア DC サーボモータの位置決め特性」, 電気学会論文誌 D (産業応用部門誌) , 115 巻, 3 号, pp.204-210, 1995.
- [18] 渡辺嘉二郎, 芹沢一雅, 侯麗雅, 「学習係数の適応調整によるニューラルネットワーク学習の高速化」, 計測自動制御学会論文集, 30 巻, 9 号, pp.1093-1099, 1994.
- [19] 神田有洋, 藤田聡, 阿江忠, 「予測によるニューラルネット誤差逆伝搬アルゴリズムの高速化」, 電子情報通信学会論文誌 (D-II) , 76 巻, 1 号, pp.132-139, 1993.

- [20] J.P.Cater, Successfully using peak learning rates of 10 (and greater) in back-propagation networks with the heuristic learning algorithm, *Proceedings of the IEEE International Conference on Neural Networks*, pp.645-651, 1987.
- [21] Zhang, Y., Mao, K., Leigh, S. et al., A parametric study of 3D printed polymer gears, *International Journal of Advanced Manufacturing Technology*, vol.107, no.11-12, pp.4481-4492, 2020.
- [22] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, D. L. Alkon, Accelerating the Convergence of the Back-Propagation Method, *Biological Cybernetics*, vol.59, pp.257-263, 1988.
- [23] Hiroshi Fukuda, Masato Odagaki, Osamu Hiwaki, Identification of Stimulated Sites Using Artificial Neural Networks Based on Transcranial Magnetic Stimulation-Elicited Motor Evoked Potentials and Finger Forces, *IEEJ Transactions on Electronics, Information and Systems*, vol.130, no.12, pp.2204-2209, 2010.
- [24] 落合慶広, 戸田尚宏, 臼井支朗, 「重みの振動を抑制する階層型ニューラルネットワークの学習法－Kick Out 法－」, 電気学会論文誌C (電子・情報・システム部門誌), 113 巻, 12 号, pp1154-1162, 1993.
- [25] Keihiro Ochiai, Naohiro Toda, Shiro Usui, New Accelerated Learning Algorithm to Reduce the Oscillation of Weights in Multilayered Neural Networks, *IJCNN-92 (Baltimore)*, vol.I, pp.914-919, 1992.
- [26] Qian Shi, Hak-Keung Lam, Bo Xiao, Shun-Hung Tsai, Adaptive PID controller based on Q -learning algorithm, *CAAI Transactions on Intelligence Technology*, vol.3, no.4, pp.235-244, 2018.
- [27] R.A.Jacobs, Increased Rates of Convergence Through Learning Rate Adaptation, *Neural Networks*, vol.1, pp.295-307, 1988.
- [28] Masakazu Morita, Qingjiu Huang, Mimpei Morishita, Online-Offline Integrated Learning Method of the Neural Network control for a Servo System, *International Journal of Innovative Computing, Information and Control*, vol.17, no.4, pp.1447-1464, 2021.
- [29] Masakazu Morita, Qingjiu Huang, Mimpei Morishita, Compensation of the Servo System on Online-Offline Integrated Learning Method of the Neural Network Control, *Proceedings of Annual Conference on Engineering and Information Technology*, pp.74-83, Osaka, Japan, March 24-26, 2020.

- [30] 森田昌一, 黄慶九, 森下明平, ニューラルネットワーク制御のオンライン・オフライン融合型学習法におけるサーボシステムの補償, 計測自動制御学会システムインテグレーション部門講演会, pp.2484-2487, 香川県, 12月12-14日, 2019年.
- [31] 江口誠, 喬夢醒, 大森浩充, 山崎由大, 金子成彦, 「AI-フィードフォワードを有するフィードバック誤差学習によるディーゼルエンジン燃焼制御」, 自動車技術会論文集, 49巻, 2号, pp.230-234, 2018.
- [32] Hiroaki Gomi, Mitsuo Kawato, Neural network control for a closed-loop System using Feedback-error-learning, *Neural Networks*, vol.6, no.7, pp.933-946, 1993.
- [33] 五味裕章, 川人光男, 「フィードバック誤差学習による閉ループシステムの学習制御」, システム制御情報学会論文誌, 4巻, 1号, pp.37-47, 1991.
- [34] Mitsuo Kawato, Kazunori Furukawa, R Suzuki, A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movement, *Biological Cybernetics*, vol.57, no.3, pp.169-185, 1987.
- [35] Masakazu Morita, Qingjiu Huang, Mimpei Morishita, An Update Rule of Parameters of Online-Offline Integrated Learning Method of Neural Network Control, *International Journal of Innovative Computing, Information and Control*, vol.18, no.3 (tentative), pp.1-IJICIC-2110-009, 2022, (Accepted, In Press).
- [36] 赤津観, 「史上最強カラー図解 最新版 モータ技術のすべてがわかる本」, ナツメ社, 296p, 2012.
- [37] 武田洋次, 松井信行, 森本茂雄, 本田幸夫, 「埋込磁石同期モータの設計と制御」, オーム社, 170p, 2001.
- [38] 横山修一, 濱根洋人, 小野垣仁, 「基礎と実践 制御工学入門」, コロナ社, 222p, 2009.
- [39] 小坂学, 「mbed マイコンによるモータ制御設計法」, 科学情報出版株式会社, 198p, 2013.
- [40] 森本茂雄, 真田雅之, 「省エネモータの原理と設計法～永久磁石同期モータの基礎から設計・制御まで～」, 科学情報出版株式会社, 245p, 2013.
- [41] 佐藤和也, 下本陽一, 熊澤典良, 「はじめての現代制御」, 講談社, 232p, 2012.
- [42] 浜田望, 松本直樹, 高橋徹, 「現代制御理論入門」, コロナ社, 284p, 1997.

- [43] 石井健一郎, 上田修功, 前田英作, 村瀬洋, 「わかりやすいパターン認識」, オーム社, 206p, 1998.
- [44] 斎藤康毅, 「ゼロから作る Deep Learning – Python で学ぶディープラーニングの理論と実装」, 300p, 2016.
- [45] Sung-Yoon Jung, Jinseok Hong, Kwanghee Nam, Current Minimizing Torque Control of the IPMSM Using Ferrari's Method, *IEEE Transactions on Power Electronics*, vol.28, no.12, 2013.
- [46] G. Cardano, *The Rules of Algebra: (Ars Magna)*. New York, USA: Dover, 2007.
- [47] 木本雅也, 清水忠昭, 吉村宏紀, 井須尚紀, 菅田一博, 「砂時計型ニューラルネットワークの多段化による LSP パラメータ圧縮特性の改善」, 情報処理学会論文誌, 46 巻, 3 号, pp.845-848, 2005.
- [48] 吉村宏紀, 清水忠昭, 井須尚紀, 菅田一博, 「多段接続砂時計型ニューラルネットワーク雑音除去フィルタを用いた適応的雑音除去」, 電気学会論文誌 C (電子・情報・システム部門誌), 120 巻, 4 号, pp.507-515, 2000.

一般社団法人システム制御情報学会、オーム社、科学情報出版株式会社、講談社、コロナ社、ナツメ社より図の引用および改変の許可をいただきました。