修 士 論 文

非線形最適レギュレータ問題への強化学習の適用

北陸先端科学技術大学院大学
知識科学研究科知識社会システム学専攻

内藤 浩行

2000 年 3 月

修 士 論 文

非線形最適レギュレータ問題への強化学習の適用

指導教官　吉田 武稔　助教授

北陸先端科学技術大学院大学
知識科学研究科知識社会システム学専攻

850062 内藤 浩行

審査委員： 吉田 武稔 助教授 (主査)
中森 義輝 教授
本多 卓也 教授

2000 年 2 月

要 旨

別紙

目 次

図 目 次

表 目 次

第 1 章

序論

1.1 研究の背景

最適制御問題とは，システムを表す状態方程式と制約条件のもとで与えられた評価関数を最小化する制御問題である．この時，制約条件や評価関数は目的や要請に応じてどのように複雑な定式化も可能である．しかしながら，従来数多くの研究が行われている最適制御問題は線形のシステムの状態方程式であり，評価関数が単一の汎関数で表される簡潔な問題である [10].

対して，非線形のシステムは，非線形系特有の複雑さを含んでいるため解析的に解を導くことは難しいことが知られている．従って，線形システムの結果を用いることで非線形システムの解析や設計が行われている．すなわち，非線形システムの平衡点近傍にてシステムを線形問題として捉え直し，線形システム問題として解くという手法が良く用いられる [12]．しかしながら，この解法はあくまでも近似的な手法であるため，動的計画法や最小実現法を用いて直接的に非線形システムの解が求められている [6][12]．だが，非線形システムの解が求まったとしても解の性質上，実装が困難，もしくは，不可能であることが指摘されている [3].

このことから，非線形システムに対して，ファジィ，遺伝的アルゴリズムやニューラルネットワーク等のヒューリスティックな解法を用いてそのシステムの実装を行うという試みもある [6][11].

だが，これらの実装法とは異なる新たなヒューリスティックな実装法として強化学習が提案されている [1][24].

強化学習とは，動物心理学などに用いられて来た用語である．工学的には，未知なる環境から報酬を得て罰から逃れるような適当な行動を試行錯誤的に獲得する，という適応現象を模倣したシステムである．このシステムを機械学習のから捉え直すと，R.S.Sutton や C.J.C.H.Watkins によって提唱された学習問題のクラスとなる．また，この学習法は，状態の評価に遅れが存在し，かつ，状態遷移に離散的な不連続性や不確実性をふくむシステムに対して適応的な解が得られる，と考えられ着目されている [2].

この学習法を用いて，現在，行われている研究分野としては，倒立振子制御問題，ジョブショップ問題，スケジューリング問題などがあるが [4][14][17][22]，新たなる他分野への応用が期待される．その一つの分野として制御理論，動的システムの最適制御問題である最適レギュレータ問題がある [15][16][19].

1.2 従来研究

非線形システムへのヒューリスティックな解法の適用については，いくつかの研究成果がある．本研究では，強化学習以外のヒューリスティックな手法については，本研究の範囲を超えるのでそれぞれに応じた参考書を見て頂きたい．非線形システムに強化学習を応用した研究成果としては，内藤ら (1999)，堀内ら (1999)，Barto ら (1983)，Zhang ら (1995) や Crites(1996) らによるものが知られている．内藤ら，堀内らや Barto らは倒立振子制御問題に対して強化学習を応用している [4][5][14]．Zhang らはジョブショップスケジューリング問題に対して強化学習を応用し [22]，Crites らはディスパッチ問題に対して強化学習を用いている [17].

さらに，強化学習の応用が提案されている分野の一つとして，制御理論，特に最適制御問題への応用があり，Bradtke ら (1993，1994)，Frueh ら (1998) らの研究が知られている [15][16][19]．Bradtke らによって実装された問題は線形時不変離散システムであり，そのシステムを適応最適レギュレータ制御問題として解いている [15][16]．Frueh も，線形時不変離散システムを研究対象として定義し，それを最適レギュレータ問題として捉え直し，強化学習問題として実装を行っている [19].

また，強化学習は有限離散マルコフ決定過程に基づくため，離散的な問題に有効な解法である．そのため，より一般的な連続系の問題に対して強化学習を用いるための拡張が，堀内ら (1999) や Munos(1997) によって行われている [5][18]．堀内らは，ファジィ推論を用いることで連続系システムを強化学習問題として定式化を行い，その実装を行っている

[5]．Munos は，連続系の最適制御問題を解くために収束強化学習アルゴリズムを提案している [18].

以上のような研究成果などがある．いずれも非線形システムをヒューリスティックに解く，もしくは，その実装を行うための研究が行われている．だが，システムの解析に強化学習が使われることは少ない．

一方，非線形システムの解析を行った研究成果として田中による研究結果がある [3]．田中は，Yoshida らの結果である非線形系の最適レギュレータ問題を数値シミュレーションにて実装を行い，そのシステムの解析を行った [20][21]．その結果，田中は Yoshida らの結果を実装する際に打ち切り次数を考慮しなければならないことを提案している．

1.3 研究の目的

ここまでの議論から，非線形システムの解析とその実装を行うために強化学習を用いることは，十分な意義があると考えられる．

そこで，本研究では，非線形システムの最適レギュレータ問題の理論解を実装する際に強化学習を用いることで，困難であった数値シミュレーションを行うことを目的とし，さらに，本手法の有用性も検証する．

すなわち，研究対象としてべき級数で制御則が与えられる非線形システムを考える [20][21]．その際，有限級数として次数を打ち切らなければ制御則を実装することはできない．その上，次数における近似の度合は対象となる非線形システムによって異なる．そこで，経験と直観に頼っていた制御則の打ち切り次数の導出を強化学習問題として捉え，レギュレータの設計を図る．同時に，数値シミュレーションにより打ち切り次数に対して考察を与える [26].

1.4 本稿の構成

本論文の構成を述べる．

第 1 章では，研究の背景を述べ，本研究を行うに至った従来研究と本研究の目的，さらに概要についてまとめている．

第 2 章では，強化学習についてまとめている．R.S.Sutton と A.G.Brato による著作 “Reinforcement Learning” と畝見による解説記事を簡潔にまとめたものである．

第 3 章では，線形最適レギュレータ問題の概要と本研究にて実装を行った非線形システムの最適レギュレータ問題について述べる．また，非線形系の最適レギュレータ問題の数値例を示すことで，本研究で扱う制御問題の検証を行い，問題点を指摘する．

第 4 章では，本研究の問題設定を行い，定式化を行う．さらに，本研究の目的である強化学習を用いて数値シミュレーションを行い，そして結果を提示している．

そして，最後に第 5 章では，本研究の結論を述べる．

第 2 章

強化学習

強化学習は，学習問題の一つの手法である．すなわち，学習者は環境の中でなんらかの行為を起こすエージェントであり，エージェントは各時間ステップにおいて得られる状態から行為を決定する．そして，エージェントは実際にとった行為に対して環境から報酬あるいは罰が与えられる．報酬の大きさは，多くの場合，過去数ステップの行動に対して求められる．結果として，エージェントの目的はある時間長さにわたる報酬の重み和を最大化することになる．

言い替えれば，強化学習とは，問題を解くためにその問題に対して当てはめられる解法とも言える．

以下，次節では，強化学習について紹介する．まずは，強化学習の構成要素をまとめる．そして，構成要素を踏まえて強化学習の枠組について概要を述べていく．さらに，実現方法について紹介する．より詳しくは，畝傍による解説記事や R.S.Sutton と A.G.Barto による著作 “Reinforcement Learning” にまとめられている [1][24].

2.1 構成要素

ここでは，強化学習の構成要素について記している．

- 環境　強化学習問題の問題領域を定義するものである．
- エージェント　意志決定や学習を行う．

- 政策　エージェントが知覚する特定の環境の状態に対して，エージェントが取れる行為を選択するためのものである．ここで “状態” とは，学習システムにとっての外部からの入力であり，環境からの感覚入力や学習者の内部状態，あるいは，それらの組合せでもある．また “行為” とは，エージェントがある状態において行われる行動のことである．

- 報酬　エージェントが，政策に基づき行為を意志決定し実行した結果，環境からエージェントが受け取るものである．

- 価値関数　エージェントが獲得した報酬を次の機会の意志決定に行かすためにある方法 (アルゴリズム) で学習し，学習内容を保持するためのものである．価値関数の価値とは，その状態ないしその状態-行為の組みから最終状態までの累積報酬を指す．

以上が基本となる構成要素であるが，上記で述べたいくつかの要素に対する付加的な概念を以下に付記する．

- エージェントの行為選択には大きく分けて 2 通りある．一方は，様々な状態と行為を試す “情報獲得行為”．もう一方は，既に学習した結果から最大価値を実現する行為を選択する “報酬獲得行為” がある．

- 価値関数には，状態に依存する “状態価値関数” と，ある特定の状態と行為によって定義される “行為価値関数” がある．どちらの関数も特定の状態ないし状態-行為の組みの真の価値 (期待値) を，限られた試行から学習された結果を保持するものである．

- 価値関数は，状態，もしくは，状態-行為の組みによる表 (ルックアップテーブル) として表されるものと，線形ベクトル表現やニューラルネットなどにより関数近似によって表現される方法がある．大規模問題に対応するためには関数近似による表現が必要となる．

以上の構成要素の概念を用いて強化学習問題を捉え直すと，強化学習問題を解くということは環境からの報酬を基に状態価値もしくは行為価値を推定して累積報酬を最大化する政策をみつけることとなる．

このような強化学習問題において，より一般的な連続時間係を扱うことも考えられるが，実際にはほとんど場合において離散時間系を前提として研究が進められている．なぜならば，強化学習アルゴリズムがマルコフ性を前提としているためである．

2.2 枠組

Fig.2.1に前節にて述べた強化学習の構成要素の繋がりを示す．Fig.2.1の流れについて簡

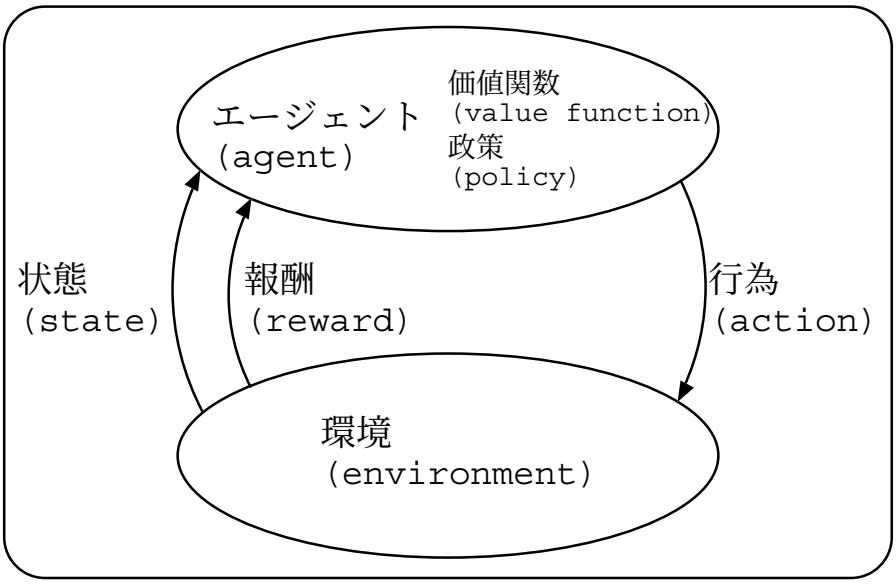

Fig. 2.1: The agent-environment interaction in reinforcement learning.

単に説明する．エージェントは環境に対して各時間ステップにおいて得られる状態から価値関数を更新し，政策を導き，行為を決定する．そして，エージェントは実際にとった行為に対して環境から報酬あるいは罰，そして状態が与えられ，新たに価値関数を更新していく，という流れとなる．報酬の大きさは，多くの場合，過去数ステップの行動に対して求められる．結果として，エージェントの目的はある時間長さにわたる報酬の重み和を最大化することになる．

エージェントは，学習によって得られた状態の評価の見積りをもとに行動を決定する．だが，その時点の評価の見積りを最大にするような行動選択が，必ずしも最適な決定となるとは限らない．なぜなら，強化学習ではエージェントの経験はエージェント自身の行動に強く依存するからである．学習一般に関して，経験の内容によって学習結果が大きく異

なるのは当然だが，強化学習ではエージェントの行動選択が経験の内容を左右する．よって，最適な政策を見付けるためには環境に対して十分な探索を行う必要がある．

状態の複雑さと行為の複雑さは強化学習問題を更に分類する際の重要な尺度となる．詳しくは，状態の複雑さ，状態変化の不確実さと文脈依存性，環境の変動，行為選択肢の複雑さなどが問題とされる．

最も単純な強化学習問題の形は，状態と行動選択肢がともに離散的かつ数え上げ可能で，すべての状態と行動選択肢を表現するに十分な記憶がエージェント側に用意でき，ひとつの状態に関して最適な行為は一つであり，環境も安定であるという場合である．実際に動物のモデルとして，あるいはなんらかの問題への応用を考えると，このような理想的な状況は想定しにくい．だが，強化学習の特徴を捉えるための足掛かりとしては十分である．工学的には，信号の区間分類による記号化や問題設定の単純化によって有効に使える形となっている場合もある．

また，報酬の見積りも重要である．あらかじめ，どのような分布で存在するかがわかっていれば，それに応じた学習方法をとることが可能な場合もある．例えば，報酬が 0 または -1 であれば，-1 が与えられるまで時間をできるだけ引き延ばすような戦略が有効となる．また，価値関数の最大値が明らかになっておれば，学習中にその値に近い性能が得られるようになった時点で，探索を中止するという戦略を取ることができる．

2.3 実現方法

強化学習は問題を解くための枠組を与える．そのため，その実現方法は様々な種類がある．例えば，Temporal Difference (TD 法)，Q-learning がある．

ここでは，本論文にて採用した Q-learning について簡単に紹介する．

2.3.1 Q-learning

強化学習の手法の 1 つに C.J.C.H.Watkins によって考案された Q-learning がある．Table 2.1にそのアルゴリズムを記す．Q-learning は状態 s_t だけでなく，状態 s_t と行為 a_t の組みに対して $Q(s_t, a_t)$ 値と呼ばれる評価値の見積りを行いつつ学習を進める．すなわち，時刻 t の時点に状態 s_t において行為 a_t を実行した結果，状態は s_{t+1} に遷移し，報酬 r_t が得られる．その時，$Q(s_t, a_t)$ 値は以下の (2.1) 式より逐次的に導出される．そし

て，$Q(s_t, a_t)$ 値をもとづいて行為 a_t の決定を行う．よって，$Q(s_t, a_t)$ 値を導く行為価値関数を Q 関数という．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})) \tag{2.1}$$

(2.1) 式について Fig.2.2を用いて説明する．Fig.2.2の縦軸は学習回数，横軸は時間軸である．支点は状態 s_t を表しており，矢印は $Q(s_t, a_t)$ 値である．また，学習進む程矢印が太くなり，その価値が高まっていることを表している．よって，学習を始めた時点 (一番下の平面) では，一つの支点において多数の行為 a_t があるが，学習を進めるにしたがって個々の支点において $Q(s_t, a_t)$ 値に応じた行為を選択するようになり，より良い政策をみつけることができる．ここで，α は学習率，γ は割引率である [24]．学習率は過去の価値を現在の価値にどの程度反映させるかを決定するパラメータであり，割引率は未来の価値を現在の価値にどの程度反映させるかを決定するパラメータである．

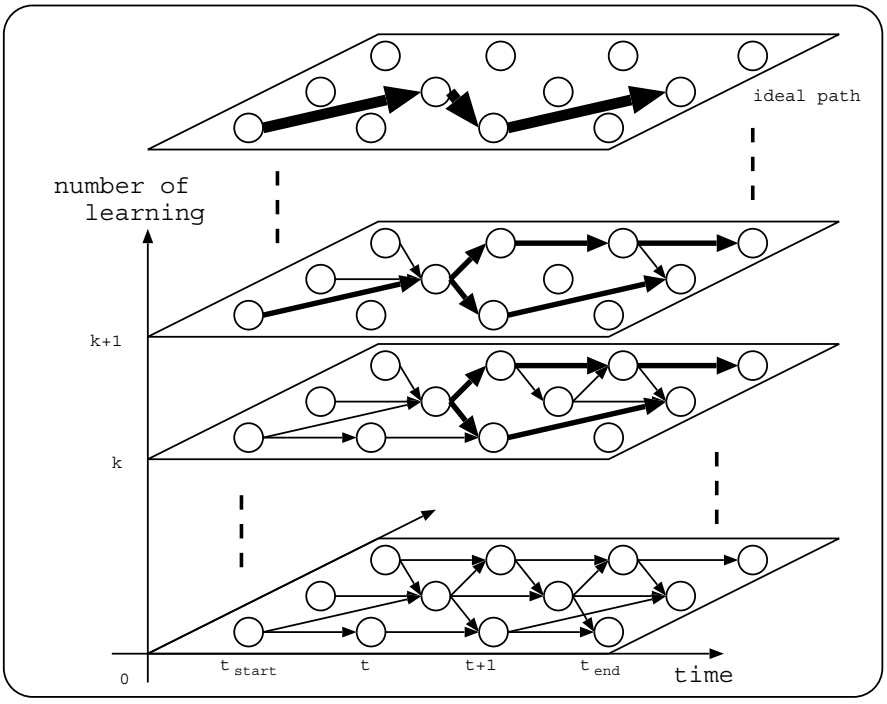

Fig. 2.2: The process of learning in reinforcement learning.

また，行為決定の方法は何通りかあり，代表的なものとして，以下の (2.2) 式に示すような Boltzmann 分布等に基づく確率的な行為選択法や ϵ-Greedy 法 (どん欲法) 等がある．

$$P(a_t|s_t) = \frac{e^{(Q(s_t,\, a_t)/T)}}{\sum_{b \in actions} e^{(Q(s_t,\, b_t)/T)}} \tag{2.2}$$

ただし，T は温度定数と呼ばれ，この値が大きいほど行動はよりランダムになり積極的な探索を行う．ϵ-Greedy 法は，ϵ % の確率で情報獲得行為を行い，$(1-\epsilon)$ % の確率で報酬獲得行為を行う行為選択法である．本稿では，ϵ-Greedy 法を用いている．

Q 関数の実現方法はさまざまなものがある．最も，単純な方法はすべての状態と行動の組合せに付いての表，ルックアップテーブルを作り Q 値を保持し，(2.1) 式にしたがって，Q 値を更新する方法である．ただし，ルックアップテーブルの大きさが利用可能な計算機の記憶容量を超えるような問題には使えない．よって，Q 関数をニューラルネットワークなどによって近似することで表現することもある．

Initialize $Q(s_t, a_t)$ arbitrarily (s_t:state, a_t:action). Repeat(for each episode): Initialize s_t. Repeat(for each episode): • Choose a_t from s_t using policy derived from $Q(s_t, a_t)$ (e.g.,ϵ-greedy). • Take action a_t, observer state s_t, observer reward r_t. • Update the equation (2.1). • $s_t \leftarrow s_{t+1}$. until s is terminal.

Table 2.1: Algorithm of Q-learning.

第 3 章

最適レギュレータ問題

本章では，最適レギュレータ問題についてまとめている．最初に最適レギュレータ問題の概要を述べ，次に，数式を使用して線形システムの最適レギュレータ 問題を定義し，その解法を提示する [12]．そして，本研究で用いる非線形システムにおける最適レギュレータ問題の概要を紹介する [3][20][21]．最後にその数値例を示す．

3.1 概要

システムがなんらかの入力により望ましい制御が行われ安定であったとする．しかしながら，そのシステムが突発的な外乱によって平衡点から外れてしまったとする．この時なんらかの新たな入力によって元の安定な状態に戻す制御問題を “最適レギュレータ問題” と呼ぶ．

ただ，あまりに短時間で平衡な状態に戻すことは，新たな入力，つまり，操作入力が現実的ではない．だからといって操作入力を小さくしすぎるのは効率がよくない．そこで，妥協案として操作入力のエネルギーとシステムのエネルギーの総和を，時間区分を $0-\infty$ で積分した 2 次形式評価関数にて表し，この評価関数を最小にする操作入力を用いて，システムを安定な状態に戻すことでシステムを効率よく制御することができることを保証する．より詳しくは，参考文献にまとめられている [7][9][10][12]．

3.2 最適レギュレータ問題の定式化

3.2.1 線形システム

この小節は，線形システムにおける最適レギュレータ問題の定式化する．詳細は参考文献などを参照せよ [10][12].

問題 3.1

構成すべきシステムを (3.1) 式のような m 入力の線形システムとする．

$$\dot{\boldsymbol{x}} := \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \neq 0 \tag{3.1}$$

ただし，$\boldsymbol{x} := [x_1 x_2 \cdots x_n]^T \in \mathcal{R}^{n\times 1}$ は状態変数，係数行列 $\boldsymbol{A}$，$\boldsymbol{B}$ はそれぞれ，$\boldsymbol{A} \in \mathcal{R}^{n\times n}$，$\boldsymbol{B} \in \mathcal{R}^{n\times m}$ であり，入力 $\boldsymbol{u}$ は $\boldsymbol{u} \in \mathcal{R}^{m\times 1}$ とする．

次に，式 (3.1) の評価関数 $\boldsymbol{J}$ を，

$$\boldsymbol{J} := \int_0^{+\infty} \{\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}\}dt \tag{3.2}$$

とする．ただし，$\boldsymbol{Q} \in \mathcal{R}^{n\times n}$ を準正定値対称行列，$\boldsymbol{R} \in \mathcal{R}^{m\times m}$ を正定値対称行列とする．

この時，評価関数 $\boldsymbol{J}$ を最小にする入力 $\boldsymbol{u}$，$t \in [0, +\infty]$ を求めよ． ■

この問題は，2 次形式評価関数 $\boldsymbol{J}$ の重み，$\boldsymbol{R} > 0$，$\boldsymbol{Q} \geq 0$ を定めると，一般的な線形最適レギュレータ問題の解が求まる．以下ではその解について述べる．

定理 3.1 ([10][12])

(3.1) 式において $\{\boldsymbol{A},\boldsymbol{B}\}$ は可制御とする．このとき，(3.2) 式を最小にする入力 $\boldsymbol{u}$ は，状態フィードバック

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x} \tag{3.3}$$

で与えられる．ただし，$\boldsymbol{S}$ はリッカチ行列方程式

$$\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A} - \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} + \boldsymbol{Q} = 0 \tag{3.4}$$

を満足する正定値対称行列である．この時，評価関数 $\boldsymbol{J}$ の最小値は

$$\boldsymbol{J}_{\min} = \boldsymbol{x}_0^T\boldsymbol{S}\boldsymbol{x}_0 \tag{3.5}$$

となる． □

3.2.2 多項式・非線形システム

本小節では，本研究テーマて扱う，べキ級数で記述された非線形システムに関する最適レギュレータ問題の定式化を行う．詳細は参考資料を参照せよ [3][20][21].

問題 3.2

構成すべきシステムを (3.6) 式のような m 入力のベキ級数システムとする．

$$\dot{\boldsymbol{x}} := \boldsymbol{A}\boldsymbol{x} + \sum_{k=2}^{\infty} \boldsymbol{F}_k \boldsymbol{x}^{[k]} + \boldsymbol{B}\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \neq 0 \tag{3.6}$$

ただし，$\boldsymbol{x} := [x_1 x_2 \cdots x_n]^T \in \mathcal{R}^{n\times 1}$ の状態変数,

$$N(n; p) := \left(\begin{array}{c} n+p-1 \\ p \end{array} \right) = \frac{(n+p-1)!}{p!(n-1)!}, \quad p > 0 \tag{3.7}$$

としたとき，次数の高い順に並べたベクトル $\boldsymbol{x}^{[p]} \in \boldsymbol{R}^{N(n;\, p)\times 1}$ は第 i 要素が次のように定義される．

$$\begin{aligned} \boldsymbol{x}^{[p]}\text{の第 } i \text{ 要素} &:= C_{np}(p_1^i, p_2^i, \cdots, p_n^i) \prod_{j=1}^{n} x_j^{p_j^i}, \\ C_{np}(p_1^i, p_2^i, \cdots, p_n^i) &:= \sqrt{\frac{p!}{p_1^i! p_2^i! \cdots p_n^i!}}, \\ p &:= \sum_{j=1}^{n} p_j^i, \\ p, p_j &\in \mathcal{R}, \\ j = 1, 2, \ldots, n, &\quad i = 1, 2, \cdots, N(n; p) \end{aligned} \tag{3.8}$$

また，係数行列 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{F}_k$ はそれぞれ，$\boldsymbol{A} \in \mathcal{R}^{n\times n}$，$\boldsymbol{B} \in \mathcal{R}^{n\times m}$，$\boldsymbol{F}_k \in \mathcal{R}^{n\times N(n;\, p)}$ であり，入力 $\boldsymbol{u}$ は $\boldsymbol{u} \in \mathcal{R}^{m\times 1}$ とする．

次に，(3.6) 式の評価関数 $\boldsymbol{J}$ を，

$$\boldsymbol{J} := \int_0^{+\infty} \{\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R}\boldsymbol{u}\} dt \tag{3.9}$$

とする．ただし，$\boldsymbol{Q} \in \mathcal{R}^{n\times n}$ を準正定値対称行列，$\boldsymbol{R} \in \mathcal{R}^{m\times m}$ を正定値対称行列とする．

この時，評価関数 $\boldsymbol{J}$((3.9) 式) を最小にする入力 $\boldsymbol{u}$，$t \in [0, +\infty]$ を求めよ． ■

以上の定式化を用いて，2 次形式評価関数 $\boldsymbol{J}$ の重み $\boldsymbol{R} > 0$，$\boldsymbol{Q} \geq 0$ を定めると，一般的な線形最適レギュレータ問題の解と同等の方法で問題 3.2 の解が求まることが Yoshida らによって提案されている [20][21]．以下ではその解についての概要を付記する．

定理 3.2 (Yoshida ら [20][21])

(3.6) 式において $\{\boldsymbol{A},\boldsymbol{B}\}$ は可制御とする．このとき，(3.9) 式を最小にする入力 $\boldsymbol{u}$ は，状態フィードバック

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T \sum_{k=1}^{\infty} \boldsymbol{S}_k \boldsymbol{x}^{[k]} \tag{3.10}$$

を得る．その際，$\boldsymbol{S}_k \in \boldsymbol{R}^{n\times N(n;k)}$ は代数方程式

$$\left(\boldsymbol{A}^T\boldsymbol{S}_1 + \boldsymbol{S}_1\boldsymbol{A} - \boldsymbol{S}_1\boldsymbol{K}\boldsymbol{S}_1 + \boldsymbol{Q}\right)\boldsymbol{x} = 0, \tag{3.11}$$

$$\left(\bar{\boldsymbol{A}}^T\boldsymbol{S}_k + \boldsymbol{S}_k\bar{\boldsymbol{A}}_{[k]} + \boldsymbol{X}_k\right)\boldsymbol{x}^{[k]} = 0, \quad k = 2, 3, \ldots \tag{3.12}$$

で与えられる．ただし，$\boldsymbol{X}_k$ は，

$$\boldsymbol{X}_2\boldsymbol{x}^{[2]} := (\boldsymbol{S}_1\boldsymbol{F}_2 + \bar{\boldsymbol{S}}_{2,\,1})\boldsymbol{x}^{[2]}, \tag{3.13}$$

$$\boldsymbol{X}_k\boldsymbol{x}^{[k]} := \left[\boldsymbol{S}_1\boldsymbol{F}_k + \sum_{j=2}^{k-1}\hat{\boldsymbol{S}}_{[j,\,k-j+1]} + \sum_{j=2}^{k}\bar{\boldsymbol{S}}_{j,\,k-j+1}\right]\boldsymbol{x}^{[k]}, \quad k = 3, 4, \ldots, \tag{3.14}$$

$$\hat{\boldsymbol{S}}_{[j,\,k-j+1]}\boldsymbol{x}^{[k]} := \boldsymbol{S}_j(\boldsymbol{F}_{k-j+1} - \boldsymbol{K}\boldsymbol{S}_{k-j+1})_{[j,\,k-j+1]}\boldsymbol{x}^{[k]},$$

$$\bar{\boldsymbol{S}}_{j,\,k-j+1}\boldsymbol{x}^{[k]} := \frac{\partial \boldsymbol{x}^{[j]^T}}{\partial \boldsymbol{x}}\boldsymbol{F}_j^T\boldsymbol{S}_{k-j+1}\boldsymbol{x}^{[k-j+1]},$$

$$\boldsymbol{S}_j(\boldsymbol{F}_{k-j+1} - \boldsymbol{K}\boldsymbol{S}_{k-j+1})_{[j,\,k-j+1]} := \frac{\partial \boldsymbol{x}^{[k]}}{\partial \boldsymbol{x}^T}\boldsymbol{S}_j\boldsymbol{x}^{[j]}(\boldsymbol{F}_{k-j+1} - \boldsymbol{K}\boldsymbol{S}_{k-j+1})\boldsymbol{x}^{[k-j+1]},$$

$$\bar{\boldsymbol{A}} := \boldsymbol{A} - \boldsymbol{K}\boldsymbol{S}_1, \tag{3.15}$$

$$\boldsymbol{K} := \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T, \tag{3.16}$$

$$\bar{\boldsymbol{A}}_{[k]} := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}C_{np}\,(\boldsymbol{E}_{ij})_{[k]}\,C_{np}^{-1}, \quad k = 2, 3, \ldots, \tag{3.17}$$

$(\boldsymbol{E}_{ij})_{[p]}$ は $N(p_1, \ldots, p_i - 1, \ldots, p_j + 1, \ldots, p_n) \times N(p_1, p_2, \ldots, p_n)$

要素が p_i となる $N(n; p) \times N(n; p)$ の行列である． (3.18)

で導く．ただし，$a_{ij} \in \mathcal{R}$ は行列 A の i 行 j 列の成分である．この時，評価関数 $\boldsymbol{J}$ の最小値は

$$\boldsymbol{J}_{\min} = \sum_{k=1}^{\infty}\frac{1}{k+1}\boldsymbol{x}_0^T\boldsymbol{S}_k\boldsymbol{x}_0^{[k]} \tag{3.19}$$

となる． □

注意 3.1

(3.10) 式を実装するには，有限級数として次数を打ち切る必要がある． □

3.2.3 非線形最適レギュレータ問題の数値例

ここでは，上記した非線形最適レギュレータ問題の数値例を述べる．そして，本研究で用いる非線形システムについて検証し考察を行う．

数値例 3.1

今回数値例の対象となったシステムを以下のように定義し，

$$\dot{\boldsymbol{x}} := \boldsymbol{A}\boldsymbol{x} + \boldsymbol{F}_3\boldsymbol{x}^{[3]} + \boldsymbol{B}\boldsymbol{u}, \ \ \boldsymbol{x}(0) = \boldsymbol{x}_0 \neq 0 \tag{3.20}$$

評価関数を，

$$\boldsymbol{J} := \int_0^\infty \left[\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} \right] dt \tag{3.21}$$

とする．その時，$\boldsymbol{J}$ ((3.21) 式) を最小にする入力 $\boldsymbol{u}$((3.10) 式) を用いて，システム ((3.20) 式) の解析を数値シミュレーションにより行う．ただし，(3.20)-(3.21) 式の係数行列を

$$\boldsymbol{A} := \begin{bmatrix} 3.5 & -8.2 \\ 9.5 & -4.5 \end{bmatrix}, \tag{3.22}$$

$$\boldsymbol{B} := \begin{bmatrix} -2 & 2 \\ -4 & -3 \end{bmatrix},$$

$$\boldsymbol{F}_3 := \begin{bmatrix} -0.3 & 1.8 & 0.3 & 0.5 \\ -1.8 & -2.7 & -0.5 & -0.7 \end{bmatrix},$$

$$\boldsymbol{Q} := \begin{bmatrix} 10 & 1 \\ 1 & 5 \end{bmatrix},$$

$$\boldsymbol{R} := \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \tag{3.23}$$

$$\boldsymbol{x} := [x_1, x_2]^T \in \mathcal{R}^2,$$

$$\boldsymbol{x}^{[3]} := [x_1^3, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, x_2^3]^T \in \mathcal{R}^4,$$

$$\boldsymbol{u} := [u_1, u_2] \in \mathcal{R}^2$$

とする．また，(3.10) 式の入力 $\boldsymbol{u}$ は有限次数で打ち切られる．

まずは，$\boldsymbol{u} = 0$ となるシステム (3.20) 式の状態変数 $\boldsymbol{x}$ のトラジェクトリ (軌道) を Fig.3.1 に示す．行列 $\boldsymbol{A}$ の固有値は $-0.5000 \pm j7.8677$ であり $\boldsymbol{A}$ は安定行列である．自由応答の

トラジェクトリの安定な平衡点は $(0,0), (\pm 2.6934, \mp 0.2530)$ となる．よって，トラジェクトリが初期値 $\boldsymbol{x}_0$ に依存することが，Fig.3.1より確認できる．すなわち，ある程度大きな初期値が与えられとき，原点に収束せず，安定な平衡点である $(\pm 2.6934, \mp 0.2530)$ に収束する時があることがわかる．

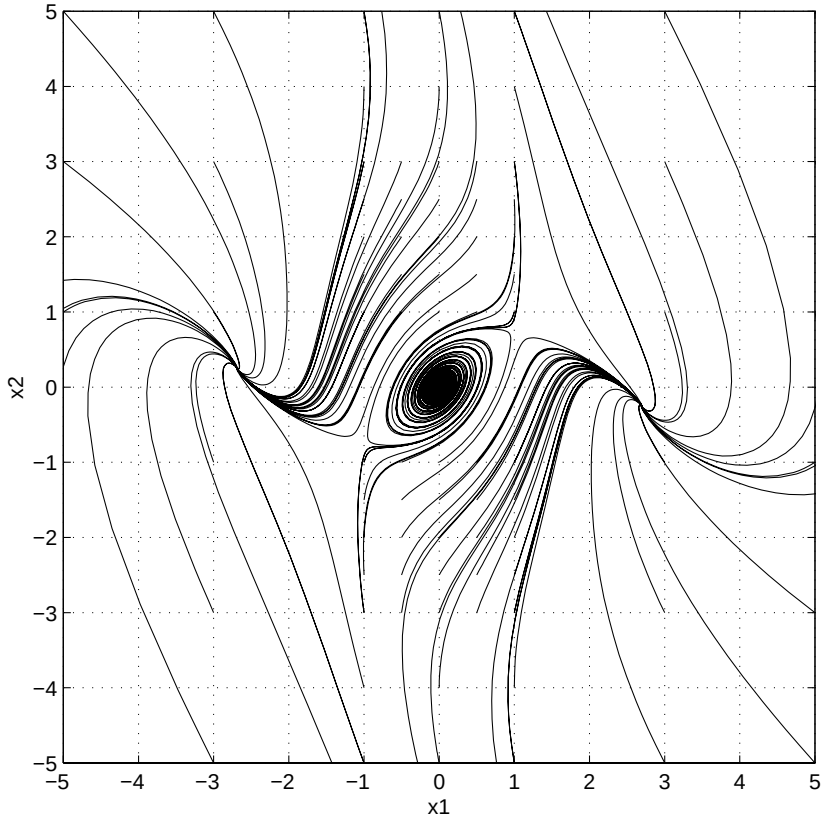

Fig. 3.1: Phase trajectories of the system when $u = 0$.

では，1 次 (線形) の次数で打ち切られたレギュレータ

$$\boldsymbol{u}_1 := -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}_1\boldsymbol{x} \tag{3.24}$$

を定義し，$\boldsymbol{u} = \boldsymbol{u}_1$ の際のシステム (3.20) 式の状態変数 $\boldsymbol{x}$ のトラジェクトリを数値シミュレーションにより検証する．ただし，(3.11) 式と (3.12) 式から得られるリカッチ方程式の解 $\boldsymbol{S}_1$ は，(3.25) 式となる．

$$\boldsymbol{S}_1 = \begin{bmatrix} 11.66 & -0.275 \\ -0.275 & 4.334 \end{bmatrix} \times 10^{-2} \tag{3.25}$$

その際のトラジェクトリを以下の Fig.3.2に述べる．自由応答 (Fig.3.1) の時とは異なり，安定な平衡点に $(\pm 2.6934, \mp 0.2530)$ に収束することはなくなる．どのような初期値が与えられても原点に収束することが判明する．

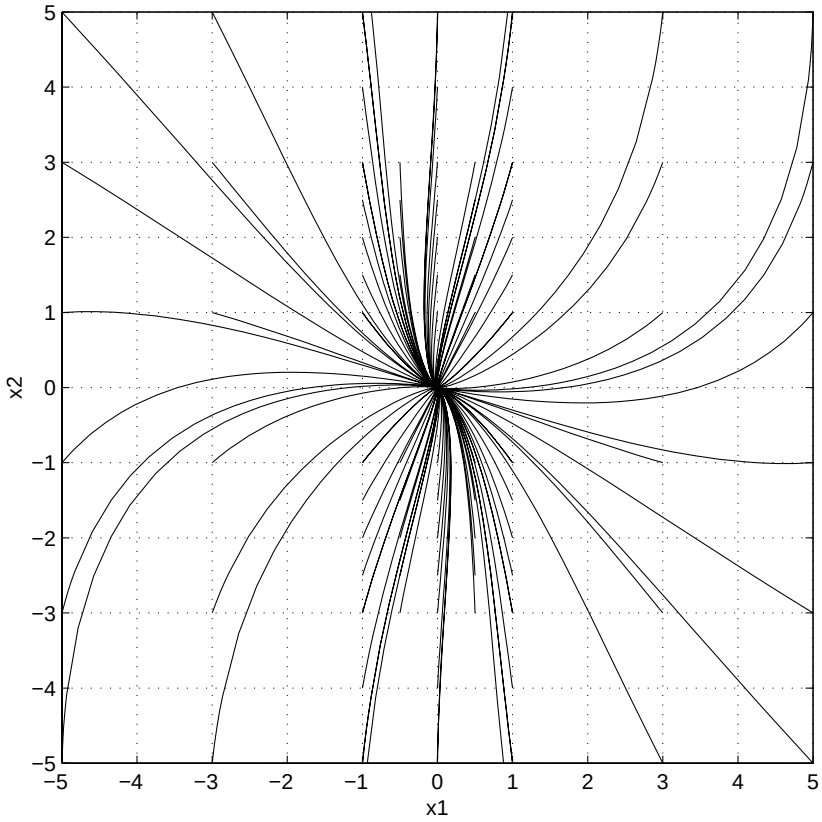

Fig. 3.2: Phase trajectories of the system when $u = u_1$.

また，2 次の次数で打ち切られたレギュレータは，以下 (3.26) 式のように 1 次の次数で打ち切られたレギュレータと同じであるため略す．

$$\boldsymbol{u}_2 := \boldsymbol{u}_1 \tag{3.26}$$

さらに，3 次の次数で打ち切られたレギュレータ

$$\boldsymbol{u}_3 := -\boldsymbol{R}^{-1}\boldsymbol{B}^T \sum_{k=1}^{3} \boldsymbol{S}_k \boldsymbol{x}^{[k]} \tag{3.27}$$

を定義し，$\boldsymbol{u} = \boldsymbol{u}_3$ の際のシステム (3.20) 式の状態変数 $\boldsymbol{x}$ のトラジェクトリを数値シミュレーションにより検証する．ただし，(3.11) 式と (3.12) 式から得られるリカッチ方程式の

解 $\boldsymbol{S}_2$，$\boldsymbol{S}_3$ は，

$$\boldsymbol{S}_2 = 0, \tag{3.28}$$

$$\boldsymbol{S}_3 = \begin{bmatrix} -4.490 & 14.23 & -5.336 & 0.865 \\ 8.385 & -5.347 & 1.499 & -2.937 \end{bmatrix} \times 10^{-4} \tag{3.29}$$

となる．$\boldsymbol{S}_1$ は，(3.25) 式である．その際のトラジェクトリを以下の Fig.3.3に述べる．1 次の次数で打ち切られたレギュレータ時と同様の形となる．ただ，初期値が大きくなるにつれて，Fig.3.2 とは異なる箇所が表れる．

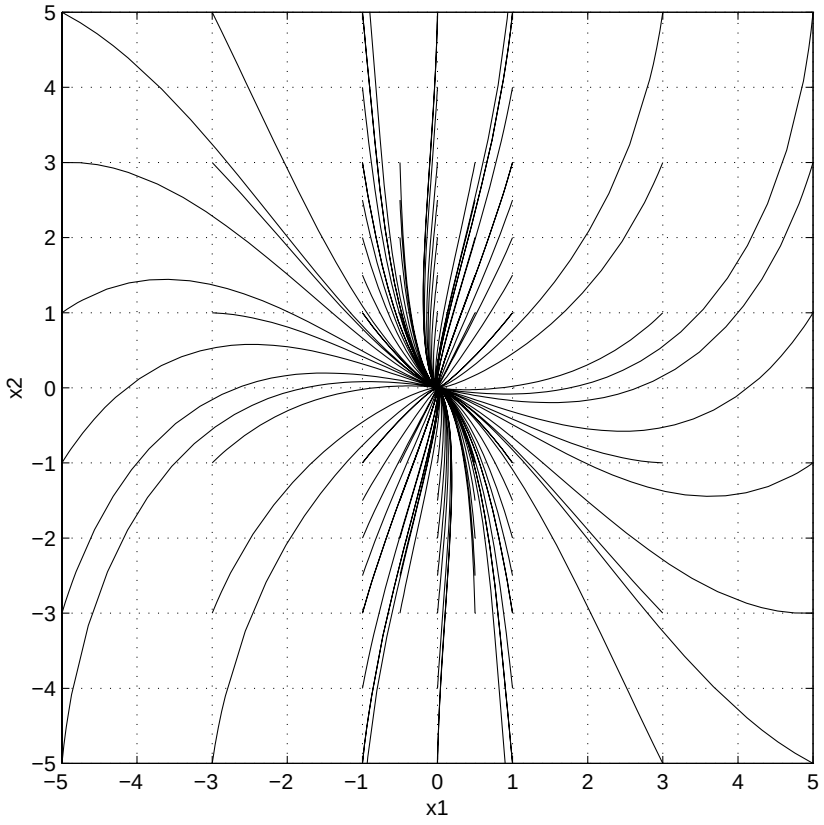

Fig. 3.3: Phase trajectories of the system when $u = u_3$.

最後に，初期値を $(x_1, x_2) = (6, 6)$ とした時に $\boldsymbol{u} = 0$,$\boldsymbol{u} = \boldsymbol{u}_1$,$\boldsymbol{u} = \boldsymbol{u}_3$ をそれぞれ入力した時の時間応答を Fig.3.4にて示す．上図が，初期値を $x_1 = 6$ とした時の x_1 のシステム ((3.20) 式) 応答を表しており，下図が，初期値を $x_2 = 6$ とした時の x_2 のシステム応答を示している．また，線の種類によって次数の違いを表わしている．

鎖線が，システム ((3.20) 式) の自由応答であり，実線が，1 次の次数で打ち切られたレギュレータ ((3.24) 式) によって補償されたシステムの応答であり，破線が，3 次の次数で

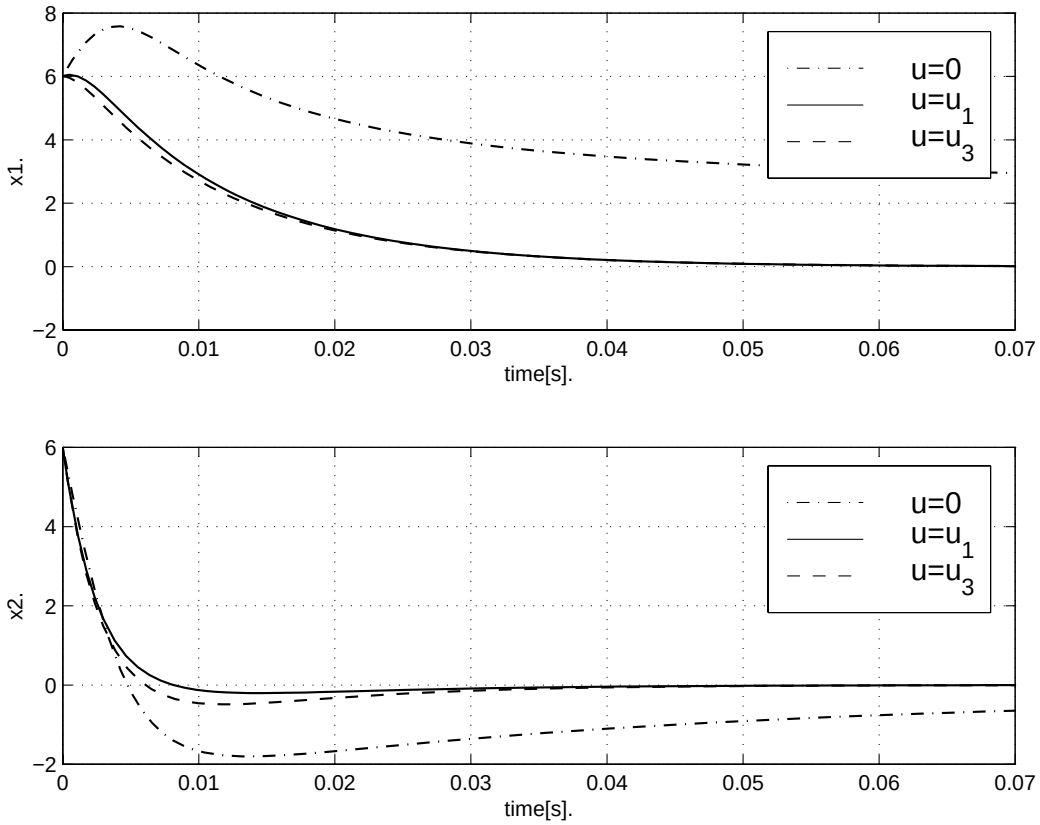

Fig. 3.4: Response of system : $u = 0$, $u = u_1$, $u = u_3$.

打ち切られたレギュレータ ((3.27) 式) によって補償されたシステムの応答である．0 次の次数で打ち切られたレギュレータ，すなわち，自由応答は収束せず．1，3 次の次数で打ち切られたレギュレータでは，原点に収束していることがわかる．ただ，3 次の次数で打ち切られたレギュレータでは，オーバーシュートが見られる． □

注意 3.2

レギュレータを高次にしたときの影響は，初期値 $\boldsymbol{x}_0$ が大きい場合にオーバーシュートとして現れることが指摘されている．この原因は，$\boldsymbol{x}_0$ が小さい場合には $\boldsymbol{x}^{[k]}$ の値が非常に小さくなり，微分方程式の線形部分 (行列 $\boldsymbol{A}$) の影響が支配的となるのに対し，$\boldsymbol{x}_0$ が大きい場合には $\boldsymbol{x}^{[k]}$ の値が非常に大きくなるので，レギュレータの高次項がシステムに作用するためと考えられる．また，最適レギュレータであれば k 次項による悪影響を $k+1$ 次以上の成分で補償できるものが，準最適レギュレータでは k 次までで高次項が途切れて

しまうため，k 次項自身あるいはそれ以下の項の成分で補償しきれなかった悪影響が残る結果となったと考えられる．

より詳しくは，参考文献 “非線形システムの最適レギュレータに関する研究” にまとめられている [3]. □

第 4 章

強化学習による打ち切り次数の導出

本章では，前章にて定義された非線形最適レギュレータ問題の (3.20)-(3.21) 式を実装する際に，導出せねばならない打ち切り次数を導出する強化学習の一例をまとめ，数値シミュレーションを述べる．使用した言語は matlab とし，pc 上にて実装を行っている．

4.1 定式化

問題 4.1

1 次の次数で打ち切られたレギュレータ (以下では，$\boldsymbol{u}_1$ と略す．)，もしくは，3 次の次数で打ち切られたレギュレータ (以下では，$\boldsymbol{u}_3$ と略す．) と以下の (4.1) 式によって構成される非線形システムでの $\boldsymbol{x}$ の応答に対して，強化学習の概念を用いることにより，非線形システムの特性により適した打ち切り次数の導出を数値シミュレーションにより行う．(ただし，以下では，断わらない限り (4.1) 式の非線形システムを単にシステムと略す．) すなわち，強化学習にて，$\boldsymbol{u}_1$ もしくは $\boldsymbol{u}_3$ の内から準最適な入力の選択を行う．まず，環境について，以下の (4.1) 式のように定義する．

$$\dot{\boldsymbol{x}} := \boldsymbol{A}\boldsymbol{x} + \boldsymbol{F}_3\boldsymbol{x}^{[3]} + \boldsymbol{B}\boldsymbol{u},\ \ \boldsymbol{x}(0) = \boldsymbol{x}_0 \neq 0 \tag{4.1}$$

ただし，(4.1) 式のそれぞれの係数行列は以下の (4.2)-(4.3) 式とする．

$$\boldsymbol{A} := \begin{bmatrix} 3.5 & -8.2 \\ 9.5 & -4.5 \end{bmatrix}, \tag{4.2}$$

$$\boldsymbol{B} := \begin{bmatrix} -2 & 2 \\ -4 & -3 \end{bmatrix},$$
$$\boldsymbol{F}_3 := \begin{bmatrix} -0.3 & 1.8 & 0.3 & 0.5 \\ -1.8 & -2.7 & -0.5 & -0.7 \end{bmatrix} \tag{4.3}$$

報酬 r_t は以下の (4.4) 式とする．

$$r_t := -\int_t^{t'} \left[\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u} \right] dt \tag{4.4}$$

ただし，t' は t の次時刻であり，今回のシミュレーションでは $0.005[s]$ 刻みで現時刻 t から次時刻 t' へ更新される．また，$\boldsymbol{Q}$ と $\boldsymbol{R}$ に関しては，以下のように定義する．

$$\boldsymbol{Q} := \begin{bmatrix} 10 & 1 \\ 1 & 5 \end{bmatrix},$$
$$\boldsymbol{R} := \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

(4.4) 式は，2 次形式評価関数の形である．この定義より，エージェントは報酬 r_t(今回の定義では，罰となる) を小さくなるように学習を進めることになる．よって，評価関数を最小化する行為 a_t を探す強化学習問題となり，最適レギュレータ問題を解くのに適した定義となる．

また，時間の更新間隔を $0.005[s]$ としているのは，今回用いた数値例におけるシステムでは，これ以上短い更新間隔では r_t が小さくなりすぎるため学習効果が，簡単には得られないためである．また，同一の状態 s_t (同一のセル) 上にて複数回学習が行われる可能性を防ぐ効果もある．ちなみに，1 学習過程の終了条件 $r_t < 0.0001$ も同様の理由による．

行為 a_t は，$\boldsymbol{u}_1$((4.5) 式)，または，$\boldsymbol{u}_3$((4.6) 式) を用いる．

$$\boldsymbol{u}_1 := -\boldsymbol{R}^{-1} \boldsymbol{B}^T \boldsymbol{S}_1 \boldsymbol{x}, \tag{4.5}$$
$$\boldsymbol{u}_3 := -\boldsymbol{R}^{-1} \boldsymbol{B}^T \sum_{k=1}^{3} \boldsymbol{S}_k \boldsymbol{x}^{[k]} \tag{4.6}$$

ただし，$\boldsymbol{u}_2$ は (4.5) 式と同じ ($\boldsymbol{u}_2 = \boldsymbol{u}_1$) であるから今数値例では考慮しない．また，(4.5) 式と (4.6) 式の係数行列 $\boldsymbol{S}_k$ $(k = 1, 2, 3)$ は，(3.11) 式と (3.12) 式から得られるリカッチ方程式から求められ，以下の (4.7)-(4.9) 式の様になる．

$$\boldsymbol{S}_1 = \begin{bmatrix} 11.66 & -0.275 \\ -0.275 & 4.334 \end{bmatrix} \times 10^{-2}, \tag{4.7}$$

$$\boldsymbol{S}_2 = 0, \tag{4.8}$$

$$\boldsymbol{S}_3 = \begin{bmatrix} -4.490 & 14.23 & -5.336 & 0.865 \\ 8.385 & -5.347 & 1.499 & -2.937 \end{bmatrix} \times 10^{-4} \tag{4.9}$$

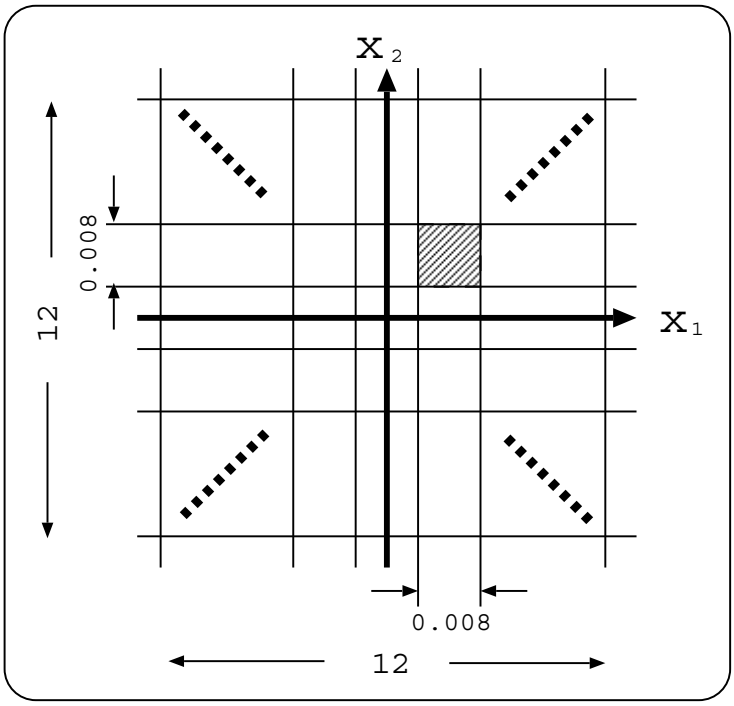

Fig. 4.1: State s_t

状態 s_t は，(4.1) 式で表されるシステムの状態変数 $\boldsymbol{x}$ を用い，ルックアップテーブルにて表す．すなわち，以下の Fig.4.1のように，一辺 12 のからなる平面を一辺 0.008 の端片に区切った斜線部分 (セル) が状態 s_t となる．ちなみに，今シミュレーションにおけるセルの総量は 2253001 個となる．

個々のパラメータの値について説明を捕捉する．状態 s_t を表すセルの一辺の大きさを 0.008 としているのは，時間の更新間隔 $0.005[s]$ が行われた時に，同一セル上で学習が行われないサイズにまでセルを小さくする必要があるためである．よって，セルの一辺の大きさを 0.008 とする．そして，そのセルを本計算機環境におけるメモリぎりぎりまでの大きさまで，セルを敷き詰めると，ルックアップテーブルの大きさはおおよそ 12 となる．よって，本シミュレーションではルックアップテーブルの一辺を 12 とする．その結果，3 次のレギュレータを選択することによってオーバーシュートが発生していることが視認できる大きさの初期値 (今シミュレーションでは，$|x_1| \geq 6, |x_2| \geq 6$) ともなっている．

$Q(s_t, a_t)$ 値は，以下の (4.10) 式より更新される．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})) \tag{4.10}$$

行為選択はϵ-Greedy($\epsilon = 0.01$) 法とする．なお，学習のパラメータである学習率と割引率は，それぞれ，$\alpha = 0.5$，$\gamma = 0.9$ として学習を行い，(4.4) 式が 0.0001 以下の値になった時に，一つの学習過程が終了する．

以上の非線形最適レギュレータ問題を強化学習の枠組で捉え直したものを以下 (Table 4.1) に示す． ■

環境	(4.1) 式
状態 s_t	システムの状態変数 $\boldsymbol{x}$ ，Fig.4.1
報酬 r_t	(4.4) 式
行為 a_t	(4.5) 式もしくは (4.6) 式
$Q(s_t, a_t)$ 値の更新	(4.10) 式
行為選択	ϵ-Greedy 法　($\epsilon = 0.01$)
現時刻 t から次時刻 t' の更新	$0.005[s]$
1 学習過程の終了条件	$\lvert r_t \rvert < 0.0001$
学習率 α	0.5
割引率 γ	0.9

Table 4.1: Reinforcement learning problem.

4.2　数値シミュレーション

ここでは，学習を行った数値例を示す．最初にその結果についてまとめ，そして，考察を述べる．

数値例 4.1

前節の定義を用いて，強化学習にて (4.1) 式の非線形システムの特性に適した打ち切り次数の導出を行う．まず，初期値 ((4.11) 式) を与えた場合の数値シミュレーションを以下の

Fig. 4.2に示す.

$$(x_1, x_2) = (6, 6) \tag{4.11}$$

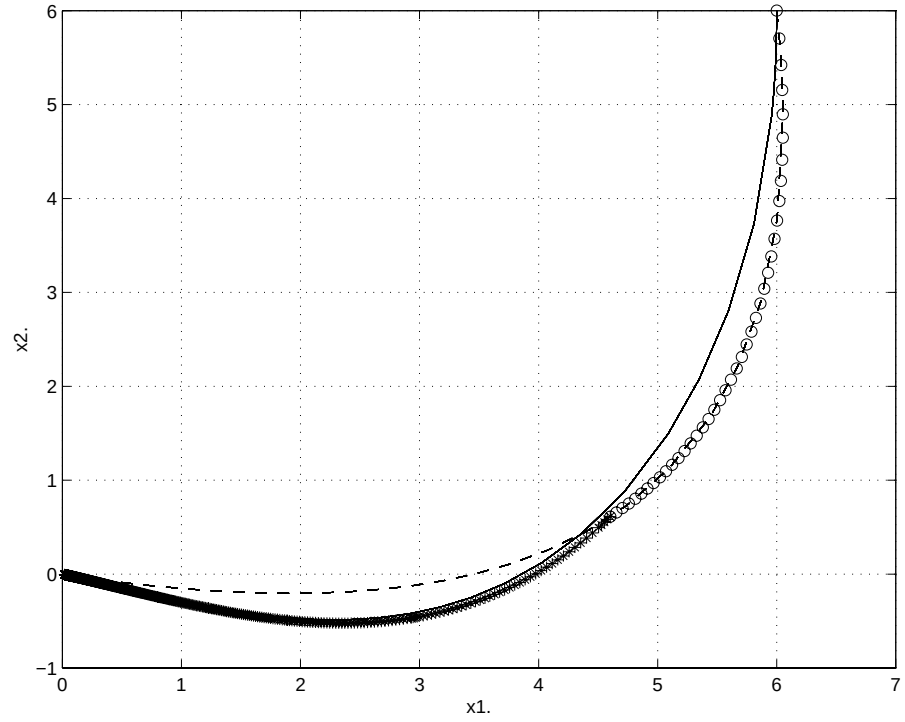

Fig. 4.2: Trajectories after reinforcement learning and $u = u_1$ and $u = u_3$.

Fig.4.2は，$\boldsymbol{u}_1$ もしくは$\boldsymbol{u}_3$ が選択され補償されたシステムのトラジェクトリと，学習後の打ち切り次数によって補償されたシステム (以下では，単に学習後システムと略す．) のトラジェクトリを表している．すなわち，星線と円線にて表されるのが学習後システムのトラジェクトリである．星線は $\boldsymbol{u}_3$ によって補償されたシステム (以下では，$\boldsymbol{u}_3$ システムと略す．) のトラジェクトリであり，円線は $\boldsymbol{u}_1$ によって補償されたシステム (以下では，$\boldsymbol{u}_1$ システムと略す．) のトラジェクトリーである．また，破線は $\boldsymbol{u}_1$ として，実線は $\boldsymbol{u}_3$ として，それぞれ選択されたトラジェクトリーである．学習後システムにおけるトラジェクトリの遷移は，初期値 $x_1 = 6, x_2 = 6$ からは $\boldsymbol{u}_1$ が選択される．その後，およそ， $x_1 = 4.6, x_2 = 0.5$ の時点で切り替えしが行われ， $\boldsymbol{u}_3$ だけが 13 回選択され原点に収束している．結局，合計 14 回の打ち切り次数の切り替えしを行う．

よって，Fig.4.2より，より良い打ち切り次数とは，単純に高次数であれば良いとは限らず，状態 s_t (今シミュレーションでは状態変数) に応じて打ち切り次数を切り替えさなけ

ればならないことが判明する．

次に，時間応答を Fig.4.3にて示す．上図が，初期値を $x_1 = 6$ とした時の x_1 のシステムの応答を表しており，下図が，初期値を $x_2 = 6$ とした時の x_2 のシステムの応答を示している．また，線の種類によって次数の違いを表す．実線が，$\boldsymbol{u}_1$ システムの応答である．重なっているため確認が不可能となっているが，鎖線が，学習後システムの応答である．また，破線が，$\boldsymbol{u}_3$ システムの応答である．

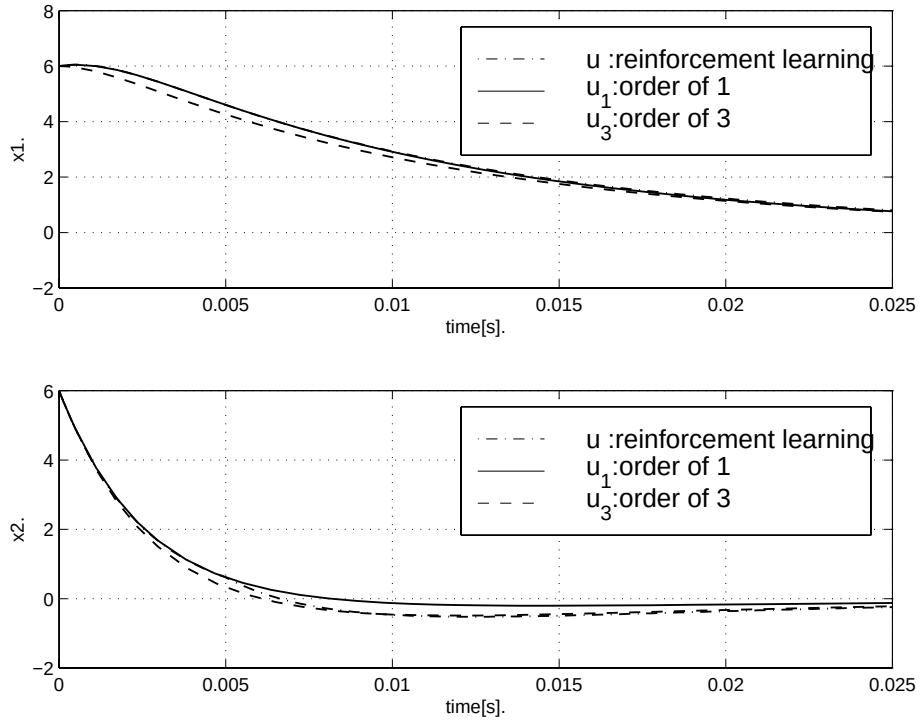

Fig. 4.3: Response of system : u:reinforcement learning，u_1:order of 1，u_3:order of 3．

したがって，オーバーシュートがおよそ最大になる $t = 0.0012[s]$ の時点では，オーバーシュートを起こす次数の打ち切り次数が選択されていることが Fig.4.3より明らかになる．すなわち，オーバーシュートの発生を抑えるために $\boldsymbol{u}_1$ が選択されるわけではないことが判明する．

また，$\boldsymbol{u}_1$ システムもしくは $\boldsymbol{u}_3$ システムの評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ と学習後システムの評価値の変化を以下の Fig.4.4に示す．この Fig. 4.4の縦軸は $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ を表し，横軸は時間を表す．また，破線は $\boldsymbol{u}_1$ システムの評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ ，鎖線は $\boldsymbol{u}_3$ システムの評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ を表し，実線は学習後システムの評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ である．

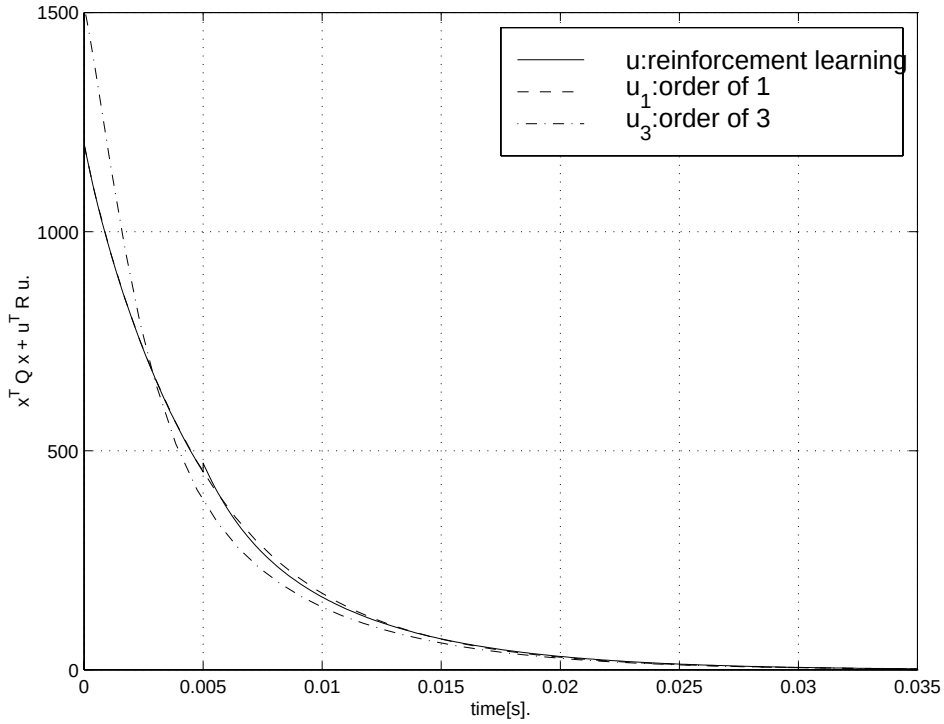

Fig. 4.4: Shift of $x^T Rx + u^T Qu$.

Fig.4.4は，強化学習による学習後のシステムの評価関数$\boldsymbol{J}$ の値が，打ち切り次数が“ 1 ”もしくは“ 3 ”だけに固定されることで補償されるシステムの評価関数$\boldsymbol{J}$ の値よりも小さくなることを表している．また，打ち切り次数の切り替えしが行われるタイミングを表している．すなわち，鎖線と実線が交差する付近 $t = 0.0025[s]$ で，切り替えしが必要であることがわかる．その上，具体的なそれぞれの評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ の値は以下のTable 4.2の様になる．参考までに評価関数$\boldsymbol{J}$ の理論値 ((3.19) 式) を付記する．

	(6，6)
学習後の評価値	6.3507
打ち切り次数 1 の評価値	6.6592
打ち切り次数 3 の評価値	6.7426
理論値	6.2439

Table 4.2: Evaluation value.

したがって，打ち切り次数を固定したまま制御を行うよりも，打ち切り次数を切り替えすことで状況に応じた入力 $\boldsymbol{u}$ を選択できるようになり，より良い制御が行われるということが，Table 4.2より判明する．

さらに，Fig.4.4の測定直後 $t = 0$ の $\boldsymbol{u}_1$ システムもしくは $\boldsymbol{u}_3$ システムに関する評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ の差は，

$$\boldsymbol{u}_3 - \boldsymbol{u}_1 = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}_3\boldsymbol{x}^{[3]} \tag{4.12}$$

に関連するものと推定できる．よって，(4.12) 式と初期値の大きさが，$\boldsymbol{u}_1$ システムもしくは $\boldsymbol{u}_3$ システムの評価値 $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ の差異を決定づけ，打ち切り次数の切り替えしが行われると考えられる．言い替えると，初期値が大きくなればなるほど (4.12) 式と評価値の値が反映されるので，システムの状態変数 $\boldsymbol{x}$ 値が大きい間は，次数の低い打ち切り次数が選択される．その後，システムの状態変数 $\boldsymbol{x}$ の値が原点に漸近すると，高次数の打ち切り次数が選択されることが判明する．以上より，ベキ級数で制御則が与えられるシステムの最適レギュレータ問題を解く際に，大きな初期値を与える場合には，$\boldsymbol{Q}$ をより大きく $\boldsymbol{R}$ をより小さくすることが必要であると考えられる．

最後に，初期値 ((4.11) 式) を複数回 (今回は 50 回) 選び，そのたびに最適なトラジェクトリを辿る打ち切り次数の遷移が選択されている確率を以下の Fig.4.5で示す．横軸は学習回数 (今回は 50 回) を表している．学習により最適な打ち切り次数の遷移が導出され，選択されていることが，Fig.4.5より明らかになる． □

数値例 4.2

前小節より，$\boldsymbol{Q}$ をより大きく $\boldsymbol{R}$ をより小さくすることで，(4.12) 式と初期値の影響を小さくできるのではないかと推測を行った．そこで，$\boldsymbol{Q}$ と $\boldsymbol{R}$ を以下のように定義し，数値シミュレーションを行う．その他の係数行列は，同様のままである．すなわち，

$$\boldsymbol{A} := \begin{bmatrix} 3.5 & -8.2 \\ 9.5 & -4.5 \end{bmatrix},$$

$$\boldsymbol{B} := \begin{bmatrix} -2 & 2 \\ -4 & -3 \end{bmatrix},$$

$$\boldsymbol{F}_3 := \begin{bmatrix} -0.3 & 1.8 & 0.3 & 0.5 \\ -1.8 & -2.7 & -0.5 & -0.7 \end{bmatrix},$$

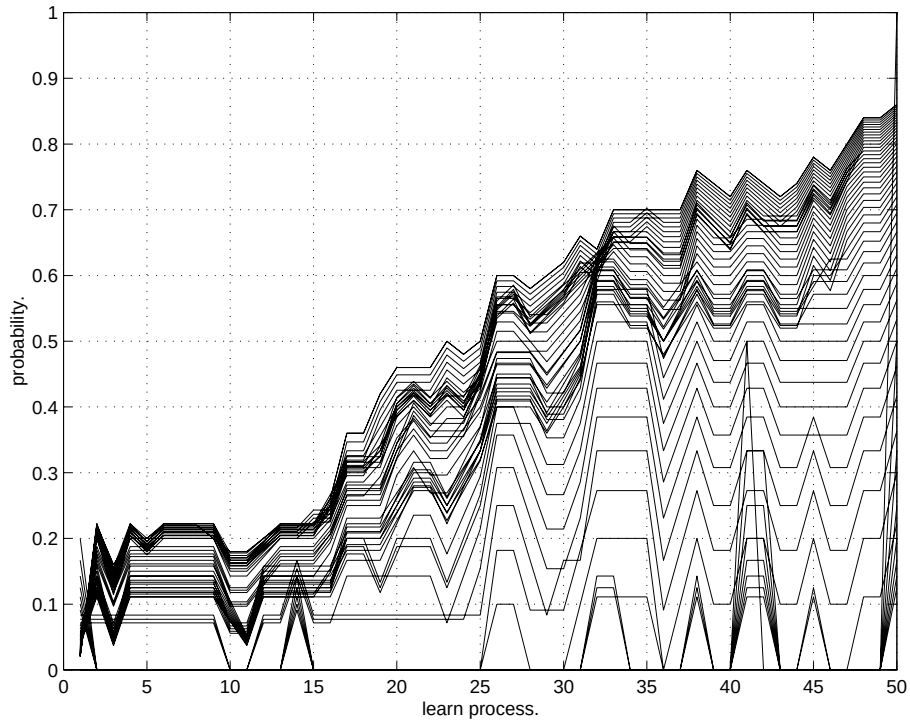

Fig. 4.5: Probability by which the order of truncation of the semi-best control law after learning is selected.

$$\boldsymbol{Q} := \begin{bmatrix} 10 & 1 \\ 1 & 10 \end{bmatrix},$$

$$\boldsymbol{R} := \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix}$$

とする．

その結果のトラジェクトリーを以下の Fig.4.6に示す．表記，記号は，Fig.4.2の時と同様である．したがって，初期値として，$(x_1, x_2) = (6, 6)$ を与えたとしても，$\boldsymbol{Q}$ をより大きく $\boldsymbol{R}$ をより小さく変更することで (4.12) 式と初期値の影響を減少させることが，Fig.4.6より示される．その結果，制御開始時点 $t = 0$ であり，かつ，初期値が大きいにもかかわらず高次の打ち切り次数が選択されていることが判明する．また，Fig.4.7より，オーバーシュートが抑えられているのがわかる．しかも，制御の開始時点にて高次の打ち切り次数を用いたとしても，打ち切り次数の切り替えしが行われる方が，より良い制御を行っていることが Fig.4.6より示される．また，評価値についても Table 4.3と Fig.4.8 として付記

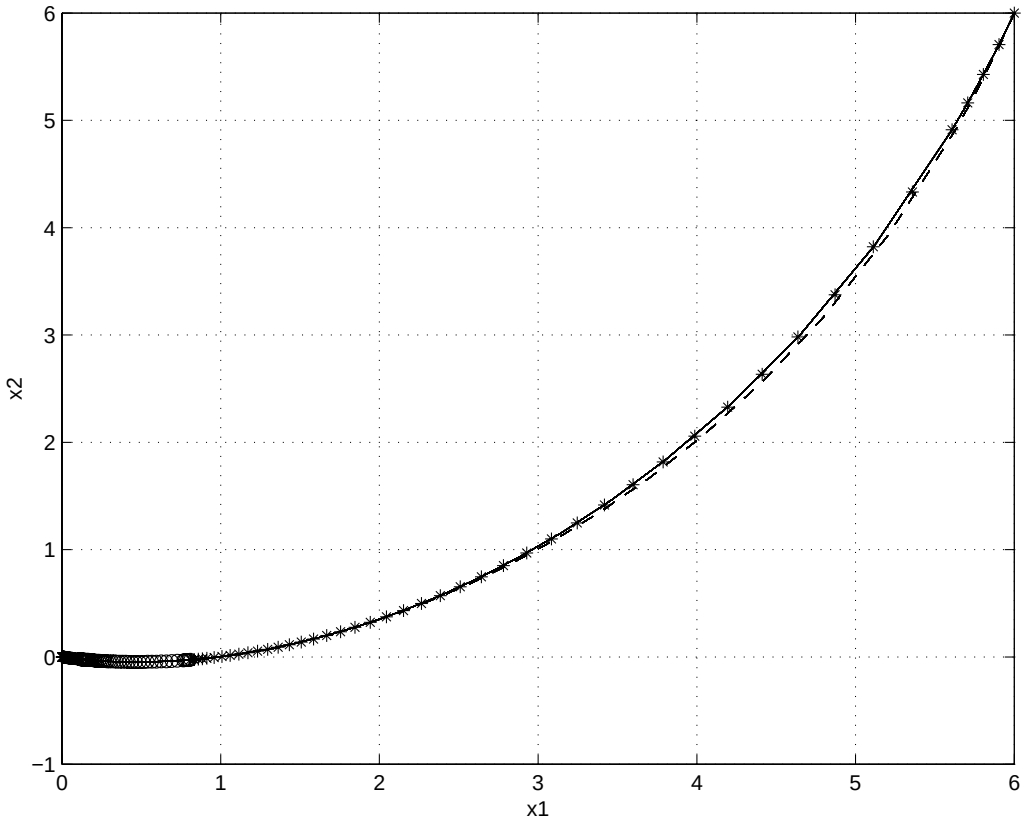

Fig. 4.6: Trajectories after reinforcement learning and $u = u_1$ and $u = u_3$.

しておく. □

	(6, 6)
学習後の評価値	1.4728
打ち切り次数 1 の評価値	1.5342
打ち切り次数 3 の評価値	1.5475
理論値	1.3783

Table 4.3: Evaluation value.

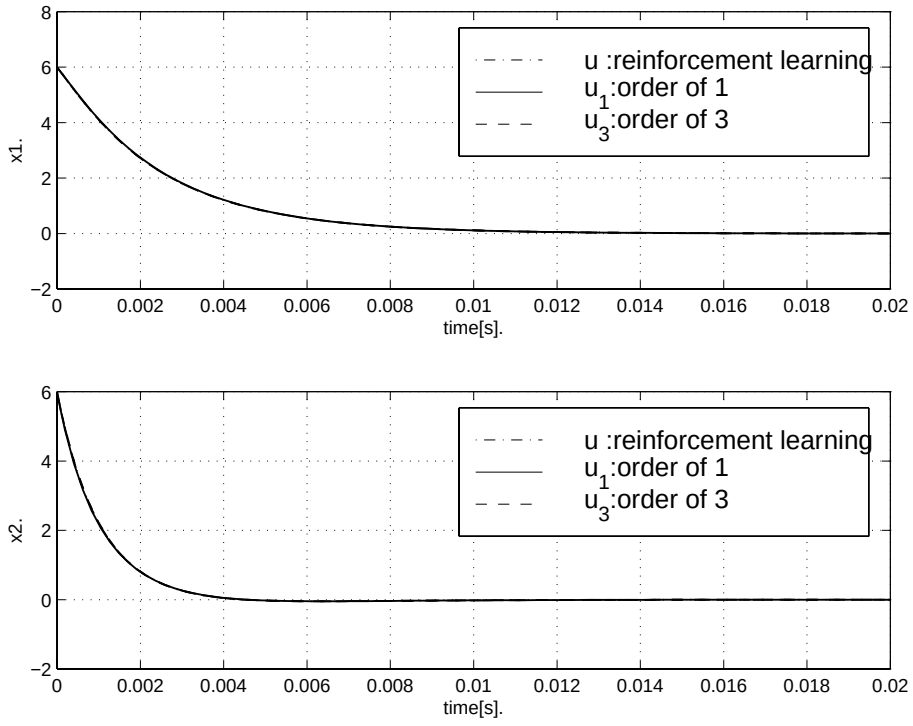

Fig. 4.7: Response of system : u:reinforcement learning, u_1:order of 1, u_3:order of 3.

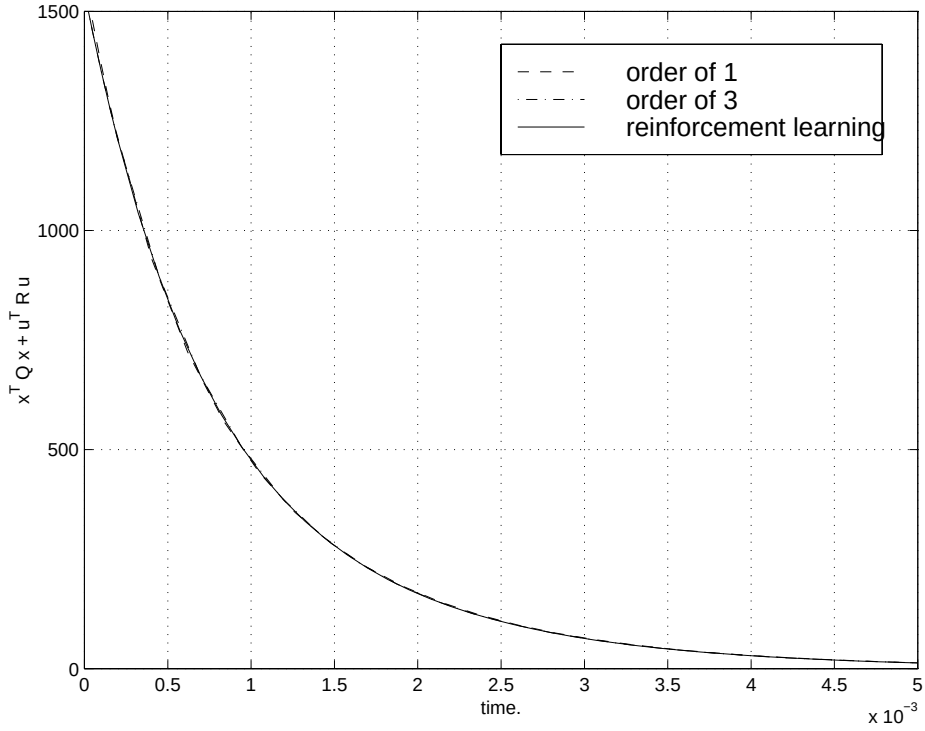

Fig. 4.8: Shift of $x^T Rx + u^T Qu$.

第 5 章

結論

5.1 まとめ

本研究では以下のことを行った.

- ベキ級数にて制御則が与えられる非線形システムに関する制御則の打ち切り次数の導出を強化学習にて行った.

その結果，次のことが明らかになった.

- より良い打ち切り次数とは，単純に高次数であれば良いとは限らず，状況に応じて打ち切り次数を切り替えす必要があること.
- 打ち切り次数を切り替えすことで，状況に応じた入力 $\boldsymbol{u}$ を選択できることにより，より良い制御を行なえるということ.

上記の結果，学習結果より示されたグラフから，オーバーシュートと打ち切り次数には関連が無いことが判明した．そこで，評価値の変化とシステムの時間応答により，$\boldsymbol{Q}$ と $\boldsymbol{R}$ に関する設定法に付いて検討した．その設定法に基づいて数値シミュレーションを行い，オーバーシュートが無くなることを実証した.

5.2 課題

以下のような課題がある.

- この学習プログラムは，$\boldsymbol{x} = [x_1, x_2]$ となる状態変数の場合にしか対応していない．
- ルックアップテーブルを用いて価値関数を表したため，大きな状態空間を扱う学習プログラムとなっていないこと．

第 6 章

謝辞

2 年間にわたり本研究全般に関してきめ細かい御指導と御鞭撻を賜わりました，吉田武稔助教授に心から感謝の意を表します．同様に御支援を賜わりました，主指導教官の桜井彰人教授に心から感謝の意を表します．

そして，本研究とは異なる分野での興味深い研究の御指導をして頂きました，副テーマ指導教官の中森義輝教授に深い感謝の意を表します．

また，本研究に関して御助言を賜わりました，本講座の教授である Gu Jifa 教授に感謝の意を表します．さらに，本研究に関して的確な御助言を賜わりました，佐藤当秀氏，田中雄介氏，田野勇二氏，丁子英樹氏，波当根亮氏，福島仁氏，山本夏江氏に深く感謝し，以後のご活躍をお祈り致します．

最後に，研究生活を共にした，複合システム論講座 Gu・吉田研究室の皆様に厚く御礼を申し上げます．

参考文献

[1] 畝見，“強化学習”， 人口知能学会，Vol．9，No．6，pp．830-836，1994.

[2] 木村，“部分マルコフ過程決定下での強化学習:確率的傾斜法による接近”，Ph．D．thesis，東京工業大学，1997.

[3] 田中，“非線形システムの最適レギュレータに関する研究”，Ph．D．Subthesis，北陸先端科学技術大学院大学，1998.

[4] 内藤，中森，吉田，“ファジィ推論を適応した強化学習の一考察”， システム/情報部門シンポジウム *'99* 講演論文集，pp．255-260，1999.

[5] 堀内，藤野，片井，椹木，“連続値入出力を扱うファジィ内挿型 Q-Learning の提案”，計測自動制御論文集，Vol．35，No．2，pp．271-279，1999.

[6] 石島，島，石動，山下，三平，渡部，非線形システム論， 計測自動制御学会，コロナ社， 1995.

[7] 伊藤， 自動制御概論， 昭晃堂，1983.

[8] 児玉，須田，システム制御のためのマトリクス理論，計測自動制御学会，コロナ社，1978.

[9] 示村， 線形システム解析入門，コロナ社，1987.

[10] 志水，最適制御の理論と計算法，コロナ社，1994.

[11] 日本ファジィ学会，講座ファジィ５，ファジィ制御，日本ファジィ学会，日刊工業新聞社， 1993.

[12] 浜田，松本、高橋，現代制御理論入門，コロナ社，1997.

[13] K. Zhou. and J. C. Doyle and K. Glover, *Robust and Optimal Control*, Prentice Hall, 1995. (劉，羅 (共訳)，ロバスト最適制御，コロナ社，1997)

[14] A. G. Barto, el. al., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems", *IEEE Transactions on Systems Man and Cybernetics*, Vol. SMC-13, No. 5, pp. 834-846, 1983.

[15] S. J. Bradtke, "Reinforcement learning applied to linear quadratic regulation", *Advances in Neural Information Processing Systems: Proceedings of the 1992 Conference*, pp. 295-302. 1993.

[16] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic contro using policy iteration", *American Control Conference*: *Proc.* , pp. 3475-3479. 1994.

[17] R. H. Crites, and A. G. Barto, "Improving Elevator Perfomance Using Reinforcement Learning ", *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference*, pp. 1017-1023. 1996.

[18] R. Munos, "A Convergent Reinforcement learning Algorithm in the continuous case based on a Finite Difference Method" *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 826-831. 1997

[19] J. A. Frueh, and M. Q. Phan, "Linear Quadratic Optimal Learning Control(LQL) " *Proceedings of the 37th IEEE Confrernce on Decision & Control* pp. 678-683. 1998

[20] T. Yoshida and K. A. Loparo, "Quadratic Regulatory Theory for Analytic Nonlinear Systems with Additive Controls", *Automatica*, vol. 25, no. 4, pp. 531-544, 1989.

[21] T. Yoshida, *Quadratic Regulator Theory for Analytic Nonlinear Systems with Additive Controls*, Ph. D. thesis, Case Western Reserve University, Cleveland, Ohio. 1984.

[22] W. Zhang. and T. G. Dietterich, "Reinforcement learning applied to Job-shop Scheduling", *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1114-1120. 1995.

[23] G. L. Blankenship, "Lie Theory and Moment Stability Problem in Stochastic Differential Equation", *Proceedings of the IFAC75 6th World Congress*, pp. 33.2.1-36.2.8. 1975.

[24] R. S. Sutton. and A. G. Barto, *Reinforcement Learning An Introduction*, The MIT Press, 1998.

[25] M. L. Puterman, *Markov Decision Processes Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., 1994.

[26] The MATH WORKS Inc., *Using MATLAB*, The Math Works Inc., 1997.

第 A 章

付録

A.1　付録

高次のリカッチ方程式を解く場合に役立つように，$\bar{\boldsymbol{A}}_{[p]}$ を計算する方法についてまとめておく．より詳しくは，参考文献にまとめられている [23]．計算機に実装する場合にも役立つと思われる．

A.2　基底ベクトルの変換による$\bar{A}_{[p]}$の計算法

A.2.1　$\bar{A}_{[p]}$の導出

k 次のリカッチ方程式の中に出てくる $\bar{\boldsymbol{A}}_{[p]}$ の計算法について説明する．

$$
\begin{aligned}
\dot{\tilde{x}}_1 &= 0 \\
\dot{\tilde{x}}_2 &= 0 \\
&\vdots \\
\dot{\tilde{x}}_i &= \tilde{x}_j \\
\dot{\tilde{x}}_{i+1} &= 0 \\
&\vdots \\
\dot{\tilde{x}}_n &= 0
\end{aligned}
$$

(A.1)

を，

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{E}}_{ij}\tilde{\boldsymbol{x}}, \quad \tilde{\boldsymbol{E}}_{ij} : (i, j) \text{ 要素が } 1, \text{ 他の要素が } 0 \text{ となる行列} \tag{A.2}$$

と表す．同様にして一般項 $\tilde{x}_1^{p_1}\tilde{x}_2^{p_2}\cdots\tilde{x}_n^{p_n}(p = \sum_{i=1}^{n} p_i)$ の場合，

$$\frac{d}{dt}\left(\tilde{x}_1^{p_1}\tilde{x}_2^{p_2}\cdots\tilde{x}_n^{p_n}\right) = p_i\tilde{x}_1^{p_1}\tilde{x}_2^{p_2}\cdots\tilde{x}_i^{p_i-1}\cdots\tilde{x}_j^{p_j+1}\cdots\tilde{x}_n^{p_n}$$

(ただし $i = j$のとき　$\dot{\tilde{x}}_i = \tilde{x}_j, otherwise\dot{\tilde{x}}_i = 0$)

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{E}}_{ij}\tilde{\boldsymbol{x}}$$

すなわち

$$\dot{\tilde{\boldsymbol{x}}}^{[p]} = \left(\tilde{\boldsymbol{E}}_{ij}\right)_{[p]}\tilde{\boldsymbol{x}}^{[p]} \tag{A.3}$$

となる．ただし，$\left(\tilde{\boldsymbol{E}}_{ij}\right)_{[p]} : N(p_1, \ldots, p_i - 1, \ldots, p_j + 1, \ldots, p_n) \times N(p_1, p_2, \ldots, p_n)$ 要素が p_i となる $N(n;p) \times N(n;p)$ の行列である．また，$||\boldsymbol{x}^{[p]}|| = ||\boldsymbol{x}||^p$ となるように正規化係数行列を導入する．$i.e.$,

$$\boldsymbol{x}^{[p]} = \boldsymbol{C}_{np}(p_1, p_2, \ldots, p_n)\tilde{\boldsymbol{x}}^{[p]} \tag{A.4}$$

行列 C_{np} は対角要素に

$$\begin{aligned}\boldsymbol{C}_{np}(p_1, p_2, \ldots, p_n) &= \sqrt{\begin{pmatrix} p \\ p_1 \end{pmatrix}\begin{pmatrix} p - p_1 \\ p_2 \end{pmatrix}\cdots\begin{pmatrix} p - p_1 - p_2\cdots p_{n-1} \\ p_n \end{pmatrix}} \\ &= \sqrt{\frac{p!}{p_1!p_2!\ldots p_n!}}\end{aligned} \tag{A.5}$$

をもつ $N(n;p) \times N(n;p)$ の対角行列である．

(A.4) 式を微分方程式 (A.3) 式に代入して得られる次式

$$\dot{\tilde{\boldsymbol{x}}}^{[p]} = \left(\tilde{\boldsymbol{E}}_{ij}\right)_{[p]}\boldsymbol{C}_{np}^{-1}\boldsymbol{x}^{[p]} \tag{A.6}$$

を，(A.4) を直接時間微分して得られる式

$$\dot{\boldsymbol{x}}^{[p]} = \boldsymbol{C}_{np}\dot{\tilde{\boldsymbol{x}}}^{[p]} \tag{A.7}$$

に代入すると $\dot{\boldsymbol{x}}^{[p]} = \left(\boldsymbol{E}_{ij}\right)_{[p]}\boldsymbol{x}^{[p]}$ であるから

$$\left(\boldsymbol{E}_{ij}\right)_{[p]} = \boldsymbol{C}_{np}\left(\tilde{\boldsymbol{E}}_{ij}\right)_{[p]}\boldsymbol{C}_{np}^{-1} \tag{A.8}$$

となる．これを使えば $\bar{\boldsymbol{A}}_{[p]}$ を

$$\bar{\boldsymbol{A}}_{[p]} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\left(\boldsymbol{E}_{ij}\right)_{[p]} \tag{A.9}$$

で表せるので

$$\dot{\boldsymbol{x}}^{[p]} = \bar{\boldsymbol{A}}_{[p]}\boldsymbol{x}^{[p]} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\left(\boldsymbol{E}_{ij}\right)_{[p]}\right)\boldsymbol{x}^{[p]} \tag{A.10}$$

である．

$$\dot{\tilde{\boldsymbol{x}}}^{[p]} = \tilde{\boldsymbol{A}}_{[p]}\tilde{\boldsymbol{x}}^{[p]} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\left(\tilde{\boldsymbol{E}}_{ij}\right)_{[p]}\right)\tilde{\boldsymbol{x}}^{[p]} \tag{A.11}$$

との関係から $\bar{\boldsymbol{A}}_{[p]}$ と $\tilde{\boldsymbol{A}}_{[p]}$ の間には

$$\bar{\boldsymbol{A}}_{[p]} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\boldsymbol{C}_{np}\left(\tilde{\boldsymbol{E}}_{ij}\right)_{[p]}\boldsymbol{C}_{np}^{-1} = \boldsymbol{C}_{np}\tilde{\boldsymbol{A}}_{[p]}\boldsymbol{C}_{np}^{-1} \tag{A.12}$$

すなわち

$$\tilde{\boldsymbol{A}}_{[p]} = \boldsymbol{C}_{np}^{-1}\bar{\boldsymbol{A}}_{[p]}\boldsymbol{C}_{np} \tag{A.13}$$

の関係を得る．以上から $(\tilde{\boldsymbol{E}}_{ij})_{[p]}$ によって $\tilde{\boldsymbol{A}}_{[p]}$ を求め，$\bar{\boldsymbol{A}}_{[p]} = \boldsymbol{C}_{np}\tilde{\boldsymbol{A}}_{[p]}\boldsymbol{C}_{np}^{-1}$ により $\boldsymbol{A}_{[p]}$ を求める．

A.2.2 $\bar{A}_{[2]}$の導出例

$n = 2$ の場合，index は $N(p_1, p_2) = 1 + p_2$ $(p = p_1 + p_2)$ である．次に，$(\boldsymbol{E}_{ij})_{[p]}$ の計算を行う．ここでは

$$\boldsymbol{A} := \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{A.14}$$

とする．

E_{11}の計算:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}_{11}\boldsymbol{x} \Rightarrow \begin{cases} \dot{x}_1 = x_1 \\ \dot{x}_2 = 0 \end{cases} \text{より} \tag{A.15}$$

$$\frac{d}{dt}\left(x_1^{p_1}x_2^{p_2}\right) = p_1 x_1^{p_1}x_2^{p_2} \tag{A.16}$$

$p = 2$ の場合,

$$\frac{d}{dt}(x_1^2 x_2^0) = 2x_1^2$$
$$\frac{d}{dt}(x_1^1 x_2^1) = x_1 x_2$$
$$\frac{d}{dt}(x_1^0 x_2^2) = 0$$
$$\text{すなわち } (\boldsymbol{E}_{11})_{[2]} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.17}$$

E_{12}の計算:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}_{12}\boldsymbol{x} \Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 0 \end{cases} \text{ より} \tag{A.18}$$

$$\frac{d}{dt}\left(x_1^{p_1} x_2^{p_2}\right) = p_1 x_1^{p_1 - 1} x_2^{p_2 + 1} \tag{A.19}$$

$p = 2$ の場合,

$$\frac{d}{dt}(x_1^2 x_2^0) = 2x_1 x_2$$
$$\frac{d}{dt}(x_1^1 x_2^1) = x_2^2$$
$$\frac{d}{dt}(x_1^0 x_2^2) = 0$$
$$\text{すなわち } (\boldsymbol{E}_{12})_{[2]} = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.20}$$

E_{21}の計算:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}_{21}\boldsymbol{x} \Rightarrow \begin{cases} \dot{x}_1 = 0 \\ \dot{x}_2 = x_1 \end{cases} \text{ より} \tag{A.21}$$

$$\frac{d}{dt}\left(x_1^{p_1} x_2^{p_2}\right) = p_2 x_1^{p_1 + 1} x_2^{p_2 - 1} \tag{A.22}$$

$p=2$ の場合,

$$\frac{d}{dt}(x_1^2 x_2^0) = 0$$
$$\frac{d}{dt}(x_1^1 x_2^1) = x_1^2$$
$$\frac{d}{dt}(x_1^0 x_2^2) = 2x_1 x_2$$

$$\text{すなわち } (\boldsymbol{E}_{21})_{[2]} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \tag{A.23}$$

E_{22}の計算:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}_{22}\boldsymbol{x} \Rightarrow \begin{cases} \dot{x}_1 = 0 \\ \dot{x}_2 = x_2 \end{cases} \text{ より} \tag{A.24}$$

$$\frac{d}{dt}\left(x_1^{p_1} x_2^{p_2}\right) = p_2 x_1^{p_1} x_2^{p_2} \tag{A.25}$$

$p=2$ の場合,

$$\frac{d}{dt}(x_1^2 x_2^0) = 0$$
$$\frac{d}{dt}(x_1^1 x_2^1) = x_1 x_2$$
$$\frac{d}{dt}(x_1^0 x_2^2) = 2x_2$$

$$\text{すなわち } (E_{21})_{[2]} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \tag{A.26}$$

したがって,

$$\tilde{\boldsymbol{A}}_{[2]} = a(\boldsymbol{E}_{11})_{[2]} + b(\boldsymbol{E}_{12})_{[2]} + c(\boldsymbol{E}_{21})_{[2]} + d(\boldsymbol{E}_{22})_{[2]} \tag{A.27}$$

$$= \begin{bmatrix} 2a & 2b & 0 \\ c & a+d & b \\ 0 & 2c & 2d \end{bmatrix} \tag{A.28}$$

一方，$\boldsymbol{x}^{[2]} = \boldsymbol{C}_{np}\tilde{\boldsymbol{x}}^{[2]}$ から

$$\boldsymbol{C}_{np} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{C}_{np}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.29}$$

なので

$$\bar{\boldsymbol{A}}_{[2]} = \boldsymbol{C}_{np}\tilde{\boldsymbol{A}}_{[2]}\boldsymbol{C}_{np}^{-1} = \begin{bmatrix} 2a & \sqrt{2}b & 0 \\ \sqrt{2}c & a+d & \sqrt{2}b \\ 0 & \sqrt{2}c & 2d \end{bmatrix} \tag{A.30}$$

を得る．同様の手法にて

$$\bar{\boldsymbol{A}}_{[3]} = \begin{bmatrix} 3a & \sqrt{3}b & 0 & 0 \\ \sqrt{3}c & 2a+d & 2b & 0 \\ 0 & 2c & a+2d & \sqrt{3}b \\ 0 & 0 & \sqrt{3}c & 3d \end{bmatrix} \tag{A.31}$$

を得ることができる．