

平成 24 年度

# 修士学位論文

題 目

内部状態と行動をベースとした  
ロボットの内部状態予測手法の提案

提 出 者

室蘭工業大学大学院工学研究科

情報電子工学系

専攻

平成 23 年 4 月 入学

氏 名 杉本 大志

提出年月日 平成 25 年 1 月 31 日

室蘭工業大学大学院

# 修士論文題目

内部状態と行動をベースとした  
ロボットの内部状態予測手法の提案

専攻名 情報電子工学系 専攻  
氏名 杉本 大志

## 論文概要

近い将来、日常生活の場に、人の仕事を手助けするロボットの登場が予想される。複雑な環境では、ロボットの望ましい行動は環境によって異なる為、起こり得る全ての状況を想定し、適切な行動を予めロボットにプログラムしておく事は困難である。また、ロボットは、予め決められたコマンドに応じて行動するだけでなく、人の様に環境の変化に合わせて行動を決定する事が望ましいが、それら全てを予測し、認識出来る様にしておく事も難しい。

この様な場合、ロボットが自身の現在の内部状態と行動から、次に遷移する内部状態を予測する事により、次に採るべき行動を決定する事が出来る。

そこで本研究では、『状態予測からの行動決定』に着目する。学習器であるOnline SVRを用いてロボットの為の『状態の先読み予測』の考え方を提案し、予測した状態からロボットが素早く望ましい行動を決定する方法について論じてゆく。

|                 |                   |
|-----------------|-------------------|
| 修 士 論 文 審 査 委 員 | 主 査 畑 中 雅 彦       |
|                 | 委 員 本 田 泰         |
|                 | 委 員 佐 賀 聡 人       |
|                 | 委 員 倉 重 健 太 郎     |
| 審 査 年 月 日       | 平 成 25 年 2 月 12 日 |

平成 24 年度 情報電子工学特別研究 II 博士前期課程論文

研究テーマ

内部状態と行動をベースとした  
ロボットの内部状態予測手法の提案

室蘭工業大学 大学院工学研究科 情報電子工学系専攻  
コンピュータ知能学コース 認知ロボティクス研究室

杉本 大志 (学籍番号: 11054034)

提出日付 平成 25 年 1 月 31 日  
研究期間 平成 23 年 4 月 1 日 ~ 平成 25 年 1 月 31 日

指導教官 倉重 健太郎

## 内容梗概

近い将来, 日常生活の場に, 人の仕事を手助けするロボットが登場する事が予想される. 複雑な環境では, ロボットの望ましい行動は環境によって異なる為, 起こり得る全ての状況を想定し, 適切な行動を予めロボットにプログラムしておく事は困難である. また, ロボットは, 予め決められたコマンドに応じて行動するだけでなく, 人の様に環境の変化に合わせて行動を決定する事が望ましいが, それら全てを予測し, 認識出来る様にしておく事も難しい.

この様な場合, ロボットが自身の現在の内部状態と行動から, 次に遷移する内部状態を予測する事により, 次に採るべき行動を決定する事が出来る.

そこで本研究では, 『状態予測からの行動決定』に着目する. 学習器である **Online SVR** を用いてロボットの為の 『状態の先読み予測』の考え方を提案し, 予測した状態からロボットが素早く望ましい行動を決定する方法について論じてゆく.

## Abstract

In the near future, robots are expected to provide extensive support for us in our daily lives, with the desired behavior of the robots varying according to their owners and the environment. However, in the complicated environment, because of desired behavior of the robots are different, so it is virtually impossible to predict all possible situations and to preprogram the robot with all suitable behavioral patterns for each of the possible situations. Furthermore, it is desirable that robots decide individual action of instruction adapt environment and in addition to the pre-registered commands, like a human. Nevertheless, it is also difficult to predict all the possible actions and to recognize them.

In this case, robots can predict next state from current state and action that should be selected.

In the present study focused on “**decide action using the state-prediction.**” The purposes of this research is “**state advance-prediction**” using **Online SVR** as a learner. And this thesis presents the results of these studies and discusses methods that enable a robot decides its desirable behavior quickly, using “**state that predicted**”.

# 目次

|         |                                  |    |
|---------|----------------------------------|----|
| 第 1 章   | 序論                               | 1  |
| 1.1     | はじめに                             | 1  |
| 1.2     | 機械学習                             | 1  |
| 1.2.1   | 教師付き学習                           | 1  |
| 1.2.2   | 教師なし学習                           | 2  |
| 1.2.3   | 半教師付き学習                          | 2  |
| 1.2.4   | 強化学習                             | 2  |
| 1.3     | 従来研究とその問題点                       | 3  |
| 1.4     | 本研究の目的                           | 3  |
| 1.5     | 本研究のアプローチ                        | 3  |
| 1.6     | 本論文の構成                           | 4  |
| 第 2 章   | Online Support Vector Regression | 5  |
| 2.1     | Kernel Trick                     | 5  |
| 2.2     | Support Vector Machine           | 7  |
| 2.3     | Support Vector Regression        | 7  |
| 2.4     | Online SVR                       | 9  |
| 第 3 章   | 行動を基にした状態予測の考え方                  | 11 |
| 3.1     | 逐次内部状態予測 – 1 サンプル時刻先の状態予測        | 11 |
| 3.2     | 追加学習を組合わせた逐次内部状態予測               | 14 |
| 3.3     | 長期内部状態予測 – $n$ サンプル時刻先の内部状態予測    | 15 |
| 第 4 章   | 自走式二輪型倒立振り子への応用                  | 19 |
| 4.1     | LEGO MINDSTORMS NXT              | 19 |
| 4.1.1   | NXTway-GS                        | 19 |
| 4.2     | 現代制御論                            | 20 |
| 4.2.1   | 概要                               | 20 |
| 4.2.1.1 | 可制御性                             | 21 |
| 4.2.1.2 | 可観測性                             | 21 |
| 4.2.2   | 状態フィードバック制御                      | 22 |
| 4.2.3   | 制御系の状態方程式と安定性                    | 22 |
| 4.2.4   | 安定とは                             | 23 |
| 4.2.4.1 | 安定の定義                            | 24 |
| 4.2.4.2 | 正定と準正定                           | 25 |

|         |                                  |    |
|---------|----------------------------------|----|
| 4.2.5   | 最適レギュレータ                         | 25 |
| 4.2.6   | サーボ制御                            | 26 |
| 4.3     | NXTway-GS のモデリングと状態フィードバックゲインの導出 | 27 |
| 4.3.1   | NXTway-GS のモデルと物理パラメータについて       | 27 |
| 4.3.2   | NXTway-GS のモデリング                 | 28 |
| 4.3.3   | 状態方程式の導出                         | 32 |
| 4.3.4   | コントローラ的设计                        | 35 |
| 4.3.4.1 | 入出力                              | 35 |
| 4.3.4.2 | 安定性                              | 35 |
| 4.3.4.3 | 設計                               | 35 |
| 第 5 章   | シミュレーションによる検証実験とその検討             | 37 |
| 5.1     | 実験環境                             | 37 |
| 5.2     | 逐次学習と追加学習を組合わせた逐次内部状態予測          | 38 |
| 5.2.1   | 外乱を与えない場合の訓練データ                  | 38 |
| 5.2.2   | 予測可能な外乱を与えた場合の訓練データ              | 41 |
| 5.2.3   | 突発的な外乱を与えた場合の訓練データ               | 44 |
| 5.3     | 逐次学習と追加学習を組合わせた長期内部状態予測          | 47 |
| 5.3.1   | 突発的な外乱を与えた場合の訓練データ               | 47 |
| 5.4     | 実験結果                             | 48 |
| 5.4.1   | 逐次学習と追加学習を組合わせた逐次内部状態予測          | 48 |
| 5.4.1.1 | 外乱を与えない場合の訓練データに対する逐次内部状態予測      | 48 |
| 5.4.1.2 | 予測可能な外乱を与えた場合の訓練データに対する逐次内部状態予測  | 50 |
| 5.4.1.3 | 突発的な外乱を与えた場合の訓練データに対する逐次内部状態予測   | 53 |
| 5.4.2   | 逐次学習と追加学習を組合わせた長期内部状態予測          | 55 |
| 5.5     | 実験結果に対する検討                       | 58 |
| 5.5.1   | 逐次学習と追加学習を組合わせた逐次内部状態予測          | 58 |
| 5.5.2   | 逐次学習と追加学習を組合わせた長期内部状態予測          | 58 |
| 5.6     | まとめ                              | 58 |
| 第 6 章   | 結論                               | 59 |
| 6.1     | 成果のまとめ                           | 59 |
| 6.2     | 今後の展望                            | 59 |
| 参考文献    |                                  | 63 |
| 研究業績    |                                  | 67 |

# 目次

|      |  |    |
|------|--|----|
| 2.1  | Kernel Trick と SVR の組み合わせによる, 非線型関数の回帰 . . . . .                       | 5  |
| 2.2  | SVR の概要 . . . . .  | 8  |
| 2.3  | Parrella が提案した手法の概要 . . . . .  | 9  |
| 3.1  | Parrella が提案した手法の概要 (図 2.3 の再掲) . . . . .                              | 11 |
| 3.2  | 内部状態と行動から未来を予測する考え方 . . . . .  | 12 |
| 3.3  | 追加学習を組合わせた逐次内部状態予測 . . . . .   | 15 |
| 3.4  | 長期内部状態予測の為のブロック図 . . . . .   | 17 |
| 3.5  | 逐次内部状態予測部のブロック図 . . . . .  | 17 |
| 4.1  | 自走式二輪型倒立振子 NXTway-GS の概観 . . . . .                                     | 19 |
| 4.2  | 典型的な状態空間モデル . . . . .  | 21 |
| 4.3  | 可制御と可観測な系 . . . . .  | 21 |
| 4.4  | 状態フィードバックによる制御系の例 . . . . .  | 22 |
| 4.5  | 閉ループ極の望ましい位置 . . . . .   | 23 |
| 4.6  | リアプノフの意味での安定 . . . . .   | 24 |
| 4.7  | 漸近安定 . . . . .   | 25 |
| 4.8  | 自走式二輪型倒立振子のモデル . . . . .   | 27 |
| 4.9  | 自走式二輪型倒立振子の側面・平面図 . . . . .  | 28 |
| 4.10 | NXTway-GS の入出力 . . . . .   | 35 |
| 4.11 | NXTway-GS 用サーボ制御器のブロック線図 . . . . .                                     | 35 |
| 4.12 | 本研究で提案する長期内部状態予測の構成図 (図 3.4 の再掲) . . . . .                             | 36 |
| 5.1  | 内部状態 $x_1$ (駆動輪回転角度 $\theta$ ) の訓練データ (外乱を与えない場合) . . . . .            | 38 |
| 5.2  | 内部状態 $x_2$ (車体傾斜角度 $\psi$ ) の訓練データ (外乱を与えない場合) . . . . .               | 39 |
| 5.3  | 内部状態 $x_3$ (駆動輪回転角速度 $\dot{\theta}$ ) の訓練データ (外乱を与えない場合) . . . . .     | 39 |
| 5.4  | 内部状態 $x_4$ (車体傾斜角速度 $\dot{\psi}$ ) の訓練データ (外乱を与えない場合) . . . . .        | 40 |
| 5.5  | 行動 $a$ (制御入力) の訓練データ (外乱を与えない場合) . . . . .                             | 40 |
| 5.6  | 内部状態 $x_1$ (駆動輪回転角度 $\theta$ ) の訓練データ (予測可能な外乱を与えた場合) . . . . .        | 41 |
| 5.7  | 内部状態 $x_2$ (車体傾斜角度 $\psi$ ) の訓練データ (予測可能な外乱を与えた場合) . . . . .           | 42 |
| 5.8  | 内部状態 $x_3$ (駆動輪回転角速度 $\dot{\theta}$ ) の訓練データ (予測可能な外乱を与えた場合) . . . . . | 42 |
| 5.9  | 内部状態 $x_4$ (車体傾斜角速度 $\dot{\psi}$ ) の訓練データ (予測可能な外乱を与えた場合) . . . . .    | 43 |
| 5.10 | 行動 $a$ (制御入力) の訓練データ (予測可能な外乱を与えた場合) . . . . .                         | 43 |
| 5.11 | 内部状態 $x_1$ (駆動輪回転角度 $\theta$ ) の訓練データ (突発的な外乱を与えた場合) . . . . .         | 44 |
| 5.12 | 内部状態 $x_2$ (車体傾斜角度 $\psi$ ) の訓練データ (突発的な外乱を与えた場合) . . . . .            | 45 |



|      |   |    |
|------|---|----|
| 5.13 | 内部状態 $x_3$ (駆動輪回転角速度 $\dot{\theta}$ ) の訓練データ (突発的な外乱を与えた場合) . . . . . | 45 |
| 5.14 | 内部状態 $x_4$ (車体傾斜角速度 $\dot{\psi}$ ) の訓練データ (突発的な外乱を与えた場合) . . . . .    | 46 |
| 5.15 | 行動 $a$ (制御入力) の訓練データ (突発的な外乱を与えた場合) . . . . .                         | 46 |
| 5.16 | 内部状態 $x_1$ に対する逐次予測の結果 (外乱を与えない場合) . . . . .                          | 48 |
| 5.17 | 内部状態 $x_2$ に対する逐次予測の結果 (外乱を与えない場合) . . . . .                          | 49 |
| 5.18 | 内部状態 $x_3$ に対する逐次予測の結果 (外乱を与えない場合) . . . . .                          | 49 |
| 5.19 | 内部状態 $x_4$ に対する逐次予測の結果 (外乱を与えない場合) . . . . .                          | 50 |
| 5.20 | 内部状態 $x_1$ に対する逐次予測の結果 (予測可能な外乱を与えた場合) . . . . .                      | 51 |
| 5.21 | 内部状態 $x_2$ に対する逐次予測の結果 (予測可能な外乱を与えた場合) . . . . .                      | 51 |
| 5.22 | 内部状態 $x_3$ に対する逐次予測の結果 (予測可能な外乱を与えた場合) . . . . .                      | 52 |
| 5.23 | 内部状態 $x_4$ に対する逐次予測の結果 (予測可能な外乱を与えた場合) . . . . .                      | 52 |
| 5.24 | 内部状態 $x_1$ に対する逐次予測の結果 (突発的な外乱を与えた場合) . . . . .                       | 53 |
| 5.25 | 内部状態 $x_2$ に対する逐次予測の結果 (突発的な外乱を与えた場合) . . . . .                       | 54 |
| 5.26 | 内部状態 $x_3$ に対する逐次予測の結果 (突発的な外乱を与えた場合) . . . . .                       | 54 |
| 5.27 | 内部状態 $x_4$ に対する逐次予測の結果 (突発的な外乱を与えた場合) . . . . .                       | 55 |
| 5.28 | 内部状態 $x_1$ に対する状態予測の結果 . . . . .                                      | 56 |
| 5.29 | 内部状態 $x_2$ に対する状態予測の結果 . . . . .                                      | 56 |
| 5.30 | 内部状態 $x_3$ に対する状態予測の結果 . . . . .                                      | 57 |
| 5.31 | 内部状態 $x_4$ に対する状態予測の結果 . . . . .                                      | 57 |

# 表目次

|     |                              |    |
|-----|------------------------------|----|
| 4.1 | センサの特性 . . . . .             | 20 |
| 4.2 | アクチュエータの特性 . . . . .         | 20 |
| 4.3 | NXTway-GS の物理パラメータ . . . . . | 29 |
| 5.1 | シミュレーション条件 . . . . .         | 37 |

---

# 第 1 章 序論

本章では、本研究の背景と目的、内容、および本論文の構成について述べる。

## 1.1 はじめに

動的な環境下で適切に動作するロボットの研究は、古くから数多く行われている [1, 2, 3]。例として家庭、工場等とあらゆる作業空間内で人間の代わりに活躍する、様々なロボットの開発が行われている [4]。しかしながら、動的な環境下に於いてロボットを人間の様に行動させる事は容易ではない [5]。我々人間は或る環境下で作業をする際、周囲の環境の変化や、自身の次の状態を予測し、それに合わせた行動を無意識の内に選択している。具体的には、過去の経験と記憶から作業をスムーズに行う為に、自分自身の姿勢や力の入れ具合を予測して、無意識の内に制御をしながら操作を行っている [6]。これは、あまりに多くの事を意識して行動していると、即座の対応が難しくなり、目的の行動を達成できず、『損をする』場合がある。例えば人間の歩行は、環境及び身体に関する感覚入力に応じて適切な調節がなされる結果、常にリズムカルで安定した動作となる。この時『脳』が意識すべきは歩行と環境であり、その環境に合わせて『個々の関節をどの程度変化させるのか』は無意識で行われる。この予測によって、普段の社会生活に於いて人間はボディバランスを制御し、転倒や接触等のリスクを低減している [6]。

同様にロボットの場合は、予測を行わない場合、行動選択に対する制御処理の負荷が大きくなる事が考えられる。この為、人間の作業の代行として、ロボットが動的な環境下にて或る作業を実現する場合に於いても、人間同様に『状態予測からの行動決定』が重要である。これに関連した研究のアプローチとして、ロボットに対し機械学習を適用する事で、環境に適した行動を自律的に学習させるものが在る。

## 1.2 機械学習

人間は過去に行なった問題とよく似た問題を解決する時、以前に解いた経験を基に、より効率的に解決出来る学習能力を持っている。機械学習 (machine learning) [7] とは、言語やゲームなどをはじめとした人間の様々な知的活動の中で、人間が自然に行っているパターン認識や経験則を導き出したりする様な活動を、計算機を用いて実現する為の技術や理論、またはソフトウェアなどの総称である。一般に機械学習は、教師付き学習、教師なし学習、半教師付き学習、強化学習の 4 種類に大別される。

### 1.2.1 教師付き学習

人間がテニスのサーブを練習する時、この様に打てば右に飛ぶ、左に飛ぶ、強く飛ぶと云った、一般に模範となるものがある。この時、練習者はこの模範を見る事で、次第にテニスのサーブが上手になる。

教師付き学習 (supervised learning) は、入力 (質問) と出力 (答え) の組からなる訓練データを用いて、そ

の背後に潜んでいる入出力関係 (関数) を学習する問題である。ひとたび関数を巧く学習する事が出来れば、学習していない入力に対する出力を予測する事が出来る様になる。即ち学習器は未知の状況に汎化出来、且つ可能な限り少ない訓練データから最高の汎化能力を獲得する事が、教師付き学習の研究の目標となる。教師付き学習問題は、標本から関数を近似する問題に定式化出来る。

教師付き学習に於いて高い汎化能力を獲得する為には、学習器 (モデル) の複雑さを適切に決定する事が重要である。もしモデルが単純過ぎれば、学習したい真の関数を表現する事が出来ず、豊富な訓練データを用いても汎化能力は向上しない。一方モデルが複雑な場合、学習したい真の関数を表現する事は出来るが、訓練データに含まれるノイズの影響を強く受ける為、比較的少数の訓練データしか得られない現実的な場面に於いては良い汎化性能は得られない。モデル選択は、教師付き学習に於ける最も基礎的かつ中心的な研究課題である。一般にモデル選択は、汎化能力を最大にするモデルを見つける事により行なわれる。従って、モデル選択に於いて、如何に精度の良い汎化能力推定法を構成するかと言う事になる。

本研究に於いては、『状態予測からの行動決定』を行う過程にて、この教師付き学習の考え方をを用いている。

### 1.2.2 教師なし学習

教師なし学習 (unsupervised learning) は、文字通り教師がいない状況での学習であり、出力 (答え) が無く入力データのみが与えられる。教師なし学習の目的は状況によって異なり、数学的にはっきりと定式化出来ない場合が多い。例えば、入力データの似たもの同士をグループ化するクラスタリングがその典型的な例である。クラスタリングではデータの類似度をどう定義するかが重要であるが、どの類似度が用いるのが適切かを定量的に評価する基準はなく、主観的に決めることが多い。

この教師なし学習は自己組織化 (self-organization) と呼ばれる事もある。教師なし学習には、主成分分析・独立成分分析 [8] 等の様に信号を特徴的な成分に分解するもの、混合正規分布モデル [9]、自己組織化マップ [10] 等の様に離散化しクラス分けするものの2つがある。

### 1.2.3 半教師付き学習

教師付き学習では入力と出力の組からなる訓練データが与えられ、教師なし学習では入力だけの訓練データが与えられる。半教師付き学習 (semi-supervised learning) は、これらの中間の状況に対応する学習問題であり、入力と出力の組からなる訓練データに加え入力だけの訓練データも与えられる。半教師付き学習の目標は、教師付き学習と同じく高い汎化能力を獲得する事に在る。

半教師付き学習では、入出力両方が揃っている訓練データの数は少なく、入力だけの訓練データの数は非常に多い場合を考える事が典型的である。このような状況では、少数の入出力データだけでなく多数の入力データも用いる事により、より高い汎化能力が獲得出来るものと期待される。

### 1.2.4 強化学習

強化学習 (reinforcement learning) とは、あるエージェントの行動を司る政策関数、即ち状態から行動への変換関数を学習する問題である。政策関数と云う入出力関係を学習する問題であるから、強化学習の目的は教師付き学習と同じであるが、教師付き学習と異なり、出力データを直接観測する事が出来ない。即ち、教師なしの状況で政策関数を学習しなければならない。但し、教師なし学習と異なり、エージェントの行動に対して報酬が与えられる。この報酬の情報を活用して、将来的に得られる報酬の和が最大になるように政策関数を学習する事が強化学習の目的となる。

典型的な強化学習法では、政策関数を直接学習するのではなく、価値関数と呼ばれる、状態と行動から将

来の報酬の和への関数を学習し、学習した価値関数から政策関数を生成する。

### 1.3 従来研究とその問題点

従来『予測』を取り扱う研究としては、株価に対して時系列予測を適用した株価予測の研究<sup>[11]</sup>や、飛行ロボットやロボットアームをプラントと見做した予測制御<sup>[12, 13]</sup>、複数の状態予測モデルと報酬予測モデルの組合せによって出力が決定されるモジュール強化学習方式<sup>[14]</sup>が挙げられる。

また、“Online SVR”と呼ばれる、イタリアの Parrella が提案したバッチ SVR を発展させた手法も存在する<sup>[15]</sup>。これは従来使われてきたバッチ SVR<sup>[16, 17]</sup>が持っていた問題点である、訓練データを全て揃えなければ学習出来ない(バッチ学習)、と言う点をオンライン学習により解決したものである。

但し、ここで挙げた先行研究の内、文献[11]は純粋に株価を時系列データとして捉え、それが未来でどのような状態に変化するかを予測している。また、文献[12, 13]については、制御量と操作量について着目し、これらの状態の変化を予測し、フィードフォワード制御として予測結果を用いてプラントの制御を行っている。そして、文献[14]では、変化する環境や報酬に合わせて複数の状態と報酬の予測モデルを適宜切り替えながら学習制御を行っている。最後に文献[15]では、時系列データとして与えた一次元データを逐次与える事で、そのデータに対する回帰分析を行っている。つまり、これらの先行研究では、予測を間接的に用いた制御を行うか、純粋に予測そのものについて検討が為されているが、何れも状態に対する予測に焦点を当てたものではない。従って、従来の手法によって『状態予測からの行動決定』を実現する事は出来ない。

その理由として、ロボットは現在の環境の状態を感知し、一つの行動を実行する。状態と行動によって環境は新しい状態に遷移し、それに対して報酬をロボットに渡す。これらの相互作用を通して、ロボットは与えられたタスクを遂行する目的行動を学習する<sup>[18]</sup>。この為、学習で得られた経験を利用して行動するロボットの行動は、決定論に加えて、ロボットの行動が環境に作用する因果律にも影響される。例えば、自動車の動きそのものは物理法則に従うが、交差点で直進するか曲がるかを決めるのは運転手の都合だ。つまり運転手が選んだ行動が、次の状態及び環境に影響を与える事になる。この様にロボットは、状況に応じて複数の選択肢から一つを選択する。

以上の様に、ロボットの場合、自身が選択した行動によって、未来に決定される状態が無数に存在する為、未来の状態を予測する事が難しい。従って、ロボットの採った行動が未来に影響を与える事になると言う、ロボット特有の問題が存在する。その為、従来からの手法として注目されてきた時系列予測<sup>[19]</sup>の様に、単純な時系列データの予測では、その精度と考え方に限界が生じる事となる。

この様な理由から、ロボットの為の『状態予測』の考え方が必要となる。

### 1.4 本研究の目的

そこで本研究では、現在の内部状態と行動から、未来の内部状態を予測する手法を提案する。これによって、自身の採る行動によって変化する内部状態を予測する事が可能となる。

また適用対象として、自走式二輪型倒立振子を用いる。訓練データとして、自走式二輪型倒立振子に対して倒立姿勢安定化制御を継続し実施した際の内部状態と制御入力を用い、これに提案手法を適用する。この検証結果から、提案手法の有用性を確認する。

### 1.5 本研究のアプローチ

この内部状態の予測を実現する上で、本研究では文献[18]で述べられている様に、状態(内部状態)と行動によって新たな状態に遷移する点を踏まえ、ロボットの現在の内部状態と採る行動から、次にロボットが

遷移する内部状態を予測する。また、未来の内部状態は、現在の内部状態と行動の他にも、過去に経験した行動や内部状態にも依存するものと考えられる。そこで、内部状態の予測を行う為に、Online SVR を『学習器』として採用する。この“Online SVR”に対して現状態と行動を訓練セットとして与える事で、未来の内部状態の予測を実現する。

### 1.6 本論文の構成

本論文は 6 章から構成されている。各章の概要を以下に述べる。

- 第 1 章 研究の背景と内容、および目的を述べ、本研究に於ける提案手法を対象に適用し、その有効性を確認する事を本研究の目標とした。
- 第 2 章 本研究に於いて主に扱う“Online SVR”について解説する。併せて、その基礎となるバッチ SVR, Kernel Trick についても概説する。
- 第 3 章 本研究で扱う『状態予測』の考え方と、Online SVR の改良について解説する。また、状態フィードバックを用いた長期内部状態予測についての考え方について説明する。
- 第 4 章 本研究にて適用対象としている自走式二輪型倒立振子“NXTway-GS”について解説する。また、この NXTway-GS に対する状態予測についての適用方法について説明する。
- 第 5 章 シミュレーションによる検証実験について述べ、その結果について議論する。
- 第 6 章 本研究で得られた結論を述べる。

## 第 2 章 Online Support Vector Regression

本研究では、行動を基とした『状態予測』の実現に“**Online SVR**”を用いる。本章では、本研究を遂行する上で重要な手法となる“**Online SVR**”について解説する。

尚、**Online SVR** に関する文献が少ない事から、本理論の理解を助ける為に、**SVM/SVR** の概要、及び **Kernel Trick** についても併せて本章にて解説する。これは本手法に於いて、重要な基礎理論となっている。そこで各者の差違、及びこれらが持つ問題点についても同時に解説する。

### 2.1 Kernel Trick

本来、**Online** やバッチ方式に関わらず、**SVR** はそれ単体であると、獲得できるモデルは線型な特徴を持つ訓練データに限られる。しかし、**Kernel Trick** を回帰作業の前に訓練データに適用する事により、疑似的に非線型な特徴を持つ訓練データに対しても回帰モデルを獲得する事が可能となる。これを端的に示す概念図が図 2.1 である。

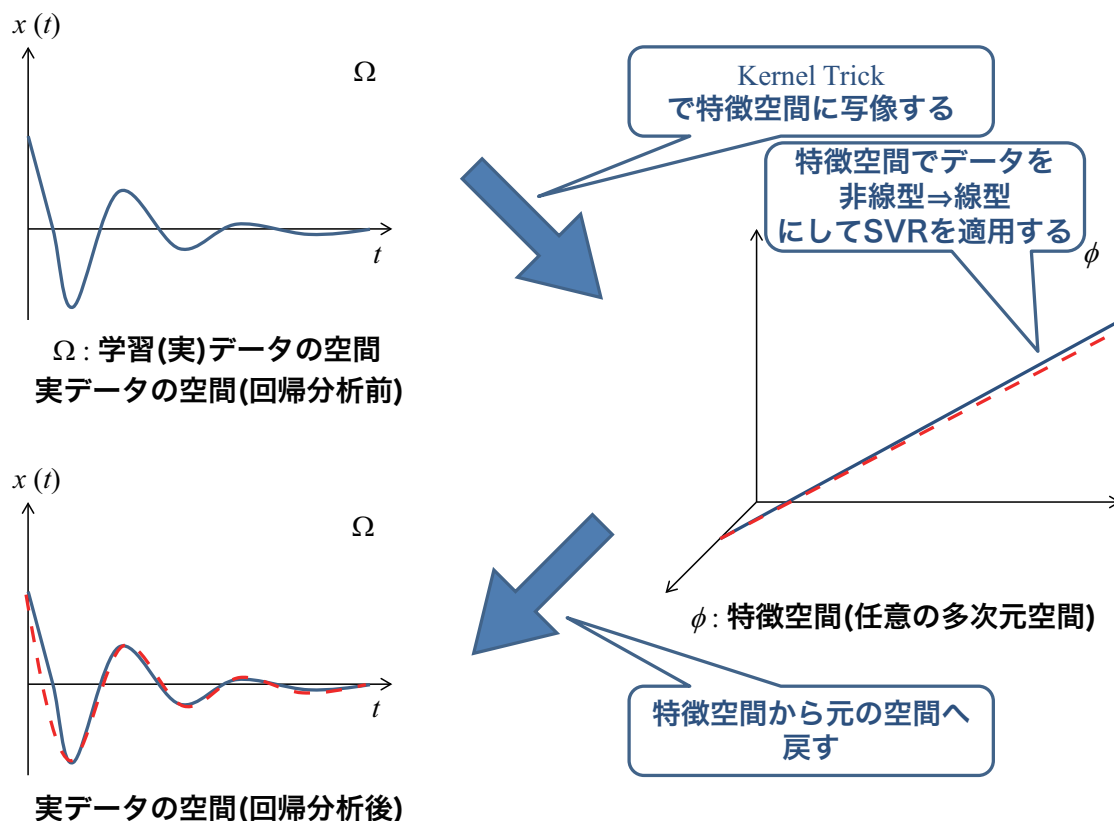


図 2.1 Kernel Trick と SVR の組み合わせによる、非線型関数の回帰

線型空間  $\Omega$  では線型回帰が出来ない訓練データを高次元の特徴空間  $\phi$  に写像し、 $\phi$  上で SVR により線型回帰を行う。その後、 $\phi$  から  $\Omega$  へ空間を戻す事によって、元の線型空間上に於ける回帰が可能となる。この結果を用いて、非線型な特徴を持った新たな入力に対する予測が可能となる。

この様に適切な特徴空間を設定する時、一般に特徴空間の次元  $h$  を、訓練データの次元  $n$  より大きく定義する事で、非線型な訓練データのサンプルは線形分離可能になる。この考え方をを用いる事で、特徴空間上で SVR を適用すると、この空間で表現された回帰平面から回帰関数を学習し、計算により予測する事も可能となるのである。より具体的には、非線型な訓練データを Ternel Trick により特徴空間に線型データとして写像し、その疑似的な線型なデータに対して回帰計算と予測計算を行う事によって、非線型な訓練データに対する回帰関数の導出と予測データの導出が可能となる。

ソフトマージン法<sup>[16]</sup>を用いる事で、線型分離可能ではない場合に対して、線型しきい素子のパラメータを求める事が出来る様になる。しかし、ソフトマージン法を用いたとしても、本質的に非線型で複雑な識別課題に対しては、必ずしも良い性能の識別器を構成出来るとは限らない。本質的に非線型な問題に対応する為の方法として、特徴ベクトルを非線型変換して、その空間で線型の識別を行う **Kernel Trick**<sup>[20]</sup> と呼ばれている方法が知られている。この方法を用いる事で、SVM/SVR や Neural Network<sup>[21]</sup> 等の学習器性能が飛躍的に向上する。

一般に、線型分離可能性はサンプル数が大きくなればなる程難しくなり、逆に、特徴空間ベクトルの次元が大きくなる程易しくなる。例えば、特徴ベクトルの次元が訓練サンプルの数よりも大きいなら、どの様なラベル付けに対しても線型分離可能である。しかし、高次元への写像を行うと、次元の増加に伴い汎化能力が低下する。また、難しい問題を線型分離可能にする為には、訓練サンプルと同程度の大きな次元に写像しなければならないので、結果的に膨大な計算量が必要となる。そこで注目されている手法が、この **Kernel Trick** である。

内積が定義されている空間で写像  $\Phi(\mathbf{x})R^p \rightarrow R^r (p \ll r)$  を考える。 $R^p$  を入力空間、 $R^r$  を特徴空間と呼ぶ。ここで、 $\mathbf{x} = [x_1 \ \dots \ x_p] \in R^p$  に対して、 $\phi_i(\mathbf{x})(i = 1, \dots, r)$  を用いて、 $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \ \phi_2(\mathbf{x}) \ \dots \ \phi_r(\mathbf{x})]^T$  と定義する。或る関数  $K : R^r \times R^r \rightarrow R$  について、

$$K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = \sum_{i=1}^r \phi_i(\mathbf{x})\phi_i(\mathbf{y}) \quad (2.1.1)$$

を満たす様な  $\Phi$  が存在する時、この  $K$  を Kernel 関数と呼ぶ。ここで、 $\langle \cdot, \cdot \rangle$  は内積を表す。

$\Phi$  は  $K$  には陽に表れないので、 $\Phi(\mathbf{x})$  の内積形さえ定義されていれば、 $\phi_i(\mathbf{x})$  を計算する必要は無い。即ち、 $r$  がどの様に大きくなろうとも、その点は  $K$  に直接影響を及ぼさない為、低次元空間に於ける計算で高次元空間のデータを扱う事が可能となる。この様に、低次元空間で高次元空間のデータを扱う事を **Kernel Trick**(カーネル法) と呼ぶ。

$\Phi(\mathbf{x})$  の内積を定義するには、Mercer の定理が用いられる<sup>[22]</sup>。

定理 (Mercer の定理)  $\chi$  をベクトル空間とする。 $x, y \in \chi$  の関数  $K$  が内積の形

$$K(x, y) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x)\phi_i(y), \quad \gamma_i \geq 0 \quad (2.1.2)$$

と定義する条件は、 $K$  が対称関数であり、且つ、半正定値を満たす事である。

ここで対称関数とは、任意の  $x, y$  に対し、 $K(x, y) = K(y, x)$  を満たす関数であり、半正定値とは、任意の  $f$  に対し、

$$\int \int_{\chi} K(x, y)f(x)f(y)dxdy \geq 0 \quad (2.1.3)$$



が成り立つ事である。但し、 $\phi(x)$  は  $K(x, y)$  の固有関数で、

$$\int_{\mathcal{X}} K(x, y)\phi_i(x) = \gamma_i\phi_i(x) \quad (2.1.4)$$

である。この定理より、或る Kernel 関数  $K$  が与えられると、その値は、特徴空間に於ける内積の値となる。

Kernel 関数の主要な例としては、以下の 5 種類が挙げられる。

$$\text{Polynomial Kernel} \quad K_p(x, y) = (1 + x^T y)^p \quad (2.1.5)$$

$$\text{Gaussian Kernel} \quad K_g(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\delta^2}\right) \quad (2.1.6)$$

$$\text{Radial Basis Function Kernel; RBF} \quad K_r(x, y) = \exp\left(-\frac{\|x - y\|^2}{\delta^2}\right) \quad (2.1.7)$$

$$\text{Sigmoid Kernel} \quad K_s(x, y) = \frac{1}{1 + \exp(-\beta xy)} \quad (2.1.8)$$

$$\text{Perceptron Kernel} \quad K_m(x, y) = \tanh(axy - \theta) \quad (2.1.9)$$

特に本研究では、式 (2.1.7) で示される非常に汎用的な類似度関数であり、ロボット制御に関わらず多くの場合で良好な結果を与える **RBF Kernel** を用いる事とする [23]。

## 2.2 Support Vector Machine

パターン認識手法の一つである **Support Vector Machine(SVM)** と呼ばれる手法が在る [16]。SVM は **Neural Network** 等それまでの手法と比べ汎化性能が高いと知られており、最近注目を集めている。サポートベクトルマシンとは、2 クラスのパターン認識手法であり、訓練データを用い最適な識別境界を決定する技術である。与えられたデータ集合が線形分離可能な場合、不可能な場合両方に適応可能である。線形分離が可能な場合は、識別関数が線形関数、線形分離不可能な場合は、識別関数が非線形関数 (Kernel 関数) となる。その線形分離不可能な場合に識別関数に Kernel 関数を適用する事で、識別関数の非線形性を実現させ、適切な分類結果を得る事が出来る。

最適な識別境界とは、マージンを最大にする事である。マージン (Margin) とは、識別境界とクラスに所属する最も近い個体 (これは、訓練データの中でサポートベクトルと呼ばれる) との距離を指す。マージンを最大にするという事が、高い汎化性能を発揮する事に繋がり、マージン最適化が二次計画問題に帰着する為、局所最適解に陥る事が無く、唯一つの解が導かれる。

## 2.3 Support Vector Regression

**Support Vector Regression(SVR)** は、機械学習の判別問題に於いて注目されている SVM の回帰問題への拡張である。PAC 学習 [24] の枠組みを用い汎化誤差の上限を最小化する様に学習を行う為、高い汎化性能を持つものとされている。

SVR は、 $\pm\epsilon$  の誤差の範囲内 ( $\epsilon$ -tube) で出来るだけ全ての訓練データを取める様な回帰直線  $f(\mathbf{x}) = \mathbf{a}\mathbf{x} + b$  を求める (図 2.2)。遠く外れてしまう様な点は、パターン認識と同様に緩和変数を加味する事によって制限する。許容誤差は、 $\epsilon$  に依存している。つまり、この許容誤差  $\epsilon$  により外れ値をカットする事が出来る様になり、観測誤差またはノイズの影響を低減する事が可能となる。これを利用する事により、ロバストに制御器またはモデルのパラメータを推定出来る。

次の回帰直線

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w} + b \quad (2.3.1)$$

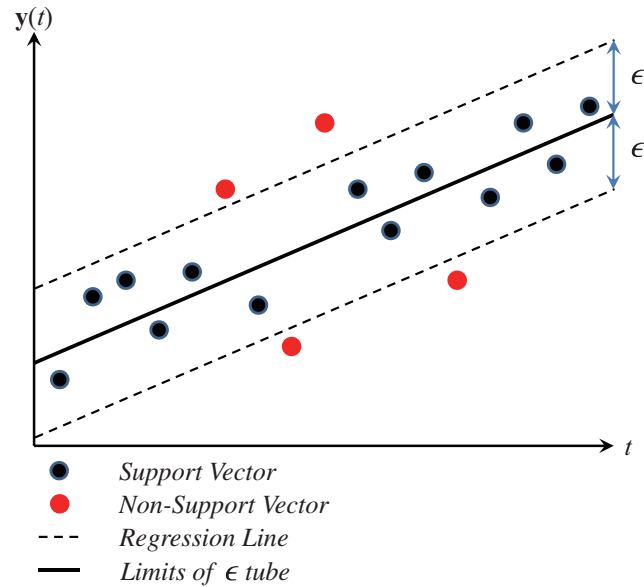


図 2.2 SVR の概要

と回帰直線とサンプルの残差を  $r$  として、つぎの  $\epsilon$  許容誤差 ( $\epsilon$ -insensitive error) を考える.

$$\xi(\mathbf{r}) = \begin{cases} 0 & \text{if } |\mathbf{r}| < \epsilon \\ |\mathbf{r}| - \epsilon & \text{otherwise} \end{cases} \quad (2.3.2)$$

このとき, サンプル  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$  について, 次の最適化問題を考える.

$$\min_{\mathbf{w}, b} \sum_{i=1}^N \xi(y_i - f(\mathbf{x}_i)) + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (2.3.3)$$

但し,  $\lambda$  は正則化パラメータである (ラグランジュ未定乗数と同一である).

この双対問題は, 次の二次計画問題に書き換えられる.

$$\begin{aligned} \min_{\alpha_i, \alpha_i^*} & \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) \\ & + \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) \mathbf{x}_i^T \mathbf{x}_j \end{aligned} \quad (2.3.4)$$

$$\text{subject to: } \begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq \frac{1}{\lambda} \\ \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \end{cases}$$

この式 (2.3.3), (2.3.4) が SVR と呼ばれる. 誤差関数には  $\epsilon$  許容誤差以外に Huber 関数<sup>[25]</sup> なども用いられる. サンプルの内積のみを用いるので, 非線型 SVM 同様 **Kernel Trick** が利用可能であり, 非線形回帰分析も可能となる. Kernel Trick を適用した場合は

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \quad (2.3.5)$$

となり, 直接的に煩雑な内積計算を行う事無く内積の値を Kernel 関数より導く事が可能となる. これによって, 内積計算の省略のみならず, 非線型写像  $\phi(\mathbf{x})$  の具体的な形態を知る必要も無くなる.

## 2.4 Online SVR

最近は、識別問題への適用にとどまらず、ロボットの制御、認識に SVM や SVR(以下、バッチ SVR と呼ぶ) を適用した研究が行われている。小林らは、バッチ SVR を用いて、倒立振子の逆システムを学習させた [26]。この研究では、予め状態フィードバック制御で得られた入出力データを用いて学習し、これにより得た SVR モデルを姿勢制御に適用する事で、SVR が初期値に対してロバストであり、ロボットの学習制御への応用に有効である事を、シミュレーションによって示している。

ただ、このバッチ SVR の欠点の一つとして、一括学習でのみ動作するという事が挙げられる。これらの手法では、全ての訓練データを用いて学習し、新たな訓練データが追加された場合は、既に存在する訓練データに加えて再学習する事が必要となる。この問題については様々な解決策が提案されている。その問題点を解決するアプローチとして、各種の動作データをリアルタイム学習させ、学習で得た結果から一括予測を行い、ロボットを制御する研究が行われている [12, 13]。この点から、リアルタイム動作を必要とするロボット制御にも有用である事が示されている。その一方で、逐次的な予測の手法が示されておらず、予測データをリアルタイムに活用する様な制御への応用が難しい。

その中でも Parrella は、逐次学習の手法を SVR に適用した **Online SVR** [15] を提案している。このアルゴリズムを適用する事により、学習器が最初から再学習をする事なく訓練データの追加 (追加学習) と削除 (忘却) が出来るとしている。

本アルゴリズムでは、開始時に回帰直線  $f$  と Margin 関数  $h$  を計算する。

$$f(\mathbf{x}_c) = \sum_{i=1}^l (\theta_i Q_{ic}) + b \quad (2.4.1)$$

$$h(\mathbf{x}_c) = f(\mathbf{x}_c) - y_c \quad (2.4.2)$$

ここで、添え字  $c$  は回帰に關与するデータであるサポートベクトルを意味している。

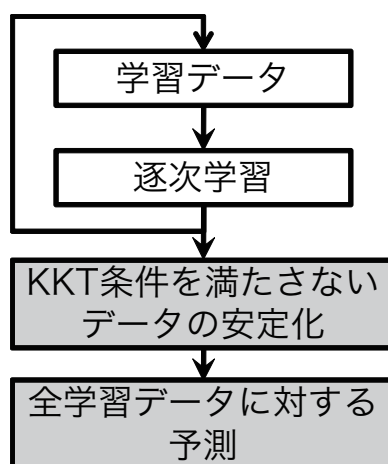


図 2.3 Parrella が提案した手法の概要

その後、サンプルが追加される度に、回帰に必要なパラメータを以下の様に更新し続ける。

$$\beta = \begin{bmatrix} \beta_b \\ \beta_{s_1} \\ \vdots \\ \beta_{s_{l_s}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & Q_{s_1 s_1} & \cdots & Q_{s_1 s_{l_s}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Q_{s_{l_s} s_1} & \cdots & Q_{s_{l_s} s_{l_s}} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ Q_{s_1 c} \\ \vdots \\ Q_{s_{l_s} c} \end{bmatrix} = -\mathbf{R} \cdot \begin{bmatrix} 1 \\ Q_{s_1 c} \\ \vdots \\ Q_{s_{l_s} c} \end{bmatrix} \quad (2.4.3)$$

$$\gamma = \begin{bmatrix} \Delta Q_{n_1 c} \\ \vdots \\ \Delta Q_{n_{l_n} c} \end{bmatrix} + \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & Q_{n_1 s_1} & \cdots & Q_{n_1 s_{l_s}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Q_{n_{l_n} s_1} & \cdots & Q_{n_{l_n} s_{l_s}} \end{bmatrix} \beta \quad (2.4.4)$$

$$\theta_c = \theta_c + \Delta\theta_c \quad (2.4.5)$$

$$\theta_i = \theta_i + \Delta\theta_i \quad (2.4.6)$$

$$b = b + \Delta b \quad (2.4.7)$$

$$\begin{bmatrix} \Delta b \\ \Delta\theta_{s_1} \\ \vdots \\ \Delta\theta_{s_{l_s}} \end{bmatrix} = \beta \Delta\theta_c \quad (2.4.8)$$

$$h(\mathbf{x}_i) = h(\mathbf{x}_i) + \Delta h(\mathbf{x}_i) \quad (2.4.9)$$

$$\begin{bmatrix} \Delta h(\mathbf{x}_{n_1}) \\ \vdots \\ \Delta h(\mathbf{x}_{n_{l_n}}) \end{bmatrix} = \gamma \Delta\theta_c \quad (2.4.10)$$

ここで、 $Q_{ic}$  は Kernel 関数により構築される Kernel 行列であり、 $\theta$  はラグランジュ乗数を示している。

しかし Parrella は、自身の手法によると、回帰モデル  $y = \mathbf{w}^T \mathbf{x} + b$  に含まれる重み係数ベクトル  $\mathbf{w}$  を計算する際、浮動小数点演算を行う事で数値的不安定性が生じる可能性があるとしている。これによる演算誤差により、計算結果に不適切な数値を出力する事があると指摘している。この問題については、KKT 条件 [27] を満たさなかった訓練データを忘却し、これまでに追加された訓練データ系列の最後尾に、その訓練データ追加し改めて学習させるという事で対処している [15] (図 2.3, またこの操作を “Stabilization<sup>[15]</sup>”, 安定化と称している)。この為、安定化を行わず逐次学習を実行した後予測した場合、各訓練データの値から逸脱した結果を示す可能性が考えられる。

また、この手法では、再学習を避ける形で訓練データを逐次的に与える事は可能だが、次時刻における出力の予測を、逐次的に与える方法については述べられていない。この為、ロボット制御などに代表されるリアルタイム制御へ応用するには、あらかじめ入出力データを与えモデルを学習させるのではなく、訓練データが時間とともに変化する事を考慮し、データを逐次与えて学習させると同時に、次時刻の出力を予測する手法を与える必要がある。

この問題に対し我々は、Online SVR 上でリアルタイムに逐次予測する手法を提案し、その有用性を確認している [28]。次章では、我々が先に提案している逐次予測を拡張し、『状態の先読み予測』を実現する為の定式化を考えてゆく。

## 第3章 行動を基にした状態予測の考え方

我々は、文献 [28] に於いて、Online SVR 上でリアルタイムに逐次予測する手法を提案している。しかし、当文献では、予測の定義に於いて、状態及び行動と言う概念を取り入れていない。この為、『状態と行動によって環境は新しい状態に遷移する』と言う考え方を基に、次状態を、現在の行動と内部状態から予測する様に再定義する必要がある。

本章では、前章で示した理論を基礎とし、その問題を解決する事で『状態予測』を実現する事を考えてゆく。先ず始めに、1 サンプル時刻先に於ける内部状態を予測する逐次内部状態予測を説明、定義する。更にその考えを拡張する事で、 $n$  サンプル時刻先に於ける内部状態を予測する長期内部状態予測を定義、説明する。

### 3.1 逐次内部状態予測 – 1 サンプル時刻先の状態予測

Online SVR の基本的なアイディアは、バッチ SVR が持つ欠点である再学習を回避する為に考え出されたものであり、逐次学習の考え方を導入したものである。本手法により、最初の学習データから訓練を再開する事なく、学習器内の要素を追加したり削除する事である。つまり、図 2.3 に示した様に、訓練の際に学習データを逐次的に与える事が出来る。一方で回帰関数を予測する際は、学習データが与えられた範囲のみを対象とし、更に訓練が終了してから予測が行われる。この為、Online SVR では、逐次学習は可能だが、逐次予測は実現されていない。

この問題に対し我々は文献 [28] に於いて、Online SVR 上でリアルタイムに逐次予測する手法を提案して

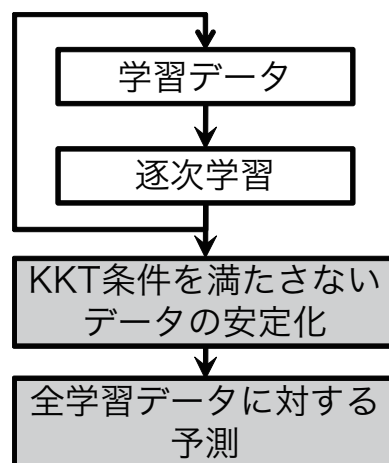


図 3.1 Parrella が提案した手法の概要 (図 2.3 の再掲)

いる。これは過去のサポートベクトルと新たに入力された訓練データ  $\mathbf{x}_i$ , そして逐次更新されるパラメー

タ  $b'$  を用いて、次時刻に於ける出力  $y_{N+1}$  の予測式を構成する事で、Online SVR による逐次予測を提案しているものである。この手法に於いて予測式は

$$\hat{y}_{N+1} = \begin{cases} 0 & \text{if } N = 0 \\ \Delta\theta & \text{if } N = 1 \\ \mathbf{k}_{sv}(x_N)^\top (\mathbf{K}_{sv} + \lambda \mathbf{I}_l)^{-1} \mathbf{y}_{sv} + b' & \text{otherwise} \end{cases} \quad (3.1.1)$$

で定義しており、また、式 (3.1.1) に於ける  $b'$  は、現在までのバイアス値の積算値として求めている。

本研究の目的は、内部状態と行動から、未来の内部状態を予測する事にある。この考え方を示したものが図 3.2 である。この図に於いて、現時刻  $t_1$  に於いて行動  $\mathbf{a}(t_1)$  を実行する事によって、内部状態は  $\mathbf{x}(t_2)$  に変化する。過去の内部状態と行動の組み合わせを学習する事により、次時刻  $t_2$  に於いて行動  $\mathbf{a}(t_2)$  を実行した場合の未来の内部状態を予測する事が可能となる。この予測式 (3.1.1) を適用する事で、Parrella の指摘す

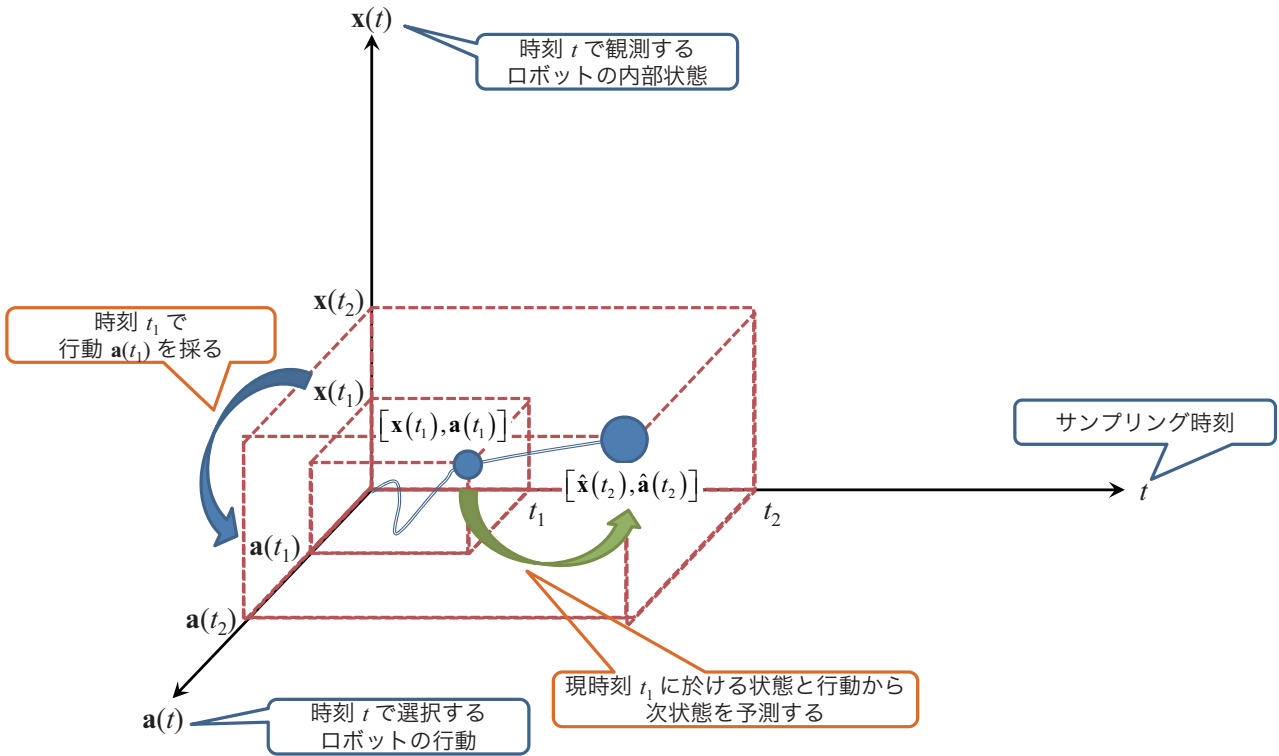


図 3.2 内部状態と行動から未来を予測する考え方

る問題を回避し、次状態の予測が可能となる。但し、本予測の定義に於いて、状態及び行動と云う概念は取り入れていない。

そこで、本研究ではこの式 (3.1.1) を拡張し、入力を次の様に、現状態と行動を行列の形式でまとめて再定義する事を考える。

$$\mathbf{z}_t = [\mathbf{x}_t \mid a_t] \quad (3.1.2)$$

上式 (3.1.2) に於いて、 $\mathbf{x}_t$  は時刻  $t$  でのロボットの内部状態、 $a_t$  は  $t$  での行動を示している。

この時、内部状態  $\mathbf{x}_t$  に対する予測値  $\hat{x}_{t+1,i}, i \in \dim \hat{\mathbf{x}}_{t+1}$  を、センサなどより得た内部状態  $\mathbf{z}_t = [x_{t,1}, \dots, x_{t,n} | a_t]$  からなるデータを用い推定する事を考える。SVR モデル  $y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$  で示される回帰関数及び訓練セット  $\{\mathbf{x}_j, y_j\}$  に対して、次式 (3.1.3) で示される正則化項付き誤差自乗和関数  $J$  を考える [29]。

$$J(\mathbf{w}) = \frac{1}{2} \sum_{j=1}^l \{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \}^2 + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \quad (\lambda \geq 0) \quad (3.1.3)$$

ここで,  $\lambda$  は正則化パラメータである. この  $J$  を最小化する  $\mathbf{w}$  は  $J(\mathbf{w})$  の傾きを 0 と等置する (つまり  $\partial J(\mathbf{w})/\partial \mathbf{w} = 0$ ) 事によって得られる. よって,

$$\mathbf{w} = -\frac{1}{\lambda} \sum_{j=1}^l \{\mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j\} \phi(\mathbf{x}_j) = \sum_{j=1}^l a_j \phi(\mathbf{x}_j) = \Phi^\top \mathbf{a} \quad (3.1.4)$$

$$\text{where } \mathbf{a} = [a_1 \ \dots \ a_l]^\top, \ a_j = -\frac{1}{\lambda} \{\mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j\}$$

となる. 但し,  $\Phi$  は計画行列 (design matrix)<sup>[30]</sup> と呼ばれ, その第  $j$  列は  $\phi(\mathbf{x}_j)^\top$  である. ここでパラメータベクトル  $\mathbf{w}$  を  $\Phi \mathbf{a}$  に置換すると,

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^\top \Phi \Phi^\top \Phi \Phi^\top \mathbf{a} - \mathbf{a}^\top \Phi \Phi^\top \mathbf{y} + \frac{1}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2} \mathbf{a}^\top \Phi \Phi^\top \mathbf{a} \quad (3.1.5)$$

ここで, 計画行列  $\Phi$  を用いて Gram 行列  $\mathbf{K} = \Phi \Phi^\top$  を定義する<sup>[31]</sup>. この時  $\mathbf{K}$  の要素は

$$K_{jm} = \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}_m) = k(\mathbf{x}_j, \mathbf{x}_m) \quad (3.1.6)$$

となる. 即ちこれは既出の Kernel 関数となる対称行列である. Gram 行列によって正則化項付き誤差自乗和関数を書き換えると,

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^\top \mathbf{K} \mathbf{K} \mathbf{a} - \mathbf{a}^\top \mathbf{K} \mathbf{y} + \frac{1}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2} \mathbf{a}^\top \mathbf{K} \mathbf{a} \quad (3.1.7)$$

のようになる. これを  $\mathbf{a}$  について解くと

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_l)^{-1} \mathbf{y} \quad (3.1.8)$$

となる. 従って, この結果を用いて SVR モデルに対して新たに  $\mathbf{x}$  を入力した時, この  $\mathbf{x}$  に対する予測結果  $\hat{y}(\mathbf{x})$  は次式に従って導かれる.

$$\hat{y}(\mathbf{x}) = \mathbf{w} \phi(\mathbf{x}) + b = \mathbf{a}^\top \Phi \phi(\mathbf{x}) + b = \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbf{I}_l)^{-1} \mathbf{y} + b \quad (3.1.9)$$

$$\text{where } \mathbf{k}(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}) \ \dots \ k(\mathbf{x}_l, \mathbf{x})]^\top$$

ここで,  $b$  はバイアス項,  $\mathbf{k}$  は特徴空間に於いて内積を求める写像関数を示しており,

$$b = \epsilon + x_t - \sum_{j=1}^l \theta_j k(\mathbf{x}_j^\top \mathbf{x}_j) \quad (3.1.10)$$

である<sup>[17, 29]</sup>. また,  $\epsilon$  は  $\epsilon$ -insensitive 損失関数<sup>[17, 32]</sup> に於ける不感帯域である.

この式を基に, 本研究で目的の一つとしている Online SVR に於ける状態の逐次予測の実現を考える. ここでは, センサなどからの入力が  $n$  次元の場合を考え,  $\dim \mathbf{x}_i = n$  で与えられる事を前提とする. 基本的な考え方としては, 過去のサポートベクトルから, 1 時刻先の値を予測するものである. 過去の  $l$  個のサポートベクトル  $\mathbf{z}_{s_k}$  ( $k \in l$ ) を用いて導かれるカーネル関数  $k$  から構成される Gram 行列  $\mathbf{K}_{sv}$  を,

$$k(\mathbf{z}_{s_i}, \mathbf{z}_{s_j}) = \phi(\mathbf{z}_{s_i})^\top \cdot \phi(\mathbf{z}_{s_j}) = Q_{s_i, s_j} \quad (3.1.11)$$

とする事で,

$$\mathbf{K}_{sv} = \begin{bmatrix} Q_{s_1, s_1} & Q_{s_1, s_2} & \cdots & Q_{s_1, s_l} \\ Q_{s_2, s_1} & Q_{s_2, s_2} & \cdots & Q_{s_2, s_l} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{s_l, s_1} & Q_{s_l, s_2} & \cdots & Q_{s_l, s_l} \end{bmatrix} \quad (3.1.12)$$

と定義する. また, 新たな入力  $\mathbf{z}_t$  とそれまでのサポートベクトルとの特徴空間に於ける内積を求める写像関数  $\mathbf{k}_{sv}(\mathbf{z}_t)$  については

$$\mathbf{k}_{sv}(\mathbf{z}_t) = \left[ k(\mathbf{z}_{s_1}, \mathbf{z}_t) \quad \dots \quad k(\mathbf{z}_{s_l}, \mathbf{z}_t) \right]^\top \quad (3.1.13)$$

とする.

以上の定義より, 新たな入力  $\mathbf{z}_t$  に対する内部状態の予測値  $\hat{x}_{t+1,i}$  を考える.

ここで, Online SVR に於いては, 先述の通り学習データが追加されるに従い逐次学習を繰り返すものであった. 学習データとしてサンプル  $c$  が追加された場合, ラグランジュ乗数, バイアス項は

$$\theta' = \theta + \Delta\theta \quad (3.1.14)$$

$$b' = b + \Delta b \quad (3.1.15)$$

$$\sum_{q \in S} Q_{pq} \Delta\theta_q + \Delta b = -Q_{pc} \Delta\theta_c \quad \text{where } p \in S \quad (3.1.16)$$

$$\sum_{q \in S} \Delta\theta_q = -\theta_c \quad (3.1.17)$$

により更新される<sup>[15]</sup>. ここで,  $S$  は学習に關与する訓練データであるサポートベクトルの集合  $\{s_k\}$ , ( $k \in l$ ) である. つまり, 新たなサンプル  $c$  に対するラグランジュ乗数とバイアスは, 現在までの積算値から求められる. そこで, 上式から, 予測式の構成を行う.

先ず,  $t = 0$  の時は, 初期状態にて如何なるデータも与えられていないので,

$$\hat{x}_{t+1,i} = \hat{x}_{1,i} = 0 \quad (3.1.18)$$

とする. 次に  $t = 1$ , つまり一つ目の学習データが与えられた場合を考える. この時は  $\mathbf{x}_1$  以外に学習データが存在せず, 特徴空間における超平面が構成できない事から, サポートベクトル自体も存在しない. この為,

$$\Delta b_i = \Delta\theta_i \quad (3.1.19)$$

と導かれる. よって,

$$\hat{x}_{t+1,i} = \hat{x}_{2,i} = \Delta b_i = \Delta\theta_i \quad (3.1.20)$$

となる. 続いて,  $t \geq 2$  の場合を考える. このときは,  $\mathbf{x}_2$  以外にも学習データが存在するので, 特徴空間に於ける超平面が構成出来, サポートベクトルが存在する. よって,

$$\hat{x}_{t+1,i} = \mathbf{k}_{sv}(\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda \mathbf{I}_l)^{-1} \mathbf{z}_{sv} + b'_i \quad (3.1.21)$$

となる. 以上をまとめると,

$$\hat{x}_{t+1,i} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}(\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda \mathbf{I}_l)^{-1} \mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad \text{when } i \in \dim \mathbf{x}_t \quad (3.1.22)$$

という予測式が得られる. また,  $b'$  は, 現在までのバイアス値の積算値として求められる. この予測式を適用する事により, 次状態の予測が可能となる.

## 3.2 追加学習を組合わせた逐次内部状態予測

前節では, 逐次学習による逐次内部状態予測の考え方を説明してきた. 但し逐次学習に於いてデータセットの長さを固定にした場合, 定められたデータセットに達しない限り学習が終わらず, 予測が出来ないという可能性が存在する. そこで, データセット長をデータ数  $N$  によって設定し, 現在のデータと直近  $N - 1$



データを基点として訓練と予測を行う事を考える必要がある. つまり, 一つずつ訓練データが増えるに従い, 追加学習 (Incremental Learning) を組み合わせ学習と予測する事を考える (図 3.3).

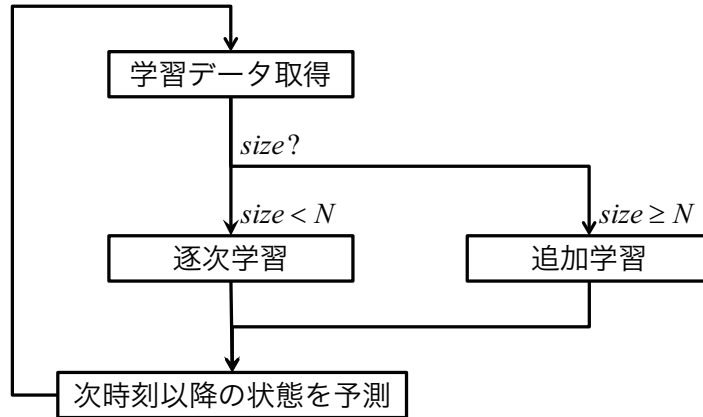


図 3.3 追加学習を組合わせた逐次内部状態予測

Online SVR では,  $N - 1$  番目の訓練データを学習した時点で学習に用いるパラメータを更新する. つまり,  $N - 1$  番目のデータを学習した時点で重みベクトル  $\mathbf{w}_{N-1}$ , バイアス項  $b$  として, その訓練データの特徴をパラメータとして抽象化している [15]. そこで本研究では, その点に着目し,  $N - 1$  番目の訓練データと  $N$  番目の訓練データを学習し, その結果を用いて 1 サンプル時刻先に於ける内部状態を予測する事を考える. そこで, バッチ SVR を組合わせ追加学習を行う.

$N - 1$  番目にてサポートベクトルの集合  $S_{N-1}$ , 重みベクトル  $\mathbf{w}_{N-1}$ , バイアス項  $b$  が定義されて更新されたものとする. この時  $N$  番目のデータが与えられた時は

$$\min_{\mathbf{w}_N, b} \sum_{j=N-1}^N \xi(y_{ji} - f(\mathbf{x}_j)) + \frac{\lambda}{2} \|\mathbf{w}_N\|^2 \quad i \in \dim \mathbf{x}_j \quad (2.3.3)$$

を最小化する. これによって, 最適解として  $\mathbf{w}_N, b$  が得られる. これらは  $N$  番目の訓練データに於ける特徴となっている. この結果を  $N - 1$  番目の訓練データを学習した時点でのパラメータと統合する事によって, 追加学習が実現する. つまり,

$$\mathbf{w} = [\mathbf{w}_{N-1} \quad \mathbf{w}_N] \quad (3.2.1)$$

$$b' = b \quad (3.2.2)$$

$$S = \{s_1, \dots, s_{N-1}, s_N\} \quad (3.2.3)$$

である. これによって, 式 (3.1.22) で示した予測式を用い, 追加学習を行いつつ逐次内部状態予測を行う事が可能となる.

### 3.3 長期内部状態予測 – $n$ サンプル時刻先の内部状態予測

本研究では, 現在の内部状態と行動から, 次に遷移する内部状態を予測する事を目的としている. つまり 1 サンプル時刻先では, 現在の内部状態と現在採った行動から内部状態を予測出来る (逐次内部状態予測). この考え方を  $n$  サンプル時刻先に於ける内部状態の予測にまで拡張した場合, 2 サンプル時刻以降に於ける状態と行動は, 何らかの制御則によって決定される事となる.

予測対象となるモデルの動的挙動が, 適切に離散時間での非線型状態空間モデルとして記述する事が出

来るとするならば、次式の様になる。

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, a_k) \quad (3.3.1)$$

このモデルは、何らかの制御によって目標状態に内部状態を移動する事が出来る事から、以降の内部状態が予測出来るという事である。従って、 $x_{t-1}$  が知られていると、 $k = t$  を起動した場合

$$\begin{aligned} \mathbf{x}(t+1) &= f_t[\mathbf{x}(t), a(t)] \\ &\vdots \\ \mathbf{x}(t+n) &= f_{t+n-1}[\mathbf{x}(t+n-1), a(t+n-1)] \end{aligned} \quad (3.3.2)$$

によって連続的に導かれる [33]。

この式 (3.3.2) で示された考え方を、行動と内部状態を組合わせて予測する方法に適用した場合のブロック図が、次図 3.4 となる。

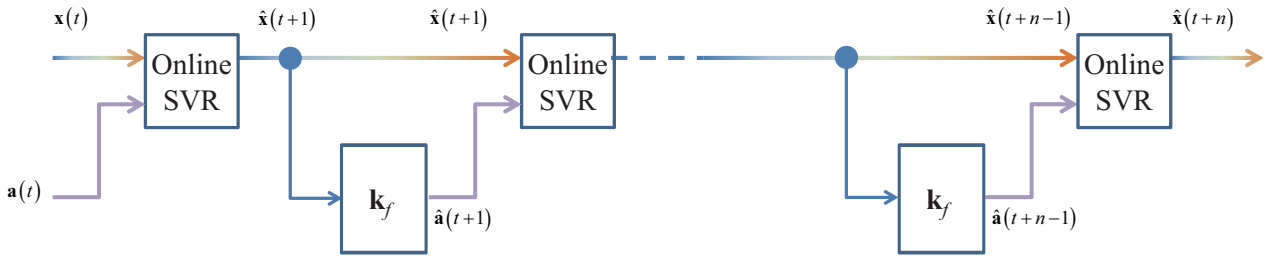


図 3.4 長期内部状態予測の為のブロック図

図 3.4 は、ロボットの内部状態と、その時採る行動から、未来の内部状態を予測する概念図である。或るサンプリング時刻  $t$  で一つの Online SVR ブロックに時系列データ  $\mathbf{x}_t$  と行動  $\mathbf{a}_t$  を与える事で、1 サンプル時刻先の内部状態  $\hat{\mathbf{x}}_{t+1}$  が予測出来る。ここで、 $\hat{\cdot}$  は、得られる予測値を表現する。そして、予測した内部状態を、行動を決定する為の状態フィードバックゲイン  $\mathbf{k}_f$  に入力する事によって、新たに未来の行動  $\hat{\mathbf{a}}_{t+1}$  をも予測出来る様になる。つまり、任意のサンプリング時刻  $t+n-1$  ( $n \geq 3$ ) に於いて

$$\mathbf{a}_{t+n-1} = \mathbf{k}_f \mathbf{x}_{t+n-1} \tag{3.3.3}$$

$$\mathbf{z}_{t+n-1} = \begin{bmatrix} \mathbf{x}_{t+n-1} & | & \mathbf{a}_{t+n-1} \end{bmatrix} \tag{3.3.4}$$

$$x_{t+n,i} = \mathbf{k}_{sv} (\mathbf{z}_{t+n-1})^\top (\mathbf{K}_{sv} + \lambda \mathbf{I}_i)^{-1} \mathbf{z}_{sv} + b'_i \quad \text{where } i \in \dim \mathbf{x}_{t+n-1} \tag{3.3.5}$$

の様に、行動を内部状態を互いに導く事によって、長期的な内部予測が可能となる。

また、図 3.4 に於ける“Online SVR” ブロックは、前節 3.1 及び 3.2 にて定義した逐次内部状態予測部によって構成されている (図 3.5)。

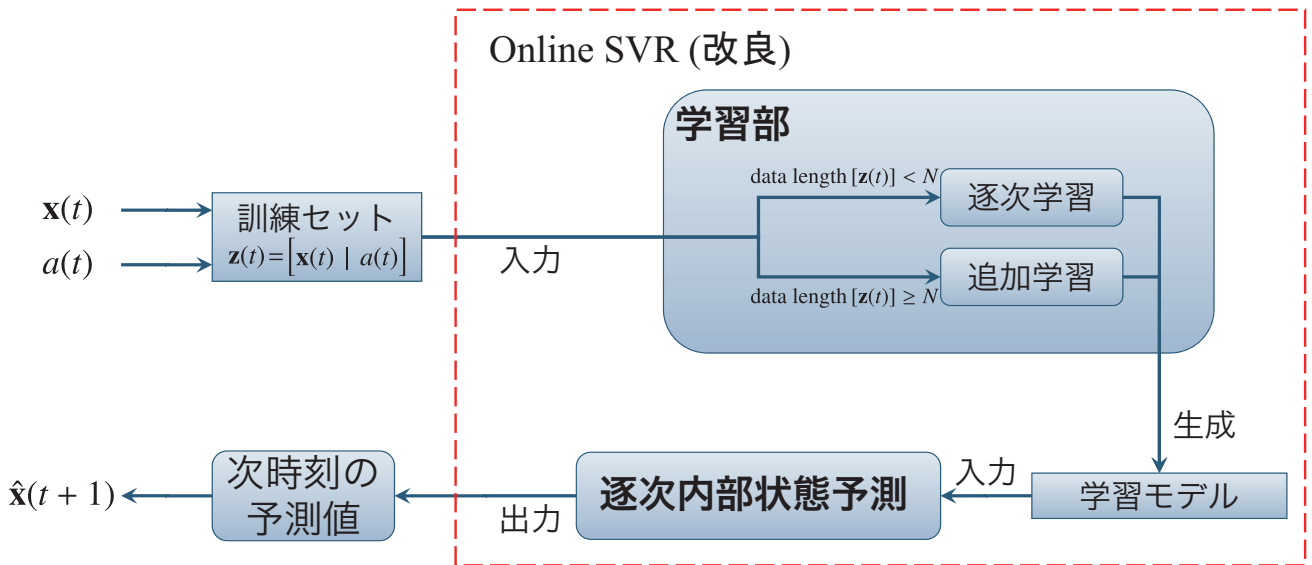


図 3.5 逐次内部状態予測部のブロック図

ここで、 $\text{data length} [\mathbf{z}(t)]$  は、 $\mathbf{z}_t$  のデータセット長を示しており、データの次元を表す  $\dim \mathbf{z}_t$  とは異なる意味を持つ。



## 第4章 自走式二輪型倒立振子への応用

本章では、本研究にて適用対象としている自走式二輪型倒立振子“NXTway-GS”について解説する。また、このNXTway-GSに対する『状態予測』についての適用方法について説明する。本章に於いて『行動』と『制御』は同義である。

### 4.1 LEGO MINDSTORMS NXT

LEGO社が提供するプログラム可能なロボットパーツセットである。レゴブロックでロボットのハードウェアを作成し、プログラミングソフトROBOLABによって、レゴブロックに似たブロックをパソコン上の画面で組み上げソフトウェアを作成する事で、思い通りに動作するロボットを作成出来る。

#### 4.1.1 NXTway-GS

NXTway-GSは、NXTを用いて作成した自走式二輪型倒立振子である(図4.1)。本研究で適用対象となるモデルはこれである。

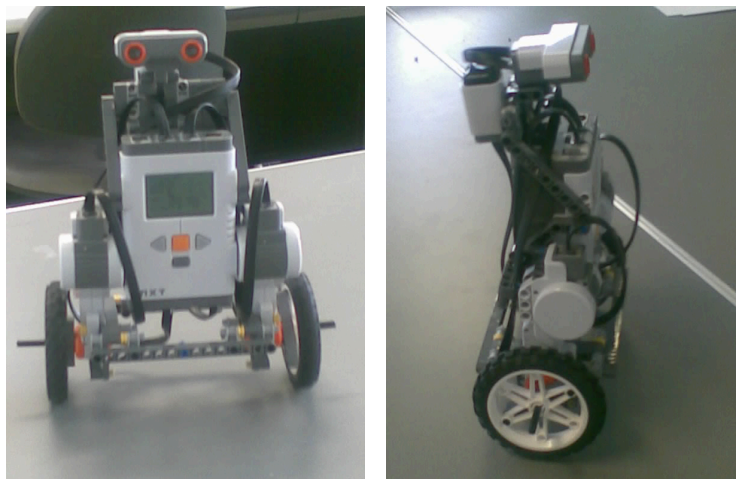


図 4.1 自走式二輪型倒立振子 NXTway-GS の概観

車体背部に HiTechnic 製ジャイロセンサを装着し、車体傾斜角度をリアルタイムに計測する事が可能となっている。

■センサ・アクチュエータについて NXTway-GS が装着し使用しているセンサおよびアクチュエータの特性を、表 4.1 および表 4.2 に示す [34]。

ここで、データタイプの `int32`, `uint16`, `int8` はそれぞれ、32bit 符号付き整数, 16bit 符号無し整数, 8bit

表 4.1 センサの特性

| センサ       | 出力値   | 単位          | データタイプ | 最大サンプル数 [1/sec.] |
|-----------|-------|-------------|--------|------------------|
| ロータリエンコーダ | 回転角度  | [deg.]      | int32  | 1000             |
| 超音波センサ    | 距離    | [cm]        | int32  | 50               |
| ジャイロセンサ   | 傾斜角速度 | [deg./sec.] | uint16 | 300              |

表 4.2 アクチュエータの特性

| アクチュエータ | 入力値 | 単位  | データタイプ | 最大サンプル数 [1/sec.] |
|---------|-----|-----|--------|------------------|
| DC モータ  | PWM | [%] | int8   | 1000             |

符号付き整数である。

## 4.2 現代制御論

### 4.2.1 概要

状態方程式(state equation) と呼ばれる一階の常微分方程式として表現された制御対象に対し、力学系を始めとする種々の数学的な成果を応用して、フィードバック形の安定性、時間応答や周波数応答等を評価して望みの挙動を達成する事を目的とする理論である。状態方程式に含まれる状態変数にベクトルを選ぶ事が出来る為、多入力多出力系の表現が容易となり、複雑な系に対して多くの成果が得られる様になった。

古典制御論を始めとする従来の制御理論に於いては、専ら単一の入力に対する単一の出力の応答を取り扱っていた。つまり、入出力の関係を示すものは、一つのまとまった形の伝達関数であった。

しかしながら、世の中に存在するシステムを考えると、入力から出力に達するまでの間には多くの要素が在る事が普通である。これら各要素間の信号にも着目して、系の内部構成が応答にどの様に影響するかをも考慮し、更に入力についても多くの入力を同時に考え様という、所謂多入力多出力系(多変数系とも)として扱う様に考えられたのが、現代制御論である。

この現代制御論では、次の形の連立 1 階微分方程式系を制御対象として扱う。

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (4.2.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (4.2.2)$$

式 (4.2.1) を状態方程式(state equation)、式 (4.2.2) を出力方程式(output equation) と云い、この 2 つをまとめてシステム方程式(system equation) と呼ぶ。各変数のサイズ及び名称を以下に示す。

$\mathbf{x}(t)(n \times 1)$  状態 (変数)

$\mathbf{u}(t)(n \times 1)$  入力 (変数)

$\mathbf{y}(t)(m \times 1)$  出力 (変数)

この場合、各行列のサイズは

$$\mathbf{A}(n \times n), \mathbf{B}(n \times r), \mathbf{C}(m \times n), \mathbf{D}(m \times r)$$

である。 $\mathbf{u}(t)$  から  $\mathbf{y}(t)$  への直達項 (feedthrough term) の在る系は稀であり、通常は  $\mathbf{D} = 0$  である。

この式 (4.2.1), (4.2.2) で記述される系をブロック線図で表現すると、図 4.2 の様になる。

ここで、 $\mathbf{u}(t), \mathbf{x}(t), \mathbf{y}(t)$  はベクトルであるので、これらは各変数を含んでいる。よって、その中には実

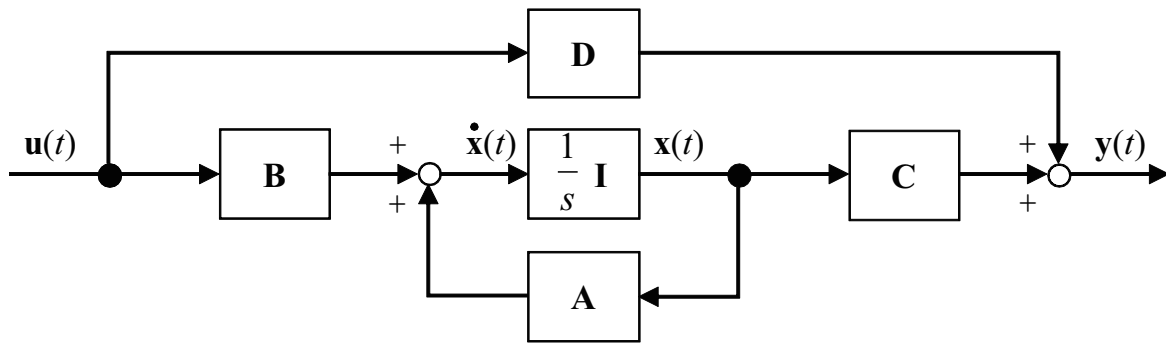


図 4.2 典型的な状態空間モデル

際に制御出来るものと出来ないもの、或いは観測出来るものと出来ないものが在る。そこで、可制御性(controllability)、可観測性(observability) という概念を考える [35, 36]。

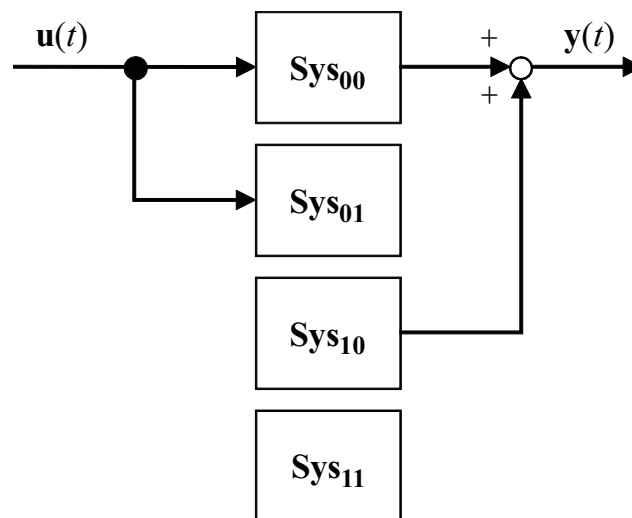


図 4.3 可制御と可観測な系

以降、制御対象のモデルとして、

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \quad (4.2.3)$$

と云う状態方程式表現を用いる。

#### 4.2.1.1 可制御性

或る制御入力  $\mathbf{u}(t)$  によって有限時間  $t_f \geq 0$  に系の初期状態  $\mathbf{x}(0)$  から、任意の最終状態  $\mathbf{x}(t_f)$  に到達出来る系を可制御(controllable) と言う。図 4.3 の場合、 $\mathbf{Sys}_{00}$  と  $\mathbf{Sys}_{01}$  は可制御である。

行列に着目すると、次の  $n \times nr$  の複合行列が  $n$  の階数(rank) である時、可制御となる。

$$\left[ \mathbf{B} | \mathbf{A}\mathbf{B} | \mathbf{A}^2\mathbf{B} | \cdots | \mathbf{A}^{n-1}\mathbf{B} \right] \quad (4.2.4)$$

#### 4.2.1.2 可観測性

出力  $\mathbf{y}(t)$  を有限時間  $0 \leq t \leq t_f$  の間に観測する事により、時刻 0 に於ける全ての状態  $\mathbf{x}(0)$  を求める事が出来るならば、可観測(observable) と言う。図 4.3 の場合、 $\mathbf{Sys}_{00}$  と  $\mathbf{Sys}_{10}$  は可観測である。

行列に着目した時, 次の  $n \times nr$  の複合行列が  $n$  の階数であれば可観測となる.

$$[C^T | A^T C^T | A^{T^2} C^T | \dots | A^{T^{n-1}} C^T] \quad (4.2.5)$$

### 4.2.2 状態フィードバック制御

本研究で考える制御とは, 状態  $\mathbf{x}(t)$  を速やかに  $\mathbf{0}$  にする事である. 『 $\mathbf{x}(t) = \mathbf{0}$ 』を, その系が安定に動作している望ましい状態と考える. 外乱により  $\mathbf{x}(t)$  が  $\mathbf{0}$  からずれても, 制御によって速やかに  $\mathbf{0}$  に戻したいとする.

制御対象が不安定な場合を例にとる. この例では, 式 (4.2.3) の  $\mathbf{A}$  の固有値に実部が正のものが在る場合である. この場合, 何の制御もしなければ ( $\mathbf{u}(t) = \mathbf{0}$  で  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$  となると) 状態  $\mathbf{x}(t)$  は  $\mathbf{0}$  から遠ざかり, 終いには発散する. これを抑えるには, 式 (4.2.3) にて何らかの  $\mathbf{u}(t)$  を作用させ,  $\mathbf{x}(t)$  の動きを抑えなければならない. ここで, 不安定な制御対象に対し, 制御によって発散を防ぐ事を, 安定化(stabilization) という.

逆に, 制御対象が安定な場合には, 放っておいても ( $\mathbf{u}(t) = \mathbf{0}$  でも)  $\mathbf{x}(t) \rightarrow \mathbf{0}$  となる. しかし, その収束の速さは  $\mathbf{A}$  の固有値により決し,  $\mathbf{A}$  の固有値が虚軸に近い程収束は遅くなる. 出来るだけ速く  $\mathbf{0}$  に収束する事が望ましいので, やはり何らかの  $\mathbf{u}(t)$  を作用させ,  $\mathbf{x}(t)$  の収束速度を調整する必要がある.

従って,  $\mathbf{u}(t)$  を何らかの方法で算出し, 式 (4.2.3) の制御対象に作用させ,  $\mathbf{x}(t)$  を速やかに  $\mathbf{0}$  にする事が求められる.

ここで, 入力  $\mathbf{u}(t)$  の算出法を考える. 制御対象に図 4.2 で表現される状態空間モデルを扱うものとする. そして, 行列  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  については各要素の値が既知であり, 制御系設計時に利用出来るものを仮定する.

ここでの制御の目的は, 出来るだけ速く  $\mathbf{x}(t) \rightarrow \mathbf{0}$  とする事である.  $\mathbf{x}(t)$  を  $\mathbf{0}$  に戻す様なコントローラをレギュレータ(regulator)と呼ぶ. 制御対象に於ける状態  $\mathbf{x}(t)$  が  $\mathbf{y}(t)$  として観測可能な場合には, レギュレータに定数行列を用いる方法が在る. つまり, 或る定数行列を  $\mathbf{F}$  とし

$$\mathbf{u}(t) = -\mathbf{F}\mathbf{x}(t) \quad (4.2.6)$$

の  $\mathbf{u}(t)$  を制御入力とする. この方法を状態フィードバック(state feedback) という. なお, 式 (4.2.6) における定数行列  $\mathbf{F}$  は状態フィードバックゲイン(state feedback gain) と呼ばれる. 下図は状態フィードバックを用いた制御系の一例である.

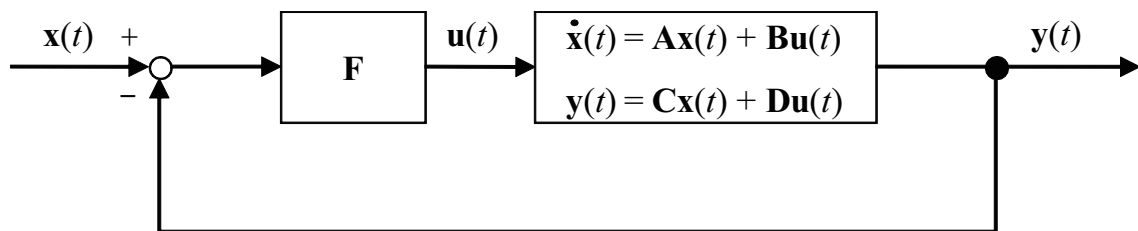


図 4.4 状態フィードバックによる制御系の例

### 4.2.3 制御系の状態方程式と安定性

式 (4.2.6) の状態フィードバックゲイン  $\mathbf{F}$  は, 状態量  $\mathbf{x}(t)$  を制御入力  $\mathbf{u}(t)$  に変換する定数行列である.  $\mathbf{F}$  の設定により, 制御は左右される.

制御に望まれる第一の条件は, 制御系が安定である事, つまり制御系の内部に在る変数が  $\mathbf{0}$  に収束する事である. 制御系の内部変数が満たしている方程式は制御系の状態方程式(フィードバック系の状態方程



式)と呼ばれ,この式が安定条件を満たす事が望ましい。

図 4.4 のフィードバック系に於いては,制御対象とコントローラと云う 2 つの系が在る. この場合では,コントローラは定数行列  $\mathbf{F}$  であり,コントローラの内部に変数は無い. 内部変数を持っているのは制御対象であり,それが  $\mathbf{x}(t)$  である. そこで  $\mathbf{x}(t)$  について成立する方程式を導く。

先ず,制御対象で式 (4.2.3) が成立している. また  $\mathbf{u}(t)$  が式 (4.2.6) で示されるので,これを式 (4.2.3) に代入する事で

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{F}\mathbf{x}(t) \\ &= (\mathbf{A} - \mathbf{B}\mathbf{F})\mathbf{x}(t)\end{aligned}\quad (4.2.7)$$

を得る. これが図 4.4 の制御系の状態方程式である。

こうして求めた式 (4.2.7) の系の変数  $\mathbf{x}(t)$  が  $\mathbf{0}$  に収束するかどうかを考える. ここで,次の定理を用い考察する。

定理 (線型自由システムが漸近安定である必要十分条件) 線型自由システム

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \quad (4.2.8)$$

が漸近安定である必要十分条件は,  $\mathbf{A}$  のすべての固有値の実部が負である事である。

この様に,系の安定性は行列  $\mathbf{A}$  の固有値により決まる.  $\mathbf{A}$  が  $n \times n$  行列なら,その固有値は  $n$  個あり,それら全ての実部が負なら安定であり,一つでも正であれば不安定となる。

この式 (4.2.8) の  $\mathbf{A}$  を,  $\mathbf{A} - \mathbf{B}\mathbf{F}$  に置き換えて考えると,式 (4.2.7) の系が安定である為の必要十分条件は,  $\mathbf{A} - \mathbf{B}\mathbf{F}$  の全ての固有値の実部が負である事となる。

この様な,制御系の状態方程式の安定性を決める行列の固有値を,閉ループ極(フィードバック系の極)と呼ばれる. 図 4.4 のフィードバック系の場合,その極は  $\mathbf{A} - \mathbf{B}\mathbf{F}$  の固有値として決まる。

制御系を安定にする為,  $\mathbf{F}$  の設定に於いては,閉ループ極の実部が負となる事を念頭に考えなければならない. そして,  $\mathbf{x}(t)$  は素早く  $\mathbf{0}$  に収束する事が望ましい為,  $\mathbf{A} - \mathbf{B}\mathbf{F}$  の固有値が,複素平面上で虚軸から離れた位置にある様に  $\mathbf{F}$  を設定すべきである. その望ましい位置は,図 4.5 のおおよそ斜線部の辺りである。

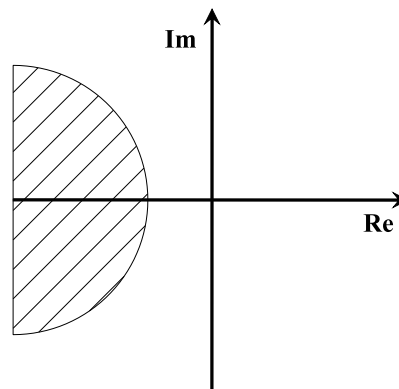


図 4.5 閉ループ極の望ましい位置

#### 4.2.4 安定とは

安定判別には,リアプノフの第 2 法則が用いられる. これは,『微分方程式の安定問題を,解に対する明白な知識を持たずに与えられた形の式を用い判別する』方法である。

この方法は線型, 非線型の区別無く全ての系に適用出来る. 線型時不変系の場合には

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \tag{4.2.9}$$

また非線型, 時変の場合は

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}, t) \tag{4.2.10}$$

と云う形で扱う. この場合, 全ての  $t$  に対し  $\dot{\mathbf{x}} = 0$  となる点  $\mathbf{x} = \mathbf{x}_e$  を平衡点という. 線型時不変の場合,  $\mathbf{A}$  が正則の時は平衡点  $\mathbf{x}_e$  は 0 の部分のみだが,  $\mathbf{A}$  が特異の場合には平衡点は多数存在する. 非線型の場合にも平衡点が多数存在する事がある.

#### 4.2.4.1 安定の定義

安定性については, 幾つかの定義が与えられている [37].

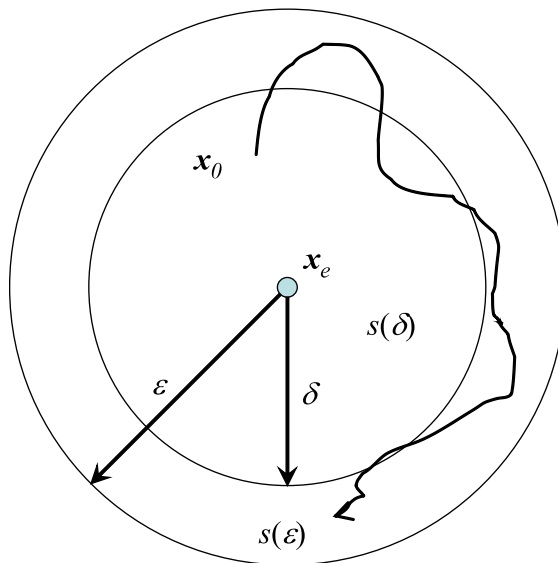


図 4.6 リアプノフの意味での安定

■安定 (stable)[リアプノフの意味での安定] 或る  $s(\delta)$  の中から出発する解軌道が図 4.6 に示す様に  $t \rightarrow \infty$  となっても, 或る状態空間  $s(\epsilon)$  よりも外に出ない時安定と言う. 即ち

$$0 \leq \mathbf{x}_0 \leq \delta \tag{4.2.11}$$

の初期条件から出発して全時間範囲に対し

$$0 \leq \mathbf{x}(t) \leq \epsilon \tag{4.2.12}$$

である.

■漸近安定 (asymptotically stable) 上述の安定性を満足していると同時に, 原点  $\mathbf{x}_e$  の十分近くから出発した軌道が最終時間に於いて  $\mathbf{x}_e$  に収束する場合を漸近安定と言う. 即ち図 4.7 に示す様に  $t = t_0$  に於いて

$$\|\mathbf{x}(0) - \mathbf{x}_e\| \geq \delta \tag{4.2.13}$$

より出発し,  $t \geq T + t_0$  の時間では

$$\|\mathbf{x}(t) - \mathbf{x}_e\| \geq \mu \tag{4.2.14}$$

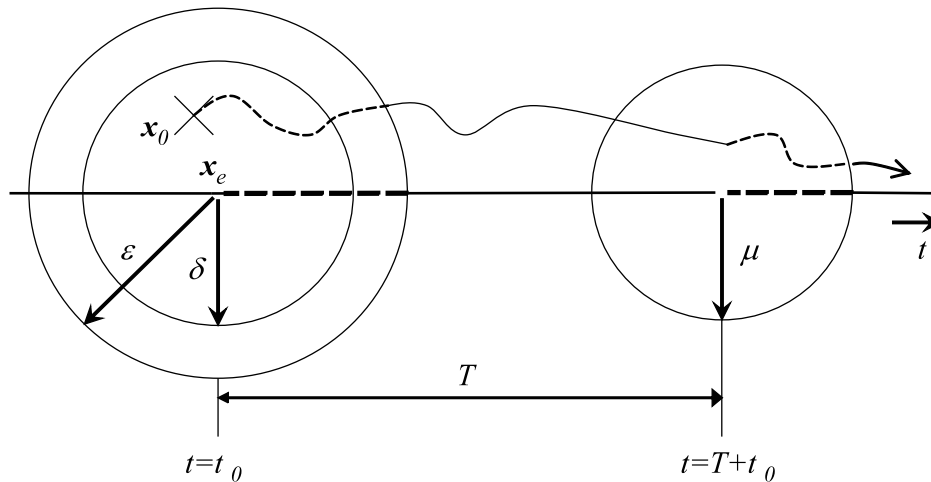


図 4.7 漸近安定

となる場合で、 $\mu$  は正実数であるので、十分時間が経つと  $\mu$  以上離れているものは無い事を示している。この収束の仕方が  $\mathbf{x}_0$  に依らず均一に行われる時、均一漸近安定 (equiasymptotically stable) と言い、 $t_0$  に依らず均一漸近安定の時、一様漸近安定 (uniformly asymptotically stable) という。

#### 4.2.4.2 正定と準正定

リアプノフの第2法則による安定判別は、『状態空間に於いて、原点を包囲するあらゆる大きさの閉局面を考え、この閉局面を  $V$  で表示した場合、この  $V$  は状態変数  $x_1, x_2, \dots, x_n$  より成る関数であり、原点では  $V = 0$ 、且つ  $V \neq 0$  の全状態空間では確実に正であるものとする。この  $V$  を Lyapunov 関数<sup>[38]</sup> と言う。そして或る状態点の軌道が  $V$  で表示される閉局面を全て外側から内側に向けて貫く時は、その系は安定であり、内側から外側に向けて貫く時は不安定である』と言う事である。

正定 (positive definite) とは、 $x_1 \neq 0, x_2 \neq 0, \dots, x_n \neq 0$  の時、 $V(x) > 0$  で、 $x_1 = x_2 = \dots = x_n = 0$  の時  $V(x) = 0$  の場合を指し、 $V(x) \geq 0$  の時を準正定 (positive semidefinite) と言う。

### 4.2.5 最適レギュレータ

制御系を設計する前に把握すべき制限や、得られる制御性能には妥協点がある。このような問題点を解決する一つの方策として、或るコストを表現する評価関数を最小にする様な制御則を構成する手段がある。その一例が、最適フィードバック制御則による最適レギュレータ (linear-quadratic regulator, LQR) である。

可制御である定係数線型システム

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (4.2.15)$$

に於いて、二次形式評価関数 (quadratic criterion function)

$$J[\mathbf{x}(t), \mathbf{u}(t)] = \int_0^{\infty} (\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)) dt \quad (4.2.16)$$

を最小にする最適制御入力

$$\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t), \mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \quad (4.2.17)$$

なる状態フィードバック制御則となる。

ここで、 $\mathbf{P}(n \times n)$  はリカッチ行列方程式 (Riccati matrix equation)

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \quad (4.2.18)$$

を満たす正定唯一解である。尚,  $\mathbf{R}, \mathbf{Q}$  は正定行列とする。

では何故, 評価関数 (4.2.16) を最小にする制御入力 が式 (4.2.17) で与えられるかについて考える [39]。

先ず始めに, 制御入力 (4.2.17) を系 (4.2.15) に適用しフィードバック形を構成した場合の系の安定性を調べる。  $\mathbf{B}^T \mathbf{P} = -\mathbf{R}\mathbf{K}$  を用い, 式 (4.2.18) は

$$\mathbf{P}(\mathbf{A} + \mathbf{BK}) + (\mathbf{A} + \mathbf{BK})^T \mathbf{P} = -(\mathbf{Q} + \mathbf{K}^T \mathbf{R}\mathbf{K}) \quad (4.2.19)$$

となる。ここで  $(\mathbf{Q} + \mathbf{K}^T \mathbf{R}\mathbf{R}^{-1} \mathbf{K}) > 0$  であるから正定な  $\mathbf{P}$  が存在すればフィードバック系

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}) \mathbf{x}(t) \quad (4.2.20)$$

は漸近安定となる。よって制御入力 (4.2.17) はフィードバック形を安定化する事が判る。

では, 次に最適性を考える。評価関数 (4.2.16) の積分の中を計算する。

$$\begin{aligned} \mathbf{x}^T \mathbf{Q}\mathbf{x} + \mathbf{u}^T \mathbf{R}\mathbf{u} &= \mathbf{x}^T (\mathbf{P}\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} - \mathbf{P}\mathbf{A} - \mathbf{A}^T \mathbf{P}) \mathbf{x} + \mathbf{u}^T \mathbf{R}\mathbf{u} \\ &= \mathbf{x}^T \mathbf{P}\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\mathbf{x} + \mathbf{u}^T \mathbf{R}\mathbf{u} + \mathbf{x}^T \mathbf{P}\mathbf{B}\mathbf{u} + \mathbf{u}^T \mathbf{B}^T \mathbf{P}\mathbf{x} - \mathbf{x}^T \mathbf{P}\dot{\mathbf{x}} - \dot{\mathbf{x}}^T \mathbf{P}\mathbf{x} \\ &= (\mathbf{u} + \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\mathbf{x})^T \mathbf{R} (\mathbf{u} + \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\mathbf{x}) - \dot{\mathbf{x}}^T \mathbf{P}\mathbf{x} \end{aligned} \quad (4.2.21)$$

である。ここで,  $t \rightarrow \infty$  のとき  $\mathbf{x} \rightarrow 0$  である。よって, 式 (4.2.21) の両辺を 0 から  $\infty$  まで積分すると

$$J[\mathbf{x}(t), \mathbf{u}(t)] = \mathbf{x}^T(0) \mathbf{P}\mathbf{x}(0) + \int_0^\infty \left\{ (\mathbf{u} + \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\mathbf{x})^T \mathbf{R} (\mathbf{u} + \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\mathbf{x}) \right\} dt \quad (4.2.22)$$

となる。今,  $\mathbf{R}$  は正定であると仮定しているので

$$\mathbf{u} = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\mathbf{x} \quad (4.2.23)$$

が成り立つ時  $J$  は最小となり, 最小値は

$$\min J = \mathbf{x}^T(0) \mathbf{P}\mathbf{x}(0) \quad (4.2.24)$$

となる。

尚, 評価関数 (4.2.16) の被積分項の第 1 項は解軌道の過渡特性に対する評価を, 第 2 項は制御入力のエネルギー消費に対する評価を与えている。最適レギュレータは, 重み行列  $\mathbf{R}, \mathbf{Q}$  に応じ, 過渡特性と入力エネルギー消費の釣り合いを決めるものである。つまり,  $\|\mathbf{R}\|$  を  $\|\mathbf{Q}\|$  に比べて大きく選ぶと, 過渡特性を犠牲にしても, 入力エネルギー消費を抑える事が出来,  $\|\mathbf{Q}\|$  を  $\|\mathbf{R}\|$  に比べて大きく選ぶと, 逆に入力エネルギー消費が大きくなっても, 過渡特性を改善する事が出来る。制御入力 (4.2.17) を見ると解る様に,  $\|\mathbf{R}\|$  を大きくすると状態フィードバックゲイン  $\|\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}\|$  は小さく,  $\|\mathbf{R}\|$  を小さくすると, 状態フィードバックは逆に大きくなる。制御系の漸近安定性は保証されているので, 状態フィードバックゲインが大きい程過渡応答は改善される。

## 4.2.6 サーボ制御

サーボ制御は, 物体の位置, 方位, 姿勢等を制御量とし, 目標値に追従する様に制御する方法である。サーボ (servo) の語源はラテン語の “*servus*”(英語の *slave*・*servant* の意) である。

ファクトリーオートメーションやロボット分野では欠かせない技術となっている。コントローラ (司令部) がサーボアンプ (制御部) を通してサーボモータ (駆動・検出部) を制御し, サーボモータは制御の状態を確認し制御部にフィードバックする。

## 4.3 NXTway-GS のモデリングと状態フィードバックゲインの導出

## 4.3.1 NXTway-GS のモデルと物理パラメータについて

NXTway-GS を図 4.8 に示す自走式二輪型倒立振子としてモデリングし、運動方程式を導出する。

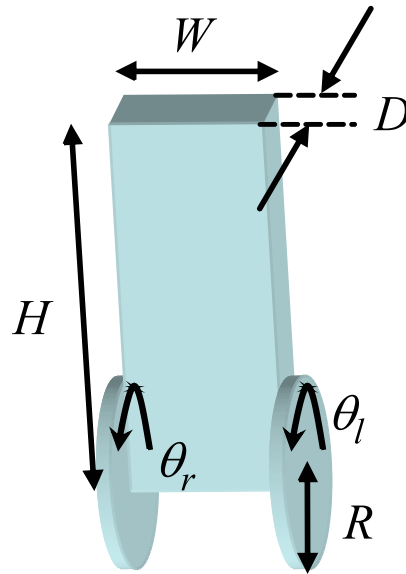


図 4.8 自走式二輪型倒立振子のモデル

上図をそのまま扱うと、座標の定義が不明確となるので、側面図 (図 4.9(a)) と平面図 (図 4.9(b)) に分け、座標や角度を考える。

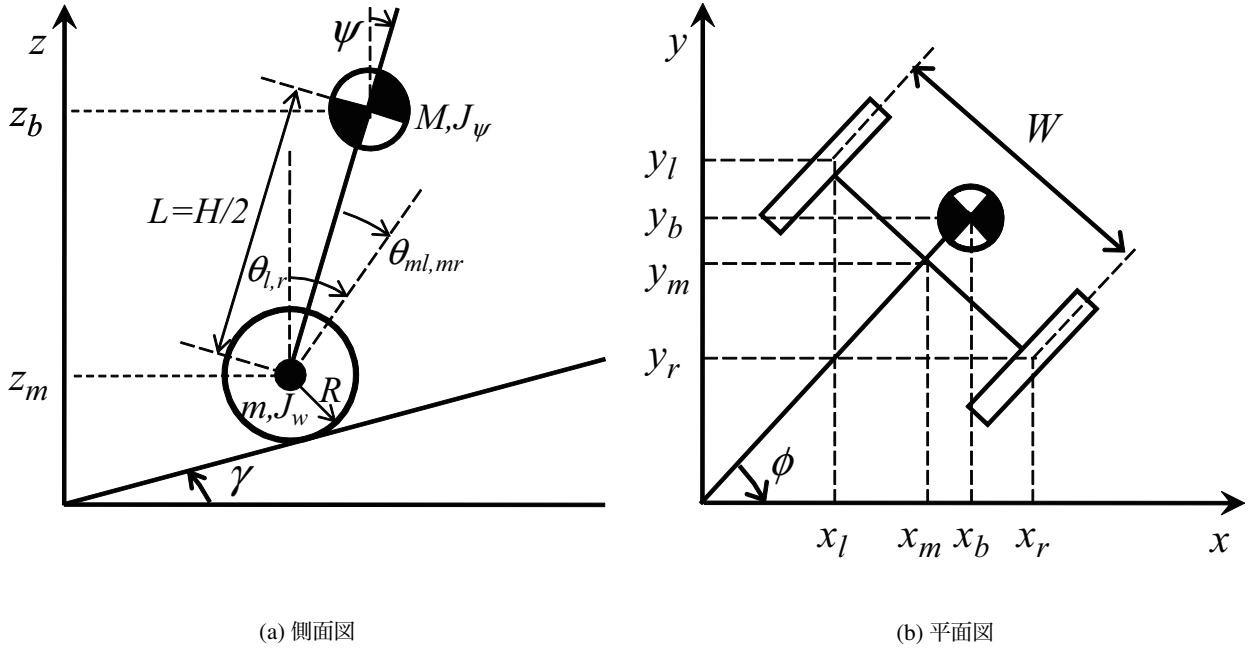


図 4.9 自走式二輪型倒立振り子の側面・平面図

上図にて、 $\psi$  は実機の傾斜角度を、 $\theta_{l,r}$  は車輪の回転角度を、 $\theta_{ml,mr}$  は直流モータの回転角度を示しており、添え字の  $l,r$  は左右を表している。

運動方程式を立てる前に、NXTway-GS の物理パラメータを知る必要がある。表 4.3 は、NXTway-GS の各種物理パラメータである。

### 4.3.2 NXTway-GS のモデリング

では、図 4.9 の座標系を用い、自走式二輪型倒立振り子の運動方程式を求め、時刻  $t = 0$  で自走式二輪型倒立振り子の向きが  $x$  軸の正方向とする時、次式で各座標は導かれる。

$$\begin{bmatrix} \theta & \phi \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\theta_l + \theta_r) & \frac{R}{W}(\theta_r - \theta_l) \end{bmatrix} \quad (4.3.1)$$

$$\begin{bmatrix} x_m & y_m & z_m \end{bmatrix} = \begin{bmatrix} R\theta \cos \phi & R\theta \sin \phi & R + R\theta \sin \phi \end{bmatrix} \quad (4.3.2)$$

$$\begin{bmatrix} x_r & y_r & z_r \end{bmatrix} = \begin{bmatrix} x_m + \frac{1}{2}W \sin \phi & y_m - \frac{1}{2}W \cos \phi & z_m \end{bmatrix} \quad (4.3.3)$$

$$\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} = \begin{bmatrix} x_m - \frac{1}{2}W \sin \phi & y_m + \frac{1}{2}W \cos \phi & z_m \end{bmatrix} \quad (4.3.4)$$

$$\begin{bmatrix} x_b & y_b & z_b \end{bmatrix} = \begin{bmatrix} x_m + L \sin \psi \cos \phi & y_m + L \sin \psi \sin \phi & z_m + L \cos \psi \end{bmatrix} \quad (4.3.5)$$

ここで、 $\theta_{l,r} \equiv \theta_l, \theta_r$  とする。

次に、全体としての移動を計算する為、並進方向の運動エネルギー (並進運動エネルギー, translation kinetic energy)  $T_1$  を求める。並進運動エネルギーは、一般に次式で与えられる。ここで、 $m$  は質量、 $x$  は変位である。

$$\begin{aligned} T_T &= \frac{1}{2}mv^2 \\ &= \frac{1}{2}m\dot{x}^2 \end{aligned} \quad (4.3.6)$$

表 4.3 NXTway-GS の物理パラメータ

| 定数       | 数値                 | 単位                    | 意味                     |
|----------|--------------------|-----------------------|------------------------|
| $g$      | 9.81               | [m/sec <sup>2</sup> ] | 重力加速度                  |
| $m$      | 0.03               | [kg]                  | 車輪 1 本あたりの重量 [40, 41]  |
| $R$      | 0.04               | [m]                   | 車輪半径                   |
| $J_w$    | $mR^2/2$           | [kgm <sup>2</sup> ]   | 車輪の慣性モーメント             |
| $M$      | 0.635              | [kg]                  | 車体重量 [40, 41]          |
| $W$      | 0.14               | [m]                   | 車体幅                    |
| $D$      | 0.04               | [m]                   | 車体奥行き                  |
| $H$      | 0.144              | [m]                   | 車体高さ                   |
| $L$      | $H/2$              | [m]                   | 車輪中心から車体重心までの距離        |
| $J_\psi$ | $ML^2/3$           | [kgm <sup>2</sup> ]   | 車体慣性モーメント (ピッチ)        |
| $J_\phi$ | $M(W^2 + D^2)/12$  | [kgm <sup>2</sup> ]   | 車体慣性モーメント (ヨー)         |
| $J_m$    | $1 \times 10^{-5}$ | [kgm <sup>2</sup> ]   | DC モータ慣性モーメント [34]     |
| $R_m$    | 6.69               | [Ω]                   | DC モータ抵抗 [42]          |
| $K_b$    | 0.468              | [V·sec/rad.]          | DC モータ逆起電力定数 [42]      |
| $K_t$    | 0.317              | [N·m/A]               | DC モータトルク定数 [42]       |
| $n$      | 1                  | [1]                   | ギア比 [34]               |
| $f_m$    | 0.0022             | [1]                   | 車体と DC モータの間の摩擦係数 [34] |
| $f_w$    | 0                  | [1]                   | 車輪と路面間の摩擦係数 [34]       |

これに従うと、 $T_1$  は次の様になる。

$$T_1 = \frac{1}{2}m(\dot{x}_l^2 + \dot{y}_l^2 + \dot{z}_l^2) + \frac{1}{2}m(\dot{x}_r^2 + \dot{y}_r^2 + \dot{z}_r^2) + \frac{1}{2}M(\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2) \quad (4.3.7)$$

続いて、全体としての回転を計算する為、回転方向の運動エネルギー (回転運動エネルギー, rotational kinetic energy) $T_2$  を求める。回転運動エネルギーは、一般に次式 (4.3.8) で与えられる。ここで、 $J$  は慣性モーメント、 $\theta$  は角度である。

$$\begin{aligned} T_R &= \frac{1}{2}J\omega^2 \\ &= \frac{1}{2}J\dot{\omega}^2 \end{aligned} \quad (4.3.8)$$

これに従うと、 $T_2$  は次の様になる。

$$T_2 = \frac{1}{2}J_w\dot{\theta}_l^2 + \frac{1}{2}J_w\dot{\theta}_r^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2}J_\phi\dot{\phi}^2 + \frac{1}{2}n^2J_m(\dot{\theta}_l - \dot{\psi})^2 + \frac{1}{2}n^2J_m(\dot{\theta}_r - \dot{\psi})^2 \quad (4.3.9)$$

位置エネルギー  $U$  を求める。位置エネルギーは、一般に次式で与えられる。ここで、 $h$  は高さである。

$$U_P = mgh \quad (4.3.10)$$

これに従うと、 $U$  は次の様になる。

$$U = mgz_l + mgz_r + Mgz_b \quad (4.3.11)$$

それでは、ラグランジュ方程式 (Lagrange equation)<sup>[43]</sup> を導く。

ラグランジアン  $L$  を求めると、

$$L = T_1 + T_2 - U \quad (4.3.12)$$

となる. 一般化座標として

- $\theta$  : 左右車輪の平均回転角度
- $\psi$  : 車体の傾斜角度 (ピッチ角度)
- $\phi$  : 車体の平面回転角度 (ヨー角度)

の 3 変数を用いると, ラグランジュ方程式は次の様に与えられる.

$$\frac{d}{dt} \left( \frac{\partial}{\partial \dot{\theta}} L \right) - \frac{\partial}{\partial \theta} L = F_{\theta} \quad (4.3.13)$$

$$\frac{d}{dt} \left( \frac{\partial}{\partial \dot{\psi}} L \right) - \frac{\partial}{\partial \psi} L = F_{\psi} \quad (4.3.14)$$

$$\frac{d}{dt} \left( \frac{\partial}{\partial \dot{\phi}} L \right) - \frac{\partial}{\partial \phi} L = F_{\phi} \quad (4.3.15)$$

上式 3 本の左辺を計算する. 式 (4.3.13) については

$$\begin{aligned} \frac{\partial}{\partial \theta} L &= (2m + M) R^2 \dot{\theta} + (2m + M) R^2 \dot{\theta} \sin^2 \gamma + MLR \dot{\psi} \cos \psi - MLR \sin \gamma \dot{\psi} \sin \psi + 2J_w \dot{\theta} + 2n^2 J_m (\dot{\theta} - \dot{\psi}) \\ \frac{d}{dt} \left( \frac{\partial}{\partial \dot{\theta}} L \right) &= (2m + M) R^2 \ddot{\theta} + (2m + M) R^2 \ddot{\theta} \sin^2 \gamma + MLR \ddot{\psi} \cos \psi - MLR \dot{\psi}^2 \sin \psi \\ &\quad - MLR \sin \gamma \ddot{\psi} \sin \psi - MLR \sin \gamma \dot{\psi}^2 \cos \psi + 2J_w \ddot{\theta} + 2n^2 J_m (\ddot{\theta} - \ddot{\psi}) \\ \frac{\partial}{\partial \theta} L &= (2m + M) R^2 \theta \dot{\phi}^2 + MLR \dot{\phi}^2 \sin \psi - Rg (2m + M) \sin \gamma \end{aligned}$$

となるので,

$$\begin{aligned} & \left[ (2m + M) R^2 + (2m + M) R^2 \sin^2 \gamma + 2J_w + 2n^2 J_m \right] \ddot{\theta} - \left[ MLR \sin \gamma \sin \psi - MLR \cos \psi - 2n^2 J_m \right] \ddot{\psi} \\ & - \left[ MLR \sin \gamma \cos \psi + MLR \sin \psi \right] \dot{\psi}^2 - \left[ (2m + M) R^2 \theta + MLR \sin \psi \right] \dot{\phi}^2 - Rg (2m + M) \sin \gamma = F_{\theta} \end{aligned} \quad (4.3.16)$$

となる.

次に式 (4.3.14) については

$$\begin{aligned} \frac{\partial}{\partial \psi} L &= MLR \dot{\theta} \cos \psi + ML^2 \dot{\psi} - MLR \dot{\theta} \sin \gamma \sin \psi + J_{\psi} \dot{\psi} - 2n^2 J_m (\dot{\theta} - \dot{\psi}) \\ \frac{d}{dt} \left( \frac{\partial}{\partial \dot{\psi}} L \right) &= MLR \ddot{\theta} \cos \psi - MLR \dot{\theta} \dot{\psi} \sin \psi + ML^2 \ddot{\psi} - MLR \ddot{\theta} \sin \gamma \sin \psi - MLR \dot{\theta} \sin \gamma \cos \psi + J_{\psi} \ddot{\psi} - 2n^2 J_m (\ddot{\theta} - \ddot{\psi}) \\ \frac{\partial}{\partial \psi} L &= -MLR \dot{\theta} \dot{\psi} \sin \psi + MLR \theta \dot{\phi}^2 \cos \psi + ML^2 \sin \psi \cos \psi \dot{\phi}^2 - MLR \dot{\theta} \sin \gamma \dot{\psi} \cos \psi + MgL \sin \psi \end{aligned}$$

となるので,

$$\begin{aligned} & \left[ MLR \cos \psi - MLR \sin \gamma \sin \psi - 2n^2 J_m \right] \ddot{\theta} + \left[ J_{\psi} + 2n^2 J_m + ML^2 \right] \ddot{\psi} - MgL \sin \psi \\ & - \left[ MLR \sin \gamma \cos \psi + MLR \dot{\psi} \sin \psi - MLR \dot{\psi} \sin \psi - MLR \sin \gamma \dot{\psi} \cos \psi \right] \dot{\theta} \\ & - \left[ MRL \theta \cos \psi + ML^2 \sin \psi \cos \psi \right] \dot{\phi}^2 = F_{\psi} \end{aligned} \quad (4.3.17)$$

となる.

続いて, 式 (4.3.15) について見る.  $T_2$  を計算の都合上, 次の様にまとめる.

$$T_2 = \frac{1}{2} J_{\psi} \dot{\psi}^2 + \frac{1}{2} J_{\phi} \dot{\phi}^2 + \frac{1}{2} J_w (\dot{\theta}_l^2 + \dot{\theta}_r^2) + \frac{1}{2} n^2 J_m \left\{ (\dot{\theta}_l - \dot{\psi})^2 + (\dot{\theta}_r - \dot{\psi})^2 \right\} \quad (4.3.9)$$



ここで,

$$(\dot{\theta}_l - \dot{\psi})^2 + (\dot{\theta}_r - \dot{\psi})^2 = \dot{\theta}_l^2 + \dot{\theta}_r^2 - 4\dot{\psi}\dot{\theta} + 2\dot{\psi}^2$$

であるので, 式 (4.3.1) の  $\phi$  を時間微分し二乗した式

$$\dot{\phi}^2 = \frac{R^2}{W^2} (\dot{\theta}_r - \dot{\theta}_l)^2$$

を展開し整理すると

$$\frac{W^2}{R^2} \dot{\phi}^2 + 2\dot{\theta}_r\dot{\theta}_l = \dot{\theta}_r^2 + \dot{\theta}_l^2$$

となる. これを  $T_2$  に代入する事で

$$T_2 = \frac{1}{2} J_\psi \dot{\psi}^2 + \frac{1}{2} J_\phi \dot{\phi}^2 + \frac{1}{2} J_w \left( \frac{W^2}{R^2} \dot{\phi}^2 + 2\dot{\theta}_r\dot{\theta}_l \right) + \frac{1}{2} n^2 J_m \left( \frac{W^2}{R^2} \dot{\phi}^2 + 2\dot{\theta}_r\dot{\theta}_l - 4\dot{\psi}\dot{\theta} + 2\dot{\psi}^2 \right) \quad (4.3.18)$$

となる. これを使い左辺を計算すると

$$\begin{aligned} \frac{\partial}{\partial \dot{\phi}} L &= \frac{1}{2} m W^2 \dot{\phi} + 2m R^2 \theta^2 \dot{\phi} + M R^2 \theta^2 \dot{\phi} + M L^2 \dot{\phi} \sin^2 \psi \\ &\quad + 2MLR\dot{\theta}\dot{\phi} \sin \psi + J_\phi \dot{\phi} + \frac{1}{2} (J_w + n^2 J_m) \frac{W^2}{R^2} \dot{\phi} \\ \frac{d}{dt} \left( \frac{\partial}{\partial \dot{\phi}} L \right) &= \frac{1}{2} m W^2 \ddot{\phi} + 2m R^2 \theta^2 \ddot{\phi} + 4m R^2 \theta \dot{\theta} \dot{\phi} \\ &\quad + M R^2 \theta^2 \ddot{\phi} + 2M R^2 \theta \dot{\theta} \dot{\phi} + M L^2 \ddot{\phi} \sin^2 \psi + 2M L^2 \dot{\phi} \dot{\psi} \sin \psi \cos \psi \\ &\quad + 2MLR\dot{\theta}\dot{\phi} \sin \psi + 2MLR\ddot{\theta} \sin \psi + 2MLR\dot{\theta}\dot{\psi} \cos \psi \\ &\quad + J_\phi \ddot{\phi} + \frac{1}{2} (J_w + n^2 J_m) \frac{W^2}{R^2} \ddot{\phi} \\ \frac{\partial}{\partial \phi} L &= 0 \end{aligned}$$

となるので, ラグランジュ方程式 (4.3.15) は

$$\begin{aligned} \left[ \frac{1}{2} m W^2 + J_\phi + \frac{1}{2} \frac{W^2}{R^2} (J_w + n^2 J_m) + (2m + M) R^2 \theta^2 + 2MLR\theta \sin \psi + M L^2 \sin^2 \psi \right] \ddot{\phi} \\ + 2 \left[ (2m + M) R^2 \theta \dot{\theta} + M L^2 \dot{\psi} \sin \psi \cos \psi + MLR (\dot{\theta} \sin \psi + \dot{\psi} \cos \psi) \right] \dot{\phi} = F_\phi \quad (4.3.19) \end{aligned}$$

となる.

さて, 次に右辺の一般化力を求める.

DC モータの駆動トルクと粘性摩擦を考慮すると, 次式で一般化力は示される.

$$[F_\theta, F_\psi, F_\phi] = \left[ \frac{1}{2} (F_l + F_r), F_\psi, \frac{1}{2} (F_l + F_r) \right] \quad (4.3.20)$$

$$F_l = n K_t i_l + f_m (\dot{\psi} - \dot{\theta}_l) - f_w \dot{\theta}_l \quad (4.3.21)$$

$$F_r = n K_t i_r + f_m (\dot{\psi} - \dot{\theta}_r) - f_w \dot{\theta}_r \quad (4.3.22)$$

$$F_\psi = -n K_t i_l - n K_t i_r - f_m (\dot{\psi} - \dot{\theta}_l) - f_m (\dot{\psi} - \dot{\theta}_r) \quad (4.3.23)$$

尚,  $i_{l,r}$  は DC モータに流れる電流である. DC モータのアクチュエータは PWM 制御の為, 電流制御は不可能である. よって, 電流から電圧の式を求める事を考える. DC モータの方程式は, 一般に次式で与えられる.

$$L_m \dot{i}_{l,r} = v_{l,r} + K_b (\dot{\psi} - \dot{\theta}_{l,r}) - R_m i_{l,r} \quad (4.3.24)$$

DC モータのインダクタンスが十分小さいとし、その効果を見捨ると、電流は

$$i_{l,r} = \frac{v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r})}{R_m} \quad (4.3.25)$$

となる。これを式 (4.3.20), (4.3.21), (4.3.22), (4.3.23) に代入する事により、一般化力は電圧を用いて表す事が出来る。

$$F_\theta = \frac{\alpha}{2}(v_l + v_r) - (\beta + f_w)\dot{\theta} + \beta\dot{\psi} \quad (4.3.26)$$

$$F_\psi = -\alpha(v_l + v_r) + 2\beta\dot{\theta} - 2\beta\dot{\psi} \quad (4.3.27)$$

$$F_\phi = \frac{R}{W}\alpha(v_r - v_l) - \left(\beta + \frac{W}{R}f_w\right)\dot{\phi} \quad (4.3.28)$$

ここで、 $\alpha, \beta$  は以下で与えられる。

$$\alpha = \frac{nK_t}{R_m} \quad (4.3.29)$$

$$\beta = \frac{nK_t K_b}{R_m} + f_m \quad (4.3.30)$$

### 4.3.3 状態方程式の導出

それでは、前節で求めたラグランジュ方程式より、状態方程式を導出する。

自走式二輪型倒立振子は、運動方程式に非線型関数である三角関数が含まれている事から判る様に、非線型システム (non-linear system) である。この為、状態方程式は

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \quad (4.3.31)$$

となる。ここで

$$\mathbf{x} = [\theta \quad \psi \quad \dot{\theta} \quad \dot{\psi}]^T \quad (4.3.32)$$

で、 $\mathbf{f}(\mathbf{x}, u)$  は非線型なベクトル関数である。

この非線型システムのままでオブザーバやコントローラを設計するのは困難である。一般的には、主に制御を行う動作点の周囲で非線型システムを線型システムに近似し (線型化, linearlization), 線型化された状態方程式に対し、状態オブザーバや安定化コントローラを設計する。つまり、非線型システムについては、シミュレータには非線型モデルを用いて構成し、状態オブザーバや安定化コントローラについては線型近似されたモデルに対して設計したものをを用いる。

ここでは、鉛直近傍で線型化する。つまり、 $\psi \rightarrow 0$  とする事で、

$$\sin \psi \rightarrow \psi$$

$$\cos \psi \rightarrow 1$$

$$\dot{\psi}^2, \sin^2 \gamma, \sin \gamma \sin \psi \text{ 等 2 次項は無視する}$$

とする。これにより、運動方程式 (4.3.16), (4.3.17), (4.3.19) は次式の通りとなる。

$$\left[(2m + M)R^2 + 2J_w + 2n^2 J_m\right]\ddot{\theta} + (MLR - 2n^2 J_m)\ddot{\psi} - Rg(M + 2m)\sin \gamma = F_\theta \quad (4.3.33)$$

$$(MLR - 2n^2 J_m)\ddot{\theta} + (ML^2 + J_\psi + 2n^2 J_m)\ddot{\psi} - MgL\psi = F_\psi \quad (4.3.34)$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}(J_w + n^2 J_m)\right]\ddot{\phi} = F_\phi \quad (4.3.35)$$

式 (4.3.33) および式 (4.3.34) は  $\theta$  と  $\psi$  の連立方程式, 式 (4.3.35) は  $\phi$  のみの式になっている事が判る. そこで, 式 (4.3.33) と式 (4.3.34) を行列で表現すると, 次式の様になる.

$$\mathbf{E} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \mathbf{F} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \mathbf{G} \begin{bmatrix} \theta \\ \psi \end{bmatrix} = \mathbf{H} \begin{bmatrix} v_l \\ v_r \end{bmatrix} + \begin{bmatrix} Rg(M+2m) \sin \gamma \\ 0 \end{bmatrix} \quad (4.3.36)$$

ここで

$$\begin{aligned} \mathbf{E} &= \begin{bmatrix} (2m+M)R^2 + 2J_w + 2n^2J_m & MLR - 2n^2J_m \\ MLR - 2n^2J_m & ML^2 + J_\psi + 2n^2J_m \end{bmatrix} \\ \mathbf{F} &= \begin{bmatrix} \beta + f_w & -\beta \\ -2\beta & 2\beta \end{bmatrix} \\ \mathbf{G} &= \begin{bmatrix} 0 & 0 \\ 0 & -MgL \end{bmatrix} \\ \mathbf{H} &= \begin{bmatrix} \frac{\alpha}{2} & \frac{\alpha}{2} \\ -\alpha & -\alpha \end{bmatrix} \end{aligned} \quad (4.3.37)$$

である. そして  $\gamma$  は, 傾斜路の進行方向傾斜角度である. 本研究に於いては  $\gamma = 0.0$  とする.

また, 式 (4.3.35) は

$$K\ddot{\phi} + I\dot{\phi} = J(v_r - v_l) \quad (4.3.38)$$

$$I = \beta + \frac{W}{R}f_w$$

$$J = \frac{R}{W}\alpha$$

$$K = \frac{1}{2}mW^2 + j_\phi + \frac{W^2}{2R^2}(J_w + n^2J_m)$$

$$(4.3.39)$$

となる.

式 (4.3.36) を  $\mathbf{E} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}^T$  について整理すると,

$$\mathbf{E} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = -\mathbf{F} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \mathbf{G} \begin{bmatrix} \theta \\ \psi \end{bmatrix} + \mathbf{H} \begin{bmatrix} v_l \\ v_r \end{bmatrix} + \begin{bmatrix} Rg(M+2m) \sin \gamma \\ 0 \end{bmatrix} \quad (4.3.40)$$

となるので, 両辺に  $\mathbf{E}^{-1}$  を左から掛けると,  $A_{ij}$  を行列  $\mathbf{A}$  の  $(i, j)$  成分とする事で

$$\begin{aligned} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= -\mathbf{E}^{-1}\mathbf{F} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \mathbf{E}^{-1}\mathbf{G} \begin{bmatrix} \theta \\ \psi \end{bmatrix} + \mathbf{E}^{-1}\mathbf{H} \begin{bmatrix} v_l \\ v_r \end{bmatrix} + \mathbf{E}^{-1} \begin{bmatrix} Rg(M+2m) \sin \gamma \\ 0 \end{bmatrix} \\ &= -\frac{1}{E} \begin{bmatrix} E_{22} & -E_{12} \\ -E_{21} & E_{11} \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \frac{1}{E} \begin{bmatrix} E_{22} & -E_{12} \\ -E_{21} & E_{11} \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \theta \\ \psi \end{bmatrix} \\ &\quad + \frac{1}{E} \begin{bmatrix} E_{22} & -E_{12} \\ -E_{21} & E_{11} \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} + \frac{1}{E} \begin{bmatrix} E_{22} & -E_{12} \\ -E_{21} & E_{11} \end{bmatrix} \begin{bmatrix} Rg(M+2m) \sin \gamma \\ 0 \end{bmatrix} \\ &= -\frac{1}{E} \begin{bmatrix} E_{22}F_{11} - E_{12}F_{21} & E_{22}F_{12} - E_{12}F_{22} \\ -E_{21}F_{11} + E_{11}F_{21} & -E_{21}F_{12} + E_{11}F_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \frac{1}{E} \begin{bmatrix} 0 & -E_{12}G_{22} \\ 0 & E_{11}G_{22} \end{bmatrix} \begin{bmatrix} \theta \\ \psi \end{bmatrix} \\ &\quad + \frac{1}{E} \begin{bmatrix} E_{22}H_{11} - E_{12}H_{21} & E_{22}H_{12} - E_{12}H_{22} \\ -E_{21}H_{11} + E_{11}H_{21} & -E_{21}H_{12} + E_{11}H_{22} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} + \frac{1}{E} \begin{bmatrix} E_{22}Rg(M+2m) \sin \gamma \\ -E_{21}Rg(M+2m) \sin \gamma \end{bmatrix} \end{aligned} \quad (4.3.41)$$

となる.

次に, 状態方程式を立てるにあたり, 状態量と操作入力を考える. ここでは, 状態量  $\mathbf{x}_1, \mathbf{x}_2$  および操作入力  $\mathbf{u}$  として, 以下の変数を用いる事とする.

$$\mathbf{x}_1 = [\theta \quad \psi \quad \dot{\theta} \quad \dot{\psi}]^T \quad (4.3.42)$$

$$\mathbf{x}_2 = [\phi \quad \dot{\phi}]^T \quad (4.3.43)$$

$$\mathbf{u} = [v_l \quad v_r]^T \quad (4.3.44)$$

式 (4.3.36), (4.3.40) より, 自走式二輪型倒立振り子の状態方程式を求める.

$$\begin{aligned} \ddot{\theta} &= \frac{1}{E} E_{12} G_{22} \psi - \frac{1}{E} (E_{22} F_{11} - E_{12} F_{21}) \dot{\theta} - \frac{1}{E} (E_{22} F_{12} - E_{12} F_{22}) \dot{\psi} \\ &\quad + \frac{1}{E} (E_{22} H_{11} - E_{12} H_{21}) v_l + \frac{1}{E} (E_{22} H_{12} - E_{12} H_{22}) v_r + \frac{1}{E} E_{22} R g (M + 2m) \sin \gamma \\ \ddot{\psi} &= -\frac{1}{E} E_{11} G_{22} \psi + \frac{1}{E} (E_{21} F_{11} - E_{11} F_{21}) \dot{\theta} + \frac{1}{E} (E_{21} F_{12} - E_{11} F_{22}) \dot{\psi} \\ &\quad - \frac{1}{E} (E_{21} H_{11} - E_{11} H_{21}) v_l - \frac{1}{E} (E_{21} H_{12} - E_{11} H_{22}) v_r - \frac{1}{E} E_{21} R g (M + 2m) \sin \gamma \end{aligned}$$

以上の 2 式から, 次を得る.

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u} + \mathbf{S} \quad (4.3.45)$$

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u} \quad (4.3.46)$$

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{E} E_{12} G_{22} & -\frac{1}{E} (E_{22} F_{11} - E_{12} F_{21}) & -\frac{1}{E} (E_{22} F_{12} - E_{12} F_{22}) \\ 0 & -\frac{1}{E} E_{11} G_{22} & \frac{1}{E} (E_{21} F_{11} - E_{11} F_{21}) & \frac{1}{E} (E_{21} F_{12} - E_{11} F_{22}) \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{E} (E_{22} H_{11} - E_{12} H_{21}) & \frac{1}{E} (E_{22} H_{12} - E_{12} H_{22}) \\ -\frac{1}{E} (E_{21} H_{11} - E_{11} H_{21}) & -\frac{1}{E} (E_{21} H_{12} - E_{11} H_{22}) \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{E} E_{22} R g (M + 2m) \sin \gamma \\ -\frac{1}{E} E_{21} R g (M + 2m) \sin \gamma \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{I}{K} \end{bmatrix}$$

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 0 \\ -\frac{J}{K} & \frac{J}{K} \end{bmatrix}$$

## 4.3.4 コントローラ的设计

## 4.3.4.1 入出力

図 4.10 は, NXTway-GS の入出力の関係である. NXTway-GS に対する入力は, 式 (4.3.44) で示した通り, 左右の DC モータの電圧とする. ここでは, 左右の DC モータに印加する電圧は等しい ( $v_l = v_r$ ) ものとする. また出力は, センサで計測した, 左右車輪の平均角度  $\theta$  と, 車体の傾斜角速度  $\dot{\psi}$  とする.

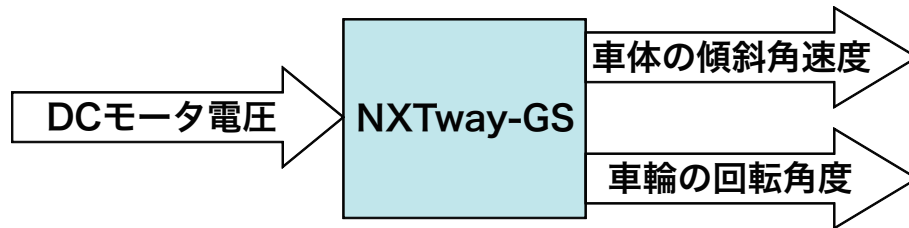


図 4.10 NXTway-GS の入出力

本研究では,  $\dot{\psi}$  を数値積分する事により  $\psi$  を求めている.

## 4.3.4.2 安定性

NXTway-GS は倒立振り子であり, 倒立状態は不安定である. 安定した倒立状態を保つには, 車体の傾斜角を検出し, NXTway-GS が倒れない様に制御する必要がある. 本研究では, 車体が倒れそうになったら, それと同じ方向に NXTway-GS を移動させて, 倒立状態を維持する様に制御する事を考える.

## 4.3.4.3 設計

■最適レギュレータによるコントローラ設計 本研究では, NXTway-GS の倒立制御手法として, サーボ制御を適用する. サーボ制御の目標値には, 左右車輪の平均回転角度  $\theta$  を選択する. 図 4.11 は, NXTway-GS 用サーボ制御器のブロック線図である.

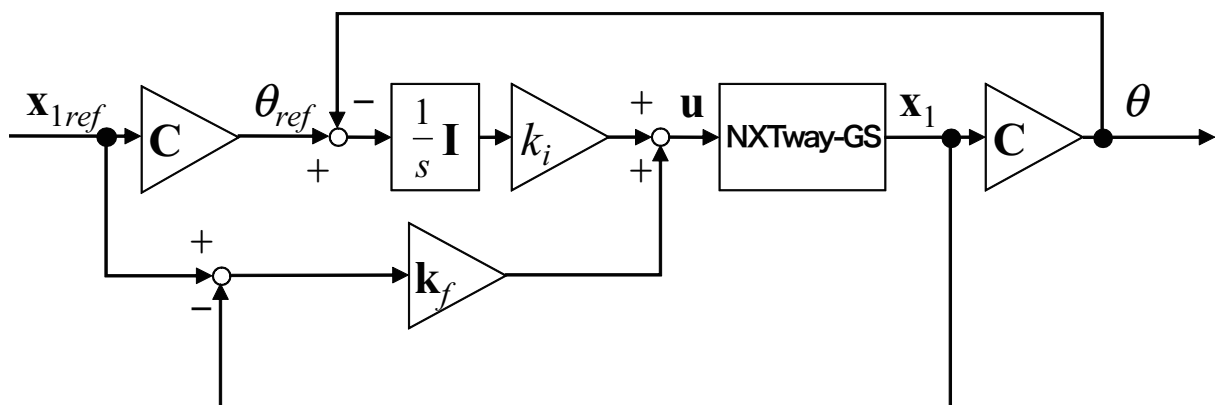


図 4.11 NXTway-GS 用サーボ制御器のブロック線図

系が可制御では無くなる為,  $\theta$  以外の状態をサーボ制御の目標値にしていない. 制御入力のゲイン計算に

は最適レギュレータを適用する. 最適レギュレータの重み行列  $\mathbf{Q}, \mathbf{R}$  には, 次の値を用いる [40, 41, 34].

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^2 \end{bmatrix} \quad (4.3.47)$$

$$\mathbf{R} = \begin{bmatrix} 1 \times 10^3 & 0 \\ 0 & 1 \times 10^3 \end{bmatrix} \quad (4.3.48)$$

ここで,  $Q_{22}$  は車体の傾斜角度に対する重み,  $Q_{55}$  は車輪の平均角度とその目標値の偏差の時間積分に対する重みである. この計算により, ゲインの数値は次の通りとなる [40, 41].

$$\mathbf{k}_f = [-0.8703034 \quad -31.997816 \quad -1.1565952 \quad -2.7887344] \quad (4.3.49)$$

本研究に於ける長期内部状態予測では, 常に倒立振子を安定化させる様, 内部状態の予測結果から行動を決定する事とする. 従って, 上式 (4.3.49) で示された状態フィードバックゲイン  $\mathbf{k}_f$  を用いて, 状態に対して決定する行動を定める (図 4.12).

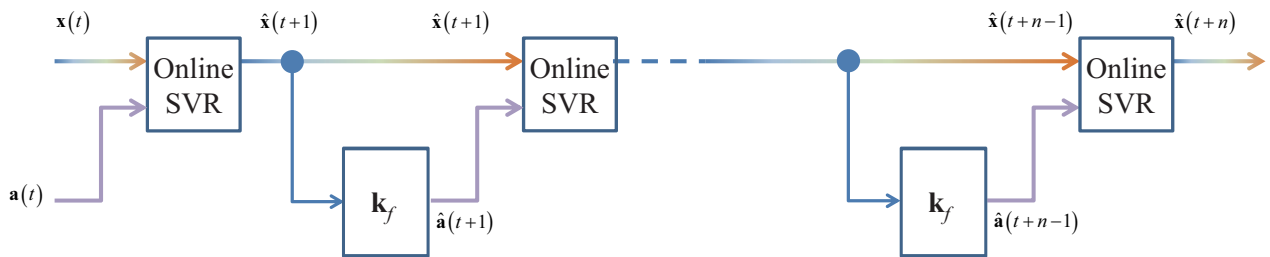


図 4.12 本研究で提案する長期内部状態予測の構成図 (図 3.4 の再掲)

# 第5章 シミュレーションによる検証実験とその検討

本章では、提案手法である状態予測の有効性を検証する為に、シミュレータを用いた検証実験を取り扱う。本検証実験では、訓練データとして、自走式二輪型倒立振子に対して倒立姿勢安定化制御を継続し実施した際の内部状態と制御入力を用いる。

また、これらの実験結果について議論する。

## 5.1 実験環境

検証実験全体に於いて使用したシミュレーション条件を、下表に示す。

表 5.1 シミュレーション条件

| パラメータ           | 数値     | 意味              | 単位     |
|-----------------|--------|-----------------|--------|
| $\lambda$       | 0.001  | 正則化パラメータ        | [1]    |
| $\epsilon$      | 0.005  | 不感帯域            | [1]    |
| $\beta$         | 30     | カーネルパラメータ       | [1]    |
| $\psi_0$        | 0.0262 | 振子の初期傾斜角度       | [rad.] |
| $\gamma_0$      | 0.0    | 傾斜路進行方向傾斜角度     | [rad.] |
| $t_s$           | 0.05   | サンプリング時刻        | [sec.] |
| $t_{d1,start}$  | 2.0    | 予測可能な外乱を与える開始時刻 | [sec.] |
| $t_{d1,finish}$ | 5.0    | 予測可能な外乱を与える終了時刻 | [sec.] |
| $t_{d2,start}$  | 2.0    | 突発的な外乱を与える開始時刻  | [sec.] |
| $t_{d2,finish}$ | 2.5    | 突発的な外乱を与える終了時刻  | [sec.] |
| $A_{d1}$        | 0.1    | 予測可能な外乱の振幅      | [V]    |
| $A_{d2}$        | 0.2    | 突発的に与える外乱の振幅    | [V]    |
| $f_{d1}$        | 10     | 予測可能な外乱の周波数     | [Hz]   |
| $N$             | 89     | 初期データセット長       | —      |

## 5.2 逐次学習と追加学習を組合わせた逐次内部状態予測

追加学習を組合わせる事で新たに定義した, 逐次内部状態予測の有効性を検討する.

本検証では, 以下のケースで制御を行った倒立振子の内部状態と制御応答を, 訓練データとして用いる.

- 外乱を与えない場合
- 時刻  $t_{d1,start}$  から  $t_{d1,finish}$  にかけて, 振幅  $A_{d1}$ [V], 周波数  $f_{d1}$ [Hz] で表される正弦波が外乱として混入した場合 (予測可能な外乱を与えた場合)
- 時刻  $t_{d2,start}$  から  $t_{d2,finish}$  にかけて, 振幅  $A_{d2}$ [V] で表される突発的な外乱が混入した場合 (突発的な外乱を与えた場合)

図 5.1 から図 5.15 で示した訓練データを  $t_s$  でサンプリングした系列を使用する.

シミュレーションではこのデータを  $t_s$  毎に逐次与え学習させると共に, 内部状態を逐次内部状態予測によって予測させる.

### 5.2.1 外乱を与えない場合の訓練データ

訓練データとして用いた倒立振子の内部状態と制御応答を図 5.1 から図 5.5 に示す.

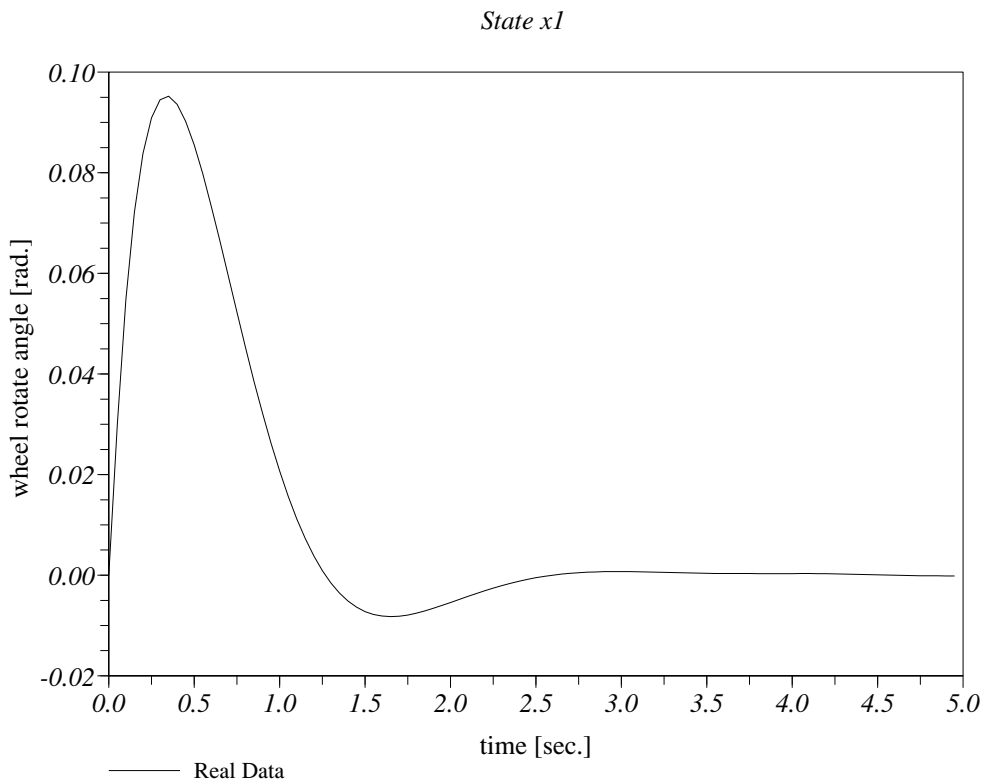
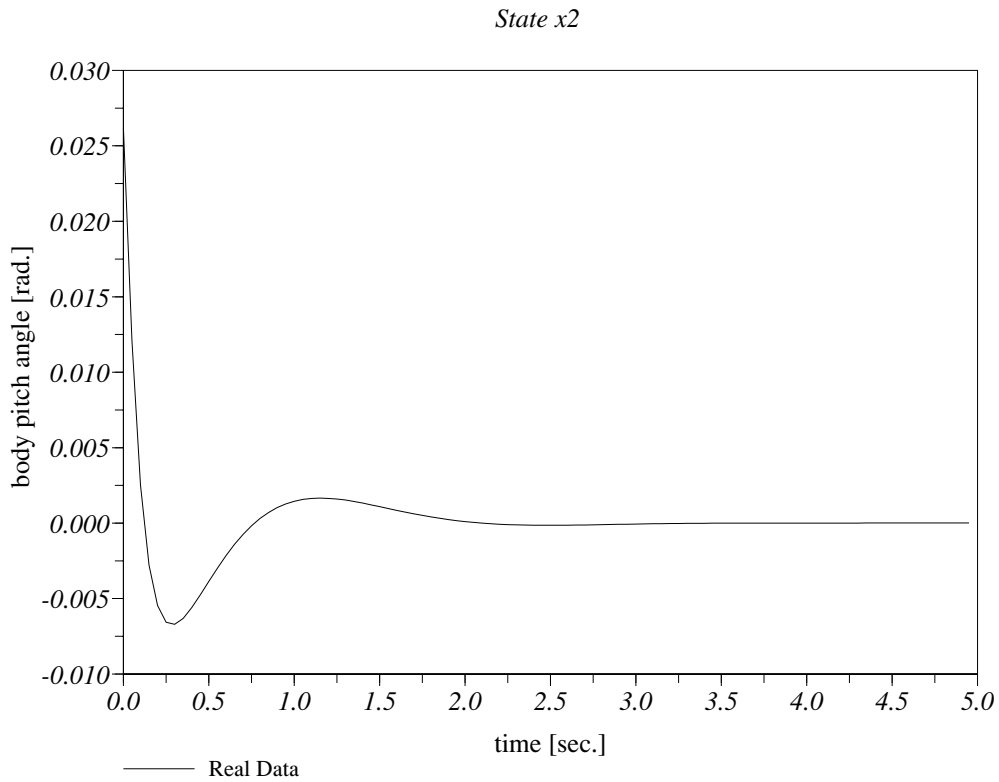
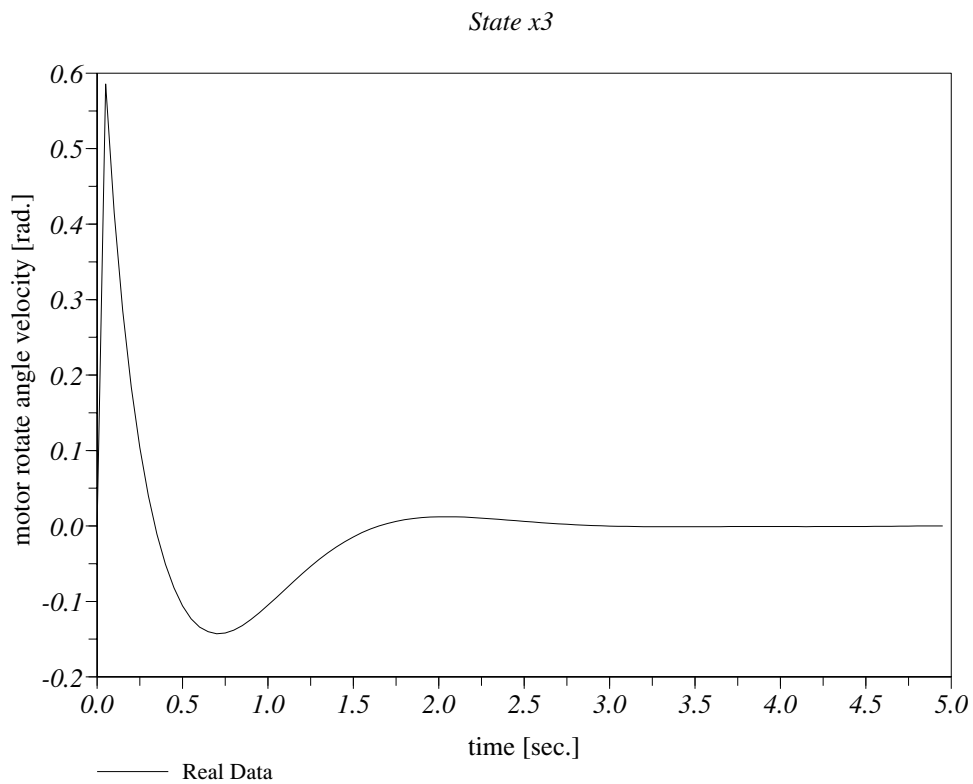


図 5.1 内部状態  $x_1$ (駆動輪回転角度  $\theta$ ) の訓練データ (外乱を与えない場合)



図 5.2 内部状態  $x_2$ (車体傾斜角度  $\psi$ ) の訓練データ (外乱を与えない場合)図 5.3 内部状態  $x_3$ (駆動輪回転角速度  $\dot{\theta}$ ) の訓練データ (外乱を与えない場合)

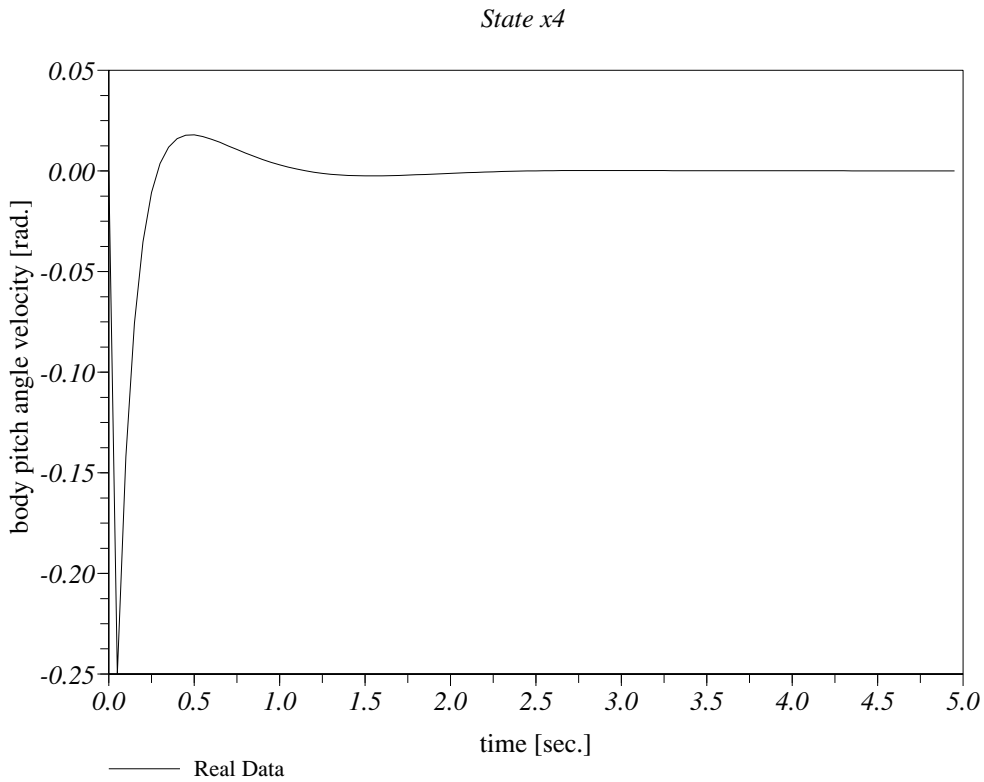


図 5.4 内部状態  $x_4$ (車体傾斜角速度  $\dot{\psi}$ ) の訓練データ (外乱を与えない場合)

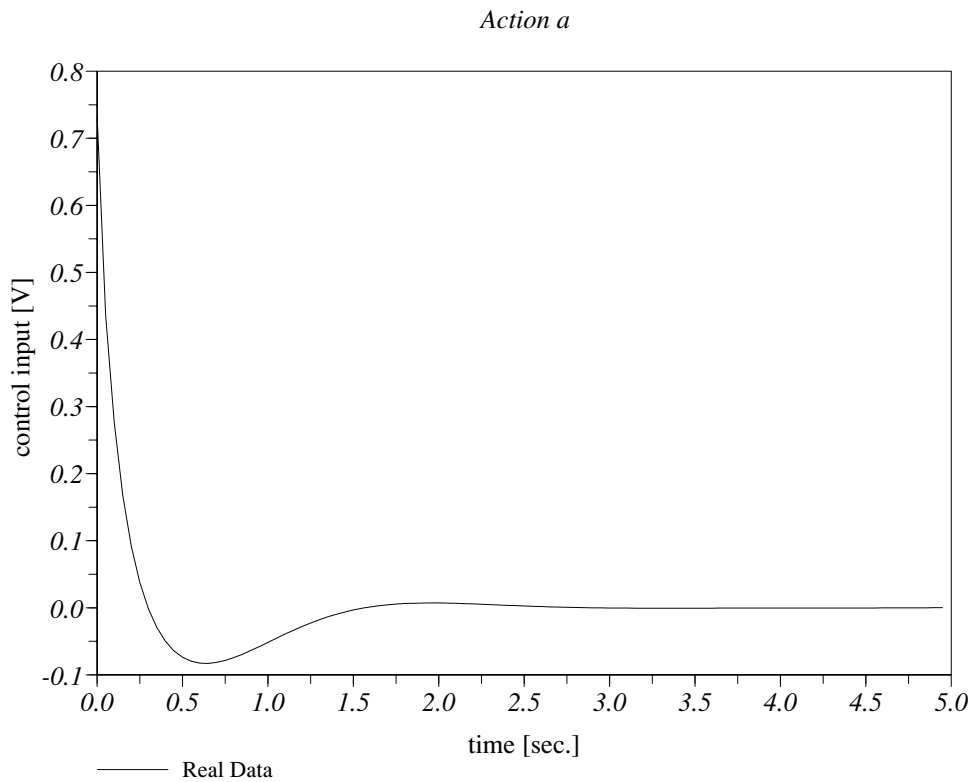


図 5.5 行動  $a$ (制御入力) の訓練データ (外乱を与えない場合)

## 5.2.2 予測可能な外乱を与えた場合の訓練データ

訓練データとして用いた倒立振子の内部状態と制御応答を図 5.6 から図 5.10 に示す。

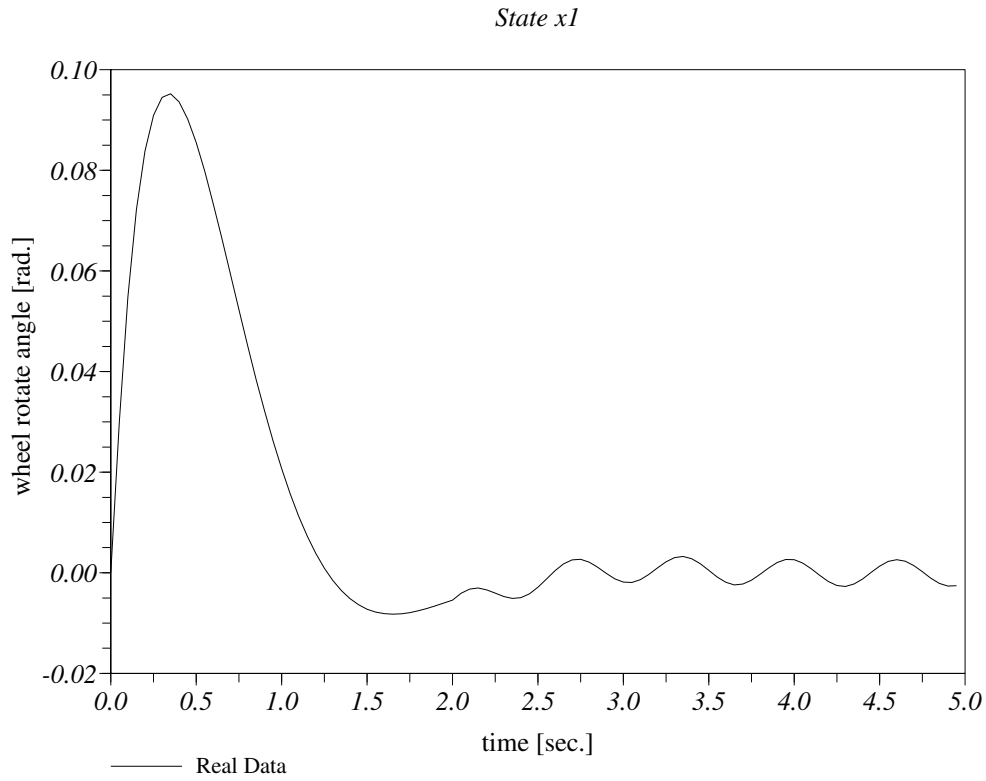


図 5.6 内部状態  $x_1$ (駆動輪回転角度  $\theta$ ) の訓練データ (予測可能な外乱を与えた場合)

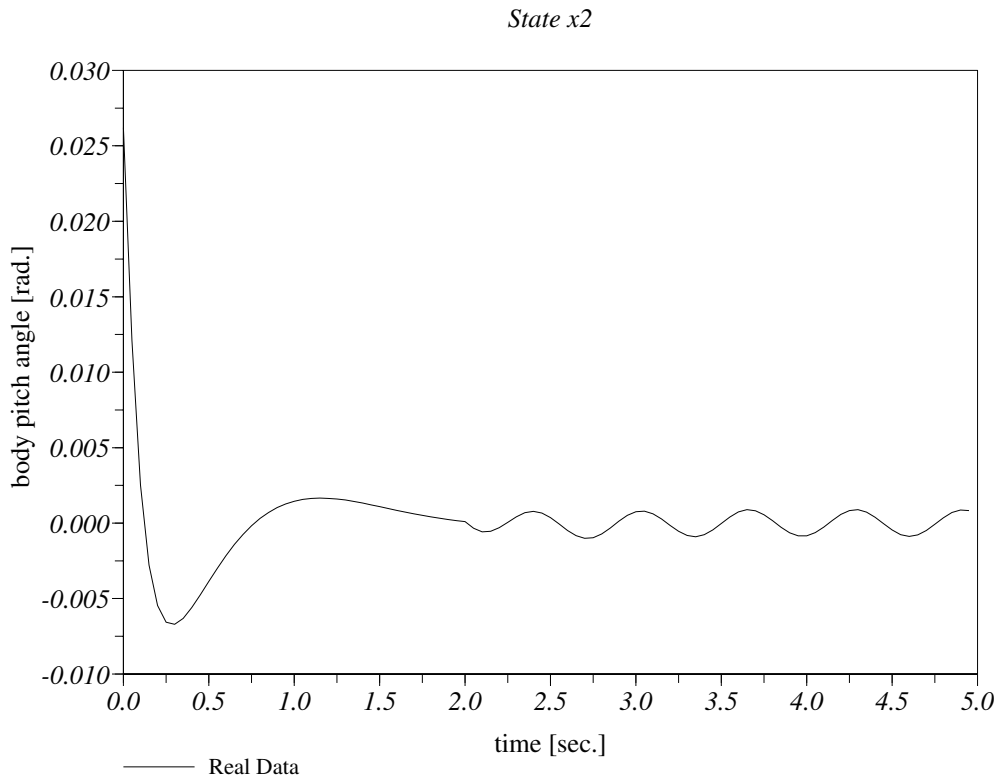


図 5.7 内部状態  $x_2$ (車体傾斜角度  $\psi$ ) の訓練データ (予測可能な外乱を与えた場合)

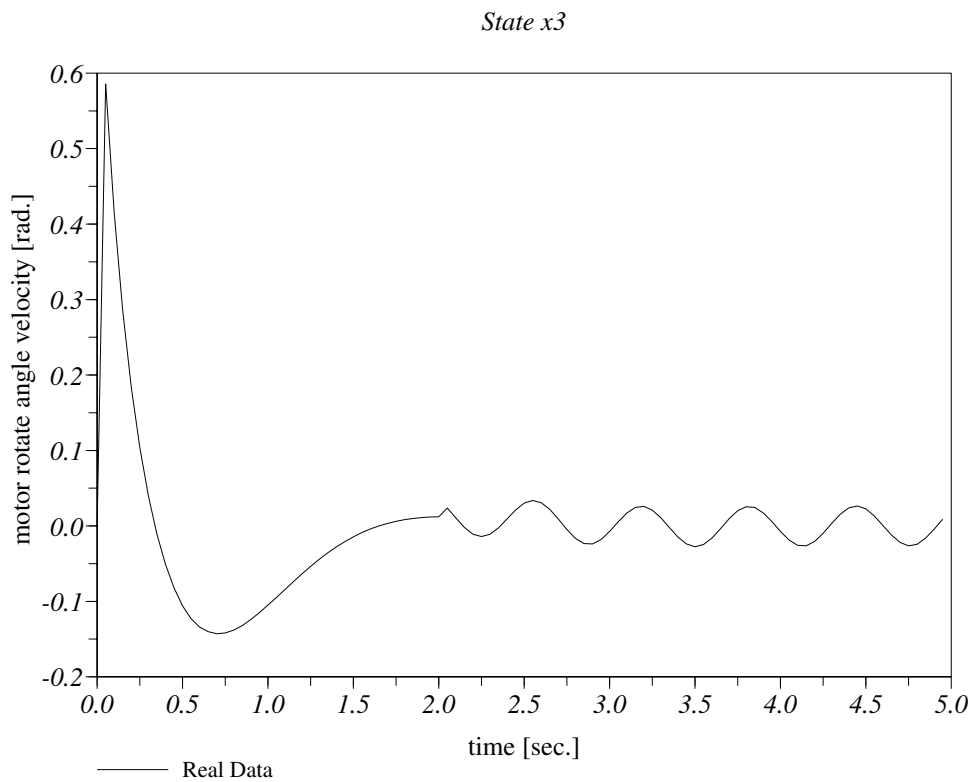
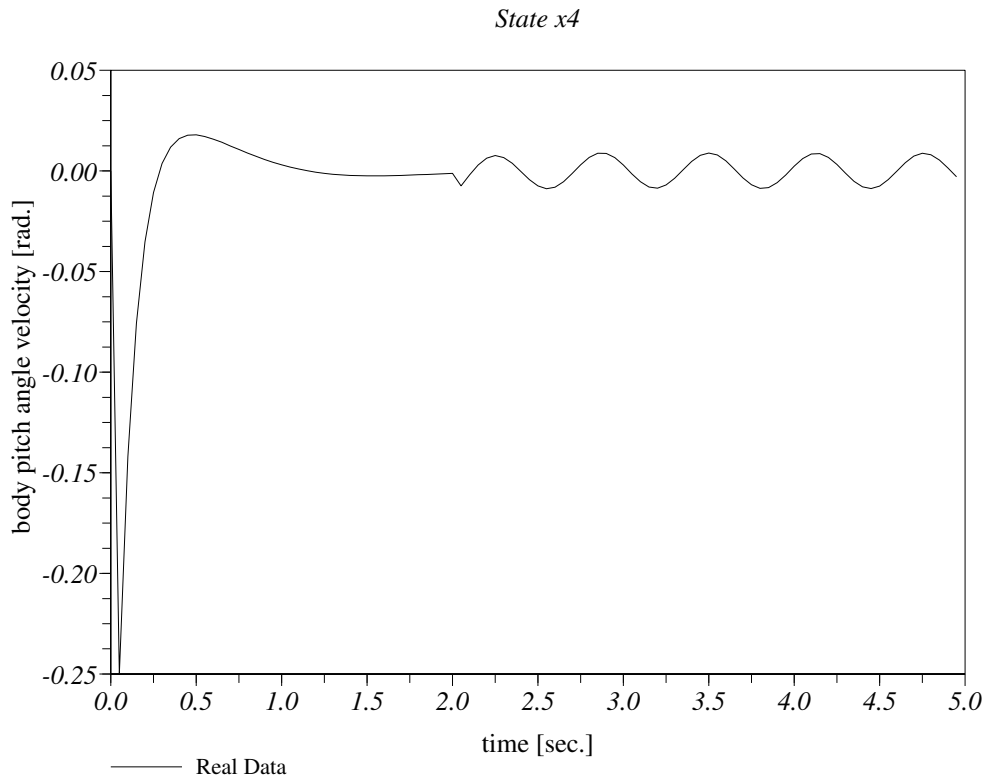
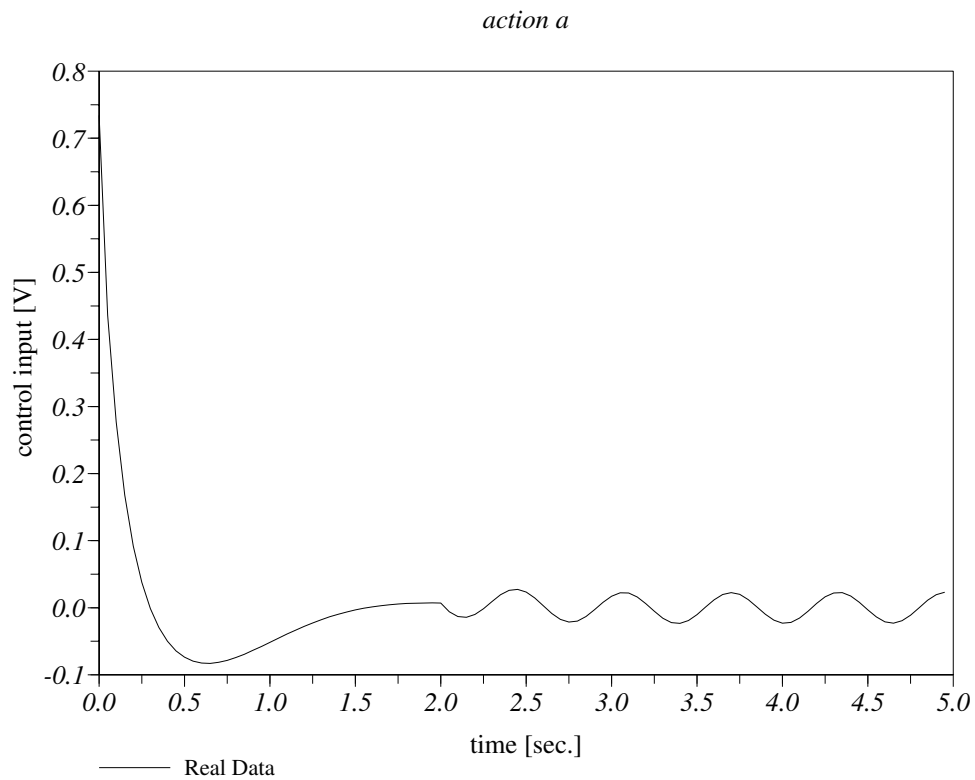


図 5.8 内部状態  $x_3$ (駆動輪回転角速度  $\dot{\theta}$ ) の訓練データ (予測可能な外乱を与えた場合)

図 5.9 内部状態  $x_4$ (車体傾斜角速度  $\dot{\psi}$ ) の訓練データ (予測可能な外乱を与えた場合)図 5.10 行動  $a$ (制御入力) の訓練データ (予測可能な外乱を与えた場合)

### 5.2.3 突発的な外乱を与えた場合の訓練データ

訓練データとして用いた倒立振子の内部状態と制御応答を図 5.11 から図 5.15 に示す.

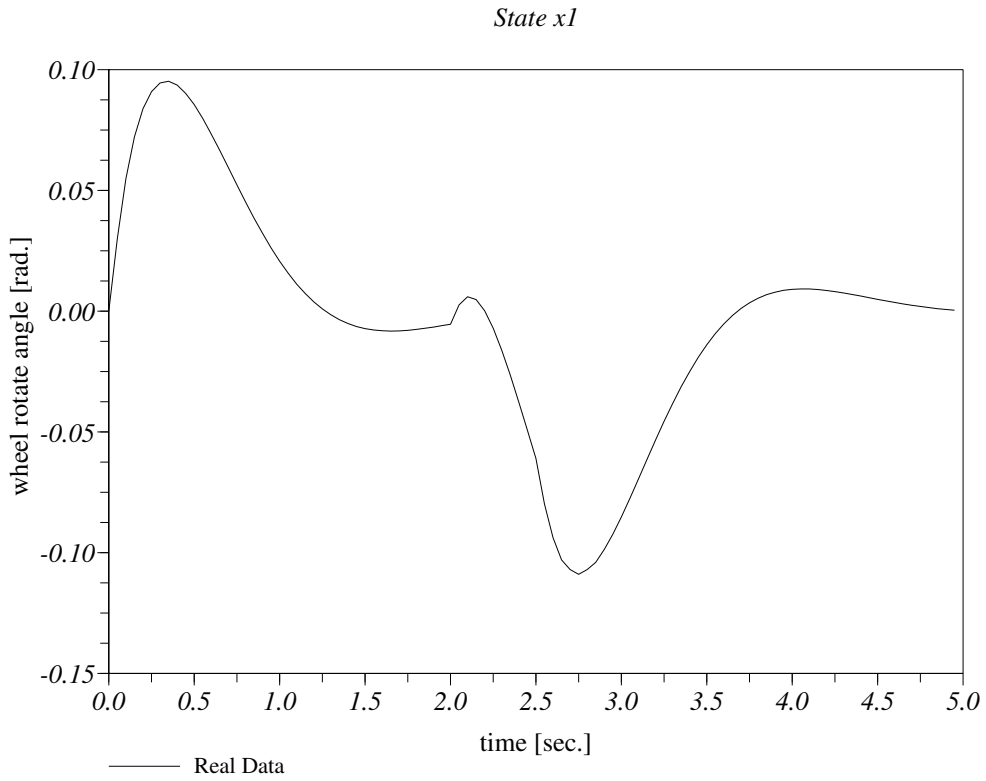
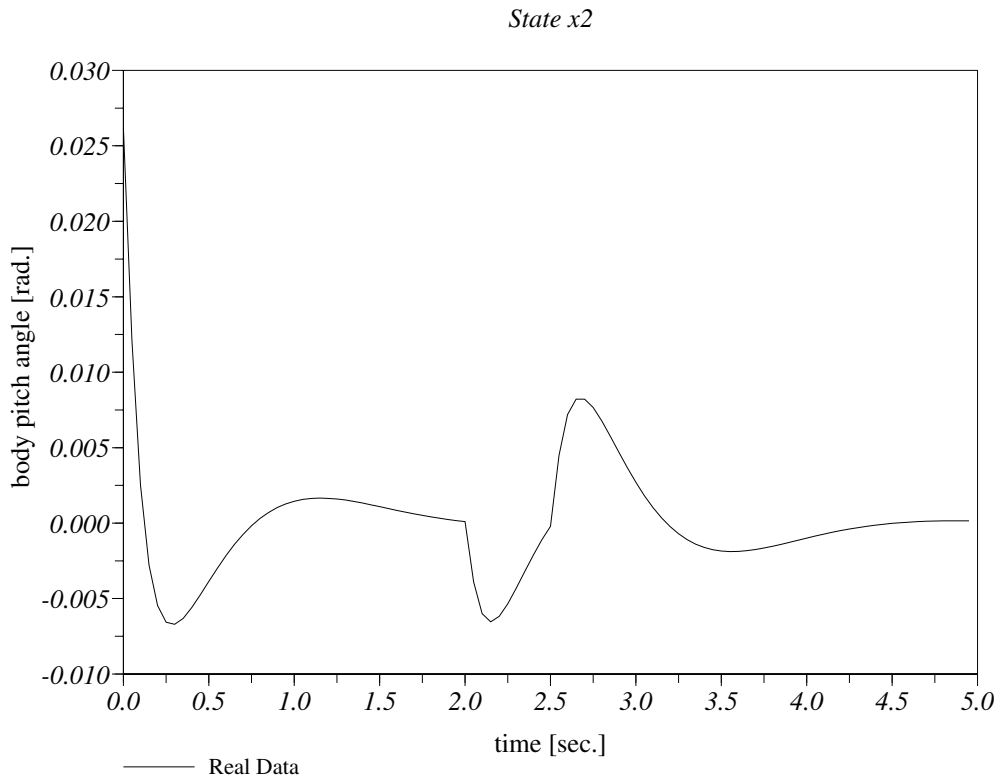
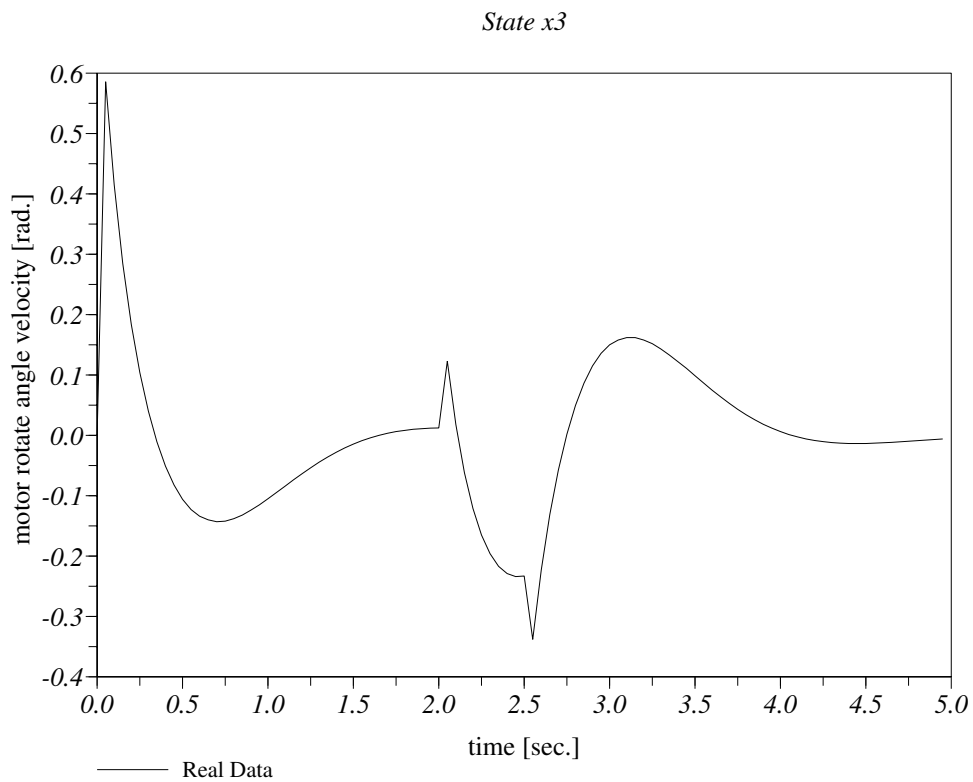


図 5.11 内部状態  $x_1$ (駆動輪回転角度  $\theta$ ) の訓練データ (突発的な外乱を与えた場合)

図 5.12 内部状態  $x_2$ (車体傾斜角度  $\psi$ ) の訓練データ (突発的な外乱を与えた場合)図 5.13 内部状態  $x_3$ (駆動輪回転角速度  $\dot{\theta}$ ) の訓練データ (突発的な外乱を与えた場合)

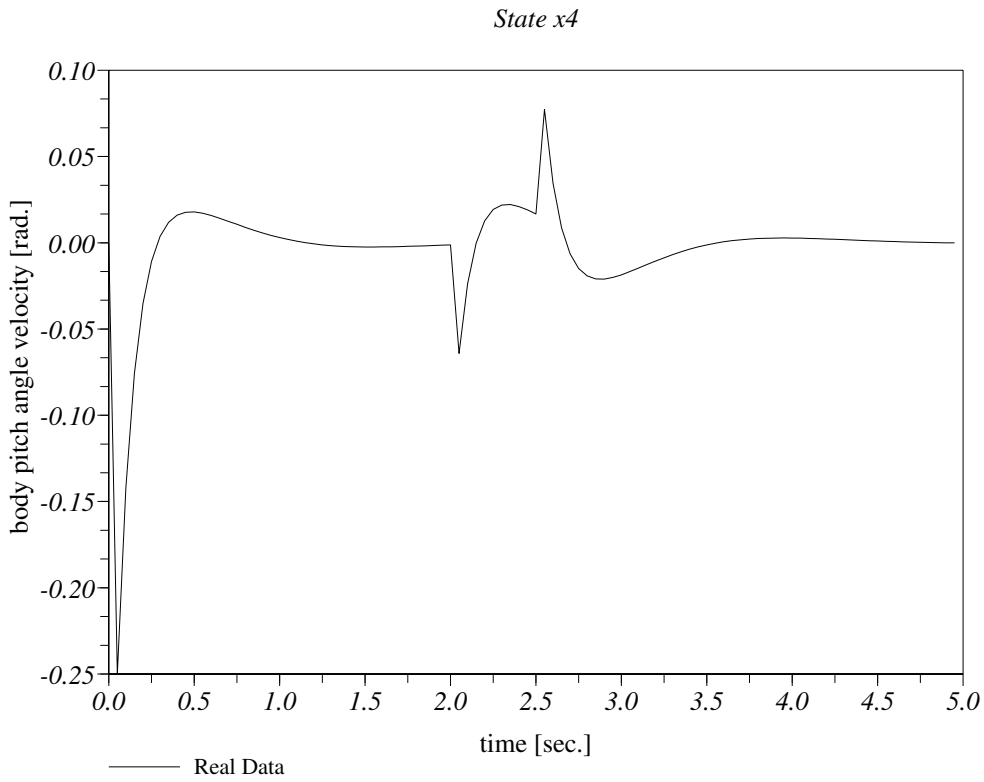


図 5.14 内部状態  $x_4$ (車体傾斜角速度  $\dot{\psi}$ ) の訓練データ (突発的な外乱を与えた場合)

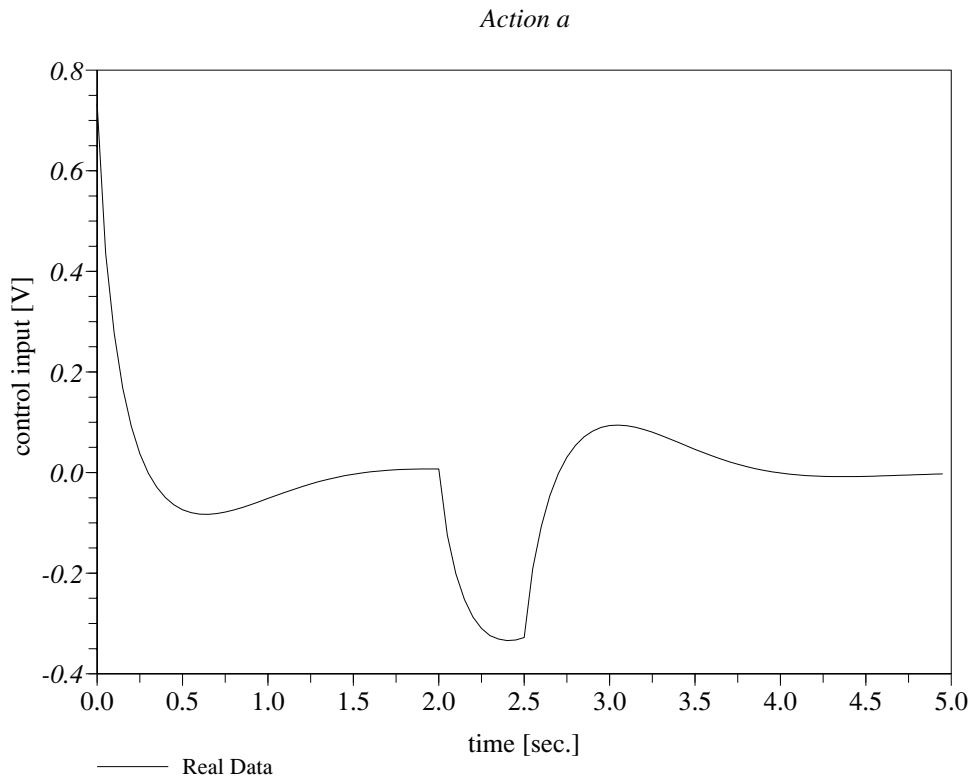


図 5.15 行動  $a$ (制御入力) の訓練データ (突発的な外乱を与えた場合)



## 5.3 逐次学習と追加学習を組合わせた長期内部状態予測

予測した内部状態から行動を決定し、長期的に内部状態を予測する事で定義した、長期内部状態予測の有効性を検討する。

本検証では、時刻  $t_{d2,start}$  から  $t_{d2,finish}$  にかけて、振幅  $A_{d2}$ [V] で表される突発的な外乱が混入した場合で制御を行った倒立振子の内部状態と制御応答を、訓練データとして用いる。図 5.1 から図 5.15 で示した訓練データを  $t_s$  でサンプリングした系列を使用する。

シミュレーションでは、このデータを  $t_s$  毎に逐次与え学習させると共に、内部状態を各サンプリング時刻に於いて長期内部状態予測によって予測させる。

### 5.3.1 突発的な外乱を与えた場合の訓練データ

訓練データとして用いた倒立振子の内部状態と制御応答を図 5.11 から図 5.15 に示す。

## 5.4 実験結果

### 5.4.1 逐次学習と追加学習を組合わせた逐次内部状態予測

逐次内部状態予測により予測した結果を, 図 5.16 から図 5.27 に示す.

#### 5.4.1.1 外乱を与えない場合の訓練データに対する逐次内部状態予測

外乱を与えなかった訓練データに対する予測結果を図 5.1 から図 5.5 に示す.

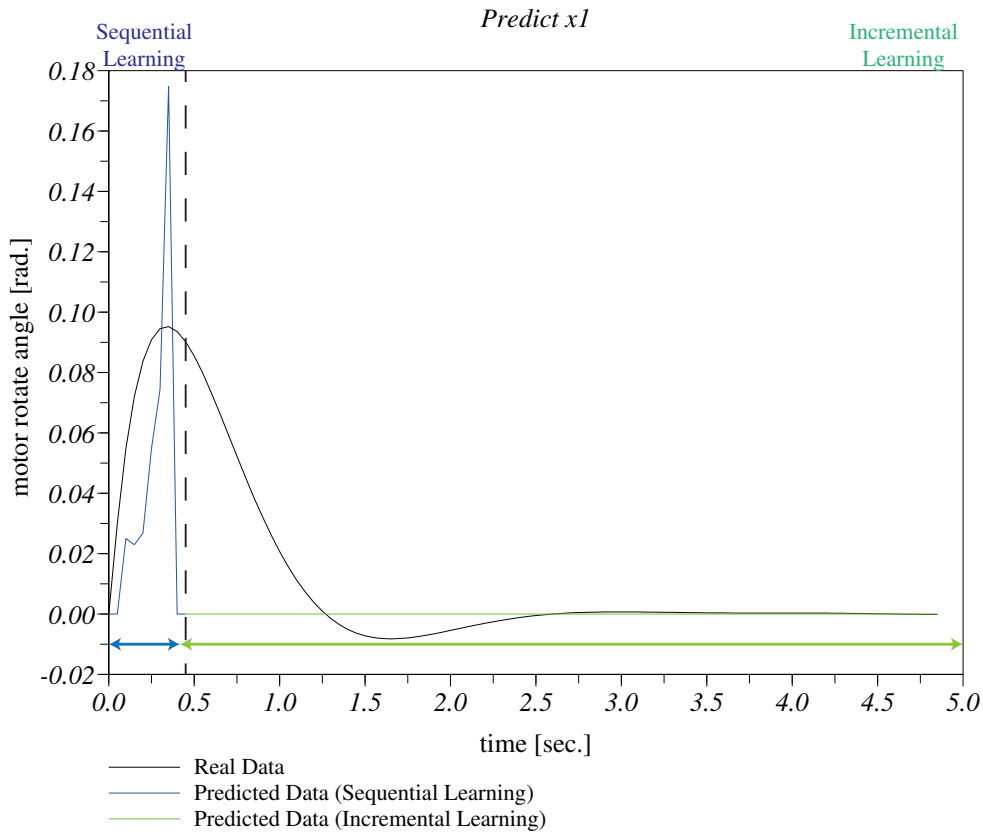


図 5.16 内部状態  $x_1$  に対する逐次予測の結果 (外乱を与えない場合)

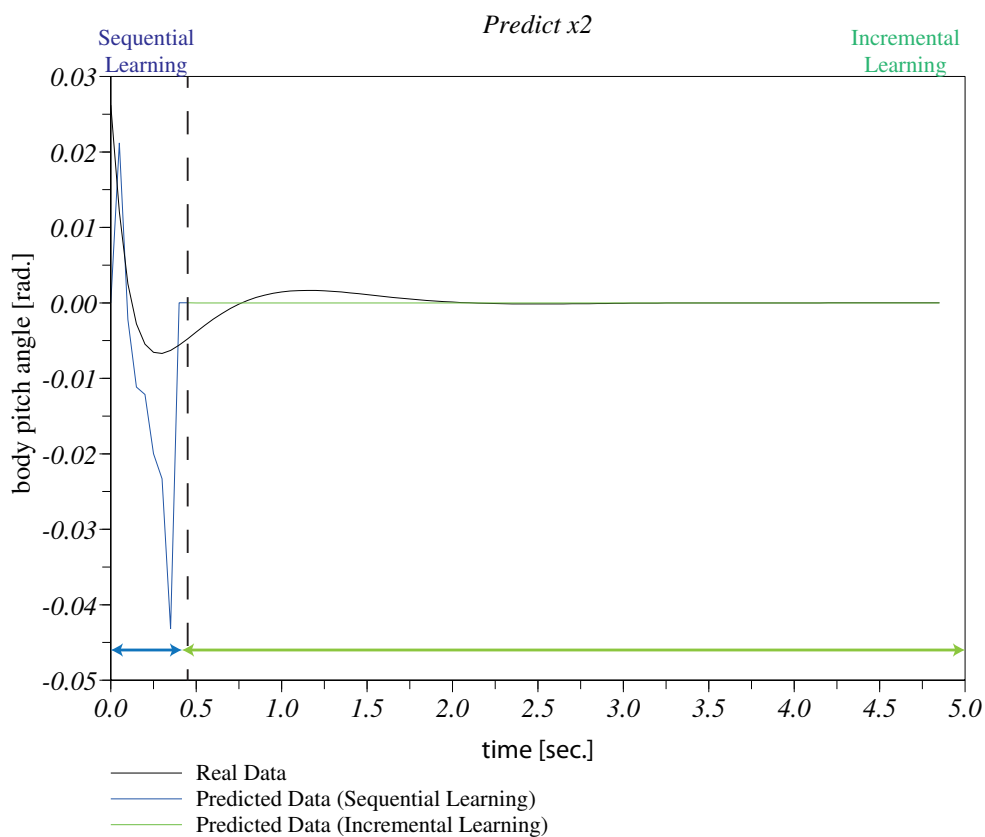


図 5.17 内部状態  $x_2$  に対する逐次予測の結果 (外乱を与えない場合)

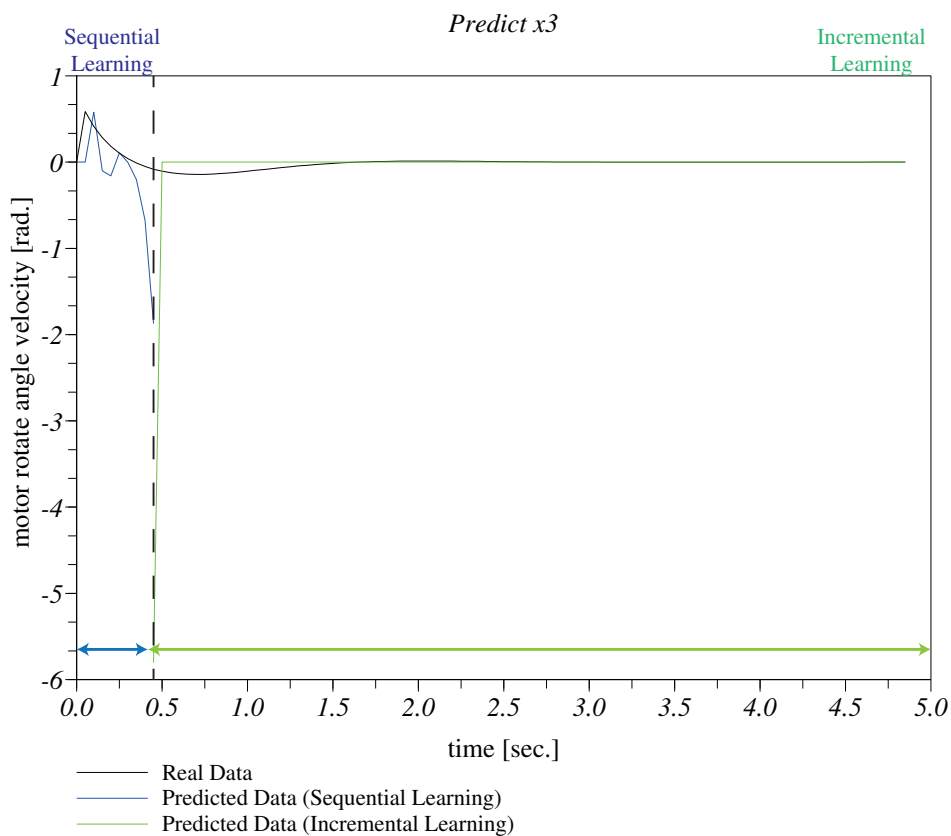


図 5.18 内部状態  $x_3$  に対する逐次予測の結果 (外乱を与えない場合)

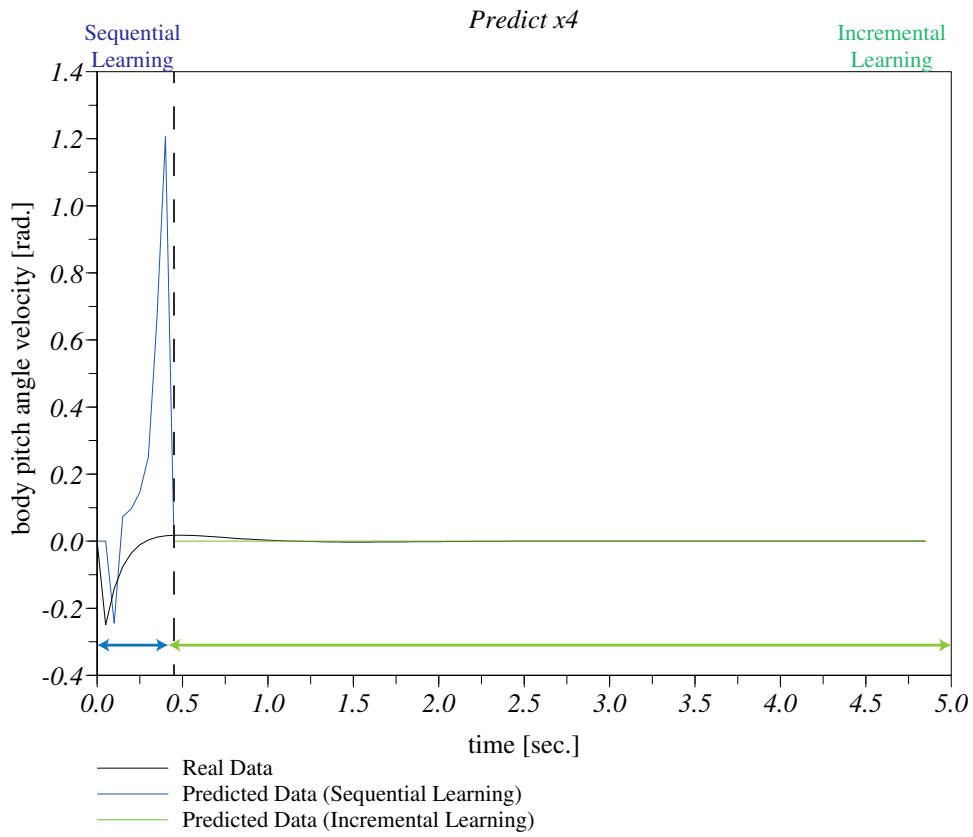


図 5.19 内部状態  $x_4$  に対する逐次予測の結果 (外乱を与えない場合)

#### 5.4.1.2 予測可能な外乱を与えた場合の訓練データに対する逐次内部状態予測

予測可能な外乱を与えた訓練データに対する予測結果を図 5.20 から図 5.23 に示す.

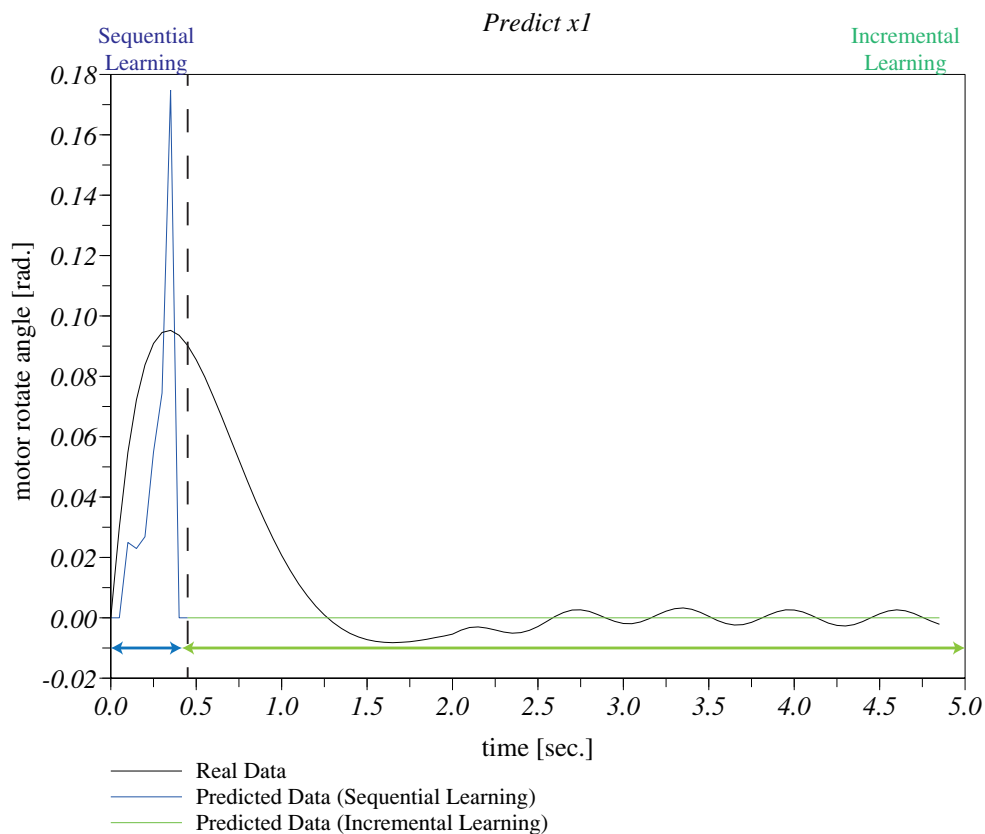


図 5.20 内部状態  $x_1$  に対する逐次予測の結果 (予測可能な外乱を与えた場合)

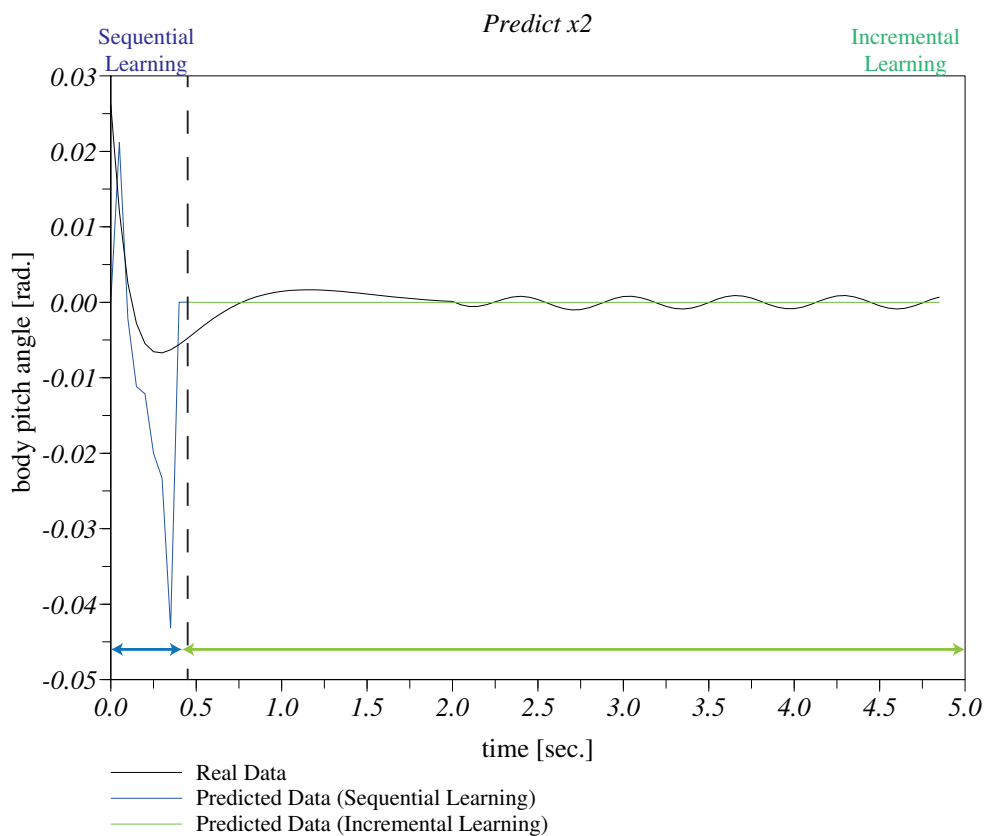


図 5.21 内部状態  $x_2$  に対する逐次予測の結果 (予測可能な外乱を与えた場合)

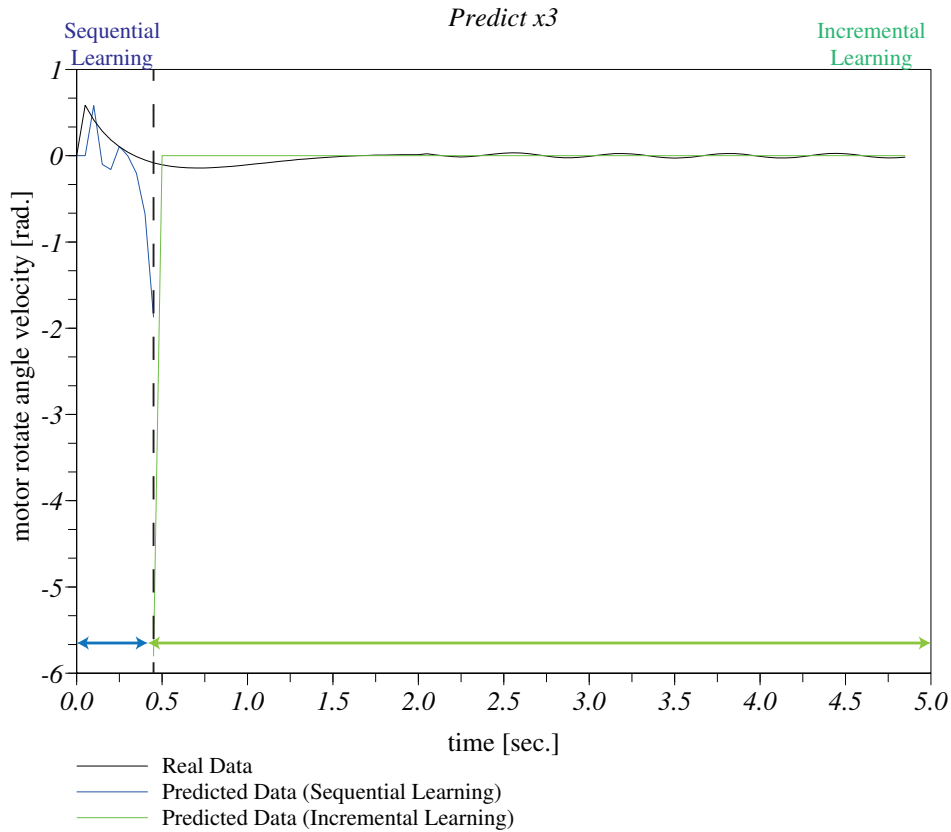


図 5.22 内部状態  $x_3$  に対する逐次予測の結果 (予測可能な外乱を与えた場合)

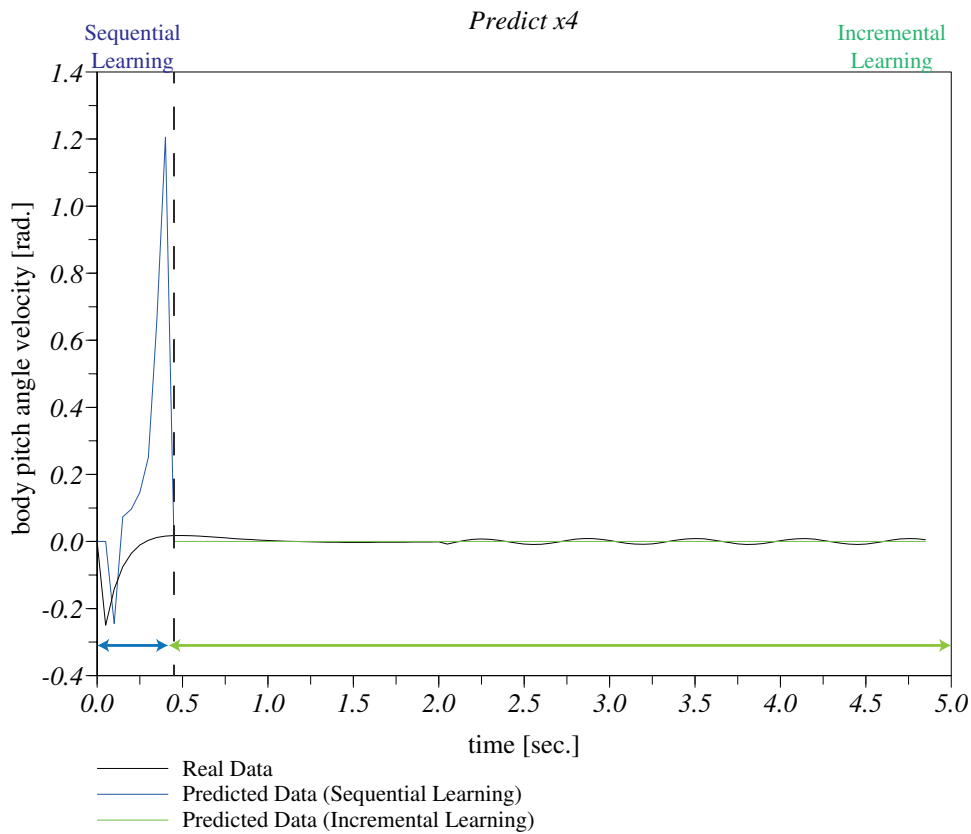


図 5.23 内部状態  $x_4$  に対する逐次予測の結果 (予測可能な外乱を与えた場合)

## 5.4.1.3 突発的な外乱を与えた場合の訓練データに対する逐次内部状態予測

突発的な外乱を与えた訓練データに対する予測結果を図 5.24 から図 5.27 に示す。

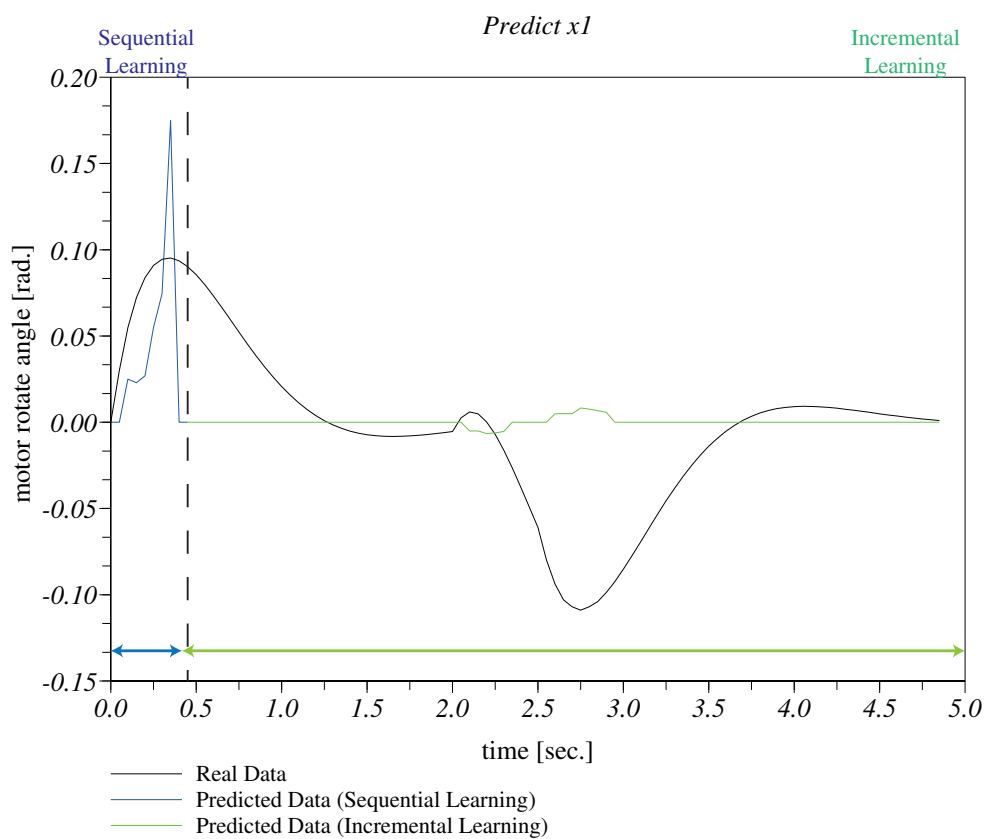


図 5.24 内部状態  $x_1$  に対する逐次予測の結果 (突発的な外乱を与えた場合)

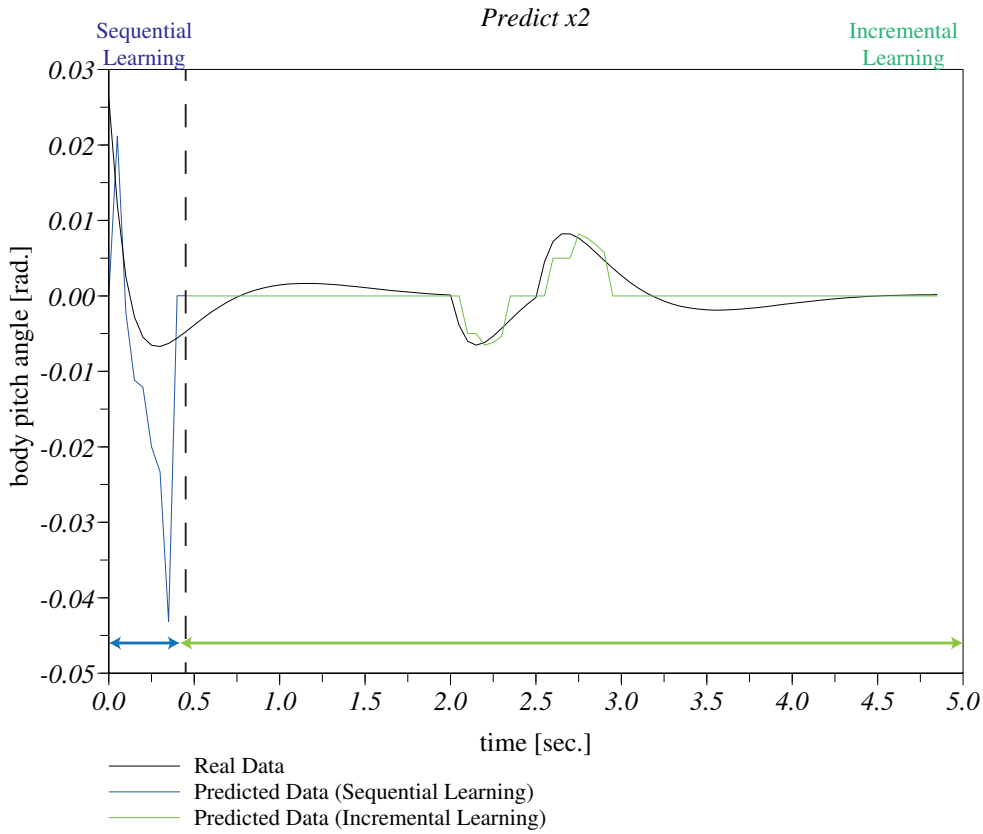


図 5.25 内部状態  $x_2$  に対する逐次予測の結果 (突発的な外乱を与えた場合)

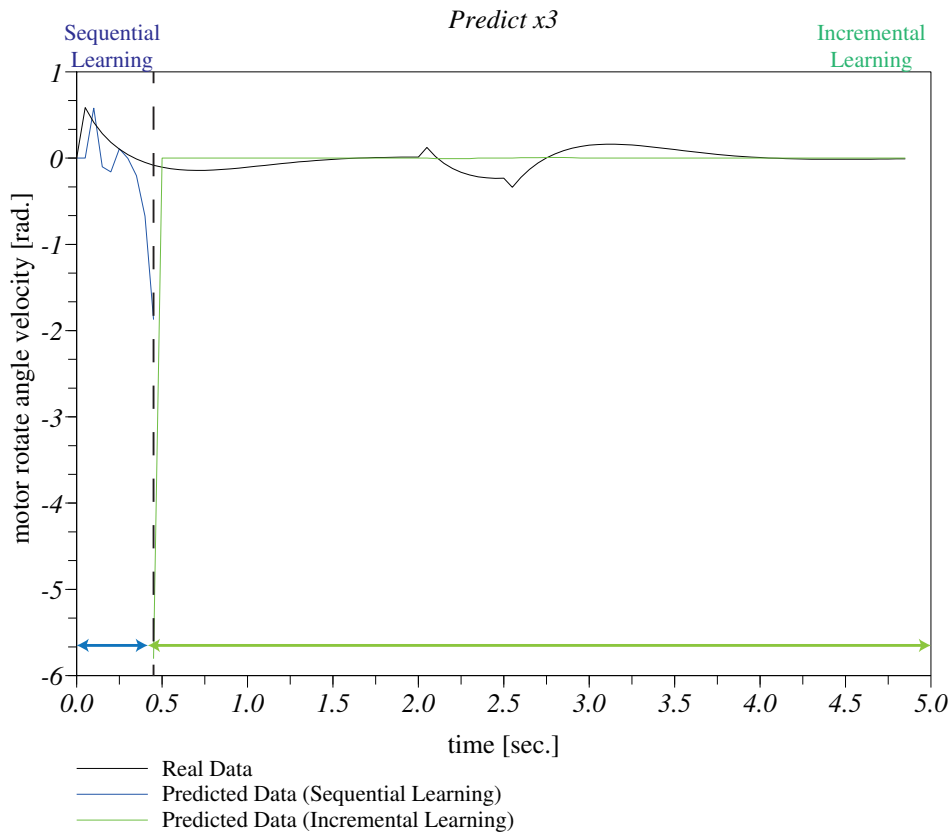


図 5.26 内部状態  $x_3$  に対する逐次予測の結果 (突発的な外乱を与えた場合)



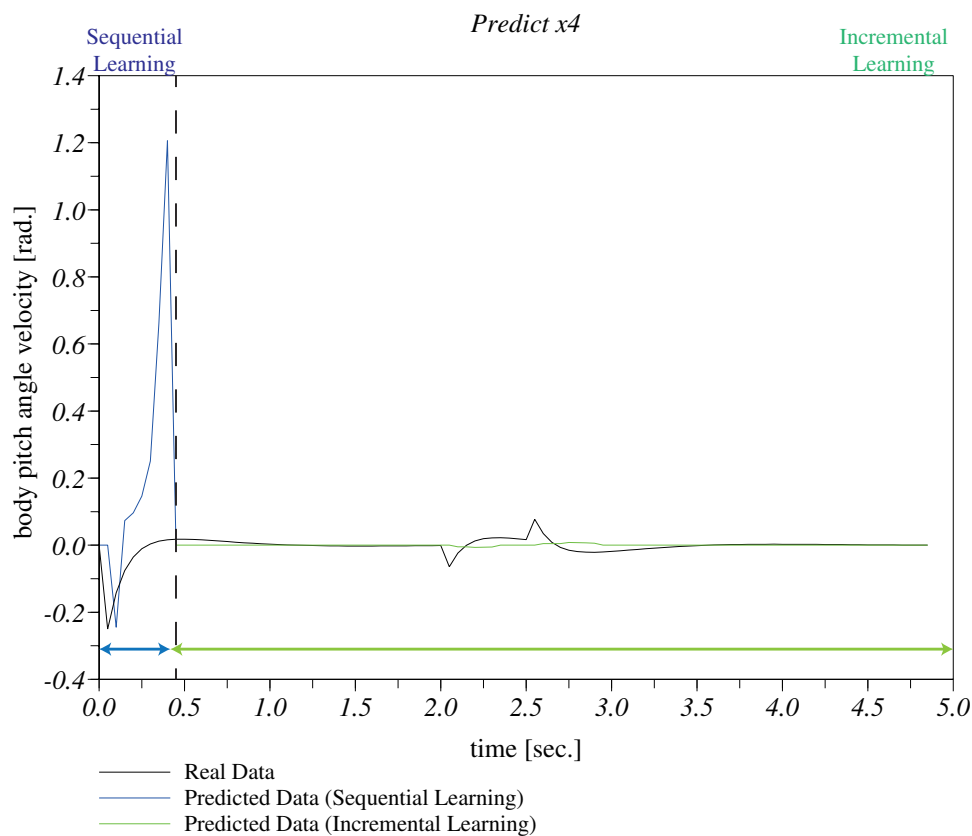


図 5.27 内部状態  $x_4$  に対する逐次予測の結果 (突発的な外乱を与えた場合)

#### 5.4.2 逐次学習と追加学習を組合わせた長期内部状態予測

突発的な外乱を与えた訓練データに対して, 内部状態を長期予測した結果を, 図 5.28 から図 5.31 に示す.

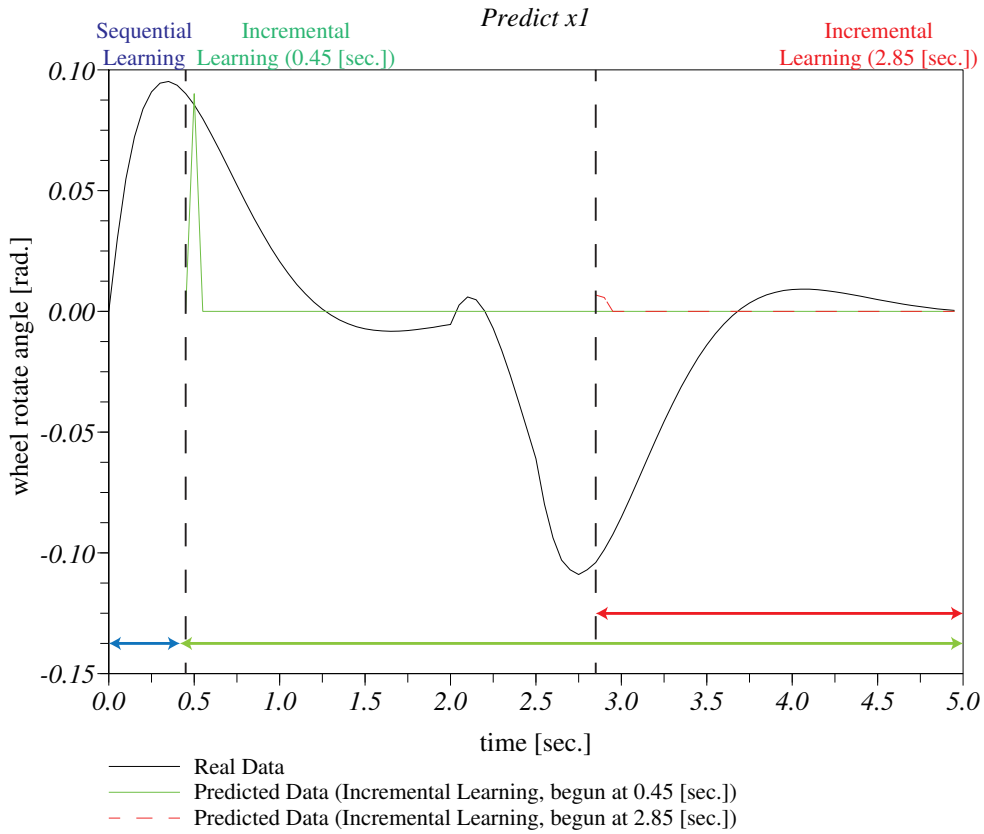


図 5.28 内部状態  $x_1$  に対する状態予測の結果

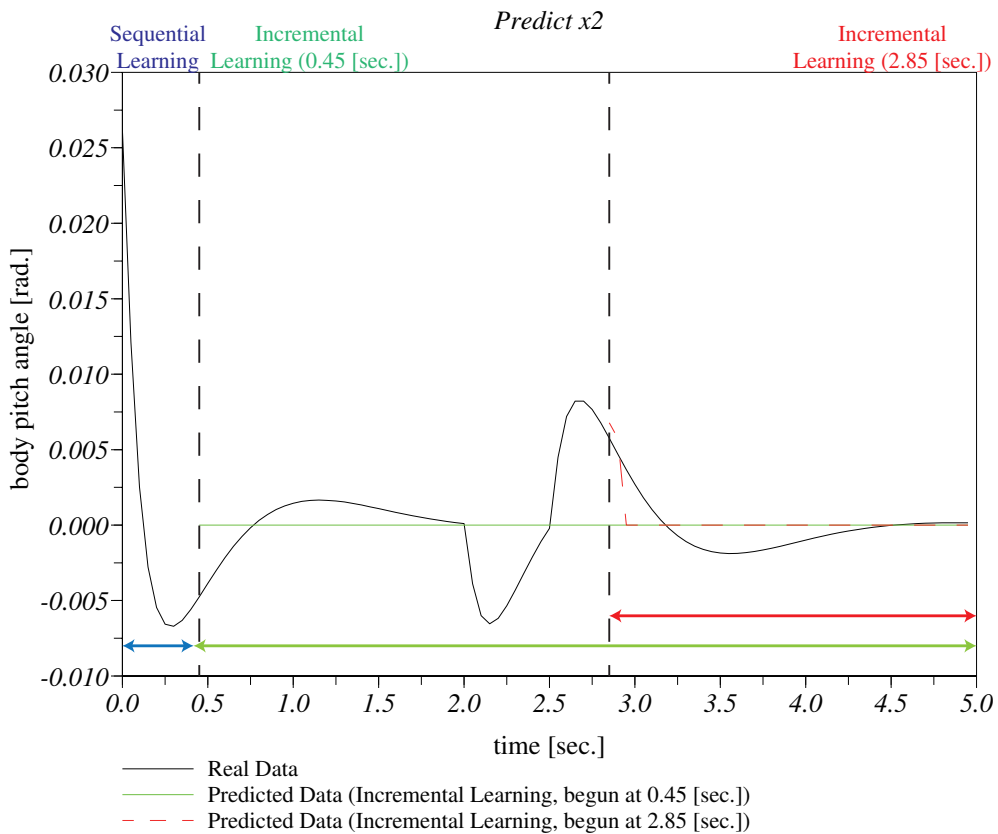
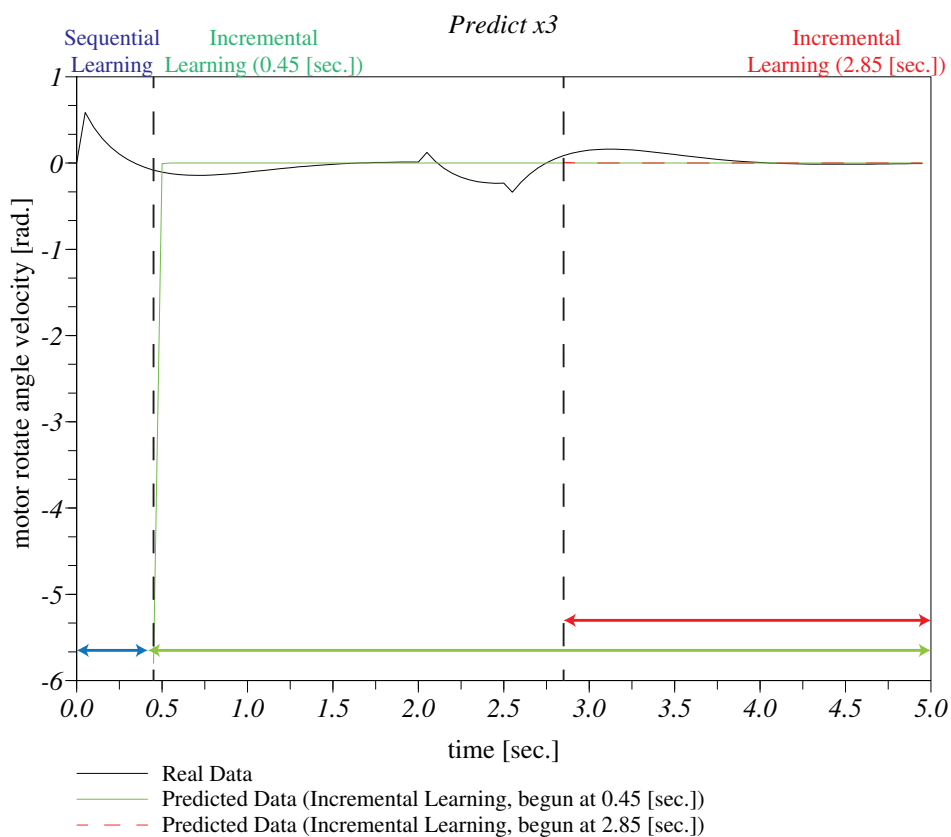
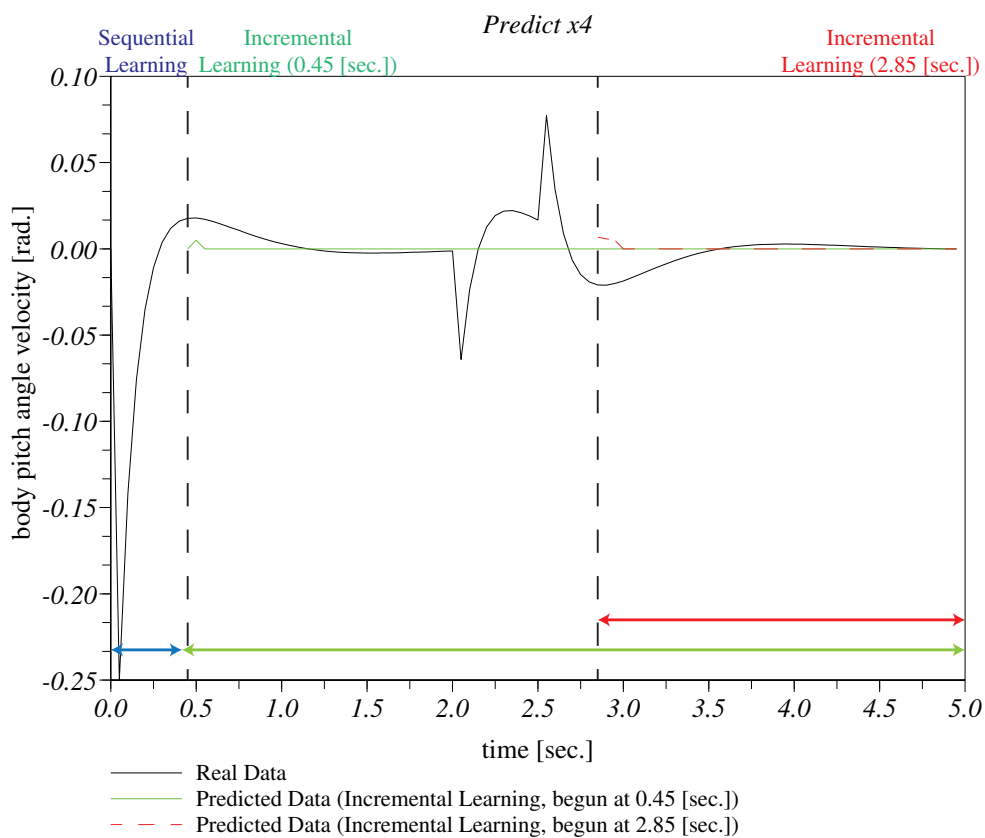


図 5.29 内部状態  $x_2$  に対する状態予測の結果

図 5.30 内部状態  $x_3$  に対する状態予測の結果図 5.31 内部状態  $x_4$  に対する状態予測の結果

## 5.5 実験結果に対する検討

### 5.5.1 逐次学習と追加学習を組合わせた逐次内部状態予測

図 5.16 から図 5.27 を見ると、学習当初は訓練データの出力を正確には予測出来ていないが、時間の経過に従い、学習データの変化に追従し、その傾向をよく表した結果となっている事が判る。特に 4.0[sec.] 以降の定常状態に於いては予測精度が或る程度高くなっており、実用に値するものと考えられる。一方で、傾向が変化する部分と、外乱の混入により状態が変化する部分に於いては最も訓練データとの差が生じているが、時間の経過に従い、訓練データを追加学習する事でその差も不感帯域にまで低下している事が判る。

結果として、訓練データの持つ特徴をよく掴み、倒立振子に外乱が混入しても安定化に転じるような予測結果を得ているものと結論付ける。

### 5.5.2 逐次学習と追加学習を組合わせた長期内部状態予測

図 5.28 から図 5.31 を見る。緑線で示される、0.45[sec.] から開始される長期状態予測の結果に注目すると、学習当初は訓練データの出力を正確には予測出来ていないが、時間の経過に従い、長期予測開始以降は倒立振子が安定化され、安定状態にとどまり続ける事を示す予測結果を得ている事が判る。一方で 2.0[sec.] 付近で外乱を与えているが、この付近で予測結果が大きく異なっている。これは、0.45[sec.] までで得た内部状態を基に、0.50[sec.] 以降の内部状態と行動の予測を繰り返しており、直接的に訓練データを用いていない為である。続いて赤線で示した 2.85[sec.] にて開始される長期状態予測の結果に着目する。この時点では、外乱の混入により状態が変化する部分までを追加学習によって学習している事から、安定化に至る内部状態の変化を予測し、その後の安定状態を予測出来ている。また訓練データとの予測誤差についても、時間の経過に従い不感帯域にまで低下している事が判る。

結果として、本手法に於いても訓練データの持つ特徴をよく掴み、倒立振子に外乱が混入しても安定化に転じるような予測結果を得ているものと結論付ける。

## 5.6 まとめ

検証実験では、状態フィードバックにより安定化した自走式二輪型倒立振子の制御応答を訓練データとし、内部状態と行動(制御入力)を基に、未来の内部状態を予測した。実験結果より、傾向の変化する部分に於いても安定化がされる様な内部状態を予測し、時間の経過に従い正確な予測が可能となる結果を得た。これにより、提案手法の有用性を示した。

---

## 第6章 結論

本章では、本研究の結果を総括する。

### 6.1 成果のまとめ

本研究では、ロボットが動的な環境下にて或る作業を実現する場合に於いても、人間と同様に『状態予測からの行動決定』が重要である点に着目した。そこで、ロボットの為に『状態予測』の考え方を与える事を目的とした。より具体的には、未来のロボットの内部状態がロボット自身の選択した行動に影響する点を考慮し、現在の内部状態とその時点で採る行動から、未来の内部状態を予測する手法を提案した。この目標を達成する為に、以下の事項を行った。

- (1) Parrella が提案した Online SVR に於いて、訓練データとして用いる事が出来るデータを、内部状態及び行動として定義される多次元データを取り扱える様拡張した。
- (2) 多次元データに対して追加学習を定義した。
- (3) 多次元データに対して逐次内部状態予測を定義した。
- (4) 多次元データと追加学習に対して長期内部状態予測を定義した。

これら提案した手法を、適用対象である自走式二輪型倒立振子の制御応答に対しシミュレーションにて逐次内部状態予測及び長期内部状態予測を実施した。その結果として、外乱の混入により状態が変化する部分に於いては最も訓練データとの差が生じているが、時間の経過に従い、訓練データを追加学習する事でその差も不感帯域にまで低下する事が判り、これによって訓練データの持つ特徴をよく掴み、倒立振子に外乱が混入しても安定化に転じるような予測結果を得る事が出来た。

従って、内部状態と行動を組合わせて未来の内部状態を予測する事によって、傾向の変化する部分に於いても安定化がされる様な内部状態を予測し、時間の経過に従い正確な予測が可能となる結果を得た。これにより、提案手法の有用性を確かめたものと結論付ける。

### 6.2 今後の展望

本研究では、逐次学習と追加学習を組合わせる事で、予測を実現した。この方法では、学習の初期に於いてはデータセットに訓練データが揃わない限り学習が完了しない為、有効な予測結果が得られないものと考えられる。これを端的に示したものが、図 5.27 に於ける学習初期の部分である。これを行動決定に用いる事で、その結果を直接的に以降の学習と予測、及び行動に影響を与える為、効率的な状態予測と行動決定が達成出来ない可能性が在る。

従って今後は、予測や行動決定の精度向上の為にノイズ削減や計算コストの軽減を図る為に、次元を削減する<sup>[44]</sup>事が一点目の課題として挙げられる。二点目としては、実世界に於ける生物をモデルとして、迅速

な予測と行動決定を可能とする手法を提案する事が挙げられる。具体的には、実世界に於ける生物を考えた時、彼らは複雑な環境下で素早く適切な行動を行う為に、予測を使っている。具体的には、事前に感覚器からの入力を予測する事で、事前に適切な行動を選定し、

- 素早く行動を実現する (時間を短縮させている)
- 行動決定のコストを下げる (感覚-行動の汎化により無意識下での行動選択を実現させている)

事を行っていると考えられている。特に環境が変化し続け、環境認識に対する行動決定に時間的制約がある状況下では、素早く最適な行動選択を行う必要がある。現状態を認識し、それに対する行動選択を行う場合、最短でも時刻ゼロに於ける反応は不可能である。そこで、事前に環境を予測し、取るべき行動を大まかにでも決定しておくことで、

- 瞬間的な反応
- 短時間での行動決定と探索

を行える事が必要となる。これをシステムとして実現するには、階層的に予測を行うシステムを構築する必要がある。その為に、今後は、本研究で提案した未来予測を活用する事で、

- 下位層では短期の未来予測を活用し、制御レベルでの行動選択を行う
- 上位層では長期の未来予測を活用し、タスク単位の行動選択を行う

手法を提案する必要がある。

## 謝辞

本研究の推進および、講演論文、ポジションペーパー、そして本論文の執筆に際しまして終始熱心なご指導を頂きました国立大学法人室蘭工業大学助教 倉重 健太郎先生に、心より感謝の意を表します。また、研究での助言を頂いた畑中 雅彦教授、佐賀 聡人教授、本田 泰准教授、鈴木 幸司教授に感謝申し上げます。

そして、本研究に関わって頂いた認知ロボティクス研究室の皆様 (木島 康隆 先輩、中南 義典 先輩、宮崎 愛央 先輩、梅津 祐介 君、北山 直樹 君、澁谷 和 さん、高泉 昇太郎 君、沼田 利伸 君、三浦 丈典 君、木村 敏久 君、挾間 重直 君、平間 経太 君) に感謝申し上げます。

これまでのコンピュータ知能学コースでの勉強を無くして、この研究は成し得ませんでした。今日までご指導頂いた同講座の全教員にこの場を借りて御礼申し上げます。





---

## 参考文献

- [1] Sebastian Thrun, Wolfram Burgard, Dieter Fox : Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series), The MIT Press (2005)
- [2] 浅香俊一, 石川繁樹 : 動的な環境における自律移動ロボットの行動制御, 日本ロボット学会誌 Vol.12 No.4, pp.583-589 (1994)
- [3] 神田崇行, 石黒浩, 小野哲雄, 今井倫太, 前田武志, 中津良平 : 研究用プラットフォームとしての日常活動型ロボット “Robovie” の開発, 電子情報通信学会論文誌 D-I Vol.J85-D-I No.4, pp.380-389 (2002)
- [4] 国際ロボット連盟 : 産業用ロボットの明るい展望 ~ ロボットが百万人以上の雇用を創出 ~ (2011)
- [5] 十河卓司, 木元克美, 石黒浩, 石田亨 : 分散視覚システムによる移動ロボットの誘導, 日本ロボット学会誌 Vol.17 No.7, pp.1-7 (1999)
- [6] 西岡基夫, 岡田明, 宮野道雄, 森一彦, 山下久仁子, 中山圭以 : 繰り返し作業における操作方略の変化と動作特性に関する基礎的研究 - パフォーマンスデータと生理的データからみた評価方法の検討 -, 生活科学研究誌 Vol.7 居住環境分野, pp.45-55 (2008)
- [7] Thomas Mitchell : Machine Learning, The McGraw-Hill Companies (1997)
- [8] 銅谷賢治 : 臨時別冊数理科学 SGC ライブラリ 60 「計算神経科学への招待」 脳の学習機構の理解を目指して, サイエンス社 (2007)
- [9] M.O. Abu-Shawiesh, F.M. Al-Athari, H.F. Kittani : Confidence Interval for the Mean of a Contaminated Normal Distribution, Journal of Applied Sciences, pp.2835-2840 (2009)
- [10] Teuvo Kohonen : Self-Organizing Maps, Springer (2000)
- [11] 植田英三郎 : 時系列解析による株価予測, 大阪府立大学経済研究 53(3), pp.95-111 (2007)
- [12] Jongho Shin, H. Jin Kim, Sewook Park and Youdan Kim : Model predictive flight control using adaptive support vector regression, Neurocomputing 73(4-6): pp.1031-1037 (2010)
- [13] Younggeun Choi, Shin-Young Cheong and Nicolas Schweighofer : Local Online Support Vector Regression for Learning Control, Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation Jacksonville, FL, USA (2007)
- [14] 杉本徳和, 鮫島和行, 銅谷賢治, 川人光男 : 複数の状態予測と報酬予測モデルによる強化学習と行動目標の推定, 電子情報通信学会誌 D-II, 87(2), pp.683-694 (2004)
- [15] Francesco Parrella : Online Support Vector Regression, A Thesis presented for the degree of Information Science, Department of Information Science, University of Genoa, Italy (2007)
- [16] Corinna Cortes and V. N. Vapnik : Support-Vector Networks, Machine Learning, 20 (1995)
- [17] V. N. Vapnik : The Nature of Statistical Learning Theory, Springer, New York (1995)
- [18] 高橋泰岳, 浅田稔 : 実ロボットによる行動学習のための状態空間の漸次的構成, 日本ロボット学会誌 Vol.17 No.1, pp.118-124 (1999)
- [19] Davidson, J. E. H., D. F. Hendry, F. Srba and S. Yeo : Econometric Modelling of the Aggregate Time Series Relationship Between Consumers' Expenditure and Income in the United Kingdom, Economic

- Journal 88, pp661-692 (1978)
- [20] M. Aizerman, E. Braverman and L. Rozonoer : Theoretical foundations of the potential function method in pattern recognition learning, Automation and Remote Control 25, pp.821-837 (1964)
- [21] J. J. HOPFIELD : Neural networks and physical systems with emergent collective computational abilities, Proc. NatL Acad. Sci. USA Biophysics Vol. 79, pp.2554-2558 (1982)
- [22] Adriaan Zaanen : Linear Analysis, North Holland Publishing Co. (1960)
- [23] Masashi Sugiyama, Hiroataka Hachiya, Christopher Towell, Sethu Vijayakumar : Geodesic Gaussian Kernels for Value Function Approximation, Autonomous Robots, Vol.25 No.3, pp.287-304 (2008)
- [24] L. Valiant : A theory of the learnable, Communications of the ACM, 27 (1984)
- [25] Huber, Peter J. : Robust Estimation of a Location Parameter, Annals of Statistics 53, pp.73-101 (1964)
- [26] 小林正幸, 小西康夫, 石垣博行 : サポートベクター回帰モデルによる倒立振子の学習制御, 日本機械学会 Dynamics and Design Conference 2003, No.03-7, pp.108-1-108-5 (2003)
- [27] Kuhn, H. W. and Tucker, A. W. : Nonlinear programming, Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, pp.481-492 (1951)
- [28] 杉本大志, 倉重健太郎 : オンラインサポートベクトル回帰におけるリアルタイム予測の検討, 2011 年度精密工学会北海道支部学術講演会講演論文集, pp59-60 (2011)
- [29] Christopher M. Bishop : Pattern Recognition and Machine Learning (Information Science and Statistics), Springer (2006)
- [30] B. S. Everitt : Cambridge Dictionary of Statistics (2nd edition), Cambridge, UK: Cambridge University Press., ISBN 0-521-81099-X (2002)
- [31] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui and Michael I. Jordan : Learning the Kernel Matrix with Semidefinite Programming, The Journal of Machine Learning Research 5, pp.27-72 (2004)
- [32] F. Girosi : An equivalence between sparse approximation and Support Vector Machines, Neural Computation, 10(6):1455-1480 (1998) NJ, USA c2009, table of contents ISBN: 978-1-4244-4523-3 (2009)
- [33] Adam Mills, Adrian Wills, Brett Ninness : Nonlinear model predictive control of an inverted pendulum, ACC'09 Proceedings of the 2009 conference on American Control Conference, pp.2335-2340, IEEE Press Piscataway,
- [34] Y.Yamamoto: NXTway-GS Model-Based Design -Control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT-, CYBERNET SYSTEMS CO.,LTD. (2008)
- [35] Brian D.O. Anderson, John B. Moore : Optimal Control: Linear Quadratic Methods. Englewood Cliffs, NJ: Prentice Hall, ISBN 978-0-13-638560-8 (1990)
- [36] Stanley G.M., Mah R.S.H. : Observability and Redundancy Classification in Process Networks, Chem. Engng. Sci. 36, pp.1941-1954 (1981)
- [37] 安定の定義 <http://ysserve.int-univ.com/Lecture/ControlMecha2/node18.html>
- [38] Khalil, H.K. : Nonlinear systems, Prentice Hall Upper Saddle River, NJ (1996)
- [39] 最適レギュレータの説明 <http://www.me.saga-u.ac.jp/~sato/regulator.pdf>
- [40] 杉本大志, 吉村斎, 阿部司, 大村功 : Scilab/Scicos による組み込みシステムのモデルベース開発に関する研究, 2010 年度精密工学会春季大会学術講演会講演論文集, pp.343-344 (2010)
- [41] 杉本大志, 吉村斎, 阿部司, 大村功 : Scilab/Scicos による組み込みシステムのモデルベース開発に関する研究 — 自動コード生成機能の開発 —, 日本機械学会ロボティクス・メカトロニクス講演会'10 講演論文集 (ROBOMECH'10), 2A2-C10 (2010)
- [42] LEGO Mindstorms NXT [http://web.mac.com/ryo\\_watanabe/iWeb/Ryo%27s%20Holiday/](http://web.mac.com/ryo_watanabe/iWeb/Ryo%27s%20Holiday/)

---

LEGO%20Mindstorms%20NXT.html

[43] R. Courant, D. Hilbert : Methoden der Mathematischen Physik, Berlin, Springer (1931)

[44] Kenji Fukumizu : Dimensionality Reduction in Regression with Positive Definite Kernels, Proceedings of the Institute of Statistical Mathematics Vol.53, No.2, pp.189-200 (2005)



---

## 研究業績

- [1] 杉本大志, 倉重健太郎: オンラインサポートベクトル回帰におけるリアルタイム予測の検討, 2011 年度精密工学会北海道支部学術講演会講演論文集, pp59-60 (2011)

# オンラインサポートベクトル回帰におけるリアルタイム予測の検討

室蘭工業大学 ○ 杉本 大志, 倉重 健太郎

## 要 旨

非線型サポートベクトル回帰では、過去のカーネル行列から統計的な予測が可能だが、学習データを逐次追加した場合、再学習すると同時にすべてのデータを保存する必要がある。この問題の解決策のひとつとして、逐次学習によって再学習を回避した Online SVR が提案されているが、逐次的な予測の手法が示されておらず、リアルタイム制御への応用が難しい。

本稿では、この問題を解決することで、Online SVR 上でリアルタイムに逐次予測する手法を提案し、その有用性を検討する。

## 1. はじめに

機械学習の分野はここ数年で拡大しており、この成果を利用することで数多くの新技術も成長を遂げている。機械学習における既存のアルゴリズムのうち最も有名であり、最近のトレンドとなっている理論として、識別問題のためのサポートベクトルマシン (SVM)<sup>[1]</sup>、回帰分析のためのサポートベクトル回帰 (SVR)<sup>[2]</sup> があげられる。本来の SVM, SVR は線型分離問題、線型回帰分析のみを対象に考案されたものであるが、Kernel Trick<sup>[3]</sup> と呼ばれる手法を適用することで応用範囲はさらに拡大し、注目されるようになった。

最近では、識別問題への適用にとどまらず、ロボットの制御、認識に SVM や SVR (以下、一括学習による SVR とよぶ) を適用した研究が行われている。小林らは、一括学習による SVR を用いて、倒立振り子の逆システムを学習させた<sup>[4]</sup>。この研究では、あらかじめ状態フィードバック制御で得られた入出力データを用いて学習し、これにより得た SVR モデルを姿勢制御に適用することで、SVR が初期値に対してロバストであり、ロボットの学習制御への応用に有効であることを、シミュレーションによって示した。

この SVM, SVR の欠点の一つとして、一括学習でのみ動作するということがあげられる。これらの手法では、すべての学習データを用いて学習し、新たな学習データが追加された場合は、すでに存在する学習データに加えて再学習することが必要となる。この問題については様々な解決策が提案されている。そのなかでも Parrella は、逐次学習の手法を SVR に適用した Online SVR<sup>[5]</sup> を提案している。このアルゴリズムを適用することにより、学習器が最初から再学習をすることなく学習データの追加 (追加学習) と削除 (忘却) を行うことができるとしている。

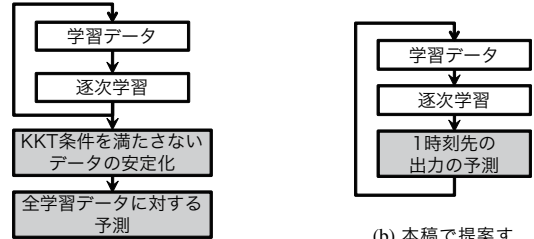
しかし Parrella は、自身の手法によると、回帰モデル  $y = \mathbf{w}^T \mathbf{x} + b$  に含まれる重み係数ベクトル  $\mathbf{w}$  を計算する際、浮動小数点演算を行うことで数値的不安定性が生じる可能性があるとしている。これによる演算誤差により、計算結果に不適切な大きな数値を出力することがあると指摘している。この問題については、KKT 条件<sup>[6]</sup> を満たさなかった学習データを忘却し、これまでに追加された学習データ系列の最後尾にその学習データ追加し改めて学習させるということで対処している<sup>[5]</sup> (図 1(a)、またこの操作を “Stabilization<sup>[5]</sup>”, 安定化と称している)。このため、安定化を行うことなく逐次学習を行った後予測した場合、各学習データの値から逸脱した結果を示す可能性が考えられる。また、この手法では、再学習を避ける形で学習データを逐次的に与えることは可能だが、次時刻における出力の予測を、逐次的に与える方法については述べられていない。このため、ロボット制御などに代表されるリアルタイム制御へ応用するには、あらかじめ入出力データを与えモデルを学習させるのではなく、学習データが時間とともに変化することを考慮し、データを逐次与えて学習させると同時に、次時刻の出力を予測する手法を与える必要がある。

そこで本稿では、Parrella による  $\mathbf{w}$  の計算手法を利用しないことで、数値的不安定性を回避することを考える。そして、逐次的に学習データを与え Online SVR により学習させると同時に、これまでの学習から得た過去のサポートベクトルと、新たに入力された学習データ、あわせて学習により更新されるパラメータを用いることで、予測式を構成する。この予測式を適用し、逐次学習と組み合わせることで、Parrella の提案では述べられていなかった逐次予測の手法を提案する。

## 2. 逐次予測の構成

### 2.1 提案手法の流れ

本稿では、Online SVR が学習データを逐次取得し学習すると同時に、学習における更新式をもとに次時刻における出力 (以下、次状態とよぶ) を予測する手法を提案する。本手法の流れを図 1(b) に示す。両者を比較すると、本稿で述べる提案手法では学習結果に対する安定化を行わず、また逐次学習とともに次状態の予測を実施している。



(a) Parrella が提案した手法の概要

(b) 本稿で提案する逐次予測の概要

図 1 システム構成の比較

### 2.2 新入力に対する逐次予測の考え方

新入力  $\mathbf{x}_{N+1}$  に対する予測値  $\hat{y}_{N+1}$  を、センサなどより得た過去の観測値  $\mathbf{y} = [y_1, \dots, y_N]^T$  からなるデータを用い推定する。一括学習による SVR において  $\mathbf{x}_{N+1}$  に対する  $\hat{y}_{N+1}$  は、

$$\hat{y}_{N+1} = \mathbf{w}^T \phi(\mathbf{x}_{N+1}) + b \quad (2.1)$$

$$= \mathbf{a}^T \Phi \phi(\mathbf{x}_{N+1}) + b \quad (2.2)$$

$$= \mathbf{k}(\mathbf{x}_{N+1})^T (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{y} + b$$

で予測できることが知られている。ここで、 $b$  はバイアス項、 $\lambda$  は正則化パラメータ、 $\Phi$  はデザイン行列と呼ばれ、その第  $n$  列は  $\phi(\mathbf{x}_n)^T$  であり、

$$\mathbf{w} = \sum_{i=1}^N (a_i - a_i^*) \phi(\mathbf{x}_i) = \sum_{i=1}^N \theta_i \phi(\mathbf{x}_i) = \Phi^T \mathbf{a} \quad (2.3)$$

$$b = \epsilon + y_i - \sum_{i=1}^N \theta_i k(\mathbf{x}_i^T \mathbf{x}_i) \quad (2.4)$$

$$\mathbf{k}(\mathbf{x}_{N+1}) = (k(\mathbf{x}_1, \mathbf{x}_{N+1}), \dots, k(\mathbf{x}_N, \mathbf{x}_{N+1}))^T \quad (2.5)$$

$$\mathbf{K} = \Phi \Phi^T, K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) \quad (2.6)$$

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \quad (2.7)$$

である。また、 $\epsilon$  は  $\epsilon$ -insensitive 損失関数<sup>[2,7]</sup> における insensitive 帯域である。これはつまり、予測が、学習パターンの目標変数  $y$  の線型結合から求められる、ということの意味している。

この式をもとに、本稿が目的としている Online SVR における逐次予測の実現を考える。ここでは、センサなどからの入力が 1 次元の場合を考え、 $\dim(\mathbf{x}_i) = 1$ 、つまり学習データ  $x_i$  で与えられることを前提とする。基本的な考え方としては、過去のサポートベクトルから、1 時刻先の値を予測するものである。過去の  $l$  個のサポートベクトル  $x_{s_k}$  ( $k \in l$ ) を用いて導かれるカーネル関数  $k$  から構成される Gram 行列  $\mathbf{K}_{sv}$  を、

$$k(x_{s_i}, x_{s_j}) = \phi(x_{s_i})^T \cdot \phi(x_{s_j}) = Q_{s_i, s_j} \quad (2.8)$$

とすることで、

$$\mathbf{K}_{sv} = \begin{bmatrix} Q_{s_1, s_1} & Q_{s_1, s_2} & \cdots & Q_{s_1, s_l} \\ Q_{s_2, s_1} & Q_{s_2, s_2} & \cdots & Q_{s_2, s_l} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{s_l, s_1} & Q_{s_l, s_2} & \cdots & Q_{s_l, s_l} \end{bmatrix} \quad (2.9)$$

と定義する。また、新入力  $x_{N+1}$  とそれまでのサポートベクトルとの特徴空間における内積を求める写像関数  $\mathbf{k}_{sv}(x_{N+1})$  については

$$\mathbf{k}_{sv}(x_{N+1}) = (k(x_{s_1}, x_{N+1}), \dots, k(x_{s_l}, x_{N+1}))^T \quad (2.10)$$

とする。

以上の定義より、新入力  $x_{N+1}$  に対する予測値  $\hat{y}_{N+1}$  を計算することを考える。

ここで、Online SVR においては、先述のとおり学習データが追加されるに従い逐次学習を繰り返すものであった。学習データとしてサンプル  $c$  が追加された場合、ラグランジュ乗数、バイアス項は

$$\theta' = \theta + \Delta\theta \quad (2.11)$$

$$b' = b + \Delta b \quad (2.12)$$

$$\sum_{j \in S} Q_{ij} \Delta\theta_j + \Delta b = -Q_{ic} \Delta\theta_c \quad \text{where } i \in S \quad (2.13)$$

$$\sum_{j \in S} \Delta\theta_j = -\theta_c \quad (2.14)$$

により更新される<sup>[5]</sup>。ここで、 $S$  はサポートベクトルの添え字集合  $\{s_k\}$  である。つまり、新たなサンプル  $c$  に対するラグランジュ乗数とバイアスは、現在までの積算値から求められる。そこで、上式から、予測式の構成を行う。

まず、 $N=0$  のときは、初期状態でいかなるデータも与えられていないので、

$$\hat{y}_{N+1} = \hat{y}_1 = 0 \quad (2.15)$$

とする。つぎに  $N=1$ 、つまり一つ目の学習データが与えられた場合を考える。このときは  $x_1$  以外に学習データが存在せず、特徴空間における超平面が構成できないことから、サポートベクトル自体も存在しない。このため、

$$\Delta b = \Delta\theta \quad (2.16)$$

と導かれる。よって、

$$\hat{y}_{N+1} = \hat{y}_2 = \Delta b = \Delta\theta \quad (2.17)$$

となる。つづいて、 $N > 1$  の場合を考える。このときは、 $x_N$  以外にも学習データが存在するので、特徴空間における超平面が構成でき、サポートベクトルが存在する。よって、

$$\hat{y}_{N+1} = \mathbf{k}_{sv}(x_{N+1})^T (\mathbf{K}_{sv} + \lambda \mathbf{I})^{-1} \mathbf{y}_{sv} + b' \quad (2.18)$$

となる。以上をまとめると、

$$\hat{y}_{N+1} = \begin{cases} 0 & \text{if } N=0 \\ \Delta\theta & \text{if } N=1 \\ \mathbf{k}_{sv}(x_{N+1})^T (\mathbf{K}_{sv} + \lambda \mathbf{I})^{-1} \mathbf{y}_{sv} + b' & \text{otherwise} \end{cases} \quad (2.19)$$

という予測式が得られる。また、 $b'$  は、現在までのバイアス値の積算値として求められる。この予測式を適用することで、Parrella の指摘する問題を回避し、次状態の予測が可能となる。

### 3. 検証

#### 3.1 検証で用いるパラメータ

本稿では、検証にあたってカーネル法を適用した Online SVR を用い、カーネル関数には次式で定義される RBF カーネルを使用する。

$$k(x_1, x_2) = \exp(-\beta \|x_1 - x_2\|^2) \quad (3.1)$$

ここで、 $\beta$  はスケールパラメータである。また、学習には下表 1 に示すパラメータを扱うこととする。

表 1 実験に用いる各種定数

| 定数         | 数値    |
|------------|-------|
| $\lambda$  | 0.002 |
| $\epsilon$ | 0.06  |
| $\beta$    | 30    |

#### 3.2 検証データに適用した結果とその検討

本検証では、予測対象の入力データとして、周波数 1.5[Hz]、6.8[sec.] 以前は振幅 1、6.8[sec.] 以降は振幅 0.5 の正弦波を生成し、これを 0.02[sec.] 間隔で、0[sec.] から 11.8[sec.] までサンプリングした系列を使用した。

シミュレーションでは、このデータを逐次与え学習させるとともに、次状態を逐次予測させた。図 2 はそのときの結果である。比較対象として、Parrella による、学習終了後に安定化を実施しなかった場合の予測結果も併載する。提案手法の結果を見ると、学習の初期と、傾向が変化する部分（以下、傾向変化部とよぶ）では、学習データの出力を正確には予測できていないが、時間の経過に従い、学習データの変化に追従し、その傾向をよく表した結果となっていることがわかる。また、Parrella による予測法から得た予測結果と、提案手法による予測結果を比較すると、学習終了後の予測結果によく似た予測結果を、逐次予測により得ることができた。そして安定化を要する部分でも、不自然な変動を起こすことなく、学習データの傾向を表すような予測が達成されていると考えられる。

また、図 3 は、逐次予測による予測値と、その時刻における実データとの予測誤差を示したデータである。この予測誤差  $e_t$  は、

$$e_t = |\hat{y}_t - y_t| \quad (3.2)$$

により算出した。Parrella による予測法を用いた結果を見ると、その予測誤差は終始 0.05 程度となっているが、0.02[sec.] で誤

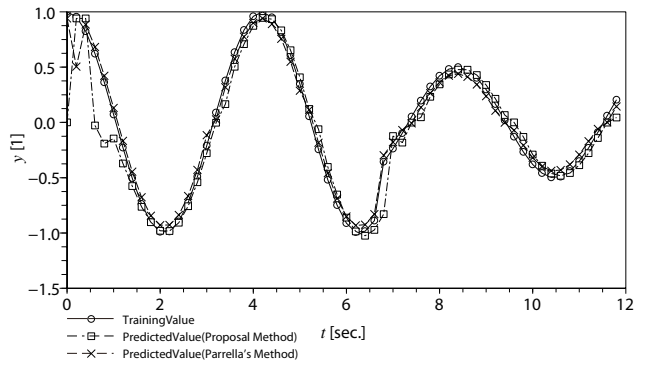


図 2 逐次予測を適用した場合の結果

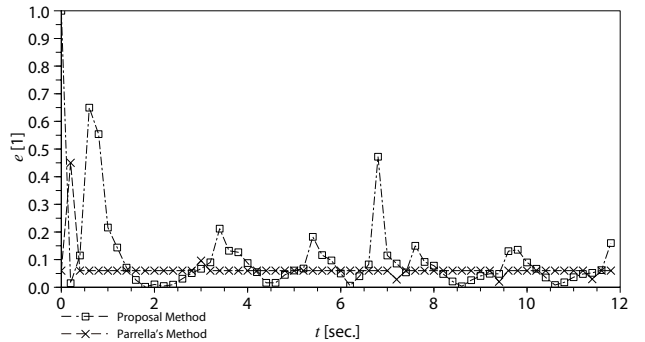


図 3 予測結果と実データとの誤差

差が最大となっていることがわかる。これは、学習終了後に安定化をしなかったためである。一方、提案手法では、学習当初と傾向変化部では誤差が最大となっているが、その後は誤差が減少し、2[sec.] 以降は最大でも 0.2 程度の値をとり、Parrella の予測誤差の近傍で振動していることがわかる。これは、正弦波データの符号が変化するときにタイミングが同じであり、 $y=0$  近傍における予測が影響しているものと考えられる。以上から、Parrella と同等な予測結果を、逐次的に得ることができたと考ええる。

### 4. 結論

本稿では、過去のサポートベクトルと新たに入力された学習データ、そして逐次更新されるパラメータを用いて次時刻における出力の予測式を構成することで、Online SVR による逐次予測の手法について提案した。また、この提案手法を、シミュレーションにより周波数が一定の正弦波データに適用することで、時間の経過に従い正確な予測が可能となる結果を得た。これにより、提案手法の有用性を示した。

今後の計画としては、

- 実機に適用した場合の検証。
- ロボット制御に有用なカーネル関数の提案。
- 学習データのサンプリング間隔と回帰精度の関係についての考察。
- 学習制御系としての応用。

が挙げられる。

### 参考文献

- [1] Corinna Cortes and V. N. Vapnik : Support-Vector Networks, Machine Learning, 20 (1995)
- [2] V. N. Vapnik : The Nature of Statistical Learning Theory, Springer, New York (1995)
- [3] M. Aizerman, E. Braverman, and L. Rozonoer : Theoretical foundations of the potential function method in pattern recognition learning, Automation and Remote Control 25: 821837 (1964)
- [4] 小林正幸, 小西康夫, 石垣博行 : サポートベクター回帰モデルによる倒立振子の学習制御, 日本機械学会 Dynamics and Design Conference 2003, No.03-7 (2003)
- [5] Francesco Parrella : Online Support Vector Regression, A Thesis presented for the degree of Information Science, Department of Information Science, University of Genoa, Italy (2007)
- [6] Kuhn, H. W. and Tucker, A. W. : Nonlinear programming, Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, pp.481-492 (1951)
- [7] F. Girosi : An equivalence between sparse approximation and Support Vector Machines, Neural Computation, 10(6):1455-1480 (1998)

内部状態と行動をベースとした

平成二二二年度入学

杉本 大志

ロボットの内部状態予測手法の提案

情報電子工学系専攻