



ロボットにおける状態行動対予測に関する研究

メタデータ	言語: jpn 出版者: 公開日: 2016-06-08 キーワード (Ja): キーワード (En): 作成者: 杉本, 大志 メールアドレス: 所属:
URL	https://doi.org/10.15118/00008926

平成 27 年度 生産情報システム工学特別研究 博士後期課程論文

研究テーマ

ロボットにおける
状態行動対予測に関する研究

室蘭工業大学 大学院工学研究科 生産情報システム工学専攻
認知ロボティクス研究室

杉本 大志 (学籍番号: 13092001)

提出日付 平成 28 年 2 月 25 日

研究期間 平成 25 年 4 月 1 日 ~ 平成 27 年 2 月 25 日

目次

第 1 章	序論	1
1.1	はじめに	1
1.2	従来研究	2
1.2.1	機械学習	2
1.2.1.1	教師付き学習	2
1.2.1.2	教師なし学習	3
1.2.1.3	半教師付き学習	3
1.2.1.4	強化学習	4
1.2.2	従来研究とその問題点	4
1.3	本研究の目的	6
1.4	本研究のアプローチ	6
1.5	従来研究に対する位置付けとして	7
1.6	本論文の構成	7
第 2 章	本研究で扱う基礎技術	9
2.1	Kernel Trick	9
2.2	Support Vector Machine	12
2.3	Support Vector Regression	12
2.4	Online SVR	14
2.5	現代制御論	17
2.5.1	概要	17
2.5.1.1	可制御性	18
2.5.1.2	可観測性	19

2.5.2	状態フィードバック制御	19
2.5.3	制御系の状態方程式と安定性	20
2.5.4	安定とは	22
2.5.4.1	安定の定義	22
2.5.4.2	正定と準正定	24
2.5.5	最適レギュレータ	24
2.5.6	サーボ制御	26
第 3 章	状態行動対予測と行動補正の考え方	27
3.1	逐次状態予測 – 1 サンプル時刻先の状態予測	27
3.2	追加学習を組合わせた逐次状態予測	32
3.3	長期状態予測 – n サンプル時刻先の状態予測	33
3.4	状態行動対予測への拡張	36
3.4.1	Online SVR を用いた状態行動対予測	36
3.4.2	Online SVR と LQR を用いた状態行動対予測	36
3.5	状態行動対予測を利用した行動補正	38
3.5.1	ロボットに於ける行動補正の考え方	38
3.5.2	行動補正で用いる重み係数の設計	39
3.5.2.1	固定の重み係数を設計する	39
3.5.2.2	動的に変更できる重み係数を設計する (1)	40
3.5.2.3	動的に変更できる重み係数を設計する (2)	42
3.5.2.4	行動補正の安定性について	43
3.6	まとめ	43
第 4 章	提案システムと処理の流れ	45
4.1	システム構成	45
4.1.1	N -ahead State-Action Pair Predictor ブロック	45
4.1.2	State-Feedback Stabilizer ブロック	47
4.1.3	Action Decision Maker ブロック	47
4.1.4	Robot (Inverted Pendulum) ブロック	48
4.2	LEGO MINDSTORMS NXT	48
4.2.1	NXTway-GS	48

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	49
4.3.1	NXTway-GS のモデルと物理パラメータについて	49
4.3.2	NXTway-GS のモデリング	52
4.3.3	状態方程式の導出	56
4.3.4	コントローラ的设计	59
	4.3.4.1 入出力	59
	4.3.4.2 安定性	60
	4.3.4.3 設計	60
4.4	まとめ	61
第 5 章	実験 I: 倒立振子に対する状態行動対予測の適用	63
5.1	実験の目的	63
5.2	事前実験	64
5.2.1	システム構築	64
5.2.2	学習データの取得	65
5.3	検証実験	66
5.3.1	実験設定	66
	5.3.1.1 Online SVR のみで状態行動対予測を実施した場合	66
	5.3.1.2 Online SVR と LQR を組み合わせて状態行動対予測を実施した 場合	66
5.3.2	実験結果とその検討	68
	5.3.2.1 Online SVR のみで状態行動対予測を実施した場合	68
	5.3.2.2 Online SVR と LQR を組み合わせて状態行動対予測を実施し た場合	73
5.4	まとめ	78
第 6 章	実験 II: 倒立振子に対する行動補正 -固定係数による信頼度付き状態行動対予 測結果を 行動決定に反映した場合-	79
6.1	実験の目的	79
6.2	事前実験	80
6.2.1	システム構築	80
6.2.2	学習データの取得	81

6.3	検証実験	82
6.3.1	実験設定	82
6.3.2	実験結果とその検討	83
6.4	まとめ	86
第7章	実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対 予測結果を 行動決定に反映した場合-	87
7.1	実験の目的	87
7.2	アクチュエータへ予測可能な外乱を持続的に与え続けた場合	88
7.2.1	事前実験	88
7.2.1.1	システム構築	88
7.2.1.2	学習データの取得	89
7.2.2	検証実験	90
7.2.2.1	実験設定	90
7.2.2.2	実験結果とその検討	92
7.3	外部コマンドとして前進後退指令を与え路面の形状を変化させた場合	95
7.3.1	事前実験	95
7.3.1.1	システム構築	95
7.3.1.2	学習データの取得	96
7.3.2	検証実験	97
7.3.2.1	実験設定	97
7.3.2.2	実験結果とその検討	99
7.4	外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合	102
7.4.1	事前実験	103
7.4.1.1	システム構築	103
7.4.1.2	学習データの取得	104
7.4.2	検証実験	104
7.4.2.1	実験設定	104
7.4.2.2	実験結果とその検討	106
7.5	まとめ	109
第8章	結論	111

目次

1.1	状態から行動を予測する	5
1.2	状態と行動を関連付けて予測する	6
2.1	Kernel Trick と SVR の組み合わせによる, 非線型関数の回帰	10
2.2	SVR の概要	13
2.3	Parrella が提案した手法の概要	15
2.4	典型的な状態空間モデル	18
2.5	可制御と可観測な系	19
2.6	状態フィードバックによる制御系の例	20
2.7	閉ループ極の望ましい位置	22
2.8	リアプノフの意味での安定	23
2.9	漸近安定	23
3.1	Parrella が提案した手法の概要 (図 2.3 の再掲)	28
3.2	状態と行動から未来を予測する考え方	29
3.3	追加学習を組合わせた逐次状態予測	32
3.4	長期状態予測の為のブロック図	35
3.5	逐次状態予測部のブロック図	35
3.6	Online SVR のみを用いた予測システムの概要	37
3.7	Online SVR と LQR を用いた予測システムの概要	37
3.8	状態行動対予測によって得られた予測値列を使い 所望の挙動へ現在の行動を 補正する	39
3.9	状態行動対予測によって得られる予測値列	40
3.10	状態行動対予測によって得た予測値のばらつき	41

4.1	状態行動対予測を用いた 行動補正システム	46
4.2	状態行動対予測による予測システム	46
4.3	Online SVR と LQR を用いた予測システムの概要 (図 3.7 の再掲)	47
4.4	自走式二輪型倒立振子 NXTway-GS の概観	48
4.5	自走式二輪型倒立振子のモデル	50
4.6	自走式二輪型倒立振子の側面・平面図	50
4.7	NXTway-GS の入出力	60
4.8	NXTway-GS 用サーボ制御器のブロック線図	60
4.9	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)	61
5.1	NXTway-GS の側面・平面図 (図 4.6 の再掲)	64
5.2	事前実験の為のシステム構成 (1)	65
5.3	モデルへ与える外乱入力 $\mathbf{d}(t)$ (1)	65
5.4	状態 $x_{1,1}$ (車輪回転角度 θ) の予測結果 (Online SVR のみのとき)	69
5.5	状態 $x_{1,2}$ (ピッチ角度 ψ) の予測結果 (Online SVR のみのとき)	69
5.6	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測結果 (Online SVR のみのとき)	70
5.7	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測結果 (Online SVR のみのとき)	70
5.8	行動 u (コントローラからの最適入力) の予測結果 (Online SVR のみのとき)	71
5.9	状態 $x_{1,1}$ (車輪回転角度 θ) の予測誤差 (Online SVR のみのとき)	71
5.10	状態 $x_{1,2}$ (ピッチ角度 ψ) の予測誤差 (Online SVR のみのとき)	71
5.11	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測誤差 (Online SVR のみのとき)	72
5.12	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測誤差 (Online SVR のみのとき)	72
5.13	行動 u (コントローラからの最適入力) の予測誤差 (Online SVR のみのとき)	72
5.14	状態 $x_{1,1}$ (車輪回転角度 θ) の予測結果 (Online SVR と LQR を組み合わせたとき)	73
5.15	状態 $x_{1,2}$ (ピッチ角度 ψ) の予測結果 (Online SVR と LQR を組み合わせたとき)	73
5.16	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測結果 (Online SVR と LQR を組み合わせたとき)	74
5.17	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測結果 (Online SVR と LQR を組み合わせたとき)	74
5.18	行動 u (コントローラからの最適入力) の予測結果 (Online SVR と LQR を組み合わせたとき)	75

5.19	状態 $x_{1,1}$ (車輪回転角度 θ) の予測誤差 (Online SVR と LQR を組み合わせたとき)	75
5.20	状態 $x_{1,2}$ (ピッチ角度 ψ) の予測誤差 (Online SVR と LQR を組み合わせたとき)	76
5.21	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測誤差 (Online SVR と LQR を組み合わせたとき)	76
5.22	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測誤差 (Online SVR と LQR を組み合わせたとき)	76
5.23	行動 u (コントローラからの最適入力) の予測誤差 (Online SVR と LQR を組み合わせたとき)	77
6.1	NXTway-GS の側面・平面図 (図 4.6 の再掲)	80
6.2	事前実験の為のシステム構成 (2)	81
6.3	モデルへ与える外乱入力 $\mathbf{d}(t)$ (2)	81
6.4	状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (1)	84
6.5	状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (1)	84
6.6	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (1)	84
6.7	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (1)	85
6.8	行動 u (コントローラからの最適入力) に対する補正結果 (1)	85
7.1	NXTway-GS の側面・平面図 (図 4.6 の再掲)	89
7.2	事前実験の為のシステム構成 (3)	89
7.3	モデルへ与える外乱入力 $\mathbf{d}(t)$ (3)	90
7.4	状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (2)	92
7.5	状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (2)	93
7.6	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (2)	93
7.7	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (2)	94
7.8	行動 u (コントローラからの最適入力) に対する補正結果 (2)	94
7.9	NXTway-GS の側面・平面図 (図 4.6 の再掲)	95
7.10	事前実験の為のシステム構成 (4)	96
7.11	人間からモデルへ与える指令入力 (1)	97
7.12	二輪型倒立振子が自走する環境 (床の形状)	97
7.13	状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (3)	99
7.14	状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (3)	100

7.15	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (3)	100
7.16	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (3)	100
7.17	行動 u (コントローラからの最適入力) に対する補正結果 (3)	101
7.18	二輪型倒立振子の走行位置の x 軸上に於ける時間変化	101
7.19	二輪型倒立振子の重心位置の z 軸上に於ける時間変化	102
7.20	NXTway-GS の側面・平面図 (図 4.6 の再掲)	103
7.21	事前実験の為のシステム構成 (5)	104
7.22	人間からモデルへ与える指令入力 (2)	104
7.23	状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (4)	107
7.24	状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (4)	107
7.25	状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (4)	108
7.26	状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (4)	108
7.27	行動 u (コントローラからの最適入力) に対する補正結果 (4)	109
7.28	LQR のみの場合との移動距離の差異	109

表目次

4.1	センサの特性	49
4.2	アクチュエータの特性	49
4.3	NXTway-GS の物理パラメータ	51
5.1	事前実験に係わる実験パラメータ (1)	66
5.2	検証実験時の Online SVR に於ける学習パラメータ	67
5.3	検証実験時の Online SVR 及び LQR に於けるパラメータ (1)	68
6.1	事前実験に係わる実験パラメータ (2)	82
6.2	検証実験時の Online SVR 及び LQR に於けるパラメータ (2)	83
6.3	検証実験に係わる実験パラメータ (1)	83
7.1	事前実験に係わる実験パラメータ (3)	90
7.2	検証実験時の Online SVR 及び LQR に於けるパラメータ (3)	91
7.3	検証実験に係わる実験パラメータ (2)	91
7.4	事前実験に係わる実験パラメータ (4)	97
7.5	検証実験時の Online SVR 及び LQR に於けるパラメータ (4)	98
7.6	検証実験に係わる実験パラメータ (3)	98
7.7	事前実験に係わる実験パラメータ (5)	105
7.8	検証実験時の Online SVR 及び LQR に於けるパラメータ (4)	105
7.9	検証実験に係わる実験パラメータ (3)	106

第 1 章 序論

本研究では、ロボットが採った行動によって、そのロボットが置かれた状態が変わる様な場合を想定し、この場合に対してロボットが適切な行動が選択出来るような方策の提案を考える。

本章では、本研究の背景と目的、内容、従来研究との差異と、本研究の位置づけ、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, 4, 5, 6, 7, 8, 9, 10, 11]. 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 [11]. 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 [4, 12, 13, 14]. 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 [14]. 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』場合がある為だ. 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 [15]. この時『脳』が意識すべきは歩行、即ちその時採る行動と環境であり、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』という行動決定は無意識で行われる. この予測からの行動決定によって、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と言える.

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、ロボットに対し予測の仕組みを取り入れ、環境に適した行動を自律的に学習させるものが複数存在している。

1.2 従来研究

動的環境に於いてロボットがタスクを達成する場合、そのロボットは環境を認識して行動を決定する事によって、その変化に対応しつつタスクを達成する能力を求められる。このような問題に対しては、ロボットの制御そのものに焦点を当てた研究や、パラメータの観点から論じる研究が挙げられる。それらの研究については後述するが、基礎となる学習手法については、次の機械学習が主となっている。

1.2.1 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)^[16, 17] 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、人間が自然と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。

1.2.1.1 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、主に『状態予測』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。

1.2.1.2 教師なし学習

教師なし学習 (unsupervised learning) は、文字通り教師が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^[18]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^[19]、自己組織化マップ^[20]等の様に離散化しクラス分けするものの2つがある。

1.2.1.3 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。

1.2.1.4 強化学習

強化学習 (reinforcement learning) とは [21], 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。

1.2.2 従来研究とその問題点

未知環境に於ける動作に関する研究は, これまでも提案がなされてきている [22, 23, 24, 25]。いくつかの研究に於いては, 先述の様な動的な環境の制御に於いて, ロボットの状態やパラメータに着眼し, この視点からロボット制御に関する知見を与えている [26, 27]。この様に, ロボットを動的環境下で制御するには, ロボットの状態を知ると共に, 行動計画の中で次にロボットがどのような行動を採るかを知る事も必要である。この時, 状態と行動を知るには, 置かれた環境の変化と共に, ロボットの正確なモデルを知る必要がある。この問題に対しては, 大まかに次の 4 点に従来行われている研究を分類出来る。

1 点目として, 強化学習を行動獲得に用いる分野が存在する [28, 29]。但し, 局所解に対する回避策や学習時間に対する課題が残されている。2 点目として, 対象となるシステムをプラントと見做す事で, 制御手法として一般性が非常に高く, 逐次的に最適入力を獲得するモデル予測制御 (MPC) の考え方が用いられている [30, 31, 32]。これらの手法に於いては, 計算コストによるハードウェア負荷と計算遅延, 或いは動作する環境の変化に柔軟に対応出来るかという点に於いて, 議論の余地が残っている [33, 34, 35, 36, 37]。その一方 3 点目として, Extended Kalman Filter (EKF)^[1, 38] や Unscented Kalman Filter (UKF)^[1] を組み合わせる事で, モデルの線型化を回避してロボットの動作生成を実現している手法が存在する [39, 40, 41, 42]。しかし, 用いるフィルタがしばしば不安定となる場合や, 非線形モデルをそのまま扱う事が難しい点, そして定めるべきパラメータが多くなる点や計算時間の問題が未だ残されている。続いて 4 点目として,

Gaussian Mixture Regression (GMR) の様なノンパラメトリック回帰手法を用い, 対象が持つ非線形性を近似するフレームワークを用いる事で, ロボット制御への応用を目指した研究も存在する [43, 44]. この場合, 時間に依存しないモデルを学習する事が出来るが, 推定すべきパラメータの数が膨大になるという点もある.

他方, これに関連する研究として, サポートベクトルマシン (SVM)^[45] やサポートベクトル回帰 (SVR)^[45, 46] を適用したものが挙げられ [30, 31, 47, 48, 49, 50, 51, 52], これらはロバストな結果を示している. しかし, 当該する研究に於いては外乱を考慮しておらず, またロボットの状態変化や行動の関係を検討に含めていない.

これを端的に示したものが図 1.1 である. この様に, 従来研究についてまとめると, 状態を用

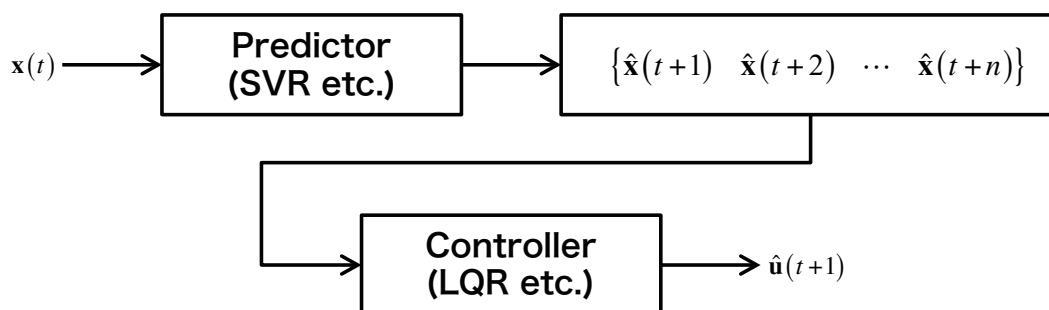


図 1.1 状態から行動を予測する

いる事で制御入力, 即ち行動を間接的に予測する事で制御を行うか, 或いは純粹に予測そのものについて検討が為されているが, 何れも状態と行動の関係性に焦点を当て, これら両者を関連付けて予測しているものではない.

動的な環境を考える. この様な環境では, 例として予期せぬ負荷やバックラッシ, 或いは静摩擦や動摩擦の発生が想定出来る. この時, ロボットが決定した行動が, ハードウェア的な制約条件によって遅延する事もあり, 結果として, その行動自体が, その状態に即した行動が採れるとは必ずしも言えない事となる. 従って, 従来手法によって『状態予測からの行動決定』を実現する事は困難である.

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 [28]. 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 [53, 54] 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を来す事となる。

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。

1.3 本研究の目的

ロボットに対して予測の技法を適用する例と問題点については、前小節までに述べた。これを以て、ロボットの状態や行動を学習し予測する考え方が不可欠である事を示した。

そこで本研究では、状態と行動を関連付ける事を考える。つまり、現在の状態と行動から、未来の状態を予測する手法である、状態行動対予測を提案する。これによって、自身の採る行動によって変化する状態を予測する事が可能となり、ロボットの動的変化に対する適応も可能となると言える。そして、未来の状態と行動の予測結果から、現在採るべき行動を決める方策も提案する。

1.4 本研究のアプローチ

そこで、本研究では、ロボットが持つ状態と採る行動を関連付け、この両者の特徴を学習する事によって、未来の状態と行動を予測し、その結果に基づき決定する (図 1.2)。この状態の予測

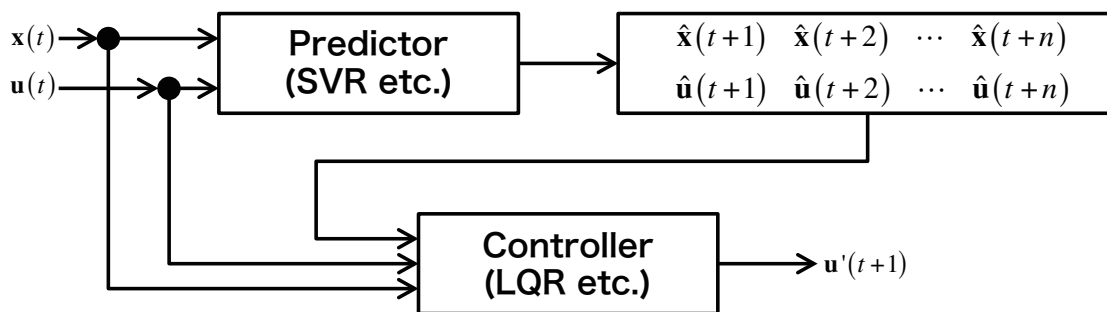


図 1.2 状態と行動を関連付けて予測する

を実現する上で、本研究では文献 [28] で述べられている様に、状態と行動によって新たな状態

に遷移する点を踏まえ、ロボットが現時点で持つ状態と採る行動から、次にロボットが遷移する状態を予測する仕組みを考える。また、未来の状態は、現在の状態と行動の他にも、過去に経験した行動や状態にも依存するものと考えられる。そこで、状態の予測を行う為に、後述する 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、未来の状態の予測と行動の決定を実現する。

1.5 従来研究に対する位置付けとして

本研究で対象とするロボットは、動作点の近傍で線型化出来る二輪型倒立振子とする。また、本研究が提唱する状態行動対予測では、Online SVR を用いる。これは Parella が提案した Online SVR^[46] に基づいている。先に提案したこの手法は、ロボットの状態と、その際に採る行動を状態行動対として拡張の上定義する。状態と行動を交互に予測する事で、遠い将来のロボットの状態や行動の予測結果が得られる。この方法を用いる事により、我々はロボットが将来的にとられる事に適した行動が得られる。先述した各種の従来手法は、パラメトリックな考え方に基づいている。しかし我々が今回提案する手法では、ノンパラメトリックな立場に基づいている。この点で提案手法は、対象ロボットに対する厳密なパラメータまたはモデルを必要としない^[55]。この違いにて、提案手法は従来手法に比べ有意な差があるものとする^[55]。

また、この考え方を提案する事によって、モデル予測制御とは異なる形で、ロボットがおかれた状態を考慮した行動選択をより迅速に決定出来るものとする。

1.6 本論文の構成

本論文は 8 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、状態と行動の予測を行う手法として、主に取り扱う“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。また、行動を決定する際に用いる最適レギュレータについても、現代制御論の点から説明する。

第 3 章 本研究が意図するロボットに対する予測の必要性を示すとともに、これを実現するために本研究が提唱する『状態行動対予測』の概念を述べる。また、この状態行動対予測を

第 1 章 序論

実現するために必要となる, Online SVR の改良について解説する. そして, 状態フィードバック制御の考え方をを用いた状態行動対予測についても説明する. 併せて, 状態行動対予測を用いて, ロボットが採るべき最適な行動を常に獲得する考え方を説明する. ここでは, 未来の予測値を考慮した優先度付き行動補正を考える.

第 4 章 前章で説明した手法をシステムとして構成した際の, 各サブシステムの機能とその概要について概説する.

第 5 章 本研究では, 二輪型倒立振子に対して提案手法を各々適用し, その妥当性を検討する. そこで本章では, 自走式二輪型倒立振子 “NXTway-GS” に対して状態行動対予測を適用し, 計算機シミュレーションによって, この提案手法の有効性を検証する.

第 6 章 本章では, 状態行動対予測より得た未来の予測値を用いて, 将来採るべき行動を考慮した最適行動を現時刻で獲得することを考える. そこで, 前章と同じく倒立振子に対して状態行動対予測を適用すると共に行動補正を組み合わせる. そして計算機シミュレーションによってその有効性を検証する. 特に本章では, 未来の予測値に対し固定の重み係数を乗じることによって, 予測誤差の影響を低減する事を試みる.

第 7 章 本章では, 状態行動対予測より得た未来の予測値を用いて, 将来採るべき行動を考慮した最適行動を現時刻で獲得することを考える. そこで, 前章と同じく倒立振子に対して状態行動対予測と行動補正を組み合わせ, 現時点に於ける行動を補正する. そして計算機シミュレーションによってその有効性を検証する. 特に本章では, 未来の予測値の統計的特性に着目し, 動的に重み係数を決定する. この重み係数を行動決定に直接的に作用させる事により, 予測誤差の影響低減に寄与する事を確認する.

第 8 章 本研究で得られた結論を総括すると共に, 本研究で提案した手法の応用範囲に関する検討や, 今後の課題について展望してゆく.

第 2 章 本研究で扱う基礎技術

本章では, 本研究で提案する状態行動対予測で必要となる『状態』と『行動』を組み合わせて予測する為の手法として必要となる, “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. また, 未来に於ける行動を決定或いは予測する際に必要となる最適レギュレータについても, 現代制御論の点から説明する.

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.

線型空間 Ω 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 ϕ に写像し, ϕ 上で SVR により線型回帰を行う. その後, ϕ から Ω 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

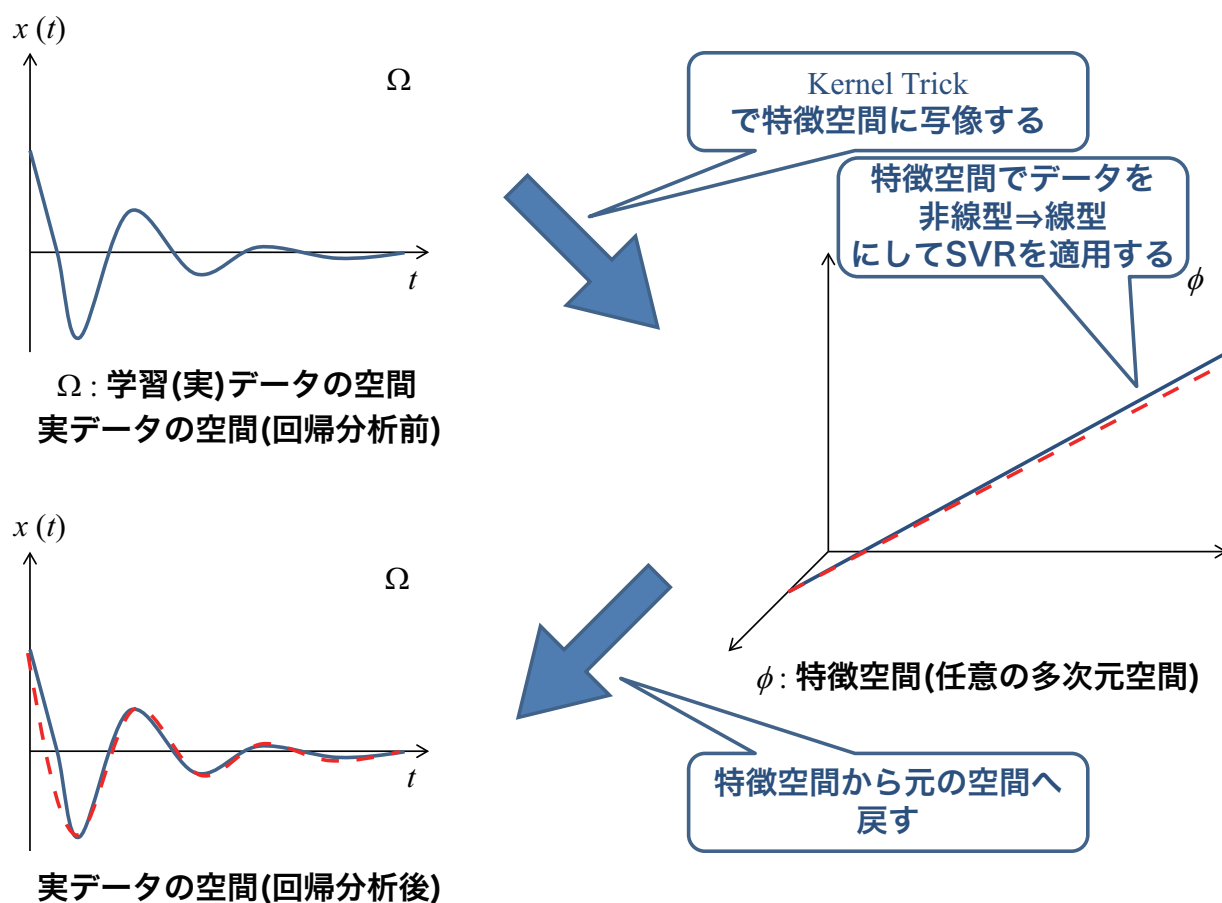


図 2.1 Kernel Trick と SVR の組み合わせによる、非線型関数の回帰

Kernel Trick^[56] 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^[45]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**^[56] が知られている。この方法を用いる事で、SVM/SVR や Neural Network^[57]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、どのよう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、 $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \ \phi_2(\mathbf{x}) \ \dots \ \phi_r(\mathbf{x})]^\top$ 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 Φ 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。

Φ 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 [58]。

定理 (**Mercer** の定理) χ 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、

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

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

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

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 [59, 60]。

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 [45]。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 [61] 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 (ϵ -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

$f(\mathbf{x}) = \mathbf{ax} + b$ 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, ϵ に依存している. つまり, この許容誤差 ϵ 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

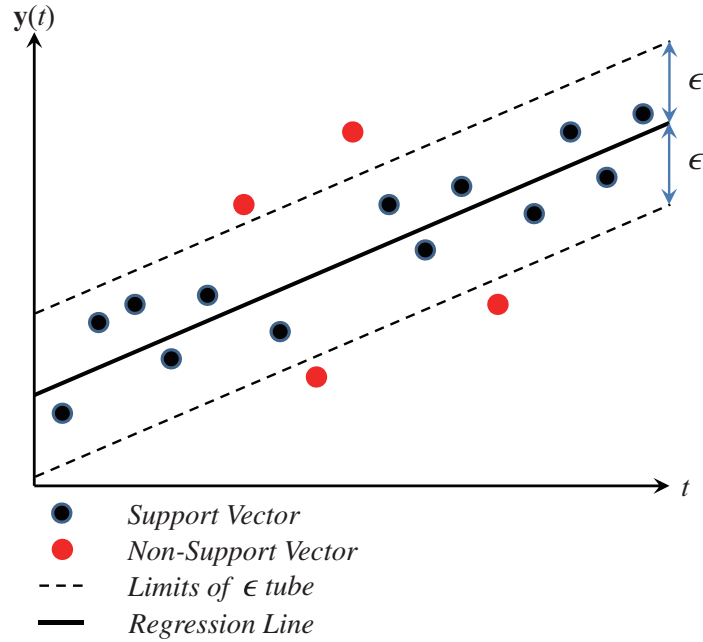


図 2.2 SVR の概要

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

と回帰直線とサンプルの残差を r として, つぎの ϵ 許容誤差 (ϵ -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, λ 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).

この最適化問題は, 次の双対問題に書き換えられる.

$$\begin{aligned} \min_{\alpha_i, \alpha_i^*} & \epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) \\ & + \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to: } & \begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq \frac{1}{\lambda} \\ \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \end{cases} \end{aligned} \quad (2.3.4)$$

この式 (2.3.3), (2.3.4) が SVR と呼ばれる. 誤差関数には ϵ 許容誤差以外に Huber 関数 [62] 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, 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 [63]. 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 [30, 31]. 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**^[46] 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 は回帰に関与するデータであるサポートベクトルを意味している.

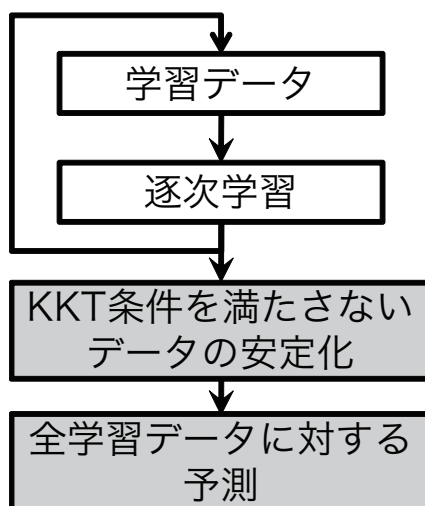


図 2.3 Parrella が提案した手法の概要

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.

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

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

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

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

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

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

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

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

ここで, Q_{ic} は Kernel 関数により構築される Kernel 行列であり, θ 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 $y = \mathbf{w}^\top \mathbf{x} + b$ 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^[64]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^[46](図 2.3, またこの操作を “Stabilization^[46]”, 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^[65]。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。

2.5 現代制御論

2.5.1 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。

$$\dot{\mathbf{x}}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad (2.5.1)$$

$$\mathbf{y}(t) = \mathbf{Cx}(t) + \mathbf{Du}(t) \quad (2.5.2)$$

式 (2.5.1) を状態方程式(state equation)、式 (2.5.2) 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 (2.5.1), (2.5.2) で記述される系をブロック線図で表現すると, 図 2.4 の様になる.

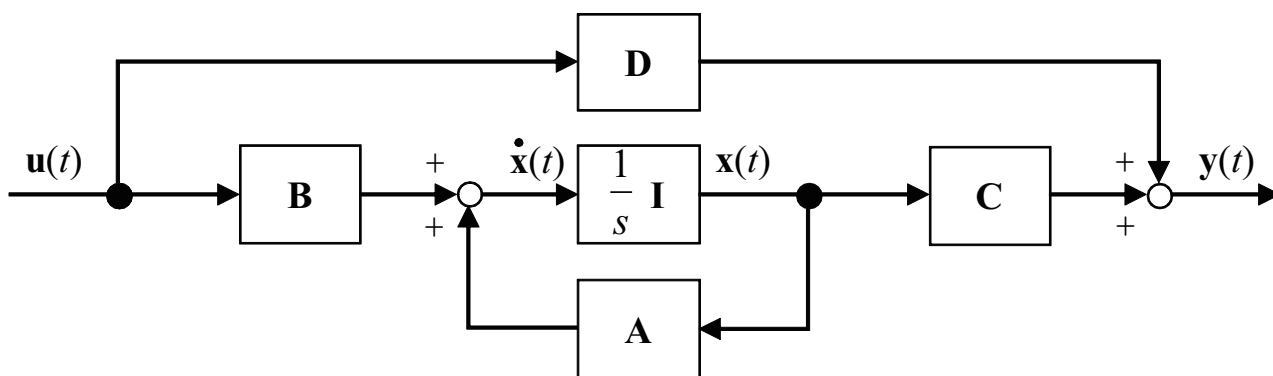


図 2.4 典型的な状態空間モデル

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 [66, 67].

以降, 制御対象のモデルとして,

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \quad (2.5.3)$$

と云う状態方程式表現を用いる.

2.5.1.1 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) と云う. 図 2.5 の場合, \mathbf{Sys}_{00} と \mathbf{Sys}_{01} は可制御である.

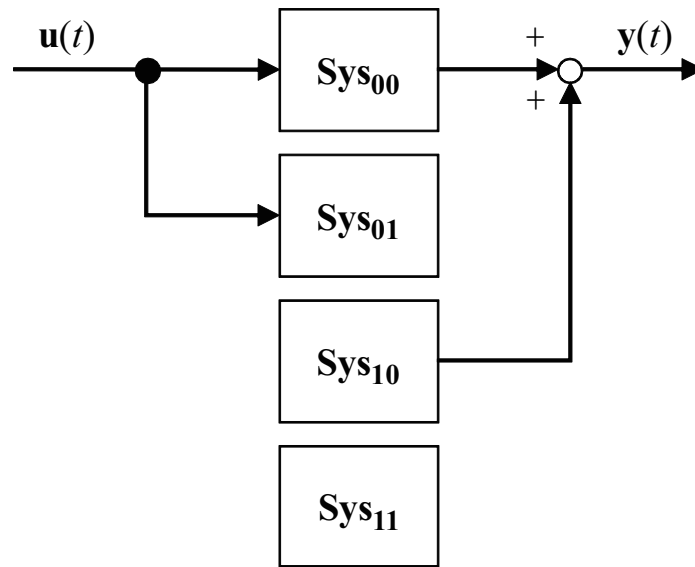


図 2.5 可制御と可観測な系

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.

$$\left[\mathbf{B} \mid \mathbf{A}\mathbf{B} \mid \mathbf{A}^2\mathbf{B} \mid \cdots \mid \mathbf{A}^{n-1}\mathbf{B} \right] \quad (2.5.4)$$

2.5.1.2 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 2.5 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.

$$\left[\mathbf{C}^T \mid \mathbf{A}^T \mathbf{C}^T \mid \mathbf{A}^{T^2} \mathbf{C}^T \mid \cdots \mid \mathbf{A}^{T^{n-1}} \mathbf{C} \right] \quad (2.5.5)$$

2.5.2 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 (2.5.3) 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 ($\mathbf{u}(t) = 0$ 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 (2.5.3) 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 (2.5.3) 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 2.4 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ある。つまり、或る定数行列を \mathbf{F} とし

$$\mathbf{u}(t) = -\mathbf{F}\mathbf{x}(t) \tag{2.5.6}$$

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 (2.5.6) 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。

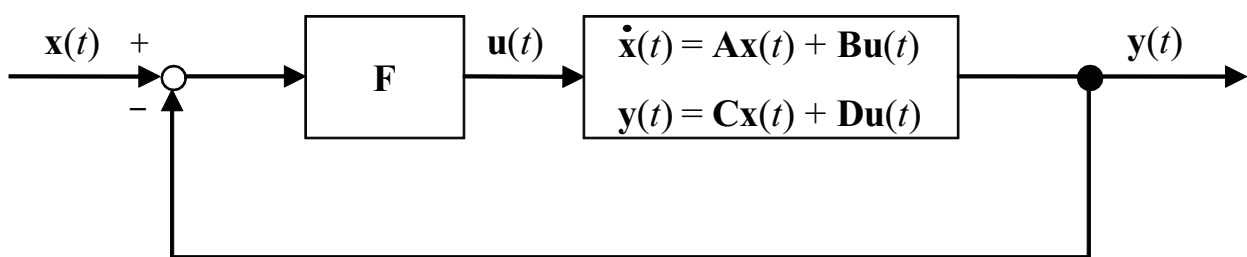


図 2.6 状態フィードバックによる制御系の例

2.5.3 制御系の状態方程式と安定性

式 (2.5.6) 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 2.6 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 (2.5.3) が成立している。また $\mathbf{u}(t)$ が式 (2.5.6) で示されるので、これを式 (2.5.3) に代入する事で

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{F}\mathbf{x}(t) \\ &= (\mathbf{A} - \mathbf{B}\mathbf{F})\mathbf{x}(t)\end{aligned}\tag{2.5.7}$$

を得る。これが図 2.6 の制御系の状態方程式である。

こうして求めた式 (2.5.7) 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

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)\tag{2.5.8}$$

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 (2.5.8) の \mathbf{A} を、 $\mathbf{A} - \mathbf{B}\mathbf{F}$ に置き換えて考えると、式 (2.5.7) 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 2.6 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 2.7 のおおよそ斜線部の辺りである。

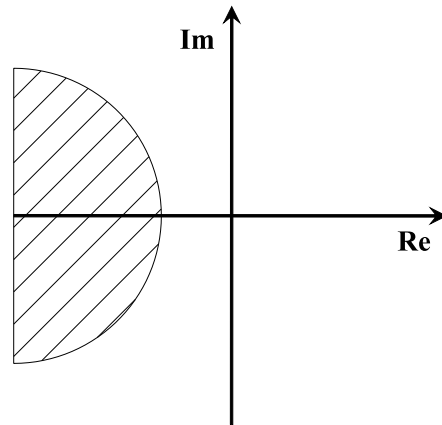


図 2.7 閉ループ極の望ましい位置

2.5.4 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

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \tag{2.5.9}$$

また非線型、時変の場合は

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}, t) \tag{2.5.10}$$

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。

2.5.4.1 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 [68]。

■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 2.8 に示す様に $t \rightarrow \infty$ となっても、或る状態空間 $s(\varepsilon_0)$ よりも外に出ない時安定と言う。即ち

$$0 \leq \mathbf{x}_0 \leq \delta \tag{2.5.11}$$

の初期条件から出発して全時間範囲に対し

$$0 \leq \mathbf{x}(t) \leq \varepsilon_0 \tag{2.5.12}$$

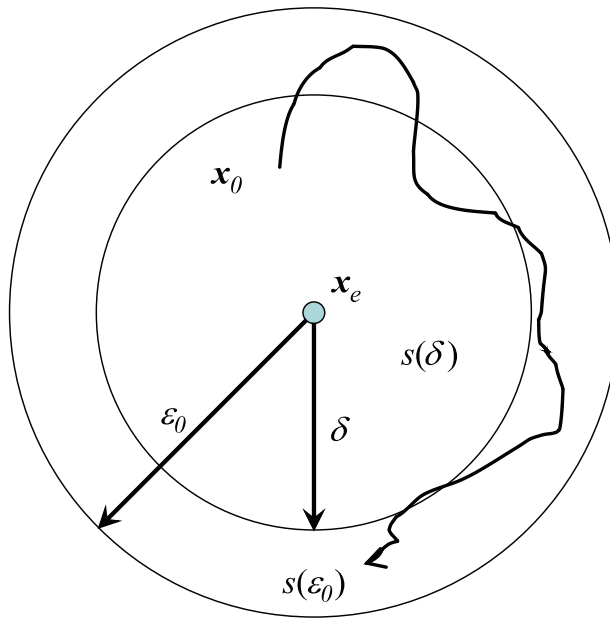


図 2.8 リアプノフの意味での安定

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 2.9 に示

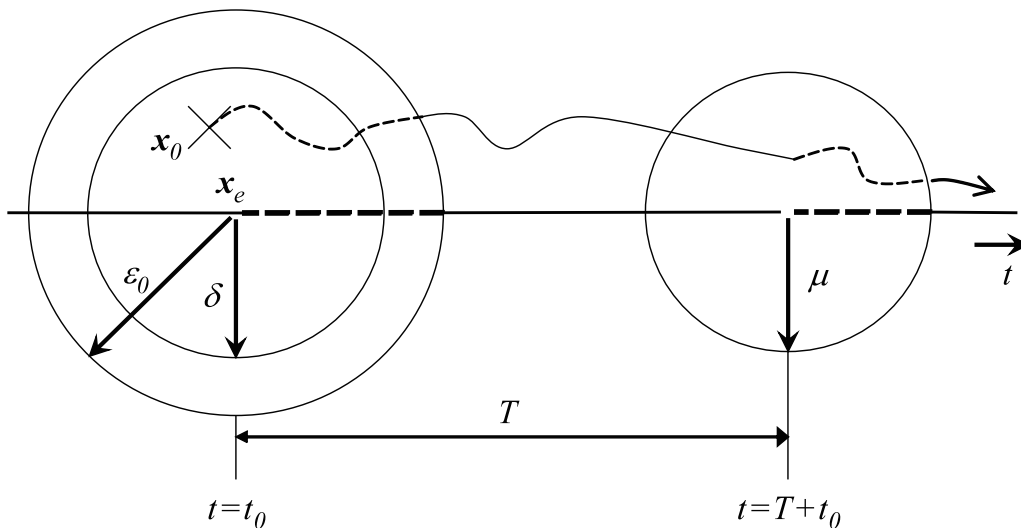


図 2.9 漸近安定

す様に $t = t_0$ に於いて

$$\|\mathbf{x}(0) - \mathbf{x}_e\| \leq \delta \tag{2.5.13}$$

より出発し, $t \geq T + t_0$ の時間では

$$\|\mathbf{x}(t) - \mathbf{x}_\varepsilon\| \leq \mu \quad (2.5.14)$$

となる場合で, μ は正実数であるので, 十分時間が経つと μ 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.

2.5.4.2 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^[69] 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.

2.5.5 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. この様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

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.5.15)$$

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

$$J[\mathbf{x}(t), \mathbf{u}(t)] = \int_0^\infty (\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)) dt \quad (2.5.16)$$

を最小にする最適制御入力

$$\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t), \mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \quad (2.5.17)$$

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)

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \quad (2.5.18)$$

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 (2.5.16) を最小にする制御入力 が式 (2.5.17) で与えられるかについて考える [70].

先ず始めに, 制御入力 (2.5.17) を系 (2.5.15) 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 (2.5.18) は

$$\mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{K}) + (\mathbf{A} + \mathbf{B}\mathbf{K})^T\mathbf{P} = -(\mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K}) \quad (2.5.19)$$

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

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P})\mathbf{x}(t) \quad (2.5.20)$$

は漸近安定となる. よって制御入力 (2.5.17) 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 (2.5.16) の積分の中を計算する.

$$\begin{aligned} \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} &= \mathbf{x}^T(\mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} - \mathbf{P}\mathbf{A} - \mathbf{A}^T\mathbf{P})\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} \\ &= \mathbf{x}^T\mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} + \mathbf{x}^T\mathbf{P}\mathbf{B}\mathbf{u} + \mathbf{u}^T\mathbf{B}^T\mathbf{P}\mathbf{x} - \mathbf{x}^T\mathbf{P}\dot{\mathbf{x}} - \dot{\mathbf{x}}^T\mathbf{P}\mathbf{x} \\ &= (\mathbf{u} + \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x})^T \mathbf{R} (\mathbf{u} + \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x}) - \dot{\mathbf{x}}^T\mathbf{P}\mathbf{x} \end{aligned} \quad (2.5.21)$$

である. ここで, $t \rightarrow \infty$ のとき $x \rightarrow 0$ である. よって, 式 (2.5.21) の両辺を 0 から ∞ まで積分すると

$$J[\mathbf{x}(t), \mathbf{u}(t)] = \mathbf{x}^T(0)\mathbf{P}\mathbf{x}(0) + \int_0^\infty \left\{ (\mathbf{u} + \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x})^T \mathbf{R} (\mathbf{u} + \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x}) \right\} dt \quad (2.5.22)$$

となる. 今, \mathbf{R} は正定であると仮定しているので

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x} \quad (2.5.23)$$

が成り立つ時 J は最小となり, 最小値は

$$\min J = \mathbf{x}^T(0)\mathbf{P}\mathbf{x}(0) \quad (2.5.24)$$

となる.

尚, 評価関数 (2.5.16) 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 (2.5.17) 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。

2.5.6 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。

第3章 状態行動対予測と行動補正の 考え方

本研究が意図するロボットに対する予測の必要性を示すとともに、これを実現するために本研究が提唱する『状態行動対予測』の概念を述べる。

我々は、文献 [65] 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、現在の行動と状態から予測する様に再定義する必要がある。そこで本章では、この状態行動対予測を実現するために必要となる、Online SVR の改良について解説する。そして、状態フィードバック制御の考え方をを用いた状態行動対予測についても説明する。

まず始めに、1 サンプル時刻先に於ける状態を予測する逐次状態予測を説明、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状態を予測する長期状態予測を定義、説明する。これらの理論を基にし、状態行動対予測の考え方を明らかにしてゆく、その後、状態行動対予測を用いて、ロボットが採るべき最適な行動を常に獲得する行動補正の考え方を説明する。ここでは、未来の予測値を考慮した優先度付き行動補正を考える。

3.1 逐次状態予測 – 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 [65] に於いて、Online SVR 上でリアルタイムに逐次予測する手

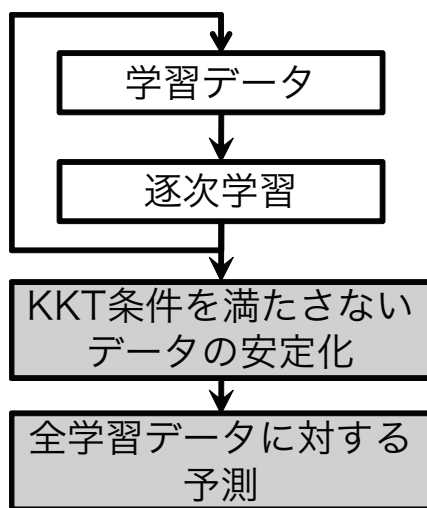


図 3.1 Parrella が提案した手法の概要 (図 2.3 の再掲)

法を提案している。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、状態と行動から、未来の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 $\mathbf{u}(t_1)$ を実行する事によって、状態は $\mathbf{x}(t_2)$ に変化する。過去の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 $\mathbf{u}(t_2)$ を実行した場合の未来の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

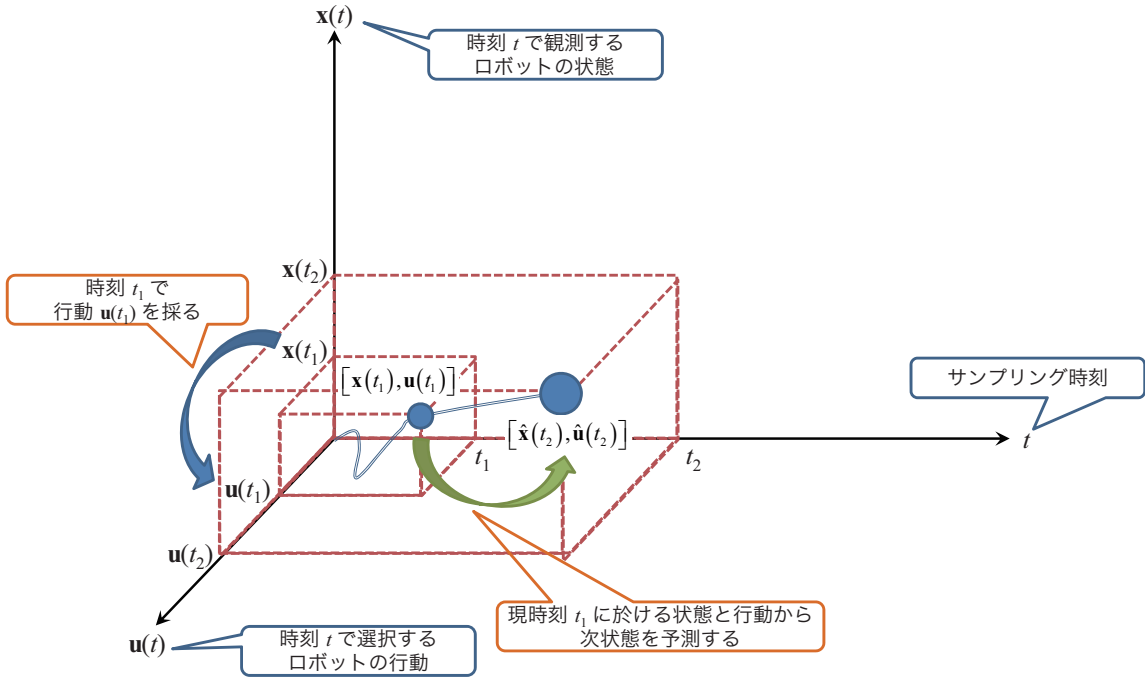


図 3.2 状態と行動から未来を予測する考え方

とめて再定義する事を考える.

$$\mathbf{z}_t = [\mathbf{x}_t \quad | \quad \mathbf{u}_{t-1}] \quad (3.1.2)$$

上式 (3.1.2) に於いて, \mathbf{x}_t は時刻 t でのロボットの状態, \mathbf{u}_t は t での行動を示している.

この時, 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状態 $\mathbf{z}_t = [x_{t,1}, \dots, x_{t,n} | u_{t-1,1}, \dots, u_{t-1,n}]$, ($u_{t-1} = 0$ when $t = 0$) 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 [71].

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

ここで, λ 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,

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

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

となる。但し、 Φ は計画行列 (design matrix)^[72] 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 Φ を用いて Gram 行列 $\mathbf{K} = \Phi\Phi^\top$ を定義する^[73]。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、

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

の様になる。これを \mathbf{a} について解くと

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

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。

$$\begin{aligned} \hat{y}(\mathbf{x}) &= \mathbf{w}\phi(\mathbf{x}) + b = \mathbf{a}^\top \Phi\phi(\mathbf{x}) + b = \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \lambda\mathbf{I}_t)^{-1} \mathbf{y} + b \\ \text{where } \mathbf{k}(\mathbf{x}) &= \left[k(\mathbf{x}_1, \mathbf{x}) \quad \dots \quad k(\mathbf{x}_j, \mathbf{x}) \right]^\top \end{aligned} \quad (3.1.9)$$

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、

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

である^[17, 71]。また、 ϵ は ϵ -insensitive 損失関数^[17, 74] 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

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

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^[46]. 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状態予測

前節では, 逐次学習による逐次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).

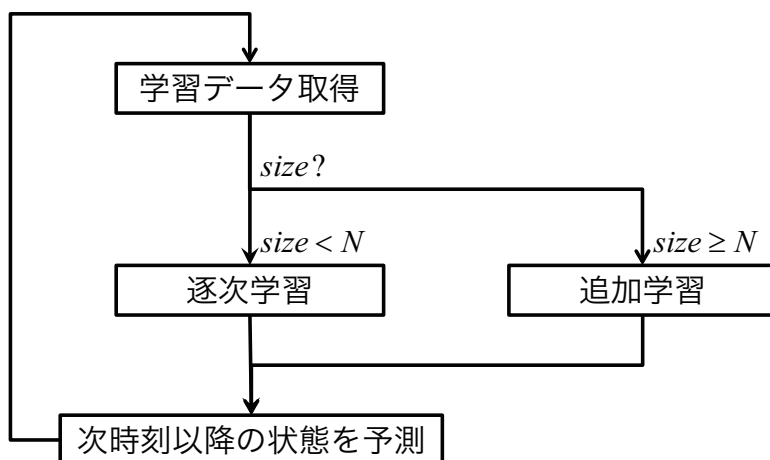


図 3.3 追加学習を組合わせた逐次状態予測

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 [46]. 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 サンプルング時刻先に於ける状態を予測する事を考える. そこで, バッチ SVR を組み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

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

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状態予測を行う事が可能となる.

3.3 長期状態予測 – n サンプルング時刻先の状態予測

本研究では, 現在の状態と行動から, 次に遷移する状態を予測する事を目的としている. つまり 1 サンプルング時刻先では, 現在の状態と現在採った行動から状態を予測出来る (逐次状態予測). この考え方を n サンプルング時刻先に於ける状態の予測にまで拡張した場合, 2 サンプルング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, 次式の様になる.

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

このモデルは, 何らかの制御によって目標状態に状態を移動する事が出来る事から, 以降の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

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

によって連続的に導かれる [75].

この式 (3.3.2) で示された考え方を, 行動と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.

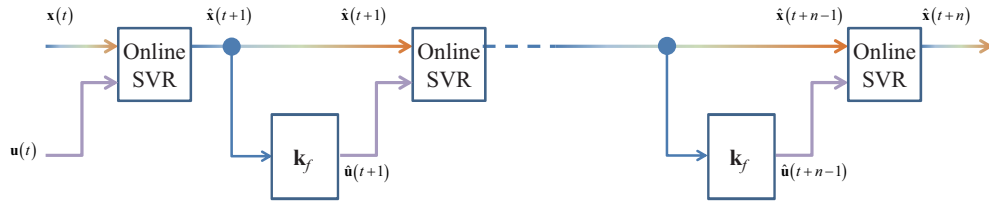


図 3.4 長期状態予測の為のブロック図

図 3.4 は、ロボットの状態と、その時採る行動から、未来の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{u}_t を与える事で、1 サンプル時刻先の状態 $\hat{\mathbf{x}}_{t+1}$ 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 $\hat{\mathbf{u}}_{t+1}$ 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

$$\mathbf{u}_{t+n-1} = \mathbf{k}_f \mathbf{x}_{t+n-2} \tag{3.3.3}$$

$$\mathbf{z}_{t+n-1} = \begin{bmatrix} \mathbf{x}_{t+n-1} & | & \mathbf{u}_{t+n-2} \end{bmatrix} \tag{3.3.4}$$

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

の様に、行動を状態を互いに導く事によって、長期的な状態予測が可能となる。

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

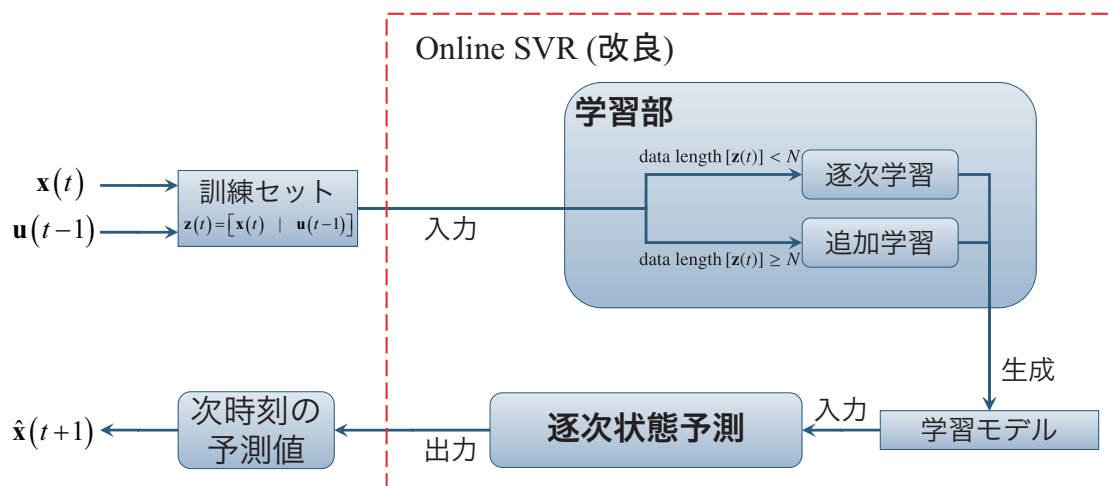


図 3.5 逐次状態予測部のブロック図

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。

3.4 状態行動対予測への拡張

前節までに示した逐次状態予測, 及び長期状態予測を組み合わせ, 状態と行動それぞれに対しての予測式を定式化する事を考える. 状態と行動を対として扱い, それぞれを予測する事によって, 状態行動対予測が実現される. 本研究では, Online SVR と LQR を組み合わせる事によって, 状態行動対予測の実現を試みた.

3.4.1 Online SVR を用いた状態行動対予測

この状態行動対予測を実現する上で, 式 (3.3.5) を拡張する. ここでは, Online SVR を状態及び行動に於ける予測器として採用する. 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. 状態 \mathbf{x}_t と \mathbf{z}_t を対とする事で ${}^x\mathbf{z}_t = \begin{bmatrix} \mathbf{z}_t & | & \mathbf{x}_t \end{bmatrix}$ を, 行動 \mathbf{u}_{t-1} と \mathbf{z}_t を対とする事で ${}^a\mathbf{z}_t = \begin{bmatrix} \mathbf{z}_t & | & \mathbf{u}_{t-1} \end{bmatrix}$ を定義する. これを式 (3.3.5) へ適用する事により, 次式を得る.

$$\hat{x}_{t+1,i} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}({}^x\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_t)^{-1} {}^x\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_t$

$$\hat{u}_{t+1} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}({}^a\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_t)^{-1} {}^a\mathbf{z}_{sv} + b' & \text{otherwise} \end{cases} \quad (3.4.2)$$

これによって, 現在の状態と行動から, 次時刻に於ける状態と行動のそれぞれを予測出来る. 式 (3.4.1) を **State Predictor**, 式 (3.4.2) を **Action Predictor** と定義する事で, 図 3.6 に示す予測システムを構築出来る.

3.4.2 Online SVR と LQR を用いた状態行動対予測

ここでは, 次時刻に於ける状態を予測する際に Online SVR を, 行動を予測する際に LQR をそれぞれ採用する. 状態 \mathbf{x}_t と \mathbf{z}_t を対とする事で ${}^x\mathbf{z}_t = \begin{bmatrix} \mathbf{z}_t & | & \mathbf{x}_t \end{bmatrix}$ を定義する. これを式 (3.3.5)

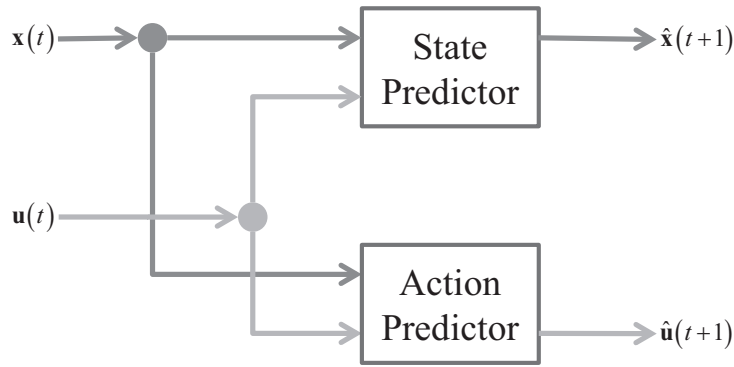


図 3.6 Online SVR のみを用いた予測システムの概要

へ適用する事により, 次式を得る.

$$\hat{x}_{t+1,i} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}(x\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_l)^{-1} x\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_t$

$$\hat{u}_{t+1} = \mathbf{k}_f \hat{\mathbf{x}}_{t+1} \quad (3.4.3)$$

予測対象のモデルが非線型離散状態空間モデルとして記述出来る時, 上式を適用出来る. ここで, \mathbf{k}_f は LQR より与えられる状態フィードバックゲインである. これによって, やはり現在の状態と行動から, 次時刻に於ける状態と行動のそれぞれを予測出来る. 式 (3.4.1) を **State Predictor**, 式 (4.1.1) を **Action Predictor** と定義する事で, 図 3.7 に示す予測システムを構築出来る.

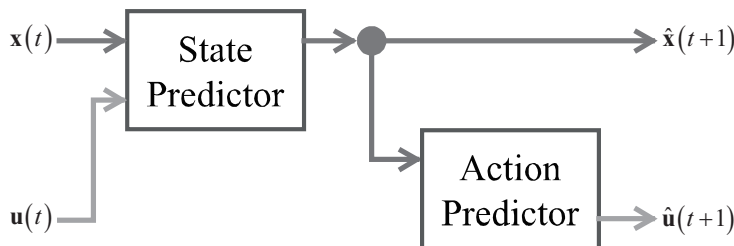


図 3.7 Online SVR と LQR を用いた予測システムの概要

3.5 状態行動対予測を利用した行動補正

前節に掛けて、状態行動対予測についての考え方を論じてきた。ここでは、状態行動対予測を従来使われてきた最適制御と組み合わせる事を考える。状態と行動の組合せで表現出来るロボット制御に対し提案手法を適用する事で、未来の行動を学習器によって予測する。そしてこの予測した未来の行動を現在採る行動に反映させる事で、行動の補正を行う仕組みを実現する。また、未来に採るべき行動の予測値に含まれる予測誤差を考慮し、生成する行動には重み付けを行う事で、その影響を低減する。この考え方を適用する事で、最適制御方法と組み合わせで予測された状態を使用して、すぐにその望ましい行動を補正し決定する手段について説明する。特に本研究では、予測によって得た将来の行動に対して重み係数を設計し、これを基本として今後採るべき行動を決定する。

ここでは、大まかに

- (i) 固定の重み係数を設計する
- (ii) 予測結果の『ばらつき』に注目し、重み係数を動的に変更出来る様設計する

2通りの仕組みを考える。また、古い時刻に於いて予測した結果に対しては重要度を下げる様な仕組みについても同時に検討する。この考え方を実現する事によって、現在の行動をより柔軟に補償出来るものと考えられる。行動補正に於いては **Action Predictor** には LQR を適用し、式 (4.1.1) に従い予測値を導出する。

3.5.1 ロボットに於ける行動補正の考え方

動的環境に於けるロボット制御では、これまでの行動と状態を基にして未来の状態を予測し、その結果に即した行動を直ちに用いる事により、最適な動作が実現出来ると言える。ここでは、状態行動対予測を利用する事で予測可能な外乱が与えられた場合の、倒立振子の車体傾斜角(以下、ピッチ角)の変化を小さく留める様な最適行動を獲得する事を考える。そこで、外乱を補正する上で必要となる行動を生成する為に、図 3.8 の様に、状態行動対予測で鰐予測結果を用いて現在の行動を補正する事を考える。本研究では、適用例として自走式二輪型倒立振子の倒立姿勢維持制御を想定しており、状態フィードバックゲイン \mathbf{k}_f によって最適制御を行うと共に、状態行動対予測によって未来に採るべき行動を組合せて、倒立振子への制御入力を補正する事を考える。状態行動対予測を適用する事によって、現時刻 t から任意の $t+N$ 時刻先までの遠い

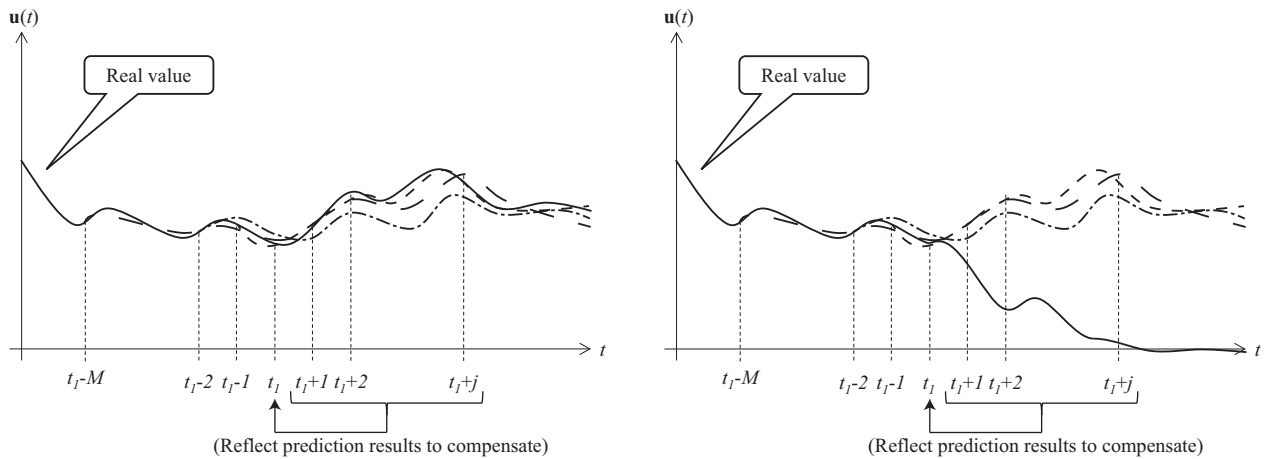


図 3.8 状態行動対予測によって得られた予測値列を使い 所望の挙動へ現在の行動を補正する

未来にかけて採られるであろう行動の予測結果の集合 (予測値列) が得られる. この予測値列を使う事により, 現時刻で決定した行動に未来で採る行動を加えて補正する事が可能となる. ただし, 予測結果から得られた行動を実際に用いる場合, 予測の対象となる時刻が現時刻に比べて相対的に遠くなると, その長さに比例して予測誤差が無視出来ない程度に蓄積される. そこで, この点を考慮しつつ行動を補正する手段を考える.

3.5.2 行動補正で用いる重み係数の設計

先述の様に, 状態行動対予測を時刻 t にて適用する事で, 現時刻 t から任意の $t+N$ 時刻先までの遠い未来にかけて採られるであろう行動の予測値列が得られる. この予測値列を使う事により, 現時刻で決定した行動に未来で採る行動を加えて補正する事が可能となる. その一方で, 予測結果を用いる場合, 予測の対象となる時刻が現時刻に比べて相対的に遠くなると, その長さに比例して予測誤差が無視出来ない程度に蓄積される. そこで補正に用いる予測結果に対し重み付けを行い, 補正に用いる値の重要度を設定すると共に予測誤差の影響を低減する事を試みる. また, 用いる予測値に於いても, 予測を始めた時刻にも注目し, 重み付けする事考える.

3.5.2.1 固定の重み係数を設計する

補正に用いる予測結果に対し重み付けを行い, 補正に用いる値の重要度を設定すると共に予測誤差の影響を低減する事考える. つまり, 現時刻より遠い未来の時間で採る行動程, その行動に対する重要度を低下させる. 即ち, 或る時刻 t にて, 未来の時刻 $t+1$ から $t+N$ にかけて

予測した行動 $\hat{\mathbf{u}}_{t+i}, 1 \leq i \leq N$ に対して

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t+i) \quad (3.5.1)$$

と定義する. 上式の α_i は, 各時刻に於ける予測値に対する重みである. この $\mathbf{u}_s^{(t+1, t+N)}(t)$ が, 行動補正に用いる未来の行動を生成するブロック **Action Decision Maker** の機能となる. この時, **Action Decision Maker** への入力となる *N-ahead State-Action Pair Predictor* による行動の予測値には, 外乱 $\mathbf{d}(t)$ が明示的に含まれている. この点に着目すると, 外乱を前もって対応出来る様な行動を作り出す事で, 未来に於いて最適な行動を獲得出来るものと考えられる. つまり図 3.8 内の **Action Decision Maker** に該当するブロックによって決定した行動 $\mathbf{u}_s^{(t+1, t+N)}(t)$, および状態フィードバックが計算した行動 $\mathbf{u}_p(t)$ を使う事で,

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.2)$$

となる. ここで, 重み α_i は, 次式によって決定する. C_f は任意の整数とする.

$$\alpha_i = \frac{N - i + 1}{C_f N} \quad (3.5.3)$$

これによって, 現時刻より遠い時間に位置する予測値ほど重要度を低下させる重みを定義出来る.

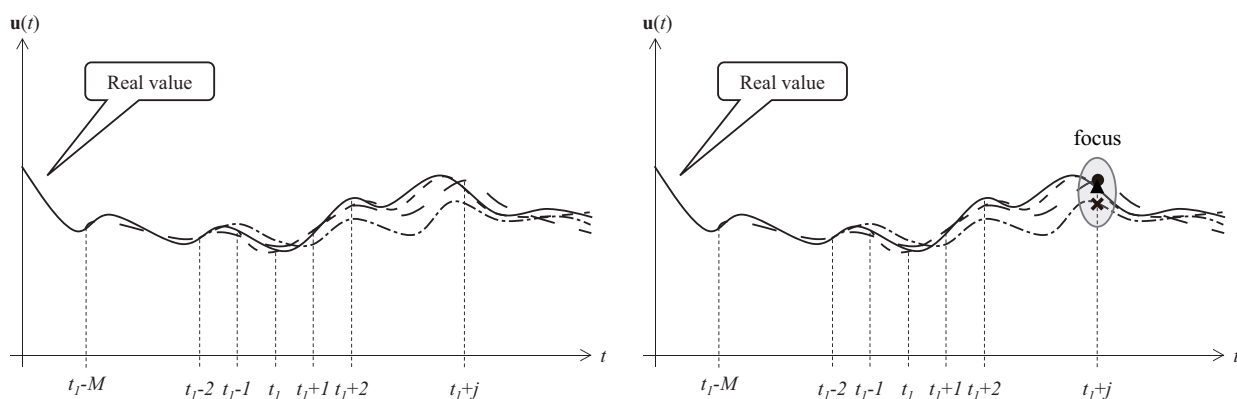


図 3.9 状態行動対予測によって得られる予測値列

3.5.2.2 動的に変更できる重み係数を設計する (1)

ここでは, 予測値の予測開始時間と任意の対象時間に於ける『ばらつき』に着目し, その時点での予測値の信頼性を考える. 状態行動対予測を適用した場合, 各サンプリング時刻での予測が可能となる. そこで, 或る時刻 $t_1 + j$ での予測値について着目し考えてみる (図 3.10). まず入

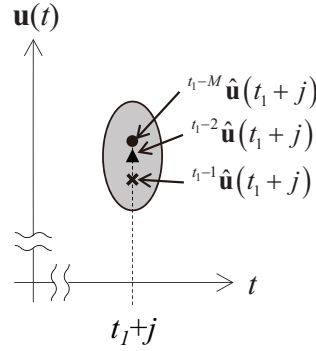


図 3.10 状態行動対予測によって得た予測値のばらつき

力 $\mathbf{u}(t+j)$ に対するそれぞれの予測値を計算する. 本研究に於いては, この結果得られたそれぞれの時刻に於ける \mathbf{u} の予測値に対し, 時刻 t_1+j に於ける入力標準偏差 σ_j を求める事とする. この σ_j を用いる事で, 予測結果に応じて柔軟に変化する重み係数を定義する. またこの時, $t+j$ が時刻 t から離れる程, その時刻に於ける予測値の重みを小さくする事として, 補正量の増加を避ける. この考えに基づき, 次式を得る.

$$\alpha_j = C_\sigma \cdot \sigma_j \quad (3.5.4)$$

$$\sigma_j = \sigma \left[t_1^{-M} \hat{\mathbf{u}}(t_1 + j), t_1^{-M+1} \hat{\mathbf{u}}(t_1 + j), \dots, t_1^{-i} \hat{\mathbf{u}}(t_1 + j), \dots, t_1^{-1} \hat{\mathbf{u}}(t_1 + j) \right] \quad (3.5.5)$$

ここで, C_σ は小さな正の実数であり, $\sigma(\cdot)$ は \cdot の集合に対する標準偏差を求めるシンボルである.

この α_j を用いる事で, “ N -ahead State-Action Pair Predictor” ブロックによって, 現時刻 t から任意の時刻 $t+N$ までの遠い未来にかけて採られるであろう行動の予測値列が得られる. この予測値列を使う事により, 現時刻で決定した行動に未来で採る行動を加えて補正する事が可能となる. 即ち, 或る時刻 t にて, 未来の時刻 $t+1$ から $t+N$ にかけて予測した行動 $\hat{\mathbf{u}}_{t+i}, 1 \leq i \leq N$ に対して

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t+i) \quad (3.5.6)$$

と定義する. 上式の α_i は各時刻に於ける予測値に対する重みである. この $\mathbf{u}_s^{(t+1, t+N)}(t)$ が, 行動補正に用いる未来の行動を生成するブロック “Action Decision Maker” の機能となる. この時, “Action Decision Maker” への入力となる “ N -ahead State-Action Pair Predictor” による行動の予測値には, 外乱 $\mathbf{d}(t)$ が明示的に含まれている. この点に着目すると, 事前に外乱に対応出来る行動を作り出す事で, 将来に於いて最適な行動を獲得出来る. つまり図 3.8 内の “Action

Decision Maker” に該当するブロックによって決定した行動 $\mathbf{u}_s^{(t+1, t+N)}(t)$, および状態フィードバックが計算した行動 $\mathbf{u}_p(t)$ を使う事で

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.7)$$

となる.

3.5.2.3 動的に変更できる重み係数を設計する (2)

続いて, 用いる予測値に於いても, 予測を始めた時刻に注目し, 最新の時刻程重要度を高める [76] 事を試みる. 図 3.9 で示した楕円枠の部分に着目する (図 3.10 は拡大部分である). この時刻 $t_1 + j$ に於ける予測値について考えてみる.

それぞれの時刻 $(t_1 - k), (0 \ll M \leq k \leq N)$ に於ける入力 $\mathbf{u}(t_1 + j)$ の予測値に注目する. まず入力 $\mathbf{u}(t + j)$ に対するそれぞれの予測値を計算する. この時得られたそれぞれの時刻に於ける \mathbf{u} の予測値に対し, 時刻 $(t_1 + j)$ に於ける入力の標準偏差 σ_j を求める. ここで, この σ_j を用い重み α_j を求める. 標準偏差の特性から, 予測結果の集まりがより拡散している場合は重みを小さくし逆に予測結果がある範囲に収束している場合は大きい重みを定義する様にする. またこの時, $t + j$ が時刻 t から離れる程, その時刻に於ける予測値の重みを小さくする事として, 補正量の増加を避ける. これに従い, 次のような式を与える.

$$\alpha_j = \frac{C_{\sigma 1}}{C_{\sigma 2} + \sigma_j} \quad (3.5.8)$$

$$\sigma_j = \sigma \left(\frac{w_M}{w_\Sigma} t_1 - M \hat{\mathbf{u}}(t_1 + j), \frac{w_{M+1}}{w_\Sigma} t_1 - M + 1 \hat{\mathbf{u}}(t_1 + j), \dots, \frac{w_1}{w_\Sigma} t_1 + 1 \hat{\mathbf{u}}(t_1 + j) \right) \quad (3.5.9)$$

ここで, $C_{\sigma 1}, C_{\sigma 2}$ は 1 より小さな正の実数であり, $\sigma(\cdot)$ は, \cdot の集合に対する標準偏差を求めるシンボルである. また式 (3.5.9) にて, w_M, w_{M+1}, \dots, w_1 は加重移動平均に於ける重み, w_Σ は重みの総和である.

この α_j を用いる事で, “N-ahead State-Action Pair Predictor” ブロックによって, 現時刻 t から任意の $t + N$ 時刻先までの遠い未来にかけて採られるであろう行動の予測値列が得られる. この予測値列を使う事により, 現時刻で決定した行動に未来で採る行動を加えて補正する事が可能となる. ただし, 予測結果から得られた行動を実際に用いる場合, 予測の対象となる時刻が現時刻に比べて相対的に遠くなると, その長さに比例して予測誤差が無視出来ない程度に蓄積される. そこで本研究では, 補正に用いる予測結果に対し重み付けを行い, 補正に用いる値の重要度を設定すると共に予測誤差の影響を低減する事を考える. つまり, 現時刻より遠い未来の時

間で採る行動程, その行動に対する重要度を低下させる. 即ち, 或る時刻 t にて未来の時刻 $t+1$ から $t+N$ にかけて予測した行動 $\hat{\mathbf{u}}_{t+i}, 1 \leq i \leq N$ に対して

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t+i) \quad (3.5.10)$$

とする. 上式の α_i は, 各時刻に於ける予測値に対する重みであり, 式 (3.5.8) を適用する. この $\mathbf{u}_s^{(t+1, t+N)}(t)$ が, 行動補正に用いる未来の行動を生成するブロック “Action Decision Maker” の機能となる. この時, “Action Decision Maker” への入力となる “N-ahead State-Action Pair Predictor” による行動の予測値には, 外乱 $\mathbf{d}(t)$ が明示的に含まれている. この点に着目すると, 外乱を前もって対応出来る様な行動を作り出す事で, 未来に於いて最適な行動を獲得出来るものと考えられる. つまり図 3.8 内の “Action Decision Maker” に該当するブロックによって決定した行動 $\mathbf{u}_s^{(t+1, t+N)}(t)$, および状態フィードバックが計算した行動 $\mathbf{u}_p(t)$ を使う事で

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.11)$$

となる.

3.5.2.4 行動補正の安定性について

ここで, 行動補正による安定性を考えてみる. $\mathbf{u}_p(t)$ は常に系が安定化する様な制御入力を出力し続ける. $\mathbf{u}_s^{(t+1, t+N)}(t)$ は遠い未来にかけてその影響を減少するような行動を出力する. また各時刻に於ける予測結果が或る一点に収束した場合は, 或る一定値に収束する様に設定している. そしてこの $\mathbf{u}_s^{(t+1, t+N)}(t)$ は外乱 $\mathbf{d}(t)$ の傾向も間接的に学習する事で, $\mathbf{d}(t)$ をキャンセルする様な制御入力を与える. 従って, 経時的には安定的な行動補正を与える事となる.

3.6 まとめ

本章では, ロボットに於ける状態と行動を定義した. また, 状態と行動を対として扱う状態行動対を説明した. また, Online SVR を拡張する事で, 本研究が提案する状態行動対予測に必要な, 状態及び行動の予測式を定式化した. また, この状態行動対予測を基に, 未来の状態と行動を用いて現在の行動を補正する考え方を述べた.

第 4 章 提案システムと処理の流れ

第 3 章に於いて、状態行動対予測と行動補正の考え方を示した。本章では、先に挙げた考え方をシステム化し、実際に制御対象の安定化を為すものとして定める事を考える。

先ずシステム全体を俯瞰し、その後各々のシステムについて概説する。また、本研究に於いて適用対象としている自走式二輪型倒立振子“**NXTway-GS**”について解説し、提案システムに組み込む為にモデリングする。本章以降に於いて、『行動』と『制御』は同義である。

4.1 システム構成

本研究で提案するシステムとしては、図 4.1 に示す通りである。動的環境に於けるロボット制御では、これまでの行動と状態を基にして未来の状態を予測し、その結果に即した行動を直ちに用いる事により最適な動作が実現出来る。本研究では特に、状態行動対予測を利用する事で予測可能な外乱が与えられた場合で、倒立振子のピッチ角の変化を小さく留める様な最適行動を獲得する事を考える。そこで、外部からの指令入力によって与えられる外乱や、モータやアクチュエータに対し直接的に与えられる外乱を補正する上で必要となる行動を生成する為に、図 4.1 で示されるシステムを考える。以下に、本システムを構成するサブシステムを紹介する。

4.1.1 N -ahead State-Action Pair Predictor ブロック

図 4.1 に示す様に、 $t-l\hat{\mathbf{u}}(t+j)$ は時刻 $t-l$ に於ける制御入力 $\mathbf{u}(t+j)$ の予測値列となっている。ここで提案手法では、状態行動対予測で得た予測結果を用いて現在の行動を補正する事を標榜している。この時使う状態行動対予測を行うサブシステムとして、“ N -ahead State-Action Pair

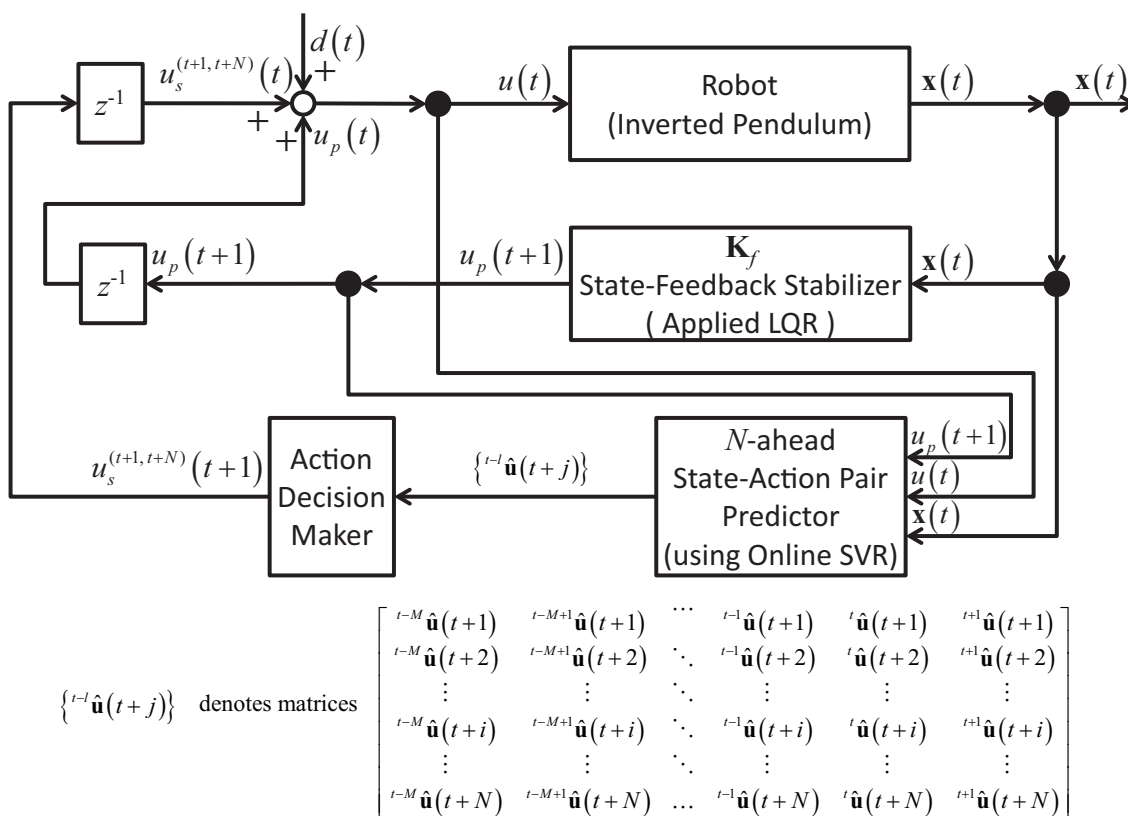


図 4.1 状態行動対予測を用いた 行動補正システム

Predictor” を構成している.

このサブシステムの内部構造は、次図 4.2 の通りである.

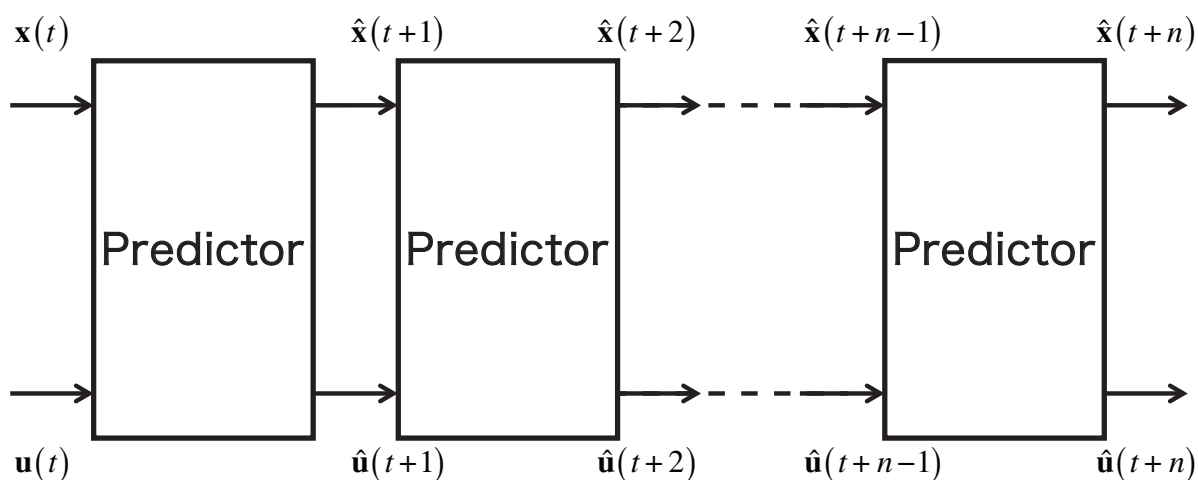


図 4.2 状態行動対予測による予測システム

上図に示す様に、概念としては状態行動対予測を連続的に行い、時刻 t から時刻 $t+n$ につ

ての状態と行動の未来の予測値を計算し続ける. 各々の Predictor の内部構成は, 図 3.7 に示す通りとなっている. また, このシステムの動作は次式に従う.

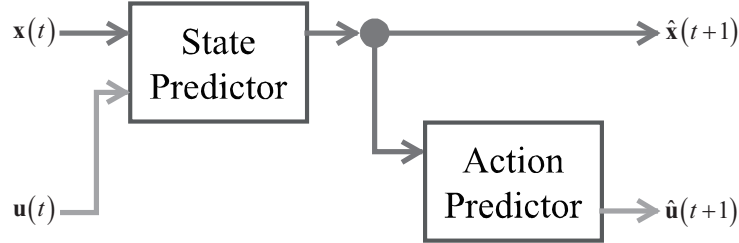


図 4.3 Online SVR と LQR を用いた予測システムの概要 (図 3.7 の再掲)

$$\hat{x}_{t+1,i} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}({}^x\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_l)^{-1} {}^x\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_t$

$$\hat{u}_{t+1} = \mathbf{k}_f \hat{\mathbf{x}}_{t+1} \quad (4.1.1)$$

従って, “ N -ahead State-action Pair Predictor” ブロックによって, 任意の現時刻 t から任意の $t + N$ 時刻先までの遠い未来にかけて採られるであろう, 行動と状態の予測値列が得られる.

4.1.2 State-Feedback Stabilizer ブロック

図 4.1 に示す様に, このブロックでは LQR に基き状態フィードバックゲイン \mathbf{K}_f を定め, \mathbf{K}_f によって現時点における最適行動を算出する. なお, この動作は状態行動対予測と並行して実施される. 具体的な \mathbf{K}_f の算出法については, 二輪型倒立振子のモデリングの項に於いて後述する.

4.1.3 Action Decision Maker ブロック

図 4.1 に示す様に, “ N -ahead State-Action Pair Predictor” で得た予測値列 ${}^{t-l}\hat{\mathbf{u}}(t + j)$ から, 現在採るべき最適行動を決定する機構を持ったサブシステムである. このブロックの動作は, 次式に従う.

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t + i) \quad (4.1.2)$$

α_i の定義については, 前章に於ける重み係数の設計方法に従う事となる.

4.1.4 Robot (Inverted Pendulum) ブロック

図 4.1 に示したシステムは、自走式二輪型倒立振子の倒立姿勢維持制御を行うものであり、状態フィードバックゲイン \mathbf{K}_f によって最適制御を行うと共に、状態行動対予測によって未来に採るべき行動を組合せて、倒立振子への制御入力を補正する。

二輪型倒立振子については後述する。

4.2 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 ROBO-LAB 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。

4.2.1 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 4.4)。本研究で適用対象となるモデルはこれである。

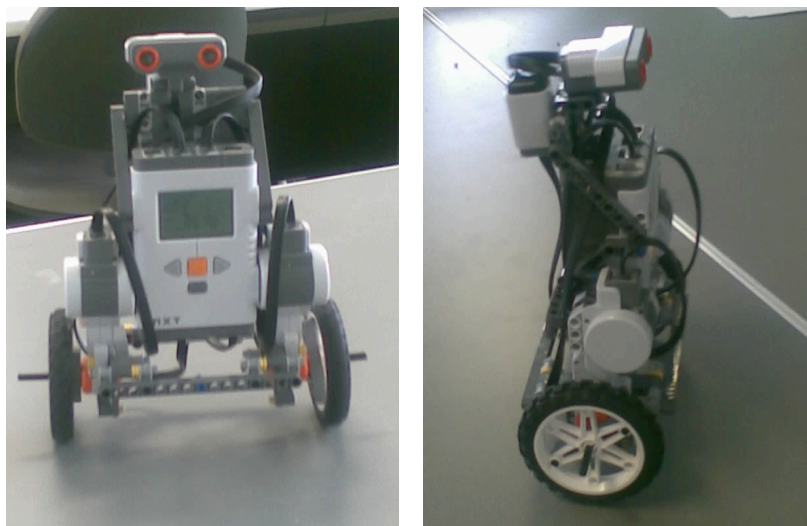


図 4.4 自走式二輪型倒立振子 NXTway-GS の概観

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。

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

■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 [77].

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

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 符号付き整数である.

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 4.5 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.

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 4.6(a)) と平面図 (図 4.6(b)) に分け, 座標や角度を考える.

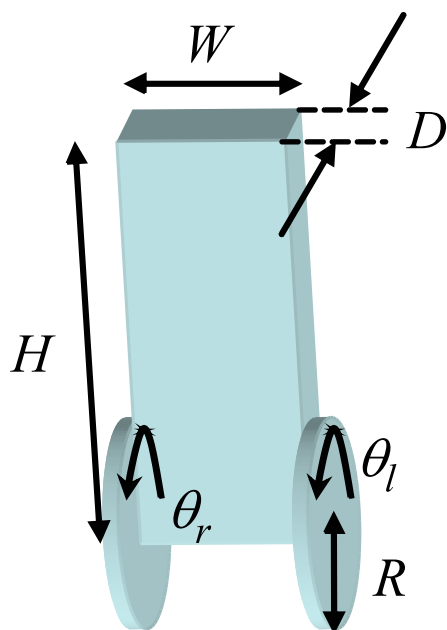
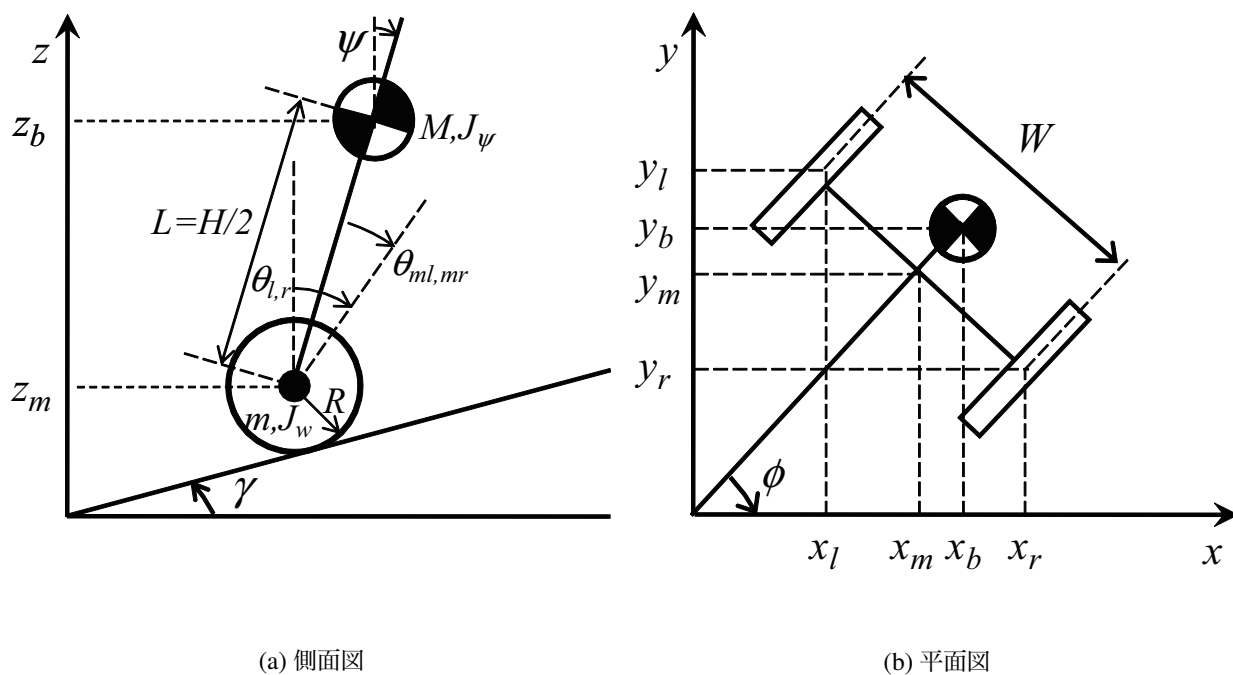


図 4.5 自走式二輪型倒立振子のモデル



(a) 側面図

(b) 平面図

図 4.6 自走式二輪型倒立振子の側面・平面図

上図にて、 ψ 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有る。表 4.3 は、

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

NXTway-GS の各種物理パラメータである.

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 ²]	重力加速度
m	0.03	[kg]	車輪 1 本あたりの重量 [78, 79]
R	0.04	[m]	車輪半径
J_w	$mR^2/2$	[kgm ²]	車輪の慣性モーメント
M	0.635	[kg]	車体重量 [78, 79]
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_ψ	$ML^2/3$	[kgm ²]	車体慣性モーメント (ピッチ)
J_ϕ	$M(W^2 + D^2)/12$	[kgm ²]	車体慣性モーメント (ヨー)
J_m	1×10^{-5}	[kgm ²]	DC モータ慣性モーメント [77]
R_m	6.69	[Ω]	DC モータ抵抗 [80]
K_b	0.468	[V·sec/rad.]	DC モータ逆起電力定数 [80]
K_t	0.317	[N·m/A]	DC モータトルク定数 [80]
n	1	[1]	ギア比 [77]
f_m	0.0022	[1]	車体と DC モータの間の摩擦係数 [77]
f_w	0	[1]	車輪と路面間の摩擦係数 [77]

4.3.2 NXTway-GS のモデリング

では、図 4.6 の座標系を用い、自走式二輪型倒立振子の運動方程式を求める。時刻 $t = 0$ で自走式二輪型倒立振子の向きが 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。

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

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、 θ 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)^[81] 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

- θ : 左右車輪の平均回転角度
- ψ : 車体の傾斜角度 (ピッチ角度)
- ϕ : 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

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

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

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

となるので、

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

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

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

となるので, ラグランジュ方程式 (4.3.15) は

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

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.

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

$$F_l = nK_t i_l + f_m (\dot{\psi} - \dot{\theta}_l) - f_w \dot{\theta}_l \tag{4.3.21}$$

$$F_r = nK_t i_r + f_m (\dot{\psi} - \dot{\theta}_r) - f_w \dot{\theta}_r \tag{4.3.22}$$

$$F_\psi = -nK_t i_l - nK_t i_r - f_m (\dot{\psi} - \dot{\theta}_l) - f_m (\dot{\psi} - \dot{\theta}_r) \tag{4.3.23}$$

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.

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

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

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

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, α, β 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, linearization), 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, となる。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。

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

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

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

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

$$\cos \psi \rightarrow 1$$

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

式 (4.3.33) および式 (4.3.34) は θ と ψ の連立方程式, 式 (4.3.35) は ϕ 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 γ 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

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

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.

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

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

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

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

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

以上の2式から、次を得る.

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

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

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

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

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

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

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

ただし、本研究に於いては傾斜路は想定しないので、 $\mathbf{S} = \mathbf{0}$ とする. また、一次元の前後移動のみを考慮するので、 $\phi = 0, \theta_{ml} = \theta_{mr}$ とし、制御入力及び外乱はそれぞれ一次元の入力、即ち $\mathbf{u}(t) = u(t), \mathbf{d}(t) = d(t)$ とする.

4.3.4 コントローラ的设计

4.3.4.1 入出力

図 4.7 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 θ と、車体の傾

斜角速度 $\dot{\psi}$ とする.

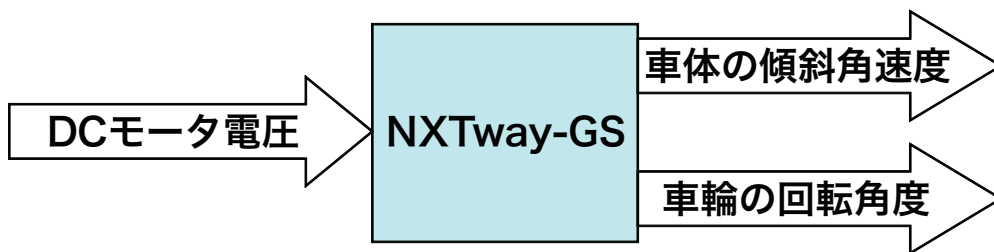


図 4.7 NXTway-GS の入出力

本研究では, $\dot{\psi}$ を数値積分する事により ψ 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, 車体の傾斜角度を検出し, 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 θ を選択する. 図 4.8 は, NXTway-GS 用サーボ制御器のブロック線図である.

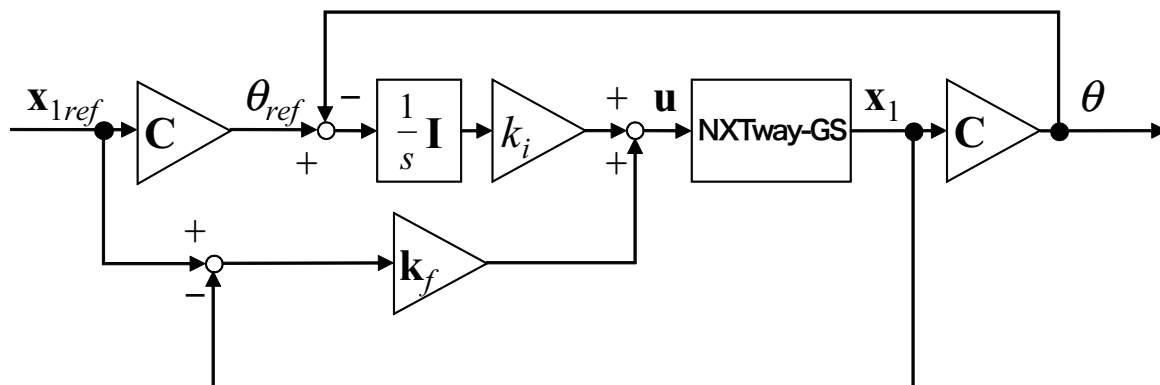


図 4.8 NXTway-GS 用サーボ制御器のブロック線図

系が可制御では無くなる為, θ 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 of 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 [78, 79, 77].

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

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

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 [78, 79].

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

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 4.9).

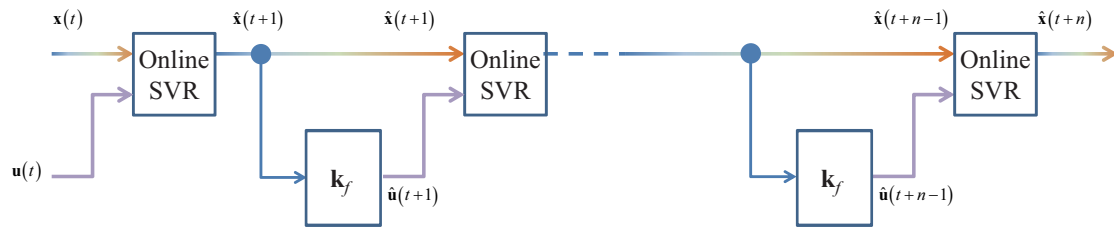


図 4.9 本研究で提案する長期内部状態予測の構成図 (図 3.4 の再掲)

4.4 まとめ

本章では, 状態行動対予測と行動補正を, 従来使われてきた LQR による状態フィードバック機構と組み合わせシステム化し, 制御対象として二輪型倒立振子へ適用する枠組みを構築した. また, システム全体を俯瞰し, その後各々のシステムについて機能を概説した. そして, 本研究に於いて適用対象としている自走式二輪型倒立振子 “NXTway-GS” について解説し, 提案システムに組込む為にモデリングした. 平面図と側面図より運動方程式を構築し, 動作点周りで線型化する事によって, システム方程式を導出した.

第 5 章 実験 I: 倒立振子に対する状態 行動対予測の適用

本章では, 将来の予測に焦点を当て, 本研究で提案した状態行動対予測の有効性を検証する. 実施の為にシミュレータを用いた検証実験を取り扱う.

適用例としては, 自走式二輪型倒立振子 “NXTway-GS” をモデル化すると共に, 実際の応答結果と提案手法の予測結果を比較し, その結果について評価する. また, ここでは Online SVR にみよって予測器を構築した場合と, Online SVR を状態予測に, LQR を行動予測に適用し予測器を構築した場合について各々検討する.

5.1 実験の目的

第 2 章および第 3 章において, ロボットが直ちに将来とるべき行動を決定出来ることのメリットと必要性を述べると共に, 我々は状態行動対予測を提案した. 本実験の目的は, Online SVR, および LQR(最適レギュレータ法) を適用した状態行動対予測を用い, 現時刻において将来の状態と行動のそれぞれを予測出来ることを確認する点にある.

提案手法を評価するための適用例として, NXTway-GS を使用し, 実際の応答との予測結果を比較する. 特に本章では, Online SVR のみで状態行動対予測を行う方法, そして Online SVR と LQR を組み合わせて状態行動対予測を行う方法のそれぞれを検証し, 各々の結果を議論してゆく.

5.2 事前実験

5.2.1 システム構築

ここでは、内部状態と行動を予測する為に用いる学習データの取得の為に事前に必要となるシミュレーション実験について説明する。本研究では NXTway-GS を、前章にて二輪型倒立振子としてモデリングした。NXTway-GS を側面図 (図 4.6(a)) と平面図 (図 4.6(b)) に分け、これに従い運動方程式を考える [78][79]。同図で現れる主要な変数は表 4.3 の通りである。

Lagrange 方程式 [81] を基に、鉛直近傍で線型化した運動方程式を立てると、次式を得る。

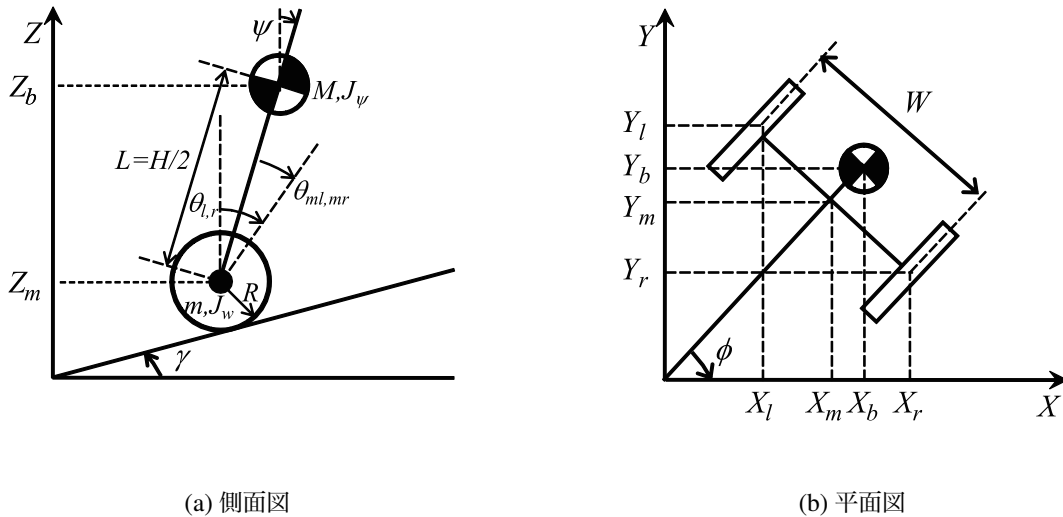


図 5.1 NXTway-GS の側面・平面図 (図 4.6 の再掲)

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

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

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

この運動方程式を $\mathbf{x}_1 = [\theta \ \psi \ \dot{\theta} \ \dot{\psi}]^T$, $\mathbf{u} = [v_l \ v_r]^T$, $y = \theta$ とすることで、

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

を得る。この状態方程式と、最適レギュレータ法を適用する事で算出したフィードバックゲインを配置した状態フィードバックにより、制御器を構成した [78][79]。この様にして得られる事

前実験の概要が図 5.2 である.

5.2.2 学習データの取得

事前実験に於いては, 予測可能な外乱をモータへ与える (図 5.3). 実験条件は表 5.1 に従う. この事前実験によって学習データが得られる.

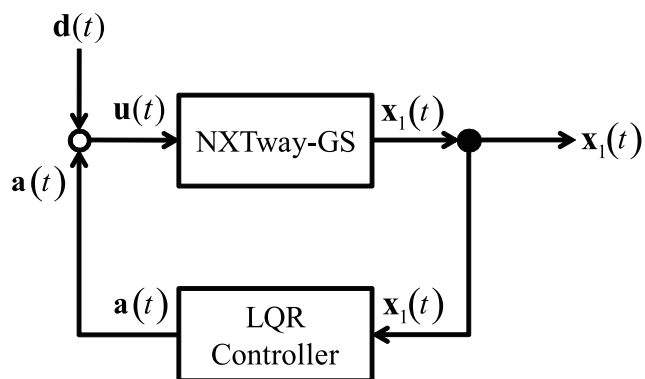


図 5.2 事前実験の為のシステム構成 (1)

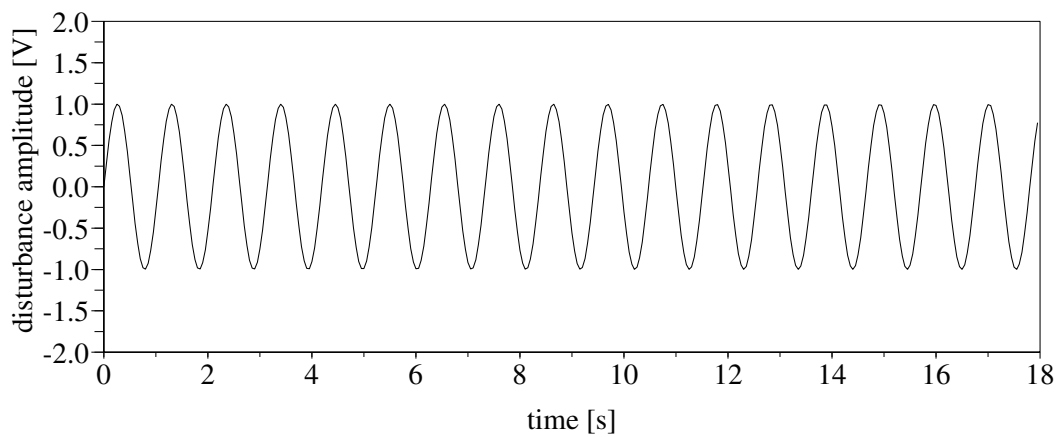


図 5.3 モデルへ与える外乱入力 $\mathbf{d}(t)$ (1)

表 5.1 事前実験に係わる実験パラメータ (1)

定数	数値	単位	意味
ψ_0	0.0262	[rad.]	初期ピッチ角度
γ_0	0.0	[rad.]	進行方向傾斜路傾斜角
t_s	0.05	[s]	サンプリングレート
$t_{d,start}$	0.0	[s]	予測可能な外乱を与え始める時間
$t_{d,finish}$	18.0	[s]	予測可能な外乱を与え終わる時間
A_{d1}	1.0	[V]	予測可能な外乱の振幅
f_{d1}	6.0	[Hz]	予測可能な外乱の周波数
N	60	—	初期データセット長

5.3 検証実験

5.3.1 実験設定

5.3.1.1 Online SVR のみで状態行動対予測を実施した場合

ここでは、学習器として、RBF カーネルを組込んだ Online SVR を用いている。学習データについては、先述の事前実験で得た制御応答と行動の数値データ (図 5.4 から図 5.8 に於ける黒色実線で示される曲線) を採用する。学習器の学習パラメータを表 5.2 に示す。

5.3.1.2 Online SVR と LQR を組み合わせて状態行動対予測を実施した場合

続いてここでは、学習器として、RBF カーネルを組込んだ Online SVR を **State Predictor** に、LQR を **Action Predictor** に用いている。学習データについては、先述の事前実験で得た制御応答と行動の数値データ (図 5.14 から図 5.18 に於ける黒色実線で示される曲線) を採用する。学習器の学習パラメータを表 5.3 に示す。尚、**Action Predictor** には LQR を用いているので、状

表 5.2 検証実験時の Online SVR に於ける学習パラメータ

定数	数値	意味
C_1	300	x_1 に対する正則化パラメータ
ϵ_1	0.03	x_1 に対する ϵ 不感帯パラメータ
β_1	30	x_1 に対する RBF カーネルパラメータ
C_2	300	x_2 に対する正則化パラメータ
ϵ_2	0.03	x_2 に対する ϵ 不感帯パラメータ
β_2	30	x_2 に対する RBF カーネルパラメータ
C_3	300	x_3 に対する正則化パラメータ
ϵ_3	0.03	x_3 に対する ϵ 不感帯パラメータ
β_3	30	x_3 に対する RBF カーネルパラメータ
C_4	300	x_4 に対する正則化パラメータ
ϵ_4	0.03	x_4 に対する ϵ 不感帯パラメータ
β_4	30	x_4 に対する RBF カーネルパラメータ
C_a	300	a に対する正則化パラメータ
ϵ_a	0.03	a に対する ϵ 不感帯パラメータ
β_a	30	a に対する RBF カーネルパラメータ

態フィードバックゲイン \mathbf{k}_f を得る為に、重み行列 \mathbf{Q}, \mathbf{R} には、次の数値を適用する。

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^2 \end{bmatrix} \quad (5.3.1)$$

$$\mathbf{R} = \begin{bmatrix} 1 \times 10^3 & 0 \\ 0 & 1 \times 10^3 \end{bmatrix} \quad (5.3.2)$$

表 5.3 検証実験時の Online SVR 及び LQR に於けるパラメータ (1)

定数	数値	意味
C_1	300	x_1 に対する正則化パラメータ
ϵ_1	0.03	x_1 に対する ϵ 不感帯パラメータ
β_1	30	x_1 に対する RBF カーネルパラメータ
C_2	300	x_2 に対する正則化パラメータ
ϵ_2	0.03	x_2 に対する ϵ 不感帯パラメータ
β_2	30	x_2 に対する RBF カーネルパラメータ
C_3	300	x_3 に対する正則化パラメータ
ϵ_3	0.03	x_3 に対する ϵ 不感帯パラメータ
β_3	30	x_3 に対する RBF カーネルパラメータ
C_4	300	x_4 に対する正則化パラメータ
ϵ_4	0.03	x_4 に対する ϵ 不感帯パラメータ
β_4	30	x_4 に対する RBF カーネルパラメータ
\mathbf{k}_f	$\begin{bmatrix} -0.870 \\ -32.2 \\ -1.16 \\ -2.81 \end{bmatrix}^T$	u に対する状態フィードバックゲイン

5.3.2 実験結果とその検討

5.3.2.1 Online SVR のみで状態行動対予測を実施した場合

図 5.4 から 5.7 は状態 \mathbf{x}_1 の予測結果を, 図 5.8 は行動 u の予測結果を示している.

\mathbf{x}_1 及び u の予測結果に於いて, ここでは図 5.4 に示した時刻 T , 即ち状態行動対予測を開始して以降の変動について議論してゆく事とする.

これら予測結果の図に於いて, 黒色実線は学習データの軌跡を, 緑色実線は予測結果の軌跡を表現している. \mathbf{x}_1 及び u の予測結果に着目すると, この予測結果はほぼリアルタイムに学習データに追従する予測値を算出している事が判る. 特に図 5.4 及び図 5.6 では, 学習データに対しよく予測結果が合致している事が確認出来る. これは, 学習データの傾向が滑らかであった事が要因として挙げられる. 一方で, 混入された外乱を含む内部状態の応答に注目すると, 時間の経過に従い安定状態である鉛直近傍へ近接する様な挙動を予測している事が見て取れる.

続いて, それぞれの状態及び行動に対する予測誤差を計算したグラフを次に示す.

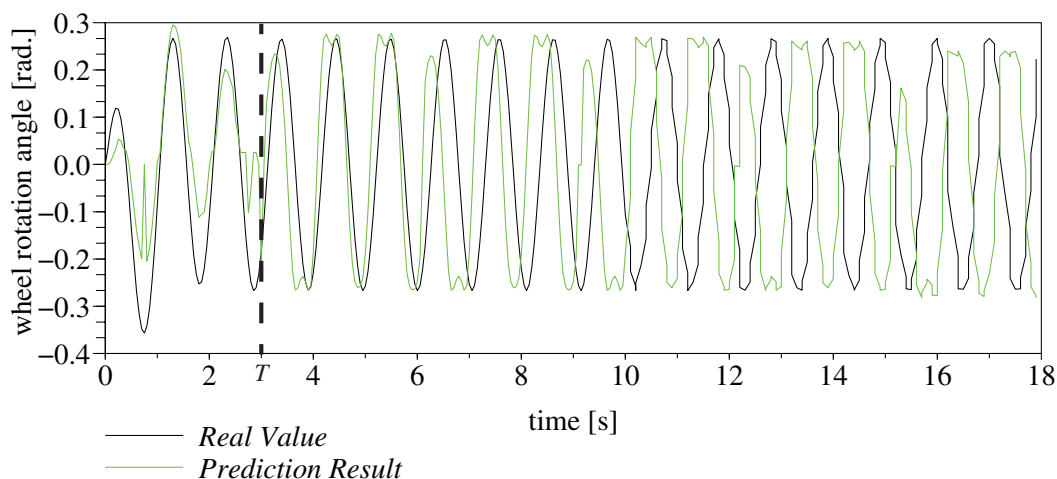


図 5.4 状態 $x_{1,1}$ (車輪回転角度 θ) の予測結果 (Online SVR のみのとき)

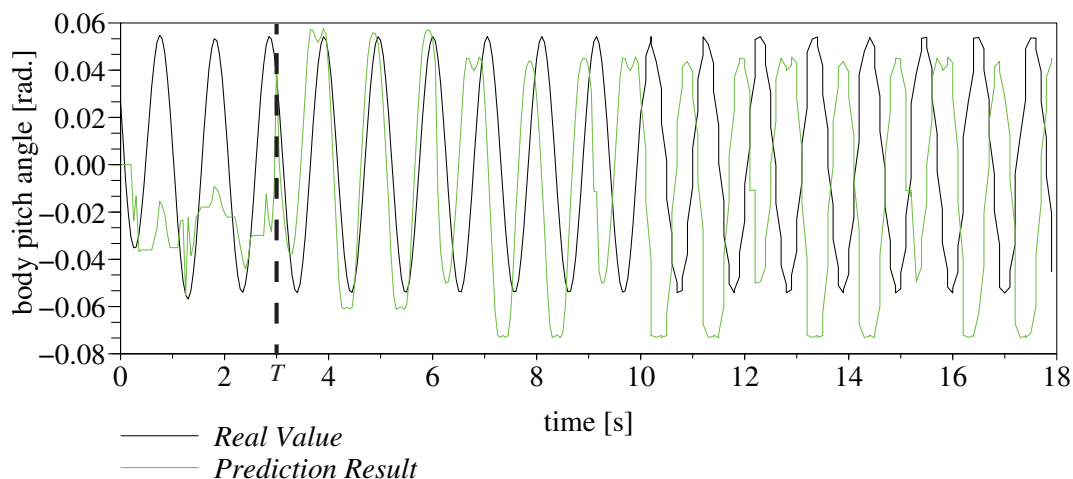


図 5.5 状態 $x_{1,2}$ (ピッチ角度 ψ) の予測結果 (Online SVR のみのとき)

ここで, 上記の予測誤差は以下に従い算出出来る.

$$e_i(t) = |\hat{x}_{1,i}(t) - x_{1,i}(t)|, \quad i \in \dim \mathbf{x}_1(t) \quad (5.3.3)$$

$$e_u(t) = |\hat{u}(t) - u(t)| \quad (5.3.4)$$

この予測誤差に注目する. 図 5.9 から図 5.12 に示した各々の状態と図 5.13 に示した行動に対する予測誤差は, 予測を開始した時刻 T より遠い時間になるほど, その時間の遠さに比例して増加している事が見て取れる. この結果について考えてみると, 未来の予測値は, Online SVR によって算出された予測値を繰返し用いる事によって導いている. その為, この様な現象を得た理由としては, 次に挙げる 2 点が考えられる.

1 点目として, 初期のパラメータ設定が挙げられる. 予測結果は, ϵ 不感帯パラメータと正則

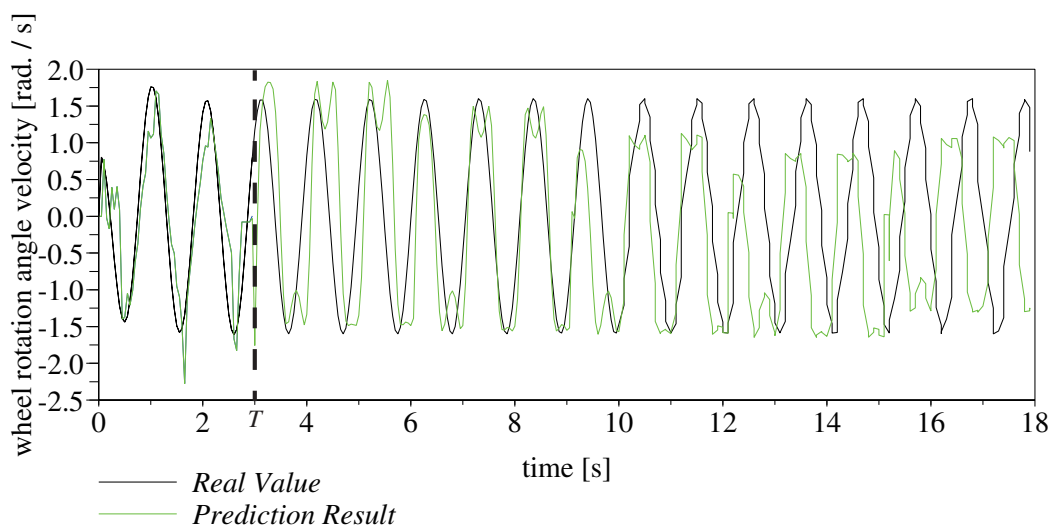


図 5.6 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測結果 (Online SVR のみのとき)

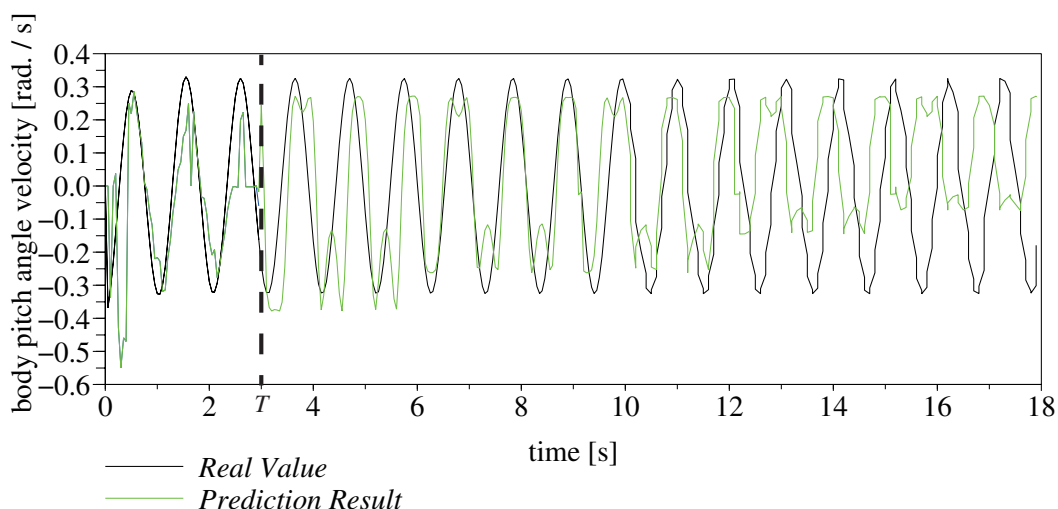


図 5.7 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測結果 (Online SVR のみのとき)

化パラメータによって決定される。ここでは、汎化性を考慮して小さな正則化パラメータと広めな不感帯パラメータを設定している。この様な場合、学習器は、外乱として与えたデータを許容誤差と認められるノイズであると見做す可能性がある。2 点目に、検証実験に於いて入力した外乱信号そのものの特性を考慮する必要がある。この場合、学習器は、小さな振幅の外乱信号を、設定した不感帯の範囲内に含まれるものと見做して無視する可能性がある。

以上を踏まえると、今回学習データに於いて示した極値の様な非線形部分を除き、予測可能な外乱に対してその後の挙動を予測すると共に、外乱の影響が時間経過に従い減少してゆく様に予測しているものと結論付ける事が出来る。

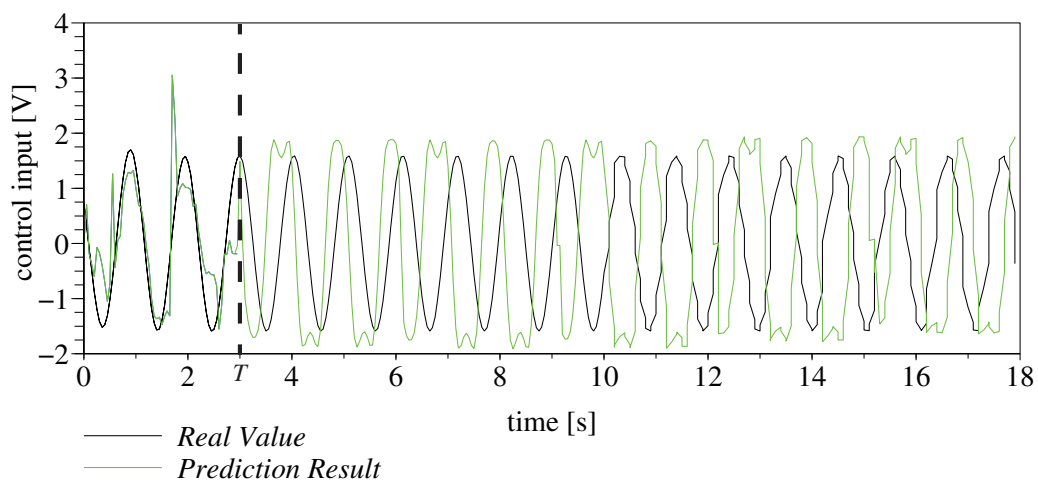


図 5.8 行動 u (コントローラからの最適入力) の予測結果 (Online SVR のみのとき)

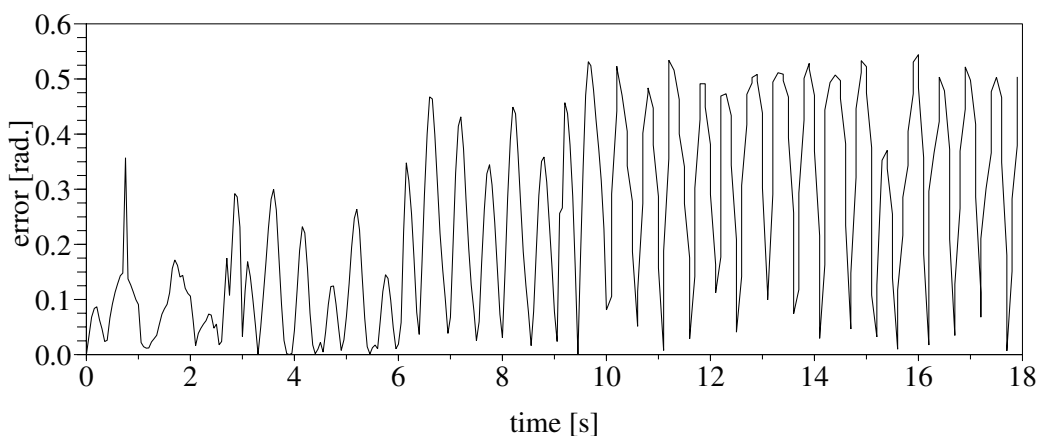


図 5.9 状態 $x_{1,1}$ (車輪回転角度 θ) の予測誤差 (Online SVR のみのとき)

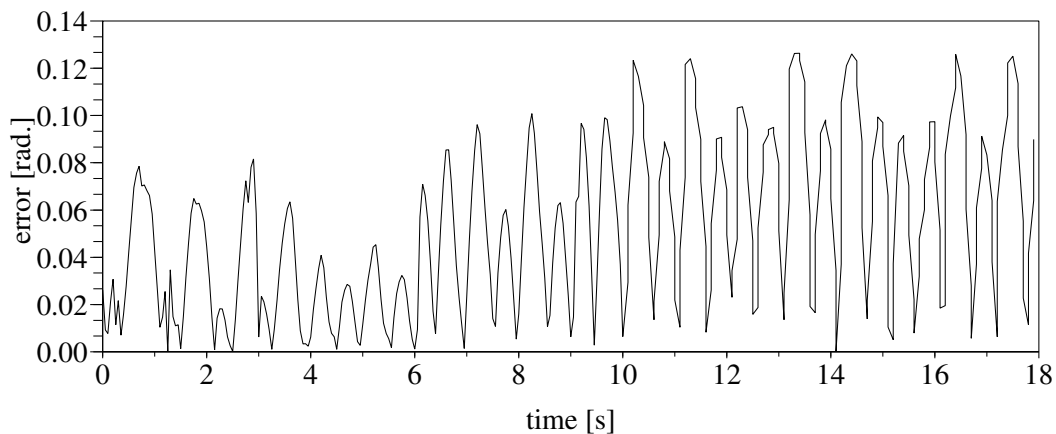


図 5.10 状態 $x_{1,2}$ (ピッチ角度 ψ) の予測誤差 (Online SVR のみのとき)

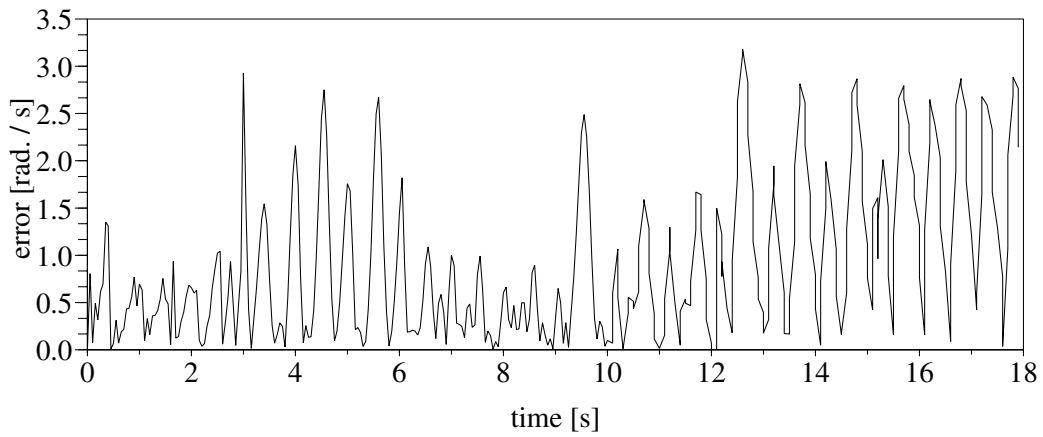


図 5.11 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測誤差 (Online SVR のみのとき)

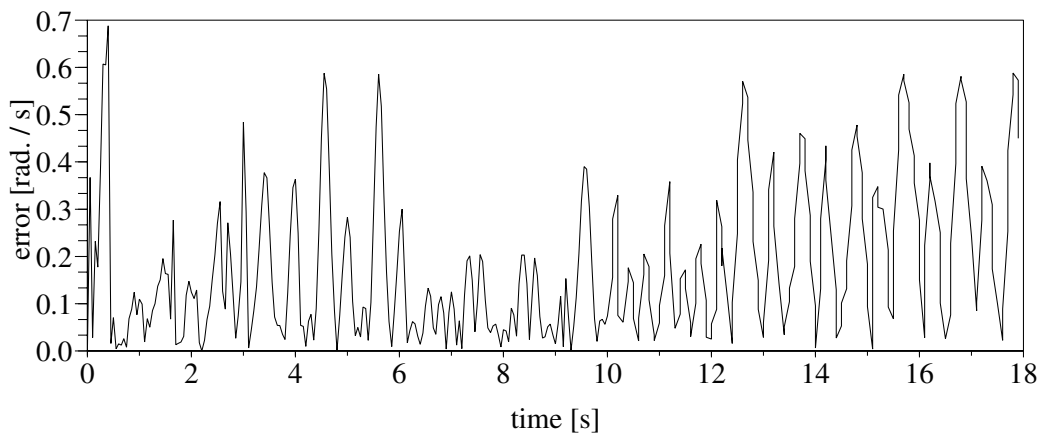


図 5.12 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測誤差 (Online SVR のみのとき)

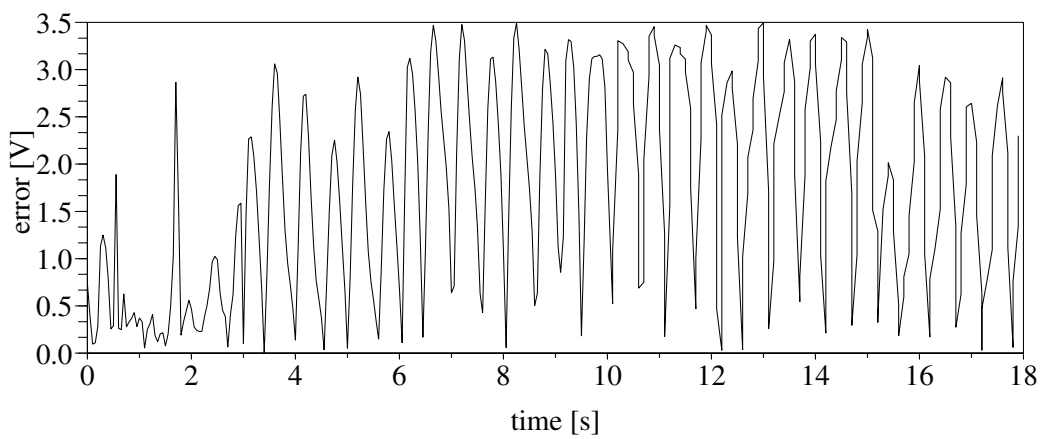


図 5.13 行動 u (コントローラからの最適入力) の予測誤差 (Online SVR のみのとき)

その一方でこの予測結果を行動決定に適用する場面を想定した場合, 予測誤差に関しては, 用いる予測結果に対して重みを付ける事によって対処出来る. 但し, 予測可能な外乱であっても, 予測結果と実際の外乱それぞれの傾向が異なる場合は, 重みによる対処のみでは難しいものと考えられる. 従って, 予測結果を行動決定に用いることが可能な範囲は有限であり, 尚且つ事前に学習したデータ数に依存した範囲になるものと考ええる. 従って, 得られた結果より, 状態行動対予測として得られる予測結果の有効な範囲考慮すれば, 適切なものとなると結論付ける.

5.3.2.2 Online SVR と LQR を組み合わせて状態行動対予測を実施した場合

図 5.14 から 5.17 は内部状態 \mathbf{x}_1 の予測結果を, 図 5.18 は行動 u の予測結果を示している.

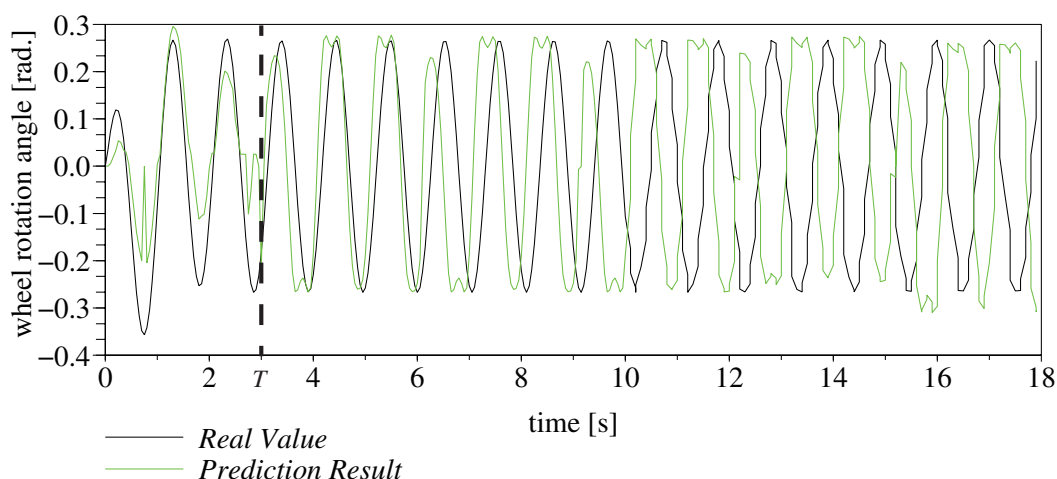


図 5.14 状態 $x_{1,1}$ (車輪回転角度 θ) の予測結果 (Online SVR と LQR を組み合わせたとき)

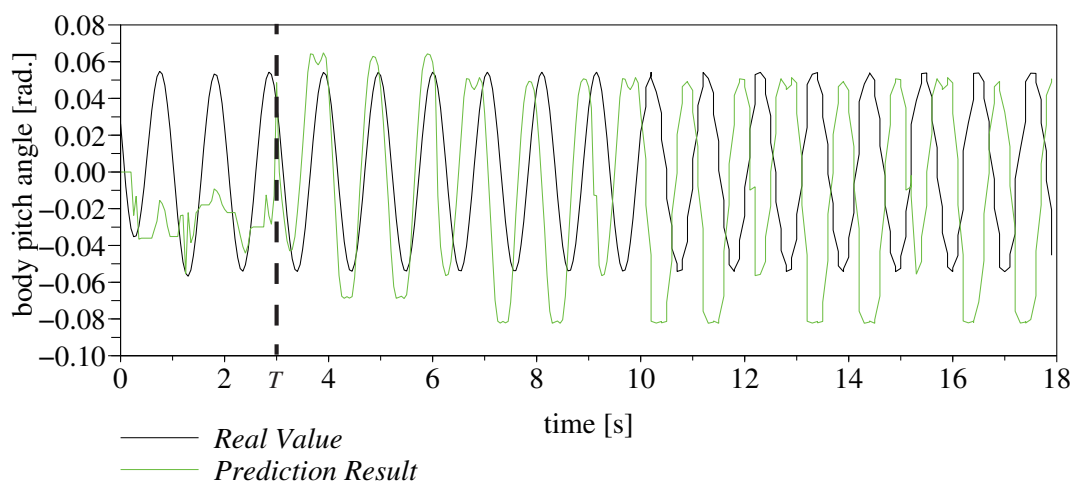


図 5.15 状態 $x_{1,2}$ (ピッチ角度 ψ) の予測結果 (Online SVR と LQR を組み合わせたとき)

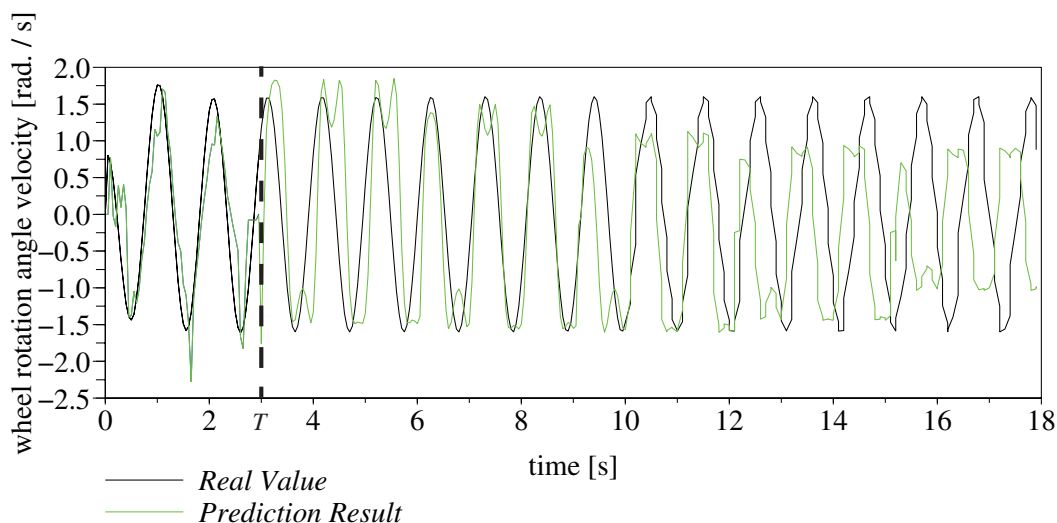


図 5.16 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) の予測結果 (Online SVR と LQR を組み合わせたとき)

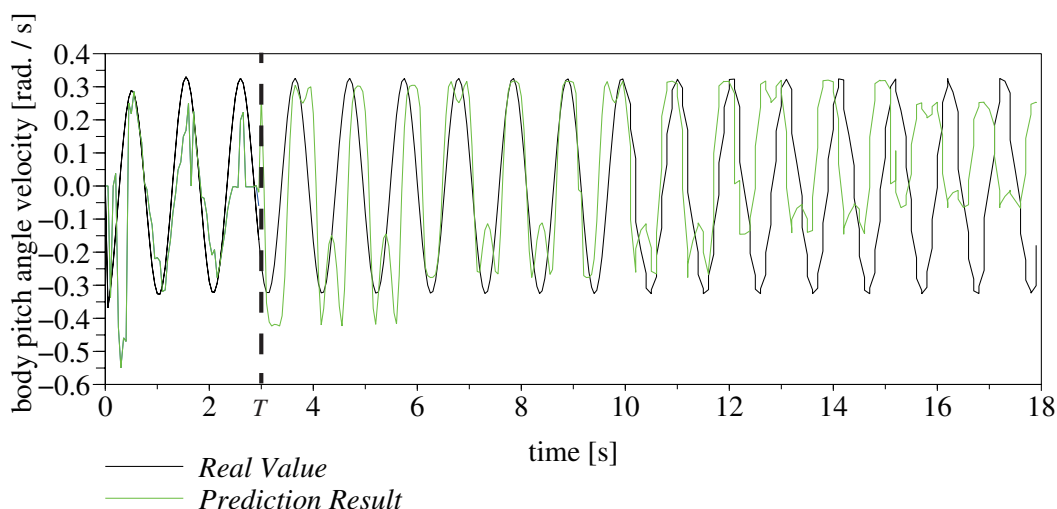


図 5.17 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測結果 (Online SVR と LQR を組み合わせたとき)

\mathbf{x}_1 及び u の予測結果に於いて、ここでは図 5.14 に示した時刻 T , 即ち状態行動対予測を開始して以降の変動について議論してゆく事とする。

これら予測結果の図に於いて、黒色実線は学習データの軌跡を、緑色実線は予測結果の軌跡を表現している。 \mathbf{x}_1 及び u の予測結果に着目すると、Online SVR のみによって状態行動対予測を実施した場合同様に、ほぼリアルタイムに学習データに追従する予測結果を導いている事が判る。図 5.14 及び図 5.17 に於いては、ほぼ学習データに等しい値を予測すると共に、その傾向を良く示す予測結果を示している事が確認出来る。ただ一方で、混入された外乱を含む内部状態の応答に注目すると、時間の経過に従い安定状態である鉛直近傍へ近接する様な挙動を予測し

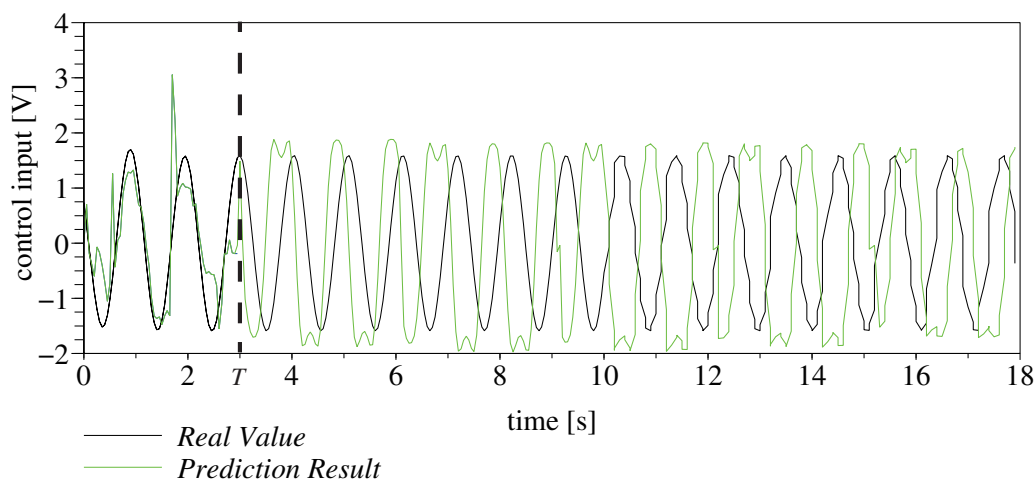


図 5.18 行動 u (コントローラからの最適入力) の予測結果 (Online SVR と LQR を組み合わせたとき)

ている事が見て取れる.

続いて, それぞれの状態及び行動に対する予測誤差を計算したグラフを次に示す.

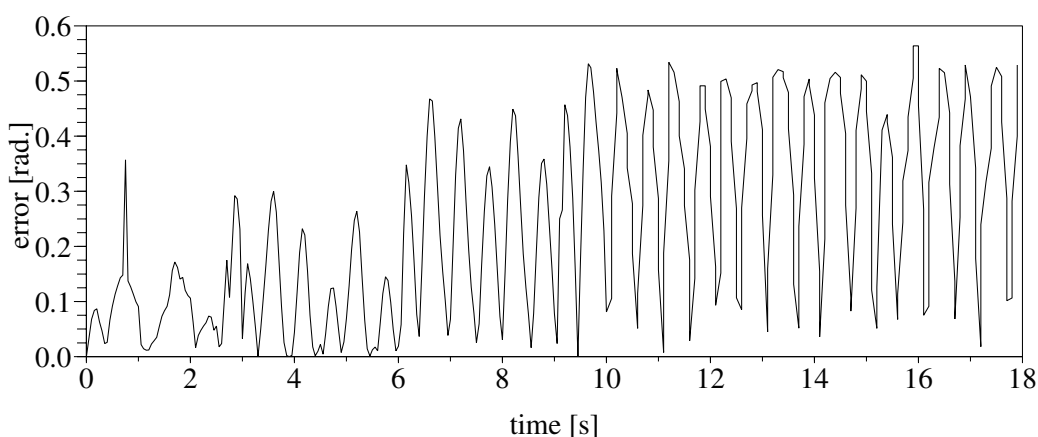


図 5.19 状態 $x_{1,1}$ (車輪回転角度 θ) の予測誤差 (Online SVR と LQR を組み合わせたとき)

ここで, 上記の予測誤差は以下に従い算出出来る.

$$e_i(t) = |\hat{x}_{1,i}(t) - x_{1,i}(t)|, \quad i \in \dim \mathbf{x}_1(t) \quad (5.3.5)$$

$$e_u(t) = |\hat{u}(t) - u(t)| \quad (5.3.6)$$

この予測誤差に注目する. 図 5.19 から図 5.22 に示した各々の状態と, 図 5.23 に示した行動に対する予測誤差は, 予測を開始した時刻 T より遠い時間になるほど, その時間の遠さに比例して増加している事が見て取れる. この結果について考えてみると, 未来の予測値は, Online SVR によって算出された予測値を繰返し用いる事によって導いている. その為, この様な現象

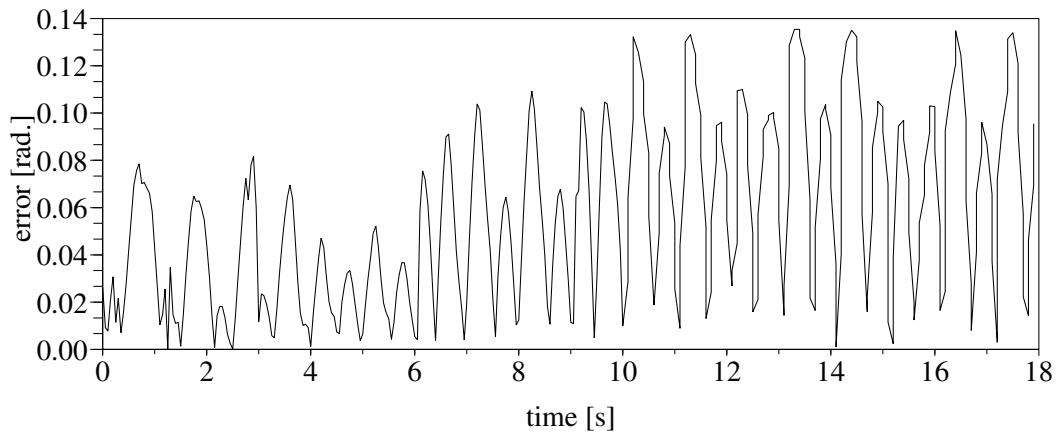


図 5.20 状態 $x_{1,2}$ (ピッチ角度 ψ) の予測誤差 (Online SVR と LQR を組み合わせたとき)

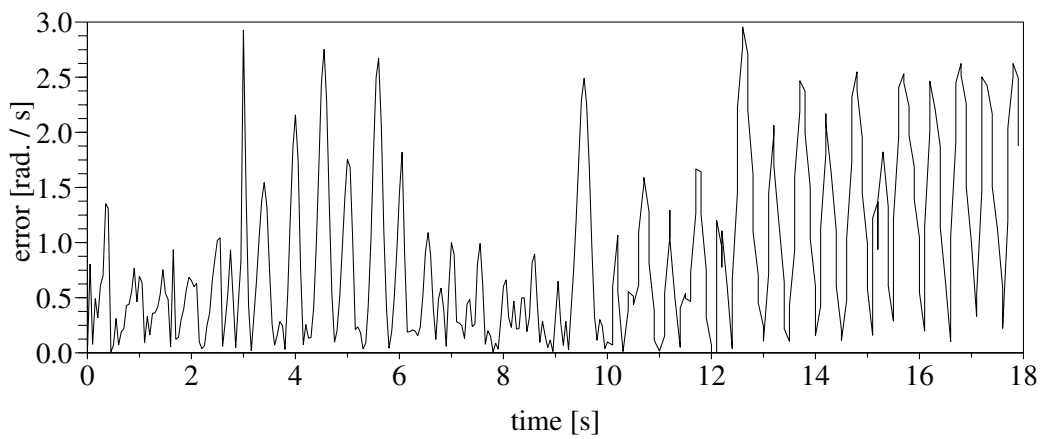


図 5.21 状態 $x_{1,3}$ (車輪回転角速度 θ) の予測誤差 (Online SVR と LQR を組み合わせたとき)

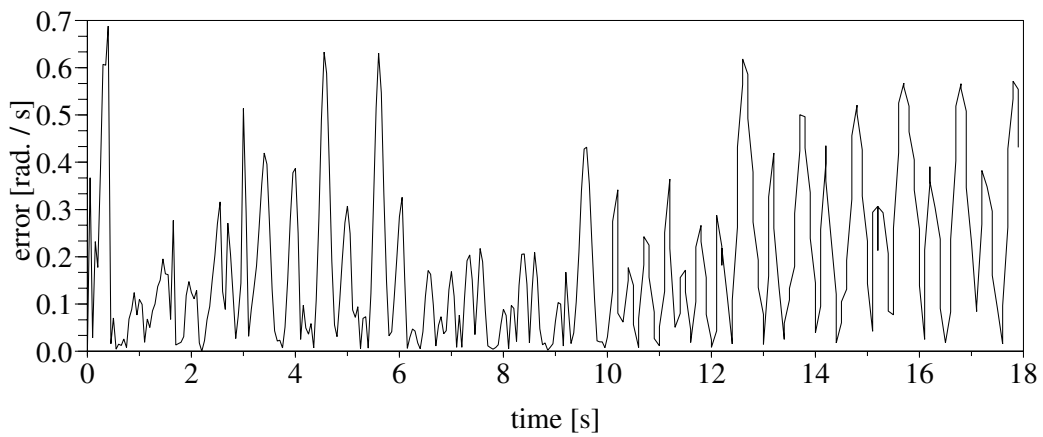


図 5.22 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) の予測誤差 (Online SVR と LQR を組み合わせたとき)

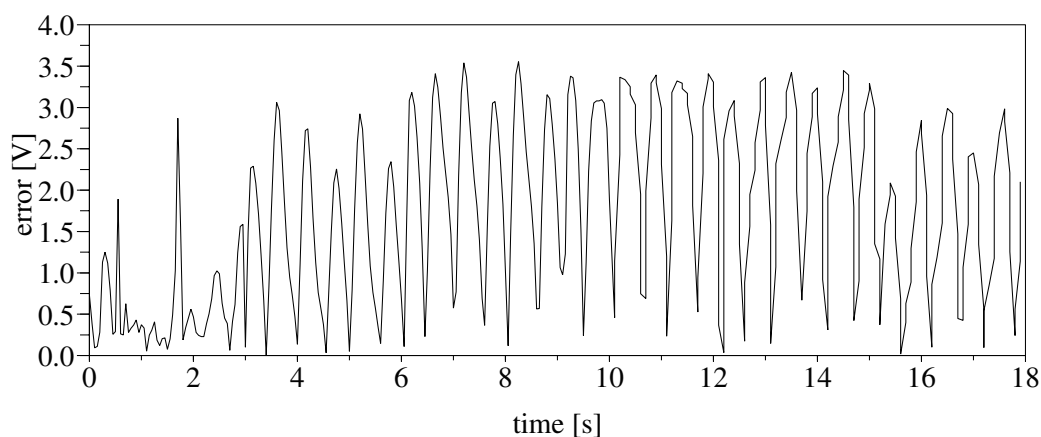


図 5.23 行動 u (コントローラからの最適入力) の予測誤差 (Online SVR と LQR を組み合わせたとき)

を得た理由としては、次に挙げる 2 点が考えられる。

1 点目として、初期のパラメータ設定が挙げられる。予測結果は、 ϵ 不感帯パラメータと正則化パラメータによって決定される。ここでは、汎化性を考慮して小さな正則化パラメータと広めな不感帯パラメータを設定している。この様な場合、学習器は、外乱として与えたデータを許容誤差と認められるノイズであると見做す可能性がある。2 点目に、検証実験に於いて入力した外乱信号そのものの特性を考慮する必要がある。この場合、学習器は、小さな振幅の外乱信号を、設定した不感帯の範囲内に含まれるものと見做して無視する可能性がある。

ただしこの実験では、Online SVR と LQR を組合わせて状態行動対を実施している。この為、LQR は予測結果を安定した状態へ遷移させる行動を得ようとする。従って、予測結果は Online SVR のみの場合に比べ、実際の制御結果に近い予測値を計算する事が考えられる。この予測結果を行動決定に適用する場面を考える。SVR と LQR を組み合わせて状態行動対予測をした場合は、SVR のみによる予測と同じく、予測誤差に関しては、用いる予測結果に対して重みを付ける事によって対処出来る。但し、予測可能な外乱であっても、予測結果と実際の外乱それぞれの傾向が異なる場合は、重みによる対処のみでは難しいものと考えられる。従って、予測結果を行動決定に用いることが可能な範囲は有限であり、尚且つ事前に学習したデータ数に依存した範囲になるものとする。従って、得られた結果より、状態行動対予測として得られる予測結果の有効な範囲考慮すれば、適切なものとなると結論付ける。

5.4 まとめ

本論文では, ロボットが保有する状態ととられる行動に着目し, Online SVR と LQR を学習器かつ予測器として適用する事により, 未来の状態と行動を予測する為の手法を提案している.

特に本章では, この提案手法を検証する為シミュレーションを実施した. 当該のシミュレーションでは, Online SVR のみによって予測器を構成する方法, あるいは Online SVR と LQR を組合わせて予測器を構成する方法の, 2 つのアプローチから検証し, その予測結果を示した. 学習データとなる応答が滑らかであるとき, 提案手法による予測結果は, 学習データにほぼ等しい傾向を示した. その一方で, 時間経過に従い予測結果は予測誤差を蓄積する事から, 予測の傾向が学習データに対して反転する結果も確認された. これは, 将来の予測に対して, 予測器が算出した予測値を直接的に用いた事が原因である.

この点を踏まえて実験結果を考えると, 得られた結果は極めて妥当なものであると結論付けられる.

第6章 実験 II: 倒立振子に対する行動補正 -固定係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

本章では, 状態行動対予測より得た未来の予測値を用いて, 将来採るべき行動を考慮した最適行動を現時刻で獲得することを考える.

そこで, 前章と同じく倒立振子に対して状態行動対予測を適用すると共に行動補正を組み合わせる. そして計算機シミュレーションによってその有効性を検証する. 特に本章では, 未来の予測値に対し固定の重み係数を乗じることによって, 予測誤差の影響を低減する事を試みる.

6.1 実験の目的

前章にて, 状態行動対予測を二輪型倒立振子へ適用し, 二輪型倒立振子が持つ状態と行動の予測を試みた. その結果として, 極めて良好な予測結果を得た.

本章では, 状態行動対予測を, 従来使われてきた最適制御と組み合わせる事を考える. 状態と行動の組合せで表現出来るロボット制御に対し提案手法を適用する事で, 未来の行動を学習器によって予測する. そしてこの予測した未来の行動を現在採る行動に反映させる事で, 行動の補正を行う仕組みを実現する. また, 未来に採るべき行動の予測値に含まれる予測誤差を考慮し, 生成する行動には重み付けを行い, その影響を低減する. 適用例として, 自走式二輪型倒立振子

第 6 章 実験 II: 倒立振子に対する行動補正 -固定係数による信頼度付き状態行動対予測結果を行動決定に反映した場合-

に LQR と組み合わせ倒立姿勢維持タスクを実行させた. この間に予測可能な外乱を持続的に与え続けた場合の動作を, 計算機シミュレーションによって確認し, その結果を検証する.

6.2 事前実験

6.2.1 システム構築

ここでは, 内部状態と行動を予測する為に用いる学習データの取得の為事前に必要となるシミュレーション実験について説明する. 本研究では NXTway-GS を, 二輪型倒立振子としてモデリングした. NXTway-GS を側面図 (図 4.6(a)) と平面図 (図 4.6(b)) に分け, これに従い運動方程式を考えている [78][79]. 同図で現れる主要な変数は表 4.3 にて既に示してある.

Lagrange 方程式 [81] を基に, 鉛直近傍で線型化した運動方程式を立てる事で, 次式を得て

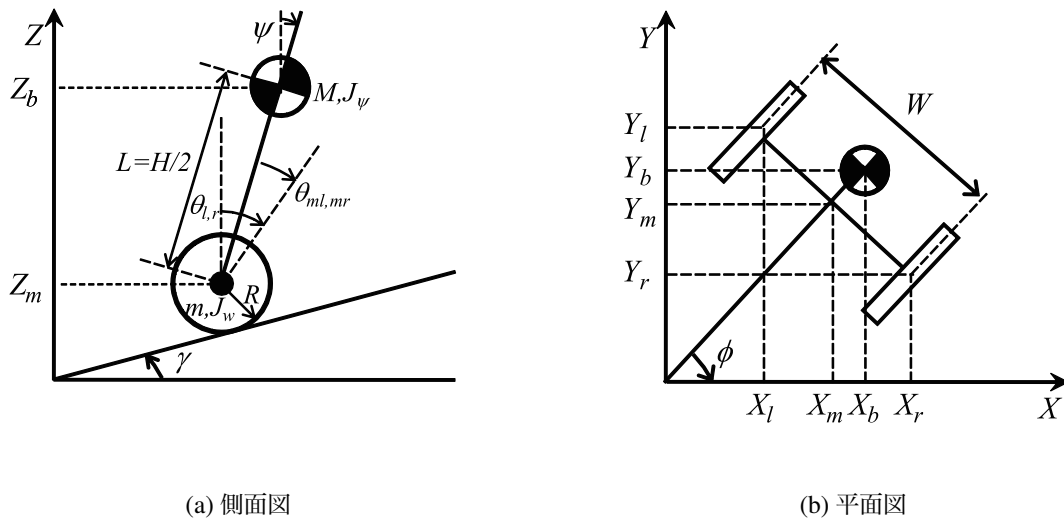


図 6.1 NXTway-GS の側面・平面図 (図 4.6 の再掲)

いる

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

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

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

またこの運動方程式を $\mathbf{x}_1 = [\theta \ \psi \ \dot{\theta} \ \dot{\psi}]^T$, $\mathbf{u} = [v_l \ v_r]^T$, $y = \theta$ とすることで,

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

を既に得ている. この状態方程式と, 最適レギュレータ法を適用する事で算出したフィードバックゲインを配置した状態フィードバックにより, 制御器を構成した [78][79]. この様にして得られる事前実験の概要が図 6.2 である.

6.2.2 学習データの取得

事前実験に於いては, 予測可能な外乱をモータへ与える (図 6.3). 実験条件は表 6.1 に従う. この事前実験より学習データが得られる.

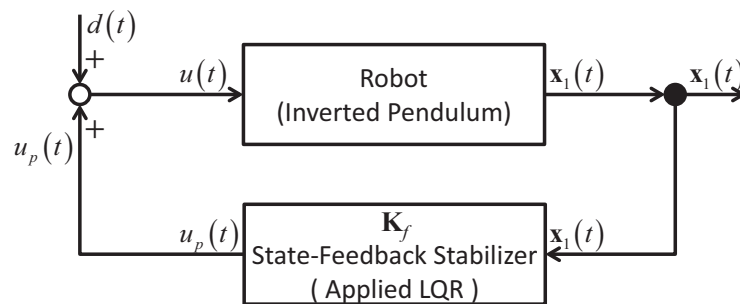


図 6.2 事前実験の為のシステム構成 (2)

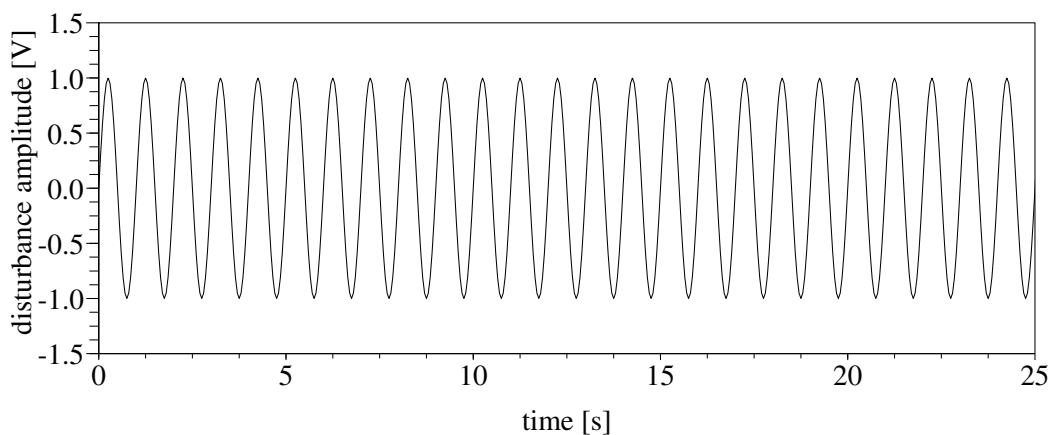


図 6.3 モデルへ与える外乱入力 $\mathbf{d}(t)$ (2)

第 6 章 実験 II: 倒立振子に対する行動補正 -固定係数による信頼度付き状態行動対予測結果を行動決定に反映した場合-

表 6.1 事前実験に係わる実験パラメータ (2)

定数	数値	単位	意味
ψ_0	0.0262	[rad.]	初期ピッチ角度
γ_0	0.0	[rad.]	進行方向傾斜路傾斜角
t_s	0.05	[s]	サンプリングレート
$t_{d,start}$	0.0	[s]	予測可能な外乱を与え始める時間
$t_{d,finish}$	25.0	[s]	予測可能な外乱を与え終わる時間
A_{d1}	1.0	[V]	予測可能な外乱の振幅
f_{d1}	4.0	[Hz]	予測可能な外乱の周波数

6.3 検証実験

6.3.1 実験設定

検証実験に於いては, 学習器として, RBF カーネルを組込んだ Online SVR を **State Predictor** に, LQR を **Action Predictor** に用いている. 学習データについては, 先述の事前実験で得た制御応答と行動の数値データ (図 6.4 から図 6.8 に於ける黒色実線で示される曲線) を採用する. 学習器の学習パラメータを表 6.2 に, 検証実験に於けるその他の実験パラメータを表 6.3 に示す. また, 状態行動対予測に於ける **State Predictor** と **Action Predictor**, 及び行動補正については, 次式を適用する.

$$\hat{x}_{1,i}(t+1) = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}(x\mathbf{z}(t))^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_l)^{-1} x\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_1(t)$

$$\hat{u}(t+1) = \mathbf{k}_f \hat{\mathbf{x}}_1(t+1) \quad (4.1.1)$$

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{u}(t+i) \quad (3.5.1)$$

$$\mathbf{u}_p(t) = \mathbf{k}_f \cdot \mathbf{x}_1(t) \quad (6.3.1)$$

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.2)$$

表 6.2 検証実験時の Online SVR 及び LQR に於けるパラメータ (2)

定数	数値	意味
C_1	300	x_1 に対する正則化パラメータ
ϵ_1	0.02	x_1 に対する ϵ 不感帯パラメータ
β_1	30	x_1 に対する RBF カーネルパラメータ
C_2	300	x_2 に対する正則化パラメータ
ϵ_2	0.02	x_2 に対する ϵ 不感帯パラメータ
β_2	30	x_2 に対する RBF カーネルパラメータ
C_3	300	x_3 に対する正則化パラメータ
ϵ_3	0.02	x_3 に対する ϵ 不感帯パラメータ
β_3	30	x_3 に対する RBF カーネルパラメータ
C_4	300	x_4 に対する正則化パラメータ
ϵ_4	0.02	x_4 に対する ϵ 不感帯パラメータ
β_4	30	x_4 に対する RBF カーネルパラメータ
\mathbf{k}_f	$\begin{bmatrix} -0.870 \\ -32.2 \\ -1.16 \\ -2.81 \end{bmatrix}^T$	u に対する状態フィードバックゲイン

表 6.3 検証実験に係わる実験パラメータ (1)

定数	数値	単位	意味
t_s	0.05	[s]	サンプリングレート
N_s	60	—	初期の一括学習に係る初期データセット長
N_{max}	440	—	予測に用いる最大データセット長
N	20	—	行動補正で用いる予測値のステップ数
$\alpha_i, i \in N$	$\frac{N+i-1}{100N}$	—	予測した行動 $\hat{u}(t+i), i \in N$ に対する重み

6.3.2 実験結果とその検討

図 6.4 から 6.7 は状態 \mathbf{x}_1 に対する補正結果を、図 6.8 は行動 u に対する補正結果を示している。

上記の結果をそれぞれ確認すると、赤色破線で示した補正結果は時間経過に従ってゼロに収

第 6 章 実験 II: 倒立振り子に対する行動補正 -固定係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

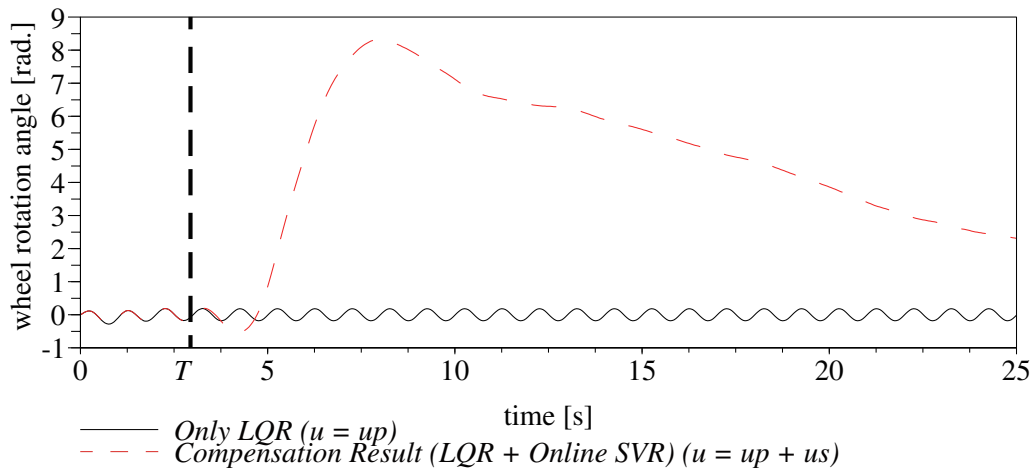


図 6.4 状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (1)

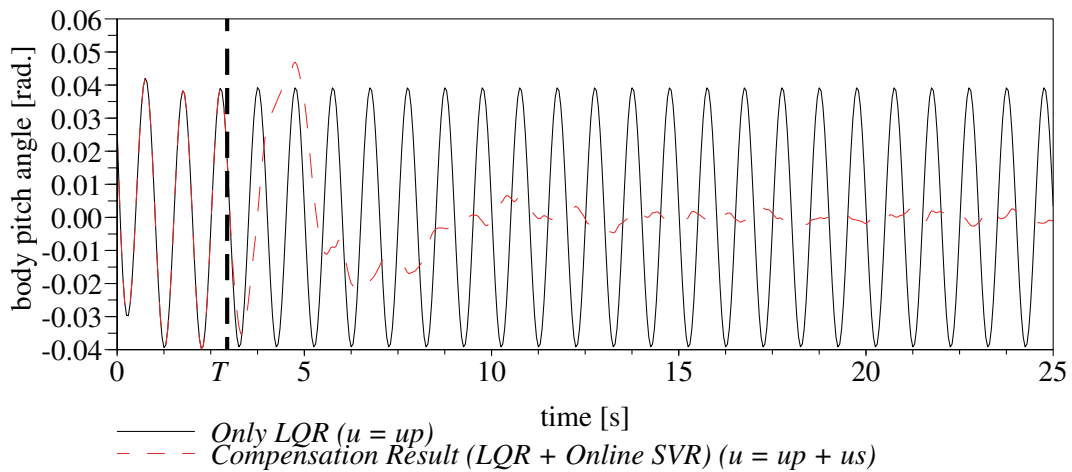


図 6.5 状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (1)

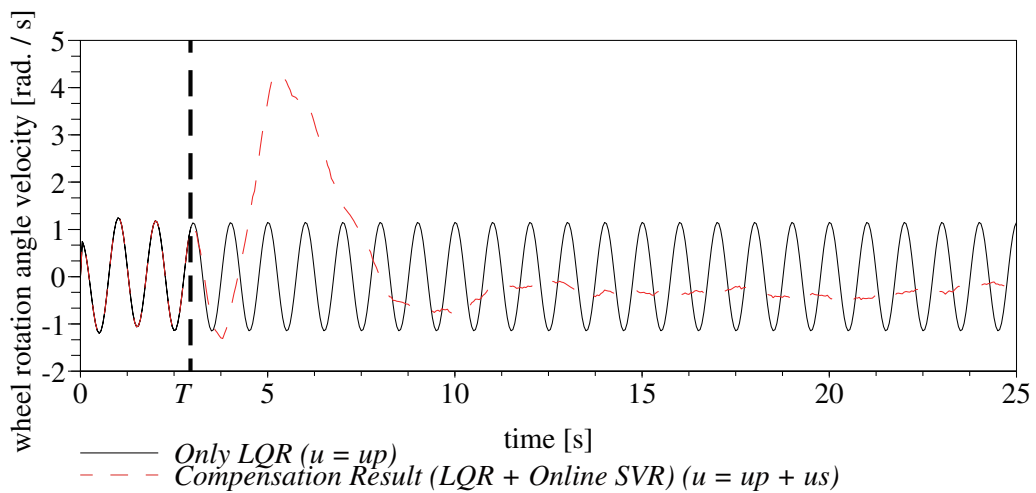


図 6.6 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (1)

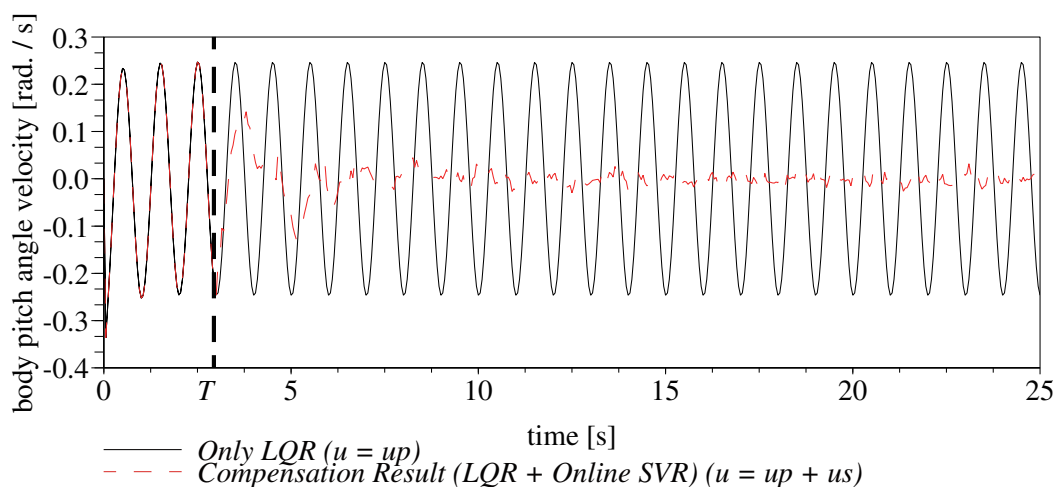


図 6.7 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (1)

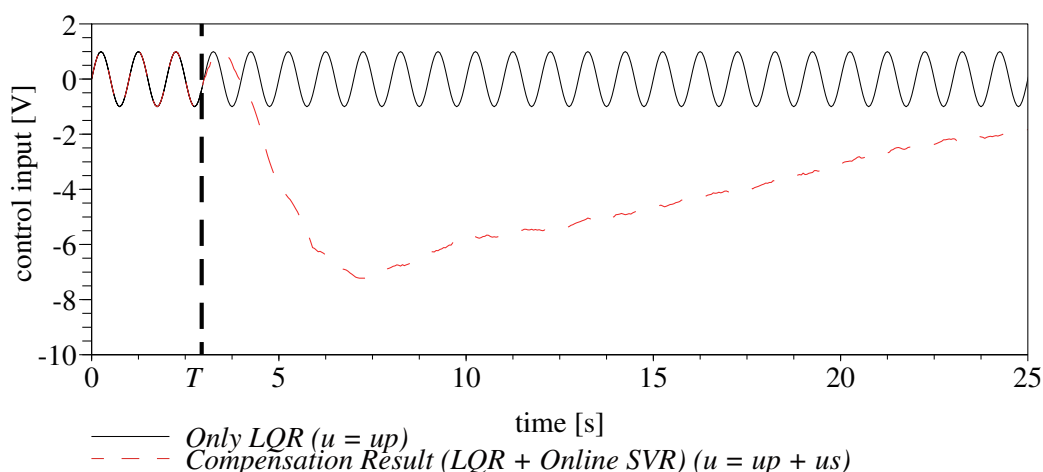


図 6.8 行動 u (コントローラからの最適入力) に対する補正結果 (1)

束するか、接近或いはゼロ周辺で振動している事が確認出来る。この傾向について考えてみる。まずこの収束の理由としては、予測器の出力が第一に要因として挙げられる。状態行動対予測による予測器は、制御対象を安定化させる為に、これまでの状態と行動を学習する。この学習を基に未来に於ける状態と行動を予測し、次時刻に於ける行動を決定する。ここで行動は LQR により予測される。つまり、この時所望の安定状態を維持しようと行動を決定し続ける。従ってこのシステムでは、状態行動対予測の結果を直接的に用いる事で、状態及び行動は経時的にゼロへ収束する様な安定化動作を実行するものと結論付けられる。

この理由から、アクチュエータに対し予測可能な外乱を与え続けた場合、予測によって二輪型倒立振子は安定状態へ遷移する様な行動を予測から決定されるものとなる。従って、状態行動対予測と行動補正を組み合わせる事によって、システムはロバストになるので、得られた実験結果

は妥当であると考える.

6.4 まとめ

本章では, 第 3 章で示した状態行動対予測から得た未来の行動を用いる事で, 毎時刻に於いてロボットが採るべき行動を補正する方法に対し, 計算機シミュレーションによる検証を試みた. 検証結果より, アクチュエータへ予測可能な外乱を加えた場合, 補正結果は LQR を組み合わせる事で経時的にゼロへ収束することを確認した. 従って, 状態行動対予測と行動補正を組み合わせる事で, 未来の予測結果を基に安定した状態へ遷移し維持する様な行動を決定出来るものと結論付ける.

第7章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

本章では, 状態行動対予測より得た未来の予測値を用いて, 将来採るべき行動を考慮した最適行動を現時刻で獲得することを考える.

そこで, 前章と同じく倒立振子に対して状態行動対予測と行動補正を組み合わせ, 現時点に於ける行動を補正する. そして計算機シミュレーションによってその有効性を検証する. 特に本章では, 未来の予測値の統計的特性に着目し, 動的に重み係数を決定する. この重み係数を行動決定に直接的に作用させる事により, 予測誤差の影響低減に寄与する事を確認する.

7.1 実験の目的

前章では, 未来に採るべき行動の予測値に含まれる予測誤差を考慮し, 固定した重みを適用した行動補正に注目した. この手法によって, 未来の予測結果を基に安定した状態へ遷移し維持するような行動を決定し続ける事を, 計算機シミュレーションによって確認した.

本章では, 状態行動対予測による予測結果の『ばらつき』に注目し, 重み係数を動的に変更する仕組みを持った行動補正の手法に注目し, この手法を LQR と組み合わせ倒立姿勢維持タスクを自走式二輪型倒立振子に実行させる. ここで, 以下に示す実験を計算機シミュレーション

によって実施し, その結果を議論してゆく.

- (i) アクチュエータへ予測可能な外乱を持続的に与え続けた場合.
- (ii) 外部コマンドとして前進後退指令を与え路面の形状を変化させた場合.
- (iii) 外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合.

7.2 アクチュエータへ予測可能な外乱を持続的に与え続けた場合

本節では, 二輪型倒立振子のアクチュエータに対し予測可能な外乱を持続的に与え続けた時の提案手法の挙動を観察し, その結果について検討する.

7.2.1 事前実験

7.2.1.1 システム構築

状態行動対予測を実施し行動補正を行う上で必要となる学習データを取得する為, 事前に実施するシミュレーションについて説明する. 本研究では NXTway-GS を, 二輪型倒立振子としてモデリングしている (第4章参照). NXTway-GS を側面図 (図4.6(a)) と平面図 (図4.6(b)) に分け, Lagrange 方程式^[81]を基に運動方程式を考える^{[78][79]}. 同図で現れる主要な変数は表4.3の通りである.

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

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

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

この運動方程式を $\mathbf{x}_1 = \begin{bmatrix} \theta & \psi & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$, $\mathbf{u} = \begin{bmatrix} v_l & v_r \end{bmatrix}^T$, $y = \theta$ とする事で,

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

を得ている. この状態方程式と, 最適レギュレータ法を適用する事で算出したフィードバックゲインを配置した状態フィードバックにより, 制御器を構成した^{[78][79]}. この様にして得られ

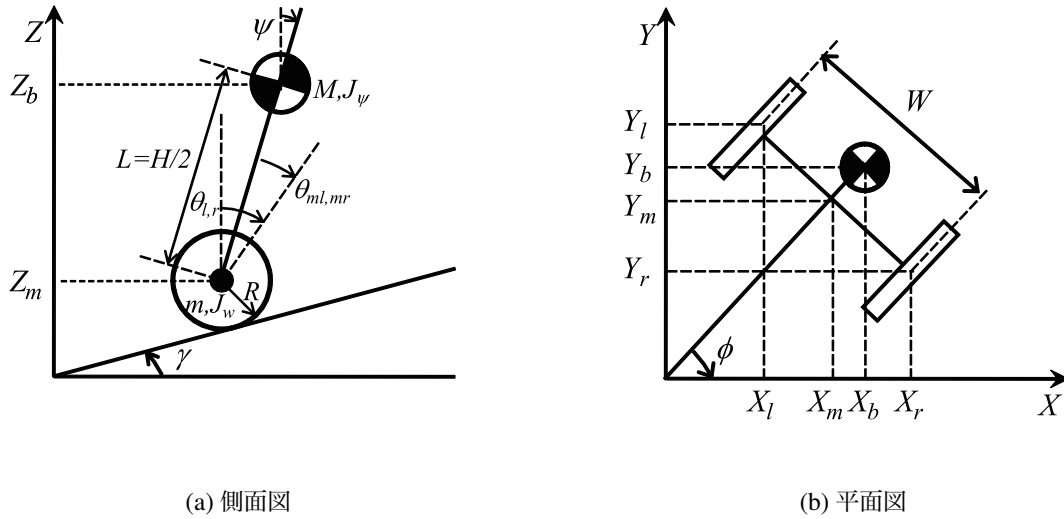


図 7.1 NXTway-GS の側面・平面図 (図 4.6 の再掲)

る事前実験の概要が図 7.2 である。

7.2.1.2 学習データの取得

事前実験に於いては、予測可能な外乱 $d(t)$ をモータへ与える (図 7.3)。ここで $d(t)$ は次式で示される。尚、実験条件は表 7.1 に従う。

$$d(t) = A_{d1} \sin(2\pi f_{d1}t) \tag{7.2.1}$$

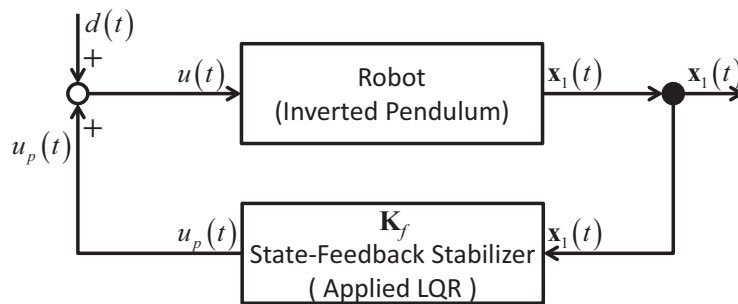


図 7.2 事前実験の為のシステム構成 (3)

第 7 章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

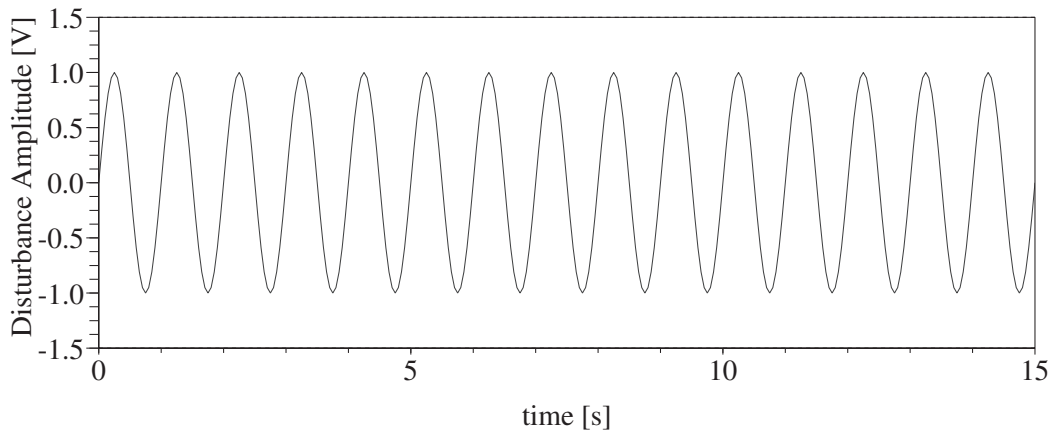


図 7.3 モデルへ与える外乱入力 $d(t)$ (3)

表 7.1 事前実験に係わる実験パラメータ (3)

定数	数値	単位	意味
ψ_0	0.0262	[rad.]	初期ピッチ角度
γ_0	0.0	[rad.]	進行方向傾斜路傾斜角
t_s	0.05	[s]	サンプリングレート
$t_{d,start}$	0.0	[s]	予測可能な外乱を与え始める時間
$t_{d,finish}$	25.0	[s]	予測可能な外乱を与え終わる時間
A_{d1}	1.0	[V]	予測可能な外乱の振幅
f_{d1}	1.0	[Hz]	予測可能な外乱の周波数

7.2.2 検証実験

7.2.2.1 実験設定

検証実験に於いては, 学習器として, RBF カーネルを組込んだ Online SVR を **State Predictor** に, LQR を **Action Predictor** に用いている. 学習データについては, 前述の事前実験で得た制御応答と行動の数値データ (図 7.4 から図 7.8 に於ける黒色実線で示される曲線) を採用する. 学習器の学習パラメータを表 7.2 に, 検証実験に於けるその他の実験パラメータを表 7.3 に示す. また, 状態行動対予測に於ける **State Predictor** と **Action Predictor**, 及び行動補正につい

7.2 アクチュエータへ予測可能な外乱を持続的に与え続けた場合

表 7.2 検証実験時の Online SVR 及び LQR に於けるパラメータ (3)

定数	数値	意味
C_1	300	x_1 に対する正則化パラメータ
ϵ_1	0.02	x_1 に対する ϵ 不感帯パラメータ
β_1	30	x_1 に対する RBF カーネルパラメータ
C_2	300	x_2 に対する正則化パラメータ
ϵ_2	0.02	x_2 に対する ϵ 不感帯パラメータ
β_2	30	x_2 に対する RBF カーネルパラメータ
C_3	300	x_3 に対する正則化パラメータ
ϵ_3	0.02	x_3 に対する ϵ 不感帯パラメータ
β_3	30	x_3 に対する RBF カーネルパラメータ
C_4	300	x_4 に対する正則化パラメータ
ϵ_4	0.02	x_4 に対する ϵ 不感帯パラメータ
β_4	30	x_4 に対する RBF カーネルパラメータ
\mathbf{k}_f	$\begin{bmatrix} -0.870 \\ -32.2 \\ -1.16 \\ -2.81 \end{bmatrix}^T$	u に対する状態フィードバックゲイン

表 7.3 検証実験に係わる実験パラメータ (2)

定数	数値	単位	意味
t_s	0.05	[s]	サンプリングレート
N_s	60	—	初期の一括学習に係る初期データセット長
N_{max}	241	—	予測に用いる最大データセット長
N	20	—	行動補正で用いる予測値のステップ数
$C_{\sigma 1}$	0.01	—	標準偏差から重みを求める際の係数 (1)
$C_{\sigma 2}$	0.5	—	標準偏差から重みを求める際の係数 (2)
N_{σ}	10	—	標準偏差を求める際の予測値のステップ数
w_{t-j}	$0.55 - 0.05j$	—	移動平均法による時刻 $(t - j)$ に於ける動的な重み係数

第 7 章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

ては, 次式を適用する.

$$\hat{x}_{1,i}(t+1) = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}(\mathbf{x}\mathbf{z}(t))^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_l)^{-1} \mathbf{x}\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_1(t)$

$$\hat{\mathbf{u}}(t+1) = \mathbf{k}_f \hat{\mathbf{x}}_1(t+1) \quad (4.1.1)$$

$$\alpha_j = \frac{C_{\sigma 1}}{C_{\sigma 2} + \sigma_j} \quad (3.5.8)$$

$$\sigma_j = \sigma \left(\frac{w_M}{w_\Sigma} t_1 - M \hat{\mathbf{u}}(t_1 + j), \frac{w_{M+1}}{w_\Sigma} t_1 - M + 1 \hat{\mathbf{u}}(t_1 + j), \dots, \frac{w_1}{w_\Sigma} t_1 + 1 \hat{\mathbf{u}}(t_1 + j) \right) \quad (3.5.9)$$

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t+i) \quad (3.5.10)$$

$$\mathbf{u}_p(t) = \mathbf{k}_f \cdot \mathbf{x}_1(t) \quad (7.2.2)$$

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.11)$$

7.2.2.2 実験結果とその検討

図 7.4 から 7.7 は状態 \mathbf{x}_1 に対する補正結果を, 図 7.8 は行動 \mathbf{u} に対する補正結果を示している.

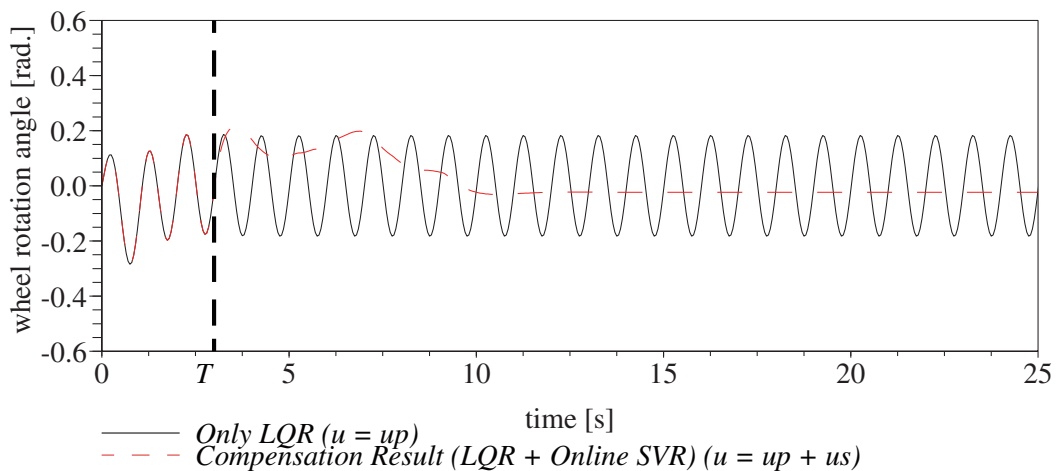
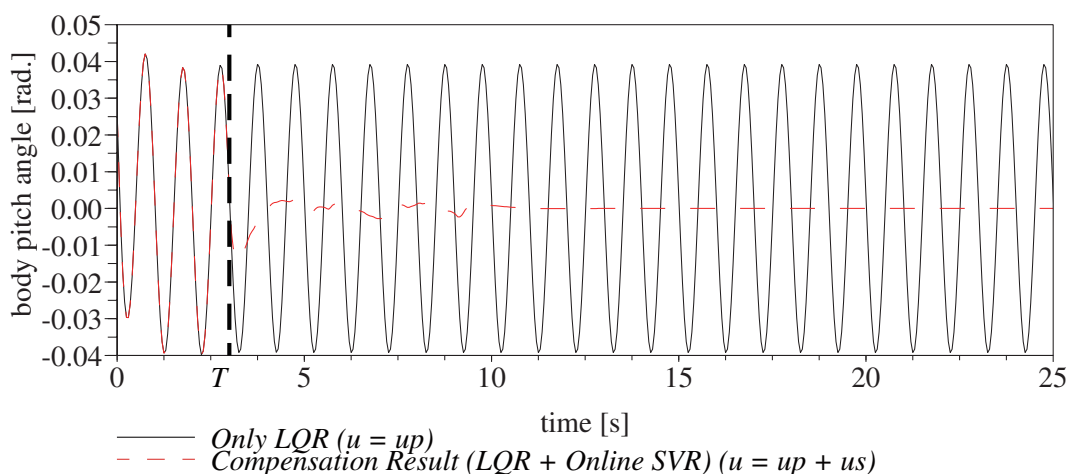
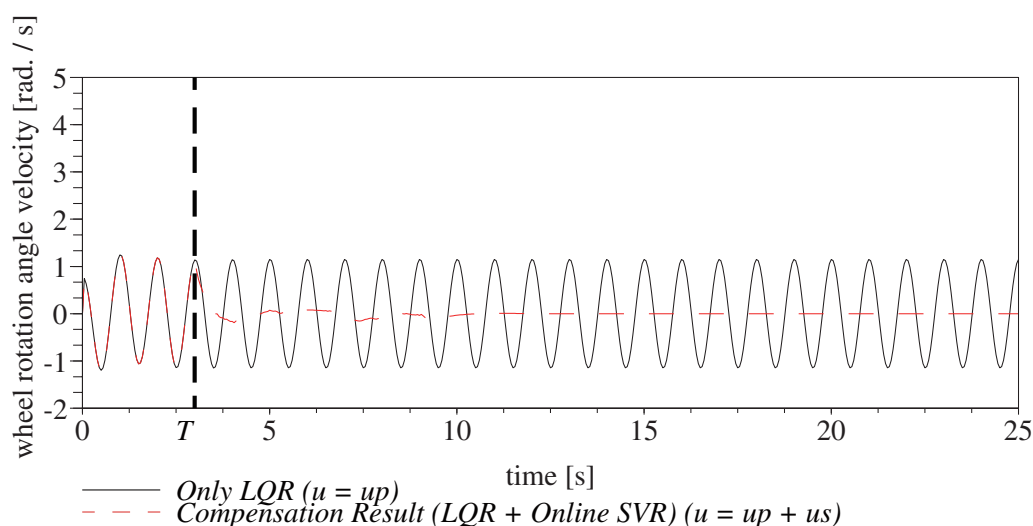


図 7.4 状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (2)

図 7.4 から図 7.7 に於ける赤色一点破線は各状態 $\mathbf{x}_1 = [\theta \ \psi \ \dot{\theta} \ \dot{\psi}]^T$, 図 7.8 に於ける赤色一点破線は行動 \mathbf{u} の補正結果を示している. 実際に予測結果を用い制御を行っている部分は $t = 3.00$ [s] 以降である. ここではこの点を踏まえ, \mathbf{x}_1 及び \mathbf{u} の補正結果に於いて時刻

図 7.5 状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (2)図 7.6 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (2)

$T = 3.00[s]$ 以降の変動について議論してゆく事とする。

先ず図 7.4 に着目すると、車輪回転角 θ が大きく振れている部分が確認出来る。これより、外乱に対して本体を大きく前後に移動する事により、ピッチ角の変動を僅かに留めようとしている事が判る。次に図 7.5 に注目すると、提案手法と LQR を組合わせた結果は、LQR のみに比べピッチ角が僅かに下回っている事が判る。続いて図 7.6、図 7.7 に着目すると、外乱に先んじて角速度が変動している事が判る。図 7.8 については、予測結果を用いて行動を補正している事から、LQR のみに比べ採られる行動が時間的に『先取り』されている事が判る。ところで本研究では、Online SVR を用い状態行動対を予測している為、行動を予測する際には、行動に対する予測誤差に加え、各々の状態を予測した時の予測誤差も含まれる。この為学習器が導いた未来

第7章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

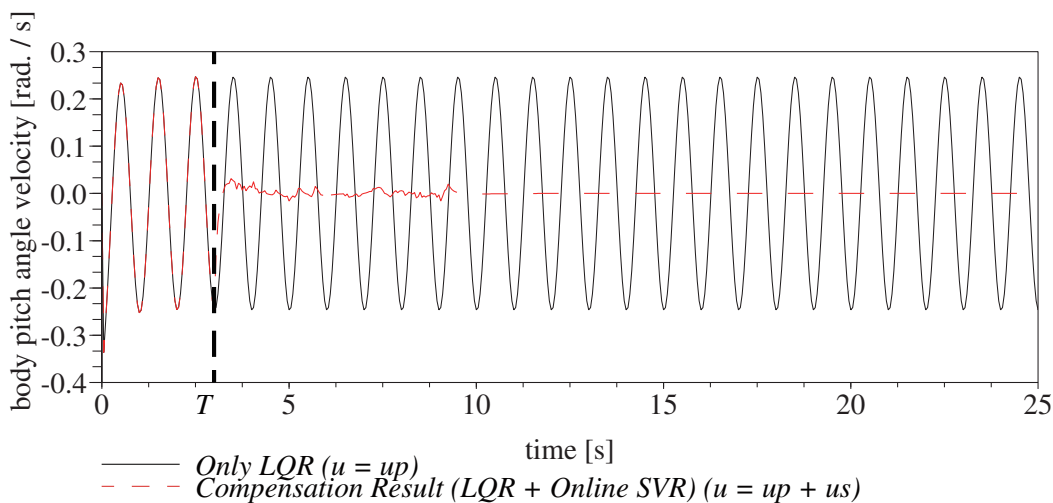


図 7.7 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (2)

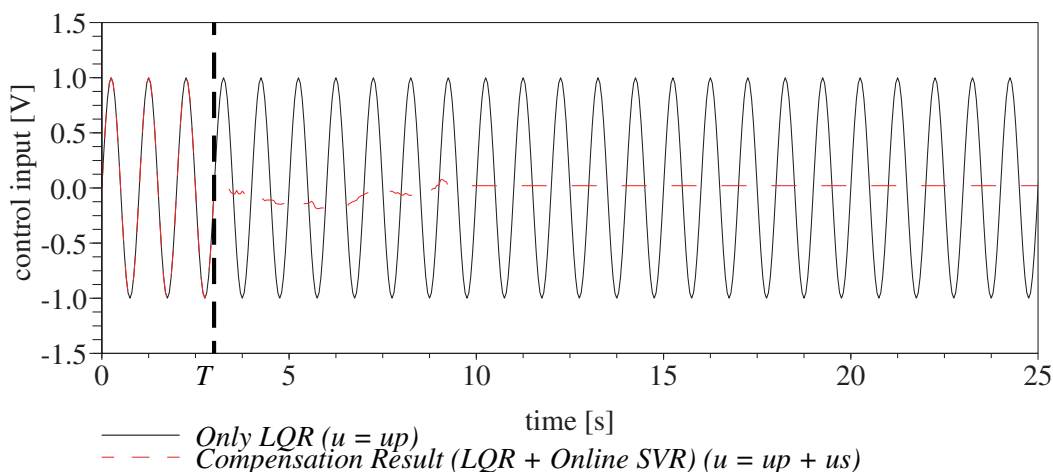


図 7.8 行動 u (コントローラからの最適入力) に対する補正結果 (2)

に採るべき行動には、或る程度の誤差が含まれる。これが原因となり、図 7.4 の様に 3.00[s] 以降で大きく移動する部分が生じたり、目標軌道である $\psi(t) = 0$ 近傍で振動する等の結果が確認されるものとする。但し、この点を踏まえても提案手法により、ピッチ角は安定的に低減出来、目標とするゼロ近傍で僅かに振動する現象を確認出来ている事から、提案手法の有用性を確認したものと結論付ける。

7.3 外部コマンドとして前進後退指令を与え路面の形状を変化させた場合

本節では、二輪型倒立振子に提案手法を適用し、倒立姿勢維持タスクを実行させる。この時、二輪型倒立振子に外部コマンドとして前進指令を与える事で、二輪型倒立振子が自走する。これによって自機が位置する路面の形状が変化する。この変化する艦きゅおかでどうさせた場合を例にとり、提案手法の挙動をシミュレーションによって観察し、その結果について検討する。

7.3.1 事前実験

7.3.1.1 システム構築

状態行動対予測を実施し行動補正を行う上で必要となる学習データを取得する為、事前実施するシミュレーションについて説明する。本研究では NXTway-GS を、二輪型倒立振子としてモデリングしている(第4章参照)。NXTway-GS を側面図(図4.6(a))と平面図(図4.6(b))に分け、Lagrange 方程式^[81]を基に運動方程式を考える^{[78][79]}。同図で現れる主要な変数は表4.3の通りである。

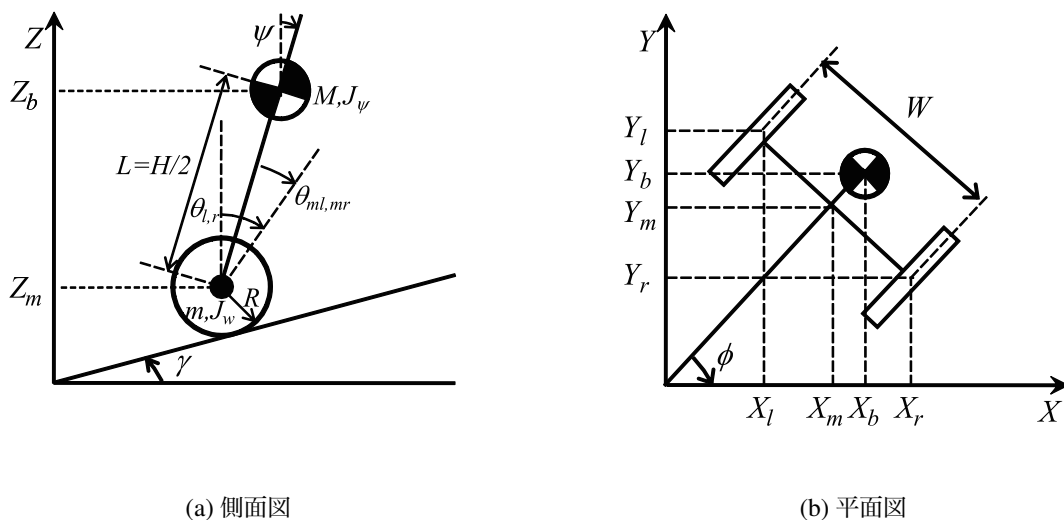


図 7.9 NXTway-GS の側面・平面図(図 4.6 の再掲)

第7章 実験 III: 倒立振り子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

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

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

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

この運動方程式を $\mathbf{x}_1 = [\theta \ \psi \ \dot{\theta} \ \dot{\psi}]^T$, $\mathbf{u} = [v_l \ v_r]^T$, $y = \theta$ とする事で、

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

を得ている。この状態方程式と、最適レギュレータ法を適用する事で算出したフィードバックゲインを配置した状態フィードバックにより、制御器を構成した [78][79]。この様にして得られる事前実験の概要が図 7.10 である。

7.3.1.2 学習データの取得

事前実験に於いては、+1 を前進、0 を静止とし、人間が外部から指令入力を与えるものとする (図 7.11)。これによって、NXTway-GS は平坦な路面と、うねりのある路面を走行する事となる (図 7.12)。ここで、NXTway-GS が走行する路面の形状は、次式 (7.3.1) で定義される。

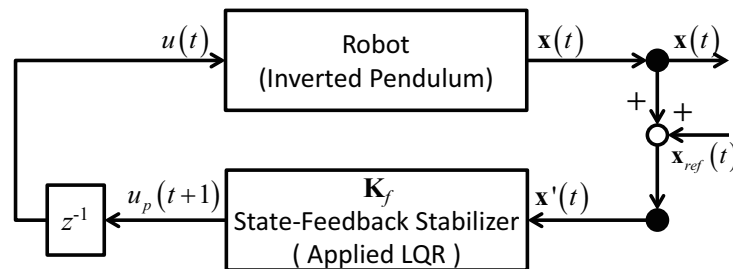


図 7.10 事前実験の為のシステム構成 (4)

$$z = \begin{cases} 1.5 \times 0.15 \sin [2 \times 0.125 \cdot \pi \cdot (x + 0.25)] & \text{if } (x < -0.25) \\ 0 & \text{if } (-0.25 \leq x \leq 4) \\ 0.15 \sin [2 \times 0.125 \cdot \pi \cdot (x + 4)] & \text{if } (x > 4) \end{cases} \quad [\text{cm}] \quad (7.3.1)$$

尚、実験条件は表 7.4 に従う。この事前実験によって、検証実験時に必要となる学習データが得られる。

7.3 外部コマンドとして前進後退指令を与え路面の形状を変化させた場合

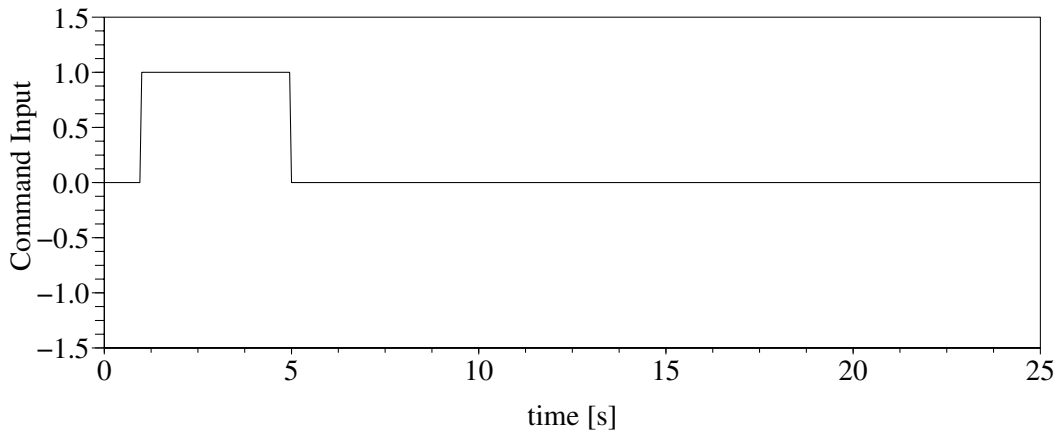


図 7.11 人間からモデルへ与える指令入力 (1)

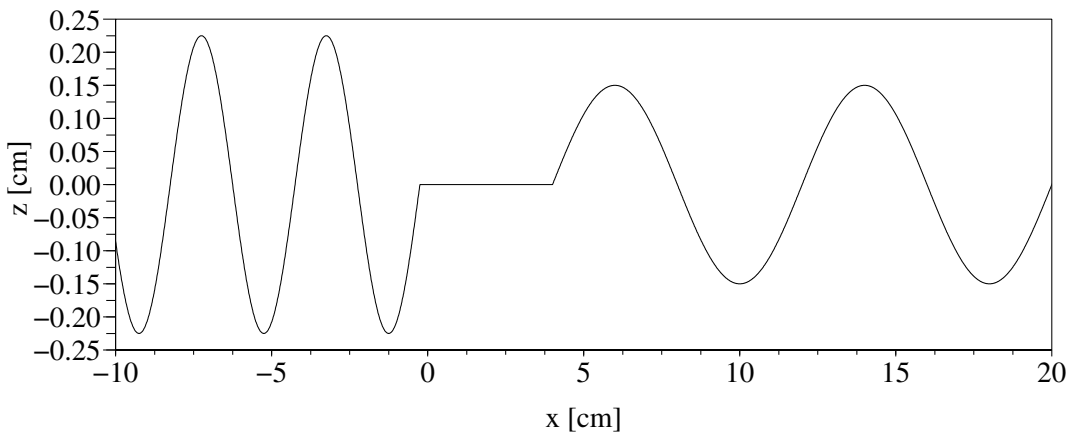


図 7.12 二輪型倒立振子が自走する環境 (床の形状)

表 7.4 事前実験に係わる実験パラメータ (4)

定数	数値	単位	意味
ψ_0	0.0262	[rad.]	初期ピッチ角度
γ_0	0.0	[rad.]	進行方向傾斜路傾斜角
t_s	0.05	[s]	サンプリングレート

7.3.2 検証実験

7.3.2.1 実験設定

検証実験に於いては、学習器として、RBF カーネルを組込んだ **Online SVR** を **State Predictor** に、**LQR** を **Action Predictor** に用いている。学習データについては、先述の事前実験で得た制

第7章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

御応答と行動の数値データ (図 7.13 から図 7.17 に於ける黒色実線で示される曲線) を採用する. 学習器の学習パラメータを表 7.8 に, 検証実験に於けるその他の実験パラメータを表 7.9 に示す. また, 状態行動対予測に於ける **State Predictor** と **Action Predictor**, 及び行動補正につ

表 7.5 検証実験時の Online SVR 及び LQR に於けるパラメータ (4)

定数	数値	意味
C_1	300	x_1 に対する正則化パラメータ
ϵ_1	0.02	x_1 に対する ϵ 不感帯パラメータ
β_1	30	x_1 に対する RBF カーネルパラメータ
C_2	300	x_2 に対する正則化パラメータ
ϵ_2	0.02	x_2 に対する ϵ 不感帯パラメータ
β_2	30	x_2 に対する RBF カーネルパラメータ
C_3	300	x_3 に対する正則化パラメータ
ϵ_3	0.02	x_3 に対する ϵ 不感帯パラメータ
β_3	30	x_3 に対する RBF カーネルパラメータ
C_4	300	x_4 に対する正則化パラメータ
ϵ_4	0.02	x_4 に対する ϵ 不感帯パラメータ
β_4	30	x_4 に対する RBF カーネルパラメータ
\mathbf{k}_f	$\begin{bmatrix} -0.870 \\ -32.2 \\ -1.16 \\ -2.81 \end{bmatrix}^T$	u に対する状態フィードバックゲイン

表 7.6 検証実験に係わる実験パラメータ (3)

定数	数値	単位	意味
t_s	0.05	[s]	サンプリングレート
N_s	60	—	初期の一括学習に係る初期データセット長
N_{max}	501	—	予測に用いる最大データセット長
N	20	—	行動補正で用いる予測値のステップ数
C_σ	0.05	—	標準偏差から重みを求める際の係数
N_σ	10	—	標準偏差を求める際の予測値のステップ数

いては, 次式を適用する.

$$\hat{x}_{1,i}(t+1) = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}(x\mathbf{z}(t))^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_l)^{-1} x\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_1(t)$

$$\hat{u}(t+1) = \mathbf{k}_f \hat{\mathbf{x}}_1(t+1) \quad (4.1.1)$$

$$\alpha_j = C_\sigma \cdot \sigma_j \quad (3.5.4)$$

$$\sigma_j = \sigma \left[{}^{t_1-M} \hat{\mathbf{u}}(t_1+j), {}^{t_1-M+1} \hat{\mathbf{u}}(t_1+j), \dots, {}^{t_1-i} \hat{\mathbf{u}}(t_1+j), \dots, {}^{t_1+1} \hat{\mathbf{u}}(t_1+j) \right] \quad (3.5.5)$$

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t+i) \quad (3.5.6)$$

$$\mathbf{u}_p(t) = \mathbf{k}_f \cdot \mathbf{x}_1(t) \quad (7.3.2)$$

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.7)$$

7.3.2.2 実験結果とその検討

図 7.13 から 7.16 は状態 \mathbf{x}_1 に対する補正結果を, 図 7.17 は行動 u に対する補正結果を示している. ここで改めて確認すると, 実際に予測結果を用い制御を行っている部分は,

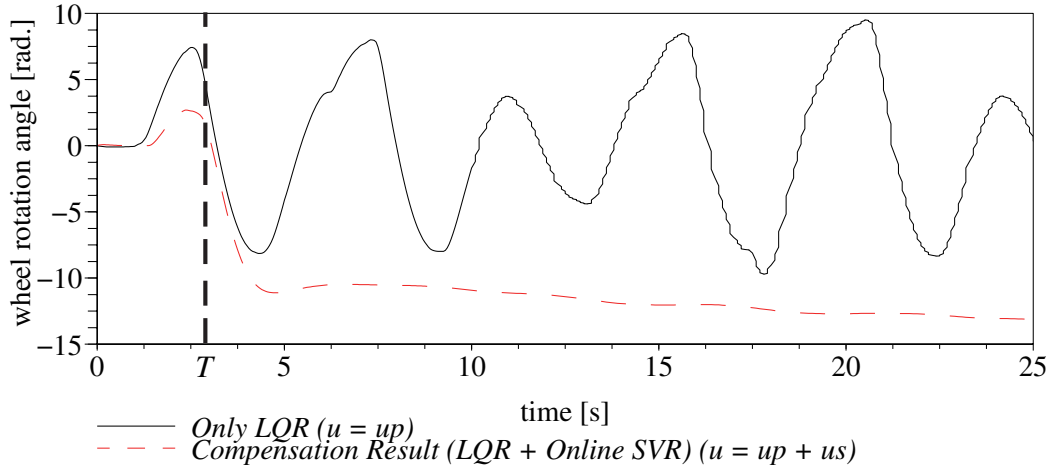


図 7.13 状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (3)

$T = 3.00[\text{s}]$ 以降である. この点を踏まえ, それぞれの補正結果に於いて時刻 T 以降の変動について議論してゆく事とする. まず図 7.14 に注目すると, 提案手法は LQR のみに比べピッチ角が大きく下回っている事が判る. ここで, $\psi(t) = 0$ 近傍に注目し提案手法の挙動を確認すると, 鉛直近傍である 0 近傍で僅かに振動している. 次に図 7.13 に着目すると, 車輪回転角 θ が大きく振れている部分が確認出来る. これより, 指令入力に対して本体を大きく前後に移動する事

第7章 実験 III: 倒立振り子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

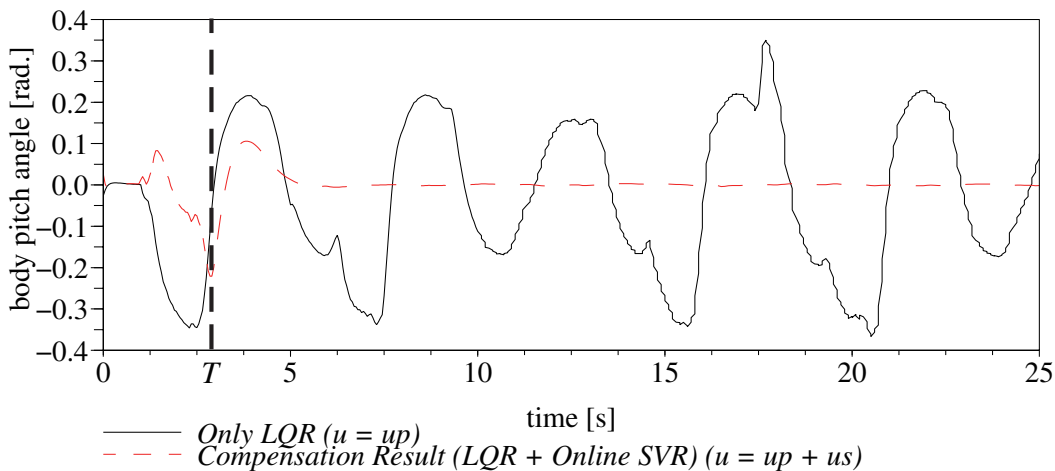


図 7.14 状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (3)

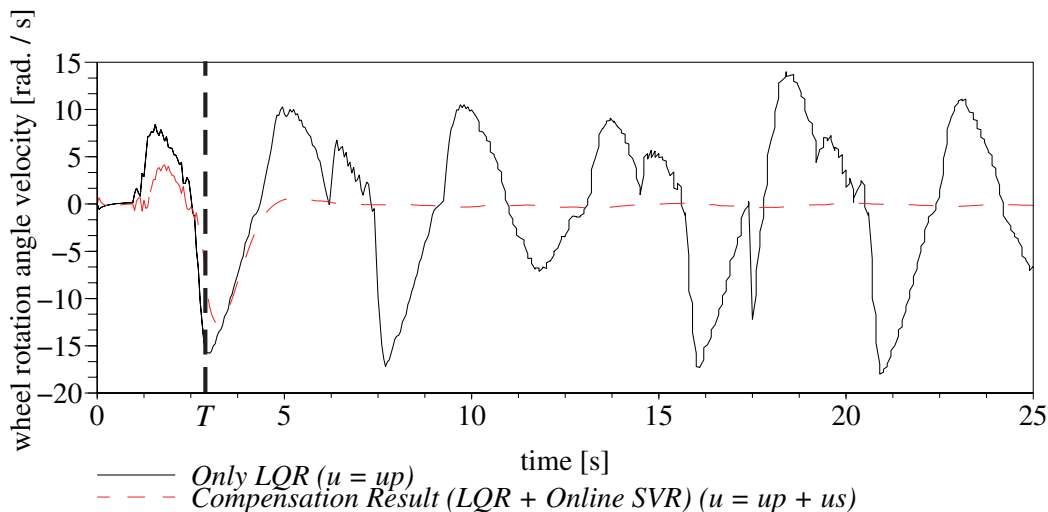


図 7.15 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (3)

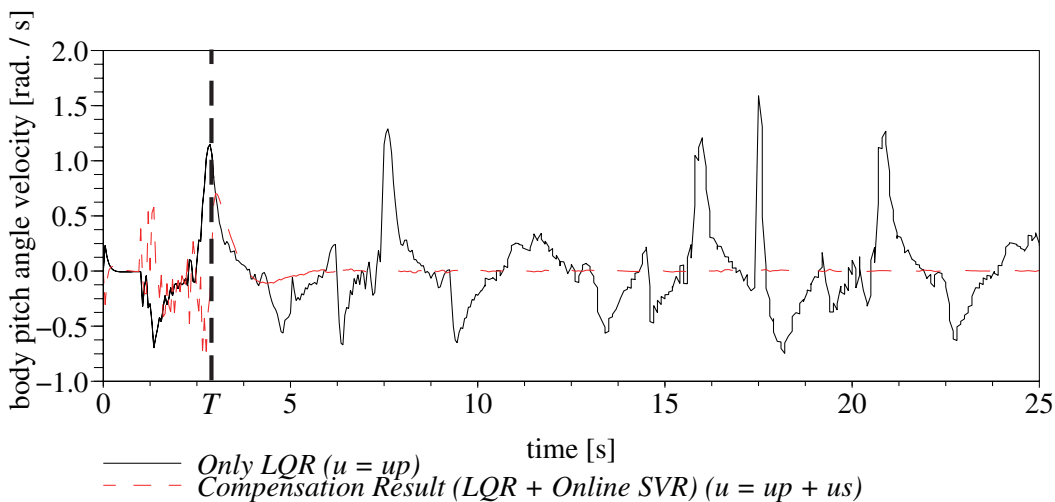


図 7.16 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (3)

7.3 外部コマンドとして前進後退指令を与え路面の形状を変化させた場合

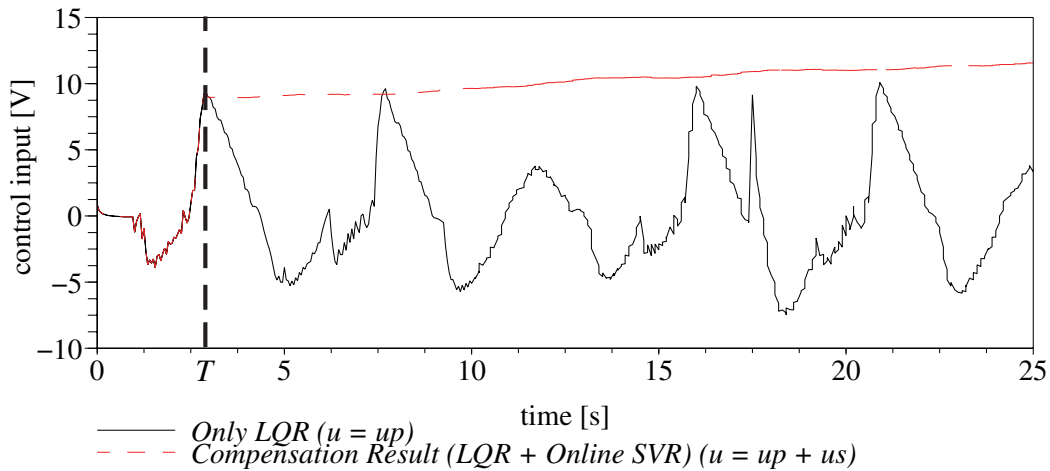


図 7.17 行動 u (コントローラからの最適入力) に対する補正結果 (3)

により、ピッチ角の変動を僅かに留めようとしている事が判る。続いて図 7.15, 図 7.16 に着目すると、ピッチ角速度は速やかに 0 近傍に速度が収束すると共に、車輪回転角速度についても速度は 0 近傍で僅かに変動する程度に抑えられている。図 7.17 については、予測結果を用いて行動を補正している事から、LQR のみに比べ採られる行動が時間的に『先取り』されていると言える。この結果から、指令入力によって移動するが、振子のピッチ角を鉛直近傍に抑えている事が確認出来る。この事実より、提案手法は路面の変動に対してロバストであると言える。続

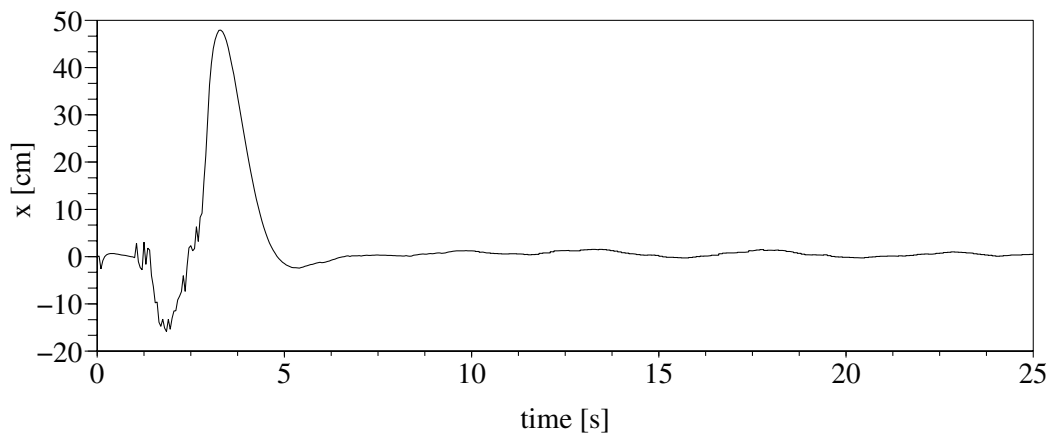


図 7.18 二輪型倒立振子の走行位置の x 軸上に於ける時間変化

いて、図 7.18 及び図 7.19 に着目したい。このグラフでは、指令入力によって得た移動距離と、二輪型倒立振子の重心位置の経時的変動を記録したものとなっている。この移動距離については、

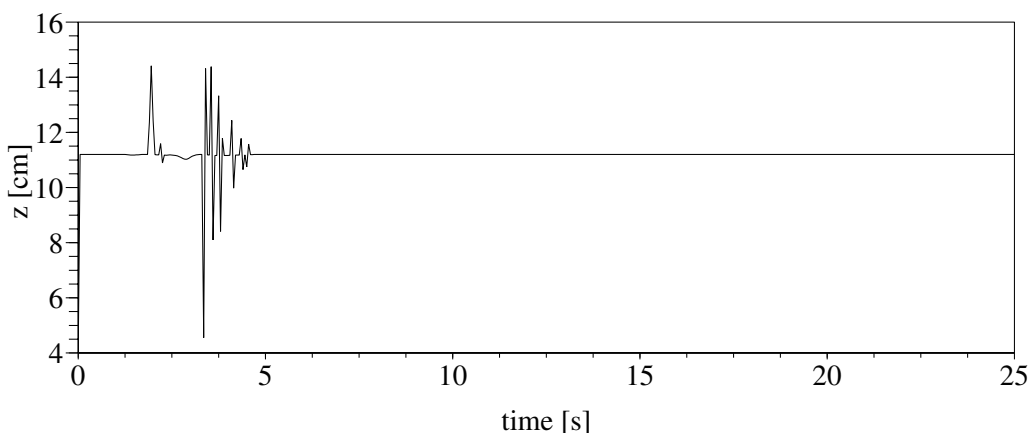


図 7.19 二輪型倒立振子の重心位置の z 軸上に於ける時間変化

式 (7.3.3) に, 重心位置については式 (7.3.4) 従い算出している.

$$x = 100 \cdot R \cdot \int \dot{\theta}(t) dt \quad [\text{cm}] \quad (7.3.3)$$

$$z_m = 100 \cdot \left[R + R \sin \gamma \cdot \int \dot{\theta}(t) dt + L \cos \left\{ \int \dot{\psi}(t) dt \right\} \right] \quad [\text{cm}] \quad (7.3.4)$$

先ず, 図 7.18 に示した移動距離に着目する. 図 7.11 で与えられた指令入力によって, 二輪型倒立振子は一旦後方へ移動した後, 大きく前方へ移動している事が確認出来る. この指令入力によって, 二輪型倒立振子は起伏のある路面に接近している事が判る. その後, 再び平坦な路面へ移動し, 以降は起伏のある路面と平坦な路面を往復する様になっている. この結果より, NXTway-GS は予測結果を用いて自律的に移動し, 自身のピッチ角を鉛直近傍で維持する様, 路面に合わせて行動していると言える. これに関係する結果として, 図 7.19 を見ると, 指令入力を与えられた時点では重心位置が大きく変動しているが, これ以降は大きく重心位置は変動していない. この事実より, 提案手法では, 予測結果を用いて自律的に路面を移動し, それ自身が路面状況に応じバランスを取れていると言える. これらの結果より, 提案手法によって, ロボットは所望の安定状態を維持することから, 提案手法の有用性を確認出来たものと考えられる.

7.4 外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合

本節では, 二輪型倒立振子に提案手法を適用し, 倒立姿勢維持タスクを実行させる. この時人間が外部から予測可能な外乱を継続的に与え続けた場合の動作をシミュレーションによって観察し, その結果について検討する.

7.4.1 事前実験

7.4.1.1 システム構築

状態行動対予測を実施し行動補正を行う上で必要となる学習データを取得する為、事前に実施するシミュレーションについて説明する。本研究では NXTway-GS を、二輪型倒立振子としてモデリングしている (第 4 章参照)。NXTway-GS を側面図 (図 4.6(a)) と平面図 (図 4.6(b)) に分け、Lagrange 方程式 [81] を基に運動方程式を考える [78][79]。同図で現れる主要な変数は表 4.3 の通りである。

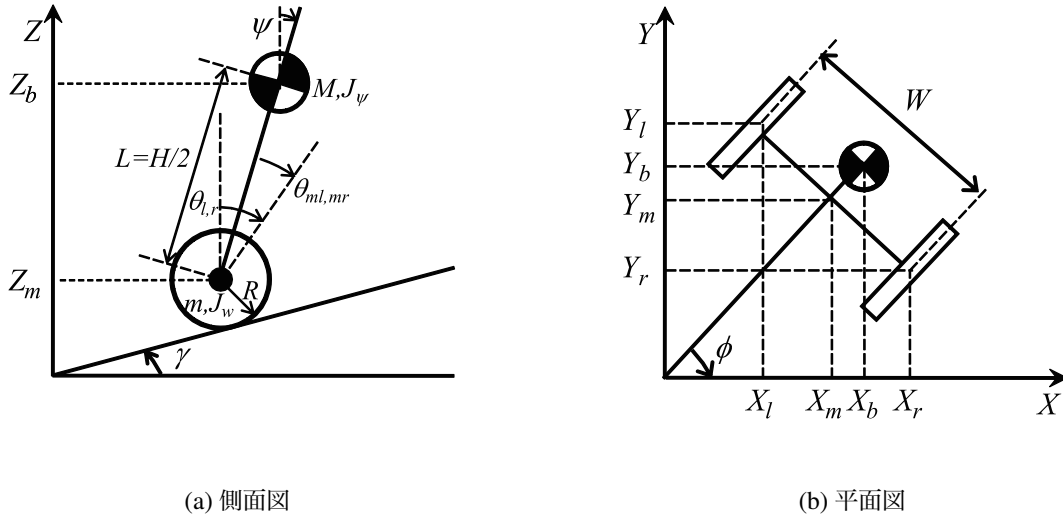


図 7.20 NXTway-GS の側面・平面図 (図 4.6 の再掲)

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

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

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

この運動方程式を $\mathbf{x}_1 = \begin{bmatrix} \theta & \psi & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$, $\mathbf{u} = \begin{bmatrix} v_l & v_r \end{bmatrix}^T$, $y = \theta$ とする事で、

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

第 7 章 実験 III: 倒立振り子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を 行動決定に反映した場合-

を得ている. この状態方程式と, 最適レギュレータ法を適用する事で算出したフィードバックゲインを配置した状態フィードバックにより, 制御器を構成した [78][79]. この様にして得られる事前実験の概要が図 7.21 である.

7.4.1.2 学習データの取得

事前実験に於いては, +1 を前進, 0 を静止, -1 を後退とし, 人間が外部から指令入力として予測可能な外乱を与える (図 7.22). 尚, 実験条件は表 7.7 に従う. この事前実験より学習データが得られる.

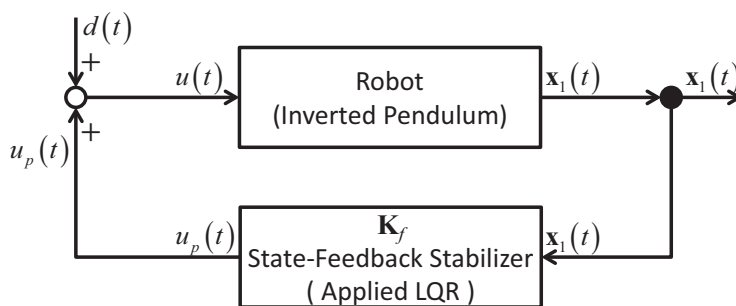


図 7.21 事前実験の為のシステム構成 (5)

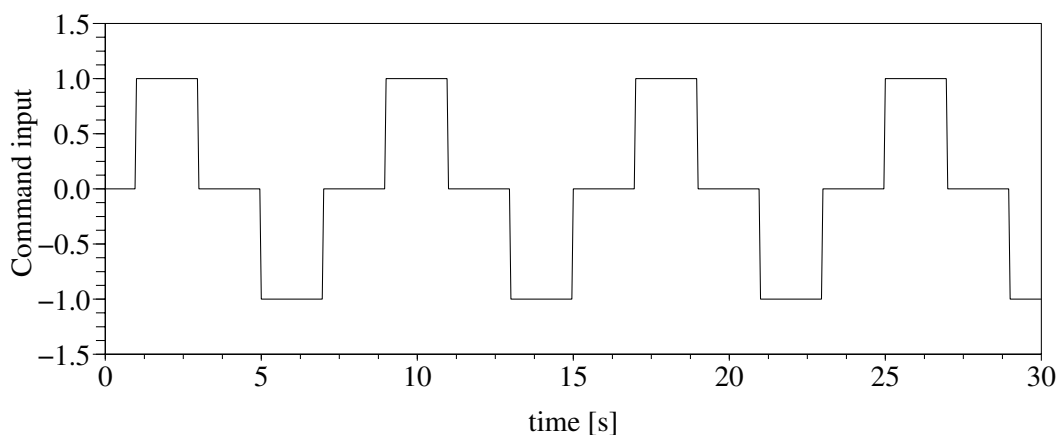


図 7.22 人間からモデルへ与える指令入力 (2)

7.4.2 検証実験

7.4.2.1 実験設定

検証実験に於いては, 学習器として, RBF カーネルを組込んだ Online SVR を **State Predictor** に, LQR を **Action Predictor** に用いている. 学習データについては, 前述の事前実験で得た制

7.4 外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合

表 7.7 事前実験に係わる実験パラメータ (5)

定数	数値	単位	意味
ψ_0	0.0262	[rad.]	初期ピッチ角度
γ_0	0.0	[rad.]	進行方向傾斜路傾斜角
t_s	0.05	[s]	サンプリングレート
$t_{d,start}$	1.0	[s]	予測可能な外乱を与え始める時間
$t_{d,finish}$	30.0	[s]	予測可能な外乱を与え終わる時間

御応答と行動の数値データ (図 7.13 から図 7.17 に於ける黒色実線で示される曲線) を採用する。学習器の学習パラメータを表 7.8 に、検証実験に於けるその他の実験パラメータを表 7.9 に示す。また、状態行動対予測に於ける **State Predictor** と **Action Predictor**, 及び行動補正につ

表 7.8 検証実験時の Online SVR 及び LQR に於けるパラメータ (4)

定数	数値	意味
C_1	300	x_1 に対する正則化パラメータ
ϵ_1	0.02	x_1 に対する ϵ 不感帯パラメータ
β_1	30	x_1 に対する RBF カーネルパラメータ
C_2	300	x_2 に対する正則化パラメータ
ϵ_2	0.02	x_2 に対する ϵ 不感帯パラメータ
β_2	30	x_2 に対する RBF カーネルパラメータ
C_3	300	x_3 に対する正則化パラメータ
ϵ_3	0.02	x_3 に対する ϵ 不感帯パラメータ
β_3	30	x_3 に対する RBF カーネルパラメータ
C_4	300	x_4 に対する正則化パラメータ
ϵ_4	0.02	x_4 に対する ϵ 不感帯パラメータ
β_4	30	x_4 に対する RBF カーネルパラメータ
\mathbf{k}_f	$\begin{bmatrix} -0.870 \\ -32.2 \\ -1.16 \\ -2.81 \end{bmatrix}$	u に対する状態フィードバックゲイン

表 7.9 検証実験に係わる実験パラメータ (3)

定数	数値	単位	意味
t_s	0.05	[s]	サンプリングレート
N_s	60	—	初期の一括学習に係る初期データセット長
N_{max}	241	—	予測に用いる最大データセット長
N	20	—	行動補正で用いる予測値のステップ数
C_σ	0.05	—	標準偏差から重みを求める際の係数
N_σ	10	—	標準偏差を求める際の予測値のステップ数

いては, 次式を適用する.

$$\hat{x}_{1,i}(t+1) = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}(x\mathbf{z}(t))^\top (\mathbf{K}_{sv} + \lambda\mathbf{I}_l)^{-1} x\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (3.4.1)$$

when $i \in \dim \mathbf{x}_1(t)$

$$\hat{u}(t+1) = \mathbf{k}_f \hat{\mathbf{x}}_1(t+1) \quad (4.1.1)$$

$$\alpha_j = C_\sigma \cdot \sigma_j \quad (3.5.4)$$

$$\sigma_j = \sigma \left[{}^{t_1-M}\hat{\mathbf{u}}(t_1+j), {}^{t_1-M+1}\hat{\mathbf{u}}(t_1+j), \dots, {}^{t_1-i}\hat{\mathbf{u}}(t_1+j), \dots, {}^{t_1+1}\hat{\mathbf{u}}(t_1+j) \right] \quad (3.5.5)$$

$$\mathbf{u}_s^{(t+1, t+N)}(t) = \sum_{i=1}^N \alpha_i \hat{\mathbf{u}}(t+i) \quad (3.5.6)$$

$$\mathbf{u}_p(t) = \mathbf{k}_f \cdot \mathbf{x}_1(t) \quad (7.4.1)$$

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1, t+N)}(t) + \mathbf{d}(t) \quad (3.5.7)$$

7.4.2.2 実験結果とその検討

図 7.23 から 7.26 は状態 \mathbf{x}_1 に対する補正結果を, 図 7.27 は行動 u に対する補正結果を示している. ここで改めて確認すると, 実際に予測結果を用い制御を行っている部分は, $T = 3.00[\text{s}]$ 以降である. この点を踏まえ, それぞれの補正結果に於いて時刻 T 以降の変動について議論してゆく事とする.

先ず図 7.24 に注目すると, 提案手法は LQR のみに比べピッチ角が大きく下回っている事が判る. 次に図 7.23 に着目すると, 車輪回転角 θ が大きく振れている部分が確認出来る. これより, 外乱に対して本体を大きく前後に移動する事により, ピッチ角の変動を僅かに留めようとしている事が判る. 一方で図 7.25 及び図 7.26 に着目すると, LQR のみによる従来的手段と比べ,

7.4 外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合

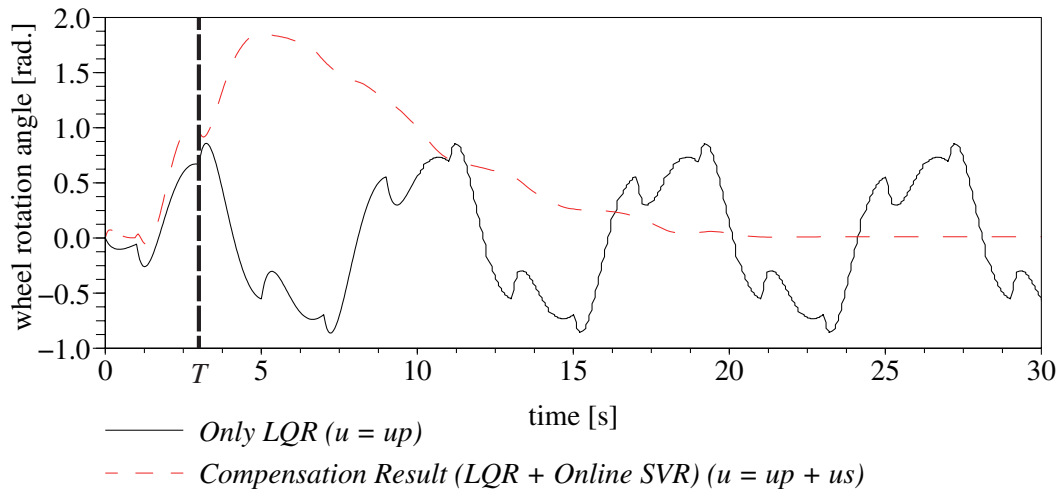


図 7.23 状態 $x_{1,1}$ (車輪回転角度 θ) に対する補正結果 (4)

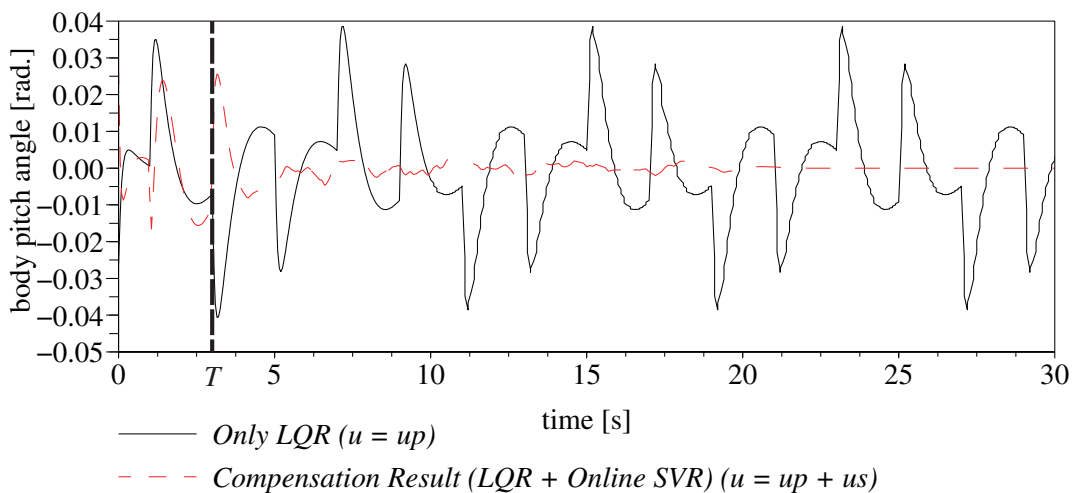


図 7.24 状態 $x_{1,2}$ (ピッチ角度 ψ) に対する補正結果 (4)

速度的な変化は時間経過に従い殆ど見られない傾向を示している。図 7.27 については、予測結果を用いて行動を補正している事から、LQR のみに比べ採られる行動が時間的に『先取り』されている事が判る。更に図 7.25, 図 7.26 に着目すると、外乱に先んじて角速度が変動している事が判る。この結果から、指令入力によって移動するが、振子のピッチ角を鉛直近傍に抑えている事が確認出来る。

続いて、図 7.28 に着目したい。このグラフでは、指令入力によって得た移動距離を、提案手法、LQR のそれぞれを適用した場合で比較したものとなっている。この移動距離については、次式に従い算出している。

$$x = 100 \cdot R \cdot \int \dot{\theta}(t) dt \quad [\text{cm}] \quad (7.3.3)$$

第7章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を行動決定に反映した場合-

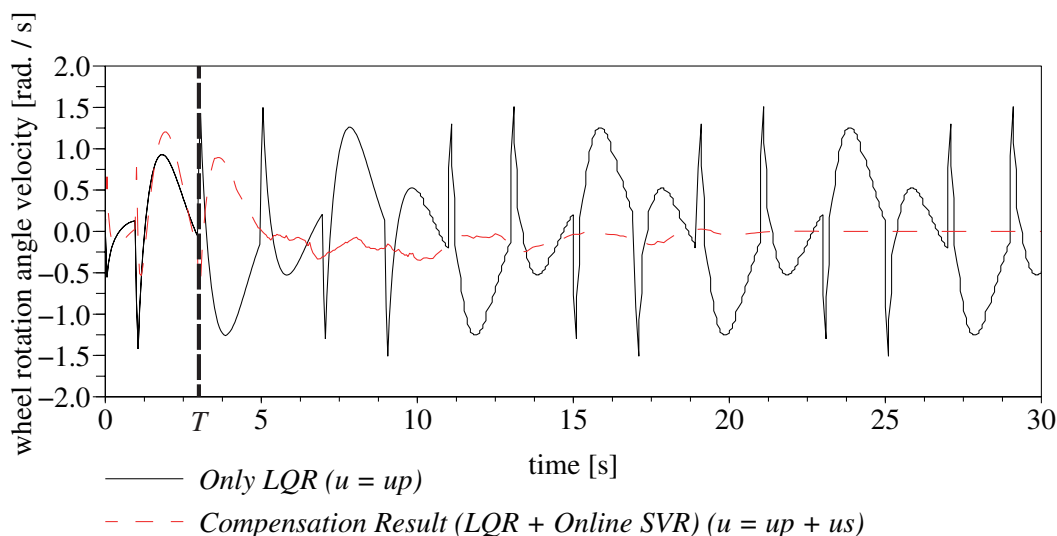


図 7.25 状態 $x_{1,3}$ (車輪回転角速度 $\dot{\theta}$) に対する補正結果 (4)

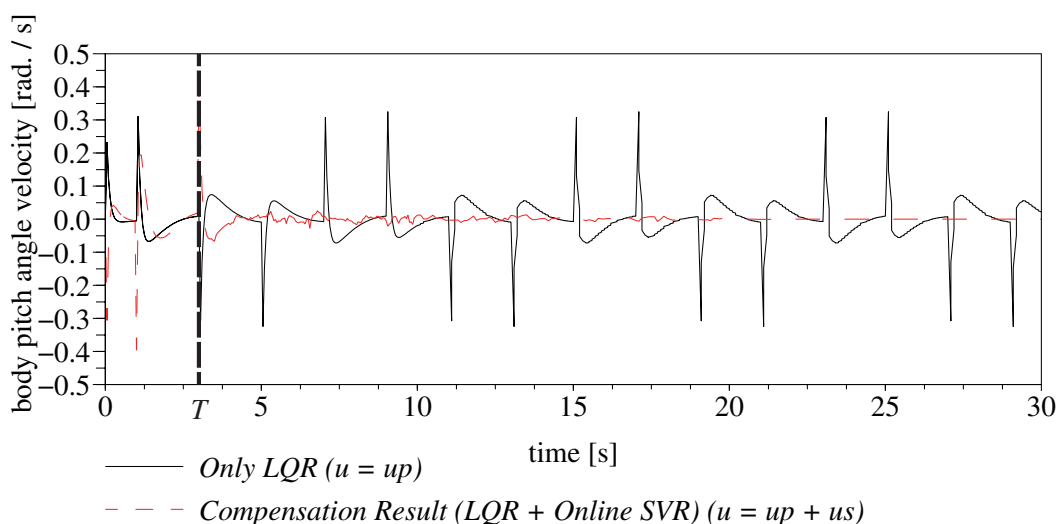


図 7.26 状態 $x_{1,4}$ (ピッチ角速度 $\dot{\psi}$) に対する補正結果 (4)

各々の結果を見ると、指令入力に従って前進或いは後退している事が確認出来る。また距離について見ると、LQR のみの場合に比べて提案手法の方が長く移動している事が判る。これは、前後移動から静止にかけてのプロセスに於いて、LQR のみの場合はバランスをとる為に前後への移動を必ず採らなければならない、その分の移動距離を相殺してしまっているからであると考えられる。一方で提案手法は 15[s] を過ぎたあたりから、ほとんど前後移動をしていない事も確認出来る。これは、外乱に対して目標軌道である $\psi(t) = 0$ 近傍を維持しようと提案手法が最適な行動を決定した為であると考えられる。但し、この点を踏まえても、提案手法により外乱が与えられた場合でもピッチ角は安定的に低減出来ている事から、提案手法の有用性を確認出来たもの

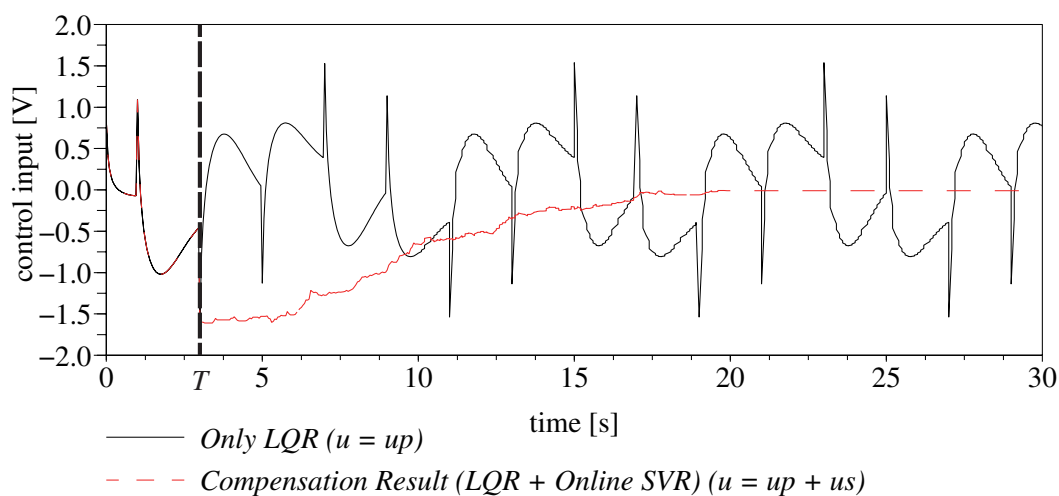


図 7.27 行動 u (コントローラからの最適入力) に対する補正結果 (4)

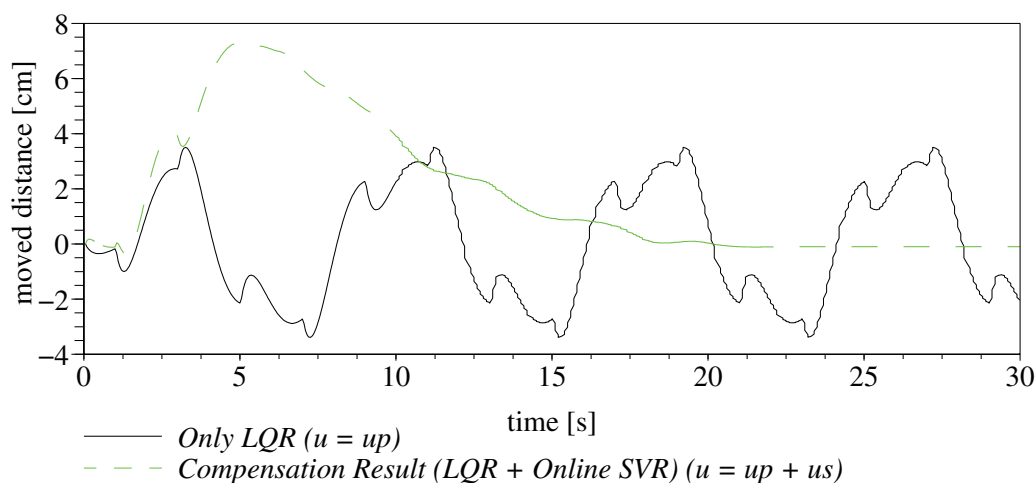


図 7.28 LQR のみの場合との移動距離の差異

と結論付ける。

7.5 まとめ

本章では、状態行動対予測を基にした行動補正において、予測結果の統計的性質によって動的に重みを変化させる方法について、計算機シミュレーションによる検証を試みた。検証結果より、次の事項を確認した。

- (i) アクチュエータへ予測可能な外乱を持続的に与え続けた場合、姿勢安定の状態を保ちつつピッチ角が安定的に低減出来る事を確認した。

第7章 実験 III: 倒立振子に対する行動補正 -統計的係数による信頼度付き状態行動対予測結果を行動決定に反映した場合-

- (ii) 外部コマンドとして前進後退指令を与え路面の形状を変化させた場合, 予測結果に基づいて自律的に移動し, 姿勢安定の状態を保ちピッチ角が安定する行動を採る事を確認した.
- (iii) 外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合, 姿勢安定の状態を保ちつつピッチ角が安定的に収束する事を確認した.

また, 固定の重み係数による行動補正に比べ回転角数やピッチ角の触れ幅, 収束度合いが優れている事も確認出来た. 従って, 予測結果の統計的性質によって重みを動的に変化させ行動を補正する事で, 対象の状態を素早く安定化するような行動を決定出来るものと結論付ける.

第 8 章 結論

本論文では, ロボットが持つ状態と, それ自身が採る行動により, これらが緊密に関係する事で所望のタスクを達成する事を示した. この考え方の下, これら 2 つの関係性に着目する事で, 将来の状態と行動を予測する状態行動対予測を提案した. また, 状態行動対予測によって予測した未来の行動を, 現在採る行動に反映させる事で, 未来を考慮して最適な行動に補正できる手法を提案した. 本章では, 本研究の結果を総括すると共に, 将来展望について述べる.

本研究では, ロボットが持つ状態と採る行動との関係性に注目し, この 2 つを関連付ける必要性を論じると共に, これを対とする事で未来の状態と行動を予測出来る事を示した. また, 従来の最適制御と提案手法を組み合わせることで, 未来を考慮した最適行動を決定出来る事を論じ, 検証実験でその有用性を確認した.

この課題を達成する為に, 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, 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状態予測を定義した.
- (5) 状態予測を拡張し, 状態行動対を定義する事で, これの未来の挙動を予測する状態行動対予測を提案した.
- (6) 状態行動対予測で得た予測結果を基に, 従来の最適制御と組み合わせる事で, 未来を考慮した最適行動を決定する行動補正を提案した.
- (7) 行動補正の際に生じる可能性のある予測誤差について, 得られた予測値に標準偏差を基にした重み付けを行い動的に変化させる事により, この影響を低減する方法を提案した.

本研究では、自走式二輪型倒立振子を適用例として採用した上でモデリングした。このモデルに対して状態行動対予測を適用する事で未来の状態と行動を予測出来るか、シミュレーションによって検討した。また、この状態行動対予測を用いた行動補正を同じく二輪型倒立振子に適用する事で、未来の状態と行動を基に現在の行動が補正出来る事をシミュレーションによって確認した。これらを実施する為に、各々の提案手法を、制御対象となる自走式二輪型倒立振子に適用し、次の様な状況を想定しシミュレーションした。

- (1) アクチュエータへ予測可能な外乱を持続的に与え続けた場合の未来の状態と行動を、状態行動対予測によって予測する。
- (2) アクチュエータへ予測可能な外乱を持続的に与え続けた場合の挙動を、状態行動対予測と行動補正を組み合わせ安定化させる。
- (3) 外部コマンドとして前進後退指令を与え路面の形状を変化させた場合の挙動を、状態行動対予測と行動補正を組み合わせ安定化させる。
- (4) 外部コマンドとして予測出来る程度の『周期的な外乱』を与えた場合の挙動を、状態行動対予測と行動補正を組み合わせ安定化させる。

この結果として、状態行動対予測は学習データに対しその傾向を示す結果を得た。また、状態行動対予測と行動補正を組み合わせた場合、外乱の傾向を状態と行動の関係から直接的に学習する事で、経時的に速やかに安定する様な行動を決定し続けた。また外部コマンドに由って環境が変化する場合も、状態と行動の関係から採るべき行動を先取りする事で、環境の変化に応じて安定化する様な行動を決定し続ける事を確認した。

従って、状態と行動を対として組み合わせる事によって、未来に於ける状態と行動を正確に予測出来ると共に、外乱や環境の変化にロバストな行動を採るよう補正出来る事を確認した。この事より、本論文が主目標としていた状態行動対予測と行動補正の有用性を確認したものと結論付ける。

本研究では、状態と行動の組み合わせで動作を表現できるロボットを対象として状態行動対予測の考え方を提案した。この方法では、ハードウェア的制約等により、計算ステップの遅延により行動が数ステップ遅れて決定される事を考慮に入れている。つまり、状態行動対予測をする事でロボットの行動決定の遅延にも対応出来るものと考えられる。従って、ロボットの自律的行動決定のみならず、操作者と遠隔した地に存在するロボットを操作する様な状況下に於いても、命令遅延の影響を低減させる補助として応用出来るものと考えられる。

その一方で課題も残されている。本研究で提案した状態行動対予測はその仕組みから、時間

の経過に従って予測誤差が累積する。このため、予測を開始した時刻から離れた時刻に着目した場合、その時の予測値が実際に与えられる値と異なる事が考えられる。この為、状態行動対予測によって得た結果の内、行動補正などで用いる事が出来る予測値の範囲は限られる事となる。この点を考慮した上で、今後は実用上用いる事が出来る予測値の範囲を理論的に示す事、また用いる事が出来る予測値の範囲を拡大する事を検討する必要がある。

また、本研究に於いては、未来予測に於いて必要となる各種パラメータを一意に決定して検証実験を繰り返した。仮に実環境での実装と動作に至った場合、計算機シミュレーションとは異なる要因が複数存在する為、適切なパラメータを特定する事は困難である。そこで今後は、状態行動対予測の途中で学習パラメータを最適化する様な仕組みを定式化する必要があるものと考えられる。また、行動補正に於いても同様の問題点が指摘出来る。重み係数に関しては動的な決定を実現しているものの、行動補正に用いる為の予測値列の使用データ長を固定していた。この点についても、状態と行動の経時的変化や、状態行動対予測の予測結果から自律的に最適化出来る様な仕組みを定式化する必要があるものとする。

謝辞

本論文は、筆者が室蘭工業大学大学院工学研究科の博士後期課程 (生産情報システム工学専攻) に於いて遂行した研究成果をまとめたものです。

この間、本研究を実施する機会を与えて戴き、公私にわたり暖かいご助言を賜り、そして本研究の推進と各種論文、本論文の執筆に終始熱心なご指導を頂きました室蘭工業大学准教授 倉重健太郎先生に、衷心より、御礼申し上げます。

また、研究及びプレゼンテーションでのご助言を頂いた畑中 雅彦教授、佐賀 聡人教授、本田 泰准教授、鈴木 幸司教授、渡邊 真也准教授、工藤 康生准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。

最後に、私の進路を支持しここまで支えてくれた母と今は亡き祖母に、心より感謝の意を表します。

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] 杉浦 孔明, 岩橋 直人: 参照点に依存した確率モデルを用いた連続動作の認識と生成, 人工知能学会全国大会論文集 22 回, 3E3-6, pp.1-4 (2008)
- [5] Denis F. Wolf and Gaurav S. Sukhatme. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. *Autonomous Robots* 19:53-65, Springer, Netherlands, 2005.
- [6] Dieter Fox, Wolfram Burgard and Sebastian Thrun. Markov Localization for Mobile Robots in Dynamic Environments. *Journal of Artificial Intelligence Research* 11:391-427, 1999.
- [7] Mohammad Abdel Kareem Jaradata, Mohammad Al-Rousanb and Lara Quadanb. Reinforcement based Mobile Robot Navigation in Dynamic Environment. *Robotics and Computer-Integrated Manufacturing* 27:135-149, 2011.
- [8] Ellips Masehian and Yalda Katebi. Sensor-Based Motion Planning of Wheeled Mobile Robots in Unknown Dynamic Environments. *Journal of Intelligent & Robotic Systems*, DOI:10.1007/s10846-013-9837-3, 2013.
- [9] Mohammed Faisal, Ramdane Hedjar, Mansour Al Sulaiman and Khalid Al-Mutib. Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment. *International Journal of Advanced Robotic Systems*, DOI: 10.5772/54427, 2012.
- [10] Fabrizio Abrate, Basilio Bona, Marina Indri, Stefano Rosa and Federico Tibaldi. Multi-robot Map Updating in Dynamic Environments. *Distributed Autonomous Robotic Systems*

- Springer Tracts in Advanced Robotics 83:147-160, 2013.
- [11] 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)
- [12] 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)
- [13] Jong Jin Park, Collin Johnson, and Benjamin Kuipers, “Robot Navigation with MPEPC in Dynamic and Uncertain Environments: From Theory to Practice,” IROS 2012 Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs, 2012.
- [14] 湯田陽子, 有我祐一, 渡部慶二, 村松鋭一, 遠藤茂：生物の反射モデルのロボット制御への応用, 計測自動制御学会東北支部第 245 回研究集会, 資料番号 245-27 (2008)
- [15] 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)
- [16] Thomas Mitchell. Machine Learning, The McGraw-Hill Companies, 1997.
- [17] V. N. Vapnik. The Nature of Statistical Learning Theory, Springer, New York, 1995.
- [18] 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)
- [19] 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.
- [20] Teuvo Kohonen. Self-Organizing Maps, Springer, 2000.
- [21] R. S. Sutton, A. G. Barto. Reinforcement Learning, The MIT Press, 1998.
- [22] 山田誠二, 室田盛道：自己組織化マップを用いた移動ロボットによる 行為系列からの環境認識, 日本ロボット学会誌 Vol.17 No.6, pp.855-864 (1999)
- [23] 井上晃, 井上健司, 大川善邦：複数移動障害物の行動予測に基づく 自律移動ロボットのオンライン回避行動計画, 日本ロボット学会誌 Vol.15 No.2, pp.249-260 (1997)
- [24] 前田 雄介, 原 崇之, 新井 民夫：躍度最小モデルを用いた動作予測に基づく 人間-ロボット協調作業, 日本機械学会論文集. C 編 68(675), pp.3367-3372 (2002)
- [25] Amalia F. Foka and Panos E. Trahanias. Predictive Control of Robot Velocity to Avoid Obstacles in Dynamic Environments. In Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, 1:370-375, 2003.

-
- [26] Edvard Naerum, H. Hawkeye King, and Blake Hannaford, “Robustness of the Unscented Kalman Filter for State and Parameter Estimation in an Elastic Transmission,” In *Proceedings of the Robotics: Science and Systems*, 2009.
- [27] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda, “Purposive Behavior Acquisition for a Robot by Vision-Based Reinforcement Learning,” *Journal of the Robotics Society of Japan*, vol.13, no.1, pp.68-74, 1995.
- [28] 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)
- [29] 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)
- [30] 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.
- [31] 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.
- [32] 田崎勇一, 井村順一: 平面 2 足歩行ロボットのグラフ型モデル予測制御, *日本ロボット学会誌* Vol.24 No.5, pp.663-671 (2006)
- [33] Erik Schuitema, Lucian Busoniu, Robert Babuska, and Pieter Jonker, “Control Delay in Reinforcement Learning for Real-Time Dynamic systems: A Memoryless Approach,” In *Proceedings of Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference, pp.3226-3231, IEEE, 2010.
- [34] Thomas J. Walsh, Ali Nouri, Lihong Li, and Micheal L. Littman, “Planning and Learning in Environments with Delayed Feedback,” *Machine Learning: ECML 2007*, pp.442-453, 2007.
- [35] Yang Su, Kok Kiong Tan, and Tong Heng Lee, “Computation Delay Compensation for Real Time Implementation of Robust Model Predictive Control,” In *Proceedings of Industrial Informatics (INDIN)*, 2012 10th IEEE International Conference, pp.242-247, IEEE, 2012.
- [36] Cunjia Liu, Wen-Hua Chen, and John Andrews, “Model Predictive Control for Autonomous Helicopters with Computational Delay,” In *Proceedings of Control 2010, UKACC International Conference*, pp.1-6, IET, 2010.
- [37] Giancarlo Marafioti, Sorin Olaru, and Morten Hovd, “State Estimation in Nonlinear Model

- Predictive Control, Unscented Kalman Filter Advantages,” *Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences* Vol.384, pp.305-313, 2009.
- [38] Ramazan Havangi, Mohammad Ali Nekoui, and Mohammad Teshnehlab, “Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization,” *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 2, No 2, pp.15-23, 2010.
- [39] Niko Sünderhauf, Sven Lange, and Peter Protzel, “Using the Unscented Kalman Filter in Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control,” In *Proceedings of IEEE International Workshop on SSRR 2007*, pp.1-6, 2007.
- [40] Mohammad Ali Badamchizadeh, Iraj Hassanzadeh, and Mehdi Abedinpour Fallah, “Extended and Unscented Kalman Filtering Applied to a Flexible-Joint Robot with Jerk Estimation,” *Discrete Dynamics in Nature and Society*, Vol. 2010, Article ID 482972, 2010.
- [41] J. G. Iossaqui, J. F. Camino, and D. E. Zampieri, “Slip Estimation Using The Unscented Kalman Filter for The Tracking Control of Mobile Robots,” In *Proceeding of the International Congress of Mechanical Engineering - COBEM*, pp.1-10, 2011.
- [42] 山北 昌毅 : UKF (Unscented Kalman Filter) って何？, システム制御情報学会誌 Vol.50, No.7, pp.261-266 (2006)
- [43] 尾形和哉, 橋口宏衛 : 混合正規分布モデルを用いた 自律移動型ロボットの自己位置推定問題の検討, 日本機械学会論文集 (C 編) Vol.79 No.807, pp.248-261 (2013)
- [44] Sylvain Calinon, Danilo Bruno, Darwin G. Caldwell, “A Task-Parameterized Probabilistic Model with Minimal Intervention Control,” *ICRA*, pp.3339-3344, 2014.
- [45] Corinna Cortes and V. N. Vapnik, “Support-Vector Networks,” *Machine Learning*, 20, 1995.
- [46] Francesco Parrella. *Online Support Vector Regression*, A Thesis presented for the degree of Information Science, Department of Information Science, University of Genoa, Italy, 2007.
- [47] Lihua Jiang, Mingcong Deng, A. Inoue. *SVR based Obstacle Avoidance and Control of a Two Wheeled Mobile Robot*. In *Proceeding of Innovative Computing, Information and Control*, 2007. ICICIC '07. Second International Conference, Okayama, DOI: 10.1109/ICICIC.2007.553, 2007.
- [48] Zhijun Li, Kun Yang and Yipeng Yang, “Support Vector Machine based Optimal Control for Mobile Wheeled Inverted Pendulums with Dynamics Uncertainties,” In *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009 held jointly with the 2009 28th Chinese Control Conference, 2009.

-
- [49] Foudil Abdessemed, "SVM-Based Control System for a Robot Manipulator," *Int. J. Adv. Robotic Sy*, 2012, vol. 9, 247, DOI: 10.5772/51192, 2012.
- [50] Yi Wan, Hailong Huang, Jing Tian, "Dynamic Self-Adaptive Reliability Control for Electric-Hydraulic Systems," *Advances in Mechanical Engineering*, Vol.7, No.2, 767401, 2014.
- [51] J. B. Gao, S. R. Gunn, and C. J. Harris, "Mean field method for the support vector machine regression," *Neurocomputing*, Vol.50, pp.391- 405, 2003.
- [52] Muhsin Hassan, Dino Isa, Rajprasad Rajkumar, Nik Ahmad Akram, Roselina Arelhi, "Reducing Support Vector Machine Classification Error by Implementing Kalman Filter," *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.5, No.9, pp.10-18, 2013.
- [53] 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.
- [54] 植田英三郎 : 時系列解析による株価予測, 大阪府立大學經濟研究 53(3), pp.95-111 (2007)
- [55] Eisaku Maeda and Hiroshi Murase, "Kernel-Based Nonlinear Subspace Method for Pattern Recognition," *Systems and Computers in Japan*, Vol. 33, Issue 1, pp.38-52, 2002.
- [56] 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.
- [57] 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.
- [58] Adriaan Zaanen. *Linear Analysis*, North Holland Publishing Co. , 1960.
- [59] Masashi Sugiyama, Hirotaka Hachiya, Christopher Towell, Sethu Vijayakumar, "Geodesic Gaussian Kernels for Value Function Approximation," *Autonomous Robots*, Vol.25 No.3, pp.287-304, 2008.
- [60] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Machine Learning Research*, vol.11, pp.1471-1490, 2010.
- [61] L. Valiant. A theory of the learnable, *Communications of the ACM*, 27, 1984.
- [62] Huber, Peter J. : Robust Estimation of a Location Parameter, *Annals of Statistics* 53, pp.73-101, 1964.

参考文献

- [63] 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)
- [64] H. W. Kuhn, A. W. Tucker, “Nonlinear programming,” Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, pp.481-492, 1951.
- [65] 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)
- [66] Brian D.O. Anderson, John B. Moore. Optimal Control: Linear Quadratic Methods. Englewood Cliffs, NJ: Prentice Hall, ISBN 978-0-13-638560-8, 1990.
- [67] Stanley G.M., Mah R.S.H. , “Observability and Redundancy Classification in Process Networks,” Chem. Engng. Sci. 36, pp.1941-1954, 1981.
- [68] 安定の定義 <http://ysserve.int-univ.com/Lecture/ControlMecha2/node18.html>
- [69] H. K. Khalil. Nonlinear systems, Prentice Hall Upper Saddle River, NJ, 1996.
- [70] 最適レギュレータの説明 <http://www.me.saga-u.ac.jp/~sato/regulator.pdf>
- [71] Christopher M. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.
- [72] B. S. Everitt. Cambridge Dictionary of Statistics (2nd edition), Cambridge, UK: Cambridge University Press., ISBN 0-521-81099-X, 2002.
- [73] 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.
- [74] F. Girosi, “An equivalence between sparse approximation and Support Vector Machines,” Neural Computation, 10(6):1455-1480, 1998.
- [75] 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, NJ, USA c2009, table of contents ISBN: 978-1-4244-4523-3, 2009.
- [76] John Devcic, “Weighted Moving Averages: The Basics,” 2006. [Online]. Available: <http://www.investopedia.com/articles/technical/060401.asp>
- [77] Y.Yamamoto. NXTway-GS Model-Based Design -Control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT-, CYBERNET SYSTEMS CO.,LTD. , 2008.

-
- [78] 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)
- [79] 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)
- [80] LEGO Mindstorms NXT http://web.mac.com/ryo_watanabe/iWeb/Ryo%27s%20Holiday/LEGO%20Mindstorms%20NXT.html
- [81] R. Courant, D. Hilbert. Methoden der Mathematischen Physik, Berlin, Springer, 1931.
- [82] Kenji Fukumizu, “Dimensionality Reduction in Regression with Positive Definite Kernels,” In Proceedings of the Institute of Statistical Mathematics Vol.53, No.2, pp.189-200, 2005.

研究業績

査読付き投稿論文

- [1] Masashi Sugimoto, Kentarou Kurashige, “A Study of Effective Prediction Methods of the State-action Pair for Robot Control using Online SVR,” *Journal of Robotics and Mechatronics*, Vol.27, No.5, pp.469-479, 2015.
- [2] Masashi Sugimoto, Kentarou Kurashige, “Future Motion Decisions using State-action Pair Predictions,” *International Journal of New Computer Architectures and Their Applications (IJNCAA)*, Vol.5, NO.2, pp.79-93, DOI: 10.17781/P001896, 2015.

査読付き国際会議プロシーディングス

- [1] Masashi Sugimoto, Kentarou Kurashige, “The Proposal for Prediction of Internal Robot State Based on Internal State and Action,” In proceedings of IWACHII2013 CD-ROM, SS1-2, 2013.
- [2] Masashi Sugimoto, Kentarou Kurashige, “The Proposal for Deciding Effective Action using Prediction of Internal Robot State Based on Internal State and Action,” In proceedings of 2013 International Symposium on Micro-NanoMechatronics and Human Science, pp.221-226, 2013.
- [3] Masashi Sugimoto, Kentarou Kurashige, “Real-time Sequentially Decision for Optimal Action using Prediction of the State-Action Pair,” In proceedings of 2014 International Symposium on Micro-NanoMechatronics and Human Science, pp.199-204, 2014.
- [4] Masashi Sugimoto, Kentarou Kurashige, “The Proposal for Real-time Sequential-decision for Optimal Action using Flexible-weight Coefficient based on the State-Action Pair,” 2015 In proceedings of IEEE Congress on Evolutionary Computation (CEC2015), pp.544-551, 2015.
- [5] Masashi Sugimoto, Kentarou Kurashige, “The Proposal for Compensation to the Action of Motion Control based on the Prediction of State-action Pair”, In proceedings of the International Conference on Electronics and Software Science, pp.35-48, 2015.

査読無し講演論文

- [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, pp.59-60 (2011)
- [2] 杉本 大志, 倉重 健太郎: “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”, 第 31 回日本ロボット学会学術講演会, RSJ2013AC3I2-04 (2013)
- [3] 杉本 大志, 倉重 健太郎: “状態行動対予測を用いた ロボット制御における最適な未来行動の決定”, 第 33 回日本ロボット学会学術講演会, RSJ2015AC3C3-02 (2015)
- [4] 杉本 大志, 倉重 健太郎: “ロボット制御における 状態行動対予測を用いた未来行動決定の検討”, システム・情報部門学術講演会 2015, pp.1181-1185 (2015)