

論文 / 著書情報
Article / Book Information

題目(和文)	Koopmanアプローチを活用したモデル予測制御に関する研究
Title(English)	
著者(和文)	金井政樹
Author(English)	Masaki Kanai
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12363号, 授与年月日:2023年3月26日, 学位の種別:課程博士, 審査員:山北 昌毅,三平 満司,,中島 求,塚越 秀行,早川 朋久
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12363号, Conferred date:2023/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis



Tokyo Institute of Technology

令和 4 年度 博士論文

Koopman アプローチを活用した
モデル予測制御に関する研究

Study of Model Predictive Control
Utilizing Koopman Approach

2023 年 3 月

東京工業大学
工学院 システム制御系
金井 政樹
指導教員 山北 昌毅 准教授

Abstract

Model predictive control (MPC) is a versatile optimal control method that can determine the optimal input by explicitly considering the nonlinearities and constraints of the target system. In MPC, the accuracy of the prediction model has a significant impact on control performance. On the other hand, if a complex prediction model with nonlinearity is used, the computational cost of the optimization is increased at each control cycle. Therefore, the challenge is to improve prediction accuracy by taking nonlinearities into account while reducing the computational cost for real-time executions.

On the other hand, MPC methods utilizing lifting linearization based on Koopman's theory have been attracting attention in recent years. Lifting linearization is a method that lifts the state to a high-dimensional space using observation functions (observables), which are nonlinear outputs of the target system, and expresses the time evolution of the observables as linear dynamics. This allows us to obtain a high-dimensional linear model (lifted linear model) that expresses the nonlinear characteristics of the target system. By using this, it is possible to construct a linear model predictive control (LMPC) and is expected to significantly reduce the computational cost.

However, an accurate lifted linear model for a nonlinear system generally tends to have very large orders. This results in a relatively high computational cost, even for LMPC. On the other hand, when lifting linearization is performed in low dimensions using a limited number of observables, the prediction accuracy is generally deteriorated. For these reasons, even if utilizing the lifting linearization, there remains a challenge in achieving both improved prediction accuracy and reduced computational cost. Therefore, the objective of this research is to propose MPC methods that combines high control performance based on accurate prediction with low computational cost by utilizing the Koopman approach.

The lifted linear model obtained based on Koopman theory is generally an approximate model of the nonlinear system. Therefore, regardless of the order of the model, prediction errors are inevitable. Therefore, this study takes the approach of constructing an LMPC with prediction error correction for a low order lifted linear model to achieve both high control performance based on accurate prediction and low computational cost.

First, we consider the case where the target system is known and propose an approximate lifting linearization that derives a lower order lifted linear model based on a known physical model. Then, we propose an LMPC that considers the approximate lifted linear model and the prediction error correction that takes advantages of the features of the lifting linearization. The proposed method introduces constraints that numerical relationships based on observables are satisfied between the elements in the lifted state at each step on the predic-

tion horizon. Along with the additional constraints, we also introduce corrective inputs to ensure the feasibility of the optimization problem.

Then, to deal with the case where the target system is unknown, we propose an LMPC with prediction error correction using a prediction model identified in a data-driven manner. Here, we consider a nonlinear input affine system as the target system. It is known that for nonlinear affine systems, lifting bilinearization is more accurate than lifting linearization. For this reason, we present a method for identifying the lifted bilinear model by a data-driven manner, as well as a method for constructing an LMPC algorithm by linearizing the model at each prediction step and considering the prediction error correction method described above.

In addition, since a physical model of the target system is often available in industry, we will also consider how to apply the LMPC using the lifted bilinear model, considering the effective use of the known model. Here, we assume the situation where a nominal model with model error is obtained against the real plant. We then identify a model error model (MEM) that expresses the dynamics of the prediction error by the nominal model and consider using this model for prediction error correction. Since the model error model is generally a nonlinear system, lifting bilinearization is used to identify the MEM. The nominal model and the identified lifted bilinear MEM are then integrated to form an error-compensated model, and an LMPC using that model is then proposed.

The prediction error correction method based on the observables used in the previous methods increases the number of constraints and optimization variables according to the number of observables and the length of the prediction horizon, which limits the effectiveness of the method. Therefore, finally, we further extend the LMPC using the lifting bilinear model described above and propose a scalable LMPC that does not require additional constraints or optimization variables, so that it can be applied to practical higher-order systems and conditions with long prediction horizons. The effectiveness of the proposed method is demonstrated using a quadrotor model as an example of a practical target system.

目次

第1章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的	3
1.3	論文構成	5
1.4	数学記号	7
第2章	事前知識	8
2.1	モデル予測制御 (Model Predictive Control: MPC)	8
2.1.1	概要	8
2.1.2	線形モデル予測制御 (Linear Model Predictive Control: LMPC)	9
2.1.3	非線形モデル予測制御 (Nonlinear Model Predictive Control: NMPC)	10
2.2	拡大線形化	11
2.2.1	Koopman 作用素	11
2.2.2	拡大線形化とその例	12
2.2.3	拡大線形化の数値計算方法	13
第3章	近似拡大線形モデルを用いた線形モデル予測制御	15
3.1	はじめに	15
3.2	近似拡大線形モデルの導出	15
3.3	拡大線形モデルを用いた MPC における予測誤差補償	17
3.3.1	拡大線形モデルを用いた MPC の定式化	17
3.3.2	予測誤差補償	17
3.4	近似拡大線形モデルを用いた LMPC	19
3.4.1	事前予測状態を用いた QP 問題の定式化	19
3.4.2	事前予測状態の取得	20
3.4.3	LMPC アルゴリズム	21
3.5	数値シミュレーション	22
3.5.1	対象システム	22
3.5.2	近似拡大線形モデル	23
3.5.3	評価条件	24
3.5.4	制御性能の評価結果	24
	状態制約なしの場合	24
	状態制約ありの場合	28

3.5.5	計算時間の評価結果	32
3.6	まとめ	33
第4章	拡大双線形モデルを用いた線形モデル予測制御	34
4.1	はじめに	34
4.2	拡大双線形化	34
4.3	拡大双線形モデルの同定	36
4.4	拡大双線形モデルを用いた MPC の予測誤差補償	37
4.4.1	拡大双線形モデルを用いた MPC の定式化	37
4.4.2	予測誤差補償	38
4.5	拡大双線形モデルを用いた LMPC	39
4.5.1	事前予測状態を用いた QP 問題の定式化	39
4.5.2	事前予測状態の取得	40
4.5.3	LMPC アルゴリズム	41
4.6	数値シミュレーション	42
4.6.1	対象システム	42
4.6.2	拡大双線形モデルの同定	43
4.6.3	制御性能の評価結果	45
	状態制約なしの場合	45
	状態制約ありの場合	49
4.6.4	計算時間の評価結果	53
4.7	まとめ	55
第5章	拡大双線形モデル誤差モデルを用いた線形モデル予測制御	56
5.1	はじめに	56
5.2	モデル誤差モデル (Model-Error Model: MEM)	56
5.3	拡大双線形 MEM の同定	58
5.4	拡大双線形 MEM を用いた LMPC	60
5.4.1	誤差補償モデル	60
5.4.2	誤差補償モデルを用いた MPC の定式化	61
5.4.3	QP 問題の定式化	61
5.4.4	事前予測状態の取得	63
5.4.5	LMPC アルゴリズム	64
5.5	数値シミュレーション	65
5.5.1	対象システム	65
5.5.2	MEM の同定	65

5.5.3	評価結果	67
5.6	まとめ	72
第 6 章	拡大双線形モデルを用いた線形モデル予測制御のスケーラビリティ向上	73
6.1	はじめに	73
6.2	対象システムと拡大双線形モデルの導出	73
6.2.1	対象システム	73
6.2.2	拡大双線形モデルの導出	75
6.3	拡大双線形モデルを用いたスケーラブルな LMPC	77
6.3.1	拡大双線形モデルを用いた MPC の定式化	77
6.3.2	予測誤差補償手法の改良	77
6.3.3	事前予測状態を用いた QP 問題の定式化	79
6.3.4	LMPC アルゴリズム	80
6.4	数値シミュレーション	81
6.4.1	評価条件	81
6.4.2	制御性能の評価結果	81
6.4.3	計算時間の評価結果	85
6.5	まとめ	86
第 7 章	結論	88
付 録 A	LMPC における最適制御問題の QP 問題への変換	90
A.1	線形時不変システムの場合	90
A.2	線形時変システムの場合	92
付 録 B	平面ドローンノミナルモデルの可制御性	95
	参考文献	97
	研究業績	103
	謝辞	106

目 次

1.1	Steps of the research	3
2.1	Model predictive control	9
2.2	Lifting linearization	12
3.1	Acquisition of prior-predicted states	21
3.2	Bicycle kinematic model	22
3.3	States (without state constraint)	25
3.4	Inputs (without state constraint)	26
3.5	Corrective inputs (without state constraint)	27
3.6	Normalized objective function values (without state constraint)	28
3.7	States (with state constraint)	29
3.8	Inputs (with state constraint)	30
3.9	Corrective inputs (with state constraint)	31
3.10	Normalized objective function values (with state constraint)	32
3.11	CPU time	32
4.1	Planar drone	42
4.2	Prediction by the identified model with test data	44
4.3	States (without state constraint)	46
4.4	Inputs (without state constraint)	47
4.5	Corrective inputs (without state constraint)	48
4.6	Normalized objective function values (without state constraint)	49
4.7	States (with state constraint)	50
4.8	Inputs (with state constraint)	51
4.9	Corrective inputs (with state constraint)	52
4.10	Normalized objective function values (with state constraint)	53
4.11	CPU time	54
5.1	Error compensation structure by model-error model	57
5.2	Block diagram for data acquisition	58
5.3	Block diagram for data acquisition with controller	59
5.4	Comparison of prediction accuracy	66
5.5	States	68
5.6	Inputs	69
5.7	Corrective inputs	70

5.8	Normalized objective function values	71
5.9	CPU time	71
6.1	Quadrotor kinematic model	74
6.2	3D trajectories from 10 random initial points	82
6.3	States comparison for a single trajectory	83
6.4	Inputs comparison for a single trajectory	84
6.5	Normalized objective function values	85
6.6	CPU time	86

表 目 次

4.1	Physical parameters of the target system	43
5.1	Physical parameters of the real plant and the nominal model	65

第1章 序論

1.1 本研究の背景

近年、モデル予測制御 (Model Predictive Control: MPC) は様々な分野で注目されている。MPC では、対象システムに関する予測モデルを用いて、有限時間先までの状態変化を予測し、状態や入力に関する制約条件のもとで評価コストを最小化する入力列を計算する。このため、制御対象の非線形性や制約条件を陽に考慮し、性能限界を追及するという問題設定に適した汎用的な手法である [1–4]。このような特徴から、MPC は幅広い産業分野に適用されている [5, 6]。例えば、化学プラント [7]、スマートグリッドなどの電力システム [8, 9]、自動車の経路生成、エンジン、運転支援システム [10–12]、EV のエネルギー管理システム [13]、航空機 [14, 15]、農業 [16]、ビルシステム [17] などを対象に、非常に多岐にわたる分野においてその有用性が示されている。

MPC において、設計した評価関数に基づく高性能な制御を実現するには、予測モデルによる状態の予測精度が非常に重要である。ここで、制御対象のダイナミクスは非線形性を含むことが一般的であり、対象システムの動作条件によっては、固定の動作点近傍でダイナミクスを線形近似することができないケースも多い。そのような制御対象に対しては、非線形状態方程式で表現された予測モデルを用いることで、対象システムの振る舞いをより正確に予測することができる。しかし、そのような非線形状態方程式に基づく非線形モデル予測制御 (Nonlinear Model Predictive Control: NMPC) は、制御周期ごとに解くべき最適制御問題が非線形計画 (NonLinear Programming: NLP) 問題となり、計算コストが大きいため、使用する機器の性能によっては実時間での実行速度が問題となる。このため、予測精度と計算コストのトレードオフを解決することは、非線形システムを対象とした MPC の実用上の重要課題である。

NMPC の実時間での実行を目的に計算コストを低減する手法については、従来より多くの研究がなされている [5]。例えば、C/GMRES [18] では、連続変形法を応用し、微分方程式によって最適解の変化を追跡することで、各時刻の計算は連立 1 次方程式の求解のみとして高速化を実現している。また、アルゴリズムの工夫によって並列計算を行うことで高速化を実現する手法 [19] も提案されている。さらに、想定される初期状態に対して最適解をあらかじめオフラインで計算し、非線形フィードバック則を Neural Network (NN) によって近似する手法も提案されている [20, 21]。ただし、この手法では想定していない状態になってしまった場合や、精度の保証が問題となる。

一方で、NMPC の直接的な高速化とは別方向のアプローチとして、近年、Koopman 理論 [22, 23] に基づく拡大線形化を活用した手法に注目が集まっている [24–26]。拡大線形化は、

対象システムの非線形出力である観測関数を用いて状態を高次元空間に拡大し、観測関数の時間発展を線形ダイナミクスとして表現する手法である。これによれば、対象システムの非線形特性を表現した高次の線形モデル（拡大線形モデル）を獲得することができ、線形システムを対象として確立された幅広い解析・設計手法を適用することができる [27–32]。応用分野としては、流体システム [33–35] や電力システム [36, 37] をはじめとし、様々な対象に適用先が広がっている。

MPC においては、拡大線形モデルを予測モデルとして用いることにより、線形モデル予測制御 (Linear Model Predictive Control: LMPC) を構成することができる。これにより、最適制御問題を二次計画 (Quadratic Programming: QP) 問題として解くことができ、対象システムの非線形特性を考慮しつつ、計算コストを低減可能となることが期待できる。また、NMPC において NLP 問題を解く場合、通常得られる解は局所最適解となるが、QP 問題では大域的最適解が得られることも利点と考えられる。さらに、NMPC を対象とした並列計算などの有用な高速化手法を応用することにより、さらなる高速化の実現についても検討余地がある。このような拡大線形モデルを活用した MPC についても、ロボット [38, 39] や電力システム [40] など、幅広い対象への応用が検討されている。

拡大線形化を具体的に実行するための手法として、従来よりデータドリブンに拡大線形モデルを同定する手法が開発されている。この時、精度の良いモデルを得るためには観測関数の適切な選定が重要であり、これに対して様々な手法が提案されている。良く知られた手法としては、ある次数までの単項式などの基本的な観測関数の辞書を用いる Extended Dynamic Mode Decomposition (EDMD) がある [41]。また、最適化や NN を用いて、観測関数そのものを学習する手法 [42, 43] や、対象システムの知見から観測関数を解析的に構築する手法 [44] も提案されている。異なる方向性として、対象システムの物理的な知識を活用し、力のつり合いなどの線形な関係に着目して非線形要素を抽出して観測関数として用いる Dual Faceted Linearization (DFL) [45–47] や、それに対してさらに Radially Bounded Function (RBF) などの基底関数を追加してロバスト性を向上した Robust DFL [48] などの手法も提案されている。同様に対象システムのモデルに基づく方法として、対象システムのダイナミクスを解析的に多項式関数として近似し、観測関数を構築する手法 [44] も研究されている。また、データドリブンな同定ではなく、解析的に拡大線形化を行う手法としては、Carleman 線形化 [49, 50] が提案されている。

しかしながら、既存の手法によって対象の非線形システムに対する高精度なモデルを獲得するためには、一般的に、対象の非線形システムが比較的低次であっても、非常に多数の観測関数を用意する必要がある。具体的には、10 次元に満たない状態を持つ対象システムに対し、 10^2 オーダーの観測関数が必要となるケースもある [51, 52]。その結果として得られる拡大線形モデルは、観測関数の数に応じて高次元になってしまうため、LMPC であっても相対的に計算コストが増大するという課題がある。その一方、少数の観測関数を用いた場合、観測関数の数に応じた低次の拡大線形モデルが得られるが、一般的に、その予測精度は低下する。そのような拡大線形モデルを用いて LMPC を構成した場合、計算コストは低減できるものの、予測精度が高い非線形モデルを用いた NMPC と比較すると、性能の低下は避けられないといった課題がある。

また、前述した手法は、基本的にデータドリブンで拡大線形モデルを同定する手法である。一方で、産業界においてはモデルベースでの開発が盛んであり、制御系設計の段階で対象システムの物理モデルが得られていることも多い。上記の課題解決のために、このような物理的モデルを有効活用することについても検討の余地がある。

1.2 本研究の目的

本研究では、Koopman アプローチを活用し、高精度な予測に基づく高い制御性能と、低計算コストを両立する MPC を提案することを目的とする。

Koopman 理論に基づき得られる拡大線形モデルは、一般的には対象の非線形システムの近似モデルである。したがって、モデルの次数に関わらず、予測誤差が生じることは避けられない。そこで本研究では、低次の拡大線形モデルに対して、予測誤差補償を考慮した LMPC を構築することにより、高精度な予測に基づく高い制御性能と、LMPC の利点である低計算コストを両立するアプローチを考える。Fig. 1.1 に、本研究の流れを示す。以下では、各ステップにおける検討の概要を説明する。

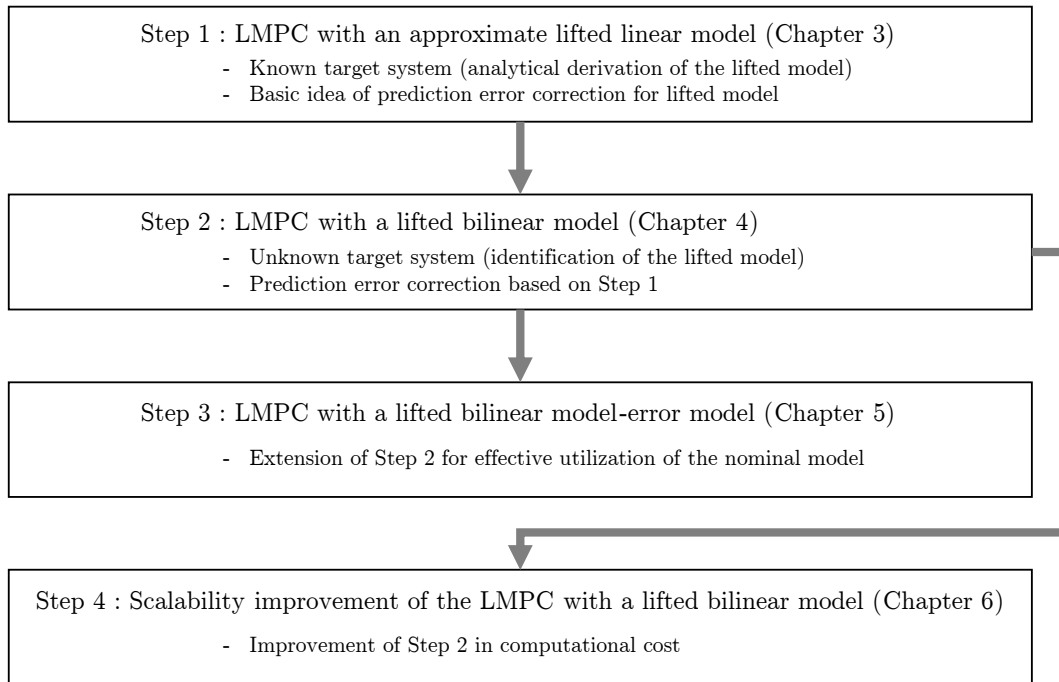


Fig. 1.1: Steps of the research

ステップ1では、対象システムが既知の場合について検討する。ここでは、対象システムに含まれる非線形項の情報を活用し、低次の拡大線形モデルを解析的に導出する近似拡大線形化を提案する。さらに、拡大線形モデルを予測モデルとする LMPC の定式化において、拡大線形化の特徴である状態の拡大を利用した予測誤差補償手法を導入する。具体的には、拡大状態の要素間において、観測関数を介した数値的關係を成立させるための制約条件と、最

適化問題の可解性を確保するための補償入力項を新たに導入する。これらの近似拡大線形化と予測誤差補償手法に基づき、高精度な予測と低計算コストを両立する LMPC を提案する。なお、本ステップで検討する予測誤差補償手法は、以降の検討内容における提案手法の基礎となる。

続いて、ステップ 2 では、対象システムが未知の場合について検討する。ここでは、対象システムが非線形入力アフィンシステムであると仮定し、その予測モデルをデータドリブンに同定することを考える。ここで、非線形入力アフィンシステムに対しては、拡大線形化よりも拡大双線形化を用いることで、より高精度なモデルを得られることが知られており [53]、先行研究では、データに基づき同定した拡大双線形モデルを用いて、対象システムが既知の場合と同等な性能の NMPC を実現する手法が提案されている [54, 55]。しかしながら、従来手法は NMPC であるために計算コストが大きいという課題がある。そこで本研究では、拡大双線形モデルをデータに基づき同定した上で、それを予測ステップごとに線形化した LMPC を考える。さらに、線形化による予測誤差を補償するためにステップ 1 の予測誤差補償手法を考慮し、拡大双線形モデルに基づく LMPC を提案する。

さらに、ステップ 3 ではステップ 2 の産業応用を見据え、対象システムのノミナルモデルが得られている場合に、その情報を有効活用する方法について検討する。産業界ではモデルベース開発が盛んであり、対象システムのノミナルモデルが得られている状況は少なくないが、モデル誤差の発生を回避することは難しい。一方で、製品設計プロセスにおいては、既存のモデルに基づき設計したコントローラを有効に活用しつつ、アドオンの性能改善といった要求もある。そこで本研究では、ノミナルモデルの予測誤差のダイナミクスを表現するモデル誤差モデルを活用し、ノミナルモデルのモデル誤差を補償することで高性能な MPC を実現することを考える。これに向け、ステップ 2 の提案手法を応用し、非線形性を有するモデル誤差モデルを拡大双線形モデルとして同定する。そして、ノミナルモデルと拡大双線形モデル誤差モデルとを統合して誤差補償モデルを構成し、さらに、誤差補償モデルを予測モデルとした LMPC を提案する。

最後に、ステップ 4 ではステップ 2 のさらなる実用性向上に向け、計算コストを低減しスケラビリティを向上するための方法を検討する。ここまでの提案手法で用いた予測誤差補償手法は、観測関数の数や予測ホライゾンの長さに応じて制約条件および最適化変数が増大する。これに伴い、計算コストが増大してしまうという問題があり、その有効性が示せる条件に制限があった。なお、この問題はステップ 1 およびステップ 3 の提案手法についても同様である。そこで、より実践的な高次のシステムや、予測ホライゾンが長い条件にも対応できるように、ここまでの予測誤差補償手法をさらに改良し、制約条件や最適化変数の追加を不要としたスケラブルな LMPC を提案する。そして、実践的な高次の対象システムの例としてクアッドロータモデルを取り上げ、提案手法の有効性を示す。

1.3 論文構成

本論文の構成を以下に示す。

1 章 序論

論文の全体としての研究背景と目的について、先行研究との対比を行いながら紹介を行った。

2 章 事前知識

本章では、本研究に関する事前知識として、まず MPC の基礎的な内容について説明する。続いて、拡大線形化に関する基礎として、非線形自律システムに対する Koopman 理論を用いた線形化について説明し、それを利用した拡大線形化の具体例について説明する。さらに、データに基づき拡大線形システムを同定する方法についても説明する。

3 章 近似拡大線形モデルを用いた線形モデル予測制御

本章では、対象システムの非線形モデルが既知であるという仮定のもと、それに基づき導出される低次の拡大線形モデルを用いて、予測誤差補償を考慮した LMPC を提案する。まず、既知の非線形モデルに含まれる非線形項に基づいて観測関数を選定し、それに基づいて低次の拡大線形モデルを近似的に導出する。そして、近似拡大線形モデルを予測モデルとし、拡大状態に関する制約条件と補償入力を考慮した予測誤差補償手法を導入し、LMPC アルゴリズムを構築する。最後に、車両モデルを扱った数値シミュレーションにより提案手法の有効性を示す。

4 章 拡大双線形モデルを用いた線形モデル予測制御

本章では、対象システムの非線形モデルが未知の入力アフィンシステムであるという仮定のもと、データに基づき同定される拡大双線形モデルと、その予測誤差補償を考慮した LMPC を提案する。まず、2 章で紹介した拡大線形化に関する事前知識をベースとして、拡大双線形化の基礎知識と、データに基づく拡大双線形モデルの同定方法について説明する。続いて、拡大双線形モデルを用いた NMPC の定式化を行い、さらに 3 章の提案手法を応用し、拡大状態に関する制約条件を利用した予測誤差補償を導入する。そして、予測ホライズン上の最適状態列の予測値である事前予測状態を活用して、拡大双線形モデルと制約条件を線形化する。これにより、拡大双線形モデルを用いた LMPC アルゴリズムを構築する。最後に、入力アフィンシステムの例として平面ドローンを対象システムとした数値シミュレーションにより、提案手法の有効性を示す。

5 章 拡大双線形モデル誤差モデルを用いた線形モデル予測制御

本章では、4 章の拡大双線形モデルを用いた提案手法を応用して、既に得られているノミナルモデルのモデル誤差を補償することで、高性能なモデル予測制御を実現する手法を提案する。まず、本研究におけるモデル誤差モデルと、それによるモデル誤差補償

の考え方について説明する．そして，対象システムおよびノミナルモデルから得られるデータに基づいて，拡大双線形モデル誤差モデルを同定する手法について説明する．続いて，ノミナルモデルと拡大双線形モデル誤差モデルを統合した，誤差補償モデルを導出する．さらに，4章の提案手法を応用して，誤差補償モデルを活用した LMPC アルゴリズムを構築する．最後に，平面ドローンを対象システムとした数値シミュレーションにより，提案手法の有効性を示す．

6 章 拡大双線形モデルを用いた線形モデル予測制御のスケラビリティ向上

本章では，4章の提案手法を発展させ，拡大双線形モデルに基づくスケラブルな LMPC を提案し，実践的な非線形システムに適用可能であることを示す．ここでは，対象システムとして 3次元空間で動作するクアッドロータを考え，その解析的な拡大双線形モデルを導出する．また，MPC の定式化において，拡大状態に関する制約条件を追加する代わりに，拡大状態の各要素が満たすべき数値的関係の誤差をフィードバックする補償入力を新たに導入する．これにより，4章の提案手法で問題であった，観測関数の数や予測ホライゾンの長さに応じた制約条件および最適化変数の増加を回避し，計算コストの増大を抑制した手法を構築する．そして，クアッドロータを対象システムとした数値シミュレーションにより，提案手法の有効性を示す．

7 章 結論

本章では，本研究で得られた成果の総括と，我々の研究の今後の発展と課題などについて述べる．

1.4 数学記号

本論文で用いる数学記号を以下にまとめる.

$\mathbb{R}$... 実数全体の集合 ($\mathbb{R}_+$ は正の実数全体の集合)

$\mathbb{R}^n$... n 次元実数ベクトル全体の集合

$\mathbb{R}^{m \times n}$... $m \times n$ 次元実数行列全体の集合

$0_{m \times n}$... m 行 n 列の零行列

I_n ... n 次元単位行列

A^T ... 行列 $A \in \mathbb{R}^{m \times n}$ の転置

A^{-1} ... 正方行列 $A \in \mathbb{R}^{n \times n}$ の逆行列

$|x|$... $x \in \mathbb{R}$ の絶対値

$\|x\|$... ベクトル $x \in \mathbb{R}^n$ のユークリッドノルム

$\|x\|_Q$... ベクトル $x \in \mathbb{R}^n$ の正方行列 $Q \in \mathbb{R}^{n \times n}$ による重み付きノルム $\|x\|_Q = \sqrt{x^T Q x}$

$L_f h$... 関数 $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ の関数 $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ に沿った微分 $L_f h = \frac{\partial h}{\partial x} f(x)$

$\text{diag}(a_1, \dots, a_n)$... スカラー $a_i \in \mathbb{R}$, $(i = 1, \dots, n)$ を対角要素にもつ対角行列

$\mathcal{N}(\mu_x, \Sigma_x)$... 平均値 μ_x , 分散 Σ_x を持つガウス分布

第2章 事前知識

本章では、本研究に関する事前知識として、まず MPC の基礎的な内容について説明する。続いて、拡大線形化に関する基礎として、非線形自律システムに対する Koopman 理論を用いた線形化と、その具体例について説明する。さらに、拡大線形化の具体的な実行方法として、対象システムの入出力データを用いて、入力を持つ制御系に対する拡大線形モデルを同定する方法を説明する。

2.1 モデル予測制御 (Model Predictive Control: MPC)

2.1.1 概要

MPC では、予測モデルに基づく対象システムの振る舞いの予測と、状態や入力に関する制約条件を陽に考慮し、最適な入力を決定する。この考え方を、Fig. 2.1 を用いて説明する。MPC では、現在時刻 k において対象システムの状態を観測し、現在時刻から予測ホライゾン N_p ステップ先までのシステムの振る舞いを考慮する。評価関数として、例えば、予測ホライゾン上における目標値 r と予測出力 y の誤差、および入力 u の大きさを評価する関数を考える。そして、評価関数を最小化するような予測ホライゾン上の最適入力列を、状態や入力に関する制約条件を考慮した最適制御問題を解いて求める。最後に、得られた最適入力列の最初の 1 ステップを実際の対象システムに入力する。この最適化は、制御周期の各サンプリング時刻で繰り返される。なお、予測ホライゾンと独立して入力ホライゾン $N_u (\leq N_p)$ を考え、予測ホライゾンにおける $N_u + 1$ ステップ目以降は N_u ステップ目と同じ入力として、最適化変数の数を減らす方法もあるが、本研究では簡単のため、 $N_u = N_p$ の場合のみを考える。

以上のような方法により、各時刻において将来の対象システムの振る舞いを最適化するような入力の計画を立て、現在の行動を決定することができる。また、最適化において予測された状態と、対象システムの実際の状態が異なっていたとしても、実際の状態を考慮して繰り返し再計画を行うため、外乱の影響も抑制することができる。しかしながら、予測モデルに基づいて将来の振る舞いの最適化を行うため、予測モデルの精度はコントローラの性能に直接的に影響することに注意する必要がある。

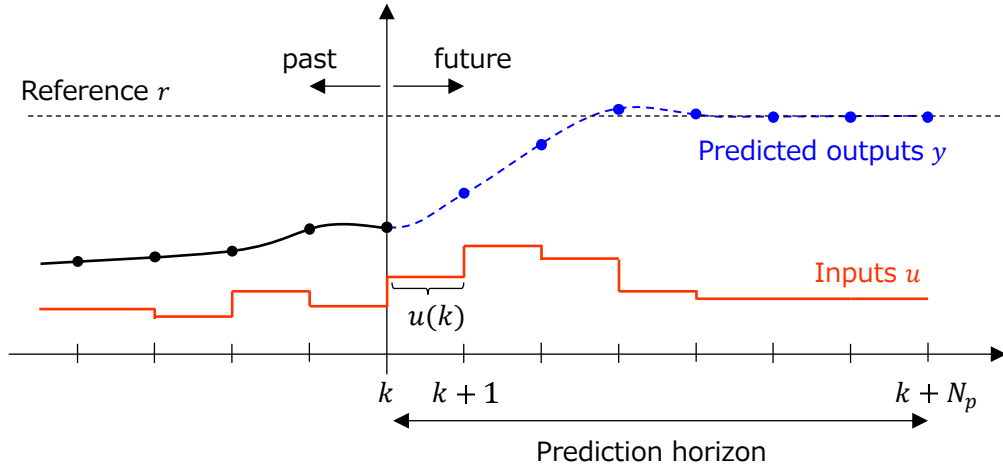


Fig. 2.1: Model predictive control

2.1.2 線形モデル予測制御 (Linear Model Predictive Control: LMPC)

対象システムが線形システムである場合や、非線形システムであってもその動作領域において線形システムとして見なせる場合には、線形システムで表現される予測モデルを用いた線形モデル予測制御 (LMPC) を適用することができる。

次の離散時間線形システムを考える。

$$x_{k+1} = Ax_k + Bu_k \quad (2.1)$$

$$y_k = Cx_k \quad (2.2)$$

ここで、 $x_k \in \mathbb{R}^n$ を状態、 $u_k \in \mathbb{R}^m$ を入力、 $y_k \in \mathbb{R}^p$ を出力とし、 k を時刻インデックスとする。また、 A, B, C はそれぞれ適切な次元の行列とする。

(2.1), (2.2) を予測モデルとした LMPC では、各時刻で解くべき最適制御問題を次のように定式化する。

$$\underset{u_k}{\text{minimize}} \quad J_l \quad (2.3a)$$

$$\text{subject to} \quad x_0 = x_t \quad (2.3b)$$

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N_p - 1 \quad (2.3c)$$

$$y_k = Cx_k, \quad k = 0, \dots, N_p \quad (2.3d)$$

$$A_c \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_c \geq 0, \quad k = 0, \dots, N_p \quad (2.3e)$$

ここで、 N_p は予測ホライズン、(2.3b) 式における x_t は時刻 t において観測 (あるいは推定) された状態とする。また、(2.3e) 式は状態と入力に関する線形制約条件を表しており、行列 A_c とベクトル b_c は適切な次元とする。また、評価関数 J_l は、次式で与えられる。

$$J_l = \|y_{N_p} - y_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|y_k - y_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2) \quad (2.4)$$

ここで, $P, Q \in \mathbb{R}^{p \times p}$ は準正定, $R \in \mathbb{R}^{m \times m}$ は正定な重み行列であり, $y_k^r, k = 0, \dots, N_p$ と $u_k^r, k = 0, \dots, N_p - 1$ はそれぞれ目標出力と目標入力とする.

最適制御問題 (2.3) は, 2 次の評価関数と線形制約条件から構成されており, 2 次計画 (Quadratic Programming: QP) 問題に変換することができる. 計算の詳細については, 付録 A.1 を参照されたい. また, 予測モデル (2.1), (2.2) が線形時変システムの場合については, 付録 A.2 を参照されたい. QP 問題は凸最適化問題であり, アクティブセット法 [56] や内点法 [57] などの高性能なアルゴリズムを適用することで, 比較的高速に大域的最適解を求めることができる.

2.1.3 非線形モデル予測制御 (Nonlinear Model Predictive Control: NMPC)

対象システムがその動作領域において強い非線形性を有する場合には, 非線形システムで表現される予測モデルを用いた非線形モデル予測制御 (NMPC) を適用することで, 対象システムに関するより正確な予測を考慮した制御を行うことができる.

次の離散時間非線形システムを考える.

$$x_{k+1} = f(x_k, u_k) \quad (2.5)$$

$$y_k = h(x_k) \quad (2.6)$$

ここで, f, h は非線形関数ベクトルであり, $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n, h: \mathbb{R}^n \rightarrow \mathbb{R}^p$ とする.

(2.5), (2.6) 式を予測モデルとした NMPC では, 各時刻で解くべき最適制御問題を次のように定式化する.

$$\underset{u_k}{\text{minimize}} \quad J_n \quad (2.7a)$$

$$\text{subject to} \quad x_0 = x_t \quad (2.7b)$$

$$x_{k+1} = f(x_k, u_k), \quad k = 0, \dots, N_p - 1 \quad (2.7c)$$

$$y_k = h(x_k), \quad k = 0, \dots, N_p \quad (2.7d)$$

$$c(x_k, u_k) \geq 0, \quad k = 0, \dots, N_p \quad (2.7e)$$

ここで, (2.7e) 式は状態と入力に関する非線形制約条件を表しており, c はベクトル値関数である. また, 評価関数 J_n は次式で与えられる.

$$J_n = \phi(y_{N_p}) + \sum_{k=0}^{N_p-1} L(y_k, u_k) \quad (2.8)$$

ここで, ϕ と L はそれぞれ終端コストとステージコストを表すスカラー値関数であり, 例えば, 出力の目標に対する誤差や入力の大きさなどからなる.

最適制御問題 (2.7) は, 非線形計画 (NonLinear Programming: NLP) 問題である. NLP 問題のためのアルゴリズムやソルバも多く開発されているが, 求解のための計算コストは, 一

般に QP 問題よりも大きく、実時間での実行速度がしばしば問題となる。実時間最適化アルゴリズムの例としては、C/GMRES [18] や、並列計算を利用した手法 [19] などが挙げられる。また、QP 問題が凸最適化問題であるのに対して、NLP 問題は非凸最適化問題であるため、最適解は局所最適解になることが一般的である。

2.2 拡大線形化

2.2.1 Koopman 作用素

Koopman 作用素とは、非線形システムに対して導入される線形作用素であり、非線形システムの状態空間で定義されるスカラー値関数（これを観測関数と呼ぶ）の時間発展を表わす作用素として定義される [26]。Koopman 作用素は、非線形自律システムを対象に、Koopman 作用素のスペクトルに基づいて非線形システムの特性を特徴づけるための理論として提案された [22, 23]。

次の離散時間の非線形自律システムを考える。

$$x_{k+1} = f_d(x_k) \quad (2.9)$$

ここで、 k は時刻インデックスであり、 $x_k \in \mathbb{R}^n$ とする。さらに、状態に関する非線形関数である観測関数を $\psi : \mathbb{R}^n \rightarrow \mathbb{C}$ とする。この時、Koopman 作用素 $\mathcal{K}$ は、観測関数に働く無限次元の線形作用素であり、次式に示すように、観測関数の (2.9) に従った時間発展を線形ダイナミクスとして表現することができる。

$$(\mathcal{K}\psi)(x_k) = \psi(f_d(x_k)) \quad (2.10)$$

また、 $x \in \mathbb{R}^n$ を状態とした連続時間の非線形自律システム

$$\dot{x}(t) = f_c(x(t)) \quad (2.11)$$

に対しては、サンプル時間 $\Delta t > 0$ に対応した Koopman 作用素の半群 $\mathcal{K}_{\Delta t}$ が存在する。この半群の無限小生成作用素 $\hat{\mathcal{K}}$ は

$$(\hat{\mathcal{K}}\psi)(x(t)) = \nabla\psi \cdot f_c(x(t)) \quad (2.12)$$

となり、これを連続時間 Koopman 作用素と呼ぶ。ここで、 $\cdot$ は $\mathbb{R}^n$ 上の内積、 ∇ は勾配演算子である。

$\hat{\mathcal{K}}$ の固有値 $\lambda \in \mathbb{C}$ を Koopman 固有値と呼ぶ。この時、Koopman 固有関数は、Koopman 固有値に対応して (2.11) の解に沿って線形な振る舞いをする関数 $\phi : \mathbb{R}^n \rightarrow \mathbb{C}$ であり、その将来時刻の値は次の関係によって表される。

$$\phi(x(t + \Delta t)) = (\mathcal{K}_{\Delta t}\phi)(x(t)) = e^{\lambda\Delta t}\phi(x(t)) \quad (2.13)$$

2.2.2 拡大線形化とその例

z を観測関数ベクトル $h : x \rightarrow z$ によって拡大された状態とすると, Koopman 作用素を用いて, 非線形システムのダイナミクスを線形システムとして表現することができる (Fig. 2.2). 次節で述べる入力を持つシステムへの拡張も含め, これを拡大線形化と呼ぶ. 一般に, Koopman 作用素は無限次元となるが, ここでは特殊な例として, 有限個の観測関数によって拡大線形化を行うことのできる例を次に示す.

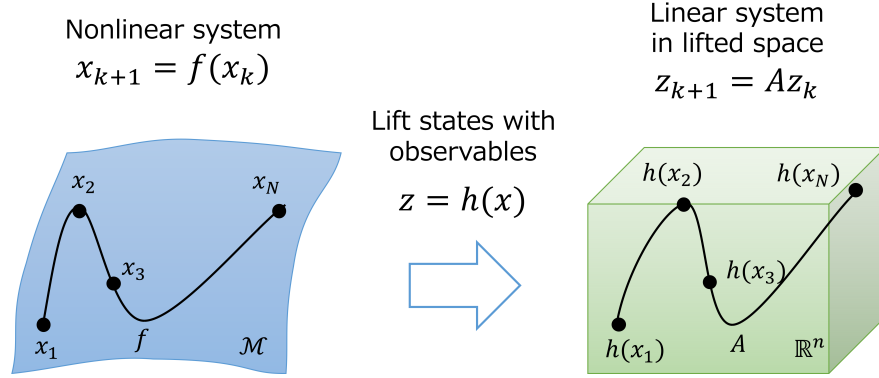


Fig. 2.2: Lifting linearization

$x = [x_1 \ x_2]^T \in \mathbb{R}^2$ を状態とする, 次の連続時間の非線形自律システムを考える.

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mu x_1 \\ \lambda(x_2 - x_1^2) \end{bmatrix} \quad (2.14)$$

ここで, $\mu, \lambda \in \mathbb{R}$ はパラメータとする. (2.14) 式のシステムに対し, 次の観測関数ベクトル $h(x)$ を考える.

$$h(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix} \quad (2.15)$$

観測関数ベクトルによって高次元空間に写像された拡大状態を $z = [z_1 \ z_2 \ z_3]^T = h(x) \in \mathbb{R}^3$ とすると, z のダイナミクスは次式のように線形システムとして表すことができる.

$$\frac{d}{dt} z = \begin{bmatrix} \mu x_1 \\ \lambda(x_2 - x_1^2) \\ 2\mu x_1 \end{bmatrix} = \begin{bmatrix} \mu z_1 \\ \lambda(z_2 - z_3) \\ 2\mu z_1 \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix} z \quad (2.16)$$

この例では, 有限個の観測関数を用いて, 元の非線形システム (2.14) 式に沿ったダイナミクスを閉じた形で表せるので, Koopman 作用素も有限次元の行列として表すことができる. しかし, 観測関数のダイナミクスがこのように閉じた形で表現できるケースは稀であり, 一般的には Koopman 作用素は無限次元の行列となる. このため, 次節で述べるように, あらかじめ選定した有限個の観測関数を用いて近似的に線形化することで拡大線形化を行う手法が広く用いられている.

2.2.3 拡大線形化の数値計算方法

Koopman 作用素は一般に無限次元となるが、あらかじめ選定した有限個の観測関数と、元の非線形システムの入出力データを用いることにより、有限次元の行列として近似的に求めることができる。良く知られた手法として、Extended Dynamic Mode Decomposition (EDMD) [41] がある。また、そのような手法は入力を持つ制御系に対しても適用可能に拡張されている [58, 59]。これらの手法によって、入力を持つ場合についても非線形システムを高次の線形システムとして表現することができれば、既存の線形制御理論を用いてコントローラやオブザーバを設計し、元の非線形システムに対して適用することも可能となる。

以降では、入力を持つ非線形システムの入出力データを用いて、拡大線形システムを同定する方法について述べる。なお、入力に関する項を無視すれば、自律系の場合についても同様の方法を適用することができる。

対象システムに関し、次のデータセット $\mathcal{D}$ が得られているとする。

$$\mathcal{D} = \{x_k, u_k\}, \quad k = 0, \dots, M \quad (2.17)$$

ここで、 $x_k \in \mathbb{R}^{n_x}$ は状態、 $u_k \in \mathbb{R}^m$ は入力であり、サンプル時間を Δt とする。

次に、 n_z 個の観測関数 $\psi_i(x), i = 1, \dots, n_z$ を導入し、これらによって構成される観測関数ベクトルを

$$h(x) = \begin{bmatrix} \psi_1(x) \\ \vdots \\ \psi_{n_z}(x) \end{bmatrix} \quad (2.18)$$

とする。この時、データセット $\mathcal{D}$ の各サンプル時刻に対応した拡大状態 z_k は、

$$z_k = h(x_k), \quad k = 0, \dots, M \quad (2.19)$$

として計算できる。

今、求めたい拡大線形モデルを、 z_k を状態とした次の離散時間線形システムとする。

$$z_{k+1} = Az_k + Bu_k \quad (2.20)$$

この時、データセット $\mathcal{D}$ に基づいて、(2.20) の係数行列 A, B は次の最適化問題を解いて求めることができる。

$$\min_{A, B} \sum_{k=0}^{M-1} \|z_{k+1} - (Az_k + Bu_k)\|^2 \quad (2.21)$$

最適化問題 (2.21) は、サンプル時刻間の拡大状態の予測誤差を最小化するように係数行列を決定する問題である。この解は、最小二乗法によって求めることができる。 $H = [A \ B] \in \mathbb{R}^{n_z \times n_z + m}$ とし、さらに、

$$Z_+ = [z_1, \dots, z_M], \quad Z_u = \begin{bmatrix} z_0 & \cdots & z_{M-1} \\ u_0 & \cdots & u_{M-1} \end{bmatrix} \quad (2.22)$$

とする．ここで， Z_+ と Z_u の各列は，各サンプル時刻における (2.20) の左辺と右辺の関係にある．この時， H の推定値は

$$\hat{H} = Z_+ Z_u^T (Z_u Z_u^T)^{-1} \quad (2.23)$$

と計算することができ， A, B はこの部分行列として得ることができる．

さらに，(2.18) の観測関数ベクトル $h(x)$ が次式の形で構成される場合を考える．

$$h(x) = \begin{bmatrix} x \\ \eta(x) \end{bmatrix} \quad (2.24)$$

ここで， $\eta(x) \in \mathbb{R}^{n_\eta}$ は対象システムの状態に関して非線形な観測関数のベクトルとする．なお， $n_\eta = n_z - n_x$ である．この時，射影行列 $P = [I_{n_x} \ O_{n_x \times n_\eta}]$ を用いて，拡大線形モデルの出力方程式は

$$x_k = P z_k \quad (2.25)$$

と表すことができる．なお，一般の観測関数ベクトルに対する出力方程式の同定方法については [24] を参照されたい．

また，最適化問題 (2.21) においては，求める係数行列のスパース性を向上して過学習の影響を抑えるため，正則化項を考慮してもよい．L1 正則化により予測精度が向上するという結果は，例えば先行研究 [60] に示されている．

第3章 近似拡大線形モデルを用いた線形モデル予測制御

3.1 はじめに

拡大線形化では、非線形システムの振る舞いを表現する高次の線形システムを得ることが可能である。しかし、得られた線形システムは、あくまで非線形システムの近似であるため、MPCの予測モデルとして利用した場合には、真の非線形ダイナミクスに対する予測誤差が生じ、制御性能が低下する場合がある。特に、低次で拡大線形化した場合には、一般的に誤差が大きくなる傾向がある。

これを踏まえ、本章では、低次の拡大線形モデルを用いた場合でも高精度な予測を可能とする、近似拡大線形化と予測誤差補償を考慮したLMPCを提案する。本章では、まず、対象システムの物理モデルが既知であるという仮定のもと、既知モデルに含まれる非線形項を活用した近似拡大線形化により、低次の時変な拡大線形モデルを導出する。続いて、拡大線形モデルを予測モデルとしたMPCを定式化し、さらに、観測関数に基づく拡大状態に関する特徴を考慮した予測誤差補償手法を導入する。そして、これらを考慮したLMPCアルゴリズムを構築し、その有効性を数値シミュレーションにより示す。

3.2 近似拡大線形モデルの導出

ここでは、対象の非線形システムのモデルが第一原理モデリングなどにより得られていることを仮定し、その情報を活用した拡大線形化を考える。対象システムとして、次の入力アフィン非線形システムを考える。

$$\dot{x} = f(x) + g(x)u \quad (3.1)$$

ここで、 $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^m$ とする。次に、 $f(x)$ に含まれる状態に関する全ての非線形項から構成されるベクトルを $\eta(x) \in \mathbb{R}^{n_\eta}$ とする。さらに、元の状態と $\eta(x)$ を縦に並べたベクトルを次のように構成する。

$$h(x) = \begin{bmatrix} x \\ \eta(x) \end{bmatrix} \quad (3.2)$$

この時、対象システムのダイナミクス (3.1) に沿った $h(x)$ の時間発展は、次のように表す

ことができる.

$$\frac{d}{dt}h(x) = \begin{bmatrix} A_x \\ A_\eta \end{bmatrix} h(x) + \begin{bmatrix} g(x) \\ B_\eta(x) \end{bmatrix} u + \begin{bmatrix} W_x \\ W_\eta(x) \end{bmatrix} \quad (3.3)$$

ここで, 行列 $A_x, A_\eta, B_\eta(x), W_x, W_\eta(x)$ は, 次の関係を満たす.

$$A_x h(x) + W_x = f(x) \quad (3.4a)$$

$$A_\eta h(x) + W_\eta(x) = \frac{\partial \eta}{\partial x} f(x) \quad (3.4b)$$

$$B_\eta(x) = \frac{\partial \eta}{\partial x} g(x) \quad (3.4c)$$

すなわち, $A_x \in \mathbb{R}^{n_x \times n_x}$ は $f(x)$ に含まれる $h(x)$ の要素に関して線形な項の係数行列, $W_x \in \mathbb{R}^{n_x}$ は $f(x)$ に含まれる定数項のベクトルである. また, $A_\eta \in \mathbb{R}^{n_\eta \times n_\eta}$ は $\eta(x)$ の時間微分 ($\dot{\eta} = \frac{\partial \eta}{\partial x} f(x) + \frac{\partial \eta}{\partial x} g(x)u$) に含まれる $h(x)$ の要素に関して線形な項の係数行列, $B_\eta(x) \in \mathbb{R}^{n_\eta \times m}$ は $\eta(x)$ の時間微分に含まれる u に関して線形な項の係数行列, $W_\eta(x) \in \mathbb{R}^{n_\eta}$ は $\eta(x)$ の時間微分に含まれる $h(x)$ の要素と u の線形結合で記述できない項のベクトルである.

以上を踏まえ, $h(x)$ を観測関数ベクトルとした拡大状態を $z := h(x) \in \mathbb{R}^{n_z}$ とする. また,

$$A := \begin{bmatrix} A_x \\ A_\eta \end{bmatrix}, \quad B(x) := \begin{bmatrix} g(x) \\ B_\eta(x) \end{bmatrix}, \quad W(x) := \begin{bmatrix} W_x \\ W_\eta(x) \end{bmatrix} \quad (3.5)$$

とすると, 非線形システム (3.1) は, 拡大状態 z に関して次のように変形することができる.

$$\dot{z} = Az + B(Pz)u + W(Pz) \quad (3.6)$$

ここで, P は z から x への射影行列であり, $P := [I_{n_x} \ O_{n_x \times n_\eta}]$ である. なお, $B(Pz)$ と $W(Pz)$ は状態 z に依存した行列であるため, (3.6) はこの段階では非線形システムであることに注意されたい.

ここで, z の時刻に応じた予測値をあらかじめ得ることができれば, それを用いて, $B(Pz)$ と $W(Pz)$ を時変行列として近似することができる. これにより, 近似的な時変の線形モデルを得ることができる. 事前に得られている時刻 t における z の予測値 (事前予測状態) を $\hat{z}(t)$ とし,

$$\hat{B}(t) := B(P\hat{z}(t)), \quad \hat{W}(t) := W(P\hat{z}(t)) \quad (3.7)$$

と定義すれば, 次の線形時変モデルが得られる.

$$\dot{z} = Az + \hat{B}(t)u + \hat{W}(t) \quad (3.8)$$

本研究では, この変形を近似拡大線形化と呼び, これにより得られた線形時変モデル (3.8) を近似拡大線形モデルと呼ぶ. 近似拡大線形モデルを用いて予測された拡大状態を, 元の状態空間に $x = Pz$ として射影すれば, 対象システム (3.1) の状態を予測することができる.

なお, (3.6) は非線形モデル (3.1) の厳密な変形となっているため, 近似拡大線形モデル (3.8) においても, 事前予測状態 $\hat{z}(t)$ の精度が良ければ, 非線形モデル (3.1) の振る舞いを精度良く近似することが可能であることに注意されたい. また, 事前予測状態は MPC の枠組みを利用すれば容易に取得することができる. この方法については後述する.

3.3 拡大線形モデルを用いた MPC における予測誤差補償

本節では、拡大線形化により得られた線形モデルを予測モデルとした MPC において、その予測誤差を補償する手法を提案する。なお、本節で扱う拡大線形モデルは、前節で示した近似拡大線形化によって得られるものに限定されないことに注意されたい。

3.3.1 拡大線形モデルを用いた MPC の定式化

まず、拡大線形モデルを用いた MPC の基本形として、各時刻で解く最適化問題を次のように構成する。

$$\underset{u_k}{\text{minimize}} \quad J_1 \quad (3.9a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (3.9b)$$

$$z_{k+1} = A_d z_k + B_d u_k + W_d, \quad k = 0, \dots, N_p - 1 \quad (3.9c)$$

$$x_k = P z_k, \quad k = 0, \dots, N_p \quad (3.9d)$$

$$A_c \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_c \geq 0, \quad k = 0, \dots, N_p \quad (3.9e)$$

ここで、 N_p は予測ホライゾン、 x_t は時刻 t において観測された状態とする。(3.9c) は離散時間の拡大線形モデルによる予測式であり、ここでは A_d, B_d, W_d を定数行列とする。さらに、 A_c, b_c は状態と入力に関する線形制約条件を構成するための行列およびベクトルである。評価関数 J_1 は次式で与えられる。

$$J_1 = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2) \quad (3.10)$$

ここで、 $P, Q \in \mathbb{R}^{n_x \times n_x}$ は準正定、 $R \in \mathbb{R}^{m \times m}$ は正定な重み行列であり、 $x_k^r, k = 0, \dots, N_p$ と $u_k^r, k = 0, \dots, N_p - 1$ はそれぞれ目標状態と目標入力とする。

最適化問題 (3.9) では、まず現在のシステムの状態 x_t を観測して (3.9b) で観測関数ベクトル $h(x)$ により拡大状態空間に写像し、予測ホライゾン上の初期ステップにおける拡大状態を得る。そして、拡大状態の変化を拡大線形モデル (3.9c) で予測し、出力関数 (3.9d) によって拡大状態を元の状態空間に射影して、評価関数 (3.9a) を最小化する最適入力列 $u_k, k = 0, \dots, N_p - 1$ を求める。その際に、状態と入力に関する線形制約条件 (3.9e) も考慮する。以上が拡大線形モデルを用いた MPC の定式化である。

3.3.2 予測誤差補償

前述したように、拡大線形モデルは元の非線形システムの近似であるため、予測誤差が生じて MPC の性能低下を招く可能性がある。特に、拡大状態のうち、元の状態に関する非線

形関数に対応する要素 ((3.2) における $\eta(x)$ に対応する要素) のダイナミクスについては、一般的には、観測関数として考慮した非線形関数よりもさらに高次の非線形項を有する。しかし、低次で近似した場合、そのような高次の非線形項は考慮しきれず、予測誤差が生じやすい。そのような予測誤差は、拡大状態のうち、元の状態に対応する要素 ((3.2) における x に対応する要素) の精度にも影響する。その結果として、正しい予測に基づいて評価関数の値を最小化する入力列を決定できず、制御性能を低下させる可能性がある。

このような誤差の影響を補償するために、拡大状態の特徴を踏まえた制約条件と、補償入力項を導入する。すなわち、まず、予測ホライズン上の各ステップにおいて、拡大状態の要素が観測関数の関係式を満たすという制約条件を設ける。さらに、最適化問題の可解性を確保するため、拡大状態のうち、非線形関数 $\eta(x)$ に対応する要素に関して、補償入力項を追加する。これらを考慮した最適化問題は次のようになる。

$$\underset{u_k, \Delta u_k}{\text{minimize}} \quad J_2 \quad (3.11a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (3.11b)$$

$$z_{k+1} = A_d z_k + B_d u_k + W_d + \begin{bmatrix} O_{n \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \quad (3.11c)$$

$$x_k = P z_k, \quad k = 0, \dots, N_p \quad (3.11d)$$

$$z_k(n+1:n_z) = \eta(z_k(1:n)), \quad k = 1, \dots, N_p \quad (3.11e)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (3.11f)$$

ここで、 $\Delta u_k \in \mathbb{R}^{n_\eta}$ は各予測ステップ k における $n_x + 1$ から n_z 番目の状態に対応した補償入力であり、 $z_k(i:j)$ は、Matlab 表記を用いて z_k の i から j 番目の要素のベクトルを表すものとする。また、評価関数 J_2 は、 Δu_k に関する重み行列を $R_\Delta \in \mathbb{R}^{n_\eta \times n_\eta}$ として、次式で与えられる。

$$J_2 = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2 + \|\Delta u_k\|_{R_\Delta}^2) \quad (3.12)$$

最適化問題 (3.11) において、追加の制約条件 (3.11e) では、拡大状態のうち元の状態に対応する $z_k(1:n_x)$ と、非線形な観測関数によって拡大した部分に対応する $z_k(n_x+1:n_z)$ が、各ステップにおいて観測関数の関係を満たすことを考慮している。この最適化問題を解くことにより、各ステップにおける拡大状態の要素が制約条件を満たすように最適な補償入力項が計算され、拡大線形モデルによる予測誤差を補償することができる。拡大状態 z_k の予測精度の改善は元の状態 x_k の予測精度の改善につながるため、MPC の制御性能も改善されると期待できる。

なお、補償入力 Δu_k は評価関数の設定によってできるだけ小さくなるように計算されるため、状態軌道の最適化は、できるだけ本来の入力を用いて実現されると期待される。また、拡大線形モデルが元の非線形モデルを精度良く近似できていれば、制約条件 (3.11e) の両辺

の誤差は小さくなり、補償入力 Δu_k も小さな値をとると考えられる。つまり、補償入力 Δu_k は予測モデルのダイナミクスの不確かさを必要に応じて補償するが、本来の入力の最適化計算を妨げないものと期待される。

ここで、上記の定式化においては、拡大状態に関する制約条件が非線形な観測関数 $\eta(\cdot)$ を用いた関係式であるため、非線形最適化問題となっている。このため、LMPC を構築するためにはこの制約条件の線形化が必要である。この方法については次節で説明する。

3.4 近似拡大線形モデルを用いた LMPC

本節では、3.2 節で導出した近似拡大線形モデルと、3.3 節で説明した予測誤差補償手法を活用した LMPC のアルゴリズムを提案する。

3.4.1 事前予測状態を用いた QP 問題の定式化

近似拡大線形化によって線形時変モデル (3.8) を得るためには、将来時刻 t における拡大状態 z の事前予測状態 $\hat{z}(t)$ をあらかじめ得る必要がある。また、最適化問題 (3.11) においては、非線形制約条件 (3.11f) が含まれており、LMPC を構築するためにはこの制約条件の線形化が必要である。

ここで、MPC では各時刻において最適な入力列を求めるのと同時に、最適な予測状態列を計算するため、前サンプル時刻における最適化問題の解を活用し、上記の両課題を解決することを考える。つまり、前サンプル時刻の最適解に基づいて事前予測状態を取得し、それに基づき、近似拡大線形化による線形時変モデル (3.8) を得るとともに、非線形制約条件 (3.11f) を線形化する。

予測ホライゾンの k ステップ目における事前予測状態を z_k^* , $k = 0, \dots, N_p$ とすると、それに対応した元の状態の予測値は $x_k^* = Pz_k^*$ として得ることができる。これを用いて、(3.6) より、近似拡大線形化による離散時間線形予測モデルは次式で与えられる。

$$z_{k+1} = A_d z_k + B_d(x_k^*) u_k + W_d(x_k^*) \quad (3.13)$$

ここで、

$$A_d = I_n + A \Delta t \quad (3.14a)$$

$$B_d(x_k^*) = B(x_k^*) \Delta t \quad (3.14b)$$

$$W_d(x_k^*) = W(x_k^*) \Delta t \quad (3.14c)$$

であり、 Δt は離散化のサンプル時間とする。

また、予測ホライゾンの k ステップ目における拡大状態に関する非線形制約条件は、一次のテーラー近似により次式のように線形制約条件で表すことができる。

$$z_k(n+1 : n_z) = \eta(x_k^*) + \left. \frac{\partial \eta}{\partial x} \right|_{x=x_k^*} (z_k(1:n) - x_k^*) \quad (3.15)$$

以上より、事前予測状態 z_k^* , $k = 0, \dots, N_p$ が得られているとしたとき、各制御周期において解くべき最適化問題は次のように定式化できる。

$$\underset{u_k, \Delta u_k}{\text{minimize}} \quad J_2 \quad (3.16a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (3.16b)$$

$$x_k^* = Pz_k^*, \quad k = 0, \dots, N_p \quad (3.16c)$$

$$z_{k+1} = A_d z_k + B_d(x_k^*) u_k + W_d(x_k^*) + \begin{bmatrix} O_{n \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \quad (3.16d)$$

$$x_k = Pz_k, \quad k = 0, \dots, N_p \quad (3.16e)$$

$$z_k(n+1:n_z) = \eta(x_k^*) + \frac{\partial \eta}{\partial x} \Big|_{x=x_k^*} (z_k(1:n) - x_k^*), \quad k = 0, \dots, N_p \quad (3.16f)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (3.16g)$$

最適化問題 (3.16) では、予測ホライズンにおける拡大状態の初期値 z_0 を求める計算式のみ非線形な計算を含むが、それ以外では、二次形式の評価関数と線形の制約条件で構成されている。このため、高速な QP ソルバで解くことが可能な定式化となっている。

3.4.2 事前予測状態の取得

次に、事前予測状態 z_k^* の取得方法を説明する。まず、ある時刻 t において最適化問題 (3.16) を解いて得られた最適入力列 $u_k, \Delta u_k$, $k = 0, \dots, N_p - 1$ と、線形予測モデル (3.13) を用いて、各ステップにおける拡大状態の最適値 z_k^{*-} , $k = 0, \dots, N_p$ を計算することができる。ここで、制御周期が離散時間線形予測モデル (3.13) のサンプル時間 Δt と一致していると仮定すると、次の時刻 $t + \Delta t$ における予測ホライズン上の状態は、最終ステップを除き、前の時刻の最適状態列 z_k^{*-} を 1 ステップシフトすることであらかじめ予測することができる。また、最終ステップについては、1 ステップ前の値を用いて近似すればよい。この計算過程の図を、Fig. 3.1 に示す。

すなわち、事前予測状態 z_k^* は次のようにして取得することができる。

$$\begin{aligned} z_k^* &= z_{k+1}^{*-}, \quad k = 0, \dots, N_p - 1 \\ z_{N_p}^* &= z_{N_p}^{*-} \end{aligned} \quad (3.17)$$

なお、制御周期が離散時間線形予測モデル (3.13) のサンプル時間 Δt と一致しない場合については、制御周期分の時間経過を考慮して、予測ホライズン上の時刻が最も近くなるように、前の時刻の最適状態列 z_k^{*-} を適切なステップ数シフトすればよい。

また、制御ループの初期時刻においては、前時刻が存在しないため、上記の方法は適用できない。そこで初期時刻においては、対象システムの初期状態に基づく拡大状態を、予測ホ

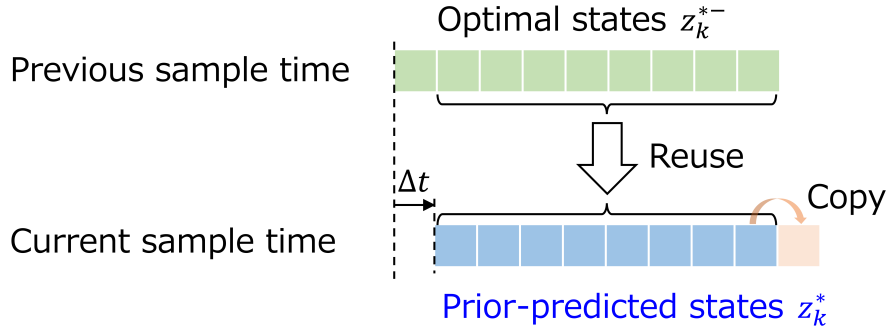


Fig. 3.1: Acquisition of prior-predicted states

ライゾンの全ステップで使用する．すなわち，対象システムの初期状態を x_{t_0} として，次のように取得することができる．

$$z_k^* = h(x_{t_0}), \quad k = 0, \dots, N_p \quad (3.18)$$

なお，初期時刻における事前予測状態を取得するための別の方法として，事前にオフラインで非線形モデルを予測モデルとした非線形最適化問題を解き，それにより得られた最適状態列を用いてもよい．その方がより高精度な値を利用でき，近似拡大線形化と非線形制約条件の近似精度が向上するため，初期時刻付近の制御性能がより改善されると期待できる．

3.4.3 LMPC アルゴリズム

以上より，提案手法の近似拡大線形モデルを用いた LMPC アルゴリズムは Algorithm 1 のようにまとめられる．

Algorithm 1 LMPC with the approximated lifted model

Require: Reference trajectory (x^r, u^r)

while Controller is running **do**

if $t=0$ **then**

 Set the prior-predicted states z_k^* by (3.18)

end if

 Observe current state x_t of the system

 Solve the optimization problem (3.16) with the prior-predicted states z_k^*

 Update the prior-predicted states z_k^* by (3.17)

 Apply first input u_0 to the system

end while

本提案手法によれば，近似拡大線形化による予測に対し，予測誤差の補償と，補償結果によるモデル精度向上を同時に行うことができる．すなわち，ある時刻において事前予測状態

の精度が低く、近似拡大線形モデル (3.13) の近似精度が低いとしても、少なくとも拡大状態のうち非線形観測関数 $\eta(x)$ に対応する要素については、拡大状態に関する制約条件 (3.15) を満たすように補償入力 Δu_k を加えることで、発生する予測誤差を低減できる。そして、誤差が低減された状態予測に基づき最適状態列を求め、次の時刻の事前予測状態を構成することで、次の時刻における近似拡大線形モデル (3.13) の精度を改善するとともに、拡大状態に関する制約条件 (3.15) の近似精度も改善される。このような反復的な精度向上効果によって、提案の LMPC アルゴリズムは、高速な求解が可能な QP 問題の形式でありながら、非線形性を考慮して高い性能を達成できる。

3.5 数値シミュレーション

本節では、直線路を目標軌道とした車両モデルを対象システムとして、数値シミュレーションにより提案手法の有効性を評価した結果を説明する。

3.5.1 対象システム

対象システムとして、Fig. 3.2 に示す二輪車両モデルを考える。

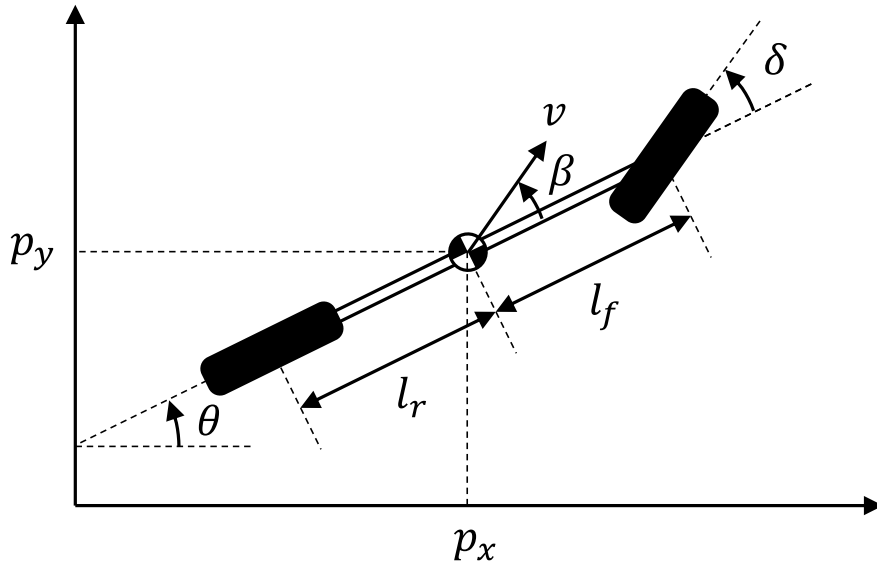


Fig. 3.2: Bicycle kinematic model

対象システムの運動学モデルのダイナミクスは、次のように与えられる。

$$\dot{x} = f(x) + g(x)u = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ 0 \\ \frac{1}{l_r} v \sin(\beta(\delta)) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ \dot{\delta} \end{bmatrix} \quad (3.19)$$

ここで、状態は $x = [p_x, p_y, v, \theta, \delta]^T$ であり、 p_x, p_y はグローバル座標系における車両の x, y 座標、 v は車両重心における速度、 θ は車体方位角、 δ は前輪の操舵角を表す。また、入力 $u = [a, \dot{\delta}]^T$ であり、 a は加速度、 $\dot{\delta}$ は角速度を表す。さらに、 $\beta(\delta) = \tan^{-1}(\frac{l_r}{l_f + l_r} \tan \delta)$ は速度ベクトル方向と車体方位角の間の滑り角である。 l_f と l_r はそれぞれ前輪と後輪から車両重心までの長さであり、ここでは $l_f = l_r = 1.5[\text{m}]$ とした。

3.5.2 近似拡大線形モデル

3.2 節の手順に従い、近似拡大線形モデルを導出する。まず、 $f(x)$ に含まれる非線形項より、観測関数ベクトル $h(x) = [x^T, \eta^T(x)]^T$ において、非線形観測関数ベクトル $\eta(x)$ は、

$$\eta(x) = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ v \sin \beta(\delta) \end{bmatrix} \quad (3.20)$$

と選ぶことができる。ここで、 $\eta(x)$ の (3.19) に沿った時間微分は、

$$\frac{d}{dt}\eta(x) = \frac{\partial \eta(x)}{\partial x} \dot{x} = \begin{bmatrix} a \cos \theta - \frac{v^2}{L} \tan \delta \sin \theta \\ a \sin \theta + \frac{v^2}{L} \tan \delta \cos \theta \\ a \tan \theta + \omega \frac{v}{\cos^2 \delta} \end{bmatrix} \quad (3.21)$$

と計算できる。したがって、(3.19) と (3.21) に基づき、(3.5) における $A, B(x), W(x)$ 行列は、次のように導出できる。

$$A = \begin{bmatrix} O_{5 \times 5} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{L} \\ 0 & 0 & 0 \end{bmatrix} \\ O_{3 \times 5} & O_{3 \times 3} \end{bmatrix}, B(x) = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \\ \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \tan \theta & \frac{v}{\cos^2 \delta} \end{bmatrix} \end{bmatrix}, W(x) = \begin{bmatrix} O_{5 \times 1} \\ \begin{bmatrix} -\frac{v^2}{L} \tan \delta \sin \theta \\ \frac{v^2}{L} \tan \delta \cos \theta \\ 0 \end{bmatrix} \end{bmatrix} \quad (3.22)$$

(3.22) を用いて、近似拡大線形化による離散時間線形予測モデルは (3.13) で得ることができる。得られた予測モデルの次数は $n_z = 8$ であり、対象システム (3.19) に含まれる最小限の非線形項に基づく低次な線形時変モデルとなっている。

3.5.3 評価条件

ここでは、直線上を移動する目標点に対する対象システム (3.19) の追従制御を行い、提案手法の有効性を検証した結果について説明する。目標軌道は、 $x_d(t) = [v_d \cos \theta_d, v_d \sin \theta_d, v_d, \theta_d, 0]$ とし、今回は $v_d = 5.6[\text{m/s}]$, $\theta_d = \pi/4[\text{rad}]$ と設定した。

本検証では、比較対象として、真のモデルを予測モデルとした NMPC(以下, NMPC)、目標軌道近傍において対象システムを線形近似したノミナルモデルを予測モデルとした LMPC(以下, ノミナル LMPC) についても評価した。

なお, NMPC のソルバには CasADi [61] をインターフェースとして IPOPT を使用し, LMPC のソルバには MATLAB の quadprog 関数を使用した。シミュレーション実行環境は, Core i5 CPU@1.20GHz, 24.0GB メモリ, Windows 10 の PC 上で動作する MATLAB 2020b を使用した。

3.5.4 制御性能の評価結果

状態制約なしの場合

まず、状態に関する制約条件なしの場合のシミュレーションを実施した。サンプル時間は $0.01[\text{s}]$, 予測ホライゾンの長さは 30 ステップ, 評価関数 J_2 における重み行列は, $P = 100 \cdot I_5$, $Q = I_5$, $R = 0.1 \cdot I_3$, $\Delta R = 0.001 \cdot I_2$ とした。対象システムの初期状態は $x_{t_0} = [2, -1, 7, 0, 0]^T$ とし, また入力の変数 u の制約条件を $|u| \leq 10, |\dot{u}| \leq \pi/4$ とした。また, 今回はノイズの影響は考慮していない。

Fig. 3.3 と Fig. 3.4 に, 各手法による状態と入力の比較結果をそれぞれ示す。ここで, 状態のグラフに関しては, 目標軌道からの偏差を示している。また, 図中の赤の破線は制約条件を示している。これらの図より, 状態と入力の両方に関して, ノミナル LMPC は真のモデルを用いた NMPC に対して大きく異なる振る舞いをしていることが分かる。これに対し, 提案手法では LMPC で入力を計算しているにもかかわらず, NMPC とほとんど差がない振る舞いを達成できている。

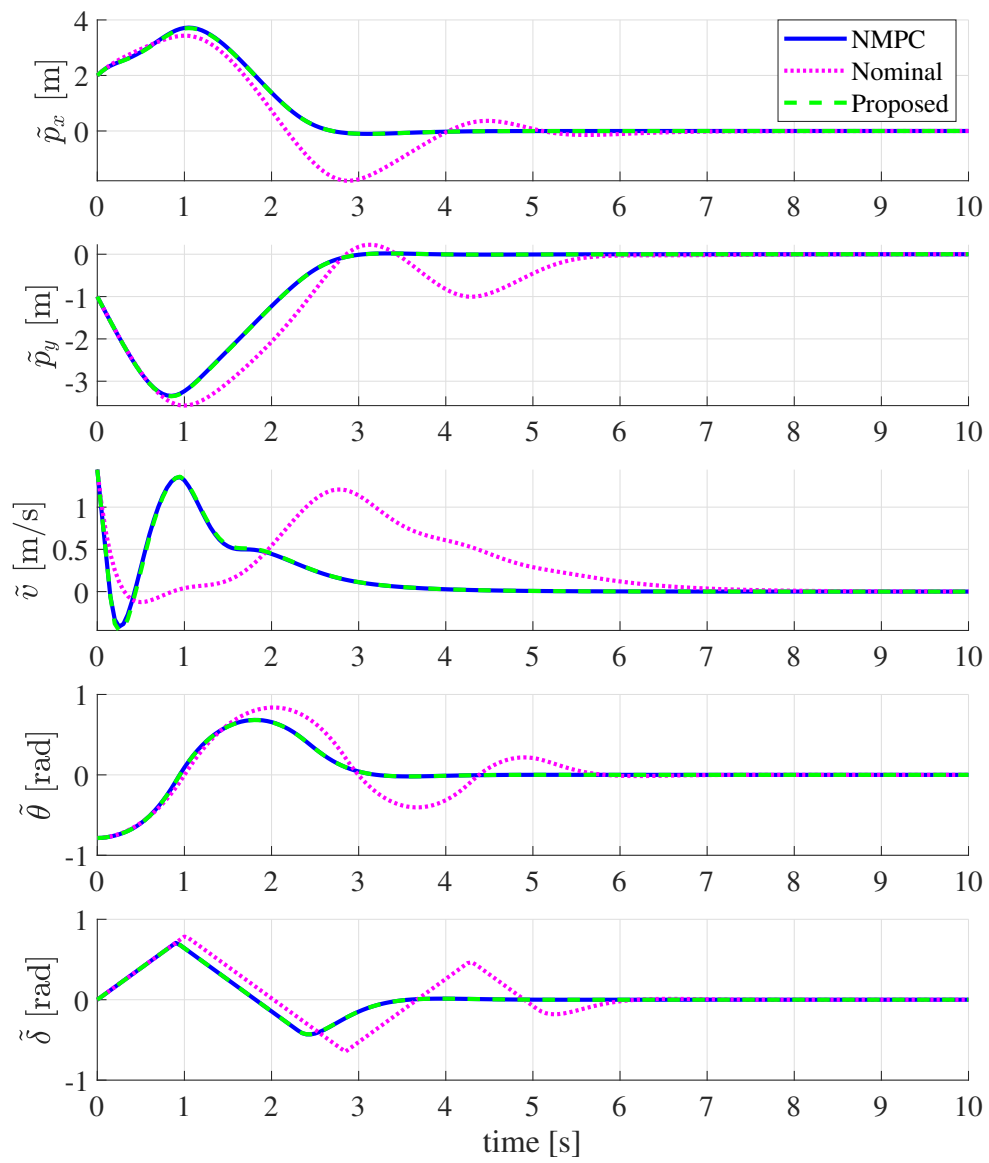


Fig. 3.3: States (without state constraint)

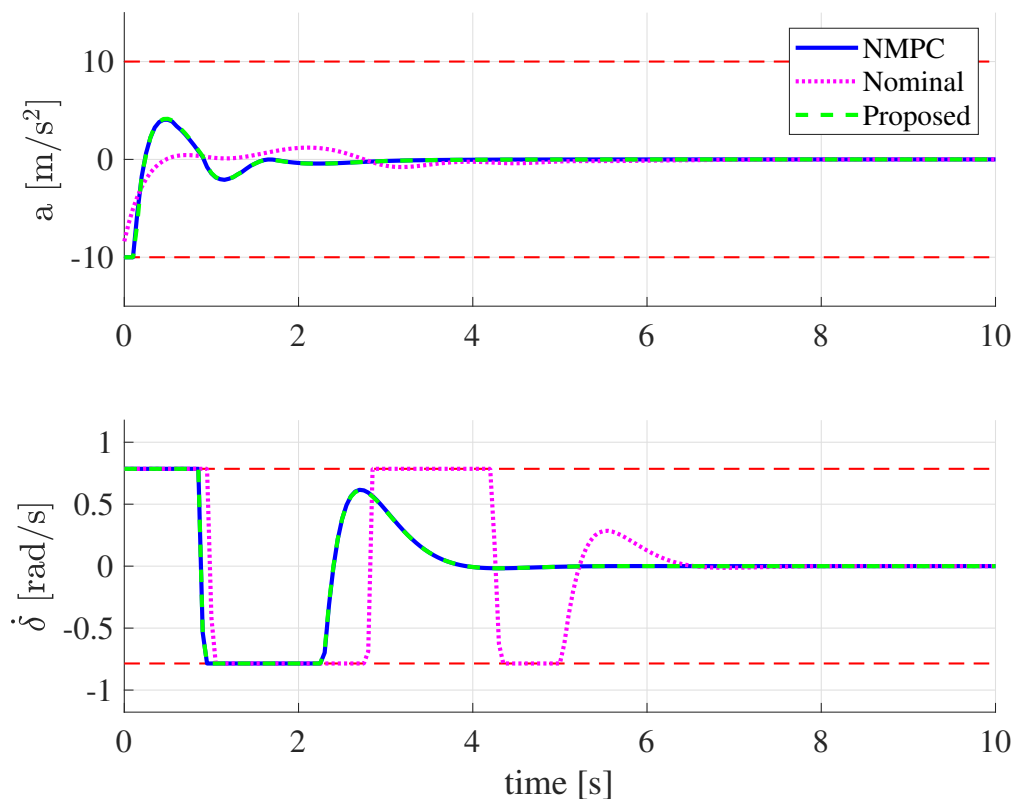


Fig. 3.4: Inputs (without state constraint)

ここで, Fig. 3.5 に, 提案手法において拡大した 3 つの次元に対応する補償入力 Δu の初期ステップの値をプロットしたグラフを示す. 補償入力, 特に過渡応答において相対的に大きな値をとって予測誤差を補償しており, 定常状態ではゼロに近い値に収束している.

これらの結果より, ノミナル LMPC における単純な線形近似モデルでは対象システムの非線形性を含めた振る舞いを適切に予測できず, 性能が低下するのに対し, 提案手法では予測誤差を補償し, NMPC と同等の性能を達成できると考えられる.

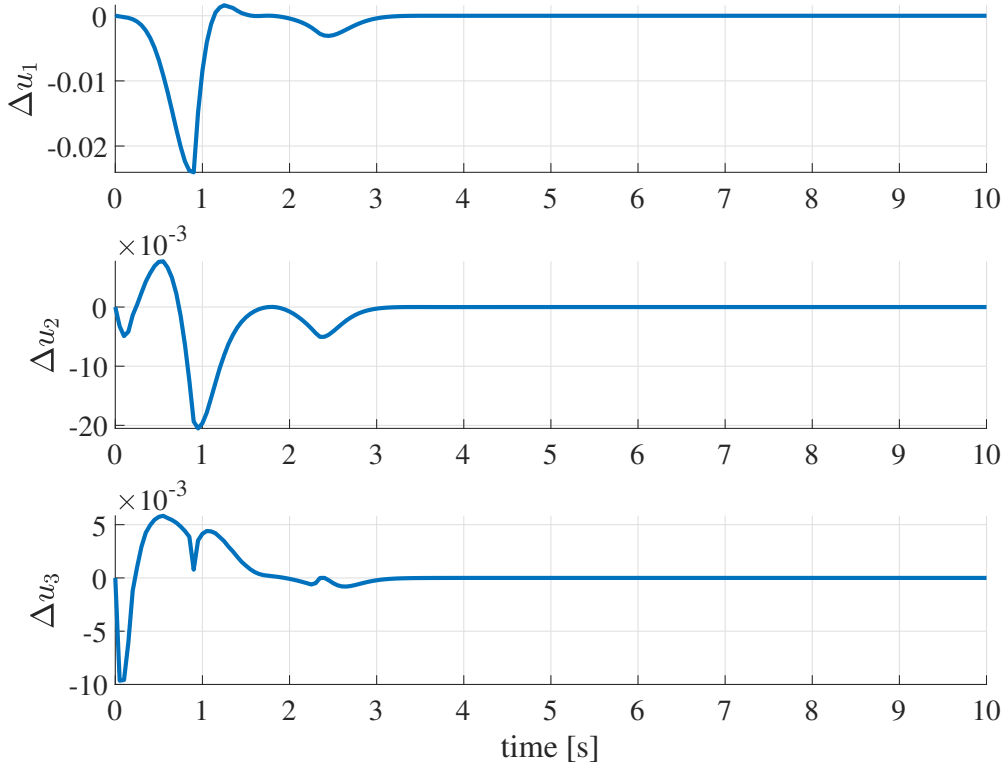


Fig. 3.5: Corrective inputs (without state constraint)

次に、同じ目標軌道を用いて、異なる初期状態を用いて 30 回のシミュレーションを行い、次の評価関数により各手法の性能を定量的に評価した。

$$J_e = \sum_{k=0}^N (\|x_k - \bar{x}\|_Q^2 + \|u_k - \bar{u}\|_R^2) \quad (3.23)$$

ここで、 N は 10[s] のシミュレーションで取得したデータ点数である。初期状態は $x_{t0} = [\pm 5, \pm 5, v_d \pm 2.7, \theta_d \pm \pi/2, \pm \pi/6]^T$ の範囲で一様にランダムに設定した。各試行において、評価関数値を NMPC の値で正規化し、全試行の平均を計算した結果を Fig. 3.6 に示す。ここで、ノミナル LMPC については全体の 23% について大きく振動的な応答となったため、ここでは目標軌道にうまく収束できた試行の結果のみを集計した。この結果より、NMPC に対し、ノミナル LMPC で評価関数値が 40% 悪化しているのに対し、提案手法では 1% しか差がなかった。したがって、異なる初期状態を用いた複数回の試行に対しても、提案手法では NMPC と同等の性能を達成可能であることを示すことができた。

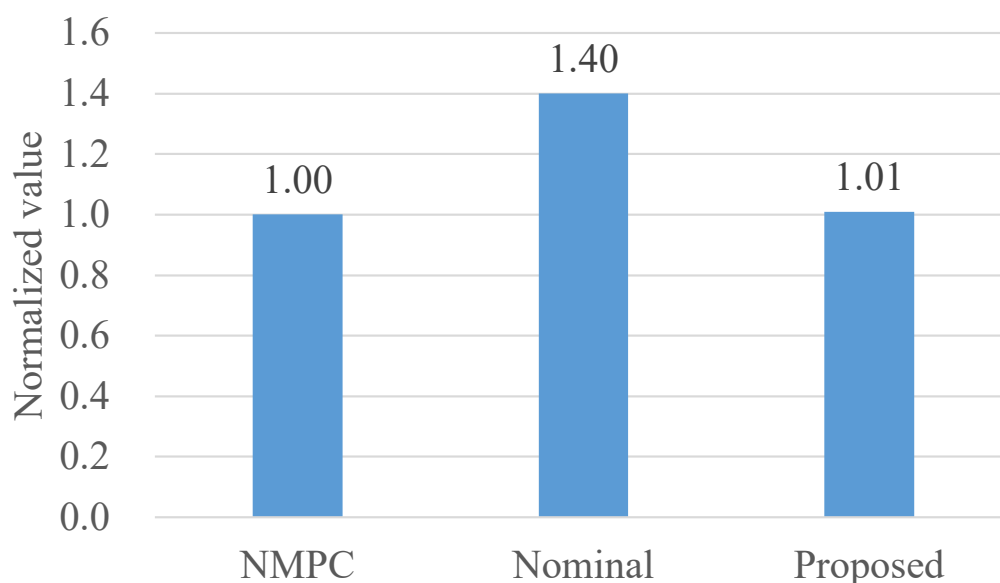


Fig. 3.6: Normalized objective function values (without state constraint)

状態制約ありの場合

続いて、状態に関する制約条件として、対象システムの操舵角の範囲を $-\frac{\pi}{6} \leq \delta \leq \frac{\pi}{6}$ とした場合のシミュレーション結果を示す。なお、その他のシミュレーション条件は前述の状態制約なしの場合と同様である。

Fig. 3.7 と Fig. 3.8 に、各手法による状態と入力の比較結果をそれぞれ示す。これらの結果より、いずれの手法においても操舵角に関する状態制約は満足できていることが分かる。その上で、状態制約がある場合においても、状態制約がない場合と同様に、ノミナル LMPC と比べて提案手法の方が NMPC に近い応答が得られていることが分かる。

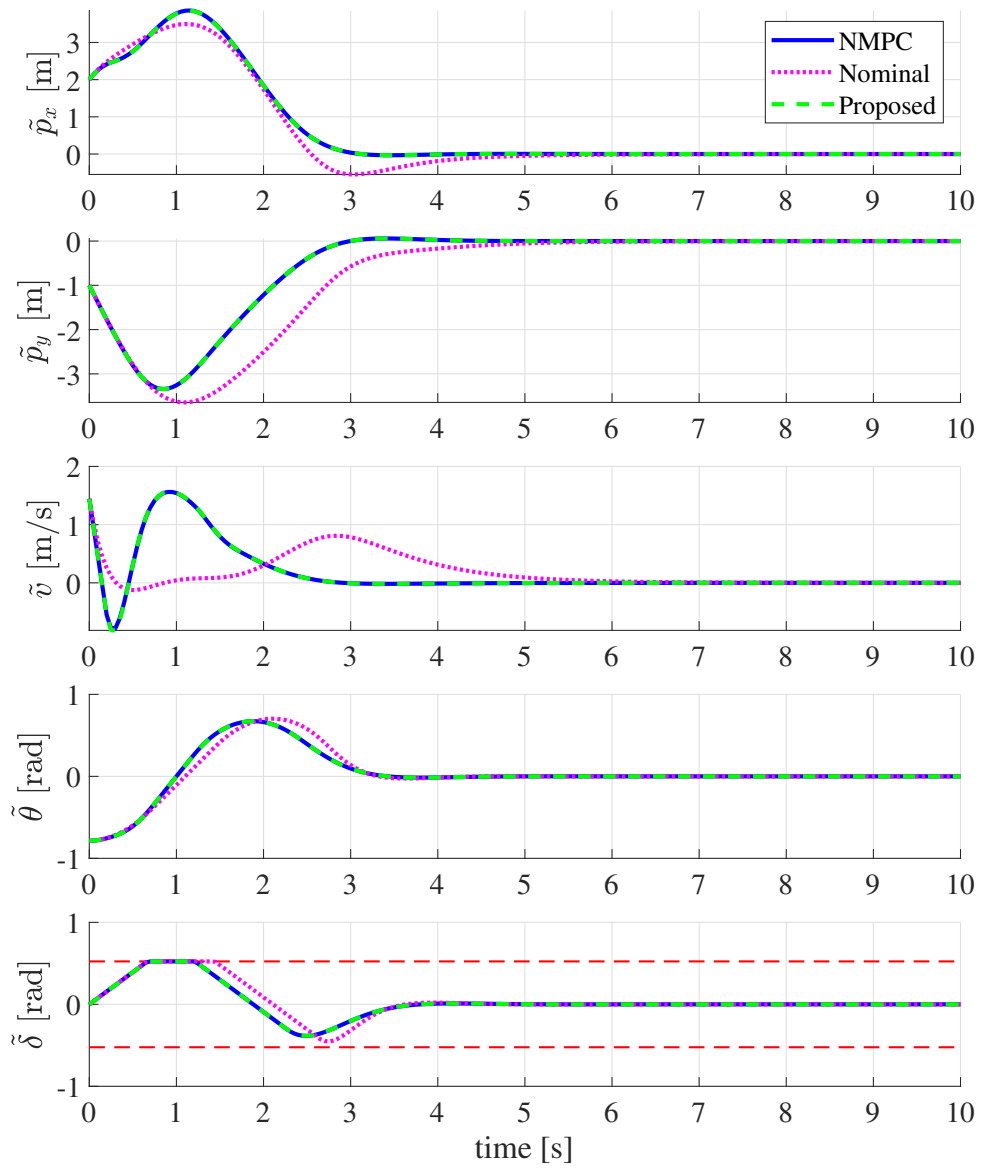


Fig. 3.7: States (with state constraint)

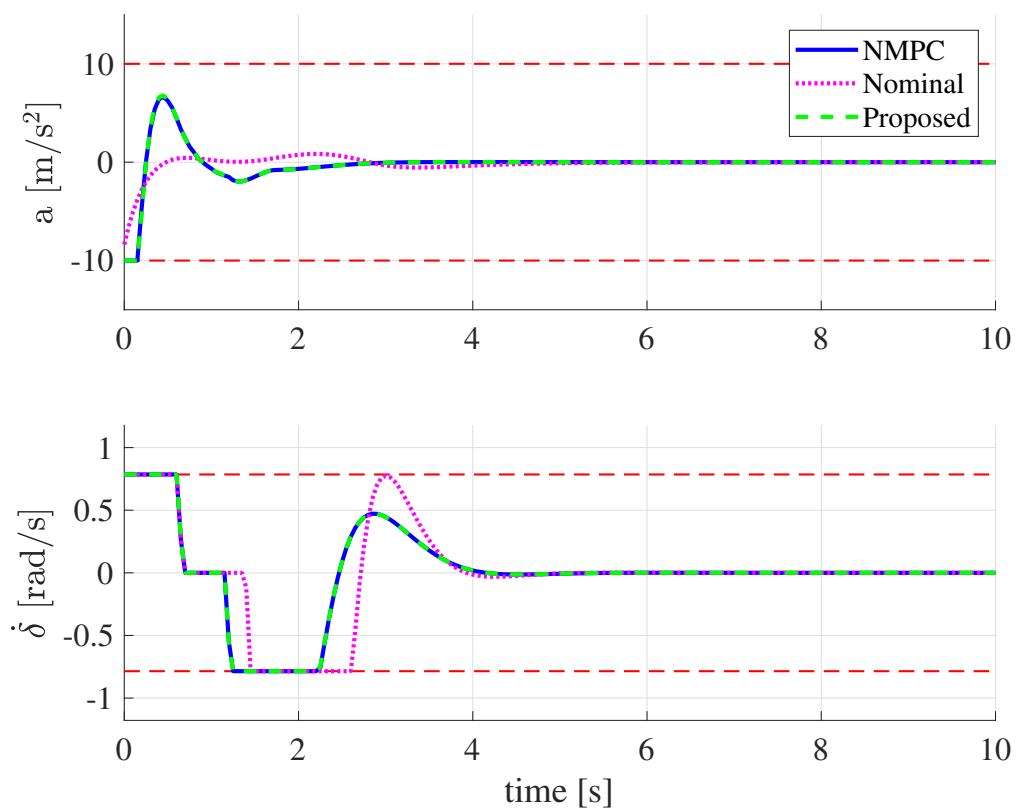


Fig. 3.8: Inputs (with state constraint)

また、Fig. 3.9 に、提案手法における補償入力 Δu の初期ステップの値をプロットしたグラフを示す。こちらも状態制約がない場合と同様に、過渡応答では補償入力が相対的に大きな値となるが、定常状態ではゼロに近い値に収束していることが分かる。

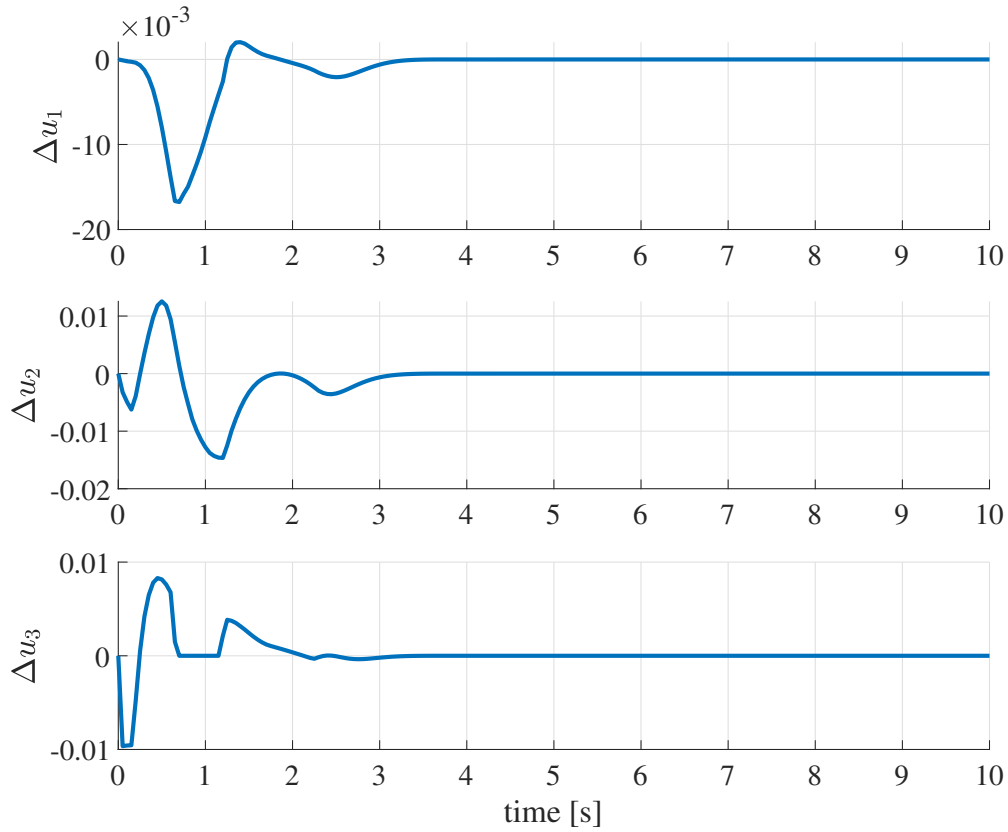


Fig. 3.9: Corrective inputs (with state constraint)

さらに、状態制約なしの場合と同様に、異なる初期状態を用いて30回のシミュレーションを行い、評価関数 (3.23) により各手法の性能を評価した。ここで、初期状態の選び方と評価関数値の計算方法については同様である。ただし、今回は状態制約を設定した影響により、NMPCでも目標値に追従できないケースが全体の20%あったため、今回はNMPCで目標値に追従できた試行の結果のみを集計した。評価結果を Fig. 3.10 に示す。この結果より、NMPC に対し、ノミナル LMPC で評価関数値が11%悪化しているのに対し、提案手法では状態制約なしの場合と同様に、1%しか差がなかった。したがって、状態制約ありの場合についても、異なる初期状態からの複数回の試行に対して、提案手法はNMPCと同等の性能を達成可能であることを示すことができた。

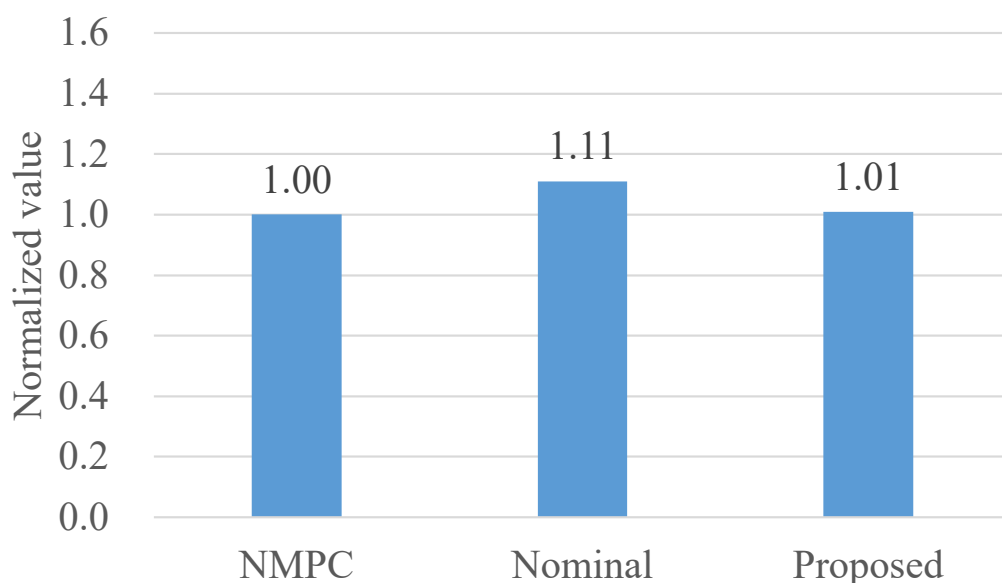


Fig. 3.10: Normalized objective function values (with state constraint)

3.5.5 計算時間の評価結果

最後に、計算時間の評価結果について述べる．Fig. 3.11 に、予測ホライズン $N_p = 10, 20, 30$ の場合の、各手法の最適化ソルバの求解時間を示す．なお、ここでは状態制約なしの場合の集計結果を示す．

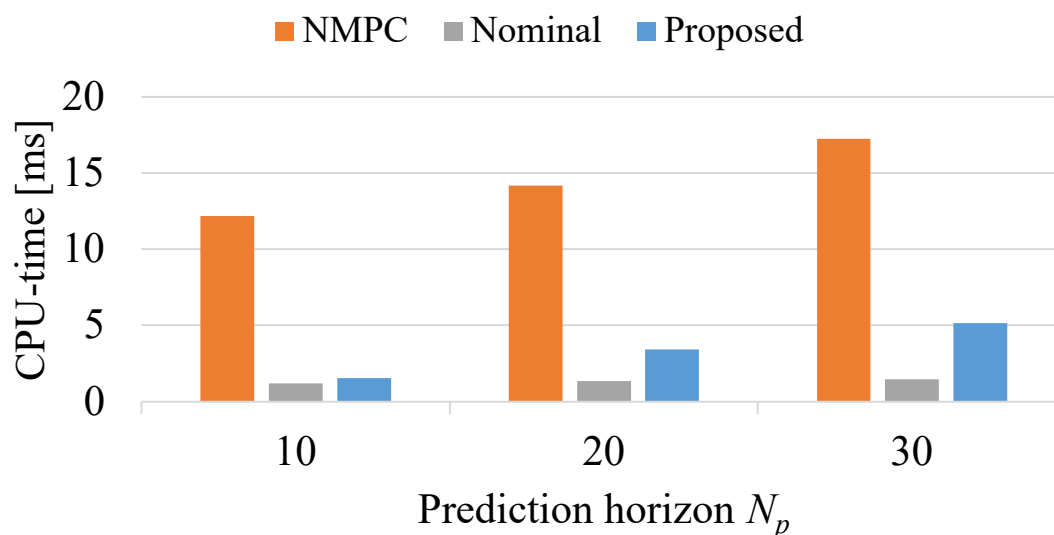


Fig. 3.11: CPU time

この結果より、提案手法では NMPC よりも大幅に計算時間を低減できていることが分かる．一方、ノミナル LMPC と比較すると提案手法の方がやや計算時間が大きいですが、これは、

提案手法ではノミナル LMPC と比較して予測モデルの次数が大きいこと、また拡大状態に関する制約条件の考慮が必要であることが理由として考えられる。

以上の結果より、提案手法を用いることで、NMPC と同等の制御性能を高速な LMPC で実現可能であることを検証することができた。

なお、事前予測状態の近傍で対象システムを線形近似した線形時変モデルを用いて MPC を構成した場合でも、提案手法と同様の制御性能を達成できる可能性がある。これに対し、提案手法のように拡大線形化を用いる手法においては、観測関数に補助的な基底関数を加える手法 [48] などを用いることによってさらに性能向上が図れるという点や、対象システムのモデルが未知の場合であっても、拡大線形化によるモデル同定と組み合わせることが可能であるという点で利点がある。

3.6 まとめ

本章では、低次の拡大線形モデルを用いて高精度な予測を実現するための、近似拡大線形化と予測誤差補償を考慮した LMPC について検討した。近似拡大線形化では、既知の非線形モデルに基づいて観測関数を定義し、さらに事前予測状態を活用することで低次の線形時変システムを導出した。また、拡大線形モデルを予測モデルとした MPC において、拡大状態に関する制約条件と補償入力項を考慮した予測誤差補償手法を提案し、これらを活用した LMPC アルゴリズムを提案した。提案手法によれば、対象システムの非線形性を考慮した高精度な予測と低計算コストを両立した MPC を実現することができる。

数値シミュレーションでは、二輪車両モデルを対象システムとして、提案手法の有効性検証を行い、真のモデルを用いた NMPC と同等の制御性能を達成でき、さらに計算時間を大幅に低減可能であることを示すことができた。

第4章 拡大双線形モデルを用いた線形モデル予測制御

4.1 はじめに

3章では対象システムの物理モデルが既知であるという仮定のもと、その知識を活用した近似拡大線形化を用いて拡大線形モデルを導出し、それをMPCの予測モデルとして活用した。これに対し本章では、対象システムの物理モデルが未知の場合を想定し、データドリブンに予測モデルを同定することを考える。ここで、例えばドローンや車両などのビークルのダイナミクスを表現する非線形入力アフィンシステムに対しては、拡大双線形モデルをデータドリブンに学習することで高精度な予測モデルを得られることが知られている [54, 55]。しかしながら、MPCの予測モデルとして拡大双線形モデルを使用すると、各時刻で解くべき最適化問題がNLP問題となるため、計算コストの面で課題があった。

そこで本章では、未知の入力アフィンシステムを対象として、データに基づき同定した拡大双線形モデルを活用した、高い予測精度と低計算コストを両立するLMPCを提案する。まず、拡大双線形化の基礎的な内容と、拡大双線形モデルの同定方法について説明する。続いて、拡大双線形モデルを用いたNMPCの定式化を行い、これに対し、3章の提案手法を応用した予測誤差補償を導入する。そして、予測ホライゾン上の最適状態列の予測値である事前予測状態を活用して、最適化問題における非線形要素を線形化する。これにより、拡大双線形モデルを用いたLMPCアルゴリズムを構築する。また、その有効性を数値シミュレーションにより示す。

4.2 拡大双線形化

対象システムとして次の入力アフィンシステムを考える。

$$\dot{x} = f(x) + g(x)u \quad (4.1)$$

ここで、 $x \in \mathbb{R}^n$ 、 $u \in \mathbb{R}^m$ である。入力アフィンシステムでは、一般に、入力状態依存の行列 $g(x)$ を介して状態変化に作用する。具体的には、例えばドローンや車両などのビークルのダイナミクスにおいて、駆動力が姿勢角に依存した回転行列を介して状態に影響するといった状況が挙げられる。

一方で、2.2節で説明した拡大線形化の数値計算方法では、選定した観測関数に基づき、対象システム (4.1) を線形モデルとして同定する。この時、入力の係数行列 B は状態に依存し

ない定数行列としてとして求められるため、状態依存の係数の影響を適切に表現できない場合がある。例えば、姿勢角の変化が大きいビークルなどに対しては、元のシステムに対し十分に精度の良いモデルが得られない可能性がある。

これに対し、入力アフィンシステムは、Koopman 正準変換 (Koopman Canonical Transformation: KCT) を用いて、ある条件下において双線形システムに変換できることが知られている [53, 54]。

$(\lambda_i, \phi_i), i = 1, \dots, n_z$ を (4.1) の自律ダイナミクス $\dot{x} = f(x)$ に関する Koopman 作用素の固有値-固有関数のペアとする。 $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}$ とし、 $z = T(x) = [\phi_1(x) \dots \phi_{n_z}(x)]^T \in \mathbb{R}^{n_z}$ とすると、KCT は次のように定義される。

$$x = Cz \quad (4.2a)$$

$$\dot{z} = Fz + \sum_{i=1}^m L_{g_i} T(x) u_i \quad (4.2b)$$

ここで、 $F \in \mathbb{R}^{n_z \times n_z}$ は Koopman 固有値 λ_i を (i, i) 成分とした対角行列である。また、(4.2a) において、元のシステム (4.1) の状態 x は、Koopman 固有関数の線形結合によって表現されていることに注意されたい。

この時、Koopman 固有関数 $\phi_i, i = 1, \dots, n_z$ によって張られる空間が、 $L_{g_i}, i = 1, \dots, m$ の不変部分空間を形成するならば、入力アフィンシステム (4.1) は、 n_z 次元空間で双線形化可能である。結果として、 $L_{g_i} T(x) = G_i z, G_i \in \mathbb{R}^{n_z \times n_z}$ となり、入力アフィンシステムは、次の Koopman 双線型形式 (Koopman Bilinear Form: KBF) で表現することができる (詳細は [53] を参照されたい)。

$$\dot{z} = Fz + \sum_{i=1}^m G_i z u_i \quad (4.3)$$

4.3 拡大双線形モデルの同定

対象システム (4.1) が未知である場合に, KBF で表される拡大双線形モデル (4.3) を数値的に同定する方法を説明する. まず, 対象システムの観測データとして, 状態・入力のデータセット

$$\mathcal{D} = \{x_k, u_k\}, \quad k = 0, \dots, M \quad (4.4)$$

が得られているとする. ここで, サンプル時間を Δt とする. なお, データ取得実験においては, 対象システムが不安定な場合, 簡易的に設計したコントローラを適用してデータを取得してもよい. また, データセット $\mathcal{D}$ は, 独立した実験により取得した複数のデータセットを連結したものとしてもよい. その場合は, 後述する Z_+ , Z_u を構成する際に, 異なるデータセットにまたがってデータの対応付けをしないように注意が必要である.

次に, 観測関数ベクトルを $h(x) \in \mathbb{R}^{n_z}$ とする. ここで, 観測関数は, 物理モデルに関する知識があれば, それに基づき選定してもよい. より一般的には, 状態に関するある次数までの単項式など, 一般的な基底関数を選定する方法もある.

ここで, (4.2a) を成立させるための単純な方法としては, 元の状態を観測関数に含めれば良い. 元の状態以外の観測関数を並べたベクトルを $\eta(x) \in \mathbb{R}^{n_\eta}$ として, 観測関数ベクトルを次の形で仮定する.

$$h(x) = \begin{bmatrix} x \\ \eta(x) \end{bmatrix} \quad (4.5)$$

今, z_k を拡大状態とし, 次の離散時間の拡大双線形モデルを考える.

$$z_{k+1} = Az_k + \sum_{i=1}^m B_i z_k u_k(i) \quad (4.6a)$$

$$x_k = Cz_k \quad (4.6b)$$

ここで, $u_k(i)$ は u_k の i 番目の成分である. データセット $\mathcal{D}$ が与えられた時, $z_k = h(x_k)$, $k = 0, \dots, M$ として, (4.6a) の各係数行列 $A, B_1, \dots, B_m$ は, 次の最適化問題を解いて求めることができる.

$$\min_{A, B_1, \dots, B_m} \sum_{k=0}^{M-1} \left\| z_{k+1} - \left(Az_k + \sum_{i=1}^m B_i z_k u_k(i) \right) \right\|^2 \quad (4.7)$$

最適化問題 (4.7) は, サンプル時刻間の予測誤差を最小化するように係数行列を決定する問題である. この解は, 最小二乗法によって求めることができる. $H = [A \ B_1 \ \dots \ B_m] \in \mathbb{R}^{n_z \times n_z(m+1)}$ とする. さらに,

$$Z_+ = [z_2, \dots, z_M], \quad Z_u = \begin{bmatrix} z_1 & \cdots & z_{M-1} \\ z_1 u_1(1) & \cdots & z_{M-1} u_{M-1}(1) \\ \vdots & \ddots & \vdots \\ z_1 u_1(m) & \cdots & z_{M-1} u_{M-1}(m) \end{bmatrix} \quad (4.8)$$

とする．この時， Z_+ と HZ_u の各列は各サンプル時刻における (4.6a) の左辺と右辺の関係にある．したがって， H の推定値は次式で計算することができる．

$$\hat{H} = Z_+ Z_u^T (Z_u Z_u^T)^{-1} \quad (4.9)$$

また，(4.5) より，(4.6b) の行列 C は次のように得られる．

$$C = \begin{bmatrix} I_n & O_{n \times n_\eta} \end{bmatrix} \quad (4.10)$$

ここでは，選定された観測関数に基づいてデータから拡大双線形モデルを同定する手法を説明したが，NN を活用して観測関数と拡大双線形モデルを同時に学習する手法も提案されている [55]．

また，上記の定式化では (4.6) において入力に関する線形項を考慮していないが，後述する数値例では非線形観測関数ベクトル $\eta(x)$ に定数項を含めることにより，実質的に入力に関する線形項を考慮している．入力に関する線形項を陽に考慮するための定式化については，[26] を参照されたい．

4.4 拡大双線形モデルを用いた MPC の予測誤差補償

本節では，前節の方法によって得られる拡大双線形モデルを予測モデルとした MPC を考える．また，後述の LMPC を構築するための準備として，予測誤差補償を導入する．

4.4.1 拡大双線形モデルを用いた MPC の定式化

ここではまず，NMPC としての定式化を考え，各時刻で解く最適化問題を次のように構成する．

$$\underset{u_k}{\text{minimize}} \quad J_1 \quad (4.11a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (4.11b)$$

$$z_{k+1} = Az_k + \sum_{i=1}^m B_i z_k u_k(i), \quad k = 0, \dots, N_p - 1 \quad (4.11c)$$

$$x_k = Cz_k, \quad k = 0, \dots, N_p \quad (4.11d)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (4.11e)$$

ここで， N_p は予測ホライズン， x_t は時刻 t において観測された状態とする． A_{con}, b_{con} は状態と入力に関する線形制約条件を構成する行列およびベクトルである．また，評価関数 J_1 は次式で与えられる．

$$J_1 = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2)$$

ここで, $P, Q \in \mathbb{R}^{n \times n}$ は準正定, $R \in \mathbb{R}^{m \times m}$ は正定な重み行列であり, $x_k^r, k = 0, \dots, N_p$ と $u_k^r, k = 0, \dots, N_p - 1$ はそれぞれ予測ホライズン上の目標状態と目標入力とする.

最適化問題 (4.11) では, 観測した現時刻の状態 x_t を観測関数ベクトルにより拡大状態空間に写像し, 予測ホライズン上の拡大状態の変化を拡大双線形モデルで予測する. さらに, 予測した各ステップの拡大状態を元の状態空間に射影して評価することで, 評価関数を最小化する最適入力列 $u_k, k = 0, \dots, N_p - 1$ を求める.

4.4.2 予測誤差補償

対象の入力アフィンスシステムに対し適切に同定された拡大双線形モデルは, 対象システムの振る舞いを比較的高精度に予測可能であると期待できる. しかし本研究では, 次節で詳細を述べるように, LMPC を構築するために拡大双線形モデルを線形時変モデルとしてさらに近似する. この近似によって, 予測誤差が増大する可能性がある.

この予測誤差の影響を補償するため, ここでは 3 章の提案手法と同様に, 最適化問題 (4.11) に対し, 拡大状態に関する制約条件と, 補償入力項を導入する. これらを考慮した最適化問題を次に示す.

$$\underset{u_k, \Delta u_k}{\text{minimize}} \quad J_2 \quad (4.12a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (4.12b)$$

$$z_{k+1} = Az_k + \sum_{i=1}^m B_i z_k u_k(i) + \begin{bmatrix} O_{n \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \quad (4.12c)$$

$$x_k = Cz_k, \quad k = 0, \dots, N_p \quad (4.12d)$$

$$z_k(n+1:n_z) = \eta(z_k(1:n)), \quad k = 1, \dots, N_p \quad (4.12e)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (4.12f)$$

ここで, $\Delta u_k \in \mathbb{R}^{n_\eta}$ は各予測ステップ k における $n+1$ から n_z 番目の状態に対応した補償入力であり, $z_k(i:j)$ は, Matlab 表記を用いて z_k の i から j 番目の要素のベクトルとする. また, 評価関数 J_2 は, Δu_k に関する重み行列を $R_\Delta \in \mathbb{R}^{n_\eta \times n_\eta}$ として次式で与えられる.

$$J_2 = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2 + \|\Delta u_k\|_{R_\Delta}^2)$$

最適化問題 (4.12) では, (4.12e) によって, 拡大状態のうち元の状態に対応する $z_k(1:n)$ と, 非線形な観測関数 $\eta(x)$ によって拡大した部分に対応する $z_k(n+1:n_z)$ が, 各ステップにおいて観測関数の関係を満たすことを考慮している. この最適化問題を解くことにより, 各ステップにおける拡大状態の要素が制約条件を満たすように最適な補償入力項が計算され, 拡大モデルによる予測誤差を補償することができる. z_k の予測精度の改善は x_k の予測精度の改善につながるため, MPC の制御性能の向上が期待できる.

ここで、補償入力 Δu_k は評価関数によってできるだけ小さくなるように計算されるため、状態軌道の最適化は、できるだけ本来の入力を用いて実現されると期待できる。また、拡大双線形モデルが元の非線形モデルを精度良く近似できていれば、補償入力 Δu_k の値は 0 に近づくと期待できる。

なお、(4.12c) は拡大双線形モデルに基づく予測モデルであるが、これは拡大線形モデルを包含している。実際に、 $\eta(x)$ の成分が定数を含み、 $B_i, \forall i$ のうち定数以外に対応する要素が全て 0 である場合、双線形項は消えて入力に関しても線形なモデルとなる。したがって、提案の予測誤差補償を考慮した MPC は、予測モデルが拡大線形モデルの場合にも適用することができる。

4.5 拡大双線形モデルを用いた LMPC

本節では、前節の予測誤差補償を考慮した MPC に基づき、事前予測状態を活用した線形化を行い、LMPC アルゴリズムを構築する。

4.5.1 事前予測状態を用いた QP 問題の定式化

最適化問題 (4.12) は、拡大双線形モデル (4.12c) に加え、拡大状態に関する非線形制約条件 (4.12e) を含むため、非線形最適化問題である。これに基づいて LMPC を定式化するため、3 章の提案手法と同様に、予測ホライゾン上の事前予測状態を用いて、拡大双線形モデル (4.12c) および非線形制約条件 (4.12e) を線形化することを考える。

まず、予測ホライゾンの k ステップ目における事前予測状態を z_k^* , $k = 0, \dots, N_p$ とする。これらに対応した元の状態の事前予測値は $x_k^* = Cz_k^*$, $k = 0, \dots, N_p$ として得ることができる。この時、予測のための拡大双線形モデルは、事前予測状態を用いて、次のように予測ホライゾン上での線形時変システムとみなすことができる。

$$\begin{aligned} z_{k+1} &= Az_k + \sum_{i=1}^m B_i z_k^* u_k(i) \\ &= Az_k + \tilde{B}(z_k^*) u_k \end{aligned} \quad (4.13)$$

ここで、 $\tilde{B}(z_k^*) = [B_1 z_k^* \dots B_m z_k^*]$ である。

次に、予測ホライゾンの k ステップ目における拡大状態に関する非線形制約条件 (4.12e) は、元の状態を事前予測状態の近傍で線形近似することにより、次のように線形制約条件で表すことができる。

$$z_k(n+1:n_z) = \eta(x_k^*) + \left. \frac{\partial \eta}{\partial x} \right|_{x=x_k^*} (z_k(1:n) - x_k^*) \quad (4.14)$$

以上より、事前予測状態 z_k^* , $k = 0, \dots, N_p$ が得られているものとしたとき、LMPC の各制御周期において解くべき最適化問題は次のように与えられる。

$$\underset{u_k, \Delta u_k}{\text{minimize}} \quad J_2 \quad (4.15a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (4.15b)$$

$$x_k^* = Pz_k^*, \quad k = 0, \dots, N_p \quad (4.15c)$$

$$z_{k+1} = Az_k + \tilde{B}(z_k^*)u_k + \begin{bmatrix} O_{n \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \quad (4.15d)$$

$$x_k = Cz_k, \quad k = 0, \dots, N_p \quad (4.15e)$$

$$z_k(n+1:n_z) = \eta(x_k^*) + \left. \frac{\partial \eta}{\partial x} \right|_{x=x_k^*} (z_k(1:n) - x_k^*), \quad k = 1, \dots, N_p \quad (4.15f)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (4.15g)$$

最適化問題 (4.15) は、初期ステップにおける拡大状態 z_0 を求める (4.15b) を除き、二次形式の評価関数と線形の制約条件で構成されており、高速な QP ソルバで解くことが可能な定式化となっている。

4.5.2 事前予測状態の取得

事前予測状態 z_k^* の取得手順は、3.4.2 項で説明した内容と同様である。まず、ある時刻 t において最適化問題 (4.15) を解いて得られた最適入力列 $u_k, \Delta u_k$ と、双線形予測モデル (4.15d) を用いて、各ステップにおける拡大状態の最適値 z_k^{*-} , $k = 0, \dots, N_p$ を計算することができる。また、制御周期が予測モデル (4.15d) のサンプル時間 Δt と一致していると仮定する。この時、事前予測状態 z_k^* は次のようにして取得することができる。

$$\begin{aligned} z_k^* &= z_{k+1}^{*-}, \quad k = 0, \dots, N_p - 1 \\ z_{N_p}^* &= z_{N_p}^{*-} \end{aligned} \quad (4.16)$$

また、制御ループの初期時刻においては、対象システムの初期状態を x_{t_0} として、事前予測状態を次のように取得することができる。

$$z_k^* = h(x_{t_0}), \quad k = 0, \dots, N_p \quad (4.17)$$

4.5.3 LMPC アルゴリズム

以上の議論より，拡大双線形モデルを用いた LMPC アルゴリズムは Algorithm 2 のようにまとめられる．

Algorithm 2 LMPC based on lifted bilinear model

Require: Reference trajectory (x^r, u^r)

while Controller is running **do**

if $t=0$ **then**

 Set the prior-predicted states z_k^* by (4.17)

end if

 Observe current state x_t of the system

 Solve the optimization problem (4.15) with the prior-predicted states z_k^*

 Update the prior-predicted states z_k^* by (4.16)

 Apply first input u_0 to the system

end while

本提案手法によれば，拡大双線形モデルに基づき近似した線形時変予測モデルに対し，予測誤差の補償と，補償結果によるモデル精度向上を同時に行うことができる．すなわち，ある時刻において事前予測状態に誤差があり，拡大双線形モデルに対して線形時変予測モデル (4.13) の近似誤差が大きいとしても，拡大状態に関する制約条件 (4.14) を満たすように補償入力 Δu_k を加えることで，発生する予測誤差を低減できる．そして，誤差が低減された状態予測に基づき最適状態列を求め，次の時刻の事前予測状態を構成することで，次の時刻における線形時変予測モデル (4.13) の精度を改善するとともに，拡大状態に関する制約条件 (4.14) の近似精度も改善される．このような反復的な精度向上効果によって，提案の LMPC アルゴリズムは，高速な求解が可能な QP 問題の形式でありながら，非線形性を考慮して高い性能を達成できる．

また，3 章の提案手法に対する違いは，3 章では既知の物理モデルに基づく近似拡大線形モデルを利用しているのに対し，本提案手法では未知の対象システムに対してデータドリブンに同定した拡大双線形モデルを利用している点である．すなわち，本提案手法は，対象システムに関する事前知識がない場合においても適用できることが利点である．

4.6 数値シミュレーション

本節では、2次元平面上で動作するドローンモデルを対象システムとして、数値シミュレーションにより提案手法の有効性を評価した結果を説明する。

4.6.1 対象システム

対象システムとして、Fig. 4.1 に示す 2次元平面上で動作するドローンモデルを考える。

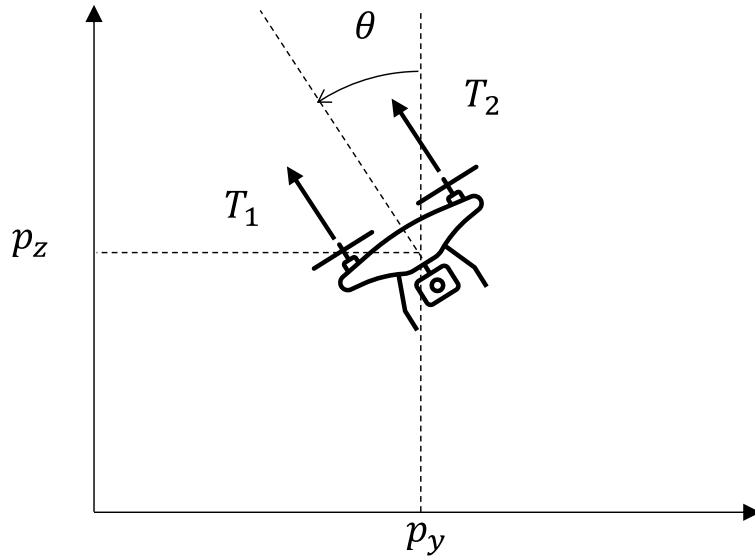


Fig. 4.1: Planar drone

対象システムは非線形入力アフィンシステムであり、そのダイナミクスは次式で与えられる。

$$\frac{d}{dt} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} = \begin{bmatrix} v_y \\ v_z \\ \omega \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta \\ \frac{1}{m} \cos \theta & \frac{1}{m} \cos \theta \\ -\frac{l_a}{I_{xx}} & \frac{l_a}{I_{xx}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (4.18)$$

ここで、 p_y, p_z は y, z 座標 [m]、 θ は角度 [rad]、 v_y, v_z は y, z 軸方向の速度 [m/s]、 ω は角速度 [rad/s]、 T_1, T_2 は左右のプロペラのスラスト力 [N] である。このシステムにおいて、状態を $x = [p_y \ p_z \ \theta \ v_y \ v_z \ \omega]^T$ 、入力を $u = [T_1 \ T_2]^T$ とする。また、物理パラメータを Table 4.1 に示す。

Table 4.1: Physical parameters of the target system

Parameter	Variable	Value
Body mass	m	2.0
Rotational inertia	I_{xx}	1.0
Arm length to propeller axis	l_a	0.2
Acceleration of gravity	g	9.81

以降では、まず対象システム (4.18) に対し、拡大双線形モデルを同定する．そして得られたモデルに基づき、本章で提案した LMPC アルゴリズム 2 を適用し、その性能を評価する．比較対象の制御手法として、真のシステムを予測モデルとした NMPC(以下, NMPC) と、平衡点近傍の線形近似モデルを予測モデルとした LMPC (以下, ノミナル LMPC) の 2 つを考える．

ここで、ノミナル LMPC における線形近似モデルは、平衡状態 $\bar{x} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ ，平衡入力 $\bar{u} = [\frac{1}{2}mg \ \frac{1}{2}mg]^T$ 近傍の一次のテーラー展開より、次式で得られる．

$$\dot{x} = Ax + Bu + W \quad (4.19)$$

ここで、

$$A = \begin{bmatrix} O_{3 \times 3} & I_3 \\ \begin{bmatrix} 0 & 0 & -g \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & O_{3 \times 3} \end{bmatrix}, \quad B = \begin{bmatrix} O_{3 \times 2} \\ \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ -\frac{l_a}{I_{xx}} & \frac{l_a}{I_{xx}} \end{bmatrix} \end{bmatrix}, \quad W = \begin{bmatrix} O_{3 \times 1} \\ \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} \end{bmatrix}$$

である．

本評価においては、特に対象システムの姿勢が大きく傾いた非線形領域における制御性能に差が現れると期待する．すなわち、非線形領域において線形近似モデルでは予測誤差が大きくなるためノミナル LMPC の性能は悪化するが、提案手法では非線形性を捉えた拡大双線形モデルを用いることで、より高い性能を発揮できると考えられる．

4.6.2 拡大双線形モデルの同定

拡大双線形モデルの同定にあたり、対象システム (4.18) はそのままでは不安定であるため、原点への安定化コントローラを適用して同定用データを取得した．安定化コントローラとして、ノミナルモデル (4.19) を用いて LQR を設計した．そして、データ取得のためのサンプル時間を 0.05[s] とし、100 点のランダムな初期状態から原点に安定化させる動作を行った．ここで、初期値は $x_0 \sim \mathcal{N}(O_{6 \times 1}, \sigma_0)$, $\sigma_0 = \text{diag}(1, 1, 0.1, 0, 0, 0)$ とした．また、信号を励起させるため、安定化フィードバック入力 $u_{st} = -Kx$ にノイズ信号 u_v を加え、対象システムに入

力した. ここで, K はノミナルモデルに基づく LQR ゲインである. また, $u_v \sim \mathcal{N}(O_{2 \times 1}, I_2)$ とした.

観測関数ベクトル $h(x) = [x^T \eta^T(x)]^T$ については, 対象システム (4.18) のダイナミクスに含まれる状態の非線形要素を考慮し, $\eta(x) = [\cos \theta \ \sin \theta \ 1]^T$ と選定した.

また, 今回は $\theta = 0$ 近傍での不可制御性を回避するために, 入力を $u = \bar{u} + \Delta u$ と分解し, 平衡入力 $\bar{u}$ からの差分 Δu を新たな入力と見なして, 拡大双線形モデルを同定した. 入力の選び方の違いによる可制御性への影響に関する考察は, Appendix B を参照されたい.

以上の条件のもと同定した拡大双線形モデルに対し, テストデータを用いて予測精度を確認した結果を Fig. 4.2 に示す. テストデータでは, 同定用データと異なる 30 点のランダムな初期値を使用し, 各軌道について 2[s] のシミュレーションを行った. 本結果より, 多くの軌道について真値に近い予測値が得られており, 対象システムの非線形性をよく捉えられていると考えられる. なお, ここでは拡大双線形モデルを線形近似せず, 双線形項を含むモデルを用いて予測を行っていることに注意されたい.

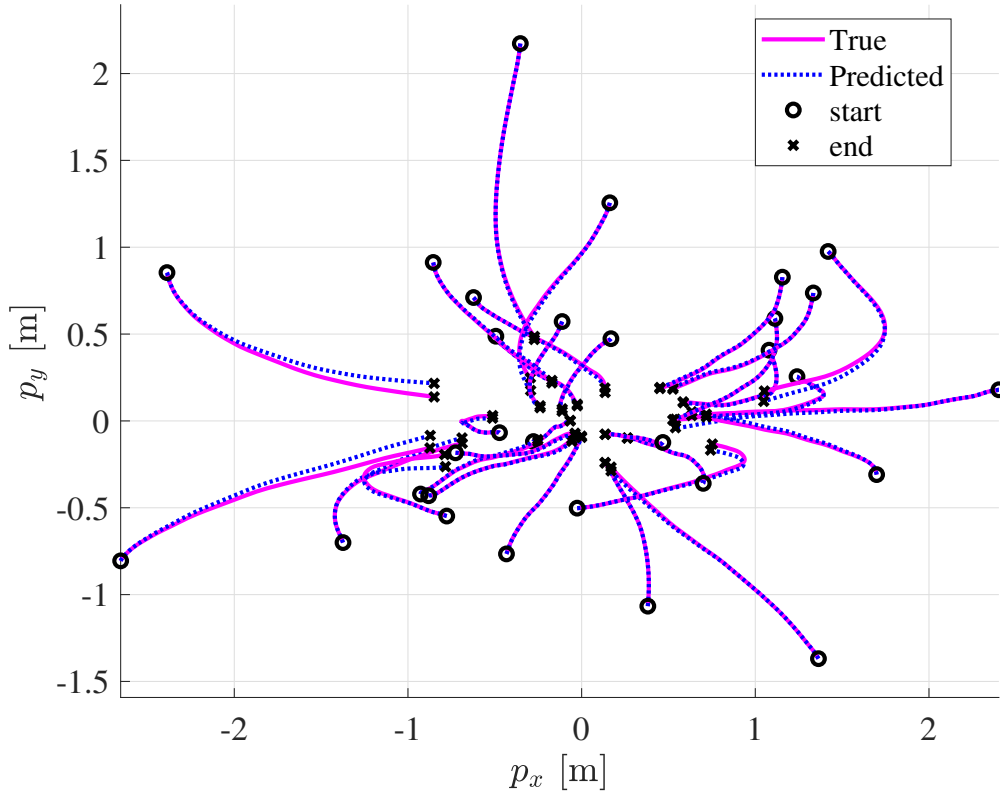


Fig. 4.2: Prediction by the identified model with test data

4.6.3 制御性能の評価結果

状態制約なしの場合

同定により得られた拡大双線形モデルを用いて、提案手法の評価を行った。まず、状態に関する制約条件なしの場合のシミュレーション結果を示す。

今回、初期状態を $[-15, 0, 0, 0, 0, 0]^T$ とし、目標状態を原点として、ドローンを水平移動させる制御を行った。シミュレーション時間は $10[s]$ 、サンプル時間は $\Delta t = 0.05[s]$ 、予測ホライズンは $N_p = 30$ 、評価関数 J_2 における重み行列は、 $P = 100 \cdot I_6$ 、 $Q = I_6$ 、 $R = 0.1 \cdot I_2$ 、 $\Delta R = 0.001 \cdot I_3$ とした。また、入力に関する制約条件を $T_1, T_2 \geq 0$ とした。なお、今回はノイズの影響は考慮していない。

また、NMPC のソルバには CasADi [61] をインターフェースとして IPOPT を使用し、提案手法およびノミナル MPC のソルバには MATLAB の quadprog 関数を使用した。シミュレーション実行環境には Core i5 CPU@1.20GHz, 24.0GB メモリ, Windows 10 の PC 上で動作する MATLAB 2020b を使用した。

Fig. 4.3, Fig. 4.4 に、各手法による状態と入力の比較結果をそれぞれ示す。なお、図中の赤色の破線は制約条件を示す。また、ここでは比較のため、提案手法における予測誤差補償を行わない場合の結果についても示している。

まず Fig. 4.3 に示す状態について、どの手法によっても原点へ安定化できている。ただし、線形近似モデルを用いたノミナル LMPC は、NMPC と比較すると大きく異なる振る舞いを示しているのに対し、提案手法では NMPC とほとんど同様の応答が得られていることが分かる。また、提案手法において、予測誤差補償なしの場合よりも、予測誤差補償ありの方が、より NMPC に近い応答となっていることが分かる。

また、Fig. 4.4 に示す入力についても同様の傾向がみられる。ただし、シミュレーション開始直後の時間帯においては、提案手法においても NMPC に対して若干の応答の違いが現れていることが分かる。

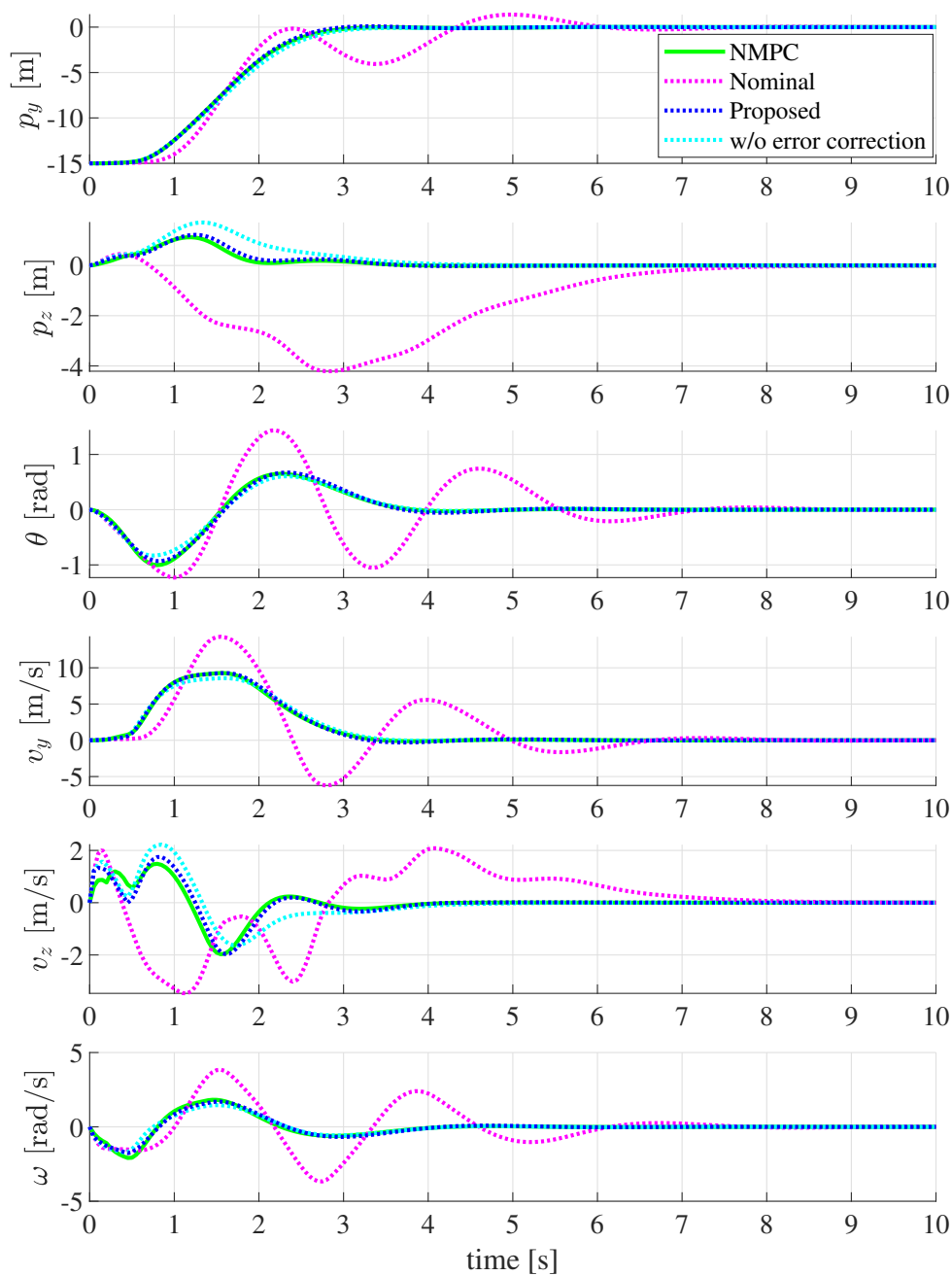


Fig. 4.3: States (without state constraint)

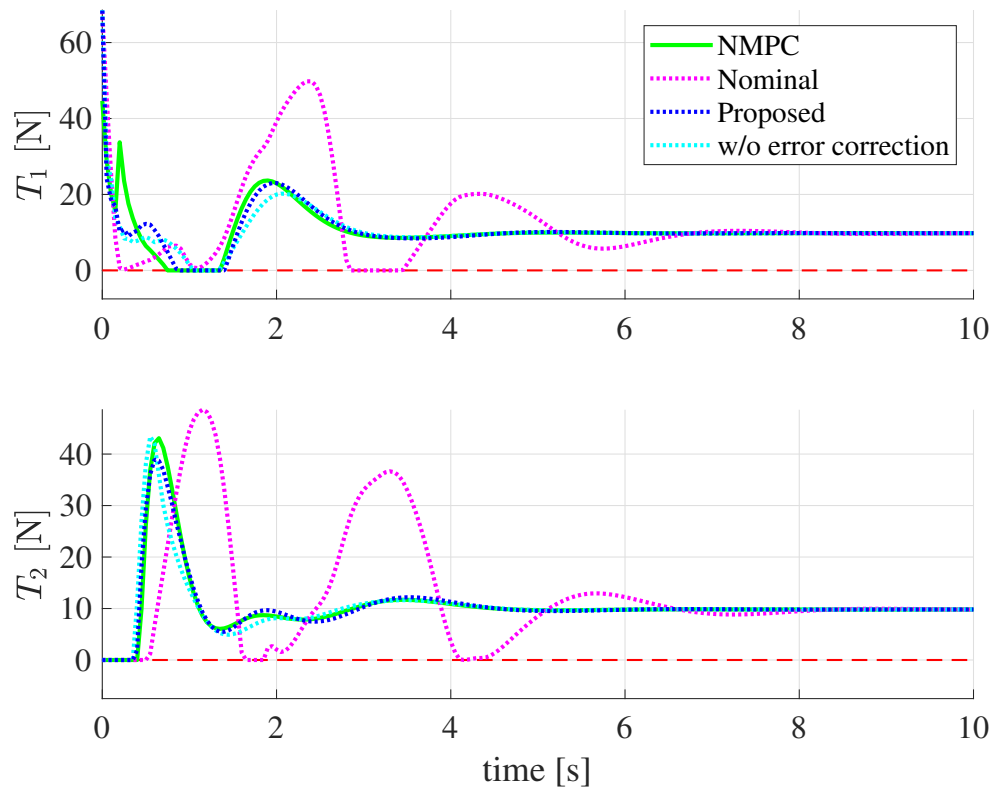


Fig. 4.4: Inputs (without state constraint)

ここで、提案手法における補償入力グラフを Fig. 4.5 に示す。このグラフより、シミュレーション開始直後、特に 1[s] 程度までの時間においては、補償入力の絶対値が相対的に大きくなっているが、それ以降は減少し、3[s] 以降ではほとんど 0 に収束していることが分かる。これは、シミュレーション開始時には事前予測状態の誤差が大きく、それによって拡大双線形モデルを近似した線形時変モデルによる予測誤差が大きいことにより、拡大状態に関する制約条件を満たすための補償入力計算されているためと考えられる。そして、反復計算によって事前予測状態の精度が向上すると、予測モデルの精度が改善され、補償入力が必要なくなって 0 に近づいてく。

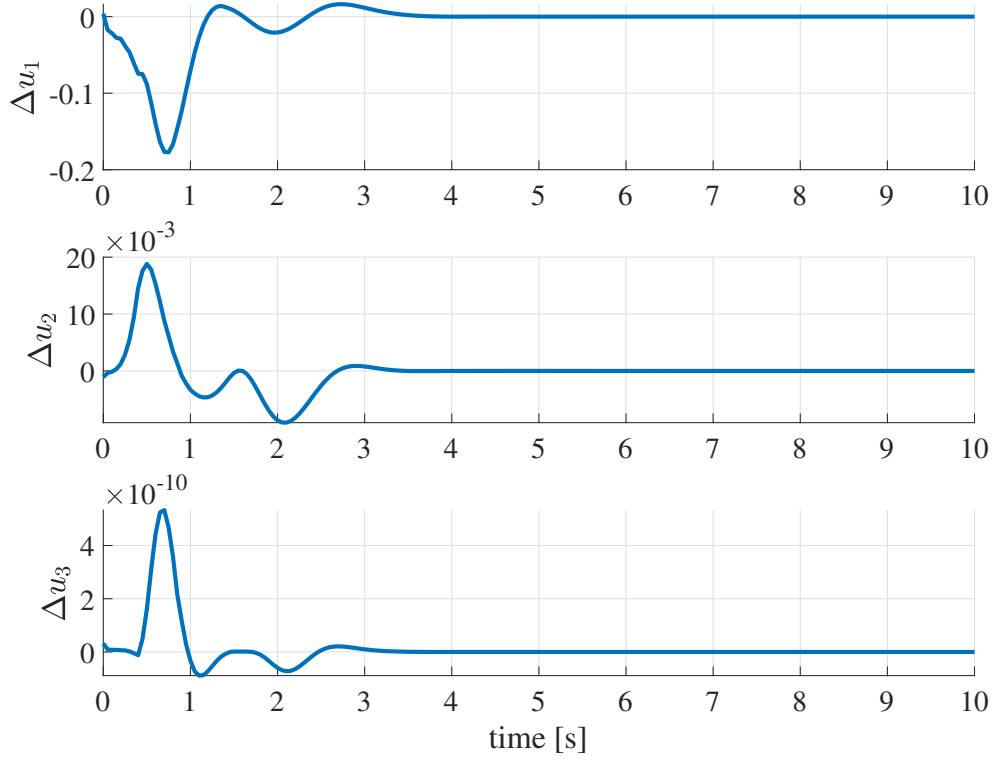


Fig. 4.5: Corrective inputs (without state constraint)

また，次の評価関数により各手法の性能を定量的に評価した．

$$J_e = \sum_{k=0}^N (\|x_k - \bar{x}\|_Q^2 + \|u_k - \bar{u}\|_R^2) \quad (4.20)$$

ここで， N は 10[s] のシミュレーションで取得したデータ点数である．NMPC の値を 1 とし て正規化した評価結果を Fig. 4.6 に示す．この結果より，ノミナル LMPC では NMPC よりも性能が 90% 低下するのに対し，提案手法では NMPC とほぼ同等の制御性能を発揮できることが示された．

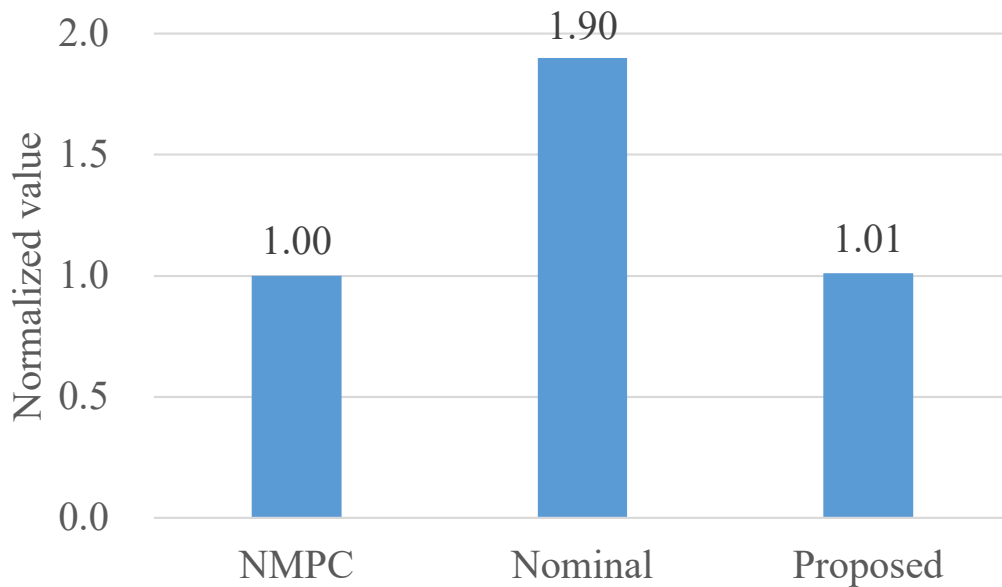


Fig. 4.6: Normalized objective function values (without state constraint)

状態制約ありの場合

次に、状態に関する制約条件として、対象システムの高度を限定し、 $|p_z| < 0.1$ とした場合のシミュレーション結果を示す。なお、今回は初期状態を $[-10, 0, 0, 0, 0, 0]^T$ とし、その他のシミュレーション条件は前述の場合と同様とした。

Fig. 4.7, Fig. 4.8 に、各手法による状態と入力の比較結果をそれぞれ示す。状態制約がある場合においても、制約がない場合と同様に、ノミナル LMPC と比べて提案手法の方が NMPC に近い応答が得られている。ノミナル LMPC と提案手法の特に大きな違いとして、ノミナル LMPC では p_z に関する制約条件に対してたびたび違反が見られるが、提案手法では制約条件を満足できている。これは、提案手法の方が状態の予測精度が優れているためであると考えられる。

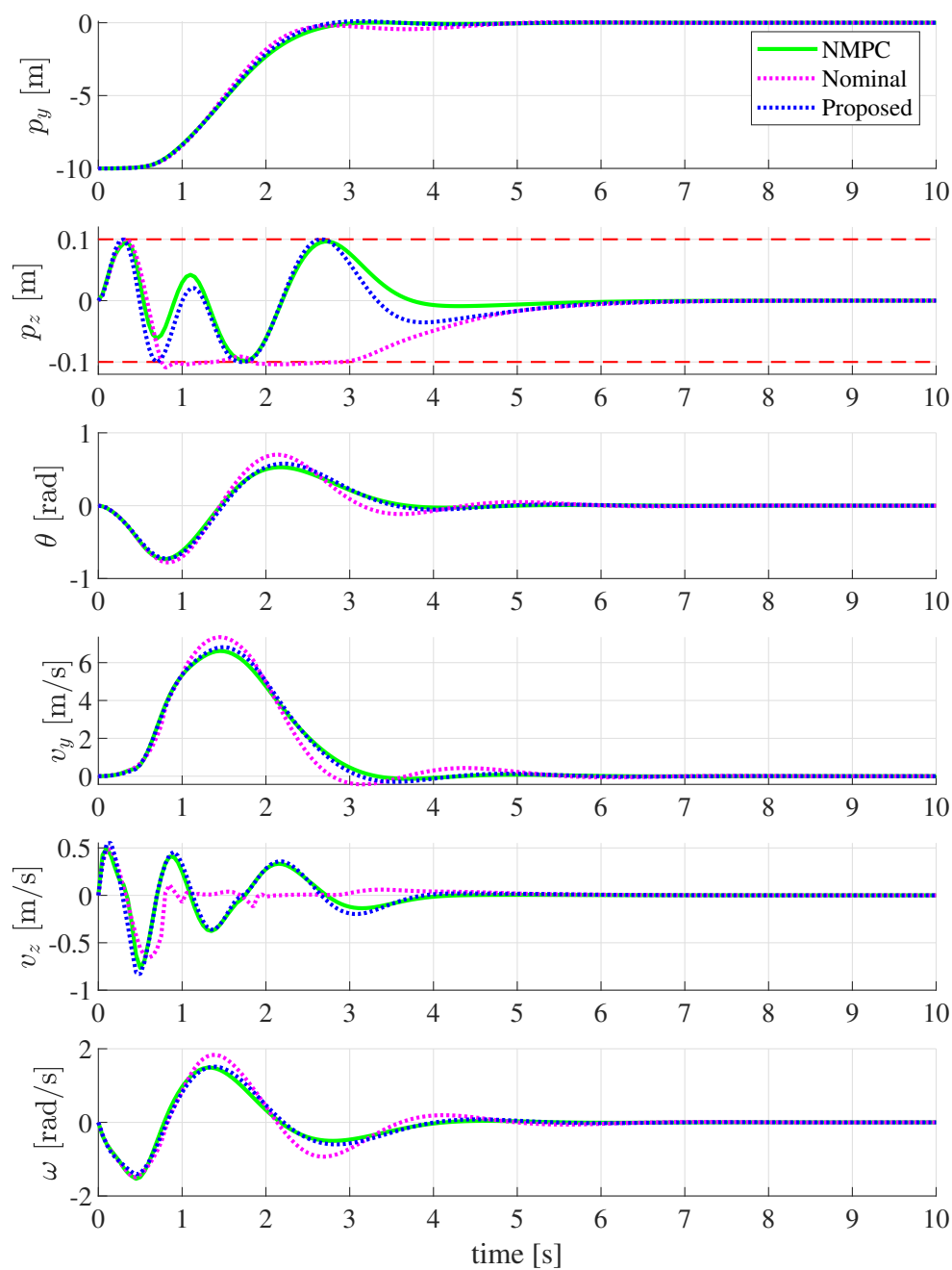


Fig. 4.7: States (with state constraint)

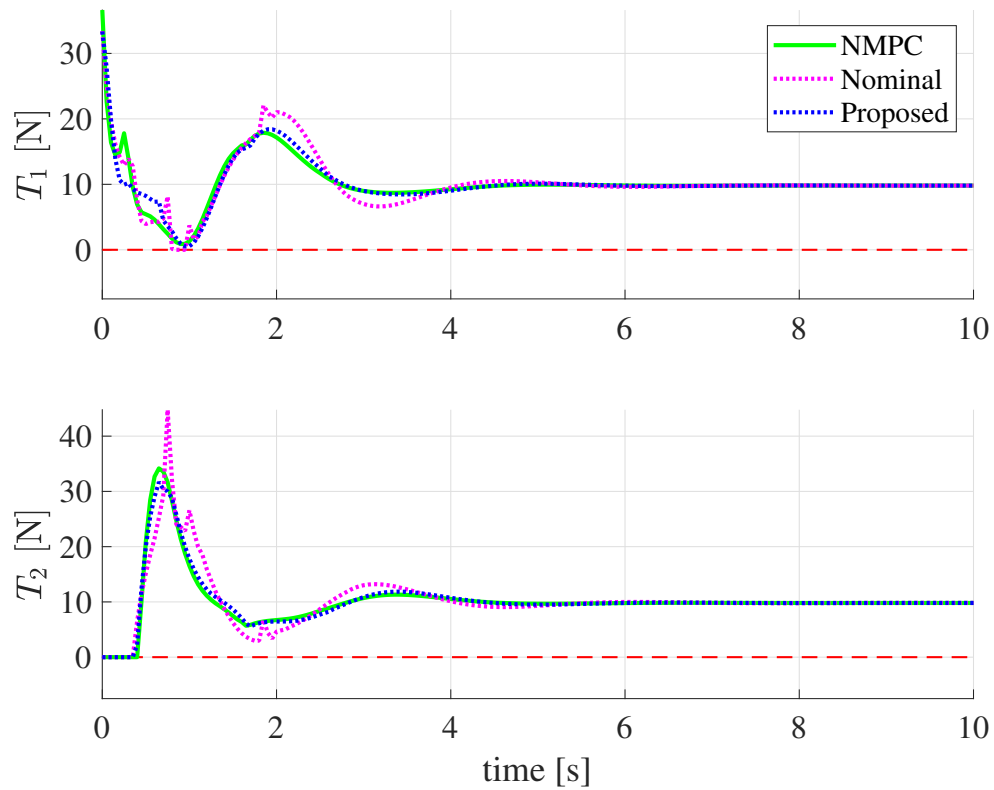


Fig. 4.8: Inputs (with state constraint)

また，提案手法における補償入力グラフを Fig. 4.9 に示す．補償入力についても，状態制約がない場合と同様に，シミュレーション開始直後は相対的に大きな値をとっているが，3[s] 以降ではほとんど 0 に収束しており，事前予測状態の誤差が大きい間は拡大状態の予測誤差を補償し，予測モデルの精度が改善されるにつれ値が小さくなっているものと考えられる．

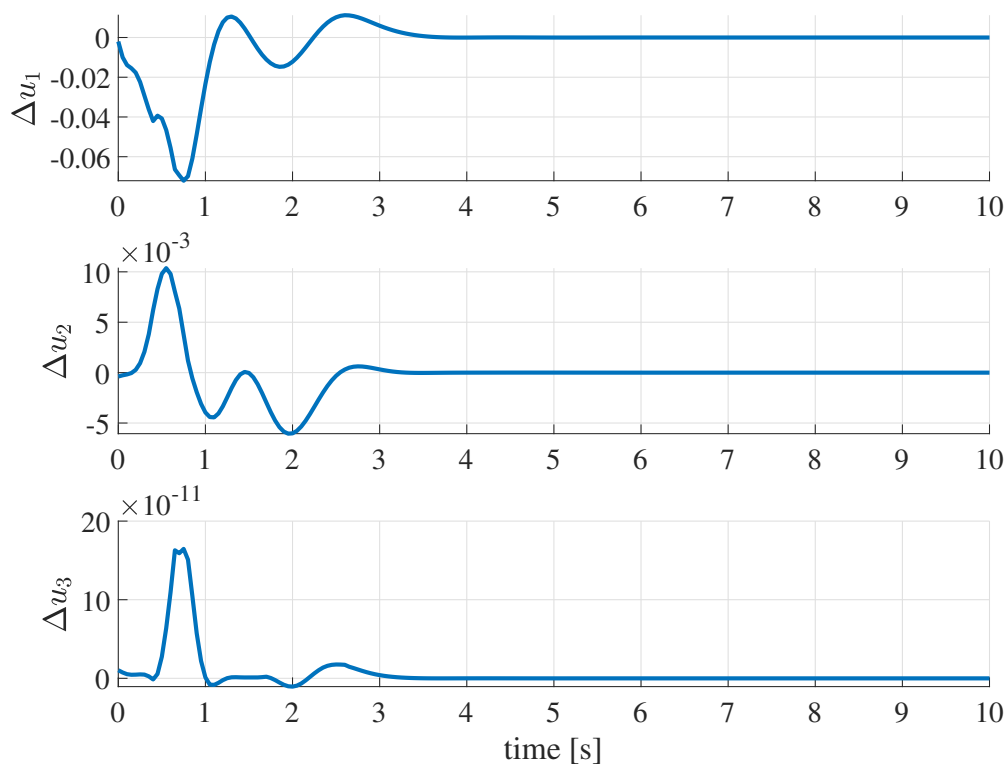


Fig. 4.9: Corrective inputs (with state constraint)

NMPC を 1 として正規化した評価関数 (4.20) の値を, Fig. 4.10 に示す. 今回は状態制約によって対象システムの動作範囲を限定したため, ノミナル LMPC においても状態制約なしの場合ほどの大きな性能低下は無かった. しかしこの場合においても, 提案手法の方がノミナル LMPC よりもさらに優れており, NMPC とほぼ同等の制御性能を発揮できることが示された.

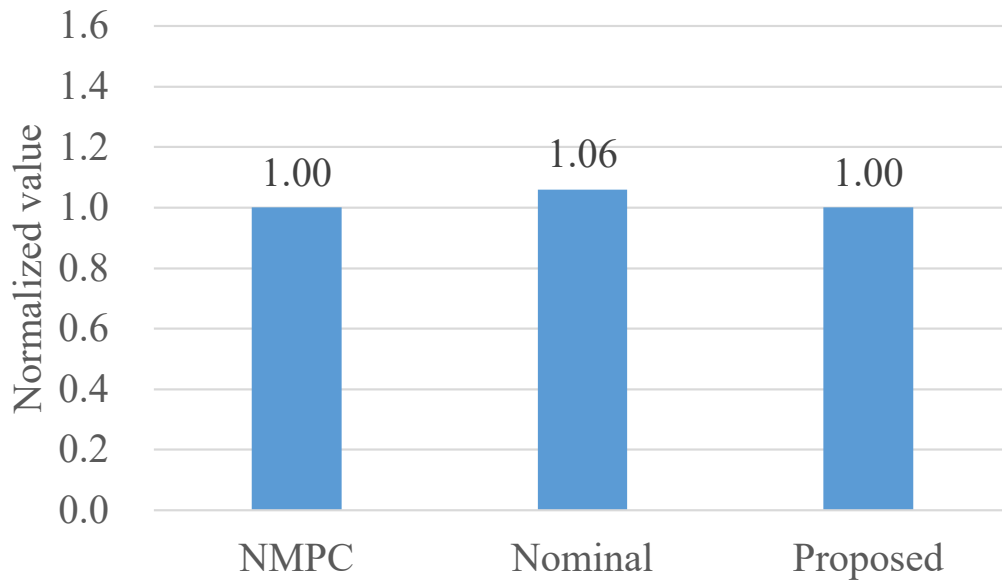
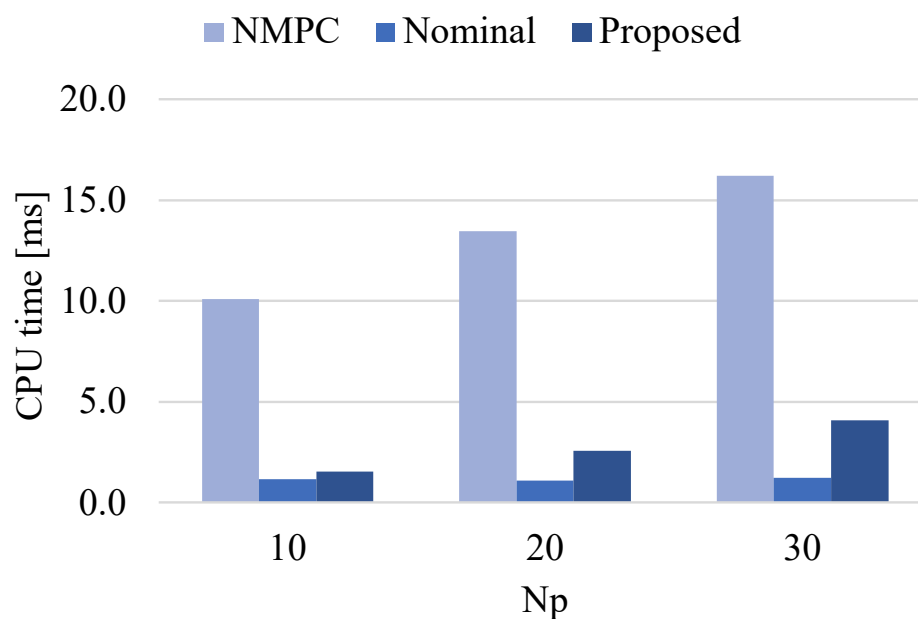


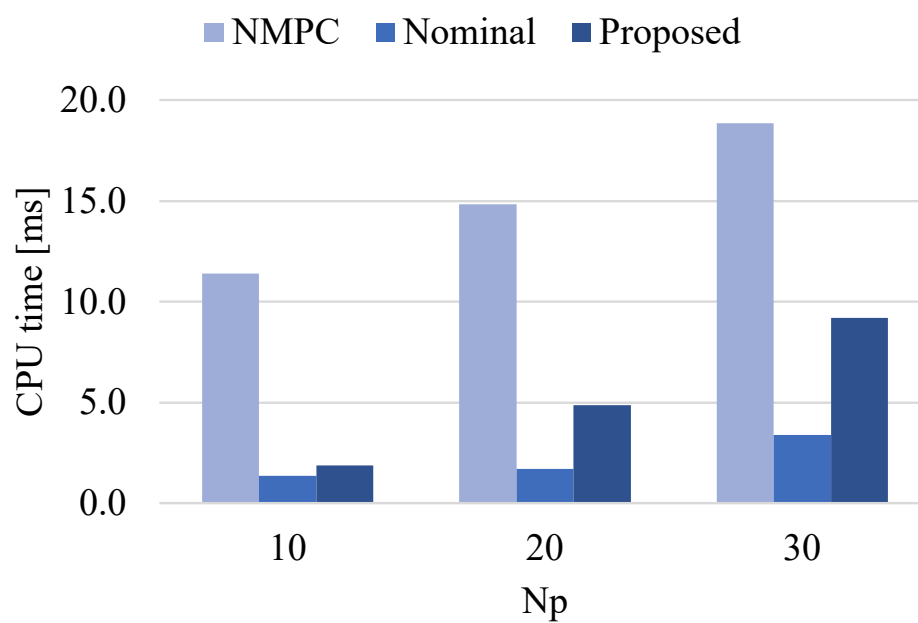
Fig. 4.10: Normalized objective function values (with state constraint)

4.6.4 計算時間の評価結果

最後に、計算時間の評価結果について述べる．Fig. 4.11 に、予測ホライズン $N_p = 10, 20, 30$ の場合の、各手法の最適化ソルバの実行時間を示す．Fig. 4.11(a) と Fig. 4.11(b) は、状態制約なしの場合と制約条件ありの場合の評価結果をそれぞれ示している．



(a) Without state constraint



(b) With state constraint

Fig. 4.11: CPU time

この結果より、状態制約の有無に関わらず、提案手法では NMPC よりも大幅に計算時間を短縮できていることが分かる。一方、ノミナル LMPC と比較すると提案手法の方が計算時間は長くなっている。この理由は、提案手法の方がモデルの次数が大きいことと、拡大状態に関する制約条件を課していることが影響していると考えられる。

以上の結果より、提案手法では NMPC と同等の制御性能を持ちながら、線形化によって NMPC よりも低計算コストで実行可能であることを示し、提案手法の有効性を示すことができた。

4.7 まとめ

本章では、未知の入力アフィンシステムを対象とし、データに基づき拡大双線形モデルを同定し、それを活用して高い予測精度と低計算コストを両立する LMPC を提案した。拡大双線形モデルは、対象システムの状態と入力データのデータセットに基づき、最小二乗法によって数値的に同定することができる。提案手法では、そのようにして求めた拡大双線形モデルを予測モデルとした MPC において、拡大状態に関する制約条件を考慮した予測誤差補償と、事前予測状態を活用した非線形要素の線形化を行うことで、高精度な予測が可能な LMPC を実現した。

数値シミュレーションでは、2次元平面上で動作するドローンモデルを対象システムとして、提案手法の有効性検証を行った。その結果、真のモデルを用いた NMPC と同等の制御性能を達成し、計算時間も大幅に低減可能であるという提案手法の有効性を示すことができた。

第5章 拡大双線形モデル誤差モデルを用いた線形モデル予測制御

5.1 はじめに

本章では，4章の提案手法を応用し，既に得られているノミナルモデルのモデル誤差を補償することで，高性能なMPCを実現する手法について検討する．このために，ノミナルモデルの予測誤差のダイナミクスを表現するモデル誤差モデル (Model-Error Model: MEM) を活用した補償を考える．MEMのコンセプト自体は過去の研究で示されており，MEMを考慮したMPCについても研究されている [62, 63]．従来研究においては，MEMは線形なダイナミクスと有界なゲインの組合せによってモデル化されており，これにより不確かさの範囲を評価することでMPCのロバスト性を向上している．しかし，この方法ではノミナルモデルの予測の不確かさを評価することはできるが，誤差を補償して予測精度を向上させることは難しい．

ここで，一般に実システムは非線形システムであるため，MEMを高精度に同定するためには，それに起因する非線形性を正確に捉える必要がある．このために，本章では，4章で説明した拡大双線形化を活用してMEMを同定することを考える．

以上を踏まえ，本章では，Koopmanアプローチに基づき非線形特性を表現可能な拡大双線形MEMを同定し，それによりノミナルモデルの予測誤差を補償することで，高性能なMPCを実現する手法を提案する．まず，MEMと，それによるモデル誤差補償の考え方について説明する．次に，対象システムおよびノミナルモデルから得られるデータに基づいて，拡大双線形MEMを同定する手法について説明する．続いて，ノミナルモデルが線形システムである場合を仮定し，ノミナルモデルと拡大双線形MEMを統合した誤差補償モデルを導出する．そして，4章の提案手法を応用して，誤差補償モデルを活用したLMPCアルゴリズムを構築し，数値シミュレーションによりその有効性を示す．

5.2 モデル誤差モデル (Model-Error Model: MEM)

対象システムの実プラントのダイナミクスが次の入力アフィンシステムで表されるとする．

$$\dot{x} = f_p(x) + g_p(x)u \quad (5.1)$$

ここで, $x \in \mathbb{R}^n$ を状態, $u \in \mathbb{R}^m$ を入力とする. また, 物理的知見などに基づき事前に得られている, 対象システムに関するノミナルモデルを次式で表す.

$$\dot{\hat{x}} = f_m(\hat{x}) + g_m(\hat{x})u \quad (5.2)$$

ここで, $\hat{x} \in \mathbb{R}^n$ はノミナルモデルの状態とし, 入力の u 次数は実プラントと同じと仮定する.

そして, 共通の入力 u によって駆動された時の, 実プラント (5.1) に対するノミナルモデル (5.2) の状態誤差を $\tilde{x} := x - \hat{x}$ と定義する. この時, u を入力とする $\tilde{x}$ のダイナミクスを表現したモデルを, モデル誤差モデル (Model-Error Model: MEM) と呼ぶ. MEM は, ノミナルモデルの状態 $\hat{x}$ をパラメータと捉え, 次式のように表すことができる.

$$\begin{aligned} \dot{\tilde{x}} &= f_p(x) - f_m(\hat{x}) + (g_p(x) - g_m(\hat{x}))u \\ &:= f_e(\tilde{x}, \hat{x}) + g_e(\tilde{x}, \hat{x})u \end{aligned} \quad (5.3)$$

ここで,

$$\begin{aligned} f_e(\tilde{x}, \hat{x}) &= f_p(\hat{x} + \tilde{x}) - f_m(\hat{x}) \\ g_e(\tilde{x}, \hat{x}) &= g_p(\hat{x} + \tilde{x}) - g_m(\hat{x}) \end{aligned}$$

である.

MEM(5.3) を高精度に得ることができれば, ノミナルモデル (5.2) による状態の予測誤差を MEM によって補償することで, 実プラント (5.1) の状態をより高精度に予測することができる. ノミナルモデル (5.2) と MEM (5.3) は共通の入力によって駆動されるため, MEM を用いた誤差補償モデルは Fig. 5.1 に示す形で構成することができる.

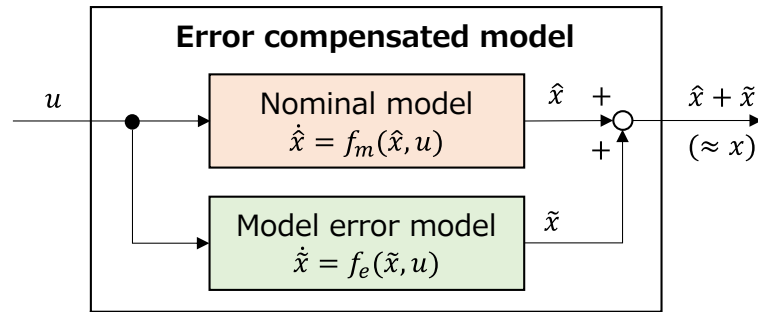


Fig. 5.1: Error compensation structure by model-error model

5.3 拡大双線形 MEM の同定

実プラントは一般に非線形システムであるため、ノミナルモデルが線形システムの場合であったとしても、MEM は非線形システムである。そこで本節では、4.3 節の方法を応用して、拡大双線形化を用いて非線形性を有する MEM を同定する手順を説明する。ここでは MEM を離散時間モデルとして同定する。まず、同定に用いるデータとして、実プラントの状態 x 、ノミナルモデルの状態 $\hat{x}$ 、共通の入力 u に関するデータセット

$$\mathcal{D} = \{x_k, \hat{x}_k, u_k\}, \quad k = 0, \dots, M \quad (5.4)$$

が得られているとする。ここで、 k は時刻インデックスであり、サンプル時間を Δt とする。実プラントとノミナルモデルの状態のサンプル値を用いると、状態誤差 $\tilde{x}_k = x_k - \hat{x}_k$, $k = 0, \dots, M$ を計算できる。このデータ取得のブロック図を Fig. 5.2 に示す。

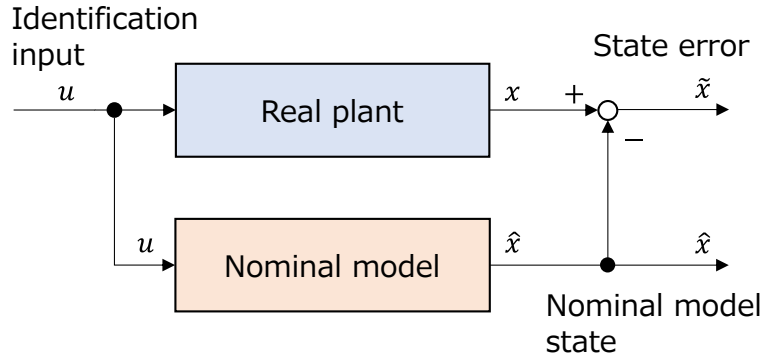


Fig. 5.2: Block diagram for data acquisition

また、実プラントが不安定な場合や、対象とする動作領域でのデータを取得するため、例えばノミナルモデルを用いた安定化コントローラを実プラントに適用した上でデータを取得してもよい。その場合のデータ取得のブロック図を Fig. 5.3 に示す。この構成の場合、安定化入力 u^{st} に対して励起信号 v を加えた信号 u を同定用入力として、実プラントとノミナルモデルそれぞれに入力すればよい。

$\mathbb{R}^{n_z \times n_z(m+1)}$ とし, さらに,

$$Z_+ = [z_2, \dots, z_M], \quad Z_u = \begin{bmatrix} z_1 & \cdots & z_{M-1} \\ z_1 u_1(1) & \cdots & z_{M-1} u_{M-1}(1) \\ \vdots & \ddots & \vdots \\ z_1 u_1(m) & \cdots & z_{M-1} u_{M-1}(m) \end{bmatrix} \quad (5.8)$$

とすると, H の推定値は次式で計算することができる.

$$\hat{H} = Z_+ Z_u^T (Z_u Z_u^T)^{-1} \quad (5.9)$$

また, (5.6) より, (5.5b) の行列 C^e は次式で得られる.

$$C^e = \begin{bmatrix} I_n & O_{n \times n_\eta} \end{bmatrix} \quad (5.10)$$

5.4 拡大双線形 MEM を用いた LMPC

本節では, 線形ノミナルモデルを対象として, 拡大双線形 MEM によって誤差を補償する誤差補償モデルを導出し, それを用いた MPC の定式化を行う. さらに, 4 章の手法を応用して事前予測状態を活用した線形化を行い, LMPC アルゴリズムを構築する.

5.4.1 誤差補償モデル

ノミナルモデルおよび同定された MEM を活用し, MPC の予測モデルとして用いるための誤差補償モデルを構築する. 以下では, ノミナルモデルとして離散時間線形システム

$$\hat{x}_{k+1} = A\hat{x}_k + Bu \quad (5.11)$$

を考える. このノミナルモデルに対し, 同定により得られる拡大双線形 MEM(5.5) を統合した誤差補償モデルは, 次のように構成できる.

$$\begin{bmatrix} \hat{x}_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ [B_1^e z_k \cdots B_m^e z_k] \end{bmatrix} u_k, \quad (5.12)$$

$$x_k = \begin{bmatrix} I_n & C^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} \quad (5.13)$$

状態方程式 (5.12) において, ブロックの上段がノミナルモデルのダイナミクス, ブロックの下段が MEM のダイナミクスを表している. また, 出力方程式 (5.13) は, ノミナルモデルの状態 $\hat{x}_k$ と MEM の出力である状態誤差 $\tilde{x}_k = C^e z_k$ の和となっている.

ここで, MPC において各サンプル時刻における実プラントの状態の観測値を x_t とする. この時, 予測ホライズンの初期ステップにおいてノミナルモデルの状態が実プラントの状態に一致していると考えれば, 観測関数ベクトル (5.6) を用いて, 誤差補償モデルの状態の初期値は次で与えられる.

$$\begin{bmatrix} \hat{x}_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} x_t \\ h(O_{n \times 1}, x_t) \end{bmatrix} \quad (5.14)$$

5.4.2 誤差補償モデルを用いた MPC の定式化

ここまでの関係式を用いて、誤差補償モデルを活用した MPC において各時刻で解く最適化問題は、次のように定式化できる。

$$\underset{u_k}{\text{minimize}} \quad J_1 \quad (5.15a)$$

$$\text{subject to} \quad \begin{bmatrix} \hat{x}_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} x_t \\ h(O_{n \times 1}, x_t) \end{bmatrix} \quad (5.15b)$$

$$\begin{bmatrix} \hat{x}_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ [B_1^e z_k \cdots B_m^e z_k] \end{bmatrix} u_k, \quad k = 0, \dots, N_p - 1 \quad (5.15c)$$

$$x_k = \begin{bmatrix} I_n & C^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix}, \quad k = 0, \dots, N_p \quad (5.15d)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (5.15e)$$

ここで、 N_p は予測ホライゾンである。また、評価関数 J_1 は次式で与えられる。

$$J_1 = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2)$$

ここで、 $P, Q \in \mathbb{R}^{n \times n}$ は準正定、 $R \in \mathbb{R}^{m \times m}$ は正定な重み行列であり、 $x_k^r, k = 0, \dots, N_p$ と $u_k^r, k = 0, \dots, N_p - 1$ はそれぞれ目標状態と目標入力とする。

ここで、誤差補償モデルのダイナミクス (5.15c) は非線形であるため、最適化問題 (5.15) は NLP 問題であることに注意されたい。

5.4.3 QP 問題の定式化

続いて、最適化問題 (5.15) の計算コスト低減のため、非線形要素を線形化して QP 問題として定式化することを考える。このために、4 章の手法を応用し、事前予測状態を活用した線形化と、その線形化により発生する予測誤差の補償を考える。

まず、予測ホライゾン上の状態列の事前予測値である事前予測状態を活用し、予測モデル (5.15c) における状態依存の入力係数行列を時変行列として近似することを考える。誤差補償システムの予測ホライゾン上の事前予測状態を $[\hat{x}_k^{*T} \ z_k^{*T}]^T, k = 0, \dots, N_p$ とする。また、誤差状態に対応した事前予測値は $\tilde{x}_k^* = C^e z_k^*, k = 0, \dots, N_p$ として得ることができる。この時、事前予測状態を用いて、(5.15c) は次の線形時変システムとして近似できる。

$$\begin{bmatrix} \hat{x}_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ [B_1^e z_k^* \cdots B_m^e z_k^*] \end{bmatrix} u_k \quad (5.16)$$

ここで、事前予測状態の精度が低い場合、(5.16) の (5.15c) に対する近似精度が低くなり、予測誤差が発生する恐れがある。この問題に対し、MEM が拡大双線形モデルで得られていることを利用し、4.4.2 項と同様に拡大状態に関する制約条件と、補償入力項を導入して対処する。

まず、予測ホライゾンの k ステップ目における拡大状態に関して、次の制約条件を設ける。

$$z_k(n+1:n_z) = \eta(z_k(1:n), \hat{x}_k) \quad (5.17)$$

ここで、 $z_k(i:j)$ は Matlab 表記を用いて z_k の i から j 番目の要素のベクトルを表すものとする。制約条件 (5.17) は、予測ホライゾンの k ステップ目において、MEM の拡大状態 z_k のうち元の状態に対応する部分ベクトル $z_k(1:n)$ 、拡大した部分に対応する部分ベクトル $z_k(n+1:n_z)$ と、ノミナルモデルの状態 $\hat{x}_k$ が、観測関数ベクトルの部分ベクトル $\eta(\hat{x}, \hat{x})$ による写像の数値的な関係を満たす、という条件である。

さらに、制約条件 (5.17) は非線形であるため、QP 問題を構成するためにはこれを線形化する必要がある。(5.17) を事前予測状態 $(\tilde{x}_k^*, \hat{x}_k^*)$ 近傍で線形近似すると、次の関係式が得られる。

$$z_k(n+1:n_z) = \eta(\tilde{x}_k^*, \hat{x}_k^*) + \frac{\partial \eta}{\partial \tilde{x}} \bigg|_{\substack{\tilde{x}=\tilde{x}_k^* \\ \hat{x}=\hat{x}_k^*}} (z_k(1:n) - \tilde{x}_k^*) + \frac{\partial \eta}{\partial \hat{x}} \bigg|_{\substack{\tilde{x}=\tilde{x}_k^* \\ \hat{x}=\hat{x}_k^*}} (\hat{x}_k - \hat{x}_k^*) \quad (5.18)$$

さらに、上記の制約条件を考慮した最適化問題の実行可能性を確保するために、拡大状態のうち $\eta(\tilde{x}, \hat{x})$ に対応した次元に対する補償入力 $\Delta u_k \in \mathbb{R}^{n_n}$ を導入する。線形時変モデルとして近似した誤差補償モデル (5.16) に対し補償入力 Δu_k を加えた予測モデルは次のようになる。

$$\begin{bmatrix} \hat{x}_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ [B_1^e z_k^* \cdots B_m^e z_k^*] \end{bmatrix} u_k + \begin{bmatrix} O_{n \times 1} \\ O_{n \times 1} \\ \Delta u_k \end{bmatrix} \quad (5.19)$$

以上より，補償入力 Δu_k を追加の最適化変数として，LMPC の各時刻で解くべき最適化問題は次のように与えられる．

$$\underset{u_k}{\text{minimize}} \quad J_2 \quad (5.20a)$$

$$\text{subject to} \quad \begin{bmatrix} \hat{x}_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} x_t \\ h(O_{n \times 1}, x_t) \end{bmatrix} \quad (5.20b)$$

$$\begin{bmatrix} \hat{x}_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ [B_1^e z_k^* \cdots B_m^e z_k^*] \end{bmatrix} u_k + \begin{bmatrix} O_{n \times 1} \\ O_{n \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \quad (5.20c)$$

$$x_k = \begin{bmatrix} I_n & C^e \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix}, \quad k = 0, \dots, N_p \quad (5.20d)$$

$$\begin{aligned} z_k(n+1 : n_z) &= \eta(\tilde{x}_k^*, \hat{x}_k^*) + \frac{\partial \eta}{\partial \tilde{x}} \bigg|_{\substack{\tilde{x}=\tilde{x}_k^* \\ \hat{x}=\hat{x}_k^*}} (z_k(1:n) - \tilde{x}_k^*) \\ &\quad + \frac{\partial \eta}{\partial \hat{x}} \bigg|_{\substack{\tilde{x}=\tilde{x}_k^* \\ \hat{x}=\hat{x}_k^*}} (\hat{x}_k - \hat{x}_k^*), \quad k = 1, \dots, N_p \end{aligned} \quad (5.20e)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (5.20f)$$

評価関数 J_2 は， Δu_k に関する重み行列を $R_\Delta \in \mathbb{R}^{n_\eta \times n_\eta}$ として次式で与えられる．

$$J_2 = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2 + \|\Delta u_k\|_{R_\Delta}^2)$$

最適化問題 (5.20) は，初期ステップにおける拡大状態 z_0 を求める (5.20b) を除き，二次形式の評価関数と線形の制約条件で構成されており，高速な QP ソルバで解くことが可能な定式化となっている．

5.4.4 事前予測状態の取得

事前予測状態の取得手順は，3.4.2 項および 4.5.2 項で説明した内容と同様である．まず，ある時刻 t において最適化問題 (5.20) を解いて得られた最適入力列 u_k ， Δu_k と，予測モデル (5.20c) を用いて，各ステップにおける予測状態の最適値 $\hat{x}_k^*$ ， z_k^* ， $k = 0, \dots, N_p$ を計算することができる．

また，制御周期が予測モデル (5.20c) のサンプル時間 Δt と一致していると仮定する．この

時, 事前予測状態は次のようにして取得することができる.

$$\begin{aligned} \begin{bmatrix} \hat{x}_k^* \\ z_k^* \end{bmatrix} &= \begin{bmatrix} \hat{x}_{k+1}^{*-} \\ z_{k+1}^{*-} \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \\ \begin{bmatrix} \hat{x}_{N_p}^* \\ z_{N_p}^* \end{bmatrix} &= \begin{bmatrix} \hat{x}_{N_p}^{*-} \\ z_{N_p}^{*-} \end{bmatrix} \end{aligned} \quad (5.21)$$

また, 制御ループの初期時刻においては, 対象システムの初期状態を x_{t0} として, 事前予測状態を次のように取得することができる.

$$z_k^* = h(O_{n \times 1}, x_{t0}), \quad k = 0, \dots, N_p \quad (5.22)$$

5.4.5 LMPC アルゴリズム

以上の議論より, 提案手法の LMPC アルゴリズムは Algorithm 3 のようにまとめられる.

Algorithm 3 LMPC with error-compensated model

Require: Reference trajectory (x^r, u^r)

while Controller is running **do**

 Observe current state x_t of the system

if $t=0$ **then**

 Set the prior-predicted states $\hat{x}_k^*, z_k^*$ by (5.22)

end if

 Solve the optimization problem (5.20) with the prior-predicted states $\hat{x}_k^*, z_k^*$

 Update the prior-predicted states $\hat{x}_k^*, z_k^*$ by (5.21)

 Apply first input u_0 to the system

end while

本提案手法によれば, 線形ノミナルモデルと拡大双線形 MEM を統合した誤差補償モデルを予測モデルとして, 実プラントに対し高精度に状態を予測し, 最適入力を計算することができる. これにより, 事前に得られているノミナルモデルの情報を活用した上で, そのモデル誤差による予測誤差を補償し, 高性能な MPC を実現できる. さらに, 4 章の手法を応用して事前予測状態を用いた線形化と予測誤差補償を行うことで最適化問題を QP 問題として定式化し, 高速に求解することができる.

5.5 数値シミュレーション

本節では、4章と同様に、2次元平面上で動作するドローンモデルを対象システムとして、数値シミュレーションにより提案手法の有効性を評価した結果を説明する。

5.5.1 対象システム

対象システムとして、Fig. 4.1 に示す 2 次元平面で動作するドローンを考える．対象システムのダイナミクスは次式で与えられる．

$$\frac{d}{dt} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} = \begin{bmatrix} v_y \\ v_z \\ \omega \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta \\ \frac{1}{m} \cos \theta & \frac{1}{m} \cos \theta \\ -\frac{l_a}{I_{xx}} & \frac{l_a}{I_{xx}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}, \quad (5.23)$$

ここで、 p_y, p_z は y, z 座標 [m]， θ は角度 [rad]， v_y, v_z は y, z 軸方向の速度 [m/s]， ω は角速度 [rad/s]， T_1, T_2 は左右のプロペラのスラスト力 [N] である．このシステムにおいて、状態を $x = [p_y \ p_z \ \theta \ v_y \ v_z \ \omega]^T$ ，入力を $u = [T_1 \ T_2]^T$ とする．また、パラメータは重量 $m = 2.0$ ，回転慣性 $I_{xx} = 1.0$ ，プロペラ軸までのアーム長 $l_a = 0.20$ ，重力加速度 $g = 9.81$ とした．

これに対し、ノミナルモデルとして、平衡状態 $\bar{x} = [0, 0, 0, 0, 0, 0]^T$ ，平衡入力 $\bar{u} = [\frac{1}{2}mg, \frac{1}{2}mg]^T$ 近傍での線形近似モデルを考える．また、ノミナルモデルは、実プラント (5.23) に対してパラメータ誤差を持つものとし、 m, I_{xx}, l_a について 5%, 20%, 10% の増加方向の誤差をそれぞれ仮定した．

パラメータの一覧を Table 5.1 に示す．

Table 5.1: Physical parameters of the real plant and the nominal model

Parameter	Variable	Real plant	Nominal model (error)
Body mass	m	2.0	2.1 (+5%)
Rotational inertia	I_{xx}	1.0	1.2 (+20%)
Arm length to propeller axis	l_a	0.20	0.22 (+10%)
Acceleration of gravity	g	9.81	9.81 (0%)

5.5.2 MEM の同定

MEM の同定にあたり、対象システム (5.23) を原点へ安定化させるコントローラとしてノミナルモデルに基づき設計した LQR を適用し、同定用データを取得した．データ取得のサ

サンプル時間は $0.05[\text{s}]$ とし, 100 点のランダムな初期状態から原点に安定化させる動作を行った. ここで, 初期値は $x_0 \sim \mathcal{N}(O_{6 \times 1}, \sigma_0)$, $\sigma_0 = \text{diag}(1, 1, 0.1, 0, 0, 0)$ とした. また, 信号を励起させるため, 安定化フィードバック入力 $u_{st} = -Kx$ にノイズ信号 u_v を加え, 対象システムに入力した. ここで, K はノミナルモデルに基づく LQR ゲインであり, $u_v \sim \mathcal{N}(O_{2 \times 1}, I_2)$ とした. 観測関数は, 対象システムに関する物理的知見に基づき,

$$\eta(\tilde{x}, \hat{x}) = \begin{bmatrix} \hat{\theta} \\ \cos(\hat{\theta} + \tilde{\theta}) \\ \sin(\hat{\theta} + \tilde{\theta}) \\ 1 \end{bmatrix} \quad (5.24)$$

と選定した.

以上の条件のもと MEM を同定し, それをノミナルモデルと統合して誤差補償モデルを構築した. 誤差補償モデルの予測精度を, テストデータを用いて確認した結果を Fig. 5.4 に示す. テストデータでは, 訓練データと異なる 30 点のランダムな初期値を使用し, 各軌道について $2[\text{s}]$ のシミュレーションを行った. この結果より, ノミナルモデルのみの状態予測値は真値に対して大きくずれているのに対し, MEM との統合モデルでは真値に近い予測値が得られていることが分かる.

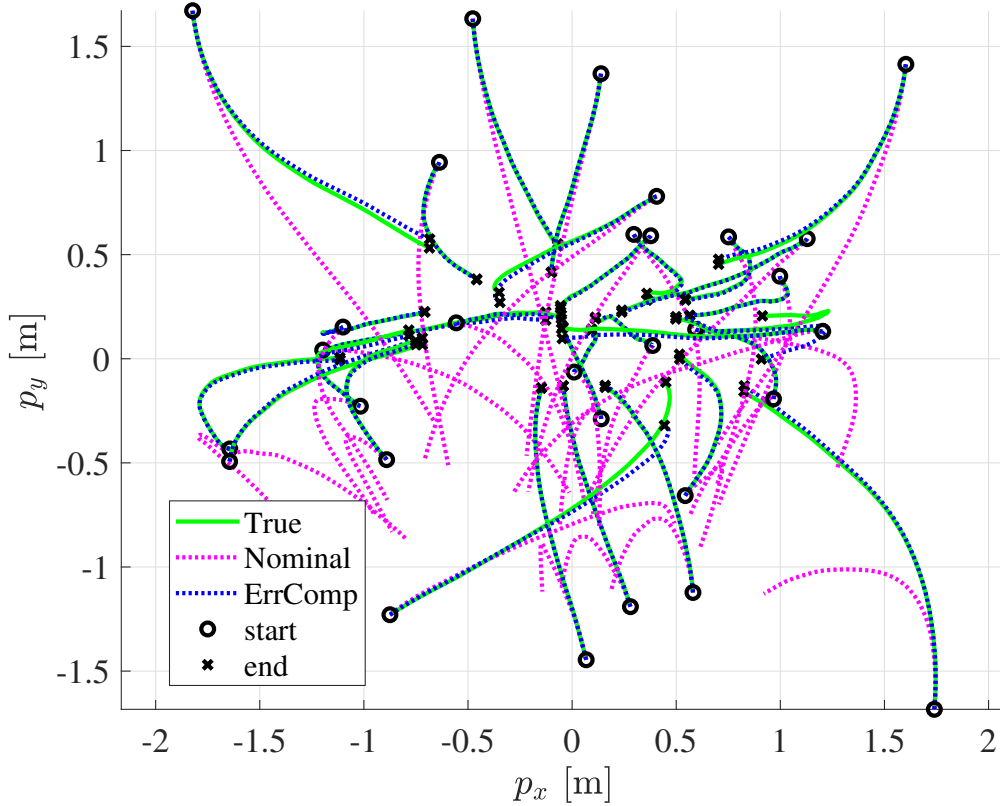


Fig. 5.4: Comparison of prediction accuracy

5.5.3 評価結果

前述の同定により得られたモデルを用いて、提案の LMPC の評価を行った．なお，比較対象として，真のシステムを予測モデルとした NMPC（以下，NMPC）と，ノミナルモデルを予測モデルとした LMPC（以下，ノミナル LMPC）の 2 つを評価した．

ここでは，5[s] ごとに y 軸に沿って 15[m] ジャンプする目標位置を与え，ドローンをアグレッシブに水平方向に反復移動させる制御を行った．サンプル時間は $\Delta t = 0.05[\text{s}]$ ，予測ホライズンの長さは $N_p = 30$ ，評価関数 J_2 における重み行列は， $P = 100 \cdot I_6$ ， $Q = I_6$ ， $R = 0.1 \cdot I_2$ ， $\Delta R = 0.001 \cdot I_4$ とした．また，入力に関する制約条件として， $T_1, T_2 \geq 0$ とした．

なお，NMPC のソルバには CasADi [61] をインターフェースとして IPOPT を使用し，提案手法および Nominal LMPC のソルバには MATLAB の quadprog 関数を使用した．シミュレーション実行環境には Core i5 CPU@1.20GHz, 24.0GB メモリ, Windows 10 の PC 上で動作する MATLAB 2020b を使用した．

Fig. 5.5, Fig. 5.6 に，各手法による状態と入力の比較結果をそれぞれ示す．まずノミナル LMPC による制御結果については，目標値に対して安定して追従できておらず，しばしば大きな誤差が発生している．これは特に姿勢変化が大きくなったときに，実プラントの非線形な特性に対する予測精度が低下していることに起因していると考えられる．これに対し提案手法では，目標値に対して安定した追従が実現できており，その応答は真のモデルを用いた NMPC とほとんど一致していることが分かる．

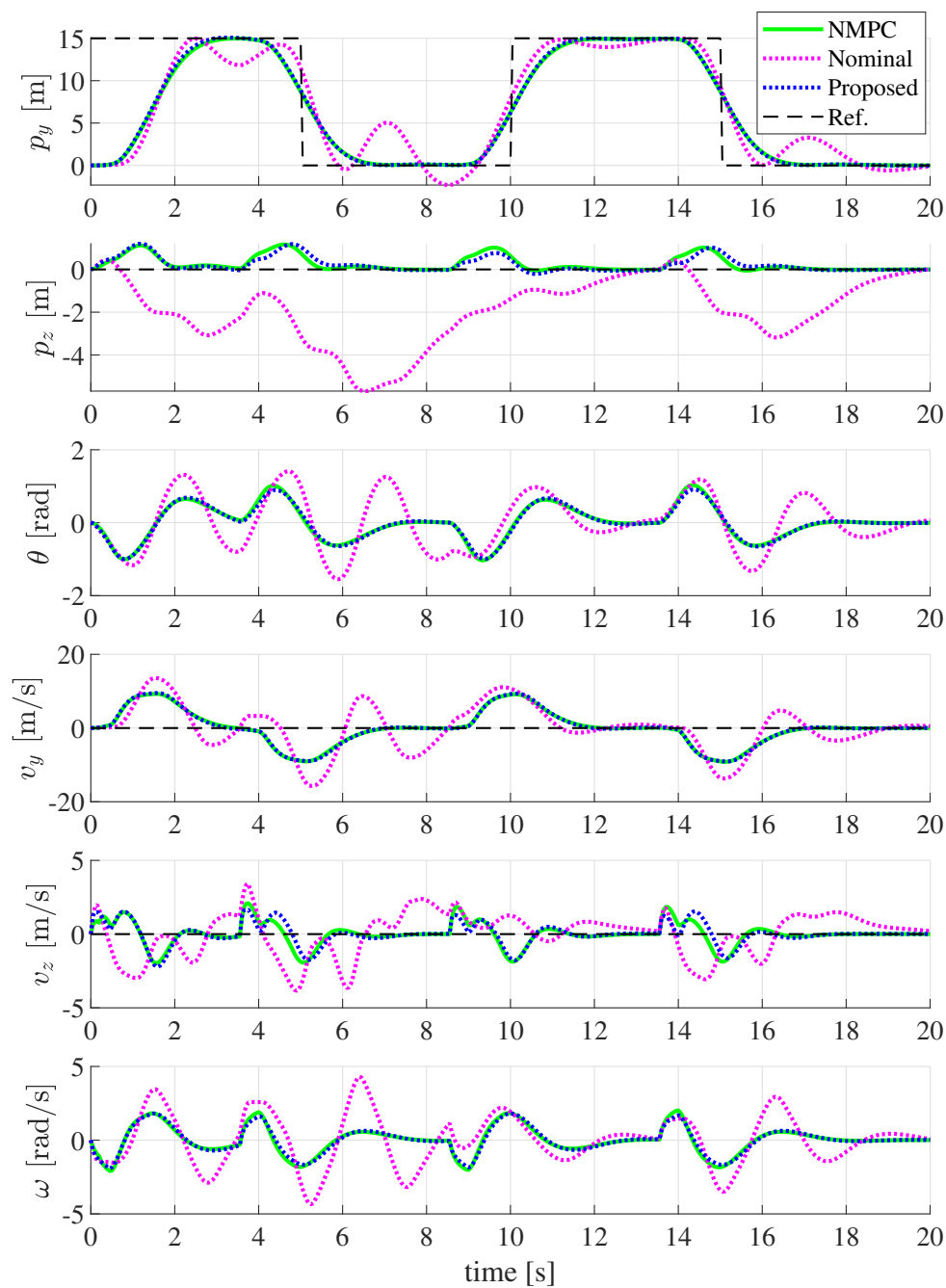


Fig. 5.5: States

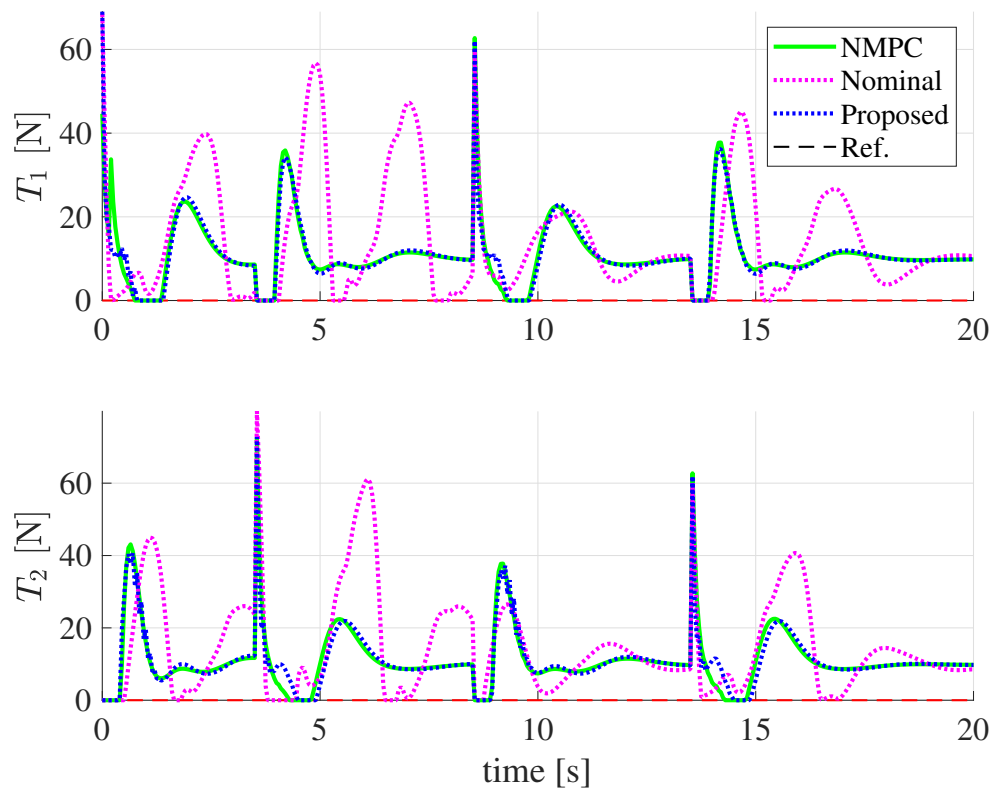


Fig. 5.6: Inputs

また，提案手法における補償入力のグラフを Fig. 5.7 に示す．グラフより，各拡大状態に対応する補償入力，目標値が切り替わる前後のタイミングで相対的に大きな値をとっていることが分かる．これにより，特に過渡応答において，補償入力によって近似による予測誤差をうまく補償できているものと考えられる．

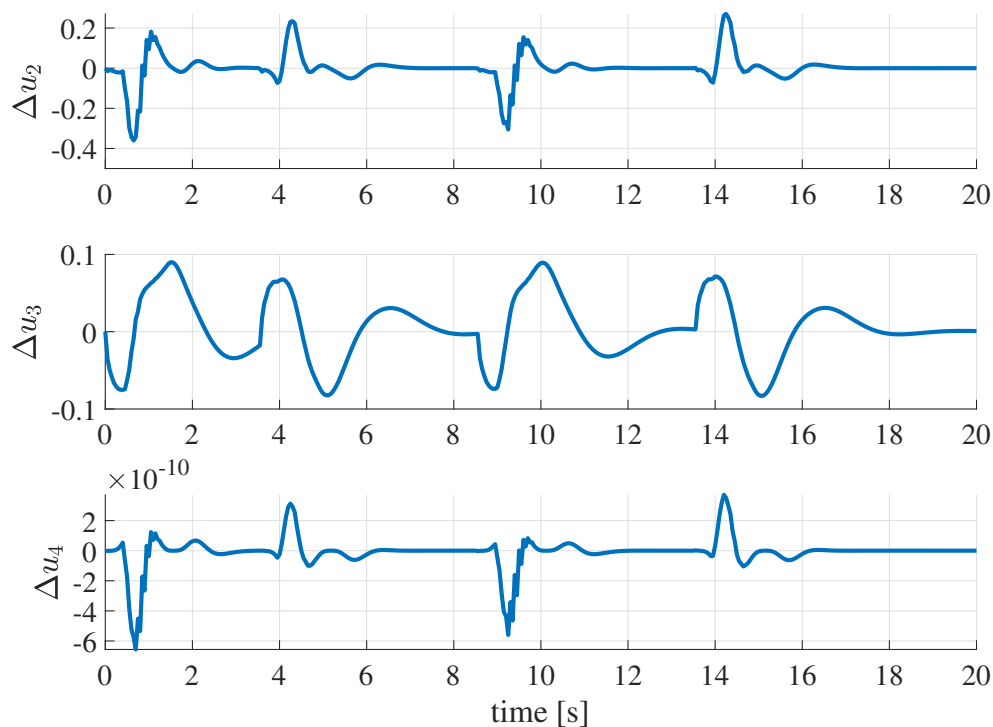


Fig. 5.7: Corrective inputs

続いて、シミュレーション時間全体に対する評価関数の値を比較した結果を、Fig. 5.8に示す。NMPCの値を1として正規化すると、Nominal LMPCでは2.07であるのに対し、提案手法では0.99であった。この結果より、ノミナルモデルのみでは2倍以上の性能悪化が発生してしまうが、提案手法により、真のモデルを用いたNMPCと同等の性能を達成できていることが分かる。

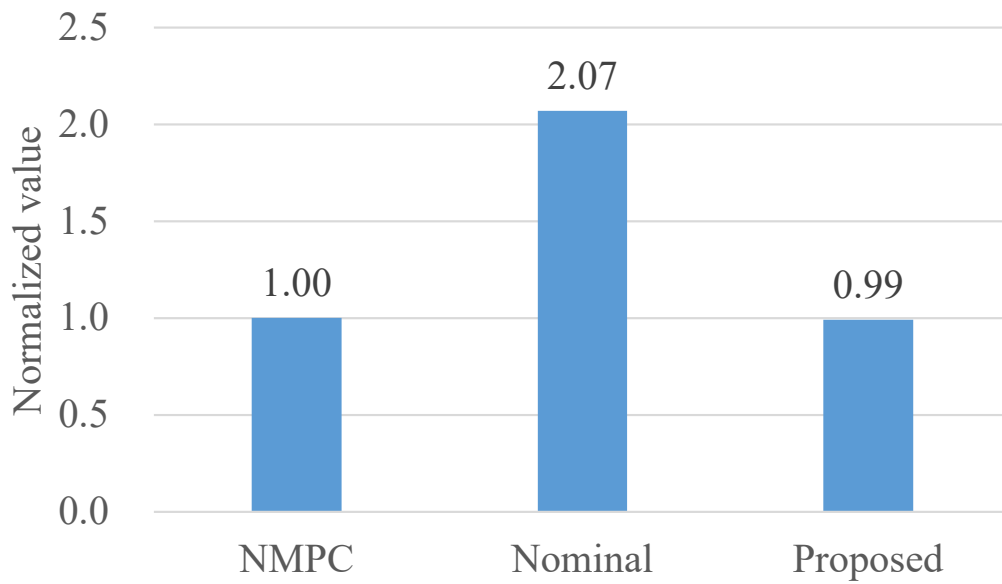


Fig. 5.8: Normalized objective function values

最後に、Fig. 5.9 に、予測ホライズン $N_p = 10, 20, 30$ の場合の、各手法の最適化ソルバの実行時間を示す。この結果より、提案手法では NMPC よりも大幅に計算時間を短縮できていることが分かる。一方、ノミナル LMPC と比較すると提案手法の方が計算時間は長い、これはモデル次数と制約条件の増加によるものと考えられる。

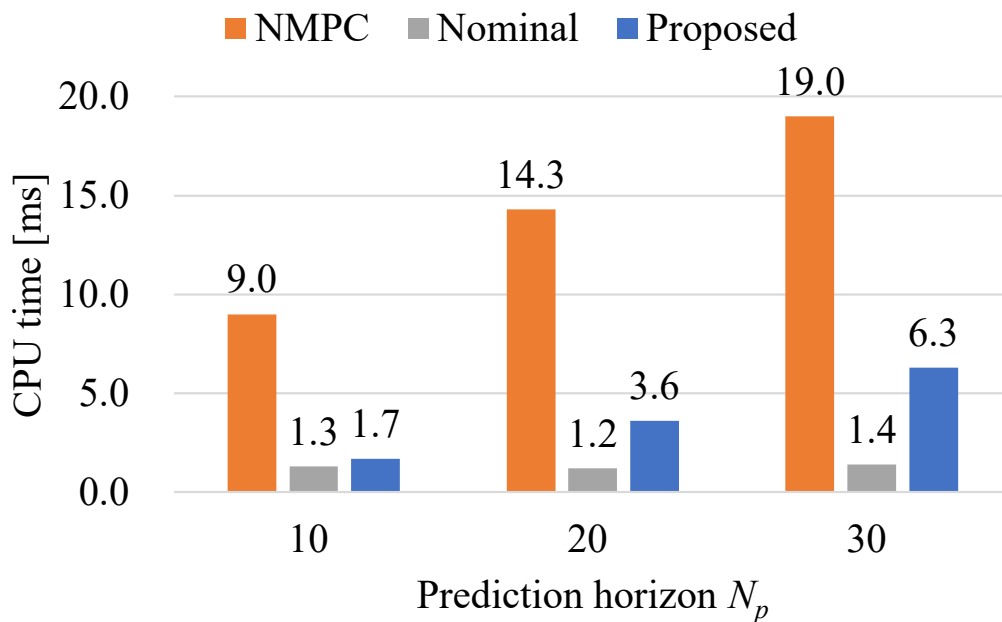


Fig. 5.9: CPU time

以上の評価結果より、提案手法は、NMPC と比べて大幅に短い時間で、ノミナル LMPC よりも高い制御性能を実現できており、計算時間と制御性能の両方に対して優位性を持つと

言える.

5.6 まとめ

本章では, 拡大双線形モデルを用いた MPC を応用し, 事前に対象システムのモデルが得られている場合に, そのモデル誤差を補償して高性能な MPC を実現する手法を提案した. まず, 事前に得られているノミナルモデルの実プラントに対するモデル誤差を拡大双線形 MEM として同定する方法を示し, ノミナルモデルと拡大双線形 MEM を統合した誤差補償モデルを導出した. さらに, 誤差補償モデルを予測モデルとした MPC において, 拡大状態に基づく制約条件による予測誤差補償と, 事前予測状態を用いた線形化を考慮することで, 高精度な予測と低計算コストを両立する LMPC を提案した.

数値シミュレーションでは, 2次元平面上で動作するドローンモデルを対象としたアグレッシブな制御を行い, 提案手法の有効性検証を行った. 真のモデルを用いた NMPC と, ノミナルモデルを用いた LMPC との比較を行った結果, 制御性能と計算コストに関して提案手法の優位性を検証することができた.

第6章 拡大双線形モデルを用いた線形モデル予測制御のスケーラビリティ向上

6.1 はじめに

これまでの各章の提案手法で用いた予測誤差補償では、観測関数の数と予測ホライゾンの長さに応じて、最適化問題における制約条件と最適化変数が増加する。このため、観測関数の数や予測ホライゾン長が大きい場合には、計算時間が大きく増加してしまうという問題があった。これまでの各章における数値シミュレーションでは比較的低次の対象システムを扱い、観測関数の数は最大でも4つ(拡大状態は10次元)、予測ホライゾンは30ステップであったが、より実践的なシステムを対象とした場合、非線形性の複雑さに応じてより多くの観測関数を用意する必要がある状況が想定される。したがって、前述の提案手法の実用性を向上するためには、スケーラビリティ向上が重要な課題である。

そこで本研究では、4章の提案手法をベースに、拡大双線形モデルに基づくスケーラブルなLMPCを提案し、実践的な非線形システムに適用可能であることを示す。本章では、実践的な非線形システムの例として、3次元空間で動作するクアッドロータを対象として扱う。ここで、クアッドロータモデルは解析的に拡大双線形化可能であるため、まずその拡大双線形モデルを導出する。さらに、MPCの定式化において、拡大状態に関する制約条件を追加する代わりに、拡大状態の各要素が満たすべき数値的関係の誤差をフィードバックする補償入力を新たに導入する。これにより、拡大双線形モデルに基づくスケーラブルなLMPCアルゴリズムを構築する。そして、クアッドロータモデルを対象システムとした数値シミュレーションにより、提案手法の有効性を示す。

6.2 対象システムと拡大双線形モデルの導出

本節では、本章で扱う対象システムと、その拡大双線形モデルの導出について説明する。

6.2.1 対象システム

本研究では、実践的な非線形特性を持った制御対象の例として、3次元空間で動作するクアッドロータを考える。ここでは、[64]で用いられているキネマティックモデルを参考にす

る．対象システムの模式図を Fig. 6.1 に示す．

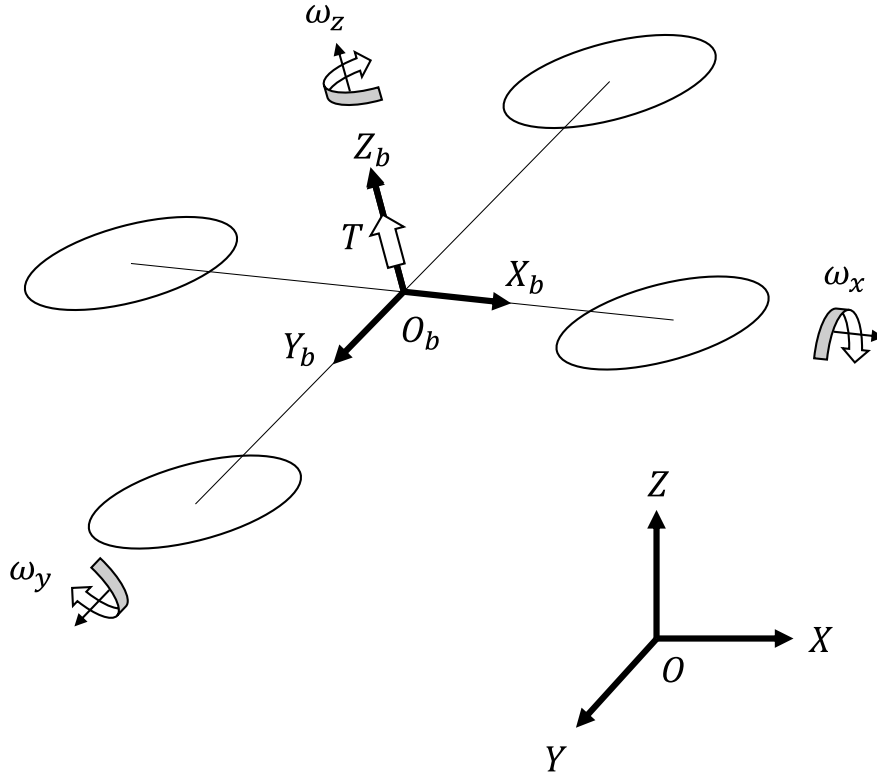


Fig. 6.1: Quadrotor kinematic model

対象システムの状態を, $x = [p^T, q^T, v^T]^T \in \mathbb{R}^{n_x}$ とする．ここで, $p = [p_x, p_y, p_z]^T$ は基準座標系 O における機体の位置 (すなわち, 機体座標系 O_b の原点), $q = [q_w, q_x, q_y, q_z]^T$ はクォータニオンで表現される機体座標系 O_b の基準座標系 O に対する姿勢, $v = [v_x, v_y, v_z]^T$ は基準座標系 O における機体の速度とする．また, 対象システムの入力を, $u = [\omega^T, T]^T \in \mathbb{R}^m$ とする．ここで, $\omega = [\omega_x, \omega_y, \omega_z]^T$ は機体座標系 O_b における角速度, T は機体重量で正規化されたスラスト力とする．以上より, 状態と入力 の次数はそれぞれ $n_x = 10, m = 4$ である．

これらの変数を用いて, 対象システムのダイナミクスは次の非線形入力アフィンシステムとして与えられる．

$$\frac{d}{dt} \begin{bmatrix} p \\ q \\ v \end{bmatrix} = \begin{bmatrix} v \\ O_{4 \times 1} \\ -g_{vec} \end{bmatrix} + \begin{bmatrix} O_{3 \times 3} & O_{3 \times 1} \\ \frac{1}{2}Q(q) & O_{4 \times 1} \\ O_{3 \times 3} & R(q) \end{bmatrix} \begin{bmatrix} \omega \\ T \end{bmatrix} \quad (6.1)$$

ここで, $g_{vec} = [0, 0, 9.81]^T$ は重力加速度ベクトルである．また, $Q(q), R(q)$ はそれぞれ次式

で表される.

$$Q(q) = \begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix} \quad (6.2)$$

$$R(q) = \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} \quad (6.3)$$

なお, クォータニオン q は, $|q| = 1$ を満たす.

6.2.2 拡大双線形モデルの導出

続いて, 対象システムの拡大双線形モデルの導出について説明する. 4.2 節で説明した入力アフィンシステムの Koopman Bilinear Form (KBF) (4.3) への変換においては, 一般には条件を満たす有限個の Koopman 固有関数の組を見つけることは難しく, KBF は無限次元となる. このため, 実用的には有限個の観測関数を選び, 4.3 節で説明したようなデータドリブンな方法により, 近似的に有限次元の拡大双線形モデルを同定する. しかし, 今回の対象システム (6.1) では, 観測関数を適切に選択することで, 解析的に拡大双線形化することができる. 以降ではその方法について説明する.

観測関数ベクトルを, 元の状態ベクトル x と, 状態ベクトルの非線形関数ベクトル $\eta(x)$ を連結したベクトル

$$h(x) = \begin{bmatrix} x \\ \eta(x) \end{bmatrix} \quad (6.4)$$

とする. ここで, $\eta(x)$ の構成要素として, クォータニオン q の要素に関する全ての 2 次の単項式と, 定数項を考える. すなわち, $\eta(x) \in \mathbb{R}^{n_\eta}$ を, 次のように構成する.

$$\eta(x) = [q_w^2, q_x^2, q_y^2, q_z^2, q_w q_x, q_w q_y, q_w q_z, q_x q_y, q_y q_z, q_z q_x, 1]^T \quad (6.5)$$

ここで, $n_\eta = 11$ である.

このようにすると, (6.1) において, まず $R(q)$ が $h(x)$ に関して線形な形で表せるため, 状態 x の時間微分は $h(x)$ に関して双線形な形で表現することができる. 続いて, $\eta(x)$ の要素のうち, クォータニオンの要素の 2 次の単項式について, その時間微分を考える. $q_a, q_b \in \{q_w, q_x, q_y, q_z\}$ とすると, $q_a q_b$ の時間微分は次のようになる.

$$\frac{d}{dt}(q_a q_b) = \dot{q}_a q_b + q_a \dot{q}_b \quad (6.6)$$

ここで, (6.1) より

$$\dot{q} = \frac{1}{2} Q(q) \omega \quad (6.7)$$

であるため, $\dot{q}_a, \dot{q}_b \in \{\dot{q}_w, \dot{q}_x, \dot{q}_y, \dot{q}_z\}$ は, いずれも q の要素の一次の項と入力変数 ω の要素のクロスタームで表される. したがって, (6.6) より, $q_a q_b$ の時間微分は q の要素の二次の項と入力変数 ω の要素のクロスタームで表され, $h(x)$ に関して双線形な形で表現することができる. また, $\eta(x)$ の要素のうち, 定数項の時間微分はゼロである.

以上より, 元の状態を観測関数ベクトルにより写像した拡大状態を $z = h(x) \in \mathbb{R}^{n_z}$ ($n_z = 21$) とすると, 対象システムは, 次のような連続時間の拡大双線形システムに厳密に変形することができる.

$$\dot{z} = A_c z + \sum_{i=1}^m B_{ci} z u(i) \quad (6.8a)$$

$$x = Pz \quad (6.8b)$$

ここで, $u(i)$ は u の i 番目の要素であり, A_c, B_{ci} ($i = 1, \dots, m$) は適切な定数行列, $P = [I_{n_x}, O_{n_x \times n_\eta}]$ は拡大状態 z から元の状態 x への射影行列である. さらに, サンプル時間を Δt , 時刻インデックスを k とすると, オイラー近似により次の離散時間の拡大双線形モデルを得ることができる.

$$z_{k+1} = A z_k + \sum_{i=1}^m B_i z_k u_k(i) \quad (6.9a)$$

$$x_k = P z_k \quad (6.9b)$$

ここで, $A = I_{n_z} + A_c \Delta t$, $B_i = B_{ci} \Delta t$ ($i = 1, \dots, m$) である.

なお, ここでは観測関数の一つに定数を含めることにより (6.8a) を導出したが, 入力に関する線形項を別途考慮することで, 定数項は観測関数から除外することができる. そのようにすることで, 拡大状態の次元をさらに減らすことができる.

6.3 拡大双線形モデルを用いたスケーラブルな LMPC

本節では、まず拡大双線形モデルを用いて NMPC としての定式化を行った上で、4 章の提案手法における課題を解決するための新たな予測誤差補償手法を導入し、拡大双線形モデルを用いたスケーラブルな LMPC アルゴリズムを提案する。

6.3.1 拡大双線形モデルを用いた MPC の定式化

拡大双線形モデルを用いた LMPC の構築に向け、まず NMPC としての定式化を行う。各時刻で解く最適化問題を次のように構成する。

$$\underset{u_k}{\text{minimize}} \quad J \quad (6.10a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (6.10b)$$

$$z_{k+1} = Az_k + \sum_{i=1}^m B_i z_k u_k(i), \quad k = 0, \dots, N_p - 1 \quad (6.10c)$$

$$x_k = Pz_k, \quad k = 0, \dots, N_p \quad (6.10d)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (6.10e)$$

ここで、 N_p は予測ホライゾン、 x_t は時刻 t において観測された状態とする。 A_{con}, b_{con} は状態と入力に関する線形制約条件を構成する行列およびベクトルである。また、評価関数 J は次式で与えられる。

$$J = \|x_{N_p} - x_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2)$$

ここで、 $P, Q \in \mathbb{R}^{n_x \times n_x}$ は準正定、 $R \in \mathbb{R}^{m \times m}$ は正定な重み行列であり、 $x_k^r, k = 0, \dots, N_p$ と $u_k^r, k = 0, \dots, N_p - 1$ はそれぞれ目標状態と目標入力とする。

以上の定式化では、拡大双線形モデルによる予測式 (6.10c) が非線形であるため、最適化問題 (6.10) は NLP 問題である。以降では、この最適化問題を、より効率的に解くことのできる QP 問題として定式化することを考える。

6.3.2 予測誤差補償手法の改良

最適化問題 (6.10) を QP 問題に変換するために、後述するように拡大双線形モデル (6.10c) を線形近似する。ここでは、この線形近似によって発生する誤差を吸収するための予測誤差補償について、4 章の提案手法（以下では、従来手法とする）を改良した手法について説明する。

従来手法では、拡大双線形モデルにおける、観測関数による状態の拡大に関する特徴を活用した予測誤差補償を提案した．具体的には、最適化問題 (6.10) に対し、次のような観測関数に基づく等式制約条件を追加した．

$$z_k(n_x + 1 : n_z) = \eta(z_k(1 : n_x)), \quad k = 1, \dots, N_p \quad (6.11)$$

これは、予測ホライゾンの各時刻ステップにおいて、拡大状態 z_k のうち、元の状態に対応する要素 $z_k(1 : n_x)$ と、非線形観測関数ベクトルに対応する要素 $z_k(n_x + 1 : n_z)$ が、観測関数 $\eta(x)$ を介した数値的な関係性を満たすようにするための制約条件である．従来手法ではさらに、最適制御問題の可解性を確保するため、拡大状態 z_k のうち、非線形観測関数ベクトルに対応した次元について、補償入力項を新たな最適化変数として導入した．すなわち、(6.10c) に対して補償入力項 Δu_k を加え

$$z_{k+1} = Az_k + \sum_{i=1}^m B_i z_k u_k(i) + \begin{bmatrix} O_{n_x \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \dots, N_p - 1 \quad (6.12)$$

として、 Δu_k を各時刻ステップ k に関して独立な新たな最適化変数として追加した．

しかしながら、これらの制約条件と最適化変数の数は、非線形観測関数ベクトル $\eta(x)$ の次数 n_η 、および予測ホライゾンの長さ N_p に伴って増大する．具体的には、制約条件 (6.11) と最適化変数 Δu_k は、ともに $n_\eta N_p$ 個増加することになる．このため、これらの制約条件と最適化変数を追加した最適化問題を、最終的に全ての非線形要素を線形近似することにより QP 問題に帰着させたとしても、 n_η や N_p の大きさによっては計算コストが大きく増大してしまうという課題がある．

この課題を解決するため、本章の提案手法では、(6.11) の両辺の差を拡大状態にフィードバックすることにより誤差補償を行うことを考える．すなわち、予測モデル (6.10c) を、次式のように変更する．

$$z_{k+1} = Az_k + \sum_{i=1}^m B_i z_k u_k(i) - \gamma(z_k - h(z_k(1 : n_x))) \quad (6.13)$$

ここで、 $\gamma \in \mathbb{R}$ はフィードバックゲインである．ここで、追加した誤差補償項のうちフィードバックゲインを除く部分は、観測関数ベクトル (6.5) より

$$\begin{aligned} z_k - h(z_k(1 : n_x)) &= \begin{bmatrix} z_k(1 : n_x) \\ z_k(n_x + 1 : n_z) \end{bmatrix} - \begin{bmatrix} z_k(1 : n_x) \\ \eta(z_k(1 : n_x)) \end{bmatrix} \\ &= \begin{bmatrix} O_{n_x \times 1} \\ z_k(n_x + 1 : n_z) - \eta(z_k(1 : n_x)) \end{bmatrix} \end{aligned} \quad (6.14)$$

となる．したがって、拡大状態 z_k のダイナミクスに対して追加した誤差補償項は、元の状態に対応する次元 $z_k(1 : n_x)$ には影響を与えず、非線形観測関数ベクトルに対応する次元 $z_k(n_x + 1 : n_z)$ のダイナミクスについて、(6.11) の両辺の差をフィードバックするものである．

この定式化によれば、従来手法における制約条件と最適化変数の追加を必要とすることなく、拡大双線形モデルの予測誤差補償を行うことができる。これによって、非線形観測関数の数と予測ホライゾンの長さに関して、前述のスケラビリティの課題に対する改善が期待できる。

6.3.3 事前予測状態を用いた QP 問題の定式化

続いて、予測誤差補償を含む拡大双線形モデル (6.13) を線形化し、各時刻で解く最適化問題を QP 問題として定式化する。このために、4 章と同様に、予測ホライゾン上の最適状態列の予測値である事前予想状態を用いた近似を考える。

ここでは簡単のため、制御周期が拡大双線形モデル (6.13) のサンプル時間 Δt と一致していると仮定する。この時、前時刻で求めた最適状態列を $z_k^*, k = 0, \dots, N_p$ とすると、事前予測状態 $z_k^*, k = 0, \dots, N_p$ は次式のように得ることができる。

$$\begin{aligned} z_k^* &= z_{k+1}^{*-}, \quad k = 0, \dots, N_p - 1 \\ z_{N_p}^* &= z_{N_p}^{*-} \end{aligned} \quad (6.15)$$

また、初期時刻においては、前時刻の最適状態列を利用できないため、次のように、初期時刻の状態 x_t に関する拡大状態を、全ての時刻ステップに使用する。

$$z_k^* = h(x_t), \quad k = 0, \dots, N_p \quad (6.16)$$

ここで、元の状態空間に射影した事前予測状態は、次のように計算できる。

$$x_k^* = Pz_k^*, \quad k = 0, \dots, N_p \quad (6.17)$$

以上のようにして得られる事前予測状態を用いて、(6.13) は、各時刻ステップにおいて次のように近似できる。

$$\begin{aligned} z_{k+1} &\approx Az_k + \sum_{i=1}^m B_i z_k^* u_k(i) - \gamma \left(z_k - \left(h(x_k^*) + \frac{\partial h}{\partial x} \Big|_{x=x_k^*} (Pz_k - x_k^*) \right) \right) \\ &= \tilde{A}(z_k^*)z_k + \tilde{B}(z_k^*)u_k + \tilde{W}(z_k^*) \end{aligned} \quad (6.18)$$

ここで、

$$\tilde{A}(z_k^*) = A - \gamma \left(I_{n_z} - \frac{\partial h}{\partial x} \Big|_{x=Pz_k^*} P \right) \quad (6.19a)$$

$$\tilde{B}(z_k^*) = [B_1 z_k^*, \dots, B_m z_k^*] \quad (6.19b)$$

$$\tilde{W}(z_k^*) = \gamma \left(h(Pz_k^*) - \frac{\partial h}{\partial x} \Big|_{x=Pz_k^*} Pz_k^* \right) \quad (6.19c)$$

である．(6.18) は，時刻ステップ k に応じて係数行列が変化する，線形時変システムとみなすことができる．

以上より，事前予測状態 z_k^* , $k = 0, \dots, N_p$ が得られているものとした時，LMPC の各制御周期において解くべき最適化問題は次のように与えられる．

$$\underset{u_k}{\text{minimize}} \quad J \quad (6.20a)$$

$$\text{subject to} \quad z_0 = h(x_t) \quad (6.20b)$$

$$z_{k+1} = \tilde{A}(z_k^*)z_k + \tilde{B}(z_k^*)u_k + \tilde{W}(z_k^*), \quad k = 0, \dots, N_p - 1 \quad (6.20c)$$

$$x_k = Pz_k, \quad k = 0, \dots, N_p \quad (6.20d)$$

$$A_{con} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_{con} \geq 0, \quad k = 0, \dots, N_p \quad (6.20e)$$

最適化問題 (6.20) は，初期ステップにおける拡大状態 z_0 を求める (6.20b) を除き，二次形式の評価関数と線形の制約条件で構成されており，高速な QP ソルバで解くことが可能な定式化となっている．

6.3.4 LMPC アルゴリズム

以上の議論より，本稿で提案する拡大双線形モデルに基づく LMPC アルゴリズムは，Algorithm 4 のようにまとめることができる．

Algorithm 4 LBM-LMPC

Require: Reference trajectory (x^r, u^r)

while Controller is running **do**

if $t=0$ **then**

 Set the prior-predicted states z_k^* by (6.16)

end if

 Observe current state x_t of the system

 Solve the optimization problem (6.20) with the prior-predicted states z_k^*

 Update the prior-predicted states z_k^* by (6.15)

 Apply first input u_0 to the system

end while

本提案手法によれば，事前予測状態を用いて線形近似した拡大双線形モデルに基づく状態予測と，観測関数に基づく誤差補償を，同時に実行することができ，反復的に事前予測状態および予測モデルの精度を改善することができる．さらに，4 章の提案手法と比較して，最適化問題 (6.20) では制約条件と最適化変数の数が増加しないため，観測関数や予測ホライゾンの長さに応じて計算コストが大幅に増大することがないという利点がある．

6.4 数値シミュレーション

本節では，6.2 節で示したクアッドロータモデルを対象システムとして，提案手法の有効性をシミュレーションにより評価した結果を説明する．

6.4.1 評価条件

本シミュレーションでは，以下の4つの手法に関する比較を行った．

- NMPC: 真のモデルに基づく NMPC
- Nom-LMPC: ノミナルモデルに基づく LMPC
- LBM-LMPC(Conv.): 4 章の提案手法による拡大双線形モデルに基づく LMPC
- LBM-LMPC(Proposed): 本章の提案手法による拡大双線形モデルに基づく LMPC

なお，Nom-LMPC については，4 章と同様に，姿勢が大きく傾いた際の非線形性を考慮の有無による性能の違いを評価することを目的とした．ここで，ノミナルモデルは，平衡状態 $\bar{x} = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]^T$ ，平衡入力 $\bar{u} = [0, 0, 0, 9.81]^T$ 近傍で，1 次のテーラー展開を行って導出した．

ここでは，上記の各手法を用いて，10 点の初期位置 p_0 と初期姿勢 q_0 を一様にランダムに取得し，それらの初期値から原点へ安定化する制御を各手法で実行した．ここで，初期位置と初期姿勢の範囲は， $[-1, -1, -1] \leq p_0^T \leq [1, 1, 1]$ ， $[0, -1, -1, -1] \leq q_0^T \leq [1, 1, 1, 1]$ とし， q_0 についてはランダムな値を取得後，ノルムが 1 となるように正規化した．また，サンプル時間は $\Delta t = 20[\text{ms}]$ とした．予測ホライゾンの長さは基本的に $N_p = 30$ とし，計算時間の評価においてのみ $N_p = 10, 20, 30$ とした．評価関数 J における重み行列は， $P = 100 \cdot I_{10}$ ， $Q = I_{10}$ ， $R = 0.1 \cdot I_4$ とした．入力に関する制約条件として， $T \geq 0$ とした．また，提案手法における誤差補償のフィードバックゲインは，行列 $\tilde{A}(z_k^*)$ の固有値の絶対値が 1 以下となるように選び， $\gamma = 1$ とした．

なお，NMPC のソルバには CasADi [61] を介して IPOPT を使用し，提案手法および Nominal MPC のソルバには MATLAB の quadprog 関数を使用した．シミュレーション実行環境には Core i5 CPU@1.20GHz, 24.0GB メモリの Windows 10 の PC 上で動作する MATLAB 2020b を使用した．

6.4.2 制御性能の評価結果

Fig. 6.2 に，それぞれの手法による全ての初期値からの軌道を，3 次元空間で表示した結果を示す．この結果より，全ての手法において，対象システムを全ての初期値から原点に安定化できているが，特に Nom-LMPC は，他の手法に対して大きく逸脱した軌道となっていることが分かる．これは，ノミナルモデルでは姿勢が大きく傾いた際のダイナミクスの非線形性を

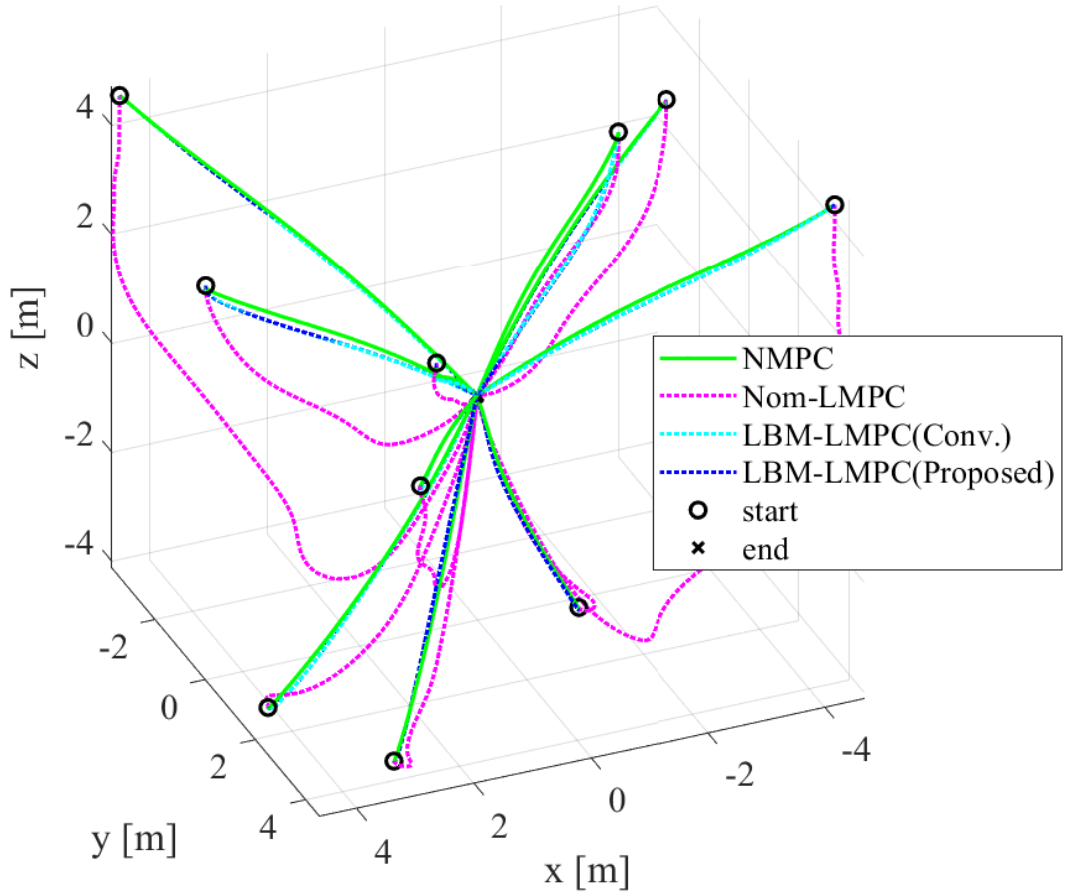


Fig. 6.2: 3D trajectories from 10 random initial points

考慮できていないためであると考えられる．一方で，NMPC に対して，LBM-LMPC(Conv.) および LBM-LMPC(Proposed) は，ほぼ同等の軌道となっていることが分かる．

次に，このうちのある初期値からの軌道に関する状態と入力の変時間変化グラフを，Fig. 6.3 および Fig. 6.4 にそれぞれ示す．これらの結果より，対象の軌道においては，Nom-LMPC では状態と入力がともに大きく乱れているのに対し，LBM-LMPC(Conv.) および LBM-LMPC(Proposed) は NMPC とほぼ同等の振る舞いを達成できていることが分かる．特に，LBM-LMPC(Conv.) と LBM-LMPC(Proposed) の挙動が全ての状態と入力に関してほとんど一致していることが分かる．これより，提案手法において導入した誤差補償は，4 章の提案手法における誤差補償とほぼ同等の効果を発揮できていると考えられる．また，ここではあるひとつの軌道に着目した結果を示したが，他の初期値からの軌道についても同様の結果がみられた．

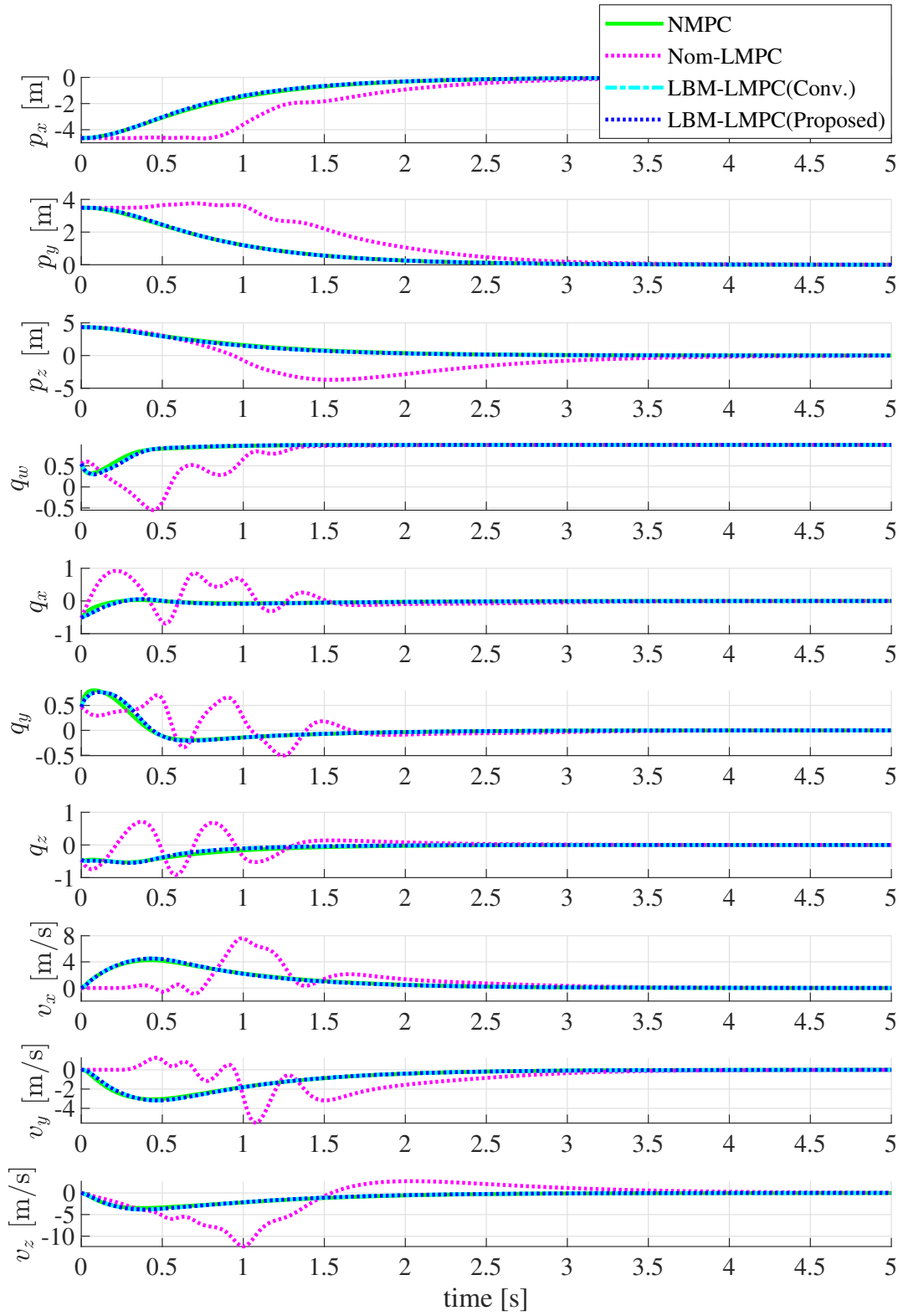


Fig. 6.3: States comparison for a single trajectory

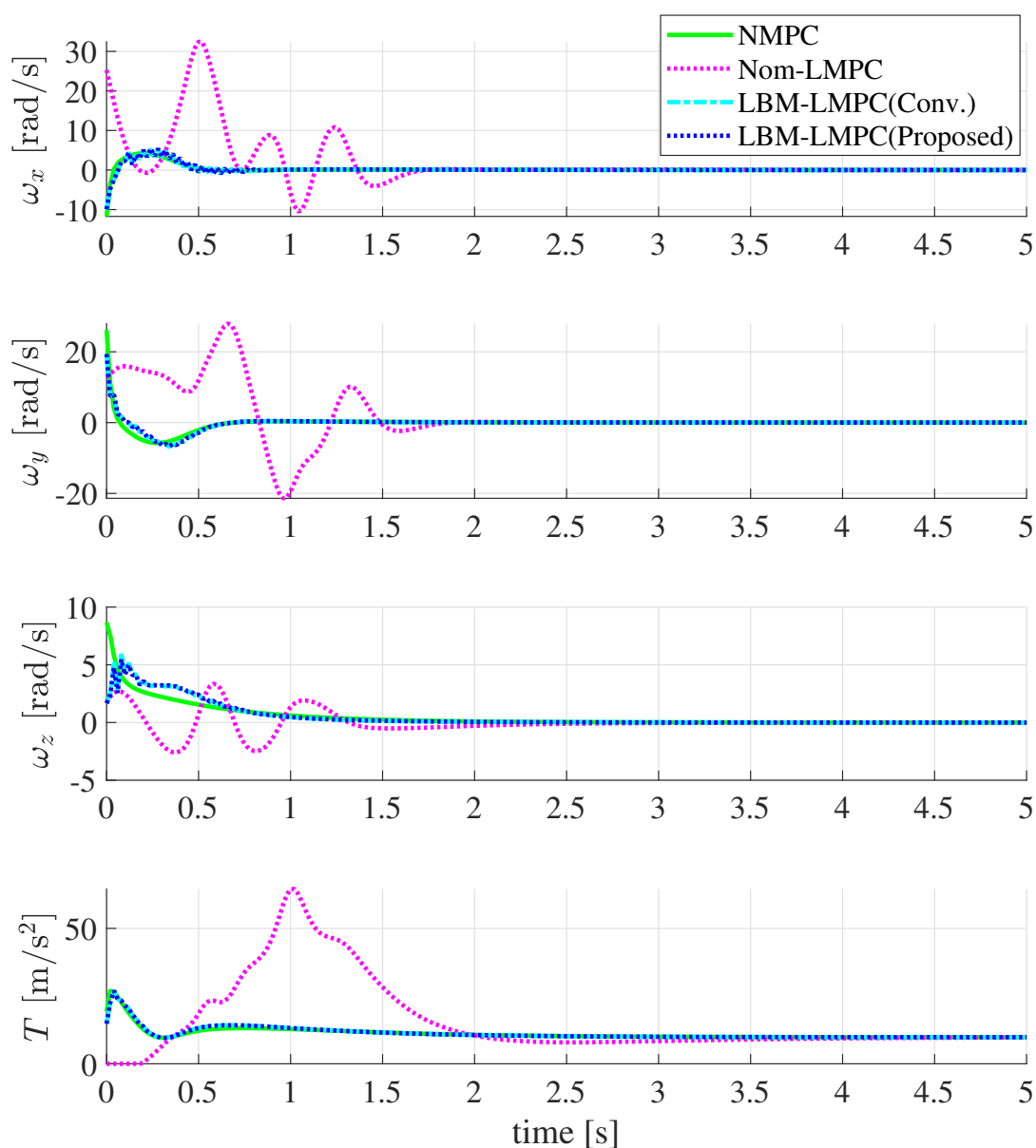


Fig. 6.4: Inputs comparison for a single trajectory

続いて、Fig. 6.5 に、シミュレーション時間全体に対する評価関数値の比較結果を示す。ここでは、各手法の値を NMPC の値で正規化し、全軌道分の平均値を算出している。この結果より、Nom-LMPC では NMPC と比較して評価関数値が 74% と大幅に悪化しているのに対し、LBM-LMPC(Conv.) と LBM-LMPC(Proposed) では、双方ともに 1% しか性能の違いが無く、高性能な制御を実現できていることがわかる。

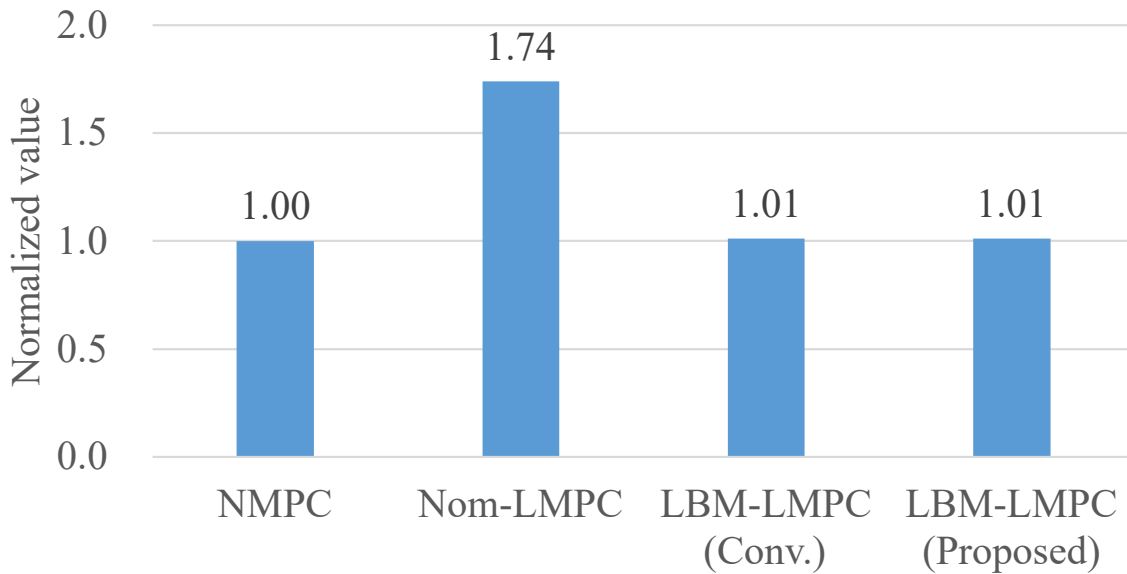


Fig. 6.5: Normalized objective function values

6.4.3 計算時間の評価結果

異なる予測ホライゾンの長さ ($N_p = 10, 20, 30$) に対して、各手法における最適化ソルバの実行時間を測定した結果を、Fig. 6.6 に示す。この結果より、まず、NMPC と LBM-LMPC(Conv.) では、予測ホライゾンの長さとともに計算時間が著しく増大していることが分かる。特に、LBM-LMPC(Conv.) においては、予測ホライゾン長が小さい場合には NMPC よりも高速な求解ができていたが、予測ホライゾン長が大きいと、むしろ NMPC よりも計算時間が大きくなってしまっている。4 章の数値シミュレーションでは、対象システムに拡大双線形化において設定した観測関数の数が少なかったため、 $N_p = 30$ の場合でも提案手法の優位性を示せたが、本章の対象システムのように、より複雑な対象では考慮すべき観測関数の数が多くなり、結果として計算時間が増大してしまうという課題があることが分かる。

これに対し、Nom-LMPC と LBM-LMPC(Proposed) では、今回用いた予測ホライゾンの範囲内では計算時間が十分に小さく、またその増加は緩やかである。この結果より、本章の提案手法 LBM-LMPC(Proposed) によれば、4 章の提案手法 LBM-LMPC(Conv.) で課題となる計算時間の増大を解決し、観測関数の数と予測ホライゾンに関してスケーラビリティを向上できていることが分かる。なお、LBM-LMPC(Proposed) の方が若干計算時間の増大幅がやや大きいのは、状態の拡大によって予測モデルの次数が大きくなっている影響であると考えられるが、今回のシミュレーション条件を踏まえれば、実用上十分にリアルタイム性を担保できると考えられる。

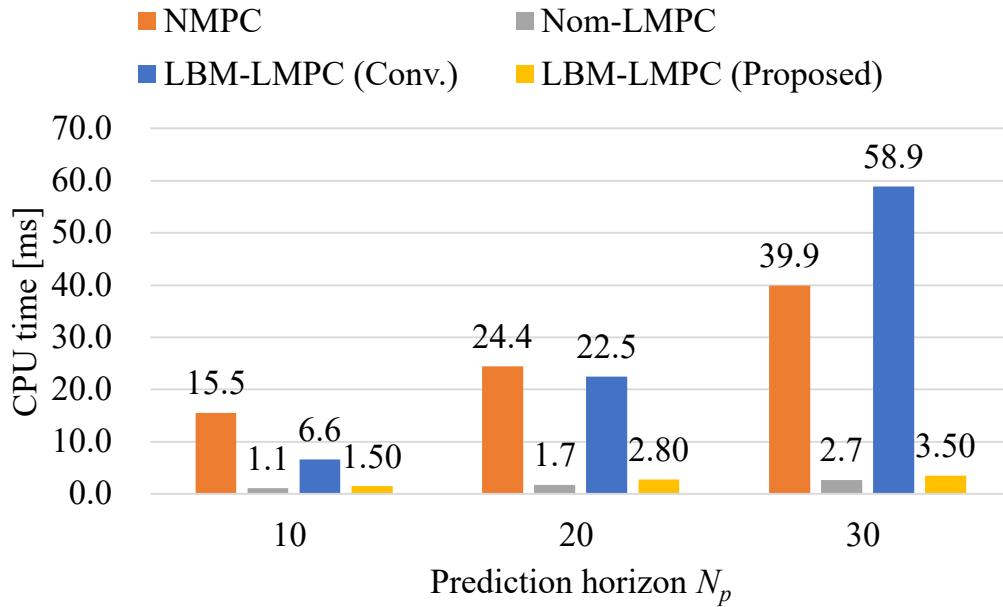


Fig. 6.6: CPU time

6.5 まとめ

本章では、4章の提案手法におけるスケーラビリティの課題を解決し、実践的な対象システムに対しても適用可能な、拡大双線形モデルに基づく LMPC を提案した。ここでは、対象システムとして3次元空間で動作するクアッドロータを取り上げ、まずその解析的な拡大双線形モデルを導出した。提案手法では、拡大双線形モデルを用いた MPC の予測モデルにおいて、拡大状態の各要素が満たすべき観測関数に基づく数値的関係の誤差をフィードバックする、新たな予測誤差補償手法を導入した。これにより、従来手法の課題であった、観測関数の数と予測ホライズンの長さに対する制約条件と最適化変数の増大を回避し、スケーラビリティを改善した LMPC アルゴリズムを構築した。

数値シミュレーションでは、対象システムであるクアッドロータに対する安定化制御を行い、提案手法の有効性を検証した。その結果、提案手法によって、真のモデルを用いた NMPC と同等の制御性能を達成し、かつ計算時間の大幅な低減が可能であることを確認できた。特に、比較対象とした従来手法では予測ホライズン長に対して計算時間が大幅に増大してしまうのに対し、提案手法では計算時間の増大を十分に抑制することができ、スケーラビリティが大幅に改善されることを確認できた。なお、本章の予測誤差補償手法は、3章と5章の提案手法に対しても同様の考え方を適用することで、スケーラビリティを改善可能であると期待できる。

なお、本章の提案手法では、拡大状態の誤差補償と、誤差補償を行うための制約条件の誤差の収束を同時に行っている。そのため、拡大双線形モデルの精度が低い場合には、拡大状態の誤差増大の影響により、これらを両立できない可能性も考えられる。このような、拡大

双線形モデルの精度に関する提案手法の有効性の詳細分析については、今後の検討課題である。また、その他の課題としては、今回はクアッドロータの解析的な拡大双線形モデルを用いたが、データドリブンなモデル同定手法と組合せた場合の有効性検証や、実機実験による有効性検証が挙げられる。

第7章 結論

本研究では，Koopman アプローチを活用し，高精度な予測に基づく高い制御性能と，低計算コストを両立する MPC を提案した．

2 章では，本研究に関する事前知識として，MPC と拡大線形化に関する基礎的な内容を紹介した．

3 章では，対象の非線形システムが既知であるという仮定のもと，その非線形要素に基づき拡大線形化を行う，近似拡大線形化を導出した．本手法では，予測ホライゾン上の最適状態列の予測値である事前予測状態を活用することで，非線形システムを高次元空間において線形時変システムとして近似する．さらに，近似拡大線形化に基づく予測モデルと，拡大状態に関する制約条件を利用した予測誤差補償を考慮した LMPC を提案した．数値シミュレーションで提案手法の有効性検証を行い，対象システムの非線形性を考慮した高精度な予測と低計算コストを両立した MPC を実現できることを示した．

4 章では，対象の非線形システムが未知の入力アフィンシステムであるという仮定のもと，データドリブンに拡大双線形モデルを同定する方法を示し，さらに得られた拡大双線形モデルに基づく LMPC を提案した．提案手法では，3 章の手法を応用し，拡大双線形モデルを事前予測状態を活用して線形化し，さらに拡大状態に関する制約条件を利用して予測誤差補償を行う．数値シミュレーションで提案手法の有効性検証を行い，未知の対象システムであってもデータに基づき非線形性を捉えた拡大双線形モデルを同定し，それに基づき高精度な予測と低計算コストを両立した MPC を実現できることを示した．

5 章では，4 章の手法を応用し，対象システムに関するノミナルモデルが得られている場合に，その実プラントに対するモデル誤差を補償することで高性能な MPC を実現する方法を提案した．まず，実プラントに対して事前に得られているノミナルモデルのモデル誤差を拡大双線形モデル誤差モデル (MEM) として同定する方法を示し，ノミナルモデルと拡大双線形 MEM を統合した誤差補償モデルを導出した．そして，誤差補償モデルを予測モデルとし，4 章の手法を応用して，事前予測状態を活用した線形化と予測誤差補償とを考慮した LMPC を提案し，その有効性を数値シミュレーションにより示した．

6 章では，4 章の提案手法における観測関数の数と予測ホライゾン長に関するスケーラビリティの課題を解決するため，新たな予測誤差補償手法を用いた拡大双線形モデルに基づく LMPC を提案した．提案手法では，拡大状態における各要素が満たすべき数値的關係の誤差をフィードバックする補償入力を新たに導入した．これによって，4 章の手法と異なり，予測誤差補償によって最適化問題における制約条件と最適化変数が増大しないため，観測関数の数や予測ホライゾン長に対する計算コストの大幅な増大を回避することができる．また，

実践的な対象システムとして3次元空間で動作するクアッドロータを取り上げ、数値シミュレーションによって、制御性能と計算時間に関して従来手法に対する提案手法の優位性を示すことができた。

これらの結果は、近年盛んに研究されているKoopman理論に基づくアプローチを活用したMPCにおいて、新たな観点からその高性能化と計算コスト低減の可能性を示したものであり、この分野の研究ならびに応用を大きく進展させると期待している。

以降では、各章の提案手法の比較に関する議論と、今後の課題について述べる。

まず、4章と5章の提案手法については、どちらもデータに基づき拡大双線形モデルを同定しLMPCを構成する点は同様であり、適切な観測関数を選定した場合、達成される制御性能についてはほぼ同等であると考えられる。4章の提案手法の利点としては、ノミナルモデルを考慮しない分、予測モデルが低次となり計算コストについて優位性がある。これに対し、5章の提案手法は、ノミナルモデルを基準に予測モデルを構築するため、同定データのPE性に関わらず、全体のモデルの挙動が大きく異なる懸念が少ないという利点がある。また、産業応用の観点では、物理ベースのモデルを取り入れることによる設計上の解釈容易性の向上や、経時変化などに起因したモデル誤差に対するコントローラ再設計工数の低減などが利点として挙げられる。

また、6章では4章の提案手法と比較して計算コストを低減する手法を提案した。ただし、前述のように、6章の提案手法では拡大状態の誤差補償と、拡大状態に関する制約条件の誤差の収束を同時に行うため、拡大双線形モデルの精度が低い場合には性能が低下する可能性がある。そのため、提案手法が有効となる条件については今後の検討課題である。これに対し、4章の提案手法では、陽に拡大状態に関する制約条件を設定し、拡大状態の誤差を強制的に補償するため、制御性能は6章の提案手法よりも高いと期待できる。ただし、計算コストが大きいというデメリットはあるため、問題のサイズに対して制御周期が十分に大きい場合には有効な手法であると言える。

本研究では、基本的に時不変な非線形入力アフィンシステムを対象として検討を行ったが、より一般的な時不変の非線形システムについても、入力的时间微分を新たな入力とし、元の入力を状態変数として捉えることにより、入力アフィンシステムに変換することが可能であり[65,66]、本研究の提案手法を適用可能であると考えられる。一方、より多様なクラスのシステムに適用するための拡張は今後の課題である。拡大線形化に関する先行研究としては、対象システムとして時変システムを扱ったもの[67,68]や、ハイブリッドシステムを扱ったもの[69]もある。これらのような研究の応用によっても、提案手法のさらなる発展が期待できる。

また、本研究では各提案手法の有効性を数値シミュレーションにより検証したが、実機実験による有効性検証は今後の重要課題である。特に、本研究では基本的に観測関数を物理的知見に基づいて選定したが、実機を用いる際にはモデルの不確かさによって適切な観測関数の選定が難しい場合がある。その影響で、十分な精度のモデルを得られない恐れもあり、そのような場合の提案手法の有効性の範囲を検証することは必要であると考えられる。また、データに基づいて適切な観測関数を学習する手法[42,43]などとも組み合わせ、さらなる実用性向上をめざすことも課題であると考えられる。

付 録 A LMPCにおける最適制御問題の QP問題への変換

A.1 線形時不変システムの場合

LMPCにおいて、最適制御問題をQP問題の形に変換することで、QPソルバを適用して解を求めることができる。本節では、予測モデルを線形時不変システムとし、次の最適制御問題を考える。

$$\underset{u_k}{\text{minimize}} \quad J_l \quad (\text{A.1a})$$

$$\text{subject to} \quad x_0 = x_t \quad (\text{A.1b})$$

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad k = 0, \dots, N_p - 1 \quad (\text{A.1c})$$

$$y_k = Cx_k, \quad k = 0, \dots, N_p \quad (\text{A.1d})$$

ここで、 $x_k \in \mathbb{R}^n$ を状態、 $u_k \in \mathbb{R}^m$ を入力、 $w_k \in \mathbb{R}^n$ を外乱項、 $y_k \in \mathbb{R}^p$ を出力とする。また、 N_p は予測ホライズン、 x_t は時刻 t におけるシステムの状態とする。評価関数 J_l は、次式で与えられるものとする。

$$J_l = \|y_{N_p} - y_{N_p}^r\|_P^2 + \sum_{k=0}^{N_p-1} (\|y_k - y_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2) \quad (\text{A.2})$$

ここで、 $P, Q \in \mathbb{R}^{p \times p}$ は準正定、 $R \in \mathbb{R}^{m \times m}$ は正定な重み行列であり、 $y_k^r, k = 0, \dots, N_p$ と $u_k^r, k = 0, \dots, N_p - 1$ はそれぞれ目標出力と目標入力とする。なお、ここでは状態と入力に関するその他の制約条件は考えないものとする。

まず、評価関数 J_l は次のように式展開できる。

$$\begin{aligned} J_l &= (y_{N_p} - y_{N_p}^r)^T P (y_{N_p} - y_{N_p}^r) + \sum_{k=0}^{N_p-1} ((y_k - y_k^r)^T Q (y_k - y_k^r) + (u_k - u_k^r)^T R (u_k - u_k^r)) \\ &= (y_0 - y_0^r)^T Q (y_0 - y_0^r) + (Y - Y_r)^T \bar{Q} (Y - Y_r) + (U - U_r)^T \bar{R} (U - U_r) \end{aligned} \quad (\text{A.3})$$

ここで,

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_{N_p} \end{bmatrix} \in \mathbb{R}^{pN_p}, \quad Y_r = \begin{bmatrix} y_1^r \\ \vdots \\ y_{N_p}^r \end{bmatrix} \in \mathbb{R}^{pN_p}, \quad \bar{Q} = \begin{bmatrix} Q & & \\ & \ddots & \\ & & Q \\ & & & P \end{bmatrix} \in \mathbb{R}^{pN_p \times pN_p}, \quad (A.4)$$

$$U = \begin{bmatrix} u_0 \\ \vdots \\ u_{N_p-1} \end{bmatrix} \in \mathbb{R}^{mN_p}, \quad U_r = \begin{bmatrix} u_0^r \\ \vdots \\ u_{N_p-1}^r \end{bmatrix} \in \mathbb{R}^{mN_p}, \quad \bar{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix} \in \mathbb{R}^{mN_p \times mN_p}$$

とした.

また, $X = [x_1^T, \dots, x_{N_p}^T]^T$ とすると, これは (A.1c) を用いて次のように式展開できる.

$$\begin{aligned} X = \begin{bmatrix} x_1 \\ \vdots \\ x_{N_p} \end{bmatrix} &= \begin{bmatrix} Ax_0 + Bu_0 + w_0 \\ Ax_1 + Bu_1 + w_1 \\ \vdots \\ Ax_{N_p-1} + Bu_{N_p-1} + w_{N_p-1} \end{bmatrix} = \begin{bmatrix} Ax_0 + Bu_0 + w_0 \\ A(Ax_0 + Bu_0 + w_0) + Bu_1 + w_1 \\ \vdots \\ \vdots \end{bmatrix} \\ &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix} x_0 + \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & & \ddots & \\ A^{N_p-1}B & A^{N_p-2}B & \dots & B \end{bmatrix} U + \begin{bmatrix} I_n & & & \\ A & I_n & & \\ \vdots & & \ddots & \\ A^{N_p-1} & A^{N_p-2} & \dots & I_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N_p-1} \end{bmatrix} \\ &= Sx_0 + TU + VW \end{aligned} \quad (A.5)$$

ここで,

$$S = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}, \quad T = \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & & \ddots & \\ A^{N_p-1}B & A^{N_p-2}B & \dots & B \end{bmatrix}, \quad (A.6)$$

$$V = \begin{bmatrix} I_n & & & \\ A & I_n & & \\ \vdots & & \ddots & \\ A^{N_p-1} & A^{N_p-2} & \dots & I_n \end{bmatrix}, \quad W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N_p-1} \end{bmatrix}$$

とした.

さらに, Y は (A.5) を用いて次のように表せる.

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_{N_p} \end{bmatrix} = \begin{bmatrix} Cx_1 \\ \vdots \\ Cx_{N_p} \end{bmatrix} = \bar{C}X \quad (A.7)$$

ここで,

$$\bar{C} = \begin{bmatrix} C & & \\ & \ddots & \\ & & C \end{bmatrix} \quad (\text{A.8})$$

とした.

以上より, (A.5), (A.7) を用いて (A.3) をさらに式展開し, 評価関数 J_l を x_0, U の関数として以下のように表現することができる.

$$\begin{aligned} J_l(x_0, U) &= (y_0 - y_0^r)^T Q (y_0 - y_0^r) \\ &\quad + (\bar{C}(Sx_0 + TU + VW) - Y_r)^T \bar{Q} (\bar{C}(Sx_0 + TU + VW) - Y_r) \\ &\quad + (U - U_r)^T \bar{R} (U - U_r) \\ &= U^T (T^T \bar{C}^T \bar{Q}^T \bar{C} T + \bar{R}) U + 2 \{ (\bar{C}(Sx_0 + VW) - Y_r)^T \bar{Q} \bar{C} T - U_r^T \bar{R} \} U + (\text{定数項}) \\ &= U^T F U + 2 G U + (\text{定数項}) \end{aligned} \quad (\text{A.9})$$

ここで,

$$\begin{aligned} F &= T^T \bar{C}^T \bar{Q}^T \bar{C} T + \bar{R} \\ G &= (\bar{C}(Sx_0 + VW) - Y_r)^T \bar{Q} \bar{C} T - U_r^T \bar{R} \end{aligned} \quad (\text{A.10})$$

とした.

$J_l(x_0, U)$ は U に関する 2 次関数となっている. したがって, 最適制御問題 (A.1) は, 次のようにベクトル U を最適化変数とした制約条件なしの QP 問題に帰着することができる.

$$\underset{U}{\text{minimize}} \quad J_l(x_0, U) \quad (\text{A.11})$$

また, このように変換された問題を, 最適制御問題 (A.1) の密形式と呼ぶ.

なお, ここでは詳細を割愛するが, 状態と入力に関する線形制約条件がある場合については, それらに関して入力ベクトル U に対する係数行列を整理することにより, 線形制約条件付きの QP 問題に容易に帰着することができる.

A.2 線形時変システムの場合

本節では, A.1 節で説明した式展開において, 予測モデルが線形時変システムである場合の違いについて説明する. ここでは A.1 節と重複する変数の説明は省略する. 次の最適制御問題を考える.

$$\underset{u_k}{\text{minimize}} \quad J_l \quad (\text{A.12a})$$

$$\text{subject to} \quad x_0 = x_t \quad (\text{A.12b})$$

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, \dots, N_p - 1 \quad (\text{A.12c})$$

$$y_k = C_k x_k, \quad k = 0, \dots, N_p \quad (\text{A.12d})$$

ここで, A_k, B_k, C_k は時変行列とする.

この時, A.1 節の (A.5) における X の式展開は次のようになる.

$$\begin{aligned}
X = \begin{bmatrix} x_1 \\ \vdots \\ x_{N_p} \end{bmatrix} &= \begin{bmatrix} A_0 x_0 + B_0 u_0 + w_0 \\ A_1 x_1 + B_1 u_1 + w_1 \\ \vdots \\ A_{N_p-1} x_{N_p-1} + B_{N_p-1} u_{N_p-1} + w_{N_p-1} \end{bmatrix} \\
&= \begin{bmatrix} A_0 x_0 + B_0 u_0 + w_0 \\ A_1 (A_0 x_0 + B_0 u_0 + w_0) + B_1 u_1 + w_1 \\ \vdots \\ \vdots \end{bmatrix} \\
&= \begin{bmatrix} A_0 \\ A_1 A_0 \\ \vdots \\ A_{N_p-1} \cdots A_0 \end{bmatrix} x_0 + \begin{bmatrix} B_0 & & & \\ A_1 B_0 & B_1 & & \\ \vdots & & \ddots & \\ A_{N_p-2} \cdots A_0 B_0 & A_{N_p-3} \cdots A_0 B_1 & \cdots & B_{N_p-1} \end{bmatrix} U \\
&\quad + \begin{bmatrix} I_n & & & \\ A_1 & I_n & & \\ \vdots & & \ddots & \\ A_{N_p-2} \cdots A_0 & A_{N_p-3} \cdots A_0 & \cdots & I_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N_p-1} \end{bmatrix} \\
&= Sx_0 + TU + VW
\end{aligned} \tag{A.13}$$

ここで,

$$\begin{aligned}
S &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}, \quad T = \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & & \ddots & \\ A^{N_p-1} B & \cdots & \cdots & B \end{bmatrix}, \\
V &= \begin{bmatrix} I_n & & & \\ A_1 & I_n & & \\ \vdots & & \ddots & \\ A_{N_p-2} \cdots A_0 & A_{N_p-3} \cdots A_0 & \cdots & I_n \end{bmatrix}, \quad W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N_p-1} \end{bmatrix}
\end{aligned} \tag{A.14}$$

とした.

また, A.1 節の (A.7) における Y は次のように表せる.

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_{N_p} \end{bmatrix} = \begin{bmatrix} C_1 x_1 \\ \vdots \\ C_{N_p} x_{N_p} \end{bmatrix} = \bar{C} X \tag{A.15}$$

ここで,

$$\bar{C} = \begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_{N_p} \end{bmatrix} \quad (\text{A.16})$$

とした.

以上が, A.1 節に対する予測モデルが時変システムの場合の違いである. (A.13), (A.15) を用いて評価関数 J_l を式展開することで, A.1 節と同様に, 最適化問題 (A.12) をベクトル U を最適化変数とした制約条件なしの QP 問題に帰着することができる.

付 録 B 平面ドローンノミナルモデルの 可制御性

対象システム (4.18) において, $\theta = 0$ 近傍で線形システムに近似する場合の, 入力を選び方による可制御性の違いについて説明する. まず, 入力を T_1, T_2 とする場合, 入力の係数行列において $\theta = 0$ とすると, 近似線形システムは次のようになる.

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} &= \begin{bmatrix} O_{3 \times 3} & I_3 \\ O_{3 \times 3} & O_{3 \times 3} \end{bmatrix} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ -\frac{l_a}{I_{xx}} & \frac{l_a}{I_{xx}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -g \\ 0 \end{bmatrix} \\ &= A_1 x + B_1 u + W_1 \end{aligned} \quad (\text{B.1})$$

この時, 係数行列 A_1, B_1 を用いて可制御性行列 $M_{c1} = [B_1 \ A_1 B_1 \ \dots \ A_1^{n-1} B_1]$ を計算すると, M_{c1} のランクは 4 となり, システム (B.1) は不可制御である.

一方で, 入力を $\Delta T_1, \Delta T_2$ とする場合を考える. まず, $T_i = \frac{1}{2} + \Delta T_i, i = 1, 2$ を (4.18) に代入すると, 次式が得られる.

$$\frac{d}{dt} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} = \begin{bmatrix} v_y \\ v_z \\ \omega \\ -g \sin \theta \\ g(\cos \theta - 1) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta \\ \frac{1}{m} \cos \theta & \frac{1}{m} \cos \theta \\ -\frac{l_a}{I_{xx}} & \frac{l_a}{I_{xx}} \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} \quad (\text{B.2})$$

これに対し, 第 1 項については $\theta = 0$ 近傍で線形近似し, 第 2 項の入力の係数行列において $\theta = 0$ とすると, $\Delta T_1, \Delta T_2$ を入力とした非線形システム (B.2) の近似線形システムは次のよ

うになる.

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} &= \begin{bmatrix} O_{3 \times 3} & I_3 \\ \begin{bmatrix} 0 & 0 & -g \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & O_{3 \times 3} \end{bmatrix} \begin{bmatrix} p_y \\ p_z \\ \theta \\ v_y \\ v_z \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ -\frac{l_a}{I_{xx}} & \frac{l_a}{I_{xx}} \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} \\ &= A_2 x + B_2 u \end{aligned} \quad (\text{B.3})$$

この時, 係数行列 A_2, B_2 を用いて可制御性行列 $M_{c2} = [B_2 \ A_2 B_2 \ \dots \ A_2^{n-1} B_2]$ を計算すると, M_{c2} のランクは 6 となり, システム (B.3) は可制御となる.

したがって, 4 章の数値シミュレーションにおいては, $\Delta T_i, i = 1, 2$ を入力として選択した.

参考文献

- [1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [2] L. Grüne, and J. Pannek, “Nonlinear model predictive control,” in *Nonlinear model predictive control*. Springer, 2017, pp. 45–69.
- [3] 大塚敏之, 非線形最適制御入門 コロナ社, 2011.
- [4] 足立修一, 管野政明, モデル予測制御: 制約のもとの最適制御 東京電機大学出版局, 2005.
- [5] 大塚敏之, “非線形モデル予測制御の研究動向”, システム/制御/情報, vol. 61, no. 2, pp. 42–50, 2017.
- [6] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on model predictive control: an engineering perspective,” *The International Journal of Advanced Manufacturing Technology*, 2021.
- [7] S. J. Qin, and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [8] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, “Model predictive control of power electronic systems: Methods, results, and challenges,” *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020.
- [9] 滑川徹, “スマートグリッドのための分散予測制御”, 計測と制御, vol. 51, no. 1, pp. 62–68, 2012.
- [10] 大塚敏之, 浜松正典, 永塚満, 川辺武俊, 向井正和, Md Abdus Samad Kamal, 西羅光, 山北昌毅, 李俊默, 橋本智昭, 実時間最適化による制御の実応用 コロナ社, 2015.
- [11] 仲田勇人, 白井隼人, 松永彰生, 富永浩之, “ディーゼルエンジン吸排気システムへのC/GMRES モデル予測制御の応用”, 計測自動制御学会論文集, vol. 54, no. 2, pp. 175–181, 2018.
- [12] A. Musa, M. Pipicelli, M. Spano, F. Tufano, F. De Nola, G. Di Blasio, A. Gimelli, D. A. Misul, and G. Toscano, “A review of model predictive controls applied to advanced driver-assistance systems,” *Energies*, vol. 14, no. 23, p. 7974, 2021.

-
- [13] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, “Model predictive control power management strategies for hevs: A review,” *Journal of Power Sources*, vol. 341, pp. 91–106, 2017.
- [14] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, “Model predictive control in aerospace systems: Current state and opportunities,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.
- [15] 濱田吉郎, “非線形モデル予測制御: 誘導制御技術としての期待と無人機による飛行実験”, 計測と制御, vol. 61, no. 2, pp. 91–96, 2022.
- [16] Y. Ding, L. Wang, Y. Li, and D. Li, “Model predictive control and its application in agriculture: A review,” *Computers and Electronics in Agriculture*, vol. 151, pp. 104–117, 2018.
- [17] D. Mariano-Hernández, L. Hernández-Callejo, A. Zorita-Lamadrid, O. Duque-Pérez, and F. S. García, “A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis,” *Journal of Building Engineering*, vol. 33, p. 101692, 2021.
- [18] T. Ohtsuka, “A continuation/GMRES method for fast computation of nonlinear receding horizon control,” *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.
- [19] H. Deng, and T. Ohtsuka, “A parallel Newton-type method for nonlinear model predictive control,” *Automatica*, vol. 109, p. 108560, 2019.
- [20] T. Parisini, and R. Zoppoli, “A receding-horizon regulator for nonlinear systems and a neural approximation,” *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.
- [21] 森安竜大, 上田松栄, 池田太郎, 永岡真, 神保智彦, 松永彰生, 中村俊洋, “機械学習によるディーゼルエンジン吸排気系の実時間 MPC 設計”, 計測自動制御学会論文集, vol. 55, no. 3, pp. 172–180, 2019.
- [22] B. Koopman, “Hamiltonian Systems and Transformations in Hilbert Space,” *Proc. Natl. Acad. Sci. USA*, vol. 17, no. 5, pp. 315–318, 1931.
- [23] B. O. Koopman, and J. v. Neumann, “Dynamical systems of continuous spectra,” *Proceedings of the National Academy of Sciences*, vol. 18, no. 3, pp. 255–263, 1932.
- [24] M. Korda, and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018.
- [25] E. Kaiser, J. N. Kutz, and S. L. Brunton, “Data-driven discovery of Koopman eigenfunctions for control,” *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035023, 2021.

-
- [26] 薄良彦, “クープマン作用素による非線形システムの制御”, 計測と制御, vol. 61, no. 2, pp. 133–138, 2022.
- [27] S. L. Brunton, and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.
- [28] S. E. Otto, and C. W. Rowley, “Koopman operators for estimation and control of dynamical systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 59–87, 2021.
- [29] P. Bevanda, S. Sosnowski, and S. Hirche, “Koopman operator dynamical models: Learning, analysis and control,” *Annual Reviews in Control*, vol. 52, pp. 197–212, 2021.
- [30] 薄良彦, “クープマン作用素による大規模非線形ダイナミカルシステムの解析”, 計測と制御, vol. 56, no. 12, pp. 931–936, 2017.
- [31] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman operator in systems and control*. Springer, 2020.
- [32] M. Netto, and L. Mili, “A robust data-driven Koopman Kalman filter for power systems dynamic state estimation,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7228–7237, 2018.
- [33] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, “Spectral analysis of nonlinear flows,” *Journal of fluid mechanics*, vol. 641, pp. 115–127, 2009.
- [34] I. Mezić, “Analysis of fluid flows via spectral properties of the Koopman operator,” *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.
- [35] Phan Tuy NM, Wells John C, Susuki Yoshihiko, Nguyen Linh V, Bonner James S, Islam Mohammad S, Kirkey William D, “Koopman modes in a near-bank region of a tidal river”, 土木学会論文集 B1 (水工学), vol. 71, no. 4, pp. I.205–I.210, 2015.
- [36] Y. Susuki, I. Mezic, F. Raak, and T. Hikiyara, “Applied Koopman operator theory for power systems technology,” *Nonlinear Theory and Its Applications, IEICE*, vol. 7, no. 4, pp. 430–459, 2016.
- [37] B. Eisenhower, T. Maile, M. Fischer, and I. Mezic, “Decomposing building system data for model validation and analysis using the Koopman operator,” in *Proceedings of the National IBPSAUSA Conference, New York, USA*, 2010.
- [38] I. Abraham, and T. D. Murphey, “Active learning of dynamics for data-driven control using Koopman operators,” *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1071–1083, 2019.

-
- [39] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, “Data-driven control of soft robots using Koopman operator theory,” *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 948–961, 2020.
- [40] M. Korda, Y. Susuki, and I. Mezić, “Power grid transient stabilization using Koopman model predictive control,” *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 297–302, 2018.
- [41] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition,” *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.
- [42] M. Korda, and I. Mezić, “Optimal construction of Koopman eigenfunctions for prediction and control,” *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5114–5129, 2020.
- [43] E. Yeung, S. Kundu, and N. Hodas, “Learning deep neural network representations for Koopman operators of nonlinear dynamical systems,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4832–4839.
- [44] M. Netto, Y. Susuki, V. Krishnan, and Y. Zhang, “On analytical construction of observable functions in extended dynamic mode decomposition for nonlinear estimation and prediction,” *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 1868–1873, 2021.
- [45] F. E. Sotiropoulos, and H. H. Asada, “Causality in dual faceted linearization of nonlinear dynamical systems,” in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 1230–1237.
- [46] H. Harry Asada, and F. E. Sotiropoulos, “Dual faceted linearization of nonlinear dynamical systems based on physical modeling theory,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 2, p. 021002, 2019.
- [47] F. E. Sotiropoulos, and H. H. Asada, “Dynamic modeling of bucket-soil interactions using Koopman-DFL lifting linearization for model predictive contouring control of autonomous excavators,” *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 151–158, 2021.
- [48] Y. Igarashi, M. Yamakita, J. Ng, and H. H. Asada, “A robust method for dual faceted linearization,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6095–6100, 2020, 21st IFAC World Congress.
- [49] T. Carleman, “Application de la théorie des équations intégrales linéaires aux systèmes d’équations différentielles non linéaires,” *Acta Mathematica*, vol. 59, pp. 63–87, 1932.

-
- [50] A. Amini, Q. Sun, and N. Motee, “Error bounds for Carleman linearization of general nonlinear systems,” in *2021 Proceedings of the Conference on Control and its Applications*. SIAM, 2021, pp. 1–8.
- [51] Y. Igarashi, M. Yamakita, J. Ng, and H. H. Asada, “MPC performances for nonlinear systems using several linearization models,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 2426–2431.
- [52] T. Chen, J. Shan, and H. Wen, “Koopman-operator-based attitude dynamics and control on $\text{so}(3)$,” in *Distributed Attitude Consensus of Multiple Flexible Spacecraft*. Springer, 2023, pp. 177–210.
- [53] D. Goswami, and D. A. Paley, “Global bilinearization and controllability of control-affine nonlinear systems: A Koopman spectral approach,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 6107–6112.
- [54] C. Folkestad, and J. W. Burdick, “Koopman NMPC: Koopman-based learning and nonlinear model predictive control of control-affine systems,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7350–7356.
- [55] C. Folkestad, S. X. Wei, and J. W. Burdick, “KoopNet: Joint learning of Koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1344–1350.
- [56] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.
- [57] A. Altman, and J. Gondzio, “Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization,” *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 275–302, 1999.
- [58] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Generalizing Koopman theory to allow for inputs and control,” *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.
- [59] M. O. Williams, M. S. Hemati, S. T. Dawson, I. G. Kevrekidis, and C. W. Rowley, “Extending data-driven Koopman analysis to actuated systems,” *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 704–709, 2016.
- [60] E. Kaiser, J. N. Kutz, and S. L. Brunton, “Sparse identification of nonlinear dynamics for model predictive control in the low-data limit,” *Proceedings of the Royal Society A*, vol. 474, no. 2219, p. 20180335, 2018.

-
- [61] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019. [Online]. Available: <https://doi.org/10.1007/s12532-018-0139-4>
- [62] L. Ljung, “Estimating linear time-invariant models of nonlinear time-varying systems,” *European Journal of Control*, vol. 7, no. 2-3, pp. 203–219, 2001.
- [63] S. Thangavel, S. Subramanian, S. Lucia, and S. Engell, “Handling structural plant-model mismatch using a model-error model in the multi-stage NMPC framework,” *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 1074–1079, 2018.
- [64] P. Foehn, and D. Scaramuzza, “Onboard state dependent LQR for agile quadrotors,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6566–6572.
- [65] Y. Sakai, M. Kanai, and M. Yamakita, “Torque demand control by nonlinear mpc for speed control of vehicles with variable valve lift engine,” *IFAC Proceedings Volumes*, vol. 43, no. 7, pp. 494–499, 2010.
- [66] H. Oyama, M. Yamakita, and H. H. Asada, “Approximated stochastic model predictive control using statistical linearization of nonlinear dynamical system in latent space,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 7153–7158.
- [67] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, “Online dynamic mode decomposition for time-varying systems,” *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 3, pp. 1586–1609, 2019.
- [68] D. Guého, P. Singla, and M. Majji, “Time-varying koopman operator theory for nonlinear systems prediction,” in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2294–2299.
- [69] J. Ng, and H. H. Asada, “Model predictive control and transfer learning of hybrid systems using lifting linearization applied to cable suspension systems,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 682–689, 2021.

研究業績

研究業績

学術雑誌論文

- Masaki Kanai, and Masaki Yamakita: “Linear model predictive control with lifted bilinear models by Koopman-based approach”, SICE Journal of Control, Measurement, and System Integration, 15(2), 162-171, 2022. (4章に関連)

国際会議発表

- Masaki Kanai and Masaki Yamakita: “Linear Model Predictive Control with Accurate Prediction Based on Low-Dimensional Lifting Linearization”, 6th IEEE Conference on Control Technology and Applications (CCTA), 2022. (3章に関連)
- Masaki Kanai and Masaki Yamakita: “Model Predictive Control with Model Error Compensation by Koopman Approach”, 48th Annual Conference of the IEEE Industrial Electronics Society (IECON), 2022. (5章に関連)
- Masaki Kanai and Masaki Yamakita: “Lifted Bilinear Model-based Linear Model Predictive Control with Scalability”, 22nd IFAC World Congress, 2023. (査読中, 6章に関連)

博士論文に関係しないもの

学術雑誌論文

- 金井 政樹, 齋藤 真司, 平田 敦: “イオン液体添加高分子複合材料のトライボロジー特性および自己潤滑コーティングへの適用”, 精密工学会誌, 76(7), 804-808, 2010.

国際会議発表

- Yasutaka Sakai, Masaki Kanai and Masaki Yamakita, “Torque demand control by non-linear MPC with constraints for vehicles with variable valve lift engine”, 2010 IEEE International Conference on Control Applications (CCA), 2010.

- Masaki Kanai and Masaki Yamakita: “Linear parameter varying error system modeling for nonlinear systems and its applications”, 2011 IEEE International Conference on Control Applications (CCA), 2011.
- Masaki Kanai and Masaki Yamakita: “On a Closed Loop Stability of Controlled Non-linear System Based on Lifting Linearization”, 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2021. (Position paper)
- Masaki Kanai, Shinji Ishihara, Ryu Narikawa and Toshiyuki Ohtsuka: “Cooperative Motion Generation Using Nonlinear Model Predictive Control for Heterogeneous Agents in Warehouse”, 6th IEEE Conference on Control Technology and Applications (CCTA), 2022.
- Shinji Ishihara, Masaki Kanai, Ryu Narikawa and Toshiyuki Ohtsuka: “A Proposal of Path Planning for Robots in Warehouses by Model Predictive Control without Using Global Paths”, 2022 Modeling, Estimation and Control Conference (MECC), 2022.
- Shinji Ishihara, Masaki Kanai, Ryu Narikawa and Toshiyuki Ohtsuka: “Comparative Study on Collision Avoidance Methods in Path Planning for Warehouse Robots Using MPC”, 48th Annual Conference of the IEEE Industrial Electronics Society (IECON), 2022.
- Shinji Ishihara, Masaki Kanai, Ryu Narikawa and Toshiyuki Ohtsuka: “Path Planning and Tracking for Heterogeneous Multiple Robots in Warehouses by Model Predictive Control Handling Trilemma-related KPIs”, 22nd IFAC World Congress, 2023. (査読中)

国際会議ポスター発表

- Masaki Kanai and Masaki Yamakita: “LPV Error System Modeling for Nonlinear Systems”, 18th IFAC World Congress, 2011.

査読なし国内会議発表

- 金井 政樹, 平田 敦: “イオン液体添加エポキシ樹脂による自己潤滑性コーティング”, トライボロジー会議 2009 春東京, 2009.
- 金井 政樹, 平田 敦: “イオン液体を添加した高分子材料の摺動特性”, 2009 年度精密工学会春季大会学術講演会, 2009.
- 金井 政樹, 山北 昌毅: “非線形システムに対する LPV 誤差システムの制御手法への応用”, 第 11 回計測自動制御学会制御部門大会, 2011.

- 金井 政樹, 山北 昌毅: “非線形システムに対する LPV 誤差システムモデリング”, 第 2 回 SICE プラントモデリングシンポジウム, 2011.

特許（主筆，国内登録のみ掲載）

- 金井 政樹, 荒井 雅嗣, 一野瀬 亮子, 小野 幸彦: “自律移動装置および自律移動システム”, 特許 5920743 号, 2016.4.22 登録.
- 金井 政樹, 濱田 朋之: “車両管制システム”, 特許 6219731 号, 2017.10.6 登録.
- 金井 政樹, 濱田 朋之, 加藤 学, 中 拓久哉, 荒井 雅嗣: “管制制御装置”, 特許 6247622 号, 2017.11.24 登録.
- 金井 政樹, 濱田 朋之: “車両管理システム”, 特許 6267059 号, 2018.1.5 登録.
- 金井 政樹, 荒井 雅嗣, 加藤 学, 濱田 朋之: “管制制御装置及び運搬車両の走行シミュレーション方法”, 特許 6426941 号, 2018.11.2 登録.
- 金井 政樹, 加藤 学, 濱田 朋之: “管制制御システム及び車載端末装置”, 特許 6498542 号, 2019.3.22 登録.
- 金井 政樹, 田中 航, 濱田 朋之: “管制制御装置及び車載通信端末装置”, 特許 6564954 号, 2019.8.21 登録.
- 金井 政樹, 濱田 朋之, 板東 幹雄, 森實 裕人: “自律走行システム”, 特許 7012769 号, 2022.1.20 登録.
- 金井 政樹, 濱田 朋之: “車両管理システム”, 特許 7064036 号, 2022.4.25 登録.
- 金井 政樹, 濱田 朋之: “車両管制システム”, 特許 7107737 号, 2022.7.19 登録.
- 金井 政樹, 濱田 朋之, 齋藤 真二郎: “作業エリア管理システム”, 特許 7129891 号, 2022.8.25 登録.

謝辞

本研究を進めるにあたり、様々な方々からご助言やご指導を賜りました。特に、指導教員である山北准教授には、社会人学生としての博士課程のご指導を快くお引き受けいただき、日々の研究推進をはじめとし、論文執筆や学会発表など非常に多くのご指導を賜りました。仕事との両立にも大変ご配慮いただき、中々成果が出ない時期にも辛抱強く、柔軟にご指導いただきましたことを深く感謝申し上げます。また、私の入学直前にC O V I D-19が流行し始め、ほとんど全てリモートでご指導いただくことになってしまいましたが、コミュニケーションが限られる中でも的確に導いていただきましたことを、重ねて感謝申し上げます。3年間のご指導のおかげで、あらためて制御工学に関して非常に多くのことを学び、またそれを研究成果としてアウトプットすることができ、研究者として大きく成長できたと実感しております。

山北研の学生のみなさまには、ともに研究室生活を過ごす中でたくさんの刺激を与えていただき、感謝申し上げます。リモートでの報告会や院ゼミで集まることがほとんどでしたが、その中でもそれぞれの研究内容を共有してもらうことで知見が広がるとともに、熱心に研究に取り組んでいる様子を見て、自身のモチベーションの向上に繋がりました。また、社会人学生である私にも研究室メンバの一員として接していただき、居心地の良い研究室生活が送れたことをとてもありがたく思っております。

私の博士課程への進学の後押し、ならびに通学中のご支援をいただいた株式会社日立製作所 研究開発グループ 自動運転研究部の部長である高橋氏、ユニットリーダである田中氏に感謝申し上げます。また、同部署の先輩であり、山北研の社会人博士としても先輩である石原氏には、進学の準備段階から在学中の研究推進に至るまで様々なご支援をいただき、深く感謝申し上げます。さらに、博士課程への進学と通学に寛大なご理解を示していただいた日立建機株式会社の皆様にも感謝申し上げます。

最後に、博士課程への進学に理解を示し、在学中も常に精神的に支えてくれた妻、そして家族に、心より感謝いたします。