Summer 2021

Robust Adaptive Model Predictive Control of Nonlinear Sample-Data Systems

Lixing Yang

Robust Adaptive Model Predictive Control of Nonlinear Sample-Data Systems

by

Lixing Yang

Bachelor of Science
Hebei University of Technology 2013

Bachelor of Science
Florida International University 2013

Master of Science
New Jersey Institute of Technology 2015

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2021

Accepted by:

Xiaofeng Wang, Major Professor

Bin Zhang, Committee Member

Enrico Santi, Committee Member

Song Wang, Committee Member

Tracey L. Weldon, Interim Vice Provost and Dean of the Graduate School

Abstract

In the past decades, model predictive control (MPC) has been widely used as an efficient tool in areas such as process control, power grids, transportation systems, and manufacturing. It provides an approach that aims to design stabilizing feedback to the system so that the performance criterion gets minimized while the state and input constraints get satisfied. In many situations, MPC may outperform other approaches to design and implement feedback control systems. Furthermore, MPC may solve optimization problems with large and practically important sets of multiple-input multiple-output (MIMO) systems efficiently. A typical implementation of MPC predicts the optimal control inputs that guarantee a certain level of optimality based on the interest of model behavior to the actual dynamical system. Many schemes of model predictive control have been addressed in the past years.

Recently, the technology development of computers, sensors, and communications make the control systems much larger and more complex than ever before. These advances also increase the need for MPC to design the controllers for complex multiple-input multiple-output systems. Besides, the advanced computation hardware has significantly improved the speed and reliability of solving optimization problems.

In general, we can differentiate the MPC scheme into linear and nonlinear model predictive control. Linear MPC refers to the MPC schemes that deal with linear models to predict the system dynamics. Besides, the constraints on the states and inputs should be linear, and the cost function can be as simple as quadratic. The optimal solutions of linear MPC rely on the dynamic models, the constraints,

and the optimal problems that aim to minimize the system performance, which is usually expressed as the cost function. Nonlinear MPC refers to the MPC schemes based on nonlinear models or non-quadratic cost functionals with corresponding nonlinear constraints on the states and inputs. Nevertheless, linear models are often inadequate in describing the optimal problem because of higher product quality specifications, increasing productivity demands, tighter environmental regulations, and the requirements of operating conditions. In this case, people need to use nonlinear MPC to describe the models accurately.

This dissertation studies several MPC algorithms for solving nonlinear continuous-time systems with uncertainties. The work focuses on systems with disturbances, system discretization, and explicit model predictive control. Meanwhile, we ensure these algorithms may provide a series of feasible solutions and stabilize the control systems along the prediction horizon.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter is organized as follows: introduces the background of model predictive control in Section 1.1, presents the basic knowledge of MPC in Section 1.2, and follows with prior work on MPC and dissertation organization in Section 1.3.

1.1 Background

Human activities implement the guidance of the basic concept of model predictive control (MPC) in many areas. It provides the idea that people may obtain the maximum benefit with minimum spend using a present decision according to a prediction of future based on the current situation. For example, a chess player tries to predict all the possible moves as far as possible to determine the actions that will achieve minor losses. Meanwhile, the next move will be determined according to the new situation based on the reaction of the opponent. Once the opponent has made a move, the player will only need to add one more step to the previous prediction. The schemes get repeated until the end of the game. Moreover, the prediction horizon is determined by the length of the match.

Three features need to be discussed in the scheme. To begin with, the prediction steps of each iteration are limited by the calculation ability of the player as it may have kinds of choices for each move, which rapidly increases the cost and complexity with the number of predicted moves. Secondly, it is crucial to decide a move among a variety of choices. In general, a decision aims to get the best performance based

on evaluating possible actions. A good decision balances the cost of planning the strategy to win and obtaining benefit from the opponent. Typically, it is difficult to set a fixed criterion for the judgment as it is better to seize the advantage with a weak opponent while it is wiser to achieve the strategy with a strong opponent. Finally, a crucial point to win the match depends on the length of prediction moves since a sequence of moves based on short-term predictions may lead to a bad result. In this case, the side with longer-term prediction moves may have the ability to eliminate the sequences of wrong actions and win.

The above aspects explain the critical points of model predictive control. We may derive the match as an MPC problem by describing the model, condition, and objective in mathematics, and we can get the prediction results by solving an optimization problem. Similarly, as chess matches, the criterion is usually determined by minimizing a cost function among the possible choices. People use the first element of the sequence as the control input to the system, and the prediction process is repeated with the new states. Besides, it is important that the control signal may guarantee the robustness of the system so that the system may reach its destination as the chess match.

In the 1960s, the research on control theory focused on three areas:

1. Maximum principle – producing necessary conditions for open-loop optimal control problems;
2. Dynamic programming – solving optimal feedback control problems;
3. Lyapunov stability – giving sufficient conditions to ensure the stability of control systems.

During this period, the linear quadratic regulator (LQR) was derived as one of the most critical methods for solving optimal control problems. LQR provides the approach for solving the optimization problem to linear time-invariant MIMO systems with linear time-invariant state feedback controls. The control performance is always

a quadratic function, which presents the system energy in most cases. Meanwhile, the robustness is guaranteed by driving the system to be asymptotically stable under common assumptions. Most importantly, the optimal solution is explicitly provided as a feedback gain formula which makes it easy to compute.

Nevertheless, few applications of LQR have been used in actual engineering models during the following decades. The main reason for this is the weakness of LQR considering the actuator saturation, which is called the constraint condition in the optimization problem. In contrast, the former control method that has been widely used for years – PID control – owns the ability to suit the actuator saturation in practice. To overcome the weakness of LQR, the Pontryagin minimum principle is proposed as a theory of optimal control that includes the effects of saturation.However, three flaws prevent the approach from being used practically: open-loop control law, difficulty computing the solution, and finite prediction horizon.

Nowadays, most existing control systems are operated in continuous time. Two approaches may solve the continuous-time optimization problem. One method is to solve the problem continuously and discretize the problem within a tiny period. The other method is to discretize the problem first and solve the discrete-time problem via the technique such as linear, quadratic, convex, and nonlinear programming. As an open-loop control method, its optimal solution mainly depends on the initial states, which is a significant disadvantage to system performance compared with feedback controls. In this case, it is a good idea to insert a closed-loop controller into the problem.

People find the open-loop controller can be estimated by a closed-loop controller via recomputing the optimal problem at each discretized point with a new initial state obtained in the previous optimal problem. Once the optimization problem is feasible in each process and the system may achieve stability, the model predictive control theory gets accomplished with the last method computation.

In general, the action of the MPC algorithm is obtained by solving a finite horizon open-loop optimal control problem according to a feedback control law in each sampling interval. We will use the first movement of the optimization from the sequence of optimal solutions as the input of the plant. Meanwhile, the iterations get repeated at shifted time-horizon using the most recently available state information as a new initial condition to avoid accumulating the errors along the time-horizon. In this case, we may simplify the computation by making the problem piecewise continuous.

1.2 Basic of Model Predictive Control

In the 1970s, linear MPC is invented and has become one of the critical advanced control strategies since then. It is widely used in the area of industrial applications. It may optimize the control performance based on the dynamic model while the constraints get held. We can find the overviews and applications of linear MPC techniques in [1] and [2]. So far, the theory of linear MPC grows maturely, and it has been well studied in aspects such as computation method, system modeling, and control strategy. We are going to describe the process of MPC theoretically in continuous time as a feedback control method.

1.2.1 The Principle of Predictive Control

The model predictive control scheme can be treated as a repeated process on solving finite horizon open-loop optimal control problem (FHOCP) subjects to system dynamic, states constraint, and input constraint. As shown in Figure 1.1 of [3], the controller predicts the dynamic behavior over the prediction horizon T_p by determining the input in order to minimize the system performance based on the sampled initial conditions at t. Ideally, if there do not exist model disturbance, model-plant mismatch, and computation difficulty. In that case, we can solve the

optimization problem along prediction horizon until any time as T_p can be determined freely, and the problem can be solved at one time. However, it is always difficult to find the systems that satisfy the three conditions above in most of time. In this case, we need to divide the model and solve the problem recursively, which means the process in Figure 1.1 gets repeated and moves forward along the time horizon.

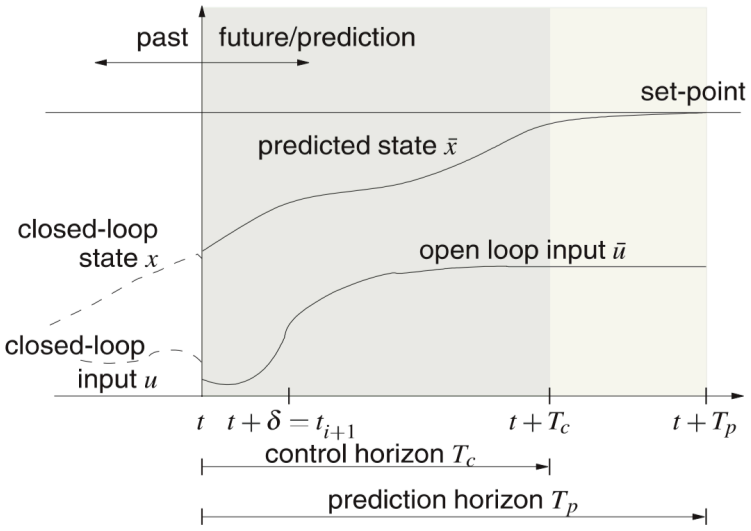

Figure 1.1: Principle of MPC

In general, we usually make the optimal solutions be piecewise-continuous for simplicity. And the optimal inputs remain the same value as it is calculated at t until the next sampling time t_{i+1}. Besides, the length of the sampling period δ can be determined according to the demands of accuracy.

Then we can summarize the MPC scheme process as:

1 Sampling states of the system;

2 Calculating FHOCP that minimizing the system performance with constraints over the prediction horizon based on dynamic model and sampled state;

3 Implement the first step optimal input until the next sampling iteration.

1.2.2 Mathematical Formulation of State Feedback NMPC

Consider a continuous-time system:

$$\dot{x}(t) = f(x(t), u(t)),\ x(t_k) = x_k$$

where the system state $x \in \mathcal{X} \subset \mathbb{R}^n$, the control input $u \in \mathcal{U} \subset \mathbb{R}^m$, and $f : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$. Meanwhile, we assume $f(x, u)$ as local Lipschitz, i.e., there exists a positive constant L_f so that $\forall x, y \in \mathcal{X}, u \in \mathcal{U}$, the following inequality holds:

$$\|f(x, u) - f(y, u) \leq L_f \|x - y\|.$$

Typically, the constraints may convex with the form as

$$\begin{aligned}\mathcal{U} =& \{u \in \mathbb{R}^m | u_{min} \leq u_i \leq u_{max}\} \\ \mathcal{X} =& \{x \in \mathbb{R}^n | x_{min} \leq x_i \leq x_{max}\}\end{aligned}$$

where $u_{min}, u_{max}, x_{min}, x_{max}$ are scalars and u_i, x_i are elements of input and state vector.

We solve the FHOCP problem with the prediction horizon $[t_k, t_k + T_p)$ at t_k based on the sampled state $x(t_k)$. Besides, we set the optimal input to be piecewise-continuous with the length of $\delta (\delta \ll T_p)$, which means the next prediction routine starts at $t_k + \delta$. Then we may get the following optimal control problem

$$\begin{aligned}\textbf{Problem 1} \qquad & \min_{u \in \mathcal{U}} J[u|x(t_k)] \\ \text{s.t. } & \dot{x}(t) = f(x(t), u(t)),\ x(t_k) = x_k,\ t \in [t_k, t_k + T_p) \\ & u(t) \in \mathcal{U},\ x(t) \in \mathcal{X},\ x(t_k + T_p) \in \mathcal{X}_f \\ & J[u|x(t_k)] = \int_{t_k}^{t_k + T_p} L(x(t), u(t)) dt + V_f(x(t_k + T_p))\end{aligned}$$

where x_k is sampled from the plant at t_k, $\mathcal{X}_f$ is the terminal set that the final state stays inside. Meanwhile, we define $J[u|x(t_k)]$ as system performance, $L(x(t), u(t))$ as

stage cost, and $V_f(x(t_k + T_p))$ as terminal cost. In general, we may present

$$L(x(t), u(t)) = x(t)^T Q x(t) + u(t)^T R x(t)$$
$$V_f(x(t_k + T_p)) = x(t_k + T_p)^T H x(t_k + T_p)$$

in quadratic form which is same as Lyapunov equations, and Q, R, H are semi-positive symmetric matrices. With the sampled state x_k, we can solve Problem 1 subject to the dynamic function and the constraints on the state and input. Besides, the solution of the problem is used as the prediction to the control input. We define the optimal solution as $u^*(t)$ and it applies to the plant until $t_k + \delta$ that can be presented as

$$u(t) = u^*(t_k), \ t \in [t_k, t_k + \delta).$$

For the discussion on feasibility and stability of MPC algorithm, we will provide them with corresponding proof in the following chapters in detail.

1.3 Problem Description of Model Predictive Control

1.3.1 Systems with Disturbances

It is common to meet the situation that there exist disturbances or uncertainties in dynamic models, such as the sampling errors to actual systems, the external interferences to operate plants, and the deviations caused by transmission delays. In this case, it is necessary to discuss the problem of the systems with disturbances under the MPC scheme.

To begin with, there comes the study showing that the standard MPC algorithm may admit a certain level of robustness concerning minor uncertainties. As it discusses the conditions that may derive the stabilization for discrete-time nonlinear systems with uncertainties under model predictive control schemes in [4]. The authors prove that exponential stability theorems may still achieve asymptotic stability with

compact uncertainties inserted. Furthermore, the authors analyze the stability of constrained nonlinear discrete-time systems with bounded additive uncertainty in [5]. The authors provide sufficient conditions that may guarantee the framework's robust stability and feasibility using invariance set theory. In the paper of [6], the authors present some robust stability results for nonlinear constrained discrete-time systems under model predictive control schemes without any particular properties of the terminal cost on deriving such approaches. The authors introduce two assumptions: the value function can be upper bound by a $\mathcal{K}_\infty$ function of the state, and the measurement should be detectable in the MPC algorithm. And these assumptions have been treated as common conditions for solving MPC problems nowadays. Then the authors show the system under previous assumptions may admit stability with small disturbances.

Another approach to overcome the impact of the disturbances is to formulate MPC as min-max optimization problems, which is to optimize the control performance under the worst-case uncertainty. Comparing with standard MPC schemes, it may improve the control performance since it always considers the impact of the disturbances. In the paper of [7, 8], the authors first provide a notion of MPC algorithm under min-max optimization formulation. And the system feedback is also presented in the control implementation. The authors show that it does not need to consider all possible disturbance realizations, and the optimization is always processed under the extreme disturbance realizations. Nevertheless, the scheme may have high computation costs on solving the optimization problems with large horizons. In this case, the method may have a compelling performance in solving the system within small horizons. Meanwhile, the authors also consider the possible feasibility problems such that one fixed control profile may ensure the formulation of min-max MPC along the prediction horizon. It presents a min-max MPC scheme for discrete-time nonlinear systems in [9, 10]. The authors propose that the approach may be practical under

classical min-max MPC problem set-up, and the input-to-state stability is ensured. To begin with, the authors show that input-to-state practical stability can be guaranteed under closed-loop min-max MPC systems, and the corresponding explicit bounds on the evolution of the state have been provided. To obtain a controller that may derive the state around the desired point, the authors offer a priori sufficient conditions for robust stability under the input-to-state stability (ISS) framework. The authors prove that such conditions imply the overall system uniformly ultimately bounded. Finally, the authors formulate sufficient conditions for ISS of min-max nonlinear MPC using a dual-mode approach. In this case, the system may get robustly asymptotically stable without assuming that the disturbance needs to converge to zero as the state converges to the origin.

Furthermore, people introduce an adaptive model predictive control method, allowing the model to estimate and predict the disturbances over the prediction horizon. It should be a promising way to improve the control performance since the worst-case uncertainties are not affine to the overall system, comparing with the min-max MPC scheme. In the paper of [11], the authors consider the problem of adaptive receding horizon model predictive control, which is the earliest paper focusing on nonlinear systems. It presents an approach to estimate the uncertainties, and the estimation procedure is employed in a receding horizon controller. In this case, it may achieve an accurate prediction while there exists deviation between the estimated parameter and actual parameter. Furthermore, the overall system may admit stability with the allowance for the deviation. The authors demonstrated the problem focuses on identifying the regions where the uncertainties should stay inside accordingly in [12], which is a different approach for adaptive model predictive control as previous research. By generating a set-valued measurement of uncertainties, it may minimize the impact of parameter identification error. Meanwhile, it may reduce the conservativeness of the computation by implementing such estimation

to the optimization procedure. Furthermore, the feasibly and stability of overall system get guaranteed with constraints to state or input within the framework of robust nonlinear MPC. It provides an adaptive MPC design technique for nonlinear constrained systems with parametric uncertainties in [13, 14]. The authors develop an approach that may generate the unknown parameter via a parameter estimator. Specifically, the adaptive identifier formulated based on the Lipschitz projection operator. Furthermore, a general min-max approach is considered combining with the adaptive estimator. And the scheme of robust adaptive MPC is proposed by solving the optimization problem of parameterized uncertain nonlinear systems subject to state and input constraints. Similar to the previous work, it may reduce the conservativeness while the robustness is still ensured under the adaptive MPC scheme.

The work in [4–6] shows that the standard MPC algorithm admits a certain level of robustness concerning minor uncertainties. Nevertheless, such marginal robustness may not be enough in practice. An alternative method is to formulate MPC as min-max optimization problems, which optimize the control performance under the worst-case uncertainty (see [7–10] and reference therein). However, such min-max approaches amplify the negative impact of uncertainties, either address the robustness issue or degrade the control performance. It only approximates the uncertainties at each sampling point in [11], while the errors may keep changing inside each prediction interval. In the paper [12], the authors derive an adaptive region for the uncertainties. However, it may generate unnecessary computation costs without an accurate estimation of the uncertainties. The most related work is [13, 14], where estimators are embedded in the min-max MPC framework. Nevertheless, the projection operator cannot be pre-calculated, which may impact the stability of the system.

1.3.2 System Discretization

In computer-controlled systems, the MPC schemes are often implemented in discrete time for digital implementation. In this case, the sampling and computation of optimal control inputs should be triggered in a discrete manner. And the models used in the finite-horizon optimal control problem (FHOCP) should also be discrete.

As a common approach, the system is always discretized under fixed time manners. The discretized models get expressed as piecewise constant models along the prediction horizon. Meanwhile, the sampling periods follow the same periodic time manner. Then the optimization problems are generated using such discrete-time models with corresponding constraints. Typically, the authors derive algorithms of continuous-time systems under model predictive control schemes in [15–19]. The robustness of each algorithm gets guaranteed under appropriate stabilizing properties. The approaches in the four articles provide a standard method of model discretization. In the work of [20, 21], the authors propose the schemes of sampled-data model predictive control for continuous-time linear parameter varying (LPV) systems. The stability properties of the proposed MPC are discussed by modeling the closed-loop systems with piecewise constant input. Meanwhile, the sampling interval is not necessarily to be periodic with such linear systems. In this case, the approaches are efficient to reduce computation conservativeness and achieve better performance.

The main idea of the event-triggered strategy is that the optimal control problem is solved only under a specific event condition rather than making the computation every step as classic MPC schemes. The authors propose a robust event-triggered model predictive control scheme for linear time-invariant discrete-time systems with bounded stochastic disturbances in [22]. The probability distributions of the disturbances have been pre-determined. Meanwhile, the authors set the event condition for the situation the state runs out of a given region so that the control

values get updated periodically. The authors employ the event-triggered profile under Tube MPC methods. And the robustness of the overall system gets guaranteed with the constraints on the input and state. The same strategy is considered in [23] to design an MPC scheme for continuous-time event-based systems. The authors introduced a state estimator with a bounded covariance matrix. The error between estimated state and sampled state then gets implemented into the MPC formulation. Meanwhile, the authors prove that the overall closed-loop system is input-to-state stable (ISS) to the estimation error.

In the papers of [24, 25], the authors consider an event-triggered strategy of model predictive control schemes for discrete-time systems. The control law gets updated with the event condition that a $\mathcal{K}_\infty$ function of measurement error reaches the critical region constructed by the system state. The authors show that the event-triggered rule may generate effectively under either linear or nonlinear plants. Furthermore, the event-triggered profile gets extended to a self-triggered formulation. The triggered condition is determined by the previous measurement error to avoid the need of monitoring the plant continuously. Further research [26] proposes a novel event-triggered model predictive control scheme for nonlinear continuous-time systems with uncertainties. In this work, the triggering condition of continuous-time systems follows the same idea as previous papers. And the robustness analysis gets provided to ensure the overall system is uniformly ultimately bounded under the proposed framework.

The authors present an event-triggered control scheme for linear continuous-time systems based on a combination of model predictive control (MPC) and integral sliding lode (ISM) control in [27]. Similarly, the objective of such a control framework may minimize the number of transmissions and save computation cost, while the robustness gets ensured under the constraints. The authors introduce the sliding mode approach to compensate the uncertainties of the system, and the MPC

framework may optimize the control performance with corresponding constraints. The triggering condition is determined as the controller gets updated when the error between the measured state and computed state reaches the limitation. The paper presents an event-triggered implementation of MPC for linear discrete-time systems with disturbances under the same triggering condition in [28]. The stability conditions are analyzed by the comparison of event-triggered implementation and common time-triggered scheme. An explicitly computable set is provided based on the disturbance bound and the event threshold accordingly. The exact triggering condition is studied in the paper [29] under the event-triggered model predictive control (MPC) scheme for nonlinear continuous-time systems subject to bounded disturbances. The authors investigate sufficient conditions for ensuring feasibility and stability. Furthermore, it is shown that the length of the prediction horizon, the range of disturbance bound, and the level triggering condition may directly impact the system stability.

Self-triggered model predictive control proposes an approach that may adjust the utilization of sampling procedure along the prediction horizon other than the conventional periodic sampling. Self-triggered MPC follows the strategy that each sampling interval is determined online based on the current state of the system to achieve a low average sampling rate. In this case, it may minimize the control communication and reduce the number of control updates. The authors introduce a self-triggered MPC strategy for discrete-time linear systems in [30]. The triggered interval is determined by the length of the controlling time under the current input strategy. There are three features been studied under the control formulation. To begin with, it guarantees the asymptotic stability of the system with corresponding constraints. Moreover, the feedback law and the triggering condition get well designed. Finally, the overall performance shows a significant reduction in the usage of sampling resources. The authors propose a self-triggered MPC strategy

for discrete-time nonlinear systems in [31] as an extension to the preliminary paper. As an improvement of the previous work, the self-triggered MPC framework owns the flexibility to determine an appropriate input strategy within limited options rather than employing the same strategy through the whole iteration. A similar self-triggered MPC scheme is introduced for linear systems with additive bounded disturbances in [32]. The self-triggered formulation is derived under Tube MPC. And the authors proposed a constraint tightening method to ensure the robustness of the overall system.

In the paper of [33], the authors propose a self-triggered model predictive control scheme for a robotic model, which is a continuous-time nonlinear nonholonomic system with additive disturbances. The triggering mechanism is considered based on the discussion of the stability, which may drive the state into a compact set and ensure the framework gets uniformly ultimately bounded. Furthermore, the simulation result illustrates the efficiency of the proposed approach. The authors present a self-triggered MPC approach for the linear time-invariant (LTI) process in [34]. It discussed the sufficient conditions to ensure the stability of the system. Meanwhile, it provided an explicit solution such that the optimization problem gets pre-solved offline, and the controller gets implemented from a lookup table of state feedback gains. Furthermore, the simulation examples show the efficiency performance of the scheme.

The sampled-data MPC has been discussed in [15–19] as periodic models. However, it is conservative for these approaches to estimate control periods sometimes. In this case, the control task may have greater utilization than actual needs. It may lead to a significant over-provisioning to the operating system. To overcome the problem, people present the method of aperiodic sampled-data MPC, which considers linear uncertain continuous-time systems in the work of [20, 21]. However, the models used in the FHOCP are in continuous time. Meanwhile,

event-triggered MPC is presented as another approach. The models are defined as discrete-time systems in the work of [22, 24, 25, 28]. It only considers the linear continuous-time system in the work of [23, 27], which the system can be discretized easily. In the paper [26], it derives the overall scheme in continuous time. The authors of [29] linearize the systems arbitrarily, which disregards the impact of linearization error on stability. Besides, self-triggered MPC also works on solving the problem. Different from event-triggered MPC, the sampling/computation instant expressed as a function relative to the past information. The works of [30–32, 34] only consider the strategies under discrete-time systems. And the authors of [33] generate the optimization problem in continuous time. In conclusion, all previous work is not friendly for digital implementation to practical models; either the models are defined as discrete-time systems or set to be linear systems.

1.3.3 Explicit Model Predictive Control

As an efficient approach, explicit MPC may partially solve the finite horizon optimal control problem (FHOCP) offline, which may reduce the online computation time on solving the FHOCP for the discrete-time model. Explicit model predictive control is a method that can pre-calculate the FHOCP problem as an explicit function of the states and reference vectors rather than make the whole computation in real-time, which is a crucial advancement compare with common MPC approaches [35, 36]. It can prevent the application of MPC in several contexts, such as the expense of computation, delay of control schemes, and feasibility of states. Typically, explicit MPC may solve the optimization problem using multi-parametric programming technique. The optimal control input can be expressed as the function contains sampling state, initial input, and reference conditions. In most cases, the optimal input is piecewise affine with concerning the states [18, 37, 38], and the MPC controller can be expressed in linear form.

The authors of [39] first introduce an algorithm to determine the feedback control law for discrete-time linear time-invariant systems, which may minimize the quadratic performance criterion with constraints on inputs and states. The online computation of the finite horizon optimal control problem gets simplified as explicitly piecewise linear functions to reduce the computational complexity. Then we may solve the optimization using the Hamilton-Jacobi-Bellman equation for discrete-time linear constrained systems under the controller structure. And the explicit solutions can be obtained by solving multi-parametric quadratic programs with the parameters based on of the state vector. The technique aims to avoid the procedure of online computation, while the stability and performance properties of model predictive control still get inherited. The authors show an efficient performance of the algorithm; however, the scheme is not intended to replace the online computation approach, especially for some industry systems with extensive applications. Furthermore, the authors extend the previous work by studying the geometry properties of the polyhedral partition in [40]. Then preliminary strategy gets improved that may exclude the conditions such as the unnecessary partitioning, the solution to the linear programming problems for determining the interior point of each critical region, and the solution to the quadratic programming problem.

Explicit model predictive control may derive the topics such as trajectory following, disturbances impaction, time-varying constraints, and criteria comparison. In the work of [41], the authors study the explicit model predictive control schemes for discrete-time linear time-invariant systems with constraints on inputs and states. The authors compare the control performances under the criteria of 1-norm and ∞-norm. Based on the discussion, the authors provide the conditions for the weighting matrices to ensure robustness. Another work on investigating the performance of explicit MPC algorithm is presented in [42]. It considers the explicit scheme for multiparametric nonlinear integer programming problems, such as the cost function and constraints in

nonlinear fashions. The author examines the main theoretical properties and proposes an adjusted solution under sufficient conditions.

The authors propose an approximate multi-parametric algorithm based on the explicit model predictive control scheme in [43]. The approach allows the implementation of bounded disturbance in real-time systems, while the trajectories may converge to the pre-defined fixed sets, and robustness of the systems get guaranteed. The authors analyze the impact of implemented errors and provide alternative approaches to ensure stability. Meanwhile, the corresponding explicit control law is provided. The author derives the formulation of explicit model predictive control for nonlinear discrete-time system under the scheme of explicit linear state feedback approximation in [44] as an extension to previous work. The author provides the strategy to linearize overall system, and the robustness gets ensured if approximation error gets bounded inside the stability margin.

The previous work of [39–43] consider deriving the scheme under linear discrete-time systems. And [44] provides an extension of the formulation to nonlinear discrete-time systems.

1.3.4 Contribution

In our work of Chapter 2, with relatively accurate estimation and prediction, one expects to narrow down the region where the uncertainties stay. Compare with preliminary strategies, we estimate the uncertainty using fast adaptation. Meanwhile, we can predict the set-valued measurement of uncertainty over the prediction horizon at each computation cycle based on the estimation. Then the adaptive regions of the uncertainties get structured. Finally, the conservativeness due to the worst-case synthesis can be significantly reduced.

We introduce a sporadic MPC algorithm for nonlinear continuous-time systems based on the Lebesgue approximation in Chapter 3. The sampling is triggered by a

self-triggering method, which may discretize the system directly. And the sampling and prediction step-length are determined in an aperiodic manner accordingly.

We investigate an explicit adaptive MPC(EAMPC) algorithm for nonlinear continuous-time systems in Chapter 4. We discretize the original system and develop an adaptive discrete-time estimator to approximate the discretization error. And we predict the error within a bounded region over each prediction iteration. In this case, the online computation is mainly for function evaluation based on multi-parametric programming, which may save computation costs dynamically. And the MPC controller maps the state into a lookup table of linear gains.

1.3.5 Dissertation Organization

We are going to present the approaches on modeling systems with disturbance adaptively, discretizing model using Lebesgue approximation, and solving FHOCP of nonlinear continuous-time systems explicitly in the following chapters. The dissertation is organized as follows: Chapter 2 introduces an MPC algorithm on solving nonlinear continuous systems with state-dependent uncertainties adaptively; Chapter 3 presents a discrete-time MPC algorithm for nonlinear sampled-data systems with uncertainties based on Lebesgue approximation; Chapter 4 describes an adaptive MPC scheme on solving the nonlinear continuous-time system explicitly, Chapter 5 provides the examples for the application of the algorithm in Chapter 4; Chapter 6 conclude current work along with a discussion on future research.

Chapter 2

Adaptive Model Predictive Control of Nonlinear Systems with State-Dependent Uncertainties

This chapter studies adaptive model predictive control (AMPC) with time-varying and potentially state-dependent uncertainties. We propose an estimation and prediction architecture within the min-max MPC framework. An adaptive estimator is presented to estimate the set-valued measures of the uncertainty using piecewise constant adaptive law, which can be arbitrarily accurate if the sampling period in adaptation is small enough. Based on such measures, a prediction scheme is provided that predicts the time-varying feasible set of the uncertainty over the prediction horizon. We show that if the uncertainty and its first derivatives are locally Lipschitz, the stability of the system with AMPC can always be guaranteed under the standard assumptions for traditional min-max MPC approaches, while the AMPC algorithm enhances the control performance by efficiently reducing the size of the feasible set of the uncertainty in min-max MPC setting.

2.1 Problem Formulation

Notations: We denote by $\mathbb{R}^n$ the n-dimensional real vector space, by $\mathbb{R}^+$ the set of the real positive numbers, and by $\mathbb{R}_0^+$ the set of the real non-negative numbers.

We use $\|\cdot\|$ to denote the Euclidean norm of a vector and the induced 2-norm of a matrix.

Definition 1. *A continuous function $\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. A function $\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ belongs to class $\mathcal{K}_\infty$ if it belongs to class $\mathcal{K}$ and $\lim_{r\to\infty} \alpha(r) = \infty$.*

Definition 2. *A set $\mathcal{S} \subseteq \mathbb{R}^n$ that contains the origin in its interior is called a robustly positive invariant (RPI) set for the system governed by $\dot{x} = g(x, w)$ (with the set Ω and the terminal time T) if for all $x_0 \in \mathcal{S}$, the state trajectory $x(t)$ with $x(0) = x_0$ satisfies $x(t) \in \mathcal{S}$ for any $t \in [0, T]$ and any $w(t) \in \Omega$.*

Definition 3. *The state $x(t)$ of a system $\dot{x} = f(x)$ is called uniformly ultimately bounded (UUB) with ultimate bound b if there exist positive constants b and c, independent of $t_0 \geq 0$, and for every $a \in (0, c)$, there is $T = T(a, b) \geq 0$, independent of t_0, such that $\|x(t_0)\| \leq a$ implies $\|x(t)\| \leq b$ for any $t \geq t_0 + T$.*

Consider a multi-input-multi-output (MIMO) state-feedback system:

$$\dot{x}(t) = f(x(t), u(t)) + \Delta(t, x(t)),\ x(0) = x_0 \tag{2.1}$$

where $x : \mathbb{R}_0^+ \to \mathcal{X}$ is the system state, $u : \mathbb{R}_0^+ \to \mathcal{U}$ is the control input, $f : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is a known function satisfying $f(0, 0) = 0$, and $\Delta : \mathbb{R}_0^+ \times \mathbb{R}^n \to \mathbb{R}^n$ is unknown. The compact sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ describe the state constraint and the input constraint, respectively. In other words,

$$x(t) \in \mathcal{X},\ u(t) \in \mathcal{U} \tag{2.2}$$

must hold for any $t \geq 0$. The function $f(x, u)$ satisfies the following condition over $x, y \in \mathcal{X}$ and $u, v \in \mathcal{U}$:

$$\|f(x, u) - f(y, v)\| \leq l_x\|x - y\| + l_u\|u - v\|, \quad \forall x \in \mathcal{X},\ u \in \mathcal{U} \tag{2.3}$$

where $l_x, l_u \in \mathbb{R}^+$. The uncertainty $\Delta(t,x)$ is assumed to satisfy the following condition:

Assumption 1. *Given a positive constant T, there exist constants $l_\Delta, l_t, b_\Delta, l_{\Delta_t}, b_{\Delta_t}, l_{\Delta_x}, b_{\Delta_x} \in \mathbb{R}^+$ such that for any $t, \tau \geq 0$ satisfying $|t-\tau| \leq T$ and any $x, y \in \mathcal{X}$,*

$$\|\Delta(t,x) - \Delta(\tau,y)\| \leq l_\Delta \|x-y\| + l_t|t-\tau| \tag{2.4}$$

$$\|\Delta(t,0)\| \leq b_\Delta \tag{2.5}$$

$$\left\|\frac{\partial \Delta(t,x)}{\partial t}\right\| \leq l_{\Delta_t}\|x\| + b_{\Delta_t}$$

$$\left\|\frac{\partial \Delta(t,x)}{\partial x}\right\| \leq l_{\Delta_x}\|x\| + b_{\Delta_x}$$

hold. Moreover the positive constants $l_\Delta, l_t, b_\Delta, l_{\Delta_t}, b_{\Delta_t}, l_{\Delta_x}$, and b_{Δ_x} are known.

Remark 1. *This assumption is to ensure the growth rate of the uncertainty is bounded so that it can be predicted over the prediction horizon. Such an assumption can often be found in adaptive control literature [45]. A weaker assumption is that the function $\Delta(t,x)$ and its first derivatives are uniformly bounded for any $t \geq 0$ and any $x \in \mathcal{X}$. Our approach can be easily extended under this weaker assumption. However, those uniform bounds are usually conservative because they should be valid for any $x \in \mathcal{X}$. This chapter focuses on the Lipschitz assumption which will result in less conservative and state-dependent upper bounds on $\Delta(t,x)$ and its first derivatives.*

Given the fact that x should stay in a compact set $\mathcal{X}$, we can define a constant θ and a set Ω as follows:

$$\theta = l_\Delta \max_{x \in \mathcal{X}} \|x\| + b_\Delta \tag{2.6}$$

$$\Omega = \{w \in \mathbb{R}^n \mid \|w\| \leq \theta\}. \tag{2.7}$$

We can see $\Delta(t, x(t)) \in \Omega$ for any $t \geq 0$, given inequalities (5.13) and (5.16) in Assumption 1.

The objective of this chapter is to stabilize the system described in equation (5.11) subject to the state and input constraints in (5.14).

2.2 Adaptive MPC

To stabilize the system subject to the state and input constraints, we consider the MPC approach. Let $\{t_k\}_{k=0}^{\infty}$ denote a time sequence with $t_{k+1} = t_k + \delta$ and $t_0 = 0$ where $\delta \in \mathbb{R}^+$ is the MPC iteration period. The related continuous-time min-max MPC algorithm is proposed as follows: At time t_k, the controller solves the constrained optimal control problem over the time interval $[t_k, t_k + T)$, where T is the length of the prediction horizon greater than δ:

$$\begin{aligned} \textbf{Problem 2}: \quad & \min_{\hat{u}(t)\in\mathcal{U}} \quad \max_{\hat{\Delta}(t)\in\mathcal{W}_k(t)} \quad J[\hat{u}, \hat{\Delta}] \\ & \text{s.t. } \dot{\hat{x}}(t) = f(\hat{x}(t), \hat{u}(t)) + \hat{\Delta}(t), \\ & \hat{x}(t) \in \mathcal{X} \\ & \hat{x}(t_k + T) \in \mathcal{X}_f. \end{aligned}$$

for any $\forall t \in [t_k, t_k + T)$, where

$$J[\hat{u}, \hat{\Delta}] \triangleq \int_{t_k}^{t_k+T} L(\hat{x}(t), \hat{u}(t))dt + V_f(\hat{x}(t_k + T))$$

$L : \mathcal{X} \times \mathcal{U}$ is the running cost, $V_f : \mathbb{R}^n \to \mathbb{R}_0^+$ is the terminal cost, $\hat{\Delta} : \mathbb{R}_0^+ \to \mathbb{R}^n$ describes the uncertainty in the model, $\mathcal{X}_f \subset \mathbb{R}^n$ is the terminal set, and $\mathcal{W}_k(t)$ is the predicted feasible set at the kth iteration that covers the uncertainty over $[t_k, t_k + T)$. Once this problem is solved at the kth iteration, then the optimal input, $\hat{u}_k^*(t)$, over $[t_k, t_k + \delta)$ will be applied to the plant, i.e.

$$u(t) = \hat{u}_k^*(t), \quad \forall t \in [t_k, t_k + \delta).$$

Meanwhile, the next iteration starts at t_{k+1}.

Remark 2. *Problem 2 implies an open-loop formulation of min-max MPC. We use this framework as a carrier to discuss how adaptation can be introduced in MPC. Notice that our method is also applicable to feedback MPC using closed-loop predictions. More details on applying AMPC in a feedback MPC framework can be found in [14].*

In this MPC framework, there are still several components to be determined: $\mathcal{W}_k(t)$ over the time interval $[t_k, t_k + T)$, V_f, and $\mathcal{X}_f$. In traditional min-max MPC approaches, the set $\mathcal{W}_k(t)$ is usually defined as $\mathcal{W}_k(t) \triangleq \Omega$, based on which V_f and $\mathcal{X}_f$ can be selected [8, 10]. However, if the set Ω is very large, the worst-case analysis will result in very conservative results, which lead to poor control performance or even infeasibility of MPC. In our approach, we actively estimate the uncertainty $w(t)$ and quantify the estimation error using an adaptive estimator. With the estimated information, it is expected to obtain a smaller set $\mathcal{W}_k(t)$ over the prediction horizon, compared with Ω, which will help reducing conservativeness. The detailed AMPC algorithm will be discussed in the following sections.

This section introduces the AMPC algorithm. It includes three steps as shown in Figure 2.1: (i) estimate the uncertainty; (ii) predict the set $\mathcal{W}_k(t)$ over $[t_k, t_k+T)$; and (iii) solve Problem 2 for $u_k^*(t)$. Notice that once the first two steps are completed, the third step can be solved using dynamic programming [46]. Therefore, the following discussion will mainly focus on uncertainty estimation and prediction and stability analysis.

2.2.1 Adaptive Estimation

This subsection introduces the algorithm to estimate the uncertainty $\Delta(t, x(t))$. We expect to identify the set where $\Delta(t, x(t))$ stays. To obtain a tight estimation, we

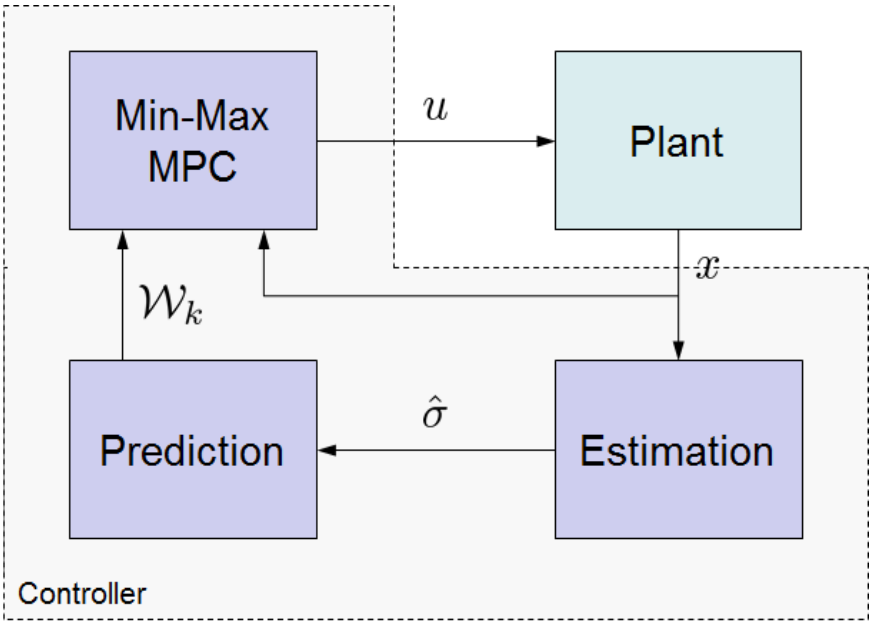

Figure 2.1: AMPC Procedure

take advantage of fast adaptation. The adaptive estimator is defined by

$$\dot{z}(t) = -a\tilde{x}(t) + f(x(t), u(t)) + \hat{\sigma}(t),\ z(0) = x_0 \tag{2.8}$$

where $z : \mathbb{R}_0^+ \to \mathbb{R}^n$ is the state of the estimator, $a \in \mathbb{R}^+$ is an arbitrarily chosen positive constant and $\tilde{x}(t) = z(t) - x(t)$ is the state estimation error. The signal, $\hat{\sigma}(t)$, is updated according to the piecewise constant adaptation law:

$$\hat{\sigma}(t) = -\Upsilon(T_s)\tilde{x}(kT_s) \tag{2.9}$$

for $t \in [kT_s, (k+1)T_s)$ and $k = 0, 1, 2, \cdots$, where $T_s \in \mathbb{R}^+$ ($T_s \ll \delta$) is the sampling period, and

$$\Upsilon(T_s) = \frac{a}{e^{aT_s} - 1}.$$

Remark 3. *The parameter a has to be positive to ensure convergence of $\tilde{x}(t)$. On one hand, the larger a is, the faster $\tilde{x}(t)$ converges. On the other hand, a should not be too large from the robustness perspective. Also notice that the selection of a should keep $\Upsilon(T_s)$ away from being close to 0 or ∞ in order to prevent potential numerical issues.*

The performance of this estimator is established in the following lemma:

Lemma 1. *Consider the system* (5.11) *and the estimator defined by equations* (4.9) *and* (2.9)*. If Assumption 1 is satisfied, then the following inequality*

$$\|\Delta(t, x(t)) - \hat{\sigma}(t)\| \leq \gamma(T_s) \tag{2.10}$$

holds for any $t \geq 0$*, where* $\gamma : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ *is defined by*

$$\gamma(T_s) = 2rT_s + \|1 - e^{-aT_s}\|\theta \tag{2.11}$$
$$r = l_{\Delta_t} \max_{x \in \mathcal{X}} \|x\| + b_{\Delta_t} + (l_{\Delta_x} \max_{x \in \mathcal{X}} \|x\| + b_{\Delta_x}) \left(\max_{x \in \mathcal{X}, u \in \mathcal{U}} \|f(x, u)\| + \theta \right)$$

and θ *is defined in* (2.6)*.*

Proof. Consider the system (5.11) and the estimator in (4.9). The error dynamics of $\tilde{x}(t)$ satisfies

$$\dot{\tilde{x}}(t) = -a\tilde{x}(t) + \hat{\sigma}(t) - \Delta(t, x(t)),\ \tilde{x}(0) = 0.$$

With the sampling period T_s, the dynamics can be discretized as

$$\tilde{x}(t) = e^{-a(t-kT_s)}\tilde{x}(kT_s) + \int_{kT_s}^{t} e^{-a(t-\tau)} \left(\hat{\sigma}(\tau) - \Delta(\tau, x(\tau))\right) d\tau.$$

for any $t \in [kT_s, (k+1)T_s)$. By the adaptive law in equation (2.9), $\hat{\sigma}(\tau)$ is constant over $[kT_s, (k+1)T_s)$, which is equal to $\hat{\sigma}(kT_s)$. So the preceding equation can be rewritten as

$$\tilde{x}(t) = e^{-a(t-kT_s)}\tilde{x}(kT_s) + \int_{kT_s}^{t} e^{-a(t-\tau)} d\tau \hat{\sigma}(kT_s) - \int_{kT_s}^{t} e^{-a(t-\tau)} \Delta(\tau, x(\tau)) d\tau$$

for any $t \in [kT_s, (k+1)T_s)$. Since $\tilde{x}(t)$ is continuous, the preceding equation implies

$$\begin{aligned}
\tilde{x}(kT_s + T_s) =& e^{-aT_s}\tilde{x}(kT_s) + \int_{kT_s}^{kT_s+T_s} e^{-a(kT_s+T_s-\tau)} d\tau \hat{\sigma}(kT_s) \\
&- \int_{kT_s}^{kT_s+T_s} e^{-a(kT_s+T_s-\tau)} \Delta(\tau, x(\tau)) d\tau \\
=& e^{-aT_s}\tilde{x}(kT_s) + \frac{1}{a}\left(1 - e^{-aT_s}\right)\hat{\sigma}(kT_s) - \int_{kT_s}^{kT_s+T_s} e^{-a(kT_s+T_s-\tau)} \Delta(\tau, x(\tau)) d\tau
\end{aligned}$$

which implies

$$\tilde{x}(kT_s + T_s) = -\int_{kT_s}^{kT_s+T_s} e^{-a(kT_s+T_s-\tau)}\Delta(\tau, x(\tau))d\tau \tag{2.12}$$

given the adaptive law in equation (2.9).

Let us now consider the term at the right-hand side in equation (2.12). Since the exponential function $e^{-a(kT_s+T_s-\tau)}$ is always positive and $\Delta(t, x(t))$ is continuous, by the first mean value theorem, there exist $\tau^* \in [kT_s, (k+1)T_s]$ such that

$$\begin{aligned}\int_{kT_s}^{kT_s+T_s} e^{-a(kT_s+T_s-\tau)}\Delta(\tau, x(\tau))d\tau &= \int_{kT_s}^{kT_s+T_s} e^{-a(kT_s+T_s-\tau)}d\tau \cdot \Delta(\tau^*, x(\tau^*)) \\ &= \frac{1}{a}\left(1 - e^{-aT_s}\right)\Delta(\tau^*, x(\tau^*)).\end{aligned}$$

Applying this equation into equation (2.12) implies

$$\tilde{x}(kT_s + T_s) = -\frac{1}{a}\left(1 - e^{-aT_s}\right)\Delta(\tau^*, x(\tau^*)).$$

Meanwhile, the adaptive law in equation (2.9) means

$$\tilde{x}(kT_s + T_s) = -\frac{1}{a}(e^{aT_s} - 1)\hat{\sigma}(t)$$

for any $t \in [kT_s + T_s, kT_s + 2T_s)$. Combining these two equations yields

$$\Delta(\tau^*, x(\tau^*)) = e^{aT_s}\hat{\sigma}(t).$$

Therefore, for any $t \in [kT_s + T_s, kT_s + 2T_s)$,

$$\begin{aligned}&\|\Delta(t, x(t)) - \hat{\sigma}(t)\| \\ &\leq \|\Delta(t, x(t)) - e^{-aT_s}\Delta(\tau^*, x(\tau^*))\| \\ &\leq \|\Delta(t, x(t)) - \Delta(\tau^*, x(\tau^*))\| + \|(1 - e^{-aT_s})\Delta(\tau^*, x(\tau^*))\| \\ &\leq \int_{\tau^*}^{t} \|\dot{\Delta}(\tau, x(\tau))\|d\tau + \|1 - e^{-aT_s}\|\|\Delta(\tau^*, x(\tau^*))\| \\ &\leq \int_{kT_s}^{kT_s+2T_s} \|\dot{\Delta}(\tau, x(\tau))\|d\tau + \|1 - e^{-aT_s}\|\|\Delta(\tau^*, x(\tau^*))\|\end{aligned} \tag{2.13}$$

where the last inequality holds because $\tau^* \geq kT_s$.

Note that for any $t \geq 0$,

$$\begin{aligned}\frac{d}{dt}\Delta(t, x(t)) =& \frac{\partial \Delta}{\partial t} + \frac{\partial \Delta}{\partial x}\dot{x} \\ =& \frac{\partial \Delta}{\partial t} + \frac{\partial \Delta}{\partial x}(f(x,u) + \Delta(t,x)).\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\left\|\dot{\Delta}(t, x(t))\right\| \\ \leq& \left\|\frac{\partial \Delta}{\partial t}\right\| + \left\|\frac{\partial \Delta}{\partial x}\right\| (\|f(x,u)\| + \|\Delta(t,x)\|) \\ \leq& \, l_{\Delta_t}\|x\| + b_{\Delta_t} + (l_{\Delta_x}\|x\| + b_{\Delta_x}) \left(\max_{x\in\mathcal{X}, u\in\mathcal{U}} \|f(x,u)\| + l_\Delta \|x\| + b_\Delta\right) \\ \leq& \, l_{\Delta_t} \max_{x\in\mathcal{X}} \|x\| + b_{\Delta_t} + (l_{\Delta_x} \max_{x\in\mathcal{X}} \|x\| + b_{\Delta_x}) \left(\max_{x\in\mathcal{X}, u\in\mathcal{U}} \|f(x,u)\| + l_\Delta \max_{x\in\mathcal{X}} \|x\| + b_\Delta\right) \\ =& \, r\end{aligned}$$

where the second inequality comes from Assumption 1. Applying the preceding inequality and $\|\Delta(\tau^*, x(\tau^*))\| \leq \theta$ into inequality (2.13) yields

$$\|\Delta(t, x(t)) - \hat{\sigma}(t)\| \leq 2rT_s + \|1 - e^{-aT_s}\|\theta$$

which completes the proof. □

Remark 4. *The adaptive estimator in* (4.9) *and* (2.9) *is an extension of the piecewise-constant adaptive law proposed in* $\mathcal{L}_1$ *adaptive control literature [45, 47, 48] to nonlinear systems with a much simpler adaptation structure. It enables us to derive a tight and uniform bound on* $\|\Delta(t, x(t)) - \hat{\sigma}(t)\|$. *Notice that the bound* $\gamma(T_s)$ *can be arbitrarily small as long as the sampling period* T_s *is small enough, which implies that the estimation accuracy is subject to the hardware limitation. Another advantage of using the piecewise-constant adaptive law is the ease of implementation in computer-controlled systems, compared with projection-based adaptive laws [45, 49]. Notice that a small* T_s *will not significantly increase the computation workload because the major computational cost in the AMPC is to solve Problem 2, which takes place every* δ *unit-of-time and* δ *is much larger than* T_s.

With inequality (2.10), we know that $\Delta(t, x(t))$ will stay in the ball centered at $\hat{\sigma}(t)$ with the radius $\gamma(T_s)$. Accordingly, at time t_k, we have

$$\Delta(t_k, x(t_k)) \in \Lambda_k \triangleq \{w \in \mathbb{R}^n \mid \|w - \hat{\sigma}(t_k)\| \leq \gamma(T_s)\}. \tag{2.14}$$

The set Λ_k will be used as the initial set for the prediction of the feasible set of the uncertainty, $\mathcal{W}_k(t)$, over the prediction horizon $[t_k, t_k + T)$.

2.2.2 Prediction of Feasible Sets for the Uncertainty

As discussed in Subsection 2.2.1, the adaptive estimator can only provide set-valued estimates of $\Delta(t, x(t))$ in the past (by time t). In the MPC framework, however, one important step is to predict the state trajectory. It requires the future information on $\Delta(t, x)$ as indicated in Problem 2. Therefore, at time t_k we must predict and quantify the change of $\Delta(t, x)$ over the time interval $[t_k, t_k + T)$, i.e. we must predict the set $\mathcal{W}_k(t)$ over $[t_k, t_k + T)$, in which $\Delta(t, x)$ stays for sure.

Consider the error dynamics of $\epsilon(t) = x(t) - x(t_k)$ over $[t_k, t_k + T)$:

$$\begin{aligned}
\frac{d}{dt}\|\epsilon(t)\| \leq \|\dot{\epsilon}(t)\| = \|\dot{x}(t)\| &= \|f(x(t), u(t)) + \Delta(t, x(t))\| \\
&\leq l_x\|x(t)\| + l_u\|u(t)\| + l_\Delta\|x(t)\| + b_\Delta \\
&\leq (l_x + l_\Delta)\,\|x(t)\| + \xi \\
&\leq (l_x + l_\Delta)\,\|x(t) - x(t_k)\| + (l_x + l_\Delta)\,\|x(t_k)\| + \xi
\end{aligned}$$

where the second inequality comes from inequality (5.12) and $\xi = l_u \max_{u\in\mathcal{U}} \|u\| + b_\Delta$. Solving this differential inequality over $[t_k, t_k + T)$ with the initial condition $\epsilon(t_k) = 0$ implies

$$\|\epsilon(t)\| \leq \left(\|x(t_k)\| + \frac{\xi}{l_x + l_\Delta}\right)\left(e^{(l_x+l_\Delta)(t-t_k)} - 1\right).$$

Applying this inequality into inequality (5.13) yields

$$\begin{aligned}\|\Delta(t,x(t))-\Delta(t_k,x(t_k))\| &\le l_\Delta\|x(t)-x(t_k)\|+l_t(t-t_k)\\ &\le l_\Delta\left(\|x(t_k)\|+\frac{\xi}{l_x+l_\Delta}\right)\left(e^{(l_x+l_\Delta)(t-t_k)}-1\right)+l_t(t-t_k).\end{aligned}$$

Since $\|\Delta(t_k,x(t_k))-\hat{\sigma}(t_k)\| \le \gamma(T_s)$ as shown in Lemma 5.11, the preceding inequality indicates

$$\begin{aligned}\|\Delta(t,x(t))-\hat{\sigma}(t_k)\| &\le \|\Delta(t_k,x(t_k))-\hat{\sigma}(t_k)\|+\|\Delta(t,x(t))-\Delta(t_k,x(t_k))\|\\ &\le \gamma(T_s)+l_\Delta\left(\|x(t_k)\|+\frac{\xi}{l_x+l_\Delta}\right)\left(e^{(l_x+l_\Delta)(t-t_k)}-1\right)+l_t(t-t_k)\\ &\triangleq \phi(x(t_k),t-t_k),\end{aligned}$$

which means that

$$\Delta(t,x(t))\in\hat{\mathcal{W}}_k(t)$$

holds for any $t\in[t_k,t_k+T)$, where

$$\hat{\mathcal{W}}_k(t)\triangleq\left\{w\in\mathbb{R}^n \mid \|w-\hat{\sigma}(t_k)\|\le\phi(x(t_k),t-t_k)\right\}. \tag{2.15}$$

Note that $\hat{\mathcal{W}}_k(t_k)=\Lambda_k$ where Λ_k is defined in equation (2.14). In fact, $\hat{\mathcal{W}}_k(t)$ can be regarded as the expansion of Λ_k along the prediction horizon $[t_k,t_k+T)$.

To summarize, we have the following lemma:

Lemma 2. *The uncertainty $\Delta(t,x(t))$ always stays inside the set $\hat{\mathcal{W}}_k(t)$ defined in equation (2.15) for any $t\in[t_k,t_k+T)$.*

With this lemma and equation (4.8), we know that $\Delta(t,x(t))$ will stay in both Ω and $\hat{\mathcal{W}}_k(t)$. Meanwhile, it is inside $\hat{\mathcal{W}}_{k-1}(t)$ because $\hat{\mathcal{W}}_{k-1}(t)$ is a set-valued estimate of $\Delta(t,x(t))$ at time t_{k-1}. Accordingly, we define the feasible set of $\Delta(t,x(t))$ by

$$\mathcal{W}_k(t)\triangleq\Omega\cap\hat{\mathcal{W}}_k(t)\cap\hat{\mathcal{W}}_{k-1}(t) \tag{2.16}$$

for any $t \in [t_k, t_k + T)$, which can be computed at time t_k and $\hat{\mathcal{W}}_{-1}(t) = \Omega$. The reason to include $\hat{\mathcal{W}}_{k-1}(t)$ in the definition is to ensure $\mathcal{W}_k(t) \subseteq \mathcal{W}_{k-1}(t)$ for any $t \in [t_k, t_{k-1} + T)$, which will be used later in stability analysis. In general, most elements in $\hat{\mathcal{W}}_k(t)$ will be included in $\hat{\mathcal{W}}_{k-1}(t)$ over $[t_k, t_{k-1}+T)$ since $\hat{\mathcal{W}}_k(t)$ is always a more accurate set-valued estimate of $\Delta(t, x)$ compared with $\hat{\mathcal{W}}_{k-1}(t)$, which is computed at t_{k-1}.

The variation in $\mathcal{W}_k(t)$ is described in Figure 2.2. At $t = t_k$, we have $\hat{\mathcal{W}}_k(t_k) = \Lambda_k$ as mentioned before. Since $\gamma(T_s)$ can be very small given a short sampling period T_s, the set $\hat{\mathcal{W}}_k(t_k)$ will be small, according to its definition in equation (2.15), which implies that $\mathcal{W}_k(t_k)$ is also small. As time elapses, $\hat{\mathcal{W}}_k(t)$ continuously grows until it completely covers Ω defined in equation (4.8). Then $\mathcal{W}_k(t)$ becomes equal to Ω. In this process, the growth rate plays an important role. The smaller the rate is, the slower $\hat{\mathcal{W}}_k(t)$ expands, which will lead to a smaller feasible set for $\Delta(t, x)$ for Problem 2. As the result, Problem 2 will admit a less conservative solution.

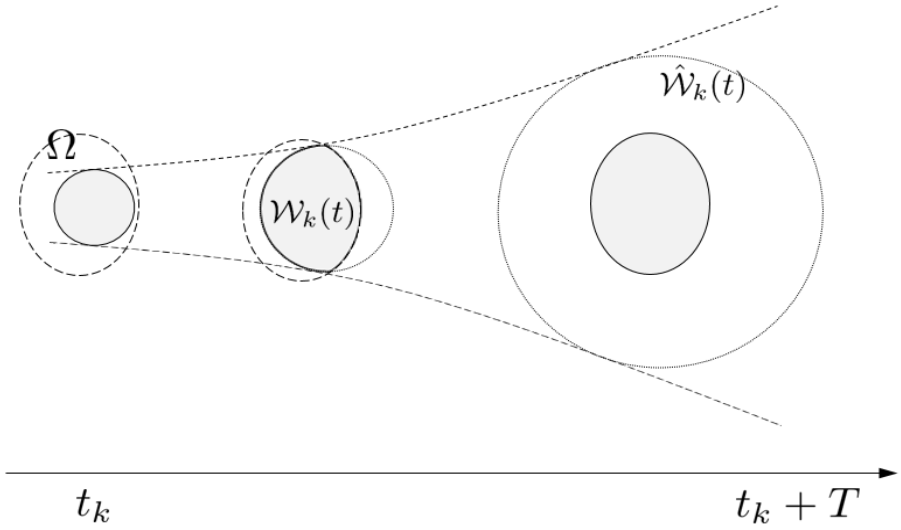

Figure 2.2: Prediction of the sets containing $\Delta(t, x(t))$

With the definition of the time-varying set $\mathcal{W}_k(t)$, the AMPC algorithm is summarized as follows.

Table 2.1: Adaptive MPC Algorithm

At time $t = t_0$,
1 Compute $\gamma(T_s)$ according to (2.11);
2 Run the estimator defined in (4.9) and (2.9);
At time $t = t_k$,
3 Predict $\mathcal{W}_k(t)$ over $[t_k, t_k + T)$ based on (2.16);
4 Solve Problem 2 for $u_k^*(t)$ over $[t_k, t_k + T)$;
5 Set $u(t) = u_k^*(t)$ for all $t \in [t_k, t_k + \delta)$.

Remark 5. *Notice that the uncertainty does not depend on the inputs in our formulation. When the system contains input-dependent uncertainties, the proposed AMPC algorithm cannot be directly applied because the growth rate of the uncertainties may be unbounded due to possible discontinuity in the input $u(t)$. One possible solution is to add constraints on the input in Problem 2 so that the change in $u(t)$ can be bounded. Meanwhile, the set $\hat{\mathcal{W}}_k(t)$ has to be re-defined since it may grow in a piecewise continuous manner. We will investigate state/input-dependent uncertainties in the future.*

Remark 6. *Notice that the set $\mathcal{W}_k(t)$ provides a more accurate set-valued estimate of the uncertainty, compared with the traditional min-max MPC which considers the worst-case static set, Ω, for the uncertainty. With more accurate estimates of the uncertainty in Problem 2, the cost function $J[\hat{u}, \hat{\Delta}]$ can be further minimized which leads to higher control performance.*

2.2.3 Stability Analysis

This subsection discussion the stability of the AMPC algorithm. We first introduce the following assumptions that are standard in MPC formulation [10, 19, 50].

Assumption 2. *There exist a positive constant $c \in \mathbb{R}^+$, class $\mathcal{K}$ function β, α_1, $\alpha_2 : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, and a function $h : \mathbb{R}^n \to \mathbb{R}^m$ with $h(0) = 0$ such that:*

(i) *$\mathcal{X}_f \subseteq \mathcal{X}_U \triangleq \{x \in \mathcal{X} \mid h(x) \in \mathcal{U}\}$, and $0 \in int(\mathcal{X}_f)$;*

(ii) *$\mathcal{X}_f$ is a RPI set for system* (5.11) *in closed-loop with $u = h(x)$ and $\Delta(t,x) \in \Omega$;*

(iii) *$L(x,u) \geq \beta(\|x\|)$ holds for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$;*

(iv) *$\alpha_1(\|x\|) \leq V_f(x) \leq \alpha_2(\|x\|)$ for all $x \in \mathcal{X}_f$;*

(v) *The following inequality holds for all $x \in \mathcal{X}_f$, and $w \in \Omega$.*

$$\frac{\partial V_f(x)}{\partial x}\left(f(x,h(x))+w\right) \leq -L(x,h(x)) + c. \tag{2.17}$$

With Assumption 2, the main result is presented as follows.

Theorem 1. *Consider the system* (5.11) *controlled by the AMPC law. If Assumptions 1 and 2 hold, then the overall closed-loop system is uniformly ultimately bounded.*

Proof. First, let us define multiple Lyapunov functions for the closed-loop system. For any $t \in [t_k, t_{k+1})$, let

$$V_k(t,x) \triangleq \min_{\hat{u}(\tau)\in\mathcal{U}} \max_{\hat{\Delta}(\tau)\in\mathcal{W}_k(\tau),\forall\tau\in[t,t_k+T)} \int_t^{t_k+T} L(\hat{x}_k(\tau),\hat{u}(\tau))d\tau + V_f(\hat{x}_k(t_k+T))$$

where $\hat{x}_k(t)$ is the predicted state at the kth iteration. Notice that given $\hat{x}_k(t) = x$, we have

$$\begin{aligned}
&\min_{\hat{u}^0(\tau)\in\mathcal{U},\,\hat{\Delta}(\tau)=0} \int_t^{t_k+T} L(\hat{x}_k^0(\tau),\hat{u}^0(\tau))d\tau + V_f(\hat{x}_k^0(t_k+T)) \leq V_k(t,x) \\
\leq &\min_{\hat{u}^\Omega(\tau)\in\mathcal{U}} \max_{\hat{\Delta}(\tau)\in\Omega} \int_t^{t_k+T} L(\hat{x}_k^\Omega(\tau),\hat{u}^\Omega(\tau))d\tau + V_f(\hat{x}_k^\Omega(t_k+T))
\end{aligned}$$

where $\hat{x}_k^0(\tau)$ and $\hat{x}_k^\Omega(\tau)$ are generated under $(0,\hat{u}^0)$ and $(\Omega,\hat{u}^\Omega)$, respectively. By Assumption (iii) and (iv), we know that there exist two class $\mathcal{K}$ functions χ_1, χ_2 and a positive constant ζ, independent of k, such that $\chi_1(\|x\|) \leq V_k(t,x) \leq \chi_2(\|x\|) + \zeta$.

With the Lyapunov function $V_k(t,x)$, we have

$$\begin{aligned}
V_k(t_k,x(t_k)) &= \min_{\hat{u}(\tau)\in\mathcal{U}} \max_{\hat{\Delta}(\tau)\in\mathcal{W}_k(\tau),\forall\tau\in[t_k,t_k+T)} \int_{t_k}^{t_k+T} L(\hat{x}_k(\tau),\hat{u}(\tau))d\tau + V_f(\hat{x}_k(t_k+T)) \\
&= \int_{t_k}^{t_k+T} L(\hat{x}_k^*(\tau),\hat{u}_k^*(\tau))d\tau + V_f(\hat{x}_k^*(t_k+T)) \\
\hat{x}_k(t_k) &= \hat{x}_k^*(t_k) = x(t_k)
\end{aligned}$$

where $\hat{x}_k^*(t)$ and $\hat{u}_k^*(t)$ are the optimal state and input, respectively, generated at the kth iteration with the initial condition $x(t_k)$. Also, we use $\hat{\Delta}_k^*(t)$ to denote the corresponding worst-case disturbance. Note that the input $\hat{u}_k^*(t)$ will be applied to the plant over the time interval $[t_k, t_{k+1})$.

Next, let us define a feasible control input that can be applied at t_{k+1}:

$$\hat{u}_{k+1}(t) \triangleq \begin{cases} \hat{u}_k^*(t), & \forall t \in [t_{k+1}, t_k + T) \\ h(\hat{x}_{k+1}(t)), & \forall t \in [t_k + T, t_{k+1} + T) \end{cases}$$

where $\hat{x}_{k+1}(t)$ is the predicted state generated by $\hat{u}_{k+1}(t)$ with an arbitrary disturbance $\hat{\Delta}_{k+1}(t) \in \mathcal{W}_{k+1}(t)$ over $[t_{k+1}, t_{k+1} + T)$. The corresponding cost is then

$$J_{k+1}(\hat{u}_{k+1}, \hat{\Delta}_{k+1}) = \int_{t_{k+1}}^{t_{k+1}+T} L(\hat{x}_{k+1}(\tau), \hat{u}_{k+1}(\tau))d\tau + V_f(\hat{x}_{k+1}(t_{k+1} + T))$$
$$\hat{x}_{k+1}(t_{k+1}) = x(t_{k+1}).$$

We now examine the difference between V_k and J_{k+1}, which is shown as.

$$\begin{aligned} & J_{k+1}(\hat{u}_{k+1}, \hat{\Delta}_{k+1}) - V_k(t_k, x(t_k)) \\ &= \int_{t_{k+1}}^{t_{k+1}+T} L(\hat{x}_{k+1}, \hat{u}_{k+1})d\tau + V_f(\hat{x}_{k+1}(t_{k+1} + T)) - V_k(t_k, x(t_k)) \\ &= \underbrace{\int_{t_{k+1}}^{t_k+T} L(\hat{x}_{k+1}, \hat{u}_{k+1})d\tau + \int_{t_k}^{t_{k+1}} L(x, u)d\tau + V_f(\hat{x}_{k+1}(t_k + T))}_{\Phi_1} - V_k(t_k, x(t_k)) \\ &\quad + \underbrace{\int_{t_k+T}^{t_{k+1}+T} L(\hat{x}_{k+1}, \hat{u}_{k+1})d\tau + V_f(\hat{x}_{k+1}(t_{k+1} + T)) - V_f(\hat{x}_{k+1}(t_k + T))}_{\Phi_2} \\ &\quad - \int_{t_k}^{t_{k+1}} L(x, u)d\tau. \end{aligned} \tag{2.18}$$

Integrating inequality (2.17) in Assumption 2 over the time interval $[t_k + T, t_{k+1} + T)$, we have

$$\Phi_2 \leq \delta c. \tag{2.19}$$

Let us now consider Φ_1. Notice that Φ_1 is the cost of system (5.11) over $[t_k, t_k + T)$ with the initial condition $x(t_k)$, the input $u_k^*(t)$, and the disturbance

$$\tilde{\Delta}(t) \triangleq \begin{cases} \Delta(t, x(t)), & \forall t \in [t_k, t_{k+1}) \\ \hat{\Delta}_{k+1}(t), & \forall t \in [t_{k+1}, t_k + T) \end{cases}.$$

Based on the definition of $\mathcal{W}_k(t)$ in equations (2.16), we know $\mathcal{W}_{k+1}(t) \subseteq \mathcal{W}_k(t)$ for $t \in [t_{k+1}, t_k + T)$. Therefore, $\hat{\Delta}_{k+1}(t) \in \mathcal{W}_{k+1}(t) \subseteq \mathcal{W}_k(t)$ and $\tilde{\Delta}(t) \in \mathcal{W}_k(t)$. Also, we realize that $V_k(t_k, x(t_k))$ is the cost of system (5.11) over $[t_k, t_k + T)$ with the same initial condition $x(t_k)$, the same input $u_k^*(t)$, but the worst-case disturbance over $\mathcal{W}_k(t)$. Consequently, we can conclude that $\Phi_1 - V_k(t_k, x(t_k)) \leq 0$. Applying this inequality and inequality (2.19) into inequality (2.18) implies

$$J_{k+1}(\hat{u}_{k+1}, \hat{\Delta}_{k+1}) - V_k(t_k, x(t_k)) \leq \delta c - \int_{t_k}^{t_{k+1}} L(x, u) d\tau.$$

Since $\hat{\Delta}_{k+1}(t)$ is any disturbance in $\mathcal{W}_{k+1}(t)$, we have

$$\max_{\hat{\Delta}_{k+1}(\tau) \in \mathcal{W}_{k+1}(\tau)} J_{k+1}(\hat{u}_{k+1}, \hat{\Delta}_{k+1}) - V_k(t_k, x(t_k)) \leq \delta c - \int_{t_k}^{t_{k+1}} L(x, u) d\tau.$$

and therefore

$$\begin{aligned} & V_{k+1}(t_{k+1}, x(t_{k+1})) - V_k(t_k, x(t_k)) \\ =& \min_{\hat{u}_{k+1}(\tau) \in \mathcal{U}} \max_{\hat{\Delta}_{k+1}(\tau) \in \mathcal{W}_{k+1}(\tau)} J_{k+1}(\hat{u}_{k+1}, \hat{\Delta}_{k+1}) - V_k(t_k, x(t_k)) \\ \leq& \delta c - \int_{t_k}^{t_{k+1}} L(x(\tau), u(\tau)) d\tau \\ \leq& \delta c - \int_{t_k}^{t_{k+1}} \beta(\|x(\tau)\|) d\tau \end{aligned}$$

where the last inequality comes from item (iii) in Assumption 2. Notice that for any $t \in [t_k, t_{k+1})$, $V_k(t, x(t)) \leq V_k(t_k, x(t_k))$ holds. Therefore, by Theorem 2.5 in [10], we can conclude that the overall closed-loop system is uniformly ultimately bounded. □

2.3 Simulation

This section presents the simulation results that demonstrate the performance of the proposed AMPC algorithm. The system under consideration is the simplified model

of the short-period dynamics of an aircraft [51]:

$$\dot{x} = \begin{bmatrix} -0.1686 & 0.9562 \\ -0.4002 & -0.3393 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1.5313 \end{bmatrix} u + \Delta(t, x)$$

where the state $x = [x_1,\ x_2]^\top$ includes the angle of attack and the pitch rate and the input $u(t)$ represents the elevator deflection. The state must satisfy $|x_1(t)| \leq 2$ and $|x_2(t)| \leq 2$ for any $t \geq 0$. The input constraint is $-2 \leq u(t) \leq 2$. We set the prediction horizon to be $T = 1.5$ and $\delta = 0.1$. The running cost function is $L(x, u) = x^\top x + 0.01u^\top u$ and the terminal cost function is $V_f(x) = x^\top x$. With this setting and $\mathcal{X}_f = \mathcal{X}$, Assumption 2 can be verified.

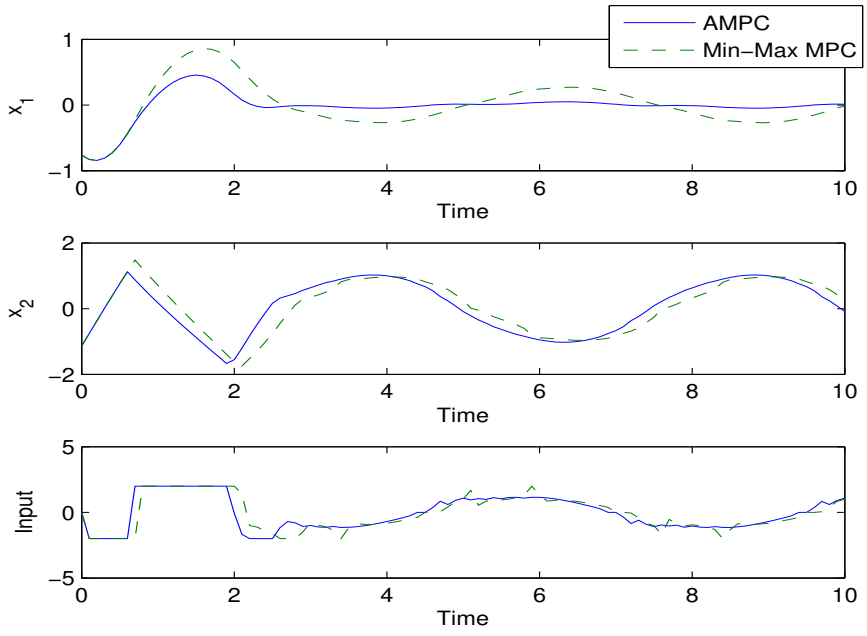

Figure 2.3: Comparison between AMPC and min-max MPC

In the first simulation, the external disturbance is assumed to be a sinusoidal signal $\Delta(t, x) = [1, 1]^\top \sin(0.4\pi t)$, which is independent of the state. The system is run for 10 seconds. We compare the simulation results with the min-max MPC approach. Figure 2.3 shows the state trajectories of the systems under AMPC (solid line) and min-max MPC (dashed line). From the top two plots, we see that x_1 by

AMPC stays much closer to the origin than that generated by min-max MPC, while the trajectories of x_2 are comparable for both cases. The input trajectories are also comparable as as shown in the bottom plot, which suggests that AMPC admits a smaller cost over the running time.

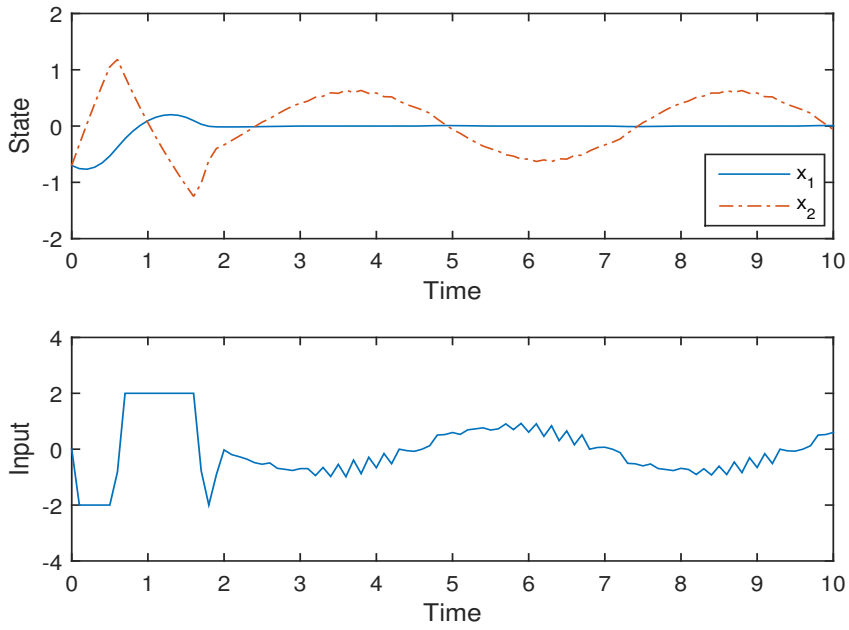

Figure 2.4: The states and the inputs generated by AMPC

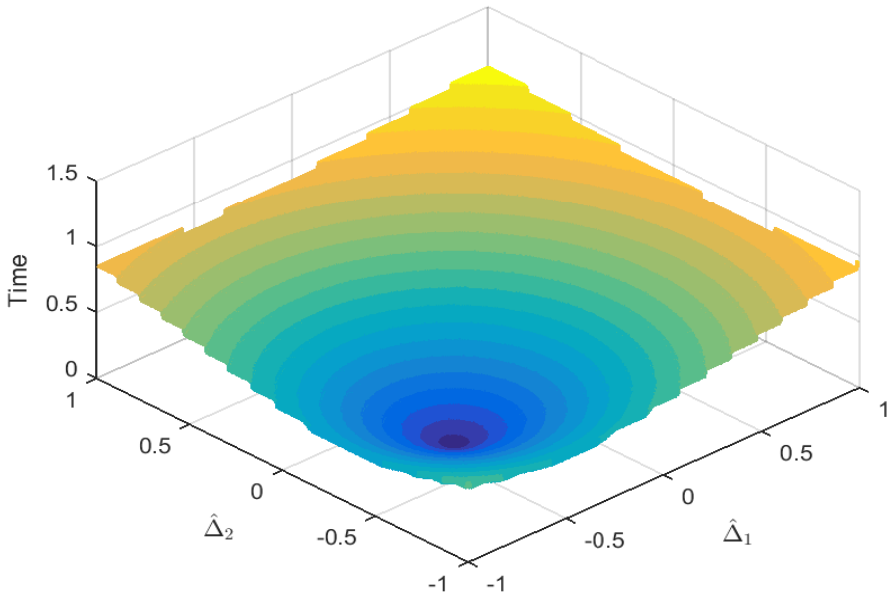

Figure 2.5: The feasible sets of the uncertainty $\hat{\mathcal{W}}_k(t)$ for $t \in [t_k, t_k + T)$

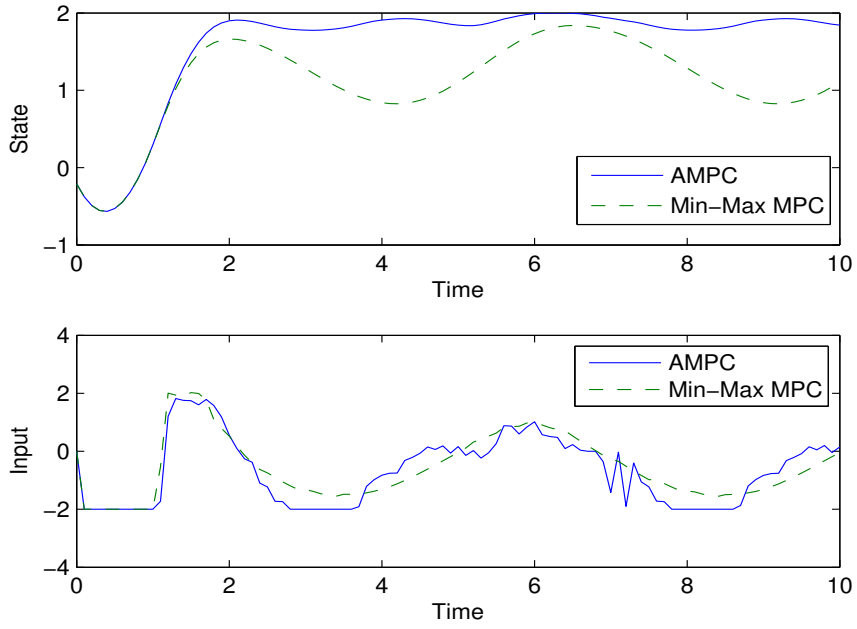

Figure 2.6: Comparison between AMPC and min-max MPC

The second simulation considers state-dependent uncertainties, where

$$\Delta(t,x) = \begin{bmatrix} 0.3 & 0.1 \\ 0 & -0.2 \end{bmatrix} x + \begin{bmatrix} 0.2\sin(0.4\pi t) \\ 0.2\sin(0.4\pi t) \end{bmatrix}.$$

Figure 2.4 shows that the system under AMPC achieves similar performance to the first simulation, which is not surprising. Figure 2.5 plots the predicted set of the uncertainty, $\hat{\mathcal{W}}_k(t)$, over $[t_k, t_k + T)$, computed at $t_k = 3$ using equations (2.15). It can be observed that the expansion of the set is consistent to the plot in Figure 2.2.

The third simulation examines whether the proposed AMPC algorithm can be applied to tracking problems. The objective is to transfer the output $y(t) = x_1$ to the reference level 2, i.e., $y(t)$ tracks 2. The running cost function is

$$L(x,u) = (x_1 - 2)^\top (x_1 - 2) + 0.01u^\top u$$

and the terminal cost is chosen to be $V_f(x) = (x_1 - 2)^T (x_1 - 2)$. The trajectory of the output $y(t)$ obtained from applying the AMPC algorithm is plotted in Figure 2.6. Again, it is compared with the output of the min-max MPC approach with $\mathcal{W}_k = \Omega$. Thanks to the adaptive estimator and the reduced feasible set for the uncertainty $\mathcal{W}_k(t)$, we can see that the performance of the proposed AMPC is superior to that of the min-max MPC from Figure 2.6.

2.4 Conclusion

As discussed in context, the proposed AMPC algorithm enhances the control performance by increasing estimation and prediction accuracy. We realize that timing is very important in this framework: (i) the estimation error can be very small with a short sampling period in adaptation; and (ii) if the uncertainty changes fast, the set $\mathcal{W}_k(t)$ will converge to Ω very soon in prediction, which minimizes the benefit of using adaptation (similar situations happen when the prediction horizon is overlong). Further work will be done to study the impact of timing parameters on control

performance and AMPC feasibility. Another observation is that the entire framework in this chapter is continuous-time, which matches the idea of fast adaptation in the adaptive control literature. However, implementing such continuous-time AMPC will be computationally expensive. The continuous-time min-max optimization in Problem 2 is challenging, especially when the plant is nonlinear. A more reasonable way is to seek discrete-time AMPC algorithms, even for continuous-time plants. The recent advances on sampled-data systems in fact provide the possibility [52, 53]. This is also one of the reasons to introduce the piecewise constant adaptive law, as mentioned in Remark 4. Developing discrete-time AMPC algorithms will be the work to be done in the future.

Chapter 3

Lebesgue-Approximation-Based Model Predictive Control for Nonlinear Sampled-Data Systems with Measurement Noises

For computer-controlled systems, the MPC algorithms should be in discrete-time in order to implement on actual systems. In this case, the sampling and computation of optimal control inputs are triggered in discrete-time manners; and the models used in the finite-horizon optimal control problem (FHOCP) are discrete-time. This chapter presents a discrete-time MPC algorithm, based on Lebesgue approximation, for nonlinear sampled-data systems. In this algorithm, the sampling instants are triggered by a self-triggered scheme. The predictive model in the FHOCP is iterated in an aperiodic manner subject to the Lebesgue approximation model. Sufficient conditions are derived on feasibility and stability of the closed-loop systems with the guarantee of exclusion of Zeno behavior.

3.1 Problem Formulation

Definition 1. *A continuous function $\alpha : \mathbb{R}_o^+ \to \mathbb{R}_0^+$ belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0, \lim_{s\to\infty} \alpha(s) = \infty$.*

Definition 2. *The state $x(t)$ of a system $\dot{x} = f(x)$ is called uniformly ultimately bounded (UUB) with ultimate bound b if there exist positive constants b and c,*

independent of $t_0 \geq 0$*, and for every* $a \in (0, c)$*, there is* $T = T(a, b) \geq 0$*, independent of* t_0 *, such that* $\|x(t_0)\| \leq a$ *implies* $\|x(t)\| \leq b$ *for any* $t \geq t_0 + T$*.*

Consider a nonlinear continuous system:

$$\dot{x}(t) = f(x(t), u(t)),\ x(t_0) = x_0 \tag{3.1}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ is the system state, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, and $f : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}^n$. The sets $\mathcal{X}$,$\mathcal{U}$ are the constraint sets on the state and input, respectively. We assume $f(x, u)$ is locally Lipschitz, i.e., there exists a positive constant L_f so that $\forall x, y \in \mathcal{X}, u \in \mathcal{U}$, the following inequality holds:

$$\|f(x, u) - f(y, u) \leq L_f \|x - y\|.$$

The main idea of MPC is described as follows: at time t_k, the system obtains the sampled state $\tilde{x}(t_k)$ and uses it as the initial point, the controller solves the FHOCP over the prediction horizon $[t_k, t_k + T_k]$, where T_k is the horizon length, and identify the next sampling time instant t_{k+1}. Then the calculated optimal control input will be applied to the plant over the time interval ($t_{k+1} \leq t_k + T_k$). The next sampling and prediction will start at t_{k+1}.

We assume the sampled state contains measurement noise, i.e.,

$$\bar{x}(t_k) = x(t_k) + \omega(t_k)$$

where $\omega(t_k)$ is the noise satisfying

$$\|\omega(t_k) \leq \sigma\|. \tag{3.2}$$

In our discrete-time MPC framework, the sampling time instants and the calculation of FHOCP must be discrete. In general, the next sampling time t_{k+1} can be expressed as

$$t_{k+1} = t_k + \phi_k$$

where $\phi_k \in \mathbb{R}^+$ is the inter-sampling time interval to be defined. The cost function of the FHOCP at the kth computation is

$$J[\hat{u}|x(t_k)] = \int_{t_k}^{t_k+T_k} L(\hat{x}, \hat{u})d\tau + V_f(\hat{x}(t_k + T_k)) \tag{3.3}$$

where $\hat{x} : \mathbb{R}^+ \to \mathbb{R}^n$ and $\hat{u} : \mathbb{R}^+ \to \mathbb{R}^m$ are the predictive state and input, respectively, $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+$ is the running cost function, and $V_f : \mathbb{R}^n \to \mathbb{R}^+$ is the terminal cost function.

Our objective is to design a design a completely discrete-time and cost-efficient MPC algorithm to stabilize system (4.10), in the presence of measurement noises and state/input constraints.

3.2 LAMPC Algorithm

This section presents the LAMPC algorithm. To begin with, we define the LAM for FHOCP starts at time t_k ,

$$\begin{aligned}
\hat{x}_k^{i+1} &= \hat{x}_k^i + d\hat{x}_k^i \frac{\hat{D}_k^i}{\|d\hat{x}_k^i\|},\ \hat{x}_k^0 = \hat{x}_k \\
t_k^{i+1} &= t_k^i + \frac{\hat{D}_k^i}{\|d\hat{x}_k^i\|},\ t_k^0 = t_k \\
d\hat{x}_k^i &= f(\hat{x}_k^i, \hat{u}_k^i),
\end{aligned} \tag{3.4}$$

where $\hat{x}_k^i, \hat{u}_k^i$, and $\hat{D}_k^i = D(\hat{x}_k^i, \hat{u}_k^i)$ are the predictive state, input, and state-dependent discretization level of LAM at the ith iteration, respectively. The threshold function $D : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+$ will be discussed later.

Let $N \in \mathbb{N}$ be the number of iterations, and the horizon length can be described as $T_k = t_k^N - t_k^0$. Using zero-order-hold approximation $\hat{x}(\tau)\hat{x}_k^i, \hat{u}(\tau) = \hat{u}_k^i$ for any

$\tau \in [t_k^i, t_k^{i+1})$, and the cost function (4.11) can be rewritten as

$$\begin{aligned} J[\hat{u}_k|\bar{x}(t_k)] &= \sum_{i=0}^{N-1} \int_{t_k^i}^{t_k^{i+1}} L(\hat{x}(\tau), \hat{u}(\tau))d\tau + V_f(\hat{x}(t_k^N)) \\ &= \sum_{i=0}^{N-1} L(\hat{x}_k^i, \hat{u}_k^i)(t_k^{i+1} - t_k^i) + V_f(\hat{x}_k^N) \\ &= \sum_{i=0}^{N-1} L(\hat{x}_k^i, \hat{u}_k^i)\frac{\hat{D}_k^i}{\|d\hat{x}_k^i\|} + V_f(\hat{x}_k^N). \end{aligned}$$

Then we can state the FHOCP at t_k as a discrete-time optimal control problem with $\hat{x}_k^0 = \bar{x}(t_k)$:

$$\begin{aligned} V(\hat{x}(t_k)) &= \min_{\hat{u}_k^i \in \mathcal{U}} J[\hat{u}_k|\hat{x}(t_k)] \\ \text{subject to} \quad \hat{x}_k^{i+1} &= \hat{x}_k^i + d\hat{x}_k^i \frac{\hat{D}_k^i}{\|d\hat{x}_k^i\|},\ i = 0, 1, \cdots, N-1 \\ \hat{x}_k^i &\in \mathcal{X}_i,\ \hat{x}_k^N \in \mathcal{X}_N = \mathcal{X}_f \end{aligned} \tag{3.5}$$

where $\mathcal{X}_i$ are the constraint on the predictive state $\hat{x}_k^i$, and $\mathcal{X}_N$ is the set of terminal states.

Let $\hat{u}_k^{i,*}$ $(i = 0, \cdots, N-1)$ be the optimal solutions and $\hat{x}_k^{i,*} \in \mathcal{X}_i$ be the corresponding optimal states. Accordingly, $d\hat{x}_k^i$ and $\hat{D}_k^i$ (4.1) in (4.12) can be represented as

$$\hat{D}_k^{i,*} = D(\hat{x}_k^{i,*}, \hat{u}_k^{i,*}),\ d\hat{x}_k^{i,*} = f(\hat{x}_k^{i,*}, \hat{u}_k^{i,*}).$$

With the optimal solution, we can define the next sampling time instant t_{k+1}, following the time iteration as (4.12)

$$t_{k+1} = t_k^{1,*} = t_k + \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}. \tag{3.6}$$

The LAMPC algorithm is summarized as follows.

Table 3.1: LAMPC Algorithm Routine

At time $t = t_k$	
1	Sample the state and obtain $\bar{x}(t_k)$;
2	Solve the FHOCP (4.13) for $\hat{x}_k^{i,*}, \hat{u}_k^{i,*}$;
3	Apply the optimal solution $\hat{u}_k^{0,*}$ to the plant, i.e, set $u(t) = \hat{u}_k^{0,*}$ over $[t_k, t_{k+1})$;
4	Start the next sampling and computation cycle at time t_{k+1} defined by (3.6)

3.3 Feasibility

In order to guarantee feasibility of LAMPC, the constraint sets $\mathcal{X}_i$ in the FHOCP must be correctly defined. Notice that we do not have direct access to $x(t)$, but only access to $\hat{x}_k^{i,*}$. Therefore, to ensure $x(t) \in \mathcal{X}$, the idea is to reduce the constraint set $\mathcal{X}$ by certain amount to obtain $\mathcal{X}_i$ and hopefully $\hat{x}_k^{1,*} \in \mathcal{X}_1$ can imply $x(t) \in \mathcal{X}$. To begin with, we first study the error dynamics between the LAM (4.12) and the continuous-time dynamics (4.10). Let $z(t)$ be defined as

$$z(t) = \bar{x}(t_k) + d\hat{x}_k^{0,*}(t - t_k),\ \forall t \in [t_k, t_{k+1}].$$

Note that $z(t_k) = \hat{x}_k^{0,*} = \bar{x}(t_k)$ and $z(t_{k+1}) = \hat{x}_k^{1,*}$

Lemma 1. *Consider the system* (4.10) *and the signal* $z(t)$*, then for any* $t \in [t_k, t_{k+1}]$*, the following inequality holds*

$$\|x(t) - z(t)\| \leq \epsilon_k \triangleq \hat{D}_k^{0,*}(e^{L_f \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}} - 1) + \sigma e^{L_f \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}}$$

where σ is defined in (3.2).

Proof. According to the definition of $z(t)$, we may get that

$$\dot{z} = d\hat{x}_k^{0,*} = f(\hat{x}_k^{0,*}, \hat{u}_k^{0,*}),\ \forall t \in [t_k, t_{k+1}].$$

Let us set the error as $e(t) = x(t) - z(t)$, and we can get its dynamics as:

$$\dot{e}(t) = f(x(t), \hat{u}_k^{0,*}) - f(\bar{x}(t_k), \hat{u}_k^{0,*})$$

which leads to the inequality

$$\begin{aligned}
\frac{d}{dt}\|e(t)\| &\leq \|\dot{e}(t)\| \\
&= \|f(x(t), \hat{u}_k^{0,*}) - f(\bar{x}(t_k), \hat{u}_k^{0,*})\| \\
&\leq L_f\|x(t) - \bar{x}(t_k)\| \\
&= L_f\|x(t) - z(t) + z(t) - x(t_k)\| \\
&\leq L_f\|e(t)\| + L_f\|z(t) - z(t_k) + z(t_k) - \bar{x}(t_k)\| \\
&\leq L_f\|e(t)\| + L_f\|z(t) - z(t_k)\| \\
&\leq L_f\|e(t)\| + L_f\|d\hat{x}_k^{0,*}(t - t_k)\| \\
&\leq L_f\|e(t)\| + L_f\|d\hat{x}_k^{0,*}(t_{k+1} - t_k)\| \\
&= L_f\|e(t)\| + L_f\|d\hat{x}_k^{0,*}\|\frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|} \\
&= L_f\|e(t)\| + L_f\hat{D}_k^{0,*}.
\end{aligned}$$

Solving the inequality with $\|e(t_k)\| \leq \sigma$, we obtain

$$\begin{aligned}
\|e(t)\| &\leq \hat{D}_k^{0,*}(e^{L_f(t-t_k)} - 1) + \sigma e^{L_f(t-t_k)} \\
&\leq \hat{D}_k^{0,*}(e^{L_f(t_{k+1}-t_k)} - 1) + \sigma e^{L_f(t_{k+1}-t_k)} \\
&= \hat{D}_k^{0,*}(e^{L_f\frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}} - 1) + \sigma e^{L_f\frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}}
\end{aligned}$$

for any $t \in [t_k, t_{k+1}]$. □

Let

$$\epsilon = \sup \epsilon_k = \max_{x\in\mathcal{X}, u\in\mathcal{U}} D(x,u)(e^{L_f\frac{D(x,u)}{\|f(x,u)\|}} - 1) + \sigma e^{L_f\frac{D(x,u)}{\|f(x,u)\|}}$$

and

$$\mathcal{X}_1 \triangleq \mathcal{X} - 2\epsilon$$

where $\mathcal{X}_1$ is the Pontryagin difference between $\mathcal{X}$ and a ball $\mathcal{B}(2\epsilon)$ centered at the origin with the radius 2ϵ as it shows in . With the result of Lemma 1, if $\bar{x}(t_0) \in \mathcal{X} - \epsilon, x(t) \in \mathcal{X}$ can be guaranteed for any $t \in [t_k, t_{k+1})$.

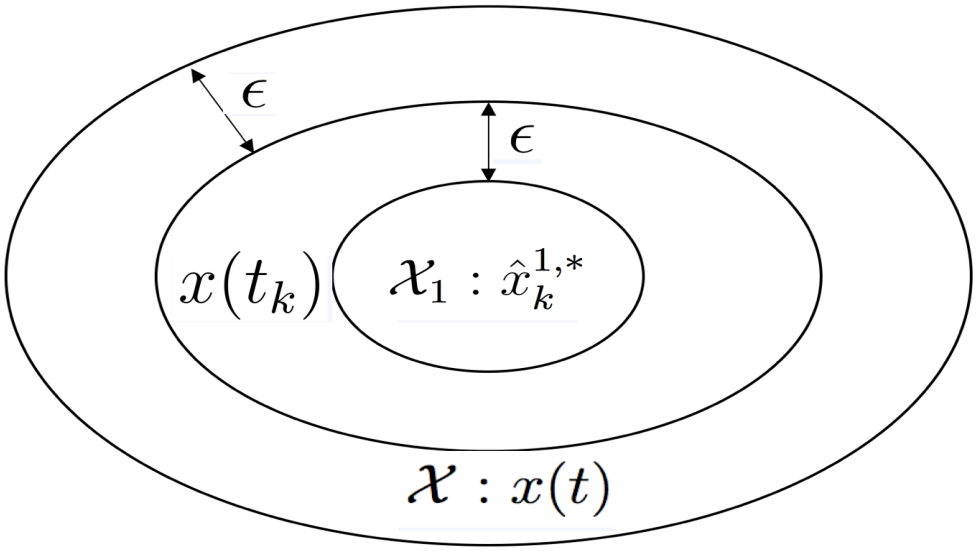

Figure 3.1: The relationship between $\mathcal{X}$ and $\mathcal{X}_1$

Assumption 1. *To ensure feasibility, we assume that for any $x, y \in \mathcal{X}, u \in \mathcal{U}$, there exists a positive constant L_s such that*

$$\|D(x,u)\frac{f(x,u)}{\|f(x,u)\|} - D(y,u)\frac{f(y,u)}{\|f(y,u)\|}\| \leq L_S\|x-y\| \tag{3.7}$$

and there exists a function $h(x) : \mathbb{R}^n \in \mathbb{R}^m,\ h(0) = 0$ such that

$$0 \in int(\mathcal{X}_f) \tag{3.8}$$

$$\mathcal{X}_f + \epsilon(L_s+1)^{N-1} \subset \mathcal{X}_U \triangleq \{x \in \mathcal{X}_{N-1} | h(x) \in \mathcal{U}\} \tag{3.9}$$

$$x \in \mathcal{X}_f + \epsilon(L_s+1)^{N-1} \Rightarrow x + D(x,h(x))\frac{f(x,h(x))}{\|f(x,h(x))\|} \in \mathcal{X}_f \tag{3.10}$$

And we construct an admissible control input for the LAM at the $(k+1)$th computation with the initial condition $\hat{x}_{k+1}^0 = x(t_{k+1})$:

$$\hat{u}_{k+1}^i = \begin{cases} \hat{u}_k^{i+1,*},\ i = 0, 1, \cdots, N-2 \\ h(x_{k+1}^i).\ i = N-1 \end{cases} \tag{3.11}$$

We compute the optimization input from 0 to $(N-1)$th step in each computation cycle. As a result, the ith control input calculated in the current cycle can be mapped to the $(i-1)$th input in next iteration.

With $\hat{u}^i_{k+1}$ and $\hat{x}^0_{k+1} = \bar{x}(t_{k+1})$, the LAM can generate the states at the $(k+1)$th computation cycle as $\hat{x}^i_{k+1}$. Then we will study the difference between $\hat{x}^{i-1}_{k+1}$ and $\hat{x}^{i,*}_k$.

Lemma 2. *Given inequality* (3.7)*, the following inequality holds*

$$\|\hat{x}^{i-1}_{k+1} - \hat{x}^{i,*}_k\| \leq \epsilon_k (L_s + 1)^{i-1},\ i = 1, 2, \cdots, N.$$

Proof. Let's prove the statement using mathematical induction. To begin with, it is obvious that the inequality agrees with the conclusion in Lemma 1 for $i = 0$. Assuming the inequality holds,

According to (4.12), we can get that

$$\begin{aligned}
\hat{x}^{i+1,*}_k &= \hat{x}^{i,*}_k + D(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k) \frac{f(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k)}{\|f(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k)\|} \\
\hat{x}^{i}_{k+1} &= \hat{x}^{i-1}_{k+1} + D(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k) \frac{f(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k)}{\|f(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k)\|}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\|\hat{x}^{i}_{k+1} - \hat{x}^{i+1,*}_k\| \\
=&\|\hat{x}^{i-1}_{k+1} + D(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k) \frac{f(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k)}{\|f(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k)\|} - \hat{x}^{i,*}_k - D(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k) \frac{f(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k)}{\|f(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k)\|}\| \\
\leq&\|\hat{x}^{i-1}_{k+1} - \hat{x}^{i,*}_k\| + \|D(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k) \frac{f(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k)}{\|f(\hat{x}^{i-1}_{k+1}, \hat{u}^{i,*}_k)\|} - D(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k) \frac{f(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k)}{\|f(\hat{x}^{i,*}_k, \hat{u}^{i,*}_k)\|}\|.
\end{aligned}$$

According to (3.7), we can get that

$$\begin{aligned}
\|\hat{x}^{i}_{k+1} - \hat{x}^{i+1,*}_k\| &\leq (1 + L_s)\|\hat{x}^{i-1}_{k+1} - \hat{x}^{i,*}_k\| \\
&\leq \epsilon_k (L_s + 1)^p,
\end{aligned}$$

which finishes the proof by following the same rule of initial assumption. □

According to Lemma 2, we can define the reduced constraint sets in the FHOCP at each iteration as

$$\mathcal{X}_i \triangleq \mathcal{X} - \epsilon(\sum_{p=1}^{i} (L_s + 1)^{p-1} + 1),\ i = 1, \cdots, N-1. \tag{3.12}$$

And Figure 3.2 shows the relationship between $\mathcal{X}$ and constraint set $\mathcal{X}_i$.

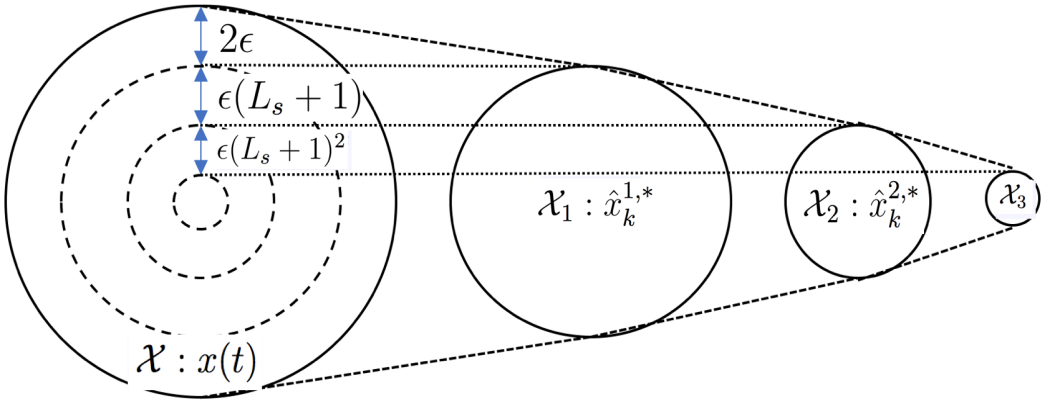

Figure 3.2: The relationship between $\mathcal{X}$ and $\mathcal{X}_3$

Theorem 1. *With* (3.12) *holds, along with the assumptions of* (3.8)-(3.10)*, the FHOCP of* (4.13) *is always feasible.*

Proof. We will prove that for $\hat{x}_k^{i,*} \in \mathcal{X}_i, i = 1, 2, \cdots, N$, there always have $\hat{x}_{k+1}^i \in \mathcal{X}_i, \hat{x}_{k+1}^N \in \mathcal{X}_f$ and $\hat{u}_{k+1}^i \in \mathcal{U}$ as previous iteration, which makes it exists a feasible solution $\hat{u}_{k+1}^i$ of the optimization problem (4.13) at the $(k+1)$th iteration.

To begin with, let us show that $\hat{u}_{k+1}^i \in \mathcal{U}, i = 0, 1, \cdots, N-1$. Based on (3.11), we may get that $\hat{u}_{k+1}^i = \hat{u}_k^{i+1,*} \in \mathcal{U}, i = 0, 1, \cdots, N-2$. As it for $i = N-1$, we can get that

$$\|\hat{x}_{k+1}^{N-1} - \hat{x}_k^{N,*}\| \leq \epsilon(L_s+1)^{N-1},\ \hat{x}_k^{N,*} \in \mathcal{X}_f$$

according to Lemma 2. With (3.9) holds, it may conclude that

$$\hat{x}_{k+1}^{N-1} \in \mathcal{X}_f + \epsilon(L_s+1)^{N-1} \subset \mathcal{X}_U \tag{3.13}$$

which ensure that $\hat{u}_{k+1}^{N-1} \in \mathcal{U}$.

Secondly, we will show that $\hat{x}_{k+1}^i \in \mathcal{X}_i, i = 1, \cdots, N-1$ and $\hat{x}_{k+1}^N \in \mathcal{X}_f$. Based on (3.13), we may have $\hat{x}_{k+1}^N \in \mathcal{X}_f$ according to the assumption of (3.10). Meanwhile, we can get that

$$\|\hat{x}_{k+1}^i - \hat{x}_k^{i+1,*}\| \leq \epsilon(L_s+1)^i,\ \hat{x}_k^{i+1,*} \in \mathcal{X}_{i+1},\ \hat{x}_k^{N,*} \in \mathcal{X}_f \subset \mathcal{X}_{N-1}$$

with Lemma 2 holds. Then we conclude that

$$\hat{x}_{k+1}^i \in \mathcal{X}_{i+1} + \epsilon(L_s+1)^i = \mathcal{X} - \epsilon(\sum_{p=1}^{i}(L_s+1)^{p-1}+1) = \mathcal{X}_i,\ i = 1, \cdots, N-1.$$

which completes the proof. □

3.4 Stability

This section discusses stability of the closed-loop system. Before presenting the main results, we will first define the threshold function $D(x,u)$ because this function will directly affect system stability. If the threshold $\hat{D}_k^i$ is too large, the LAM may significantly deviate from the actual system, which may lead to instability. Let

$$D(x,u) = \frac{\|f(x,u)\|}{L_f}\log(\max\{\frac{\rho L(x,u)}{\|f(x,u)\|},\delta\}+1) \tag{3.14}$$

where δ is an arbitrarily small positive constant and ρ is a positive constant to be determined. Such a choice of the threshold can avoid Zeno behavior since $\frac{D(x,u)}{\|f(x,u)\|} \geq \frac{\log\delta+1}{L_f}$ always holds and therefore $t_k^{i+1} - t_k^i \geq \frac{\log\delta+1}{L_f}$ according to (4.12)

Assumption 2. *Assume the following inequalities holds:*

$$\beta(\|x\|) \leq \frac{L(x,u)}{\|f(x,u)\|},\ \gamma(\|x\|) \leq D(x,u),\ V_f(x) \leq \alpha(\|x\|) \tag{3.15}$$

$$V_f(x + D(x,h(x))\frac{f(x,h(x))}{\|f(x,h(x))\|}) - V_f(x) \leq -L(x,h(x))\frac{D(x,h(x))}{\|f(x,h(x))\|} \tag{3.16}$$

where $L(x,u)$ is the stage cost function; $\beta,\gamma,\alpha : \mathbb{R} \to \mathbb{R}$ are class $\mathcal{K}_\infty$ functions. Besides, we assume there exist the following Lipschitz continuous conditions

$$\begin{aligned} |V_f(x) - V_f(y)| \leq & L_{V_f}\|x-y\| \\ |L(x,u)\frac{D(x,u)}{\|f(x,u)\|} - L(y,u)\frac{D(y,u)}{\|f(y,u)\|}| \leq & L_c\|x-y\|. \end{aligned}$$

Note that the inequalities are common assumption conditions in MPC algorithms.

Theorem 2. *Suppose the hypotheses in Assumption 1, 2 along with Theorem 1 hold, if there exist the condition that*

$$\begin{aligned}
&\rho\frac{L(x,u)}{\|f(x,u)\|} \geq \theta(e^{L_f\frac{D(x,u)}{\|f(x,u)\|}}-1),\ \rho\in(0,1)\\
&\theta = L_c\frac{(L_s+1)^{N-1}-1}{L_s}+L_{V_f}(L_s+1)^{N-1}
\end{aligned} \tag{3.17}$$

$$\|\bar{x}(t_k)\| \geq \max\{\gamma^{-1}(\frac{2\rho\sigma}{1-\rho}),\beta^{-1}(\frac{\theta}{\rho})\} \tag{3.18}$$

the system (4.10) *under the MPC algorithm* (4.13) *is UUB.*

Proof. Let $J[\hat{u}_{k+1}|\bar{x}(t_{k+1})]$ be the cost of the FHOCP generated by $\hat{u}^i_{k+1}$ in (3.11) with initial condition $\bar{x}(t_{k+1})$. Consider

$$\begin{aligned}
&J[\hat{u}_{k+1}|\bar{x}(t_{k+1})]-V(\bar{x}(t_k)) \qquad (3.19)\\
=&\sum_{i=0}^{N-1}L(\hat{x}^i_{k+1},\hat{u}^i_{k+1})\frac{\hat{D}^i_{k+1}}{\|d\hat{x}^i_{k+1}\|}+V_f(\hat{x}^N_{k+1})-V(\bar{x}(t_k))\\
=&\sum_{i=0}^{N-2}L(\hat{x}^i_{k+1},\hat{u}^i_{k+1})\frac{\hat{D}^i_{k+1}}{\|d\hat{x}^i_{k+1}\|}+L(\hat{x}^{N-1}_{k+1},\hat{u}^{N-1}_{k+1})\frac{\hat{D}^{N-1}_{k+1}}{\|d\hat{x}^{N-1}_{k+1}\|}+V_f(\hat{x}^{N-1}_{k+1})-V_f(\hat{x}^{N-1}_{k+1})\\
&+V_f(\hat{x}^N_{k+1})+L(\hat{x}^{0,*}_k,\hat{x}^{0,*}_k)\frac{\hat{D}^{0,*}_k}{\|d\hat{x}^{0,*}_k\|}-L(\hat{x}^{0,*}_k,\hat{x}^{0,*}_k)\frac{\hat{D}^{0,*}_k}{\|d\hat{x}^{0,*}_k\|}-V(\bar{x}(t_k))\\
\leq&\{\sum_{i=0}^{N-2}L(\hat{x}^i_{k+1},\hat{u}^i_{k+1})\frac{\hat{D}^i_{k+1}}{\|d\hat{x}^i_{k+1}\|}+V_f(\hat{x}^{N-1}_{k+1})+L(\hat{x}^{0,*}_k,\hat{x}^{0,*}_k)\frac{\hat{D}^{0,*}_k}{\|d\hat{x}^{0,*}_k\|}-V(\bar{x}(t_k))\}\\
&+\{L(\hat{x}^{N-1}_{k+1},\hat{u}^{N-1}_{k+1})\frac{\hat{D}^{N-1}_{k+1}}{\|d\hat{x}^{N-1}_{k+1}\|}+V_f(\hat{x}^N_{k+1})-V_f(\hat{x}^{N-1}_{k+1})\}-L(\hat{x}^{0,*}_k,\hat{x}^{0,*}_k)\frac{\hat{D}^{0,*}_k}{\|d\hat{x}^{0,*}_k\|}
\end{aligned}$$

and we mark the first two parts of polynomial as Φ, Ψ. According to (3.16), we may get

$$\Psi = L(\hat{x}^{N-1}_{k+1},\hat{u}^{N-1}_{k+1})\frac{\hat{D}^{N-1}_{k+1}}{\|d\hat{x}^{N-1}_{k+1}\|}+V_f(\hat{x}^N_{k+1})-V_f(\hat{x}^{N-1}_{k+1})\leq 0, \tag{3.20}$$

since $\hat{u}^{N-1}_{k+1}=h(\hat{x}^{N-1}_{k+1})$ and $\hat{x}^N_{k+1}=\hat{x}^{N-1}_{k+1}+D(\hat{x}^{N-1}_{k+1},\hat{u}^{N-1}_{k+1})\frac{f(\hat{x}^{N-1}_{k+1},\hat{u}^{N-1}_{k+1})}{\|f(\hat{x}^{N-1}_{k+1},\hat{u}^{N-1}_{k+1})\|}$.

Let us consider about the sign of Φ. Note that the first term of Φ can be written as

$$\sum_{i=0}^{N-2}L(\hat{x}^i_{k+1},\hat{u}^i_{k+1})\frac{\hat{D}^i_{k+1}}{\|d\hat{x}^i_{k+1}\|}=\sum_{i=1}^{N-1}L(\hat{x}^{i-1}_{k+1},\hat{u}^{i-1}_{k+1})\frac{\hat{D}^{i-1}_{k+1}}{\|d\hat{x}^{i-1}_{k+1}\|}.$$

According to (4.13),

$$V(\bar{x}(t_k)) = \sum_{i=0}^{N-1} L(\hat{x}_k^{i,*}, \hat{u}_k^{i,*}) \frac{\hat{D}_k^{i,*}}{\|d\hat{x}_k^{i,*}\|} + V_f(\hat{x}_k^{N,*}).$$

And we can rewrite Φ as

$$\Psi = \sum_{i=1}^{N-1} L(\hat{x}_{k+1}^{i-1}, \hat{u}_{k+1}^{i-1}) \frac{\hat{D}_{k+1}^{i-1}}{\|d\hat{x}_{k+1}^{i-1}\|} - \sum_{i=1}^{N-1} L(\hat{x}_k^{i,*}, \hat{u}_k^{i,*}) \frac{\hat{D}_k^{i,*}}{\|d\hat{x}_k^{i,*}\|} + V_f(\hat{x}_{k+1}^{N-1}) - V_f(\hat{x}_k^{N,*}).$$

By the Lipschitz conditions of Assumption 2, along with Lemma 2, we can write the inequality as

$$\begin{aligned} \Psi &\leq \sum_{i=1}^{N-1} L_c \|\hat{x}_{k+1}^{i-1} - \hat{x}_k^{i,*}\| + L_{V_f} \|\hat{x}_{k+1}^{N-1} - \hat{x}_k^{N,*}\| \\ &\leq \sum_{i=1}^{N-1} L_c \epsilon_k (L_s + 1)^{i-1} + L_{V_f} \epsilon_k (L_s + 1)^{N-1} \\ &\triangleq \epsilon_k \theta \end{aligned} \tag{3.21}$$

Then we may represent (3.19) as

$$J[\hat{u}_{k+1}|\bar{x}(t_{k+1})] - V(\bar{x}(t_k)) \leq -L(\hat{x}_k^{0,*}, \hat{x}_k^{0,*}) \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|} + \epsilon_k \theta$$

with the conclusion of (3.20) and (3.21). Therefore, it may establish that

$$\begin{aligned} & V(\bar{x}(t_{k+1})) - V(\bar{x}(t_k)) \\ =& \min_{\hat{u}_{k+1}} J[\hat{u}_{k+1}|\bar{x}(t_{k+1})] - V(\bar{x}(t_k)) \\ \leq& - L(\hat{x}_k^{0,*}, \hat{u}_k^{0,*}) \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|} + \epsilon_k \theta \\ =& - L(\hat{x}_k^{0,*}, \hat{u}_k^{0,*}) \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|} + \hat{D}_k^{0,*} (e^{L_f \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}} - 1)\theta + \sigma (e^{L_f \frac{\hat{D}_k^{0,*}}{\|d\hat{x}_k^{0,*}\|}} - 1)\theta + \sigma\theta \\ \leq& - \frac{L(\hat{x}_k^{0,*}, \hat{u}_k^{0,*})}{\|f(\hat{x}_k^{0,*}, \hat{u}_k^{0,*})\|} (\hat{D}_k^{0,*} - \rho \hat{D}_k^{0,*} - \rho\sigma) + \sigma\theta \end{aligned}$$

with the definition of ϵ_k and (3.17). Besides, we may get

$$\hat{D}_k^{0,*} - \rho \hat{D}_k^{0,*} \geq 2\rho\sigma$$

by combining the conditions of (3.15) and (3.18). According to the definition of $\beta(\|x\|)$ in (3.15) and (3.18), the preceding inequality implies that

$$\begin{aligned} V(\bar{x}(t_{k+1})) - V(\bar{x}(t_k)) \leq & - \frac{L(\hat{x}_k^{0,*}, \hat{u}_k^{0,*})}{\|f(\hat{x}_k^{0,*}, \hat{u}_k^{0,*})\|}\rho\sigma + \sigma\theta \\ \leq & - \beta(\|\bar{x}(t_k)\|)\rho\sigma + \sigma\theta \\ \leq & 0. \end{aligned} \tag{3.22}$$

Meanwhile, by (3.16) in Assumption 2, we can get that

$$V_f(\hat{x}_k^{i+1}) - V_f(\hat{x}_k^i) \leq -L(\hat{x}_k^i, h(\hat{x}_k^i))\frac{D(\hat{x}_k^i, h(\hat{x}_k^i))}{\|f(\hat{x}_k^i, h(\hat{x}_k^i)\|}.$$

Summing up the inequality above for $i = 0, 1, \cdots, N-1$, then it may have

$$V_f(\hat{x}_k^N) - V_f(\hat{x}_k^0) \leq - \sum_{i=0}^{N-1} L(\hat{x}_k^i, h(\hat{x}_k^i))\frac{D(\hat{x}_k^i, h(\hat{x}_k^i))}{\|f(\hat{x}_k^i, h(\hat{x}_k^i)\|}.$$

According to the definition of $V(\bar{x}(t_k))$,

$$\begin{aligned} V(\bar{x}(t_k)) \leq & \sum_{i=0}^{N-1} L(\hat{x}_k^i, h(\hat{x}_k^i))\frac{D(\hat{x}_k^i, h(\hat{x}_k^i))}{\|f(\hat{x}_k^i, h(\hat{x}_k^i)\|} + V_f(\hat{x}_k^N) \\ \leq & V_f(\hat{x}_k^0) = V_f(\bar{x}(t_k)) \leq \alpha(\|\bar{x}(t_k)\|) \end{aligned}$$

together with (3.22) holds, it implies that $\bar{x}(t_k)$ will be uniformly ultimately bounded, which also leads to the conclusion that $x(t)$ is UUB. □

3.5 Simulation

This section shows how the LAMPC works on a nonlinear system with measurement noises. We consider the crane model in [54] with the excitation angle ϕ and the horizontal trolley position p:

$$\begin{aligned} \dot{p}(t) &= v(t) \\ \dot{v}(t) &= u(t) \\ \dot{\phi}(t) &= \omega(t) \\ \dot{\omega}(t) &= -gsin(\phi(t)) - u(t)cos(\phi(t)) - b\omega(t) \end{aligned}$$

where $x = (p, v, \phi, \omega)^T$ is the state and u is the control input. The control input must satisfy $-0.5 \leq u(t) \leq 0.5$. Besides, we use the parameters $b = 0.2J$ and $g = 9.81m/s^2$. The measurement noise $\omega(t_k)$ satisfies $\|\omega(t_k)\| \leq 0.05$.

The running cost function and the terminal cost function are defined as

$$L(x, u) = |f(x, u)| \cdot (|x| + |u| + 1), \ V_f(x) = 5|x|.$$

The threshold function $D(x, u)$ is defined by (3.14). The BARON solver [55] is used to solve the nonlinear FHOCP.

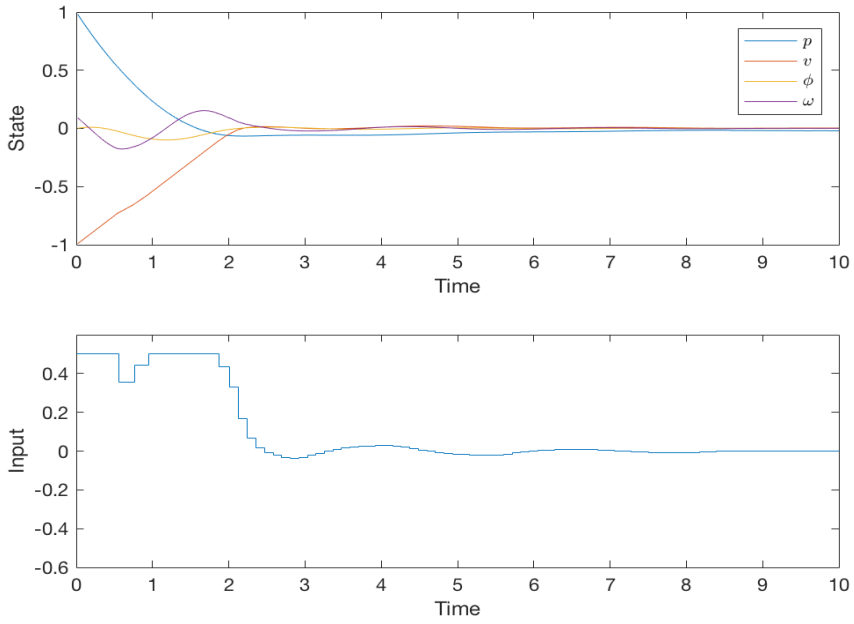

Figure 3.3: The state and input trajectory generated by LAMPC

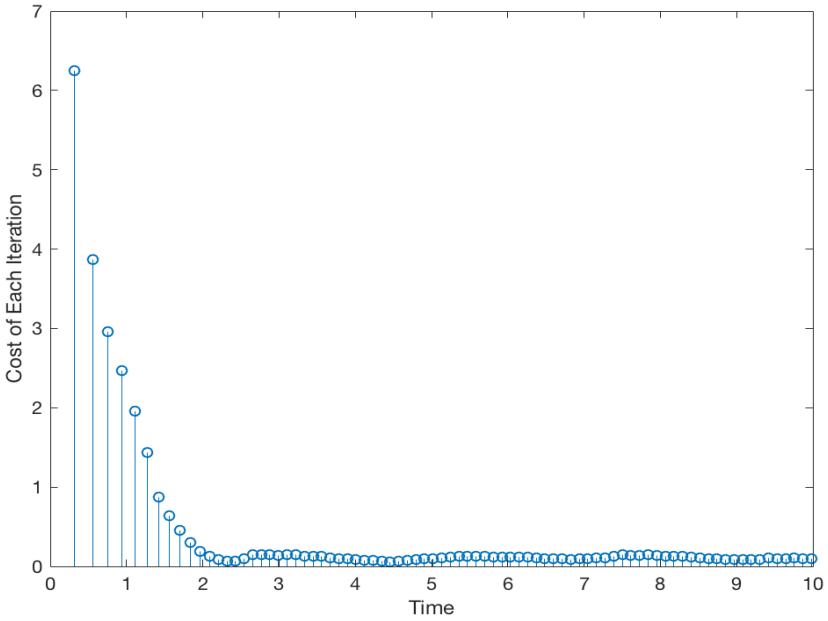

Figure 3.4: The history of $V(\bar{x}(t_k))$

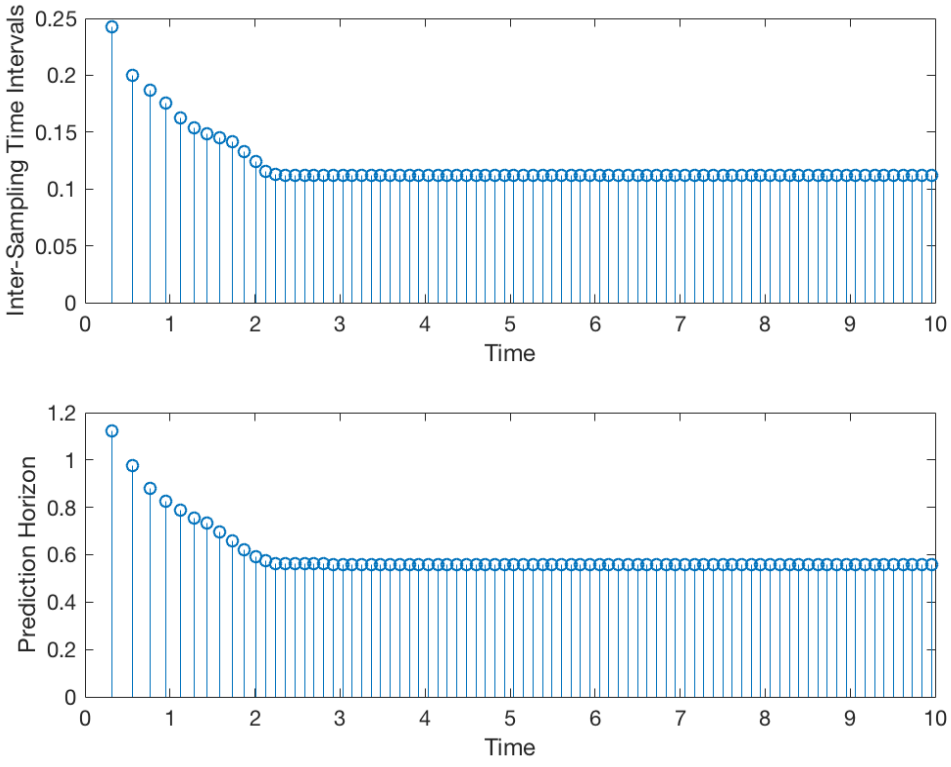

Figure 3.5: The inter-sampling time intervals and the prediction horizons

As it shows, Figure 3.3 plot the the state and input trajectories of the system. It is obvious that the system converges to a small neighborhood of the origin and the constraints are not violated. Figure 3.4 plots the trajectory of $V(\bar{x}(t_k))$. We can see that $V(\bar{x}(t_k))$ keeps decreasing until being close to zero. This is consistent to the theoretical results. And Figure 3.5 shows the history of the inter-sampling time intervals generated by the self-triggered scheme (top) and the length of prediction horizons at each sampling instants (bottom). It is clear that those intervals are time-varying until the state stays around the steady state. Also notice that they are strictly greater than zero.

3.6 Conclusion

This chapter presents a LAMPC algorithm for nonlinear continuous-time systems with measurement noises. A self-triggered method is used to trigger the sampling and computation and the LAM is designed to discretize the FHOCP. We show that with appropriate design of the threshold function in the LAM, even when the sampled state contains measurement noises, the LAMPC can still guarantee the system to be UUB. Meanwhile, this work can be applied to output-feedback systems as long as the observer is well designed such that the error between the estimated state and the actual state is uniformly bounded as stated in (3.2).

Chapter 4

Explicit Adaptive Model Predictive Control for Nonlinear Continuous-Time System with Uncertainty

As we discussed in Chapter 2, adaptive MPC (AMPC) may practically achieve stability in the presence of uncertainties with tight estimation and prediction. Nevertheless, the existing work focuses on continuous-time controllers where the finite horizon optimal control problem (FHOCP) is formulated in continuous-time. Solving such an FHOCP online will take a significant amount of computation resources. It will prevent applications of AMPC in systems with fast dynamics. Moreover, in practice, many controllers are implemented in digital environments, which require discrete-time control algorithms that can comprise intermittent sampling and task jitters in the system.

This chapter investigates an explicit adaptive model predictive control algorithm for nonlinear continuous-time systems. In this algorithm, we develop a discrete-time adaptive estimator to approximate the discretization error over the prediction horizon and the upper bound of the estimation error is derived. The finite-horizon optimal control problem (FHOCP) is formulated in discrete-time subject to the linearized system model and the state/input constraints. And multi-parametric programming is applied to find the explicit solution to the proposed FHOCP as a function based on current state and the estimated uncertainties. Given the resulting close-loop system with mixed continuous/discrete-time behaviors, we derive sufficient conditions on

feasibility of the proposed EAMPC and show that the system will be uniformly ultimately bounded under the proposed algorithm. Finally, we provide simulations to demonstrate the performance of the algorithms.

4.1 Basic of Explicit MPC

Consider a linear discrete-time model

$$x_{k+1} = Ax_k + Bu_k \tag{4.1}$$

where x_k is sampled from the plant, $\mathcal{U}, \mathcal{X}$ are the constraints to the input and state with the shape of polyhedral as

$$\begin{aligned} \mathcal{U} &= \{u \in \mathbb{R}^m | u_{min} \leq u_i \leq u_{max} \\ \mathcal{X} &= \{x \in \mathbb{R}^n | x_{min} \leq x_i \leq x_{max}\} \end{aligned} \tag{4.2}$$

where $u_{min}, u_{max}, x_{min}, x_{max}$ are scalars and u_i, x_i are the elements of the input and state, and $\mathcal{X}_f$ is the terminal set that the final state stays inside. Meanwhile, we define $J[u_k|x_k]$ as the system performance with

$$\begin{aligned} L(x_k, u_k) &= \frac{1}{2} x_k^T Q x_k + \frac{1}{2} u_k^T R u_k \\ F(x_N) &= \frac{1}{2} x_N^T P x_N \end{aligned}$$

as stage cost and terminal cost, and Q, R, P are semi-positive symmetric matrices.

We can always present the future state as

$$x_{k+i} = A^i x_k + \sum_{j=0}^{i-1} A^j B u_{k+i-1-j}.$$

By using it to substitute the states into (4.1) and (4.2), we can represent Problem 3 in the form as

$$\begin{aligned} V_z(x_k) &= \min_z \frac{1}{2} z^T H z \\ \text{s.t. } Gz &\leq W + Sx_k \end{aligned} \tag{4.3}$$

where $z \triangleq U + H^{-1}F^T x_k, U = \begin{bmatrix} u_k & u_{k+1} & \cdots & u_{k+N-1} \end{bmatrix}^T$ and H, G, W, S are the matrices of appropriate dimensions. The problem of the multi-parametric programming can be solved through Karush-Kuhn-Tucker (KKT) conditions as

$$\begin{aligned} &Hz + G^T\lambda = 0,\ \lambda \in \mathbb{R}^q; \\ &\lambda_i(G^i z - W^i - S^i x_k) = 0, \\ &\lambda_i \geq 0,\ j = 1, ..., q \\ &Gz - W - Sx_k \leq 0. \end{aligned}$$

And we can get the inequalities

$$\begin{aligned} GH^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x_k) \leq & W + Sx_k \\ (\tilde{G}H^{-1}\tilde{G}^T)^{-1}(\tilde{W} + \tilde{S}x_k) \leq & 0 \end{aligned} \tag{4.4}$$

for active constraints where $\tilde{S}, \tilde{G}, \tilde{W}$ are the matrices constructed by the rows in original matrices for $\lambda_j \neq 0$. Then we may get the critical region R_0 satisfying (4.4) and divide the constraints set into different regions $R_1 \cdots R_5$ based on the boundary of the R_0 as it shows in Figure 4.1. In this case, the optimization problem gets simplified and the optimal solutions and be presented by the central point of each polyhedron once x gets into corresponding area.

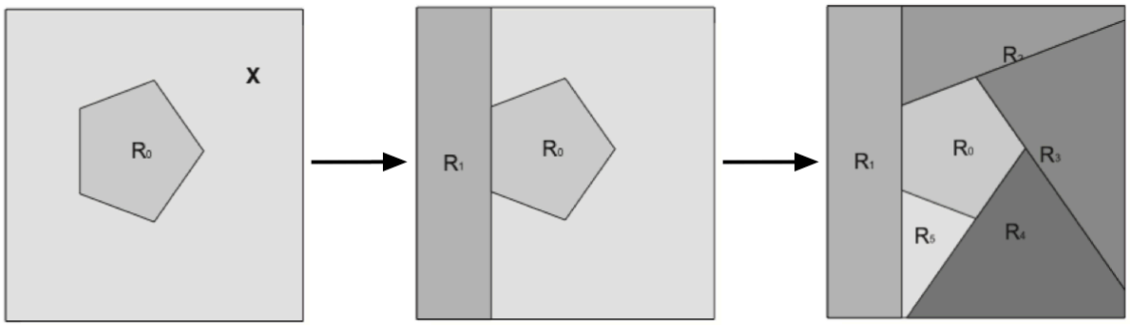

Figure 4.1: Division of Constraint Set $\mathcal{X}$

For the process of selecting active constraints on composing (4.4) and dividing corresponding regions can be pre-computed offline, we can get the full sequence of

optimized solutions by applying sampled state x_k during online computation and use the element u_k as the current input. In conclusion, explicit MPC does not need complex real-time computation compares with common MPC approaches.

4.2 Explicit MPC Scheme

In this section, we will provide the approach for solving the optimal problem as Problem 3. We can always get x_{k+i} in terms of $x(t_k)$ and $U = [u_k, ..., u_{N-1}]^T$ so that the summing function can be represented as

$$J[U|x(t_k)] = \sum_{i=0}^{N-1} (x_{k+i}^T Q x_{k+i}^T + u_{k+i}^T R u_{k+i}^T) + x_{k+N}^T P x_{k+N} - x_k^T Q x_k$$
$$= \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix}^T \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & P \end{bmatrix} \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix} + \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix}^T \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & R \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix}.$$

Meanwhile, we can present the states vector as

$$\begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x_k + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix}.$$

Let us make $\bar{A} = [A, ..., A^N]^T$ and present the matrices as

$$\bar{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & P \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & R \end{bmatrix}.$$

And we can get that

$$\begin{aligned} J[U|x(t_k)] =&[\bar{A}x_k + \bar{B}U]^T\bar{C}[\bar{A}x_k + \bar{B}U] + U^T\bar{D}U \\ =&x_k^T\bar{A}^T\bar{C}\bar{A}x_k + 2x_k^T\bar{A}^T\bar{C}\bar{B}U + U^T\bar{B}^T\bar{C}\bar{B}U + U^T\bar{D}U \end{aligned}$$

so that we may determine that $H = 2\bar{D}, F = 2\bar{A}\bar{C}\bar{B}$. If we have boundary condition as

$$x_{k+i} \leq v,\ i = 0, 1, \cdots, n$$

where v is the upper bound of the element, then it can be presented as

$$\bar{A}x_k + \bar{B}U \leq V, \text{ where } V = \begin{bmatrix} v & v & \cdots & v \end{bmatrix}^T_{1\times Nn}$$

which makes the shape of constrain set $\mathcal{X}$ as a convex polyhedral. We can represent Problem 3 as

$$\begin{aligned} \min_U\ J[U|x(t_k)] =&x_k^T\bar{A}^T\bar{C}\bar{A}x_k + 2x_k^T\bar{A}^T\bar{C}\bar{B}U + U^T\bar{B}^TG\bar{B}U + U^T\bar{D}U \\ \bar{B}U \leq&V - \bar{A}x_k \end{aligned}$$

and we can set $G = \bar{B}, S = \bar{D}^{-1}\bar{A}\bar{C}\bar{B} - \bar{A}$. Then the parameters of (4.3) get determined, and we may solve the problem via using multi-parametric programming as previous section.

4.3 Problem Formulation

Notations: We denote by $\mathbb{R}^n$ the n-dimensional real vector space, by $\mathbb{R}^+$ the set of real positive numbers, by $\mathbb{R}_0^+$ the set of real non-negative numbers, and by $\mathbb{Z}_0^+$ the set of non-negative integers. We use $\|\cdot\|$ to denote the Euclidean norm of a vector and the induced 2-norm of a matrix. Given a positive constant d, let $\mathcal{B}(d) = \{x \in \mathbb{R}^n \mid \|x\| \leq d\}$. Given two sets $\mathcal{X}, \mathcal{S} \subseteq \mathbb{R}^n$, $\mathcal{X} + \mathcal{S}$ is the Minkowski sum of these two sets.

Definition 1. *A continuous function $\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.*

Definition 2. *The state $x(t)$ of a system $\dot{x} = f(x)$ is called uniformly ultimately bounded (UUB) with ultimate bound b if there exist positive constants b and c, independent of $t_0 \geq 0$, and for every $a \in (0, c)$, there is $T = T(a, b) \geq 0$, independent of t_0 , such that $\|x(t_0)\| \leq a$ implies $\|x(t)\| \leq b$ for any $t \geq t_0 + T$.*

Consider a nonlinear multi-input-multi-output (MIMO) state-feedback system:

$$\dot{x} = f(x) + Bu(t),\ x(0) = x_0 \tag{4.5}$$

where $x : \mathbb{R}_0^+ \to \mathcal{X}$ is the system state, $u : \mathbb{R}_0^+ \to \mathcal{U}$ is the control input, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a known nonlinear continuously differentiable function. The compact sets $\mathcal{X} \subset \mathbb{R}_n^+, \mathcal{U} \subset \mathbb{R}_m^+$ describe the state constraint and the input constraint, respectively, i.e.,

$$x(t) \in \mathcal{X},\ u(t) \in \mathcal{U}$$

must hold for any $t \geq 0$. Besides, we assume $f(x)$ is locally Lipschitz that satisfies

$$\|f(x) - f(y)\| \leq l_x \|x - y\| \tag{4.6}$$

for any $x, y \in \mathcal{X}$, where $l_x \in \mathbb{R}^+$. Then we can get that

$$\|f(x)\| \leq l_x \max_{x \in \mathcal{X}} \|x\| = f_{\max}. \tag{4.7}$$

Let T to be the sampling period. So the state $x(t_k)$ is sampled at $t_k = kT$. Let us denote $x_k = x(t_k)$ for simplicity.

When the sampled state x_k is obtained, a discrete-time N-step FHOCP will be solved. And the FHOCP with $\bar{x}_k^0 = x_k$ under EAMPC framework is formulated as

follows:

$$\textbf{Problem 3:} \quad \min_{\bar{u}_k^i \in \mathcal{U}} \quad J[\bar{u}_k | x_k, \sigma_k]$$
$$\text{s.t.} \quad \bar{x}_k^{i+1} = \bar{x}_k^i + BT\bar{u}_k^i + T\sigma_k, \ i = 0, 1, \cdots, N-1$$
$$\bar{x}_k^i \in \bar{\mathcal{X}}_i, \ \bar{x}_k^N \in \bar{\mathcal{X}}_N = \mathcal{X}_f \tag{4.8}$$

where

$$J[\bar{u}_k | x_k, \sigma_k] = \sum_{i=0}^{N-1} (\bar{x}_k^{iT} Q \bar{x}_k^i + \bar{u}_k^{iT} R \bar{u}_k^i) + \bar{x}_k^{NT} P \bar{x}_k^N \tag{4.9}$$

$\bar{x}_k^i, \bar{u}_k^i$ are the predicted state and input generated at $t_{k+i} = (k+i)T$ at the kth computation cycle. In this formulation, Q, R, P are semi-positive symmetric matrices, σ_k is a parameter that estimates the discretization error at t_k, and $\bar{\mathcal{X}}_i$ is the constraint set used in FHOCP of the predicted state. The optimal solution $\bar{u}_k^{0,*}$ will be applied to the plant as the input during $t \in [t_k, t_{k+1})$, i.e.,

$$u(t) = \bar{u}_k^{0,*}, \quad \forall t \in [t_k, t_{k+1}).$$

We may present the system dynamic of (4.8) as

$$\bar{y}_k^i = A^i \bar{y}_k^0 + \sum_{j=0}^{i-1} A^j E \bar{u}_k^{i-1-j}, \ \bar{y}_k^i = [\bar{x}_k^i; T\sigma_k]$$

where $A = [I_n \ I_n; 0 \ I_n], E = [BT; \ 0]$. And we are going to use the approach of explicit MPC to solve the optimize problem as previous section during each prediction iteration.

Finally, the next computation cycle starts at t_{k+1} by solving the same problem but a different initial state $\bar{x}_{k+1}^0 = x_{k+1}$ and a different parameter σ_{k+1} in order to get a close-loop FHOCP. The control objective is stabilize the continuous-time system under the framework (4.8).

4.4 Adaptive Estimator

In this section, we design the discrete-time adaptive estimator. To begin with, let us take integral on both sides of equation (5.11) over $[t_k, t_{k+1})$

$$\int_{t_k}^{t_{k+1}} \dot{x}\ dt = \int_{t_k}^{t_{k+1}} (f(x) + Bu(t))\, dt$$

which provides the discretized system

$$x_{k+1} = x_k + BTu_k + \int_{t_k}^{t_{k+1}} f(x)dt \tag{4.10}$$

where $u_k = u(t_k)$ as $u(t)$ is a constant over $[t_k, t_{k+1})$.

Let us consider the adaptive estimator

$$\hat{x}_{k+1} = \Gamma\epsilon_k + x_k + BTu_k + T\sigma_k,\ \hat{x}_0 = x_0 \tag{4.11}$$

where $\hat{x}_k$ is the state of the estimator, $\epsilon_k = \hat{x}_k - x_k$, Γ is an arbitrarily chosen positive constant over $(0, 1)$, σ_k is estimated by the following adaptive law

$$\sigma_k = -\frac{\Gamma}{T}\epsilon_k + f(x_k). \tag{4.12}$$

Note that the estimator runs independent of the optimizer.

Lemma 1. *Consider the system* (4.10) *and the estimator* (4.11)*. The inequality*

$$\begin{aligned} \|f(x) - \sigma_k\| \leq & l_x(f_{\max} + \max_{u\in\mathcal{U}} \|Bu\|)(t - t_k + \Gamma T) \\ \triangleq & \phi(t - t_k) \end{aligned} \tag{4.13}$$

holds for any $t \in [t_k, t_{k+N})$*, where* $f_{\max}$ *is defined in* (5.12)*.*

Proof. By (4.10) and (4.11), the error ϵ_{k+1} satisfies

$$\begin{aligned} \epsilon_{k+1} = & Tf(x_k) - \int_{t_k}^{t_{k+1}} f(x)dt \\ = & Tf(x_k) - Tf(x(\tau_k^*)) \end{aligned}$$

for $\tau_k^* \in [t_k, t_{k+1})$ by applying the first mean value theorem. Equation (4.12) also implies that

$$\sigma_{k+1} = -\frac{\Gamma}{T}\epsilon_{k+1} + f(x_{k+1}).$$

Therefore, it exists that

$$\sigma_{k+1} = \Gamma f(x(\tau_k^*)) - \Gamma f(x_k) + f(x_{k+1})$$

for all $k \in \mathbb{Z}_0^+$. Consider the difference between $f(x)$ and σ_k for $t \in [t_k, t_{k+N})$:

$$\begin{aligned}
&\|f(x) - \sigma_k\| \\
\leq &\|f(x) - f(x_k)\| + \|f(x_k) - \sigma_k\| \\
\leq &\|f(x) - f(x_k)\| + \left\|-\Gamma f(x(\tau_{k-1}^*)) + \Gamma f(x_{k-1}))\right\| \\
\leq &l_x\|x(t) - x_k\| + l_x\Gamma\|x(\tau_{k-1}^*) - x_{k-1}\| \\
\leq &l_x(f_{\max} + \max_{u\in\mathcal{U}}\|Bu\|)(t - t_k + \Gamma T) \triangleq \phi(t - t_k)
\end{aligned}$$

and the proof gets finished. □

Remark 1. *The bound ϕ can be arbitrarily close to 0, if T is small enough and Γ is close to 1. It implies that when the sampling rate is high enough, the estimation error can be very small. Besides, σ_k is always converges to $f(x)$ as the system reaches stability.*

Lemma 1 suggests that we can use the constant σ_k to approximate $f(x)$ over the prediction horizon $[t_k, t_{k+N})$ and formulate the discrete-time dynamics constraint by represent the system in (5.11) as follows:

$$\bar{x}_k^{i+1} = \bar{x}_k^i + BT\bar{u}_k^i + T\sigma_k$$

where $x_k(t_{k+i}) = \bar{x}_k^i$. It shows the routine of EAMPC in Figure 5.3 with corresponding description as Table 5.2.

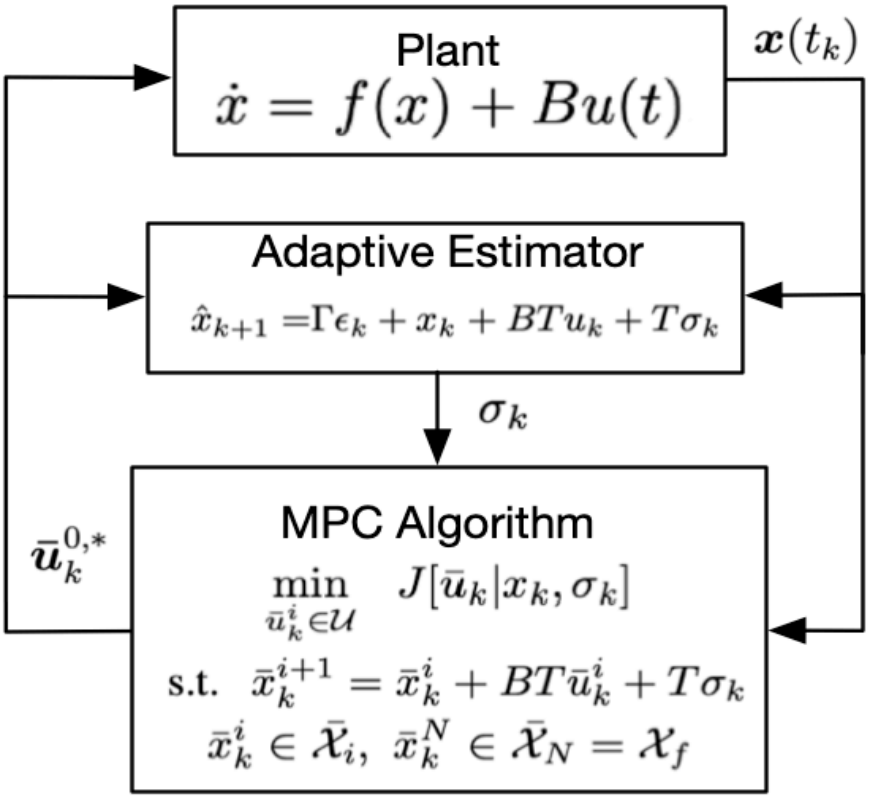

Figure 4.2: The Procedure of EAMPC

Table 4.1: EAMPC Algorithm Routine

1. At t_k, run the estimator defined in (4.11) and compute σ_k by (4.12);
2. Solve the FHOCP in (4.8) with the parameters x_k and σ_k;
3. Apply $\bar{u}_k^{0,*}$ as the input to the plant for $t \in [t_k, t_{k+1})$;
4. Repeat Step 1-3 until the state researches the destination.

4.5 Feasibility Analysis

This section discusses the feasibility of EAMPC. To begin with, we are going to discuss the condition which may ensure $x(t) \in \mathcal{X}$ under the constrains of (4.8). The basic idea is making $\mathcal{X}_1$ to be a reduced set of $\mathcal{X}$ based on the difference between $x(t_{k+1})$ and $\bar{x}_k^1$, so that $\bar{x}_k^1 \in \mathcal{X}_1$ implies $x(t) \in \mathcal{X}$ for $t \in [t_k, t_{k+1}]$. Let us construct a

continuous-time system over $t \in [t_k, t_{k+1}]$ as

$$\dot{z}_k = B\bar{u}_k^{0,*} + \sigma_k = Bu_k + \sigma_k,\ z_k(t_k) = x(t_k) = \bar{x}_k^0 \tag{4.14}$$

according to (4.8), so that

$$z_k(t) = z_k(t_k) + (Bu_k + \sigma_k)(t - t_k) \tag{4.15}$$

and $z_k(t_{k+1}) = \bar{x}_k^1$. Consider the error between $z_k(t)$ and $x(t)$ over the time interval $t \in [t_k, t_{k+1}]$ with same control input $\bar{u}_k^{0,*}$ and initial state $x(t_k)$, the following Lemma holds.

Lemma 2. *Consider the systems* (5.11) *and* (4.14)*, the following inequality holds*

$$\|x(t) - z_k(t)\| \leq T\phi(T) \tag{4.16}$$

for any $t \in [t_k, t_{k+1}]$.

Proof. Comparing the systems (5.11) and (4.14), we may get the dynamic of error $e(t) = x(t) - z_k(t)$ as

$$\dot{e} = f(x) - \sigma_k$$

which may lead to the differential inequality

$$\frac{d}{dt}\|e(t)\| \leq \|f(x) - \sigma_k\|.$$

Then we may solve the inequality as

$$\|e(t)\| \leq T\phi(T)$$

for any $t \in [t_k, t_{k+1}]$. □

With (4.16) holds, we may define the constraint set $\mathcal{X}_1$ and ensure $x(t) \in \mathcal{X}$ over the prediction horizon by the following theorem.

Theorem 1. *Assume the hypotheses in Lemma 2 holds, if* $x(t_0) \in \mathcal{X} - \mathcal{B}(T\phi(T))$ *and set*

$$\mathcal{X}_1 \triangleq \mathcal{X} - 2\mathcal{B}(T\phi(T)) \tag{4.17}$$

then $x \in \mathcal{X}$ *for any* $t \geq t_0$ *under the model of* (4.8).

Proof. We may present $z_k(t)$ as

$$z_k(t) = (1-\theta)z_k(t_k) + \theta z_k(t_{k+1}),\ \theta = \frac{t - t_k}{t_{k+1} - t_k}$$

based on (4.15) for any $t \in [t_k, t_{k+1}]$ so that $\theta \in [0, 1]$. Let us discuss the statement in a recursive way. Firstly, at the moment of t_0, we have

$$z_0(t_0) = x(t_0) = \bar{x}_0^0 \in \mathcal{X} - \mathcal{B}(T\phi(T))$$

with the assumption holds, and we may also get

$$z_0(t_1) = \bar{x}_0^1 \in \mathcal{X}_1 = \mathcal{X} - 2\mathcal{B}(T\phi(T)).$$

In this case, we may find that

$$z_0(t) \in \mathcal{X} - \mathcal{B}(T\phi(T)),\ t \in [t_0, t_1]$$

as $\mathcal{X} - \mathcal{B}(T\phi(T))$ is a compact set for both of $z_0(t_0), z_0(t_1)$. According to (4.16), it may imply that $x(t) \in \mathcal{X}$ for any $t \in [t_0, t_1]$. Secondly, for the situation of t_1, we may get that

$$z_1(t_1) = x(t_1) = \bar{x}_1^0 \in \mathcal{X}_1 + \mathcal{B}(T\phi(T)) = \mathcal{X} - \mathcal{B}(T\phi(T))$$

with (4.16) holds. Together with the definition

$$z_1(t_2) = \bar{x}_1^1 \in \mathcal{X}_1 = \mathcal{X} - 2\mathcal{B}(T\phi(T))$$

we may get that

$$z_1(t) \in \mathcal{X} - \mathcal{B}(T\phi(T)),\ t \in [t_1, t_2]$$

and $x(t) \in \mathcal{X}$ for any $t \in [t_1, t_2]$ as similar as previous iteration. And the proof gets completed by keep the processing along the prediction horizon with the conclusion that $x(t) \in \mathcal{X}$ for any $t \in [t_k, t_{k+1}]$. □

Let us construct a control sequence for (4.8) at the $(k+1)$st FHOCP with the initial condition $\bar{x}_{k+1}^0 = x_{k+1}$ as

$$\bar{u}_{k+1}^i = \begin{cases} \bar{u}_k^{i+1,*}, \ i = 0, 1, \cdots, N-2 \\ h(\bar{x}_{k+1}^i), \ i = N-1 \end{cases} \tag{4.18}$$

where $\bar{u}_k^{i+1,*}$ donates the optimal solution of the kth FHOCP, $\bar{x}_{k+1}^i$ is the predicted state generated by $\bar{u}_{k+1}^i$ at the $(k+1)$st computation cycle, and $h(x) : \mathbb{R}^n \to \mathbb{R}^m$ is a feedback law that will be defined later. We will consider $\bar{u}_k^{i+1,*}, h(\bar{x}_{k+1}^{N-1})$ as one possible solution to the $(k+1)$st FHOCP.

According to (4.10) and (4.11), it yields that

$$\begin{aligned} \|\epsilon_{k+1} - \epsilon_k\| \leq & \Big\| Tf(x_k) - Tf(x_{k-1}) - \int_{t_k}^{t_{k+1}} f(x)dt \\ & + \int_{t_{k-1}}^{t_k} f(x)dt \Big\| \\ \leq & Tl_x \|x_k - x_{k-1}\| + T\|f(x(\tau_k^*)) - f(x(\tau_{k-1}^*))\| \\ \leq & T^2(l_x+1)(f_{\max} + \max_{u \in \mathcal{U}} \|Bu\|) \end{aligned}$$

Given the Lipschitz condition on $f(x)$, there exist the inequality that

$$\begin{aligned} \|\sigma_{k+1} - \sigma_k\| &\leq \|f(x_{k+1} - f(x_k))\| + \frac{\Gamma}{T}\|\epsilon_{k+1} - \epsilon_k\| \\ &\leq l_x \|x_{k+1} - x_k\| + \frac{\Gamma}{T}\|\epsilon_{k+1} - \epsilon_k\| \\ &\leq T(\Gamma l_x + \Gamma + l_x)(f_{\max} + \max_{u \in \mathcal{U}} \|Bu\|) \\ &\triangleq \Lambda(T). \end{aligned} \tag{4.19}$$

Lemma 3. *There exists the relationship between states of neighboring prediction routine as the following inequality*

$$\|\bar{x}_{k+1}^i - \bar{x}_k^{i+1,*}\| \leq \gamma_i \triangleq T\phi(T) + iT\Lambda(T). \tag{4.20}$$

holds for $i = 0, 1, \cdots, N-1$.

Proof. We prove the statement using mathematical induction. According to Lemma 2, it is obvious that (4.20) holds for $i = 0$ as

$$\begin{aligned}\|\bar{x}_{k+1}^0 - \bar{x}_k^{1,*}\| =& \|x(t_{k+1}) - \bar{x}_k^{1,*}\| \\ \leq& T\phi(T).\end{aligned}$$

Assuming (4.20) holds for $i = p-1$, i.e.,

$$\|\bar{x}_{k+1}^{p-1} - \bar{x}_k^{p,*}\| \leq \gamma_{p-1} = T\phi(T) + (p-1)\Lambda(T).$$

Notice that

$$\begin{aligned}\bar{x}_k^{p+1,*} =& \bar{x}_k^{p,*} + BT\bar{u}_k^{p,*} + T\sigma_k \\ \bar{x}_{k+1}^{p} =& \bar{x}_{k+1}^{p-1} + BT\bar{u}_k^{p,*} + T\sigma_{k+1}.\end{aligned}$$

Therefore, by the inequality (4.19), we have

$$\begin{aligned}\|\bar{x}_{k+1}^{p} - \bar{x}_k^{p+1,*}\| \leq& \|\bar{x}_{k+1}^{p-1} - \bar{x}_k^{p,*}\| + T\Lambda(T) \\ \leq& T\phi(T) + pT\Lambda(T) \\ =& \gamma_p\end{aligned}$$

which means that (4.20) holds for $i = p$. Besides, we may get $\lim_{T\to 0,\Gamma\to 1}\gamma_i = 0$ as $\lim_{T\to 0}\Lambda(T) = 0$ and $\lim_{T\to 0,\Gamma\to 1}\phi(T) = 0$. □

With Theorem 1 and Lemma 3 holds, we may construct the constraint set for $i = 1, 2, \cdots, N-1$ as:

$$\bar{\mathcal{X}}_i \triangleq \mathcal{X} - \mathcal{B}(\varepsilon_i),\ \varepsilon_i = (i+1)T\phi(T) + \sum_{p=0}^{i-1} pT\Lambda(T). \tag{4.21}$$

Assumption 1. *It satisfies the following conditions that:*

$$h(0) = 0,\ 0 \in \mathit{interior}(\bar{\mathcal{X}}_N),\ \bar{\mathcal{X}}_N + \mathcal{B}(\gamma_{N-1}) \subset \mathcal{X}_U \triangleq \{x \in \bar{\mathcal{X}}_{N-1} \mid h(x) \in \mathcal{U}\} \tag{4.22}$$

$$x \in \bar{\mathcal{X}}_N + \mathcal{B}(\gamma_{N-1}) \Rightarrow x + BTh(x) + Tf(x) \in \bar{\mathcal{X}}_N - \mathcal{B}(T\phi(NT)) \tag{4.23}$$

where $\phi(NT)$ *is defined in* (4.13).

Theorem 2. *If Assumption 1 hold and the FHOCP in* (4.8) *is feasible at* t_0*, then the FHOCP is always feasible for* $k = 0, 1, \cdots$ *under the constraint sets defined in* (4.21).

Proof. We will prove that if $\bar{x}_k^{i,*} \in \bar{\mathcal{X}}_i$ and $u_k^{i,*} \in \mathcal{U}$, then $\bar{x}_{k+1}^i \in \bar{\mathcal{X}}_i$ and $\bar{u}_{k+1}^i \in \mathcal{U}$, which means that they are feasible to the FHOCP at the $(k+1)$st computation.

To begin with, let us show that $\bar{u}_{k+1}^i \in \mathcal{U}$ for $i = 0, 1, \cdots, N-1$. Based on (4.18), we know that $\bar{u}_{k+1}^i = \bar{u}_k^{i+1,*} \in \mathcal{U}, i = 0, 1, \cdots, N-2$. For $i = N-1$, we know by Lemma 3 that

$$|\|\bar{x}_{k+1}^{N-1}\| - \|\bar{x}_k^{N,*}\|| \leq \|\bar{x}_{k+1}^{N-1} - \bar{x}_k^{N,*}\| \leq \gamma_{N-1}.$$

Since $\bar{x}_k^{N,*} \in \bar{\mathcal{X}}_N$, with (4.22), it implies that

$$\bar{x}_{k+1}^{N-1} \in \bar{\mathcal{X}}_N + \mathcal{B}(\gamma_{N-1}) \subset \mathcal{X}_U$$

and therefore $\bar{u}_{k+1}^{N-1} = h(\bar{x}_{k+1}^{N-1}) \in \mathcal{U}$.

Secondly, we will show that $\bar{x}_{k+1}^i \in \bar{\mathcal{X}}_i, i = 1, \cdots, N-1$. By Lemma 3, we know

$$|\|\bar{x}_{k+1}^i\| - \|\bar{x}_k^{i+1,*}\|| \leq \|\bar{x}_{k+1}^i - \bar{x}_k^{i+1,*}\| \leq \gamma_i.$$

Also, we have

$$\bar{x}_k^{i+1,*} \in \bar{\mathcal{X}}_{i+1} \quad \text{and} \quad \bar{x}_k^{N,*} \in \bar{\mathcal{X}}_N \subset \bar{\mathcal{X}}_{N-1}.$$

Therefore,

$$\bar{x}_{k+1}^i \in \bar{\mathcal{X}}_{i+1} + \mathcal{B}(\gamma_i) = \bar{\mathcal{X}}_i, \ i = 1, \cdots, N-1.$$

for $i = 1, \cdots, N-1$.

Finally, we show that $\bar{x}_{k+1}^N \in \bar{\mathcal{X}}_N$. Since

$$\bar{x}_{k+1}^N = \bar{x}_{k+1}^{N-1} + BTh(\bar{x}_{k+1}^{N-1}) + T\sigma_{k+1}$$

there exists the error $T\phi(NT)$ between σ_{k+1} and $f(x)$ according to (4.13). With (4.23) holds, we may ensure $\bar{x}_{k+1}^N \in \bar{\mathcal{X}}_N$ and the proof gets finished. □

4.6 Stability

It discusses the stability of the system under EAMPC algorithm in this section.

Assumption 2. *There exist the positive constants* $l_{V_f}, l_L, d \in \mathbb{R}^+, \rho \in (0,1)$*, and class* $\mathcal{K}_\infty$ *functions* $\alpha, \eta : \mathbb{R} \to \mathbb{R}$*, such that*

(i) for any $x, y \in \mathcal{X}$ *and* $u \in \mathcal{U}$

$$\begin{aligned} |V_f(x) - V_f(y)| \leq & l_{V_f}\|x-y\| \\ |L(x,u) - L(y,u)| \leq & l_L\|x-y\| \\ L(x,u) \geq \eta(\|x\|),\ & V_f(x) \leq \alpha(\|x\|) \end{aligned} \tag{4.24}$$

(ii) for any $x \in \bar{\mathcal{X}}_N + \mathcal{B}(\gamma_{N-1})$

$$V_f(x + BTh(x) + Tf(x)) - V_f(x) \leq -L(x, h(x)) + d \tag{4.25}$$

Theorem 3. *If Assumption 1–2 hold and the FHOCP in* (4.8) *is feasible, the system is uniformly ultimately bounded.*

Proof. Let us consider

$$\begin{aligned} & J[\bar{u}|x_{k+1}, \sigma_{k+1}] - V(x_k) \\ = & \sum_{i=0}^{N-1} L(\bar{x}_{k+1}^i, \bar{u}_{k+1}^i) + V_f(\bar{x}_{k+1}^N) - V(x_k) \\ = & \sum_{i=0}^{N-2} L(\bar{x}_{k+1}^i, \bar{u}_{k+1}^i) + V_f(\bar{x}_{k+1}^{N-1}) + L(\bar{x}_k^{0,*}, \bar{u}_k^{0,*}) \\ & - V(x_k) + L(\bar{x}_{k+1}^{N-1}, \bar{u}_{k+1}^{N-1}) + V_f(\bar{x}_{k+1}^N) \\ & - V_f(\bar{x}_{k+1}^{N-1}) - L(\bar{x}_k^{0,*}, \bar{u}_k^{0,*}). \end{aligned} \tag{4.26}$$

Meanwhile, there exists

$$\begin{aligned} & V_f(\bar{x}_{k+1}^{N-1} + BTh(\bar{x}_{k+1}^{N-1}) + Tf(x)) - V_f(\bar{x}_{k+1}^{N-1}) \\ \leq & - L(\bar{x}_{k+1}^{N-1}, h(\bar{x}_{k+1}^{N-1})) + d \end{aligned}$$

with (4.25) holds. As we have

$$\begin{aligned}
&|V_f(\bar{x}_{k+1}^{N-1} + BTh(\bar{x}_{k+1}^{N-1}) + Tf(x)) - V_f(\bar{x}_{k+1}^{N})| \\
=&l_{V_f}\|\bar{x}_{k+1}^{N-1} + BTh(\bar{x}_{k+1}^{N-1}) + Tf(x) - \bar{x}_{k+1}^{N}\| \\
\leq&l_{V_f}T\phi(NT)
\end{aligned}$$

according to (4.13). Then we may get

$$\begin{aligned}
&V_f(\bar{x}_{k+1}^{N}) - V_f(\bar{x}_{k+1}^{N-1}) + L(\bar{x}_{k+1}^{N-1}, h(\bar{x}_{k+1}^{N-1})) \\
\leq&d + l_{V_f}T\phi(NT).
\end{aligned} \tag{4.27}$$

Also, notice that

$$V(x_k) = \sum_{i=0}^{N-2} L(\bar{x}_k^{i+1,*}, \bar{u}_k^{i+1,*}) + L(\bar{x}_k^{0,*}, \bar{u}_k^{0,*}) + V_f(\bar{x}_k^{N,*})$$

then we may consider the first four terms in(4.26) as

$$\begin{aligned}
&\sum_{i=0}^{N-2} L(\bar{x}_{k+1}^{i}, \bar{u}_{k+1}^{i}) - \sum_{i=0}^{N-2} L(\bar{x}_k^{i+1,*}, \bar{u}_k^{i+1,*}) \\
&+ V_f(\bar{x}_{k+1}^{N-1}) - V_f(\bar{x}_k^{N,*}) \\
\leq& \sum_{i=0}^{N-2} |L(\bar{x}_{k+1}^{i}, \bar{u}_{k+1}^{i}) - L(\bar{x}_k^{i+1,*}, \bar{u}_k^{i+1,*})| \\
&+ |V_f(\bar{x}_{k+1}^{N-1}) - V_f(\bar{x}_k^{N,*})| \\
\leq& \sum_{i=0}^{N-2} l_L\|\bar{x}_{k+1}^{i} - \bar{x}_k^{i+1,*}\| + l_{V_f}\|\bar{x}_{k+1}^{N-1} - \bar{x}_k^{N,*}\| \\
\leq& \sum_{i=0}^{N-2} l_L\gamma_i + l_{V_f}\gamma_{N-1} = l_L\varepsilon_{N-1} + l_{V_f}\gamma_{N-1}
\end{aligned} \tag{4.28}$$

where the last inequality comes from Lemma 3. Applying (4.27) and (4.28) into (4.26) yields

$$\begin{aligned}
&J[\bar{u}|x_{k+1}, \sigma_{k+1}] - V(x_k) \\
\leq&\underbrace{l_L\varepsilon_{N-1} + l_{V_f}\gamma_{N-1} + d + l_{V_f}T\phi(NT)}_{\zeta} - L(x_k, u_k).
\end{aligned}$$

Then we can further derive this inequality as follows:

$$\begin{aligned} V(x_{k+1}) - V(x_k) =& \min_{\bar{u}} J[\bar{u}|x_{k+1}, \sigma_{k+1}] - V(x_k) \\ \leq& \zeta - L(x_k, u_k) \\ \leq& \zeta - \eta(\|x_k\|) \end{aligned}$$

which implies x_k gets uniformly ultimately bounded. As $V(x(t)) \leq V(x_k)$ for any $t \in [t_k, t_{k+1})$, which may conclude that $x(t)$ is also uniformly ultimately bounded. □

4.7 Simulation

This section presents simulation results that demonstrate the performance of EAMPC algorithm. Consider the example as

$$\begin{cases} \dot{x}_1 =& 0.1x_1^2 - 0.25x_2 + 2u_1 \\ \dot{x}_2 =& 0.5x_1 - 0.1x_2^3 + u_2 \end{cases}$$

where $x = [x_1\ x_2]^T \in \mathcal{X} \triangleq \{x \in \mathbb{R}^2 |\ \|x\|_\infty \leq 5\}, u = [u_1\ u_2]^T \in \mathcal{U} \triangleq \{u \in \mathbb{R}^2 |\ \|u\|_\infty \leq 5\}$. Note that we use uniform norm to present distance of vectors in order to match the conditions on solving multi-parametric programming problem. We set the step length $T = 0.05$, the prediction steps to be 4 in each iteration, and the parameter of discretize estimator $\Gamma = 0.04$. And we solve the problem via multi-parameter quadratic programming (MPQP) by using toolbox [56].

To begin with, let us check whether Assumption 1, 2 may get satisfied as common conditions in MPC formulation. We can get the value of $l_x = 2.5, f_{max} = 12.5$. Then we may present (4.13) as

$$\phi(t - t_k) = 56.25(t - t_k + 0.002).$$

Furthermore, there exists the coefficients that

$$\|\sigma_{k+1} - \sigma_k\| \leq \Lambda(T) = 59.4T = 2.97.$$

Besides, we may always let

$$h(x) = -B^{-1}(x/T + f(x))$$

in order to guarantee that

$$x + BTh(x) + Tf(x) = 0 \in \bar{\mathcal{X}}_N - \mathcal{B}(NT\phi)$$

for (4.23). And we may determine the feasible sets as

$$\begin{aligned}
\bar{\mathcal{X}}_0 =& \mathcal{X} = \{\|x\|_\infty \leq 5\} \\
\bar{\mathcal{X}}_1 =& \mathcal{X} - \mathcal{B}(0.15) = \{\|x\|_\infty \leq 4.85\} \\
\bar{\mathcal{X}}_2 =& \mathcal{X} - \mathcal{B}(0.44) = \{\|x\|_\infty \leq 4.56\} \\
\bar{\mathcal{X}}_3 =& \mathcal{X} - \mathcal{B}(0.88) = \{\|x\|_\infty \leq 4.12\} \\
\bar{\mathcal{X}}_N =& \mathcal{X} - \mathcal{B}(1.48) = \{\|x\|_\infty \leq 3.52\}
\end{aligned}$$

Then the conditions of Assumption 1 get satisfied and we may present the corresponding relationship of each set as the following figure. To simplify the computation, we are going to use $\bar{\mathcal{X}}_N$ as the uniform boundary along the prediction horizon.

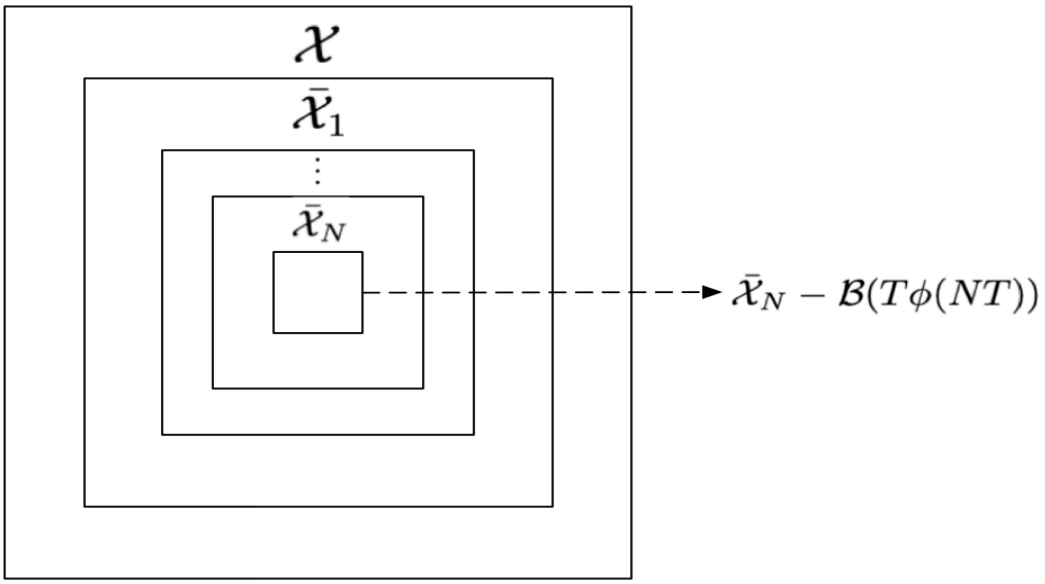

Figure 4.3: Relationship between each Constraint Sets

Let us construct the functions of stage cost and terminal cost as

$$L(x,u) = \frac{1}{4}x^T x + \frac{1}{4}u^T u$$
$$V_f(x) = \frac{1}{4}x^T x$$

then we may get $l_{V_f} = 2.5,\ l_L = 5$, and the $\mathcal{K}_\infty$ functions can be defined as

$$\eta(\|x\|) = \frac{1}{4}x^T x,\ \alpha(\|x\|) = x^T x.$$

And we may find that

$$V_f(x + BTh(x) + Tf(x)) - V_f(x) \leq -L(x, h(x)).$$

In this case, the conditions of Assumption 2 get guaranteed.

Let us run the example under EAMPC and common online MPC to compare the performance of each algorithm. According to Figure 4.5, the system of online MPC get stabilized more quickly and the optimization get stopped around $t = 5$. However, the trajectory of EAMPC in Figure 5.4 runs more gentle and smoother compare with online MPC, which means the control of EAMPC may prevent the system states with sudden change.

As it shows in the following table, the computation of EAMPC takes 5.78 seconds with total cost of 363.96 for 25 prediction length, while it spends 52.99 seconds with total cost of 69.04 for online MPC under same computation steps. The computation time of online MPC is much longer than EAMPC although the total cost of online MPC is low, which lead to a rough trajectory of online MPC. In conclusion, we may find that EAMPC owns a better performance than common online MPC approaches on dealing with nonlinear FHOCP.

Table 4.2: Comparison of EAMPC and online MPC

	Running Time	Total Cost	Total Length
EAMPC	5.78s	363.96	25
online MPC	52.99s	69.04	

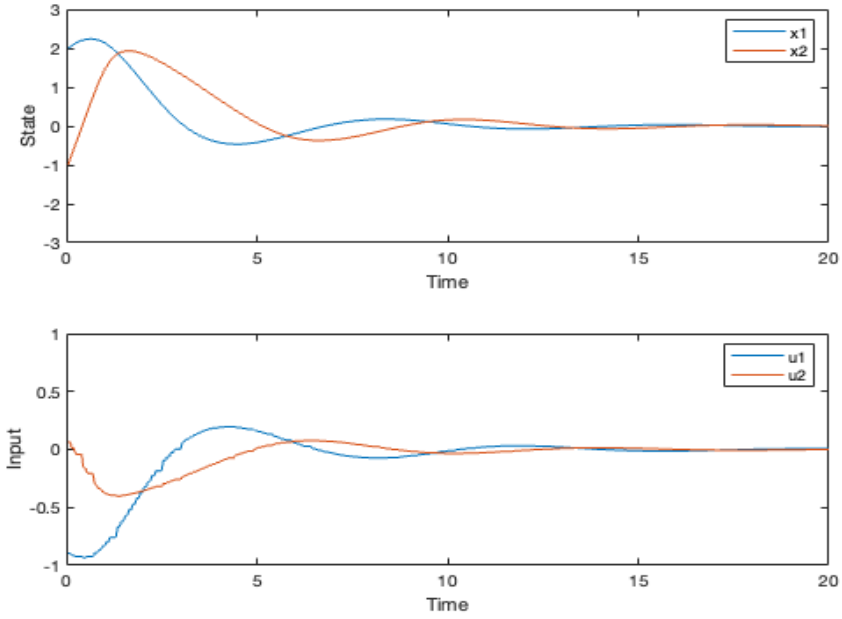

Figure 4.4: The State and Input Trajectory Generated by EAMPC

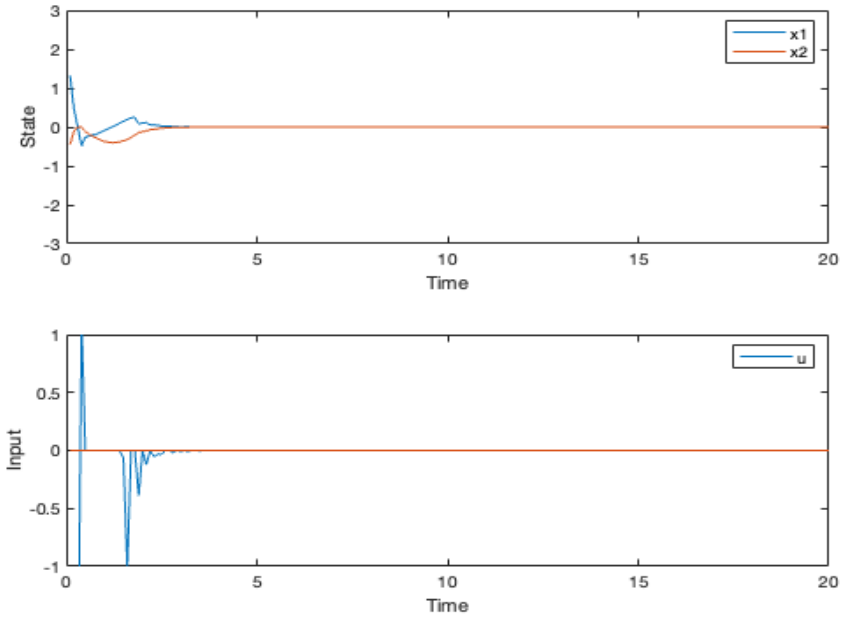

Figure 4.5: The State and Input Trajectory Generated by online MPC

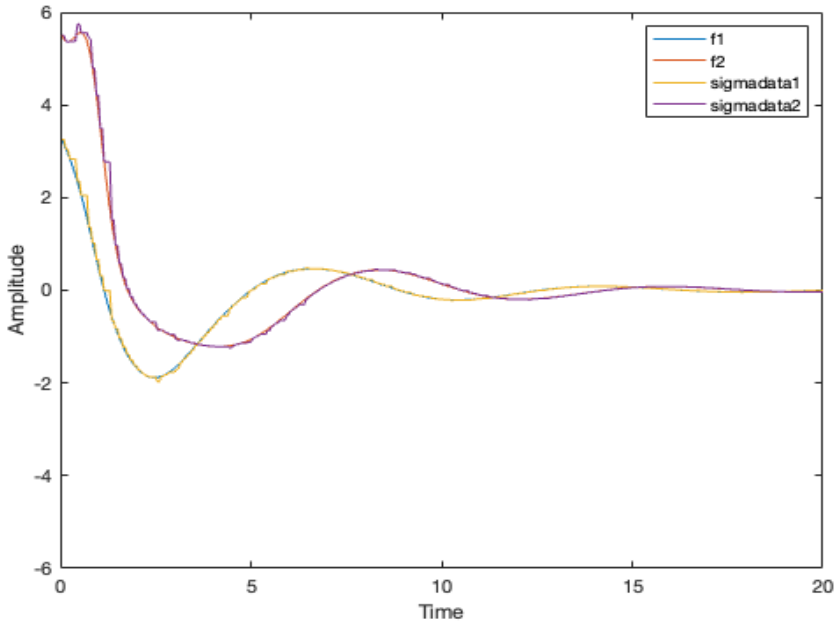

Figure 4.6: The Trajectory of $f(x)$ v.s σ_k

Finally, it shows the approximation of $f(x)$ with σ_k via the discretize estimator in Figure 5.7. We may find that the trajectory of σ_k always follows $f(x)$. And it works well along the prediction horizon.

4.8 Conclusion

This chapter presents an adaptive explicit MPC algorithm for nonlinear continuous-time system with uncertainty. It solves the optimization problems by linearizing and discretizing the original model. Besides, it provides a method to approximate the error generated by remodeling the original system adaptively. Based on these works, it may improve the performance of the system and save the computation cost. Finally, we will discuss the impact of linearization on control performance and corresponding feasibility for future work.

Chapter 5

Explicit Adaptive Model Predictive Control on robotic manipulators

Robotics is a complex field with diverse areas such as physics, statics, dynamics, electronics, control theory, and computer programming, to name a few. In our research, we are interested in the control of robotic manipulators. In this chapter, we study the approach to building the mathematic model of robotic manipulators and operating the system using the algorithm of explicit adaptive model predictive control. We review the knowledge of Lagrange's equation, which is a common approach to denote robot dynamics. Besides, we introduce a fundamental geometric structure of robot manipulators, such as manipulator configuration, kinematic, and inverse kinematic based on the content of [57]. And we are going to provide two examples of the approach to derive robotic manipulators under explicit adaptive model predictive control algorithm.

5.1 Derivation of Manipulator Dynamics

Lagrange's equation aims to present the motion systems in the form as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \tag{5.1}$$

where q is defined as joint-variable vector, τ is the vector of generalized forces, and L is the difference between the kinetic energy K and potential energy P as

$$L = K - P. \tag{5.2}$$

Typically, q consists of the elements as variable angle θ_i and/or variable length d_i of each joint while τ contains torque n_i and/or force f_i of each joint.

In general, the dynamics of robots are always been presented in the form as

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau \tag{5.3}$$

with the name as Lagrange's mechanics, where $M(q)$ is defined as inertia matrix, $V(q,\dot{q})$ is Coriolis/centripetal vector, and $G(q)$ is gravity vector. It is necessary to determine the kinetic energy K and potential energy P, along with Lagrange's equation, in order to get the general form (5.3).

To begin with, let's discuss the kinetic energy of a point on the ith link in a general manipulator model. We can determine the position of the point based on the kinematics as

$$r = T_i {}^i r, \ {}^i r = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T_{4\times 1}$$

where T_i is a 4×4 homogeneous transformation matrix contains the scalar joint variables $q_1, q_2, \cdots, q_i$, ${}^i r$ is the position of the point in the reference coordinates of the ith link. According to the knowledge of kinematics, we can get the velocity of the point in the base coordinates as

$$v = \frac{dr}{dt} = \frac{d(T_i {}^i r)}{dt} = \sum_{j=1}^{i} \frac{\partial T_i}{\partial q_j} \dot{q}_j {}^i r$$

and we can replace the upper summation limit i by n which is the total number of links of in the manipulator as it is obvious that

$$\frac{\partial T_i}{\partial q_j} = 0, \ j > i.$$

Then we can determine the kinetic energy of the point with infinitesimal mass dm and velocity of v as

$$\begin{aligned} dK_i =& \frac{1}{2} v^T v dm =,\ v = \begin{bmatrix} v_x & v_y & v_z & 0 \end{bmatrix}^T \\ =& \frac{1}{2} \text{trace} \left[v v^T \right] dm \\ =& \frac{1}{2} \text{trace} \left[\frac{dT_i}{dt} {}^i r \, {}^i r^T [\frac{dT_i}{dt}]^T \right] \\ =& \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \text{trace} \left[\frac{\partial T_i}{\partial q_j} {}^i r \, {}^i r^T [\frac{\partial T_i}{\partial q_k}]^T \right] \dot{q}_j \dot{q}_k dm \end{aligned}$$

Thus the total kinetic energy of link i can be given as

$$K_i = \int dK_i = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \text{trace} \left[\frac{\partial T_i}{\partial q_j} I_i [\frac{\partial T_i}{\partial q_k}]^T \right] \dot{q}_j \dot{q}_k,\ I_i = \int {}^i r \, {}^i r^T dm$$

and the detail of pseudo-inertia matrix I_i is provided in [57]. As it for the total kinetic energy of the manipulator, we can present it as

$$\begin{aligned} K =& \sum_{i=1}^{n} K_i \\ =& \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \text{trace} \left[\frac{\partial T_i}{\partial q_j} I_i [\frac{\partial T_i}{\partial q_k}]^T \right] \dot{q}_j \dot{q}_k \qquad (5.4) \\ =& \frac{1}{2} \dot{q}^T M(q) \dot{q} \qquad (5.5) \end{aligned}$$

where we make

$$\dot{q} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \cdots & \dot{q}_n \end{bmatrix}^T_{1 \times n}$$

and $M(q)$ as a $n \times n$ symmetric inertia matrix with the elements of

$$m_{jk}(q) = \sum_{i=\max j,k}^{n} \text{trace} \left[\frac{\partial T_i}{\partial q_j} I_i [\frac{\partial T_i}{\partial q_k}]^T \right],\ \frac{\partial T_i}{\partial q_j} = 0,\ j > i.$$

With (5.5), it provides a convenient expression for kinetic energy in terms of known quantities and the joint variable vector q. Besides, $M(q)$ should be symmetric and positive definite.

For the next step, let's determine the potential energy of a general manipulator. For the link i with the mass m_i at the center of gravity

$$^{i}\bar{r} = \begin{bmatrix} \bar{x} & \bar{y} & \bar{z} & 1 \end{bmatrix}^T$$

in the coordinates of its frame i, the potential energy of the link can be given by

$$P_i = -m_i g^T T_i {}^{i}\bar{r},\ g = \begin{bmatrix} g_x & g_y & g_z & 0 \end{bmatrix}^T$$

where g is gravity vector according to the base coordinates. For the most of cases, we can express

$$g = \begin{bmatrix} 0 & 0 & -9.81 & 0 \end{bmatrix}^T$$

while the robot is operated on the earth. Therefore, we can get the total potential energy as

$$\begin{aligned} P(q) &= -\sum_{i=1}^{n} m_i g^T T_i {}^{i}\bar{r} \\ &= -\sum_{i=1}^{n} g^T T_i I_i e_4,\ e_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T. \end{aligned} \tag{5.6}$$

According to (5.2),(5.5), and (5.6), we can determine

$$\begin{aligned} L(q,\dot{q}) =& K(q,\dot{q}) - P(q) \\ =& \frac{1}{2}\dot{q}^T M(q)\dot{q} - P(q) \end{aligned} \tag{5.7}$$

With (5.7) hold, we can also get the components of (5.1) as

$$\begin{aligned} \frac{\partial L}{\partial \dot{q}} =& \frac{\partial K}{\partial \dot{q}} = M(q)\dot{q} \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} =& M(q)\ddot{q} + \dot{M}(q)\dot{q} \\ \frac{\partial L}{\partial q} =& \frac{\partial K(q,\dot{q})}{\partial q} - \frac{\partial P(q)}{\partial q} \end{aligned} \tag{5.8}$$

Finally, we can express the Lagrange's mechanics as

$$\begin{aligned} M(q)\ddot{q} + V(q,\dot{q}) + G(q) =& \tau \\ V(q,\dot{q}) = \dot{M}(q)\dot{q} - \frac{\partial K(q,\dot{q})}{\partial q},\ G(q) =& \frac{\partial P(q)}{\partial q} \end{aligned} \tag{5.9}$$

by substituting (5.8) into (5.1), which is the general form of manipulator dynamics (5.3).

As it for the dynamic function in state space, we may represent (5.3) as Newton-Euler equation

$$\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -M^{-1}(q)(V(q,\dot{q}) + G(q)) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(q) \end{bmatrix} u(t),\ u(t) = \tau \tag{5.10}$$

where τ is treated as input $u(t)$, joint-variable vector q and its derivative $\dot{q}$ is defined as the element of system state.

We provide an example for the approach on constructing the Lagrange's equation. Consider a two-link planar elbow arm with the knowledge of joint variable vector and generalized force vector

$$q = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T,\ \tau = \begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^T$$

as it shows in the figure.

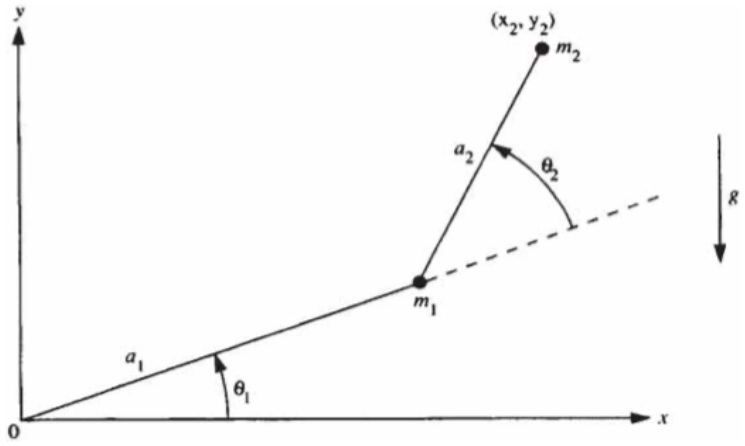

Figure 5.1: Two-link Planar Elbow Arm

To begin with, we may build the kinematic matrices as

$$T_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & -\sin(\theta_1)\sin(\theta_2) & 0 & a_1\cos(\theta_1) + a_2\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_1)\sin(\theta_2) & \cos(\theta_1)\cos(\theta_2) & 0 & a_1\sin(\theta_1) + a_2\sin(\theta_1)\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assuming the mass of each link is concentrated at its ends then we may get

$$I_i = \begin{bmatrix} a_i^2 m_i^2 & 0 & 0 & a_i m_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_i m_i & 0 & 0 & m_i \end{bmatrix}, \ i = 1, 2.$$

According to (5.4) (5.6), we can determine the kinetic and potential energy as

$$\begin{aligned} K =& \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} \text{trace}\left[\frac{\partial T_i}{\partial q_j} I_i [\frac{\partial T_i}{\partial q_k}]^T\right] \dot{q}_j \dot{q}_k \\ =& \frac{1}{2} m_1 a_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 a_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 a_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 a_1 a_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2)\cos(\theta_2), \\ P =& -\sum_{i=1}^{n} g_i^T I_i T_i e_4, \ e_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \\ =& m_1 g a_1 \sin(\theta_1) + m_2 g a_1 \sin(\theta_1) + m_2 g a_2 \sin(\theta_1 + \theta_2). \end{aligned}$$

Then we can get the normal form (5.3) according to (5.8) as

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau$$

$$M(q) = \begin{bmatrix} (m_1+m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2\cos(\theta_2) & m_2a_2^2 + m_2a_1a_2\cos(\theta_2) \\ m_2a_2^2 + m_2a_1a_2\cos(\theta_2) & m_2a_2^2 \end{bmatrix}$$

$$V(q,\dot{q}) = \begin{bmatrix} -m_2a_1a_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2)\sin(\theta_2) \\ m_2a_1a_2\dot{\theta}_1^2\sin(\theta_2) \end{bmatrix}$$

$$G(q) = \begin{bmatrix} (m_1+m_2)ga_1\cos(\theta_1) + m_2ga_2\cos(\theta_1+\theta_2) \\ m_2ga_2\cos(\theta_1+\theta_2) \end{bmatrix}$$

and the example gets finished.

5.2 Deriving An Manipulator Dynamic Model Using EAMPC

To begin with, let us consider an manipulator model as:

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) = \tau$$

where $\theta = [\theta_1 \ \theta_2]^T$ is the vector of angles, $\tau = [\tau_1 \ \tau_2]^T$ is torque input, $M(\theta)$ is defined as inertia matrix, $V(\theta,\dot{\theta})$ is Coriolis/centripetal vector, and $G(\theta)$ is gravity vector. We are going to represent the Lagrange's mechanics (5.10) as Hamiltonian formulation in order to linearize the input, so that the dynamic structure of EAMPC gets satisfied. Let us define generalized momentum by

$$p = \frac{\partial L}{\partial \dot{\theta}} = M(\theta)\dot{\theta} \tag{5.11}$$

so that the kinetic energy (5.5) can be expressed as

$$\begin{aligned} K =& \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} \\ =& \frac{1}{2}p^T M^{-1}(\theta)p \\ =& \frac{1}{2}p^T\dot{\theta}. \end{aligned} \tag{5.12}$$

Let us define the manipulator Hamiltonian as

$$\begin{aligned} H =& K + P \\ =& 2K - (K - P) \\ =& p^T \dot{\theta} - L \end{aligned}$$

and the corresponding equations of motion should be

$$\frac{\partial H}{\partial p} = \dot{\theta} \tag{5.13}$$

and

$$\begin{aligned} \frac{\partial H}{\partial \theta} =& - \frac{\partial L}{\partial \theta} \\ =& - \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} + \tau \\ =& - \dot{p} + \tau \end{aligned} \tag{5.14}$$

according to (5.1) and (5.11). We can also represent (5.14) as

$$\begin{aligned} \dot{p} =& - \frac{\partial H}{\partial \theta} + \tau \\ =& - \frac{\partial (K + P)}{\partial \theta} + \tau \\ =& - \frac{1}{2}\frac{\partial}{\partial \theta}(p^T M(q) p) - \frac{\partial P}{\partial \theta} + \tau \\ =& - \frac{1}{2}(I_n \bigotimes p^T)\frac{\partial M^{-1}(\theta) p}{\partial \theta} - G(\theta) + \tau \end{aligned} \tag{5.15}$$

based on (5.9) and (5.12), where $G(\theta)$ is the gravity vector and $\bigotimes$ is the Kronecker product. Combining (5.13) (5.16), we may get that

$$\frac{d}{dt}\begin{bmatrix} \theta \\ p \end{bmatrix} = \begin{bmatrix} M^{-1}(\theta) p \\ -\frac{1}{2}(I_n \bigotimes p^T)\frac{\partial M^{-1}(\theta) p}{\partial \theta} - G(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ I_n \end{bmatrix} \tau \tag{5.16}$$

so that we obtain the system with linearize form of input.

For specification, we have

$$V(\theta,\dot{\theta}) = \begin{bmatrix} -\dot{\theta}_2 m_2 L_1 l_{m2} \sin\theta_2 & -(\dot{\theta}_2+\dot{\theta}_1) m_2 L_1 l_{m2} \sin\theta 2 \\ \dot{\theta}_1 m_2 L_1 l_{m1} \sin\theta 2 & 0 \end{bmatrix},$$

$$G(\theta) = \begin{bmatrix} g m_1 l_{m1} \sin\theta_1 + g m_2 L_1 \sin\theta_1 + g m_2 l_{m2} \sin(\theta_1+\theta_2) \\ g m_2 l_{m2} \sin(\theta_1+\theta_2) \end{bmatrix}$$

$$M(\theta) = \begin{bmatrix} m_1 l_{m1}^2 + m_2(L_1^2 + l_{m2} + 2L_1 l_{m2}\cos\theta_2) + I_1 + I_2 & m_2(l_{m2}^2 + L_1 l_{m2}\cos\theta_2) + I_2 \\ m_2(l_{m2}^2 + L_1 l_{m2}\cos\theta_2) + I_2 & m_2 l_{m2}^2 + I_2 \end{bmatrix}$$

with the parameters show in the following table.

Table 5.1: Specification of Parameters

m_1	m_2	L_1	L_2	l_{m1}	l_{m2}	I_1	I_2
kg	kg	m	m	m	m	$kg \cdot m^2$	$kg \cdot m^2$
2.970	0.540	0.400	0.355	0.124	0.146	0.11175	0.00708

We use (5.16) to build the system dynamic, and our objective is deriving the manipulator via optimized path under explicit adaptive MPC algorithm. The simulation finished in 12 seconds with step length $T = 0.1$, $\Gamma = 0.1$, 2000 optimization iterations, and 40 prediction steps in each iteration.

As we can see from Figure 5.2, the initial state of the angles is $(1, -2)^T$, and it converges to the origin smoothly, as well as the trajectory of the input torque. Besides, the estimator shows a good performance on tracking $f(x)$ in Figure 5.3. It shows that the explicit adaptive MPC algorithm may have an excellent efficiency to implement the practical systems.

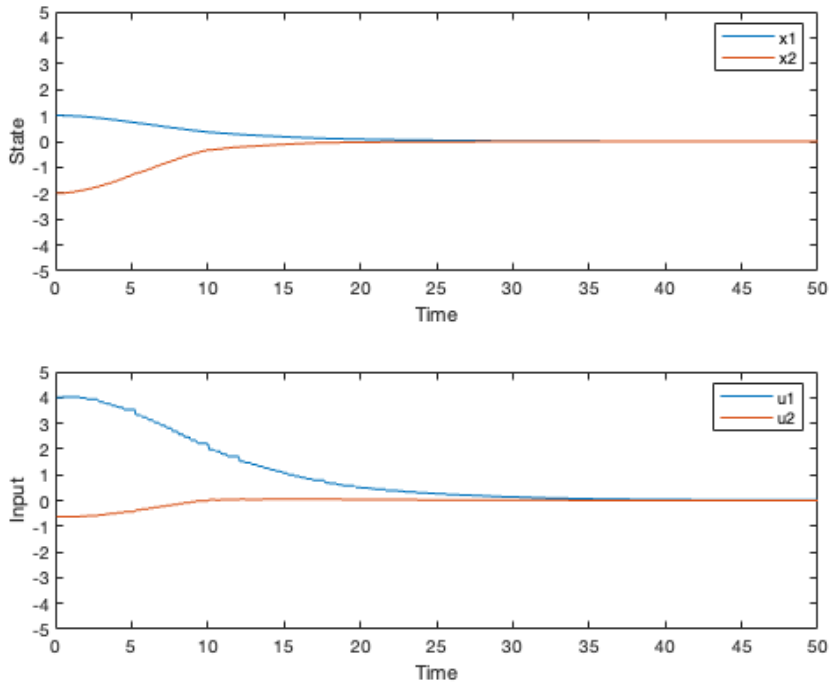

Figure 5.2: The Trajectory of Angles and Torque

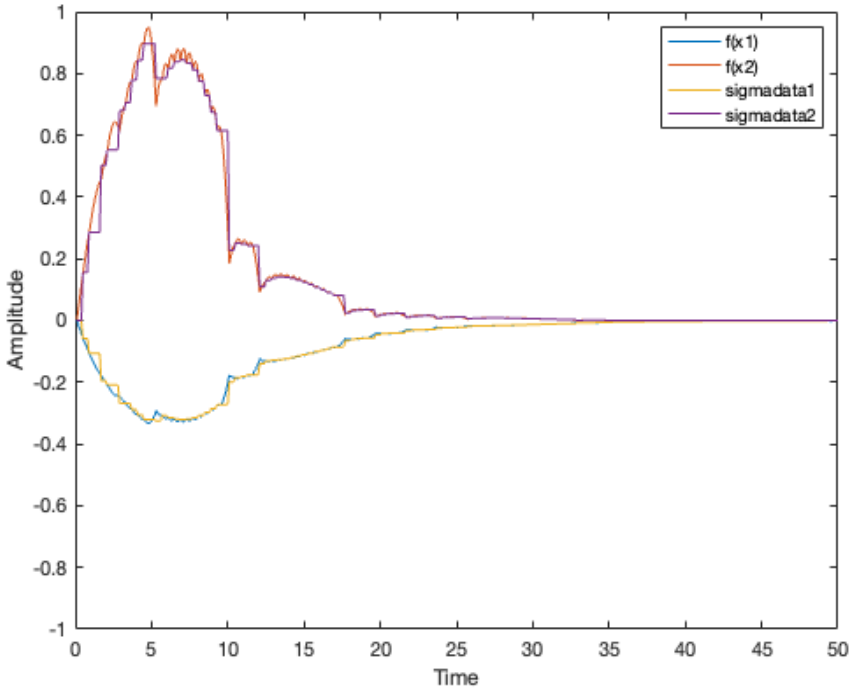

Figure 5.3: The Trajectory of $f(x)$ v.s σ_k

5.3 Deriving An Actual Manipulator Using EAMPC

In this section, we are going to operate an actual robotic manipulator under the EAMPC algorithm. The PincherX 100 Robot Arm is developed by Interbotix as it shows in Figure 5.6, which featuring the DYNAMIXEL X-Series Smart Servo Motors that may offer a high resolution of 4096 positions. And the system may provide the data such as temperature monitoring, positional feedback, voltage levels, load, and compliance settings. As the heart of the PincherX 100, Robotis DYNAMIXEL U2D2 in Figure 5.4 enables easy access to DYNAMIXEL Wizard software. Finally, PincherX 100 offers 4 degrees of freedom and a full 360 degree of rotation. It shows the technical specifications of PincherX 100 in Table 5.2 and the default joint limits of each servo motor in Table 5.3.

Table 5.2: Technical Specifications of PincherX 100

PincherX-100	
Degrees of Freedom	4
Reach	300mm
Total Span	600mm
Accuracy	5mm
Working Payload	50g
Total Servos	5
Wrist Rotate	No

Table 5.3: Default Joint Limits

Joint	Min	Max	Servo ID(s)
Waist	$-180°$	$180°$	1
Shoulder	$-111°$	$107°$	2
Elbow	$-121°$	$92°$	3
Wrist Angle	$-180°$	$180°$	4
Gripper	30mm	74mm	5

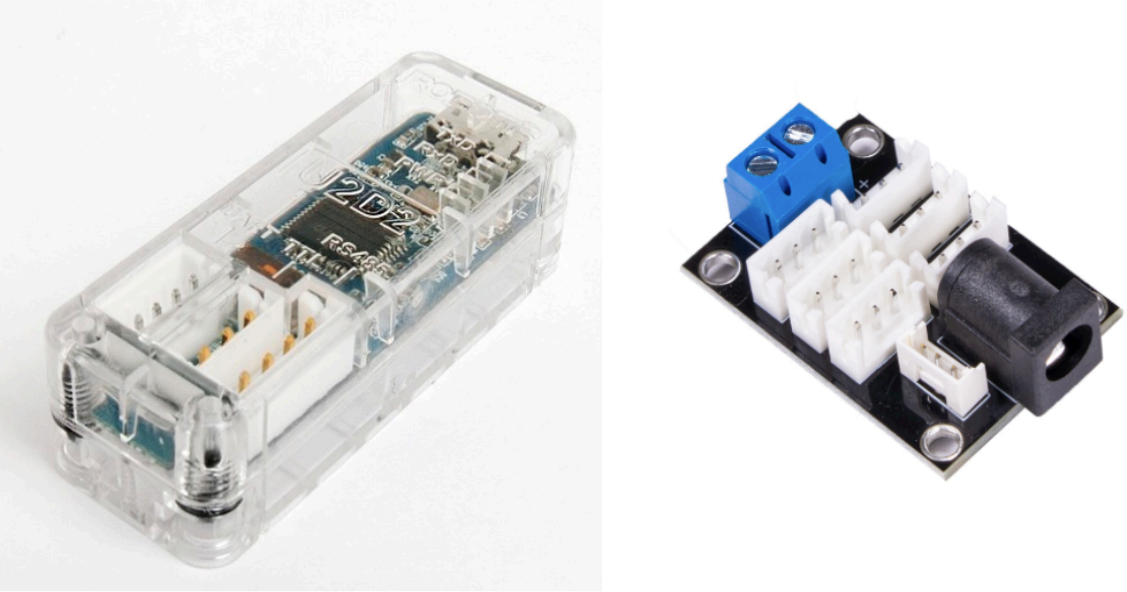

Figure 5.4: PincherX 100 U2D2 and Power Hub

The DYNAMIXEL servos support up to 360 degrees of rotation, making the PincherX 100 Robot Arm has a 30cm horizontal reach from the center of the base to the gripper with a total span of 60cm as it shows in Figure 5.5. Besides, an advantage of the X-Series manipulator is we can design the gripper fingers for different projects under X-Series gripper carriages. We may download the CAD files, design the structures, and make customization via 3D print.

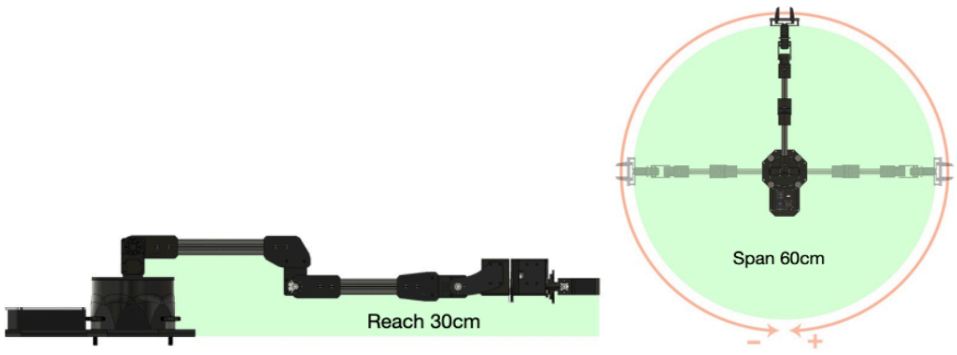

Figure 5.5: PincherX 100 Reach

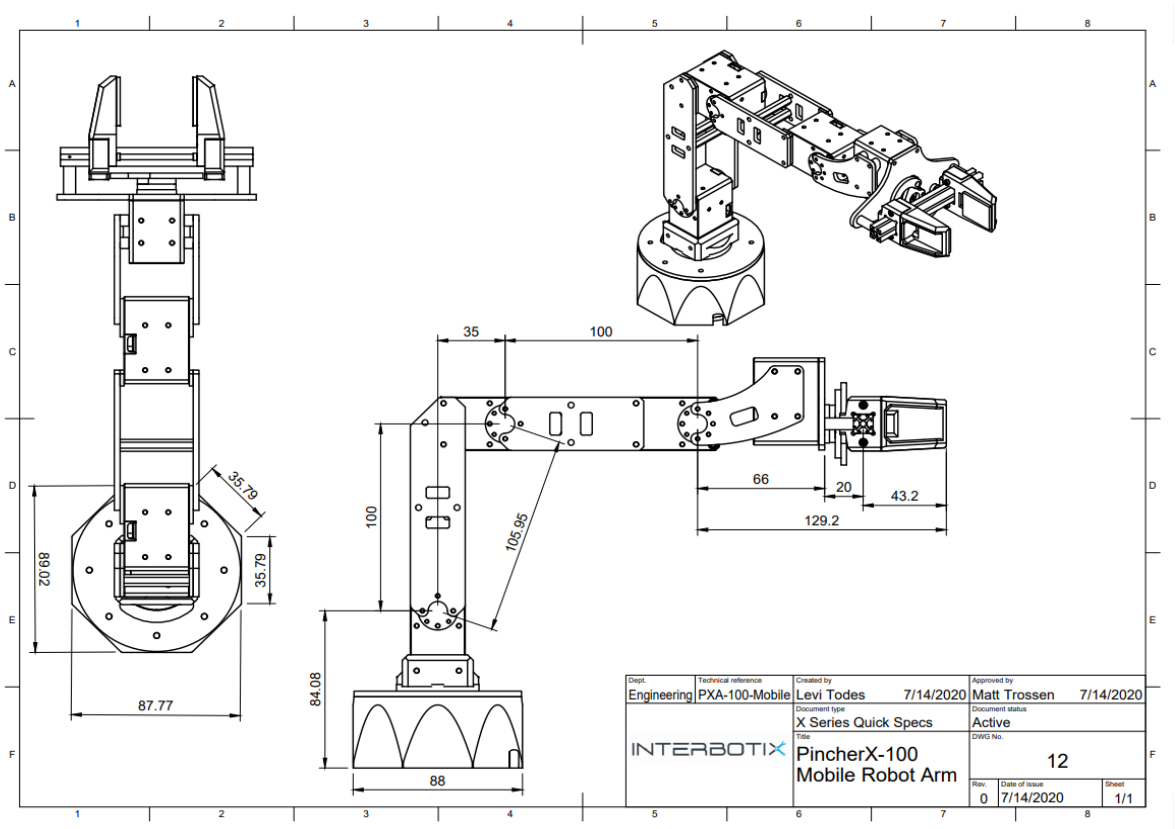

Figure 5.6: PincherX Technical Drawings

As a software development kit, DYNAMIXEL SDK provides DYNAMIXEL control functions using packet communication. The API of DYNAMIXEL SDK is designed for DYNAMIXEL actuators and DYNAMIXEL-based platforms. We may operate the actuators under all three operating systems: Windows, Linux, and MacOS, and it supports various programming languages such as C, C++, C#, Python, Java, MATLAB, and LabVIEW. In this case, we can configure register settings such as PID gains, Control Modes (position, velocity, current, or PWM) to tune joint motions via programming languages. We may also get feedback information such as firmware updates, diagnostics, configuration and testing, data plotting, generating, and monitoring DYNAMIXEL packets. Finally, the libraries of DYNAMIXEL servos may abstract away the serial communication layer allowing developers to concentrate their time on higher-level code.

Let us discuss the approaches to operate the PincherX 100 Robot Arm under the EAMPC algorithm. To begin with, we are going to design the system dynamic and corresponding control objective. The position of the links in manipulator systems should be determined by Denavit-Hartenberg parameters, which consist of four parameters associated with a particular convention for attaching reference frames to the links of the manipulator, such as

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & I \end{bmatrix} = \begin{bmatrix} R & P \\ O & I \end{bmatrix}$$

where $R = [n, o, a]$ that presents the rotation matrix with vector n –the cosine value of the angel between current x-axis and the three axises in ordinary coordinate system, vector o – the cosine value of the angel between current y-axis and the three axises in ordinary coordinate system, and vector a – the cosine value of the angel between current z-axis and the three axises in ordinary coordinate system; P presents the position vector which is the position should be appeared in ordinary coordinate system; O presents the zero matrix, and I presents the proportion between the unit length. Besides, there exist three forms of the rotation matrix as the current coordinate system get rotated compare with ordinary coordinate system such as

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_x) & -sin(\theta_x) \\ 0 & sin(\theta_x) & cos(\theta_x) \end{bmatrix}, \; R_y(\theta_y) = \begin{bmatrix} cos(\theta_y) & 0 & sin(\theta_y) \\ 0 & 1 & 0 \\ -sin(\theta_y) & 0 & cos(\theta_y) \end{bmatrix}$$

$$R_z(\theta_z) = \begin{bmatrix} cos(\theta_z) & -sin(\theta_z) & 0 \\ sin(\theta_z) & cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then we may determine the Denavit-Hartenberg matrix according to the parameters of Figure 5.6 and corresponding transformation rules as

$$T = P_1 R_z(\theta_1) R_x(\theta_2) P_2 P_3 R_x(\theta_3) P_4 R_x(\theta_4) P_4$$

$$P1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 89.45 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; P2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; P3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -35 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; R_z(\theta_1) = \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & 0 \\ sin(\theta_1) & cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\theta_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_2) & -sin(\theta_2) & 0 \\ 0 & sin(\theta_2) & cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; R_x(\theta_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_3) & -sin(\theta_3) & 0 \\ 0 & sin(\theta_3) & cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\theta_4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_4) & -sin(\theta_4) & 0 \\ 0 & sin(\theta_4) & cos(\theta_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\theta_1, \cdots, \theta_4$ present the rotation angle of each joint. And the position of the gripper finger in ordinary coordinate system should be $P = T \times \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. We find that the running velocity can be expressed as

$$\dot{x} = \frac{d}{dt} P.$$

Then we determine the discretized system dynamic and control objective as

$$\begin{aligned} &\min\ J[u_k|x_k] \\ \text{s.t. } &x_{k+1} = x_k + T_s u_k \\ &x_k \in \mathcal{X} \subset \mathbb{R}^m \\ &J[u_k|x_k] = \sum_{k=0}^{N-1} L(x_k, u_k) + F(x_N) \end{aligned}$$

where x_k presents the position of the gripper finger, u is the angular velocity of each joint, $T_s = 0.1$, and the constraint $\mathcal{X} = \{x_k \in \mathbb{R}^3 \mid x_1 \in (-300, 300), x_2 \in (-300, 300), x_3 \in (54.45, 389.45)\}$. And the control objective is always finds the shortest path operating the gripper finger to reach the destination.

Specifically, we have

$$u_k = \dot{P} = \frac{\partial}{\partial \theta_k} P \begin{bmatrix} \pm\omega & \pm\omega & \pm\omega & \pm\omega \end{bmatrix}^T,\ \theta_k = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T,\ \omega = 6.28 rad/s$$

where ω is angular velocity of each servo motor and the sign of each angular velocity is determined according to the changing angle of each motor. Meanwhile, we also set the rules on computing θ_k as

$$\begin{aligned} &\min_{\theta_k \in \Theta}\ J[\theta_k|u_k] \\ \text{s.t. } &u_k = \frac{1}{T_s} \frac{\partial}{\partial \theta_k} P \begin{bmatrix} \pm\omega & \pm\omega & \pm\omega & \pm\omega \end{bmatrix}^T,\ \omega = 6.28 rad/s \\ &J[u_k|x_k] = L(\theta_k - \theta_{k-1}) \end{aligned}$$

where $\Theta = \{\theta \in \mathbb{R}^4 |\ \theta_1 \in (0, 2\pi), \theta_2 \in (0, \pi), \theta_3 \in (-\pi, 0), \theta_4 \in (-\pi/2, \pi/2)\}$. In this case, we may always find the minimum movements of each joint to reach the destination, so that the energy on operating the manipulator get optimized.

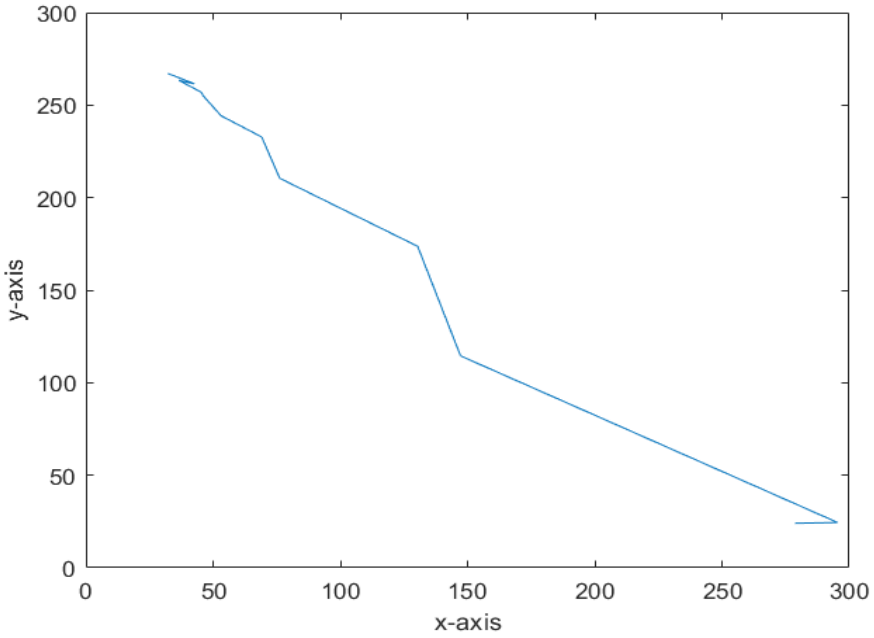

Figure 5.7: The Trajectory of Gripper

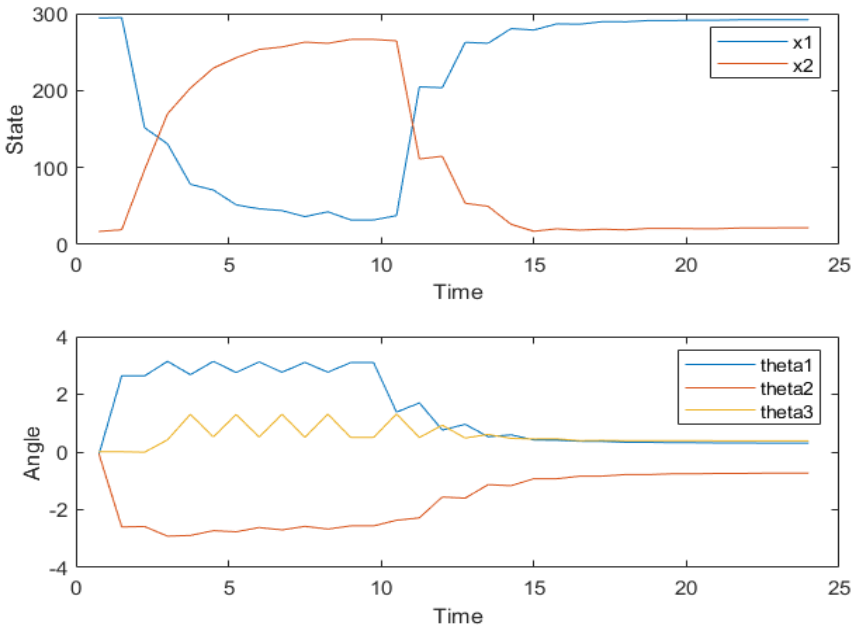

Figure 5.8: The Trajectory of Position and Angle

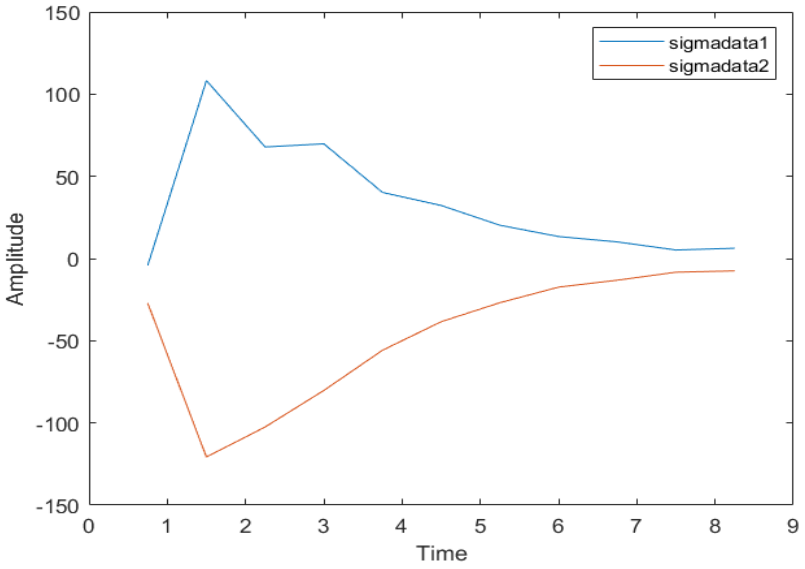

Figure 5.9: The Trajectory of σ_k

Based on previous settings, we operate the PincherX 100 manipulator arm with the code generated on Matlab. We set the step length as $T = 0.75$, the parameter of discretize estimator $\Gamma = 0.5$, the prediction steps of each iteration $n = 10$, and the total computation circle $N = 50$. Besides, we also set a threshold that the computation gets stopped as the difference between current angles and the desired angles less or equal to 0.0012 rad. And the actions of online computation and manipulator operation spend 9 seconds.

As it shown in in Figure 5.7, the control objective is find the shortest path to reach the destination for the gripper. In Figure 5.8,the trajectory of the state converge to the reference position $[35; 269.45]$ gently. And the rotation angle of each joint starts to adjust the value around the target after $t = 4$. It shows the trajectory of the estimation error σ_k shown in Figure 5.9. We may find there exists a big disturbance to the sampling state before $T = 5$, which caused by the vibration from the servo motors. And estimator provides a positive support to operate the manipulator.

In conclusion, the EAMPC algorithm shows an efficient performance to the operation of PincherX 100 manipulator arm.

Chapter 6

Conclusion and Future Work

This chapter intends to to emphasize the main contribution of the dissertation. The chapter ends with explaining the works for future research.

6.1 Conclusion

This dissertation presents three algorithms for solving the MPC problems in practical application, which may improve the optimization accuracy, control efficiency, and computation cost.

To begin with, we present an AMPC algorithm for systems with time-varying and state-dependent uncertainties in Chapter 2. In this algorithm, we estimate the uncertainty using fast adaptation. With the set-valued estimation of the uncertainty as the initial condition, we can predict the set-valued measurement of uncertainty over the prediction horizon at each computation cycle. Based on the prediction, a min-max finite-horizon optimization problem is solved for the control input. We show that the proposed adaptive uncertainty estimator can ensure uniform boundedness of the estimation error, and the derived uniform bound can be arbitrarily small by increasing the sampling rate in adaptation. Furthermore, the uncertainty is guaranteed to stay inside the predicted feasible sets, which can be much tighter than the traditional min-max MPC approach. It, therefore, allows the proposed AMPC to achieve better control performance. We show that if the uncertainty and its first derivatives are local Lipschitz, the stability of the system under AMPC can always be guaranteed under

standard assumptions for min-max MPC approaches. Simulation results show that the control performance can be improved by the proposed AMPC scheme remarkably, comparing with the traditional min-max approach.

Secondly, we introduce a sporadic MPC algorithm for nonlinear continuous-time systems based on Lebesgue approximation in Chapter 3. The sampling is triggered by a self-triggering method, and the model is discretized in an aperiodic manner. The basic idea of the Lebesgue approximation model (LAM) in this chapter is to calculate the state and the period at the same time when a linear approximation of state meets a threshold function. With such an aperiodic model, the predicted horizon gets enlarged comparing with periodic models if we set the same prediction iterations.

Besides, we investigate an explicit adaptive MPC(EAMPC) algorithm for nonlinear continuous-time systems in Chapter 4. We develop an adaptive discrete-time estimator to approximate the discretization error over the prediction horizon with bounded estimation error. Accordingly, the finite-horizon optimal control problem (FHOCP) is formulated discretely with the linearized system model. And multi-parametric programming is applied in each prediction routine to find the explicit solution of FHOCP subject to state and estimated uncertainty. In this case, online computation is mainly for function evaluation based on multi-parametric programming, and it can save computation costs dynamically. Given the resulting closed-loop system with mixed continuous/discrete-time behaviors, we derive sufficient conditions on the feasibility of the proposed EAMPC and show the system is uniformly ultimately bounded under the proposed algorithm.

Finally, we apply the EAMPC algorithm to manipulator models to study its efficiency to operate practical systems. We provide the approach to build the mathematical model of robotic manipulators and set corresponding control objectives under the EAMPC algorithm. We introduce knowledge such as Lagrange's mechanics,

Hamiltonian formulation, and Denavit-Hartenberg parameters. In conclusion, the performance of such models is efficient under the EAMPC algorithm.

6.2 Future Work

For the future work, we consider a general framework for the co-design of discrete-time MPC law and the associated scheduling schemes, where the discrete-time model used in MPC approximates the behavior of the plant with state-dependent sampling periods. We are going to study the stability of sampled-data systems controlled by discrete-time model predictive control (MPC) algorithms. Sufficient conditions are derived to guarantee uniform ultimate boundedness of the resulting closed-loop system. The results can be applied to most existing model approximation methods with either fixed or time-varying sampling rates.

We are going to relax the assumption of first-order approximation and considers a more general framework of discrete time MPC for continuous-time systems that is suitable to various approximation models with either time-triggered or event/self-triggered scheduling schemes. The discrete-time model used in MPC can approximate the behavior of the plant with state-dependent sampling periods. The allowable model approximation error is also state-dependent and therefore admits a tighter error bound in MPC framework, compared with the error bound related to period only.

Another research topic mainly focuses on deriving the system and taking advantage of explicit model predictive control, which may pre-solve the optimization problems offline and save online computation. We are going to apply a sporadic time manner to the original system as well as the structure of the explicit computation. We may follow the rules to design the aperiodic time manner based on the content of Chapter 3. Most importantly, it may overcome the limitation of the original system, such as the form of the system input in Chapter 4. Finally, the stability should

get satisfied by following the basic requirements of event/self-triggered scheduling schemes, and computation steps admit the standard explicit model predictive control procedure.

A variety of research on nonlinear model predictive control (NMPC) has been derived. Most schemes assume that the full state information can be obtained entirely. Nevertheless, the condition may not easily achieve in practice. In this case, it is important to study the approaches to estimate the system states except sampling from the plant directly. As shown in recent works, the strategy of combining sampled-data NMPC for continuous-time systems with high-gain observers may overcome this problem.

According to current research, this technique may estimate the output robustly and ensure a fast convergence of the overall system. Moreover, it has shown the high-gain observer can recover the system performance completely under a sufficiently observer gain and a bounded controller. However, it is also known that the measurement noise may have a significant impact on the stability of the system, through the current research shows the good robust performance when uncertainty and disturbances. One purpose of our future work on NMPC is to construct a high-gain observer that takes the form of a piecewise linear function and achieves a fast state estimation. We are going to design an adaptive control scheme based on a high-gain observer, which may overcome the impact of measurement noise adaptively. Meanwhile, the high-gain observer may be discretized via the Lebesgue approximation so that the computation cost gets saved under the discrete-time model. Furthermore, we may apply the explicit model predictive control strategy to reduce the computation time on solving the optimization problem.

PUBLICATIONS

1. Yang L, Wang X. Lebesgue-Approximation-Based Model Predictive Control for Nonlinear Sampled-Data Systems with Measurement Noises[C]//2018 Annual American Control Conference (ACC). IEEE, 2018: 3147-3152.

2. Yang L, Wang X. A Framework for Predictive Control of Sampled-Data Systems Using Sporadic Model Approximation[C]//2021 Annual American Control Conference (ACC).

3. Yang L, Dauchert S, Luo, J, and Wang X. Self-triggered predictive control of nonlinear systems using approximation model. IEEE Transactions on Systems, Man and Cybernetics: Systems., Peer review.

4. Wang X, Yang L, Sun Y, and Deng K. Adaptive model predictive control of nonlinear systems with state-dependent uncertainties. International Journal of Robust and Nonlinear Control, 2017, 27(17): 4138-4153.

5. Wang X, Yang L. Sporadic model predictive control using Lebesgue approximation[C]//2017 American Control Conference (ACC). IEEE, 2017: 5768-5773.

6. Tao J, Yang L, Wu Z G, et al. Lebesgue-Approximation Model Predictive Control of Nonlinear Sampled-Data Systems[J]. IEEE Transactions on Automatic Control, 2019.

Bibliography

[1] S. J. Qin and T. A. Badgwell. "An overview of nonlinear model predictive control applications". In: *Nonlinear model predictive control*. Springer, 2000, pp. 369–392.

[2] S. J. Qin and T. A. Badgwell. "A survey of industrial model predictive control technology". In: *Control engineering practice* 11.7 (2003), pp. 733–764.

[3] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss. "State and output feedback nonlinear model predictive control: An overview". In: *European journal of control* 9.2-3 (2003), pp. 190–206.

[4] P. O. Scokaert, J. B. Rawlings, and E. S. Meadows. "Discrete-time stability with perturbations: Application to model predictive control". In: *Automatica* 33.3 (1997), pp. 463–470.

[5] D Limón Marruedo, T Alamo, and E. Camacho. "Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC". In: *Proceedings of the American Control Conference*. Vol. 1. IEEE. 2002, pp. 364–369.

[6] G. Grimm, M. J. Messina, S. Tuna, and A. Teel. "Nominally robust model predictive control with state constraints". In: *Proceedings of the 42nd IEEE Conference on Decision and Control*. Vol. 2. IEEE. 2003, pp. 1413–1418.

[7] P. Scokaert and D. Mayne. "Min-max feedback model predictive control for constrained linear systems". In: *IEEE Transactions on Automatic Control* 43.8 (1998), pp. 1136–1142.

[8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[9] J. Löfberg. *Minimax approaches to robust model predictive control*. Ph.D. Thesis, 2003.

[10] M. Lazar, D Munoz de la Pena, W. Heemels, and T Alamo. "On input-to-state stability of min–max nonlinear model predictive control". In: *Systems & Control Letters* 57.1 (2008), pp. 39–48.

[11] D. Mayne and H Michalska. "Adaptive receding horizon control for constrained nonlinear systems". In: *Proceedings of the 32nd IEEE Conference on Decision and Control*. IEEE. 1993, pp. 1286–1291.

[12] D. DeHaan and M. Guay. "Adaptive robust mpc: A minimally-conservative approach". In: *Proceedings of the American Control Conference*. IEEE. 2007, pp. 3937–3942.

[13] V. Adetola, D. DeHaan, and M. Guay. "Adaptive model predictive control for constrained nonlinear systems". In: *Systems & Control Letters* 58.5 (2009), pp. 320–326.

[14] V. Adetola and M. Guay. "Robust adaptive MPC for constrained uncertain nonlinear systems". In: *International Journal of Adaptive Control and Signal Processing* 25.2 (2011), pp. 155–167.

[15] M. Farina and R. Scattolini. "Tube-based robust sampled-data MPC for linear continuous-time systems". In: *Automatica* 48.7 (2012), pp. 1473–1476.

[16] F. A. Fontes. "A general framework to design stabilizing nonlinear model predictive controllers". In: *Systems & Control Letters* 42.2 (2001), pp. 127–143.

[17] L.-S. Hu, B. Huang, and Y.-Y. Cao. “Robust digital model predictive control for linear uncertain systems with saturations”. In: *IEEE Transactions on Automatic Control* 49.5 (2004), pp. 792–796.

[18] L. Magni and R. Scattolini. “Model predictive control of continuous-time nonlinear systems with piecewise constant control”. In: *IEEE Transactions on Automatic Control* 49.6 (2004), pp. 900–906.

[19] M. Rubagotti, D. M. Raimondo, A. Ferrara, and L. Magni. “Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems”. In: *IEEE Transactions on Automatic Control* 56.3 (2011), pp. 556–570.

[20] T. Raff, D. Sinz, and F. Allgower. “Model predictive control of uncertain continuous-time systems with piecewise constant control input: a convex approach”. In: *American Control Conference, 2008*. IEEE. 2008, pp. 1109–1114.

[21] T. Shi and H. Su. “Sampled-data MPC for LPV systems with input saturation”. In: *IET Control Theory & Applications* 8.17 (2014), pp. 1781–1788.

[22] F. D. Brunner, W. Heemels, and F. Allgöwer. “Robust Event-Triggered MPC for Constrained Linear Discrete-Time Systems with Guaranteed Average Sampling Rate”. In: *IFAC-PapersOnLine* 48.23 (2015), pp. 117–122.

[23] J. Sijs, M. Lazar, and W. Heemels. “On integration of event-based estimation and robust MPC in a feedback loop”. In: *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*. ACM. 2010, pp. 31–40.

[24] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. “Event-triggered control for discrete-time systems”. In: *American Control Conference (ACC), 2010*. IEEE. 2010, pp. 4719–4724.

[25] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. "Event-triggered strategies for decentralized model predictive controllers". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 10068–10073.

[26] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. "Novel event-triggered strategies for model predictive controllers". In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE. 2011, pp. 3392–3397.

[27] A. Ferrara, G. P. Incremona, and L. Magni. "Model-based event-triggered robust MPC/ISM". In: *Control Conference (ECC), 2014 European*. IEEE. 2014, pp. 2931–2936.

[28] D. Lehmann, E. Henriksson, and K. H. Johansson. "Event-triggered model predictive control of discrete-time linear systems subject to disturbances". In: *Control Conference (ECC), 2013 European*. IEEE. 2013, pp. 1156–1161.

[29] H. Li and Y. Shi. "Event-triggered robust model predictive control of continuous-time nonlinear systems". In: *Automatica* 50.5 (2014), pp. 1507–1513.

[30] J. B. Berglind, T. Gommans, and W. Heemels. "Self-triggered MPC for constrained linear systems and quadratic costs". In: *IFAC Proceedings Volumes* 45.17 (2012), pp. 342–348.

[31] T. Gommans and W. Heemels. "Resource-aware MPC for constrained nonlinear systems: A self-triggered control approach". In: *Systems & Control Letters* 79 (2015), pp. 59–67.

[32] F. D. Brunner, W. Heemels, and F. Allgöwer. "Robust self-triggered MPC for constrained linear systems". In: *Control Conference (ECC), 2014 European*. IEEE. 2014, pp. 472–477.

[33] A. Eqtami, S. Heshmati-alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos. "Self-triggered model predictive control for nonholonomic systems". In: *Control Conference (ECC), 2013 European*. IEEE. 2013, pp. 638–643.

[34] E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. "Self-Triggered Model Predictive Control for Network Scheduling and Control1". In: *IFAC Proceedings Volumes* 45.15 (2012), pp. 432–438.

[35] A. Alessio and A. Bemporad. "A survey on explicit model predictive control". In: *Nonlinear model predictive control*. Springer, 2009, pp. 345–369.

[36] C. N. Jones, M. Barić, and M. Morari. "Multiparametric linear programming with applications to control". In: *European Journal of Control* 13.2-3 (2007), pp. 152–170.

[37] A. Lasheen, M. S. Saad, H. M. Emara, and A. L. Elshafei. "Continuous-time tube-based explicit model predictive control for collective pitching of wind turbines". In: *Energy* 118 (2017), pp. 1222–1233.

[38] L. Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

[39] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems". In: *Automatica* 38.1 (2002), pp. 3–20.

[40] P. TøNdel, T. A. Johansen, and A. Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions". In: *Automatica* 39.3 (2003), pp. 489–497.

[41] A. Bemporad, F. Borrelli, M. Morari, et al. "Model predictive control based on linear programming˜ the explicit solution". In: *IEEE transactions on automatic control* 47.12 (2002), pp. 1974–1985.

[42] A. Bemporad. "Multiparametric nonlinear integer programming and explicit quantized optimal control". In: *Proc. of the 42nd IEEE Conference on Decision and Control*. 2003, pp. 3167–3172.

[43] M. de la Pena, A. Bemporad, and C. Filippi. "Robust explicit MPC based on approximate multi-parametric convex programming". In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 3. IEEE. 2004, pp. 2491–2496.

[44] T. A. Johansen. "Approximate explicit receding horizon control of constrained nonlinear systems". In: *Automatica* 40.2 (2004), pp. 293–300.

[45] N. Hovakimyan and C. Cao. $\mathcal{L}_1$ *Adaptive Control Theory*. Philadelphia, PA: SIAM, 2010.

[46] D. P. Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 2. Athena Scientific Belmont, MA, 1995.

[47] X. Wang and N. Hovakimyan. "$\mathcal{L}_1$ adaptive controller for nonlinear time-varying reference systems". In: *Systems & Control Letters* 61.4 (2012), pp. 455–463.

[48] J. Che and C. Cao. "$\mathcal{L}_1$ adaptive control of system with unmatched disturbance by using eigenvalue assignment method". In: *Proceedings of the 51st IEEE Conference on Decision and Control*. IEEE. 2012, pp. 4823–4828.

[49] S. Sastry and M. Bodson. *Adaptive control: stability, convergence and robustness*. Dover Publications, 2011.

[50] R. Findeisen, L. Imsland, F. Allgöwer, and B. Foss. "Towards a sampled-data theory for nonlinear model predictive control". In: *New Trends in Nonlinear Dynamics and Control and their Applications*. Springer, 2003, pp. 295–311.

[51] E. A. Morelli. "Global Nonlinear Parametric Modeling with Application to F-16 Aerodynamics". In: *Proceedings of American Control Conference*. Vol. 2. Philadelphia, PA, 1998, pp. 997–1001.

[52] P. Tabuada. "Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks". In: *IEEE Transactions on Automatic Control* 52.9 (2007), pp. 1680–1685.

[53] X. Wang, E. Kharisov, and N. Hovakimyan. "$\mathcal{L}_1$ Adaptive Control of Uncertain Networked Control Systems". In: *Proceedings of the American Control Conference*. IEEE. 2013.

[54] B. Houska, H. J. Ferreau, and M. Diehl. "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range". In: *Automatica* 47.10 (2011), pp. 2279–2285.

[55] M. Tawarmalani and N. V. Sahinidis. "A polyhedral branch-and-cut approach to global optimization". In: *Mathematical Programming* 103.2 (2005), pp. 225–249.

[56] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari. "Multi-parametric toolbox 3.0". In: *2013 European control conference (ECC)*. IEEE. 2013, pp. 502–510.

[57] F. L. Lewis, D. M. Dawson, and C. T. Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.