

Aperiodic Model Predictive Control for Networked Control Systems

February 2018

Kazumune Hashimoto

KEIO UNIVERSITY

DOCTORAL THESIS

**Aperiodic Model Predictive Control
for Networked Control Systems**

Author:

Kazumune HASHIMOTO

Supervisor:

Prof. Shuichi ADACHI

School of Fundamental Science and Technology

February 2018

Acknowledgements

First of all, I would like to express my sincere appreciation to my advisor Prof. Shuichi Adachi, who had provided valuable comments and advices on my PhD studies. I would also like to thank you for giving many advices to improve my presentation and academic writing skills both in Japanese and English. In addition, I'm indebted to you for providing many contacts with Japanese researchers and opportunities to attend many domestic and international conferences. Without these supports, it would not be possible to achieve the research contributions obtained so far and grow as a research engineer.

I would also like to thank Prof. Dimos V. Dimarogonas (KTH Royal Institute of Technology, Sweden) for many years' collaboration. I am deeply indebted to you for having valuable discussions on my research and checking many journal and conference papers written in English. In particular, my two months' stay at KTH as a visiting researcher not only improved my research results but also expanded my research field.

I also thank all labmates both at Adachi Lab. and at Automatic Control Lab. at KTH, who gave me many comments and discussions on my PhD research. In addition, sometimes I could relax by playing sports, eating together, and traveling with you. Without that I would not refresh my mind and achieve the research contributions until now. I would not forget the precious time, and look forward to meeting you again soon.

Finally, I would like to thank my sister and parents for all your support and encouragement. Thank you very much for allowing me to leave the company and become a PhD student to dive into academic fields. All your support to continue my research career and my pursuit to be a research engineer is so appreciated. Thank you very much.

January 2018
Kazumune Hashimoto

Contents

Acknowledgements	iii
Abstract	xv
1 Introduction	1
1.1 Event-triggered and Self-triggered control	1
1.2 Model Predictive Control	4
1.3 Contributions and Outline of thesis	6
2 Basic methodologies	15
2.1 Event-triggered and Self-triggered control	15
2.1.1 Event-triggered control	15
2.1.2 Self-triggered control	17
2.1.3 Event-triggered and Self-triggered MPC	18
2.2 Theoretical background of MPC	19
2.2.1 Feasibility and stability	21
2.2.2 Discussions	24
2.3 Summary	26
3 Aperiodic MPC for linear systems	27
3.1 Multiple discretizations approach	27
3.1.1 Problem formulation	27
3.1.2 Self-triggered strategy	38
3.1.3 Feasibility and Stability analysis	41
3.1.4 Simulation results	42
3.2 Contractive set-based approach	54
3.2.1 Set-Invariance theory	55
3.2.2 Self-triggered strategy	62
3.2.3 Feasibility and Stability analysis	63
3.2.4 Discussions on computational complexity	64
3.2.5 Simulation results	66
3.3 Summary	71

4	Aperiodic MPC for Nonlinear Input-affine systems	73
4.1	Problem formulation	74
4.2	Self-triggered strategy	80
4.3	Choosing sampling time intervals	86
4.4	Stability analysis	92
4.5	Simulation results	95
4.6	Summary	101
5	Aperiodic MPC for General Nonlinear systems	111
5.1	Problem formulation	112
5.2	Feasibility analysis	118
5.3	Event-triggered strategy	121
5.4	Self-triggered strategy	128
5.5	Stability analysis	129
5.6	Simulation results	131
5.7	Summary	142
6	Conclusion and future work	145
6.1	Conclusion	145
6.2	Future work	146
A	Mathematical preliminaries	149
A.1	Lyapunov Stability	149
A.2	Gronwall-Bellman inequality	150
	Bibliography	151

List of Figures

1.1	Networked Control System	2
1.2	Illustration of time-triggered control and event and self-triggered control. In the figure, blue arrows represent transmission time instants.	2
1.3	Basic idea of Model Predictive Control	5
2.1	Event-triggered control architecture incorporating the Event-Triggering Mechanism (ETM).	16
3.1	Illustration of the piece-wise constant control policy.	28
3.2	Sampling patterns considered in this paper. Blue lines represent the transmission time intervals.	31
3.3	Optimal piecewise constant control policy for the i -th sampling pattern (blue line) and the admissible control sequence for the $(i - 1)$ -th pattern $\bar{u}_{i-1}(t_k)$ (black circles).	34
3.4	State trajectories by implementing Algorithm 3.1 (blue solid lines) and the periodic MPC (red dotted lines).	44
3.5	Applied control inputs by applying Algorithm 3.1 (blue solid line) and periodic MPC with sampling time interval 0.1 (red dotted line).	45
3.6	Number of transmissions during the time interval $t \in [0, 300]$ and the average calculation time against the number of sampling patterns M	46
3.7	State trajectories of x_1 and x_2 by implementing Algorithm 3.1 (blue solid lines) and the periodic MPC (red dotted lines).	49
3.8	State trajectories of x_3 and x_4 by implementing Algorithm 3.1 (blue solid lines) and the periodic MPC (red dotted lines).	50
3.9	Applied control inputs under Algorithm 3.1 (blue line) and periodic MPC with sampling time interval 0.1 (red dotted line).	51
3.10	Number of transmission time instants during the time period $t \in [0, 300]$ and the calculation time against the number of sampling patterns M	53
3.11	λ -contractive sets \mathcal{S}_1 ($\lambda = 1.0$), \mathcal{S}_2 ($\lambda = 0.9$), and \mathcal{S}_3 ($\lambda = 0.8$). The contractive sets are illustrated with different shades of blue. The gray region represents \mathcal{X}	59
3.12	Number of decision variables used for Problem 3.1 and Problem 3.2.	65
3.13	State trajectories of x_1 and x_2 by implementing Algorithm 3.2 (blue solid lines) and Algorithm 3.1 (red dotted lines).	67

3.14	State trajectories of x_3 and x_4 by implementing Algorithm 3.2 (blue solid lines) and Algorithm 3.1 (red dotted lines).	68
3.15	Applied control inputs by applying Algorithm 3.2 (blue solid line) and Algorithm 3.1 (red dotted line).	70
3.16	Calculation times against M under Algorithm 3.1 (blue) and Algorithm 3.2 (red).	70
4.1	The illustration of three regions Σ_V , Φ , Φ_f , and an example of optimal state trajectory $x^*(\xi)$	75
4.2	Based on the optimal control trajectory (black line), the controller picks up N control input samples (red circles) and these samples are transmitted to the plant and applies them as sample-and-hold fashion (red line).	78
4.3	The way to find sampling intervals. L.H.S and R.H.S are the evolutions of left-hand-side and right-hand side in (4.26).	103
4.4	The illustration of Φ and the restricted terminal region Φ_f	103
4.5	State trajectories of x_1 and x_2 by implementing Algorithm 4.1 (blue solid lines) and the periodic MPC (red dotted lines).	104
4.6	State trajectories of x_3 and x_4 by implementing Algorithm 1 (blue solid lines) and the periodic MPC (red dotted lines).	105
4.7	Applied control inputs by applying Algorithm 4.2 (blue solid line) and periodic scheme with sampling time interval 0.1 (red dotted line).	106
4.8	State variables for a vehicle regulation problem in two dimensions.	106
4.9	Trajectory of the vehicle by applying Algorithm 4.2.	107
4.10	Control trajectory of v and ω implementing Algorithm 4.2.	108
4.11	Trajectory of the vehicle by applying periodic MPC scheme with 0.88 sampling time interval.	109
4.12	Trajectory of the vehicle by applying periodic MPC scheme with 0.1 sampling time interval.	109
4.13	Number of transmission instants against the number of control samples N	110
5.1	Graphical representation of the two regions Φ , Φ_f , and the optimal state trajectory \hat{x}^* (blue solid line). T_k^* denotes the time interval to reach Φ_f	116
5.2	The illustration of the problem presented in (P.1). The figure shows the left hand side (black solid) and the right hand side (black dotted) of (5.19).	123
5.3	The figure illustrates the problem of violating the feasibility described in (P.2).	124
5.4	State trajectories of x_1 and x_2 by implementing Algorithm 5.1 (blue solid lines) and the periodic MPC (red dotted lines).	133
5.5	State trajectories of x_3 and x_4 by implementing Algorithm 5.1 (blue solid lines) and the periodic MPC (red dotted lines).	134

5.6	Control inputs by applying Algorithm 5.1 (blue line) and the periodic scheme (red dotted line).	135
5.7	Trajectory of the vehicle by applying Algorithm 5.1.	137
5.8	Control trajectory of v and ω implementing Algorithm 5.1 and the periodic one without disturbances (red dotted line).	138
5.9	Trajectory of the vehicle by applying the periodic MPC with 1.0 sampling time interval.	139
5.10	Trajectory of the vehicle by applying Algorithm 5.1 with large disturbance sets \mathcal{W}	140
5.11	Trajectory of the vehicle by applying the periodic MPC with 0.1 sampling time interval.	141

List of Tables

3.1	Convergence time when the state trajectory enters the region ($\ x\ \leq 0.001$) and the number of transmission instants.	45
3.2	Convergence time when the state trajectory enters the region ($\ x\ \leq 0.001$) and the number of transmission instants during the time period $t \in [0, 30]$	52
3.3	Number of decision variables used in Algorithm 3.1 ($N_p = 50$) and Algorithm 3.2.	65
3.4	Convergence time when the state trajectory enters the region ($\ x\ \leq 0.001$) and the number of transmission instants during the time period $t \in [0, 30]$	69
3.5	Calculation times against M under Algorithm 3.1 and Algorithm 3.2. . .	71
4.1	Convergence time when the state trajectory enters the region ($\ x\ \leq 0.001$) and the number of transmission instants	97
4.2	Convergence time when the state trajectory enters around the origin and the number of transmission instants.	100
4.3	Convergence time when the trajectory of the vehicle enters around the origin and the number of transmission instants.	100
5.1	Convergence time when the state trajectory enters Φ and the number of transmission instants.	135
5.2	Convergence time when the state trajectory enters around the origin and the number of transmission instants.	140
5.3	Convergence time when the state trajectory enters Φ and the number of transmission instants.	142

List of Symbols, Abbreviations

\mathbb{N}	set of integers
$\mathbb{N}_{\geq 0}$	set of non-negative integers
$\mathbb{N}_{> 0}$	set of positive integers
\mathbb{R}	set of real numbers
$\mathbb{R}_{\geq 0}$	set of non-negative real numbers
$\mathbb{R}_{> 0}$	set of positive real numbers
\mathbb{R}^n	n -dimensional Euclidean space
$\mathbb{R}^{n \times m}$	the set of all $n \times m$ real matrices
A^T	transpose of A
A^{-1}	inverse of A
$\lambda_{\min}(A)$	minimum eigenvalue of A
$\lambda_{\max}(A)$	maximum eigenvalue of A
I_n	$n \times n$ identity matrix
$A \cap B$	intersection of the sets A and B
$A \cup B$	union of the sets A and B
$\text{co}\{v_1, \dots, v_N\}$	convex hull of the vertices v_1, \dots, v_N
Ψ_S	the function $\Psi_S : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is the gauge function if defined as $\Psi_S(x) = \inf\{\mu : x \in \mu S, \mu \geq 0\}$ for a given set $S \subset \mathbb{R}^n$
\mathcal{K}_∞	a function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a class \mathcal{K}_∞ function if it is continuous, strictly increasing, $\alpha(0) = 0$, and $\alpha(\infty) = \infty$
$\ x\ $	Euclidean norm of $x \in \mathbb{R}^n$
$\ x\ _Q$	weighted norm defined as $\ x\ _Q = \sqrt{x^T Q x}$
MPC	Model Predictive Control
OCP	Optimal Control Problem
NCSs	Networked Control Systems
LTI	Linear Time Invariant

Abstract

Networked Control Systems (NCSs) are systems whose sensors, actuators, and controllers are spatially distributed over communication channels. On one hand, it is well-known that a major concern is the energy consumption of battery powered devices due to the network communications. Thus, it is of great importance to reduce communication frequencies between the plant and the controller. On the other hand, many control systems are typically subject to hard constraints, such as actuator saturations. Therefore, this thesis proposes a control framework by applying ‘aperiodic control’, which achieves communication reduction in NCSs, and ‘Model Predictive Control (MPC)’, which takes into account hard constraints. Throughout the thesis, this combinational control scheme is referred to as ‘aperiodic MPC’.

In Chapter 1, the purpose and the outline of this thesis are given. The purpose of the thesis is to provide a control framework to reduce communication frequencies between controller and the plant, while at the same time guaranteeing both control performance and the satisfaction of hard constraints.

In Chapter 2, some basic methodologies of aperiodic control and MPC are given.

In Chapter 3, a control problem of linear systems is considered to formulate an aperiodic model predictive control. In the aperiodic formulation, the timings for sensors to transmit state measurements are determined based on Lyapunov stability, so that the stabilization of the system and communication reduction for NCSs can be achieved.

In Chapter 4, an aperiodic formulation of MPC is proposed for nonlinear input-affine systems, which are thus provided for a more general class of systems than the ones in Chapter 2. As with Chapter 2, the way to communicate between the plant and the controller is given based on Lyapunov stability.

In Chapter 5, an aperiodic formulation of MPC is proposed for a more general class of systems than the ones presented in Chapter 3 and 4. In particular, the author will derive a threshold between the predictive states and the actual state, such that feasibility of the optimal control problem and stability are both guaranteed. The derived threshold is provided as a criterion for the communication timing between the plant and the controller, so that both control performance and communication reduction are achieved.

In Chapter 6, some conclusions are provided.

Chapter 1

Introduction

1.1 Event-triggered and Self-triggered control

With the advent of communication technologies, there has been a growing trend of introducing a communication network in many control applications, such as manufacturing plants, autonomous robots, traffic systems, and so on [1]. Typically, a control system whose sensors, actuators, and controllers are spatially distributed and connected over communication channels is often referred to as *Networked Control Systems* (NCSs). The illustration of NCSs is shown in Fig. 1.1. The advantage of incorporating the communication network is that: (i) it enables to save maintenance cost by eliminating physical components, such as redundant wirings; (ii) it allows to increase flexibility to reconfigure a system for building up new control architectures; (iii) it allows to control a system *remotely* in distant areas. Consequently, NCSs are becoming more and more ubiquitous and have seen an increasing attention in recent years, see e.g., [1], [2] for survey papers.

In NCSs, introducing the communication network has raised new challenges with regard to network uncertainties and constraints. For example, network delays and packet losses are typically present while transmitting control signals or sensor data over a communication channel. It is well-known, that the presence of delays and packet losses can potentially degrade control performance or even destabilizing the system [3]. In view of this, various results have

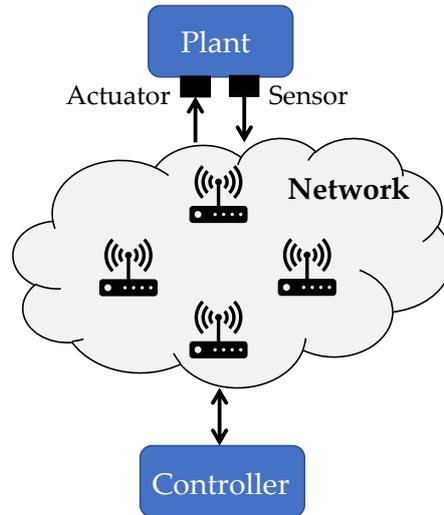


FIGURE 1.1: Networked Control System

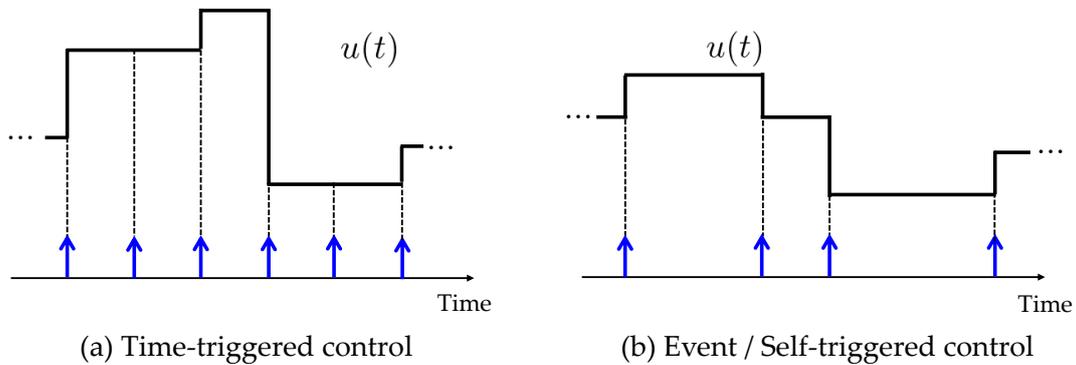


FIGURE 1.2: Illustration of time-triggered control and event and self-triggered control. In the figure, blue arrows represent transmission time instants.

been appeared to analyze the relation among network uncertainties, control performance, and stability, see e.g., [3]–[6].

Another main challenge of NCSs lies in the fact that NCSs are subject to a limited nature of *communication and computation resources*, which will be the main focus in this thesis. In sensor networks, sensor and relay nodes are typically battery driven that are equipped with a frugal battery capacity. Therefore, sensor nodes are subject to a limited amount of available energy, and designing appropriate feedback controllers to save the energy consumptions is a crucial problem to be solved. To tackle this problem, two major control schemes have been proposed; *event-triggered control* and *self-triggered control* [7], which

are collectively referred to as *aperiodic control*. In both control strategies, the objective is to reduce communication frequencies between the plant and the controller. Specifically, sensor data and control signals are exchanged over a communication network *only when* they are needed. In contrast to the typical time-triggered control framework where control inputs are executed periodically, event-triggered and self-triggered control require the executions in an *aperiodic* fashion. The illustration of the time-triggered control and the aperiodic control is depicted in Fig. 1.2. The aperiodic scheme can potentially lead to energy savings of battery powered devices, since the communication over the network is known to be one of the crucial energy consumers.

Event-triggered control and self-triggered control are essentially different in the sense that in the former case communication times are determined in the plant side, while in the latter case those are determined in the controller side. In the event-triggered control framework, the plant determines suitable communication times by continuously monitoring sensor measurements and evaluate (event-triggered) conditions, which are derived based on stability or closed-loop control performance (for details, see Chapter 2). Only when these conditions are violated, then the communication events are triggered. So far, the event-triggered framework has been analyzed for many different types of systems, including linear systems [8]–[15], nonlinear systems [16]–[18], and distributed control systems [19], [20]. In the self-triggered case, on the other hand, the controller pre-determines the next communication time as soon as the current sensor measurements are received from the plant. The reader can refer to many results also for the self-triggered case for linear systems [21]–[23] and nonlinear systems [24]. Moreover, some experimental validations of applying the event-triggered and self-triggered control schemes have been also provided, see e.g., [25]–[27].

In summary, event-triggered and self-triggered control have been proposed as promising control strategies to reduce communication frequencies between plant and controller system, which aims to save the energy consumption of

battery powered devices in NCSs. While there has been a growing attention of these strategies and many theoretical results have been proposed as illustrated above, only a few attention may be paid for designing the aperiodic strategies for *constrained* systems, where certain constraints such as actuator saturations or physical constraints need to be explicitly taken into account. This motivates us to introduce the concept of *model predictive control*, as provided in the next section.

1.2 Model Predictive Control

In many control applications including NCSs, it is typical that the control systems are subject to *hard constraints*. For instance, control signals are in general bounded due to actuator saturations. Autonomous robots such as unmanned ground vehicles must avoid colliding with obstacles and humans. In robot manipulators, the joint angles may be restricted to be within a certain range due to the structural constraint. In flight control, pitch angles must be small enough to achieve comfort for the passengers.

Model Predictive Control (MPC), which is often referred to as receding horizon control, offers a significant advantage in dealing with such hard constraints as illustrated above. The idea of MPC is illustrated in Fig. 1.3. Roughly speaking, the controller repeatedly solves a finite horizon optimal control problem online to compute optimal control actions over a prediction horizon, based on the knowledge of current state information and future system behavior from the plant dynamics. After solving the optimal control problem, only the current optimal control action is applied and the optimal control problem is again solved at the next update time. In this manner, MPC scheme allows to guarantee constraint satisfactions explicitly by solving a *constrained* optimal control problem.

MPC is categorized as an advanced control technology and firstly saw successful control applications in process industries [28]–[30]. With the advance

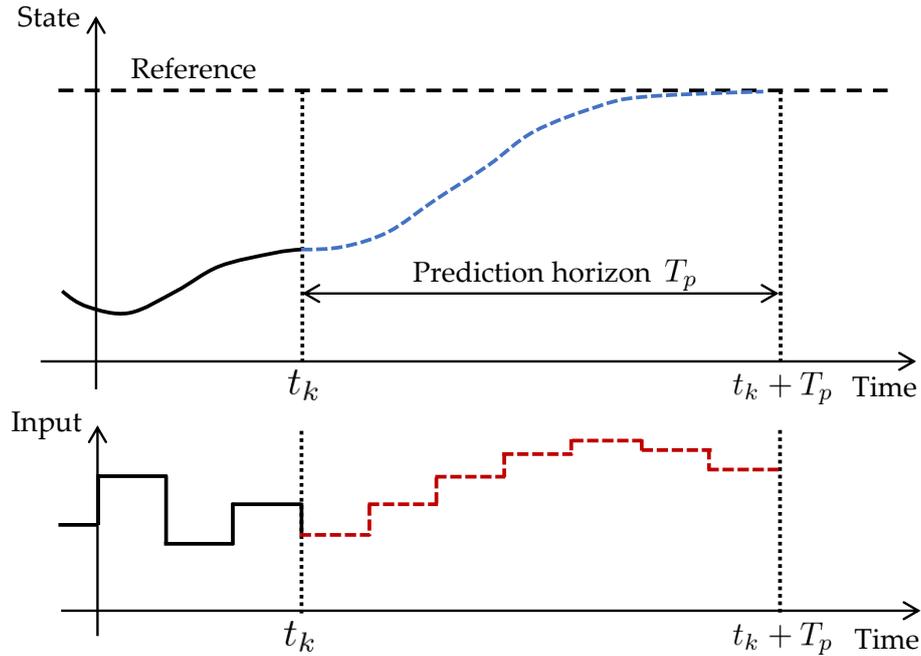


FIGURE 1.3: Basic idea of Model Predictive Control

of computation power, MPC is now applied in various control applications, including automobiles and aircrafts [31], [32], formation control of multi-vehicles [33], [34], power systems [35], medical engineering [36], [37], to name a few. Moreover, MPC has also seen a number of theoretical progresses for both linear and nonlinear systems. For instance, in [38] the authors showed asymptotic stability of an equilibrium point by applying MPC for nonlinear control systems. In particular, they showed that the optimal cost, which is regarded as a Lyapunov function candidate, is shown to decrease by introducing the notion of *terminal constraint*. Roughly speaking, the terminal constraint imposes that the predictive state at the terminal time must belong to a local region around the origin, in which a linear state-feedback controller exists to stabilize the system. In recent years, the authors showed in [39], [40] that asymptotic stability is still guaranteed, by using a sufficient long prediction horizon, without having to impose such terminal constraint. Robustness properties are also analyzed by various researchers, see e.g., [41]–[45] and the references therein. For example,

[41] showed that the robustness properties of MPC are related to the ones of infinite horizon optimal control problem, by introducing the notion of Fake Algebraic Reccati Equation (FARE). Moreover, Input-to-State Stability (ISS) of MPC against bounded external disturbances has been shown in [44], [45], where the optimal cost is shown to be an ISS Lyapunov function candidate.

As above, the concept of MPC has received a lot of attention for the past decades in terms of both control applications and theory. Note that in the earlier MPC applications as well as the theoretical results illustrated in the aforementioned papers, there has been a fundamental assumption that control inputs are updated in a *periodic fashion*, i.e., the optimal control problem is solved periodically under a specific sampling time period. Thus, this formulation may not be preferable, especially for NCSs, since solving the optimal control problem periodically leads to a high energy consumption of battery powered devices due to the periodic communication between the plant and the controller. Therefore, it is more useful to consider, like event-triggered and self-triggered strategies, that the controller solves the optimal control problem *only when it is needed*, instead of periodically. This motivates us to consider the concept of *event-triggered MPC* and *self-triggered MPC*, which serves as the main contribution of this thesis.

1.3 Contributions and Outline of thesis

The main contribution of this thesis is to blend the above two important control concepts; namely, *event-triggered and self-triggered control*, which allows to achieve energy-savings for NCSs, and *Model Predictive Control*, which mainly allows to deal with systems subject to hard constraints. In the following, the author call this combinational control scheme as *event-triggered MPC* and *self-triggered MPC*, which are collectively referred to as *aperiodic MPC*. Specifically, the author considers NCSs architecture illustrated in Fig. 1.1, where the plant

system is subject to hard constraints and the controller implements MPC framework, and aims to derive event and self-triggered strategies to determine suitable communication times (i.e., communication times to solve an optimal control problem). Note that, as previously mentioned, it has been fundamentally assumed in the standard MPC that the optimal control problem is solved periodically. Thus, this thesis is dedicated to provide theoretical analysis of MPC when the optimal control problem is solved *aperiodically*. More specifically, the communication times between the plant and the controller are determined based on stability and feasibility of MPC, which will be analyzed for various systems including linear and nonlinear systems.

Overall, the aperiodic schemes proposed in this thesis are categorized into three parts according to the system description of the plant; linear systems (Chapter 3), nonlinear input-affine systems (Chapter 4) and general nonlinear systems (Chapter 5). Thus, the class of plant dynamics will be considered more and more in general as the chapter moves forward. For each chapter, the author analyzes the corresponding stability and feasibility of MPC in order to formulate the aperiodic strategy. The structure of this thesis is described below.

Chapter 2: — Basic methodologies

In this chapter, the author reviews some basic methodologies of event-triggered and self-triggered control. Moreover, some theoretical results of MPC, such as recursive feasibility and asymptotic stability of periodic MPC scheme are provided by following the theoretical result provided earlier in [38]. The proofs for the theoretical results are key ingredients to derive the event and self-triggered strategies provided in the remaining chapters.

Chapter 3 — Aperiodic MPC for linear systems

In this chapter, a control problem of Linear-Time-Invariant (LTI) systems is

given, and an aperiodic formulation of MPC is proposed. In particular, the author proposes two different types of self-triggered MPC frameworks that aim to achieve the communication reduction for NCSs. In the first approach, the author formulates a set of optimal control problems such that the controller obtains stabilizing control inputs under multiple *candidates* of transmission time intervals. Among the multiple solutions, the controller then selects a suitable one such that both control performance and communication load are taken into account. Asymptotic stability of the origin is ensured by using Lyapunov techniques, where the Lyapunov function is induced by the optimal cost. Although the first approach guarantees asymptotic stability, it may lead to a high computational load as it requires to solve multiple optimization problems online. Therefore, the author secondly proposes an alternative strategy that aims to overcome the computational drawback of the first proposal. The key idea is to incorporate the notion of *contractive set* when formulating the optimal control problem. As will be seen in this chapter, incorporating the contractive set can potentially reduce the size of decision variables compared to the first approach, while at the same time guaranteeing both feasibility of the optimal control problem and asymptotic stability. Some simulation results are also illustrated to validate the proposed control schemes. To summarize, the contribution of this chapter is

- Two self-triggered MPC schemes for LTI systems are proposed.
- In both schemes, asymptotic stability and feasibility of the optimal control problem are guaranteed.
- Some simulation examples illustrate the effectiveness of the proposed approaches.

The results presented in Chapter 3 are related to the following journal paper:

- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Aperiodic Sampled-Data Control via Explicit Transmission Mapping: A Set Invariance Approach," IEEE Transactions on Automatic Control (to appear).*

Moreover, the results are also related to the following peer-reviewed conference papers:

- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered Model Predictive Control for Continuous-Time Systems: A Multiple Discretizations Approach," in Proceedings of the 55th IEEE Conference on Decision and Control (IEEE CDC), 2016, pp. 3078-3083.*
- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered control for constrained systems: a contractive set based approach," in Proceedings of American Control Conference (ACC), 2017, pp. 1011-1016.*

Chapter 4 — Aperiodic MPC for Nonlinear Input-affine systems

In this chapter, the author proposes aperiodic MPC schemes for nonlinear input-affine systems, which provides for a wider class of systems than the linear systems considered in Chapter 3. Here, an aperiodic formulation is given in a self-triggered fashion, by evaluating the optimal cost as a Lyapunov function candidate. Namely, communication times are determined only when the optimal cost is not guaranteed to decrease. Additionally, the bandwidth limitation of communication network will be taken into account, which means that the controller is restricted to transmit a limited number of control samples. Specifically, the controller not only solves the optimal control problem but also *discretizes* the obtained optimal control input trajectory into several control input samples, so that these can be transmitted as a packet to the plant. The discretizing method is to some extent relevant to *roll-out event-triggered control*, which is introduced in [46]. In this approach the authors proposed a way to pick up the transmission time step for linear discrete time systems, and then show that

the proposed control policy provides better performance than the conventional periodic optimal control in terms of the reduced value function. In contrast to the result presented in [46], the author proposes a way to adaptively select sampling time intervals to reduce the communication load. While this may lead to additional optimization problems, an efficient way of choosing the sampling intervals will be given. Moreover, while the results presented in [46] considers linear systems, the author deals with nonlinear systems. Finally, some simulation examples are given to validate the proposed self-triggered scheme by considering both linear and nonlinear control systems. To summarize, the contribution of this chapter is

- A self-triggered MPC scheme for nonlinear input-affine systems is proposed.
- Stability under the sample-and-hold implementation is shown by guaranteeing that the optimal cost as a Lyapunov function candidate is strictly decreasing.
- An efficient way to adaptively select suitable control samples that should be transmitted to the plant will be given.
- Some simulation examples are given to validate the proposed scheme by considering both linear and non-linear systems.

The results given in this chapter are related to the following journal paper:

- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered Model Predictive Control for Nonlinear Input-Affine Dynamical Systems via Adaptive Control Samples Selection," IEEE Transactions on Automatic Control, vol. 62, no. 1, pp. 177-189, 2017.*

Moreover, the results are also related to the following peer-reviewed conference paper:

- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered Nonlinear Model Predictive Control for Networked Control Systems," in Proceedings of American Control Conference (ACC), 2015, pp. 4239-4244*

Chapter 5 — Aperiodic MPC for General Nonlinear systems

In this chapter, the author will propose aperiodic MPC schemes for nonlinear systems. The aperiodic formulations proposed in this chapter can be applicable to general nonlinear systems (including input-affine systems), which are thus provided for a wider class of systems than the ones presented in both Chapter 3 and 4. Moreover, the author considers the case when systems are *perturbed* by additive bounded disturbances. In the previous chapters, the aperiodic formulations are derived by evaluating the optimal cost as a Lyapunov function candidate. In this chapter, on the other hand, the author will provide an alternative evaluation to guarantee stability and derive the triggering strategies. In the stability derivations, the author instead evaluates the *time interval* when the optimal state trajectory enters a local region around the origin. By guaranteeing that this time interval becomes smaller as the optimal control problem is solved, it is ensured that the state enters a prescribed set in finite time. The triggering strategies are firstly derived in an event-triggered manner, and the self-triggered strategy is secondly proposed as a sufficient condition to the event-triggered strategy.

The derivation of the new stability is motivated by the fact that the aperiodic formulation provided in Chapter 4 includes Lipschitz constant parameters for the stage and terminal cost. Since these parameters are in fact characterized by the *maximum* distance of the state from the origin, the triggering condition becomes largely affected by the state domain considered in the problem formulation. That is, as a larger state domain is considered, the event-triggered condition becomes more conservative. Furthermore, if the exact state domain is not known (e.g., if there exists no physical limitations), these parameters are not

known explicitly. Depending on the problem formulation, therefore, it is not desirable to include these parameters in the event-triggered condition. Since the proposed approach presented in this chapter does not evaluate the optimal cost as a Lyapunov function candidate, the corresponding event-triggered conditions do not include such un-suitable parameters. The author will also illustrate through a simulation example that the proposed approach attains much less conservative result than the result presented in Chapter 4. To summarize, the contribution of this chapter is

- The author proposes event-triggered and self-triggered MPC schemes for nonlinear systems with additive bounded disturbances.
- Stability analysis is given *without* evaluating the optimal cost as a Lyapunov function candidate.
- Some simulation examples illustrate the effectiveness of the proposed approach. In particular, the author shows that less conservative result is achieved than the approach presented in Chapter 4.

The results presented in this chapter are related to the following journal paper:

- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Event-triggered Intermittent Sampling for Nonlinear Model Predictive Control," Automatica, vol. 81, pp. 148-155, 2017.*

Moreover, the results are also related to the following peer-reviewed conference papers:

- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "A Collision-free Communication Scheduling for Nonlinear Model Predictive Control," in Proceedings of the 20th IFAC World Congress (IFAC WC), 2017, pp. 8939-8944.*
- *K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Time-constrained Event-triggered Model Predictive Control for Nonlinear Continuous-time Systems," in Proceedings of the 54th IEEE Conference on Decision and Control (IEEE CDC),*

2015, pp. 4326-4331.

Chapter 6 — Conclusion and Future work

In this chapter, conclusions of this thesis and future works are provided. In particular, the author discusses some other variants of MPC schemes such as tube-based MPC, unconstrained MPC, and stochastic MPC, and provide potential applicability of the proposed approaches.

Chapter 2

Basic methodologies

In this chapter, the author reviews some basic concepts of event-triggered and self-triggered control. Moreover, some theoretical backgrounds of standard MPC are provided. In particular, some established results of *recursive feasibility* and *asymptotic stability* of MPC are given, by following the preliminary work presented in [38]. The proofs for these analysis are useful tool to formulate event-triggered and self-triggered strategies proposed in later chapters.

2.1 Event-triggered and Self-triggered control

2.1.1 Event-triggered control

In this section, an overview of the event-triggered control is given. The basic concept of the event-triggered control is illustrated in Fig. 2.1.

As shown in the figure, the event-triggered control integrates the Event Triggering Mechanism (ETM). The ETM is responsible for deciding sampling time instants to transmit state or output measurements to the controller. Basically, the ETM evaluates the so-called event-triggered condition, which is derived from control performance or stability. To illustrate an example, let us consider the following Linear-Time-Invariant (LTI) system:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2.1)$$

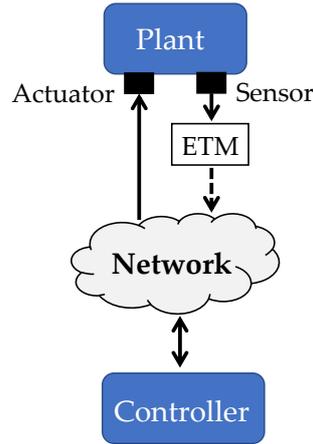


FIGURE 2.1: Event-triggered control architecture incorporating the Event-Triggering Mechanism (ETM).

where $x \in \mathbb{R}^n$ denotes the state and $u \in \mathbb{R}^m$ denotes the control input. Suppose that a static state feedback controller $u(t) = Kx(t)$ is applied, where K is chosen such that $(A + BK)$ is Hurwitz. Denote by t_k , $k \in \mathbb{N}$ the transmission time instants determined by the ETM. During the sampling time period $t \in [t_k, t_{k+1}]$, the controller is given in a zero-order-hold fashion, i.e.,

$$u(t) = Kx(t_k), \quad \forall t \in [t_k, t_{k+1}). \quad (2.2)$$

As mentioned above, the ETM determines the sampling time instants $t_k \in \mathbb{R}$ by evaluating the event-triggered condition. One possible way to derive the event-triggered condition is to evaluate the Lyapunov function candidate [7], [9], i.e., $V(x(t)) = x(t)^\top Px(t)$, where P is an appropriately chosen positive definite matrix such that $(A + BK)^\top P + P(A + BK) = -Q$ holds for a given $Q \succ 0$. Namely, sensor measurements are transmitted only when the Lyapunov function $V(x(t))$ is *not* guaranteed to decrease. In [9], the event-triggered condition has been derived based on this Lyapunov function as

$$\|x(t) - x(t_k)\| \leq \sigma \|x(t_k)\|, \quad (2.3)$$

where t denotes the current time, t_k represents the latest sampling time from

t , and $\sigma > 0$ denotes an appropriately chosen constant parameter. Namely, at the current time instant t , the ETM checks if the error between the current state measurement $x(t)$ and the previous state at the latest sampling time $x(t_k)$ exceeds a certain threshold according (2.3), and if (2.3) is *violated* at t , then the ETM sets the communication time as $t_{k+1} = t$ and transmit $x(t_{k+1})$ to the controller. It has been shown in [9] that the event-triggered condition in (2.3) yields $\dot{V}(x(t)) \leq -a\|x(t)\|^2, \forall t \in \mathbb{R}$ for some $a > 0$. Namely, the triggering condition according to (2.3) renders the closed loop system asymptotically (exponentially) stable in the sense of Lyapunov stability, while executing control inputs only when it is needed.

In NCSs, the ETM is equipped in the plant side, and it monitors the state to evaluate the condition (2.3) to determine the transmission time instants. As described in the introduction, energy expenditure of battery powered devices can be potentially saved by exchanging state and control inputs over communication network only when the event-triggered condition (2.3) is violated. In addition to Lyapunov stability as mentioned above, some other performance criteria to derive the event-triggered condition has been proposed, such as \mathcal{L}_2 and \mathcal{L}_∞ gain stability [12], [47], and Input-State-Stable (ISS) Lyapunov stability [16]. While the above example considers linear systems, event-triggered strategies for nonlinear systems have been also proposed, see e.g., [16], [18]. When full state information is not available, event-triggered mechanisms with output feedback controller are utilized, see e.g., [12].

2.1.2 Self-triggered control

In the event-triggered control, the ETM requires to monitor the sensor measurements continuously. While the continuous monitoring is available if the plant is equipped with a dedicated analog hardware, it may not be the case with a digital platform. To overcome this issue, a self-triggered paradigm has

been proposed as an alternative to the event-triggered control [7]. In the self-triggered strategy, for each communication time $t_k \in \mathbb{R}$, the controller directly determines the next communication time t_{k+1} , based on the current state measurement $x(t_k)$. Namely, t_{k+1} is determined as

$$t_{k+1} = t_k + \Gamma(x(t_k)), \quad (2.4)$$

where $\Gamma : \mathbb{R}^n \rightarrow \mathbb{R}_{>0}$ denotes a given mapping that maps the (current) state to the transmission time interval ($t_{k+1} - t_k$). In contrast to the event-triggered control, the self-triggered control directly determines the next communication time according to (2.4), which does *not* require continuous monitoring of the state. In [24], the mapping Γ is constructed by extending the event-triggered condition (2.3) for nonlinear control systems. Some other variants of the self-triggered control schemes are proposed, such as the one by evaluating ISS Lyapunov stability [23], [48]. In NCSs, the controller determines both control inputs and communication times according to (2.4), and these are transmitted over a communication network. As a consequence, both sensor and communication systems can be completely shut down, and, therefore, energy savings of battery powered devices can be achieved.

2.1.3 Event-triggered and Self-triggered MPC

Event-triggered and self-triggered Model Predictive Control (MPC) are the variants of event-triggered and self-triggered strategies, in which the controller implements a MPC framework. That is, control inputs are computed by solving an optimal control problem in an online fashion, and these are transmitted over a communication network only when it is necessary. As previously mentioned in the introduction, MPC framework is useful when control systems are subject to hard constraints, such as actuator saturations. However, it is typical in MPC framework [49], that the controller solves an optimal control problem

periodically, which may induce a high communication load and it is thus of importance to introduce the aperiodic control framework. When implementing event-triggered or self-triggered schemes, one needs to design suitable event-triggered conditions similarly to (2.3) or the mapping Γ that determines the next communication time. The key concepts for deriving these conditions are recursive feasibility and Lyapunov stability, as the details described in the next section.

2.2 Theoretical background of MPC

In this section, some theoretical results of periodic MPC are reviewed. Let us consider applying MPC to the following system:

$$\dot{x}(t) = \phi(x(t), u(t)) \quad (2.5)$$

for $t \in \mathbb{R}$, where $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $x(t) \in \mathbb{R}^n$ denotes the state, and $u(t) \in \mathbb{R}^m$ denotes the control variable. Assume that the state and control input must satisfy the following constraints:

$$x(t) \in \mathcal{X} \subseteq \mathbb{R}^n, \quad u(t) \in \mathcal{U} \subseteq \mathbb{R}^m, \quad (2.6)$$

where \mathcal{X} and \mathcal{U} are assumed to be compact, convex and contain the origin in their interiors. The following standard assumptions are made (see, e.g., [38]):

Assumption 2.1. (i) The function $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is twice continuously differentiable, and the origin is an equilibrium point, i.e., $\phi(0, 0) = 0$; (ii) The system (2.5) has a unique, absolutely continuous solution for any initial state $x(0)$ and any piecewise continuous control $u : [0, \infty) \rightarrow \mathcal{U}$.

Let $t_k = k\Delta$, $k \in \mathbb{N}_{\geq 0}$ with $t_0 = 0$ be the update time instants when OCPs are solved, where $\Delta \in \mathbb{N}_{>0}$ denotes a given sampling time. That is, at t_k the controller finds an optimal control and state trajectories by solving an optimal

control problem, based on the state measurement $x(t_k)$ and the predictive behavior according to the system dynamics in (2.5). The following cost function to be minimized is considered:

$$J(x(t_k), u(\cdot)) = \int_{t_k}^{t_k+T_p} (\|x(\xi)\|_Q^2 + \|u(\xi)\|_R^2) d\xi + \|x(t_k + T_p)\|_{P_f}^2, \quad (2.7)$$

where $Q = Q^\top \succ 0$, $R = R^\top \succ 0$ are the matrices for the stage cost, $P_f = P_f^\top$ is the terminal cost, and $T_p > 0$ denotes the prediction horizon. The Optimal Control Problem (OCP) is formulated as follows:

Problem 2.1 (Optimal Control Problem). *For any $t_k, k \in \mathbb{N}$ and given $x(t_k)$, find an optimal control and a state trajectory $u^*(\xi), x^*(\xi)$ for all $\xi \in [t_k, t_k + T]$, by minimizing the cost function $J(x(t_k), u(\cdot))$, subject to the following constraints:*

$$\begin{cases} \dot{x}(\xi) = \phi(x(\xi), u(\xi)), \quad \forall \xi \in [t_k, t_k + T_p] \end{cases} \quad (2.8)$$

$$\begin{cases} u(\xi) \in \mathcal{U}, \quad x(\xi) \in \mathcal{X}, \quad \forall \xi \in [t_k, t_k + T_p] \end{cases} \quad (2.9)$$

$$\begin{cases} x(t_k + T_p) \in \Phi, \end{cases} \quad (2.10)$$

where $\Phi = \{x \in \mathbb{R}^n : x^\top P x \leq \varepsilon^2\}$ for a given $\varepsilon > 0$. □

The first constraint in (2.8) represents the constraint that the state should follow the dynamics. The second constraint in (2.9) represents the constraints that the state and input should remain in the constraint sets \mathcal{X}, \mathcal{U} , respectively. The third constraint in (2.10) is the so-called *terminal constraint*, in which the terminal state $x(t_k + T)$ should be in Φ_f . The terminal constraint is kind of an *artificial* constraint in order to ensure asymptotic stability of the origin. The set $\Phi = \{x \in \mathbb{R}^n : x^\top P x \leq \varepsilon^2\}$ is so-called *terminal set*, which will be characterized as a local set around the origin where a stabilizing, linear state feedback controller exists according to the following assumption:

Assumption 2.2. *There exists a local controller $\kappa(x) = Kx \in \mathcal{U}$, satisfying*

$$\frac{\partial V_f}{\partial x} \phi(x, \kappa(x)) \leq -x^\top (Q + K^\top R K) x \quad (2.11)$$

for all $x \in \Phi$, where $V_f = x^\top P x$.

Assumption 2.2 assumes that the controller $\kappa(x) = Kx \in \mathcal{U}$ stabilizes the system in the sense that the Lyapunov function $V_f = x^\top P x$ is guaranteed to decrease. Therefore, the set $\Phi = \{x \in \mathbb{R}^n : x^\top P x \leq \varepsilon\}$ is an invariant set for the nonlinear systems (2.5) under the state feedback controller $\kappa(x) = Kx$. In [38], they showed that there always exist a non-empty set Φ and the corresponding stabilizing controller $\kappa(x)$ such that Assumption 2.2 is fulfilled under the assumption of stabilizability of the linearized system around the origin.

In the standard setup of periodic MPC, the optimal control input trajectory $u^*(\xi)$ is applied until the next update time $t_{k+1} = t_k + \Delta$. Thus, the closed-loop system for $t \in [t_k, t_{k+1})$ is given by

$$\dot{x}(t) = \phi(x(t), u^*(t)), \quad t \in [t_k, t_{k+1}). \quad (2.12)$$

Note that since there exists no disturbance, the actual state at t_{k+1} corresponds to the optimal predictive state at t_{k+1} , i.e., we have $x(t_{k+1}) = x^*(t_{k+1})$.

2.2.1 Feasibility and stability

Having formulated the basic problem set-up, some established results are reviewed that have been analysed in the literature; namely, *recursive feasibility* and *stability*. The concept of recursive feasibility states that the existence of a feasible solution to the OCP at the initial time t_0 implies the feasibility for all the update times afterwards t_k , $k \in \mathbb{N}_{>0}$. The concept of stability states that the system is stabilized towards the origin asymptotically (i.e., $x(t_k) \rightarrow 0$ as $k \rightarrow \infty$). As will be seen later, this can be achieved by showing that the optimal cost as a Lyapunov function candidate is guaranteed to decrease.

Theorem 2.1 (Recursive feasibility [38]). *Suppose that Assumption 2.2 holds and the OCP defined in Problem 2.1 has a solution at t_k , providing an optimal control*

input $u^*(\xi)$ and the corresponding state trajectory $x^*(\xi)$ for all $\xi \in [t_k, t_k + T]$. Then, Problem 2.1 has a solution at $t_{k+1} = t_k + \Delta$. \square

Proof. The detailed proof is given in [38] and the overview is described below. Consider the following dual mode controller as a feasible control candidate:

$$\bar{u}(\xi) = \begin{cases} u^*(\xi), & \xi \in [t_{k+1}, t_k + T_p] \\ \kappa(\bar{x}(\xi)), & \xi \in (t_k + T, t_{k+1} + T_p], \end{cases} \quad (2.13)$$

where $\kappa(\cdot)$ denotes the state-feedback controller defined in Assumption 2.2 and let $\bar{x}(\xi)$, $\xi \in [t_{k+1}, t_{k+1} + T_p]$ be given by the corresponding state trajectory by applying $\bar{u}(\xi)$, $\xi \in [t_{k+1}, t_{k+1} + T_p]$. To show that the controller in (2.13) provides a feasible control input for Problem 2.1, it is required to show that $\bar{u}(\xi) \in \mathcal{U}$, $\bar{x}(\xi) \in \mathcal{X}$ for all $\xi \in [t_{k+1}, t_{k+1} + T_p]$ and $\bar{x}(t_k + T_p) \in \Phi$. It is trivially shown that $\bar{u}(\xi) \in \mathcal{U}$, $\forall \xi \in [t_{k+1}, t_k + T]$ since we have $u^*(\xi) \in \mathcal{U}$, $\forall \xi \in [t_{k+1}, t_k + T]$. Since $x(t_{k+1}) = x^*(t_{k+1})$ and $\bar{u}(\xi) = u^*(\xi)$, for all $\xi \in [t_{k+1}, t_k + T]$, we have $\bar{x}(\xi) = x^*(\xi)$ for all $\xi \in [t_{k+1}, t_k + T_p]$. Thus, we have $\bar{x}(\xi) \in \mathcal{X}$, $\forall \xi \in [t_{k+1}, t_k + T_p]$ and $\bar{x}(t_k + T_p) \in \Phi_f$ from the terminal constraint given by (2.10). Since the local state feedback controller $\kappa(x(\xi))$ is applied for all $\xi \in (t_k + T, t_{k+1} + T_p]$ and Φ_f is an invariant set under the controller $\kappa(x(\xi))$, we have $\bar{x}(\xi) \in \Phi$, $\forall \xi \in (t_k + T_p, t_{k+1} + T_p]$. Thus, we obtain $\bar{x}(\xi) \in \Phi \subseteq \mathcal{X}$, $\forall \xi \in (t_k + T, t_{k+1} + T_p]$ and $\bar{x}(t_{k+1} + T_p)$. Therefore, applying the controller $\bar{u}(\xi)$, $\xi \in [t_{k+1}, t_{k+1} + T_p]$ in (2.13) and the corresponding state $\bar{x}(\xi)$, $\xi \in [t_{k+1}, t_{k+1} + T_p]$ satisfy all the constraints imposed in Problem 2.1. This completes the proof. \square

Theorem 2.1 states that the feasibility of Problem 2.1 at t_k implies the feasibility at t_{k+1} . This implies that Problem 2.1 is feasible for all t_k , $k \in \mathbb{N}$ as long as Problem 2.1 is feasible at the initial time t_0 . In order to guarantee the feasibility at t_0 , the prediction horizon T_p needs to be suitably chosen such that the terminal constraint $x(t_0 + T_p) \in \Phi$ is fulfilled. More specifically, T_p should be selected

to satisfy $x(0) \in \mathcal{X}(T_p)$, where

$$\mathcal{X}(T_p) = \{x(t_0) \in \mathbb{R}^n \mid \exists u(t) \in \mathcal{U}, t \in [0, T_p] : x(T_p) \in \Phi\}, \quad (2.14)$$

i.e., $\mathcal{X}(T_p)$ denotes the set of states that is reachable to Φ_f within the time T_p . For linear systems, there exist several methodologies to compute the reachability set $\mathcal{X}(T_p)$, see e.g., [50]. Although there may not exist a general framework to compute $\mathcal{X}(T_p)$ explicitly for nonlinear systems, several approximation methods have been proposed to compute $\mathcal{X}(T_p)$, see e.g., [51].

The following theorem illustrates asymptotic stability of the origin:

Theorem 2.2 (Stability [38]). *Suppose that Assumption 2.2 holds and the OCP defined in Problem 2.1 has a solution at t_0 . Then, the closed loop state trajectory is asymptotically stabilized towards the origin, i.e., $x(t_k) \rightarrow 0$ as $k \rightarrow \infty$.*

Proof. The detailed proof is given in [38] and the overview is described below. Consider the dual mode controller given by (2.13). From Theorem 2.1, the controller is feasible for the optimal control problem at t_{k+1} . Let $\bar{J}(x(t_{k+1}))$ be the corresponding cost obtained by applying (2.13), i.e.,

$$\bar{J}(x(t_{k+1})) = \int_{t_k}^{t_k+T_p} (\|\bar{x}(\xi)\|_Q^2 + \|\bar{u}(\xi)\|_R^2) d\xi + \|\bar{x}(t_k + T_p)\|_{P_f}^2. \quad (2.15)$$

Now, consider the cost difference $\bar{J}(x(t_{k+1})) - J^*(x(t_k))$. We obtain

$$\begin{aligned} & \bar{J}(x(t_{k+1})) - J^*(x(t_k)) \\ & \leq \int_{t_{k+1}}^{t_k+T_p} (\|\bar{x}(\xi)\|_Q^2 - \|x^*(\xi)\|_Q^2 + \|\bar{u}(\xi)\|_R^2 - \|u^*(\xi)\|_R^2) d\xi \\ & \quad - \int_{t_k}^{t_{k+1}} (\|x^*(\xi)\|_Q^2 + \|u^*(\xi)\|_R^2) d\xi - \|x^*(t_k + T_p)\|_{P_f}^2 + \|\bar{x}(t_{k+1} + T_p)\|_{P_f}^2. \end{aligned}$$

Since $x(t_{k+1}) = x^*(t_{k+1})$ and $\bar{u}(\xi) = u^*(\xi)$, for all $\xi \in [t_{k+1}, t_k + T]$, we have $\bar{x}(\xi) = x^*(\xi)$ for all $\xi \in [t_{k+1}, t_k + T_p]$. Moreover, from Assumption 2.2 we have

$\|\bar{x}(t_{k+1} + T_p)\|_{P_f}^2 - \|x^*(t_k + T_p)\|_{P_f}^2 \leq 0$. Therefore, we obtain

$$\begin{aligned} J^*(x(t_{k+1})) - J^*(x(t_k)) &\leq \bar{J}(x(t_{k+1})) - J^*(x(t_k)) \\ &\leq - \int_{t_k}^{t_{k+1}} (\|x^*(\xi)\|_Q^2 + \|u^*(\xi)\|_R^2) d\xi. \end{aligned} \quad (2.16)$$

Now, we have

$$\begin{aligned} J^*(x(t_1)) - J^*(x(t_0)) &\leq - \int_{t_0}^{t_1} \|x^*(\xi)\|_Q^2 d\xi \\ J^*(x(t_2)) - J^*(x(t_1)) &\leq - \int_{t_1}^{t_2} \|x^*(\xi)\|_Q^2 d\xi \\ &\vdots \end{aligned}$$

Summing over both sides of the above yields

$$\int_{t_0}^{\infty} x^\top(t) Q x(t) dt < J^*(x(t_0)) - J^*(x(\infty)) < \infty.$$

Since the function $x(t)^\top Q x(t)$ is uniformly continuous on $t \in [0, \infty)$ and $Q \succ 0$, we obtain $\|x(t)\| \rightarrow 0$ as $t \rightarrow \infty$ from Barbalat's lemma, see, e.g., [52]. This completes the proof. \square

2.2.2 Discussions

So far, a basic problem setup of periodic MPC and theoretical results of feasibility and stability are provided. Some discussions on these results and the problem setup are in order as follows. First, let us recall that the control scheme is assumed to apply the optimal control trajectory until the next update time. Namely, if the optimal control trajectory $u^*(\xi)$, $\xi \in [t_k, t_k + T]$ is obtained for some t_k , we set $u(t) = u^*(t)$ for all $t \in (t_k, t_{k+1}]$ (see (5.12)). While this scheme may be useful to guarantee stability of the origin as provided in Theorem 2.2, it may not be applicable in the practical implementation, since applying the *continuous* control trajectory requires a dedicated analog hardware so that the control input can be updated continuously. Moreover, when applying the control

scheme in networked control systems, it is required that the controller needs to transmit the continuous trajectory $u(t) = u^*(t)$ for all $t \in (t_k, t_{k+1}]$, which requires an *infinite* communication bandwidth that is physically un-realizable. In view of this problem, it may be more preferable to apply the control input in a *sample-and-hold fashion*, rather than apply the continuous trajectory. Namely, control input is applied as $u(t) = u^*(t_k)$ for all $t \in (t_k, t_{k+1}]$. Stability analysis of MPC under a sample-and-hold controller may be easily handled for Linear-Time-Invariant (LTI) systems, since we can simply obtain a discrete-time model from the continuous one via standard discretization schemes (for details, see Chapter 3) and we can analyse stability for the corresponding discretized model. For nonlinear systems, however, applying MPC under a sample-and-hold controller requires a more detailed analysis and additional constraints for guaranteeing stability. For details, see Chapter 4 for the derivation of stability condition and the corresponding aperiodic control strategies.

Another challenging aspect is to analyze the *robustness*, which shows how much model uncertainties or disturbance can affect the control performance and stability. To illustrate the motivation for this analysis, suppose that Problem 2.1 is solved at t_k , which provides the optimal control $u^*(\xi)$ and the corresponding state trajectory $x^*(\xi)$ for all $\xi \in [t_k, t_k + T_p]$. If no model uncertainties or disturbances are present in the dynamics in (2.5), the actual state at $t \in [t_k, t_{k+1})$ corresponds to the predictive state, i.e., $x(t) = x^*(t)$ for all $t \in [t_k, t_{k+1})$. When the model uncertainties or disturbances are present, on the other hand, the resulting actual state $x(t)$ is no longer equal to $x^*(t)$. One of the interesting analysis is, therefore, to analyze how much the error between predictive state and actual state is tolerated to guarantee stability. In Chapter 5, the author considers nonlinear systems that are perturbed by additive bounded disturbances, and derive such upper bound to guarantee stability and the corresponding event and self-triggered strategies.

2.3 Summary

In this chapter, some basic concepts of event-triggered and self-triggered control as well as some established results (recursive feasibility and asymptotic stability) of MPC are reviewed. Regarding feasibility, it is shown that the existence of the feasible solution to the optimal control problem implies the same for the next update time. For stability, it is shown that the optimal cost as a Lyapunov function candidate is guaranteed to decrease.

Chapter 3

Aperiodic MPC for linear systems

First of all, a control problem of Linear-Time-Invariant (LTI) systems is given to formulate aperiodic formulations of MPC. Two aperiodic MPC schemes are proposed, i.e., *multiple discretizations approach* (Chapter 3.1), and *contractive set-based approach* (Chapter 3.2). In both control schemes, the triggering strategies are proposed in a self-triggered manner, in which the controller side regulates the communication times. The two self-triggered schemes are different in the sense that latter approach incorporates the notion of contractive sets (while the former does not). As will be seen later, incorporating the contractive set in the latter case leads to the reduction of computational complexity compared to the former case, while, on the other hand, it may shrink the domain of attraction. In both cases, asymptotic stability of the origin and feasibility of the optimal control problem are theoretically shown. Also, some numerical simulations are illustrated to validate the control schemes.

3.1 Multiple discretizations approach

3.1.1 Problem formulation

Consider a networked control system depicted in Fig. 1.1. The dynamics of the plant are assumed to be given by the following LTI system:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (3.1)$$

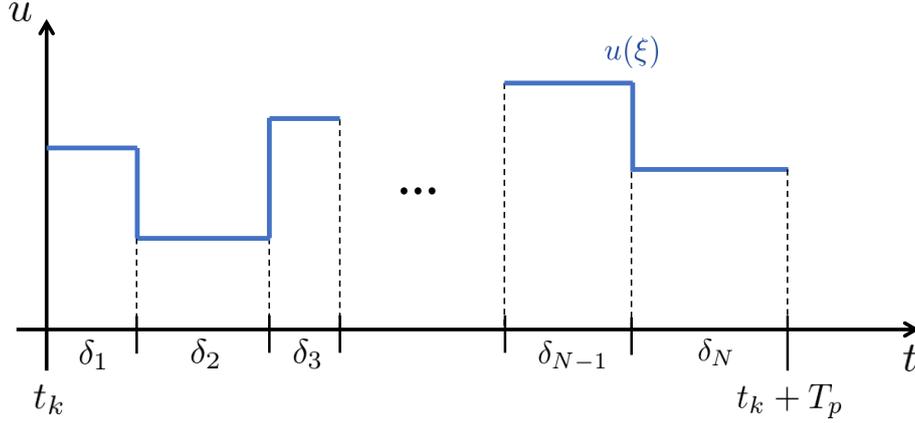


FIGURE 3.1: Illustration of the piece-wise constant control policy.

where $x(t) \in \mathbb{R}^n$ is the state and $u(t) \in \mathbb{R}^m$ is the control variable. The state and control input are assumed to be constrained as $x(t) \in \mathcal{X}$, $u(t) \in \mathcal{U}$, $\forall t \in \mathbb{R}$, where \mathcal{X} and \mathcal{U} are convex, compact and contain the origin in the interiors.

Definition 3.1 (Control objective). *The control objective for the MPC is to drive the state to the origin, i.e., $x(t) \rightarrow 0$, as $t \rightarrow \infty$.*

In the following, let $t_0 < t_1 < t_2 < \dots$ be the transmission time instants between the plant and the controller. Namely, at t_k , $k \in \mathbb{N}$, the plant transmits the state information $x(t_k)$ to the controller, and the controller solves an optimal control problem (OCP) based on the dynamics given by (3.1). The following cost function to be minimized is given:

$$J(x(t_k), u(\cdot)) = \int_{t_k}^{t_k + T_p} (\|\hat{x}(\xi)\|_Q^2 + \|u(\xi)\|_R^2) d\xi + \|x(t_k + T_p)\|_{P_f}^2, \quad (3.2)$$

where $Q \succ 0$, $R \succ 0$ are the matrices for the stage cost, $P_f^\top = P_f \succ 0$ is the matrix for the terminal cost, and $T_p > 0$ is the prediction horizon. More detailed characterization of P_f will be discussed in later sections.

In order to derive a self-triggered strategy, let us first consider that the control input $u(\xi)$, $\xi \in [t_k, t_k + T_p]$ is constrained to be piece-wise constant with different sampling intervals, $\delta_1, \delta_2, \dots, \delta_N$, as shown in Fig. 3.1. This discretizing scheme is motivated as follows. The solution of the OCP by minimizing

the cost (3.2) is in general given by a continuous trajectory of the optimal control input, say $u^*(\xi)$, for all $\xi \in [t_k, t_k + T_p]$ (see the problem formulation in Chapter 2). If the optimal control input *could* be applied until t_{k+1} , i.e., $u^*(\xi)$, $\xi \in [t_k, t_{k+1})$, then we could utilize the classic MPC result to guarantee the asymptotic stability of the origin. However, applying the continuous trajectory of the control input is not suited for practical NCSs applications in terms of the two aspects. First, transmitting continuous control trajectory over the network requires an infinite-transmission bandwidth, which is un-realizable. Second, implementing the exact continuous control input is difficult for embedded control system architectures, since they only deal with samples as a discrete time domain, resulting in applying the control input eventually as a sampled-and-hold implementation at a high frequency. As the actual control trajectory for this case possibly differs from the optimal control trajectory, it fails to guarantee the asymptotic stability of the origin.

The OCP under the piece-wise constant control policy considered in this chapter thus provides the optimal control sequence at discrete sampling intervals, i.e., $\{u^*(t_k), u^*(t_k + \delta_1), \dots, u^*(t_k + \sum_{j=1}^N \delta_j)\}$ rather than the whole control trajectory $u^*(\xi)$, $\xi \in [t_k, t_k + T_p]$. As the procedure of transmitting control samples, the following steps are considered; (i) the controller transmits the optimal control sample $u^*(t_k)$ to the plant; (ii) the plant then applies $u^*(t_k)$ at constant until $t_{k+1} = t_k + \delta_1$; (iii) the plant sends back a new state measurement $x(t_{k+1})$ to the controller to solve the next OCP at t_{k+1} . Under the procedure described above, the transmission time interval is then given by $t_{k+1} - t_k = \delta_1$.

Applying the above transmission procedure not only allows the controller to transmit control command as a sample, but also allows us to formulate the OCP in the discrete time domain. The main difference of the problem formulation with respect to the periodic (or event-triggered) MPC for general discrete time systems is, however, that we are now free to select the sampling time intervals $\delta_1, \dots, \delta_N$ in an appropriate way. Although there is a flexibility to select $\delta_1, \dots, \delta_N$, these intervals must be carefully determined such that:

- (i) The asymptotic stability of the origin is guaranteed under MPC with the piece-wise constant control policy.
- (ii) The reduction of communication load is achieved through the self-triggered formulation.

In the following, one possible way to determine the sampling time intervals $\delta_1, \dots, \delta_N$ is given such that the above problems can be tackled. By making use of the flexibility of selecting the sampling time intervals, consider at first that we have *multiple patterns* of sampling time intervals, i.e., we have M ($M \in \mathbb{N}_{\geq 1}$) different sampling patterns in total, where each i -th ($i \in \{1, 2, \dots, M\}$) sampling pattern has N_i sampling intervals, $\delta_1^{(i)}, \delta_2^{(i)}, \dots, \delta_{N_i}^{(i)}$. More specifically, in this thesis the sampling patterns shown in Fig. 3.2 are given. Stated formally, for given $M, N_p \in \mathbb{N}_{\geq 1}$, where $M < N_p$ and N_p represents the maximum number of sampling intervals among all patterns, and $\delta = T_p/N_p$, the sampling time intervals for the i -th ($i \in \{1, 2, \dots, M\}$) pattern are given by

$$\delta_1^{(i)} = i\delta, \quad \delta_j^{(i)} = \delta, \quad j = 2, 3, \dots, N_i, \quad (3.3)$$

with $N_i = N_p - i + 1$. That is, the 1st pattern has the same interval: $\delta_1^{(1)} = \dots = \delta_{N_p}^{(1)} = \delta$. The 2nd pattern is the same as the 1st pattern only except the first sampling interval: $\delta_1^{(2)} = 2\delta, \delta_2^{(2)} = \dots = \delta_{N_p-1}^{(2)} = \delta$. Similarly, for the general i -th pattern we have $\delta_1^{(i)} = i\delta$, and δ for the remaining intervals. The controller solves the corresponding OCPs under all sampling patterns above, and then selects one sampling pattern according to the self-triggered strategy proposed in the next section.

The main motivation of using the sampling patterns shown in Fig. 3.2, is that it allows to evaluate the trade-off between the transmission interval and the control performance quantitatively. According to the transmission procedure given in the previous subsection, the transmission time interval is given by $\delta_1^{(i)} = i\delta$. Thus, using larger patterns leads to longer transmission intervals. From the self-triggered point of view, it is desirable to have larger patterns.

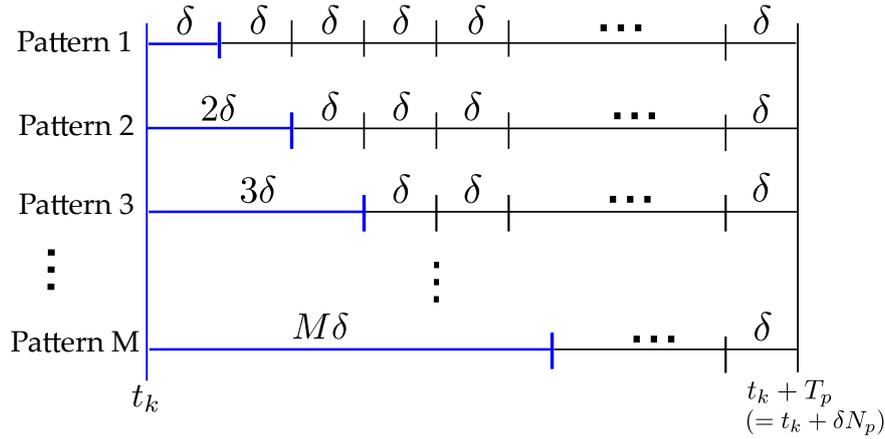


FIGURE 3.2: Sampling patterns considered in this paper. Blue lines represent the transmission time intervals.

However, as we will see in the analysis that follows, the control performance instead becomes worse; this will be proved by the fact that the optimal cost becomes larger as larger patterns are selected. In later sections, the author will provide a framework of selecting one sampling pattern, such that the trade-off between the transmission time interval and the control performance can be taken into account.

For the i -th sampling pattern, denote

$$\mathbf{u}_i(t_k) = \{u_i(t_k), u_i(t_k + i\delta), u_i(t_k + (i+1)\delta), \dots, u_i(t_k + (N_p - 1)\delta)\} \quad (3.4)$$

as the control input sequence to be applied. Note that $u_i(t_k + i\delta)$ is used after $u_i(t_k)$, as $u_i(t_k)$ is applied for the time interval $i\delta$. The cost given by (3.2) under the i -th sampling pattern can be re-written as

$$\begin{aligned} J_i(x(t_k), \mathbf{u}_i(t_k)) &= \int_0^{i\delta} \{ \|x(t_k + \xi)\|_Q^2 + \|u_i(t_k)\|_R^2 \} d\xi \\ &\quad + \sum_{n=i}^{N_p-1} \int_0^{\delta} \{ \|x(t_k + n\delta + \xi)\|_Q^2 + \|u_i(t_k + n\delta)\|_R^2 \} \\ &\quad + \|x(t_k + N_p\delta)\|_{P_f}^2, \end{aligned}$$

where the total cost is separated by each component of the control sequence

$\mathbf{u}_i(t_k)$. Here it is denoted as J_i instead of J to emphasize that the piece-wise constant control policy under the i -th sampling pattern is used. By computing each integral in the above equation, the total cost for the i -th sampling pattern can be translated into a summation of costs:

$$J_i(x(t_k), \mathbf{u}_i(t_k)) = F(x(t_k), u_i(t_k), i\delta) + \sum_{n=i}^{N_p-1} \{F(x(t_k + n\delta), u_i(t_k + n\delta), \delta)\} \\ + \|x(t_k + N_p\delta)\|_{P_f}^2,$$

where $F(x(t), u(t), i\delta)$ denotes a new stage cost given by

$$F(x(t), u(t), i\delta) = \int_0^{i\delta} \|x(t + \xi)\|_Q^2 + \|u(t)\|_R^2 d\xi \\ = \tilde{x}(t)^\top \Gamma(i\delta) \tilde{x}(t),$$

where $\tilde{x}(t) = [x^\top(t) \ u^\top(t)]^\top$ and

$$\Gamma(i\delta) = \begin{bmatrix} \int_0^{i\delta} A_\xi^\top Q A_\xi d\xi & \int_0^{i\delta} B_\xi^\top Q A_\xi d\xi \\ \int_0^{i\delta} A_\xi^\top Q B_\xi d\xi & \int_0^{i\delta} (B_\xi^\top Q B_\xi + R) d\xi \end{bmatrix}$$

with $A_\xi = e^{A\xi}$, $B_\xi = \int_0^\xi e^{A\tau} d\tau B$. The OCP for the i -th sampling pattern is now formulated as follows.

Problem 3.1 (Optimal Control Problem for i). *For given $x(t_k) \in \mathcal{X}$ and $i \in \mathcal{M}$, find a sequence of control inputs $\mathbf{u}_i(t_k) = \{u_i(t_k), u_i(t_k + i\delta), \dots, u_i(t_k + (N_p - 1)\delta)\}$ and the corresponding sequence of states $\mathbf{x}_i(t_k) = \{x_i(t_k), x_i(t_k + i\delta), \dots, x_i(t_k +$*

$N_p\delta\}$, by minimizing the cost $J_i(x(t_k), \mathbf{u}_i(t_k))$, subject to the following constraints:

$$\begin{cases} x(t_k + i\delta) = A_{i\delta}x(t_k) + B_{i\delta}u_i(t_k) \end{cases} \quad (3.5)$$

$$\begin{cases} x(t_k + (n+1)\delta) \\ = A_\delta x(t_k + n\delta) + B_\delta u_i(t_k + n\delta), \quad \forall n \in \{i, i+1, \dots, N_p-1\} \end{cases} \quad (3.6)$$

$$\begin{cases} x_i(t_k + n\delta) \in \mathcal{X}, \quad \forall n \in \{i, i+1, \dots, N_p\} \end{cases} \quad (3.7)$$

$$\begin{cases} u_i(t_k + n\delta) \in \mathcal{U}, \quad \forall n \in \{0, i, i+1, \dots, N_p-1\} \end{cases} \quad (3.8)$$

$$\begin{cases} x(t_k + N_p\delta) \in \Phi. \end{cases} \quad (3.9)$$

□

The constraints (3.5) and (3.6) represent the dynamics by applying the control sequence $\mathbf{u}_i(t_k)$, and (3.7), (3.8) represent the constraints for the state and the control input. The last constraint (3.9) represents the terminal state penalty, where $\Phi = \{x \in \mathbb{R}^n : x^\top P_f x \leq \varepsilon\}$ for a given $\varepsilon > 0$. Let

$$\begin{aligned} \mathbf{u}_i^*(t_k) &= \{u_i^*(t_k), u_i^*(t_k + i\delta), \dots, u_i^*(t_k + (N_p - 1)\delta)\} \\ \mathbf{x}_i^*(t_k) &= \{x_i^*(t_k), x_i^*(t_k + i\delta), \dots, x_i^*(t_k + N_p\delta)\} \end{aligned}$$

be the optimal control and the corresponding state sequence with $x_i^*(t_k) = x(t_k)$, obtained by solving Problem 3.1. Further denote $J_i^*(x(t_k)) = J_i(x(t_k), \mathbf{u}_i^*(t_k))$ as the optimal cost.

Similarly to Assumption 2.2, consider that the matrix P_f and ε are chosen such that the following condition on the terminal region Φ is satisfied:

Assumption 3.1. *There exists a local state feedback controller $\kappa(x) = Kx \in \mathcal{U}$, satisfying*

$$x(t_k + \delta)^\top P_f x(t_k + \delta) - x^\top(t_k) P_f x(t_k) \leq -F(x(t_k), Kx(t_k), \delta) \quad (3.10)$$

for all $x(t_k) \in \Phi$, where $x(t_k + \delta) = (A_\delta + B_\delta K)x(t_k)$. □

Assumption 3.1 will be used to guarantee that the optimal cost decreases along the time by an appropriate selection of the sampling pattern. Since the

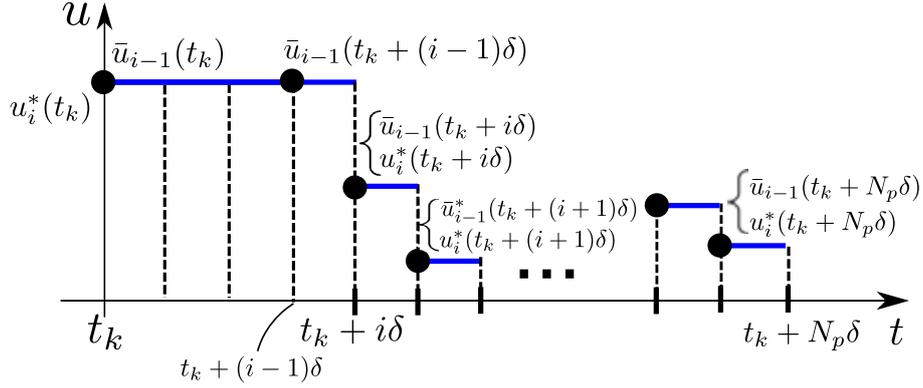


FIGURE 3.3: Optimal piecewise constant control policy for the i -th sampling pattern (blue line) and the admissible control sequence for the $(i-1)$ -th pattern $\bar{u}_{i-1}(t_k)$ (black circles).

system (3.1) is assumed to be stabilizable, the local controller $\kappa(x)$ and Φ satisfying (3.10), can be found off-line by following the procedure presented in, e.g., [38]. To arrive at the self-triggered strategy, we will in the following derive some useful properties for the optimal costs obtained under different sampling patterns. These properties are key ingredients to quantify the control performances for the self-triggered strategy, as well as for the asymptotic stability provided in later sections.

Lemma 3.1. *Suppose that Problem 3.1 admits a solution at t_k under each sampling pattern $i \in \{1, 2, \dots, M\}$, which provides the optimal costs $J_i^*(x(t_k))$ for all $i \in \{1, \dots, M\}$. Then we have*

$$J_1^*(x(t_k)) \leq J_2^*(x(t_k)) \leq J_3^*(x(t_k)) \cdots \leq J_M^*(x(t_k)). \quad (3.11)$$

□

Proof. Let $\mathbf{u}_i^*(t_k)$, $\mathbf{x}_i^*(t_k)$, $i \in \{1, 2, \dots, M\}$ be the optimal control and the corresponding state sequence obtained by Problem 3.1 under the i -th sampling pattern. The illustration of the corresponding optimal piece-wise constant control policy is depicted in Fig. 3.3. Under the i -th ($i \geq 2$) sampling pattern, $u_i^*(t_k)$ is applied at constant for all $t \in [t_k, t_k + i\delta)$ as shown in Fig. 3.3. The control policy for the i -th ($i \geq 2$) sampling pattern is thus admissible also for the $(i-1)$ -th

sampling pattern, as $u_i^*(t_k)$ is applied for $t \in [t_k, t_k + (i-1)\delta) \in [t_k, t_k + i\delta)$. More specifically, let

$$\bar{\mathbf{u}}_{i-1}(t_k) = \{\bar{u}_{i-1}(t_k), \bar{u}_{i-1}(t_k + (i-1)\delta), \dots, \bar{u}_{i-1}(t_k + (N_p - 1)\delta)\},$$

where $\bar{u}_{i-1}(t_k) = u_i^*(t_k)$, $\bar{u}_{i-1}(t_k + (i-1)\delta) = u_i^*(t_k)$ and

$$\bar{u}_{i-1}(t_k + j\delta) = u_i^*(t_k + j\delta), \quad j = i, \dots, N_p - 1,$$

and $\bar{\mathbf{x}}_{i-1}(t_k) = \{\bar{x}_{i-1}(t_k), \bar{x}_{i-1}(t_k + (i-1)\delta), \dots, \bar{x}_{i-1}(t_k + N_p\delta)\}$ be the corresponding state sequence with $\bar{x}_{i-1}(t_k) = x(t_k)$ (see the illustration of $\bar{\mathbf{u}}_{i-1}$ in Fig. 3.3). Then, $\bar{\mathbf{u}}_{i-1}(t_k)$ provides a feasible solution to Problem 1 under the $(i-1)$ -th pattern, satisfying all constraints (3.5), (3.6), (3.8) and (3.9). The last constraint (3.9) is obtained by the fact that $\bar{x}_{i-1}(t_k + N_p\delta) = x_i^*(t_k + N_p\delta) \in \Phi$. Since $\bar{\mathbf{u}}_{i-1}$ is a feasible controller for the $(i-1)$ -th pattern, we obtain

$$\begin{aligned} J_{i-1}^*(x(t_k)) &\leq J_{i-1}(x(t_k), \bar{\mathbf{u}}_{i-1}(t_k)) \\ &= J_i(x(t_k), \mathbf{u}_i^*(t_k)) \\ &= J_i^*(x(t_k)), \end{aligned} \tag{3.12}$$

and the above inequality holds for all $i \in \{2, 3, \dots, M\}$. The proof is thus complete. \square

Lemma 3.1 states that the 1st pattern provides the best control performance in the sense that the optimal cost takes the minimum value among all patterns, and moreover, the control performance becomes worse as larger patterns are selected. The next lemma states that the optimal cost is guaranteed to decrease whenever the 1st pattern is used:

Lemma 3.2. *Suppose that the i -th pattern was used at t_{k-1} and the next time to solve the OCP is given by $t_k = t_{k-1} + i\delta$. Then, under Assumption 3.1, the optimal cost*

satisfies

$$J_1^*(x(t_k)) - J_i^*(x(t_{k-1})) \leq -F(x(t_{k-1}), u_i^*(t_{k-1}), i\delta). \quad (3.13)$$

□

Proof. Let

$$\mathbf{u}_i^*(t_{k-1}) = \{u_i^*(t_{k-1}), u_i^*(t_k), \dots, u_i^*(t_k + (N_p - i - 1)\delta)\}$$

$$\mathbf{x}_i^*(t_{k-1}) = \{x_i^*(t_{k-1}), x_i^*(t_k), \dots, x_i^*(t_k + (N_p - i)\delta)\}$$

be the optimal control input and the corresponding state sequence obtained at t_{k-1} under the i -th pattern. From the constraint (3.9), we have $x_i^*(t_k + (N_p - i)\delta) \in \Phi$. At t_k , consider the following control and the corresponding state sequence for the 1st pattern; $\bar{\mathbf{u}}_1(t_k) = \{\bar{u}_1(t_k), \bar{u}_1(t_k + \delta), \dots, \bar{u}_1(t_k + (N_p - 1)\delta)\}$, $\bar{\mathbf{x}}_1(t_k) = \{\bar{x}_1(t_k), \bar{x}_1(t_k + \delta), \dots, \bar{x}_1(t_k + N_p\delta)\}$, where each component of $\bar{\mathbf{u}}_1(t_k)$ is given by

$$\bar{u}_1(t_k + j\delta) = \begin{cases} u_i^*(t_k + j\delta), & \text{for } j = 0, \dots, N_p - i - 1 \\ \kappa(\bar{x}_1(t_k + j\delta)), & \text{for } j = N - i, \dots, N_p - 1. \end{cases} \quad (3.14)$$

Applying the local controller κ from $t_k + (N_p - i)\delta$ is admissible since we have $\bar{x}_1(t_k + (N_p - i)\delta) = x_i^*(t_k + (N_p - i)\delta) \in \Phi$. Thus $\bar{\mathbf{u}}_1(t_k)$ is a feasible controller for Problem 1 under the 1st sampling pattern, and the upper bound of the difference between $J_1^*(x(t_k))$ and $J_i^*(x(t_{k-1}))$ is given by

$$J_1^*(x(t_k)) - J_i^*(x(t_{k-1})) \leq J_1(x(t_k), \bar{\mathbf{u}}_1(t_k)) - J_i(x(t_{k-1}), \mathbf{u}_i^*(t_{k-1})). \quad (3.15)$$

Some calculations of the right hand side in (3.15) yield (3.13). The derivation is given as follows. The optimal cost for the i -th pattern at t_{k-1} is given by

$$\begin{aligned} J_i(x(t_{k-1}), \mathbf{u}_i^*(t_{k-1})) &= F(x(t_{k-1}), u_i^*(t_{k-1}), i\delta) + F(x_i^*(t_k), u_i^*(t_k), \delta) \\ &\quad + \sum_{n=1}^{N_p-i-1} F(x_i^*(t_k + n\delta), u_i^*(t_k + n\delta), \delta) \\ &\quad + \|x_i^*(t_k + (N_p - i)\delta)\|_{P_f}^2. \end{aligned}$$

Furthermore, the cost at t_k under the 1st sampling pattern with $\bar{\mathbf{u}}_1(t_k)$ in (3.14), is given by

$$\begin{aligned} J_1(x(t_k), \bar{\mathbf{u}}_1(t_k)) &= F(x(t_k), \bar{u}_1(t_k), \delta) + \sum_{n=1}^{N_p-1} F(\bar{x}_1(t_k + n\delta), \bar{u}_1(t_k + n\delta), \delta) \\ &\quad + \|\bar{x}_1(t_k + N_p\delta)\|_{P_f}^2. \end{aligned}$$

From (3.14), we have $\bar{u}_1(t_k + j\delta) = u_i^*(t_k + j\delta)$ for $j = 0, \dots, N_p - i - 1$, and thus $\bar{x}_1(t_k + j\delta) = x_i^*(t_k + j\delta)$ for $j = 0, \dots, N_p - i$. The difference between $J_1(x(t_k), \bar{\mathbf{u}}_1(t_k))$ and $J_i(x(t_{k-1}), \mathbf{u}_i^*(t_{k-1}))$, which is denote as $\Delta J_k = J_1(x(t_k), \bar{\mathbf{u}}_1(t_k)) - J_i(x(t_{k-1}), \mathbf{u}_i^*(t_{k-1}))$ is then given by

$$\begin{aligned} \Delta J_k &= -F(x(t_{k-1}), u_i^*(t_{k-1}), i\delta) + \sum_{n=N_p-i}^{N_p-1} F(\bar{x}_1(t_k + n\delta), \kappa(\bar{x}_1(t_k + n\delta), \delta) \\ &\quad - \|\bar{x}_1(t_k + (N_p - i)\delta)\|_{P_f}^2 + \|\bar{x}_1(t_k + N_p\delta)\|_{P_f}^2. \end{aligned}$$

From (3.10), we have $\|\bar{x}_1(t_k + N_p\delta)\|_{P_f}^2 - \|\bar{x}_1(t_k + (N_p - 1)\delta)\|_{P_f}^2 \leq -F(\bar{x}_1(t_k + (N_p - 1)\delta), \kappa(\cdot), \delta)$. By using this inequality, we obtain

$$\begin{aligned} \Delta J_k &\leq -F(x(t_{k-1}), u_i^*(t_{k-1}), i\delta) + \sum_{n=N_p-i}^{N_p-2} F(\bar{x}_1(t_k + n\delta), \kappa(\bar{x}_1(t_k + n\delta), \delta) \\ &\quad - \|\bar{x}_1(t_k + (N_p - i)\delta)\|_{P_f}^2 + \|\bar{x}_1(t_k + (N_p - 1)\delta)\|_{P_f}^2. \end{aligned}$$

Similarly above, by recursively using the inequality from (3.10), we obtain $\|\bar{x}_1(t_k + (N_p - j - 1)\delta)\|_{P_f}^2 - \|\bar{x}_1(t_k + (N_p - j - 1)\delta)\|_{P_f}^2 \leq -F(\bar{x}_1(t_k + (N_p - j - 1)\delta), \kappa(\cdot), \delta)$.

1) δ), $\kappa(\cdot, \delta)$. for $j \in \{1, 2, \dots, i-1\}$, and thus

$$J_1(x(t_k), \bar{\mathbf{u}}_1(t_k)) - J_i(x(t_{k-1}), \mathbf{u}_i^*(t_{k-1})) \leq -F(x(t_{k-1}), u_i^*(t_{k-1}), i\delta),$$

and this yields (3.13), completing the proof. \square

3.1.2 Self-triggered strategy

In this section the author proposes the self-triggered strategy. The key idea of the framework is to select the best pattern in the sense that it provides the largest possible transmission time interval, while satisfying some conditions to obtain the desired control performance. In the following proposed algorithm, denote $i_k, k \in \mathbb{N}$ as the sampling pattern selected by the controller to transmit the corresponding optimal control sample $u_{i_k}^*(t_k)$.

Algorithm 3.1: (Self-triggered MPC via multiple-discretizations approach)

Initialization : At the initial time t_0 , the controller solves Problem 1 only for $i = 1$, based on $x(t_0)$. The controller then transmits the optimal control sample $u_1^*(t_0)$ to the plant, i.e., $i_0 = 1$. The plant applies the constant controller $u_1^*(t_0)$ until $t_1 = t_0 + \delta$, and sends back $x(t_1)$ to the controller as a new state measurement. For the non-initial time $t_k, k \in \mathbb{N}_+$, do the following:

- (i) The plant transmits the current state information $x(t_k)$ to the controller.
- (ii) Based on $x(t_k)$, the controller solves Problem 3.1 1 for all $i \in \{1, \dots, M\}$, which provides the optimal control sequences $\mathbf{u}_1^*(t_k), \mathbf{u}_2^*(t_k), \dots, \mathbf{u}_M^*(t_k)$, and the corresponding optimal costs $J_1^*(x(t_k)), \dots, J_M^*(x(t_k))$.
- (iii) The controller selects $i_k \in \{1, \dots, M\}$ by solving the following problem;

$$i_k = \arg \max_{i \in \{1, \dots, M\}} i, \quad (3.16)$$

subject to:

$$J_i^*(x(t_k)) \leq J_1^*(x(t_k)) + \beta \quad (3.17)$$

$$J_i^*(x(t_k)) \leq J_{i_{k-1}}^*(x(t_{k-1})) - \gamma F(x(t_{k-1}), u_{i_{k-1}}^*(t_k), i_{k-1}), \quad (3.18)$$

where β and γ are the constant parameters, satisfying $0 \leq \beta, 0 < \gamma \leq 1$.

- (iv) The controller transmits $u_{i_k}^*(t_k)$, and then the plant applies $u_{i_k}^*(t_k)$ as sample-and-hold implementation until $t_{k+1} = t_k + i_k \delta$. The plant then sends back $x(t_{k+1})$ to the controller as a new current state measurement. \square

The main point of our proposed algorithm is the way to select the optimal index i_k given in the step (iii). From Lemma 3.1, the 1st pattern provides the minimum cost among all sampling patterns. Thus, the first condition (3.17) implies that larger patterns are allowed to be selected to obtain longer transmission intervals, but the optimal cost should not go far from the 1st pattern; the optimal cost is allowed to be larger only by β from $J_1^*(x(t_k))$, so that it does not degrade much the control performance. Thus, the parameter β plays a role to regulate the trade-off between the control performance and the transmission time intervals. That is, a smaller β leads to better control performance (but resulting in more transmissions), and larger β leads to less transmissions (but resulting in worse control performance).

The second condition (3.18) takes into account the optimal cost obtained at the previous time t_{k-1} , and this aims at guaranteeing the asymptotic stability of the origin. Note that γ needs to satisfy $0 < \gamma \leq 1$. As will be described in the next section, this condition ensures that Algorithm 3.1 is always implementable. Since it is desirable to reduce the communication load as much as possible, the controller selects the pattern providing the largest transmission interval satisfying (3.17), (3.18), i.e., $i_k = \arg \max i$ in (3.16).

The main advantage of using the proposed method is that the optimal cost $J_i^*(t_k)$ can be compared not only with the previous one $J_{i_{k-1}}^*(t_{k-1})$, but also with

the current ones obtained at t_k under different sampling patterns. This allows us not only to ensure stability, but also to evaluate how much the control performance becomes better or worse according to the transmission time intervals. Note that the control performance may also be regulated through the tuning of γ in (3.18). However, due to the tight condition $0 < \gamma \leq 1$, we cannot select γ large enough such that small patterns (good control performance) are ensured to be obtained. Thus the desired control performance can be suitably specified through the first condition (3.17), rather than (3.18).

Some remarks are in order regarding Algorithm 3.1.

Remark 3.1 (Relation to move-blocking MPC). *The proposed algorithm is to some extent related to move-blocking MPC [53], in the sense that the optimal control inputs are restricted to be constant for some time period. Note that move-blocking MPC aims at reducing the computational complexity by decreasing the degrees of freedom of the optimal control problem [53]; the proposed approach, on the other hand, aims at reducing the communication load through the move-blocking technique, and the reduction of computation load is not a primary objective here.* \square

Remark 3.2 (Effect of time delays). *The main drawback of Algorithm 3.1 is the requirement of solving multiple OCPs at the same time, which clearly induces a time-delay of transmitting control samples in practical implementations. Regarding time delays, several methods have been proposed to take them into account and can also be applied to our proposed self-triggered strategy. For example, a delay compensation strategy has been proposed in [54]. When applying this approach, the maximum total time delay $\bar{\tau}_d$ needs to be upper bounded to satisfy $\delta_1^{(i)} < T_p - \bar{\tau}_d$ in order to guarantee stability. This implies that the condition $i < (T_p - \bar{\tau}_d)/\delta$ is required in addition to the conditions (3.17), (3.18) as the rule to choose the sampling pattern.* \square

Remark 3.3 (Effect of the noise or model uncertainties). *In the above formulation, any effects of model uncertainties or disturbances have not considered. However, the proposed scheme can be extended to take into account these effects by slightly modifying Lemma 3.2. Suppose that the actual state is given by $\dot{x} = Ax + Bu + w$, where*

w denotes additive uncertainties or disturbances satisfying $\|w\| \leq w_{\max}$. By utilizing Theorem 2 in [45], it can be shown that there exists a positive L_v such that $J_1^*(x(t_k)) - J_i^*(x(t_{k-1})) \leq -F(x(t_{k-1}), u_i^*(t_{k-1}), i\delta) + L_v w_{\max}$ instead of (3.13). Therefore, assuming that w_{\max} is known, the corresponding self-triggered strategy is obtained by adding $L_v w_{\max}$ to the right hand side of (3.18). Note that the first condition (3.17) does not need to be modified, since Lemma 3.1 still holds even for the disturbance case. \square

3.1.3 Feasibility and Stability analysis

One of the desirable properties of Algorithm 3.1 is to ensure that it is always implementable, i.e., we need to exclude the case when all the patterns do not satisfy both (3.17) and (3.18). Furthermore, the stability of the closed loop system under Algorithm 3.1 needs to be verified. In the following theorem, it is deduced that both of these properties are satisfied.

Theorem 3.1. *Consider the networked control system in Fig. 1.1 where the plant follows the dynamics given by (3.1) and the proposed self-triggered strategy (Algorithm 3.1) is implemented. The followings are then satisfied:*

(i) *The way to obtain the pattern i_k in step (iii) in Algorithm 1.1, is always feasible.*

That is, there exists at least one pattern i , satisfying both (3.17), (3.18) for all $k \in \mathbb{N}_{\geq 0}$.

(ii) *The closed loop system is asymptotically stabilized to the origin.* \square

Proof. The proof of (i) is obtained by showing that the 1st sampling pattern ($i = 1$) always satisfies (3.17) and (3.18). The first condition is clearly satisfied when $i = 1$ since $\beta \geq 0$. Furthermore, from Lemma 3.2, we obtain

$$\begin{aligned} J_1^*(x(t_k)) &\leq J_{i_{k-1}}^*(x(t_{k-1})) - F(x(t_{k-1}), u_{i_{k-1}}^*(t_{k-1}), i_{k-1}) \\ &\leq J_{i_{k-1}}^*(x(t_{k-1})) - \gamma F(x(t_{k-1}), u_{i_{k-1}}^*(t_{k-1}), i_{k-1}). \end{aligned}$$

Thus the second condition holds for $i = 1$. Thus, the proof of (i) is complete.

The proof of (ii) is obtained by the fact that the optimal cost decreases along the time sequence. Since the optimal cost of the selected pattern satisfies (3.18), we have

$$\begin{aligned} J_{i_1}^*(x(t_1)) - J_{i_0}^*(x(t_0)) &\leq -\gamma F(x(t_0), u_{i_0}^*(t_0), i_0) \\ &< -\gamma \int_{t_0}^{t_1} x^\top(t) Q x(t) dt \\ J_{i_2}^*(x(t_2)) - J_{i_1}^*(x(t_1)) &\leq -\gamma F(x(t_1), u_{i_1}^*(t_1), i_1) \\ &< -\gamma \int_{t_1}^{t_2} x^\top(t) Q x(t) dt \\ &\vdots \end{aligned}$$

where the derivation from the first to the second inequality follows from the definition of the stage cost F given by (3.5). Summing over both sides of the above yields

$$\gamma \int_{t_0}^{\infty} x^\top(t) Q x(t) dt < J_{i_0}^*(x(t_0)) - J_{i_\infty}^*(x(\infty)) < \infty.$$

Since the function $x^\top(t) Q x(t)$ is uniformly continuous on $t \in [0, \infty)$ and $Q \succ 0$, we obtain $\|x(t)\| \rightarrow 0$ as $t \rightarrow \infty$ from Barbalat's lemma ([52]). This completes the proof. \square

3.1.4 Simulation results

In this subsection illustrative examples are provided to validate our control schemes. The simulation was conducted on Matlab 2016a under Windows 10, Intel(R) Core(TM) 2.40 GHz, 8 GB RAM.

(*Example 3.1*): Consider the following spring-mass-damper system:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t), \quad (3.19)$$

where $x(t) = [x_1(t), x_2(t)]^\top \in \mathbb{R}^2$ with $x_1(t)$, $x_2(t)$ being, respectively, mass

position and its velocity, and $u(t) \in \mathbb{R}$ denotes the force applied to the mass. $m = 1$ is set as the mass of the point, $k = 1$ is set as the spring coefficient, and $c = 0.2$ is set as the viscous damper coefficient. For solving the optimal control problem (Problem 3.1), the matrices for the stage cost are set as $Q = I_2$, $R = 0.1$, and the prediction horizon as $T_p = 10$. Regarding the sampling patterns, we have $\delta = 0.2$, $M = 30$ and the tuning parameters are chosen as $\beta = 1$, $\gamma = 0.5$. For the terminal set satisfying Assumption 3.1, the approach presented in [38] is given to compute P_f and is obtained as

$$P_f = \begin{bmatrix} 6.82 & 1.23 \\ 1.23 & 2.43 \end{bmatrix}, \quad (3.20)$$

and $\varepsilon = 1$.

Figure 3.4 illustrates the resulting state trajectories from the initial states given by $x(0) = [2.5; 2.0]$. In the figure, blue solid lines represent the state trajectories by applying Algorithm 3.1 and red dotted lines represent the state trajectories when Problem 3.1 is solved periodically with 0.1 sampling time interval (i.e., $i_k = 1, \forall k \in \mathbb{N}$). From the result, it is shown that all state trajectories are asymptotically stabilized to the origin by applying both proposed self-triggered strategy (Algorithm 3.1) and the periodic strategy. Moreover, the convergence speed under Algorithm 3.1 seems to be slower than the periodic strategy, which indicates that the periodic scheme achieves better control performance (a more quantitative analysis is provided below). Figure 3.5 illustrates the control inputs by applying Algorithm 3.1 (blue solid lines) and the periodic one with 0.1 sampling time interval (red dotted lines). From the figure, control inputs are updated less frequently by applying Algorithm 3.1 than the periodic scheme and the communication reduction seems to be achieved. To provide a more quantitative analysis, Table 3.1 illustrates the convergence time when the state enters a small region around the origin ($\|x\| \leq 0.001$), as well as the number of transmission instants during the time period $t \in [0, 50]$.

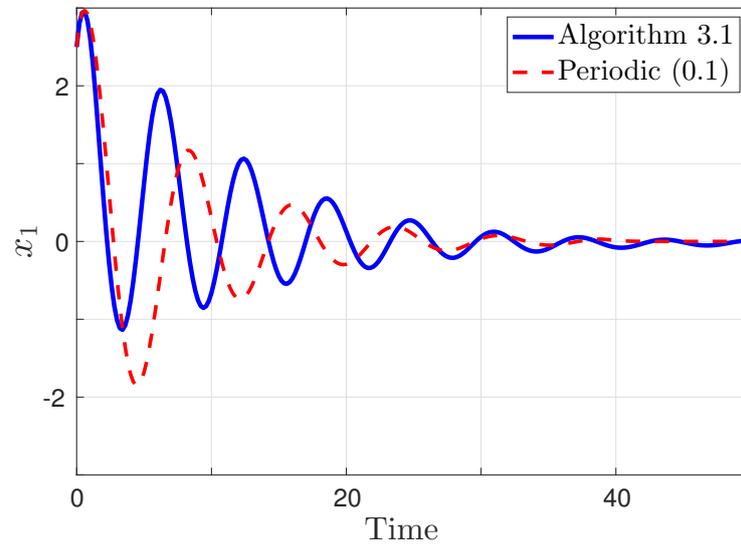
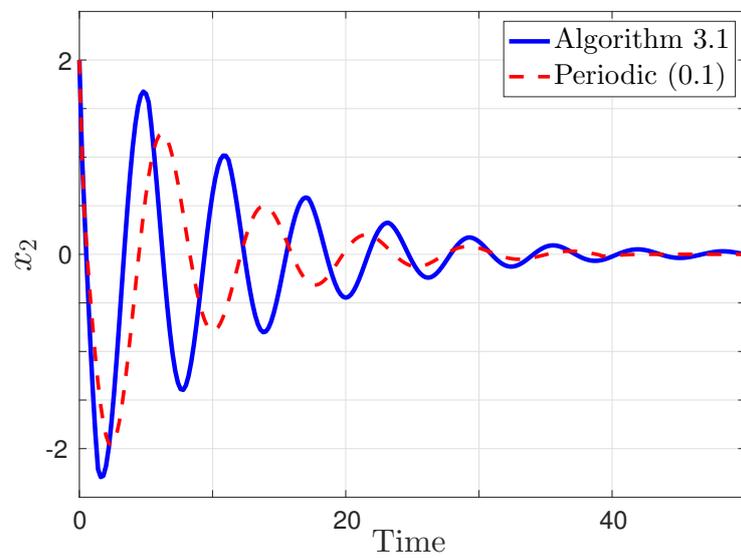
(a) Trajectories of x_1 .(b) Trajectories of x_2 .

FIGURE 3.4: State trajectories by implementing Algorithm 3.1 (blue solid lines) and the periodic MPC (red dotted lines).

TABLE 3.1: Convergence time when the state trajectory enters the region ($\|x\| \leq 0.001$) and the number of transmission instants.

	Algorithm 3.1	Periodic (0.1)
Convergence time	35.8	29.0
Transmission instants	12	251

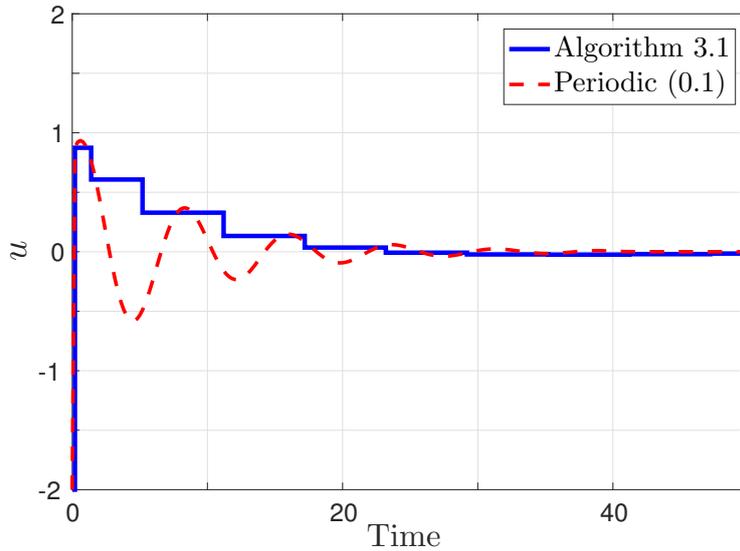
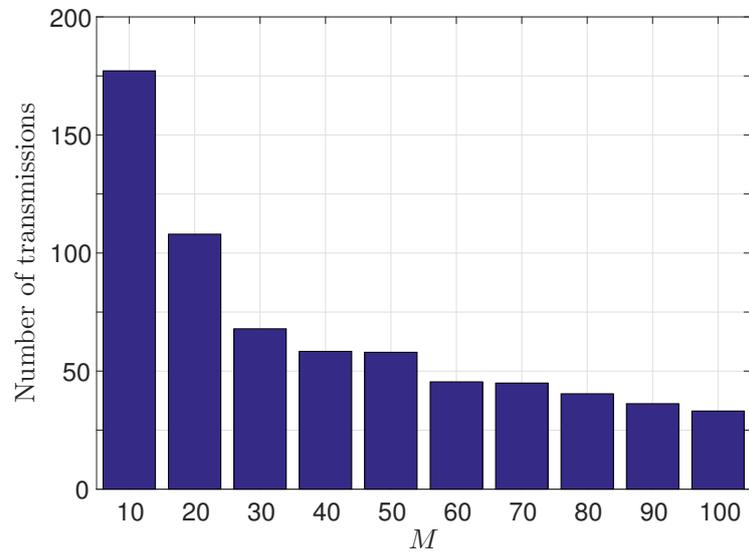


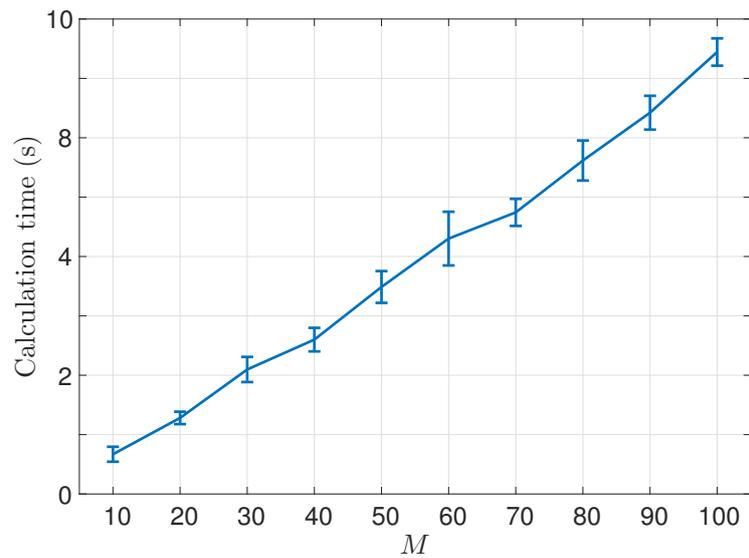
FIGURE 3.5: Applied control inputs by applying Algorithm 3.1 (blue solid line) and periodic MPC with sampling time interval 0.1 (red dotted line).

From the table, Algorithm 3.1 takes $35.8 - 29.0 = 6.8$ longer convergence time than the periodic case, which means that the control performance is degraded in contrast to the periodic case. On the other hand, the number of transmission instants by applying Algorithm 3.1 is smaller by $251 - 12 = 239$. Therefore, it is shown that the communication reduction is achieved by applying Algorithm 3.1 compared to the periodic strategy, while, on the other hand, degrading control performance.

In Fig. 3.5, the transmission time interval after the time $t = 6$ is always given by 3.0 (i.e., $i_k = M = 30$), which achieves the maximal index of sampling patterns. Therefore, it is deduced that if M (the number of sampling patterns) is selected larger, we can potentially reduce the number of transmission instants. To verify this, Algorithm 3.1 is implemented under different selection



(a) Number of transmissions



(b) Calculation time

FIGURE 3.6: Number of transmissions during the time interval $t \in [0, 300]$ and the average calculation time against the number of sampling patterns M .

of M and compute the number of transmission instants during the time interval $t \in [0, 300]$. Also, the average calculation times to compute the control input and the transmission time interval (i.e., the average calculation time from step 2 to step 3 in Algorithm 3.1) are given. The results are shown in Fig. 3.6. In Fig. 3.6(a), each bar represents the number of transmission instants for each $M = 10j$, ($j = 1, \dots, 10$), and Fig. 3.6(b) illustrates the average calculation times with the corresponding standard deviations. Indeed, it is shown from the figure that the number of transmission instants is monotonically decreasing as M is selected larger. On the other hand, the calculation time increases as M is selected larger, and this is due to the fact the number of optimal control problems to be solved increases. Thus, it is shown that we can reduce communication load by increasing the number of sampling patterns, while increasing the computational complexity of solving the optimal control problems.

To conclude, it is shown in this example that:

- Communication reduction is achieved by applying Algorithm 3.1 compared to the periodic case with 0.1 sampling time interval. On the other hand, control performance is degraded at the expense of achieving the communication reduction.
- As M is selected larger, a more communication reduction is achieved. On the other hand, the computational complexity becomes higher as M is selected larger.

Although in this example the number of transmission instants decreases as M becomes larger, it may not be the case for *unstable* systems, since the state may diverge by applying constant control signals for some time period. This will be clearly seen in the next example.

(Example 3.2): In the previous example, we consider a spring-mass-damper system that is a *stable* system. As a more interesting problem, we consider the following linearized system of an inverted pendulum on a cart (see [13]);

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -mg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/\ell & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/M\ell \end{bmatrix} u, \quad (3.21)$$

which is an *unstable* system. Here, we denote $x = [x_1 \ x_2 \ x_3 \ x_4]^\top \in \mathbb{R}^4$ with x_1, x_2 being the position of the cart and its velocity, and x_3, x_4 being the angle of the pendulum and its velocity. $u \in \mathbb{R}$ represents the force applied to the cart. We set $m = 1$ as the point mass, $M = 5$ as the mass of the cart, $\ell = 2$ as the length of the massless rod, and $g = 9.8$ as the gravity. The system is unstable having a positive eigenvalue 1.40 in matrix A . The constraint for the control input is assumed to be given by $\|u\| \leq 10$. For solving the optimal control problem (Problem 3.1), we set the matrices for the stage cost as $Q = I_2$, $R = 0.1$, and the prediction horizon as $T_p = 10$. Regarding the sampling patterns, assume $\delta = 0.1$ and $M = 30$. The matrix for the terminal constraint P_f is computed as

$$P_f = \begin{bmatrix} 29.9 & 38.3 & 139 & 89.6 \\ 38.3 & 85.0 & 320 & 207 \\ 139 & 320 & 1600 & 959 \\ 89.6 & 207 & 959 & 592 \end{bmatrix} \quad (3.22)$$

and $\varepsilon = 0.01$. The initial state is assumed to be $x(0) = [1, 0, 0, 0]$.

Figure 3.7 illustrates the resulting state trajectories of x_1 and x_2 , and Fig. 3.8 illustrates the ones of x_3, x_4 by applying Algorithm 3.1, with the tuning parameters as $\beta = 1, \gamma = 0.5$. Again, red dotted lines represent the state trajectories when Problem 3.1 is solved periodically with 0.1 sampling time interval (i.e.,

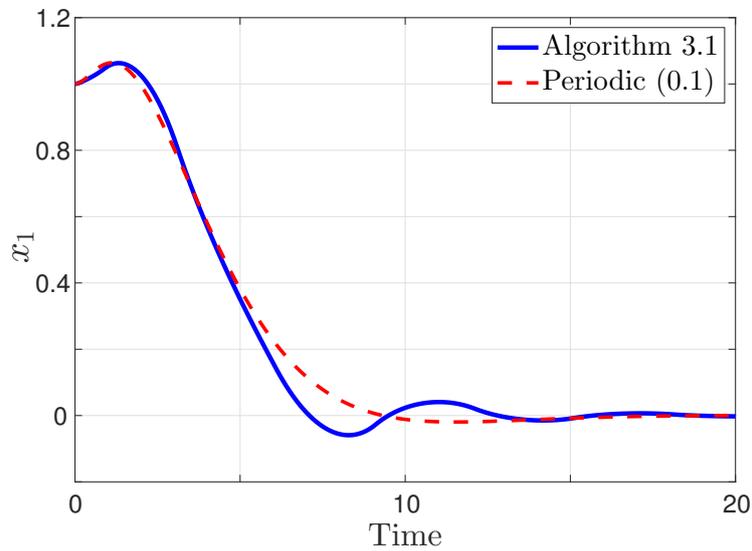
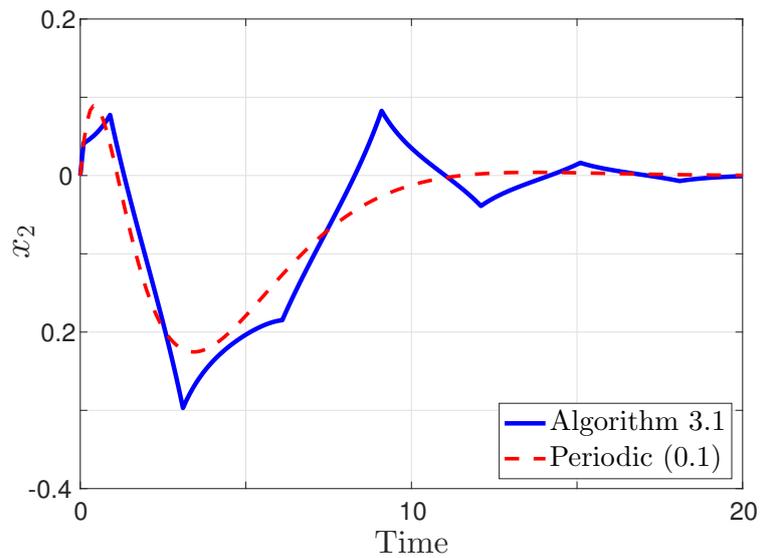
(a) State trajectories of x_1 .(b) State trajectory of x_2 .

FIGURE 3.7: State trajectories of x_1 and x_2 by implementing Algorithm 3.1 (blue solid lines) and the periodic MPC (red dotted lines).

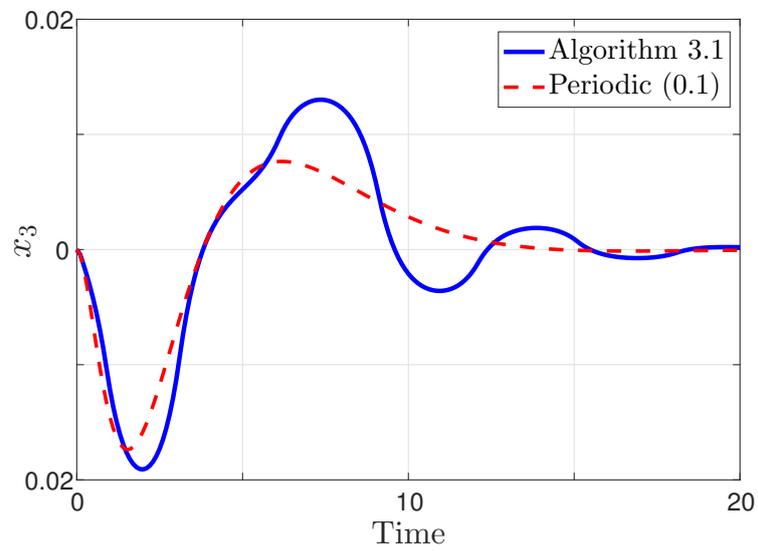
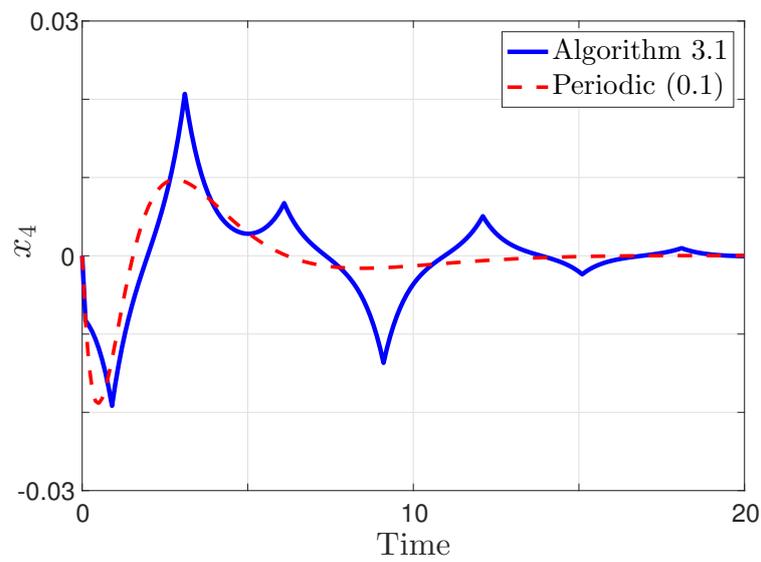
(a) State trajectories of x_3 .(b) State trajectories of x_4 .

FIGURE 3.8: State trajectories of x_3 and x_4 by implementing Algorithm 3.1 (blue solid lines) and the periodic MPC (red dotted lines).

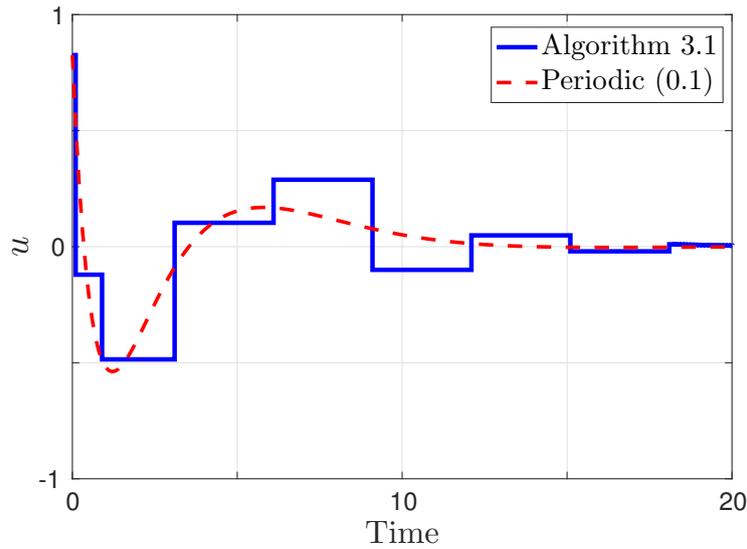


FIGURE 3.9: Applied control inputs under Algorithm 3.1 (blue line) and periodic MPC with sampling time interval 0.1 (red dotted line).

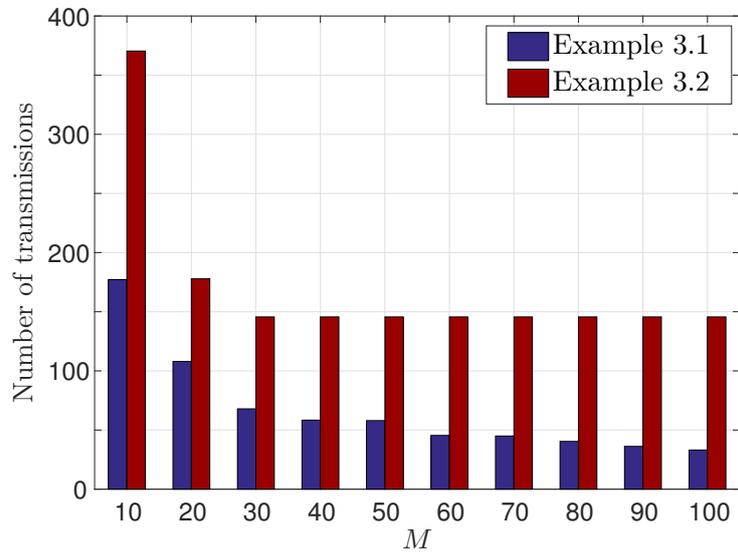
$i_k = 1, \forall k \in \mathbb{N}$). From the figure, it is shown that all state trajectories are asymptotically stabilized to the origin. Similarly to the result in Example 3.1, convergence of states seems to be slower under Algorithm 3.1 than the periodic case. For instance, the state trajectory of x_1 behaves oscillatory (see Fig. 3.7), while the periodic case does not. Figure 3.9 illustrates the corresponding control input by implementing Algorithm 3.1 (blue solid line) and the periodic scheme with 0.1 sampling time interval (red dotted line). From the figure, it is shown that control inputs are updated less frequently by applying Algorithm 3.1 than the periodic case. Table 3.2 illustrates the convergence time when the state enters the small region around the origin ($\|x\| \leq 0.001$), as well as the number of transmission time instants during the time interval $t \in [0, 30]$. Indeed, from the table the convergence time by Algorithm 3.1 is longer by $20.1 - 17.8 = 2.3$ than the periodic scheme. On the other hand, the number of transmission instants by applying Algorithm 3.1 is smaller by $301 - 15 = 286$. Thus, it is shown that the communication reduction can be achieved by applying Algorithm 3.1.

TABLE 3.2: Convergence time when the state trajectory enters the region ($\|x\| \leq 0.001$) and the number of transmission instants during the time period $t \in [0, 30]$.

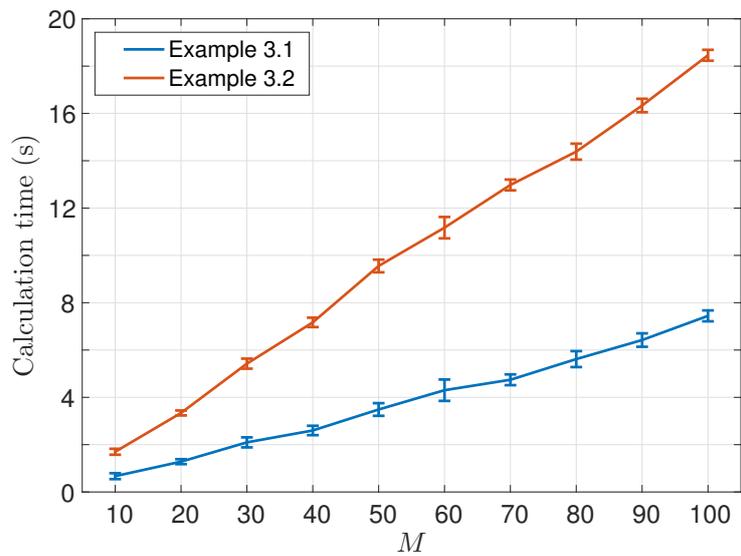
	Algorithm 3.1	Periodic (0.1)
Convergence time	20.1	17.8
Transmission instants	15	301

So far, we have considered both stable systems (Example 3.1) and unstable systems (Example 3.2) and illustrated the effectiveness of the proposed approach. Let us now compare the two results in terms of both communication load and computational complexity. To this end, Algorithm 3.1 is simulated again under different selection of M and compute the number of transmission instants during the time period $t \in [0, 300]$, as well as the average calculation time to compute control input (i.e., the average calculation time from step 2 to step 3 in Algorithm 3.1). The results are shown in Fig. 3.10. From Fig. 3.10(a), the number of transmission instants in Example 3.2 is larger than that in Example 3.1 for all M . Moreover, the number of transmission instants obtained in Example 3.2 does not decrease for all $M > 30$ while it is monotonically decreasing in the case of Example 3.1. Intuitively, this is due to the fact that for unstable systems applying a constant control signal for some time period may easily lead to a divergence of states compared to the stable case. For unstable systems, there may thus exist an inherent upper bound of the time interval during which control inputs are allowed to be constant to stabilize the system. From Fig. 3.10(b), it takes more calculation times in Example 3.2 for all M than that in Example 3.1. This is because Example 3.2 deals with higher order systems ($n = 4$) than Example 3.1 ($n = 2$) and the number of decision variables to solve Problem 3.1 becomes larger. Thus, a control problem of unstable, high-order systems is harder to be handled than stable, low-order systems as it requires a more communication load and computational complexity.

To conclude, it is shown in this example that:



(a) Number of transmission instants.



(b) Calculation time.

FIGURE 3.10: Number of transmission time instants during the time period $t \in [0, 300]$ and the calculation time against the number of sampling patterns M .

- Communication reduction is achieved by applying Algorithm 3.1 compared to the periodic scheme with 0.1 sampling time interval. On the other hand, control performance is degraded at the expense of achieving the communication reduction.
- Compared to Example 3.1, the number of transmission instants cannot get smaller for $M > 30$, since Example 3.2 considers unstable systems. Moreover, the computational complexity of solving Problem 3.1 is higher than that of Example 3.1 since Example 3.2 considers higher order systems.

3.2 Contractive set-based approach

In the previous section, the author proposes a self-triggered algorithm that solves a finite set of optimal control problems according to the discretization scheme in Fig. 3.2. Essentially, the computational complexity of solving the optimal control problem (Problem 3.1) heavily depends on the choice of prediction horizon N_p ; as N_p is selected larger, the number of decision variables becomes larger. On the other hand, the prediction horizon N_p must be selected large enough to satisfy the terminal constraint as in (3.9). More specifically, if we want to stabilize *all* states in \mathcal{X} , the prediction horizon N_p should be selected to satisfy $\mathcal{F}_{N_p} \supseteq \mathcal{X}$, where

$$\mathcal{F}_{N_p} = \{x(0) \in \mathbb{R}^n \mid \exists u(0), u(1), \dots, u(N_p - 1) \in \mathcal{U} : x(N_p) \in \Phi\}, \quad (3.23)$$

i.e., \mathcal{F}_{N_p} denotes the set of states that can reach the terminal set Φ within the time steps N_p . While the choice of N_p according to (3.23) ensures feasibility of Problem 3.1 (for the case $i = 1$) for all $x \in \mathcal{X}$, it may be sometimes a conservative choice and N_p may be selected unnecessarily long; in some control applications, it may be of interest for us to consider stabilizing the system only within a certain set $\mathcal{S} \subset \mathcal{X}$, instead of stabilizing all states in \mathcal{X} . In such case, we can potentially decrease the prediction horizon and reduce computational

complexity of solving the optimal control problem. In particular, if the region of interest \mathcal{S} is characterized by a λ -contractive set of \mathcal{X} , which will be defined soon in the next section, it is then shown that a one step horizon controller ($N_p = 1$) is sufficient to stabilize the system.

In the following subsections, a new optimal control problem will be formulated by incorporating the notion of contractive set, aiming at reducing the computational complexity with respect to the multiple-discretizations approach presented in the previous section. Similarly to the previous section, both feasibility and stability are shown for the new problem formulation, and a simulation example is illustrated to validate the proposed scheme.

3.2.1 Set-Invariance theory

Consider the Linear-Time-Invariant (LTI) system:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (3.24)$$

where $x(t) \in \mathbb{R}^n$ is the state and $u(t) \in \mathbb{R}^m$ is the control variable. Again, assume that the state and the control input must belong to the constraint sets \mathcal{X} and \mathcal{U} , where these sets are compact, convex and contain the origin in their interiors. Here, we further assume that these sets are characterized by polyhedral sets as follows:

$$\mathcal{X} = \{x \in \mathbb{R}^n : H_x x \leq h_x\}, \quad (3.25)$$

$$\mathcal{U} = \{u \in \mathbb{R}^m : H_u u \leq h_u\}, \quad (3.26)$$

where $H_x \in \mathbb{R}^{n_x \times n}$, $H_u \in \mathbb{R}^{n_u \times m}$ and h_x, h_u are appropriately sized vectors having positive components.

Let $t_0 < t_1 < t_2 < \dots$ be the transmission time instants between the plant and the controller; at t_k , $k \in \mathbb{N}$, the plant transmits $x(t_k)$ to the controller, and

the controller solves an optimal control problem to compute suitable control inputs to be applied, and determine the next communication time t_{k+1} . Suppose that the current time is t_k . Similarly to the multiple discretization approach given in the previous section, let us first discretize the system with respect to the sampling time interval δ under zero-order-hold controller, yielding

$$x(t_k + \delta) = A_\delta x(t_k) + B_\delta u(t_k), \quad (3.27)$$

where $A_\delta = e^{A\delta}$, $B_\delta = \int_0^\delta e^{As} B ds$. Namely, $x(t_k + \delta)$ represents the state at $t_k + \delta$ by applying a control input $u(t_k)$ constantly for the time interval δ .

In the following, the standard notions of *controlled invariant set* and *λ -contractive set* [55] are given, which are important definitions to characterize the invariance and convergence properties for constrained control systems.

Definition 3.2 (Controlled invariant set). *For a given $\mathcal{S} \subseteq \mathcal{X}$, \mathcal{S} is said to be a controlled invariant set in \mathcal{X} if and only if there exists a control law $g : \mathcal{X} \rightarrow \mathcal{U}$ such that:*

$$x(t_k) \in \mathcal{S} \implies A_\delta x(t_k) + B_\delta g(x(t_k)) \in \mathcal{S}. \quad (3.28)$$

□

Roughly speaking, a controlled invariant set \mathcal{S} indicates that if the state $x(t_k)$ is inside \mathcal{S} , there exists a controller such that the state remains in \mathcal{S} at $t_k + \delta$.

Definition 3.3 (λ -contractive set). *For a given $\mathcal{S} \subseteq \mathcal{X}$, \mathcal{S} is said to be a λ -contractive set in \mathcal{X} for $\lambda \in [0, 1]$, if and only if there exists a control law $g(x) \in \mathcal{U}$ such that:*

$$x(t_k) \in \mathcal{S} \implies A_\delta x(t_k) + B_\delta g(x(t_k)) \in \lambda \mathcal{S}. \quad (3.29)$$

□

A λ -contractive set \mathcal{S} indicates that if the state is inside \mathcal{S} , there exists a controller such that the state can be driven into the set $\lambda \mathcal{S}$ at the next time step. From the definition, a controlled invariant set implies a λ -contractive set with

$\lambda = 1$. In the following, several established results are reviewed for obtaining a contractive set and the corresponding properties (see, e.g., [55]). For given $\lambda \in [0, 1)$ and $\mathcal{X} \subset \mathbb{R}^n$, there are several ways to efficiently construct a λ -contractive set in \mathcal{X} . For a given compact and convex set $\mathcal{D} \subset \mathbb{R}^n$, let the mapping $\mathcal{Q}_\lambda : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be given by

$$\mathcal{Q}_\lambda(\mathcal{D}) = \{x \in \mathcal{X} : \exists u \in \mathcal{U}, A_\delta x + B_\delta u \in \lambda \mathcal{D}\}. \quad (3.30)$$

A simple algorithm to obtain a λ -contractive set in \mathcal{X} is to compute $\Omega_j \subset \mathbb{R}^n$, $j \in \mathbb{N}$ as

$$\Omega_0 = \mathcal{X}, \quad \Omega_{j+1} = \mathcal{Q}_\lambda(\Omega_j) \cap \mathcal{X}, \quad (3.31)$$

and then it holds that the set $\mathcal{S} = \lim_{j \rightarrow \infty} \Omega_j$ is λ -contractive, see e.g., [55]. If $\Omega_{j+1} = \Omega_j$ for some j , the λ -contractive set is obtained as $\mathcal{S} = \Omega_j$, which requires only a finite number of iterations. Although such condition does not hold in general, it is still shown, that the algorithm converges in the sense that for every $\lambda < \bar{\lambda} < 1$, there exists a finite $j \in \mathbb{N}_+$ such that the set Ω_j is $\bar{\lambda}$ -contractive (see *Theorem 3.2* in [55]). Several other algorithms have been recently proposed, see e.g., [56], [57] and see also [58] for a detailed convergence analysis. The following lemma illustrates the existence of a (non-quadratic) Lyapunov function in a given λ -contractive set:

Lemma 3.3 ([55]). *Let $\mathcal{S} \subset \mathcal{X}$ be a λ -contractive \mathcal{C} -set with $\lambda \in [0, 1]$ and the associated gauge function $\Psi_{\mathcal{S}} : \mathcal{S} \rightarrow \mathbb{R}_+$. Then, there exists a control law $g : \mathcal{X} \rightarrow \mathcal{U}$ such that*

$$\Psi_{\mathcal{S}}(A_\delta x(t_k) + B_\delta g(x(t_k))) \leq \lambda \Psi_{\mathcal{S}}(x(t_k)), \quad (3.32)$$

for all $x(t_k) \in \mathcal{S}$. □

Lemma 3.3 follows immediately from Definition 3.3. If $\lambda < 1$, (3.32) implies the existence of a stabilizing controller in \mathcal{S} in the sense that the output of the gauge function $\Psi_{\mathcal{S}}(\cdot)$ is guaranteed to decrease. A significant advantage here

is that a one step horizon controller is guaranteed to exist in \mathcal{S} , which stabilizes the state to the origin. The gauge function $\Psi_{\mathcal{S}}(\cdot)$ defined in \mathcal{S} is known as *set-induced Lyapunov function* in the literature; for a detailed discussion, see e.g., [55].

(*Example 3.3*): Let us go back to an example of spring-mass-damper system considered in Example 3.1:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t), \quad (3.33)$$

with $m = 1$, $k = 1$, and $c = 0.2$. The system is discretized under the sampling time interval $\delta = 0.2$. Assume that the constraint sets are given by $\mathcal{X} = \{x \in \mathbb{R}^2 : \|x\|_{\infty} \leq 4\}$, $\mathcal{U} = \{u \in \mathbb{R} : \|u\|_{\infty} \leq 2\}$. Let $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ be the corresponding λ -contractive sets with $\lambda = 1.0, 0.9, 0.8$, respectively. Figure 3.11 illustrates the contractive sets by implementing the iterative procedure presented in (3.31). In the figure, the gray region represents the constraint set \mathcal{X} and the contractive sets $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ are illustrated with different shades of blue. As shown in the figure, the contractive set becomes smaller as λ becomes smaller. Intuitively, this is due to the fact that the state is restricted to converge to a tighter region as the contractivity λ is selected smaller. \square

Based on the set-invariance theory described above, let us now formulate a new optimal control problem. For a given $\lambda \in [0, 1)$, suppose that a λ -contractive set \mathcal{S} is constructed in \mathcal{X} . Note that since \mathcal{X} is assumed to be a polyhedral, compact and convex set, one can efficiently compute the λ -contractive set through polyhedral operations according to (3.31)¹. The obtained λ -contractive

¹If the iterative procedure in (3.31) does not converge in finite time, one can stop the procedure to obtain a $\bar{\lambda}$ -contractive set ($\lambda < \bar{\lambda} < 1$) in a finite number of iterations. In such case, we can use $\bar{\lambda}$ (instead of λ) as the parameter to design the self-triggered strategies provided throughout this chapter.

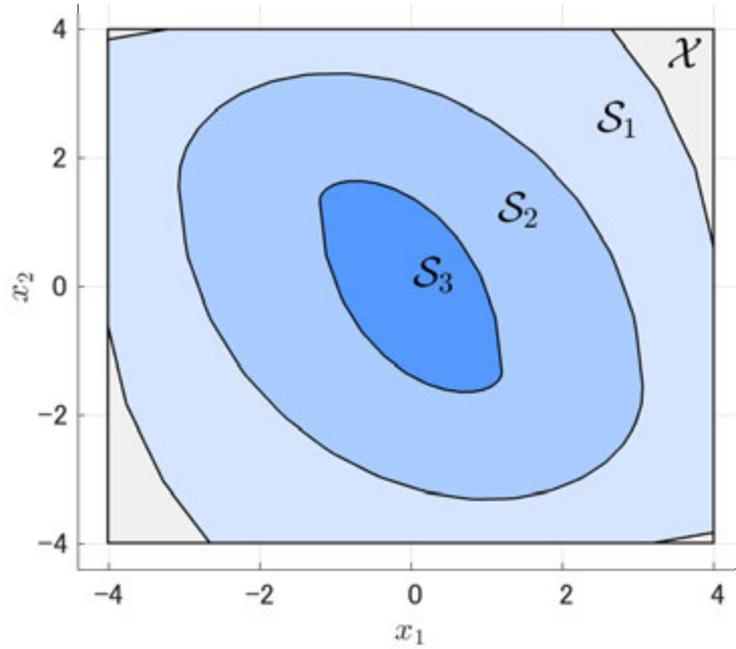


FIGURE 3.11: λ -contractive sets \mathcal{S}_1 ($\lambda = 1.0$), \mathcal{S}_2 ($\lambda = 0.9$), and \mathcal{S}_3 ($\lambda = 0.8$). The contractive sets are illustrated with different shades of blue. The gray region represents \mathcal{X} .

set \mathcal{S} can be denoted as

$$\mathcal{S} = \text{co}\{v_1, v_2, \dots, v_N\} \subseteq \mathcal{X}, \quad (3.34)$$

where $v_n, n \in \{1, 2, \dots, N\}$ represent the vertices of \mathcal{S} , and N represents the number of them.

Assumption 3.2. *The initial state is inside \mathcal{S} , i.e., $x(t_0) \in \mathcal{S}$.*

As will be seen later, Assumption 3.2 is required to guarantee feasibility of the optimal control problems for all the transmission time instants.

In the previous section, an optimal control problem is proposed for each $i \in \{1, \dots, M\}$ according to Problem 3.1, where each i indicates a candidate of transmission time interval determined by the controller. Similarly to Problem 3.1, the author proposes the following optimal control problem for each $i \in \{1, \dots, M\}$:

Problem 3.2 (Optimal Control Problem for i). For given $x(t_k)$, $i \in \{1, \dots, M\}$ and the λ -contractive set \mathcal{S} , find $u(t_k) \in \mathcal{U}$ and $\varepsilon \in \mathbb{R}$ by solving the following problem:

$$\min_{u(t_k), \varepsilon} \varepsilon, \quad (3.35)$$

subject to the following constraints:

$$\begin{cases} A_{i\delta}x(t_k) + B_{i\delta}u(t_k) \in \varepsilon\varepsilon_x\mathcal{S}, & (3.36) \\ A_{j\delta}x(t_k) + B_{j\delta}u(t_k) \in \mathcal{X}, \quad \forall j \in \{1, \dots, i\} & (3.37) \\ u(t_k) \in \mathcal{U}, \quad \varepsilon \in [0, \lambda], & (3.38) \end{cases}$$

where $A_{i\delta} = e^{A_{i\delta}}$, $B_{i\delta} = \int_0^{i\delta} e^{A_{i\delta}\tau} d\tau B$ and $\varepsilon_x = \Psi_{\mathcal{S}}(x(t_k))$. \square

In (3.36), $A_{i\delta}x(t_k) + B_{i\delta}u(t_k)$ represents a state by applying a controller $u(t_k) \in \mathcal{U}$ constantly for the time interval $i\delta$. Moreover, from the definition of the gauge function $\Psi_{\mathcal{S}}(\cdot)$ we have $x(t_k) \in \varepsilon_x\mathcal{S}$. Thus, Problem 3.2 aims to find the smallest possible scaled set $\varepsilon\varepsilon_x\mathcal{S}$, such that the state enters $\varepsilon\varepsilon_x\mathcal{S}$ (from $\varepsilon_x\mathcal{S}$) by applying a constant control input for the time interval $i\delta$. This means that a stabilizing controller is found under the transmission time interval $i\delta$. The constraint in (3.37) implies that the state must remain inside \mathcal{X} while applying a constant controller, which is imposed to guarantee the constraint satisfaction.

For given $x(t_k) \in \mathcal{S}$ and $i \in \{1, \dots, M\}$, let $u_i^*(t_k)$, ε_i^* be the optimal control input and the value of ε by solving Problem 3.2. From (3.36), the state enters $\varepsilon_i^*\varepsilon_x\mathcal{S}$ if $u_i^*(t_k)$ is applied constantly for the time interval $i\delta$, i.e., $x(t_k + i\delta) \in \varepsilon_i^*\varepsilon_x\mathcal{S}$ with $x(t_k + i\delta) = A_{i\delta}x(t_k) + B_{i\delta}u_i^*(t_k)$, which means that we have

$$\Psi_{\mathcal{S}}(x(t_k + i\delta)) \leq \varepsilon_i^*\Psi_{\mathcal{S}}(x(t_k)) \quad (3.39)$$

or

$$\Psi_{\mathcal{S}}(x(t_k + i\delta)) - \Psi_{\mathcal{S}}(x(t_k)) \leq -(1 - \varepsilon_i^*)\Psi_{\mathcal{S}}(x(t_k)) \quad (3.40)$$

with $0 \leq \varepsilon_i^* \leq \lambda < 1$. Thus, $1 - \varepsilon_i^*$ represents how much the output of the gauge function (as a Lyapunov function candidate) decreases by applying the optimal

controller $u_i^*(t_k)$ constantly for the time interval $i\delta$. That is, if $1 - \varepsilon_i^*$ becomes larger (i.e., ε_i^* becomes smaller), then the state will be closer to the origin and a better control performance is achieved.

Now, consider solving Problem 3.2 for all $i \in \mathcal{M}$, which provides different solutions under different transmission time intervals. In the following, let $\mathcal{I}(x(t_k))$ be the set of indices (transmission time intervals) where Problem 3.2 provides a feasible solution. That is,

$$\mathcal{I}(x(t_k)) = \{i \in \{1, \dots, M\} : \text{Problem 3.2 is feasible for } i\}. \quad (3.41)$$

Regarding the feasible set $\mathcal{I}(x(k))$, we obtain the following:

Lemma 3.4. *For any $x(t_k) \in \mathcal{S}$, $\mathcal{I}(x(t_k))$ is non-empty.*

Proof. In the following, it is shown that Problem 3.2 is feasible for $i = 1$ for any $x(t_k) \in \mathcal{S}$ from the property of the λ -contractive set. Let $\varepsilon_x = \Psi_{\mathcal{S}}(x(t_k))$. We have $\varepsilon_x \in [0, 1]$ since $x(t_k) \in \mathcal{S}$. Moreover, since $x(t_k) \in \varepsilon_x \mathcal{S}$, there exist $\lambda_n \in [0, 1]$, $n \in \{1, \dots, N\}$, such that $x(t_k) = \varepsilon_x \sum_{n=1}^N \lambda_n v_n$, $\sum_{n=1}^N \lambda_n = 1$, where v_n , $n \in \{1, \dots, N\}$ are the vertices of \mathcal{S} as in (3.34). From Definition 3.3, there exist $u_1, \dots, u_N \in \mathcal{U}$ such that $A v_n + B u_n \in \lambda \mathcal{S}$, $\forall n \in \{1, \dots, N\}$. Let $u(t_k) \in \mathbb{R}^m$ be given by $u(t_k) = \varepsilon_x \sum_{n=1}^N \lambda_n u_n \in \mathcal{U}$. Then we obtain

$$A_{\delta} x(t_k) + B_{\delta} u(t_k) = \varepsilon_x \sum_{n=1}^N \lambda_n (A_{\delta} v_n + B_{\delta} u_n) \in \varepsilon_x \lambda \mathcal{S} \subseteq \mathcal{X}.$$

The above inclusion implies that Problem 3.2 has a feasible solution with $u = u(k) \in \mathcal{U}$ and $\varepsilon = \lambda \in [0, \lambda]$, since the constraints (3.37), (3.38) imposed in Problem 3.2 are all fulfilled. This completes the proof. \square

3.2.2 Self-triggered strategy

Let us now present the self-triggered strategy. After solving Problem 3.2 for all $i \in \{1, \dots, M\}$, which provides the optimal (feasible) solutions of control inputs $u_i^*(t_k) \in \mathcal{U}$ and scalars $\varepsilon_i^* \in [0, \lambda]$ for all $i \in \mathcal{I}(x(k_m))$, the controller then selects a suitable transmission time interval among them. Similarly to the self-triggered strategy given in the previous section (Algorithm 3.1), the controller selects the maximal index such that both the control performance and stability are taken into account. A more detailed algorithm is described in the following overall algorithm:

Algorithm 3.2 (Self-triggered MPC strategy via contractive set)

For any transmission time instants $t_k, k \in \mathbb{N}$, do the following:

- (i) The plant transmits the current state information $x(t_k)$ to the controller.
- (ii) Based on $x(t_k)$, the controller solves Problem 3.2 for all $i \in \{1, \dots, M\}$, which provides the optimal control inputs $u_i^*(t_k) \in \mathcal{U}$, and the corresponding scalars ε_i^* for all $i \in \mathcal{I}(x(t_k))$.
- (iii) The controller picks up an optimal index $i_k \in \mathcal{I}(x(t_k))$ by solving the following problem:

$$i_k = \arg \max_{i \in \mathcal{I}(x(t_k))} i \quad (3.42)$$

subject to:

$$\varepsilon_i^* \leq \beta_2 (\varepsilon_1^*)^i, \quad (3.43)$$

$$\varepsilon_i^* \leq \gamma_2 \varepsilon_x, \quad (3.44)$$

where $\varepsilon_x = \Psi_S(x(t_k))$ and $\beta_2 \geq 1, \lambda \leq \gamma_2 < 1$ represent a given tuning weight parameter. Then, set $t_{k+1} = t_k + i_k \delta$ and $u(t_k) = u_{i_k}^*(t_k)$, and the controller transmits $u(t_k)$ and t_{k+1} to the plant.

- (iv) The plant applies $u(t_k)$ for all $t \in [t_k, t_{k+1})$. Set $k \leftarrow k + 1$, and then go back to step (i). \square

The main point of our proposed algorithm is the way to select the optimal index i_k in Step (iii). Similarly to Algorithm 3.1, (3.43) represents the constraint for achieving the control performance with respect to the optimal cost for $i = 1$ (i.e., ε_1^*), and $\beta \geq 1$ represents the associated tuning weight. As described in the previous subsection, the term ε_i^* represents how much the output of the gauge function decreases by applying the optimal controller $u_i^*(t_k)$ constantly for the time interval $i\delta$, and we will achieve better control performance if this value becomes smaller. The second condition (3.44) imposes that the value of gauge function will be smaller at $t_k + i\delta$ than the one at the current time ε_x . As will be seen the analysis that follows, this constraint is used to guarantee asymptotic stability of the origin by guaranteeing that the gauge function as a Lyapunov function candidate is strictly decreasing.

3.2.3 Feasibility and Stability analysis

In the multiple discretization approach, it has been shown that the self-triggered algorithm (Algorithm 3.1) is implementable by proving that the optimal control problem for $i = 1$ is always feasible. Also, asymptotic stability has been guaranteed by showing that the optimal cost as a Lyapunov function candidate is decreasing. In the following, it is also shown that Algorithm 3.2 is always implementable by showing that Problem 3.2 is feasible for all $t_k, k \in \mathbb{N}$. Moreover, asymptotic stability is guaranteed by showing that the output of gauge function is strictly decreasing.

Theorem 3.2. *Consider the networked control system in Fig. 1.1 where the plant follows the dynamics given by (3.24) and the proposed self-triggered strategy (Algorithm 3.2) is implemented. Suppose also that Assumption 3.2 holds. The followings are then satisfied:*

- (i) The way to select i_k in step (iii) in Algorithm 3.2, is always feasible. That is, it holds that $\mathcal{I}(x(t_k))$ is non-empty for all $k \in \mathbb{N}$, and there exists at least an index $i \in \{1, \dots, M\}$, satisfying both (3.43), (3.44) for all $k \in \mathbb{N}_{\geq 0}$.
- (ii) The closed loop system is asymptotically stabilized to the origin, i.e., $x(t) \rightarrow 0$ as $t \rightarrow \infty$. \square

Proof. We first prove $\mathcal{I}(x(t_k))$ is non-empty by showing that $x(t_k) \in \mathcal{S}$ for all $k \in \mathbb{N}$. By Assumption 3.2 we obtain $x(k_0) \in \mathcal{S}$ and thus $\mathcal{I}(x(k_0))$ is non-empty (see Lemma 3.4). Since i_0 is obtained from (3.42), we have $i_0 \in \mathcal{I}(x(k_0))$ which means that Problem 3.2 has a feasible solution for $i = i_0$. Thus, from the constraint (3.37) in Problem 3.2, we obtain $x(k_1) = A^{i_0}x(k_0) + \sum_{i=1}^{i_0} A^{i-1}Bu^*(k_0) \in \mathcal{S}$, which means that $\mathcal{I}(x(k_1))$ is non-empty. By recursively following this argument, it is shown that $x(t_k) \in \mathcal{S}$ for all $k \in \mathbb{N}$, which follows that $\mathcal{I}(x(t_k))$ is non-empty for all $k \in \mathbb{N}$.

Let us now we prove the claim (ii). Since $i_k \in \mathcal{J}(x(t_k))$, $\forall k \in \mathbb{N}$, it holds from (3.44) that:

$$\Psi_{\mathcal{S}}(x(t_{k+1})) \leq \gamma_2 \Psi_{\mathcal{S}}(x(t_k)) \quad (3.45)$$

with $\gamma_2 < 1$. Therefore, by regarding $\Psi_{\mathcal{S}}(\cdot)$ as a set-induced Lyapunov function candidate (see Lemma 3.3), the Lyapunov function is strictly decreasing and the state trajectory is asymptotically stabilized to the origin. This completes the proof. \square

3.2.4 Discussions on computational complexity

So far, two self-triggered strategies for LTI systems are proposed; *multiple discretizations approach*, and *contractive-set based approach*. In both approaches, a set of optimal control problems are provided under different transmission time intervals, and the controller determines a suitable one among them. The main difference between the two approaches is that, in the second approach the notion of contractive set has been incorporated, aiming to alleviate computational

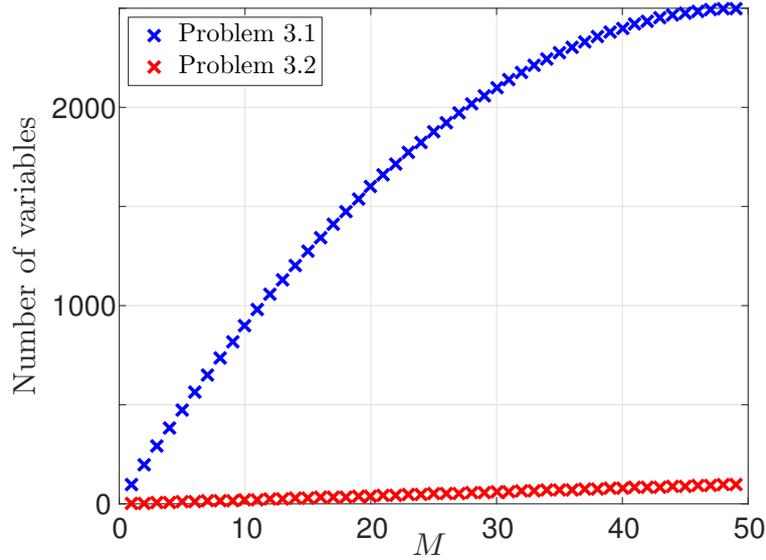


FIGURE 3.12: Number of decision variables used for Problem 3.1 and Problem 3.2.

TABLE 3.3: Number of decision variables used in Algorithm 3.1 ($N_p = 50$) and Algorithm 3.2.

M	10	20	30	40	50
Algorithm 3.1	900	1600	2100	2400	2500
Algorithm 3.2	20	40	60	80	100

complexity of solving the optimal control problems. In this section, it is illustrated that the computational complexity is indeed alleviated by evaluating the number of decision variables used in the optimal control problem.

In Problem 3.1, the total number of state and control variables used in the optimal control problem for each $i \in \{1, \dots, M\}$ is given by $2(N_p - i + 1)$, i.e., the number of state variables plus the number of control variables. Thus, the total number of decision variables is given by $2(N_p + N_p - 1 + N_p - 2 + \dots + N_p - M + 1) = 2MN_p - M^2$. On the other hand, in Problem 3.2, the total number of decision variable is given by $2M$, since only *one* state variable and *one* control variable is utilized for each $i \in \{1, \dots, M\}$. Thus, the number of decision variables in Problem 3.2 is smaller by $2MN_p - 2M - M^2$ than the ones for solving Problem 3.1. Figure 3.12 illustrates the numbers of decision variables used to solve Problem 3.1 for all $i \in \{1, \dots, M\}$ (i.e., $2MN_p - M^2$ with $N_p = 50$) and the ones

to solve Problem 3.2 (i.e., $2M$), which are plotted against different selections of M . Table 3.3 illustrates detailed number of decision variables for some M . From the figure and the table, while for both cases the number of decision variables monotonically increases as M is selected larger, Problem 3.2 uses a much smaller number of decision variables than Problem 3.1. In the simulation example in the next section, it is demonstrated that the calculation time becomes indeed smaller by incorporating the contractive set.

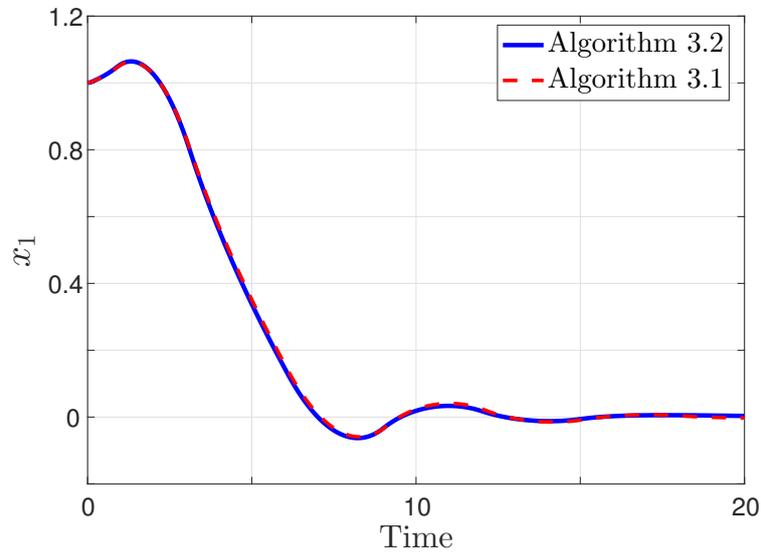
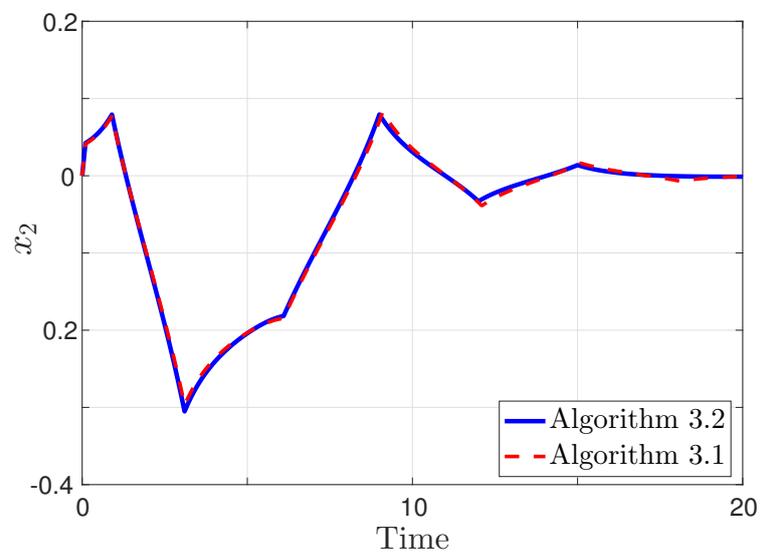
3.2.5 Simulation results

(*Example 3.4*): Similarly to Example 3.2, consider again a control problem of an inverted pendulum on a cart:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -mg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/\ell & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/M\ell \end{bmatrix} u, \quad (3.46)$$

with $m = 1$, $M = 5$, $\ell = 3$ and $g = 9.8$. The system is discretized under the sampling time interval $\delta = 0.1$, and the λ -contractive set \mathcal{S} is computed with $\lambda = 0.98$ and the tuning parameters are set as $\beta_2 = 1$, $\gamma_2 = 0.9$.

Figure 3.13 illustrates the resulting state trajectories of x_1 and x_2 , and Fig. 3.14 illustrates the ones of x_3 and x_4 . In the figure, blue lines represent the state trajectories by applying Algorithm 3.2, and red dotted lines represent the ones by applying Algorithm 3.1. While applying Algorithm 3.1, the parameters are set as $Q = I_2$, $R = 1$ and $T_p = 10$ that is the same as the one presented in the previous section (see Example 3.2). Figure 3.15 illustrates the corresponding control input, and Table 3.4 illustrates the convergence time when the state enters the small region around the origin ($\|x\| \leq 0.001$), as well as the number of transmission instants during the time interval $t \in [0, 30]$. From these results, all

(a) State trajectories of x_1 .(b) State trajectory of x_2 .FIGURE 3.13: State trajectories of x_1 and x_2 by implementing Algorithm 3.2 (blue solid lines) and Algorithm 3.1 (red dotted lines).

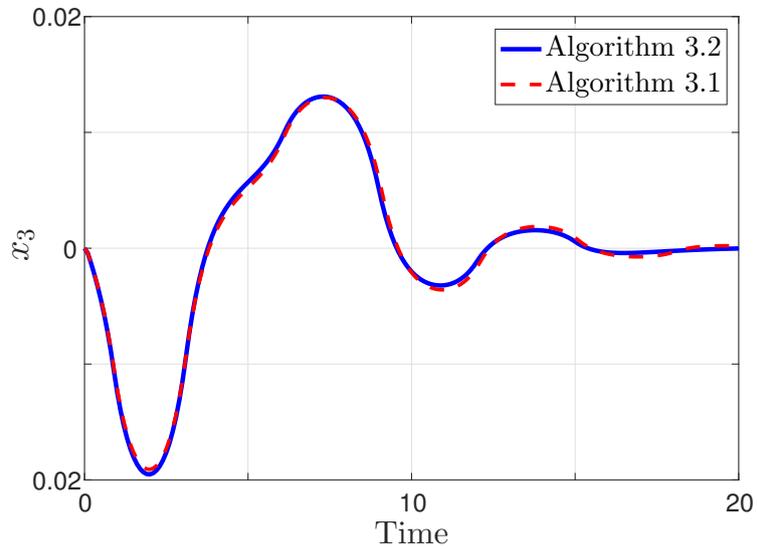
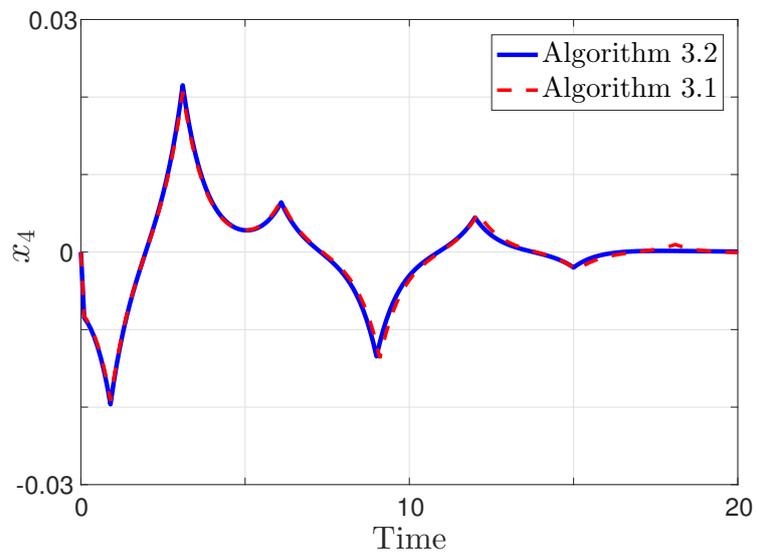
(a) State trajectories of x_3 .(b) State trajectories of x_4 .FIGURE 3.14: State trajectories of x_3 and x_4 by implementing Algorithm 3.2 (blue solid lines) and Algorithm 3.1 (red dotted lines).

TABLE 3.4: Convergence time when the state trajectory enters the region ($\|x\| \leq 0.001$) and the number of transmission instants during the time period $t \in [0, 30]$.

	Algorithm 3.2	Algorithm 3.1
Convergence time	19.9	20.1
Transmission instants	118	120

state trajectories are asymptotically stabilized to the origin by applying Algorithm 3.2, with a similar convergence and the number of transmission instants to those under Algorithm 3.1.

Let us now take a look at computational complexity for both Algorithm 3.1 and 3.2 as described in Section 3.2.4. Both Algorithm 3.1 and 3.2 are implemented under different selection of M , and compute the average calculation time to obtain the control input and the transmission time interval (i.e., the average calculation time from step (ii) to step (iii) in Algorithm 3.1, 3.2). Here, the average has been taken over all transmission time instants over the time period $t \in [0, 30]$. The results are shown in Fig. 3.16 and the concrete values are illustrated in Table 3.5 for some M . From the figure, the calculation times under both algorithms increase as the parameter M increases. This is because the number of decision variables increases for both cases as M is selected larger which has been also seen in Section 3.2.4. Still, it can be seen from the figure and the table, that the calculation time becomes much smaller by applying Algorithm 3.2 than by applying Algorithm 3.1. To conclude, it is shown in this example that:

- Algorithm 3.2 achieves similar control performance and communication reduction to Algorithm 3.1.
- The calculation time becomes smaller by applying Algorithm 3.2 than by applying Algorithm 3.1, which validates the effectiveness of incorporating the contractive set.

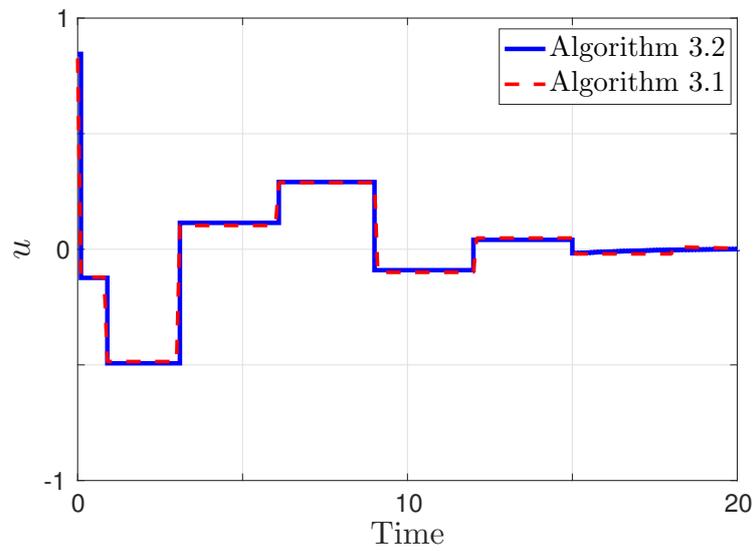


FIGURE 3.15: Applied control inputs by applying Algorithm 3.2 (blue solid line) and Algorithm 3.1 (red dotted line).

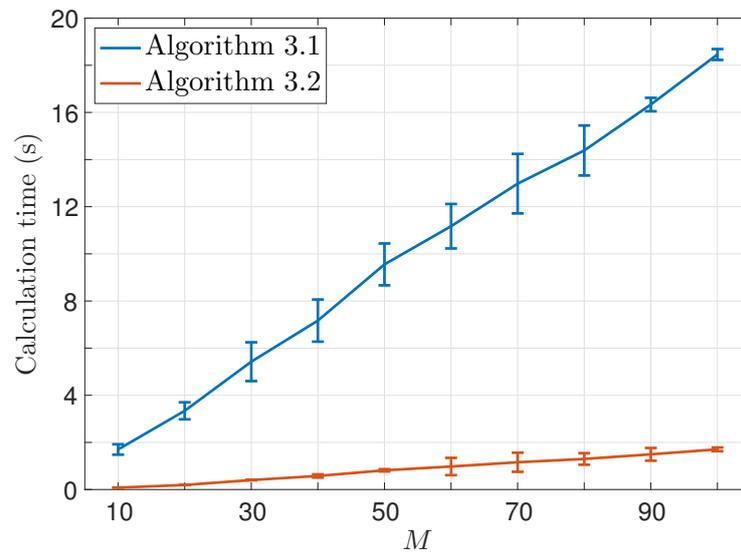


FIGURE 3.16: Calculation times against M under Algorithm 3.1 (blue) and Algorithm 3.2 (red).

TABLE 3.5: Calculation times against M under Algorithm 3.1 and Algorithm 3.2.

M	10	30	50	70	90
Algorithm 3.1	1.70	5.42	9.55	13.0	16.3
Algorithm 3.2	0.0801	0.402	0.815	1.158	1.493

3.3 Summary

In this chapter, the author proposes two different types of self-triggered MPC schemes for LTI systems. In both control schemes, communications between the plant and the controller as well as solving the optimal control problem are given only when they are needed, aiming at reducing communication load for NCSs. In the first approach, the author formulates a set of optimal control problems under a different discretization schemes, and the controller selects the suitable one among them by evaluating both control performance and stability. Feasibility and stability are rigorously shown by guaranteeing that the optimal cost as a Lyapunov function candidate is decreasing. Finally, some illustrative examples validate the effectiveness of the proposed approach by considering a control problem of spring-mass-damper systems and an inverted pendulum on a cart.

In the second approach, the author incorporates the notion of contractive set when formulating the optimal control problem, aiming at overcoming the computational drawback of the first proposal. It is shown that incorporating the contractive set achieves the reduction of computational complexity by reducing the number of decision variables required to solve the optimal control problem. The proposed self-triggered strategy is given by evaluating both the control performance and stability. Both feasibility and asymptotic stability of the origin are rigorously guaranteed by showing that the gauge function as a Lyapunov function candidate is decreasing. Finally, a simulation example validates the effectiveness of the proposed scheme by considering a control problem of an inverted pendulum. In particular, it is shown that the calculation

time is reduced by applying the proposed approach (Algorithm 3.2) by making a comparison with the multiple discretization approach (Algorithm 3.1).

Chapter 4

Aperiodic MPC for Nonlinear Input-affine systems

In the previous chapter, self-triggered strategies are derived for LTI systems. The main contribution of this chapter is to propose a new self-triggered formulation for *nonlinear* input-affine dynamical systems, which are thus provided for a wider class of systems than the one presented in the previous chapter. The triggering condition is given by evaluating the optimal cost as a Lyapunov function candidate, so that the closed-loop state trajectory is asymptotically stabilized to the origin. Moreover, the author considers the case when more than one control samples are allowed to be transmitted over a communication network. More specifically, given that the optimal control trajectory is obtained by solving the optimal control problem, the author provides a way of choosing suitable control samples that should be transmitted to the plant. Here, the control samples are selected such that both stability is guaranteed and the transmission time interval becomes as long as possible to reduce communication load. Stability analysis is given by guaranteeing that a positive inter-transmission time interval is always guaranteed and the optimal cost as a Lyapunov function candidate is decreasing. Finally, some simulation examples are given to validate the proposed self-triggered scheme by considering both linear and nonlinear control systems.

4.1 Problem formulation

Consider again the networked control system in Fig. 1.1. In this chapter, it is assumed that the dynamics of the plant are described by the nonlinear input-affine dynamical systems:

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

where $x \in \mathbb{R}^n$ denotes the state of the plant and $u \in \mathbb{R}^m$ denotes the control input. Assume that the constraint for the control input is given by $\|u\| \leq u_{\max}$.

Definition 4.1 (Control Objective). *Our control objective is to asymptotically stabilize the system (4.1) to the origin, i.e., $x(t) \rightarrow 0$ as $t \rightarrow \infty$.*

To achieve the control objective, it is assumed that the nonlinear system given by (4.1) satisfies the following conditions:

Assumption 4.1. *The nonlinear function $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is twice continuously differentiable, and the origin is an equilibrium point, i.e., $\phi(0, 0) = 0$. Moreover the function $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is Lipschitz continuous in $x \in \mathbb{R}^n$ with Lipschitz constant L_ϕ . Namely, there exists $0 \leq L_\phi < \infty$ such that*

$$\|\phi(x_1, u) - \phi(x_2, u)\| \leq L_\phi \|x_1 - x_2\| \quad (4.2)$$

for all $x_1, x_2 \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Furthermore, there exists a positive constant $L_G \geq 0$, such that $\|g(x)\| \leq L_G$ for all $x \in \mathbb{R}^n$.

Let $t_k, k \in \mathbb{N}$ be the time instants when the plant transmits the current state measurement $x(t_k)$, based on which the controller solves an Optimal Control Problem (OCP). At t_k , the controller solves the OCP involving the predictive states and the corresponding control input, which are denoted as $x(\xi), u(\xi)$, $\xi \in [t_k, t_k + T_p]$ respectively, with T_p being a prediction horizon. The following

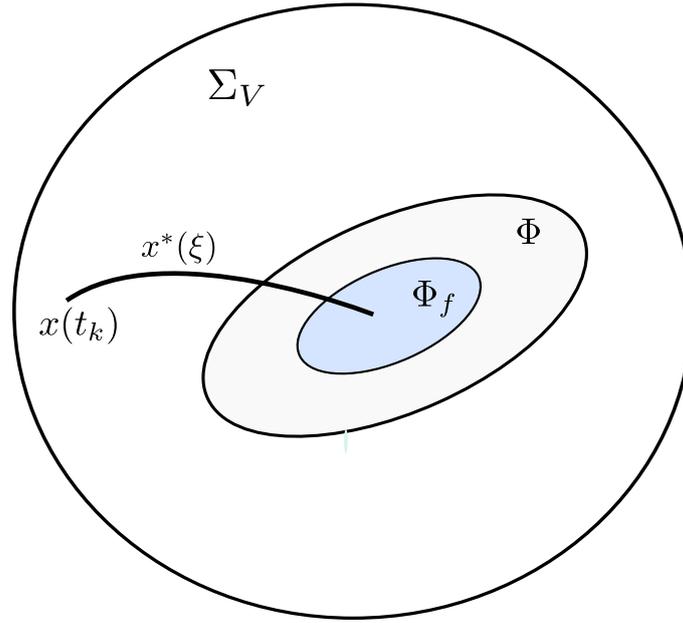


FIGURE 4.1: The illustration of three regions Σ_V , Φ , Φ_f , and an example of optimal state trajectory $x^*(\xi)$.

cost function to be minimized is given:

$$J(x(t_k), u(\cdot)) = \int_{t_k}^{t_k+T_p} F(x(\xi), u(\xi)) \, d\xi + V_f(x(t_k + T_p)),$$

where $F(x(\xi), u(\xi))$ and $V_f(x(t_k + T_p))$ are the stage and the terminal costs, which are characterized in the quadratic form as follows:

$$F(x(\xi), u(\xi)) = \|x(\xi)\|_Q^2 + \|u(\xi)\|_R^2, \quad V_f(x(t_k + T_p)) = \|x(t_k + T_p)\|_P^2. \quad (4.3)$$

In (4.3), $Q, R \succ 0$ represent the matrices for the stage cost, and $P = P^T \succ 0$ represents the matrix for the terminal cost. Based on the cost function defined above, the following OCP is proposed:

Problem 4.1 (Optimal Control Problem). At any update time t_k , $k \in \mathbb{N}_{\geq 0}$, for a given $x(t_k)$ find the optimal control input and corresponding state trajectory

$u^*(\xi), x^*(\xi), \forall \xi \in [t_k, t_k + T_p]$ that minimizes $J(x(t_k), u(\cdot))$, subject to the following constraints:

$$\begin{cases} \dot{x}(\xi) = \phi(x(\xi), u(\xi)), \xi \in [t_k, t_k + T_p] & (4.4) \\ u(\xi) \in \mathcal{U} & (4.5) \\ x(t_k + T_p) \in \Phi_f, & (4.6) \end{cases}$$

where \mathcal{U} is the control input constraint set given by

$$\mathcal{U} = \{u(\xi) \in \mathbb{R}^m : \|u(\xi)\| \leq u_{\max}, \|\dot{u}(\xi)\| \leq K_u\} \quad (4.7)$$

with $K_u > 0$ being a given positive constant. In (4.6), Φ_f denotes the terminal constraint set for a given $\varepsilon_f > 0$:

$$\Phi_f = \{x \in \mathbb{R}^n : V_f(x) \leq \varepsilon_f\}. \quad (4.8)$$

□

Regarding the control input constraint set in (4.7), the author additionally considers the constraint given by $\|\dot{u}(\xi)\| \leq K_u$, which puts a certain limit on the slope of the optimal control trajectory. Although this constraint is sometimes utilized when the actuator has a physical limitation with the rate of its position change (see e.g., [59]), in this chapter the author will make use of this constraint to guarantee asymptotic stability analyzed in the subsequent sections.

As with Chapter 3, the following assumption is made regarding stabilizability of the system around the origin:

Assumption 4.2. *There exists a positive constant $\varepsilon > 0$ and a local stabilizing controller $\kappa(x) \in \mathcal{U}$, satisfying*

$$\frac{\partial V_f}{\partial x}(f(x) + g(x)\kappa(x)) \leq -x^\top(Q + K^\top RK)x \quad (4.9)$$

for all $x \in \Phi$, where

$$\Phi = \{x \in \mathbb{R}^n : V_f(x) \leq \varepsilon\} \quad (4.10)$$

and $\varepsilon_f < \varepsilon$.

Assumption 4.2 implies that the parameter ε_f should be chosen small enough to satisfy $\varepsilon_f < \varepsilon$. Note that we have $\Phi_f \subset \Phi$. Denote by $J^*(x(t_k))$ the optimal cost obtained by solving Problem 4.1:

$$J^*(x(t_k)) = \min_{u(\cdot)} J(x(t_k), u(\cdot)).$$

Moreover, consider the following set as a stability region :

Definition 4.2. Σ_V is the set given by $\Sigma_V = \{x \in \mathbb{R}^n : J^*(x) \leq J_0\}$, where J_0 is defined such that $\Phi \subseteq \Sigma_V$.

The illustration of the three regions considered in this paper Σ_V , Φ , Φ_f , and an example of optimal trajectory $x^*(\xi)$ that is constrained to be in Φ_f by the prediction horizon T_p , are all shown in Fig. 4.1.

In this chapter, it will be shown that if the state initially starts from inside the set $x \in \Sigma_V \setminus \Phi$, the state trajectory enters Φ in finite time. Since the local control law $\kappa(x) = Kx$ is given from Assumption 4.2, the system (4.1) can be stabilized by utilizing $\kappa(x)$ once the state reaches Φ , without needing to solve the OCP. For this reason, the author considers that the control law switches from the solution to Problem 4.1 to the utilization of $\kappa(x)$ once the state enters Φ . This control scheme is in general referred to as *Dual-mode MPC*, and is adopted in many works in the literature, see e.g., [60], [42]. The following properties are satisfied for the stage and terminal costs F, V_f :

Lemma 4.1. *There exist \mathcal{K}_∞ functions $\alpha_1, \alpha_2 : \mathbb{R} \rightarrow \mathbb{R}$, such that $F(x, u) \geq \alpha_1(\|x\|)$, $V_f(x) \leq \alpha_2(\|x\|)$. Moreover, there exist $0 \leq L_F < \infty$, $0 \leq L_{V_f} < \infty$ such that $F(x, u)$ and $V_f(x)$ are Lipschitz continuous in $x \in \Sigma_V$ with the Lipschitz constants $0 < L_F < \infty$, $0 < L_{V_f} < \infty$.*

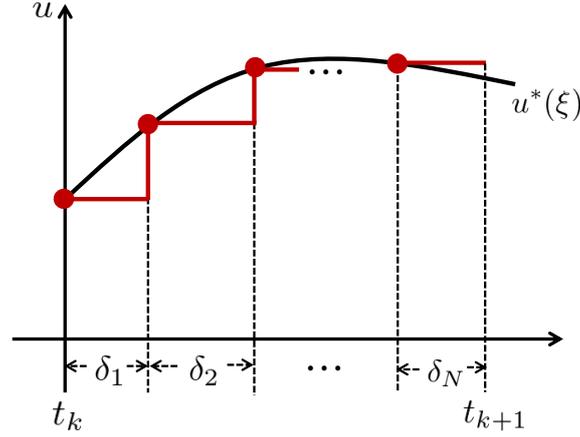


FIGURE 4.2: Based on the optimal control trajectory (black line), the controller picks up N control input samples (red circles) and these samples are transmitted to the plant and applies them as sample-and-hold fashion (red line).

Proof. For the Lipschitz continuity of $F(x, u)$ and $V_f(x)$, see the proof of Lemma 1 in [61]. The existence of \mathcal{K}_∞ functions is trivial since we have

$$F(x, u) = \|x\|_Q^2 + \|u\|_R^2 \geq \|x\|_Q^2 \geq \lambda_{\min}(Q)\|x\|^2, \quad (4.11)$$

for all $x \in \mathbb{R}^n, u \in \mathbb{R}^m$ and $V_f(x) \leq \lambda_{\max}(P)\|x\|^2, \forall x \in \mathbb{R}^n$. Hence, letting $\alpha_1(\|x\|) = \lambda_{\min}(Q)\|x\|^2$ and $\alpha_2(\|x\|) = \lambda_{\max}(P)\|x\|^2$, which are both \mathcal{K}_∞ functions, we have $F(x, u) \geq \alpha_1(\|x\|), V_f(x) \leq \alpha_2(\|x\|)$. \square

In the following, let the optimal control input and the state trajectories obtained by Problem 4.1 be given by

$$u^*(\xi), \quad x^*(\xi), \quad \xi \in [t_k, t_k + T_p], \quad (4.12)$$

where $x^*(t_k) = x(t_k)$.

Note that in earlier results of MPC framework for continuous-time systems, e.g., [61], [60], [62], [42], [38], the current and future continuous optimal control trajectory $u^*(\xi)$ is considered to be applied to the plant for $\xi \in [t_k, t_{k+1}]$. However, this situation may not be applied to the networked control systems, since sending *continuous* information requires an infinite transmission bandwidth.

Therefore, it is considered that only N ($N \in \mathbb{N}_{\geq 1}$) control input samples, i.e.,

$$\left\{ u^*(t_k), u^*(t_k + \delta_1), \dots, u^*(t_k + \sum_{i=1}^N \delta_i) \right\} \quad (4.13)$$

should be determined to be picked up by the controller and then transmitted to the plant. The plant then applies the obtained control inputs in a sample-and-hold fashion, see the illustration in Fig. 4.2. As shown in Fig. 4.2, $t_{k+1} = t_k + \sum_{i=1}^N \delta_i$ represents the next transmission time when the plant sends $x(t_{k+1})$ as the new current state information, and is obtained by the self-triggered strategy provided in the next section. Furthermore, by making use of the flexibility of selecting control samples when multiple control inputs are allowed to be transmitted (namely when $N > 1$), the author will provide an efficient way of how to pick up control samples to be transmitted, such that the reduction of the communication load is achieved.

Remark 4.1 (On the dual-mode strategy). *Since $\kappa(x)$ is a continuous control law, applying $\kappa(x)$ over the network as a dual mode strategy would in fact require an infinite transmission bandwidth. One way to avoid this issue is to apply $\kappa(x)$ under sample-and-hold fashion;*

$$u(t) = \kappa(x(t_k)), \quad t \in [t_k, t_k + \delta_l],$$

where the sampling time δ_l is constant and needs to be small enough such that asymptotic stability is still guaranteed in $x \in \Phi$; see [63] for the related analysis. Another way would be to apply $\kappa(x)$ directly at the plant as a stand-alone to stabilize the system, without needing any communication with the controller as soon as x enters Φ . This situation could be the case when the computation of $\kappa(x)$ is possible locally at the plant, while at the same time it is only feasible to solve the optimal control problems through the networked controller due to computational limitations. For this case, $\kappa(x)$ does not need to be discretized since no communication is required locally at the plant. \square

4.2 Self-triggered strategy

In this section a self-triggered condition is derived for MPC under sample-and-hold controllers. Suppose again that at t_k Problem 4.1 is solved, providing the optimal control input and the state trajectory denoted as (4.12) and the optimal cost as $J^*(x(t_k))$. Denoting $\Delta_n = \sum_{i=1}^n \delta_i < T_p$ for $1 \leq n \leq N$, let $x(t_k + \Delta_n)$ be the actual state when sample-and-hold controllers $\{u^*(t_k), \dots, u^*(t_k + \Delta_n)\}$ are applied with sampling intervals $\delta_1, \dots, \delta_n$. Moreover, let $J^*(x(t_k + \Delta_N))$ be the optimal cost obtained by solving Problem 4.1 based on the new current state $x(t_k + \Delta_N)$. Then, the self-triggered condition, which determines the next transmission time t_{k+1} , is obtained by checking if the optimal cost regarded as a Lyapunov candidate is guaranteed to decrease, i.e.,

$$J^*(x(t_k + \Delta_N)) - J^*(x(t_k)) < 0. \quad (4.14)$$

For deriving this condition more in detail, let us first recap that the following holds (see as well as *Lemma 3* in [38] or *Theorem 2.1* in [64]) :

$$J^*(x^*(t_k + \Delta_N)) - J^*(x(t_k)) \leq - \int_{t_k}^{t_k + \Delta_N} F(x^*(\xi), u^*(\xi)) d\xi, \quad (4.15)$$

where $J^*(x^*(t_k + \Delta_N))$ is the optimal cost obtained by solving Problem 4.1 if the current state at $t_k + \Delta_N$ is $x^*(t_k + \Delta_N)$. This means that the optimal cost would be guaranteed to decrease if the actual state *followed* the optimal state trajectory $x(\xi) = x^*(\xi)$ for $\xi \in [t_k, t_k + \Delta_N]$. From (4.15), we obtain

$$\begin{aligned} J^*(x(t_k + \Delta_N)) - J^*(x(t_k)) &\leq J^*(x(t_k + \Delta_N)) - J^*(x^*(t_k + \Delta_N)) \\ &\quad - \int_{t_k}^{t_k + \Delta_N} F(x^*(\xi), u^*(\xi)) d\xi, \end{aligned} \quad (4.16)$$

where $F(x^*(\xi), u^*(\xi))$, $\xi \in [t_k, t_k + T_p]$ is known at t_k when the OCP is solved.

Remark 4.2 (Feasibility of Problem 4.1). *In order to obtain the stability property given by (4.15), one can see that the feasibility of Problem 4.1 needs to be guaranteed,*

see e.g., [38]. Regarding establishing the feasibility of Problem 1, the existing procedures of event-triggered MPC (see e.g., [65]) or periodic MPC (see e.g., [38]) can be utilized; we can consider a feasible controller candidate given by $\bar{u}(s) = u^*(s)$ for all $s \in [t_{k+1}, t_k + T_p]$ and $\kappa(\bar{x}(s))$ for all $s \in (t_k + T_p, t_{k+1} + T_p]$, to obtain (4.15). However, compared with the existing procedures, the condition $\dot{\kappa}(x) \leq K_u$ is additionally required for the existence of the local controller, such that this controller candidate becomes admissible. More specifically, since we have $\dot{\kappa}(x) = \frac{\partial \kappa(x)}{\partial x} \phi(x, \kappa(x))$, K_u must satisfy

$$K_u \geq \max_{x \in \Omega(\varepsilon_f)} \left\{ \left\| \frac{\partial \kappa(x)}{\partial x} \cdot \phi(x, \kappa(x)) \right\| \right\},$$

and this needs to be computed off-line. \square

For notational simplicity in the sequel, let $E_x(\delta_1, \dots, \delta_n)$ be the upper bound of $\|x^*(t_k + \Delta_n) - x(t_k + \Delta_n)\|$ for $1 \leq n \leq N$. The following lemmas are useful to derive a more detailed expression of (4.16):

Lemma 4.2. *Under the Assumptions 4.1 – 4.4, the optimal cost $J^*(x)$ is Lipschitz continuous in $x \in \Sigma_V$, with Lipschitz constant L_J given by*

$$L_J = \left(\frac{L_F}{L_\phi} + L_{V_f} \right) e^{L_\phi T_p} - \frac{L_F}{L_\phi}. \quad (4.17)$$

Proof. Consider the optimal costs $J^*(x_1)$, $J^*(x_2)$ obtained by different initial states $x(0) = x_1$, $x(0) = x_2$. Here the current time is assumed to be 0 without loss of generality. Let $x_1^*(\xi)$, $u_1^*(\xi)$ ($x_1^*(0) = x_1$), and $x_2^*(\xi)$, $u_2^*(\xi)$ ($x_2^*(0) = x_2$) be the optimal state and control trajectory for $s \in [0, T_p]$, obtained by solving Problem 1. These optimal costs are then given by

$$J^*(x_i) = \int_0^{T_p} F(x_i^*(\xi), u_i^*(\xi)) d\xi + V_f(x_i^*(T_p)) \quad (4.18)$$

for $i = 1, 2$. Now consider the difference $J^*(x_1) - J^*(x_2)$. Assume that from the initial state x_1 , an alternative control input $\bar{u}_1(\xi) = u_2^*(\xi) \in \mathcal{U}$ ($\xi \in [0, T_p]$) is applied and let $\bar{x}_1(\xi)$ be the corresponding state obtained by applying $\bar{u}_1(\xi)$.

Also let $\bar{J}(x_1)$ be the corresponding cost. Since $J^*(x_1) \leq \bar{J}(x_1)$, we obtain

$$J^*(x_1) - J^*(x_2) \leq \int_0^{T_p} L_F \|\bar{x}_1(\xi) - x_2^*(\xi)\| d\xi + L_{V_f} \|\bar{x}(T_p) - x_2^*(T_p)\|, \quad (4.19)$$

where the Lipschitz continuities of F and V_f are used. From Gronwall-Bellman inequality, we have $\|\bar{x}_1(\xi) - x_2^*(\xi)\| \leq e^{L_\phi \xi} \|x_1 - x_2\|$ for $\xi \in [0, T_p]$. Thus, we obtain

$$\begin{aligned} J^*(x_1) - J^*(x_2) &\leq L_F \|x_1 - x_2\| \int_0^{T_p} e^{L_\phi s} ds + L_{V_f} e^{L_\phi T_p} \|x_1 - x_2\| \\ &= \left\{ \left(\frac{L_F}{L_\phi} + L_{V_f} \right) e^{L_\phi T_p} - \frac{L_F}{L_\phi} \right\} \|x_1 - x_2\|. \end{aligned}$$

Thus the proof is complete. \square

In order to derive the self-triggered condition, let us first consider the case $N = 1$ for simplicity. Denote by $E_x(\delta_1)$ the upper bound of the error between the predictive state and the actual state $\|x^*(t_k + \delta_1) - x(t_k + \delta_1)\|$. Then, we have the following lemma:

Lemma 4.3. *Suppose that Problem 4.1 is solved at t_k , which provides the optimal control trajectory $u^*(\xi)$ and the corresponding state trajectory $x^*(\xi)$ for all $\xi \in [t_k, t_k + T_p]$. Suppose also, that $u^*(t_k)$ is applied constantly for the time interval $[t_k, t_k + \delta_1]$. Then, $E_x(\delta_1)$ is given by*

$$E_x(\delta_1) = \frac{2K_u L_G}{L_\phi^2} (e^{L_\phi \delta_1} - 1) - \frac{2K_u L_G}{L_\phi} \delta_1. \quad (4.20)$$

Proof. Observe that $x(t_k + \delta_1)$ and $x^*(t_k + \delta_1)$ are given by

$$\begin{aligned} x(t_k + \delta_1) &= x(t_k) + \int_{t_k}^{t_k + \delta_1} \phi(x(\xi), u^*(t_k)) d\xi, \\ x^*(t_k + \delta_1) &= x(t_k) + \int_{t_k}^{t_k + \delta_1} \phi(x^*(\xi), u^*(\xi)) d\xi. \end{aligned}$$

We obtain

$$\|x(t_k + \delta_1) - x^*(t_k + \delta_1)\| \leq \int_{t_k}^{t_k + \delta_1} L_\phi \|x(\xi) - x^*(\xi)\| d\xi + \frac{1}{2} L_G K_u \delta_1^2, \quad (4.21)$$

where the following is used

$$\|g(x(\xi))(u^*(t_k) - u^*(\xi))\| \leq L_G K_u (\xi - t_k) \quad (4.22)$$

for all $\xi \in [t_k, t_k + \delta_1]$ from Assumption 4.1 and the control input constraint $\|\dot{u}^*(\xi)\| \leq K_u$. Therefore, by applying the Gronwall-Bellman inequality [52], we obtain

$$\|x(t_k + \delta_1) - x^*(t_k + \delta_1)\| \leq \frac{2K_u L_G}{L_\phi^2} (e^{L_\phi \delta_1} - 1) - \frac{2K_u L_G}{L_\phi} \delta_1$$

and thus we have (4.20). This completes the proof. \square

Now, the following lemma illustrates an extension to the general case of N control samples:

Lemma 4.4. *Suppose that the sample-and-hold controllers given by (4.13) are applied to the plant (4.1) from t_k . Then, the upper bound of $\|x^*(t_k + \Delta_N) - x(t_k + \Delta_N)\|$, which we denote by $E_x(\delta_1, \dots, \delta_N)$, is obtained by the following recursion for $2 \leq n \leq N$:*

$$E_x(\delta_1, \dots, \delta_n) = E_x(\delta_1, \dots, \delta_{n-1}) e^{L_\phi \delta_n} + h_x(\delta_n) \quad (4.23)$$

with $E_x(\delta_1) = h_x(\delta_1)$, where

$$h_x(t) = \frac{2K_u L_G}{L_\phi^2} (e^{L_\phi t} - 1) - \frac{2K_u L_G}{L_\phi} t. \quad (4.24)$$

Proof. Assume that $E_x(\delta_1, \dots, \delta_{n-1})$ is given for $n \geq 2$. We obtain

$$\begin{aligned} & \|x(t_k + \Delta_n) - x^*(t_k + \Delta_n)\| \\ & \leq \|x(t_k + \Delta_{n-1}) - x^*(t_k + \Delta_{n-1})\| \\ & \quad + \int_{t_k + \Delta_{n-1}}^{t_k + \Delta_n} L_\phi \|x(\xi) - x^*(\xi)\| d\xi + \frac{1}{2} L_G K_u \delta_n^2. \end{aligned} \quad (4.25)$$

The only difference between (4.21) and (4.25) is that the initial difference $\|x(t_k + \Delta_{n-1}) - x^*(t_k + \Delta_{n-1})\|$ that is upper bounded by $E_x(\delta_1, \dots, \delta_{n-1})$ is included in (4.25). By applying the Gronwall-Bellman inequality again, we obtain

$$\begin{aligned} & \|x(t_k + \Delta_n) - x^*(t_k + \Delta_n)\| \\ & \leq E_x(\delta_1, \dots, \delta_{n-1}) e^{L_\phi \delta_n} + \frac{2K_u L_G}{L_\phi^2} (e^{L_\phi \delta_n} - 1) - \frac{2K_u L_G}{L_\phi} \delta_n. \end{aligned}$$

Thus (4.23) holds. Therefore, the upper bound $E_x(\delta_1, \dots, \delta_N)$ is obtained by using $E_x(\delta_1) = h_x(\delta_1)$ at first, and then recursively using (4.23) for $n = 2, \dots, N$. This completes the proof. \square

Using Lemma 4.4, (4.16) is rewritten by

$$\begin{aligned} & J^*(x(t_k + \Delta_N)) - J^*(x(t_k)) \\ & \leq L_J E_x(\delta_1, \dots, \delta_N) - \int_{t_k}^{t_k + \Delta_N} F(x^*(\xi), u^*(\xi)) d\xi. \end{aligned}$$

Therefore, letting

$$E_x(\delta_1, \dots, \delta_N) < \frac{\sigma}{L_J} \int_{t_k}^{t_k + \Delta_N} F(x^*(\xi), u^*(\xi)) d\xi, \quad (4.26)$$

where $0 < \sigma < 1$, we obtain

$$\begin{aligned} J^*(x(t_k + \Delta_N)) - J^*(x(t_k)) &< (\sigma - 1) \int_{t_k}^{t_k + \Delta_N} F(x^*(\xi), u^*(\xi)) d\xi \\ &< 0 \end{aligned}$$

and the cost is guaranteed to decrease. In our proposed self-triggered MPC strategy, therefore, the next transmission time t_{k+1} is determined by the time when the violation of (4.26) takes place, i.e.,

$$t_{k+1} = \inf \{ \hat{t}_{k+1} \mid \hat{t}_{k+1} > t_k, \Gamma(\delta_1, \dots, \delta_N) = 0 \}, \quad (4.27)$$

where $\hat{t}_{k+1} = t_k + \sum_{i=1}^N \delta_i$ and $\Gamma(\delta_1, \dots, \delta_N)$ is given by

$$\Gamma(\delta_1, \delta_2, \dots, \delta_N) = E_x(\delta_1, \dots, \delta_N) - \frac{\sigma}{L_J} \int_{t_k}^{\hat{t}_{k+1}} F(x^*(\xi), u^*(\xi)) d\xi.$$

Note that between t_k and t_{k+1} , there exists an infinite number of patterns for the selection of sampling time intervals $\delta_1, \dots, \delta_N$. Since $E_x(\delta_1, \dots, \delta_N)$ in the left-hand-side (L.H.S) of (4.26) depends on these intervals, the way to select $\delta_1, \dots, \delta_N$ clearly affects the next transmission time t_{k+1} obtained by (4.27). In the next section, a way to adaptively select $\delta_1, \dots, \delta_N$ is proposed such that the communication load can be reduced as much as possible.

Remark 4.3 (On reducing conservativeness). *The R.H.S term in (4.26) becomes smaller as $x \rightarrow 0$, and thus the triggering condition (4.26) becomes more conservative as the state approaches origin, which needs increasingly number of transmissions. Therefore, it is important to reduce the conservativeness of the triggering condition even though the state becomes smaller. To achieve this, note in $E_x(\delta_1, \dots, \delta_N)$ that K_u stays constant for all the time. However, as $x \rightarrow 0$, we have $u \rightarrow 0$ since $\phi(0, 0) = 0$, and thus the slope of the optimal control input $\| \dot{u}^* \|$ may also become smaller. Therefore, one way to reduce the conservativeness would be to replace the constant parameter*

K_u in (4.26) with

$$K_u^*(t_k) = \max_{s \in [t_k, t_k + T_p]} \|\dot{u}^*(s)\| \leq K_u, \quad (4.28)$$

which varies at each transmission time. If we find $K_u^*(t_k) < K_u$, the upper bound E_x in (4.26) becomes smaller to achieve less conservative result. Another way of reducing the conservativeness is to make the Lipschitz constant L_J in (4.17) as small as possible, and this corresponds to achieving smaller Lipschitz constants L_F and L_{V_f} . Several ways to reduce these values, such as changing the type of norm or using control parametrizations, are described in [66]. \square

4.3 Choosing sampling time intervals

In this section an efficient way of adaptively selecting sampling intervals $\delta_1, \delta_2, \dots, \delta_N$ is given, aiming at reducing the communication load for networked control systems. In the following, let $\delta_1^*, \delta_2^*, \dots, \delta_N^*$ be the selected sampling intervals by the controller to transmit corresponding optimal control samples. In order to satisfy (4.26) as long as possible, one may select the intervals $\delta_1^*, \dots, \delta_N^*$ such that $E_x(\delta_1, \dots, \delta_N)$ is minimized. This is formulated as follows:

$$t_{k+1} = \inf \{ \hat{t}_{k+1} \mid \hat{t}_{k+1} > t_k, \Gamma(\delta_1^*, \dots, \delta_N^*) = 0 \}, \quad (4.29)$$

where $\hat{t}_{k+1} = t_k + \sum_{i=1}^N \delta_i^*$ and $\delta_1^*, \dots, \delta_N^*$ are optimal sampling time intervals between t_k and \hat{t}_{k+1} , such that $E_x(\delta_1, \dots, \delta_N)$ is minimized, i.e.,

$$\delta_1^*, \delta_2^*, \dots, \delta_N^* = \arg \min_{\delta_1, \delta_2, \dots, \delta_N} E_x(\delta_1, \dots, \delta_N), \quad (4.30)$$

subject to $\hat{t}_{k+1} = t_k + \sum_{i=1}^N \delta_i$. In this approach, it is required to solve the optimization problem (4.30) for each \hat{t}_{k+1} and check if the self-triggered condition (4.26) is satisfied. This means that the controller needs to both solve (4.30) and check (4.26) until the violation $\Gamma(\delta_1^*, \dots, \delta_N^*) = 0$ occurs. Therefore, trying to

obtain (4.29) is in fact not practical from a computational point, since the optimization problem (4.30) needs to be solved for a possibly large number of times. Moreover, since the solution to (4.30) does not provide an explicit solution, numerical calculations of solving (4.30) would become more complex as N becomes larger.

Therefore, the author proposes a following alternative algorithm to make the problem of searching for the sampling intervals easier. In contrast to the above approach, this scheme requires only N local optimizations to obtain the sampling intervals, and furthermore, a more explicit solution can be found.

Algorithm 4.1 (Choosing sampling time intervals)

- (i) Suppose that only $u^*(t_k)$ is applied for $t \geq t_k$ as a constant controller, and find the time $t_k + \tau_1$ when the triggering condition (4.26) is violated, see the illustration in Fig. 4.3 (a). We obtain $E_x(\tau_1)$ as the upper bound of $\|x^*(t_k + \tau_1) - x(t_k + \tau_1)\|$. If $N = 1$, we set $\delta_1^* = \tau_1$.
- (ii) If $N \geq 2$, we set $\delta_1^* \in [0, \tau_1]$ in the following way. Suppose that $u^*(t_k)$ and $u^*(t_k + \delta_1)$ are applied for $[t_k, t_k + \delta_1]$, $[t_k + \delta_1, t_k + \tau_1]$ respectively. This means we obtain $E_x(\delta_1, \tau_1 - \delta_1)$ as the upper bound of $\|x(t_k + \tau_1) - x^*(t_k + \tau_1)\|$. Then, find $\delta_1^* \in [0, \tau_1]$ which maximizes the difference of two upper bounds, i.e.,

$$\delta_1^* = \arg \max_{\delta_1 \in [0, \tau_1]} \{E_x(\tau_1) - E_x(\delta_1, \tau_1 - \delta_1)\},$$

see Fig. 4.3 (b). As shown in Fig. 4.3 (c), by maximizing the above difference, $u^*(t_k + \delta_1^*)$ can continue to be applied until the time when (4.26) is again violated. Denote τ_2 as the time interval when the violation of (4.26) takes place after the time $t_k + \delta_1^*$. If $N = 2$, we set $\delta_2^* = \tau_2$.

- (iii) We follow the above steps until we get N intervals. That is, given $n - 1$ sampling intervals $\delta_1^*, \dots, \delta_{n-1}^*$ for $2 \leq n < N$, find τ_n when the triggering condition is violated to obtain $E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n)$. Then, find $\delta_n^* \in [0, \tau_n]$

maximizing $E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) - E_x(\delta_1^*, \dots, \delta_n^*, \tau_n - \delta_n^*)$, i.e.,

$$\delta_n^* = \arg \max_{\delta_n \in [0, \tau_n]} \{E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) - E_x(\delta_1^*, \dots, \delta_n^*, \tau_n - \delta_n^*)\}.$$

For the last step at $n = N$, we set $\delta_N^* = \tau_N$, as the final time interval. \square

Instead of solving the optimization problem (4.30) possibly for a very large number of times, Algorithm 4.1 requires only N local optimization problems to obtain the sampling intervals $\delta_1^*, \dots, \delta_N^*$. Algorithm 4.1 may not provide the largest possible next transmission time, since it does not minimize $E_x(\delta_1, \dots, \delta_N)$. However, as we will see through several comparisons in simulation results, Algorithm 4.1 is more practical than the method to obtain (4.29), as it requires much less computation time. Furthermore, compared with (4.30) that provides no explicit solutions, the following lemma states that the solutions to the local optimization problems can be obtained by a simple numerical procedure.

Lemma 4.5. *Given $\delta_1^*, \delta_2^*, \dots, \delta_{n-1}^*$, and τ_n for $1 \leq n < N$, the transmission interval δ_n^* maximizing $E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) - E_x(\delta_1^*, \dots, \delta_n^*, \tau_n - \delta_n^*)$ in Algorithm 4.1, step (iii), is obtained by the solution to*

$$e^{L_\phi(\tau_n - \delta_n^*)} = \frac{1}{(1 - L_\phi \delta_n^*)}. \quad (4.31)$$

Furthermore, there always exists a solution of (4.31) satisfying $0 < \delta_n^* < \tau_n$.

Proof. From (4.23), $E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n)$ is given by

$$E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) = E_x(\delta_1^*, \dots, \delta_{n-1}^*)e^{L_\phi \tau_n} + h_x(\tau_n). \quad (4.32)$$

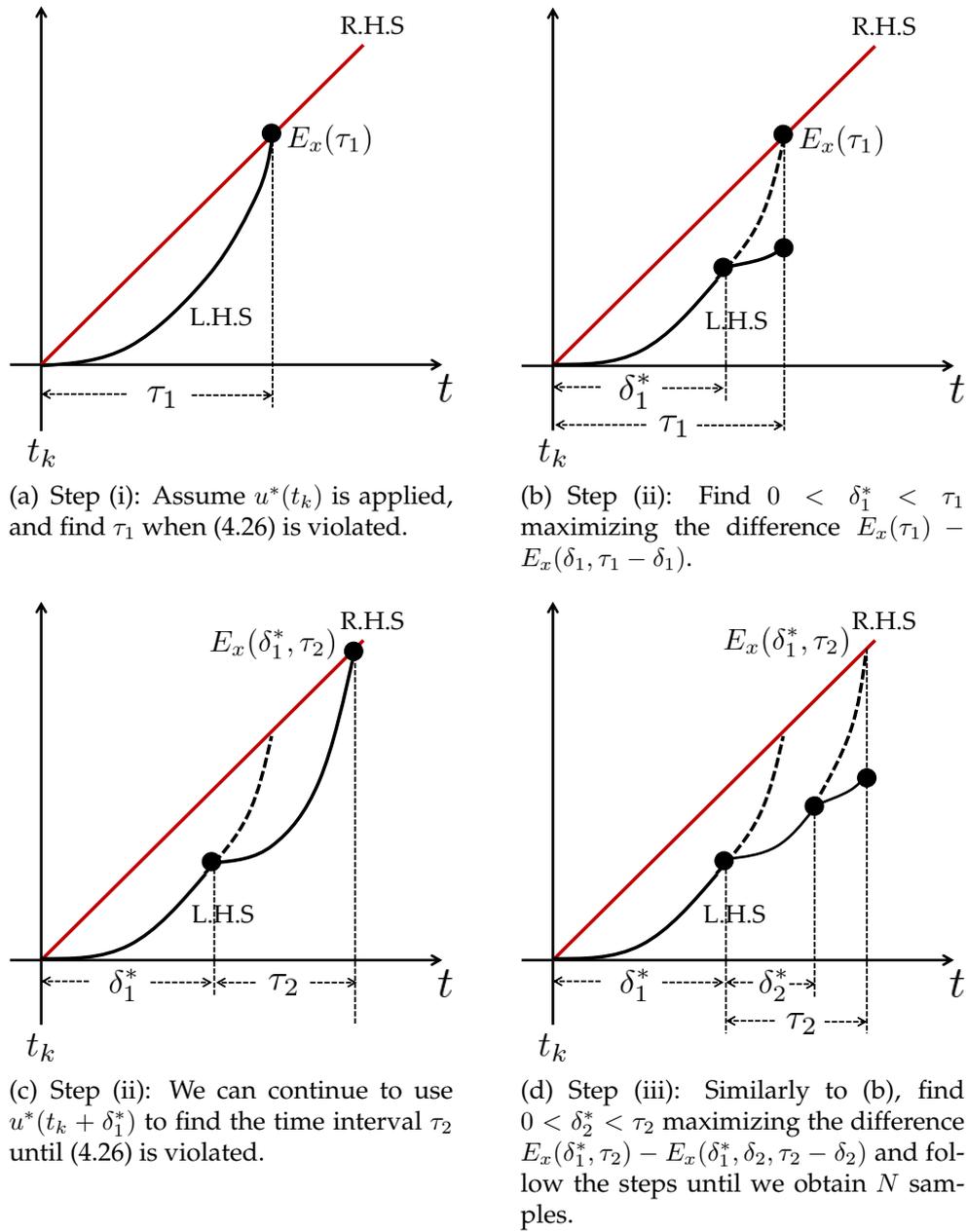


FIGURE 4.3: The way to find sampling intervals. L.H.S and R.H.S are the evolutions of left-hand-side and right-hand side in (4.26).

For $E_x(\delta_1^*, \dots, \delta_{n-1}^*, \delta_n, \tau_n - \delta_n)$, we obtain

$$\begin{aligned} & E_x(\delta_1^*, \dots, \delta_n, \tau_n - \delta_n) \\ &= E_x(\delta_1^*, \dots, \delta_{n-1}^*, \delta_n) e^{L_\phi(\tau_n - \delta_n)} + h_x(\tau_n - \delta_n) \\ &= E_x(\delta_1^*, \dots, \delta_{n-1}^*) e^{L_\phi \tau_n} + h_x(\tau_n) - \frac{2K_u L_G}{L_\phi} (e^{L_\phi(\tau_n - \delta_n)} - 1) \delta_n. \end{aligned}$$

Thus, we obtain

$$\begin{aligned} & E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) - E_x(\delta_1^*, \dots, \tau_n - \delta_n) \\ &= \frac{2K_u L_G}{L_\phi} \delta_n (e^{L_\phi(\tau_n - \delta_n)} - 1) > 0. \end{aligned} \quad (4.33)$$

Therefore, by differentiating (4.33) with respect to δ_n and solving for 0, we obtain (4.31).

Now it is shown that we can always find $0 < \delta_n^* < \tau_n$ satisfying (4.31). As $\delta_n \rightarrow 0$, we get

$$e^{L_\phi(\tau_n - \delta_n)} > \frac{1}{1 - L_\phi \delta_n}.$$

Moreover, we obtain

$$e^{L_\phi(\tau_n - \delta_n)} < \frac{1}{1 - L_\phi \delta_n}$$

as $\delta_n \rightarrow \tau_n$ if $\tau_n < 1/L_\phi$, or $\delta_n \rightarrow 1/L_\phi$ if $\tau_n > 1/L_\phi$. Therefore, there always exists δ_n^* satisfying $0 < \delta_n^* < \tau_n$. This completes the proof. \square

Lemma 4.5 states that δ_n^* can be found by solving (4.31), once τ_n is obtained. Note that the difference (4.33) is positive for any $0 < \delta_n < \tau_n$. This means that if we use larger N , then we obtain longer transmission intervals.

To conclude, the over-all self-triggered algorithm, including the OCP and Algorithm 4.1, is now stated:

Algorithm 4.2: (Self-triggered strategy via adaptive control samples selection)

- (i) At an update time t_k , $k \in \mathbb{N}_{\geq 0}$, if $x(t_k) \in \Phi$, then switch to the local controller $\kappa(x)$ to stabilize the system. Otherwise, solve Problem 4.1 to obtain $u^*(\xi)$, $x^*(\xi)$ for all $\xi \in [t_k, t_k + T_p]$.
- (ii) For a given N , calculate $\delta_1^*, \delta_2^*, \dots, \delta_N^*$ and obtain the next transmission time $t_{k+1} = t_k + \sum_{i=1}^N \delta_i^*$, according to Algorithm 4.1. Then the controller transmits the following control samples to the plant;

$$\left\{ u^*(t_k), u^*(t_k + \delta_1^*), \dots, u^*(t_k + \sum_{i=1}^N \delta_i^*) \right\}. \quad (4.34)$$

- (iii) The plant applies (4.34) in a sample-and-hold fashion, and transmits $x(t_{k+1})$ to the controller as the new current state to solve the next optimal control problem.
- (iv) $k \leftarrow k + 1$ and go back to step (i). □

Some remarks are in order below regarding Algorithm 4.2.

Remark 4.4 (*Effect of time delays*). So far some time delays arising in transmissions or calculations solving optimal control problems have been ignored. In practical applications, however, it may be important to take delays into account. A method for dealing with the delays for MPC has been proposed in [62], where the authors proposed delay compensation schemes by using forward prediction, i.e., even though the delays occur, the actual state is still able to be obtained from the system model (4.1) (see Eq. (11) in [62]). Note, however, that in order to compensate time delays and guarantee stability, the network delays need to be upper bounded. More specifically, denoting $\bar{\tau}_d$ as the total maximum time delay which could arise, then $\bar{\tau}_d$ needs to satisfy $\bar{\tau}_d < T_p - \Delta_N$ so that the inter-sampling time and the delay cannot exceed the prediction horizon T_p . Thus, assuming that $\bar{\tau}_d$ is known, the condition

$$\Delta_N < T_p - \bar{\tau}_d, \quad (4.35)$$

is required in the self-triggered strategy in addition to (4.26). □

Remark 4.5 (Effect of model uncertainties). For simplicity reasons, the effect of model uncertainties or disturbances has not yet considered. However, with a slight modification of the self-triggered condition, these effects can be taken into account. Suppose that the actual state is $x^a(t)$ and the dynamics are given by $\dot{x}^a = \phi(x^a, u) + w$ where w represents the disturbance or modeling error satisfying $\|w\| \leq w_{\max}$. In this case, the new upper bound of $\|x^*(t_k + \Delta_N) - x^a(t_k + \Delta_N)\|$, denoted as $\hat{E}_x(\delta_1, \dots, \delta_N)$ is given by

$$\hat{E}_x(\delta_1, \dots, \delta_N) = E_x(\delta_1, \dots, \delta_N) + \frac{w_{\max}}{L_\phi} (e^{L_\phi \Delta_N} - 1),$$

where Gronwall-Bellman inequality ([61]) is used for the related analysis. The corresponding self-triggered condition is thus given by replacing E_x with \hat{E}_x in (4.26). Similarly to Algorithm 4.1, it is required to obtain δ_n^* by maximizing the difference of two upper bounds \hat{E}_x . However, we can easily see that

$$\begin{aligned} & \hat{E}_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) - \hat{E}_x(\delta_1^*, \dots, \tau_n - \delta_n) \\ &= E_x(\delta_1^*, \dots, \delta_{n-1}^*, \tau_n) - E_x(\delta_1^*, \dots, \tau_n - \delta_n) \end{aligned}$$

as the effect of the disturbance can be canceled by taking the difference of the two \hat{E}_x . Thus, Algorithm 4.1 does not need to be modified as the way to obtain sampling time intervals is not affected. \square

Remark 4.6 (On the selection of the number of control samples). From (4.33) the difference of two upper bounds is always positive, so that more time is allowed for the self-triggered condition to be satisfied by setting a new sampling time (see the illustration in Fig. 4.3(c)). Thus we obtain longer transmission time intervals as N is chosen larger. However, N needs to be carefully chosen such that the network bandwidth limitation can be taken into account; large values of N may not be allowed for the network due to narrow bandwidth. Moreover, even though Algorithm 4.2 makes efficient calculations of N sampling intervals, a larger selection of N means more iterations of (4.31), which may induce larger network delays. As is already mentioned in Remark 4.4, the delays can be compensated. However, the allowable delays must be limited as shown in

(4.35). Thus, when implementing Algorithm 4.2, N needs to be appropriately selected such that it satisfies not only the constraint for network bandwidth but also for network delays fulfilling (4.35). \square

4.4 Stability analysis

In this section, stability analysis under our proposed self-triggered strategy is given. As the first step, it is shown that if the current state $x(t_k)$ is outside of Φ , there always exists a positive minimum inter-execution time for the self-triggered condition (4.26), i.e., there exists $\delta_{\min} > 0$ satisfying (4.26) for all $[t_k, t_k + \delta_{\min}]$. It will be shown only for the case where one control sample is transmitted, i.e, $N = 1$, since larger N allows for longer transmission intervals according to Lemma 4.5 and Remark 4.6.

The self-triggered condition for the case $N = 1$ is given by

$$E_x(\delta_1) < \frac{\sigma}{L_J} \int_{t_k}^{t_k + \delta_1} F(x^*(\xi), u^*(\xi)) d\xi, \quad (4.36)$$

where $x^*(t_k) = x(t_k)$ and $E_x(\delta_1) = h_x(\delta_1)$. By using $F(x, u) \geq \alpha_1(\|x\|)$ from Lemma 4.1, the condition can be replaced by

$$\int_0^{\delta_1} \left\{ \frac{\sigma}{L_J} \alpha_1(\|x^*(t_k + \eta)\|) - \frac{2K_u L_G}{L_\phi} (e^{L_\phi \eta} - 1) \right\} d\eta > 0, \quad (4.37)$$

where $h_x(\delta_1)$ is included in the integral. A sufficient condition to satisfy (4.37) is that the integrand is positive for all $0 \leq \eta \leq \delta_1$, i.e.,

$$\alpha_1(\|x^*(t_k + \eta)\|) > \frac{2K_u L_G L_J}{L_\phi \sigma} (e^{L_\phi \eta} - 1) \quad (4.38)$$

for all $0 \leq \eta \leq \delta_1$. We will thus show that if $x(t_k) \in \Sigma_V \setminus \Phi$ there exists a positive time interval $\delta_{\min} > 0$ satisfying (4.38) for all $0 \leq \eta \leq \delta_{\min}$.

Suppose at a certain time $t_k + \delta_\varepsilon$, the optimal state $x^*(t_k + \delta_\varepsilon)$ enters Φ from $x(t_k) \in \Sigma_V \setminus \Phi$, i.e., $x^*(t_k + \delta_\varepsilon) \in \partial\Phi$, and it enters $\Omega(\varepsilon_f)$ at $t_k + \delta_{\varepsilon_f}$, i.e., $x^*(t_k + \delta_{\varepsilon_f}) \in \partial\Phi_f$, as shown in Fig. 4.4. Since $\Omega(\varepsilon_f) \subset \Phi$, it holds that $\delta_{\varepsilon_f} - \delta_\varepsilon > 0$.

To guarantee the existence of δ_{\min} , the following two cases are considered:

- (i) $x^*(t_k + \eta)$ is outside of Φ_f for all the time until (4.38) is violated. That is, $x^*(t_k + \eta) \notin \Phi_f$ for all $\eta \in [0, \bar{\eta}]$, where

$$\alpha_1(\|x^*(t_k + \bar{\eta})\|) = \frac{2K_u L_G L_J}{L_\phi \sigma} (e^{L_\phi \bar{\eta}} - 1). \quad (4.39)$$

- (ii) $x^*(t_k + \eta)$ enters Φ_f by the time (4.38) is violated. That is, there exists $\eta' \in [0, \bar{\eta}]$ where we obtain $x^*(t_k + \eta') \in \partial\Phi_f$.

Denote $\delta_{\min,1}$, $\delta_{\min,2}$ as minimum inter-execution times for the above cases (i), (ii), respectively. For the case (i), it holds that $\alpha_1(\|x^*(t_k + \eta)\|) \geq \alpha_1(\alpha_2^{-1}(\varepsilon_f)) > 0$, since we have $F(x, u) \geq \alpha_1(\|x\|)$ and $V_f(x) \leq \alpha_2(\|x\|)$ from Assumption 4.1. Thus the minimum inter-execution time $\delta_{\min,1}$ is given by the time interval when the R.H.S in (4.38) reaches $\alpha_1(\alpha_2^{-1}(\varepsilon_f))$, i.e.,

$$\delta_{\min,1} = \frac{1}{L_\phi} \ln \left(1 + \frac{\sigma L_\phi \alpha_1(\alpha_2^{-1}(\varepsilon_f))}{2K_u L_G L_J} \right) > 0. \quad (4.40)$$

For the case of (ii), the minimum inter-execution time is $\delta_{\min,2} = \delta_{\varepsilon_f} - \delta_\varepsilon$, since $x(t_k) \in \Sigma_V \setminus \Phi$ and it takes at least $\delta_{\varepsilon_f} - \delta_\varepsilon$ for the state to reach Φ_f . Thus, considering both cases, the over-all minimum inter-execution time δ_{\min} is positive and given by $\delta_{\min} = \min \{\delta_{\min,1}, \delta_{\min,2}\}$.

Based on this result, we finally obtain the following stability theorem.

Theorem 4.1. *Consider the networked control system in Fig. 1.1 where the plant follows the dynamics given by (4.1), and the proposed self-triggered strategy (Algorithm 2) is implemented. Then, if the initial state starts from $x(t_0) \in \Sigma_V \setminus \Phi$, then the state is guaranteed to enter Φ in finite time.*

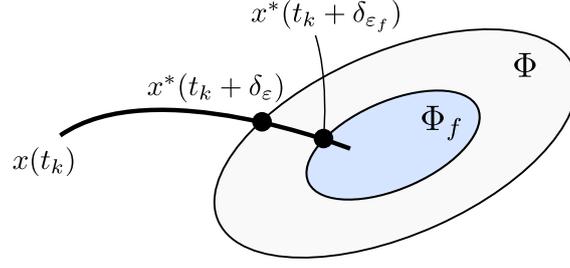


FIGURE 4.4: The illustration of Φ and the restricted terminal region Φ_f .

Proof. The statement is proved by contradiction. Starting from $x(t_0) \in \Sigma_V \setminus \Phi$, assume that the state is outside of Φ for all the time, i.e., $x(t) \in \Sigma_V \setminus \Phi$, for all $t \in [t_0, \infty)$.

Since there exists $\delta_{\min} > 0$, we obtain

$$\begin{aligned}
 & J^*(x(t_k)) - J^*(x(t_{k-1})) \\
 & < (\sigma - 1) \int_{t_{k-1}}^{t_k} F(x^*(\xi), u^*(\xi)) d\xi \\
 & < (\sigma - 1) \int_{t_{k-1}}^{t_{k-1} + \delta_{\min}} \alpha_1(\alpha_2^{-1}(\varepsilon_f)) d\xi \\
 & = -(1 - \sigma) \alpha_1(\alpha_2^{-1}(\varepsilon_f)) \delta_{\min} \\
 & = -\bar{\delta}_J < 0,
 \end{aligned}$$

where $\bar{\delta}_J = (1 - \sigma) \alpha_1(\alpha_2^{-1}(\varepsilon_f)) \delta_{\min}$. Thus, we obtain

$$\begin{aligned}
 J^*(x(t_k)) - J^*(x(t_{k-1})) & < -\bar{\delta}_J \\
 J^*(x(t_{k-1})) - J^*(x(t_{k-2})) & < -\bar{\delta}_J \\
 J^*(x(t_{k-2})) - J^*(x(t_{k-3})) & < -\bar{\delta}_J \\
 & \vdots \\
 J^*(x(t_1)) - J^*(x(t_0)) & < -\bar{\delta}_J.
 \end{aligned} \tag{4.41}$$

Summing over both sides of (4.41) yields

$$J^*(x(t_k)) < -k\bar{\delta}_J + J^*(x(t_0)) < -k\bar{\delta}_J + J_0, \tag{4.42}$$

where J_0 is defined in Definition 4.2. This implies $J^*(t_k) \rightarrow -\infty$ as $k \rightarrow \infty$, which contradicts the fact that $J^*(x(t_k)) \geq 0$. Therefore, there exists a finite time when the state enters Φ . \square

Note again that as soon as the state reaches Φ , the local control law $\kappa(x)$ is applied as a dual mode strategy. Therefore, our control objective to asymptotically stabilize the system to the origin is achieved, i.e., $x(t) \rightarrow 0$ as $t \rightarrow \infty$.

4.5 Simulation results

In this section the proposed self-triggered scheme is illustrated by considering both linear and nonlinear systems. As with the previous chapter, simulations were conducted on Matlab 2016a under Windows 10, Intel(R) Core(TM) 2.40 GHz, 8 GB RAM. As a software package, the author used Imperial College London Optimal Control Software (ICLOCS) [67], in order to solve (non)linear optimal control problems in the continuous-time domain.

(*Example 4.1*): Let us consider the following linearized system of an inverted pendulum on a cart:

$$\dot{x}(t) = \phi(x(t), u(t)) = Ax(t) + Bu(t),$$

where $x = [x_1, x_2, x_3, x_4]^T \in \mathbb{R}^4$, $u \in \mathbb{R}$ and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -mg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/\ell & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/M\ell \end{bmatrix}. \quad (4.43)$$

As with Example 3.2 and Example 3.4, x_1, x_2 represent the position of the cart and its velocity, and x_3, x_4 represents the angle of the pendulum and its velocity. Set $m = 1$ as the point mass, $M = 5$ as the mass of the cart, $\ell = 3$ as the length

of the massless rod, and $g = 9.8$ as the gravity. The constraint for the control input is assumed to be given by $\|u\| \leq 10$. The computed Lipschitz constants L_f and L_G are given by $L_f = 5.28$, $L_G = 0.501$. The stage and the terminal cost are assumed to be quadratic and given by $F(x, u) = \|x\|_Q^2 + \|u\|_R^2$ where $Q = I_4$ and $R = 0.1$.

The matrix for the terminal constraint P_f is computed as

$$P_f = \begin{bmatrix} 29.9 & 38.3 & 139 & 89.6 \\ 38.3 & 85.0 & 320 & 207 \\ 139 & 320 & 1600 & 959 \\ 89.6 & 207 & 959 & 592 \end{bmatrix} \quad (4.44)$$

and $\varepsilon = 0.43$. The parameters are set as $\varepsilon_f = 0.2$, $K_u = 1.0$, and $\sigma = 0.9$. The prediction horizon is $T_p = 10$ and the number of control sample is simply given by $N = 1$. The initial state is assumed to be $x_0 = [1, 0, 0, 0]$.

Figure 4.5 illustrates the resulting state trajectories of x_1, x_2 and Fig. 4.6 illustrates those of x_3, x_4 , by applying Algorithm 4.2 (blue solid lines) and the periodic MPC with a constant sampling time interval 0.1 (red dotted lines). Table 4.1 illustrates the resulting convergence time when the state enters the set around the origin ($\|x\| \leq 0.001$) and the total number of transmission instants during the time period $t \in [0, 30]$. Also, Fig. 4.7 illustrates the applied control input trajectory by applying the proposed scheme (blue solid line) and the periodic MPC scheme with 0.1 sampling time interval. From the figure and the table, all state trajectories are asymptotically stabilized to the origin by applying Algorithm 4.2, with providing a similar convergence to the periodic case and reducing communication load as shown in Table 4.1. Also, from Fig. 4.7 the control trajectory fulfills the input constraint $|u(t)| \leq 10$ for all $t \in \mathbb{R}$.

In Chapter 3, the author proposed a multiple discretization approach as a different self-triggered scheme, which can be applicable to this example since linear systems are considered. To compare the approach with the one presented

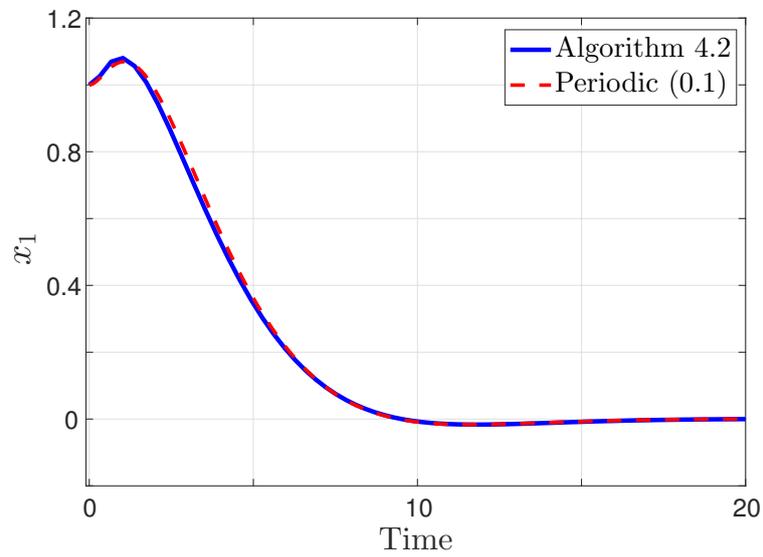
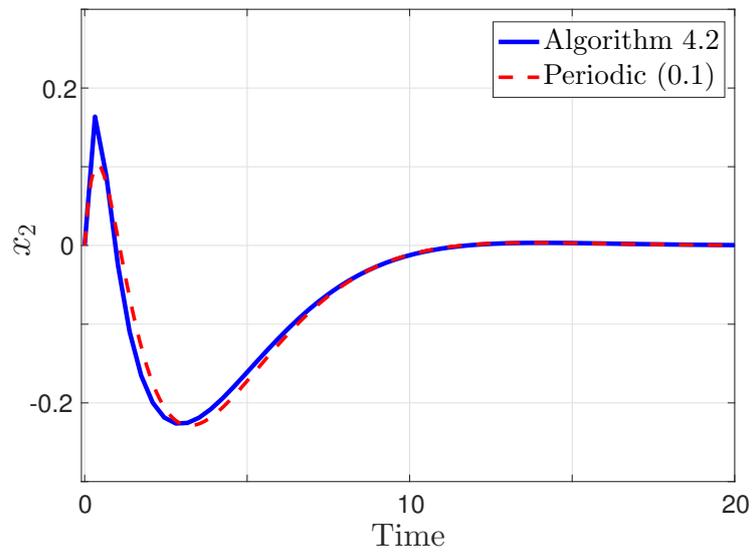
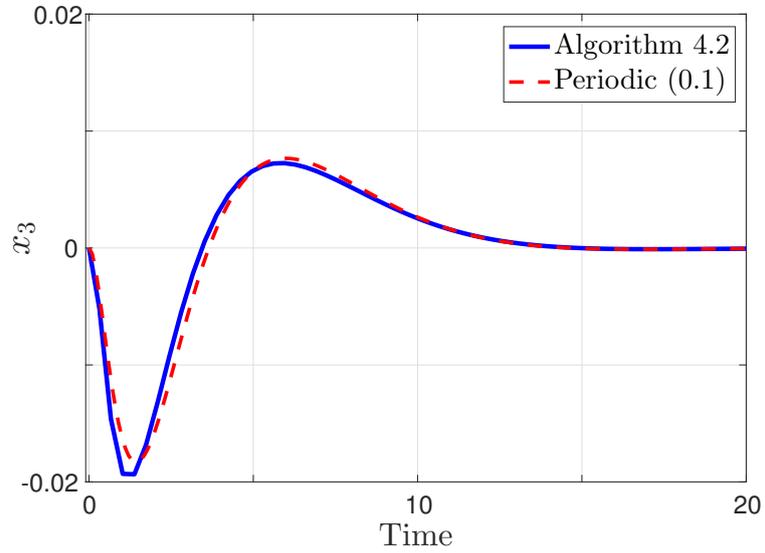
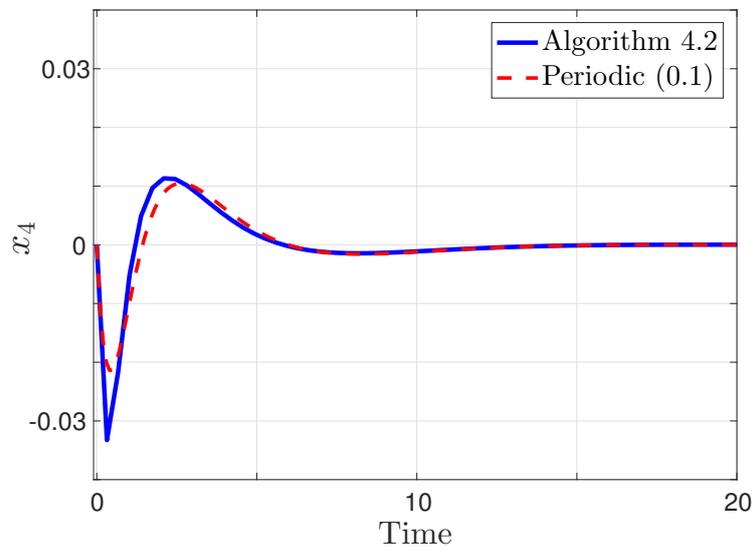
(a) State trajectories of x_1 .(b) State trajectory of x_2 .

FIGURE 4.5: State trajectories of x_1 and x_2 by implementing Algorithm 4.1 (blue solid lines) and the periodic MPC (red dotted lines).

(a) State trajectories of x_3 .(b) State trajectories of x_4 .FIGURE 4.6: State trajectories of x_3 and x_4 by implementing Algorithm 1 (blue solid lines) and the periodic MPC (red dotted lines).

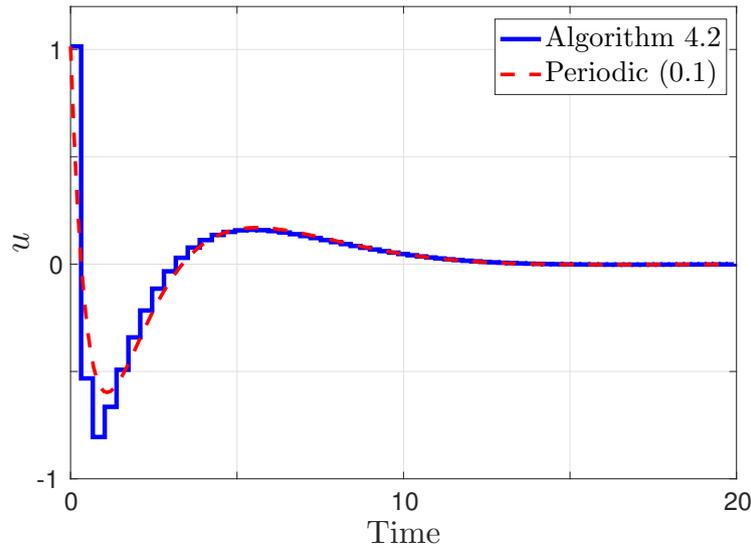


FIGURE 4.7: Applied control inputs by applying Algorithm 4.2 (blue solid line) and periodic scheme with sampling time interval 0.1 (red dotted line).

in this chapter, Table 4.1 illustrates the resulting convergence time and the number of transmission instants by applying Algorithm 3.1. When applying Algorithm 3.1, the tuning parameters are selected as $\beta = 1.0$, $\gamma = 0.5$. From the table, Algorithm 4.2 is shown to achieve a faster convergence than Algorithm 3.1. However, Algorithm 4.2 requires a more transmission instants by $74 - 15 = 19$ than Algorithm 3.1. This means that the self-triggered condition derived by Algorithm 4.2 is more conservative than Algorithm 3.1. Intuitively, this is due to the fact that a *sufficient condition* of Lyapunov stability (4.14) is derived in order to deal with nonlinear systems when deriving the self-triggered strategy. Since the optimal cost is *directly* evaluated for the multiple discretizations approach (see step (iii) in Algorithm 3.1), Algorithm 4.2 yields a more conservative result than Algorithm 3.1 for selecting transmission time intervals. In fact, we can calculate the maximum time interval for the periodic MPC scheme where the stabilization of the origin is guaranteed. Indeed, this can be simply done by starting with a very small value of the sampling time δ , and gradually increase this value until the state is destabilized. In this example, this is obtained

TABLE 4.1: Convergence time when the state trajectory enters the region ($\|x\| \leq 0.001$) and the number of transmission instants

	Algorithm 4.2	Periodic (0.1)	Algorithm 3.1
Convergence time	16.23	16.24	20.1
Transmission instants	74	201	15

as 2.5. Clearly, this critical value is much larger than the average transmission time interval obtained by Algorithm 4.2 (i.e., $16.23/74 = 0.22$). While the conservativeness is a drawback of Algorithm 4.2, it is still advantageous over Algorithm 3.1 as it can be applied for *nonlinear* systems, as illustrated in the next example.

To conclude, it is shown in this example that:

- Communication reduction is achieved by applying Algorithm 4.2 compared with the periodic case with 0.1 sampling time interval, while at the same time achieving a similar control performance.
- Due to the conservativeness of Algorithm 4.2, Algorithm 3.1 yields a more communication reduction than Algorithm 4.2.

(*Example 4.2*): The proposed approach can be applied to the nonlinear input-affine systems. To illustrate this, consider a control problem of non-holonomic vehicle in two dimensions, whose dynamics are borrowed from [33]:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (4.45)$$

where the state is denoted as $\chi = [x, y, \theta] \in \mathbb{R}^3$, consisting of the position of the vehicle $[x, y]$, and its orientation θ (see Fig. 4.8). $u = [v, \omega] \in \mathbb{R}^2$ is the control input and the constraints are assumed to be given by $\|v\| \leq \bar{v} = 2.0$ and $\|\omega\| \leq \bar{\omega} = 1.0$. This problem may be applicable to practical implementations,

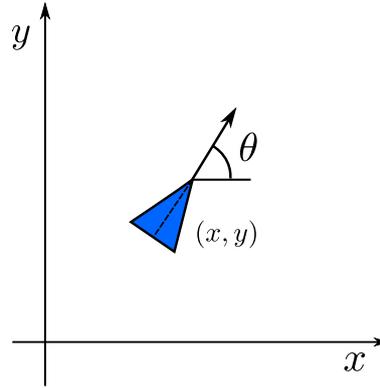


FIGURE 4.8: State variables for a vehicle regulation problem in two dimensions.

especially when the robot aims at surveying some regions interest such as hazardous areas which humans are not allowed to enter. In such dangerous areas, sensor nodes equipped in the plant side and relay nodes that deliver the sensor information are typically battery driven with limited life time of battery capacity, which thus motivates us to utilize our proposed framework. The computed Lipschitz constant L_ϕ and a positive constant L_G are given by $L_\phi = \sqrt{2}\bar{v}$ and $L_G = 1.0$. The stage and the terminal cost are given by $F = \chi^T Q \chi + u^T R u$, and $V_f = \chi^T \chi$ where $Q = I_3$ and $R = I_2$. The prediction horizon is $T_p = 6$. Since the linearized system around the origin is uncontrollable, the procedure presented in [68] is adopted to obtain a local controller satisfying Assumption 4.2, and the parameter for characterizing the terminal set is $\varepsilon = 0.8$. Set $\varepsilon_f = 0.4$ and the local controller is admissible if $K_u = 1.5$.

Figure 4.9 shows the trajectory of the vehicle under Algorithm 4.2 with $\sigma = 0.99$ and $N = 1$, starting from the initial point $[-10, -5, \pi/2]$ and its goal is the origin. The red dotted line represents the state trajectory by applying Algorithm 4.2, and the blue triangles show the position of the vehicle, where the triangle appears when control samples are transmitted to solve the OCP. The heading of the triangle shows the moving direction of the vehicle. From the figure, it is shown that the trajectory of the vehicle is asymptotically stabilized towards the origin by applying Algorithm 4.2. Figure 4.10 shows the

corresponding control inputs v, ω . From the figure, it is shown that the control inputs satisfy the constraints $\|v(t)\| \leq \bar{v} = 2.0$, $\|\omega(t)\| \leq \bar{\omega} = 1.0$, and these are updated only when they are needed by applying Algorithm 4.2. In the figure, the control inputs are given constant for the time interval $t \in [0, 2.0]$. However, the transmission has actually occurred at the time $t = 1.1$, which implies that the *useless* transmission is given even in the aperiodic control strategy. This may be due to the fact that the proposed self-triggered strategy is conservative since it has been derived based on a sufficient condition to the Lyapunov stability; namely, longer transmission time interval than the one obtained by the proposed strategy must be allowed to guarantee stabilization of the origin. The time when the state enters around the origin ($\|\chi\| \leq 0.001$) is 9.7, and the number of transmission instants until the state converges the region is given by 11 (i.e., the average transmission time interval is 0.88).

To make comparisons, Fig. 4.11 illustrates the resulting state trajectory by applying the periodic MPC scheme with 0.88 sampling time interval (i.e., $t_k = 0.88k, \forall k \in \mathbb{N}$), which is equal to the average transmission time interval by applying Algorithm 4.2. From the figure, it is shown that the state trajectory does *not* converge to the origin but is wobbling around the origin. This is due to the fact that the transmission time interval is not suitably selected to guarantee stability when the periodic scheme is employed. Figure 4.12 illustrates the resulting state trajectory by applying the periodic MPC scheme with 0.1 sampling time interval (i.e., $t_k = 0.1k, \forall k \in \mathbb{N}$), which is much smaller than the average transmission time interval by Algorithm 4.2. Table 4.2 illustrates the convergence time when the state enters the local set around the origin ($\|\chi\| \leq 0.001$), as well as the number of transmission instants until the state enters the set. From the table, Algorithm 4.2 achieves the number of transmission instants smaller by $72 - 11 = 61$ than the periodic case. On the other hand, Algorithm 4.2 requires $8.9 - 7.2 = 1.7$ longer convergence time than the periodic one, which indicates that the periodic scheme achieves better control performance. Therefore, it is shown in this example that there exists a tradeoff between achieving

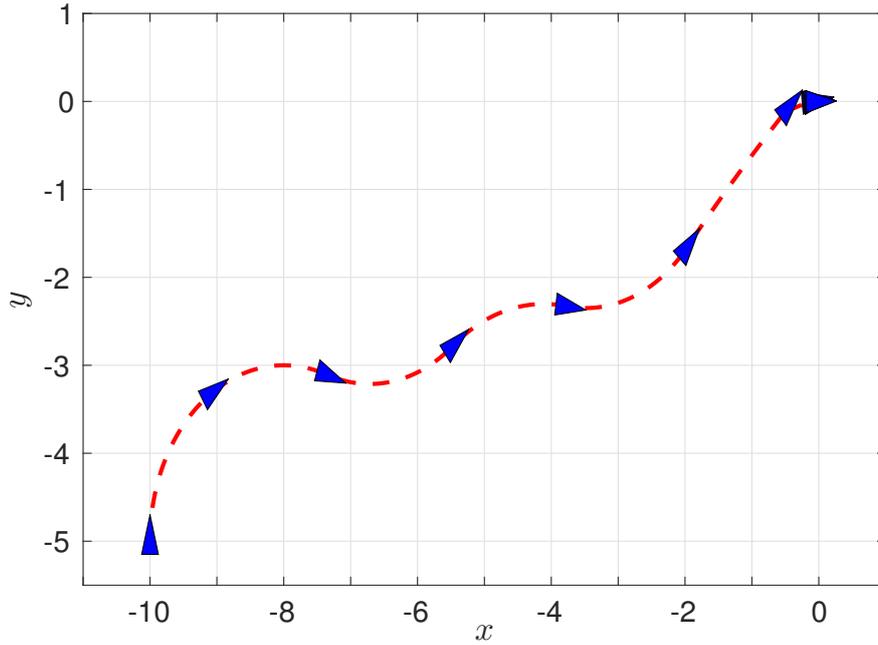


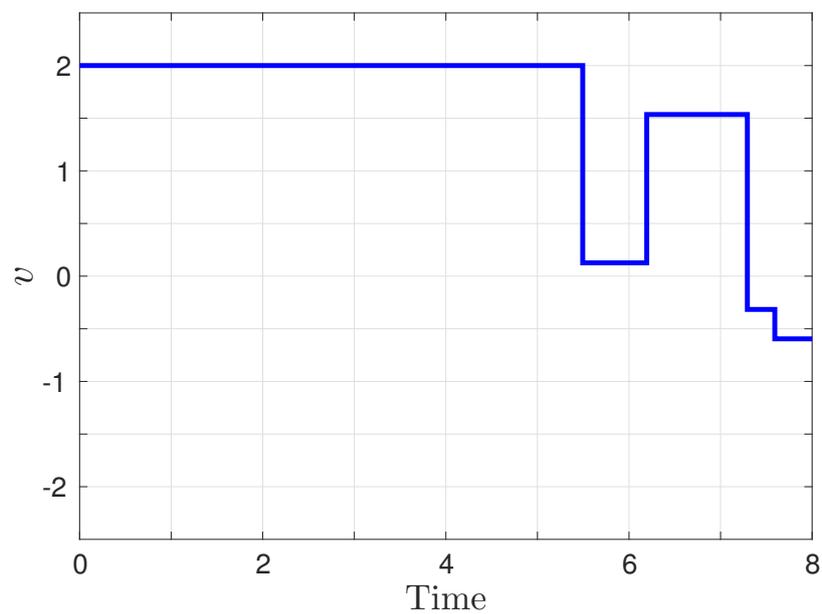
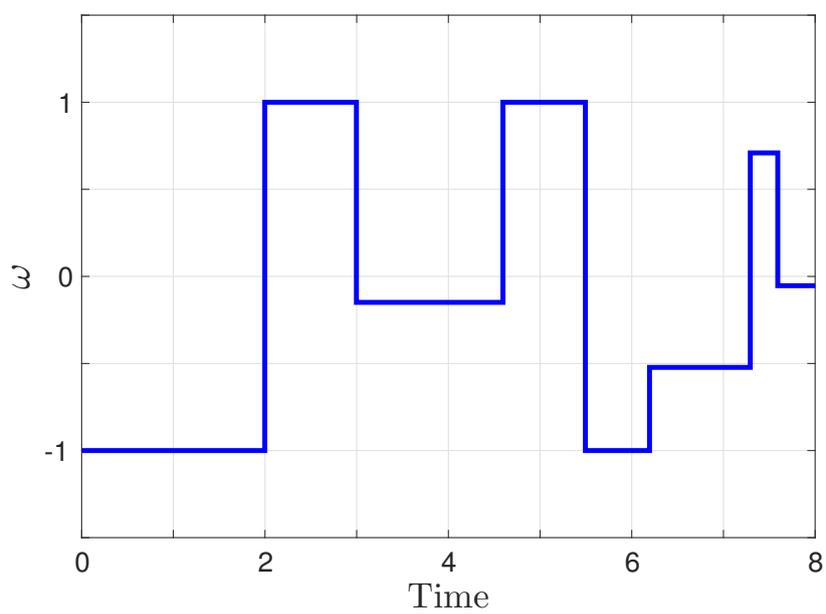
FIGURE 4.9: Trajectory of the vehicle by applying Algorithm 4.2.

TABLE 4.2: Convergence time when the state trajectory enters around the origin and the number of transmission instants.

	Algorithm 4.2	Periodic (0.1)
Convergence time	8.9	7.2
Transmission instants	11	72

the communication reduction and the control performance.

So far, we have considered the case when $N = 1$, and the number of transmission instants are expected to be smaller as N is chosen larger. To analyse how the number of transmission instants is affected by the selection of N , Algorithm 4.2 is again simulated under different selection of N . Table 4.3 illustrates the resulting convergence time when the enters the local set around the origin ($\|\chi\| \leq 0.001$) and the number of transmission instants during the time interval $t \in [0, 10]$ under different selections of N ($N = 1, 5$). From the table, all trajectories are asymptotically stabilized to the origin, with providing a similar convergence for both cases. Moreover, it is shown that selecting $N = 5$ yields $12 - 7 = 5$ smaller number of transmission instants than the case $N = 1$. Therefore, selecting $N = 5$ yields not only less communication load but also similar

(a) Control trajectory of v .(b) Control trajectory of ω .FIGURE 4.10: Control trajectory of v and ω implementing Algorithm 4.2.

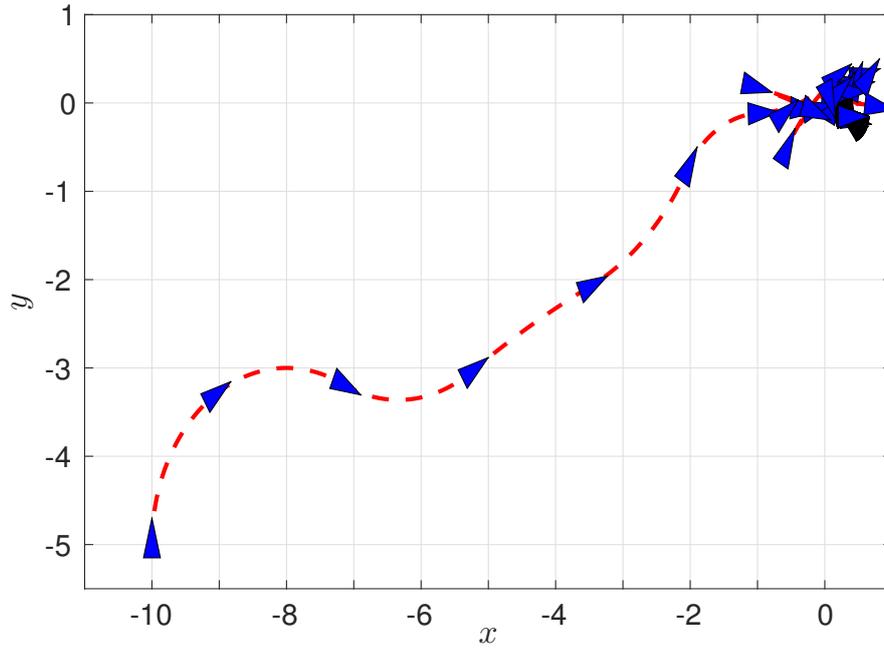


FIGURE 4.11: Trajectory of the vehicle by applying periodic MPC scheme with 0.88 sampling time interval.

TABLE 4.3: Convergence time when the trajectory of the vehicle enters around the origin and the number of transmission instants.

	Algorithm 4.2 ($N = 1$)	Algorithm 4.2 ($N = 5$)
Convergence time	9.7	9.8
Transmission instants	12	7

convergence to the case when $N = 1$.

To provide a more concrete analysis on the selection of N , Algorithm 4.2 is again simulated under different selections of N , which ranges from 1 to 100. Figure 4.13 plots the resulting number of transmission instants during the time period $t \in [0, 100]$ as a bar graph. From Fig. 4.13, the number of transmission instants tends to be smaller as N is selected larger, which means that the communication reduction is indeed achieved by increasing the number of control samples. Note that while a more communication reduction is achieved by selecting larger N , the communication bandwidth must be large enough such that N control samples can be transmitted once for each communication time.

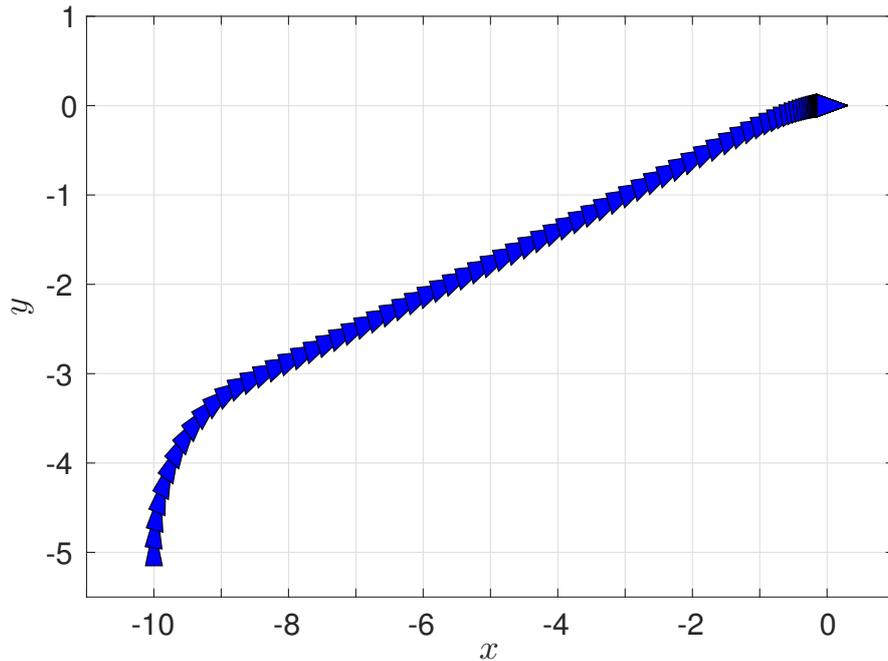


FIGURE 4.12: Trajectory of the vehicle by applying periodic MPC scheme with 0.1 sampling time interval.

To conclude, it is shown in this example that:

- Algorithm 4.2 achieves communication reduction while at the same time ensuring stability. On the other hand, control performance may be degraded at the expense of achieving the communication reduction by comparing with the periodic case with 0.1 sampling time interval.
- A more communication reduction is achieved by increasing the number of control samples.

4.6 Summary

In this chapter, the author proposes an aperiodic formulation of MPC for non-linear input-affine dynamical systems. In the proposed scheme, the controller not only solves an optimal control problem but also determine the next communication time by evaluating a self-triggered condition, which is derived based on evaluating the optimal cost as a Lyapunov function candidate. Moreover,

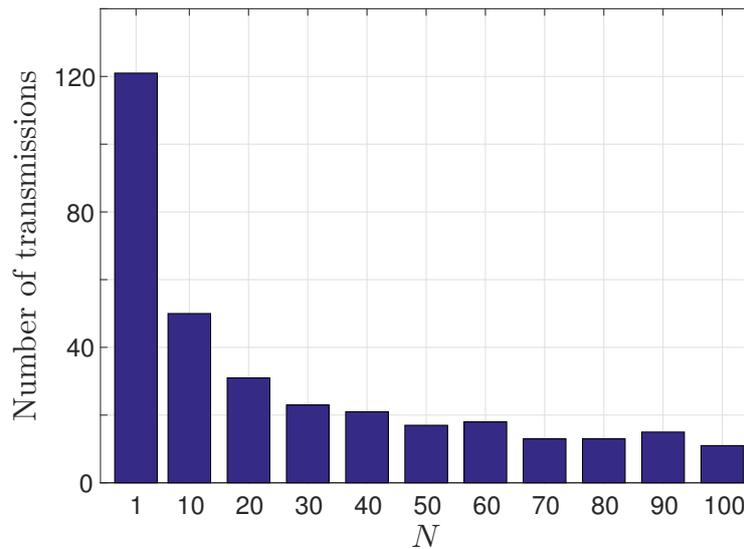


FIGURE 4.13: Number of transmission instants against the number of control samples N .

an efficient way to select the control samples is given to achieve the next communication time as long as possible. Stability under sample-and-hold implementation is shown by guaranteeing a positive minimum inter-execution time of the self-triggered strategy and showing that the optimal cost as a Lyapunov function candidate is decreasing. The proposed scheme is illustrated through several numerical examples. A control problem of an inverted pendulum on a cart is first considered and it is shown that the communication reduction is achieved by implementing the proposed scheme. While the proposed approach is applicable to nonlinear systems, it yields a more conservative result than the multiple discretization approach presented in Chapter 3, since it derives a sufficient condition of Lyapunov stability. In the second example, a control problem of vehicle regulation is considered, and the effectiveness of the proposed approach for nonlinear control systems is given. In the example, it is shown that the proposed scheme successfully achieves the communication reduction than the periodic case, while, on the other hand, degrading the control performance.

Chapter 5

Aperiodic MPC for General Nonlinear systems

The main contribution of this chapter is to propose an aperiodic MPC scheme for a more general class of systems than the ones presented in the previous chapters; namely, the author will propose an aperiodic MPC for general nonlinear systems, which are not necessarily to be input affine systems, and, moreover, are *perturbed* by additive bounded disturbances. In particular, the author will derive a threshold between the predictive states and the actual state, such that feasibility of the optimal control problem and stability are both guaranteed. The derived condition provides a key criteria to propose both event-triggered and self-triggered conditions, so that the optimal control problem is solved only when it is needed. In contrast to the triggering strategies provided in previous chapters, the optimal cost will *not* be evaluated to derive the triggering condition. Instead, the *time interval* when the optimal state trajectory enters the local set around the origin will be evaluated. An interesting feature of this scheme is that a less conservative result is obtained than the aperiodic MPC strategies for nonlinear systems presented in Chapter 4. As will be described later, this is because the proposed scheme does not include parameters (e.g., Lipschitz constant parameters for stage cost) as a potential source of conservativeness. Moreover, in the standard event-triggered strategy, it is required that the plant must monitor the state continuously, which may not only require

a dedicated analog hardware but also arise a high sensing cost. In order to alleviate such continuous requirement, the author proposes an event-triggered strategy, which evaluates an event-triggered condition at certain sampling time instants, instead of continuously. The self-triggered strategy is also given as a sufficient condition of the event-triggered strategy. Finally, some simulation examples are illustrated to validate the proposed schemes.

5.1 Problem formulation

In this section the problem formulation is defined. Consider applying MPC to the following nonlinear systems with additive disturbances:

$$\dot{x}(t) = \phi(x(t), u(t)) + w(t), \quad (5.1)$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $w(t) \in \mathbb{R}^n$ is an additive bounded disturbance. Note that in contrast to the system description (4.1) in Chapter 4, the dynamics are not necessary to be input-affine systems. The control input u and the disturbance w are assumed to satisfy the following constraints:

$$u(t) \in \mathcal{U} \subseteq \mathbb{R}^m, \quad w(t) \in \mathcal{W} \subseteq \mathbb{R}^n, \quad \forall t \geq \mathbb{R}. \quad (5.2)$$

Similarly to Chapter 4, it is assumed that the following is satisfied:

Assumption 5.1. *The nonlinear function $\phi(x, u) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is twice continuously differentiable, and the origin is an equilibrium point, i.e., $\phi(0, 0) = 0$. The constraint sets \mathcal{U} and \mathcal{W} are compact, convex and $0 \in \mathcal{U}$.*

Assumption 5.2. *For the linearized system around the origin with no disturbances;*

$$\dot{x}(t) = A_f x + B_f u, \quad (5.3)$$

where $A_f = \partial\phi/\partial x(0, 0)$ and $B_f = \partial\phi/\partial u(0, 0)$, the pair (A_f, B_f) is stabilizable.

In the following, let $t_k, k \in \mathbb{N}_{\geq 0}$ be the transmission instants when the plant transmits the state information to the controller, and let $\Delta_k = t_{k+1} - t_k$ be the transmission time intervals. Namely, at $t_k, k \in \mathbb{N}$, the controller solves an OCP based on the state measurement $x(t_k)$ and the predictive behavior of the systems described by (5.1). In this paper, the following cost to be minimized is given:

$$J(x(t_k), u(\cdot)) = \int_{t_k}^{t_k+T_k} \|\hat{x}(\xi)\|_Q^2 + \|u(\xi)\|_R^2 d\xi, \quad (5.4)$$

where $Q = Q^\top \succ 0, R = R^\top \succ 0$ and $T_k > 0$ is the prediction horizon. $\hat{x}(\xi)$ denotes the nominal trajectory of (5.1) given by

$$\dot{\hat{x}}(\xi) = \phi(\hat{x}(\xi), u(\xi)) \quad (5.5)$$

for all $\xi \in [t_k, t_k + T_k]$ with $\hat{x}(t_k) = x(t_k)$. Although the prediction horizon is given constant for any update times in the standard formulation of MPC (see Chapter 4), the author considers here that T_k is *adaptively* selected based on the previous results of OCPs. This variable horizon strategy will be a key idea to prove stability for perturbed nonlinear systems. More characterization of T_k is provided in this section when formulating the OCP.

The following property holds regarding the existence of a local controller:

Lemma 5.1. *Suppose that Assumption 5.1 holds. Then, there exists a positive constant $0 < \varepsilon < \infty$, a matrix $P_f = P_f^\top \succ 0$, and a local controller $\kappa(x) = Kx \in \mathcal{U}$, satisfying*

$$\frac{\partial V_f}{\partial x} \phi(x, \kappa(x)) \leq -\frac{1}{2}x^\top (Q + K^\top RK)x \quad (5.6)$$

for all $x \in \Phi$, where $V_f(x) = x^\top P_f x$ and

$$\Phi = \{x \in \mathbb{R}^n : V_f(x) \leq \varepsilon^2\}.$$

Furthermore, Φ is a positive invariant set for the system (5.1) with $\kappa(x) = Kx \in \mathcal{U}$, if the disturbance w satisfies $\|w\|_{P_f} \leq \hat{w}_{\max}$, where

$$\hat{w}_{\max} = \frac{\varepsilon}{4\lambda_{\max}(\hat{Q}_P)}, \quad (5.7)$$

with $\hat{Q}_P = P_f^{-1/2}(Q + K^\top RK)P_f^{-1/2}$.

Proof. Consider a linearization of (5.1) around the origin for the non-disturbance case; $\dot{x}(t) = A_f x(t) + B_f u(t)$, where $A_f = \partial\phi/\partial x(0,0)$ and $B_f = \partial\phi/\partial u(0,0)$. Since the linearized system is stabilizable from Assumption 5.1, we can find a state feedback controller $\kappa(x) = Kx$ such that $A_c = A_f + B_f K$ is Hurwitz and the closed loop system $\dot{x} = A_c x$ is thus asymptotically stable. Choose a matrix P such that the following Lyapunov equation holds: $PA_c + A_c^\top P = -(Q + K^\top RK)$ where Q and R are matrices for the stage cost defined in (5.4). Then, the time derivative of the function $V_f = x^\top P x$ along a trajectory of the nominal system $\dot{x} = \phi(x, \kappa(x))$ yields:

$$\begin{aligned} \dot{V}_f(x) &= -x^\top(Q + K^\top RK)x + 2x^\top P\phi(x) \\ &\leq -x^\top(Q + K^\top RK)x \left(1 - \frac{2\|\phi(x)\|_P}{\lambda_{\min}(\hat{Q}_P)\|x\|_P}\right), \end{aligned}$$

where $\psi(x) = \phi(x, \kappa(x)) - A_c x$, and $\hat{Q}_P = P^{-1/2}(Q + K^\top RK)P^{-1/2}$. Since $\|\psi(x)\|_P/\|x\|_P \rightarrow 0$ as $\|x\|_P \rightarrow 0$, there exists a positive constant $0 < \varepsilon_0 < \infty$ such that $\|\psi(x)\|_P/\|x\|_P \leq \lambda_{\min}(\hat{Q}_P)/4$ for $\|x\|_P \leq \varepsilon_0$. Let $0 < \varepsilon \leq \varepsilon_0$ such that for all $\|x\|_P \leq \varepsilon$, $\kappa(x) = Kx \in \mathcal{U}$. By letting $\Phi = \{x \in \mathbb{R}^n \mid V_f(x) \leq \varepsilon^2\}$, we obtain $\dot{V}_f(x) \leq -0.5x^\top(Q + K^\top RK)x$ for all $x \in \Phi$.

Now, consider the time derivative of the function V_f along a trajectory of the nonlinear system with additive disturbances $\dot{x} = f(x, \kappa(x)) + w$:

$$\begin{aligned} \dot{V}_f(x) &= -x^\top(Q + K^\top RK)x + 2x^\top P\psi(x) + 2x^\top Pw \\ &\leq -x^\top(Q + K^\top RK)x \left(1 - \frac{2\|\phi(x)\|_P}{\lambda_{\min}(\hat{Q}_P)\|x\|_P} - \frac{2\|w\|_P}{\lambda_{\min}(\hat{Q}_P)\|x\|_P} \right), \end{aligned}$$

and consider also a compact set as a boundary of Φ ; $\partial\Phi = \{x \in \mathbb{R}^n \mid V_f(x) = \varepsilon^2\}$. From above, we obtain $\dot{V}_f \leq 0$ for $x \in \partial\Phi$, if $\|w\|_P \leq \varepsilon\lambda_{\min}(\hat{Q}_P)/4$. Thus, Φ is a positive invariant set for the closed loop system $\dot{x} = \phi(x, \kappa(x)) + w$ if the disturbance satisfies $\|w\|_P \leq \varepsilon\lambda_{\min}(\hat{Q}_P)/4$. This completes the proof. \square

Definition 5.1 (Control Objective). *The control objective of MPC is to steer the state x to the local region Φ in finite time.*

Similarly to Chapter 4, applying *dual-mode MPC* is considered, in which the local controller κ is applied as soon as the state enters Φ . Note that since the plant is controlled over a network, applying the local controller $\kappa(x)$ may require a *continuous* control update and may not be suitable under limited communication capabilities. One way to avoid this issue is to apply the local controller in a *sample-and-hold fashion*, i.e., $u(t) = \kappa(x(t_k))$, $t \in [t_k, t_k + \delta]$. Here, $0 < \delta < \infty$ can be chosen small enough such that asymptotic stability is still guaranteed, see [63] for a detailed analysis. Based on the local set Φ , further define the restricted set Φ_f given by

$$\Phi_f = \{x \in \mathbb{R}^n : V_f(x) \leq \varepsilon_f^2\},$$

where $0 < \varepsilon_f < \varepsilon$. Since $\varepsilon_f < \varepsilon$, the set Φ_f is contained in Φ , i.e., $\Phi_f \subset \Phi$. An example of these regions is illustrated in Fig. 5.1.

Assumption 5.3. *The nonlinear function $\phi(x, u) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is Lipschitz continuous with the weighted matrix P_f , with the Lipschitz constant $0 \leq L_\phi < \infty$.*

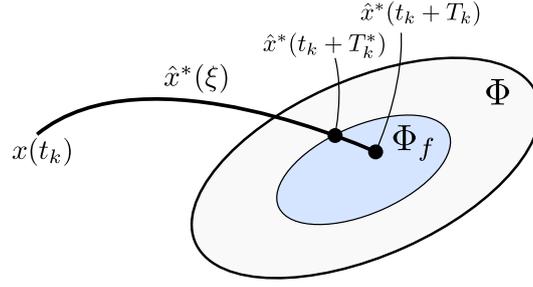


FIGURE 5.1: Graphical representation of the two regions Φ , Φ_f , and the optimal state trajectory \hat{x}^* (blue solid line). T_k^* denotes the time interval to reach Φ_f .

Namely, there exists $0 \leq L_\phi < \infty$ such that

$$\|\phi(x_1, u) - \phi(x_2, u)\|_{P_f} \leq L_\phi \|x_1 - x_2\|_{P_f} \quad (5.8)$$

for all $x_1, x_2 \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$.

Assumption 5.3 will be used to derive several conditions to guarantee feasibility of the OCP. In the formulation of MPC, the controller finds at each update time t_k , $k \in \mathbb{N}_{\geq 0}$, an optimal state and a control trajectory $\hat{x}^*(\xi)$, $u^*(\xi)$ for all $\xi \in [t_k, t_k + T_k]$, by minimizing the cost given by (5.4). Regarding constraints, it is imposed that the optimal state reaches Φ_f within the prediction horizon T_k , i.e., $\hat{x}^*(t_k + T_k) \in \Phi_f$. Since $\hat{x}^*(t_k + T_k) \in \Phi_f$, there exists a positive time interval when the optimal state enters the boundary of Φ_f . Let T_k^* ($T_k^* \leq T_k$) be such time interval obtained at t_k , i.e., $\hat{x}^*(t_k + T_k^*) \in \partial\Phi_f$. The time interval T_k^* is illustrated also in Fig. 5.1.

Based on the above notations, the following OCP is proposed:

Problem 5.1 (Optimal Control Problem). For the non-initial time t_k , $k \in \mathbb{N}_{\geq 1}$, given $x(t_k)$ and T_{k-1}^* , find the optimal control input and the corresponding state trajectory $u^*(\xi)$, $\hat{x}(\xi)$, $\forall \xi \in [t_k, t_k + T_k]$, by minimizing $J(x(t_k), u(\cdot))$, subject to the

following constraints:

$$\begin{cases} \dot{\hat{x}}(\xi) = \phi(\hat{x}(\xi), u(\xi)), \xi \in [t_k, t_k + T_k] & (5.9) \\ u(\xi) \in \mathcal{U} & (5.10) \\ \hat{x}(t_k + T_k) \in \Phi_f, & (5.11) \end{cases}$$

where $T_k = T_{k-1}^* - \alpha \Delta_{k-1}$ for a given $0 < \alpha < 1$ and $\Delta_{k-1} = t_k - t_{k-1}$. For the initial time t_0 , minimize the cost $J(x(t_k), u(\cdot))$ given by (5.4), subject to (5.9), (5.10) and $\hat{x}(t_0 + T_0) \in \Phi_f$ for a given $T_0 > 0$. \square

For the initial time t_0 , Problem 5.1 is solved with a given $T_0 > 0$. In order to guarantee feasibility at t_0 , T_0 needs to be suitably chosen such that the terminal constraint $\hat{x}(t_0 + T_0) \in \Phi_f$ is fulfilled. More specifically, T_0 should be selected to satisfy $x(t_0) \in \mathcal{X}(T_0)$, where $\mathcal{X}(T_0) = \{x(t_0) \in \mathbb{R}^n \mid \exists u(t) \in \mathcal{U}, t \in [t_0, t_0 + T_0] : \hat{x}(t_0 + T_0) \in \Phi_f\}$, i.e., $\mathcal{X}(T_0)$ denotes the set of states that can reach Φ_f within the time $t_0 + T_0$. Although there may not exist a general framework to compute $\mathcal{X}(T_0)$ explicitly for nonlinear systems, several approximation methods have been proposed to compute $\mathcal{X}(T_0)$, see e.g., [51]. The initial feasibility is essentially required for guaranteeing recursive feasibility, which is analyzed in the next section.

For the *non*-initial time $t_k, k \in \mathbb{N}_{\geq 1}$, it is required by (5.11) that the optimal state enters Φ_f within $T_k = T_{k-1}^* - \alpha \Delta_{k-1}$, where T_{k-1}^* is the time interval obtained by the previous calculation of OCP. This implies that T_k^* satisfies $T_k^* \leq T_k = T_{k-1}^* - \alpha \Delta_{k-1} < T_{k-1}^* \leq T_{k-1}$, which guarantees that the time interval T_k^* and the prediction horizon T_k become strictly smaller than the previous one at t_{k-1} . In later sections, the author will make use of this property to show that the state enters Φ in finite time.

Remark 5.1 (Terminal and without terminal constraint). *Although various analysis and control strategies have been proposed for MPC, approaches to guarantee stability can be mainly divided into two categories; the OCP with a terminal constraint (see e.g., [38]), and the OCP without a terminal constraint (see e.g., [39], [40]). While*

the OCP becomes in general harder to be solved when the terminal constraint is imposed, this paper follows the former approach to guarantee stability and to derive an event-triggered strategy. Note that our problem formulation slightly differs from the standard formulation [38], since the prediction horizon is not constant but is adaptively selected for each calculation time of the OCP. \square

When applying MPC, it is considered that the optimal control input trajectory $u^*(\xi)$ is applied until the next update time t_{k+1} , where t_{k+1} is determined by the proposed event-triggered (self-triggered) strategy. Namely, the controller transmits the optimal control trajectory $u^*(\xi)$, for all $\xi \in [t_k, t_k + T_k^*]$, and the plant applies it for all $\xi \in [t_k, t_{k+1}]$ according the event-triggered strategy. The closed-loop system for $t \in [t_k, t_{k+1})$ is thus given by

$$\dot{x}(t) = \phi(x(t), u^*(t)) + w(t), \quad t \in [t_k, t_{k+1}). \quad (5.12)$$

5.2 Feasibility analysis

The main result of this section is to provide several conditions to guarantee recursive feasibility, which states that the existence of a feasible solution at an initial update time t_0 implies the feasibility at any update times afterwards t_k , $k \in \mathbb{N}_{\geq 1}$, if the difference between the predictive and the actual state does not exceed a certain threshold. The obtained feasibility conditions are key ingredients to derive the event-triggered strategy, which will be discussed in the next section.

Theorem 5.1. *Suppose that the OCP defined in Problem 5.1 has a solution at t_k , providing an optimal control input $u^*(\xi)$ and the corresponding state trajectory $\hat{x}^*(\xi)$ for all $\xi \in [t_k, t_k + T_k]$, and the time interval T_k^* . Then, Problem 5.1 has a solution at*

$t_{k+1} (> t_k)$, if the followings are satisfied:

$$\begin{cases} \|x(t_{k+1}) - \hat{x}^*(t_{k+1})\|_{P_f} \leq (\varepsilon - \varepsilon_f)e^{-L_\phi T_k^*} & (5.13) \\ \Delta_k = t_{k+1} - t_k \leq T_k^*, & (5.14) \\ \|w\|_{P_f} \leq \tilde{w}_{\max}, & (5.15) \end{cases}$$

where \tilde{w}_{\max} is given by

$$\tilde{w}_{\max} = \frac{\lambda_{\min}(\hat{Q}_P)}{4e^{L_\phi T_0^*}}(1 - \alpha)\varepsilon_f. \quad (5.16)$$

□

Proof. Consider the following feasible control trajectory candidate:

$$\bar{u}(\xi) = \begin{cases} u^*(\xi), & \xi \in [t_{k+1}, t_k + T_k^*] \\ \kappa(\bar{x}(\xi)), & \xi \in (t_k + T_k^*, t_{k+1} + T_{k+1}], \end{cases} \quad (5.17)$$

where $T_{k+1} = T_k^* - \alpha\Delta_k$. Here we have $t_{k+1} + T_{k+1} > t_k + T_k^*$ since

$$\begin{aligned} t_{k+1} + T_{k+1} &= t_k + \Delta_k + T_k^* - \alpha\Delta_k \\ &= t_k + (1 - \alpha)\Delta_k + T_k^* > t_k + T_k^*. \end{aligned}$$

$\bar{x}(\xi)$ denotes the predictive state trajectory obtained by applying $\bar{u}(\xi)$, i.e., $\dot{\bar{x}}(\xi) = \phi(\bar{x}(\xi), \bar{u}(\xi))$ with $\bar{x}(t_{k+1}) = x(t_{k+1})$.

To prove that (5.17) is a feasible controller, it is shown that the following three claims are satisfied:

- (i) By applying $\bar{u}(\xi)$, $\xi \in [t_{k+1}, t_k + T_k^*]$, the predictive state enters Φ by the time $t_k + T_k^*$. That is, $\bar{x}(t_k + T_k^*) \in \Phi$. This ensures that applying the local controller κ from $t_k + T_k^*$ is admissible.
- (ii) $T_{k+1} = T_k^* - \alpha\Delta_k > 0$. This ensures that the time interval to reach Φ_f in the constraint (5.11) is always positive at the update time t_{k+1} .

(iii) By applying $\bar{u}(\xi)$, $\xi \in (t_k + T_k^*, t_{k+1} + T_{k+1}]$, the predictive state \bar{x} enters Φ_f by the time $t_{k+1} + T_{k+1}$. That is,

$$\bar{x}(t_{k+1} + T_{k+1}) \in \Phi_f.$$

To prove the claim (i), first use the fact that the difference between \bar{x} and \hat{x}^* is upper bounded by

$$\|\bar{x}(\xi) - \hat{x}^*(\xi)\|_{P_f} \leq \|x(t_{k+1}) - \hat{x}^*(t_{k+1})\|_{P_f} e^{L_\phi(\xi - t_{k+1})}$$

for $\xi \in [t_{k+1}, t_k + T_k^*]$. Supposing that (5.13) holds and by letting $\xi = t_k + T_k^*$, we obtain

$$\begin{aligned} \|\bar{x}(t_k + T_k^*) - \hat{x}^*(t_k + T_k^*)\|_{P_f} &\leq e^{-L_\phi T_k^*} (\varepsilon - \varepsilon_f) e^{L_\phi(t_k + T_k^* - t_{k+1})} \\ &= (\varepsilon - \varepsilon_f) e^{-L_\phi(t_{k+1} - t_k)}. \end{aligned}$$

From the triangular inequality, we obtain

$$\begin{aligned} \|\bar{x}(t_k + T_k^*)\|_{P_f} &\leq \|\hat{x}^*(t_k + T_k^*)\|_{P_f} + (\varepsilon - \varepsilon_f) e^{-L_\phi(t_{k+1} - t_k)} \\ &\leq \varepsilon_f + \varepsilon - \varepsilon_f \\ &= \varepsilon. \end{aligned}$$

Thus it holds that $\bar{x}(t_k + T_k^*) \in \Phi$ and the proof of (i) is completed.

The proof of (ii) is obtained from the fact that we have $\Delta_k \leq T_k^*$ from the event-triggered strategy, and thus $T_k^* - \alpha \Delta_k \geq (1 - \alpha) T_k^* > 0$.

Let us now prove the statement given in (iii). By using $\bar{x}(t_k + T_k^*) \in \Phi$ and from (5.6), we obtain

$$\begin{aligned} \dot{V}_f(\bar{x}(\xi)) &\leq -\frac{1}{2} \bar{x}^\top(\xi) (Q + K^\top R K) \bar{x}(\xi) \\ &\leq -\frac{1}{2} \lambda_{\min}(\hat{Q}_P) V_f(\bar{x}(\xi)) \end{aligned} \tag{5.18}$$

for $\xi \in (t_k + T_k^*, t_{k+1} + T_k^* - \alpha\Delta_k]$. Furthermore, from the Gronwall-Bellman inequality and by supposing that (5.15) holds, we obtain

$$\begin{aligned} \|\bar{x}(t_k + T_k^*)\|_{P_f} &\leq \|\hat{x}^*(t_k + T_k^*)\|_{P_f} + \frac{\tilde{w}_{\max}}{L_f} e^{L_\phi T_k^*} (1 - e^{-L_\phi \Delta_k}) \\ &\leq \varepsilon_f + \frac{(1 - \alpha)}{4L_f} \varepsilon_f \lambda_{\min}(\hat{Q}_P) (1 - e^{-L_\phi \Delta_k}). \end{aligned}$$

Denoting $\eta = \frac{(1-\alpha)}{4L_\phi} \lambda_{\min}(\hat{Q}_P)$, and by using comparison lemma, we obtain

$$\begin{aligned} V_f(\bar{x}(t_{k+1} + T_k^* - \alpha\Delta_k)) &\leq V_f(\bar{x}(t_k + T_k^*)) e^{-0.5\lambda_{\min}(\hat{Q}_P)(1-\alpha)\Delta_k} \\ &\leq \varepsilon_f^2 (1 + \eta(1 - e^{-L_\phi \Delta_k}))^2 e^{-2L_\phi \eta \Delta_k} \\ &\leq \varepsilon_f^2. \end{aligned}$$

The 3rd inequality is obtained by the fact that the function $g_\varepsilon(\Delta_k) = (1 + \eta(1 - e^{-L_\phi \Delta_k})) e^{-L_\phi \eta \Delta_k}$ is shown to be a decreasing function of Δ_k with $g_\varepsilon(0) = 1$. Thus we obtain $V_f(\bar{x}(t_{k+1} + T_k^* - \alpha\Delta_k)) \leq \varepsilon_f^2$, and the proof of (iii) is completed.

Based on above, the controller given by (5.17) provides a feasible solution to Problem 5.1 for $t_{k+1} (> t_k)$, provided that the conditions (5.14), (5.13), and (5.15) are satisfied. This completes the proof. \square

5.3 Event-triggered strategy

Suppose again that the OCP is solved at t_k , providing a pair of optimal control $u^*(\xi)$ and the corresponding state $\hat{x}^*(\xi)$ for all $\xi \in [t_k, t_k + T_k]$. Through an event-triggered condition, the author considers to determine the next OCP update time $t_{k+1} (> t_k)$ (i.e., the next communication time) such that the feasibility is ensured.

The simplest way to determine t_{k+1} might be to use the original feasibility conditions directly as the event-triggered conditions. That is, for each $t > t_k$,

check the feasibility according to (5.14) and (5.13), i.e.,

$$\|x(t) - \hat{x}^*(t)\|_{P_f} \leq (\varepsilon - \varepsilon_f)e^{-L_\phi T_k^*}, \quad (5.19)$$

$$t - t_k \leq T_k^*. \quad (5.20)$$

Only when either of the above conditions is violated, then we set $t_{k+1} = t$ as the next update time. This strategy ensures the feasibility of the OCP and may reduce a computational load of solving OCPs. However, checking the above conditions for each $t > t_k$ requires *continuous* monitoring of the state $x(t)$ and evaluations of the above conditions, which clearly leads to a high cost of sensing and a computation load.

Therefore, the author proposes here an alternative approach by relaxing the above *continuous* requirements. The key idea of our approach is to measure the state and evaluate event-triggered conditions only at certain sampling time intervals, instead of continuously. The overview of the proposed approach is described as follows. Once the OCP is solved by MPC at an update time instant, say t_k , and T_k^* is obtained, the controller computes $\delta_k^* \in \mathbb{R}_{>0}$, which represents the sampling time interval at which the event-triggered condition is evaluated. Namely, from the obtained δ_k^* the plant measures the state and checks the event-triggered condition only at $t_k + m\delta_k^*$, $m \in \mathbb{N}_{\geq 1}$, in order to determine the next update time t_{k+1} .

Regarding the proposed framework outlined above, we need to derive both mechanisms to determine δ_k^* and the event-triggered conditions. One might directly utilize (5.19), (5.20) as the event-triggered conditions, and evaluate them with a given arbitrary value of δ_k^* . However, this cannot be applied due to the following two problems regarding the violation of feasibility:

(P.1) If a large value of δ_k^* would be chosen, the feasibility would not be satisfied at the next evaluation time $t_k + \delta_k^*$. This issue is illustrated in Fig. 5.2.

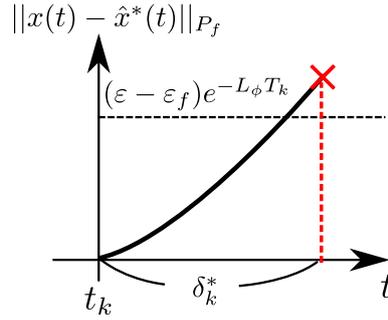


FIGURE 5.2: The illustration of the problem presented in (P.1). The figure shows the left hand side (black solid) and the right hand side (black dotted) of (5.19).

As shown in the figure, if δ_k^* would be selected too large (blue cross mark), then feasibility is not guaranteed at $t_k + \delta_k^*$.

(P.2) If we would directly use (5.19) as the event-triggered condition, the feasibility might be violated between two consecutive evaluation times. This issue is illustrated in Fig. 5.3. In the figure, blue marks represent the sequence of the left hand side in (5.19) measured at the sampling interval δ_k^* , and the red mark represents the exact time when the violation of (5.19) takes place. As shown in the figure, the feasibility is violated between two evaluation times; the event-triggered strategy fails to be obtained due to the loss of feasibility (represented as green mark). The critical problem here is that the controller does not know whether the feasibility is violated between two evaluation times; when arriving at a certain evaluation time (e.g., green cross mark in Fig. 5.3), it is possible that the error $\|x(t) - \hat{x}^*(t)\|$ already exceeds the threshold, and a loss of feasibility occurs.

In the following, the solutions to each problem above are given and provide the over-all event-triggered strategy. Consider first to solve (P.1). In order to deal with the problem, δ_k^* needs to be chosen small enough such that the feasibility is guaranteed for all $t \in [t_k, t_k + \delta_k^*]$. Thus a minimum inter-event time of the feasibility conditions given by (5.19), (5.20), is evaluated. Assume that the size of the disturbance satisfies $\|w(t)\|_{P_f} \leq \tilde{w}_{\max}, \forall t \geq t_0$, which ensures from

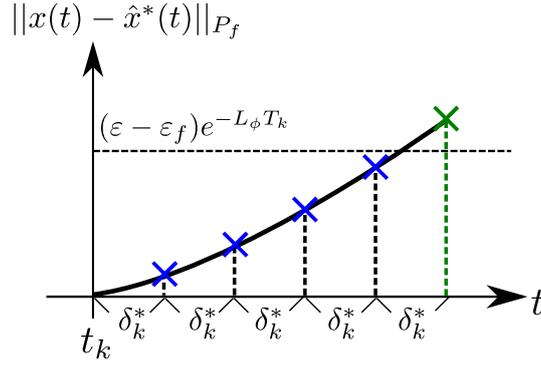


FIGURE 5.3: The figure illustrates the problem of violating the feasibility described in (P.2).

Theorem 5.1 that the effect of disturbances does not violate the feasibility. By using Gronwall-Bellman inequality, we obtain

$$\|x(t) - \hat{x}^*(t)\|_{P_f} \leq \frac{\tilde{w}_{\max}}{L_\phi} (e^{L_\phi(t-t_k)} - 1) \quad (5.21)$$

for $t \in [t_k, t_k + T_k]$. Thus, a sufficient condition to satisfy (5.19) is

$$\frac{\lambda_{\min}(\hat{Q}_P)(1 - \alpha)\varepsilon_f}{2L_\phi e^{L_\phi T_0^*}} (e^{L_\phi(t-t_k)} - 1) \leq (\varepsilon - \varepsilon_f)e^{-L_\phi T_k^*}.$$

Solving the above for t yields $t \leq t_k + \Delta_k^{\min}$, where Δ_k^{\min} is given by

$$\Delta_k^{\min} = \frac{1}{L_\phi} \ln \left(1 + \frac{2L_\phi(\varepsilon - \varepsilon_f)e^{L_\phi(T_0^* - T_k^*)}}{\lambda_{\min}(\hat{Q}_P)(1 - \alpha)\varepsilon_f} \right) > 0. \quad (5.22)$$

This implies that the condition (5.19) is satisfied for all $t \in [t_k, t_k + \Delta_k^{\min}]$. By taking into account the other feasibility condition (5.20), the over-all minimum inter-event time is now given by $\min\{\Delta_k^{\min}, T_k^*\}$. For the case we have $\Delta_k^{\min} \leq T_k^*$, the minimum inter-event time becomes Δ_k^{\min} . Thus, if the sampling time interval δ_k^* is selected such that $\delta_k^* = \gamma\Delta_k^{\min} \leq \Delta_k^{\min}$ for a given $0 < \gamma \leq 1$, the feasibility is guaranteed for all $t \in [t_k, t_k + \delta_k^*]$. On the other hand, for the case we have $\Delta_k^{\min} > T_k^*$, (5.19) is satisfied for all $t \in [t_k, t_k + T_k^*]$. This means that

(5.20) is violated earlier than (5.19). Thus, for the case we have $T_k^* < \Delta_k^{\min}$, we can directly set the next time as $t_{k+1} = t_k + T_k^*$.

Based on the above analysis, the following strategy can be provided as a solution to (P.1):

- (i) If $T_k^* \geq \Delta_k^{\min}$, then set $\delta_k^* = \gamma \Delta_k^{\min}$ for a given $0 < \gamma \leq 1$.
- (ii) If $T_k^* < \Delta_k^{\min}$, then set $t_{k+1} = t_k + T_k^*$ as the next update time.

Remark 5.2. *One may argue that the inter-event time is given by T_k^* for case (b) and that it may thus tend to 0 since T_k^* is decreasing. Note however, that $T_k^* > 0$ always holds while the MPC is implemented (i.e., $x(t_k) \notin \Phi$); if $x(t_k)$ is outside of Φ , there always exists a strictly positive time interval for the optimal state to reach Φ_f . Thus, this guarantees that the inter-event time remains always positive while implementing the MPC. \square*

Next, let us solve (P.2) by modifying the original feasibility conditions. Based on the obtained δ_k^* , the time instants to measure the state and evaluate the event-triggered condition are now given by $t_k + m\delta_k^*$, $m \in \mathbb{N}_{\geq 1}$. To avoid losing the feasibility between two evaluation times, the feasibility condition at one step *future* time is evaluated, instead of the current time instant. That is, at an evaluation time $t = t_k + m\delta_k^*$, $m \in \mathbb{N}_{\geq 1}$, the state $x(t)$ is measured and then the feasibility is checked for $t + \delta_k^*$ instead of t . If the feasibility at $t + \delta_k^*$ is guaranteed, then we move on to the next evaluation time $t + \delta_k^*$. On the other hand, if the feasibility at $t + \delta_k^*$ is not guaranteed, then we set $t_{k+1} = t$. Since we preliminary check the feasibility at one step future time, the loss of feasibility does not occur between two evaluation times.

The feasibility at one step future time can be given by modifying the original feasibility conditions. Suppose at an evaluation time $t = t_k + m\delta_k^*$, $m \in \mathbb{N}_{\geq 1}$, the feasibility at $t + \delta_k^*$ is checked based on the state measurement $x(t)$. The

difference between the actual state and the optimal state at $t + \delta_k^*$ is given by

$$\begin{aligned}
& \|x(t + \delta_k^*) - \hat{x}^*(t + \delta_k^*)\|_{P_f} \\
& \leq e^{L_\phi \delta_k^*} \|x(t) - \hat{x}^*(t)\|_{P_f} + \frac{\tilde{w}_{\max}}{L_\phi} (e^{L_\phi \delta_k^*} - 1) \\
& \leq e^{L_\phi \delta_k^*} \|x(t) - \hat{x}^*(t)\|_{P_f} \\
& \quad + \frac{\lambda_{\min}(\hat{Q}_P)}{4e^{L_\phi T_0^*}} (1 - \alpha) \varepsilon_f (e^{L_\phi \delta_k^*} - 1), \tag{5.23}
\end{aligned}$$

where the condition (5.15) is used to derive the inequality. From the feasibility conditions (5.19), (5.20), the feasibility at $t + \delta_k^*$ is guaranteed if both of the following conditions are satisfied:

$$\begin{aligned}
& \|x(t + \delta_k^*) - \hat{x}^*(t + \delta_k^*)\|_{P_f} < (\varepsilon - \varepsilon_f) e^{-L_\phi T_k^*} \\
& t + \delta_k^* - t_k \leq T_k^*.
\end{aligned}$$

From (5.23), sufficient conditions to satisfy the above equations are then given by

$$\|x(t) - \hat{x}^*(t)\|_{P_f} < (\varepsilon - \varepsilon_f) e^{-L_\phi (T_k^* + \delta_k^*)} - \frac{\lambda_{\min}(\hat{Q}_P)}{4e^{L_\phi T_0^*}} (1 - \alpha) \varepsilon_f (1 - e^{-L_\phi \delta_k^*}), \tag{5.24}$$

$$(m + 1) \delta_k^* \leq T_k^*. \tag{5.25}$$

Note that if (5.24), (5.25) are both satisfied the feasibility is guaranteed at $t + \delta_k^*$, and these conditions can be evaluated based on $x(t)$. Therefore, by using (5.24) and (5.25) as the event-triggered conditions, the violation of the feasibility between two evaluation times will not occur, providing thus a solution to (P.2).

Based on the above results, the over-all proposed algorithm of the event-triggered strategy is summarized below:

Algorithm 5.1: (Event-triggered strategy via intermittent sampling)

- (i) At any update times t_k , $k \in \mathbb{N}_{\geq 0}$, if $x(t_k) \in \Phi$, then switch to the local controller $\kappa(x)$ as a dual mode strategy. Otherwise, the plant transmits $x(t_k)$ to the controller and go to the step (ii).
- (ii) The controller solves Problem 5.1 and obtain the optimal control trajectory $u^*(\xi)$ and the corresponding state $\hat{x}^*(\xi)$ for all $\xi \in [t_k, t_k + T_k]$. Then, calculate T_k^* as the time interval when the state reaches Φ_f , i.e., $\hat{x}^*(t_k + T_k^*) \in \partial\Phi_f$. The controller then computes the sampling time δ_k^* or the next update time t_{k+1} in the following way:
- (a) If $T_k^* \geq \Delta_k^{\min}$, where Δ_k^{\min} is given by (5.22), then set $\delta_k^* = \gamma \Delta_k^{\min}$ for a given $0 < \gamma \leq 1$, and go to step (iii).
- (b) If $T_k^* < \Delta_k^{\min}$, then set $t_{k+1} = t_k + T_k^*$ and go to step (iv).
- Once these are computed, the controller transmits δ_k^* (or t_{k+1}) and the optimal control trajectory $u^*(\xi)$, $\xi \in [t_k, T_k^*]$ to the plant.
- (iii) If the plant receives t_{k+1} , it sets t_{k+1} as the next update time. If δ_k^* is received, it determines the next update time $t_{k+1} (> t_k)$ in the following way:
- (a) Set $m = 1$.
- (b) At an evaluation time $t = t_k + m\delta_k^*$, $m \in \mathbb{N}_{\geq 1}$, measure the state $x(t)$, and check the event-triggered conditions given by (5.24), (5.25).
- (c) If (5.24) and (5.25) are both satisfied, then apply $u^*(\xi)$ for $\xi \in [t, t + \delta_k^*]$. Then, set $m \leftarrow m + 1$ and go back to step (b). Otherwise, set $t_{k+1} = t$ and go to step (iv).
- (iv) $k \leftarrow k + 1$ and go back to step (i). □

Remark 5.3 (On tuning the parameter γ). If γ is chosen larger, then we obtain larger $\delta_k^* (= \gamma \Delta_k^{\min})$, and thus a smaller number of state measurements and evaluations

may be obtained. However, due to the increased value of δ_k^* , the right hand side of (5.24) becomes smaller and the event-triggered condition becomes more conservative, resulting in a larger number of OCPs. Therefore, there exists a trade-off between the number of OCPs and the number of state measurements, and the parameter γ plays an important role to regulate this trade-off. This property serves as one of the benefits of our proposed strategy, as we can now appropriately select γ according to whether we would like to focus on reducing the number of OCPs or number of state measurements.

□

5.4 Self-triggered strategy

In the self-triggered strategy, the next update time t_{k+1} is pre-determined at t_k as soon as the OCP is solved, without having to evaluate the event-triggered condition. To obtain the self-triggered strategy, recall that the minimum inter-event time of satisfying (5.19) and (5.20) is $\min\{\Delta_k^{\min}, T_k^*\}$, where Δ_k^{\min} is given by (5.22). Since Δ_k^{\min}, T_k^* can be obtained at t_k (immediately after solving the OCP), the controller can simply set t_{k+1} as

$$t_{k+1} = t_k + \min\{\Delta_k^{\min}, T_k^*\}. \quad (5.26)$$

Although considering the minimum inter-event time may lead to more conservative result than the previous even-triggered strategy, the evaluations of the event-triggered condition and the state measurements are no longer required between two update times of the OCP. Thus, the following self-triggered strategy is obtained:

Algorithm 5.2: (Self-triggered Strategy)

- (i) At any update times $t_k, k \in \mathbb{N}_{\geq 0}$, if $x(t_k) \in \Phi$, then switch to the local controller $\kappa(x)$ as a dual mode strategy. Otherwise, the plant transmits $x(t_k)$ to the controller and go to the step (ii).

- (ii) The controller solves Problem 5.1 and obtain the optimal control $u^*(\xi)$ and the corresponding state trajectory $\hat{x}^*(\xi)$ for all $\xi \in [t_k, t_k + T_k]$. Then, calculate T_k^* as the time interval when the state reaches Φ_f , i.e., $\hat{x}^*(t_k + T_k^*) \in \partial\Phi_f$. Furthermore, calculate Δ_k^{\min} according to (5.22). Then, the controller sets the next update time t_{k+1} as

$$t_{k+1} = t_k + \min\{\Delta_k^{\min}, T_k^*\} \quad (5.27)$$

and applies $u^*(t)$ for all $t \in [t_k, t_{k+1})$.

- (iii) $k \leftarrow k + 1$ and go back to step (i). □

5.5 Stability analysis

For a given initial prediction horizon $T_0 > 0$, let $\mathcal{X}(T_0)$ be the set of states such that a feasible solution to Problem 5.1 exists. The author will prove in the following that, any state trajectories starting from inside $\mathcal{X}(T_0)$ will eventually enter Φ within a prescribed finite time interval.

Theorem 5.2. *Consider the nonlinear system given by (5.1), and suppose that the event-triggered strategy (Algorithm 5.1) or the self-triggered strategy (Algorithm 5.2) is implemented. Then, for any $w(t)$ satisfying $\|w(t)\|_{P_f} \leq \min\{\hat{w}_{\max}, \tilde{w}_{\max}\}, \forall t \geq t_0$, where \hat{w}_{\max} and \tilde{w}_{\max} are given by (5.7), (5.15) respectively, any state trajectories starting from $x(t_0) \in \mathcal{X}(T_0)$ enter Φ within the time interval T_0^*/α , and remain in Φ for all the future times.*

Proof. The statement is proved by contradiction. Assume at t_k that we have $t_k - t_0 \geq T_0^*/\alpha$, and $x(t_k)$ is outside of Φ , i.e., $x(t_k) \notin \Phi$. Since $x(t_k) \notin \Phi$ and $\Phi_f \subset \Phi$, we have $T_k^* > 0$. As $x(t_0) \in \mathcal{X}(T_0)$ and $\|w(t)\|_{P_f} \leq \tilde{w}_{\max}, \forall t \geq t_0$, applying Algorithm 5.1 or Algorithm 5.2 ensures that the feasibility is guaranteed for all

t_0, t_1, \dots, t_k . Thus, we recursively obtain from (5.11) that:

$$\begin{aligned}
T_k^* &\leq T_{k-1}^* - \alpha \Delta_{k-1} \leq T_{k-2}^* - \alpha(\Delta_{k-1} + \Delta_{k-2}) \\
&\leq \dots \leq T_0^* - \alpha \sum_{l=1}^{k-1} \Delta_l \\
&= T_0^* - \alpha(t_k - t_{k-1} + t_{k-1} - t_{k-2} + \dots + t_1 - t_0) \\
&= T_0^* - \alpha(t_k - t_0).
\end{aligned}$$

Thus, by the assumption $t_k - t_0 \geq T_0^*/\alpha$, we obtain $T_k^* \leq 0$. However, this clearly contradicts to the fact that we have $T_k^* > 0$. Thus, it is shown that the state enters Φ within the time interval T_0^*/α . Furthermore, since from Lemma 5.1, Φ is a positively invariant set with the disturbance satisfying $\|w(t)\| \leq \hat{w}_{\max}$, the state remains in Φ for all future times. This completes the proof. \square

Remark 5.4 (On the novelty of convergence times). *Aside from the event-triggered strategy, one of the important results of this paper is that, by guaranteeing stability without using optimal cost, the maximum time of convergence is explicitly obtained by Theorem 5.2. Although the convergence time has been analyzed for linear discrete-time systems, e.g., [69], this paper derives it for nonlinear continuous-time systems with additive bounded disturbances.* \square

Remark 5.5 (On the control performance). *In Theorem 5.2, stability is proven by evaluating a time interval to reach Φ_f , and not by the optimal cost. Although this may be unconventional with respect to a control performance view point, our approach is advantageous and practical from an event-triggered control view point, since the event-triggered condition provides less conservative results than the approach presented in Chapter 4. Moreover, the control performance can be evaluated by tuning the parameter α ; for more details, please see Remark 5.6 below.* \square

Remark 5.6 (Convergence time v.s. Disturbance). *If α is chosen larger, then we obtain smaller T_0^*/α and faster convergence is obtained. However, this in turn means from (5.15) that the allowable size of disturbance becomes smaller, which implies that the robustness to the noise or model uncertainty may be degraded. Therefore, there*

exists a trade-off between the convergence time of the state trajectory and the allowable size of the disturbance, and this trade-off can be regulated by tuning α . \square

5.6 Simulation results

In this section the author illustrates the effectiveness of our proposed aperiodic MPC schemes. Again, simulations were conducted on Matlab 2016a under Windows 10, Intel(R) Core(TM) 2.40 GHz, 8 GB RAM. As a software package, the author used Imperial College London Optimal Control Software (ICLOCS) (see [67]), in order to solve (non)linear optimal control problems in the continuous-time domain.

(*Example 5.1*): Consider the following linearized system of an inverted pendulum on a cart:

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t),$$

where $x = [x_1 \ x_2 \ x_3 \ x_4]^T \in \mathbb{R}^4$, $u \in \mathbb{R}$ and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -mg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/\ell & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/M\ell \end{bmatrix} \quad (5.28)$$

with $m = 1$, $M = 5$, $\ell = 2$, and $g = 9.8$. The constraint for the control input is given by $\mathcal{U} = \{u \in \mathbb{R} : |u| \leq 10\}$. The computed Lipschitz constant is $L_f = 5.28$ and we have $\varepsilon = 0.08$, $\varepsilon_f = 0.05$, and $\alpha = 0.8$, $\gamma = 0.5$. The initial state is assumed to be given by $x(t_0) = [1, 0, 0, 0]$, and the initial prediction horizon is $T_0 = 10$. From Theorem 5.1, the feasibility is guaranteed if $\tilde{w}_{\max} = 8.3 \times 10^{-4}$ and from Lemma 5.1 the region Φ is positively invariant if $\hat{w}_{\max} = 2.0 \times 10^{-3}$. Taking into account both restrictions, assume that $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_{P_f} \leq 8.3 \times 10^{-4}\}$.

In this set up, from Theorem 5.2 it is guaranteed that the state enters the local set Φ within the time interval $T_0/\alpha = 12.5$.

Figure 5.4 illustrates the resulting state trajectories of x_1, x_2 , and Fig. 5.5 illustrates those of x_3, x_4 , by applying the event-triggered strategy (Algorithm 5.1). From the figure, it is shown that the state trajectories are stabilized *around* the origin by applying Algorithm 5.1. The time when the state enters the set Φ is given by 7.98, and the number of transmission instants to achieve the convergence is given by 9 (i.e., the average transmission time interval is 0.88). In the figures, the state trajectories by applying periodic MPC scheme with 0.1 sampling time interval are also illustrated. From the figure, it is shown that the trajectory also converges to the origin while achieving similar convergence to Algorithm 5.1. In both proposed scheme and the periodic scheme, the state trajectories are wobbling around the origin (see in particular Fig. 5.5) due to the effect of disturbances. Figure 5.6 illustrates the applied control input by Algorithm 5.1 (blue) and the periodic case (red dotted line). From the figure, due to the effect of disturbances, control inputs under Algorithm 5.1 behave differently from the periodic case. Note that as shown in Fig. 5.6, control inputs are given as a continuous trajectory rather than a sample-and-hold implementation, since it is assumed that for each $t_k, k \in \mathbb{N}$ the optimal control trajectory $u^*(t), \forall t \in [t_k, t_{k+1}]$ is transmitted to the plant (see (5.12)).

Table 5.1 illustrates the convergence time when the state enters the local set Φ , as well as the number of transmission instants during the time interval $t \in [0, 30]$. In Theorem 5.2, it is shown that the state trajectory converges to the local set Φ within $T_0/\alpha = 12.5$. From the table, the convergence time to Φ is indeed smaller than T_0/α , which validates the result in Theorem 5.2. The convergence time by applying Algorithm 5.1 is almost the same as the one by applying the periodic case. Moreover, the number of transmission instants by applying Algorithm 5.1 is smaller by $84 - 5 = 79$ than the periodic case, which shows that the communication reduction is achieved by applying the proposed scheme.

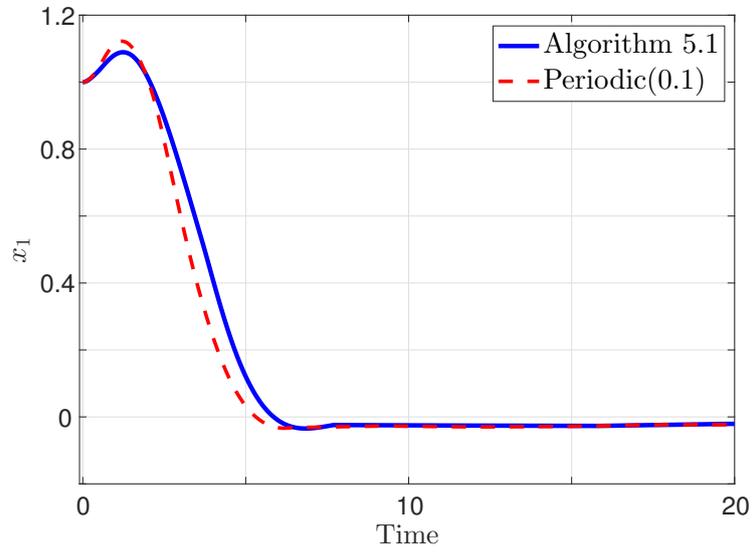
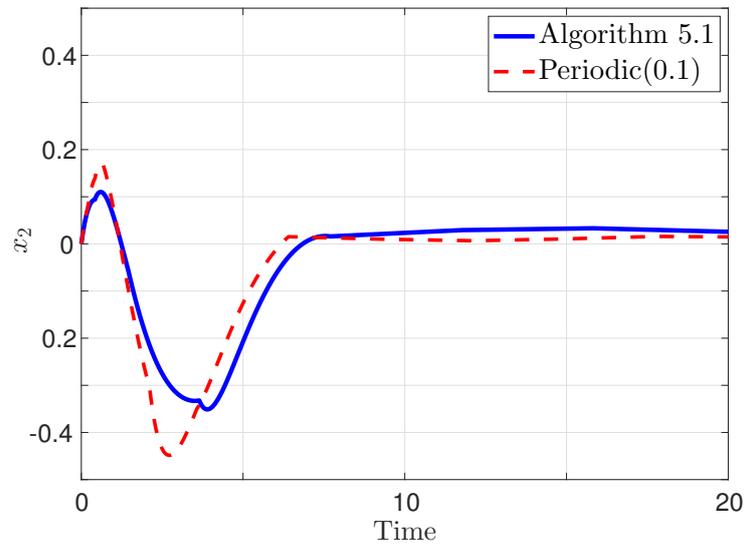
(a) State trajectories of x_1 .(b) State trajectory of x_2 .

FIGURE 5.4: State trajectories of x_1 and x_2 by implementing Algorithm 5.1 (blue solid lines) and the periodic MPC (red dotted lines).

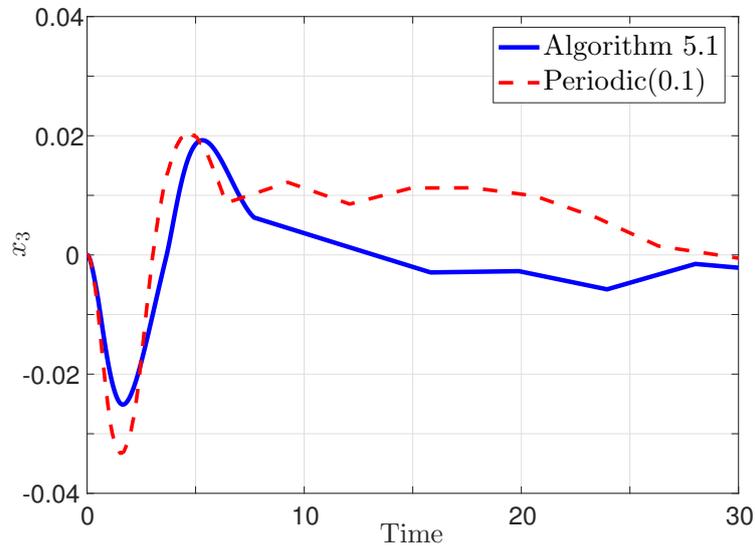
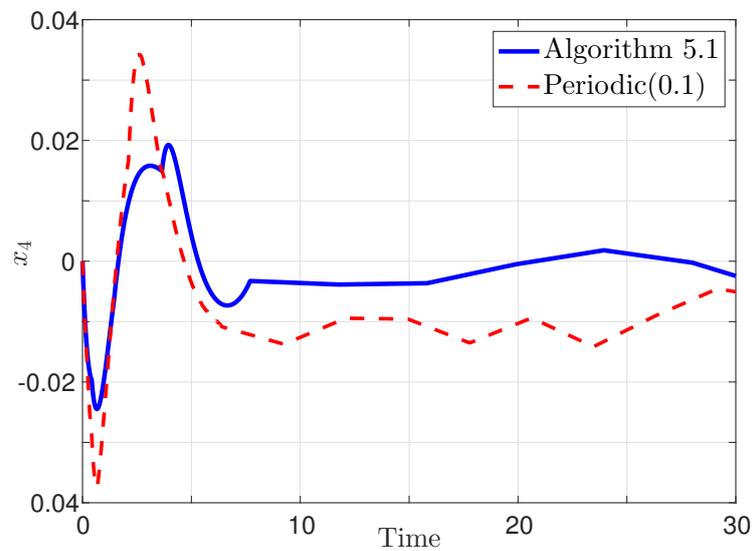
(a) State trajectories of x_3 .(b) State trajectory of x_4 .

FIGURE 5.5: State trajectories of x_3 and x_4 by implementing Algorithm 5.1 (blue solid lines) and the periodic MPC (red dotted lines).

TABLE 5.1: Convergence time when the state trajectory enters Φ and the number of transmission instants.

	Algorithm 5.1	Periodic (0.1)
T_0/α	12.5	12.5
Convergence time	7.98	7.96
Transmission instants	5	84

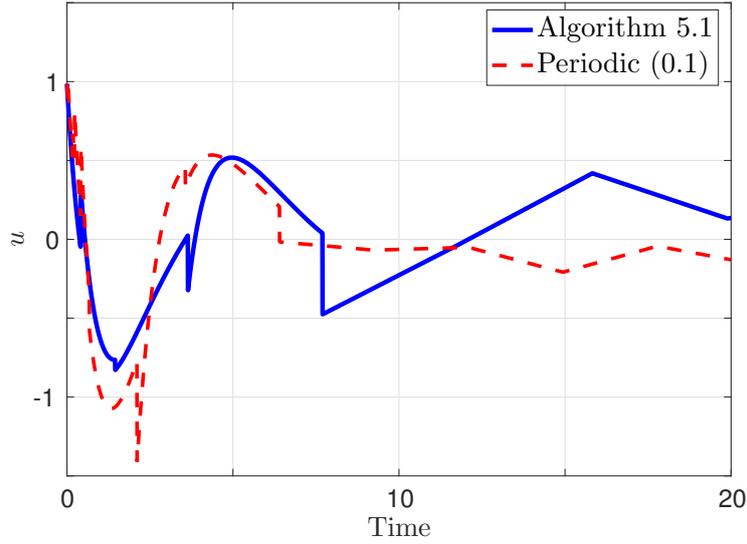


FIGURE 5.6: Control inputs by applying Algorithm 5.1 (blue line) and the periodic scheme (red dotted line).

To conclude, it is shown in this example that:

- A more communication reduction is achieved by applying Algorithm 5.1 than by applying the periodic scheme with 0.1 sampling time interval.
- By applying Algorithm 5.2, the state trajectories enter the local set Φ within T_0/α , which validates the result of Theorem 5.2.

(Example 5.2): As with Example 4.2, consider a control problem of non-holonomic vehicle in two dimensions, where the dynamics are given by

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (5.29)$$

where the state is denoted as $\chi = [x, y, \theta] \in \mathbb{R}^3$, consisting of the position of the vehicle $[x, y]$, and its orientation θ . $u = [v, \omega] \in \mathbb{R}^2$ is the control input and the constraints are assumed to be given by $\|v\| \leq \bar{v} = 2.0$ and $\|\omega\| \leq \bar{\omega} = 1.0$. The computed Lipschitz constant L_ϕ is given by $L_\phi = \sqrt{2\bar{v}}$. The stage and the terminal cost are given by $F = \chi^T Q \chi + u^T R u$, and $V_f = \chi^T \chi$ where $Q = I_3$ and $R = I_2$. The prediction horizon is $T_0 = 10$ and $\alpha = 0.8$. The parameter for characterizing the terminal set is $\varepsilon = 0.8$ and $\varepsilon_f = 0.4$. Again, assume that the initial state is given by $\chi(0) = [x(0); y(0); \theta(0)] = [-10; -5; \pi/2]$. Theorem 5.1 states that feasibility is guaranteed if $\tilde{w}_{\max} = 2.0 \times 10^{-3}$ and from Lemma 5.1 the region Φ is positively invariant if $\hat{w}_{\max} = 1.5 \times 10^{-3}$. Taking into account both restrictions, it is assumed that $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_{P_f} \leq 1.5 \times 10^{-3}\}$. In this set up, from Theorem 5.2 it is guaranteed that the state enters the local set Φ within $T_0/\alpha = 12$.

Figure 5.7 illustrates the resulting trajectory of the vehicle by applying Algorithm 5.1. From the figure, the trajectory of the vehicle is stabilized around the origin, while it is perturbed by additive bounded disturbances. Figure 5.8 illustrates the resulting control input by applying Algorithm 5.1. From the figure, it is shown that the control inputs satisfy the constraints $\|v(t)\| \leq \bar{v} = 2.0$, $\|\omega(t)\| \leq \bar{\omega} = 1.0$, and the optimal control problem is updated only when they are needed by applying Algorithm 5.1. The time when the state enters Φ is 9.0, and the number of transmission instants until the state converges the region is given by 9.0 (i.e., the average transmission time interval is 1.0). To

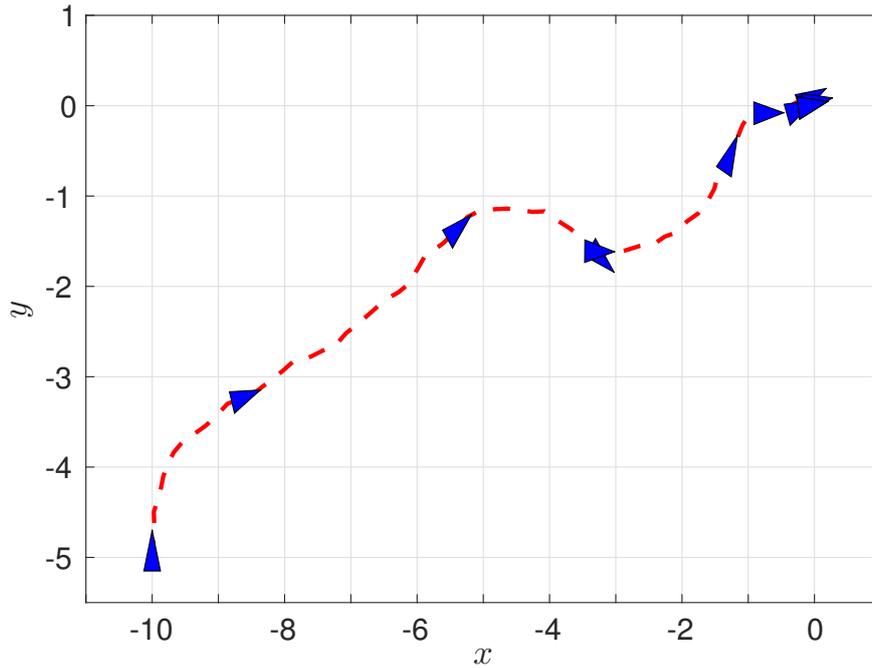


FIGURE 5.7: Trajectory of the vehicle by applying Algorithm 5.1.

make comparisons, Fig. 5.9 illustrates the resulting state trajectory by applying the periodic MPC scheme with 1.0 sampling time interval (i.e., $t_k = 1.0k$, $\forall k \in \mathbb{N}$), which is equal to the average transmission time interval by applying Algorithm 5.1. From the figure, it is shown that the state trajectory does *not* converge around the origin. Again, this is due to the fact that the transmission time interval is not suitably selected to guarantee stability when the periodic scheme is employed.

Figure 5.11 illustrates the resulting state trajectory by applying the periodic MPC scheme with 0.1 sampling time interval (i.e., $t_k = 0.1k$, $\forall k \in \mathbb{N}$), which is much smaller than the average transmission time interval by Algorithm 5.1. Table 5.2 illustrates the convergence time when the state enters Φ , as well as the number of transmission instants until the convergence is attained. From the table, Algorithm 5.1 achieves the number of transmission instants smaller by $67 - 11 = 56$ than the periodic case. On the other hand, Algorithm 5.1 requires $9.0 - 6.7 = 2.3$ longer convergence time than the periodic one, which

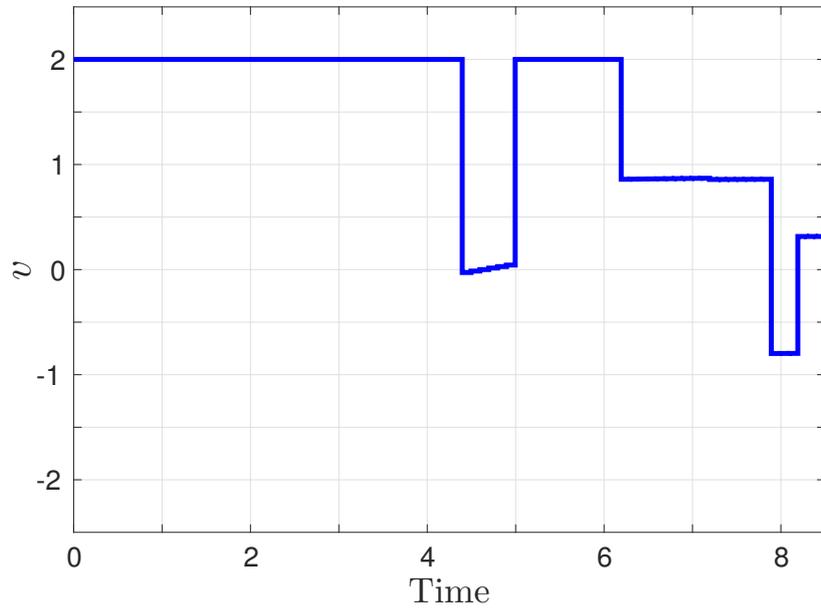
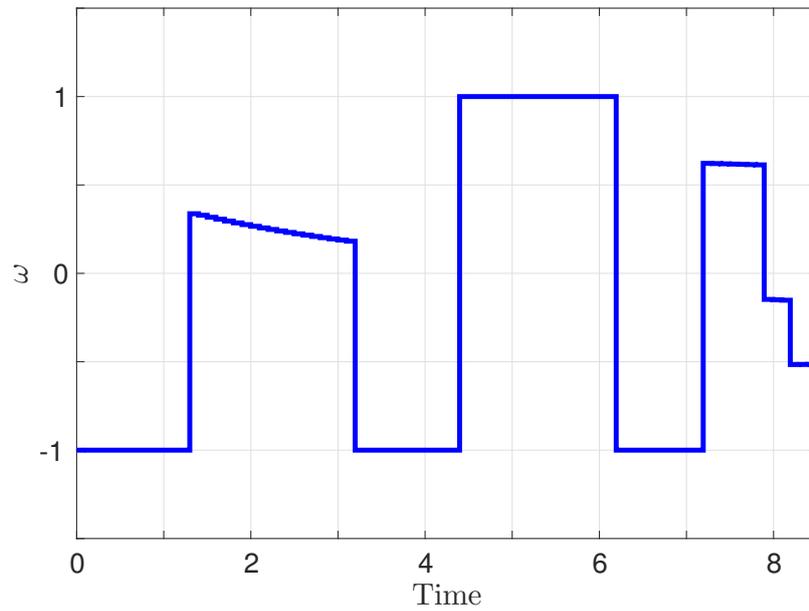
(a) Control trajectory of v .(b) Control trajectory of ω .

FIGURE 5.8: Control trajectory of v and ω implementing Algorithm 5.1 and the periodic one without disturbances (red dotted line).

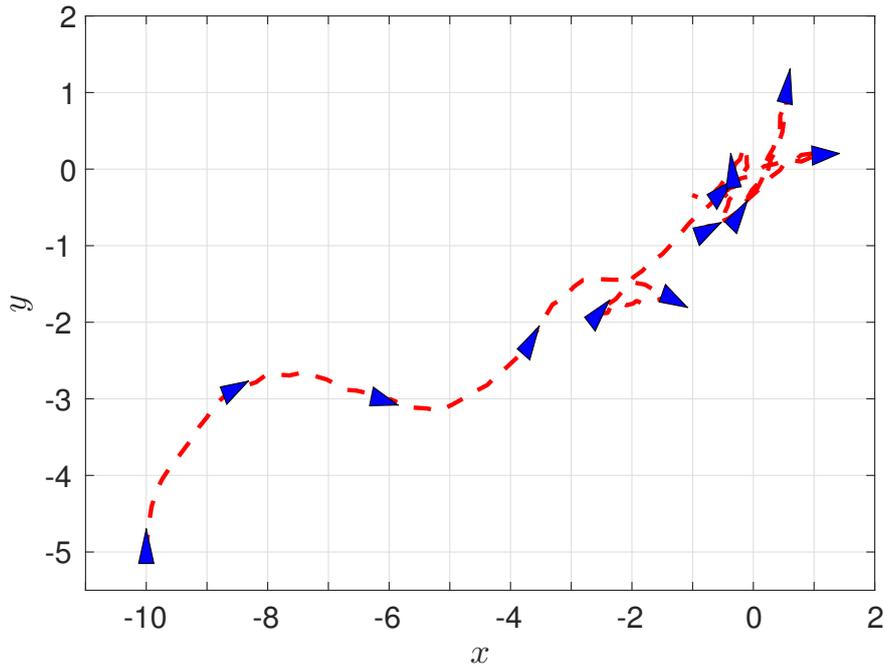
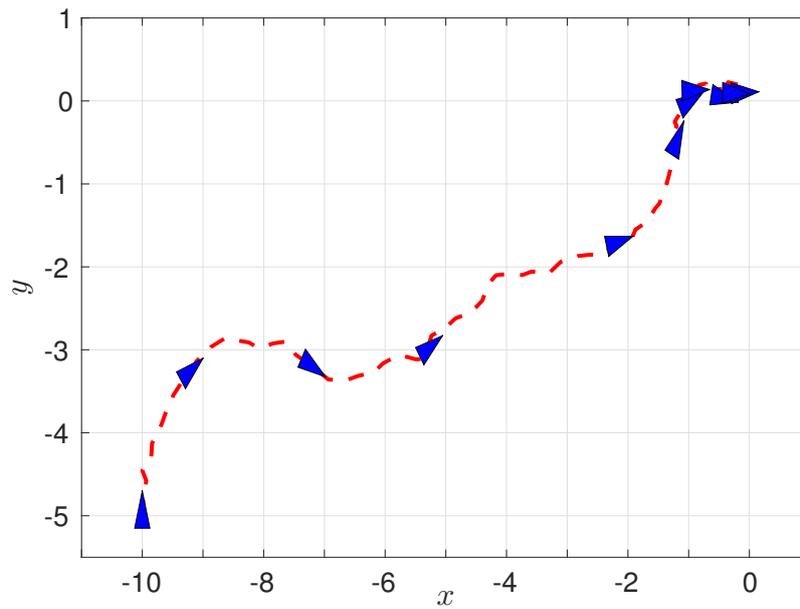


FIGURE 5.9: Trajectory of the vehicle by applying the periodic MPC with 1.0 sampling time interval.

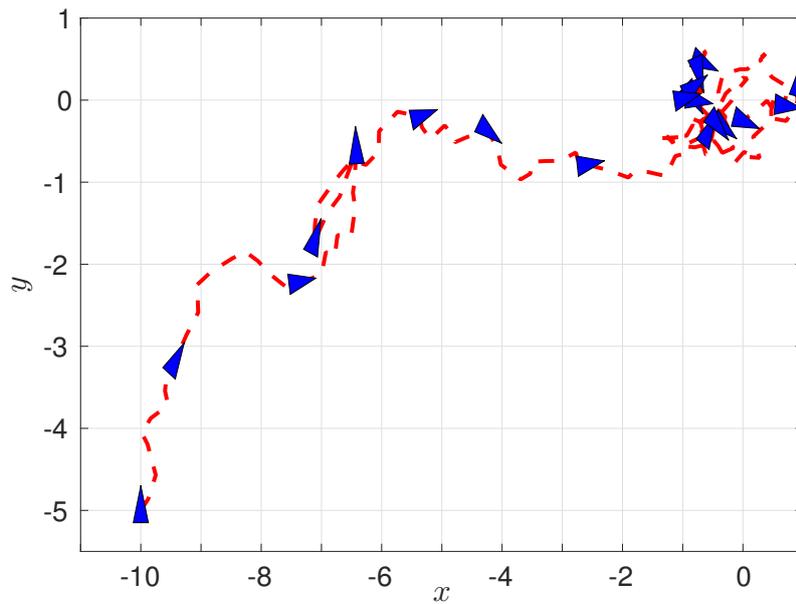
indicates that the periodic scheme achieves better control performance. Therefore, as with Chapter 4, it is shown in this example that there exists a tradeoff between achieving the communication reduction and the control performance.

To analyze the effect of disturbances, Algorithm 5.1 is again implemented with $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_{P_f} \leq 6.0 \times 10^{-3}\}$, which is larger than the former case. Fig. 5.10(a) illustrates the resulting trajectory of the vehicle by applying Algorithm 5.1. It is shown that the vehicle still enters the set Φ , even though the disturbance size is bigger than the allowable size obtained from Theorem 5.1. This is due to the fact that Theorem 5.1 provides only sufficient (conservative) conditions, which means that a larger disturbance size may be allowed to guarantee feasibility and stability. Figure 5.10(b) illustrates the resulting trajectory of the vehicle by applying Algorithm 5.1 with $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_{P_f} \leq 1.0 \times 10^{-2}\}$. In this case, it is shown that the trajectory does *not* converge Φ in finite time, which has been occurred since the disturbance size has been selected too large.

To provide a further analysis, Table 5.3 illustrates the resulting convergence



(a) $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_{P_f} \leq 6.0 \times 10^{-3}\}$.



(b) $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_{P_f} \leq 1.0 \times 10^{-2}\}$.

FIGURE 5.10: Trajectory of the vehicle by applying Algorithm 5.1 with large disturbance sets \mathcal{W} .

TABLE 5.2: Convergence time when the state trajectory enters around the origin and the number of transmission instants.

	Algorithm 5.1	Periodic (0.1)
Convergence time	9.0	6.7
Transmission instants	11	67

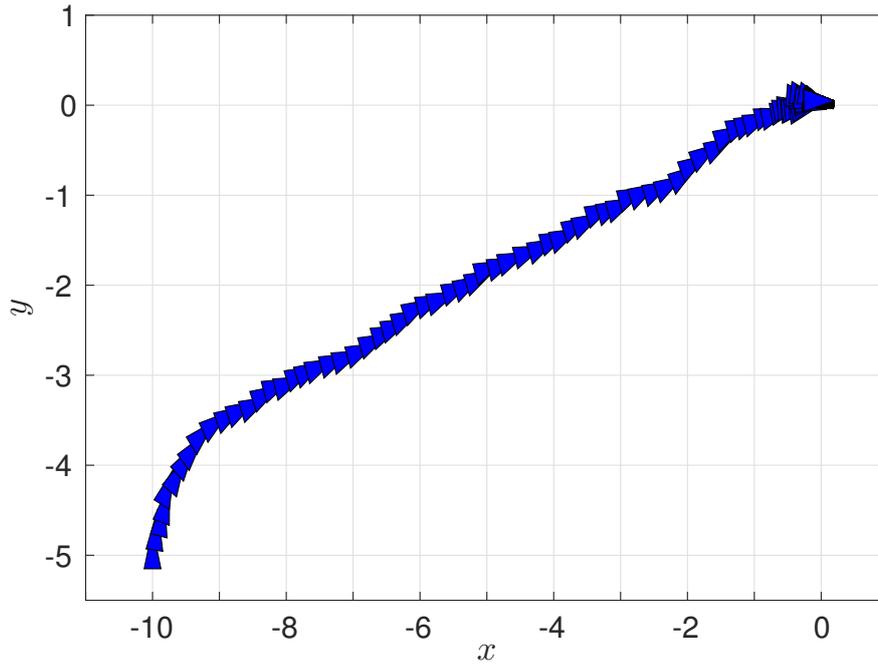


FIGURE 5.11: Trajectory of the vehicle by applying the periodic MPC with 0.1 sampling time interval.

time and the number of transmission instants until the trajectory enters the local set Φ , by applying Algorithm 5.1, 5.2, and 4.2 with disturbances taken into account (see Remark 4.5). Note that as shown in the table, T_0/α is not defined when applying Algorithm 4.2. From the table, the convergence times to the set Φ are indeed smaller than T_0/α for both Algorithm 5.1 and 5.2, which validates the theoretical result in Theorem 5.2 also for the nonlinear case. The number of transmission instants by applying Algorithm 5.1 and 5.2 becomes much smaller than Algorithm 4.2. As previously mentioned, this is due to the fact that Algorithm 4.2 tends to be conservative as it includes unsuitable parameters as a potential source of conservativeness. The number of transmission instants by applying Algorithm 5.1 (event-triggered strategy) is smaller by $18 - 12 = 6$ than the ones by applying Algorithm 5.2 (self-triggered strategy), which means that the event-triggered strategy achieves a less communication load than the self-triggered strategy. This is due to the fact that the self-triggered strategy is a sufficient condition to the event-triggered one, meaning that the latter one becomes more conservative than the former case.

TABLE 5.3: Convergence time when the state trajectory enters Φ and the number of transmission instants.

	Algorithm 5.1	Algorithm 5.2	Algorithm 4.2
T_0/α	12	12	—
Convergence time	9.5	9.5	9.4
Transmission instants	12	18	25

To conclude, it is shown in this example that:

- By applying Algorithm 5.1 and 5.2, the trajectory of the vehicle enters the local set Φ within T_0/α , which validates the result of Theorem 5.2.
- Algorithm 5.1 and 5.2 require less communication load than Algorithm 4.2, which validates that a less conservative result is obtained than by using Lyapunov stability.
- Algorithm 5.1 (event-triggered strategy) achieves less communication load than Algorithm 5.2, which is due to the fact that the latter one is more conservative than the former one.

5.7 Summary

In this chapter, an aperiodic formulation of MPC is proposed for nonlinear system under additive bounded disturbances. The new aperiodic scheme is derived based on recursive feasibility, in which the optimal control problem has a solution when the error between the predictive states and the optimal states is below a certain threshold. In the event-triggered strategy, the triggering condition is given such that the plant requires to measure the state information only at a certain sampling time instants. This leads to an alleviation of sensing cost to evaluate the event-triggered condition by not having to measure the state continuously. A self-triggered strategy is also given by deriving a sufficient

condition to the event-triggered strategy. Stability is rigorously shown by guaranteeing that the state trajectories converge to a prescribed local set Φ in finite time. Here, in contrast to the approaches presented in the previous chapters, the time interval when the state trajectory enters the local set Φ is evaluated, instead of evaluating the optimal cost. Some numerical examples validate the effectiveness of the proposed approach by considering both linear and nonlinear systems. For linear case, a control problem of an inverted pendulum on a cart is considered, and show that the state trajectory is stabilized around the origin while at the same time reducing the communication load compared to the periodic case. For nonlinear case, we consider a control problem of vehicle regulation, and show that the state trajectories are stabilized around the origin within the prescribed time interval according to the stability theorem.

Chapter 6

Conclusion and future work

Finally, some conclusions of this thesis and future works are provided in this chapter.

6.1 Conclusion

In this thesis, various formulations of aperiodic MPC schemes for networked control systems are proposed, including both linear and non-linear systems. In the proposed schemes, the plant transmits the state information to the controller and it solves an optimal control problem only when it is needed, aiming at reducing the communication load for networked control systems.

First of all, the author considers a control problem for LTI systems and proposes two self-triggered strategies (Chapter 3). The main idea is that the controller solves a multiple optimal control problems under different discretization schemes, and the controller determines suitable transmission time intervals by evaluating both control performance and communication reduction. Moreover, a method to reduce computational complexity of solving an optimal control problem is given by incorporating the notion of contractive set. The effectiveness of the proposed approaches are validated by considering several simulation examples.

Next, a self-triggered strategy is developed for nonlinear input-affine systems as provided in Chapter 4. The main contribution is that a sufficient condition to guarantee a Lyapunov stability is derived, such that the optimal cost as

a Lyapunov function candidate is guaranteed to decrease under a sample-and-hold implementation. Moreover, the author considers a case when multiple control samples are allowed to be sent, and provide an efficient strategy to determine suitable control samples to be transmitted. The proposed scheme is illustrated by considering some simulation examples, in particular a control problem of continuously stirred tank reactor system.

Finally, event and self-triggered strategies for nonlinear systems are developed in Chapter 5, which are provided for a more general class of systems than Chapter 3 and 4. The event-triggered strategy is first derived as a condition to guarantee recursive feasibility, and a sufficient condition is given to derive the corresponding self-triggered condition. Regarding stability, it is shown that the state trajectory enters a terminal region within a prescribed time interval by guaranteeing that the time interval of an optimal state trajectory to enter a terminal region is strictly decreasing until it achieves the convergence.

6.2 Future work

The proposed strategies may have the potential to apply and extend to various types of MPC formulations and applications, as the detail is described below.

In this thesis, a terminal constraint is basically imposed in the optimal control problems such that stability of the origin is guaranteed. While this constraint is useful in guaranteeing stability in a theoretical manner, such arbitrary constraint makes an optimal control problem generally hard to be solved. To overcome this issue, a MPC framework that does *not* utilize the terminal constraint has been developed in recent years, see e.g., [39], [40], and a relation to the case with terminal constraint is also discussed in [70]. In those results, feasibility and stability are analyzed *without* imposing the terminal constraint in the optimal control problem. Therefore, our future work involves deriving an aperiodic formulation of MPC for such unconstrained setup, aiming at alleviating computational complexity of solving the optimal control problem.

In Chapter 5, an aperiodic formulation of MPC has been developed for nonlinear systems under additive disturbances. In the proposed method, the basic methodology is followed from [71], where an optimal control trajectory is applied in an open-loop fashion until the next update time. However, as pointed out in [69] this open-loop formulation may yield a tight (or conservative) condition to guarantee feasibility and stability. Indeed, the maximum allowable disturbance size derived in Example 5.1 is given by $w_{\max} = 8.3 \times 10^{-4}$, which is very small. To overcome this conservativeness, the so-called *tube-based* MPC has been developed as an alternative methodology, see e.g., [69], [72]. In the tube based approach, an optimal *control policy* rather than the open loop controller is designed. Here, the control policy means that the controller designs the closed-loop, state feedback controller (e.x., $u^*(\xi) = Kx(\xi) + v$, $\forall \xi \in [t_k, t_k + T_p]$ for each update time t_k), rather than obtain the open-loop control trajectory. As has been already illustrated in [69], the tube-based strategy yields a less conservative result than the open-loop formulation. Therefore, it is of interest to extend our proposed framework to tube based MPC and develop a strategy such that a larger disturbance size is allowed to guarantee recursive feasibility.

In Chapter 5, the author considers a control problem of nonlinear systems with *bounded* disturbances. A more interesting and useful setup may be to consider system with *unbounded* disturbances, such as gaussian noise. Several MPC formulations have been proposed in the literature, such as chance constrained MPC [73]. In this approach, the authors consider probabilistic constraints, which are in general translated to the deterministic one when solving an optimal control problem (see e.g., [73]). Thus, it may be of interest and has the potential to extend our approach to the probabilistic formulation, such that unbounded disturbances can be taken into account.

In networked control systems, transmitting control packets over a communication network may induce network constraints and uncertainties, such as

limited communication bandwidth, time delays and packet losses. In this thesis, a limited nature of communication is taken into account in Chapter 3 and Chapter 4. While we can utilize several existing techniques to compensate network delays and packet losses (for details, see Remark 4.4), some theoretical challenges still remain to be considered. For example, in the approach presented in Remark 4.4, which basically employs the result in [62], the controller provides the forward prediction of the state in order to compensate network delays and packet losses. However, this forward prediction may not be applicable for systems under model uncertainties or disturbances, since the predicted states do not necessarily coincide with the actual states. Moreover, these compensation techniques do not provide an inherent robustness of the system against network delays or packet losses. Therefore, our future work involves analyzing how much network delays or packet losses can be tolerated theoretically in the system in order to guarantee feasibility and stability.

Appendix A

Mathematical preliminaries

In this appendix, some basic mathematical preliminaries that have been used in this thesis are provided.

A.1 Lyapunov Stability

Consider the autonomous system:

$$\dot{x}(t) = f(x(t)), \quad (\text{A.1})$$

where $x \in \mathbb{R}^n$. Without loss of generality, assume that the origin is an equilibrium point, i.e., $f(0) = 0$. The concept of Lyapunov stability is defined as follows.

Definition A.1 (Lyapunov stability). *The equilibrium point $x = 0$ is*

- *stable, if for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon)$ such that*

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \quad \forall t \geq 0 \quad (\text{A.2})$$

- *asymptotically stable if it is stable and δ can be chosen such that*

$$\|x(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0. \quad (\text{A.3})$$

State in words, the equilibrium point is stable if for every $\varepsilon > 0$ there exists a corresponding value of δ , which may be dependent on ε , such that the state remains in ε neighborhood of the origin. The equilibrium point is asymptotically stable if the state converges to the origin as the time goes infinity.

Lyapunov stability theorem is a useful tool to check the above stability concepts without needing to evaluate the differential equation (A.1) explicitly.

Theorem A.1 (Lyapunov stability theorem). *Let $V : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function such that $V(0) = 0$, $V(x) > 0$, $\forall x \in \mathbb{R} \setminus \{0\}$ and $\dot{V}(x) \leq 0$, $\forall x \in \mathbb{R}$. Then, the origin is stable. Moreover, if $\dot{V}(x) < 0$, $\forall x \in \mathbb{R} \setminus \{0\}$, the origin is asymptotically stable.*

Roughly speaking, the origin is stable if there exists a positive definite function V such that the time derivative is non-negative for all $x \in \mathbb{R}$. If the time derivative is strictly decreasing, then the origin is asymptotically stable.

A.2 Gronwall-Bellman inequality

Lemma A.1. *For a given $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}$ if a continuous function $y : [0, a]$ satisfies*

$$y(t) \leq \lambda + \int_a^t \mu y(\xi) d\xi \quad (\text{A.4})$$

then, we have

$$y(t) \leq \lambda \exp(\mu t) \quad (\text{A.5})$$

for all $t \in [0, a]$.

For proof, see Lemma A.1 in [52]. Lemma A.1 has been utilized in (4.21), where $y(t) = \|x(t) - x^*(t)\|$, $\mu = L_\phi$ and $\lambda = \frac{1}{2}L_G K_u \delta_1^2$.

Bibliography

- [1] R. A. Gupta and M.-Y. Chow, "Networked Control System: Overview and Research Trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
- [2] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
- [3] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.
- [4] G. C. Walsh, H. Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.
- [5] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1177–1181, 2005.
- [6] D. S. Kim, Y. S. Lee, W. H. Kwon, and H. S. Park, "Maximum allowable delay bounds of networked control systems," *Control Engineering Practice*, vol. 11, no. 11, pp. 1301–1313, 2003.
- [7] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 3270–3285.
- [8] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1992–1997, 2014.

-
- [9] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
- [10] W. P. M. H. Heemels and M. C. F. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.
- [11] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, 2013.
- [12] M. C. F. Donkers and W. P. M. H. Heemels, "Output-based event-triggered control with guaranteed \mathcal{L}_∞ -gain and decentralized event-triggering," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1362–1376, 2011.
- [13] X. Wang and M. D. Lemmon, "On event design in event-triggered feedback systems," *Automatica*, vol. 47, no. 10, pp. 2319–2322, 2011.
- [14] V. S. Dolk and W. P. M. H. Heemels, "Event-triggered control systems under packet losses," *Automatica*, vol. 80, pp. 143–155, 2017.
- [15] V. S. Dolk, D. P. Borgers, and W. P. M. H. Heemels, "Output-based and decentralized dynamic event-triggered control with guaranteed \mathcal{L}_p -gain performance and zeno-freeness," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 34–49, 2016.
- [16] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete time systems," in *Proceedings of American Control Conference (ACC)*, 2010, pp. 4719–4724.
- [17] R. Postoyan, P. Tabuada, D. Nesic, and A. Anta, "A framework for the event-triggered stabilization of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 982–996, 2014.

-
- [18] R. Postoyan, A. Anta, W. P. M. H. Heemels, P. Tabuada, and D. Nesić, "Periodic event-triggered control for nonlinear systems," in *Proceedings of the 52nd IEEE Conference on Decision and Control (IEEE CDC)*, 2013, pp. 7397–7402.
- [19] D. V. Dimagoronas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.
- [20] G. S. Seyboth, D. V. Dimagoronas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.
- [21] X. Wang and M. D. Lemmon, "Self-triggered feedback control systems with finite \mathcal{L}_2 -gain stability," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 452–467, 2009.
- [22] M. Mazo, A. Anta, and P. Tabuada, "On self-triggered control for linear systems: guarantees and complexity," in *Proceedings of European Control Conference (ECC)*, 2009, pp. 3767–3772.
- [23] M. Mazo, A. Anta, and P. Tabuada, "An iss self-triggered implementation of linear controllers," *Automatica*, vol. 46, no. 8, pp. 1310–1314, 2010.
- [24] A. Anta and P. Tabuada, "To sample or not to sample: self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.
- [25] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, "System Architectures, Protocols and Algorithms for Aperiodic Wireless Control Systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 175–184, 2013.

- [26] C. Peng, D. Yue, and M.-R. Fei, "A Higher Energy-Efficient Sampling Scheme for Networked Control Systems over IEEE 802.14.4 Wireless Networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1744–1766, 2016.
- [27] U. Tiberi, C. Faschione, K. H. Johansson, and M. D. D. Benedetto, "Energy-efficient sampling of networked control systems over IEEE 802.15.4 wireless networks," *Automatica*, vol. 49, no. 3, pp. 712–724, 2013.
- [28] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.
- [29] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [30] J. Richalet, "Industrial applications of model based predictive control," *Automatica*, vol. 29, no. 5, pp. 1251–1274, 1993.
- [31] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.
- [32] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proceedings of the 42nd IEEE Conference on Decision and Control (IEEE CDC)*, 2003, pp. 3621–3626.
- [33] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.
- [34] W. B. Dunbar, "Distributed receding horizon control of vehicle platoons: stability and string stability," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, 2012.

- [35] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodoriguez, "Model predictive control - a simple and powerful method to control power converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, 2008.
- [36] R. Hovorka, V. Canonico, L. J. Chassin, U. Haueter, *et al.*, "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes," *Physiological Measurement*, vol. 25, no. 4, pp. 905–920, 2004.
- [37] H. Lee, B. A. Buckingham, D. M. Wilson, and B. W. Bequette, "A closed-loop artificial pancreas using model predictive control and a sliding meal size estimator," *Journal of Diabetes Science and Technology*, vol. 3, no. 5, pp. 1082–1090, 2009.
- [38] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
- [39] L. Grüne, J. Pannek, and K. Worthmann, "A networked unconstrained nonlinear mpc scheme," in *Proceedings of European Control Conference (ECC)*, 2009, pp. 371–376.
- [40] A. Boccia, L. Grüne, and K. Worthmann, "Stability and feasibility of state constrained mpc without stabilizing terminal constraints," *System & Control Letters*, vol. 72, pp. 14–21, 2014.
- [41] G. D. Nicosia and R. R. Bitmead, "Fake riccati equations for stable receding-horizon control," in *Proceedings of European Control Conference (ECC)*, 1997, pp. 3294–3299.
- [42] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, 1993.

-
- [43] S. Yu, M. Reble, H. Chen, and F. Allgöwer, “Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control,” *Automatica*, vol. 50, no. 9, pp. 2269–2280, 2014.
- [44] L. Magni, D. M. Raimondo, and R. Scattolini, “Input-to-state stability for nonlinear model predictive control,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 4836–4841.
- [45] D. L. Marruedo, T. Alamo, and E. F. Camacho, “Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 4619–4624.
- [46] D. Antunes and W. P. M. H. Heemels, “Rollout event-triggered control: beyond periodic control performance,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3296–3311, 2014.
- [47] V. S. Dolk, D. P. Borgers, and W. P. M. H. Heemels, “Output-based and decentralized dynamic event-triggered control with guaranteed \mathcal{L}_p -gain performance and zeno-freeness,” *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 34–49, 2017.
- [48] M. C. F. Donkers, P. Tabuada, and W. P. M. H. Heemels, “Minimum attention control for linear systems,” *Discrete Event Dynamic Systems*, vol. 24, no. 2, pp. 199–218, 2014.
- [49] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [50] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *Proceedings of Hybrid Systems: Computation and Control (HSCC)*, 2005, pp. 291–305.

- [51] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, pp. 4042–4048.
- [52] H. K. Khalil, *Nonlinear Systems*, 3rd. Prentice Hall, 2001.
- [53] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.
- [54] P. Varutti and R. Findeisen, "Compensating network delays and information loss by predictive control methods," in *Proceedings of European Control Conference (ECC)*, 2009, pp. 1722–1727.
- [55] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 428–433, 1994.
- [56] M. Hovd, S. Oлару, and G. Bitsoris, "Low complexity constraint control using contractive sets," in *Proceedings of the 19th IFAC World Congress*, 2014, pp. 2933–2938.
- [57] S. Munir, M. Hovd, G. Sandou, and S. Oлару, "Controlled contractive sets for low-complexity constrained control," in *Proceedings of 2016 IEEE Conference on Computer Aided Control System Design (Part of 2016 IEEE Multi-Conference on Systems and Control)*, 2016, pp. 856–861.
- [58] M. S. Darup and M. Cannon, "On the computation of lambda-contractive sets for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1498–1504, 2017.
- [59] S. Liu, J. Zhang, Y. Feng, *et al.*, "Distributed model predictive control with asynchronous controller evaluations," *The Canadian Journal of Chemical Engineering*, vol. 91, no. 10, pp. 1609–1620, 2013.

- [60] H. Li and Y. Shi, "Event-triggered robust model predictive control of continuous-time nonlinear systems," *Automatica*, vol. 50, no. 5, pp. 1507–1513, 2014.
- [61] A. Eqtami, S. Heshmati-Alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Self-triggered model predictive control for nonholonomic systems," in *Proceedings of European Control Conference (ECC)*, 2013, pp. 638–643.
- [62] P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen, "Event-based model predictive control for networked control systems," in *Proceedings of Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, pp. 567–572.
- [63] L. Magni and R. Scattolini, "Model predictive control of continuous-time nonlinear systems with piecewise constant control," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 900–906, 2004.
- [64] R. Findeisen, L. Imsland, F. Allgöwer, *et al.*, "State and output feedback nonlinear model predictive control: an overview," *European Journal of Control*, vol. 9, no. 9, pp. 190–206, 2003.
- [65] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Distributed aperiodic model predictive control for multi-agent systems," *IET Control Theory and Applications*, vol. 9, no. 1, pp. 11–20, 2015.
- [66] P. Mhaskar, N. H. El-Farra, and P. D. Christofides, "Stabilization of nonlinear systems with state and control constraints using lyapunov based predictive control," *System & Control Letters*, vol. 55, no. 8, pp. 650–659, 2006.
- [67] P. Falugi, E. Kerrigan, and E. van Wyk, "Imperial College London Optimal Control Software User Guide (ICLOCS)," *Department of Electrical Engineering, Imperial College London, London, UK*, 2010.

- [68] Y. Zhu and U. Ozguner, "Robustness analysis on constrained model predictive control for nonholonomic vehicle regulation," in *Proceedings of American Control Conference (ACC)*, 2009, pp. 3896–3901.
- [69] W. Langson, I. Chrysoschoos, S. V. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.
- [70] M. S. Darup and M. Cannon, "A missing link between nonlinear mpc schemes with guaranteed stability," in *Proceedings of the 54th IEEE Conference on Decision and Control*, 2015, pp. 4977–4983.
- [71] D. L. Marruedo, T. Alamo, and E. F. Camacho, "Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 4619–4624.
- [72] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.
- [73] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *Process Systems Engineering*, vol. 45, no. 8, pp. 1743–1752, 1999.

List of Achievements

Journals

- [1] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Aperiodic Sampled-Data Control via Explicit Transmission Mapping: A Set Invariance Approach," *IEEE Transactions on Automatic Control* (to appear).
- [2] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Event-triggered intermittent sampling for nonlinear model predictive control," *Automatica*, Vol. 81, pp. 148-155, 2017.
- [3] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered Model Predictive Control for Nonlinear Input-Affine Dynamical Systems via Adaptive Control Samples Selection," *IEEE Transactions on Automatic Control*, Vol. 62, No. 1, pp. 177-189, 2017.
- [4] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Distributed aperiodic model predictive control for multi-agent systems," *IET Control Theory and Applications*, Vol. 9, No. 1, pp. 10-20, 2015.

International Conferences

- [1] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Robust safety controller synthesis using tubes," in *Proceedings of the 56th IEEE Conference on Decision and Control (IEEE CDC)*, Melbourne, Australia, 2017 (to appear).

- [2] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "A collision-free communication scheduling for nonlinear model predictive control," in Proceedings of the 20th IFAC World Congress (IFAC WC), Toulouse, France, 2017, pp. 8939-8944.
- [3] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered control for constrained systems : a contractive set-based approach," in Proceedings of American Control Conference (ACC), Seattle, USA, 2017, pp. 1011-1016.
- [4] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered model predictive control for continuous-time systems: A multiple discretizations approach," in Proceedings of the 55th IEEE Conference on Decision and Control (IEEE CDC), Las Vegas, USA, 2016, pp. 3078-3083.
- [5] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Time-constrained event-triggered model predictive control for nonlinear continuous-time systems," in Proceedings of the 54th IEEE Conference on Decision and Control (IEEE CDC), Osaka, Japan, 2015, pp. 4326-4331.
- [6] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered nonlinear model predictive control for networked control systems," in Proceedings of American Control Conference (ACC), Chicago, USA, 2015, pp. 4239-4244.
- [7] K. Hashimoto and S. Adachi, "Self-triggered Model Predictive Control for linear networked control systems," SICE International Symposium on Control Systems (SICE ISCS), Tokyo, March, 2015.
- [8] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Distributed event-based model predictive control for multi-agent systems under disturbances," in Proceedings of International conference on Network Games, Control, and Optimization (NetGCooP), Trento, Italy (2014)