Resource-aware control and estimation : an optimization-based approach

Resource-aware control and estimation: An optimization-based approach

Tom Gommans

This research is supported by the Dutch Technology Foundation (STW), which is part of the Netherlands Organisation for Scientific Research (NWO) under the VICI grant "Wireless control systems: A new frontier in automation" (no. 11382).

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3986-4

Typeset by the author using the pdf LaTeX documentation system.
Cover design: Tom Gommans.
Reproduction: CPI Koninklijke Wöhrmann, Zutphen, The Netherlands.

Resource-aware control and estimation: An optimization-based approach

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. F.P.T. Baaijens, voor een
commissie aangewezen door het College voor
Promoties, in het openbaar te verdedigen
op maandag 18 januari 2016 om 16.00 uur

door

Thomas Martinus Petrus Gommans

geboren te Helmond

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. L.P.H. de Goey
promotor:	prof.dr.ir. W.P.M.H. Heemels
copromotor:	dr. D.J. Guerreiro Tomé Antunes
leden:	prof.dr.ir. A.C.P.M. Backx
	prof.dr.-ing. F. Allgöwer (University of Stuttgart)
	prof.dr.ir. B. De Schutter (Technische Universiteit Delft)
	prof.dr. C. De Persis (Rijksuniversiteit Groningen)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsboefening.

Contents

1 Introduction **1**
1.1 Motivation 2
1.2 Resource-aware control approaches 3
1.2.1 Networked control systems 4
1.2.2 Optimization-based control 6
1.3 Objectives, contributions and outline 7
1.4 Publications 9

2 Self-triggered linear quadratic control **11**
2.1 Introduction 11
2.2 Self-triggered linear quadratic control 14
2.3 Proposed setup 15
2.4 Online implementation 19
2.5 A numerical example 22
2.5.1 Self-triggered LQR control 24
2.5.2 Self-triggered linear quadratic control with discounted cost 26
2.6 Conclusions 28

3 Resource-aware MPC for constrained nonlinear systems: A self-triggered control approach **31**
3.1 Introduction 31
3.2 Problem formulation 35
3.2.1 Standard stabilizing MPC setup 36
3.2.2 Resource-aware MPC with guaranteed performance 37
3.3 Self-triggered strategy 38
3.3.1 Approach and algorithm for self-triggered MPC 38
3.3.2 Variants of self-triggered strategy 40
3.3.3 Theoretical properties 41
3.4 Implementation considerations 45

3.5 Numerical examples 47
3.5.1 Self-triggered strategy for $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$ 48
3.5.2 Self-triggered strategy for $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ 49
3.5.3 Self-triggered strategy for $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$ 51
3.6 Conclusions 52

4 Resource-aware MPC for constrained linear systems: Two rollout approaches 55
4.1 Introduction 55
4.2 Problem formulation and preliminaries 58
4.2.1 System description 59
4.2.2 Problem formulation for resource-aware control 59
4.2.3 Preliminaries 60
4.3 Problem (A): Guarantees on control performance 61
4.3.1 A standard stabilizing MPC setup 62
4.3.2 Approach 64
4.3.3 Theoretical properties 65
4.4 Problem (B): Guarantees on resource utilization 68
4.4.1 Periodic solution to Problem (B): Multi-rate MPC 68
4.4.2 Approach 70
4.4.3 Theoretical properties 71
4.5 Sporadically changing input profiles 74
4.6 Numerical examples 75
4.6.1 Rollout approach variant (A) 75
4.6.2 Rollout approach variant (B) 78
4.7 Discussion 80
4.8 Conclusions 81

5 Resource-aware set-valued estimation for discrete-time systems 83
5.1 Introduction 83
5.2 Problem formulation 86
5.3 Set-valued estimator 87
5.3.1 Optimization problem for set-valued estimator 87
5.3.2 A priori bounds on the estimation error 89
5.4 Self-triggered set-valued estimation 94
5.5 Numerical examples 96
5.5.1 Single-output system 96
5.5.2 Multiple-output system 97
5.6 Discussion on computational complexity and implementation . . 98
5.7 Conclusions and outlook 99

6 Conclusions, recommendations and final thoughts 101
6.1 Conclusions 101
6.2 Recommendations for future work 103
6.3 Final thoughts 105

Bibliography 107

Summary 115

Acknowledgements 117

Curriculum vitae 119

Chapter 1

Introduction

In most control applications, controllers are implemented in a time-triggered fashion, in which the control tasks are executed periodically. In fact, in standard control textbooks, e.g., [14,36], the periodic time-triggered paradigm is presented as the only way to implement feedback control on digital platforms. However, updating the control signal periodically incurs periodic utilization of the system's sensors, controllers, actuators and communication infrastructure. Therefore, this design choice often leads to over-utilization of the available resources, as it might not be necessary to execute the control task every period to guarantee the desired closed-loop performance. In control applications with limited resources one might want to reconsider this classical paradigm and search for aperiodic control update schemes requiring less resources while guaranteeing a certain level of closed-loop performance. In the present thesis, this search is tackled within the scope of optimal control, where optimality is now defined not only in terms of control performance but also in terms of implementation cost, i.e., the cost of using resources to update control values. It is expected that the solutions to these multi-objective problems result in control strategies that abandon the periodic time-triggered control paradigm.

Aperiodic control strategies allow the inter-execution times of control tasks to vary in time and offer potential advantages with respect to periodic control when handling these resource limitations, but they also introduce many new interesting theoretical and practical challenges. Before elaborating on the specific objectives and contributions of this thesis, the next two sections discuss control areas where resource limitations play a crucial role, and provide an introduction to resource-aware control strategies.

Fig. 1.1. Example of a NCS: Cooperative and automated driving cars driving in a platoon (Courtesy of the Dutch Institute for Applied Scientific Research (TNO)).

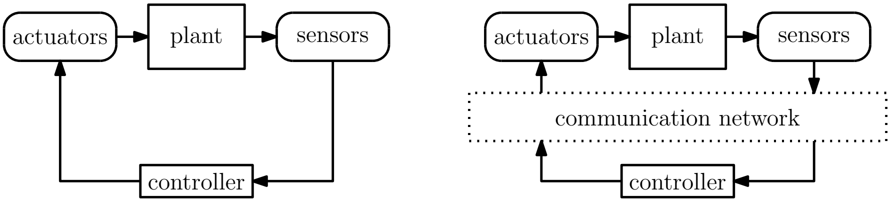

Fig. 1.2. Schematic representation of a classical control scheme (left) and a NCS (right).

1.1 Motivation

A prime example of applications where control resources, such as communication bandwidth or energy supplies, are scarce are networked control systems (NCSs). NCSs are feedback control systems, in which the communication between spatially distributed components, such as sensors, actuators and controllers, occurs through a shared communication network. An example of a NCS is shown in Figure 1.1, where cars communicate over a wireless network to perform cooperative and automated driving. Over the last decade, the study of control systems in which communication takes place via a shared network has been receiving more and more attention, see, e.g., the overview papers [54, 100] and the recent book [17]. The reason for this interest is that the use of networks offers many advantages for control systems, such as low installation and maintenance costs, reduced system wiring (in the case of wireless networks) and increased flexibility of the system. Figure 1.2 shows a schematic comparison between a classical control scheme on the left, and a networked control system on the right. In a classical control scheme, there are dedicated point-to-point connections between the sensor and the controller, and between the controller and the actuator. In NCSs the control system no longer uses dedicated point-to-point connections to transmit measurements and control updates. In fact, transmission of measurements and control updates now requires utilization of (possibly shared) limited bandwidth communication resources. In NCSs with spatially distributed wireless sensors, additional resource limitations stem from these sensors often being battery-powered. Typically, the radio chip, used to communicate sensor data, is the primary source of energy consumption in the sensors [80]. Clearly, using periodic time-triggered feedback control in a NCS requires periodic utilization of the system's resources to sample and transmit measurements and compute and update the control values at the actuators. Compared to time-triggered control, aperiodic control strategies provide more flexibility in the timing for updating the control values. Indeed, this flexibility can be exploited by reducing the number of times the control law needs to be executed in NCSs, leading to a lower network resource utilization and enhanced lifetimes of battery-powered devices.

Besides NCSs, also in other areas certain types of control input profiles are preferable from a resource point of view. The use of sparse or sporadically changing actuation signals can have several benefits, such as improved fuel efficiency or a larger lifetime of actuators that are subject to wear and tear. For instance, in [44] sparse thrust actuation signals are needed to use the fuel of a spacecraft in an efficient manner. The desire to have sparse input profiles comes from the fact that the booster rockets used for propulsion require a significant amount of fuel while operational, even when no thrust is required. Operating a spacecraft using control input profiles which utilize the boosters only when needed and turn them off as often and as long as possible is beneficial for fuel consumption. In [26] sporadically changing actuation is used in the control of an autonomous underwater vehicle in order to decrease fuel consumption and increase the deployment time. Also in overactuated systems, i.e., systems with more actuators than system states, with actuators that are subject to wear and tear it can be desirable to have a smart control allocation policy that does not require each actuator to be updated continuously in order to realize a desired level of performance. One specific example includes the usage of sparse actuation signals to control the roll of an overactuated ship [37].

1.2 Resource-aware control approaches

Given the above motivations, in this thesis we are particularly interested in designing aperiodic control strategies that only require resources when the system really needs attention, e.g., to meet certain control performance levels or satisfy constraints on states or inputs. In the design of these aperiodic controllers one can use information on the (past) states or outputs of the system to determine whether or not a control update should take place, thereby bringing feedback in the utilization of the system's resources. Such controllers taking the required utilization of resources into account are called resource-aware controllers.

It is no surprise that control areas in which resource limitations play a role also provide a rich literature on resource-aware control design. For NCS, the main two approaches that abandon the periodic time-triggered paradigm in order to ameliorate the utilization of resources are event-triggered control (ETC) and self-triggered control (STC). Other resource-aware control solutions stem from fields where the controller is obtained by solving an optimization problem, such as optimal control or model predictive control. These optimization-based areas of control provide resource-aware control solutions by cleverly modifying the optimization problems to include in the objective function not only the control performance, but also the utilization of resources.

In the the next two sections, we discuss resource-aware control approaches that have appeared in the NCS literature and in the literature on optimization-based control, respectively. This is not a black and white distinction, as ETC and STC approaches are derived from optimization-based approaches and, moreover, optimization-based approaches intend to provide ETC or STC solutions. The distinction in this chapter is mainly made for convenience. However, note that the approaches in the NCS literature are mainly driven by applications and implementation, whereas the approaches in optimization-based control focus on solving a more technical problem of optimally allocating resources.

1.2.1 Networked control systems

Considering the NCS literature, two main approaches oriented towards aperiodic input signals can be distinguished, namely, event-triggered control (ETC), see, e.g., [12, 13, 29, 30, 50, 63, 78, 85, 98], and self-triggered control (STC), see, e.g., [3,7,31,68,86,93,95]. See also [24,48,49,62,71] for recent overviews. In ETC and STC, the control law consists of two elements being a feedback controller that computes the control input, and a triggering mechanism that determines when the control input has to be updated. The difference between event-triggered control and self-triggered control is that the former is reactive, while the latter is proactive. In ETC, the triggering is based on current measurements and consists of monitoring a specific condition and when it becomes true, the control task is executed. In STC, at an update time the next update time is pre-computed based on predictions using previously received data and knowledge on the plant dynamics. Figure 1.3 shows a schematic representation of the triggering mechanisms for ETC and STC. Interestingly, in self-triggered schemes, the sensors may be shut down between updates, which allows to save additional energy. This constitutes an advantage of self-triggered strategies over event-triggered strategies. On the other hand, more information is available to the controller in event-triggered setups, as the sensors not sending measurement updates still allows conclusions to be drawn about the current measurements, based on the event conditions, see, e.g., [82, 84].

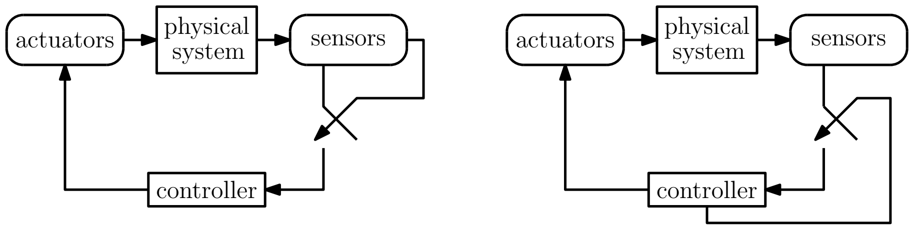

Fig. 1.3. Schematic representation of the triggering mechanism in ETC (left) and STC (right).

At present ETC and STC form popular research areas. However, two important issues have only received marginal attention: (i) the co-design of both the feedback law and the triggering mechanism, and (ii) the provision of performance guarantees (by design). To elaborate on (i), current design methods for ETC and STC are mostly emulation-based approaches, by which we mean that the feedback controller is first designed without considering the scarcity in the system's resources. The triggering mechanism is only designed in a subsequent phase, where the controller has already been fixed. Since the feedback controller is designed before the triggering mechanism, it is difficult, if not impossible, to obtain an optimal design of the combined feedback controller and triggering mechanism in the sense that the minimum number of control executions is achieved while guaranteeing closed-loop stability and a desired level of performance. Therefore such optimal design can only be pursued using a co-design approach.

Regarding (ii), only a few available ETC/STC methods provide quantitative analysis tools for control performance, such as quadratic cost, $\mathcal{L}_2$-gains or $\mathcal{H}_2$ type of criteria, and so on, see, e.g., [30,45,63,88,94,95,97,99]. For instance, in [30] one can analyze the ETC/STC loop a posteriori and evaluate what the $\mathcal{L}_\infty$-gain is, and clearly by doing this for various choices of the triggering mechanism one can (indirectly) synthesize a controller with a good closed-loop $\mathcal{L}_\infty$-gain (in balance with reasonable resource utilization) based on an iterative design process. A similar procedure can be applied for the $\mathcal{L}_2$-gain, see, e.g., [95]. Alternatively, using [63,99], one can tune the parameters of the event-triggering condition (once the controller is fixed) to obtain a desirable ultimate bound on the state. In addition, a few ETC and STC methods exist that aim at minimizing a criterion involving besides control cost also cost for utilizing resources [1,11, 27,61,72,96]. However, in most cases they do not provide guarantees in terms of standard (LQR, $\mathcal{L}_2$, $\mathcal{H}_2$) control cost (i.e., without the presence of the added cost related to resource utilization), and, in fact, due to the resulting absolute threshold in the event-triggering mechanism, these control costs are typically not finite. The case of continuous-time linear systems with a quadratic performance measure (LQR) is studied in [93, 98]. Both papers aim at arriving at ETC laws that yield the *same* cost as the optimal LQR controller, but with less resources than the continuously updating optimal LQR controller. The main idea behind the approach is to maximize the time until the next control value update, considering that the rest of the (future) controller executions will be according to standard periodic time-triggered updates. In [93], the controller design is emulation-based, whereas [98] studies a co-design method for both the feedback law and the triggering condition. However, no formal guarantees are given in these papers about the true cost realized by the ETC implementation, and the framework in [93, 98] does not offer a possibility to "trade" performance for a lower resource utilization.

1.2.2 Optimization-based control

In the context of optimal control, a rollout approach based on combined optimization over update patterns and control values was proposed in [8–10]. The rollout algorithm is a suboptimal control method for deterministic and stochastic problems that can be solved by dynamic programming [19]. In the rollout algorithm a search for optimal decisions occurs only along a (finite) lookahead horizon, assuming that from then on a "base policy" is used for which the cost-to-go is typically simple to determine. Starting from a feasible base policy, the rollout approach finds the optimal controller and cost for a set of policies that yield the same or a lower resource utilization than the base policy. By ensuring that the base policy is included in the set of policies that is considered for the optimization problem, it is shown that if the base policy produces a feasible solution, the rollout algorithm also produces a feasible solution, whose control cost is no worse than the cost corresponding to the base policy [9, 10, 19], thereby addressing (ii). Under certain assumptions, the methods in [8] outperform any periodic policy as long as this policy is used as the base policy. Regarding (i), rollout approaches are co-design approaches since they are based on a combined optimization over update patterns and control values.

In many control systems there are constraints on states and inputs that should not be violated. One of the most widely used techniques for control of constrained systems is model predictive control (MPC). MPC is a form of optimal control based on solving, at each sampling instant, a constrained finite horizon open-loop optimal control problem, using the current state of the plant as the initial state. The optimization yields an optimal control sequence and the first control value in the sequence is applied to the plant [67]. Although the classical receding horizon implementation of MPC is effective in dealing with systems with state and input constraints, typically the sampling times are chosen in a periodic time-triggered fashion, requiring the use of resources at each sampling time.

Several works already exist in the literature addressing the control of constrained systems with scarce resources. For instance in [57], a self-triggered MPC scheme is proposed which maximizes the time until the next control execution, while satisfying state and input constraints in the presence of additive disturbances. In [34, 35] a self-triggered decentralized MPC framework is presented that aims at reducing the communication between agents, as well as the number of times the agents update their control values. However, the most well known approach is based on modifying the MPC problem by appending the original MPC control cost with an ℓ_1 penalty on the input in order to obtain sparse input signals. Regularizing by the ℓ_1-norm is known to induce sparsity in the sense that, individual components of the input signal will be equal to zero, see, e.g., [6,37,38,73–76], in which the focus is on discrete-time linear systems. Also different types of sum-of-norms regularization can be used to obtain so-called group sparsity, meaning that at many time instances the entire input vector becomes zero, see, e.g., [77]. A self-triggered MPC approach for unconstrained discrete-time linear systems that avoids frequent updating of the actuators by sending only a single control value to the actuators and keeping this constant until the next execution time is presented in [53]. The strategy in [53] solves a co-design problem of simultaneously designing the control law and the triggering condition and allows trading control performance and (communication and/or actuation) resource utilization. In particular, [53] follows an ℓ_1-like regularization to solve the co-design problem by augmenting the quadratic cost function related to control performance with a penalty related to sampling the system and updating the control law.

Although ℓ_1 regularization has proven to be effective in obtaining sparse input profiles, no performance guarantees are given in terms of the original MPC cost function due to the additional penalties in the cost function.

1.3 Objectives, contributions and outline

In this thesis, the goal is to provide systematic design methodologies for optimal sampling, control and communication strategies, where optimality reflects both implementation costs (related to the required resource utilization) and control performance. In particular, the objective is to design resource-aware controllers with the following properties:

- significant reductions in resource utilization compared to time-triggered control;
- a priori closed-loop performance guarantees provided by design (in terms of the original cost function), as well as asymptotic stability and constraint satisfaction (if constraints are present);
- simultaneous design of both the feedback law and the triggering condition.

In this thesis, several resource-aware control and estimation strategies are developed. The remainder of this section highlights these contributions and provides an outline of the thesis.

In Chapter 2 we propose a self-triggered control strategy for unconstrained discrete-time linear systems subject to stochastic additive disturbances, based on performance in terms of a quadratic discounted cost. The control laws and triggering mechanisms are designed such that an a priori chosen (suboptimal) level of performance in terms of (discounted) quadratic cost is guaranteed. The approach introduces scaling parameters that can be chosen to balance the usage of resources and the degree of suboptimality compared to the optimal solution that communicates state measurements and updates the control values at every (discrete) time. The proposed methodology provides a solution to the problem of co-design of the feedback law and the triggering condition and can easily be implemented in practice as it results in a simple piecewise linear control law. This chapter is based on [40].

In Chapter 3, the ideas underlying the self-triggered control strategy in Chapter 2 are extended to a novel self-triggered MPC scheme that can deal with discrete-time systems (without disturbances) that are subject to state and input constraints. In fact, a general framework for the self-triggered MPC strategy is proposed, addressing both sporadically changing and sparse input profiles. The framework applies to discrete-time linear and nonlinear systems subject to state and input constraints and possibly non-quadratic cost functions. This chapter is based on [15, 41].

We propose two resource-aware MPC strategies for discrete-time linear systems subject to state and input constraints exploiting rollout ideas in Chapter 4. The rollout MPC approaches are based on a dynamic programming formulation of the co-design problem of both determining the feedback law and the triggering condition (see Section 1.2.2). One of the proposed rollout MPC strategies provides performance guarantees, in terms of the control cost, by design. The other proposed rollout MPC strategy provides a guaranteed (average) resource utilization, while cleverly selecting when to use the actuators in order to maximize the control performance.

The approaches presented in Chapters 2, 3 and 4 rely on the availability of the full state through measurements. In case the full state is not available it is of interest to use a resource-aware estimator that reconstructs the plant state based on measurements. In particular it is of interest to reduce the communication between the sensors and the estimator. The final contribution of the thesis is a self-triggered estimator for discrete-time linear systems subject to unknown but bounded disturbances affecting both the system states and outputs, see Chapter 5. The proposed self-triggered estimator is a set-valued estimator that employs rollout techniques to reduce the communication between the sensors and the estimator with respect to a periodic sampling strategy. Moreover, at each time instant it provides an estimate of the plant state and a guaranteed bound on the difference between the true plant state and the estimate. This information can subsequently be used in a state-based controller. This chapter is based on the paper [21].

Finally, in Chapter 6 conclusions and recommendations are presented.

1.4 Publications

The results presented in this thesis are based on the following publications.

- T.M.P. Gommans, D.J. Antunes, M.C.F. Donkers, P. Tabuada and W.P.M.H. Heemels, Self-triggered linear quadratic control, *Automatica*, 50(4), pages 1279-1287, 2014.

- J.D.J. Barradas Berglind, T.M.P. Gommans and W.P.M.H. Heemels, Self-triggered MPC for constrained linear systems and quadratic costs, *Proc. IFAC Conference on Nonlinear Model Predictive Control*, pages 342-348, 2012.

- T.M.P. Gommans and W.P.M.H. Heemels, Resource-aware MPC for constrained nonlinear systems: A self-triggered control approach, *System & Control Letters*, 79, pages 59-67, 2015.

- F.D. Brunner, T.M.P. Gommans, W.P.M.H. Heemels and F. Allgöwer, Resource-aware set-valued estimation for discrete-time linear systems, *Conference on Decision and Control*, 2015.

- T.M.P. Gommans, T.A.F. Theunisse, D.J. Antunes and W.P.M.H. Heemels, Resource-aware MPC for constrained linear systems: Two rollout approaches, *submitted*, 2015.

Other publications in the PhD period not reported in the PhD thesis.

- T.M.P. Gommans, W.P.M.H. Heemels, N.W. Bauer and N. van de Wouw, Compensation-based control for lossy communication networks, *Proc. American Control Conference*, pages 2854-2859, 2012.

- T.M.P. Gommans, W.P.M.H. Heemels, N.W. Bauer and N. van de Wouw, Compensation-based control for lossy communication networks, *International Journal of Control*, 86(10), pages 1880-1897, 2013.

- J.L.C. Verhaegh, T.M.P. Gommans and W.P.M.H. Heemels, Extension and evaluation of model-based periodic event-triggered control *Proc. European Control Conference*, 86(10), pages 1138-1144, 2013.

- F.D. Brunner, T.M.P. Gommans, W.P.M.H. Heemels and F. Allgöwer, Communication scheduling in robust self-triggered MPC for linear discrete-time system, *Workshop on Distributed Estimation and Control in Networked Systems*, 2015.

Chapter 2

Self-triggered linear quadratic control

Abstract – Self-triggered control is a recently proposed paradigm that abandons the more traditional periodic time-triggered execution of control tasks with the objective of reducing the utilization of communication resources, while still guaranteeing desirable closed-loop behavior. In this chapter, we introduce a self-triggered strategy based on performance levels described by a quadratic discounted cost. The classical LQR problem can be recovered as an important special case of the proposed self-triggered strategy. The self-triggered strategy proposed in this paper possesses three important features. Firstly, the control laws and triggering mechanisms are synthesized so that a priori chosen performance levels are guaranteed by design. Secondly, they realize significant reductions in the usage of communication resources. Thirdly, we address the co-design problem of jointly designing the feedback law and the triggering condition. By means of a numerical example, we show the effectiveness of the presented strategy. In particular, for the self-triggered LQR strategy, we show quantitatively that the proposed scheme can outperform conventional periodic time-triggered solutions.

2.1 Introduction

In many control applications, controllers are nowadays implemented using communication networks in which the control task has to share the communication resources with other tasks. Despite the fact that resources can be scarce, controllers are typically still implemented in a time-triggered fashion, in which control tasks are executed periodically. This design choice often leads to over-utilization of the available communication resources, and/or causes a limited lifetime of battery-powered devices, as it might not be necessary to execute the

This chapter is based on [40].

control task every period to guarantee desired closed-loop performance. Also in the area of 'sparse control' [37], in which it is desirable to limit the changes in certain actuator signals while still realizing specific control objectives, periodic execution of control tasks may not be optimal either. In both networked control systems with scarce communication resources and sparse control applications the *fundamental* problem of determining optimal sampling and communication strategies arises, where optimality needs to reflect both implementation cost (related to the number of communications and/or actuator changes) as well as control performance. It is expected that the solution to this problem results in control strategies that abandon the periodic time-triggered control paradigm.

Two approaches that abandon the periodic communication pattern are event-triggered control (ETC), see, e.g., [12,13,30,47,50,52,63,85,95], and self-triggered control (STC), see, e.g., [2,3,7,31,68,92,95]. Although ETC is effective in the reduction of communication or actuator movements, it was originally proposed for different reasons, including the reduction of the use of computational resources and dealing with the event-based nature of the plants to be controlled. In ETC and STC, the control law consists of two elements being a feedback controller that computes the control input, and a triggering mechanism that determines when the control input has to be updated. The difference between ETC and STC is that in the former the triggering consists of verifying a specific condition continuously and when it becomes true, the control task is triggered, while in the latter at an update time the next update time is pre-computed. ETC laws have been mostly developed for continuous-time systems, although they have also appeared for discrete-time systems, see, e.g., [27,32,45,61,72,99] and [59, Sec. 4.5]. In addition, in [12, 46, 50, 52] so-called periodic event-triggered control strategies were proposed and analyzed for continuous-time systems.

At present ETC and STC form popular research areas. However, two important issues have only received marginal attention: (i) the co-design of both the feedback law and the triggering mechanism, and (ii) the provision of performance guarantees (by design). To elaborate on (i), note that current design methods for ETC and STC are mostly emulation-based approaches, by which we mean that the feedback controller is designed without considering the scarcity in the system's resources. The triggering mechanism is only designed in a subsequent phase, where the controller has already been fixed. Since the feedback controller is designed before the triggering mechanism, it is difficult, if not impossible, to obtain an optimal design of the combined feedback controller and triggering mechanism in the sense that the minimum number of control executions is achieved while guaranteeing closed-loop stability and a desired level of performance.

Regarding (ii), only a few available ETC/STC methods provide quantitative analysis tools for control performance, such as $\mathcal{L}_2$-gains, quadratic cost, $\mathcal{H}_2$ type of criteria, and so on. For instance, in [30] one can analyze the ETC/STC loop a posteriori and evaluate what the $\mathcal{L}_\infty$-gain is, and clearly by doing this for various choices of the triggering mechanism one can (indirectly) synthesize a controller with a good closed-loop $\mathcal{L}_\infty$-gain (in balance with a reasonable communication usage) based on an iterative design process. A similar procedure can be applied for the $\mathcal{L}_2$-gain, see, e.g., [95]. Alternatively, using [63, 99], one can tune the parameters of the event-triggering condition (once the controller is fixed) to obtain a desirable ultimate bound on the state. In addition, a few ETC and STC methods exist that aim at minimizing a criterion involving besides control cost also communication cost [27, 61, 72]. However, in most cases they do not provide guarantees in terms of standard (LQR, $\mathcal{L}_2$, $\mathcal{H}_2$) control cost (i.e., without the presence of communication cost), and, in fact, due to the resulting absolute threshold in the event-triggering mechanism, these control cost are typically not finite. The case of continuous-time linear systems with a quadratic performance measure (LQR) is studied in [93, 98]. Both papers aim at arriving at ETC laws that yield the *same* cost as the optimal LQR controller, but require less communication than the continuously updating optimal LQR controller. The main idea behind the approach is to maximize the time until the next control value update, considering that the rest of the (future) controller executions will be according to standard periodic time-triggered updates. In [93], the controller design is emulation-based, whereas [98] studies a co-design method for both the feedback law and the triggering condition. However, no formal guarantees are given in these papers about the true cost realized by the ETC implementation, and the framework in [93, 98] does not offer a possibility to "trade" performance for less communication resource usage.

The main contribution of the present chapter is a novel STC strategy for discrete-time linear systems in the presence of stochastic disturbances, addressing the issues (i) and (ii) and allowing to trade guaranteed performance levels with utilization of communication resources. The contribution of this chapter is threefold:

- The methods guarantee a desired performance level based on quadratic (discounted) cost without an iterative design process. In fact, the presented strategy aims at reducing the use of communication resources, while still guaranteeing a prespecified suboptimal level of performance.
- For the considered control problem, we solve a co-design problem by jointly designing the feedback controller and the triggering mechanism.
- By means of a numerical example, we demonstrate quantitatively that aperiodic control can outperform periodic control when both control performance and communication cost are important.

Nomenclature

Let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real numbers and the set of non-negative integers (including zero), respectively. The notation $\mathbb{N}_{\geq s}$ and $\mathbb{N}_{[s,t)}$ is used to denote the sets $\{r \in \mathbb{N} \mid r \geq s\}$ and $\{r \in \mathbb{N} \mid s \leq r < t\}$, respectively, for some $s, t \in \mathbb{N}$. The inequalities $\prec, \preceq, \succ$ and $\succeq$ are used for matrices, i.e., for a square matrix $X \in \mathbb{R}^{n\times n}$ we write $X \prec 0, X \preceq 0, X \succ 0$ and $X \succeq 0$ if X is symmetric and, in addition, X is negative definite, negative semi-definite, positive definite and positive semi-definite, respectively. Sequences of vectors are indicated by bold letters, e.g., $\boldsymbol{u} = (u_0, u_1, \dots, u_M)$ with $u_i \in \mathbb{R}^{n_u}, i \in \{0, 1, \dots, M\}$, where $M \in \mathbb{N} \cup \{\infty\}$ will be clear from the context. Let X and Y be random variables. The expected value of X is denoted by $\mathbb{E}(X)$ and the conditional expectation of X given Y is denoted $\mathbb{E}[X \mid Y]$. The trace of a matrix A is denoted by $\operatorname{tr}(A)$.

2.2 Self-triggered linear quadratic control

In this section, we provide the problem formulation and the general setup for the self-triggered control strategy. We consider the control of a discrete-time LTI system given by

$$x_{t+1} = Ax_t + Bu_t + Ew_t, \tag{2.1}$$

for $t \in \mathbb{N}$, where $x_t \in \mathbb{R}^{n_x}$ is the state, $u_t \in \mathbb{R}^{n_u}$ is the input and $w_t \in \mathbb{R}^{n_w}$ is the disturbance, respectively, at discrete time $t \in \mathbb{N}$. We assume that the pair (A, B) is controllable and that $w_t,\ t \in \mathbb{N}$, are independent and identically distributed random vectors (not necessarily Gaussian distributed) with $\mathbb{E}[w_t] = 0$ and $\mathbb{E}[w_t w_t^\top] = I,\ t \in \mathbb{N}$, where $I \in \mathbb{R}^{n_w\times n_w}$ is the identity matrix. In this section, we are interested in control strategies that guarantee a certain control performance in terms of a discounted quadratic cost function

$$J = \sum_{t=0}^{\infty} \mathbb{E}\left[\alpha^t \left(x_t^\top Q x_t + 2x_t^\top S u_t + u_t^\top R u_t\right) \mid x_0\right], \tag{2.2}$$

involving the weighting matrices Q, R and S, where $\left[\begin{smallmatrix} Q & S \\ S^\top & R \end{smallmatrix}\right] \succ 0$. The discount factor $0 < \alpha \leq 1$ is assumed to be strictly less than one when $E \neq 0$ to assure that (2.2) is bounded. Note that $E = 0$ and $\alpha = 1$ allow us to consider an LQR-like framework. If we assume that the state is available at every $t \in \mathbb{N}$ and also that the control input can be updated at every $t \in \mathbb{N}$, it is well known (see, e.g., [19, Sec. 3.2]) that the optimal cost for a given initial state x_0 is given by

$$V(x_0) := x_0^\top P x_0 + \frac{\alpha}{1-\alpha}\operatorname{tr}(PEE^\top), \tag{2.3}$$

where P is the unique positive semi-definite solution to the discrete algebraic Riccati equation (DARE)

$$P = Q + \alpha A^\top P A - \left(\alpha A^\top P B + S\right)\left(R + \alpha B^\top P B\right)^{-1}\left(\alpha B^\top P A + S^\top\right) \tag{2.4}$$

and that the optimal feedback policy is described by

$$u_t^* = K^* x_t, \tag{2.5}$$

$$K^* = -\left(R + \alpha B^\top P B\right)^{-1} \left(\alpha B^\top P A + S^\top\right). \tag{2.6}$$

The control law given in (2.5) requires the transmission of measured states and updates of control actions at each sample instant $t \in \mathbb{N}$, which might not be necessary to guarantee a certain (suboptimal) performance. In this chapter, we are interested in *synthesizing* control laws that require (much) less communication between sensors, controller, and actuators (and/or less actuator movements [37]), while still providing guarantees on the quadratic performance criterion in (2.2). More specifically, we are interested in reducing the number of times the input is updated (which directly influences the number of required transmissions from sensors to controllers and controllers to actuators), while still satisfying the following performance guarantee

$$\sum_{t=0}^{\infty} \mathbb{E}\left[\alpha^t \left(x_t^\top Q x_t + 2x_t^\top S u_t + u_t^\top R u_t\right) \mid x_0\right] \leq V_{\beta_1,\beta_2}(x_0), \tag{2.7}$$

where, for $x \in \mathbb{R}^{n_x}$,

$$V_{\beta_1,\beta_2}(x) := \beta_1 x^\top P x + \beta_2 \frac{\alpha}{1-\alpha} \operatorname{tr}(PEE^\top) \tag{2.8}$$

and $\beta_1, \beta_2 \geq 1$ are scaling factors of the state-dependent and constant parts, respectively, in the optimal costs in (3). Note that $\beta_1 = \beta_2 = 1$ corresponds to requiring the same costs as the optimal time-triggered control law given by (2.5)-(2.6). The scaling factors β_1, β_2 can be chosen to balance the usage of communication resources and the degree of suboptimality. In particular, the choice $\beta_1 = \beta_2 = \beta \geq 1$ specifies a degree of suboptimality that corresponds to a percentage of the periodic control performance (2.3). This is automatically the case when $E = 0$, since in that case β_2 plays no role.

To address this problem, we propose a self-triggered strategy that aims at reducing the number of input updates. The proposed strategy solves the co-design problem of simultaneously synthesizing the next transmission time and the next corresponding control value and is based on holding the control value constant for as many steps as possible, while still guaranteeing the performance guarantee (2.7) in the end.

2.3 Proposed setup

The self-triggered strategy is based on holding the current input value as long as possible while still guaranteeing (2.7) given $\beta_1, \beta_2 \geq 1$. In fact, the control strategy will have the structure

$$\begin{cases} t_{l+1} &= t_l + M(x_{t_l}), \\ u_t &= \bar{u}_l \in \mathcal{U}(x_{t_l}), \quad t \in \mathbb{N}_{[t_l, t_{l+1})} \end{cases} \tag{2.9}$$

with $t_0 := 0$, $M : \mathbb{R}^{n_x} \to \{1, \dots, \bar{M}\}$, $\bar{M} \in \mathbb{N}$, and $\mathcal{U} : \mathbb{R}^{n_x} \rightrightarrows \mathbb{R}^{n_u}$. Hence, $\mathcal{U}$ is a set-valued map. Here, $M(x)$ denotes the time between two transmissions and $\mathcal{U}(x)$ denotes the set of possible control values when being in state x. The integer $\bar{M}$ is a predefined upper bound on the inter-transmission times, which can be taken arbitrarily large. We are interested in solving the co-design problem of both the next transmission time (through M) and the chosen control value (through $\mathcal{U}$).

Instrumental in the co-design of the mappings M and $\mathcal{U}$ will be the inequality

$$\mathbb{E}\Big[\Big(\sum_{t=t_l}^{t_{l+1}-1} \alpha^{t-t_l} \left(x_t^\top Q x_t + 2 x_t^\top S \bar{u}_l + \bar{u}_l^\top R \bar{u}_l \right) \Big) \\ + \alpha^{t_{l+1}-t_l} V_{\beta_1,\beta_2}(x_{t_{l+1}}) \mid x_{t_l} \Big] \le V_{\beta_1,\beta_2}(x_{t_l}) \tag{2.10}$$

at transmission time t_l, $l \in \mathbb{N}$. Summing (2.10) over all events $l \in \mathbb{N}$ will give the performance guarantee (2.7) as we will show in Theorem 2.1. At transmission time t_l with state x_{t_l} we aim at finding a maximal value for t_{l+1} such that there is a $\bar{u}_l$ satisfying (2.10). This results in $M(x_{t_l}) = t_{l+1} - t_l$.

To introduce the mappings $\mathcal{U}$ and M in (2.9) formally, we define for $x \in \mathbb{R}^{n_x}$, $\mathcal{U}_M(x)$ as the set of control values that can be held constant for M steps, while still satisfying (2.10) when in state x at time t_l, i.e., after a shift in time, this leads to

$$\mathcal{U}_M(x) := \Big\{ \bar{u} \in \mathbb{R}^{n_u} \;\Big|\; \mathbb{E}\Big[\Big(\sum_{j=0}^{M-1} \alpha^j (\bar{x}_j^\top Q \bar{x}_j + 2\bar{x}_j^\top S \bar{u} + \bar{u}^\top R \bar{u}) \Big) \\ + \alpha^M V_{\beta_1,\beta_2}(\bar{x}_M) \mid x \Big] \le V_{\beta_1,\beta_2}(x) \Big\}, \tag{2.11}$$

where $\bar{x}_j$, $j \in \{1, 2, \dots, M\}$, is the solution to (2.1) with $\bar{x}_0 = x$ and $u_t = \bar{u}$, $t \in \mathbb{N}$, i.e.,

$$\bar{x}_j = \bar{A}_j x + \bar{B}_j \bar{u} + \bar{E}_j^M \mathbf{w}_M, \qquad \bar{x}_0 = x \tag{2.12}$$

where for $j \in \{1, 2, \dots, M\}$, $\bar{A}_j := A^j$, $\bar{B}_j := \sum_{i=0}^{j-1} A^i B$ and $\bar{E}_j^M \in \mathbb{R}^{n_x \times M n_w}$ is given by

$$\bar{E}_j^M := \begin{bmatrix} A^{j-1}E & \dots & AE & E & 0 & \dots & 0 \end{bmatrix}, \tag{2.13}$$

where $\mathbf{w}_M := [w_0^\top, w_1^\top, \dots, w_{M-1}^\top]^\top$. We now define in (2.9), for $x \in \mathbb{R}^{n_x}$ and $\bar{M} \in \mathbb{N}$,

$$M(x) := \max \left\{ M \in \{1, 2, \dots, \bar{M}\} \mid \mathcal{U}_M(x) \neq \emptyset \right\}, \tag{2.14a}$$

$$\mathcal{U}(x) := \mathcal{U}_{M(x)}(x). \tag{2.14b}$$

The control law is now given by (2.9) and (2.14).

Remark 2.1. Note that this STC strategy is of a "greedy" nature, as at time t_l the next transmission time $t_{l+1} = t_l + M(x_{t_l})$, $l \in \mathbb{N}$, is maximized while guaranteeing (2.10) without taking into account the influence of this choice on the required number of future transmissions after t_{l+1}. ◁

Remark 2.2. In case multiple control commands can be sent in one control packet, the STC approach can be extended towards packet-based control in which a sequence of control values is transmitted to the actuators at times t_l, $l \in \mathbb{N}$. More specifically, for $t \in \mathbb{N}_{[t_l,t_{l+1})}$, the control strategy will have the structure

$$\begin{cases} t_{l+1} &= t_l + M^{\mathrm{pb}}(x_{t_l}), \\ u_t &= u^{\mathrm{pb},l}_{t-t_l} \quad \text{with} \quad (u^{\mathrm{pb},l}_0, u^{\mathrm{pb},l}_1, \dots, u^{\mathrm{pb},l}_{M-1}) \in \mathcal{U}^{\mathrm{pb}}_M(x_{t_l}), \end{cases}$$

where for $x \in \mathbb{R}^{n_x}$ we define

$$\begin{aligned} \mathcal{U}^{\mathrm{pb}}_M(x) := \Bigg\{ [u^{\mathrm{pb}}_0, u^{\mathrm{pb}}_1, \dots, u^{\mathrm{pb}}_{M-1}] \in \mathbb{R}^{n_u} \Bigg| \\ \mathbb{E}\Big[\Big(\sum_{j=0}^{M-1} \alpha^j (\tilde{x}_j^\top Q \tilde{x}_j + 2\tilde{x}_j^\top S u^{\mathrm{pb}}_j + (u^{\mathrm{pb}}_j)^\top R u^{\mathrm{pb}}_j)\Big) \\ + \alpha^M V_{\beta_1,\beta_2}(\tilde{x}_M) \mid x \Big] \leq V_{\beta_1,\beta_2}(x) \Bigg\}, \end{aligned}$$

where $\tilde{x}_j$, $j \in \{1, 2, \dots, M\}$, is the solution to (2.1) with $\tilde{x}_0 = x$ and input sequence $(u^{\mathrm{pb}}_0, u^{\mathrm{pb}}_1, \dots, u^{\mathrm{pb}}_{M-1}) \in \mathcal{U}^{\mathrm{pb}}_M(x)$. Moreover,

$$M^{\mathrm{pb}}(x) := \max \Big\{ M \in \{1, 2, \dots, \bar{M}^{\mathrm{pb}}\} \mid \mathcal{U}^{\mathrm{pb}}_M(x) \neq \emptyset \Big\},$$

where $\bar{M}^{\mathrm{pb}} \geq 2$ denotes the number of control values that can be transmitted in one packet. At time t_l, $l \in \mathbb{N}$, the state is measured and the sequence $(u^{\mathrm{pb},l}_0, u^{\mathrm{pb},l}_1, \dots, u^{\mathrm{pb},l}_{M-1}) \in \mathcal{U}^{\mathrm{pb}}(x_{t_l})$ is sent to the actuators that implement the received input sequence one-by-one. The sensors can go in standby from time $t_l + 1$ until $t_{l+1} - 1$. At time t_{l+1} the state is measured by the sensors, and the procedure is repeated. ◁

Theorem 2.1. *For fixed $\beta_1, \beta_2 \geq 1$, the control law* (2.9) *with* (2.14) *is well defined, i.e., for all $x_0 \in \mathbb{R}^{n_x}$ and all disturbance realizations $\mathbf{w} = (w_0, w_1, \dots)$, $\mathcal{U}_M(x) \neq \emptyset$ for some $M \in \{1, \dots, \bar{M}\}$. Moreover, the closed-loop system given by* (2.1), (2.9) *and* (2.14) *satisfies the performance guarantee* (2.7).

Proof. To prove well-definedness of the control law (2.9) with (2.14), for all $x \in \mathbb{R}^{n_x}$ we will show that

$$x^\top Q x + 2x^\top S \bar{u} + \bar{u}^\top R \bar{u} + \alpha \mathbb{E}[V_{\beta_1,\beta_2}(\bar{x}_1) \mid x] \leq V_{\beta_1,\beta_2}(x) \tag{2.15}$$

holds for some $\bar{u} \in \mathbb{R}^{n_u}$ showing that $\mathcal{U}_1(x) \neq \emptyset$ as (2.15) is the condition in (2.11) for $M = 1$. Suppose we are at transmission time t_l for some $l \in \mathbb{N}$ and $x_{t_l} = x$. If $\bar{u}$ is chosen as the optimal control value $K^* x$ taken from (2.5)-(2.6), then we have

$$x^\top Q x + 2x^\top S\bar{u} + \bar{u}^\top R\bar{u} + \alpha\mathbb{E}[V(\bar{x}_1) \mid x\,] = V(x),$$

Indeed, if $\bar{u} = K^* x$, then using (2.3), we have

$$\begin{aligned} x^\top Q x &+ 2x^\top S\bar{u} + \bar{u}^\top R\bar{u} + \alpha\mathbb{E}[V(\bar{x}_1) \mid x\,] \\ &= x^\top \Big(Q + 2SK^* + (K^*)^\top RK^* + \alpha(A + BK^*)^\top P(A + BK^*)\Big)x \\ &\qquad + \alpha\ \mathrm{tr}(PEE^\top) + \frac{\alpha^2}{1-\alpha}\mathrm{tr}(PEE^\top) \\ &= x^\top P x + \frac{\alpha}{1-\alpha}\mathrm{tr}(PEE^\top) = V(x), \end{aligned}$$

where in the second equality we used (2.4) and (2.6). Thus $\bar{u} = K^* x$ satisfies (2.15) for $\beta_1, \beta_2 \geq 1$. Hence, $K^* x \in \mathcal{U}_1(x)$. This shows that $\mathcal{U}(x) \neq \emptyset$ and $M(x) \geq 1$ for all $x \in \mathbb{R}^{n_x}$.

We will now prove that the control law (2.9) with (2.14) satisfies the performance guarantee (2.7). For $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$, we define $g(x, u) := x^\top Q x + 2x^\top S u + u^\top R u$. We start by fixing a given $L \in \mathbb{N}$, and notice that

$$\begin{aligned} \mathbb{E}\Big[\sum_{t=0}^{t_{L+1}-1} \alpha^t g(x_t, u_t) \mid x_0\Big] &= \mathbb{E}\Big[\sum_{l=0}^{L} \alpha^{t_l} \sum_{t=t_l}^{t_{l+1}-1} \alpha^{t-t_l} g(x_t, u_t) \mid x_0\Big] \\ &= \mathbb{E}\Big[\sum_{l=0}^{L} \alpha^{t_l} \mathbb{E}\big[\sum_{t=t_l}^{t_{l+1}-1} \alpha^{t-t_l} g(x_t, u_t) \mid x_{t_l}\big] \mid x_0\Big], \end{aligned} \tag{2.16}$$

where we used the fact that for each $l \in \mathbb{N}$

$$\mathbb{E}\Big[\alpha^{t_l} \sum_{t=t_l}^{t_{l+1}-1} \mathbb{E}[\alpha^{t-t_l} g(x_t, u_t) \mid x_{t_l}] \mid x_0\Big] = \mathbb{E}\Big[\alpha^{t_l} \sum_{t=t_l}^{t_{l+1}-1} \alpha^{t-t_l} g(x_t, u_t) \mid x_0\Big]. \tag{2.17}$$

Equation (2.17) follows from standard properties of conditional expectations [28, p.16] and the fact that the underlying stochastic process defined by (2.1) and (2.9), being a discrete-time Markov process, is also a strong Markov process [70, p.72] (then the Markov property holds at the stopping times t_l, $l \in \mathbb{N}$).

Using (2.10) in the last equation of (2.16) we obtain

$$
\begin{aligned}
&\mathbb{E}\Big[\sum_{t=0}^{t_{L+1}-1} \alpha^t g(x_t,u_t) \mid x_0\Big] \\
&\quad \le \mathbb{E}\Big[\sum_{l=0}^{L} \alpha^{t_l}\big(V_{\beta_1,\beta_2}(x_{t_l}) - \alpha^{t_{l+1}-t_l}\mathbb{E}[V_{\beta_1,\beta_2}(x_{t_{l+1}}) \mid x_{t_l}]\big) \mid x_0\Big] \\
&\quad = \mathbb{E}\Big[\sum_{l=0}^{L} \alpha^{t_l} V_{\beta_1,\beta_2}(x_{t_l}) - \alpha^{t_{l+1}} V_{\beta_1,\beta_2}(x_{t_{l+1}}) \mid x_0\Big] \\
&\quad = V_{\beta_1,\beta_2}(x_0) - \mathbb{E}[\alpha^{t_{L+1}} V_{\beta_1,\beta_2}(x_{t_{L+1}}) \mid x_0] \\
&\quad \le V_{\beta_1,\beta_2}(x_0),
\end{aligned}
\tag{2.18}
$$

where in the first equality we again used standard properties of conditional expectations and the strong Markov property and in the last inequality we used the fact that V_{β_1,β_2} takes only nonnegative values.

Since $L \le t_L \le \bar{M}L$, from (2.18) we have that $V_{\beta_1,\beta_2}(x_0)$ is also an upper bound on $\mathbb{E}[\sum_{t=0}^{L} \alpha^t g(x_t,u_t) \mid x_0]$, and, hence, we can interchange the expectation and (finite) summation operations. Taking the limit as $L \to \infty$ we obtain

$$\sum_{t=0}^{\infty} \mathbb{E}[\alpha^t g(x_t,u_t) \mid x_0] \le V_{\beta_1,\beta_2}(x_0).$$

This completes the proof. □

The above theorem shows that the required control performance in terms of the cost (2.2) is guaranteed by the proposed self-triggered control law.

For sparse control applications [37], the savings in updates of actuator values are immediately clear from the chosen setup. In case the interest is in reducing the number of communications between sensors, controller and actuators, one has to distinguish two cases. For the case of sensors co-located with the controller, the next update time can be computed at or closely to the sensors. For distributed sensors the controller can compute and broadcast the next update time. Both these implementations result in communication from sensors to controllers and controllers to actuators only at the transmission times $t_l,\ l \in \mathbb{N}$.

2.4 Online implementation

In this section, we discuss the on-line implementation of the proposed self-triggered strategy. We start by showing how to test if, for a fixed value of M, $\mathcal{U}_M(x) \neq \emptyset$, for $x \in \mathbb{R}^{n_x}$, which is needed to evaluate (2.14). Clearly, $\mathcal{U}_M(x) \neq \emptyset$

if and only if

$$\min_{\bar{u}\in\mathbb{R}^{n_u}} \mathbb{E}\Big[\Big(\sum_{j=0}^{M-1} \alpha^j \left(\bar{x}_j^\top Q\bar{x}_j + 2\bar{x}_j^\top S\bar{u} + \bar{u}^\top R\bar{u}\right)\Big) + \alpha^M V_{\beta_1,\beta_2}(\bar{x}_M) \mid x\Big] \leq V_{\beta_1,\beta_2}(x). \quad (2.19)$$

By using (2.8) and (2.12), we see that (2.19) is equivalent to

$$\min_{\bar{u}\in\mathbb{R}^{n_u}} x^\top F_M x + x^\top G_M \bar{u} + \frac{1}{2}\bar{u}^\top H_M \bar{u} + c_M \leq V_{\beta_1,\beta_2}(x),$$

where

$$\begin{aligned}
F_M &= \alpha^M \beta_1 \bar{A}_M^\top P \bar{A}_M + \sum_{j=0}^{M-1} \alpha^j \bar{A}_j^\top Q \bar{A}_j,\\
G_M &= 2\Big[\alpha^M \beta_1 \bar{A}_M^\top P \bar{B}_M + \sum_{j=0}^{M-1} \alpha^j \left(\bar{A}_j^\top Q \bar{B}_j + \bar{A}_j^\top S\right)\Big],\\
H_M &= 2\Big[\alpha^M \beta_1 \bar{B}_M^\top P \bar{B}_M + \sum_{j=0}^{M-1} \alpha^j \left(\bar{B}_j^\top Q \bar{B}_j + \bar{B}_j^\top S + S^\top \bar{B}_j + R\right)\Big],\\
c_M &= d_M + \beta_2 \alpha^M \frac{\alpha}{1-\alpha} \operatorname{tr}(PEE^\top),\\
d_M &= \alpha^M \beta_1 \operatorname{tr}\left(P\bar{E}_M^M (\bar{E}_M^M)^\top\right) + \sum_{j=1}^{M-1} \alpha^j \operatorname{tr}\left(Q\bar{E}_j^M (\bar{E}_j^M)^\top\right).
\end{aligned} \quad (2.20)$$

Note that $H_M \succ 0$ since $P \succeq 0$, $\alpha > 0$ and $\left[\begin{smallmatrix} Q & S \\ S^\top & R \end{smallmatrix}\right] \succ 0$. To find

$$\bar{u}^* := \operatorname*{argmin}_{\bar{u}\in\mathcal{U}_M(x)} x^\top F_M x + x^\top G_M \bar{u} + \frac{1}{2}\bar{u}^\top H_M \bar{u} + c_M,$$

we solve

$$\frac{\partial}{\partial \bar{u}}\left(x^\top F_M x + x^\top G_M \bar{u} + \frac{1}{2}\bar{u}^\top H_M \bar{u} + c_M\right) = 0,$$

which leads to $x^\top G_M + \bar{u}^\top H_M = 0$, and thus

$$\bar{u}^* = -H_M^{-1} G_M^\top x =: K_M x. \quad (2.21)$$

The corresponding optimal cost in the left-hand side of (2.19) is $x^\top P_M^* x + c_M$ with $P_M^* = F_M - \frac{1}{2} G_M H_M^{-1} G_M^\top$. Hence, (2.19) holds, i.e., $\mathcal{U}_M(x) \neq \emptyset$, if and only if

$$x^\top P_M^* x + \bar{c}_M \leq \beta_1 x^\top P x, \quad (2.22)$$

where

$$\bar{c}_M := d_M - \beta_2 \sum_{j=1}^{M} \alpha^j \operatorname{tr}(PEE^\top). \tag{2.23}$$

We now see that (2.14a) can be rewritten as

$$M(x) = \max\big\{M \in \{1,2,\ldots,\bar{M}\} \mid x^\top P_M^* x + \bar{c}_M \le \beta_1 x^\top P x\big\}, \tag{2.24a}$$

and the control input (2.14b) is given by

$$K_{M(x)}x \in \mathcal{U}_{M(x)}(x) \tag{2.24b}$$

The proposed control strategy, taking the form (2.9), can be simply implemented as

$$\begin{cases} t_{l+1} = t_l + M(x_{t_l}), \\ \quad u_t = K_{M(x_{t_l})} x_{t_l}, \quad t \in \mathbb{N}_{[t_l, t_{l+1})}, \end{cases} \tag{2.25}$$

where the function $M : \mathbb{R}^{n_x} \to \mathbb{N}$ is given by (2.24a).

Remark 2.3. (Deterministic case, i.e., $E = 0$) When no disturbances act on (2.1) ($E = 0$), then from (2.20) and (2.23) we obtain that $\bar{c}_M = 0$ for all M. We can conclude from (2.1) and (2.25) that any initial condition along the ray λx, $\lambda > 0$, will lead to the same triggering sequence. ◁

Remark 2.4. (Effect of disturbances, i.e., $E \neq 0$) The effect of disturbances on (2.22) is captured by $\bar{c}_M$, which is a linear function of the disturbance covariance matrix given by $EE^\top$ and can be influenced by $\beta_2 \ge 1$. In fact, from (2.20) and (2.23) it is easy to see that $\bar{c}_M$, $M \in \{1,2,\ldots,\bar{M}\}$, can be made smaller by increasing $\beta_2 \ge 1$ (for fixed β_1). Note that P_M^*, $M \in \{1,2,\ldots,\bar{M}\}$, is independent of β_2 and only depends on β_1. Hence, by increasing β_2 (with same β_1) $M(x)$ will generally become larger.

In general, $\bar{c}_M \neq 0$, $M \in \{1,2,\ldots,\bar{M}\}$, which makes the triggering rule (2.22) no longer invariant along rays (see Remark 2.3). However, if the magnitude of the state is large compared to the magnitude of the disturbance covariance matrix, then the triggering behavior is still similar to the deterministic case. Suppose now that this is not the case, i.e., for a given (small) state x, $\bar{c}_M$ plays an important role in (2.22). From (2.20) and (2.13) we can conclude that an unstable A will favor large d_M and, hence, positive $\bar{c}_M$. If $\bar{c}_M$ is larger, then *for the same state* x, $M(x)$ given by (2.24a), will be smaller, which is an indication that the average inter-transmission interval will be smaller. We will observe this in the example provided in Section 2.5. Contrarily, a stable A will favor smaller d_M, negative $\bar{c}_M$, and larger $M(x)$ *for the same state* x, which is an indication that the average inter-transmission interval will be larger. This corresponds well with intuition, as unstable systems require more attention (control updates). ◁

Remark 2.5. (Minimum inter-transmission interval) The minimum inter-transmission interval for the closed-loop system given by (2.1), (2.9) and (2.14) is defined as

$$M_{min} := \min \big\{ t_{l+1}^{x_0,\mathbf{w}} - t_l^{x_0,\mathbf{w}} \mid l \in \mathbb{N},\ x_0 \in \mathbb{R}^{n_x} \text{ and } \mathbf{w} = (w_0, w_1, \ldots),\ w_i \in \mathbb{R}^{n_w}, i \in \mathbb{N} \big\},$$

where we now included the explicit dependence of the transmission times on x_0 and $\mathbf{w} = (w_0, w_1, \ldots)$. Based on the above reasoning, we obtain

$$M_{min} = \max\{M \in \{1, 2, \ldots, \bar{M}\} \mid \forall x \in \mathbb{R}^{n_x},\ x^\top P_M^* x + \bar{c}_M \leq \beta_1 x^\top P x\}.$$

In the deterministic case, i.e., $E = 0$ and, as a consequence, $\bar{c}_M = 0$, for all M, this reduces to

$$M_{min} = \max\{M \in \{1, 2, \ldots, \bar{M}\} \mid P_M^* \preceq \beta_1 P\}. \tag{2.26}$$

◁

Remark 2.6. (Effect of β_1) For the sake of simplicity, consider that $E = 0$ yielding that $\bar{c}_M$ in (2.22) is zero for all M. If β_1 is large, then for a given state x, the condition $x^\top(P_M^* - \beta_1 P)x \leq 0$ may hold for large values of M. Hence, the system may operate in open loop for a long time, with a control value obtained from (2.19), possibly considerably moving away from the origin. We see that this is possible since the final cost in (2.19) is bounded by β_1 times the cost of a periodic implementation after M steps. As a consequence, after a choice of a large $M \in \{1, 2, \ldots, \bar{M}\}$, a considerable number of transmissions may be required afterwards. This 'greedy' effect (see also Remark 2.1) can be observed from the simulation results in Section 2.5. ◁

Note that (2.25) results in a "piecewise linear" control law, which can be obtained by checking a finite number of inequalities as in (2.22). In fact, it is easy to see that the state-space $\mathbb{R}^{n_x}$ is partitioned in regions induced by the inequalities

$$\begin{aligned} &x^\top P_M^* x + \bar{c}_M \leq \beta_1 x^\top P x, \\ &x^\top P_N^* x + \bar{c}_M > \beta_1 x^\top P x,\ N = M+1, \ldots, \bar{M}, \end{aligned}$$

for $M \in \{1, 2, \ldots, \bar{M}\}$. Note that P_M^*, $\bar{c}_M$ and P can be computed off-line. In absence of disturbances ($E = 0$), $\bar{c}_M = 0$ and these regions are *polyhedral* (but not necessarily convex).

2.5 A numerical example

In order to illustrate the self-triggered strategy presented in Section 2.2, we consider a system consisting of two masses ($m_1 = m_2 = 1$) connected by a

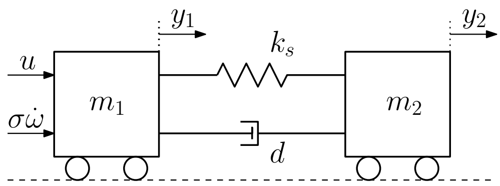

Fig. 2.1. Schematic representation of the considered system.

spring and damper, of which the continuous-time dynamics are given by

$$\begin{aligned}\ddot{y}_1 &= -k_s(y_1 - y_2) - d(\dot{y}_1 - \dot{y}_2) + u + \sigma\dot{\omega},\\ \ddot{y}_2 &= k_s(y_1 - y_2) + d(\dot{y}_1 - \dot{y}_2),\end{aligned}$$

or equivalently, using $x = [y_1 \;\; y_2 \;\; \dot{y}_1 \;\; \dot{y}_2]^\top$, by the state-space formulation

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_s & k_s & -d & d \\ k_s & -k_s & d & -d \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} (u + \sigma\dot{\omega}) \\ &=: A_c x + B_c u + E_c \dot{\omega},\end{aligned} \tag{2.27}$$

where $k_s = 5$ and $d = 1$ are the spring stiffness and damping coefficient, respectively, and $\dot{\omega}$ is a scalar unitary variance white noise process[1] and σ is a positive constant. A schematic representation of the considered system is given in Figure 2.1. The control performance is measured by a continuous-time infinite horizon discounted cost function

$$J_c = \int_0^\infty e^{-\alpha_c s} \mathbb{E}\big[\, \big(x^\top(s)x(s) + 25u^\top(s)u(s)\big) \mid x_0 \big] ds \tag{2.28}$$

for $\alpha_c \in \mathbb{R}_{\geq 0}$. To convert this continuous-time setup into a discrete-time setup, we apply *exact discretization* with sampling period $h = 0.25$, assuming a zero-order hold input between two sampling instants, which leads to a discrete-time LTI system of the form (2.1), where $A = e^{A_c h}$, $B = \int_0^h e^{A_c s} ds B_c$ and $E = \int_0^h e^{A_c s} ds E_c$. Similarly, by *exact discretization* of the continuous-time cost function (2.28), we obtain a discrete-time infinite horizon discounted cost function taking the form (2.2),with

$$\begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} = \int_0^h e^{-\alpha_c s} e^{\left[\begin{smallmatrix} A_c & B_c \\ 0 & 0 \end{smallmatrix}\right]^\top} \begin{bmatrix} I & 0 \\ 0 & 25I \end{bmatrix} e^{\left[\begin{smallmatrix} A_c & B_c \\ 0 & 0 \end{smallmatrix}\right]} ds$$

[1] Formally, (2.27) corresponds to the stochastic differential equation $dx = (A_c x + B_c u)dt + E_c d\omega$, where ω is a scalar unitary covariance Wiener process.

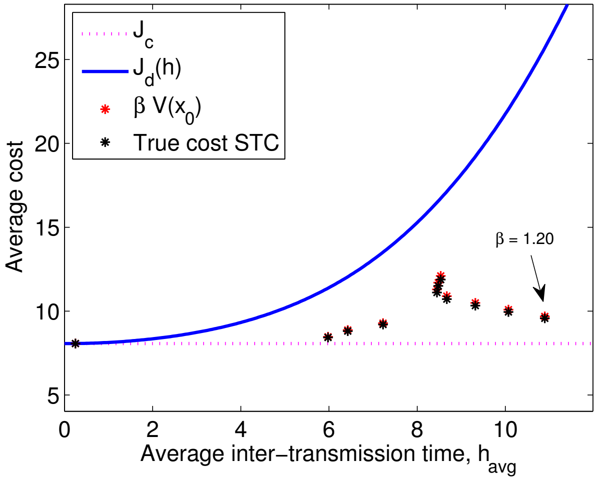

Fig. 2.2. Cost of implementation at (average) inter-transmission times for different control strategies, averaged over 100 initial conditions.

and $\alpha = e^{-\alpha_c h}$, which is exactly equal to (2.28) given the sampled-data implementation. As a consequence, all statements on cost provided below are expressed in terms of the continuous-time cost (2.28). In the remainder of this section, we consider two cases. Firstly, we consider the case where there are no disturbances acting on the system, i.e., $E = 0$, and as such, we recover an LQR like framework; secondly, we study the case where the system is subject to disturbances, i.e., $E \neq 0$.

2.5.1 Self-triggered LQR control

In this section, we evaluate the effectiveness of the self-triggered strategy in the absence of disturbances. We take $\sigma = 0$ (which implies $E = 0$) and $\alpha_c = 0$ (which implies $\alpha = 1$) and, as such, we recover an LQR like framework. To make a fair comparison with standard periodic time-triggered LQR control, note that an alternative way to reduce the required communication is to simply use larger h, i.e., to sample the system with a larger sampling period. In this way fewer communication resources are required as well, and we still obtain a guaranteed cost of the form (2.28) via (2.3) by solving the corresponding discrete-time LQR problem using (2.5). In Figure 2.2 we compare this periodic time-triggered LQR approach selecting larger sampling periods h with the STC approach with $\bar{M} = 150$. We will plot the performance with respect to the average sampling period h_{avg}. The results presented are obtained by averaging over 100 initial conditions on the four dimensional unit hypersphere. Figure 2.2 shows the performance J_c of the continuous-time controller and the performance $J_d(h)$ of the optimal *periodic* LQR controller for various sampling periods h. Moreover, for $\beta_1 \in \{1,\ 1.05,\dots,\ 1.45,\ 1.50\}$, Figure 2.2 shows both the upper bound on the (averaged) cost for the STC implementation $V_{\beta_1,\beta_2}(x_0) = \beta_1 x_0^\top P x_0$ and the (averaged) true cost of the STC strategy (computed over a finite, but sufficiently large, horizon) plotted against the 'averaged' inter-transmission times. Note that, for $E = 0$, from (2.7)-(2.8) we see that β_2 plays no role due to the absence of disturbances. In fact, for $E = 0$ we have that $V_{\beta_1,\beta_2}(x_0) = \beta_1 V(x_0)$. From Figure 2.2 we can see that the STC strategy can achieve $h_{avg} = 10.89$ for $\beta = 1.20$ with guaranteed cost 9.68, whereas periodic control with cost 9.68 requires sampling at $h = 4.46$. Hence, on average we can sample and transmit a factor 2.17 fewer by using the STC strategy while obtaining the same performance guarantees. Moreover, the cost of the periodic LQR implementation at $h = 10.89$ is 25.67, which is more than 2.5 times larger than the cost of the STC strategy at $h_{avg} = 10.89$. Hence, this shows that the STC strategy can realize combinations of average inter-transmission times h_{avg} and cost that cannot be realized by periodic time-triggered LQR implementations. Table 2.1 shows the results for $\beta_1 = 1$ to 1.3, and moreover contains the guaranteed minimum inter-transmission times $h_{min} := M_{min}h$ for the STC strategy, determined using (2.26). Interestingly, the minimal inter-transmission time h_{min} for the STC strategy with $\beta_1 = 1.20$ is $h_{min} = 3.75$, which is not much lower than the corresponding value $h = 4.46$ that is needed for a periodic LQR controller with the same performance. From both Figure 2.2 and Table 2.1 we observe that for the STC strategy, increasing β_1 above 1.20 is not useful for this example. A possible cause of this effect is the "greedy" behavior of the STC strategy as mentioned in Remark 2.6.

Remark 2.7. (Other time-triggered solutions / Feedback in the triggering mechanism) In this section, we compared our self-triggered approach with the most common time-triggered control scheme, namely, periodic (sampled-data) control, see Figure 2.2. In general, it is difficult to design alternative time-triggered solutions that, over a range of initial conditions (or for various disturbance realizations $w_t,\ t \in \mathbb{N}$), yield similar performance in terms of utilization of communication resources and control performance as the proposed STC approach. To illustrate this fact and the need for feedback in the triggering mechanism (as opposed to the "open-loop" determination of the transmission times), consider the initial conditions $x_0^{(1)} = [0.1126 \quad 0.0887 \quad 0.8330 \quad -0.5344]^\top$ and $x_0^{(2)} = [-0.9178 \quad -0.3530 \quad -0.1422 \quad 0.1130]^\top$ and let $\mathbf{t}^{x_0^{(i)}},\ i = 1,2$, denote the sequence of triggering times obtained from (2.25) for $x_0^{(1)}$ and $x_0^{(2)}$, respectively. Table 2.2 shows the results for the STC approach for these initial conditions for $\beta_1 = 1.20$ and, moreover, contains the results of simulating the system starting in $x_0^{(1)}$ but with the transmission times given by $\mathbf{t}^{x_0^{(2)}}$, and vice versa. We observe that the true costs for both initial conditions are higher when

Table 2.1. Results for self-triggered LQR strategy with different values of β_1.

β_1	h_{avg}	h_{min}	$\beta_1 V(x_0)$	True Cost
1.00	0.2500	0.2500	8.0694	8.0694
1.05	5.9777	1.7500	8.4729	8.4285
1.10	6.4252	2.5000	8.8763	8.8095
1.15	7.2258	3.2500	9.2798	9.2000
1.20	10.8945	3.7500	9.6833	9.5772
1.25	10.0702	4.0000	10.0868	9.9543
1.30	9.3188	4.7500	10.4902	10.3315

Table 2.2. Results for different transmission sequences for two initial conditions.

	$x_0^{(1)}$	$x_0^{(2)}$
$h_{avg}(\mathbf{t}^{x_0^{(1)}})$	11.1413	11.1413
$h_{avg}(\mathbf{t}^{x_0^{(2)}})$	10.6979	10.6979
$\beta_1 (x_0^{(i)})^\top P x_0^{(i)},\ i = 1,2$	2.3872	4.6968
True Cost $(\mathbf{t}^{x_0^{(1)}})$	2.3619	5.1760
True Cost $(\mathbf{t}^{x_0^{(2)}})$	2.4171	4.6408

the transmission times used are based on the other initial condition. Compared to time-triggered approaches, self-triggered control has the advantage of having "feedback" in determining the transmission times, while a time-triggered control approach selects the transmission times in an open-loop manner. Due to this feedback, self-triggered control is able to obtain better performance in terms of communication resources and control performance over a range of initial conditions (or for various disturbance realizations) than time-triggered control, in which one has to find *one* sequence of transmission times that works well for all initial conditions (and/or all disturbance realizations). ◁

2.5.2 Self-triggered linear quadratic control with discounted cost

In this section, we consider the influence of disturbances on the effectiveness of the STC strategy, meaning that we now take $E \neq 0$. More specifically, we consider the case where the actuation of the system is subject to disturbances. For $\sigma = 0.02$, $\alpha = 0.99$, $h = 0.25$ and $\bar{M} = 120$, Figure 2.3 shows the time response

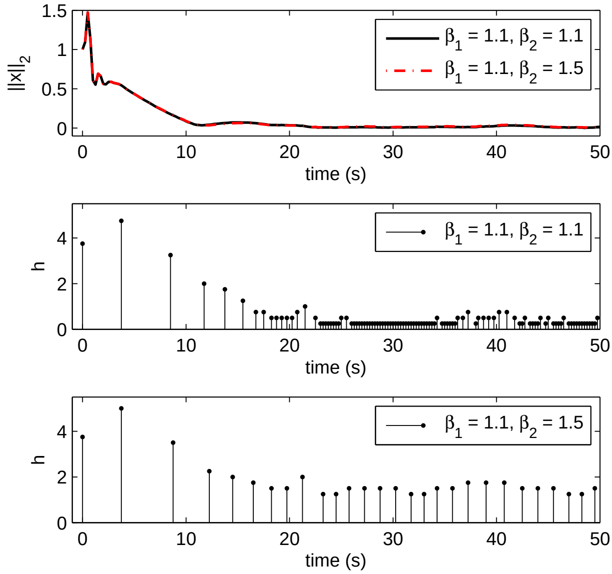

Fig. 2.3. State trajectories and inter-transmission times for STC strategy in the presence of disturbances, with $\beta_1 = \beta_2 = 1.1$ and $\beta_1 = 1.1,\ \beta_2 = 1.5$.

of the states and inter-transmission times for the self-triggered strategy with $\beta_1 = \beta_2 = 1.1$ starting from $x_0 = \begin{bmatrix}0.49 & -0.40 & 0.74 & -0.25\end{bmatrix}^\top$. We observe that during the first 15 s the STC strategy significantly reduces the required number of transmissions, despite the presence of the disturbance. However, after 15 s the STC strategy is not able to significantly reduce the number of required transmissions. This can be explained using the considerations provided in Remark 2.4. In fact, when the state x is large in comparison to the noise covariance, i.e., for about the first 15 s, we observe that the triggering strategy shows a similar average transmission rate and performance to the deterministic case. However, after 15 s the states are close to the origin and the additional term $\bar{c}_M$ due to disturbances dominates condition (2.22). By increasing β_2 (hence, giving away a bit of performance in steady state, see (2.7)-(2.8)), we can decrease $\bar{c}_M$. The results for the case where $\beta_1 = 1.1$ and $\beta_2 = 1.5$ are also shown in Figure 2.3. For $\beta_2 = 1.5$, we observe a significant reduction in communication resources, even if the state is close to the origin. These observations are confirmed by 100 Monte Carlo simulations over a finite, but sufficiently large, horizon for each of the 10 initial conditions on the four dimensional unit hypersphere. The averaged results are given in Table 2.3. With only a slight degradation in performance

Table 2.3. Results for self-triggered strategy, average over 10 initial conditions and 100 Monte Carlo simulations for each initial condition.

β_1	β_2	h_{avg} $t<15$ s	h_{avg} $t>15$ s	Cost $t<15$ s	Total Cost	$V_{\beta_1,\beta_2}(x_0)$
1.10	1.10	1.9539	0.3484	8.1106	8.1443	8.2951
1.10	1.50	3.1414	1.4447	8.1134	8.1560	8.3087

($\beta_1 = \beta_2 = 1.1$), and despite the presence of disturbances, the STC strategy reduces the required communication during transients by a factor 7.8, on average, when compared to time-triggered periodic control with $h = 0.25$. By studying the second control configuration with an increased value of β_2 to 1.5 we observe that, on average, the required communication reduces also by a factor 5.8 *after* transients.

2.6 Conclusions

In this chapter, we proposed a self-triggered control strategy for discrete-time linear systems with (discounted) quadratic cost addressing two important issues: the guarantee of desirable performance levels by co-design, and realizing a significant reduction in utilization of the system's communication and/or actuator resources (compared to periodic time-triggered control). Regarding the performance guarantees, the control laws and triggering mechanisms were designed such that an a priori chosen (suboptimal) level of performance in terms of (discounted) quadratic cost is guaranteed. Interestingly, our proposed methodology provided a solution to the problem of co-design of the feedback law and the triggering condition, a problem hardly addressed in the literature. The designed self-triggered control strategy can easily be implemented in practice as it results in a simple piecewise linear control law.

The effectiveness of the approach was illustrated by means of a numerical example, showing a significant reduction in the usage of the system‘s communication and/or actuator resources, without trading much of the optimally achievable performance. In fact, for the self-triggered LQR strategy, combinations of average sampling periods and performance levels are obtained, which are not achievable with standard periodic time-triggered LQR solutions. As such, this work is one of the first providing quantitative evidence that aperiodic control strategies, such as the STC strategy proposed in this chapter, can significantly improve beyond time-triggered periodic control. In the presence of disturbances, the self-triggered strategy also realizes a significant reduction in the usage of network resources, even at only a slight degradation in performance. As such, the proposed approach provides a viable control strategy to balance the usage of the system's resources and control performance beyond the possibilities of standard periodic time-triggered controllers.

Chapter 3

Resource-aware MPC for constrained nonlinear systems: A self-triggered control approach

Abstract – In systems with resource constraints, such as actuation limitations or limited communication bandwidth, it is desired to obtain control signals that are either sparse or sporadically changing in time to reduce resource utilization. In this chapter, we propose a resource-aware self-triggered MPC strategy for discrete-time nonlinear systems subject to state and input constraints that has three important features: Firstly, significant reductions in resource utilization can be realized without modifying the cost function by input regularization or explicitly penalizing resource usage. Secondly, the control laws and triggering mechanisms are synthesized so that a priori chosen performance levels (in terms of the original cost function) are guaranteed by design next to asymptotic stability and constraint satisfaction. Thirdly, we address the co-design problem of jointly designing the feedback law and the triggering condition. By means of numerical examples, we show the effectiveness of this novel strategy.

3.1 Introduction

In more and more control applications, it becomes essential to explicitly address resource constraints in the implementation and the design of the control law. Due to constraints on a system's actuation resources it may be desirable to have control signals that are either sparse or sporadically changing in time. The use of sparse or sporadically changing actuation signals can have several benefits, such as improved fuel efficiency or a larger lifetime of actuators subject to wear and tear. For instance, in [44] sparse thrust actuation signals are obtained to use

This chapter is based on [41].

fuel in the control of a spacecraft in an efficient manner, and in [26] sporadically changing actuation is used in the control of an autonomous underwater vehicle in order to decrease fuel consumption and increase the deployment time. Also in overactuated systems, it can be desirable to have a smart control allocation policy that does not require each actuator to be updated continuously in order to realize a desired level of performance. One specific example includes the usage of sparse actuation signals to control the roll of an overactuated ship [37]. Next to systems with constrained actuation resources, also in the area of networked control systems (NCSs) [54] with scarce communication resources there is an interest in control signals that are sparse or sporadically changing in time, depending on whether the control value is set to zero or held to the previous value, respectively, when no new information is received by the actuators, see, e.g., [81]. In case the communication in NCSs is wireless, a further resource constraint is often induced by battery-powered sensors. Indeed, the radio chip, used to communicate sensor data required to update the control law, is the primary source of energy consumption in the sensors [80]. Hence, reducing the number of times the control law needs to be executed in NCSs leads to lower network resource utilization and enhanced lifetimes of the battery-powered devices.

Given the above motivations, in this chapter, we are particularly interested in resource-aware control strategies for discrete-time nonlinear systems that have scarce (actuation and/or communication) resources, and, moreover, are subject to state and input *constraints*. One of the most widely used techniques for control of constrained systems is model predictive control (MPC). In fact, several works already exist in the literature addressing the control of constrained systems with scarce resources. The most well known approach is based on modifying the MPC problem by appending the original MPC control cost with an ℓ_1 penalty on the input in order to obtain sparse input signals. Regularizing by the ℓ_1-norm is known to induce sparsity in the sense that, individual components of the input signal will be equal to zero, see, e.g., [6, 37, 38, 73], in which the focus is on discrete-time *linear* systems. Also different types of sum-of-norms regularization can be used to obtain so-called group sparsity, meaning that at many time instances the entire input vector becomes zero, see, e.g., [77]. Although a regularization penalty on the input is effective in obtaining sparse solutions, due to the altered cost function no performance guarantees are given in terms of the original MPC cost function.

Considering the NCS literature, two main approaches oriented towards generating sporadically changing input signals (to save computation and communication resources) can be distinguished, namely, event-triggered control (ETC), see, e.g., [12, 13, 30, 50, 63, 85, 98], and self-triggered control (STC), see, e.g., [3, 7, 31, 40, 68, 93, 93, 95], see also [48] for a recent overview. In ETC and STC, the control law consists of two elements being a feedback controller that computes the control input, and a triggering mechanism that determines when the control input has to be updated. The difference between ETC and STC is that in the former the triggering consists of verifying a specific condition continuously and when it becomes true, the control task is triggered, while in the latter at an update time the next update time is pre-computed. In the context of NCSs, another popular approach to reduce utilization of communication resources (without resorting to sporadically changing input signals) in networks with messages with a high payload is packet-based predictive control in which a sequence of inputs and/or states is sent to the actuators (instead of a single input value), see, e.g., [16,43,55]. In fact, there are several works available that combine ideas from ETC and packet-based predictive control, see, e.g., [18, 32, 33]. In [32] an event-triggered MPC setup for unconstrained systems was presented using tools from input-to-state stability (ISS). Extensions of this work were presented in [33] in which ETC schemes for constrained discrete-time systems were proposed. In particular, the optimal control sequence coming from the MPC optimization problem is applied in an open-loop fashion between the event times. In [18], another ETC scheme based on using the sequence of optimal controls obtained from the MPC optimization problem is presented, for discrete-time linear systems. In addition to sending the computed MPC input sequence to the actuators, in [18], also the predicated optimal state (or output) sequence is transmitted to the sensors. An event-triggering mechanism, positioned at the sensors, is chosen such that it detects if the true states deviate from the predicted states in the MPC problem, using absolute thresholds. Note that in [18,32,33] the control law and the triggering condition are designed independently, and also that no a priori performance guarantees on the (original) control cost are provided.

The ideas in [18, 32, 33] are appealing for NCS applications, but they still require the actuators to be updated in a periodic manner (at every time step), which is undesirable in control applications with constrained actuation resources. In addition the schemes in [18, 32, 33] would lead to bursts of communication at certain points in time, which one may like to avoid, especially in networks with messages with a low payload. To ameliorate this communication burden, [73] proposes a packet-based predictive control scheme that combines a classical quadratic MPC cost function with ℓ_1 input regularization and sends the obtained sparse input sequences to the actuators. A STC approach for unconstrained discrete-time linear systems that avoids bursts in communication and frequent updating of the actuators by sending only a single control value and keeping this constant until the next execution time is presented in [53]. The strategy in [53] solves a co-design problem of simultaneously designing the control law and the triggering condition and allows trading in control performance to obtain lower (communication and/or actuation) resource utilization. In particular, [53] follows an ℓ_1-like regularization to solve the co-design problem by augmenting the quadratic cost function related to control performance with a penalty related to sampling the system (and updating the control law). Due to the additional penalties in [53,73], no performance guarantees are given in terms of the original MPC cost function. In [8], a scheme with the same purpose as in [53] is considered, also for unconstrained discrete-time linear systems. Instead of sending a single control value, [8] employs also an optimization-based approach using rollout techniques for minimizing a desired quadratic cost function with an explicit constraint on the average sampling/actuation rate. However, note that [8, 53, 73] only apply to *unconstrained* discrete-time linear systems. All these works focus on discrete-time systems, but note that also in the context of continuous-time NMPC event-triggered implementations have been proposed in [91] and [90].

This chapter provides an extension of our preliminary works [15,40]. In [40], we proposed a self-triggered control scheme with an a priori guaranteed level of performance, in terms of quadratic cost, for discrete-time linear *unconstrained* systems, which was extended in [15] to a self-triggered MPC scheme that is capable of handling state and input constraints for *linear* systems. In this work, we propose a general framework for the self-triggered MPC strategy applying to discrete-time *nonlinear* systems subject to state and input constraints and a possibly *non-quadratic* cost function. Moreover, in [15, 40], the focus is on obtaining sporadically changing input profiles, whereas this chapter shows also how the framework can be extended to include the case where sparse input profiles are desired. The proposed self-triggered control law possesses three important features: (i) significant reductions in resource utilization are obtained, (ii) a priori closed-loop performance guarantees are provided (by design) in terms of the original cost function, next to asymptotic stability and constraint satisfaction, (iii) co-design of both the feedback law and triggering condition is achieved. To elaborate on these features, we emphasize that the proposed STC approach reduces resource utilization without modifying the cost function by input regularization or explicitly penalizing resource usage, thereby being essentially different from [6, 37, 38, 53, 73, 77]. This feature enables that the proposed STC strategy will not only realize stabilization towards a desired equilibrium, but it also provides a priori guarantees on an infinite horizon cost directly related to the MPC cost (feature (ii)). More specifically, one can choose a priori a desired suboptimal level of control performance, and the controller design will automatically take this into account, while at the same time aiming for a reduction of the network and actuation resources (feature (i)). Regarding feature (iii), we note that most existing design methods for ETC and STC are emulation-based in the sense that the feedback controller is designed assuming a standard time-triggered implementation, and thus not taking the eventual event- or self-triggered implementation into account. The triggering mechanism in then designed in a subsequent phase (in which the controller is already fixed). Since the feedback controller is designed before the triggering mechanism, it is difficult, if not impossible, to obtain an optimal design of the combined feedback controller and the triggering mechanism in the sense that the minimum number of controller executions is achieved while guaranteeing stability and a certain level of closed-loop performance. In this chapter, we provide a synthesis technique that determines the next execution time of the algorithm and the applied control value *simultaneously*, thereby selecting control values that are optimized for not being updated for a maximal number of steps. The presented self-triggered MPC framework offers the flexibility to configure it for limited actuator applications (inducing the control input to be sparse) or networked control applications (by, for instance, keeping the control input constant as much as possible) by selecting proper admissible control sequences in the MPC optimal control problem. This also further extends our results in the preliminary paper [15], where only the latter case was considered. Finally, compared to packet-based solutions, note that in our solution bursts in communications can be avoided as it is the objective of the control strategy to send just a *single* control value and not a sequence of predicted future values to the actuators.

The remainder of the chapter is organized as follows. After indicating the notational conventions used in this chapter, in Section 3.2 the problem formulation is presented where a stabilizing MPC setup is described, which forms the basis for developing the self-triggered MPC strategy. The proposed self-triggered approach is discussed in Section 3.3, while Section 3.4 contains the implementation considerations. The effectiveness of the proposed approach is demonstrated by means of numerical examples in Section 3.5. Finally, Section 3.6 presents the conclusions.

Nomenclature

Let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real numbers and the set of non-negative integers (including zero), respectively. For $s, t \in \mathbb{N}$, the notation $\mathbb{N}_{\geq s}$, $\mathbb{N}_{[s,t)}$, $\mathbb{N}_{(s,t)}$ and $\mathbb{N}_{[s,t]}$ is used to denote the sets $\{r \in \mathbb{N} \mid r \geq s\}$, $\{r \in \mathbb{N} \mid s \leq r < t\}$, $\{r \in \mathbb{N} \mid s < r < t\}$ and $\{r \in \mathbb{N} \mid s \leq r \leq t\}$, respectively. Sequences of vectors are indicated by bold letters, e.g., $\mathbf{u} = (u_0, u_1, \ldots, u_M)$ with $u_i \in \mathbb{R}^{n_u}, i \in \{0, 1, \ldots, M\}$, where $M \in \mathbb{N} \cup \{\infty\}$ will be clear from the context. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a $\mathcal{K}$-function if it is continuous, strictly increasing and $\alpha(0) = 0$. The empty set is denoted by $\emptyset$.

3.2 Problem formulation

Consider a discrete-time nonlinear system of the form

$$x_{t+1} = g(x_t, u_t), \tag{3.1}$$

where $x_t \in \mathbb{R}^{n_x}$ and $u_t \in \mathbb{R}^{n_u}$ are the state and the input, respectively, at time $t \in \mathbb{N}$. We assume that $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is continuous and that $g(0, 0) = 0$, indicating that 0 is an equilibrium for (3.1) under zero input. The system (3.1) is subject to input and state constraints given by

$$u_t \in \mathbb{U} \quad \text{and} \quad x_t \in \mathbb{X}, \quad t \in \mathbb{N}, \tag{3.2}$$

where $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ and $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ are compact sets containing the origin in their interiors. For $N \in \mathbb{N}$, $x_k(x, \mathbf{u})$ denotes the solution to (3.1) at time $k \in \mathbb{N}_{[0,N]}$ initialized at $x_0 = x$ and with control input sequence given by $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1})$.

3.2.1 Standard stabilizing MPC setup

For the system (3.1), we now consider the MPC setup given by the following optimization problem. For a fixed prediction horizon $N \in \mathbb{N}_{\geq 1}$, given state $x_t = x \in \mathbb{X}$ at time $t \in \mathbb{N}$,

$$\min \quad J_N(x, \mathbf{u}) := F(x_N(x, \mathbf{u})) + \sum_{k=0}^{N-1} L(x_k(x, \mathbf{u}), u_k) \tag{3.3a}$$

$$\text{w.r.t.} \quad \mathbf{u} \in \mathcal{U}_N(x) := \Big\{ \mathbf{u} \in \mathbb{U}^N \;\Big|\; \forall k \in \{1, 2, \ldots, N-1\}, \; x_k(x, \mathbf{u}) \in \mathbb{X} \text{ and } x_N(x, \mathbf{u}) \in \mathbb{X}_T \Big\}. \tag{3.3b}$$

Here, $\mathbb{X}_T \subseteq \mathbb{X}$ is the terminal set, which is assumed to be closed and to contain the origin in its interior. The stage cost is given by $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_{\geq 0}$, where $L(0, 0) = 0$, and the terminal cost is given by $F : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$, where $F(0) = 0$. We assume that L and F are continuous functions.

A state $x \in \mathbb{X}$ is called feasible for the optimization problem (3.3) if there exists at least one admissible input sequence, i.e., $\mathcal{U}_N(x) \neq \emptyset$. The set of feasible states is denoted by $\mathbb{X}_f$, i.e., $\mathbb{X}_f = \{x \in \mathbb{X} \mid \mathcal{U}_N(x) \neq \emptyset\}$. As N is finite, for $x \in \mathbb{X}_f$, the given conditions ensure the existence of a minimum for the optimization problem (3.3). For $x \in \mathbb{X}_f$, $V(x)$ denotes the corresponding minimum value for the optimization problem (3.3). Hence, $V : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ is the MPC value function given by

$$V(x) := \min_{\mathbf{u} \in \mathcal{U}_N(x)} J_N(x, \mathbf{u}). \tag{3.4}$$

For any state $x \in \mathbb{X}_f$, a minimizer to optimization problem (3.3) is known to exist, and we use $\mathbf{u}^*(x) = (u_0^*(x), \ldots, u_{N-1}^*(x))$ to denote a particular one. Hence,

$$J_N(x, \mathbf{u}^*(x)) = \min_{\mathbf{u} \in \mathcal{U}_N(x)} J_N(x, \mathbf{u}). \tag{3.5}$$

The resulting MPC law $u^{\mathrm{mpc}} : \mathbb{X}_f \to \mathbb{U}$ is now defined as

$$u^{\mathrm{mpc}}(x) := u_0^*(x), \tag{3.6}$$

which is implemented in a receding horizon fashion as

$$u_t = u^{\mathrm{mpc}}(x_t), \; t \in \mathbb{N}. \tag{3.7}$$

To guarantee recursive feasibility and closed-loop stability in the above setup we will use the terminal set and cost method, see, e.g., [67], for which we require the following assumptions throughout the remainder of the chapter.

Assumption 3.1. *There exists $K : \mathbb{X}_T \to \mathbb{R}^{n_u}$ such that for $x \in \mathbb{X}_T$*

$$F(g(x, K(x))) - F(x) \le -L(x, K(x)), \tag{3.8a}$$
$$g(x, K(x)) \in \mathbb{X}_T, \tag{3.8b}$$
$$K(x) \in \mathbb{U}. \tag{3.8c}$$

Moreover, there exist $\alpha_1,\ \alpha_2 \in \mathcal{K}$ such that

$$L(x, u) \ge \alpha_1(\|x\|), \quad \text{for all } x \in \mathbb{X}_f, u \in \mathbb{U}, \tag{3.8d}$$
$$F(x) \le \alpha_2(\|x\|), \quad \text{for all } x \in \mathbb{X}_T. \tag{3.8e}$$

Under these standing assumptions, the optimal MPC law as in (3.7) results in a closed-loop system given by (3.1) and (3.7), which

(i) is asymptotically stable in $\mathbb{X}_f$, in the sense that

- [well-posedness]: for each $x_0 \in \mathbb{X}_f$ the corresponding trajectory x_t to (3.1) and (3.7) exists for all $t \in \mathbb{N}$;
- [Lyapunov stability]: for any $\varepsilon > 0$ there is a $\delta > 0$ such that if $x_0 \in \mathbb{X}_f$ and $\|x_0\| \le \delta$ then the corresponding trajectory x_t to (3.1) and (3.7) satisfies $\|x_t\| \le \varepsilon$;
- [attractivity]: for any $x_0 \in \mathbb{X}_f$ the corresponding trajectory x_t to (3.1) and (3.7) satisfies $\lim_{t\to\infty} x_t = 0$.

(ii) satisfies input and state constraints, i.e., (3.2).

(iii) leads to performance guarantees of the form

$$\sum_{t=0}^{\infty} L(x_t, u_t) \le V(x_0). \tag{3.9}$$

See [67] and the references therein, for the proofs and further details on the terminal set and cost method.

3.2.2 Resource-aware MPC with guaranteed performance

Clearly for all $x_0 \in \mathbb{X}_f$, the implementation of (3.7) requires the computation of the control value $u^{\mathrm{mpc}}(x_t)$ and uses communication and/or actuation resources at *each* time instant $t \in \mathbb{N}$. This is undesirable in systems with limited resources, where it may be preferable to use input profiles that reduce the required resource utilization. As already indicated in the introduction, such resource-aware input profiles are typically sparse in case actuation resources are limited or sporadically changing in case communication resources are limited.

The main focus of this work is to synthesize control laws that take scarceness of the available (actuation and/or communication) resources into account, but still guarantee certain performance requirements in terms of the infinite horizon cost as in (3.9), next to closed-loop asymptotic stability in $\mathbb{X}_f$ as in (i) and constraint satisfaction as in (ii). In particular, regarding the former, we aim at obtaining resource-aware input signals while still satisfying the performance guarantee

$$\sum_{t=0}^{\infty} L(x_t, u_t) \leq \beta V(x_0), \tag{3.10}$$

where a performance degradation index $\beta \geq 1$ is introduced to balance the degree of suboptimality and the resource utilization.

Remark 3.1. Although we focus on the MPC setup discussed in this section, the proposed self-triggered strategy can be based on any asymptotically stabilizing MPC setup that is recursively feasible such that for each $x \in \mathbb{X}_f$ there exists $u \in \mathbb{U}$ such that $V(g(x,u)) - V(x) \leq -L(x,u)$ and $g(x,u) \in \mathbb{X}_f$. ◁

3.3 Self-triggered strategy

In this section we present a so-called self-triggered approach to solve the problem formulated in Section 3.2.2, focusing on the case of sparsity. In Section 3.3.2, we present variants of the self-triggered approach based on other input profiles relevant for certain type of applications (e.g., requiring sporadically changing input profiles) and in Section 3.3.3 we derive various important theoretical properties.

3.3.1 Approach and algorithm for self-triggered MPC

To introduce the approach, consider the set of execution times $\{t_l \mid l \in \mathbb{N}\} \subseteq \mathbb{N}$ of the control algorithm that satisfies $t_{l+1} > t_l$, for all $l \in \mathbb{N}$, and $t_0 = 0$, and the corresponding control signal

$$u_t = \begin{cases} \bar{u}_l, & \text{if } t = t_l, \\ 0, & \text{if } t \in \mathbb{N}_{[t_l+1, t_{l+1})}, \end{cases} \tag{3.11}$$

where $\bar{u}_l \in \mathbb{U}$, $l \in \mathbb{N}$. At an execution time t_l the goal is to choose both the next control value $\bar{u}_l$ and the next execution time $t_{l+1} > t_l$. In fact, t_{l+1} will be selected as large as possible while still leading to the guarantee (3.10). Note that there is no actuation (and also no computation and communication) on times $t_l + 1, t_l + 2, \ldots, t_{l+1} - 1$. Instrumental in guaranteeing (3.10) will be the dissipation-like inequality

$$\sum_{t=t_l}^{t_{l+1}-1} L(x_t, u_t) \leq \beta \Big[V(x_{t_l}) - V(x_{t_{l+1}})\Big] \tag{3.12}$$

$$\text{subject to } \begin{cases} u_t = \begin{cases} \bar{u}_l \in \mathbb{U}, & \text{if} \quad t = t_l, \\ 0, & \text{if} \quad t \in \mathbb{N}_{[t_l+1, t_{l+1})}, \end{cases} \\ x_t \in \mathbb{X},\ t \in \{t_l+1, t_l+2, \dots, t_{l+1}-1\}, \\ x_{t_{l+1}} \in \mathbb{X}_f, \end{cases}$$

as we will see. In particular, at execution time t_l with state x_{t_l} the aim will now be to maximize t_{l+1} such that there exists a $\bar{u}_l$ for which the conditions in (3.12) are feasible. To formalize this setup, the notation

$$\bar{\mathcal{U}}_M(x) := \Big\{\mathbf{u} \in \mathbb{U}^M \mid \forall i \in \mathbb{N}_{[1,M-1]},\ \big(\Pi_i \mathbf{u} = 0 \text{ and } x_i(x, \mathbf{u}) \in \mathbb{X}\big),\ \text{and } x_M(x, \mathbf{u}) \in \mathbb{X}_f\Big\} \tag{3.13}$$

is introduced, in which the projection operators $\Pi_i : (\mathbb{R}^{n_u})^M \to \mathbb{R}^{n_u}$, $i \in \mathbb{N}_{[0,M-1]}$, are given by $\Pi_i \mathbf{u} = u_i$ for $\mathbf{u} = (u_0, u_1, \dots, u_{M-1})$ with $u_i \in \mathbb{R}^{n_u}$, $i \in \mathbb{N}_{[0,M-1]}$. Hence, the constraints in (3.13) specify, amongst others, that $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ is of the form $(\bar{u}, 0, \dots, 0)$ for some $\bar{u} \in \mathbb{U}$. An example of $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ is shown in Figure 3.1(a).

Using the notation in (3.13) and letting $M_l := t_{l+1} - t_l$ and $x_{t_l} = x$, inequality (3.12) can be rewritten as

$$\sum_{k=0}^{M_l-1} L(x_k(x, \mathbf{u}), \Pi_k \mathbf{u}) \leq \beta \big[V(x) - V(x_{M_l}(x, \mathbf{u}))\big] \tag{3.14}$$
$$\text{subject to } \mathbf{u} \in \bar{\mathcal{U}}_{M_l}(x),$$

where we used the time-invariance of both the system dynamics (3.1) and the constraints (3.2) to apply a time shift. Hence, at time t_l with state $x_{t_l} = x$ we aim at maximizing $M_l \in \mathbb{N}$, say $M_l^* \in \mathbb{N}$, such that (3.14) is still feasible for some $\mathbf{u} \in \bar{\mathcal{U}}_{M_l^*}(x)$. Naturally, $t_{l+1} = t_l + M_l^*$ and u_t in (3.11) is then taken as $u_t = \Pi_{t-t_l} \mathbf{u}$, $t \in \mathbb{N}_{[t_l, t_{l+1})}$. To express this formally, we define for $x \in \mathbb{X}_f$ and fixed $M_l = M$,

$$\mathcal{U}_M^{\text{st}}(x) := \Big\{\mathbf{u} \in \bar{\mathcal{U}}_M(x) \;\Big|\; (3.16) \text{ holds } \Big\}, \tag{3.15}$$

where

$$\sum_{k=0}^{M-1} L(x_k(x, \mathbf{u}), \Pi_k \mathbf{u}) \leq \beta \big[V(x) - V(x_M(x, \mathbf{u}))\big]. \tag{3.16}$$

Based on the above considerations, the following self-triggered MPC algorithm is proposed, where we use an integer $\bar{M} \in \mathbb{N}_{\geq 1}$ as a predefined upper bound on the inter-execution times. The upper bound $\bar{M}$ allows to limit the maximal time allowed between two consecutive execution times. As a consequence, it limits the number of optimization problems that are to be solved in evaluating Algorithm 3.1, on which we elaborate further in Section 3.4. In this way, the computational resources required for evaluating Algorithm 1 can be appropriately adapted, e.g., based on the available hardware.

Algorithm 3.1. *At execution time $t_l \in \mathbb{N}$, $l \in \mathbb{N}$, with corresponding state $x_{t_l} = x$, the next update time t_{l+1} and the corresponding control signal for $t \in \mathbb{N}_{[t_l,t_{l+1})}$ are given by*

$$t_{l+1} = t_l + \mathcal{M}^{\mathrm{st}}(x), \tag{3.17a}$$

$$u_t = \Pi_{t-t_l}\mathbf{u},\ t \in \mathbb{N}_{[t_l,t_{l+1})} \text{ for some } \mathbf{u} \in \mathcal{U}^{\mathrm{st}}(x), \tag{3.17b}$$

where

$$\mathcal{M}^{\mathrm{st}}(x) := \max\{M \in \{1,2,\dots,\bar{M}\} \mid \mathcal{U}^{\mathrm{st}}_M(x) \neq \emptyset\} \tag{3.18}$$

and

$$\mathcal{U}^{\mathrm{st}}(x) := \mathcal{U}^{\mathrm{st}}_{\mathcal{M}^{\mathrm{st}}(x)}(x). \tag{3.19}$$

Here, $\mathcal{M}^{\mathrm{st}} : \mathbb{R}^{n_x} \to \{1,2,\dots,\bar{M}\}$ and $\mathcal{U}^{\mathrm{st}} : \mathbb{R}^{n_x} \to \mathbb{U}^{\mathcal{M}^{\mathrm{st}}(x)}$ denote the time between two executions and the set of possible control sequences, respectively. Note that only $\Pi_0\mathbf{u}$ with $\mathbf{u} \in \mathcal{U}^{\mathrm{st}}(x)$ has to be transmitted to the actuators, as afterwards only zero inputs are implemented until a new control value is received.

3.3.2 Variants of self-triggered strategy

The STC strategy proposed in the previous section, produces solutions with input profiles that have the so-called group sparsity property [56] in the sense that $u_t = 0,\ t = \mathbb{N}_{[t_l+1,t_{l+1})}$, see Figure 3.1(a). In different applications it can be preferable to have other profiles of the input than enforced by $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ in (3.15). Our framework is general in the sense that it offers the flexibility to adopt such different input profiles. To make this more clear, consider NCS-type applications, in which it is sometimes desirable to avoid frequent (continuous) changes of the control input. Hence, it is desirable to keep the inputs constant between two execution times. The approach presented in Section 3.3 can include this easily by replacing $\bar{\mathcal{U}}_M(x)$ in (3.15) by

$$\tilde{\mathcal{U}}_M(x) := \Big\{\mathbf{u} \in \mathbb{U}^M \mid \exists \bar{u} \in \mathbb{U},\ \forall i \in \mathbb{N}_{[0,M-1]},\ \Pi_i\mathbf{u} = \bar{u},\\ \forall i \in \mathbb{N}_{[1,M-1]},\ x_i(x,\mathbf{u}) \in \mathbb{X} \text{ and } x_M(x,\mathbf{u}) \in \mathbb{X}_f\Big\}. \tag{3.20}$$

The constraints in (3.20) specify, amongst others, that $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$ is of the form $(\bar{u},\bar{u},\dots,\bar{u})$, for some $\bar{u} \in \mathbb{U}$, as depicted in Figure 3.1(b). Note that, for fixed

$M \in \mathbb{N}$, the strategy based on $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$ requires the same amount of data to be transmitted as $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$. The only difference is that in the former the actuators perform a "to hold" action, while in the latter a "to zero" strategy [81] is followed.

Both the strategies based on $\bar{\mathcal{U}}_M(x)$ and $\tilde{\mathcal{U}}_M(x)$ have n_u free control variables, but all elements of the input vector are updated simultaneously. A strategy leading to sparse solutions (instead of group sparse solutions), i.e., allowing each actuator to actuate independently but sporadically, can be obtained by replacing $\bar{\mathcal{U}}_M(x)$ in (3.15) by

$$\hat{\mathcal{U}}_M(x) = \Big\{\mathbf{u} \in \mathbb{U}^M \mid \forall j \in \mathbb{N}_{[1,n_u]},\ \mathrm{nz}(\mathbf{u}, j) \leq 1 \text{ and } \\ \forall i \in \mathbb{N}_{[1,M-1]},\ x_i(x, \mathbf{u}) \in \mathbb{X} \text{ and } x_M(x, \mathbf{u}) \in \mathbb{X}_f\Big\}, \quad (3.21)$$

in which $\mathrm{nz}(\mathbf{u}, j)$ denotes the number of non-zero elements in the set $\{u_0^{(j)}, u_1^{(j)}, \ldots, u_{M-1}^{(j)}\}$, where $\mathbf{u} = (u_0, u_1, \ldots, u_{M-1})$ and $u_i = (u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(n_u)}) \in \mathbb{R}^{n_u}$, $i = 0, 1, \ldots, M-1$. This strategy allows for sparse profiles of the form depicted in Figure 3.1(c), which are relevant in, for instance, control of overactuated systems, see, e.g., [37]. Note that due to the additional degree of freedom in the input update times (compared to group sparse profiles), we now consider $(M+1)^{n_u}$ possible discrete choices of input update patterns. However, for a fixed $M \in \mathbb{N}$, one could solve them in an ordered manner trying first the options with a minimal number of actuators that require a non-zero value. Moreover, for a fixed $M \in \mathbb{N}$, it requires more data to be communicated, as both the control values and the timing information of when the control values are to be implemented should be sent to the actuators.

Although the discussed input profiles address (actuation and/or communication) resource constraints in *all* elements of the input, the framework can also address problems where resource constraints only apply to certain elements of the input (e.g., the case where only one element u_j is transmitted over a wireless link with limited bandwidth, or the case where only the actuator corresponding to input u_j is subject to wear and tear).

For ease of exposition, in the next section we focus on the algorithm discussed in Section 3.3.1, but the resulting derivations apply *mutatis mutandis* to the cases described in this section as well, see Remark 3.2 at the end of Section 3.3.3.

3.3.3 Theoretical properties

The self-triggered MPC law given by Algorithm 3.1 has the following important properties. Recall that the conditions regarding the terminal set and terminal cost (Assumption 3.1) are standing assumptions.

Theorem 3.1. (Recursive feasibility): *Given $\beta \geq 1$, the control law as in Algorithm 3.1 is well posed, i.e., for all $x \in \mathbb{X}_f$, $\mathcal{M}^{\mathrm{st}}(x) \in \mathbb{N}_{\geq 1}$ and $\mathcal{U}^{\mathrm{st}}(x) \neq \emptyset$.*

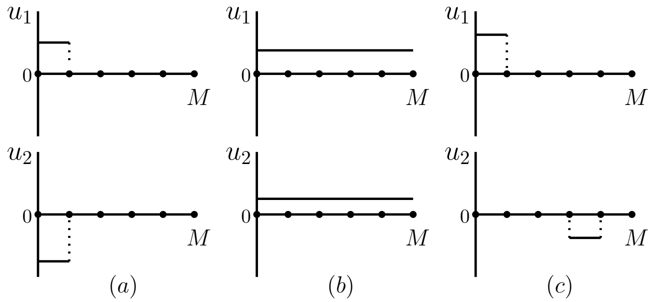

Fig. 3.1. Examples of: (a) $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$, (b) $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$, (c) $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$.

Proof. The proof is based on showing that for each $x \in \mathbb{X}_f$ there exists $u_0 \in \mathbb{U}$ such that

$$\beta \left[V(g(x,u_0)) - V(x)\right] \leq -L(x,u_0) \tag{3.22}$$

and $g(x,u_0) \in \mathbb{X}_f$. Observe that (3.22) is equivalent to the condition (3.16) for $M = 1$. In fact, we are going to show that (3.22) holds for $u_0 = u^{\mathrm{mpc}}(x)$. We will employ that $u^{\mathrm{mpc}}(x)$ was constructed based on (3.3) using the conditions (3.8a)-(3.8c) related to the terminal set and cost method. We take $\mathbf{u}^*(x) := (u_0^*(x),\ u_1^*(x),\ \ldots,\ u_{N-1}^*(x)) \in \mathcal{U}_N(x)$ and thus $u_0^*(x) = u^{\mathrm{mpc}}(x)$, see (3.5) and (3.7). Moreover, we take $\tilde{\mathbf{u}} := \big(u_1^*(x),\ \ldots,\ u_{N-1}^*(x),\ K(x_N(x,\mathbf{u}^*(x)))\big)$ and note that as $\mathbf{u}^*(x) \in \mathcal{U}_N(x)$ we have that $x_N(x,\mathbf{u}^*(x)) \in \mathbb{X}_T$, now we can invoke (3.8b) and (3.8c) to see that $g\left(x_N(x,\mathbf{u}^*(x)), K\left(x_N(x,\mathbf{u}^*(x))\right)\right) \in \mathbb{X}_T$ and $K(x_N(x,\mathbf{u}^*(x))) \in \mathbb{U}$. Hence, we conclude that $\tilde{\mathbf{u}} \in \mathcal{U}_N(g(x,u^{\mathrm{mpc}}(x)))$. By following standard arguments, see, e.g., [67], we have that

$$\begin{aligned}
&V\Big(g\big(x,u^{\mathrm{mpc}}(x)\big)\Big) - V(x) \\
&\quad\leq F\bigg(x_N\Big(g\big(x,u^{\mathrm{mpc}}(x)\big),\tilde{\mathbf{u}}\Big)\bigg) + \sum_{k=0}^{N-1} L\bigg(x_k\Big(g\big(x,u^{\mathrm{mpc}}(x)\big),\tilde{\mathbf{u}}\Big),\Pi_k\tilde{\mathbf{u}}\bigg) \\
&\qquad\quad - F\big(x_N(x,\mathbf{u}^*(x))\big) - \sum_{k=0}^{N-1} L\big(x_k(x,\mathbf{u}^*(x)),\Pi_k\mathbf{u}^*(x)\big), \\
&\quad= F\bigg(g\Big(x_N(x,\mathbf{u}^*(x)),K\big(x_N(x,\mathbf{u}^*(x))\big)\Big)\bigg) - F\big(x_N(x,\mathbf{u}^*(x))\big) \\
&\qquad\quad + L\Big(x_N\big(x,\mathbf{u}^*(x)\big),K\big(x_N(x,\mathbf{u}^*(x))\big)\Big) - L\big(x,u^{\mathrm{mpc}}(x)\big).
\end{aligned} \tag{3.23}$$

As $x_N(x,\mathbf{u}^*) \in \mathbb{X}_T$, we can invoke (3.8a) to obtain from (3.23)

$$V\left(g\left(x,u^{\mathrm{mpc}}(x)\right)\right) - V(x) \leq -L(x,u^{\mathrm{mpc}}(x)). \tag{3.24}$$

Since $\beta \geq 1$, we see that (3.24) implies (3.22) for $u_0 = u^{\mathrm{mpc}}(x)$. Clearly, $u^{\mathrm{mpc}}(x) \in \mathbb{U}$ and since we already established $\tilde{\mathbf{u}} \in \mathcal{U}_N(g(x,u^{\mathrm{mpc}}(x)))$, we also have $g(x, u^{\text{mpc}}(x)) \in \mathbb{X}_f$. Therefore, $u^{\text{mpc}}(x) \in \bar{\mathcal{U}}_1(x)$ and due to (3.24) also $u^{\text{mpc}}(x) \in \mathcal{U}_1^{\text{st}}(x)$. Hence, $\mathcal{U}^{\text{st}}(x) \neq \emptyset$ and $\mathcal{M}^{\text{st}}(x) \in \mathbb{N}_{\geq 1}$ for all $x \in \mathbb{X}_f$. □

Theorem 3.2. (Suboptimality and constraint satisfaction): *Let $\beta \geq 1$. The control law as in Algorithm 3.1 satisfies the performance guarantee given in* (3.10) *as well as the constraints given in* (3.2) *for the closed-loop system* (3.1) *with* (3.17) *for all $x_0 \in \mathbb{X}_f$.*

Proof. Condition (3.12) with the control law in (3.17), yields for $l \in \mathbb{N}$

$$\sum_{t=t_l}^{t_{l+1}-1} L(x_t, u_t) \leq \beta \left[V(x_{t_l}) - V(x_{t_{l+1}})\right] \tag{3.25}$$

and by the definition in (3.15),

$$u_t \in \mathbb{U} \text{ for } t \in \mathbb{N}_{[t_l, t_{l+1})}, x_t \in \mathbb{X} \text{ for } t \in \mathbb{N}_{(t_l, t_{l+1})} \text{ and } x_{t_{l+1}} \in \mathbb{X}_f \subseteq \mathbb{X}. \tag{3.26}$$

Summing (3.25) for $l = 0$ to $L \in \mathbb{N}$ and using $t_0 = 0$ lead to

$$\begin{array}{lll}
l = 0; & \displaystyle\sum_{t=t_0}^{t_1-1} L(x_t, u_t) & \leq \beta\left[V(x_{t_0}) - V(x_{t_1})\right] \\
l = 1; & \displaystyle\sum_{t=t_1}^{t_2-1} L(x_t, u_t) & \leq \beta\left[V(x_{t_1}) - V(x_{t_2})\right] \\
\vdots & \vdots & \vdots \\
l = L; & \displaystyle\sum_{t=t_L}^{t_{L+1}-1} L(x_t, u_t) & \leq \beta\left[V(x_{t_L}) - V(x_{t_{L+1}})\right] \quad + \\
\hline
 & \displaystyle\sum_{t=0}^{t_{L+1}-1} L(x_t, u_t) & \leq \beta\left[V(x_0) - V(x_{t_{L+1}})\right] \leq \beta V(x_0),
\end{array}$$

where in the latter inequality we used that V takes only nonnegative values. For $L \to \infty$, the desired performance guarantee (3.10) is obtained. Since we have (3.26) for all $l \in \mathbb{N}$, also the input and state constraints as in (3.2) are satisfied. □

Theorem 3.3. Stability: *Let $\beta \geq 1$. The control law given by Algorithm 3.1 renders the closed-loop system given by* (3.1) *and* (3.17) *asymptotically stable for initial conditions in $\mathbb{X}_f$.*

Proof. The proof is based on showing that V is a Lyapunov function for the closed-loop system (3.1) and (3.17). We will first show that $V(x) \leq \alpha_2(\|x\|)$ for all $x \in \mathbb{X}_T$, with α_2 as in (3.8e). Let $\tilde{\mathbf{x}} := (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N)$ denote the state sequence generated by the auxiliary system $\tilde{x}_{t+1} = g(\tilde{x}_t, K(\tilde{x}_t))$ starting from $\tilde{x}_0 \in \mathbb{X}_T$ and let $\tilde{\mathbf{u}} = (K(\tilde{x}_0),\ K(\tilde{x}_1),\ \dots,\ K(\tilde{x}_{N-1}))$ be the corresponding input sequence. Note that as $\tilde{x}_0 \in \mathbb{X}_T$, from (3.8b) and (3.8c) we have that $\tilde{\mathbf{x}} \in \mathbb{X}_T^N$ and $\tilde{\mathbf{u}} \in \mathbb{U}^N$. Now, we invoke (3.8a) to obtain

$$\begin{aligned}
F(\tilde{x}_1) - F(\tilde{x}_0) + L(\tilde{x}_0, K(\tilde{x}_0)) &\leq 0,\\
F(\tilde{x}_2) - F(\tilde{x}_1) + L(\tilde{x}_1, K(\tilde{x}_1)) &\leq 0,\\
&\vdots\\
F(\tilde{x}_N) - F(\tilde{x}_{N-1}) + L(\tilde{x}_{N-1}, K(\tilde{x}_{N-1})) &\leq 0,
\end{aligned}$$

and observe that by summing these inequalities we get $-F(\tilde{x}_0) + F(\tilde{x}_N) + \sum_{k=0}^{N-1} L(\tilde{x}_i, K(\tilde{x}_i)) \leq 0$, implying that $J_N(\tilde{x}_0, \tilde{\mathbf{u}}) \leq F(\tilde{x}_0)$ for any $\tilde{x}_0 \in \mathbb{X}_T$. Using this fact and (3.8e), for $\tilde{x}_0 = x \in \mathbb{X}_T$ we have that

$$V(x) \leq J_N(x, \tilde{\mathbf{u}}) \leq F(x) \leq \alpha_2(\|x\|). \tag{3.27}$$

From (3.10) and (3.8d), we have that for all $x \in \mathbb{X}_f$

$$\sum_{t=0}^{\infty} \alpha_1(\|x_t\|) \leq \sum_{t=0}^{\infty} L(x_t, u_t) \leq \beta V(x), \tag{3.28}$$

where x_t is the solution to the closed-loop system (3.1) and (3.17) starting from $x_0 = x \in \mathbb{X}_f$, and u_t as in (3.17b). This immediately implies that we have the attractivity property in $\mathbb{X}_f$, i.e., for all $x_0 \in \mathbb{X}_f$ we have $\lim_{t\to\infty} \|x_t\| = 0$. To establish Lyapunov stability we fix $\delta > 0$ such that $\mathcal{B}_\delta := \{x \in \mathbb{R}^{n_x} \mid \|x\| \leq \delta\} \subseteq \mathbb{X}_T$ and let $\varepsilon > 0$ be such that $\beta\alpha_2(\delta) \leq \alpha_1(\varepsilon)$. Then for $x_0 \in \mathcal{B}_\delta$, from (3.27) and (3.28) we see that

$$\sum_{t=0}^{\infty} \alpha_1(\|x_t\|) \leq \beta V(x_0) \leq \beta\alpha_2(\|x_0\|) \leq \beta\alpha_2(\delta) \leq \alpha_1(\varepsilon), \tag{3.29}$$

hence, $\alpha_1(\|x_t\|) \leq \alpha_1(\varepsilon)$ for all $t \in \mathbb{N}$, i.e., $x_t \in \mathcal{B}_\varepsilon$ for all $t \in \mathbb{N}$. This completes the proof. □

Hence, Theorems 3.1, 3.2 and 3.3 establish the properties required in the problem formulation given in Section 3.2. By maximizing at every execution time t_l the next value t_{l+1}, also significant sparsity properties, and communication reductions (following the variants in Section 3.3.2), will be realized (see the examples in Section 3.5).

Remark 3.2. Regarding the theoretical properties for the variants resulting in infrequently changing input profiles and sparse input profiles, described by $\widetilde{\mathcal{U}}_M(x)$ in (3.20) and $\hat{\mathcal{U}}_M(x)$ in (3.21), respectively, note that for $x \in \mathbb{X}_f$ and $M = 1$ we have that $\bar{\mathcal{U}}_1(x) = \tilde{\mathcal{U}}_1(x) = \hat{\mathcal{U}}_1(x)$. The proof of Theorem 3.1 is based on showing that for $x \in \mathbb{X}_f$ a feasible solution exists for $M = 1$; hence, Theorem 3.1 applies directly to the variants $\tilde{\mathcal{U}}_M(x)$ and $\hat{\mathcal{U}}_M(x)$ as well.

Note that by using $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$ or $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$ in (3.15), we see that (3.26) holds. Moreover, as (3.15) implies that (3.16) is satisfied, the performance guarantee (3.10) can be obtained analogously to the proof of Theorem 3.2.

Both the variants based on $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$ and $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$ satisfy the performance guarantee (3.10), and, moreover, are based on the terminal set and cost method, subject to the conditions given in Assumption 3.1. As a consequence, the stability proof for these variants can be obtained analogously to the proof of Theorem 3.3, by using the properties related to the terminal set and terminal cost method, i.e., (3.8a)-(3.8e), and the performance guarantee (3.10). $\lhd$

3.4 Implementation considerations

In the implementation of the control law in Algorithm 3.1, the evaluation of (3.16) plays a crucial role. Note that the term $V(x_M(x,\mathbf{u}))$ in the right-hand side of (3.16) is a priori unknown due to the fact that it depends on $\mathbf{u}$, which is yet to be constructed. We rearrange terms in (3.16) to observe that each $\mathbf{u} \in \mathcal{U}_M^{\mathrm{st}}(x)$ is characterized by

$$\beta V(x_M(x,\mathbf{u})) + \sum_{k=0}^{M-1} L(x_k(x,\mathbf{u}), \Pi_k \mathbf{u}) \leq \beta V(x), \tag{3.30a}$$

$$\text{and } \mathbf{u} \in \bar{\mathcal{U}}_M(x). \tag{3.30b}$$

To implement Algorithm 3.1, the objective is now for a given $x \in \mathbb{X}_f$ to find the largest $M \in \mathbb{N}$ for which (3.30) is feasible. For fixed $M \in \mathbb{N}$, feasibility of (3.30) for some $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ is equivalent to the minimum of the left-hand side of (3.30a) subject to $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ being smaller than the right-hand side of (3.30a). More formally, $\mathcal{U}_M^{\mathrm{st}}(x) \neq \emptyset$ if and only if

$$\min_{\mathbf{u} \in \bar{\mathcal{U}}_M(x)} \beta V(x_M(x,\mathbf{u})) + \sum_{k=0}^{M-1} L(x_k(x,\mathbf{u}), \Pi_k \mathbf{u}) \leq \beta V(x). \tag{3.31}$$

As we do not have an explicit formula for $V(x_M(x,\mathbf{u}))$, it is unclear how to verify (3.31). To resolve this issue, we replace $V(x_M(x,\mathbf{u}))$ in (3.31) by the corresponding optimization problem (3.4), which effectively results in an extension of the prediction horizon. To perform this replacement, a similar notation to the one in (3.13) is introduced for the horizon $M+N$, leading to

$$\bar{\mathcal{U}}_{M,N}(x) := \Big\{ \mathbf{u} \in \mathbb{U}^{M+N} \mid \forall i \in \mathbb{N}_{[1,M-1]},\ \Pi_i \mathbf{u} = 0,\ \forall j \in \mathbb{N}_{[1,M+N-1]}, \\ x_j(x,\mathbf{u}) \in \mathbb{X},\ \text{and } x_{M+N}(x,\mathbf{u}) \in \mathbb{X}_T \Big\}. \tag{3.32}$$

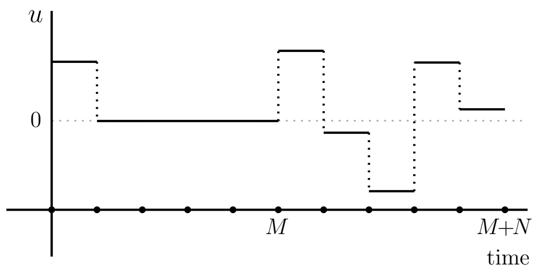

Fig. 3.2. Example of $\mathbf{u} \in \bar{\mathcal{U}}_{M,N}(x)$.

Figure 3.2 shows an example of $\mathbf{u} \in \bar{\mathcal{U}}_{M,N}(x)$.

Replacing $V(x_M(x, \mathbf{u}))$ in (3.31) by the corresponding optimization problem (3.4) and incorporating the notation in (3.32), the implementation of Algorithm 3.1 can be realized as follows.

Algorithm 3.2. *At execution time t_l and state $x_{t_l} = x$, find the largest $M \in \{1, 2, \ldots, \bar{M}\}$, denoted by M^*, such that*

$$\min_{\mathbf{u} \in \bar{\mathcal{U}}_{M,N}(x)} \beta F(x_{N+M}(x, \mathbf{u})) + \beta \sum_{k=M}^{M+N-1} L(x_k(x, \mathbf{u}), \Pi_k \mathbf{u}) + \sum_{k=0}^{M-1} L(x_k(x, \mathbf{u}), \Pi_k \mathbf{u}) \leq \beta V(x), \quad (3.33)$$

holds and let $\mathbf{u}^* \in \bar{\mathcal{U}}_{M^*,N}(x)$ be a corresponding minimizer. The next execution time $t_{l+1} = t_l + M^*$ and $\bar{u}_l = \Pi_0 \mathbf{u}^*$ in (3.11).

Note that in Algorithm 3.2, it has to be verified if the obtained minimizer in the left-hand side is smaller than the right-hand side of the inequality (3.33), i.e., smaller than $\beta V(x)$, where $V(x)$ obtained by solving the MPC problem in (3.4). Hence, the algorithm is based on solving similar nonlinear optimization problems as the original MPC setup (only over a somewhat larger horizon), but with (almost) the same number of free variables.

Comparing the standard MPC setup in Section 3.2.1, which requires using computation, actuation and/or communication resources at each time instant $t \in \mathbb{N}$, with the proposed self-triggered MPC scheme, we see that for the latter scheme the resources are only needed at the execution times t_l, $l \in \mathbb{N}$, while on times $\{t_l + 1, t_l + 2, \ldots, t_{l+1} - 1\}$ no computation, actuation and/or communication resources are required. However, note that at the execution times more computations are needed, i.e., more optimal control problems have to be solved, although the average computational load might be roughly the same as the optimal MPC setup that has to solve optimal control problems at each time instant. In this sense, (more bursty) computations are "traded" for (less) resource utilization (cf. [99]).

In particular, to obtain $M^* = \mathcal{M}^{\text{st}}(x)$ for the fixed upper bound $\bar{M} \in \mathbb{N}_{\geq 1}$, it is necessary to solve $\bar{M}+1$ optimization problems, corresponding to MPC problems with prediction horizons $N, N+1, \ldots, N+\bar{M}$, respectively (having $N+1$ vectors of free control variables). Hence, at time t_l and state $x_{t_l} = x$, one has to solve $\bar{M}+1$ optimization problems, and then the next $\mathcal{M}^{\text{st}}(x)-1$ steps, no computation (and no actuation and communication) resources are needed. This indicates that when $\mathcal{M}^{\text{st}}(x)$ is large and close to $\bar{M}$, the average number of optimization problems be solved is almost the same as for the standard MPC setup given in Section 3.2.1. However, the computations are more "bursty" now (only at t_l, $l \in \mathbb{N}$), but this gives the advantage of obtaining sparse or sporadically changing control signals with constraint satisfaction and guaranteed performance in terms of the *original* MPC cost.

Remark 3.3. An alternative implementation scheme that preserves the properties in Theorems 3.1, 3.2 and 3.3 is obtained by replacing $\mathcal{M}^{\text{st}}(x)$ in (3.18) by

$$\mathcal{M}^{\text{st}}_{\text{incr}}(x) = \left\{ M \in \mathbb{N}_{[1,\bar{M}]} \mid \mathcal{U}^{\text{st}}_m(x) \neq \emptyset,\ m \in \mathbb{N}_{[1,M]} \right\}$$

and $\mathcal{U}^{\text{st}}(x)$ in (3.19) by

$$\mathcal{U}^{\text{st}}_{\text{incr}}(x) = \mathcal{U}^{\text{st}}_{\mathcal{M}^{\text{st}}_{\text{incr}}(x)}(x).$$

This scheme incrementally increases M until (3.33) ceases to hold, and does not search for possibly larger values beyond $\mathcal{M}^{\text{st}}_{\text{incr}}(x)$ for which (3.33) can still be guaranteed. This alternative implementation requires solving $\mathcal{M}^{\text{st}}_{\text{incr}}(x)+1$ optimization problems when in state x, while in the next $\mathcal{M}^{\text{st}}_{\text{incr}}(x)$ time instances no computation, communication and actuation is needed. This indicates that for the incremental scheme that when $\mathcal{M}^{\text{st}}_{\text{incr}}(x)$ is typically large, the average number of optimization problems to be solved is almost the same as the optimal MPC setup in Section 3.2.1. We will illustrate this quantitatively in Section 3.5. ◁

Remark 3.4. In order to deal with the computational burden of standard MPC problems, it is common practice to reduce the degrees of freedom of the optimization problem by restricting the input to be constant over several time-steps. This policy is referred to as "move blocking", see, e.g., [64, 79, 87]. Move blocking is usually applied towards the end of the optimization horizon. In contrast to move blocking, the proposed self-triggered strategy fixes the input at the *beginning* of the optimization horizon (see Figure 3.2), while giving full flexibility towards the end of the horizon. This is an interesting distinction. ◁

3.5 Numerical examples

In order to illustrate the effectiveness of the self-triggered MPC scheme, three numerical examples are presented. In the first example, we consider a linear system subject to state and input constraints and we require $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$ (sporadically changing input profiles). In the second and third example we consider non-linear systems subject to state constraints. In the second example we require $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$ (group sparsity) whereas in the third example we consider a system with two inputs and consider the variant based on $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$ (sparsity). For each of the examples, we consider a quadratic cost function (3.3a) of the form

$$J_N(x, \mathbf{u}) := x_N^\top(x, \mathbf{u}) P x_N(x, \mathbf{u}) + \sum_{k=0}^{N-1} \left(x_k^\top(x, \mathbf{u}) Q x_k(x, \mathbf{u}) + u_k^\top R u_k \right),$$

where $Q \succ 0$, $R \succ 0$ and $P \succ 0$ are weighting matrices.

3.5.1 Self-triggered strategy for $\mathbf{u} \in \tilde{\mathcal{U}}_M(x)$

We consider the open-loop unstable discrete-time linear system dynamics $x_{t+1} = Ax_t + Bu_t$, $t \in \mathbb{N}$, taken from [32], where

$$A = \begin{bmatrix} 0.1 & 1.2 \\ 0.007 & 1.05 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 300 & 200 \\ 0.5 & 0.001 \end{bmatrix}. \tag{3.34}$$

The weighting matrices of the running cost are chosen as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{3.35}$$

and $P = \begin{bmatrix} 1.0002 & 0.0259 \\ 0.0259 & 4.9451 \end{bmatrix}$ is the corresponding solution to the discrete algebraic Riccati equation

$$P = Q + A^\top (P - PB(R + B^\top PB)^{-1} B^\top P)A. \tag{3.36}$$

Additionally, state constraints $-1 \leq x_t^{(1)} \leq 1$ and $-10 \leq x_t^{(2)} \leq 10$, $t \in \mathbb{N}$, and input constraints $-1 \leq u_t^{(i)} \leq 1$, $i = 1, 2$, $t \in \mathbb{N}$, are imposed on the system, where $x_t = [x_t^{(1)} \ \ x_t^{(2)}]^\top$ and $u_t = [u_t^{(1)} \ \ u_t^{(2)}]^\top$. The prediction horizon is fixed to $N = 20$ and the upper bound on the inter-transmission times is set to $\bar{M} = 150$. The terminal set $\mathbb{X}_T$ is obtained by calculating the maximal positively invariant constraint admissible set (in terms of [39,58] the $\mathcal{O}_\infty$ set) for $x_{t+1} = (A+BK)x_t$, with K the optimal unconstrained LQR state feedback gain for weights Q and R as above, i.e., $K = -(R + B^\top PB)^{-1} B^\top PA$. For the given state and input constraints we obtain that $\mathbb{X}_T = \{x \in \mathbb{X} \mid Wx \leq w\}$, where

$$W = \begin{bmatrix} -0.0064 & -1 \\ 0.0064 & 1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad w = \begin{bmatrix} 1.1578 \\ 1.1578 \\ 1.0000 \\ 1.0000 \end{bmatrix}. \tag{3.37}$$

We consider 20 initial conditions randomly and uniformly distributed in $\mathbb{X}_f$. Figure 3.3 shows the time response of the states and inputs for initial condition $x_0 = [0.58 \quad -4.70]$ with $\beta = 1.25$. We observe that even for a slight decrease in performance (i.e., an increase of at most 25% of the guaranteed upper bound on the infinite horizon cost, see (3.10)), we obtain input profiles that indeed change infrequently in time.

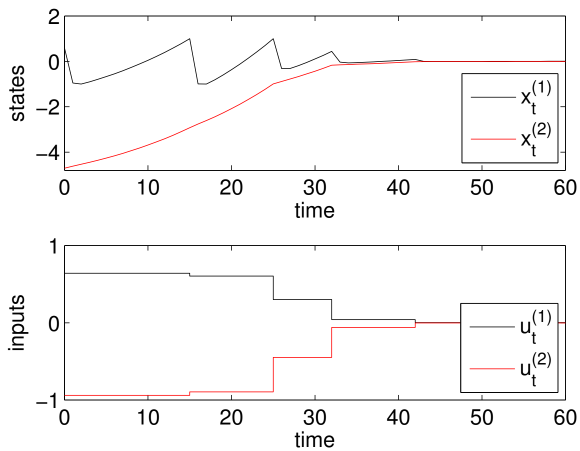

Fig. 3.3. Resulting states and inputs versus time, for $\beta = 1.25$.

Figure 3.4 shows the guaranteed upper bound on the infinite horizon cost and the true cost versus the (average) inter-transmission time for $\beta \in \{1,\ 1.025,\ \dots,\ 1.50\}$ averaged over the 20 initial conditions. The true cost is obtained by evaluating $\sum_{k=0}^{H} x_k^\top Q x_k + u_k^\top R u_k$ over (sufficiently large) simulation horizons H. We see that for a slight decrease in performance (e.g., $\beta = 1.1$) compared to the traditional MPC (with an average inter-transmission time of 1), we obtain an average inter-transmission time M_{avg} of 7.5. By increasing β to 1.45, we observe that we obtain an average inter-transmission time M_{avg} of 71.4. This shows the algorithm's capabilities of providing significant reductions in resource utilization, even at only a slight decrease in performance.

3.5.2 Self-triggered strategy for $\mathbf{u} \in \bar{\mathcal{U}}_M(x)$

We consider an open-loop unstable discrete-time nonlinear system given by

$$g(x_t, u_t) = \begin{cases} x_{t+1}^{(1)} = x_t^{(1)} + x_t^{(2)} + \frac{1}{4}(x_t^{(1)})^2, \\ x_{t+1}^{(2)} = x_t^{(2)} + \frac{1}{4}(x_t^{(2)})^2 + u_t, \end{cases} \tag{3.38}$$

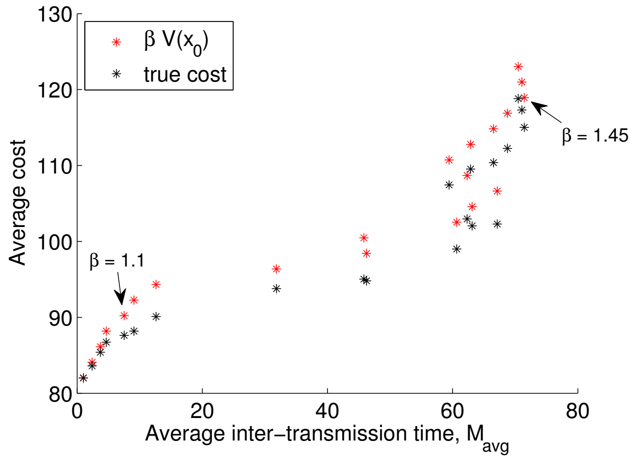

Fig. 3.4. Cost versus the (average) inter-transmission time, for $\beta \in \{1,\ 1.025,\ \ldots,\ 1.50\}$.

where $x_t = [x_t^{(1)}\ \ x_t^{(2)}]^\top$, and the weighting matrices of the running cost are chosen as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad R = 1. \tag{3.39}$$

Additionally, state constraints $-1 \leq x_t^{(i)} \leq 1$, $i = 1, 2$, $t \in \mathbb{N}$, are imposed on the system and the prediction horizon is fixed to $N = 20$. The upper bound on the inter-transmission times is set to $\bar{M} = 50$. The terminal cost and terminal set satisfying Assumption 3.1 are computed based on linearization of the system and upper bounding the linearization error, and are given by $F(x) = x^\top Px$, $x \in \mathbb{R}^{n_x}$, where $P = \begin{bmatrix} 5.8942 & 4.7384 \\ 4.7384 & 9.2263 \end{bmatrix}$, $K = -[0.4221\ \ \ 1.2439]$ and $\mathbb{X}_T = \{x \in \mathbb{X} \mid x^\top Px \leq \alpha\}$, with $\alpha = 0.2967$. We consider 20 initial conditions selected randomly and uniformly distributed in $\mathbb{X}_f$. Figure 3.5 shows the time response of the states and inputs for initial condition $x_0 = [-0.4121\ \ \ 0.9363]^\top$ with $\beta = 1.1$. We observe that even for a slight decrease in performance (i.e., an increase of at most 10% of the guaranteed upper bound on the infinite horizon cost, see (3.10)), we obtain input profiles with significant sparsity.

Figure 3.6 shows the cost versus the (average) inter-transmission time for $\beta \in \{1,\ 1.010,\ \ldots,\ 1.100,\ 1.125,\ \ldots,\ 1.250\}$ averaged over 20 initial conditions. The true infinite horizon cost (cf. (3.10)) is obtained by evaluating $\sum_{k=0}^{H} x_k^\top Qx_k + u_k^\top Ru_k$ over (sufficiently large) simulation horizons H. We see that by increasing β beyond $\beta > 1.125$ the cost increases, but there is no further reduction in resource utilization. However, with at most a 12.5% increase on the infinite

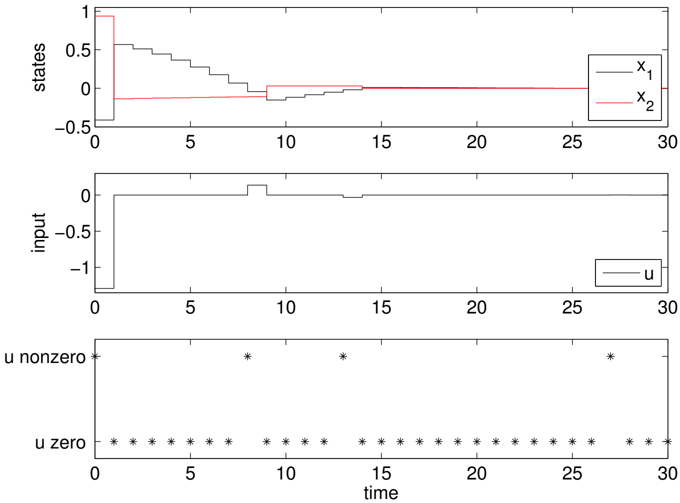

Fig. 3.5. Resulting states and inputs versus time, for $\beta = 1.1$.

horizon cost in (3.10) (i.e., $\beta = 1.125$), we obtain an average inter-transmission time M_{avg} of 12.4. Hence, for this nonlinear example, we considerably reduce the system's resource utilization with only a slight decrease in performance compared to the traditional MPC (with an average inter-transmission time of 1). This shows the algorithm's capabilities of providing significant reductions in resource utilization, even at a slight decrease in performance.

3.5.3 Self-triggered strategy for $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$

We modify system (3.38) by adding a second input to illustrate the effectiveness of the sparsity variant based on $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$ (see (3.21)). Consider the system

$$g(x_t, u_t) = \begin{cases} x_{t+1}^{(1)} = x_t^{(1)} + x_t^{(2)} + \frac{1}{4}(x_t^{(1)})^2 + u_t^{(1)}, \\ x_{t+1}^{(2)} = x_t^{(2)} + \frac{1}{4}(x_t^{(2)})^2 + u_t^{(2)}, \end{cases} \tag{3.40}$$

where $x_t = [x_t^{(1)} \ x_t^{(2)}]^\top$, $u_t = [u_t^{(1)} \ u_t^{(2)}]^\top$, and the weighting matrices of the running cost are chosen as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.41}$$

Additionally, state constraints $-1 \leq x_t^{(i)} \leq 1$, $i = 1, 2$, $t \in \mathbb{N}$, are imposed on the system and the prediction horizon is fixed to $N = 10$. Due to the combinatorial

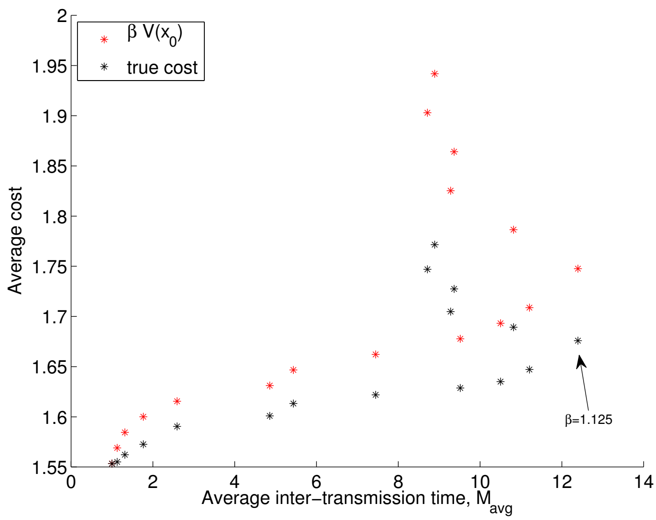

Fig. 3.6. Cost versus the (average) inter-transmission time, for $\beta \in \{1,\ 1.010,\ \ldots,\ 1.100,\ 1.125,\ \ldots,\ 1.250\}$.

nature of the required optimization problem, the upper bound on the inter-transmission times is set to $\bar{M} = 8$. The terminal cost and terminal set satisfying Assumption 3.1 are computed based on linearization of the system and upper bounding the linearization error, and are given by $F(x) = x^\top Px$, $x \in \mathbb{R}^{n_x}$, where $P = \begin{bmatrix} 3.1878 & 1.3464 \\ 1.3464 & 4.8938 \end{bmatrix}$, $K = -\begin{bmatrix} 0.5939 & 0.6732 \\ 0.0793 & 0.7737 \end{bmatrix}$ and $\mathbb{X}_T = \{x \in \mathbb{X} \mid x^\top Px \leq \alpha\}$, with $\alpha = 1.9955$. Figure 3.7 shows the time response of the states and inputs for initial condition $x_0 = [0.8404 \quad -0.5880]^\top$ with $\beta = 1.05$. As we require $\mathbf{u} \in \hat{\mathcal{U}}_M(x)$, this allows each actuator to update independently but sporadically. From Figure 3.7, we observe that the algorithm exploits this additional degree of freedom (compared to group sparse input profiles). Moreover, we again observe that, for a slight decrease in performance (i.e., an increase of at most 5% of the guaranteed upper bound on the infinite horizon cost, see (3.10)), we obtain input profiles with significant sparsity (i.e., 66% of the input values are zero).

3.6 Conclusions

In this chapter, we proposed a self-triggered control strategy for discrete-time nonlinear systems subject to state and input constraints. The proposed strategy has three important features: (i) significant reductions in resource utilization are obtained, (ii) a priori closed-loop performance guarantees are provided

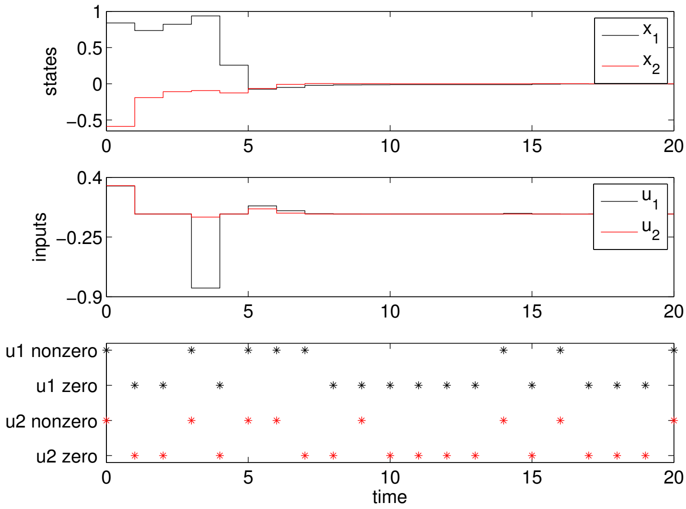

Fig. 3.7. Resulting states, inputs and indication of element-wise input sparsity versus time, for $\beta = 1.05$.

(by design), (iii) co-design of both the feedback law and triggering condition is achieved. Regarding the performance guarantees (feature (ii)), the control laws and triggering mechanisms were designed jointly such that an a priori chosen (suboptimal) level of performance in terms of the original infinite horizon cost function is guaranteed, next to asymptotic stability and constraint satisfaction. Interestingly, the presented framework is flexible in the sense that it can be configured to generate both sparse and sporadically changing input profiles thereby having the ability to serve various application domains in which different resources are scarce or expensive, e.g., communication bandwidth in networked control systems, battery power in wireless control, or fuel in space or underwater vehicles. The effectiveness of the approach was illustrated by means of numerical examples, showing a significant reduction in the usage of the system's resources, without trading much of the guaranteed achievable performance. As such, the proposed approach provides a viable control strategy to balance the usage of the system's resources and control performance.

Chapter 4

Resource-aware MPC for constrained linear systems: Two rollout approaches

Abstract – In systems with resource constraints, such as actuation limitations in sparse control applications or limited bandwidth in networked control systems, it is desirable to use control signals that are either sparse or sporadically changing in time. Therefore, we propose in this work two resource-aware MPC schemes for discrete-time linear systems subject to state and input constraints. The MPC schemes exploit ideas from rollout strategies. Rollout approaches are based on a dynamic programming formulation of the co-design problem of both determining the time instants at which the control actions are updated, and determining the new (continuous) control inputs. The first scheme provides performance guarantees by design, in the sense that it allows the user to select a desired suboptimal level of performance, where the degree of suboptimality provides a trade-off between the guaranteed closed-loop control performance on the one hand and the utilization of (communication/actuation) resources on the other hand. The second scheme provides a guaranteed (average) resource utilization, while cleverly allocating these resources in order to maximize the control performance. By means of numerical examples, we demonstrate the effectiveness of the proposed strategies.

4.1 Introduction

In more and more applications, it becomes essential to address resource constraints explicitly in the design and implementation of the control law. One important domain in which this is apparent is the field of sparse control, where certain control input profiles are preferable from a resource point of view. The use of, for instance, sparse or sporadically changing actuation signals can have several benefits, such as improved fuel efficiency or a larger lifetime of actuators that are subject to wear and tear. For instance, in [44] sparse thrust actuation signals are considered to use fuel in the control of a spacecraft in an efficient manner. In [26] sporadically changing actuation is used in the control of an autonomous underwater vehicle in order to decrease fuel consumption and increase the deployment time. Also in overactuated systems it can be desirable to have a smart control allocation policy that does not require each actuator to be updated continuously in time in order to realize a desired level of performance. One specific example includes the usage of sparse actuation signals to control the roll of an overactuated ship [37].

Another domain in which sparse control solutions are relevant is the domain of networked control systems (NCSs). In such an networked control environment controllers are no longer implemented using dedicated communication channels, but use (shared) communication networks. Since the control task has to share the communication resources with other tasks, the availability of these resources is limited and might even change over time [54]. In these NCSs it is of interest to reduce the number of times the control law is updated to lower the network resource utilization.

In both sparse control applications and NCSs it is of importance to take the resource constraints into account in the controller design. However, despite the fact that resources can be scarce, controllers are typically still implemented in a time-triggered fashion, in which control tasks are executed periodically. This design choice often leads to over-utilization of the available resources, as it might not be necessary to execute the control task every period to guarantee certain closed-loop performance. In designing control strategies for systems with limited actuation and/or communication resources, it is expected that resource-aware control strategies abandon the periodic time-triggered control paradigm and use aperiodic execution of the control tasks.

Two approaches that abandon this periodic communication pattern in NCSs are event-triggered control (ETC) and self-triggered control (STC), see [48] for a recent survey. In ETC and STC, the control law consists of two elements, namely, a feedback controller that computes the control input, and a triggering mechanism that determines when the control input has to be updated. The difference between ETC and STC is that the former is reactive, while the latter is proactive. ETC is reactive in the sense that the triggering consists of verifying a specific condition continuously and when it becomes true, the control task is triggered. In STC the next update time is pre-computed at the current update time based on predictions using previously received data and knowledge on the plant dynamics.

In this chapter we are particularly interested in the design of resource-aware (aperiodic) control strategies for systems subject to constraints on states and inputs. One of the most widely used control strategies for systems with such constraints is model predictive control (MPC). MPC is a form of optimal control based on solving, at each sampling instant, a constrained finite horizon open-loop optimal control problem, using the current state of the plant as the initial state. The optimization yields an optimal control sequence and the first control value in the sequence is applied to the plant [67].

Normally, the receding horizon implementation of MPC is time-triggered. However, several works exist in the literature that address the control of constrained systems with scarce resources. For instance [91] proposed an event-triggered MPC scheme for NCS which can cope with, and in fact, counteract bounded delays and information losses. A self-triggered MPC scheme which maximizes the time until the next control execution while satisfying state and input constraints in the presence of additive disturbances is proposed in [57]. In [35] a self-triggered decentralized MPC framework is presented that aims at reducing the communication between agents, as well as the number of times the agents update their control values. However, the most well known approach is based on modifying the MPC problem by appending the original MPC control cost with an ℓ_1 penalty on the input in order to obtain sparse input signals. Regularizing by the ℓ_1-norm is known to induce sparsity in the sense that, individual components of the input signal will be equal to zero, see, e.g., [6,37,38,73] and [74] in which the focus is on discrete-time and continuous-time linear systems, respectively. Also different types of sum-of-norms regularization can be used to obtain so-called group sparsity, meaning that at many time instants the entire input vector becomes zero, see, e.g., [77].

Although ℓ_1 regularization has proven to be effective in obtaining sparse input profiles, no performance guarantees are given in terms of the original MPC cost function due to the additional penalties included in the cost function. In [53], a self-triggered MPC approach for unconstrained discrete-time linear systems is proposed. The approach in [53] provides sporadically changing input profiles by solving an MPC problem, involving standard quadratic control costs and a penalty related to sampling the system and updating the control law. We have previously proposed a self-triggered MPC framework for constrained linear [15] and nonlinear [41] systems. The approaches in [15] and [41] are such that a reduction in resource utilization can be realized without modifying the cost function by input regularization or explicitly penalizing resource usage. The control laws and triggering mechanisms are synthesized so that a priori chosen performance levels (in terms of the original control cost function) are guaranteed by design next to asymptotic stability and constraint satisfaction.

In this chapter, we propose two stabilizing MPC strategies for systems with state and input constraints that are based on so-called rollout approaches. In rollout algorithms optimal decisions are made along a (finite) lookahead horizon, assuming that, from then on a “base policy” is used for which the cost-to-go is typically simple to compute, see [19]. Exploiting the idea of rollout algorithms in resource-aware control systems was used before in [8], but only for linear systems *without* any constraints. In this chapter, the objective is to present resource-aware MPC strategies that can handle hard constraints on states and inputs. The two novel strategies we propose solve the co-design problem of both determining the time instants on which the updating/communication of the control action take place, and selecting the new (continuous) control inputs. The first approach provides performance guarantees by design (in terms of the original control cost function), in the sense that it allows the user to select a desired suboptimal level of performance. The degree of suboptimality provides a trade-off between the guaranteed closed-loop control performance on the one hand and utilization of resources on the other hand. The second approach provides a guaranteed (average) resource utilization, while cleverly allocating these resources in order to maximize the control performance. We will illustrate the effectiveness of both approaches using a numerical example.

The remainder of this chapter is organized as follows. After indicating the notational conventions used in this paper, in Section 4.2 we provide the problem formulation and modeling preliminaries. For sparse input profiles, we present in Sections 4.3 and 4.4 our rollout approaches with a priori guarantees on control performance and guaranteed resource utilization rate, respectively. In Section 4.5 we discuss how the framework can be used to create sporadically changing input profiles. The effectiveness of the proposed approaches is demonstrated by means of a numerical example in Section 4.6. Section 4.7 contains a discussion on the complexity of the presented algorithms. Finally, in Section 4.8 we present the conclusions.

Nomenclature

By $\mathbb{R}$ and $\mathbb{N}$ we denote the set of real numbers and the set of non-negative integers (including zero), respectively. For $s, t \in \mathbb{N}$, the notation $\mathbb{N}_{[s,t]}$ is used to denote the set $\{r \in \mathbb{N} \mid s \leq r \leq t\}$. The empty set is denoted by $\emptyset$. The inequalities $\prec, \preceq, \succ$ and $\succeq$ are used for matrices, i.e., for a square matrix $X \in \mathbb{R}^{n\times n}$ we write $X \prec 0, X \preceq 0, X \succ 0$ and $X \succeq 0$ if X is symmetric and, in addition, X is negative definite, negative semi-definite, positive definite and positive semi-definite, respectively. Sequences of vectors are indicated by bold letters, e.g., $\mathbf{u} = (u_0, u_1, \dots, u_M)$ with $u_i \in \mathbb{R}^{n_u}, i \in \{0, 1, \dots, M\}$, where $M \in \mathbb{N} \cup \{\infty\}$ will be clear from the context. The projection operators $\Pi_i : (\mathbb{R}^{n_u})^M \to \mathbb{R}^{n_u}$ and $\Pi_{i:j} : (\mathbb{R}^{n_u})^M \to (\mathbb{R}^{n_u})^{j-i+1}$ for $i, j, M \in \mathbb{N}$ with $0 \leq i \leq j \leq M-1$, are defined by $\Pi_i \mathbf{u} := u_i$ and $\Pi_{i:j}\mathbf{u} := (u_i, u_{i+1}, \dots, u_j)$, respectively, for $\mathbf{u} = (u_0, u_1, \dots, u_{M-1}) \in (\mathbb{R}^{n_u})^M$.

4.2 Problem formulation and preliminaries

In this section, we provide the class of systems considered in this work. Moreover, we present two problem formulations that are of interest in designing resource-aware controllers. Finally, in this section we present the modeling preliminaries that are used throughout the remainder of the paper.

4.2.1 System description

In this chapter we consider a discrete-time linear system

$$x_{t+1} = Ax_t + Bu_t, \tag{4.1}$$

where $x_t \in \mathbb{R}^{n_x}$ and $u_t \in \mathbb{R}^{n_u}$ are the state and the input, respectively, at time $t \in \mathbb{N}$. The system (4.1) is subject to input and state constraints given by

$$u_t \in \mathbb{U} \;\text{ and }\; x_t \in \mathbb{X}, \quad t \in \mathbb{N}, \tag{4.2}$$

where $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ and $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ are convex and compact sets containing the origin in their interiors. For $N \in \mathbb{N}$, $x_k(x, \mathbf{u})$ denotes the solution to (4.1) at time $k \in \mathbb{N}_{[0,N]}$ initialized at $x_0 = x$ and with control input sequence given by $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1})$.

4.2.2 Problem formulation for resource-aware control

Standard stabilizing MPC techniques require the use of communication and/or actuation resources at *each* time $t \in \mathbb{N}$ to update the controller, see, e.g., [65,67], which may be undesirable in applications where these resources are limited. In this work, we propose resource-aware controllers that not only compute the control inputs, but also decide at which times $t \in \mathbb{N}$ resources need to be used to update the inputs to the plant u_t in order to guarantee stability, constraint satisfaction and a desired level of performance. To introduce the resource-aware controllers, we introduce $\sigma_t \in \{0, 1\}$, $t \in \mathbb{N}$, as a decision variable, indicating if at time t resources are used to update the controller ($\sigma_t = 1$) or not ($\sigma_t = 0$). We are interested in the design of a policy $\pi := (\mu_0, \mu_1, \ldots)$ which is defined as a sequence of functions $\mu_t := (\mu_t^u, \mu_t^\sigma)$, $t \in \mathbb{N}$, that map the state at each time t into control actions and resource decisions, i.e.,

$$(u_t, \sigma_t) = \mu_t(x_t), \; t \in \mathbb{N}.$$

Performance of the resource-aware controllers will be measured in terms of a multi-objective cost criterion, as it should reflect both the infinite horizon control cost

$$J_{\text{control}}(x_0, \pi) = \sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t, \tag{4.3}$$

for $Q \succ 0$ and $R \succ 0$, and the (average) resource utilization

$$J_{\text{resource}}(x_0, \pi) = \lim_{\gamma \to \infty} \frac{1}{\gamma} \sum_{t=0}^{\gamma-1} \sigma_t. \tag{4.4}$$

In designing resource-aware controllers for system (4.1) subject to the constraints (4.2) we consider two different multi-objective problem formulations, namely

(A) $\min_{\pi} \; J_{\text{resource}}(x_0, \pi)$ such that $J_{\text{control}}(x_0, \pi) \leq c_{\text{control}}(x_0)$,

(B) $\min_{\pi} \; J_{\text{control}}(x_0, \pi)$ such that $J_{\text{resource}}(x_0, \pi) = c_{\text{resource}}$,

for some control specification given by $c_{\text{control}} : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ and resource specification given by $c_{\text{resource}} \in \mathbb{R}_{(0,1]}$. For each problem, we assume that a *periodic* solution exists, which yields a finite cost, i.e., it stabilizes the system (4.1), and satisfies the state and input constraints, i.e., (4.2). Note that for Problem (B), the existence of a feasible periodic solution will boil down to requiring feasibility of a periodic controller that only updates the input once every $\frac{1}{c_{\text{resource}}}$ time instants. Hence, $c_{\text{resource}} \in \mathbb{R}_{(0,1]}$ should be selected large enough to guarantee feasibility (using periodic feedback) for the set of states of interest. Similarly, for Problem (A), the upper bound on the control cost $c_{\text{control}}(x_0)$ should be such that set of states that are feasible for a periodic solution is large enough. These periodic solutions are used as a base policy in our rollout approaches, where we aim to improve over these policies in terms of Problem (A) or (B). The approaches for Problem (A) and (B) are presented in Section 4.3 and Section 4.4, respectively.

4.2.3 Preliminaries

The problems (A) and (B) are intractable due to the (infinite) number of policies over which the optimization takes place. Consequently, the true optimal policy is hard to determine. This work provides approximate solutions by solving a (tractable) finite-horizon problem in a receding horizon fashion. We do this by considering optimization problems over a fixed prediction horizon N and restrict the number of times that resources are used to update the control input within the prediction horizon N. In fact, $N_q \in \mathbb{N}_{[1,N]}$ updates are allowed. To formulate our optimization problems, we require a parameterization of input profiles of length N with N_q control input updates. More specifically, we write $\mathbf{s} = (s_0,\ s_1,\ \ldots,\ s_{N_q-1}) \in \mathbb{S}_{N_q}$, with $\mathbb{S}_{N_q} := \{(s_0, s_1, \ldots, s_{N_q-1}) \in (\mathbb{N}_{[0,N-1]})^{N_q} \mid s_0 < s_1 < \ldots < s_{N_q-1}\}$ to denote the collection of (ordered) time schedules at which the control values are updated within the prediction horizon N and $\mathbf{v} = (v_0,\ v_1,\ \ldots,\ v_{N_q-1}) \in (\mathbb{R}^{n_u})^{N_q}$ to denote the corresponding control values. Moreover, for $\mathbf{s} \in \mathbb{S}_{N_q}$, $N_q \in \mathbb{N}_{[1,N]}$ we define $N_q(\mathbf{s}) := N_q$ as the number of nonzero inputs in the schedule $\mathbf{s}$.

There are many possible strategies for selecting the new control value at the times when no input update is specified, i.e., when $k \in \mathbb{N}_{[0,N-1]} \setminus \{s_0, s_1, \ldots, s_{N_q-1}\}$. A strategy resulting in sparse input profiles is based on setting the control value to zero if $k \in \mathbb{N}_{[0,N-1]} \setminus \{s_0, s_1, \ldots, s_{N_q-1}\}$; this will be referred to as the "*zero*" strategy. Sporadically changing input profiles can be obtained by holding the previously applied control value if no input update is satisfied, i.e., if $k \in \mathbb{N}_{[0,N-1]} \setminus \{s_0, s_1, \ldots, s_{N_q-1}\}$. This will be referred to as the "*hold*" strategy. For ease of exposition, we focus on obtaining sparse input signals using a zero strategy, although sporadically changing input signals can also be pursued, see Section 4.5 for further details.

We introduce now the mappings $M_{N,i} : \mathbb{S}_i \times \mathbb{U}^i \to \mathbb{U}^N$, for $N \in \mathbb{N}$ and $i \in \mathbb{N}_{[1,N]}$, converting the information in $\mathbf{s} \in \mathbb{S}_i$ and $\mathbf{v} \in \mathbb{U}^i$ to obtain the corresponding input sequence $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1}) \in \mathbb{U}^N$ for (4.1). Given $\mathbf{s} = (s_0, s_1, \ldots, s_{i-1}) \in \mathbb{S}_i$ and $\mathbf{v} = (v_0, v_1, \ldots, v_{i-1}) \in \mathbb{U}^i$ we define $\mathbf{u} := M_{N,i}(\mathbf{s}, \mathbf{v}) = (u_0, u_1, \ldots, u_{N-1}) \in \mathbb{U}^N$ with

$$u_k = \begin{cases} v_j, & \text{when } k = s_j \text{ for some } j \in \{0, 1, \ldots, i-1\}, \\ 0, & \text{when } k \in \mathbb{N}_{[0,N-1]} \setminus \{s_0, s_1, \ldots, s_{i-1}\}, \end{cases}$$

for $k \in \mathbb{N}_{[0,N-1]}$.

Standard receding horizon techniques in MPC are based on sampling the state and computing an optimal sequence of control inputs for each $t \in \mathbb{N}$, and applying only the first element of the computed optimal sequence to the plant, after which this procedure is repeated at time $t+1$. We consider a more general receding horizon implementation where the state is sampled and an optimization problem is solved at periodic *scheduling times* $t_l = lH$, $l \in \mathbb{N}$, for some $H \in \mathbb{N}_{[1,N]}$. In Section 4.3 we present an MPC approach for Problem (A) with $H = 1$ and in Section 4.4 we present an MPC approach for Problem (B) for a fixed $H \in \mathbb{N}_{[1,N]}$.

4.3 Problem (A): Guarantees on control performance

In this section, we propose a rollout MPC scheme that employs a one-step lookahead to improve over a standard MPC setup, in terms of optimization problem Problem (A). Our solution is based on sampling the state deciding at each time $t \in \mathbb{N}$ if the next control value can be zero or not (i.e., $H = 1$) and, consequently, each time $t \in \mathbb{N}$ is a scheduling time.

At each (scheduling) time $t \in \mathbb{N}$, given state x_t, our approach aims at reducing the number of times the control input is nonzero by solving an optimization problem based on two different schedules. The two schedules only differ in whether the input in the first step of the prediction horizon is zero or not, assuming resources are used in the remainder of the steps (i.e., the inputs are nonzero). Hence, we consider the set of schedules $\mathbb{S}^{\text{os}} := \{\mathbf{s}^0_{\text{os}}, \mathbf{s}^1_{\text{os}}\}$, where $\mathbf{s}^0_{\text{os}} := (0, 1, \ldots, N-1) \in \mathbb{S}_{N-1}$ and $\mathbf{s}^1_{\text{os}} := (1, 2, \ldots, N-1) \in \mathbb{S}_N$, note that $\mathbb{S}_N = \{\mathbf{s}^0_{\text{os}}\}$.

As mentioned at the end of Section 4.2.2, our approach relies on the existence of a stabilizing periodic solution. In this case, this periodic solution is a standard MPC scheme, which we will introduce in the next section before stating our rollout approach and its theoretical properties in Sections 4.3.2 and 4.3.3, respectively.

4.3.1 A standard stabilizing MPC setup

For the system (4.1), we now consider the MPC setup given by the following optimization problem. For a fixed prediction horizon $N \in \mathbb{N}_{\geq 1}$, given state $x_t = x \in \mathbb{X}$ at time $t \in \mathbb{N}$,

$$\min J_N(x, \mathbf{u}) \tag{4.5a}$$

$$\text{w.r.t. } \mathbf{u} \in \mathcal{U}_N(x) := \Big\{\mathbf{u} \in \mathbb{U}^N \mid \forall k \in \{0, 1, 2, \ldots, N-1\}, \; x_k(x, \mathbf{u}) \in \mathbb{X} \text{ and } x_N(x, \mathbf{u}) \in \mathbb{T}\Big\}. \tag{4.5b}$$

Here

$$J_N(x, \mathbf{u}) := F(x_N(x, \mathbf{u})) + \sum_{k=0}^{N-1} L(x_k(x, \mathbf{u}), \Pi_k \mathbf{u}), \tag{4.6}$$

and $\mathbb{T} \subseteq \mathbb{X}$ is the terminal set, which is assumed to be compact, convex and to contain the origin in its interior. In this work we focus on the case where the running cost $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_{\geq 0}$ is chosen as a quadratic function of the states and inputs and for $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ is given by

$$L(x, u) = x^\top Q x + u^\top R u \tag{4.7}$$

with $Q \succ 0$ and $R \succ 0$. Furthermore, the terminal cost $F : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ is taken for $x \in \mathbb{R}^{n_x}$ as

$$F(x) = x^\top P x, \tag{4.8}$$

where $P \succ 0$ will be defined later in this section. A state $x \in \mathbb{X}$ is said to be feasible for the optimization problem (4.5) if $\mathcal{U}_N(x) \neq \emptyset$, i.e., there is at least one admissible input sequence. The set of feasible states is denoted by $\mathbb{X}_f$, i.e.,

$$\mathbb{X}_f = \{x \in \mathbb{X} \mid \mathcal{U}_N(x) \neq \emptyset\}. \tag{4.9}$$

As N is finite, for $x \in \mathbb{X}_f$, the given conditions ensure the existence of a unique minimizer $\mathbf{u}^*(x) = \big(u_0^*(x), \ldots, u_{N-1}^*(x)\big)$ for the optimization problem (4.5) for given $x \in \mathbb{X}_f$. For $x \in \mathbb{X}_f$, $V(x)$ denotes the corresponding minimum value for the optimization problem (4.5). Hence, $V : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ is the MPC value function given by

$$V(x) := \min_{\mathbf{u} \in \mathcal{U}_N(x)} J_N(x, \mathbf{u}) = J_N(x, \mathbf{u}^*(x)). \tag{4.10}$$

The resulting MPC law $u^{\mathrm{mpc}} : \mathbb{X}_f \to \mathbb{U}$ is defined for $x \in \mathbb{X}_f$ as

$$u^{mpc}(x) := u_0^*(x), \tag{4.11}$$

which is implemented in a receding horizon fashion, leading to the policy

$$u_t = u^{\text{mpc}}(x_t), \tag{4.12}$$
$$\sigma_t = 1, \tag{4.13}$$

for $t \in \mathbb{N}$.

To guarantee recursive feasibility and closed-loop stability we will use the terminal cost and set method [65, 67], which typically assumes for the linear case considered here, that there is a $K \in \mathbb{R}^{n_u \times n_x}$ such that

$$(A + BK)^\top P(A + BK) - P \preceq -K^\top RK - Q, \tag{4.14a}$$
$$(A + BK)\mathbb{T} \subseteq \mathbb{T}, \tag{4.14b}$$
$$K\mathbb{T} \subseteq \mathbb{U} \text{ and } \mathbb{T} \subseteq \mathbb{X}. \tag{4.14c}$$

Here, we take P as the solution to the discrete algebraic Ricatti equation (DARE)

$$P = A^\top PA - (A^\top PB)(R + B^\top PB)^{-1}(B^\top PA) + Q, \tag{4.15}$$

and select

$$K = -(R + B^\top PB)^{-1}B^\top PA. \tag{4.16}$$

Note that P satisfies (4.14a) with equality.

Under these standing assumptions, the optimal MPC law as in (4.12) results in a closed-loop system given by (4.1) and (4.12), which

(i) is recursively feasible in the sense that for each $x_0 \in \mathbb{X}_f$ the corresponding solution to (4.1) and (4.12), denoted by $\{x_t\}_{t\in\mathbb{N}}$, exists for all $t \in \mathbb{N}$;

(ii) is asymptotically stable in $\mathbb{X}_f$, in the sense that

- [Lyapunov stability]: for any $\varepsilon > 0$ there is a $\delta > 0$ such that if $x_0 \in \mathbb{X}_f$ and $\|x_0\| \leq \delta$ then the corresponding trajectory $\{x_t\}_{t\in\mathbb{N}}$ to (4.1) and (4.12) satisfies $\|x_t\| \leq \varepsilon$;
- [attractivity]: for any $x_0 \in \mathbb{X}_f$ the corresponding trajectory $\{x_t\}_{t\in\mathbb{N}}$ to (4.1) and (4.12) satisfies $\lim_{t\to\infty} x_t = 0$;

(iii) satisfies input and state constraints, i.e., (4.2);

(iv) satisfies the performance guarantee

$$\sum_{t=0}^{\infty} \left(x_t^\top Qx_t + u_t^\top Ru_t\right) \leq V(x_0). \tag{4.17}$$

See [65, 67] and the references therein, for the proofs and further details on the terminal set and cost method.

4.3.2 Approach

The rollout strategy proposed in this section aims at reducing the utilization of resources compared to the standard MPC setup in Section 4.3.1, while still meeting certain performance guarantees, next to asymptotic stability and constraint satisfaction. More specifically, we require that

$$\sum_{t=0}^{\infty}(x_t^\top Qx_t + u_t^\top Ru_t) \leq (1+\beta)V(x_0), \tag{4.18}$$

for some $\beta \geq 0$, with $V(x_0)$ in (4.10). Hence, in the formulation of Problem (A) in Section 4.2.2 it holds that $c_{\text{control}} = (1+\beta)V(x_0)$ for $x_0 \in \mathbb{X}_f$. Here, $\beta \geq 0$ is a design parameter that allows to relax the obtained performance guarantee with respect to the performance (4.17) of the stabilizing MPC setup in Section 4.3.1, in order to reduce the utilization of resources.

To realize the constraint (4.18) next to a reduction in resource utilization, we multiply the cost function in (4.6) by $(1+\beta)$ and include a discount for not updating the *first* control value in the prediction horizon. Hence, for $x \in \mathbb{R}^{n_x}$ and for $\mathbf{s} \in \mathbb{S}^{\text{os}}$ and $\mathbf{v} \in \mathbb{U}^{N_q(\mathbf{s})}$ we consider the cost function

$$\begin{aligned} J_N^{\text{os}}(x, M_{N,N_q(\mathbf{s})}(\mathbf{s},\mathbf{v})) = (1+\beta)J_N(x, M_{N,N_q(\mathbf{s})}(\mathbf{s},\mathbf{v})) \\ - \Pi_0\mathbf{s}\beta L(x, \Pi_0 M_{N,N_q(\mathbf{s})}(\mathbf{s},\mathbf{v})). \end{aligned}$$

Note that $\Pi_0\mathbf{s}_{\text{os}}^0 = 0$ and $\Pi_0\mathbf{s}_{\text{os}}^1 = 1$.

For fixed $N \in \mathbb{N}_{\geq 1}$ and given state $x_t = x \in \mathbb{X}$ at time $t \in \mathbb{N}$, we solve

$$\min J_N^{\text{os}}(x, M_{N,N_q(\mathbf{s})}(\mathbf{s},\mathbf{v})) \tag{4.19a}$$

$$\text{w.r.t. } (\mathbf{s},\mathbf{v}) \in \mathcal{SV}_N^{\text{os}}(x), \tag{4.19b}$$

where

$$\mathcal{SV}_N^{\text{os}}(x) = \{\mathbf{s}_{\text{os}}^0\} \times \mathcal{U}_N(x) \cup \mathcal{SV}_{N,N-1}^0(x),$$

and

$$\mathcal{SV}_{N,N-1}^0(x) = \{(\mathbf{s},\mathbf{v}) \in \mathcal{SV}_{N,N-1}(x) \mid \Pi_0\mathbf{s} = 1\},$$

and where for fixed $N \in \mathbb{N}_{\geq 1}$ and $N_q \in \mathbb{N}_{[1,N]}$, and given state $x \in \mathbb{X}$ we define

$$\begin{aligned} \mathcal{SV}_{N,N_q}(x) := \{(\mathbf{s},\mathbf{v}) \in \mathbb{S}_{N_q} \times \mathbb{U}^{N_q} \mid \forall k \in \{0,1,2,\ldots,N-1\}, \\ x_k(x, M_{N,N_q}(\mathbf{s},\mathbf{v})) \in \mathbb{X},\ x_N(x, M_{N,N_q}(\mathbf{s},\mathbf{v})) \in \mathbb{T}\}. \end{aligned} \tag{4.20}$$

Observe that for $x \in \mathbb{R}^{n_x}$,

$$\begin{aligned} \{\mathbf{u} \in \mathbb{U}^N \mid \exists(\mathbf{s},\mathbf{v}) \in \mathcal{SV}_{N,N-1}^0(x), \mathbf{u} = M_{N,N-1}(\mathbf{s},\mathbf{v})\} \\ \subseteq \{\mathbf{u} \in \mathbb{U}^N \mid \exists(\mathbf{s},\mathbf{v}) \in \{\mathbf{s}_{\text{os}}^0\} \times \mathcal{U}_N(x), \mathbf{u} = M_{N,N-1}(\mathbf{s},\mathbf{v})\} = \mathcal{U}_N(x) \end{aligned}$$

and, consequently, the feasible states for (4.19) are

$$\mathbb{X}_f^{\text{os}} := \{x \in \mathbb{X} \mid \mathcal{SV}_N^{\text{os}}(x) \neq \emptyset\} = \mathbb{X}_f.$$

For $x \in \mathbb{X}_f$, a minimizer to (4.19) is known to exist, and we write $(\mathbf{s}_{\text{os}}^*(x), \mathbf{v}_{\text{os}}^*(x))$ to denote a particular one. For $x \in \mathbb{X}_f$, $V^{\text{os}}(x)$ denotes the corresponding value for the optimization problem (4.19). Hence, $V^{\text{os}} : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ is the rollout MPC value function given by

$$V^{\text{os}}(x) = \min_{(\mathbf{s},\mathbf{v}) \in \mathcal{SV}_N^{\text{os}}(x)} J_N^{\text{os}}(x, M_{N,N_q(\mathbf{s})}(\mathbf{s},\mathbf{v})) = J_N^{\text{os}}(x, M_{N,N_q(\mathbf{s}_{\text{os}}^*(x))}(\mathbf{s}_{\text{os}}^*(x), \mathbf{v}_{\text{os}}^*(x))).$$

An optimal sequence of inputs corresponding to $(\mathbf{s}_{\text{os}}^*(x), \mathbf{v}_{\text{os}}^*(x))$ is given for $x \in \mathbb{X}_f$ by

$$\mathbf{u}_{\text{os}}^*(x) := M_{N,N_q(\mathbf{s}_{\text{os}}^*(x))}(\mathbf{s}_{\text{os}}^*(x), \mathbf{v}_{\text{os}}^*(x)). \tag{4.21}$$

For $t \in \mathbb{N}$, the policy for (4.1) is now given by

$$u_t = \Pi_0 \mathbf{u}_{\text{os}}^*(x_t), \tag{4.22a}$$

$$\sigma_t = 1 - \Pi_0 \mathbf{s}_{\text{os}}^*(x_t). \tag{4.22b}$$

4.3.3 Theoretical properties

The rollout MPC policy given by (4.22) has the following important properties.

Theorem 4.1. (Recursive feasibility and constraint satisfaction): *For $\beta \geq 0$ and under the conditions* (4.14) *with K in* (4.16)*, the control law given by* (4.22) *is recursively feasible for all $x_0 \in \mathbb{X}_f$ in the sense that the closed-loop system* (4.1) *and* (4.22) *leads to a trajectory $\{x_t\}_{t\in\mathbb{N}}$ defined for for all $x_0 \in \mathbb{X}_f$. Moreover, the closed-loop system* (4.1) *and* (4.22) *satisfies the constraints* (4.2)*.*

Proof. The proof is based on showing that for each $x \in \mathbb{X}_f$, it holds that $x_1(x, \mathbf{u}_{\text{os}}^*(x)) \in \mathbb{X}_f$.

Let $\mathbf{u}_{\text{os}}^*(x) := (u_{\text{os},0}^*(x), u_{\text{os},1}^*(x), \ldots, u_{\text{os},N-1}^*(x))$ as in (4.21) and $\mathbf{u}_{\text{os}}^*(x) \in \mathcal{U}_N(x)$ and we define $\bar{\mathbf{u}}(x) = (u_{\text{os},1}^*(x), \ldots, u_{\text{os},N-1}^*(x), Kx_N(x, \mathbf{u}_{\text{os}}^*(x)))$. Since $\mathbf{u}_{\text{os}}^*(x) \in \mathcal{U}_N(x)$, we have that $x_N(x, \mathbf{u}_{\text{os}}^*(x)) \in \mathbb{T}$. We invoke the conditions (4.14b) and (4.14c) related to the terminal cost and set method to see that $Kx_N(x, \mathbf{u}_{\text{os}}^*(x)) \in \mathbb{U}$ and $x_1(x_N(x, \mathbf{u}_{\text{os}}^*(x)), Kx_N(x, \mathbf{u}_{\text{os}}^*(x))) \in \mathbb{T}$. Note that both $\mathbf{u}_{\text{os}}^*(x)$ and $\bar{\mathbf{u}}(x)$ contain the same sequence $(u_{\text{os},1}^*(x), \ldots, u_{\text{os},N-1}^*(x))$. Consequently, $\bar{\mathbf{u}}(x) \in \mathcal{U}_N(x_1(x, u_{\text{os},0}^*(x)))$ and thus $(\mathbf{s}_{\text{os}}^0, \bar{\mathbf{u}}(x)) \subseteq \mathcal{SV}_N^{\text{os}}(x_1(x, u_{\text{os},0}^*(x)))$. Hence, we have that $x_1(x, \mathbf{u}_{\text{os}}^*(x)) \in \mathbb{X}_f$, which proves recursive feasibility. Because $u_t = \Pi_0 \mathbf{u}_{\text{os}}^*(x_t)$ for $t \in \mathbb{N}$, we can conclude that the input and state constraints as in (4.2) are satisfied. □

Theorem 4.2. (Performance guarantee): *For $\beta \geq 0$ and given the conditions in* (4.14)*, the closed-loop system given by* (4.1) *and* (4.22) *satisfies the performance guarantee in* (4.18) *for all $x_0 \in \mathbb{X}_f$.*

Proof. Let $\mathbf{u}_{\text{os}}^*(x) = (u_{\text{os},0}^*(x), u_{\text{os},1}^*(x), \dots, u_{\text{os},N-1}^*(x)) \in \mathcal{U}_N(x)$ as in (4.21) and $\bar{\mathbf{u}}(x) = (u_{\text{os},1}^*(x), \dots, u_{\text{os},N-1}^*(x), Kx_N(x, \mathbf{u}_{\text{os}}^*(x))) \in \mathcal{U}_N(x_1(x, \mathbf{u}_{\text{os}}^*(x)))$, where the latter inclusion is established in the proof of Theorem 4.1. Moreover, let $\bar{x}_N := x_N(x, \mathbf{u}_{\text{os}}^*(x)) \in \mathbb{T}$.

To show (4.18), we will use that for $x \in \mathbb{X}_f$

$$(1+\beta)V(x) = J_N^{\text{os}}(x, \mathbf{u}^*(x)) \geq J_N^{\text{os}}(x, \mathbf{u}_{\text{os}}^*(x)) = V^{\text{os}}(x), \tag{4.23}$$

Moreover, we can write for all $x \in \mathbb{X}_f$

$$V^{\text{os}}(x) - V^{\text{os}}(x_1(x, \mathbf{u}_{\text{os}}^*(x))) \geq V^{\text{os}}(x) - J_N^{\text{os}}(x_1(x, \mathbf{u}_{\text{os}}^*(x)), M_{N,N}(\mathbf{s}_N, \bar{\mathbf{u}}(x))). \tag{4.24}$$

The right-hand side of the inequality (4.24) can be rewritten as

$$\begin{aligned}
&V^{\text{os}}(x) - V^{\text{os}}(x_1(x, \mathbf{u}_{\text{os}}^*(x))) \\
&\geq V^{\text{os}}(x) - J_N^{\text{os}}(x_1(x, \mathbf{u}_{\text{os}}^*(x)), M_{N,N}(\mathbf{s}_N, \bar{\mathbf{u}}(x))) \\
&= \big(1 + \beta(1 - \Pi_0 \mathbf{s}_{\text{os}}^*(x))\big) L(x, u_{\text{os},0}^*(x))) \\
&\quad + (1+\beta)\Bigg(\sum_{k=1}^{N-1} L(x_k(x, \mathbf{u}_{\text{os}}^*(x)), u_{\text{os},k}^*(x)) + F(x_N(x, \mathbf{u}_{\text{os}}^*(x))) \\
&\quad - F(x_N(x_1(x, \mathbf{u}_{\text{os}}^*(x)))) - \sum_{k=0}^{N-1} L(x_k(x_1(x, \mathbf{u}_{\text{os}}^*(x)), \Pi_k M_{N,N}(\mathbf{s}_N, \bar{\mathbf{u}}(x))) \Bigg) \\
&= \big(1 + \beta(1 - \Pi_0 \mathbf{s}_{\text{os}}^*(x))\big) L(x, u_{\text{os},0}^*(x))) \\
&\quad + (1+\beta)\big(L(\bar{x}_N, K\bar{x}_N) + F(\bar{x}_N) - F(x_1(\bar{x}_N, K\bar{x}_N))\big) \\
&= \big(1 + \beta(1 - \Pi_0 \mathbf{s}_{\text{os}}^*(x))\big) \big(x^\top Q x + (u_{\text{os},0}^*(x))^\top R u_{\text{os},0}^*(x)\big) \\
&\quad - (1+\beta)\bar{x}_N^\top \big(Q + K^\top R K - P + (A+BK)^\top P (A+BK)\big) \bar{x}_N
\end{aligned}$$

Because $\bar{x}_N = x_N(x, \mathbf{u}_{\text{os}}^*(x)) \in \mathbb{T}$, using (4.14a) related to the terminal cost and set method gives

$$\begin{aligned}
V^{\text{os}}(x) - V^{\text{os}}(x_1(x, \mathbf{u}_{\text{os}}^*(x))) \geq \\
\big(1 + \beta(1 - \Pi_0 \mathbf{s}_{\text{os}}^*(x))\big) \big(x^\top Q x + (u_{\text{os},0}^*(x))^\top R u_{\text{os},0}^*(x)\big).
\end{aligned}$$

Since $\Pi_0 \mathbf{s}_{\text{os}}^0 = 0$, $\Pi_0 \mathbf{s}_{\text{os}}^1 = 1$ and $\beta \geq 0$, using (4.22), for $t \in \mathbb{N}$ it holds that

$$(x_t^{\text{os}})^\top Q x_t^{\text{os}} + (u_t^{\text{os}})^\top R u_t^{\text{os}} \leq V^{\text{os}}(x_t^{\text{os}}) - V^{\text{os}}(x_1^{\text{os}}(x_t, \mathbf{u}_{\text{os}}^*(x_t^{\text{os}}))), \tag{4.25}$$

where $\{x_t^{\text{os}}\}_{t \in \mathbb{N}_{[0,T]}}$ and $\{u_t^{\text{os}}\}_{t \in \mathbb{N}_{[0,T]}}$ are the state and input trajectories produced by the closed-loop system given by (4.1) and (4.22) for some $T \in \mathbb{N}$, with $x_0^{\text{os}} =$

$x_0 \in \mathbb{X}_f$. Summing (4.25) from $t = 0$ to $t = T$ leads to

$$\begin{array}{rcl} x_0^\top Q x_0 + (u_0^{\mathrm{os}})^\top R u_0^{\mathrm{os}} & \leq & V^{\mathrm{os}}(x_0) - V^{\mathrm{os}}(x_1^{\mathrm{os}}) \\ (x_1^{\mathrm{os}})^\top Q x_1^{\mathrm{os}} + (u_1^{\mathrm{os}})^\top R u_1^{\mathrm{os}} & \leq & V^{\mathrm{os}}(x_1^{\mathrm{os}}) - V^{\mathrm{os}}(x_2^{\mathrm{os}}) \\ \vdots & & \vdots \\ (x_T^{\mathrm{os}})^\top Q x_T^{\mathrm{os}} + (u_T^{\mathrm{os}})^\top R u_T^{\mathrm{os}} & \leq & V^{\mathrm{os}}(x_T^{\mathrm{os}}) - V^{\mathrm{os}}(x_{T+1}^{\mathrm{os}}) \\ \hline \sum_{t=0}^{T} \left((x_t^{\mathrm{os}})^\top Q x_t^{\mathrm{os}} + (u_t^{\mathrm{os}})^\top R u_t^{\mathrm{os}} \right) & \leq & V^{\mathrm{os}}(x_0) - V^{\mathrm{os}}(x_{T+1}^{\mathrm{os}}) \leq V^{\mathrm{os}}(x_0), \end{array} +$$

where in the latter inequality we used that V^{os} takes only nonnegative values. Using $V^{\mathrm{os}}(x_0) \leq (1+\beta)V(x_0)$ as established in (4.23), and letting $T \to \infty$, we obtain

$$\sum_{t=0}^{\infty} \left((x_t^{\mathrm{os}})^\top Q x_t^{\mathrm{os}} + (u_t^{\mathrm{os}})^\top R u_t^{\mathrm{os}} \right) \leq (1+\beta)V(x_0), \tag{4.26}$$

which is the performance guarantee in (4.18). □

Theorem 4.3. (Closed-loop stability): *For $\beta \geq 0$, and given the conditions in* (4.14)*, the closed-loop system given by* (4.1) *and* (4.22) *is asymptotically stable for initial conditions in $\mathbb{X}_f$.*

Proof. From (4.26) and $Q \succ 0$ we obtain the existence of a $c_1 \geq 0$ such that for all $x_0 \in \mathbb{X}_f$

$$c_1 \sum_{t=0}^{\infty} \|x_t^{\mathrm{os}}\|^2 \leq \sum_{t=0}^{\infty} \left((x_t^{\mathrm{os}})^\top Q x_t^{\mathrm{os}} + (u_t^{\mathrm{os}})^\top R u_t^{\mathrm{os}} \right) \leq (1+\beta)V(x_0) \tag{4.27}$$

where $\{x_t^{\mathrm{os}}\}_{t\in\mathbb{N}}$ and $\{u_t^{\mathrm{os}}\}_{t\in\mathbb{N}}$ are the state and input trajectories produced by the closed-loop system (4.1) and (4.22) with $x_0^{\mathrm{os}} = x_0 \in \mathbb{X}_f$. From (4.27) we immediately obtain that $\lim_{t\to\infty} x_t^{\mathrm{os}} = 0$. In addition, since we have $V(x) \leq c_2\|x\|^2$ for all $x \in \mathbb{T}$ and since 0 is in the interior of $\mathbb{T}$ we also obtain Lyapunov stability from (4.27). Therefore, the closed-loop system given by (4.1) and (4.22) is asymptotically stable for all $x_0 \in \mathbb{X}_f$. □

Remark 4.1. *(Required feedback)* Note that the approach presented in this section requires access to the state x_t for all $t \in \mathbb{N}$. In the context of NCS, this means lower communication resource utilization is only obtained between the controllers and actuators, whereas the communication resources between the sensors and the controller are still used at each time $t \in \mathbb{N}$. Our solution to Problem (B) also realizes reduced resource utilization between sensors and controller, see Section 4.4. ◁

4.4 Problem (B): Guarantees on resource utilization

In this section, we propose a rollout MPC scheme that addresses Problem (B) with (average) *actuation* resource utilization $c_{\text{resource}} = \frac{1}{q}$, $q \in \mathbb{N}$. The communication resources are used at the scheduling times $t_l = lH$, $l \in \mathbb{N}$ with $H \in \mathbb{N}_{[1,N]}$; hence, the *communication* rate is $\frac{1}{H}$, where we require the following assumption on the variables N, q and H.

Assumption 4.1. *$N \in \mathbb{N}_{\geq 1}$, $q \in \mathbb{N}_{\geq 1}$ and $H \in \mathbb{N}_{\geq 1}$ are such that $N_q := \frac{N}{q} \in \mathbb{N}_{[1,N]}$, $H_q := \frac{H}{q} \in \mathbb{N}_{[1,N]}$ and $N_H := \frac{N}{H} \in \mathbb{N}_{[1,N]}$.*

The proposed rollout MPC scheme is based on the assumption that there exists a stabilizing periodic multi-rate solution ("base policy") to Problem (B) with update rate $\frac{1}{q}$ and exploits ideas from rollout strategies [19] to obtain stability, constraint satisfaction and performance guarantees. Here, that periodic solution is a stabilizing multi-rate MPC scheme that utilizes resources at a (communication and actuation) rate $\frac{1}{q}$, while satisfying state and input constraints for all $t \in \mathbb{N}$, i.e., (4.2). The multi-rate base policy is introduced in the next section. After that we state our rollout MPC approach in Section 4.4.2 and discuss important theoretical properties of our scheme in Section 4.4.3.

4.4.1 Periodic solution to Problem (B): Multi-rate MPC

We consider a multi-rate MPC approach that allows the periodic utilization of resources at a rate $\frac{1}{q}$. The schedule for the multi-rate approach is given by $\bar{\mathbf{s}} = (0, q, 2q, \ldots, (N_q - 1)q) \in \mathbb{S}_{N_q}$ and we define $\mathbb{S}^{\text{mr}}_{N_q} = \{\bar{\mathbf{s}}\} \subseteq \mathbb{S}_{N_q}$.

For fixed N, q and H satisfying Assumption 4.1, and given state $x_{t_l} = x \in \mathbb{X}$ at time $t_l = lH,\ l \in \mathbb{N}$, we consider

$$\min_{\mathbf{v}} J_N(x, M_{N,N_q}(\bar{\mathbf{s}}, \mathbf{v})) \tag{4.28a}$$

$$\text{s.t. } (\bar{\mathbf{s}}, \mathbf{v}) \in \mathcal{SV}^{\text{mr}}_{N,N_q}(x), \tag{4.28b}$$

where

$$\mathcal{SV}^{\text{mr}}_{N,N_q}(x) := \{(\bar{\mathbf{s}}, \mathbf{v}) \in \mathbb{S}^{\text{mr}}_{N_q} \times \mathbb{U}^{N_q} \mid \forall k \in \{0,1,2,\ldots,N-1\},$$
$$x_k(x, M_{N,N_q}(\bar{\mathbf{s}}, \mathbf{v})) \in \mathbb{X},\ x_N(x, M_{N,N_q}(\bar{\mathbf{s}}, \mathbf{v})) \in \mathbb{T}_{\text{mr}}\}$$

and where J_N is given by (4.6), for L in (4.7) with $Q \succ 0$, $R \succ 0$ and where $F(x) = x^\top P_{\text{mr}} x$, for $x \in \mathbb{R}^{n_x}$, where $P_{\text{mr}} = P_{\text{mr}}^\top \succ 0$ is taken as the solution to the following discrete algebraic Riccati equation

$$P_{\text{mr}} = Q_q + (A^q)^\top P_{\text{mr}} A^q -$$
$$((A^q)^\top P_{\text{mr}} B_q + S_q)(R_q + B_q^\top P_{\text{mr}} B_q)^{-1}(B_q^\top P_{\text{mr}} A^q + S_q^\top). \tag{4.29}$$

Here, $B_q := A^{q-1}B$, $Q_q := \sum_{k=0}^{q-1}(A^k)^\top Q A^k$, $S_q := \sum_{k=1}^{q-1}(A^k)^\top Q A^{k-1}B$ and $R_q := (\sum_{k=1}^{q-1} B^\top (A^{k-1})^\top Q A^{k-1} B) + R$, see e.g., [42].

By $\mathbb{T}_{\mathrm{mr}} \subseteq \mathbb{X}$ we denote the terminal set, which is assumed to be compact, convex and to contain the origin in its interior. The set of feasible states is denoted by $\mathbb{X}_f^{\mathrm{mr}}$, i.e.,

$$\mathbb{X}_f^{\mathrm{mr}} = \{x \in \mathbb{X} \mid \mathcal{SV}_{N,N_q}^{\mathrm{mr}}(x) \neq \emptyset\}. \tag{4.30}$$

For $x \in \mathbb{X}_f^{\mathrm{mr}}$, the given conditions ensure that a unique minimizer $\mathbf{v}_{\mathrm{mr}}^*(x) = \left(v_{\mathrm{mr},0}^*(x), v_{\mathrm{mr},1}^*(x), \ldots, v_{\mathrm{mr},N_q-1}^*(x)\right)$ to the optimization problem (4.28) exists. By $V^{\mathrm{mr}}(x)$ we denote the corresponding minimum in (4.28), i.e., $V^{\mathrm{mr}} : \mathbb{X}_f^{\mathrm{mr}} \to \mathbb{R}_{\geq 0}$ is the multi-rate MPC value function given by

$$V^{\mathrm{mr}}(x) = \min\{J_N(x, M_{N,N_q}(\bar{\mathbf{s}}, \mathbf{v})) \mid (\bar{\mathbf{s}}, \mathbf{v}) \in \mathcal{SV}_{N,N_q}^{\mathrm{mr}}(x)\} \tag{4.31}$$

$$= J_N(x, M_{N,N_q}(\bar{\mathbf{s}}, \mathbf{v}^*(x))). \tag{4.32}$$

The corresponding optimal sequence of control inputs for the multi-rate approach is given by

$$\mathbf{u}_{\mathrm{mr}}^*(x) = M_{N,N_q}(\bar{\mathbf{s}}, \mathbf{v}_{\mathrm{mr}}^*(x)). \tag{4.33}$$

For $t \in \mathbb{N}$, the resulting MPC policy for (4.1) is now given by

$$u_t = \Pi_{t-t_l}\mathbf{u}_{\mathrm{mr}}^*(x_{t_l}),\ t \in \mathbb{N}_{[t_l, t_{l+1})},\ l \in \mathbb{N}, \tag{4.34a}$$

$$\sigma_t = \begin{cases} 1, & \text{if } t = kq,\ k \in \mathbb{N}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.34b}$$

To guarantee recursive feasibility and closed-loop stability of this multi-rate MPC scheme we will use a terminal set and cost method. We assume that there is a $K_{\mathrm{mr}} \in \mathbb{R}^{n_x \times n_u}$ such that

$$\bar{A}_q^\top P_{\mathrm{mr}} \bar{A}_q - P_{\mathrm{mr}} \preceq -Q_q - K_{\mathrm{mr}}^\top (S_q)^\top - S_q K_{\mathrm{mr}} - K_{\mathrm{mr}}^\top R_q K_{\mathrm{mr}}, \tag{4.35a}$$

$$\bar{A}_q \mathbb{T}_{\mathrm{mr}} \subseteq \mathbb{T}_{\mathrm{mr}}, \tag{4.35b}$$

$$\bar{A}_k \mathbb{T}_{\mathrm{mr}} \subseteq \mathbb{X},\ k = 1, 2, \ldots, q-1, \tag{4.35c}$$

$$K_{\mathrm{mr}} \mathbb{T}_{\mathrm{mr}} \subseteq \mathbb{U}, \tag{4.35d}$$

where $\bar{A}_k := A^k + \bar{B}_k K_{\mathrm{mr}}$ and $\bar{B}_k := A^{k-1}B$, $k = 1, 2, \ldots, q$. Note that $\bar{B}_q = B_q$. For our approach, we use a particular choice of K_{mr}, namely

$$K_{\mathrm{mr}} = -\left(R_q + B_q^\top P_{\mathrm{mr}} B_q\right)^{-1} \left(B_q^\top P_{\mathrm{mr}} A_q + S_q^\top\right), \tag{4.36}$$

which satisfies (4.35a) with equality.

It is not hard to show that the multi-rate MPC policy proposed in this section provides input profiles using the actuation resources at a rate of $\frac{1}{q}$ and the communication resources at a rate $\frac{1}{H}$, while still satisfying constraints on states and inputs for all $t \in \mathbb{N}$, i.e., (4.2). The schedule of update times $\bar{\mathbf{s}}$ is fixed and as a result, the obtained resource-aware input profiles for the multi-rate MPC scheme are *periodic* with times $t_l = lH$, $l \in \mathbb{N}$, and actuation updates at kq, $k \in \mathbb{N}$.

4.4.2 Approach

In this section, we propose an MPC scheme that addresses variant (B), with $c_{\text{resource}} = \frac{1}{q}$, $q \in \mathbb{N}_{[1,N]}$, which is the same (average) actuation rate as the multi-rate MPC scheme in Section 4.4.1. However, the MPC scheme we propose allows for control update patterns that are *aperiodic*, and in doing so we aim to obtain better performance than the multi-rate scheme at the same update rate $\frac{1}{q}$. To obtain stability, constraint satisfaction and performance guarantees, we will exploit ideas from rollout strategies [19], as already mentioned. For the proposed rollout MPC strategy, we consider the class of schedules

$$\begin{aligned}\mathbb{S}^{\text{ro}}_{N_q,H_q} = \{(s_0, s_1, \ldots, s_{N_q-1}) \in \mathbb{S}_{N_q} \mid s_0 = 0 < s_1 < s_2 < \ldots < s_{H_q-1} < H, \\ \text{and } \forall i \in \{H_q, H_q+1, \ldots, N_q-1\},\ s_i = iq\}. \quad (4.37)\end{aligned}$$

Note that the set of schedules $\mathbb{S}^{\text{ro}}_{N_q,H_q}$ contains all schedules of length N with N_q control updates, where the last $N_q - H_q$ input updates are periodic with period q.

For fixed N, q and $H \in \mathbb{N}_{\geq 1}$ satisfying Assumption 4.1 and given state $x_{t_l} = x \in \mathbb{X}$ at time t_l, $l \in \mathbb{N}$, we consider the following optimization problem

$$\min J_N(x, M_{N,N_q}(\mathbf{s}, \mathbf{v})) \quad (4.38a)$$

$$\text{w.r.t. } (\mathbf{s}, \mathbf{v}) \in \mathcal{SV}^{\text{ro}}_{N,N_q,H_q}(x), \quad (4.38b)$$

where

$$\begin{aligned}\mathcal{SV}^{\text{ro}}_{N,N_q,H_q}(x) := \{(\mathbf{s}, \mathbf{v}) \in \mathbb{S}^{\text{ro}}_{N_q,H_q} \times \mathbb{U}^{N_q} \mid \forall k \in \{0, 1, 2, \ldots, N-1\}, \\ x_k(x, M_{N,N_q}(\mathbf{s}, \mathbf{v})) \in \mathbb{X} \text{ and } x_N(x, M_{N,N_q}(\mathbf{s}, \mathbf{v})) \in \mathbb{T}_{\text{mr}}\}.\end{aligned}$$

Note that $\mathbb{S}^{\text{mr}}_{N_q} \subseteq \mathbb{S}^{\text{ro}}_{N_q,H_q}$, and that, as a consequence, for $x \in \mathbb{X}$, it holds that $\mathcal{SV}^{\text{mr}}_{N,N_q}(x) \subseteq \mathcal{SV}^{\text{ro}}_{N,N_q,H_q}(x)$. Moreover,

$$\mathbb{X}^{\text{mr}}_f \subseteq \mathbb{X}^{\text{ro}}_f := \{x \in \mathbb{X} \mid \mathcal{SV}^{\text{ro}}_{N,N_q,H_q}(x) \neq \emptyset\}.$$

For $x \in \mathbb{X}^{\text{ro}}_f$, a combination of an optimal sequence of schedules and control values is known to exist, and we denote $\mathbf{s}^*_{\text{ro}}(x) = \left(s^*_{\text{ro},0}(x), s^*_{\text{ro},1}(x), \ldots, s^*_{\text{ro},N_q-1}(x)\right)$ and $\mathbf{v}^*_{\text{ro}}(x) = \left(v^*_{\text{ro},0}(x), v^*_{\text{ro},1}(x), \ldots, v^*_{\text{ro},N_q-1}(x)\right)$ as a particular one, i.e.,

$$(\mathbf{s}^*_{\text{ro}}(x), \mathbf{v}^*_{\text{ro}}(x)) \in \operatorname*{argmin}_{(\mathbf{s},\mathbf{v}) \in \mathcal{SV}^{\text{ro}}_{N,N_q,H_q}(x)} J_N(x, M_{N,N_q}(\mathbf{s}, \mathbf{v})). \quad (4.39)$$

For $x \in \mathbb{X}_f^{\mathrm{ro}}$, we use $V^{\mathrm{ro}}(x)$ to denote the corresponding minimum of (4.38). Hence, $V^{\mathrm{ro}} : \mathbb{X}_f^{\mathrm{ro}} \to \mathbb{R}_{\geq 0}$ is the MPC value function given by

$$V^{\mathrm{ro}}(x) = \min_{(\mathbf{s},\mathbf{v}) \in \mathcal{SV}^{\mathrm{ro}}_{N,N_q,H_q}(x)} J_N(x, M_{N,N_q}(\mathbf{s},\mathbf{v})) = J_N(x, M_{N,N_q}(\mathbf{s}^*_{\mathrm{ro}}(x), \mathbf{v}^*_{\mathrm{ro}}(x))). \tag{4.40}$$

The optimal sequence of control inputs corresponding to the optimal schedule $\mathbf{s}^*_{\mathrm{ro}}(x) = (s^*_{\mathrm{ro},0}(x), s^*_{\mathrm{ro},1}(x), \ldots, s^*_{\mathrm{ro},N_q-1}(x))$ is given by

$$\mathbf{u}^*_{\mathrm{ro}}(x) := M_{N,N_q}(\mathbf{s}^*_{\mathrm{ro}}(x), \mathbf{v}^*_{\mathrm{ro}}(x)). \tag{4.41}$$

As $x \in \mathbb{X}_f^{\mathrm{mr}} \subseteq \mathbb{X}_f^{\mathrm{ro}}$ and $\mathcal{SV}^{\mathrm{mr}}_{N,N_q}(x) \subseteq \mathcal{SV}^{\mathrm{ro}}_{N,N_q,H_q}(x)$, it holds that

$$V^{\mathrm{ro}}(x) \leq V^{\mathrm{mr}}(x). \tag{4.42}$$

For $t \in \mathbb{N}$, the resulting MPC policy for (4.1) is now given for $t \in \mathbb{N}_{[t_l, t_{l+1})}$ with $l \in \mathbb{N}$ by

$$u_t = \Pi_{t-t_l} \mathbf{u}^*_{\mathrm{ro}}(x_{t_l}), \tag{4.43a}$$

$$\sigma_t = \begin{cases} 1, & \text{if } t - t_l = s^*_{\mathrm{ro},k}(x_{t_l}) \text{ for some } k = 0, 1, \ldots, N_q - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{4.43b}$$

4.4.3 Theoretical properties

The rollout MPC policy given by (4.43) has the following important properties.

Theorem 4.4. (Recursive feasibility and constraint satisfaction): *Under Assumption 4.1 and the conditions* (4.35) *with* K_{mr} *as in* (4.36), *it holds that the control law given by* (4.43) *is recursively feasible for all* $x \in \mathbb{X}_f^{\mathrm{ro}}$ *in the sense that the closed-loop system* (4.1) *and* (4.43) *leads to a trajectory* $\{x_t\}_{t \in \mathbb{N}}$ *defined for all* $t \in \mathbb{N}$. *Moreover, the closed-loop system* (4.1) *and* (4.43) *satisfies the constraints* (4.2).

Proof. The proof is based on showing that for each $x \in \mathbb{X}_f^{\mathrm{mr}}$, it holds that $x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)) \in \mathbb{X}_f^{\mathrm{ro}}$.

Let $\hat{\mathbf{s}} = (0, q, \ldots, (H_q-1)q) \in \mathbb{S}^{\mathrm{mr}}_{H_q}$ and $\hat{\mathbf{v}}(x) = (K_{\mathrm{mr}}\hat{x}_0, K_{\mathrm{mr}}\hat{x}_1, \ldots, K_{\mathrm{mr}}\hat{x}_{H_q-1}) \in (\mathbb{R}^{n_u})^{H_q}$, where $\hat{x}_0 := x_N(x, \mathbf{u}^*_{\mathrm{ro}}(x))$ and $\hat{x}_{i+1} = \bar{A}_q \hat{x}_i$, $i = 0, 1, \ldots, H_q - 2$. We denote $\tilde{\mathbf{s}}(x) = (\Pi_{H:N-1}\mathbf{s}^*_{\mathrm{ro}}(x), \hat{\mathbf{s}})$ and $\tilde{\mathbf{v}}(x) = (\Pi_{H:N-1}\mathbf{v}^*_{\mathrm{ro}}(x), \hat{\mathbf{v}}(x))$.

We start by showing that $(\tilde{\mathbf{s}}(x), \tilde{\mathbf{v}}(x)) \in \mathcal{SV}^{\mathrm{ro}}_{N,N_q,H_q}(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)))$. As $(\mathbf{s}^*_{\mathrm{ro}}(x), \mathbf{v}^*_{\mathrm{ro}}(x)) \in \mathcal{SV}^{\mathrm{ro}}_{N,N_q,H_q}(x)$, we have that $\hat{x}_0 = x_N(x, \mathbf{u}^*_{\mathrm{ro}}(x)) \in \mathbb{T}_{\mathrm{mr}}$. Due to (4.35b), this implies $\hat{x}_i \in \mathbb{T}_{\mathrm{mr}}$, $i = 1, 2, \ldots, H_q-1$, and, consequently, $\hat{\mathbf{v}}(x) \in \mathbb{U}^H$, using (4.35d) and the fact that the origin is contained in $\mathbb{U}$. As $\hat{x}_i \in \mathbb{T}_{\mathrm{mr}}$, $i = 0, 1, \ldots, H_q-1$, we can invoke (4.35c) to see that $x_j(\hat{x}_i, M_{q,1}(0, K_{\mathrm{mr}}\hat{x}_i)) = \bar{A}_j \hat{x}_i \in \mathbb{X}$, $j = 1, 2, \ldots, q-1$, $i = 0, 1, \ldots, H_q - 1$. Lastly, note that $x_N(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)), M_{N,N_q}(\tilde{\mathbf{s}}(x), \tilde{\mathbf{v}}(x)) = x_H(\hat{x}_0, M_{H,H_q}(\hat{\mathbf{s}}, \hat{\mathbf{v}}(x)) = (\bar{A}_q)^{H_q}\hat{x}_0 \in \mathbb{T}_{\mathrm{mr}}$ due to (4.35b).

Hence, $(\tilde{\mathbf{s}}(x), \tilde{\mathbf{v}}(x)) \in \mathcal{SV}^{\mathrm{ro}}_{N,N_q,H_q}(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)))$ and thus $x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)) \in \mathbb{X}^{\mathrm{ro}}_f$. Because $x_0 \in \mathbb{X}^{\mathrm{ro}}_f$ and $u_t = \Pi_{t-t_l}\mathbf{u}^*_{\mathrm{ro}}(x_{t_l})$, for $t \in \mathbb{N}_{[t_l,t_{l+1})}$, $l \in \mathbb{N}$, we can conclude that the input and state constraints as in (4.2) are satisfied. □

Theorem 4.5. (Closed-loop stability) *Under Assumption 4.1 and the conditions* (4.35) *with* K_{mr} *as in* (4.36)*, the closed-loop system given by* (4.1) *and* (4.43) *is asymptotically stable for initial conditions in* $\mathbb{X}^{\mathrm{ro}}_f$*.*

Proof. Let $x = x_0$ and $\hat{\mathbf{u}}_i(x) = (K_{\mathrm{mr}}\hat{x}_i, 0, 0, \ldots, 0) \in (\mathbb{R}^{n_u})^q$, $i = 0, 1, \ldots, H_q - 1$, where $\hat{x}_0 := x_N(x, \mathbf{u}^*_{\mathrm{ro}}(x))$ and $\hat{x}_{i+1} = x_q(\hat{x}_i, \hat{\mathbf{u}}_i(x)) = \bar{A}_q\hat{x}_i$, $i = 0, 1, \ldots, H_q - 2$. We denote $\hat{\mathbf{u}}(x) = (\hat{\mathbf{u}}_0(x), \hat{\mathbf{u}}_1(x), \ldots, \hat{\mathbf{u}}_{H_q-1}(x))$ and $\tilde{\mathbf{u}}(x) = (\Pi_{H:N-1}\mathbf{u}^*_{\mathrm{ro}}(x), \hat{\mathbf{u}}(x))$ as in the proof of Theorem 4.1.

To show asymptotic stability, we will first study $V^{\mathrm{ro}}(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x))) - V^{\mathrm{ro}}(x) \le 0$ for $x \in \mathbb{X}^{\mathrm{ro}}_f$. We have that

$$
\begin{aligned}
&V^{\mathrm{ro}}(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x))) - V^{\mathrm{ro}}(x) \\
&\le J_N(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \tilde{\mathbf{u}}(x)) - V^{\mathrm{ro}}(x) \\
&= F(x_N(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \tilde{\mathbf{u}}(x))) + \sum_{k=0}^{N-1} L(x_k(x_H(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \tilde{\mathbf{u}}(x)), \Pi_k\tilde{\mathbf{u}}(x)) \\
&\qquad - F(\hat{x}_0) - \sum_{k=0}^{N-1} L(x_k(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \Pi_k\mathbf{u}^*_{\mathrm{ro}}(x)) \\
&= F(x_H(\hat{x}_0, \hat{\mathbf{u}}(x)) + \sum_{k=0}^{H-1} L(x_k(\hat{x}_0, \hat{\mathbf{u}}(x)), \Pi_k\hat{\mathbf{u}}(x)) \\
&\qquad - F(\hat{x}_0) - \sum_{k=0}^{H-1} L(x_k(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \Pi_k\mathbf{u}^*_{\mathrm{ro}}(x)) \\
&= -\sum_{k=0}^{H-1} L(x_k(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \Pi_k\mathbf{u}^*_{\mathrm{ro}}(x)) + \hat{x}_0^\top \Bigg((\bar{A}_q^\top)^{H_q} P \bar{A}_q^{H_q} - P_{\mathrm{mr}} \\
&\qquad + \sum_{i=0}^{H_q-1} \Big((\bar{A}_q^\top)^i \left(Q_q + K_{\mathrm{mr}}^\top S_q^\top + S_q K_{\mathrm{mr}} + K_{\mathrm{mr}}^\top R_q K_{\mathrm{mr}}\right) \bar{A}_q^i \Big) \Bigg) \hat{x}_0 \\
&\le -\sum_{k=0}^{H-1} L(x_k(x, \mathbf{u}^*_{\mathrm{ro}}(x)), \Pi_k\mathbf{u}^*_{\mathrm{ro}}(x)),
\end{aligned} \tag{4.44}
$$

where in the last step we used that

$$
\begin{aligned}
&(\bar{A}_q^\top)^{H_q} P_{\mathrm{mr}} \bar{A}_q^{H_q} - P_{\mathrm{mr}} \preceq \\
&\qquad - \sum_{i=0}^{H_q-1} \Big((\bar{A}_q^\top)^i \left(Q_q + K_{\mathrm{mr}}^\top S_q^\top + S_q K_{\mathrm{mr}} + K_{\mathrm{mr}}^\top R_q K_{\mathrm{mr}}\right) \bar{A}_q^i \Big),
\end{aligned} \tag{4.45}
$$

which can be derived from (4.35a) by pre-multiplication with $(\bar{A}_q^\top)^i$ and post-multiplication with $\bar{A}_q^i$ and observing that

$$(\bar{A}_q^\top)^{i+1} P_{\mathrm{mr}} \bar{A}_q^{i+1} - (\bar{A}_q^\top)^i P_{\mathrm{mr}} \bar{A}_q^i \preceq - (\bar{A}_q^\top)^i \left(Q_q + K_{\mathrm{mr}}^\top (S_q)^\top + S_q K_{\mathrm{mr}} + K_{\mathrm{mr}}^\top R_q K_{\mathrm{mr}}\right) \bar{A}_q^i, \quad (4.46)$$

for $i \in \{0, 1, \ldots, H_q - 1\}$. By summing (4.46) from 0 to $H_q - 1$ we obtain indeed (4.45).

Given $x_{t_l} \in \mathbb{X}_f^{\mathrm{ro}}$, note that $x_{t_{l+1}} = x_H(x_{t_l}, \mathbf{u}_{\mathrm{ro}}^*(x_{t_l}))$ and thus from (4.44) we obtain

$$V^{\mathrm{ro}}(x_{t_l}) - V^{\mathrm{ro}}(x_{t_{l+1}}) \geq \sum_{k=0}^{H-1} L(x_k(x_{t_l}, \mathbf{u}_{\mathrm{ro}}^*(x_{t_l})), \Pi_k \mathbf{u}_{\mathrm{ro}}^*(x_{t_l})). \quad (4.47)$$

Summing (4.47) from $l = 0$ to L and using $t_0 = 0$ we obtain

$$\begin{aligned} V^{\mathrm{ro}}(x_0) - V^{\mathrm{ro}}(x_{t_{L+1}}) &\geq \sum_{l=0}^{L} \sum_{k=0}^{H-1} L(x_k(x_{t_l}, \mathbf{u}_{\mathrm{ro}}^*(x_{t_l})), \Pi_k \mathbf{u}_{\mathrm{ro}}^*(x_{t_l})), \\ &= \sum_{t=0}^{LH-1} ((x_t^{\mathrm{ro}})^\top Q x_t^{\mathrm{ro}} + (u_t^{\mathrm{ro}})^\top R u_t^{\mathrm{ro}}), \quad t \in \mathbb{N}, \end{aligned}$$

where $\{x_t^{\mathrm{ro}}\}_{t\in\mathbb{N}}$ and $\{u_t^{\mathrm{ro}}\}_{t\in\mathbb{N}}$ are the state and input trajectories produced by the closed-loop system (4.1) and (4.43) for $x_0^{\mathrm{ro}} = x_0$. Taking the limit $L \to \infty$ and using the fact that $V^{\mathrm{ro}}(x)$ only takes nonnegative values, we obtain

$$V^{\mathrm{ro}}(x_0) \geq \sum_{t=0}^{\infty} ((x_t^{\mathrm{ro}})^\top Q x_t^{\mathrm{ro}} + (u_t^{\mathrm{ro}})^\top R u_t^{\mathrm{ro}}). \quad (4.48)$$

As $Q, R \succ 0$, this proves for all $x_0^{\mathrm{ro}} \in \mathbb{X}_f^{\mathrm{ro}}$ that $\lim_{t\to\infty} x_t^{\mathrm{ro}} = 0$. In addition, since we have $V^{\mathrm{ro}}(x) \leq c\|x\|^2$ for $x \in \mathbb{T}_{\mathrm{mr}}$ and since 0 is in the interior of $\mathbb{T}_{\mathrm{mr}}$ we also obtain Lyapunov stability from (4.48). Therefore, the closed-loop system given by (4.1) and (4.22) is asymptotically stable for all $x_0 \in \mathbb{X}_f$. □

Theorem 4.6. (Performance) *Under Assumption 4.1 and the conditions* (4.35) *with* P_{mr} *taken as the solution to* (4.29) *and* K_{mr} *as in* (4.36) *it holds that*

$$\sum_{t=0}^{\infty} L(x_t^{\mathrm{ro}}, u_t^{\mathrm{ro}}) \leq \sum_{t=0}^{\infty} L(x_t^{\mathrm{mr}}, u_t^{\mathrm{mr}}) = x_0^\top P_{\mathrm{mr}} x_0, \quad (4.49)$$

for $x_0 \in \mathbb{T}_{\mathrm{mr}}$*, where* $\{x_t^{\mathrm{mr}}\}_{t\in\mathbb{N}}$*,* $\{u_t^{\mathrm{mr}}\}_{t\in\mathbb{N}}$ *and* $\{x_t^{\mathrm{ro}}\}_{t\in\mathbb{N}}$*,* $\{u_t^{\mathrm{ro}}\}_{t\in\mathbb{N}}$ *are the state and input trajectories produced by the closed-loop system* (4.1) *and* (4.34)*, and the closed-loop system* (4.1) *and* (4.43)*, respectively, for initial state* $x_0 \in \mathbb{T}_{\mathrm{mr}}$*.*

Proof. Note that with P_{mr} taken as the solution to (4.29) and K_{mr} as in (4.36), we have that (4.35a) holds with equality. The first inequality in (4.49) can be obtained by combining (4.42) and (4.48) for $x_0 \in \mathbb{X}_f^{\text{mr}}$ to obtain

$$\sum_{t=0}^{\infty}((x_t^{\text{ro}})^\top Q x_t^{\text{ro}} + (u_t^{\text{ro}})^\top R u_t^{\text{ro}}) \leq V^{\text{ro}}(x_0) \leq V^{\text{mr}}(x_0).$$

As $x_0 \in \mathbb{T}_{mr}$, we can use the conditions (4.35) (where (4.35a) holds with equality) with K_{mr} as in (4.36) to see that $V^{\text{mr}}(x_0) = \sum_{t=0}^{\infty} L(x_t^{\text{mr}}, u_t^{\text{mr}}) = x_0^\top P_{\text{mr}} x_0$. □

Hence, in terms of Problem (B), both the proposed rollout MPC approaches and the multi-rate MPC approach provide a solution for $c_{\text{resource}} = \frac{1}{q}$, however, the proposed rollout MPC approach provides a guaranteed upper bound on the control cost J_{control} that is not worse than the upper bound provided by the multi-rate MPC approach. Moreover, for $x_0 \in \mathbb{T}_{\text{mr}}$ the control cost J_{control} of the proposed rollout MPC approach cost is less than, or equal to the control cost of the multi-rate MPC scheme.

Remark 4.2. One can impose restrictions on the set of schedules $\mathbb{S}_{N_q,H_q}^{\text{ro}}$, for instance for computational reasons. In fact, all theorems in this section are also valid for a reduced set of schedules $\mathbb{S}_{N_q,H_q}^{\text{ro,red}}$ that satisfies $\mathbb{S}_{N_q}^{\text{mr}} \in \mathbb{S}_{N_q,H_q}^{\text{ro,red}} \subseteq \mathbb{S}_{N_q,H_q}^{\text{ro}}$, see Section 4.7 for a discussion on the computational complexity of the presented algorithms. ◁

4.5 Sporadically changing input profiles

So far, the focus has been on obtaining sparse input profiles. However, the rollout approaches in both Section 4.3 and 4.4 can also be used to obtain sporadically changing input profiles. As mentioned before, these sporadically changing input profiles can be obtained by employing a "hold" strategy between input updates. For the hold strategy we introduce now the mappings $M_{N,i}^{\text{hold}} : \mathbb{U} \times \mathbb{S}_i \times \mathbb{U}^i \to \mathbb{U}^N$, for $N \in \mathbb{N}$ and $i \in \mathbb{N}_{[1,N]}$, converting the information in $v_{\text{old}} \in \mathbb{U}$, $\mathbf{s} \in \mathbb{S}_i$ and $\mathbf{v} \in \mathbb{U}^i$ to obtain the corresponding input sequence $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1}) \in \mathbb{U}^N$ for (4.1) based on the hold strategy. Given $v_{\text{old}} \in \mathbb{U}$, $\mathbf{s} = (s_0, s_1, \ldots, s_{i-1}) \in \mathbb{S}_i$ and $\mathbf{v} = (v_0, v_1, \ldots, v_{i-1}) \in \mathbb{U}^i$ we define $\mathbf{u} := M_{N,i}^{\text{hold}}(v_{\text{old}}, \mathbf{s}, \mathbf{v}) = (u_0, u_1, \ldots, u_{N-1}) \in \mathbb{U}^N$ with

$$u_k = \begin{cases} v_{\text{old}}, & \text{when } k < s_0, \\ v_j, & \text{when } s_j \leq k < s_{j+1}, \text{ for } j \in \{0, 1, \ldots, i-1\}, \end{cases}$$

for $k \in \mathbb{N}_{[0,N-1]}$, and some $v_{\text{old}} \in \mathbb{R}^{n_u}$. Note that for the hold strategy it is necessary to keep track of the last input value, as this value is to be applied until the first update s_0 in the schedule $\mathbf{s}$.

The base policy used in Section 4.4 is a periodic policy with $c_{\text{resource}} = \frac{1}{q}$. In obtaining sparse control profiles using the zero strategy, this implies that the input profile contains one nonzero control value every q time steps. However, when aiming for sporadically changing input profiles using a hold strategy, this base policy is a periodic policy updating the control value once every q steps, but keeping the value constant between updates. For the hold strategy, the parameters $B_q := \sum_{k=0}^{i-1} A^k B$, $R_q := (\sum_{k=1}^{q-1} B^\top \sum_{i=0}^{k-1} (A^i)^\top Q \sum_{i=0}^{k-1} (A^i) B + R) + R$ and $S_q := \sum_{k=1}^{q-1} (A^k)^\top Q \sum_{i=0}^{k-1} A^i B$ are to be used for obtaining P_{mr} from the Riccati equation (4.29) and K_{mr} in (4.36). Moreover, for the conditions related to the terminal set and terminal cost, i.e., (4.35), $\bar{A}_k := A^k + \bar{B}_k K_{\text{mr}}$ where $\bar{B}_k := \sum_{i=0}^{k-1} A^k B$, $k = 1, 2, \ldots, q$. Note that no changes are needed for the base policy used in Section 4.3, as it is based on a standard MPC setup that updates at every time.

The theoretical properties derived in Sections 4.3 and 4.4 also apply (mutatis mutandis) for our strategies when sporadically changing inputs are used. In the next section, we demonstrate the effectiveness of our rollout approaches in obtaining both sparse and sporadically changing input profiles.

4.6 Numerical examples

To illustrate the effectiveness of the rollout approaches proposed in this chapter, we consider an open-loop unstable discrete-time system as in (4.1) with

$$A = \begin{bmatrix} 1.0221 & 0.25 \\ -0.01 & 0.9988 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Additionally, state constraints $-10 \le x_t^{(i)} \le 10$, $i = 1, 2$, where $x_t = [x_t^{(1)} \;\; x_t^{(2)}]^\top$, and input constraints $-2 \le u_t \le 2$ are imposed. The weighting matrices of the running cost are chosen as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad R = 0.1.$$

Sections 4.6.1 and 4.6.2 show the results for the rollout strategies in Sections 4.3 and 4.4, respectively, where we consider both sparse and sporadically changing input profiles. For the hold strategy, we set $u_{-1} := 0$.

4.6.1 Rollout approach variant (A)

In this subsection we we will illustrate the approach of Section 4.3. The prediction horizon is fixed to $N = 60$. The terminal cost and terminal set satisfying (4.14), are given by $P = \begin{bmatrix} 6.2683 & 1.3785 \\ 1.3785 & 1.4530 \end{bmatrix}$, $K = [-0.8979 \quad -1.1564]$ and $\mathbb{T}$ is

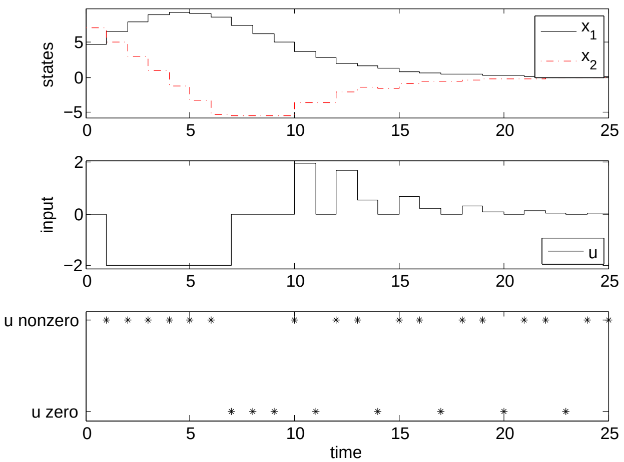

Fig. 4.1. Sparse inputs and resulting states versus time for $x_0 = [4.7239\ 7.0892]^\top$ and $g = 0.05$.

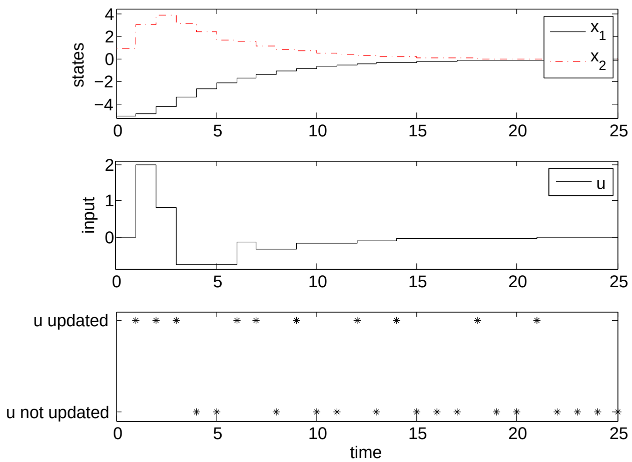

Fig. 4.2. Sporadically changing inputs and resulting states versus time for $x_0 = [-4.9630\ 0.9892]^\top$ and $\beta = 0.01$.

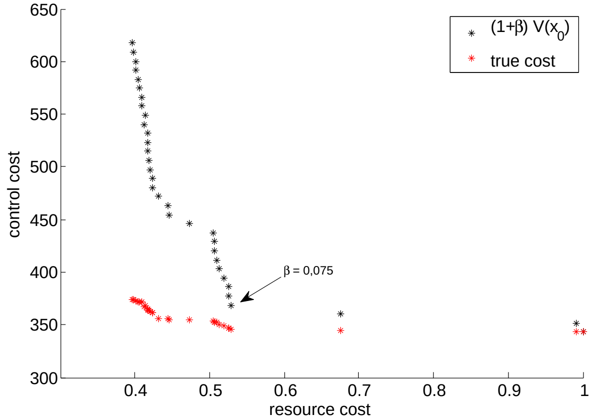

Fig. 4.3. Control cost J_{control} versus the (average) actuation rate J_{resource} for sparse input profiles for $\beta \in \{0, 0.025, 0.050, \ldots, 0.800\}$.

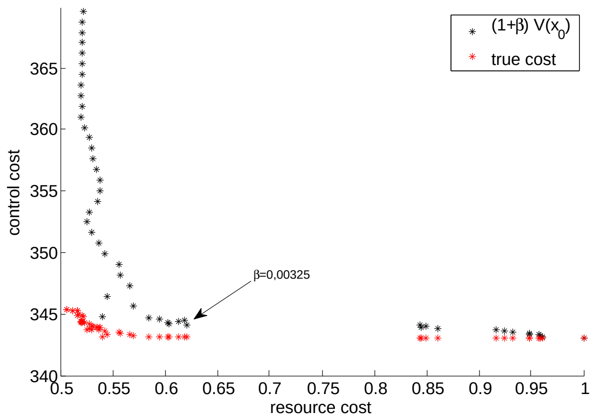

Fig. 4.4. Control cost J_{control} versus the (average) actuation rate J_{resource} for sporadically changing input profiles for $\beta \in \{0, 0.0025, 0.0050, \ldots, 0.0500, 0.0750, 0.1000, \ldots, 0.1000\}$.

chosen as $\{x \in \mathbb{X} \mid Dx \leq E\}$, where

$$D = \begin{bmatrix} -0.9525 & 0.3044 \\ 0.9525 & -0.3044 \\ 0.6133 & 0.7898 \\ -0.6133 & -0.7898 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} 14.4157 \\ 14.4157 \\ 1.3660 \\ 1.3660 \end{bmatrix}.$$

We consider 25 initial conditions randomly and uniformly distributed in $\mathbb{X}_f$. For $\beta = 0.05$, Figures 4.1 and 4.2 show the state trajectories and corresponding input signals versus time, for sparse and sporadically changing input signals, respectively. We observe that even for a slight decrease in performance (i.e., at most 1% and 5% of the guaranteed upper bound on the infinite horizon control cost (see (4.18)), for sparse and sporadically changing input profiles), both approaches obtain input profiles with a significant reduction in resource utilization. Note that the obtained sparse input profile in Figure 4.1 tends to a periodic profile when the state converges the origin (and constraints no longer play a role).

Figures 4.3 and 4.4 show both the guaranteed upper bound on the control cost $(1 + \beta)V(x_0)$ as well as the control cost (4.3) evaluated over a finite but sufficiently long simulation horizon versus the average resource cost (4.4), averaged over 25 initial conditions, for sparse input profiles with $\beta \in \{0, 0.025, 0.050, \ldots, 0.800\}$ and sporadically changing input profiles with $\beta \in \{0, 0.0025, 0.0050, \ldots, 0.0500, 0.0750, 0.1000, \ldots, 0.1000\}$, respectively. From Figure 4.3 we observe that for $\beta = 0.075$ we obtain an average resource utilization J_{resource} of 0.53. For higher values of β the resource utilization is even lower, e.g., for $\beta = 0.8$ we obtain $J_{\text{resource}} = 0.39$. However for $\beta = 0.8$ this also implies a significantly increased control cost. From Figure 4.4 we observe that for $\beta = 0.00325$ we obtain an average resource utilization of 0.62, indicating a tremendous decrease in resource utilization at only a very slight increase in performance. However, similarly to the case of sparse input profiles, increasing β further only yields minor reductions in resource utilization. These results show that the proposed rollout approach fcan realize significant reductions in the utilization of resources, even at only a slight degradation in performance. Note also that, in both Figures 4.3 and 4.4 the guaranteed upper bound is conservative as the actual control cost remains well below the bound.

4.6.2 Rollout approach variant (B)

In this subsection we we will illustrate the approach of Section 4.4. The prediction horizon is fixed to $N = 48$, and we choose $H = 12$ and $q = 4$ (satisfying Assumption 4.1). This means that $c_{\text{resource}} = \frac{1}{4}$ and that $\mathbb{S}_{N_q,H_q,\text{ro}}$ contains 220 schedules that are used for our rollout approach.

For the zero strategy, the terminal cost and terminal set satisfying (4.35) are given by $P_{\text{mr}} = \begin{bmatrix} 6.4514 & 1.4028 \\ 1.4028 & 1.4594 \end{bmatrix}$, $K_{\text{mr}} = [-0.6928 \quad -1.1614]$ and $\mathbb{T}$ is chosen as

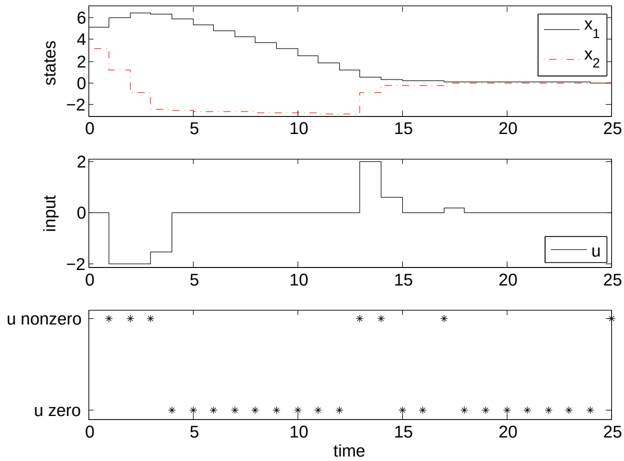

Fig. 4.5. Sparse inputs and resulting states versus time for $x_0 = [5.1174\ 3.2030]^\top$.

$\{x \in \mathbb{X} \mid D_z x \leq E_z\}$, where

$$D_z = \begin{bmatrix} -0.9785 & -0.2064 \\ 0.9785 & 0.2064 \\ 0.5123 & 0.8588 \\ -0.5123 & -0.8588 \end{bmatrix} \quad \text{and} \quad E_z = \begin{bmatrix} 4.2056 \\ 4.2056 \\ 1.4789 \\ 1.4789 \end{bmatrix}.$$

We consider 25 initial conditions randomly and uniformly distributed in $\mathbb{X}_f^{\mathrm{mr}}$. Figure 4.5 shows the sparse input signal and corresponding state trajectories versus time, for $x_0 = [5.1174\ 3.2030]^\top$. The average control cost over 25 initial conditions is 423.98 for the multi-rate strategy and 241.11 for the rollout strategy. This clearly shows the advantage of using aperiodic control updates, as with the same resource utilization the rollout approach obtains a significant reduction in the control cost.

For the hold strategy, the terminal cost and terminal set satisfying (4.35) (see Section 4.5) are given by $P_{\mathrm{mr}} = \begin{bmatrix} 7.4716 & 2.7326 \\ 2.7326 & 2.9789 \end{bmatrix}$, $K_{\mathrm{mr}} = [-0.2148\ \ -0.401]$ and $\mathbb{T}$ is chosen as $\{x \in \mathbb{X} \mid D_h x \leq E_h\}$, where

$$D_h = \begin{bmatrix} -0.7812 & -0.6243 \\ 0.7812 & 0.6243 \\ 0.4715 & 0.8819 \\ -0.4715 & -0.8819 \end{bmatrix} \quad \text{and} \quad E_h = \begin{bmatrix} 7.8371 \\ 7.8371 \\ 4.3896 \\ 4.3896 \end{bmatrix}.$$

Again, we consider 25 initial conditions randomly and uniformly distributed

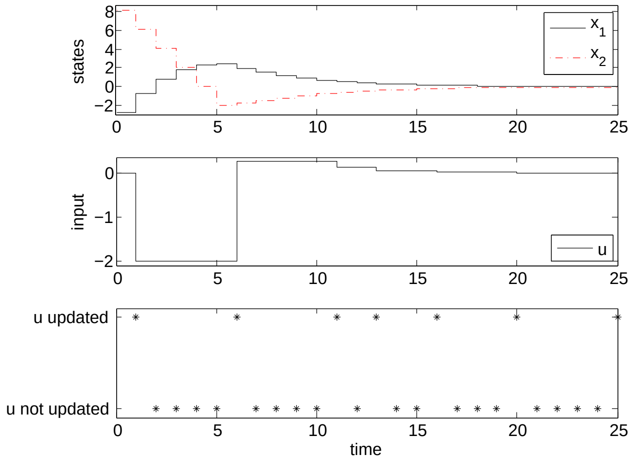

Fig. 4.6. Sporadically changing inputs and resulting states versus time for $x_0 = [-2.7010\ 8.0463]^\top$.

in $\mathbb{X}_f^{\mathrm{mr}}$. Figure 4.6 shows the sporadically changing input signal and corresponding state trajectories versus time, for $x_0 = [-2.7010\ 8.0463]^\top$. The average control cost over 25 initial conditions is 277.71 for the multi-rate strategy and 268.62 for the rollout strategy. For sporadically changing input profiles, the rollout strategy based on a "hold" approach is able to improve slightly over the multi-rate MPC approach, in terms of Problem (B).

4.7 Discussion

This section provides a discussion on the computational complexity and the implementation of both rollout approaches.

At each time $t \in \mathbb{N}$, the rollout approach presented in Section 4.3 samples the state x_t and solves 2 quadratic programing (QP) problems with N and $N-1$ free control variables. Although sampling the state at each time $t \in \mathbb{N}$ implies communication resources are used at each time step, see Remark 4.1, having feedback can be beneficial in case disturbances act on the system or if there is a mismatch between the system model (4.1) and the physical plant.

The rollout approach presented in Section 4.4 samples the state at times $t_l = lH$, $l \in \mathbb{N}$, with $H \in \mathbb{N}_{[q,N]}$. At the sampling times it solves $\frac{H!}{(H-H_q)!H_q!}$ QPs with N_q free control variables, where ! denotes the factorial operator. However, note that the next H time steps no computations are required. Moreover, (suboptimal) variants based on solving less QPs can also be used, by using less

Table 4.1. Average actuation, communication and scheduling rates for variants of the rollout approach in Section 4.4.

	OL in H	CL in H	CL in H WR
Actuation rate	$\frac{1}{q}$	$\frac{1}{q}$	$\frac{1}{q}$
Communication rate	$\frac{1}{H}$	$\frac{1}{q}$	$\frac{1}{q}$
Scheduling rate	$\frac{1}{H}$	$\frac{1}{H}$	$\frac{1}{q}$

schedules, see Remark 4.2. The obtained optimal control sequence is then applied to the system in open-loop, for the next H time steps, see (4.43). We will refer to this approach as open-loop in H (OL in H) . From a NCS perspective, OL in H implies that communication resources are used at a rate $\frac{1}{H}$. When disturbances act on the system, we envision that variants based on more frequent feedback may provide better robustness with respect to these disturbances. Two of these variants are discussed briefly below, and their average actuation, communication and scheduling rates are summarized in Table 4.1.

Closed-loop in H (CL in H): Instead of applying the optimal rollout control sequence obtained at time t_l, $l \in \mathbb{N}$, in open loop for H steps, one can *sample* the state at the H_q times when the input is to be updated (according to the optimal schedule at time t_l) until time t_{l+1}. Based on the sampled state, the remainder of the control *values* until time t_{l+1} can be re-computed using the remainder of the optimal schedule decided at time t_l, see, e.g., [22].

Closed-loop in H with rescheduling (CL in H WR): This variant is also based on *sampling* the state at the H_q times when the input is to be updated. However, next to re-computing the control *values* also the remaining *schedule* until time t_{l+1} is re-computed based on the obtained samples of the state.

The principles of implementing these variants are similar to the ones discussed in Section 4.4.

4.8 Conclusions

In this chapter, we proposed two resource-aware MPC policies for discrete-time linear systems subject to state and input constraints. The strategies solve the co-design problem of both determining the time instants on which the updating/communication related to the control tasks take place, and selecting the new (continuous) control inputs. The first approach provides performance guarantees by design, in the sense that it allows the user to select a desired level of suboptimality for the performance (in terms of the original control cost function), where the degree of suboptimality provides a trade-off between the guaranteed closed-loop control performance on the one hand and the utilization of (communication/actuation) resources on the other hand. The second approach provides a guaranteed (average) resource utilization, while cleverly allocating these resources in order to maximize the control performance (i.e., minimize the control cost). Interestingly, the presented framework is flexible in the sense that it can be configured to generate both sparse and sporadically changing input profiles thereby having the ability to serve various application domains in which different resources are scarce or expensive, e.g., communication bandwidth in networked control systems, battery power in wireless control, or fuel in space or underwater vehicles. The effectiveness of both approaches was illustrated by means of numerical examples. In particular, the first rollout strategy led to significant reductions in the usage of the system's resources, without trading much of the guaranteed achievable performance. Moreover, the rollout strategy with guaranteed resource utilization shows that allocating these resources cleverly can lead to significant reductions of the control cost. As such, the proposed approaches provides viable control strategies for systems where both control performance and limited resource utilization are important.

Chapter 5

Resource-aware set-valued estimation for discrete-time systems

Abstract – In this chapter, we propose a self-triggered estimator for discrete-time linear systems subject to unknown but bounded disturbances affecting both the system states and outputs. The proposed self-triggered estimator is a set-valued estimator that employs rollout techniques to reduce the communication between the sensors and the estimator with respect to a periodic sampling strategy. Moreover, at each time instant it provides an estimate of the plant state and a guaranteed bound on the difference between the true plant state and the estimate

5.1 Introduction

In setups where the system state is estimated by an estimator that receives measurements from the system via a communication link, the cost of communication has to be weighed against the quality of the estimation. Measurements have to be sent over a communication link for example if it is not possible or not desired to include the observer directly in the sensor or, if multiple, spatially distributed sensors are connected to a single observer. If the transmissions are made over a communication network that is shared with other components, managing the amount of communication becomes especially relevant in this case as network usage is a shared and limited resource. Other components might for example be controllers communicating with actuators over the network. A detailed survey of communication issues in networked control systems can be found in [54].

This chapter is based on [21].

Several means of reducing the required communication in control and estimation have been proposed in recent years. Two methods in particular have emerged, that is event triggering, where the sensor decides, based on the current measurement, whether to send an update to the observer/controller or not, and self triggering, where the observer/controller decides, based on the current state (estimate), when the next sensor update will take place. For a recent overview on self- and event-triggered control, including event-triggered estimation, please refer to [48]. Interestingly, in self-triggered schemes, the sensors may be shut down between updates, which allows to save additional energy. This constitutes an advantage of self-triggered strategies over event-triggered strategies. On the other hand, more information is available to the estimator in event-triggered setups, as the sensors not sending measurement updates still allows conclusions to be drawn about the current measurements, based on the event conditions, see, e.g., [82], [84].

In this chapter, we propose a self-triggered set-valued estimator which schedules its future measurements based on the current state estimate. In particular, at each point in time, the estimator computes an estimate of the system state and an upper bound on the estimation error. At periodically recurring scheduling time points, the estimator decides on the time points at which measurements are required from each sensor in the time span until the next scheduling time point. The selection of this measurement schedule is made based on the current known bound on the estimation error. The objective is to minimize the overall number of measurements subject to the constraint that the estimation error does not exceed a certain bound in the time span until the next scheduling time point. Thus, it is guaranteed that the actual estimation error always stays below this bound. The parameters of the set-valued estimator and the scheduling function are designed based on a certain well-chosen base schedule of measurements which is always a feasible choice for the scheduler, thereby guaranteeing an upper bound on the average sampling frequency. Additional constraints on the measurement schedules may be included such that for example at most one sensor can send its measurements over the communication system at a given point in time or that there exists a lower bound on the time between communications. The set-valued estimator used in this work is based on linear programming, while the scheduling function requires the evaluation of scalar inequalities, allowing an efficient implementation of the proposed scheme.

The optimization over update patterns in self-triggered control and estimation was proposed in [8–10], where the disturbances and noise are assumed to be Gaussian distributed, and thus, unbounded. Other recent results on self-triggered state estimation can for example be found in [4,5], where input-to-state stability under self-triggered output-feedback control is shown. The estimators used in [4, 5] are a type of Kalman filter and do not provide an online bound on the current estimation error. Hence, the current bound on the estimation error cannot be used to reduce future communications, as is proposed in the present chapter. In [25], asymptotic stability of uncertain systems under self-triggered output-feedback *control* with continuous output *measurement* is shown. For an introduction to set-valued estimation see for example Chapter 10 of [20] and the references therein. Set-valued estimation based on self-triggered measurements is proposed in [69], where interval estimators are employed to estimate the state of a special class of nonlinear systems. There, each time a measurement is taken, the time until the next measurement is maximized, while guaranteeing a certain bound on the estimation error. However, the results rely on the assumption that a measurement at any given point in time ensures the estimation error to be bounded by an a priori given bound that decreases with time. In [83], a set-valued self-triggered estimator for linear parameter-varying systems is presented, using ellipsoidal bounding techniques. Measurements are required at each point in time, while the self-triggered mechanism decides how much computations have to be performed at a given point in time, based on the current bound on the estimation error.

As the estimator proposed in this chapter provides an upper bound on the estimation error, it may be used in a control scheme that guarantees hard constraints on the system state and input. Model predictive control (MPC) is an example of a class of such control schemes. It is straightforward to combine the results in this chapter with for example the robust output feedback MPC scheme in [66] in order to obtain a robust self-triggered output feedback MPC scheme.

The remainder of this chapter is organized as follows. After indicating the notational conventions used in this chapter and providing an important lemma, Section 5.2 provides the problem formulation. A set-valued estimator is proposed in Section 5.3, which forms the basis for developing the scheduler presented in Section 5.4. The effectiveness of the approach is demonstrated by means of numerical examples in Section 5.5. Section 5.6 presents a discussion on the implementation and computational aspects of the approach. Finally, Section 5.7 presents the conclusions and an outlook.

Notation: The set of non-negative integers is denoted by $\mathbb{N}$, the set of real numbers by $\mathbb{R}$. For $a, b \in \mathbb{R}$ we define $\mathbb{N}_{[a,b]} := \mathbb{N} \cap [a, b]$. By $\mathbb{N}_{>0}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$ we denote the positive integers, positive reals, and nonnegative reals, respectively. For $n \in \mathbb{N}$, the n-dimensional identity matrix is denoted by I_n and $\mathbb{1}_n$ denotes a vector of length n, where each element is 1. The Kronecker product is denoted by $\otimes$. For a square matrix $X \in \mathbb{R}^{n\times n}$ we write $X \succ 0$ and $X \succeq 0$ if X is symmetric and, in addition, X is positive definite and positive semi-definite, respectively. For $X \in \mathbb{R}^{m\times n}$, inequalities of the type $X \geq 0$, should be interpreted element-wise, i.e., all entries of X are nonnegative. For a symmetric matrix X, we write $\begin{bmatrix} X_1 & X_2 \\ \star & X_3 \end{bmatrix}$ to denote $\begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3 \end{bmatrix}$. For $A_i \in \mathbb{R}^{n\times n}$, and $a, b \in \mathbb{N}$, $a \leq b$, we denote $\Pi_{i=a}^{b} A_i = A_a \dots A_b$. For $a > b$, we define $\Pi_{i=a}^{b} A_i := I_n$. For a vector $\mu = [\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(M)}]^\top$, $M \in \mathbb{N}$, $\mu^{(i)} \in \{0, 1\}$, $i = 1, 2, \dots$, we use $S = \mathrm{diag}(\mu)$ to denote a matrix with $\mu^{(1)}, \mu^{(2)}, \dots$, on its diagonal. The following Farkas-type lemma will be used multiple times in this chapter.

Lemma 5.1 ([51])**.** *Let $G \in \mathbb{R}^{m\times n}$, $g \in \mathbb{R}^m$, $F \in \mathbb{R}^{r\times n}$, $f \in \mathbb{R}^r$. If $\{x \in \mathbb{R}^n \mid Gx \leq g\} \neq \emptyset$, then the following two statements are equivalent.*

(i) $\{x \in \mathbb{R}^n \mid Gx \leq g\} \subseteq \{x \in \mathbb{R}^n \mid Fx \leq f\}$

(ii) There exists a matrix $P \in \mathbb{R}^{r\times m}$, $P \geq 0$, such that $PG = F$ and $Pg \leq f$

5.2 Problem formulation

Consider the system

$$x_{t+1} = Ax_t + w_t, \tag{5.1a}$$
$$y_t = Cx_t + v_t, \tag{5.1b}$$

where $x_t \in \mathbb{R}^{n_x}$ is the state, $y_t \in \mathbb{R}^{n_y}$ is the output, and where $w_t \in \mathbb{W}$ and $v_t \in \mathbb{V}$ are the unknown but bounded disturbances affecting the system state and output, respectively, at discrete time $t \in \mathbb{N}$. Here the sets $\mathbb{W}$ and $\mathbb{V}$ contain the origin and are given by $\mathbb{W} := \{w \in \mathbb{R}^{n_x} \mid H_w w \leq h_w\}$ and $\mathbb{V} := \{v \in \mathbb{R}^{n_y} \mid H_v v \leq h_v\}$, for $H_w \in \mathbb{R}^{n_w\times n_x}$, $h_w \in \mathbb{R}^{n_w}$, $H_v \in \mathbb{R}^{n_v\times n_y}$, and $h_v \in \mathbb{R}^{n_v}$.

In this chapter, we consider the case where output measurements are not available to the estimator for all $t \in \mathbb{N}$. Consider the periodic scheduling times $t_l = lM$, $l \in \mathbb{N}$, $M \in \mathbb{N}_{>0}$. At time t_l, $l \in \mathbb{N}$, let $\boldsymbol{\mu}_l = [\boldsymbol{\mu}_{l,1}, \ldots, \boldsymbol{\mu}_{l,M}] \in \mathcal{M} \subseteq \{0,1\}^{n_y\times M}$, $\boldsymbol{\mu}_{l,i} \in \{0,1\}^{n_y}$, $i \in \mathbb{N}_{[1,M]}$, denote the schedule determining which output measurements are available to the estimator at time instances $t = t_l + i$, $i \in \mathbb{N}_{[1,M]}$. Here $\mathcal{M}$ denotes the set of allowed measurement schedules. In particular, the information about the plant output that is available to the estimator at time $t = t_l + i$ is given by $[\bar{y}_{l,1}, \ldots, \bar{y}_{l,i}]$, where $\bar{y}_{l,i} = S_{\boldsymbol{\mu}_{l,i}} y_{t_l+i}$ for $i \in \mathbb{N}_{[1,M]}$, where $S_{\boldsymbol{\mu}_{l,i}} = \mathrm{diag}(\boldsymbol{\mu}_{l,i})$. At every time instant $t \in \mathbb{N}$, the estimator should provide an estimate of the state $\hat{x}_t \in \mathbb{R}^{n_x}$ combined with a bound $a_t \in \mathbb{R}_{\geq 0}$, such that the estimation error $e_t := x_t - \hat{x}_t$ satisfies $e_t \in a_t\mathcal{G}$ for a fixed set $\mathcal{G} \subseteq \mathbb{R}^{n_x}$. The information available to the estimator at time t consists of the available output information in the time span $\mathbb{N}_{[t_l+1,t_l+i]}$, where $t = t_l + i$, the state estimate and the bound on the estimation error at time t_l, i.e., $\hat{x}_{t_l}$ and a_{t_l}. Here $\mathcal{G} = \{x \in \mathbb{R}^{n_x} \mid Gx \leq g\}$ with $G \in \mathbb{R}^{n_g\times n_x}$ and $g \in \mathbb{R}^{n_g}_{\geq 0}$ is a bounded polyhedron with the origin in its interior. The self-triggered set-valued estimator is defined by the functions $\Gamma_i : \{0,1\}^{n_y\times M} \times \mathbb{R}^{n_x} \times \mathbb{R} \times \mathbb{R}^{n_y} \times \ldots \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_x+1}$, $i \in \mathbb{N}_{[1,M]}$, and $\Omega : \mathbb{R} \to \{0,1\}^{n_y\times M}$, such that for $l \in \mathbb{N}$, $i \in \mathbb{N}_{[1,M]}$,

$$[\hat{x}_{t_l+i}^\top, a_{t_l+i}]^\top = \Gamma_i(\boldsymbol{\mu}_l, \hat{x}_{t_l}, a_{t_l}, \bar{y}_{t_l+1}, \ldots, \bar{y}_{t_l+i}), \tag{5.2a}$$
$$\boldsymbol{\mu}_l = \Omega(a_{t_l}), \tag{5.2b}$$

given $\hat{x}_0 \in \mathbb{R}^{n_x}$ and $a_0 \in \mathbb{R}_{\geq 0}$, such that $G(x_0 - \hat{x}_0) \leq a_0 g$. The goal of this chapter is to design Γ_i, $i \in \mathbb{N}_{[1,M]}$, and Ω, such that the number of non-zero elements in $\boldsymbol{\mu}_l$ is minimized at times t_l, $l \in \mathbb{N}$, while guaranteeing $G(x_t - \hat{x}_t) \leq a_t g$ and $a_t \leq a_{\max}$, for all $t \in \mathbb{N}$, for a given $a_{\max} \in \mathbb{R}_{>0}$. We make the following technical assumption on the output disturbances.

Assumption 5.1. *For all $v \in \mathbb{R}^{n_y}$ and all $\boldsymbol{\mu}_l \in \mathcal{M}$, $l \in \mathbb{N}$, it holds that $H_v v \leq h_v$ implies that $H_v S_{\boldsymbol{\mu}_{l,i}} v \leq h_v$, for all $i \in \mathbb{N}_{[1,M]}$.*

Remark 5.1. *This assumption is for example satisfied in the case where the output y_t is composed of n_y independent scalar measurements, i.e., $y_t^{(j)} = C^{(j)} x_t + v_t^{(j)}$, where $v^{(j)} \in [\underline{v}^{(j)}, \overline{v}^{(j)}]$, $\underline{v}^{(j)}, \overline{v}^{(j)} \in \mathbb{R}$, $0 \in [\underline{v}^{(j)}, \overline{v}^{(j)}]$, $j \in \mathbb{N}_{[1,n_y]}$, $t \in \mathbb{N}$. Then it can be assumed without loss of generality, that $H_v = I_{n_y} \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix} \in \mathbb{R}^{n_v \times n_y}$ and $h_v \in \mathbb{R}^{n_v}_{\geq 0}$, where $n_v = 2n_y$. The structure of H_v and $S_{\boldsymbol{\mu}_{l,i}}$ then readily implies that the statement in the assumption holds.*

5.3 Set-valued estimator

In this section, we design the functions Γ_i, $i \in \mathbb{N}_{[1,M]}$, defining the set-valued estimator. Moreover, we provide a priori bounds on the future guaranteed bounds a_{t_l+i}, $i \in \mathbb{N}_{[1,M]}$, on the estimation error for a given schedule $\boldsymbol{\mu}_l$.

Given $\hat{x}_{t_l}$, a_{t_l} and a schedule $\boldsymbol{\mu}_l$, $l \in \mathbb{N}$, the information available to the set-valued estimator at times $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ is given by

$$G(x_{t_l} - \hat{x}_{t_l}) \leq a_{t_l} g \tag{5.3a}$$

$$H_w w_j \leq h_w, \quad j \in \mathbb{N}_{[t_l,t-1]}, \tag{5.3b}$$

$$H_v v_j \leq h_v, \quad j \in \mathbb{N}_{[t_l+1,t]}, \tag{5.3c}$$

$$\bar{y}_{l,j} = S_{\boldsymbol{\mu}_{l,j}} \left(C(A^j x_{t_l} + \sum_{k=0}^{j-1} A^{j-1-k} w_{t_l+k}) + v_{t_l+j} \right), \quad j \in \mathbb{N}_{[1,t-t_l]}, \tag{5.3d}$$

from which our set-valued estimator should obtain $\hat{x}_t$ and the smallest $a_t \geq 0$ such that

$$G(x_t - \hat{x}_t) \leq a_t g, \tag{5.4}$$

for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$.

5.3.1 Optimization problem for set-valued estimator

In the following, we will derive the optimization problem that defines the set-valued estimator. Using Assumption 5.1, from (5.3c) and (5.3d) it follows that $H_v \bar{y}_{l,j} - H_v S_{\boldsymbol{\mu}_{l,j}} \left(C(A^j x_{t_l} + \sum_{k=0}^{j-1} A^{j-1-k} w_{t_l+k}) \right) \leq h_v$, $j \in \mathbb{N}_{[1,t-t_l]}$. Hence, the prior information given at times $t_l + i$, $i \in \mathbb{N}_{[1,M]}$, $l \in \mathbb{N}$, according to (5.3) can be expressed as

$$F_i^{\boldsymbol{\mu}_l,\text{prior}} \begin{bmatrix} x_{t_l} \\ w_{t_l} \\ \vdots \\ w_{t_l+i-1} \end{bmatrix} \leq f_i^{\text{prior}}(\hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \ldots, \bar{y}_{l,i}), \tag{5.5}$$

where

$$F_i^{\boldsymbol{\mu}_l,\text{prior}} = \begin{bmatrix} G & 0 & 0 & \cdots & 0 \\ 0 & H_w & 0 & \cdots & 0 \\ 0 & 0 & H_w & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \ldots & H_w \\ \Xi_{1,1} & \Xi_{1,0} & 0 & \cdots & 0 \\ \vdots & \vdots & & & \\ \Xi_{i,i} & \Xi_{i,i-1} & \Xi_{i,i-2} & \cdots & \Xi_{i,0} \end{bmatrix} \tag{5.6}$$

with $\Xi_{i,j} := -H_v S_{\boldsymbol{\mu}_{l,i}} C A^j$, and

$$f_i^{\text{prior}}(\hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \ldots, \bar{y}_{l,i}) = \begin{bmatrix} G\hat{x}_{t_l} + a_{t_l} g \\ h_w \\ h_w \\ \vdots \\ h_w \\ -H_v \bar{y}_{l,1} + h_v \\ \vdots \\ -H_v \bar{y}_{l,i} + h_v \end{bmatrix}. \tag{5.7}$$

Using the expression

$$x_{t_l+i} = A^i x_{t_l} + \sum_{j=0}^{i-1} A^{i-1-j} w_{t_l+j}, \tag{5.8}$$

for $i \in \mathbb{N}_{[1,M]}$, the posteriori information that the estimator should determine at time $t = t_l + i$ is $\hat{x}_{t_l+i}$ and a_{t_l+i} such that (5.4) is satisfied, which can be expressed as

$$F_i^{\text{post}} \begin{bmatrix} x_{t_l} \\ w_{t_l} \\ \vdots \\ w_{t_l+i-1} \end{bmatrix} \leq f_i^{\text{post}}(\hat{x}_{t_l+i}, a_{t_l+i}), \tag{5.9}$$

where

$$F_i^{\text{post}} = G \left[A^i \;\; A^{i-1} \;\; \ldots \;\; I \right] \tag{5.10}$$

and

$$f_i^{\text{post}}(\hat{x}_{t_l+i}, a_{t_l+i}) = G\hat{x}_{t_l+i} + a_{t_l+i}g. \tag{5.11}$$

By Lemma 5.1, (5.5) implies (5.9) if and only if there exists a matrix $P_i \in \mathbb{R}^{n_g \times n_g + i(n_w+n_v)}$ with $P_i \geq 0$, such that $P_i F_i^{\boldsymbol{\mu}_l,\text{prior}} = F_i^{\text{post}}$ and $P_i f_i^{\text{prior}}(\hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \dots, \bar{y}_{l,i}) \leq f_i^{\text{post}}(\hat{x}_{t_l+i}, a_{t_l+i})$. Hence, the problem of finding an estimate $\hat{x}_{t_l+i}$ based on the given prior information such that the estimation error described by a_{t_l+i} is minimized, can be expressed as the following linear program.

Problem 5.1. *Given $\hat{x}_{t_l}$, a_{t_l}, $\boldsymbol{\mu}_l$ and $\bar{y}_{l,1}, \dots, \bar{y}_{l,i}$, for $i \in \mathbb{N}_{[1,M]}$, $l \in \mathbb{N}$, solve*

$$\begin{bmatrix} \hat{x}_{t_l+i}(\boldsymbol{\mu}_l, \hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \dots, \bar{y}_{l,i}) \\ a_{t_l+i}(\boldsymbol{\mu}_l, \hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \dots, \bar{y}_{l,i}) \end{bmatrix} = \underset{[\hat{x}^\top, a]^\top \in \mathbb{R}^{n_x+1}}{\operatorname{argmin}} \quad \min_{P_i \in \mathbb{R}^{n_g \times n_g + i(n_w+n_v)}} a \tag{5.12}$$

$$\begin{aligned} s.t. \quad & a \geq 0, \\ & P_i \geq 0, \\ & P_i F_i^{\boldsymbol{\mu}_l,\text{prior}} = F_i^{\text{post}}, \\ & P_i f_i^{\text{prior}}(\hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \dots, \bar{y}_{l,i}) \leq f_i^{\text{post}}(\hat{x}, a). \end{aligned}$$

This optimization problem defines the functions Γ_i, $i \in \mathbb{N}_{[1,M]}$, in (5.2a). The following lemma states that the proposed estimator provides a valid bound on the estimation error.

Lemma 5.2. *Let $\hat{x}_{t_l} \in \mathbb{R}^{n_x}$, $a_{t_l} \in \mathbb{R}_{\geq 0}$, $w_j \in \mathbb{R}^{n_x}$, $j \in \mathbb{N}_{[t_l,t-1]}$, $v_j \in \mathbb{R}^{n_y}$, $j \in \mathbb{N}_{[t_l+1,t]}$ and $\bar{y}_{l,j+1} \in \mathbb{R}^{n_y}$, $j \in \mathbb{N}_{[0,t-t_l-1]}$ be given for some $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ and $l \in \mathbb{N}$ such that the constraints in (5.3) are satisfied. Let further $i = t - t_l$, let $\hat{x} \in \mathbb{R}^{n_x}$ and $a \in \mathbb{R}$ satisfy the constraints in Problem 5.1, and let x_t satisfy (5.8). Then it holds that $G(x_t - \hat{x}) \leq ag$.*

Proof. The statement follows from the derivations leading up to Problem 5.1. □

5.3.2 A priori bounds on the estimation error

At times t_l, $l \in \mathbb{N}$, the scheduling function Ω proposed in Section 5.4 provides a schedule $\boldsymbol{\mu}_l$ that minimizes the required communication in the time span $\mathbb{N}_{[t_l+1,t_l+M]}$ while guaranteeing bounds on a_t, $t \in \mathbb{N}_{[t_l+1,t_l+M]}$. We want these bounds to be explicit functions of a_{t_l}, such that they can be evaluated efficiently online for a large number of schedules. In the following, we derive *affine* functions for the bounds on a_t, $t \in \mathbb{N}_{[t_l+1,t_l+M]}$, based on the assumption that the optimization in Problem 5.1 only takes place over a_{t_l+i}, and that $\hat{x}_{t_l+i}$ is chosen (suboptimally) as the output of some linear estimator[1]. This is formalized in the following assumption.

[1]Note that Problem 5.1 is feasible for any fixed $\hat{x} \in \mathbb{R}^{n_x}$ if $a \in \mathbb{R}_{\geq 0}$ is chosen sufficiently large.

Assumption 5.2. *For any schedule* $\boldsymbol{\mu}_l \in \mathcal{M}$, $l \in \mathbb{N}$, *there exist Luenberger-like estimator gains* $L_i^{\boldsymbol{\mu}_l} \in \mathbb{R}^{n_x \times n_y}$, $\lambda_i^{\boldsymbol{\mu}_l}$ *and* $\rho_i^{\boldsymbol{\mu}_l}$, $i \in \mathbb{N}_{[1,M]}$, *such that for any* w_{t_l+i-1}, v_{t_l+i}, $i \in \mathbb{N}_{[1,M]}$, *satisfying* $H_w w_{t_l+i-1} \leq h_w$ *and* $H_v v_{t_l+i} \leq h_v$, $i \in \mathbb{N}_{[1,M]}$, *and any* x_{t_l}, $\hat{x}_{t_l}$ *and* a_{t_l} *with* $G(x_{t_l} - \hat{x}_{t_l}) \leq a_{t_l} g$, *the inequality*

$$G(x_{t_l+i} - \hat{x}_{t_l+i}) \leq (\lambda_i^{\boldsymbol{\mu}_l} a_{t_l} + \rho_i^{\boldsymbol{\mu}_l}) g \tag{5.13}$$

is satisfied for all $i \in \mathbb{N}_{[1,M]}$, *where*

$$x_{t_l+i} = A x_{t_l+i-1} + w_{t_l+i-1}, \tag{5.14}$$

$$\hat{x}_{t_l+i} = A \hat{x}_{t_l+i-1} + L_i^{\boldsymbol{\mu}_l} \left(\bar{y}_{l,i} - S_{\boldsymbol{\mu}_{l,i}} C A \hat{x}_{t_l+i-1} \right) \tag{5.15}$$

with

$$\bar{y}_{l,i} = S_{\boldsymbol{\mu}_{l,i}} \left(C x_{t_l+i} + v_{t_l+i} \right). \tag{5.16}$$

Remark 5.2. *Less conservative a priori bounds on* a_t, $t \in \mathbb{N}_{[t_l+1, t_l+M]}$, *based on* a_{t_l} *for* $l \in \mathbb{N}$, *can be achieved by employing computation-heavy min-max optimization. This matter is subject to future research.*

In the following, we will show how, for a given schedule $\boldsymbol{\mu}_l$ and given estimator gains $L_i^{\boldsymbol{\mu}_l}$, $i \in \mathbb{N}_{[1,M]}$, minimal $\lambda_i^{\boldsymbol{\mu}_l}, \rho_i^{\boldsymbol{\mu}_l}$, $i \in \mathbb{N}_{[1,M]}$, satisfying Assumption 5.2, can be obtained by linear programming. The inequalities in (5.3a) to (5.3c) can be written as

$$F_i^{\text{LP,prior}} z_{t_l,i} \leq f_i^{\text{LP,prior}}, \tag{5.17}$$

where

$$z_{t_l,i} := [x_{t_l}^\top - \hat{x}_{t_l}^\top, a_{t_l}, w_{t_l}^\top, \ldots, w_{t_l+i}^\top, v_{t_l+1}^\top, \ldots, v_{t_l+i+1}^\top]^\top,$$

$$F_i^{\text{LP,prior}} := \begin{bmatrix} G & -g & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & I_i \otimes H_w & 0 \\ 0 & 0 & 0 & I_i \otimes H_v \end{bmatrix} \tag{5.18}$$

and

$$f_i^{\text{LP,prior}} := \begin{bmatrix} 0 \\ 0 \\ \mathbb{1}_i \otimes h_w \\ \mathbb{1}_i \otimes h_v \end{bmatrix}. \tag{5.19}$$

Using the recursive procedure in (5.15) to obtain the estimates $\hat{x}_{t_l+i}$ leads to the error dynamics

$$\begin{aligned}(x_{t_l+i} - \hat{x}_{t_l+i}) = {} & (A - L_i^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,i}} C A)(x_{t_l+i-1} - \hat{x}_{t_l+i-1}) \\ & + (I_{n_x} - L_i^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,i}} C) w_{t_l+i-1} - L_i^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,i}} v_{t_l+i},\end{aligned} \tag{5.20}$$

for $i \in \mathbb{N}_{[1,M]}$, or, equivalently,

$$(x_{t_l+i} - \hat{x}_{t_l+i}) = \underbrace{\left(\prod_{j=0}^{i-1}(A - L_{i-j}^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,i-j}} CA)\right)}_{=:\Theta_i}(x_{t_l} - \hat{x}_{t_l})$$
$$+ \sum_{j=0}^{i-1} \underbrace{\left(\prod_{k=0}^{i-2-j}(A - L_{i-k}^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,i-k}} CA)\right)(I_{n_x} - L_{j+1}^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,j+1}} C)}_{=:\Phi_{i,j}} w_{t_l+j}$$
$$+ \sum_{j=0}^{i-1} \underbrace{\left(\prod_{k=0}^{i-2-j}(A - L_{i-k}^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,i-k}} CA)\right)(-L_{j+1}^{\boldsymbol{\mu}_l} S_{\boldsymbol{\mu}_{l,j+1}})}_{=:\Psi_{i,j}} v_{t_l+j+1} \quad (5.21)$$

for $i \in \mathbb{N}_{[1,M]}$. Hence, the inequality in (5.13) can be written as

$$F_i^{\mathrm{LP,post}}(\lambda_i^{\boldsymbol{\mu}_l}) z_{t_l,i} \leq f_i^{\mathrm{LP,post}}(\rho_i^{\boldsymbol{\mu}_l}), \quad (5.22)$$

where

$$F_i^{\mathrm{LP,post}}(\lambda_i^{\boldsymbol{\mu}_l}) := \begin{bmatrix} G\Theta_i & -\lambda_i^{\boldsymbol{\mu}_l} g & G\Phi_{i,0} & \dots & G\Phi_{i,i-1} & G\Psi_{i,0} & \dots & G\Psi_{i,i-1} \end{bmatrix} \quad (5.23)$$

and $f_i^{\mathrm{LP,post}}(\rho_i^{\boldsymbol{\mu}_l}) := \rho_i^{\boldsymbol{\mu}_l} g$. By Lemma 5.1, if there exists a $z_{t_l,i} \in \mathbb{R}^{n_x+1+i(n_w+n_v)}$ satisfying (5.17), then (5.17) implies (5.22) if and only if there exists a matrix $P_i \in \mathbb{R}^{n_g \times n_g+1+i(n_w+n_v)}$ with $P_i \geq 0$, such that $P_i F_i^{\mathrm{LP,prior}} = F_i^{\mathrm{LP,post}}(\lambda_i^{\boldsymbol{\mu}_l})$ and $P_i f_i^{\mathrm{LP,prior}} \leq f_i^{\mathrm{LP,post}}(\rho_i^{\boldsymbol{\mu}_l})$. Hence, given estimator gains $L_i^{\boldsymbol{\mu}_l}$, $i \in \mathbb{N}_{[1,M]}$, the following linear program can be used to obtain minimal $\lambda_i^{\boldsymbol{\mu}_l}, \rho_i^{\boldsymbol{\mu}_l}$, $i \in \mathbb{N}_{[1,M]}$, satisfying Assumption 5.2.

Problem 5.2.

$$\begin{bmatrix} \lambda_i^{\boldsymbol{\mu}_l} \\ \rho_i^{\boldsymbol{\mu}_l} \end{bmatrix} = \underset{[\lambda,\rho]^\top \in \mathbb{R}^2}{\mathrm{argmin}} \min_{P \in \mathbb{R}^{n_g \times n_g+1+i(n_w+n_v)}} \lambda + \rho \quad (5.24)$$
$$\text{s.t.} \quad \lambda \geq 0,\ \rho \geq 0,\ P \geq 0,$$
$$PF_i^{\mathrm{LP,prior}} = F_i^{\mathrm{LP,post}}(\lambda),$$
$$Pf_i^{\mathrm{LP,prior}} \leq f_i^{\mathrm{LP,post}}(\rho).$$

Note that due to the block-structure of $F_i^{\mathrm{LP,prior}}$ and the zero entries in $f_i^{\mathrm{LP,prior}}$, there is no interdependence between the variables λ and ρ in the optimization problem, such that they could be minimized using independent optimization problems yielding the same results.

The following lemma establishes that $\lambda_i^{\boldsymbol{\mu}_l}$ and $\rho_i^{\boldsymbol{\mu}_l}$ computed according to Problem 5.2 actually provide a bound on a_{t_l+i}.

Lemma 5.3. *Let Assumption 5.2 hold true and let* (5.5) *be satisfied. Then, the constraints of Problem 5.1 are feasible for* $a = \lambda_i^{\boldsymbol{\mu}_l} a_{t_l} + \rho_i^{\boldsymbol{\mu}_l}$, $i \in \mathbb{N}_{[1,M]}$, *and some* $\hat{x} \in \mathbb{R}^{n_x}$.

Proof. By noting that $\bar{y}_{l,i} = S_{\boldsymbol{\mu}_{l,i}} y_{t_l+i}$ and that $S_{\boldsymbol{\mu}_{l,i}} S_{\boldsymbol{\mu}_{l,i}} = S_{\boldsymbol{\mu}_{l,i}}$, $i \in \mathbb{N}_{[1,M]}$, it follows that satisfaction of (5.5) implies the existence of $v_{t_l+j} \in \mathbb{R}^{n_y}$ for $j \in \mathbb{N}_{[1,i]}$, with $H_v v_{t_l+j} \leq h_v$ such that $\bar{y}_{l,j}$ satisfies (5.3d) for $j \in \mathbb{N}_{[1,i]}$. From this it follows that (5.17) is satisfied. By Assumption 5.2, (5.17) implies that $G(x_{t_l+i} - \hat{x}_{t_l+i}) \leq (\lambda_i^{\boldsymbol{\mu}_l} a_{t_l} + \rho_i^{\boldsymbol{\mu}_l}) g$ for the specific choice of $\hat{x}_{t_l+i}$ in (5.15). The last inequality is, noting the definition of x_{t_l+i}, equivalent to (5.9) with $a_{t_l+i} = \lambda_i^{\boldsymbol{\mu}_l} a_{t_l} + \rho_i^{\boldsymbol{\mu}_l}$. As x_{t_l} and w_{t_l+j} for $j \in \mathbb{N}_{[1,i]}$ were arbitrary, we can conclude that (5.5) implies (5.9) with $a_{t_l+i} = \lambda_i^{\boldsymbol{\mu}_l} a_{t_l} + \rho_i^{\boldsymbol{\mu}_l}$ (and the specific choice of $\hat{x}_{t_l+i}$ in (5.15)). By Lemma 5.1, this implies the existence of a matrix $P_i \in \mathbb{R}^{n_g \times n_g + i(n_w + n_v)}$ satisfying the constraints in Problem 5.1, thereby completing the proof. □

The scheduling function Ω proposed in Section 5.4 is defined by a minimization problem with the number of communications in the schedule as its cost function. By guaranteeing that a certain base schedule $\bar{\boldsymbol{\mu}} := (\bar{\mu}_1, \ldots, \bar{\mu}_M) \in \mathcal{M}$ is always a feasible solution of this optimization problem, it is ensured that the actual number of communications in the selected schedule is less than or equal to the number of communications required for the base schedule. The requirement that $a_t \leq a_{\max}$, $t \in \mathbb{N}$, is enforced through constraints in this optimization problem. Hence, it is necessary, that there exists some estimator using the base schedule $\bar{\boldsymbol{\mu}}$, that has certain convergence properties. In particular, we assume the existence of a linear estimator using the base schedule such that the set $\mathcal{G}$ is contractive (see, e.g., [20]) under the associated estimation error dynamics in the absence of disturbances. That is, there exists a $\bar{\lambda} \in [0, 1)$ such that

$$Gx \leq g \;\Rightarrow\; G \prod_{j=0}^{M-1} (A - L_{M-j}^{\bar{\boldsymbol{\mu}}_l} S_{\bar{\mu}_{i-j}} CA) x \leq \bar{\lambda} g, \tag{5.25}$$

where $L_i^{\bar{\boldsymbol{\mu}}}$, $i \in \mathbb{N}_{[1,M]}$ are the Luenberger-like gains in Assumption 5.2, associated with the base schedule $\bar{\boldsymbol{\mu}}$. The following assumption provides a concise characterization of the required properties of $\mathcal{G}$ and $\bar{\boldsymbol{\mu}}$.

Assumption 5.3. *It holds that* $\lambda_M^{\bar{\boldsymbol{\mu}}} < 1$.

The question remains how to compute the Luenberger-like gains. For the base schedule we assume that these gains are computed using the periodic Riccati equation [89]

$$X_i = AX_{i+1}A^\top - AX_{i+1}C_i^\top (C_i X_{i+1} C_i^\top + R_i)^{-1} C_i X_{i+i} A^\top, \tag{5.26}$$

with weighting matrices $Q_i \in \mathbb{R}^{n_x \times n_x}$, $R_i \in \mathbb{R}^{n_y \times n_y}$, where $Q_i, R_i \succ 0$, $i \in \mathbb{N}_{[1,M]}$, for the periodic time-varying system

$$x_{t+1} = Ax_t, \tag{5.27a}$$
$$y_t = C_t x_t, \tag{5.27b}$$

where $C_{t+M} = C_t$, $t \in \mathbb{N}$ and $C_i = S_{\bar{\mu}_i} CA$, $i \in \mathbb{N}_{[1,M]}$, moreover, the gains are given by

$$L_i^{\bar{\boldsymbol{\mu}}} = (C_i X_{i+1} C_i^\top + R_i)^{-1} C_i X_{i+i} A^\top \tag{5.28}$$

This approach, while guaranteeing a converging estimation error, also allows including knowledge about the disturbance distributions in the form of the weighting matrices Q_i and R_i. The gains $L_i^{\boldsymbol{\mu}}$ for schedules $\boldsymbol{\mu}$ other than the base schedule $\bar{\boldsymbol{\mu}}$, can in principle be chosen arbitrarily. However, it makes sense to choose them in way such that the estimation error remains small when using these particular gains in a Luenberger-like estimator, even if it can not be guaranteed that the estimation error converges. In order to achieve a small estimation error, we select the gains $L_i^{\boldsymbol{\mu}}$ based on the following semi-definite program.

Problem 5.3. *Find*

$$\max_{Y_0, Y_i \in \mathbb{R}^{n_x \times n_x}, D_i \in \mathbb{R}^{n_x \times n_x}, Z_i \in \mathbb{R}^{n_x \times n_y}, i \in \mathbb{N}_{[1,M]}} \quad \sum_{i=0}^{M} \operatorname{trace}(Y_i) \tag{5.29}$$

$$\begin{aligned}
\text{s.t.} \quad & Y_j = Y_j^\top \succ 0, \\
& \frac{1}{\alpha} Y_0 - Y_M \succeq 0, \qquad (5.30)\\
& \begin{bmatrix} D_i + D_i^\top - Y_{i-1} & D_i Q_i^{\frac{1}{2}} & Z_i R_i^{\frac{1}{2}} & D_i A - Z_i S_{\mu_i} CA \\ \star & I_{n_x} & 0 & 0 \\ \star & \star & I_{n_y} & 0 \\ \star & \star & \star & Y_i \end{bmatrix} \succeq 0, \\
& i \in \mathbb{N}_{[1,M]}, \; j \in \mathbb{N}_{[0,M]},
\end{aligned}$$

where $\boldsymbol{\mu} := (\mu_1, \dots, \mu_M)$, $Q_i = Q_i^\top \succeq 0$ *and* $R_i = R_i^\top \succ 0$, $i \in \mathbb{N}_{[1,M]}$, *are given matrices of appropriate dimensions, and* $\alpha \in (0, 1]$.

Besides the constraint 5.30, this type of LMI is standard in the literature, see for example [60] and the references therein. A bisection approach is employed to obtain the maximal value of α for which the problem is still feasible. The estimator gains are then obtained as $L_i^{\boldsymbol{\mu}} := D_i^{-1} Z_i$, $i \in \mathbb{N}_{[1,M]}$. The reasoning behind Problem 5.3 is the following. For $\alpha = 1$ the feasibility of the constraints implies convergence of the estimation error, if the resulting gains are employed. Maximizing α aims for resulting estimators that are as close to convergence as possible.

The necessary offline computations are summarized in the following algorithm.

Algorithm 1 Offline computations

1: Solve the periodic Riccati equations (5.26) for the base schedule $\bar{\boldsymbol{\mu}}$ and obtain the Luenberger-like gains $L_i^{\bar{\boldsymbol{\mu}}}$, $i \in \mathbb{N}_{[1,M]}$, from (5.28).
2: Compute a contractive polytope $\mathcal{G} := \{x \in \mathbb{R}^{n_x} \mid Gx \leq g\}$ containing the origin in its interior satisfying (5.25).
3: For all schedules $\boldsymbol{\mu} \in \mathcal{M} \setminus \{\bar{\boldsymbol{\mu}}\}$, solve Problem 5.3 to obtain the Luenberger-like gains $L_i^{\boldsymbol{\mu}}$, $i \in \mathbb{N}_{[1,M]}$.
4: For all schedules $\boldsymbol{\mu} \in \mathcal{M}$, solve Problem 5.2 and obtain $\lambda_i^{\boldsymbol{\mu}}, \rho_i^{\boldsymbol{\mu}}$, $i \in \mathbb{N}_{[1,M]}$.

5.4 Self-triggered set-valued estimation

In this section, we will propose a scheduling function that decides online, based on the current bound on the estimation error a_{t_l}, which measurement schedule $\boldsymbol{\mu}_l$ to use in the time span $\mathbb{N}_{[t_l+1,t_l+M]}$. For any $\boldsymbol{\mu}_l \in \mathcal{M}$, with $\boldsymbol{\mu}_{l,i} = [\boldsymbol{\mu}_{l,i}^{(1)}, \ldots, \boldsymbol{\mu}_{l,i}^{(n_y)}]^\top$, $i \in \mathbb{N}_{[1,M]}$, we define $s(\boldsymbol{\mu}_l) := \sum_{i=1}^{M} \sum_{j=1}^{n_y} \boldsymbol{\mu}_{l,i}^{(j)}$ indicating the total number of sensor measurements communicated during the time span $\mathbb{N}_{[t_l+1,t_l+M]}$. At time t_l, $l \in \mathbb{N}$, given an estimate $\hat{x}_{t_l}$, and an error bound a_{t_l}, the measurement schedule for the times $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ is given as the solution of the following optimization problem, which defines the function Ω in (5.2b).

Problem 5.4. *At time t_l, given $\hat{x}_{t_l}$ and a_{t_l},*

$$\boldsymbol{\mu_l}(a_{t_l}) = \arg\min_{\boldsymbol{\mu} \in \mathcal{M}} \{ s(\boldsymbol{\mu}) \mid \lambda_M^{\boldsymbol{\mu}} a_{t_l} + \rho_M^{\boldsymbol{\mu}} \leq a_{\min}, \lambda_i^{\boldsymbol{\mu}} a_{t_l} + \rho_i^{\boldsymbol{\mu}} \leq a_{\max}, i \in \mathbb{N}_{[1,M-1]} \},$$

with the additional parameter $a_{\min} \in \mathbb{R}_{>0}$, $a_{\min} \leq a_{\max}$.

That is, the schedule with the least number of required measurements is selected which still guarantees the upper bound $a_t \leq a_{\max}$ for $t \in \mathbb{N}_{[t_l+1,t_l+M]}$. The inclusion of the additional parameter $a_{\min}$ and its associated constraint in Problem 5.4 endows the optimization problem with the property of recursive feasibility, meaning that if the problem is feasible at t_0, it will remain feasible for all t_l, $l \in \mathbb{N}$. In particular, the following assumption ensures that the base schedule $\bar{\boldsymbol{\mu}}$ is always a feasible (but not necessarily optimal) solution to Problem 5.4.

Assumption 5.4. *It holds that*

1. $a_{\min} \geq \frac{\rho_M^{\bar{\boldsymbol{\mu}}}}{1-\lambda_M^{\bar{\boldsymbol{\mu}}}}$ *and*
2. $a_{\max} \geq \max\{a_{\min}, \lambda_i^{\bar{\boldsymbol{\mu}}} a_{\min} + \rho_i^{\bar{\boldsymbol{\mu}}} \mid i \in \mathbb{N}_{[1,M]}\}$.

Lemma 5.4. *Let $\hat{x}_{t_l}$ and a_{t_l} be given such that $G(x_{t_l} - \hat{x}_{t_l}) \leq a_{t_l} g$, where $a_{t_l} \leq a_{\min}$ for some $l \in \mathbb{N}$. Then $\bar{\boldsymbol{\mu}}$ is a feasible solution to Problem 5.4. Moreover, if $\boldsymbol{\mu}_l$ is computed by Problem 5.4 and a_t is computed by Problem 5.1 using schedule $\boldsymbol{\mu}_l$ for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$, then it holds that $a_t \leq a_{\max}$ for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ and Problem 5.4 is feasible for a_{t_l+M}. Moreover, it holds that $G(x_{t_l+M} - \hat{x}_{t_l+M}) \leq a_{t_l+M} g$, where $a_{t_l+M} \leq a_{\min}$.*

Proof. By Assumption 5.4, if $a_{t_l} \leq a_{\min}$, then it holds that $\lambda_M^{\bar{\boldsymbol{\mu}}} a_{t_l} + \rho_M^{\bar{\boldsymbol{\mu}}} \leq \lambda_M^{\bar{\boldsymbol{\mu}}} a_{\min} + (1 - \lambda_M^{\bar{\boldsymbol{\mu}}}) a_{\min} = a_{\min} \leq a_{\max}$ and $\lambda_i^{\bar{\boldsymbol{\mu}}} a_{t_l} + \rho_i^{\bar{\boldsymbol{\mu}}} \leq \lambda_i^{\bar{\boldsymbol{\mu}}} a_{\min} + \rho_i^{\bar{\boldsymbol{\mu}}} \leq a_{\max}$, $i \in \mathbb{N}_{[1,M-1]}$, such that the base schedule $\bar{\boldsymbol{\mu}}$ is a feasible, but not necessarily an optimal solution of Problem 5.4. By Lemma 5.3, it holds that $a_t \leq \lambda_{t-t_l}^{\boldsymbol{\mu}_l} a_{t_l} + \rho_{t-t_l}^{\boldsymbol{\mu}_l}$ for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ and all $\boldsymbol{\mu}_l \in \mathcal{M}$, $l \in \mathbb{N}$, such that the satisfaction of the constraints in Problem 5.4 implies that $a_t \leq a_{\max}$ for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ and $a_{t_l+M} \leq a_{\min}$, which implies feasibility of Problem 5.4 for a_{t_l+M}. Finally, by Lemma 5.2, it holds that $G(x_{t_l+M} - \hat{x}_{t_l+M}) \leq a_{t_l+M} g$, thereby completing the proof. □

The following theorem states the main results about the self-triggered set-valued estimation scheme.

Theorem 5.5. *Let $\hat{x}_{t_0}$ and a_{t_0} be given such that $G(x_{t_0} - \hat{x}_{t_0}) \leq a_{t_0} g$, where $a_{t_0} \leq a_{\min}$. Let $\boldsymbol{\mu}_l$ be decided by Problem 5.4 for all $l \in \mathbb{N}$ and let $\hat{x}_t$ and a_t be computed by Problem 5.1 using schedule $\boldsymbol{\mu}_l$ for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$ and all $l \in \mathbb{N}$. Then, it holds that $s(\boldsymbol{\mu}_l) \leq s(\bar{\boldsymbol{\mu}})$ for all $l \in \mathbb{N}$. Furthermore, it holds that $G(x_t - \hat{x}_t) \leq a_{\max} g$ for all $t \in \mathbb{N}$.*

Proof. As the requirements in Lemma 5.4 are satisfied at initialization, it follows by induction that $\bar{\boldsymbol{\mu}}$ is a feasible solution to Problem 5.4 for all $l \in \mathbb{N}$. As the scheduler selects a schedule $\boldsymbol{\mu}_l$ with the least number of measurements, it follows that $s(\boldsymbol{\mu}_l) \leq s(\bar{\boldsymbol{\mu}})$ for all $l \in \mathbb{N}$. Furthermore, by Lemma 5.2, it holds that $G(x_t - \hat{x}_t) \leq a_t g$ for all $t \in \mathbb{N}$. By Lemma 5.4 it follows that $a_t \leq a_{\max}$ for all $t \in \mathbb{N}_{[t_l+1,t_l+M]}$, $l \in \mathbb{N}$, and, hence, $G(x_t - \hat{x}_t) \leq a_{\max} g$, $t \in \mathbb{N}$, thereby completing the proof. □

The overall self-triggered estimation algorithm can be summarized as follows.

Algorithm 2 Self-triggered set-valued estimator

Initialize $\hat{x}_0$ and a_0 such that $G(x_0 - \hat{x}_0) \leq a_0 g$.
$t_0 := 0$.
for $l \in \mathbb{N}$ **do**
  Obtain $\boldsymbol{\mu}_l(a_{t_l})$ by solving Problem 5.4.
  for $i \in \mathbb{N}_{[1,M]}$ **do**
    Obtain measurement $\bar{y}_{l,i}$ according to t_l, i and $\boldsymbol{\mu}_l$.
    By solving Problem 5.1, compute $\hat{x}_{t_l+i}(\boldsymbol{\mu}_l, \hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \dots, \bar{y}_{l,i})$ and $a_{t_l+i}(\boldsymbol{\mu}_l, \hat{x}_{t_l}, a_{t_l}, \bar{y}_{l,1}, \dots, \bar{y}_{l,i})$.
  end for
  $t_{l+1} := t_l + M$.
end for

Remark 5.3. *The properties of the self-triggered estimation scheme are determined by the design parameters $\mathcal{M}$ and $\bar{\boldsymbol{\mu}}$. The former contains all possible communication schedules, which allows any type of constraints on the communication to be imposed, such as restrictions on the number of sensors that can communicate simultaneously, an upper bound on the average number of measurement updates in a given time span, or a lower bound on the time between two measurement updates. Via Assumption 5.4, the latter provides an upper bound on the estimation error.*

5.5 Numerical examples

In this section we provide two numerical examples demonstrating the effectiveness of the proposed scheme.

5.5.1 Single-output system

In the first example, we consider a system as defined in Section 5.2 with the system matrices given by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{5.31}$$

The bounds on the disturbances are given by $\mathbb{W} = [-0.01, 0.01]^2$ and $\mathbb{V} = [-0.5, 0.5]$. The base schedule was chosen as $\bar{\boldsymbol{\mu}} = [1\ 1\ 1\ 1\ 1\ 1]$. For the periodic Riccati equation and the LMIs in the offline computations in Algorithm 1, we chose $Q_i = I_2$ and $R_i = 0.546$, $i \in \mathbb{N}_{[1,6]}$. We chose $R_i = R$, $i \in \mathbb{N}_{[1,6]}$, and tuned R in order to minimize the minimally possible $a_{\max}$ in Assumption 5.4. For all examples in this chapter, we used 10 bisection steps for obtaining α in

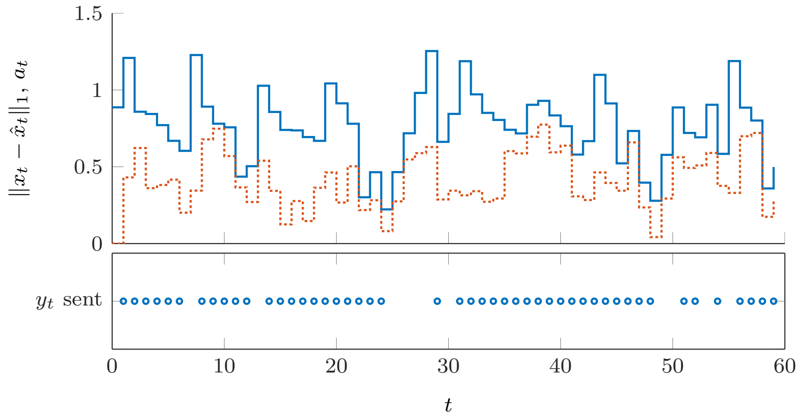

Fig. 5.1. The upper plot shows the evolution of the real (dashed) output error $\|x_t - \hat{x}_t\|_1$ and the upper bound a_t (solid) computed by the estimator. The lower plot shows the times at which measurements are sent.

Problem 5.3, starting from $\alpha = 1$. We chose $\mathcal{M}$ to be the set of all possible schedules of length $M = 6$. We chose $\mathcal{G} = \{e \in \mathbb{R}^2 \mid \|e\|_1 \leq 1\}$, such that $e_t \in a_t\mathcal{G} \Leftrightarrow \|e_t\|_1 \leq a_t$. The values for $a_{\min}$ and $a_{\max}$ in Assumption 5.4 were chosen as $a_{\min} = \frac{\rho_M^{\bar{\boldsymbol{\mu}}}}{1-\lambda_M^{\bar{\boldsymbol{\mu}}}} = 0.8869$ and $a_{\max} = \max\{a_{\min}, \lambda_i^{\bar{\boldsymbol{\mu}}} a_{\min} + \rho_i^{\bar{\boldsymbol{\mu}}} \mid i \in \mathbb{N}_{[1,M]}\} = 1.2638$. With this choice of parameters, the self-triggered set-valued estimator guarantees bounds on the estimation error which are comparable to those guaranteed by a well-tuned Luenberger estimator receiving measurements at every point in time. The initial conditions were set to $\hat{x}_0 = x_0 = [0\ 0]^\top$ and $a_0 = a_{\min}$. We ran 100 simulations for $t \in \mathbb{N}_{[0,59]}$ with uniformly independently distributed random disturbances $w_t \in \mathbb{W}$ and $v_t \in \mathbb{V}$. Plots of the guaranteed and actual estimation error for one of the simulations are shown in Figure 5.1, as are the times when measurements were transmitted. The average number of transmitted output measurements was 88% of that for the base schedule. This example shows that a self-triggered set-valued estimator is able to reduce the number of communicated measurements while guaranteeing the *same* (worst-case) bounds on the estimation error as a set-valued estimator that receives measurements at every point in time. Note that increasing the maximal bounds on the estimation error allows to further reduce the number of communicated measurements, as we show in the next example.

5.5.2 Multiple-output system

In the second example, we show that our self-triggered set-valued estimation scheme can be employed to decide on which sensors to choose for future measurements under the constraint that only one sensor may send its measurements at a given point in time. Let $a^*_{\min} = \frac{\rho^{\bar{\boldsymbol{\mu}}}_M}{1-\lambda^{\bar{\boldsymbol{\mu}}}_M}$ and $a^*_{\max} = \max\{a^*_{\min}, \lambda^{\bar{\boldsymbol{\mu}}}_i a^*_{\min} + \rho^{\bar{\boldsymbol{\mu}}}_i \mid i \in \mathbb{N}_{[1,M]}\}$, i.e., $a^*_{\min}$ and $a^*_{\max}$ satisfy Assumption 5.4 with equality. In this example we will furthermore show that by selecting $a_{\min} > a^*_{\min}$ and $a_{\max} > a^*_{\max}$, our self-triggered set-based estimator is able to significantly reduce the required communication, at the price of obtaining a larger (but guaranteed) bound on the estimation error. We again consider a system as defined in Section 5.2, the system matrices are given by

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.32}$$

Note that the system is not detectable if only one the two output channels of the system is used. The bounds on the disturbances are given by $\mathbb{W} = [-0.01, 0.01]^3$ and $\mathbb{V} = [-0.5, 0.5]^2$. The base schedule was chosen as $\bar{\boldsymbol{\mu}} = \left[\begin{smallmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{smallmatrix}\right]$. For the periodic Riccati equation and the LMIs in the offline computations in Algorithm 1, we chose $Q_i = I_3$ and $R_i = I_2$, $i \in \mathbb{N}_{[1,4]}$. We chose $\mathcal{M}$ to be the set of all possible schedules of length $M = 4$, where at most 1 sensor is updated at any given point in time. We chose $\mathcal{G} = \{e \in \mathbb{R}^3 \mid \|e\|_\infty \leq 1\}$, such that $e_t \in a_t\mathcal{G} \Leftrightarrow \|e_t\|_\infty \leq a_t$. We obtain $a^*_{\min} = 1.0978$ and $a^*_{\max} = 2.6410$, and we chose $a_{\min} = 2$ and $a_{\max} = 4.5$. The initial conditions were set to $\hat{x}_0 = x_0 = [0\ 0\ 0]^\top$ and $a_0 = a_{\min}$. We ran 100 simulations for $t \in \mathbb{N}_{[0,59]}$ with uniformly independently distributed random disturbances $w_t \in \mathbb{W}$ and $v_t \in \mathbb{V}$. Plots of the guaranteed and actual estimation error for one of the simulations are shown in Figure 5.2, as are the times when measurements were transmitted. The average number of transmitted output measurements was 63% of that for the base schedule. In comparison, running a simulation for $a_{\min} = a^*_{\min} + \epsilon$, and $a_{\max} = \max\{a^*_{\min}, \lambda^{\bar{\boldsymbol{\mu}}}_i a^*_{\min} + \rho^{\bar{\boldsymbol{\mu}}}_i \mid i \in \mathbb{N}_{[1,M]}\} + \epsilon$, for $\epsilon = 0.5 \cdot 10^{-4}$, yielded an average number of transmitted output measurements was 99.23% of that for the base schedule. Here, ϵ is included for numerical reasons.

5.6 Discussion on computational complexity and implementation

In this section we briefly discuss issues regarding the computations involved with and the implementation of the proposed scheme. The offline computations in Algorithm 1 scale with the number of schedules. If $\mathcal{M} = \{0, 1\}^{n_y \times M}$ the number of possible schedules scales exponentially with $n_y M$.

The online computations require solving the LP related to the set-valued estimator once for each $t \in \mathbb{N}$, as given in Problem 5.1. These computations can be done efficiently. For the examples in Section 5.5, the required computation times are in the order of milliseconds on a desktop computer. At times t_l, $l \in \mathbb{N}$,

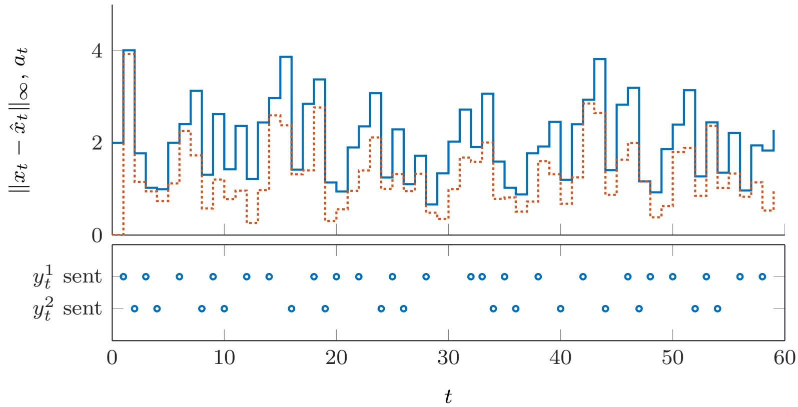

Fig. 5.2. The upper plot shows the evolution of the real (dashed) output error $\|x_t - \hat{x}_t\|_\infty$ and the upper bound a_t (solid) computed by the estimator. The lower plot shows the times at which measurements are sent over each of the two output channels.

it is required to solve Problem 5.4 to obtain the next schedule. This schedule selection involves checking scalar inequalities, M for each schedule, which can be done in milliseconds on a desktop computer, even when the number of schedules is in the order of 10^4.

The measurement schedules that result from the self-triggered set-valued estimator have to be communicated to the sensors at times t_l, $l \in \mathbb{N}$, as these inform the sensors at which times they should communicate their measurements in the timespan $\mathbb{N}_{[t_l, t_l+M]}$. Communicating the schedules to the sensors will incur additional communication. As the schedules are sequences of binary values with length M, the amount of *information* that is to be communicated to the sensors is rather small. The efficiency with which this information can be communicated will strongly depend on the communication protocol that is used, which is beyond the scope of this work.

5.7 Conclusions and outlook

In this chapter, we have proposed a self-triggered set-valued estimator with guaranteed bounds on the estimation error and the number of required measurement updates. As shown in the examples, the estimator is able to exploit its current knowledge of the size of the estimation error in order to decrease the average number of required measurements when compared to a periodic update schedule. The scheme is also able to schedule the measurements of multiple sensors.

Future work includes the improvement of the a priori bounds on the estimation error based on (offline) min-max optimization. Further, the prediction of the estimation error bounds may also be improved by more involved online computations based on solving linear programs similar to those of Problem 5.1, which then leads to a trade-off between computational demand and network usage. Finally, the set-valued estimator might benefit from an increased number of parameters describing the estimation error, when compared to the scalar a_t currently used. This also requires careful study in the future.

Chapter 6

Conclusions, recommendations and final thoughts

In this chapter, we present concluding remarks, recommendations for future work and final thoughts.

6.1 Conclusions

In this thesis, control strategies have been developed that are tailored to control problems where communication and/or actuation resource limitations play an important role. The need for these strategies is motivated by applications and developments in the area of NCSs and sparse control applications. In these areas of control the communication and/or actuation resources available for executing the control tasks are limited. Despite the fact that these resources are scarce, they are still typically used at each periodic sampling instant based on conventional time-triggered control strategies. In this thesis we provided aperiodic control strategies that only require resources when the system really needs attention, e.g., to meet a certain level of control performance or satisfy constraints on states or inputs.

More specifically, we provided systematic design methodologies for optimal sampling, control and communication strategies, where optimality reflects both implementation costs (related to the required resource utilization) and control performance. In fact, several resource-aware control strategies were developed with the following crucial properties:

- significant reductions in resource utilization compared to time-triggered control;

- a priori closed-loop performance guarantees provided by design (in terms of the original cost function), as well as asymptotic stability and constraint satisfaction (if constraints are present);
- simultaneous design of both the feedback law and the triggering condition.

We will now describe these contributions in more detail.

Motivated by the three desired properties, in Chapter 2 we proposed a STC strategy for obtaining sporadically changing input profiles for discrete-time linear systems with (discounted) quadratic cost. Regarding the performance guarantees, the control laws and triggering mechanisms were designed such that an a priori chosen (sub-optimal) level of performance in terms of (discounted) quadratic cost is guaranteed by design. Our proposed methodology provided a solution to the problem of co-design of the feedback law and the triggering condition, a problem hardly addressed in the literature. The designed STC strategy can easily be implemented in practice as it results in a simple piecewise linear control law. The effectiveness of the approach was illustrated by means of a numerical example, showing a significant reduction in the usage of the system's resources, without trading much of the optimally achievable performance. In fact, for the self-triggered LQR strategy, combinations of average resource utilization and control performance levels are obtained, which are not achievable with standard periodic time-triggered LQR solutions. As such, this work is one of the first providing quantitative evidence that aperiodic control strategies, such as the STC strategy proposed in this thesis, can significantly improve beyond time-triggered periodic control. Also in the presence of disturbances, the STC strategy realizes a significant reduction in the usage of network resources with only a slight degradation in performance.

In Chapter 3, the ideas underlying the self-triggered control strategy in Chapter 2 were extended to a novel self-triggered MPC scheme that can deal with discrete-time nonlinear systems (without disturbances) that are subject to state and input constraints and possibly non-quadratic cost functions. In fact, a general framework for the self-triggered MPC strategy was proposed that can be configured to generate both sparse and sporadically changing input profiles, thereby having the ability to serve various application domains. The effectiveness of the approach was illustrated by means of numerical examples, again showing a significant reduction in the usage of the system's resources, without trading much of the achievable performance.

Two resource-aware MPC strategies for discrete-time linear systems subject to state and input constraints exploiting rollout ideas were presented in Chapter 4. For both sparse and sporadically changing input profiles, the strategies provided a solution to the co-design problem. The first strategy allows the user to select a desired level of suboptimality for the performance (in terms of the original control cost function) and minimizes the number of times resources are used to update the input, while guaranteeing the desired level of performance (as well as stability and constraint satisfaction). The second strategy provides a guaranteed (average) resource utilization, while cleverly allocating these resources in order to maximize the control performance (i.e., minimize the control cost). The effectiveness of both approaches was illustrated by means of numerical examples. In these examples, we observe that the first rollout strategy led to significant reductions in the usage of the system's resources, without trading much of the guaranteed achievable performance. The second strategy demonstrates that clever allocation of the available resources can lead to significant reductions of the control cost.

In conclusion, the approaches proposed in Chapters 2-4, provide viable control strategies to balance the usage of the system's resources and control performance beyond the possibilities of standard periodic time-triggered controllers. Interestingly, the presented frameworks are flexible in the sense that they can be configured to generate both sparse and sporadically changing input profiles, thereby having the ability to serve various application domains in which different types of resources are scarce or expensive, e.g., communication bandwidth in networked control systems, battery power in wireless control, or fuel in space or underwater vehicles.

In Chapter 5, we proposed a resource-aware estimator for discrete-time linear systems that reconstructs the plant state based on measurements, in situations where the full state is not available to the controller. To reduce the communication between the sensors and the estimator with respect to a periodic sampling strategy, we proposed a set-valued estimator that employs rollout techniques in deciding at which times to take measurements. Moreover, at each time instant an estimate of the plant state and a guaranteed bound on the difference between the true plant state and the estimate were provided. This information can subsequently be used in a state-based controller. As demonstrated by means of numerical examples, the proposed estimator was able to exploit its current knowledge of the size of the estimation error in order to decrease the average number of required measurements when compared to a periodic update schedule.

6.2 Recommendations for future work

"In theory, there is no difference between practice and theory.
In practice, there is."

J.L.A. van de Snepscheut

Although various relevant contributions to the field of resource-aware control design are made in this thesis, a comprehensive system theory for resource-aware control is currently not (yet) available. Therefore, several recommendations for future research are given below, which can contribute to the deployment of resource-aware controllers in a large variety of control applications.

(Robustness w.r.t. model mismatch) The resource-aware control laws proposed in this thesis employ information about the current state or output in combination with a model of the plant in order to make decisions on future control values and the times at which they are to be applied. A mismatch between the model that is used in the design of the resource-aware control laws and the actual plant to be controlled might lead to errors in both the control values as well as the times at which they are applied. In comparison, for model based-time triggered control strategies only the control values are affected by this mismatch (as the timing at which they are applied is already fixed). The resource-aware control approaches developed in this thesis demonstrate that, by using knowledge about the plant state combined with a model of the plant dynamics, significant reductions in resource utilization can be obtained (without sacrificing much of the obtainable performance). The effects of a mismatch between the model used in the control design and the plant to be controlled were not treated in this thesis. However, this issue requires careful study in the future.

(Robustness w.r.t. disturbances) Similar to a mismatch between the model and the plant to be controlled, also disturbances acting on the system can cause errors in both the control values as well as the times at which these values are applied. In situations where the system is affected by disturbances, it is important to include any available knowledge on these disturbances in the design of the resource-aware controllers. Chapters 2 and 5 study resource-aware control and estimation strategies, respectively, where the presence of disturbances is accounted for in the design of the controller/estimator. Moreover, in [23], a robust self-triggered MPC approach is presented along the lines of the ideas in Chapter 3 and a robust rollout MPC scheme in line with 4 is presented in [22]. In spite of the availability of these results, further exploration of robust resource-aware control techniques is required in order to cope with the many different types of disturbances present in various control applications.

(Including network-induced artifacts) Due to the inherent limitations of resources in NCSs, the results in this thesis offer great potential in this area of control. However, in NCS there are many challenges such as packet dropouts, time-varying delays and communication constraints, which were not considered in this thesis. Of course, by reducing the network usage the effects of these artifacts are smaller. For instance, if the network usage is smaller also the number of packet losses will decrease. Still these artifacts can have a major influence on stability and performance, if they are not carefully addressed in the design of the controller and are deserving a rapidly growing interest in the NCSs literature. In implementing the presented results in the context of NCSs, it is important that these network-induced phenomena are included in the analysis and design of resource-aware control approaches.

(Compare with methods available in the literature) Many interesting results on resource-ware control have appeared in the recent literature, inspired by a large variety of areas in control where resource limitations play a role. Although the effectiveness of the resource-aware control laws in this thesis has been demonstrated by means of various numerical examples, a study comparing the proposed strategies with similar strategies available in the literature is not treated in this thesis. Such a comparison study could provide further insight in the particular advantages and disadvantages of the techniques developed in this thesis.

6.3 Final thoughts

Many theoretical and experimental results appearing in the recent literature on resource-aware control show that these strategies are capable of reducing the utilization of resources required to execute the control tasks, while retaining a satisfactory closed-loop performance. In spite of these recent developments, the actual deployment of this novel control paradigm in relevant applications is still rather marginal. To foster the further development of resource-aware control approaches in the future, it is therefore important to validate these strategies in practice. Moreover, even though many interesting results are currently available, the system theory for resource-aware control is far from being mature, certainly compared to the vast literature on time-triggered (periodic) control. The contributions in this thesis and future explorations, possibly along the lines of the recommendation given above, can lead to such a mature system theory for resource-aware control strategies. However, most importantly, the results in this thesis demonstrate that combinations of average resource utilization and control performance levels can be obtained, which are not achievable with standard periodic time-triggered solutions.

Bibliography

[1] M. Abdelrahim, R. Postoyan, J. Daafouz, and D. Nešić. Co-design of output feedback laws and event-triggering conditions for linear systems. In *Proc. IEEE Conference on Decision and Control*, pages 3560–3565, 2014.

[2] J. Almeida, C. Silvestre, and A.M. Pascoal. Self-triggered state feedback control of linear plants under bounded disturbances. In *Proc. IEEE Conference on Decision and Control*, pages 7588–7593, 2010.

[3] J. Almeida, C. Silvestre, and A.M. Pascoal. Self-triggered output feedback control of linear plants. In *Proc. American Control Conference*, pages 2831–2836, 2011.

[4] J. Almeida, C. Silvestre, and A.M. Pascoal. Self-triggered output feedback control of linear plants in the presence of unknown disturbances. *IEEE Transactions on Automatic Control*, 59(11):3040–3045, 2014.

[5] J. Almeida, C.J. Silvestre, and A.M. Pascoal. Observer based self-triggered control of linear plants with unknown disturbances. In *Proc. American Control Conference*, pages 5688–5693, 2012.

[6] M. Annergren, A. Hansson, and B. Wahlberg. An ADMM algorithm for solving ℓ_1 regularized MPC. In *Proc. IEEE Conference on Decision and Control*, pages 4486–4491, 2012.

[7] A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010.

[8] D.J. Antunes and W.P.M.H. Heemels. Rollout event-triggered control: Beyond periodic control performance. *IEEE Transactions on Automatic Control*, 59(12):3296–3311, 2014.

[9] D.J. Antunes, W.P.M.H. Heemels, J.P. Hespanha, and C.J. Silvestre. Scheduling measurements and controls over networks - part I: Rollout strategies for protocol design. In *Proc. American Control Conference*, pages 2036–2041, 2012.

[10] D.J. Antunes, W.P.M.H. Heemels, J.P. Hespanha, and C.J. Silvestre. Scheduling measurements and controls over networks - part II: Rollout strategies for simultaneous protocol and controller design. In *Proc. American Control Conference*, pages 2042–2047, 2012.

[11] J. Araujo, A. Teixeira, E. Henriksson, and K.H. Johansson. A down-sampled controller to reduce network usage with guaranteed closed-loop performance. In *Proc. IEEE Conference on Decision and Control*, pages 6849–6856, 2014.

[12] K.-E. Arzén. A simple event-based PID controller. In *Proc. IFAC World Congress*, pages 423–428, 1999.

[13] K.J. Åström and B.M. Bernhardsson. Comparison of periodic and event based sampling for first order stochastic systems. In *Proc. IFAC World Congress*, pages 301–306, 1999.

[14] K.J. Åström and B. Wittenmark. *Computer controlled systems*. Prentice Hall, 1997.

[15] J.D.J. Barradas Berglind, T.M.P. Gommans, and W.P.M.H. Heemels. Self-triggered MPC for constrained linear systems and quadratic costs. In *Proc. IFAC Conference on Nonlinear Model Predictive Control*, pages 342–348, 2012.

[16] A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. In *Proc. IEEE Conference on Decision and Control*, volume 2, pages 2133–2138, 1998.

[17] A. Bemporad, W.P.M.H. Heemels, and M Johansson. *Networked control systems*. Springer, 2010.

[18] D. Bernardini and A. Bemporad. Energy-aware robust model predictive control based on noisy wireless sensors. *Automatica*, 48(1):36–44, 2012.

[19] D.P. Bertsekas. *Dynamic programming and optimal control, vol. II*. Athena Scientific, 3rd edition, 2007.

[20] F. Blanchini and S. Miani. *Set-theoretic methods in control*. Birkhäuser, 2008.

[21] F.D. Brunner, T.M.P. Gommans, Heemels, W.P.M.H., and F. Allgöwer. Resource-aware set-valued estimation for discrete-time systems. In *IEEE Conference on Decision and Control*, 2015.

[22] F.D. Brunner, T.M.P. Gommans, W.P.M.H. Heemels, and F. Allgöwer. Communication scheduling in robust self-triggered mpc for linear discrete-time system. In *Workshop on Distributed Estimation and Control in Networked Systems*, 2015.

[23] F.D. Brunner, W.P.M.H. Heemels, and F. Allgöwer. Robust self-triggered mpc for constrained linear systems. In *Proc. European Control Conference*, pages 472–477, 2014.

[24] C.G. Cassandras. The event-driven paradigm for control, communication and optimization. *Journal of Control and Decision*, 1(1):3–17, 2014.

[25] X. Chen and F. Hao. Observer-based event-triggered control for certain and uncertain linear systems. *IMA Journal of Mathematical Control and Information*, 30(4):527–542, 2013.

[26] M. Chyba, S. Grammatico, V.T. Huynh, J. Marriott, B. Piccoli, and R.N. Smith. Reducing actuator switchings for motion control of autonomous underwater vehicles. In *Proc. American Control Conference*, pages 1406–1411, 2013.

[27] R. Cogill. Event-based control using quadratic approximate value functions. In *Proc. Joint IEEE Conference on Decision and Control and Chinese Control Conference*, pages 5883–5888, 2009.

[28] M.H.A. Davis. *Markov models and optimization.* Monographs on Statistics and Applied Probability 49, Chapman and Hall, London, 1993.

[29] V.S. Dolk and W.P.M.H. Heemels. Dynamic event-triggered control under packet losses: The case with acknowledgements. In *Proc. International Conference on Event-based Control, Communication, and Signal Processing*, pages 1–7, 2015.

[30] M.C.F. Donkers and W.P.M.H. Heemels. Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$ gain and improved and decentralized event-triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, 2012.

[31] M.C.F. Donkers, P. Tabuada, and W.P.M.H. Heemels. Minimum attention control for linear systems. *IEEE Discrete Event Dynamic Systems*, 22:1–22, 2012.

[32] A. Eqtami, D.V. Dimarogonas, and K.J. Kyriakopoulos. Event-triggered control for discrete-time systems. In *Proc. American Control Conference*, pages 4719–4724, 2010.

[33] A. Eqtami, D.V. Dimarogonas, and K.J. Kyriakopoulos. Event-triggered strategies for decentralized model predictive controllers. In *Proc. IFAC World Congress*, pages 10068–10073, 2011.

[34] A. Eqtami, S. Heshmati-Alamdari, D.V. Dimarogonas, and K.J. Kyriakopoulos. Self-triggered model predictive control for nonholonomic systems. In *Proc. European Control Conference*, pages 638–643, 2013.

[35] A. Eqtami, S. Heshmati-Alamdari, D.V. Dimarogonas, and K.J. Kyriakopoulos. A self-triggered model predictive control framework for the cooperation of distributed nonholonomic agents. In *Proc. IEEE Conference on Decision and Control*, pages 7384–7389, 2013.

[36] D.J. Franklin, G.F. Powell and A. Emami-Naeini. *Feedback control of dynamic systems.* Prentice Hall, 2006.

[37] M. Gallieri and J.M. Maciejowski. ℓasso MPC: Smart regulation of over-actuated systems. In *Proc. American Control Conference*, pages 1217–1222, 2012.

[38] M. Gallieri and J.M. Maciejowski. Stabilising terminal cost and terminal controller for ℓasso-mpc: Enhanced optimality and region of attraction. In *Proc. European Control Conference*, pages 524–529, 2013.

[39] E.G. Gilbert and K. Tan Tin. Linear systems with state and control constraints: The theory and applications of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36:1008–1020, 1991.

[40] T.M.P. Gommans, D.J. Antunes, M.C.F. Donkers, P. Tabuada, and W.P.M.H. Heemels. Self-triggered linear quadratic control. *Automatica*, 50(4):1279–1287, 2014.

[41] T.M.P. Gommans and W.P.M.H. Heemels. Resource-aware mpc for constrained nonlinear systems: A self-triggered control approach. *System & Control Letters*, 79:59–67, 2015.

[42] R. Gondhalekar and C.N. Jones. Model predictive control of linear periodic systems: A unified framework including control of multirate and multiplexed systems. In *Proc. IEEE Conference on Decision and Control*, pages 6351–6358, 2009.

[43] L. Greco, A. Chaillet, and A. Bicchi. Exploiting packet size in uncertain nonlinear networked control systems. *Automatica*, 48(11):2801–2811, 2012.

[44] E.N. Hartley, M. Gallieri, and J.M. Maciejowski. Terminal spacecraft rendeveuz and capture with ℓasso model predictive control. *International Journal of Control*, 86(11):2104–2113, 2013.

[45] W.P.M.H. Heemels and M.C.F. Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, 2013.

[46] 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*, 58(4):847–861, 2013.

[47] W.P.M.H. Heemels, R.J.A. Gorter, A. van Zijl, P.P.J. van den Bosch, S. Weiland, W.H.A. Hendrix, and M.R. Vonder. Asynchronous measurement and control: A case study on motor synchronization. *Control Engineering Practice*, 7:1467–1482, 1999.

[48] W.P.M.H. Heemels, K.H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *Proc. IEEE Conference on Decision and Control*, pages 3270–3285, 2012.

[49] W.P.M.H. Heemels, K.H. Johansson, and P Tabuada. Event-triggered and self-triggered control. In J. Baillieul and T. Samad, editors, *Encyclopedia of Systems and Control*. Springer, London, 2014.

[50] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, 2008.

[51] J-C. Hennet. Une extension du lemme de Farkas et son application au problème de régulation linéaire sous contraintes. *Comptes-Rendus de l'Académie des Sciences, Série I*, 308:415–419, 1989.

[52] T. Henningsson, E. Johannesson, and A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, 2008.

[53] E. Henriksson, D.E. Quevedo, H. Sandberg, and K.H. Johansson. Self-triggered model predictive control for network scheduling and control. In *Proc. IFAC International Symposium on Advanced Control of Chemical Processes*, pages 432–438, 2012.

[54] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[55] W. Hu, G. Liu, and D. Rees. Event-driven networked predictive control. *IEEE Transactions on Industrial Electronics*, 54(3):1603–1613, 2007.

[56] J. Huang and T. Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

[57] M. Kögel and Findeisen R. On self-triggered reduced-attention control for constrained systems. In *Proc. IEEE Conference on Decision and Control*, pages 2795–2801, 2014.

[58] I. Kolmanovsky and E.G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Egineering*, 4:317–367, 1998.

[59] D. Lehmann. *Event-based state-feedback control.* Logos Verlag, Berlin, 2011.

[60] D. Li, Y. Xi, and P. Zheng. Constrained robust feedback model predictive control for uncertain systems with polytopic description. *International Journal of Control*, 82:1267–1274, 2009.

[61] L. Li and M. Lemmon. Weakly coupled event triggered output feedback control in wireless networked control systems. In *Proc. Allerton Conference on Communication, Control and Computing*, pages 572–579, 2011.

[62] Z. Liu, Q. Wang and D.H. Zhou. A survey of event-based strategies on control and estimation. *Systems Science and Control Engineering*, 2:90–97, 2014.

[63] J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

[64] J.M. Maciejowski. *Predictive control with constraints.* Prentice Hall, 2000.

[65] D.Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[66] D.Q. Mayne, S.V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[67] D.Q. Mayne, J.B. Rawlings, C.B. Rao, and P.O.M. Stockaert. Constrained model predictive control: Stability and optimality. *Automatica*, (36):789–814, 2000.

[68] M. Mazo Jr., A. Anta, and P. Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46:1310–1314, 2010.

[69] N. Meslem and C. Prieur. State estimation based on self-triggered measurements. In *Proc. IFAC World Congress*, pages 86–91, 2014.

[70] S.P. Meyn and R.L. Tweedie. *Markov chains and stochastic stability.* Springer-Verlag, 1993.

[71] M. Miskowicz. *Event-based control and signal processing.* Apple Academic Press Inc., 2015.

[72] A. Molin and S. Hirche. On the optimality of certainty equivalence for event-triggered control systems. *IEEE Transactions on Automatic Control*, 58(2):470–474, 2013.

[73] M. Nagahara and D.E. Quevedo. Sparse representations for packetized predictive networked control. In *Proc. IFAC World Congress*, pages 84–89, 2011.

[74] M. Nagahara, D.E. Quevedo, and D. Nešić. Maximum hands-off control: A paradigm of control effort minimization. *IEEE Transactions on Automatic Control*, 61(4):to appear, 2016.

[75] M. Nagahara, D.E. Quevedo, and J. Ostergaard. Packetized predictive control for rate-limited networks via sparse representation. In *Proc. IEEE Conference on Decision and Control*, pages 1362–1367, 2012.

[76] M. Nagahara, D.E. Quevedo, and J. Ostergaard. Sparse packetized predictive control for networked control over erasure channels. *IEEE Transactions on Automatic Control*, 59(7):1899–1905, 2014.

[77] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Trajectory generation using sum-of-norms regularization. In *Proc. IEEE Conference on Decision and Control*, pages 540–545, 2010.

[78] R. Postoyan, P. Tabuada, D. Nešić, and A. Anta. A framework for the event-triggered stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 60(4):982–996, 2015.

[79] S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. *Chemical Process Control*, 93(316):232–256, 1997.

[80] V. Raghunathan, C. Schurgers, S. Park, and M.B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, 2002.

[81] L. Schenato. To zero or to hold control inputs with lossy links? *IEEE Transactions on Automatic Control*, 54(5):1093–1099, 2009.

[82] J. Sijs, B. Noack, and U.D. Hanebeck. Event-based state estimation with negative information. In *Proc. International Conference on Information Fusion*, pages 2192–2199, 2013.

[83] D. Silvestre, P. Rosa, J.P. Hespanha, and C.J. Silvestre. Self-triggered set-valued observers. In *Proc. European Control Conference*, 2015.

[84] Y.S. Suh, V.H. Nguyen, and Y.S. Ro. Modified Kalman filter for networked monitoring systems employing a send-on-delta method. *Automatica*, 43:332–338, 2007.

[85] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

[86] U. Tiberi and K.H. Johansson. On the robustness of self-triggered sampling of nonlinear control systems. *arXiv preprint 1510.00564*, 2015.

[87] P. Tøndel and T.A. Johansen. Complexity reduction in explicit model predictive control. In *Proc. IFAC World Congress*, 2002.

[88] S. Trimpe and R. D'Andrea. Event-based state estimation with variance-based triggering. *IEEE Transactions on Automatic Control, Special Issue on Control of Cyber-Physical Systems*, 59(12):3266–3281, 2014.

[89] A. Varga. On solving periodic Riccati equations. *Numerical Linear Algebra with Applications*, 15(9):809–835, 2008.

[90] P. Varutti and R. Findeisen. Event-based NMPC for networked control systems over UDP-like communication channels. In *Proc. American Control Conference*, pages 3166–3171, 2011.

[91] P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen. Event-based model predictive control for networked control systems. In *Proc. IEEE Conference on Decision and Control*, pages 567–572, 2009.

[92] M. Velasco, J.M. Fuertes, and P. Marti. The self triggered task model for real-time control systems. In *Proc. IEEE Real-Time Systems Symposium*, pages 67–70, 2003.

[93] M. Velasco, P. Marti, J. Yepez, F.J. Ruiz, J.M. Fuertes, and E. Bini. Qualitative analysis of a one-step finite-horizon boundary for event-driven controllers. In *Proc. IEEE Joint Conference Decision and Control and European Control Conference*, pages 1662–1667, 2011.

[94] J.L.C. Verhaegh, T.M.P. Gommans, and W.P.M.H. Heemels. Extension and evaluation of model-based periodic event-triggered control. In *Proc. European Control Conference*, pages 1138–1144, 2013.

[95] X. Wang and M. Lemmon. Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability. *IEEE Transactions on Automatic Control*, 54(3):452–467, 2009.

[96] W. Wu, S. Reimann, D. Gorges, and S. Liu. Suboptimal event-triggered control for time-delayed linear systems. *IEEE Transactions on Automatic Control*, 60(5):1386–1391, 2015.

[97] W. Wu, S. Reimann, and S. Liu. Event-triggered control for linear systems subject to actuator saturation. In *Proc. IFAC World Congress*, pages 9492–9497, 2014.

[98] J. Yepez, M. Velasco, P. Marti, E.X. Martin, and J.M. Fuertes. One-step finite horizon boundary with varying control gain for event-driven networked control systems. In *IEEE Industrial Electronics Society*, pages 2606–2611, 2011.

[99] J.K. Yook, D.M. Tilbury, and N.R. Soparkar. Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators. *IEEE Transactions on Control Systems Technology*, 10(4):503–518, 2002.

[100] M.S. Zhang, W. Branicky and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems*, 21(1):84–99, 2001.

Summary

Resource-aware control and estimation: An optimization-based approach

Recent developments in computer and communication technologies are leading to an increasingly networked and wireless world. In the context of networked control systems this raises new challenging questions, especially when the computation, communication and energy resources of the system are limited. Similar questions also arise in sparse control, where the system's actuation resources are limited.

In both networked control and sparse control it is important to efficiently use the available resources by limiting the control actions to instances when the system really needs attention. Unfortunately, the classical time-triggered control and estimation paradigms are based on executing sensing and actuation actions periodically in time (irrespective of the state or output of the system) rather than when the system needs attention. Therefore, it is of interest to consider aperiodic control and estimation schemes as a resource-aware alternative, as it seems much more natural to sample the state or output based on well-designed event conditions. By determining the executing times based on state or output information, we effectively bring *feedback* in the sensing, communication and actuation processes. The conditions specifying the event conditions considered in this thesis are of a self-triggered nature, meaning that the controller and/or estimator decide, based on the information currently available, when to schedule the next sensing and/or actuation instance. We will design the triggering conditions leading to optimal sampling, control and communication strategies using an optimization-based approach, where optimality will reflect both implementation costs (related to the number of communications and/or actuator changes) as well as control performance.

Firstly, we propose a self-triggered control strategy for discrete-time linear systems subject to disturbances, based on performance in terms of a quadratic

discounted cost. The presented self-triggered strategy possesses three important features, namely: the control laws and triggering mechanisms are synthesized so that a priori chosen performance levels are guaranteed by design; they realize significant reductions in the usage of communication/actuation resources; and finally, the co-design problem of jointly designing the feedback law and the triggering condition is addressed. The ideas underlying this self-triggered control strategy are also exploited to derive a novel self-triggered control scheme that can deal with nonlinear systems that are subject to state and input constraints (without disturbances).

Secondly, we propose two resource-aware MPC strategies for discrete-time linear systems subject to state and input constraints exploiting rollout ideas. The presented rollout approaches are based on a dynamic programming formulation of the co-design problem of both determining the feedback law and the triggering condition. One of the proposed rollout approaches provides performance guarantees, in terms of the control cost, by design. The other proposed rollout approach provides a guaranteed (average) resource utilization, while cleverly selecting when to use each of the actuators in order to maximize the control performance.

Finally, all the approaches above rely on the availability of the full state through measurements. In case the full state is not available it is of interest to use a resource-aware estimator that reconstructs the plant state based on measurements. The final contribution of the thesis is a self-triggered estimator for discrete-time linear systems subject to unknown but bounded disturbances affecting both the system states and outputs. The proposed self-triggered estimator is a set-valued estimator that employs rollout techniques to reduce the communication between the sensors and the estimator compared to a periodic sampling strategy. Moreover, at each time instant it provides an estimate of the plant state and a guaranteed bound on the difference between the true plant state and the estimate. This information is relevant for further use in state-based controllers such as the ones proposed in other parts of the thesis.

In summary, this work presents several optimization-based approaches to obtain optimal resource-aware control and estimation strategies, where optimality reflects both implementation costs and control performance. The effectiveness of the presented strategies is demonstrated by means of various numerical examples.

Acknowledgements

Een avontuur van vier jaar heeft geleid tot dit proefschrift. Ondanks dat slechts mijn naam op de voorkant staat, hebben velen hier direct of indirect aan bijgedragen. Daarom wil ik een aantal mensen bedanken.

Allereerst wil ik mijn promotor Maurice Heemels en mijn copromotor Duarte Antunes bedanken. Maurice, bedankt voor de prettige samenwerking en je begeleiding. Ik heb veel waardering voor je enthousiasme, expertise en je betrokkenheid gedurende dit avontuur. Duarte, thanks for your guidance, enthusiasm and for sharing your vast know-how during our discussions.

Then, I would like to thank Frank Allgöwer for hosting me at the University of Stuttgart. Frank, thank you for giving me this great opportunity. I have enjoyed my time at the IST very much, for which I'd like to express my gratitude to the IST group, and in particular to Florian Brunner. Florian, thank you for the pleasant and productive collaboration, fist in Stuttgart and later in Eindhoven. Last but not least, thanks for arranging all the logistics for my stay in Stuttgart.

In addition to the people above, I'd like to express my gratitude to Paulo Tabuada, Tijs Donkers and Nick Bauer for their collaboration and contributions.

I would also like to thank the master students Jesús, Jan, Bahadir and Thomas. It has been my pleasure working with you, thanks for your hard work and your contributions to this dissertation.

I extend my sincere gratitude to the members of my reading committee. Ton Backx, Frank Allgöwer, Bart De Schutter and Claudio De Persis, thanks for reviewing this dissertation and for taking part in my promotion.

Een plezierige werksfeer is belangrijk om goed werk te kunnen leveren. Daarom wil ik graag de collega's van CST en D&C bedanken voor de gezelligheid op -1, tijdens de frequente koffiepauzes, gedurende de uitjes en op conferenties. In het bijzonder wil ik Benjamin, Sava en Niek bedanken voor de gezellige sfeer op kantoor en voor de vele discussies, ook al waren deze lang niet altijd aan onderzoek gerelateerd. Special thanks go out to Sinan, Nick, Menno, Bas and Behnam, for their friendship during, and after office hours.

Naast mijn collega's wil ik ook graag mijn vrienden bedanken voor de nodige afleiding en ontspanning. Paul, Erik, Henri, Mieke, Anne, John, Ben en Yvonne, bedankt voor onze langdurige vriendschap, de gezelligheid en de avonturen in de bergen. Bas, ons rondje Noordkaap was een perfect intermezzo op het schrijven van dit proefschrift. Gaaf dat je ook de tent achterop hebt gebonden en mee bent gegaan, het was een mooi avontuur! Sinan, thanks for sharing a passion for bikes and for our many (on/off road) trips. Ceylan, thanks for introducing me to the delicious Turkish cuisine, and especially for putting up with Sinan and me talking about bikes all the time. Dan wil ik ook graag de kaartclub bedanken, Tim, Joep, Lemmen, Willems, nog even en we zijn zo geoefend dat we daadwerkelijk allemaal de spelregels kennen. Maar of we nu een kaartje leggen of niet, dat doet er weinig toe, want gezellig is het toch wel! Rob, Dave bedankt voor jullie vriendschap en de vele gezellige avonden.

Jess, thank you for your endless love and support. Thanks for your words of encouragement and for all the fun adventures!

Tot slot wil ik mijn familie bedanken, Pap, Mam en Bart bedankt voor jullie onvoorwaardelijke vertrouwen en liefde! Zonder jullie steun en aanmoediging was ik nooit zo ver gekomen.

Tom Gommans
December 2015
Eindhoven

Curriculum vitae

Tom Gommans was born on April 14, 1987 in Helmond, the Netherlands. He received both his B.Sc. degree (2008, cum laude) and his M.Sc. degree (2011, cum laude) in Mechanical Engineering at the department of Mechanical Engineering at the Eindhoven University of Technology. As part of this curriculum he carried out a internship in the Centre for Complex Dynamic Systems at the University of Newcastle, Australia. His masters thesis was entitled "Compensation-based control for lossy communication networks" and was supervised by Maurice Heemels, Nathan van de Wouw en Nick Bauer.

In September 2011, Tom started his Ph.D. research in the Control Systems Technology group, at the department of Mechanical Engineering at the Eindhoven University of Technology, under the supervision of Maurice Heemels and Duarte Antunes. The main results of this research are reported in this thesis and focus on resource-aware control and estimation strategies for systems with limited resources. The research is part of the VICI project "Wireless control systems: A new frontier in automation".