

Domagoj Tolić

Candidate

Electrical and Computer Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Rafael Fierro

, Chairperson

Meeko Oishi

Ricardo Sanfelice

Francesco Sorrentino

Estimation and Stability of Nonlinear Control Systems Under Intermittent Information with Applications to Multi-Agent Robotics

by

Domagoj Tolić

B.S. in Electrical Engineering, University of Zagreb, 2005
M.S. in Electrical Engineering, University of Zagreb, 2007

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

**Doctorate of Philosophy
Engineering**

The University of New Mexico

Albuquerque, New Mexico

December, 2012

©2012, Domagoj Tolić

Dedication

To my parents, Verica and Petar, for their support and kind word when most needed.

Acknowledgments

It is a pleasure to thank those who made this dissertation possible. First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Rafael Fierro, for his friendly advice, patience, enthusiasm and encouragement through past four years leading to the completion of this dissertation.

I am also indebted to Prof. Ricardo G. Sanfelice from University of Arizona, Tucson, for his expertise and fruitful discussions that improved Chapter 4. In addition, my gratitude goes to other two members of my Ph.D committee, Prof. Meeko Mitsuko K. Oishi and Prof. Francesco Sorrentino, for their time and helpful comments. This list would not be complete without acknowledging constructive suggestions from Prof. Chaouki T. Abdallah while I was preparing my comprehensive exam.

Finally, a special thanks goes to fellow students from the MARHES Lab: Ivana Palunko, Nicola Bezzo, Titus Appel, Andres Cortez, and Jose Marcio Luna.

Estimation and Stability of Nonlinear Control Systems Under Intermittent Information with Applications to Multi-Agent Robotics

by

Domagoj Tolić

B.S. in Electrical Engineering, University of Zagreb, 2005

M.S. in Electrical Engineering, University of Zagreb, 2007

Ph.D, Engineering, University of New Mexico, 2012

Abstract

This dissertation investigates the role of intermittent information in estimation and control problems and applies the obtained results to multi-agent tasks in robotics.

First, we develop a stochastic hybrid model of mobile networks able to capture a large variety of heterogeneous multi-agent problems and phenomena. This model is applied to a case study where a heterogeneous mobile sensor network cooperatively detects and tracks mobile targets based on intermittent observations. When these observations form a satisfactory target trajectory, a mobile sensor is switched to the pursuit mode and deployed to capture the target. The cost of operating the sensors is determined from the geometric properties of the network, environment and probability of target detection. The above case study is motivated by the Marco Polo game played by children in swimming pools.

Second, we develop adaptive sampling of targets' positions in order to minimize energy consumption, while satisfying performance guarantees such as increased probability of detection over time, and no-escape conditions. A parsimonious *predictor-corrector* tracking

filter, that uses geometrical properties of targets' tracks to estimate their positions using imperfect and intermittent measurements, is presented. It is shown that this filter requires substantially less information and processing power than the Unscented Kalman Filter and Sampling Importance Resampling Particle Filter, while providing comparable estimation performance in the presence of intermittent information.

Third, we investigate stability of nonlinear control systems under intermittent information. We replace the traditional periodic paradigm, where the up-to-date information is transmitted and control laws are executed in a periodic fashion, with the event-triggered paradigm. Building on the small gain theorem, we develop input-output triggered control algorithms yielding stable closed-loop systems. In other words, based on the currently available (but outdated) measurements of the outputs and external inputs of a plant, a mechanism triggering when to obtain new measurements and update the control inputs is provided. Depending on the noise environment, the developed algorithm yields stable, asymptotically stable, and \mathcal{L}_p -stable (with bias) closed-loop systems. Control loops are modeled as interconnections of hybrid systems for which novel results on \mathcal{L}_p -stability are presented. Prediction of a triggering event is achieved by employing \mathcal{L}_p -gains over a finite horizon in the small gain theorem. By resorting to convex programming, a method to compute \mathcal{L}_p -gains over a finite horizon is devised.

Next, we investigate optimal intermittent feedback for nonlinear control systems. Using the currently available measurements from a plant, we develop a methodology that outputs when to update the control law with new measurements such that a given cost function is minimized. Our cost function captures trade-offs between the performance and energy consumption of the control system. The optimization problem is formulated as a Dynamic Programming problem, and Approximate Dynamic Programming is employed to solve it. Instead of advocating a particular approximation architecture for Approximate Dynamic Programming, we formulate properties that successful approximation architectures satisfy. In addition, we consider problems with partially observable states, and propose Particle

Filtering to deal with partially observable states and intermittent feedback.

Finally, we investigate a decentralized output synchronization problem of heterogeneous linear systems. We develop a self-triggered output broadcasting policy for the interconnected systems. Broadcasting time instants adapt to the current communication topology. For a fixed topology, our broadcasting policy yields global exponential output synchronization, and \mathcal{L}_p -stable output synchronization in the presence of disturbances. Employing a converse Lyapunov theorem for impulsive systems, we provide an average dwell time condition that yields disturbance-to-state stable output synchronization in case of switching topology. Our approach is applicable to directed and unbalanced communication topologies.

Contents

List of Figures	xiv
List of Tables	xix
1 Introduction	1
1.1 Motivation	3
1.2 Related Work	4
1.2.1 Motion Planning	4
1.2.2 Estimation Under Intermittent Information	5
1.2.3 Stability Under Intermittent Information	6
1.2.4 Optimal Intermittent Feedback	9
1.2.5 Decentralized Intermittent Feedback	10
1.3 Contributions of Dissertation	12
1.4 Organization of Dissertation	15
2 Marco Polo Game	17

Contents

2.1	Problem Statement and Assumptions	18
2.2	Mathematical Preliminaries and Definitions	21
2.2.1	Hybrid Modeling	21
2.2.2	Cooperative Multi-Target Tracking Modeling	24
2.3	Mobile Hybrid Networks	29
2.4	Simulations and Numerical Results	30
2.5	Conclusion	31
3	Estimation Under Intermittent Information	34
3.1	Problem Statement and Assumptions	36
3.2	Tracking Methodology	38
3.2.1	No-Escape Property of the Approach	44
3.3	Simulation Results	46
3.3.1	Estimation Performance of the Filters	46
3.3.2	Multi-agent Tracking and Filtering Algorithms' Analysis	50
3.3.3	Complexity Analysis	51
3.4	Improving the Marco Polo Game	53
3.5	Experimental Verification of Tracking Methodology	53
3.6	Conclusion	55
4	Stability Under Intermittent Information	66

Contents

4.1	Motivational Example	67
4.2	Problem Formulation and Assumptions	70
4.3	Mathematical Preliminaries	74
4.3.1	Notation	74
4.3.2	Hybrid Systems	75
4.3.3	Types of Stability	75
4.4	Methodology	77
4.4.1	Modeling Approach	77
4.4.2	Why \mathcal{L}_p -gains Over a Finite Horizon?	80
4.4.3	\mathcal{L}_p -Stability of Hybrid Systems	81
4.4.4	Extensions of Previous Work	83
4.4.5	Minimization of $\ A\ $ is a Convex Problem	85
4.5	Input-Output Triggering	87
4.5.1	Input-Output Triggering for Case 1	91
4.5.2	Input-Output Triggering for Case 2	93
4.5.3	Input-Output Triggering for both Case 2 and Case 3	93
4.6	Case Study - Trajectory Tracking	95
4.7	Conclusion	98
5	Optimal Self-Triggering	102
5.1	Problem Statement and Assumptions	104

Contents

5.2	Methodology	107
5.2.1	Input-Output Triggering via the Small Gain Theorem	107
5.2.2	Dynamic Programming	108
5.2.3	Approximate Dynamic Programming - Value Iteration	109
5.2.4	Approximation Architecture	110
5.2.5	Partially Observable States	113
5.3	Case Study - Trajectory Tracking	114
5.4	Conclusion	118
6	Decentralized Output Synchronization	120
6.1	Mathematical Preliminaries	121
6.1.1	Notation	121
6.1.2	Graph Theory	122
6.1.3	Stability Notions	123
6.1.4	Switched Systems and Average Dwell-Time	124
6.2	Problem Statement	125
6.3	Methodology	128
6.3.1	Decentralized Topology Discovery for Directed Graphs	128
6.3.2	Partitioning the Agents	129
6.3.3	Designing Broadcasting Instants	130
6.4	Stability under Switching Topology	136

Contents

6.4.1	Switching without Disturbances	137
6.4.2	Switching with Disturbances	141
6.5	Example	142
6.6	Conclusions and Future Work	145
A	Proofs of main results from Chapter 4	146
References		155

List of Figures

1.1	An illustration of the physical and cyber components interacting in our robotics lab.	16
2.1	Cooperation of UAVs and UGVs.	18
2.2	An illustrative detail of a simulation. Red squares represent UGVs, the green square represents a UGV in the pursuit mode, purple squares depict UGVs, circles denote the corresponding sensing regions, blue polygons represent targets, black rectangles represent obstacles, and the blue shaded cone-like area is a C-target. Red shaded polygons represent areas with poor communication.	32
2.3	The algorithm developed for simulating the Marco Polo game.	33
3.1	Cooperation of UAVs and UGVs illustrating the notation in Chapter 3. .	37
3.2	Parts of the Field-Of-View of an omnidirectional sensor.	43
3.3	The adaptive sampling rate algorithm developed in Chapter 3 for multi-agent tracking scenarios.	57
3.4	A block diagram of the methodology developed for multi-agent tracking scenarios.	58

List of Figures

3.5	Estimates and actual paths using different filters: (a) An estimate of the circular path using the Periodic Filter (#det. = 17); and, (b) An estimate of the circular path using the Adaptive Filter (#det. = 23).	59
3.6	Estimates and actual paths using different filters: (a) An estimate of the zig-zag path using the Periodic Filter (#det. = 100); (b) An estimate of the zig-zag path using the Adaptive Filter (#det. = 25); (c) An estimate of the zig-zag path using a UKF filter (#det. = 25); and, (d) An estimate of the zig-zag path using a PF filter (#det. = 25).	60
3.7	Snapshots of the multi-agent scenario: (a) The initial state of a cooperative scenario including 4 sensors and 2 targets. The simulation snapshot is edited in order to visualize the notation and terminology introduced in Section 3.1; and, (b) The final state of a cooperative scenario including 4 sensors and 2 targets. Purple squares represent sensor platforms, purple circles represent the sensors' FOV boundaries, blue polygons represent targets, while yellow diamonds represent detections. Red dotted circles (i.e., the circle and the line) represent fitting circles, red stars are the targets' waypoints, blue hashed curves are the targets' paths, purple hashed curves are the sensors' paths, green crosses are the initial positions of the targets and green circles are estimates of targets' positions.	61

List of Figures

3.8	Estimation error in time: (a) A sample of the estimation error for the circular path using the Periodic Filter with $\rho = 0.1$ and $\Gamma = 17$. The title of the plot comprises statistics of 50 Monte Carlo simulations; (b) A sample of the estimation error for the circular path using the Periodic Filter with $\rho = 0.3$ and $\Gamma = 17$. The title of the plot comprises statistics of 50 Monte Carlo simulations; (c) A sample of the estimation error for the zig-zag path using the Adaptive Filter with $\rho = 0.1$ and $\Gamma = 3$. The title of the plot comprises statistics of 50 Monte Carlo simulations; and, (d) A sample of the estimation error for the zig-zag path using the Adaptive Filter with $\rho = 0.3$ and $\Gamma = 3$. The title of the plot comprises statistics of 50 Monte Carlo simulations.	62
3.9	Snapshots of the implemented capturing strategy in the comprehensive Marco Polo scenario: (a) The UGV painted green is deployed to capture the target located in the lower-left part of the AOI; and, (b) A snapshot just before the target is captured.	63
3.10	Snapshots of the experiments along with Matlab plots constructed from the experimental data. Notice the correspondence with the real picture. (a) A snapshot of the first experimental setup; and, (b) A snapshot of the second experimental setup.	63
3.11	Experimental data of the quadrotor attitude while tracking waypoints provided by the Adaptive Filter.	64
3.12	Snapshots of the experimental setup involving a quadrotor and Rovio. (a) The beginning of the experiment; (b) Rovio is driving backwards; (c) Rovio is turning; and, (d) The end of the experiment.	65
4.1	An illustration of the trajectory tracking problem considered in Chapter 4.	68

List of Figures

4.2	A diagram of a control system with the plant and controller interacting over a communication network with intermittent information updates. The three switches indicate that the information between the plant and controller are exchanged at discrete time instants belonging to a set \mathcal{T}	70
4.3	A comparison of the intersampling intervals τ_i 's obtained for different notions of \mathcal{L}_p -gains. The abbreviation UG stands for 'Unified Gain'. Red stems indicate time instants when changes in $\hat{\omega}_p$ occur. The solid blue line indicates τ_i 's generated via the methodology devised in this chapter.	71
4.4	Interconnection of the nominal hybrid system Σ_n^δ and the output error hybrid system Σ_e^δ	87
4.5	A realistic scenario illustrating input-output triggering: (a) States x of the tracking system; (b) Norm of (x, e) ; (c) Values of intersampling intervals τ_i 's between two consecutive transmissions. Red stems indicate time instants when changes in δ happen; and, (d) A detail from Figure 5.4(c).	100
4.6	A realistic scenario illustrating input-output triggering using the unified gains: (a) States x of the tracking system; (b) Norm of (x, e) ; and (c) Values of intersampling intervals τ_i 's between two consecutive transmissions. Red stems indicate time instants when changes in δ happen.	101
5.1	A diagram of a plant and controller with discrete transmission instants and communication channels giving rise to <i>intermittent feedback</i>	105
5.2	An illustration of the trajectory tracking problem considered in Chapter 5.	115
5.3	An approximation $\hat{V}^*(\hat{x})$ of the optimal value function $V^*(x)$ for $\omega_p = (1, 1)$ depicted as a function of $\hat{x}_1 \in [-70, 70]$ and $\hat{x}_2 \in [-70, 70]$ when $\hat{x}_3 = 0$	117

List of Figures

- 5.4 An illustration of the optimal input-output triggering: (a) State x of the tracking system; (b) Norm of (x, e) ; (c) Values of the sampling period τ between two consecutive transmissions. Red stems indicate time instants when changes in ω_p happen; and, (d) A detail from Figure 5.4(c). 119
- 6.1 The graph partition $\mathcal{P}_1 = \{1, 3\}$, $\mathcal{P}_2 = \{2, 5\}$ and $\mathcal{P}_3 = \{4\}$ obtained via Algorithm 1. Nodes belonging to different partitions are colored differently. 130
- 6.2 Interconnection of the nominal and the output error dynamics. 134
- 6.3 Outputs of the agents for: (a) Scenario without disturbance; and, (b) Scenario with disturbance. Magenta dots indicate switching instants. 144

List of Tables

3.1	A comparison of standard filters with the Periodic Filter for the circular path	49
3.2	Statistics of the Adaptive Filter for the circular path	50
3.3	A comparison of the standard filters with the Periodic Filter for the zig-zag path	50
3.4	Statistics of the Adaptive Filter for the zig-zag path	50
3.5	A complexity comparison of the standard filters with the filter we developed. RTPSs are obtained with $\#particles = 1000$ for PF, and $\Gamma = 17$ for our filter.	52

Chapter 1

Introduction

Advances in the cyber world such as communications, networking, sensing, computing, storage, and control, as well as in the physical world such as materials, hardware, and power sources, all give rise to a new engineering paradigm – Cyber-Physical System (CPS). The CPS paradigm extends even further the concepts of safety, decentralization, scalability, reconfigurability, and robustness of *systems comprised of subsystems* that share the same physical world [1]. The subsystems of interest can have only physical components (e.g., people, animals, flocks, herds, schools) or both cyber and physical components (e.g., individually accomplished intelligent sensors and autonomous robots equipped with significant communication and computational capabilities). Examples of CPSs are everywhere: transportation systems (e.g., road trains, smart bridges), emergency systems, management of social networks, biological systems, smart sensors, advanced electric power grids, automated manufacturing, entertainment, gaming, haptic systems, etc. As depicted in Figure 1.1, our robotics lab can be thought of as a CPS. Even though control, communication and computation will naturally continue to develop as disciplines on their own right, it is finally time to admit that all those disciplines coexist having the same goal – to satisfy evergrowing demands of the industrial and civil sectors. More than ever, CPS emphasizes the necessity of integrating the control, communication and computation communities, and

Chapter 1. Introduction

relaxing traditionally ingrained boundaries between these communities in order to exploit the full potential of both the cyber and physical world.

In order to address the demands of the modern world articulated within the CPS paradigm, the control community has recently put under scrutiny its fundamental concept – feedback. These efforts tackle the question: “How often should information between systems be exchanged in order to meet a desired performance?” The desired performance can be estimation quality or stability. Estimation under intermittent information is the topic of Chapter 3 while Chapter 4 is concerned with stability of nonlinear control systems under *intermittent information*. Under the term intermittent information we refer to both *intermittent feedback* (a user-designed property of a system as in [2], [3], [4] and [5]) and *intrinsic properties* of control systems such as packet collisions, sampling period, processing time, network throughput, scheduling protocols, delays, lossy communication channels, occlusions of sensors or a limited communication/sensing range (see [6] and the references in [2]). Obviously, intermittent information are present in almost all real-life applications. Therefore, the study of systems under intermittent information is a critical area of research. User-designed intermittent feedback is motivated by rational use of expensive resources at disposition in an effort to decrease energy consumption, and processing and sensing requirements. Consequently, autonomy and the life span of the components increase. We believe this dissertation is a step towards design of low-power and low-cost real-time systems coupled by distributed control laws and estimation schemes resulting in a more robust and reliable performance [7]. In addition, intermittent feedback allows *multitasking* (different functionalities of a system at the same time) by not consuming resources all the time for a sole task. Allocation of the Central Processing Unit (CPU) time between several tasks is known as *task scheduling* [8].

1.1 Motivation

Every day, we witness numerous achievements and improvements in both technology and science. The field of robotics is not an exception. Technology pushes hardware limits further providing processors decreasing in size and increasing in performance, more advanced robots and sensors. As a consequence, networks of commercially available UAVs and UGVs¹ are capable of successfully resolving complicated problems such as search and rescue missions, monitoring urban environments or endangered species, landmine or intruder detection, and pursuit-evasion problems. At the same time, science invents methods and techniques to solve these challenging problems more efficiently. This means better coordination of large heterogeneous robot networks, improvements in planning, sensing and estimation requirements along with higher flexibility, robustness and fault tolerance of the networks.

Managing networks of heterogeneous robots for accomplishing a common goal, such as, detecting, tracking, and eventually capturing one or more targets, autonomously is very challenging because of the limited availability of information about the targets, processing power, and energy at disposition. When the robots are equipped with embedded wireless systems, information about the targets and the workspace can be obtained directly, through onboard sensors and microprocessors, or indirectly, through wireless communication devices that allow the robots to exchange information with each other, or with a central station. Modern technologies and embedded systems allow sensor measurements to be processed and communicated in real time, allowing the robots to estimate environmental and target characteristics with accuracy and precision that can increase rapidly over time. Therefore, the performance of these networks can be significantly improved by decision making approaches that adapt the sensing and motion policies online, based on the latest sensor measurements. Despite the availability of onboard sensors and processors, these policies must also be applicable when little or no target measurements are available,

¹*Unmanned Aerial Vehicles and Unmanned Ground Vehicles, respectively.*

Chapter 1. Introduction

and must minimize energy and processing power, without compromising the estimation performance and/or stability of the control system, in an effort to prolong the mission.

1.2 Related Work

Since the work presented in this dissertation builds on several areas of research, we divide the literature review in several subsections, i.e., motion planning, estimation under intermittent information, stability under intermittent information, optimal intermittent feedback and decentralized intermittent feedback.

1.2.1 Motion Planning

A considerable number of methodologies for coordination of robotic networks and motion planning has been proposed in recent years. Distributed control of synchronous robotic networks with an emphasis on communication protocols and geometric notions relevant in motion coordination are described in [9]. The problem of maintaining connectivity for a dynamic multi-agent network via hybrid modeling is investigated in [10]. An overview of stochastic hybrid models is given in [11]. Planning algorithms are thoroughly explained in [12], while [13] describes autonomous mobile robots from sensing, decision making and application perspectives. A hybrid modeling framework for robust maneuver-based motion planning algorithms for nonlinear systems with symmetries is proposed in [14]. Cell decomposition approaches are covered in [15], whereas a specific case of cell decomposition is implemented in [16]. All aforementioned approaches focus only on certain problems when modeling multi-agent cooperation. For our work, we need a broad, yet simple enough, model of mobile multi-agent networks.

Our hybrid model of multi-agent networks is presented in Chapter 2.

1.2.2 Estimation Under Intermittent Information

Traditionally, the concept of intermittent information along with its repercussions has been a subject of research in the area of Networked Control Systems (NCSs). A comprehensive survey on NCSs is found in [6]. According to [6], NCSs are spatially distributed systems for which the communication between sensors, actuators, and controllers is supported by a shared communication network. From this perspective, intermittent properties might be a consequence of packet losses and delays resulting in out-of-sequence measurements and in partial knowledge of a process of interest. A dynamic programming approach to the concept of intermittent information in NCS settings is considered in [17]. In [17] the authors model each component of an NCS as a potential decision maker that makes control and communication decisions. Robustness of systems with missing measurements is addressed in [18]. In more recent works, [19] investigates estimation for nonlinear dynamical systems over packet-dropping networks while [20] deals with a decentralized detection problem using wireless passive sensors. Several papers address extensions of Kalman Filters (KFs) to scenarios with intermittent observations ([21], [22], [23], [24]). Furthermore, there are several extensions of [23] made by some of its authors. The aforementioned papers consider a linear deterministic process that is being estimated buried in white Gaussian noise where loss of information is modeled as a Bernoulli process (e.g., [22], [23] and [24]) or as a Poisson process (e.g., [21]). An extension of KFs applied to a distributed target tracking problem with a fusion center is proposed in [25] in order to account for unreliable communication channels. The authors in [25] propose an approach in which sensors transmit local sufficient statistics (i.e., a summary of past local observations) rather than raw measurements.

Other works regarding the problem of intermittent information in multi-agent applications are found in [26] and [27]. The authors in [26] investigate intermittent communication using a measurable vector field in a simultaneous tracking and formation control scenario. The work in [27] uses Bayesian Filtering (BF) approximated by Particle Filtering (PF),

Chapter 1. Introduction

intermittent observations and sensors with a limited Field-Of-View (FOV) in a search and tracking scenario. In [27] the authors also deal with a Hospitality Map (HMap) as a way to represent non-analytic information of the environment in which the scenario takes place. Using entropy as the measure of quality of an estimate, nonlinear processes are being estimated and nonlinear measuring methods, as a consequence of limited FOV, are implemented. A cooperative probabilistic search of an area within a sequential decision-making framework based on BF is presented in [28]. Comprehensive studies of PFs are found in [29] and [30]. The idea of combining several models of a target with the aforementioned filters in a unifying framework is known as Interacting Multiple Model (IMM) filter ([31], [32], [33]). Drawbacks of this approach applied to intermittent scenarios are presented in [27]. A relatively high number of targets' models and corresponding filters leads to a greater computational load of the IMM approach [33]. Since we are interested in real-time applications, IMM is not included in this dissertation. The work in [34] deals with a multi-agent coordination under periodic connectivity.

Our approach to estimation under intermittent information is presented in Chapter 3.

1.2.3 Stability Under Intermittent Information

Traditional digital control provides rules of thumb to determine stabilizing sampling periods for linear systems (e.g., 20 times the time constant of the dominant pole [35]) and does not take into account effects of communication between interacting systems. When it comes to nonlinear systems, approximate discrete-time models are derived and analyzed because nonlinear systems, in general, cannot be discretized in closed form [36]. The general consensus, except for [37], is that faster sampling increases stability margins and improves performance of digital control systems. A very educative example where the fast sampling and processing are essential in order to achieve desirable performance is control of UAVs [38].

Chapter 1. Introduction

Recent approaches regarding stability under intermittent information can be classified as follows:

- (i) Small gain theorem approaches [39], [40], [41];
- (ii) Dissipativity or passivity-based approaches [42], [43];
- (iii) Input-to-State Stability (ISS) approaches [5], [44], [45], [46]; and
- (iv) Other approaches [3], [4], [47], [48].

The authors in [3] apply intermittent feedback to Model-Based Networked Control Systems (MB-NCS) in the presence of a state observer, and variable periods (stochastic and with upper bounds) of closed-loop and open-loop control. By analyzing eigenvalues of state matrices, both sufficient and necessary conditions for linear control systems, and sufficient conditions for nonlinear control systems are provided. The work in [4] studies the stabilization problem for a class of nonlinear chaotic systems by means of periodically intermittent control. Using Lyapunov theory and Linear Matrix Inequalities (LMIs), sufficient conditions for exponential stability are provided, and a suboptimal control law is presented. The work in [47] utilizes Lyapunov theory and develops event triggered trajectory tracking for control affine nonlinear systems.

In general, CPSs consist of multiple subsystems. Consequently, they are characterized by multiple time scales. Instead of trying to synchronize all time scales and dealing with time-driven systems, *event-triggered* and *self-triggered* realizations of intermittent feedback are proposed in [5], [44], [45], [46] and [42]. It should be noted that event-driven modeling is traditionally found in the area of computer science being yet another example of the interdisciplinary approach fostered by CPS. In these event-driven approaches, one defines a desired performance, and sampling (i.e., transmission of up-to-date information) is triggered when an event representing imperiled performance occurs. The

Chapter 1. Introduction

work in [46] provides a comprehensive insight into triggered sampling with an emphasis on event-triggering. It applies event-triggering to control, estimation and optimization tasks. The work in [42] utilizes the dissipative formalism of nonlinear systems (see [49] for more), and employs passivity properties of feedback interconnected systems in order to reach an event-triggered control strategy for stabilization of passive and output passive systems. In self-triggered approaches, the current sample is used to determine the next sampling instance, i.e., to predict the occurrence of the triggering event. In comparison with event-triggering, where sensor readings are constantly obtained and analyzed in order to detect events (even though the control signals are updated only upon event detection), self-triggering decreases requirements posed on sensors and processors in embedded systems. Event detection related problems are investigated in [50] and [51]. The authors in [45] extend ideas presented in [5], and develop *state-triggering*: self-triggering based on the value of the system state in the last feedback transmission. Using isochronous manifolds on state-dependent homogeneous systems, a less conservative self-triggered policy (i.e., sampling instants are more similar to the event-triggered ones) is presented in [44]. Finally, the work in [48] investigates self-triggered coordination of robotic networks based on spatial partitioning techniques. Since [48] considers geometric properties of possible future positions of the agents and not the actual controller-plant interconnection, it is more similar to reachability analysis (for example, [52] and [53]) than to the aforementioned works and the work presented in Chapter 4.

The work that has grabbed our attention is presented in [39] and [40]. The authors in [39] and [40] present a framework in which one first designs a controller without taking into account a communication network and then, in the second step, one determines how often control and sensor signals have to be transmitted over the network so that the closed-loop system remains stable. This framework models Networked Control Systems (NCSs) as hybrid systems ([54], [55]), and utilizes the small gain theorem ([56], [57]) to study stability.

Chapter 1. Introduction

In comparison with the approach in [39] and [40], most of the aforementioned approaches appear to be more restrictive and less general in terms of types of stability reached under intermittent information, and requirements on the system in the absence of a communication network. In addition, [39] and [40] consider dynamic controllers, external inputs and output feedback. For example, [3] does not consider systems with external inputs, and the results for nonlinear systems are provided for a class of exponentially stable closed loop systems without communication networks. The authors in [4] do not consider external inputs, and exponential stability and systems with specific nonlinearities are analyzed. The work in [47] investigates state feedback for control affine nonlinear systems and controllers without dynamics. Furthermore, the ISS approaches assume state feedback, controllers without dynamics, and do not consider external inputs. In addition, the results of [45] are applicable to state-dependent homogeneous systems and polynomial systems. The work in [42] analyses passive plants, proportional feedback and does not take into account external inputs.

Our results regarding stability under intermittent information are found in Chapter 4.

1.2.4 Optimal Intermittent Feedback

Similar problems to the problem considered in Chapter 5 are discussed in [58] and [59]. The authors in [58] balance the control performance versus the network cost by choosing the appropriate time delay-controller pair. The work in [59] investigates optimal control of hybrid systems based on ideas from dynamic and convex programming. While [58] associates costs with each of time delay-controller pairs, the work in [59] associates costs with switches between controllers. The optimization methods from [58] and [59] boil down to optimal control of switched systems (see [60] and [61]). However, the results of [60] and [61] are not applicable in Chapter 5. The authors of [61] focus on problems in which a prespecified sequence of active subsystems is given, and then seek both the optimal

Chapter 1. Introduction

switching instants and the optimal continuous inputs. [60] does not make any assumptions about the number of switches nor the mode sequence; instead, they are determined by the solution of the problem. The authors in [60] develop their methodology for problems that include two modes, and provide directions for how to extend the methodology to problems with several distinct modes. However, this extension becomes intractable as the number of modes grows.

Motivated by [61], we adopt Approximate Dynamic Programming (ADP) as the strategy for tackling our problem (see [62] and [63]). ADP is a set of methods for solving sequential decision-making problems under uncertainty by alleviating the computational burden of the infamous *curse of dimensionality* in Dynamic Programming (DP) (see [64] for more). In theory, DP solves a wide spectrum of optimization problems providing an optimal solution. In practice, straightforward implementations of DP algorithms are deemed computationally intractable for most of the applications. Therefore the need for efficient ADP methods. However, comprehensive analyses and performance guarantees of these approximate methods are still unresolved (except in very special settings), and present a critical area of research. In literature, ADP is also known as *reinforcement learning* [65] and *neuro-dynamic programming* [66]. Furthermore, ADP methods are extensively used in *operational research* [64].

Our work regarding the problem of optimal intermittent feedback is found in Chapter 5.

1.2.5 Decentralized Intermittent Feedback

Recent years have witnessed an increasing interest in decentralized control of multi-agent systems (refer to [67], [68], [69], [70], [71], [43] and [72]). Decentralized control is characterized by local interactions among agents, i.e., interconnected systems. Consequently, each agent exchanges information only with its neighbors. When compared with central-

Chapter 1. Introduction

ized control, decentralized control avoids a single point of failure which in turn increases robustness of multi-agent systems, scales better as the number of agents increases, allows for inexpensive and simple agents, lowers the implementation cost, and is sometimes an intrinsic property of multi-agent systems. The problem of synchronizing agents' outputs is a typical problem solved in a decentralized fashion (e.g., [68] and [43]). On the other hand, decentralized control poses significant theoretical and practical challenges due to, among others, limited bandwidth and connectivity of multi-agent networks.

Information exchange among neighbors is instrumental for coordination as discussed in [67], [68], [70], [71], [43] and [72]. According to [73], two models of *wireless networks* supporting distributed (i.e., decentralized) control are typically considered in the literature. The first model is the *radio network model* that buys into worst-case thinking – concurrent transmissions cancel each other because of interference, and potential message collision cannot be detected at a receiver's end. The second model is the *local model* that abstracts away media access issues allowing the agents to concurrently communicate with all neighbors. Clearly, the local model is too optimistic. In order to reconcile these two models, in Chapter 6 we partition the set of agents into subsets with the following property: when all agents in a subset broadcast simultaneously, the wireless network is collision free. Basically, we do not allow agents, that belong to different partitions, to broadcast at the same time due to possibility of message collisions. Consequently, agents in the same partition synchronously broadcast their outputs via wireless. Benefits of synchronous wireless networks, such as a constant bit rate with increased Committed Information Rate (CIR) and Quality of Service (QoS), increased tolerance to interference, low and predictable latency, are recognized in several commercially available wireless services (refer to [74] and [75]). Their asynchronous counterpart leads to unpredictable time-varying communication delays and packet dropouts that must be taken into account when analyzing the performance of multi-agent systems (see [70] and [71]). When communication protocols with Carrier Sense Multiple Access (CSMA) and Collision Avoidance (CA) are used to handle message collisions in asynchronous wireless networks, the communication delays increase and im-

Chapter 1. Introduction

pair the performance of cooperative missions [74]. This increase in communication delays is even more evident as distances between agents increase [74] and in underwater applications [76].

Synchronous wireless is an example of the *time-triggered* communication paradigm [77]. This paradigm excels with respect to predictability, composability, error detection and error containment (see [78] and [77]). Another widely used paradigm is the *event-triggered* communication paradigm which excels with respect to flexibility and resource efficiency [78]. As [78] suggests, a combination of the time- and event-triggered paradigm is often beneficial. In Chapter 6, we take advantage of the predictability in synchronous wireless networks to detect possible changes in the communication topology among the agents. When a receiver does not receive a message in an allotted time interval, we say that an event has occurred and a decentralized topology discovery algorithm is triggered (e.g., [79], [80], [81] and [82]). Notice that event-triggered implementations of decentralized control, such as in [43] and [72], do not allow for switching topology.

Our work regarding the problem of decentralized output synchronization of heterogeneous systems with intermittent communication and switching topology is found in Chapter 6.

1.3 Contributions of Dissertation

In Chapter 2, we present a comprehensive hybrid network model able to capture a wide range of multi-agent problems. By applying the model to a multi-target tracking case study, we demonstrate its versatility and flexibility. The multi-target tracking case study is motivated by the Marco Polo game (introduced in [83]) where a network of mobile robotic sensors must track and capture mobile targets based on the information obtained through cooperative detections of the sensors. This pursuit-evasion game combines cooperative

Chapter 1. Introduction

multi-target tracking, distributed estimation, intermittent sensing/communication and geometric properties of sensor networks. Specifically, we extend the previous work on the Marco Polo game in order to consider more realistic scenarios.

The main contributions of Chapter 3 are threefold:

- a) A computationally feasible real-time tracking strategy that guarantees no-escaping targets while obtaining noisy measurements (detections) of the targets in an adaptive manner;
- b) The estimation strategy that requires less knowledge about targets and noise than the conventional approaches; and
- c) The quality of targets' positions estimate is better or comparable with the conventional approaches.

Our strategy includes *prediction-correction* stages that are found in Kalman Filtering (KF) and Bayesian Filtering (BF). In addition, Chapter 3 complements the approach in Chapter 2 by adding the estimation component which in turn significantly improves the capturing strategy.

The main contributions of Chapter 4 are fourfold:

- a) The design of an input-output triggered sampling policy yielding stability of nonlinear systems employing the small gain theorem;
- b) Consideration of realistic communication channels and sensors in the stability analysis;
- c) The formulation of novel conditions for \mathcal{L}_p -stability (over a finite horizon) of hybrid systems; and
- d) The design of a novel method for calculating \mathcal{L}_p -gains over a finite horizon by resorting to convex programming.

Chapter 1. Introduction

In addition, our approach does not require construction of storage or Lyapunov functions which can be quite a difficult task for a given problem.

The main contributions of Chapter 5 are threefold:

- a) Formulation of the optimal self-triggering problem as a Dynamic Programming (DP) problem;
- b) Employment of Particle Filters (PFs) fed by intermittent feedback to account for partially observable states; and
- c) Formulation of properties that successful approximation architectures in ADP approaches satisfy.

To the best of our knowledge, the problem of optimal intermittent feedback has yet to be addressed.

The contributions of Chapter 6 are fourfold:

- a) The design of broadcasting instants for each partition of agents yielding stability for a fixed topology;
- b) Consideration of directed and unbalanced topologies;
- c) The formulation of an *average dwell time* condition leading to stability with switching topology; and
- d) Stability analysis that takes into account disturbances.

We point out that [43] considers balanced and fixed topologies while [72] considers undirected and fixed topologies.

1.4 Organization of Dissertation

Using the Marco Polo game played by children in swimming pools, Chapter 2 introduces the problem of intermittent information as far as control and estimation is concerned. In Chapter 3 we focus on estimation under intermittent information while in Chapter 4 we investigate stability under intermittent information. Chapter 5 deals with the problem of optimal intermittent feedback. Finally, a decentralized output synchronization problem under intermittent communication and switching topology is investigated in Chapter 6.

Chapter 1. Introduction

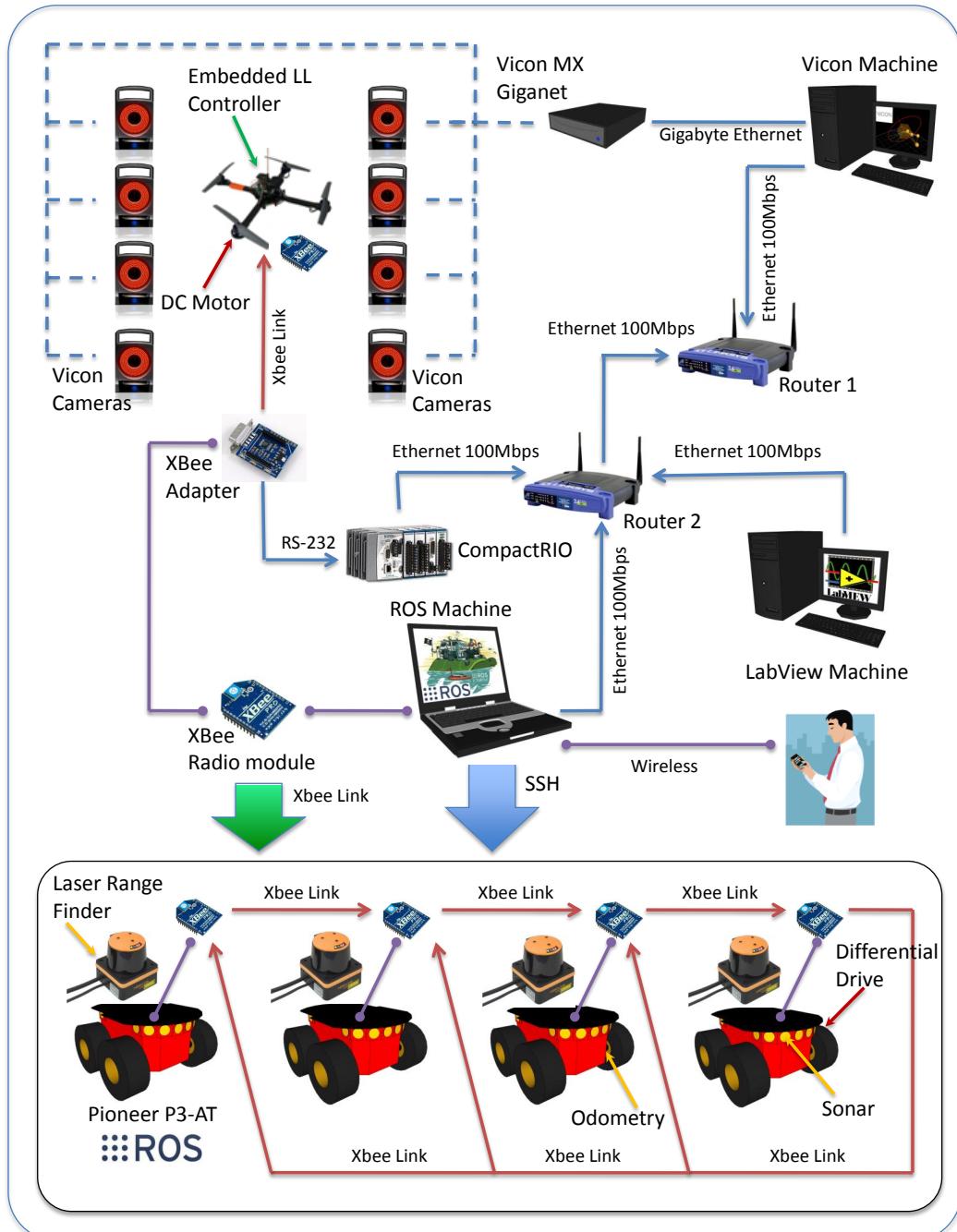


Figure 1.1: An illustration of the physical and cyber components interacting in our robotics lab.

Chapter 2

Marco Polo Game

In this chapter, using the Marco Polo game played by children in swimming pools, we introduce the concept of intermittent information which in turn gives rise to hybrid modeling. We present a comprehensive hybrid network model able to capture a wide range of multi-agent problems. By applying this hybrid network model to a multi-target tracking case study, we demonstrate its versatility and flexibility. The multi-target tracking case study is motivated by the Marco Polo game (introduced in [83]) where a network of mobile robotic sensors must track and capture mobile targets based on the information obtained through cooperative detections of the sensors. This pursuit-evasion game combines cooperative multi-target tracking, distributed estimation, intermittent sensing/communication and geometric properties of the sensor network and environment. Specifically, we extend the previous work from [83] and [84] in order to consider more realistic scenarios. An illustration of a multi-target environment is provided in Figure 2.1 where aerial and ground sensors are used to detect and capture mobile targets in the plane. More details regarding this chapter are found in [85].

The remainder of this chapter is organized as follows. In Section 2.1 we state the multi-agent pursuit-evasion problem and assumptions considered in this chapter. A de-



Figure 2.1: Cooperation of UAVs and UGVs.

tained analysis of the problem is provided in Section 2.2 while stochastic hybrid modeling and geometric optimization methods are proposed to solve the problem. Mathematical details of the developed hybrid model are conveyed in Subsection 2.2.1. Subsection 2.2.2 is reserved for concepts and definitions regarding the application of the hybrid model and geometric methods to the multi-target pursuit-evasion case study. In Section 2.3 we further investigate specifics of the applied hybrid model. Simulations and numerical results are provided in Section 2.4. Finally, we provide conclusions in Section 2.5.

2.1 Problem Statement and Assumptions

The problem considered in this chapter is stated here.

Problem 1 *Given a heterogeneous set \mathcal{P} of N pursuers and a set \mathcal{T} of M targets moving within a specified game area \mathcal{S} , find a set of control policies of sensors which maximizes the*

Chapter 2. Marco Polo Game

total sensing reward, and minimizes the total time required to capture targets in \mathcal{T} that have been positively detected. The objectives of sensors in the detection mode are to (i) avoid obstacles; (ii) maximize the probability of cooperatively detecting unobserved tracks; and (iii) maximize the probability of detecting partially-observed tracks. The objectives of sensors in the pursuit mode are to (i) avoid obstacles; and (ii) minimize the time required to capture a positively detected target based on its fully-observed track and the first $k - 1$ detections, where $k \in \mathbb{N}$.

In this chapter, targets (or evaders) move in a piece-wise straight fashion with uniformly distributed orientations. We require that the scalar product of the initial velocity and current velocity is positive. This requirement assures traversing nature of targets. Moreover, the changes of targets' directions occur randomly within a time interval based on the properties of the targets, environment, and user's preferences. The sensors communicate among themselves when their mode changes or when they detect a target (i.e., event-driven communication). A sensor detects evaders when they enter its circular sensing region (i.e., event-driven measurements). Hence, time instances of the detections are randomly distributed and cannot be anticipated (i.e., intermittent detections). Consequently, a transition from one behavior (i.e., mode) of a sensor to another is based purely on stochastic events. Circular sensors are also called isotropic or omnidirectional sensors. An example of the sensor with a circular Field-Of-View (FOV) is an omnidirectional camera [86].

After obtaining $k - 1$ independent detections of a target, a sensor is deployed to obtain the k^{th} independent observation of the target. We define a C-target as the region in the game area \mathcal{S} where the probability that a deployed sensor will obtain the k^{th} observation of the target is above a threshold ϵ determined by user's preferences (energy at disposal, the desired percentage of captured targets, etc.), and properties of both sensors and targets. C-targets can be approximated with cone-like areas as illustrated in Figure 2.2. In order to cover the C-target with a finite sensing region, the sensor moves orthogonally to the bisector of the C-target within the boundaries of the C-target. We require k independent

Chapter 2. Marco Polo Game

detections before deploying a sensor into pursuit mode in order to avoid false alarms. The decision of which sensor to deploy for either of the aforementioned tasks is based on the reward function (defined in Section 2.2). This reward function is designed for straight line moving targets in [84]. The maximal probability of detecting the targets cooperatively using [84] is obtained when the sensors are grouped in the corners of a rectangular game area (for a comprehensive discussion see [87]). To accommodate the optimality criteria for the targets in this chapter, we maximize the reward function with the constraint that a certain minimal area coverage must be satisfied.

Next, in order to make the problem more realistic, we consider heterogeneous sensors (i.e., sensors with different properties) installed on robotic platforms with different functionalities. The sensors have different sensing regions while the platforms are UGVs and UAVs. Sensors on UGVs have smaller sensing areas than those on UAVs. In addition, UGVs are slower than UAVs, but UGVs can capture targets whereas UAVs cannot. Unlike ground vehicles, aerial vehicles can fly over obstacles. When pursuers are UAVs, a sensor's FOV represents the area in \mathcal{S} monitored by the pursuer while the altitude of the pursuer is kept constant by an autopilot. The properties of both sensors and platforms are taken into account for motion planning. Throughout this chapter, we refer to both the sensors and associated platforms as sensors.

Lastly, we assume that there are areas in the environment where communication is not possible or is very poor. We model these areas as virtual obstacles that sensors would avoid but targets can enter. Virtual obstacles due to communication are introduced in [88].

The uniform distribution is used throughout this chapter because it is ‘the most random’ distribution. The use of any other zero-mean distribution for modeling directions of the targets gives even better results in the sense of fulfilling the problem objectives.

2.2 Mathematical Preliminaries and Definitions

Mathematical preliminaries are divided in two subsections. In the first subsection, we present concepts and details of the stochastic hybrid model developed. In the second subsection, the methods brought together in order to successfully solve the Marco Polo case study are conveyed. We combine cell decomposition, geometric optimization and track coverage into a cohesive framework.

2.2.1 Hybrid Modeling

Roughly speaking, hybrid systems are dynamical systems that involve the interaction of different types of dynamics - discrete jumps and continuous flows. Discrete states are related to different modes of behavior such as sensing, pursuit or obstacle avoidance. Generally, these modes have different goals, continuous dynamics, control laws, sensing and communication policies.

A hybrid automaton is often used as the modeling language for hybrid systems. Merging discussions from [11] and [54], we define the following:

Definition 1 *A hybrid automaton H is a tuple $H = (\mathcal{Q}, \mathcal{X}, f, \Upsilon, \mathcal{U}, \Delta, \mathcal{D}, \text{Init}, \text{Dom}, \mathcal{E}, G, R)$ that describes the evolution of*

- discrete state variables $q \in \mathcal{Q}$ and continuous state variables $x \in \mathcal{X}$,
- control inputs $v \in \Upsilon$ and $u \in \mathcal{U}$, and
- stochastic or disturbance inputs $\delta \in \Delta$ and $d \in \mathcal{D}$

by means of four functions:

- a vector field $f(\cdot, \cdot, \cdot, \cdot) : \mathcal{Q} \times \mathcal{X} \times \mathcal{U} \times \mathcal{D} \rightarrow \mathcal{X}$,

Chapter 2. Marco Polo Game

- a domain set $\text{Dom}(\cdot, \cdot, \cdot) : \mathcal{Q} \times \Upsilon \times v \rightarrow P(\mathcal{X})$,
- a guard sets $G(\cdot, \cdot, \cdot) : \mathcal{E} \times \Upsilon \times \Delta \rightarrow P(\mathcal{X})$, and
- a reset function $R(\cdot, \cdot, \cdot, \cdot) : \mathcal{E} \times \mathcal{X} \times \Upsilon \times \Delta \rightarrow \mathcal{X}$,

where $\text{Init} \subseteq \mathcal{Q} \times \mathcal{X}$ is a set of initial states and $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ is a set of edges.

In the above definition $P(\mathcal{X})$ denotes the power set of \mathcal{X} . Furthermore, we refer to $(q, x) \in \mathcal{Q} \times \mathcal{X}$ as the state of H . We assume that sets \mathcal{Q} , Υ and Δ are countable and that $\mathcal{X} = \mathbb{R}^n$ (or $\mathbb{C}^n, \mathbb{S}^n$), $\mathcal{U} \subseteq \mathbb{R}^m$, and $\mathcal{D} \subseteq \mathbb{R}^p$ for integers n, m , and p , where \mathbb{C}^n is the n -dimensional complex space, and \mathbb{S}^n is the n -dimensional sphere. It should be noted that the function f is a mapping to TX (tangent space of X) given by

$$\dot{x}(t) = f(t, q_0, x_0, x(t), u(t), d(t)), \quad t \in \mathbb{R}_{t \geq 0}, \quad (2.1)$$

and is required to be continuous. With (2.1), the evolution $x : \mathbb{R}_{t \geq 0} \rightarrow \mathcal{X}$ of the dynamical system starting at some initial state $(q_0, x_0) \in \text{Init}$, with input $u : \mathbb{R}_{t \geq 0} \rightarrow \mathcal{U}$ and stochastic input $d : \mathbb{R}_{t \geq 0} \rightarrow \mathcal{D}$, is given. The set of nonnegative real numbers is denoted $\mathbb{R}_{t \geq 0}$.

To characterize the evolution of the state of a hybrid automaton, one needs an appropriate set of times. Such a set has to capture both continuous intervals (over which the continuous evolution takes place) and discrete points in time (when discrete transitions occur). This set of times is called a hybrid time set. From [54], we have,

Definition 2 A hybrid time set is a sequence of intervals $\tau = \{I_0, I_1, \dots, I_N\} = \{I_i\}_{i=0}^N$, finite or infinite (i.e., $N = \infty$ is allowed) such that

- $I_i = [t_i, t_{i+1}], \forall i < N$;
- if $N < \infty$ then either $I_N = [t_N, t_{N+1}]$ or $I_N = [t_N, t_{N+1})$;
- $t_i \leq t_{i+1}, \forall i$.

Chapter 2. Marco Polo Game

In multi-agent applications, each agent can be represented as a hybrid automaton. Such hybrid automata form a Mobile Hybrid Network (MHN) of agents (e.g., networks of sensors and pursuers) that are able to interact within the network and with members of other networks. MHNs can be modeled such that each node represents a mobile agent (sensor or target) with communication, sensing, and control capabilities. Following some ideas from [9], we define an MHN as follows:

Definition 3 *A mobile hybrid network Σ is a tuple $\Sigma = (\mathcal{I}, \mathcal{A}, \mathcal{G}_c, \mathcal{G}_s, \mathcal{G}_h)$, where $\mathcal{I} = \{1, \dots, N\}$ is the set of unique identifiers representing agents in the network, $\mathcal{A} = \{H_i\}_{i \in \mathcal{I}}$ is a set of control systems (physical agents) with processing power modeled as hybrid automata, $\mathcal{G}_c = \{\mathcal{V}, \mathcal{E}_c\}$ is a directed communication graph, where \mathcal{V} is the set of nodes and \mathcal{E}_c is the communication edge map, and $\mathcal{G}_s = \{\mathcal{V}, \mathcal{E}_s\}$ is a directed sensing graph, where \mathcal{E}_s is the sensing edge map. Finally, $\mathcal{G}_h = \{\mathcal{V}, \mathcal{E}_h\}$ is a directed control graph with the set of nodes \mathcal{V} and the control edge map \mathcal{E}_h . If $H_i = H_j, \forall i, j \in \mathcal{I}$, the network is uniform. Otherwise, the network is heterogeneous.*

Several graphs are needed to capture interactions of the agents within the network and environment. In some cases, agents can 'hear' but not 'see' each other. The design of the control graph \mathcal{G}_h involves the assignment of control policies for each agent. The set \mathcal{E}_h is related to the communication \mathcal{E}_c and sensing \mathcal{E}_s graphs. An edge between two nodes in the control graph can be created only if a communication edge or sensing edge exists. Furthermore, processing capabilities of agents (recall the CPS paradigm) bring parallel processing (e.g., coverage optimization), hierarchical structure, (distributed) control and estimation into focus.

Based on the characteristics of the events (upon which the discrete modes of hybrid automata might change) occurring in the network, a hybrid network could be synchronous, asynchronous or a combination of both. That bring us to the following definition:

Definition 4 A synchronous hybrid network is a set of hybrid systems where exists a scheduled increasing sequence of time instants $\mathbb{T} = \{t_k\}_{k \in \mathbb{N}}$ or a sequence of events $\mathbb{E} = \{e_k\}_{k \in \mathbb{N}}$ that take place at $\mathbb{T}_e = \{t_{e_k}\}_{k \in \mathbb{N}}$ when executions of the hybrid automata happen.

On the other hand, there is no such a sequence in asynchronous networks. In the networks that are a combination of both, subsets of the network's elements are not mutually synchronized, but the elements within each subset are synchronized.

The evolution (or solution) of a hybrid automaton could be deterministic, nondeterministic and stochastic. Nondeterministic hybrid automata are those with a certain freedom in defining the solution. For deterministic hybrid automata, we have that, for a given input and initial state, its state is uniquely defined at any instant of time in the future. A refinement of nondeterministic models in order to obtain better analysis of uncertain systems calls for a stochastic hybrid model. In stochastic hybrid models, uncertainties (failures, duration of operations, switching between modes, etc.) are modeled as random variables or random processes in order to include probabilistic phenomena.

2.2.2 Cooperative Multi-Target Tracking Modeling

We consider a pursuit-evasion game where the set \mathcal{P} of N heterogeneous robotic sensors has to detect, pursue and capture M randomly moving targets members of the set \mathcal{T} . Elements of the sets \mathcal{P} and \mathcal{T} are denoted \mathcal{P}_i and \mathcal{T}_i respectively. With $\mathcal{I}_{\mathcal{P}}$ we denote the index set of \mathcal{P} , and with $\mathcal{I}_{\mathcal{T}}$ the index set of \mathcal{T} . The game takes place in a polygonal Area-Of-Interest (AOI) $\mathcal{S} \subset \mathbb{R}^2$ with the boundary $\partial\mathcal{S}$. The area \mathcal{S} is populated by n fixed and convex obstacles $\{\mathcal{O}_1, \dots, \mathcal{O}_n\} \subset \mathcal{S}$. The geometry of the i^{th} pursuer is assumed to be a convex polygon, denoted \mathcal{A}_i , with a configuration $q_{\mathcal{P}_i}(t)$ that specifies its position and orientation at time t with respect to a fixed (or inertial) Cartesian frame $\mathcal{F}_{\mathcal{S}}$ related to \mathcal{S} . When dealing with targets, the position and orientation of the i^{th} target at time t are comprised

Chapter 2. Marco Polo Game

in $q_{\mathcal{T}_i}(t)$. Let us point out that, in general, $q_{\mathcal{P}_i}(t)$ and $q_{\mathcal{T}_i}(t)$ are 3-dimensional vectors, i.e., $q_{\mathcal{P}_i}(t) = [X_{\mathcal{P}_i}(t) \ Y_{\mathcal{P}_i}(t) \ \theta_{\mathcal{P}_i}(t)]^T$, and $q_{\mathcal{T}_i}(t) = [X_{\mathcal{T}_i}(t) \ Y_{\mathcal{T}_i}(t) \ \theta_{\mathcal{T}_i}(t)]^T$. Notice that the orientation of pursuers does not play a significant role since we assume omnidirectional sensors and holonomic sensor platforms. Therefore, we have

$$\dot{q}_{\mathcal{P}_i}(t) = u_i(t), \quad (2.2)$$

where $u_i(t) \in \mathbb{R}^2$ is the input of the pursuer \mathcal{P}_i .

In order to proceed, let us introduce the following definitions. From [89] we have:

Definition 5 A continuous-time random process is a family of random variables $X(t)$ where t ranges over a specified interval of time.

Definition 6 We say that $X(t)$ is a continuous-time Markov process if for $0 \leq s_0 < \dots < s_{n-1} < s < t$ we have $\Pr\{X(t) \in B | X(s) = x, X(s_{n-1}) = x_{n-1}, \dots, X(s_0) = x_0\} = \Pr\{X(t) \in B | X(s) = x\}$ where \Pr denotes probability.

As a consequence of Definition 5, a target \mathcal{T}_i is a continuous-time random process of the random variables $\theta_{\mathcal{T}_i}(t)$, where $t \in [t_0^i, \infty)$ (t_0^i is the time instant when the i^{th} target entered \mathcal{S}). The random variables $\theta_{\mathcal{T}_i}(t)$ describe the target's orientation. A three-dimensional real valued vector function, whose components are continuous functions, maps the family of random variables $\theta_{\mathcal{T}_i}(t)$ into $q_{\mathcal{T}_i}(t)$ using

$$\begin{aligned} \dot{X}_{\mathcal{T}_i}(t) &= v_{\mathcal{T}_i}(t) \cos \theta_{\mathcal{T}_i}(t), \\ \dot{Y}_{\mathcal{T}_i}(t) &= v_{\mathcal{T}_i}(t) \sin \theta_{\mathcal{T}_i}(t), \\ \dot{\theta}_{\mathcal{T}_i}(t) &= 0, \quad t \notin \{t_k\}_{k \in \mathbb{N}}, \\ \theta_{\mathcal{T}_i}(t_k^+) &= \delta_k, \end{aligned} \quad (2.3)$$

and therefore, \mathcal{T}_i is a Markov process. The third component of the vector function experiences uniformly distributed jumps $\delta_k \in \mathbb{R}$ when events t_k , $k \in \mathbb{N}$, occur and does not change between consecutive t_k 's.

Chapter 2. Marco Polo Game

The first two components of \mathcal{T}_i are piece-wise linear, while the third component is piece-wise constant with discontinuities at time instants when orientation changes occur. The implemented time span between two consecutive instants is uniformly distributed in interval $[T_{i,\min}, T_{i,\max}]$, where $T_{i,\min}, T_{i,\max} > 0$ are determined by the user. This ensures absence of Zeno behavior (refer to [55] and [54]). A change of direction δ_k at t_k is uniformly distributed in the interval $(\theta_{\mathcal{T}_i}(t_0^i) - \pi/2, \theta_{\mathcal{T}_i}(t_0^i) + \pi/2)$.

The maximal translational speed of all sensors and targets is known, and $v_{\mathcal{P},\max} > v_{\mathcal{T},\max}$. While sensors can move with any speed in $[0, v_{\mathcal{P},\max}]$, it is assumed that the speed of every target is uniformly distributed in $[v_{\mathcal{T},\min}, v_{\mathcal{T},\max}]$, where $v_{\mathcal{T},\min} > 0$.

Let $\mathcal{C}_{free,i}$ denote the configuration space of the i^{th} sensor that is free of obstacles and other sensors. Let $\mathcal{F}_{\mathcal{A}_i}$ denote a moving Cartesian frame embedded in \mathcal{A}_i . If we assume that \mathcal{B}_i (the geometry of the sensor's Field-Of-View) and \mathcal{A}_i are both rigid, then $q_{\mathcal{P}_i}(t)$ also specifies the position of every point in \mathcal{B}_i (or \mathcal{A}_i) relative to $\mathcal{F}_{\mathcal{S}}$. Using the $k - 1$ individual sensors' detections up to time τ where $k \in \mathbb{N}$, it is possible to identify the area in \mathcal{S} where the sensor may obtain measurements of a target with the probability higher than some threshold ϵ . That leads to the following definition and proposition.

Definition 7 Target \mathcal{T}_j in \mathcal{S} maps in the i^{th} sensor configuration space \mathcal{C} to the C-target region $\mathcal{CR}_j = \{q_i \in \mathcal{C} \mid \Pr\{\mathcal{B}_i \cap \mathcal{T}_i\} > \epsilon, \forall t \geq \tau, i \in \mathcal{I}_{\mathcal{P}}, j \in \mathcal{I}_{\mathcal{T}}\}$.

Proposition 1 C-target can be approximated with cone-like areas (shown in Figure 2.2).

Proof 1 Let \mathcal{F}_i be a Cartesian coordinate system associated with the C-target of target i . Let its y-axis be a minimum squared error line with respect to $k - 1$ detection of the target. Note that this y-axis is a bisector of the C-target. The x-axis contains the point of the $(k - 1)^{\text{th}}$ detection. For the sake of simplicity, let us assume that the change of direction happens periodically so that the distance covered by the target in one period is d . Let us define sums of independent random variables $A_n = \sum_{i=1}^n d \cos \theta_i$ and $B_n =$

Chapter 2. Marco Polo Game

$\sum_{i=1}^n d \sin \theta_i$. Expectation of A_n is 0 and of B_n is $\frac{2nd}{\pi}$ while the variances are $\frac{nd}{2}$ and $n \frac{d\pi^2 - 8d^2}{2\pi^2}$, respectively. Therefore, as n grows, the likelihood that the target could be found further away from the bisector increases. Moreover, the target is more likely to progress along the bisector. Since $\frac{B_n - B_{n-1}}{A_n - A_{n-1}} = \tan \theta_n$, using $\Pr\{0 \leq \tan \theta_n \leq \alpha\} = \frac{1-\epsilon}{2}$ we can approximate the boundaries of the C-target with lines. Their slopes are α and $\pi - \alpha$ with respect to \mathcal{F}_i where $\alpha = \tan(\pi \frac{1-\epsilon}{2})$. Intersections of the lines with the x -axis of \mathcal{F}_i are given by $\pm r_i \epsilon$ where r_i is the radius of the sensor.

A target with $k - 1$ independent detections is called a partially-observed target. A sensor receiving the highest reward is deployed to investigate the C-target of the partially-observed target that makes the maneuvers described in Section 2.1. After gathering k independent observations of the target \mathcal{T}_i , a sensor with the greatest reward is deployed to capture the target. Our goal is to estimate future positions of the target and use a pursuit strategy that maximizes the likelihood of capturing the target (this motivates the work presented in Chapter 3). Therefore, based on k intermittent observations of the target's position, the following capturing policy is proposed and implemented in this section: *Move to the point of the last detection. Afterwards, move to the intersection of minimum squared error line and $\partial\mathcal{S}$.* The error that is minimized is the sum of squared distances of k detection points and the line.

In order to reduce the computational complexity due to the uncountable space \mathbb{R}^2 , a cell decomposition of \mathcal{S} is implemented. We discretize the environment using the uniform cell decomposition. The uniform cell decomposition is used because of its implementation tractability and optimal dispersion (see [12]). This optimal dispersion (with respect to the Euclidean norm) is important since the sensing regions are circles with finite areas. From the cell decomposition, a connectivity graph \mathcal{G} is obtained. Every sensor has its own connectivity graph. Cells forming $\mathcal{C}_{free,i}$ are divided into void and observation cells. Void cells are cells with the probability of detection of partially-observed targets less than the threshold ϵ , while observation cells are those with the probability larger than ϵ . The

Chapter 2. Marco Polo Game

connectivity graph as well as void and observation cells of each sensor have to be updated as the game progresses. Cells in the decomposition (nodes of the graph) are denoted k_i , and sensors (except when in the pursue mode and when being initially placed) move among the cells' centroids.

Next, we define the underlying performance index in order to achieve the goals stated in Section 2.1. The sensing objectives are expressed in terms of a reward function that represents the improvement in the overall probability of detection that would be obtained by moving from a configuration $q_{\mathcal{T}_i}(t_1) \in k_l$ to a configuration in an adjacent cell $q_{\mathcal{T}_i}(t_2) \in k_i$ (obviously, $t_1 < t_2$) taking into account distance. The reward function is as follows:

$$R(k_l, k_i) = P_{\mathcal{R}}(k_i) + \Delta P_{\mathcal{S}}^k(k_l, k_i) - d(k_l, k_i), \quad (2.4)$$

where $P_{\mathcal{R}}$ is the probability of cooperatively detecting a partially-observed target, $\Delta P_{\mathcal{S}}^k$ is the gain in the probability of cooperatively detecting unobserved targets and $d(k_l, k_i)$ is the distance between centroids of the cells. An unobserved target is a target with less than $k - 1$ independent observations. These probability density functions are obtained using the methodology based on geometric properties of sensors and the area-of-interest described in [84]. The performance index (2.4) is maximized with the constraint that minimal area coverage has to be satisfied as stated in Section 2.1. This is due to the fact that the maximal value of $P_{\mathcal{S}}^k$, contributing as the difference in (2.4), is obtained when the sensors are grouped in the corners of \mathcal{S} . Based on the reward function, a sensor with maximal reward along a path in the graph is deployed. The terms in the reward function are weighted based on user's preferences. While choosing a sensor for pursuit, more weight is put on the distance term. In order to find a sensor \mathcal{P}_i with maximal path reward, we use the graph searching algorithm A*. After finding the optimal sensor, we determine a control input $u_i(t)$ in (2.2) that corresponds to the optimal path.

2.3 Mobile Hybrid Networks

The sensors (or pursuers), considered in the previous section, form an MHN denoted $\Sigma_{\mathcal{P}}$, and the targets (or evaders), considered in the previous section, form an MHN denoted $\Sigma_{\mathcal{T}}$. The sensors are fully connected, and the associated control graph is omitted since the sensors do not perform a coordinated motion as a group (e.g., keeping some formation) to accomplish the goal. Through communication, sensors exchange their current position. Therefore, a sensing graph is redundant. However, the sensing capability is essential for target tracking. The sensors are able to sense targets (i.e., a sensing between nodes of different networks takes place). Every new observation triggers an exchange of information between the sensors causing mode changes, i.e., network $\Sigma_{\mathcal{P}}$ is synchronized by events related to the targets. On the other hand, collision avoidance with other sensors yields $\Sigma_{\mathcal{P}}$ asynchronous behavior.

When considering the sensors as processing units, estimation of the targets' position and sensors' motion planing are distributed among the sensors. In other words, the sensor group estimates the random processes (i.e., targets' positions) joining collected information, while the control policy for each sensor is obtained considering only its configuration space. Each sensor calculates its reward function given by (2.4) and communicates it to others. The ground sensors' modes are:

- *sensing (static or mobile with obstacle avoidance),*
- *pursuing (with obstacle avoidance), and*
- *communicating and updating information.*

Aerial sensors do not have *pursuing* mode. Collisions among the sensors are avoided using model prediction. Since sensors' motion planing is distributed, such collisions are possible. Using one step look ahead, collisions are avoided by switching to *avoiding obstacles* mode.

Chapter 2. Marco Polo Game

On the other hand, we assume that the targets cannot communicate with each other, and each target is independent from other targets. Hence, network $\Sigma_{\mathcal{T}}$ is asynchronous. Evolution of $\Sigma_{\mathcal{T}}$ is stochastically modeled as described in Section 2.2 (see Definitions 1 and 3). The targets are only capable of sensing obstacles in their vicinity. Targets' modes are:

- *active* (*obstacle avoidance* or *changing direction*), and
- *captured*.

Collision avoidance among the targets and obstacles is obtained using model prediction. Targets 'look' one step ahead, and if there is a collision, they switch to the *obstacle avoidance* mode. The algorithm in Figure 2.3 shows relations between the aforementioned modes and events that trigger transitions from one mode to the other.

2.4 Simulations and Numerical Results

The information-driven sensor planning and pursuit strategies described in previous sections are integrated in a simulator. A pseudo-code of the implemented algorithm is shown in Figure 2.3. We use $k = 3$ since it is found in [90] that, from a geometric point of view, it is a convenient number of detections for estimating target tracks in the absence of false alarms. The initial positions of the sensors maximize the value of $P_{\mathcal{S}}^k$ constrained to the required area coverage. The environment is a square measuring 10 by 10 meters, while the sensors have the area of 0.25 square meters. Two sensors are UAVs with the sensing radius 1.5 m and five sensors are UGVs - three with the sensing radius 1 m and two with the sensing radius 1.25 m. Initially, all sensors are in the *static sensing* mode and each UGV is a candidate to switch to the *mobile sensing* mode, i.e., the *pursuit* mode. Three targets enter the environment with uniformly distributed locations along $\partial\mathcal{S}$, headings and velocities.

Chapter 2. Marco Polo Game

Since sensors present obstacles to each other, obstacle maps are not static and \mathcal{C}_{free} has to be determined for each sensor and updated during the game. The computations of obstacle and coverage maps are time consuming and there is no need to perform them frequently since the sensors perform optimal maneuvers (short in duration), and their position is optimal (they are waiting for the targets). Hence, these computations are performed periodically as stated in Figure 2.3.

Robustness of the proposed strategies is verified by introducing noise in the targets' position and velocity. As a consequence, obtained results are slightly impaired meaning that fewer targets are successfully captured comparing to scenarios without noise.

2.5 Conclusion

This chapter presents a comprehensive stochastic hybrid model of mobile agent networks able to capture a wide range of multi-agent phenomena. A hybrid network consists of mobile agents modeled as hybrid systems with processing capabilities. Versatility and flexibility of the model are demonstrated by the cooperative multi-target tracking case study where demanding tracking and pursuit goals are fulfilled.

Chapter 2. Marco Polo Game

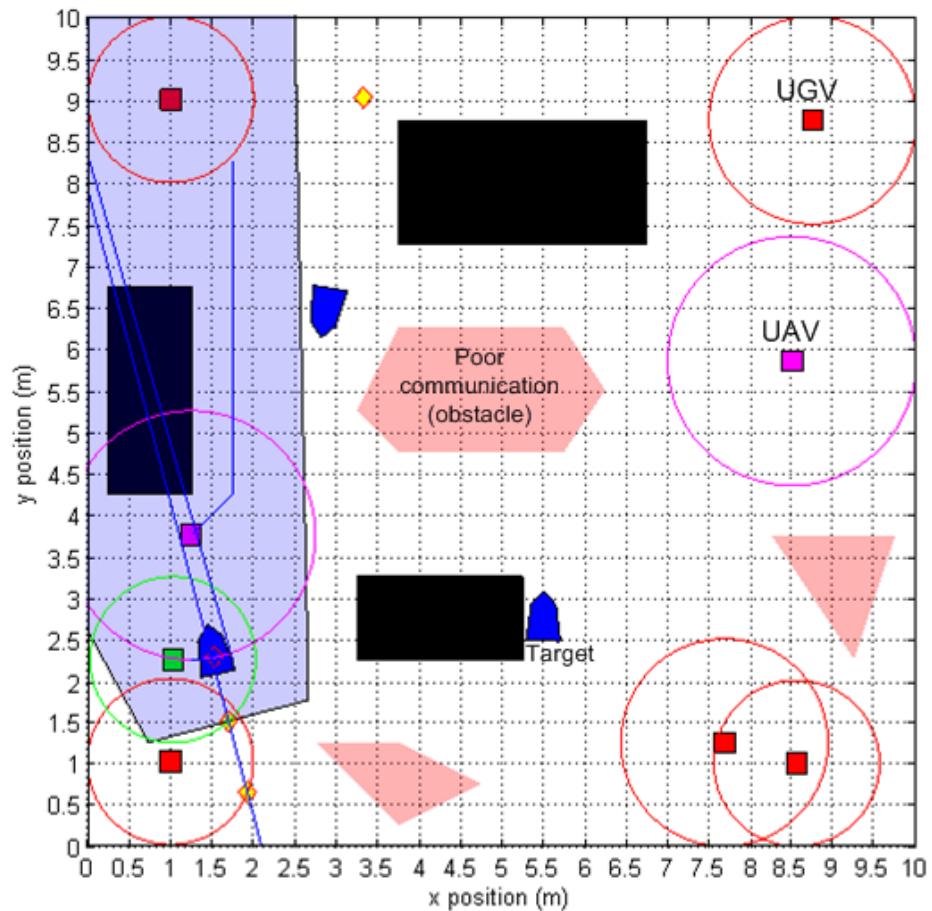


Figure 2.2: An illustrative detail of a simulation. Red squares represent UGVs, the green square represents a UGV in the pursuit mode, purple squares depict UGVs, circles denote the corresponding sensing regions, blue polygons represent targets, black rectangles represent obstacles, and the blue shaded cone-like area is a C-target. Red shaded polygons represent areas with poor communication.

Chapter 2. Marco Polo Game

```

1. Perform initial optimal sensor placement
2. Decompose environment into  $C_{free}$  and  $C_{obstacle}$  cells
3. for all Sensors do
4.     Calculate obstacle and coverage maps
5. end for
6. while Game not over do
7.     for all Targets do
8.         if Time for change of direction then
9.             Change direction
10.            end if
11.            Update position (avoid obstacles, collisions and add noise)
12.        end for
13.        for all Sensors do
14.            if Sensor update interval then
15.                Calculate obstacle and coverage maps
16.            end if
17.            Update position (do maneuver, avoid obstacles and collisions)
18.            if Position changed then
19.                Update sensors' network information
20.            end if
21.            Detect targets
22.            if A target beneath capture threshold then
23.                Remove target
24.                Update sensors' network information
25.                End associated pursuit or investigations
26.            end if
27.        end for
28.        if Pursued target beneath capture threshold then
29.            Remove target
30.            End pursuit
31.            Update sensors' network information
32.        end if
33.        if Detection then
34.            Update sensors' network information
35.            if Target detections =  $k - 1$  then
36.                Hypothesize target track
37.                Calculate observation cells
38.                for all Sensors that have not detected this target do
39.                    Calculate path and reward to investigate target
40.                end for
41.                Deploy a sensor with the greatest reward
42.                Determine maneuver
43.            else if Target detections =  $k$  then
44.                for all Sensors not in pursuit do
45.                    Calculate path and reward to pursue target
46.                end for
47.                Deploy a sensor with the greatest reward
48.            end if
49.        end if
50.    end while

```

Figure 2.3: The algorithm developed for simulating the Marco Polo game.

Chapter 3

Estimation Under Intermittent Information

In this chapter, we develop a geometric approach for estimation of mobile agents' positions considering motions of the mobile agents as higher order Markov chains. From kinematic models of mobile agents, one can infer possible geometries of targets' maneuvers. Since every maneuver takes time for completion, we interpret maneuvers as higher order Markov chains. Due to intermittent information, we are not able to capture every erratic maneuver, but rather try to capture tendencies in the motion of mobile agents. Erratic maneuvers are not energy efficient, and therefore, represent unnatural behavior. In addition, erratic maneuvers are usually a consequence of poorly designed control systems, and are unwanted behaviors of agents. Our approach does not require the exact model of evaders nor measurement and process noise, and is applicable to a class of evaders. The only assumptions on the measurement noise are to be white, zero-mean (a non-zero mean noise can always be transformed into a zero-mean noise based on the sensor's characteristics) and with finite variance (fulfilled for all commercially available sensors). Notice that Kalman Filtering (KF) and Bayesian Filtering (BF) exploit the (first order) Markov property of a process being estimated.

Chapter 3. Estimation Under Intermittent Information

The Bayesian inference is a well established field on its own right along with its drawbacks (see, for example, the discussions in [12] and [91]). Bayesian approaches use probability density functions (pdfs), prior and importance distributions (i.e., importance sampling), and can be applied to nonlinear, nonadditive and non-Gaussian settings. Tracking problems that utilize KF or BF are usually homogeneous (with no inputs). In order to perform complex maneuvers and motions, robots need inputs. Consequently, applications of either KF or BF assume the knowledge of robots' inputs. Also, KF and BF require fairly accurate models of noise and evader's dynamics in order to converge. In real-life applications, such knowledge is rarely available (e.g., unknown or intelligent evaders). The more intermittent information become, the greater process noise levels in KF and BF have to be for the filters to converge and work properly in order to account for the greater uncertainty. Levels of the process noise might become so great and conceal accurate models of a process and noise. Hence, it is questionable whether the accurate modeling is crucial when applying the conventional filters to scenarios with intermittent measurements.

A target with purely stochastic behavior can be interpreted as a target with a certain level of intelligence. In such a case, one may consider a game-theoretic approach. A brief overview of game-theoretic approaches in robotics is given in [12]. Our choice of targets with random decisions (behaviors) is justified by one of the most important result of game theory – a saddle point always exists for a zero-sum game. The importance of the information available to the players on the outcome of a pursuit-evasion game is investigated in [92]. From the game-theoretic viewpoint, the goal of a target is to maximize the distance from the sensors, while the goal of the sensors is the opposite. Since in this section we are interested in intermittent information, the results of game theory are used merely for justifying and interpreting strategies and behaviors of sensors and targets (e.g., targets are not aware of sensors, but perform unpredictable motions). For instance, in order not to lose a target from its FOV, a sensor makes decisions to prevent the target to slip out of the sensor's FOV (a worst-case decision strategy).

This chapter differs from the large portion of existing literature because intermittent information is not a stochastic process that agents cannot control. Based on this observation, it is similar to [17], game theory and the idea of intermittent feedback. Sensors decide when to obtain new measurements of targets and plan their motion. Hence, we change the cause and the consequence from the NCS point of view, but the problem remains the same - how to accurately estimate targets' positions with the most rational use of expensive resources? The answer to this question leads to a better understanding of intermittent information concepts.

The remainder of the chapter is organized as follows. Section 3.1 states the problem of estimation under intermittent information and assumptions considered herein, presents a novel approach for solving the problem, and provides a stability analysis of the approach. We compare the estimation performance, computational load, scalability and complexity of UKF, BF and of our filter under intermittent information in Section 3.3. In Section 3.4 we incorporate our tracking filter in a complex heterogeneous scenario based on the Marco polo game enhancing the work presented in Chapter 2. Experimental results are presented in Section 3.5. Finally, we provide a summary in Section 3.6.

3.1 Problem Statement and Assumptions

In order to make this section self contained and since this section has more general assumptions than Chapter 2, we repeat some details already stated in Chapter 2.

We consider a pursuit-evasion game where the set \mathcal{P} of N heterogeneous robotic sensors has to detect, track and eventually capture M randomly moving heterogeneous targets members of the set \mathcal{T} . Elements of the sets \mathcal{P} and \mathcal{T} are denoted \mathcal{P}_i and \mathcal{T}_i , respectively. With $\mathcal{I}_{\mathcal{P}}$ we denote the index set of \mathcal{P} , and with $\mathcal{I}_{\mathcal{T}}$ the index set of \mathcal{T} . The game takes place in a polygonal area-of-interest (AOI) $\mathcal{S} \subset \mathbb{R}^2$ with boundary $\partial\mathcal{S}$. The area \mathcal{S} is

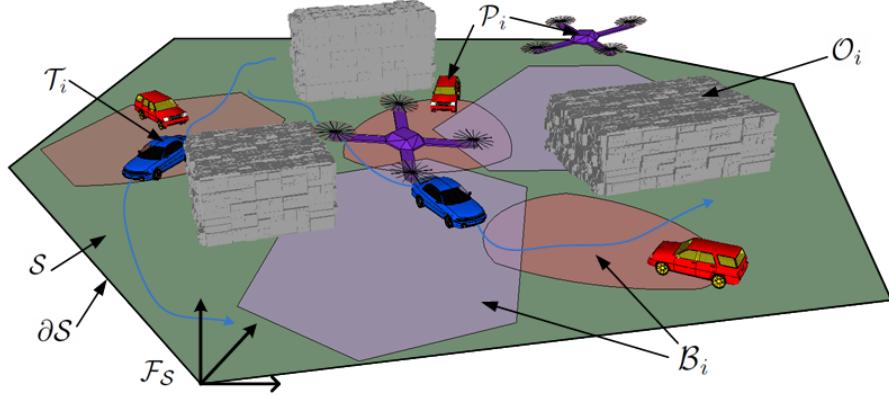


Figure 3.1: Cooperation of UAVs and UGVs illustrating the notation in Chapter 3.

populated by n fixed and convex obstacles $\{\mathcal{O}_1, \dots, \mathcal{O}_n\} \subset \mathcal{S}$. The geometry of the i^{th} pursuer's platform is assumed to be a convex polygon denoted by \mathcal{A}_i , with a configuration $q_{\mathcal{P}_i}(t)$ that specifies its position and orientation at time t with respect to a fixed (or inertial) Cartesian frame $\mathcal{F}_{\mathcal{S}}$ related to \mathcal{S} . Similarly, a FOV of the i^{th} pursuer is assumed to be a convex set denoted by \mathcal{B}_i . For the sake of simplicity, we will use $\mathcal{A}_i(t)$ and $\mathcal{B}_i(t)$ instead of expressing them through $q_{\mathcal{P}_i}(t)$. When modeling targets, position and orientation of the i^{th} target at time t are comprised in $q_{\mathcal{T}_i}(t)$. An illustration of the pursuit-evasion game is given in Figure 2.1.

The problem considered herein can be postulated as follows:

Problem 2 Given a set \mathcal{P} of N mobile sensors with dynamics $\dot{q}_{\mathcal{P}_i}(t) = f_i(t, q_{\mathcal{P}_i}, u_i), i = 1, \dots, N$, and FOVs $\{\mathcal{B}_1(t), \dots, \mathcal{B}_N(t)\}$ that are deployed to track, and eventually capture a set \mathcal{T} of M targets with dynamics $\dot{q}_{\mathcal{T}_j}(t) = g_j(t, q_{\mathcal{T}_j}, u_j), j = 1, \dots, M$, find a set of control policies $\{u_1(t), \dots, u_N(t)\}$ and find sampling policies $\delta_j^i(t) \in \{0, 1\}$ for the mobile sensors resulting in minimal exposure of sensors and no-escaping targets that are being tracked. Next, find (or choose from the existing ones) an estimation method that takes the best use of intermittent measurements and limited knowledge of targets (e.g., intelligent and unknown targets) with respect to estimation quality, complexity, scalability

and computational load of the method.

Note that the estimation method is independent of the sampling policy because intermittent measurements can be a consequence of environment (e.g., target occlusion) or characteristics of sensors and communication channels (e.g., collisions and channel throughput). A sampling policy can be just one of many reasons for intermittent measurements. The sampling policy considered in this section is motivated by the Marco Polo game presented in Chapter 2.

3.2 Tracking Methodology

We use the stochastic hybrid model of mobile agents derived in Chapter 2 to formalize the methodology presented in this section. A sensor (pursuer) \mathcal{P}_i has the following model:

$$\dot{q}_{\mathcal{P}_i}(t) = u_i(t), \quad (3.1)$$

where $u_i(t) \in \mathbb{R}^2$ is the input of the pursuer \mathcal{P}_i . Sensors can be in *idle* or in *tracking* mode. An idle sensor is a sensor not tracking any target at the moment.

Targets (evaders) are modeled as unicycles:

$$\begin{aligned} \dot{X}_{\mathcal{T}_i}(t) &= v_{\mathcal{T}_i}(t) \cos \theta_{\mathcal{T}_i}(t), \\ \dot{Y}_{\mathcal{T}_i}(t) &= v_{\mathcal{T}_i}(t) \sin \theta_{\mathcal{T}_i}(t), \\ \dot{\theta}_{\mathcal{T}_i}(t) &= \omega_{\mathcal{T}_i}(t), \end{aligned} \quad (3.2)$$

where $v_{\mathcal{T}_i}(t) \in [v_{i,\min}, v_{i,\max}]$ and $\omega_{\mathcal{T}_i}(t) \in [\omega_{i,\min}, \omega_{i,\max}]$ are inputs. In our work we assume that $v_{i,\min}, v_{i,\max}, \omega_{i,\min}, \omega_{i,\max} \in \mathbb{R}$ for every $i \in \{1, \dots, N\}$. The position (i.e, the first two components of $q_{\mathcal{T}_i}$) of the i^{th} target is denoted $q_{\mathcal{T}_i}^p$. For starters, let us assume that $v_{\mathcal{T}_i}(t)$ and $\omega_{\mathcal{T}_i}(t)$ cannot be described probabilistically (i.e., pdfs are not given and cannot be inferred by via learning methods). The probabilistic model of targets from Chapter 2 is considered in Section 3.4.

Chapter 3. Estimation Under Intermittent Information

The targets (3.2) can be considered as a generalization of the Reeds-Shepp car. A difference is that, with zero linear velocity, targets can rotate while being at the same position. More generally, a turning radius $r(t)$ is not fixed but is a function of linear and angular velocity, i.e., $r(t) = v(t)/\omega(t)$. Hence, the model in (3.2) is fairly general and possesses behaviors of holonomic (Section 3.4), car-like, and simple dynamical (Section 3.5) models of vehicles. According to [12], the (shortest) path of the Reeds-Shepp car consists of circles and lines. Therefore, we approximate targets' paths with circles where lines are circles having infinite radius. Using the method explained in [93], a least square algorithm for fitting a circle to a number of given points is implemented. For more regarding least square fitting curves refer to [94].

Sensors' estimation of the i^{th} target's position is based on the fitting circles (a *prediction stage*). Our algorithm considers Γ_i detections, where $\Gamma_i \geq 3$ is a natural number, when calculating a fitting circle. Also, targets' velocities are being estimated from the same set of detections. In other words, sensors are estimating both targets' velocities and positions, while the targets' orientations are tangents to corresponding fitting circles. It should be pointed out that last Γ_i observations are not always taken for estimation. If an estimate using some older Γ_i detections is precise enough (comprised in ρ_j using the expression given in the algorithm in Figure 3.3), there is no need to perform new estimation. However, if there is a discrepancy between the current estimate and the latest detection, the corresponding fitting circle is updated by taking into account more recent detections (a *correction stage* using novel information). If even now the estimate is not accurate enough, obtain new detections until the estimate is accurate enough. Hence, our filter decides when to obtain new measurements based on the quality of the estimated velocity and position. This is the adaptive feature of the filter.

A detection can be obtained only if a target is inside a sensor's FOV, and at time instances that are multiples of T_s . Period T_s accounts for limited sensing and processing frequencies of pursuers. Idle pursuers make detections of targets entering their FOVs be-

Chapter 3. Estimation Under Intermittent Information

cause an idle sensor detects changes occurring in 'Escape Area' of its FOV (defined with (3.8)) on every multiple of T_s . The motivation is to decrease processing requirements. If several targets are entering the FOV, randomly choose which one to follow.

Since we are implementing intelligent pursuers (rational decision makers), the best strategy for them comes from the game theory approach. Based on the current detection and the knowledge of the maximum linear velocity of a target, a sensor calculates the time t_i^{esc} the i^{th} target needs to leave its FOV (using expression (3.4) and $\max\{|v_{i,\min}|, |v_{i,\max}|\}$, where $|\cdot|$ is the absolute value) and, based on that data, makes a new detection. We name this version of the filter Adaptive Filter. We also develop another version of the filter based on the fitting circle approach that makes observations of targets periodically. For that reason, the sensors are not decision makers, but blindly follow targets in this version. We name this version Periodic Filter. The motivation for this version is the fact that conventional filters (KF and BF) in literature sample processes periodically; therefore, a comparison with the existing filters is more relevant. In addition, the sampling period can easily be varied and the corresponding performance analyzed. For an estimate of targets' future positions, we use the expected value obtained using fitting circles.

More formally, we have a Markov process \mathcal{T}_i given by (2.3) and its position is being measured by a sensor \mathcal{P}_j modeled as follows:

$$y_j^i(kT_s) = H_j(q_{\mathcal{T}_i}(kT_s)) + \nu_j(kT_s), \quad (3.3)$$

where $k \in \{0, 1, 2, \dots\}$. Function H_j is highly nonlinear since the sensor's FOV is limited. From (3.3), it is obvious that measurements could be taken only at multiples of T_s . The measurement noise in (3.3) is additive and is denoted $\nu_j(kT_s)$. The only necessary assumptions on the noise is to be white and with a finite variance. Instead of kT_s , we will use simply k for denoting time in the remainder of the section. Since our filter does not require availability of input values, a process noise is not important and therefore neglected while designing our filter. Since it is not of our interest in this section, observability of the position $q_{\mathcal{T}_i}^p$ of the i^{th} target being tracked is implicitly assumed.

Chapter 3. Estimation Under Intermittent Information

After obtaining a new detection at time k , the minimal distance $d_{ij}(m)$ at time m from \mathcal{T}_i to $\partial\mathcal{B}_j(m)$, where $\partial\mathcal{B}_j(m)$ is the boundary of the FOV of \mathcal{P}_j at time instant m , is given by:

$$d_{ij}(m) = \inf_{b \in \partial\mathcal{B}_j(m), q_{\mathcal{T}_i}^p \in \{\mathcal{T}_i(m) | y_i^j(k)\}} \{||q_{\mathcal{T}_i}^p - b||\}, \quad (3.4)$$

where $|| \cdot ||$ represents the Euclidean norm, and $\{\mathcal{T}_i(m) | y_i^j(k)\}$ is the set of points in \mathbb{R}^2 reachable by the i^{th} target at instant m given the detection $y_i^j(k)$ and $m \geq k$. A more formal treatment of reachable sets is found in [52] and [53].

At any time instant k , the expectation of an estimate of the position of \mathcal{T}_i is given as the expectation of a conditional expectation (so called *smoothing property*):

$$E[q_{\mathcal{T}_i}(k)] = E[E[q_{\mathcal{T}_i}(k) | \{y_{j'_1}^i(k'_1), \dots, y_{j'_{\Gamma_i}}^i(k'_{\Gamma_i})\}]], \quad (3.5)$$

where $\{j'_1, \dots, j'_{\Gamma_i}\} \subseteq \mathcal{P}$ and $k'_i \leq k$. It should be pointed out that, based on the accuracy of the estimate, the set $\{k'_1, \dots, k'_{\Gamma_i}\}$ does not have to include the newest Γ_i detections. In our approach, expression (3.5) is calculated using the geometric approach. Note that our design allows for any filter in this calculation. In Subsections 3.3.1 and 3.3.3 we provide advantages and disadvantages of the conventional filters and the filter we developed. Based on the given application, one can choose the most appropriate filter. The effect of measurement noise ν_j is averaged out using $\Gamma_i > 3$ since a circle is uniquely defined with 3 points. Furthermore, the current velocity (or acceleration) of \mathcal{T}_i is estimated based on Γ_i detections of the target. Velocities are calculated for intervals between two consecutive observation and a Least Mean Square (LMS) line is calculated. Hence, the slope of the line represents the acceleration.

In addition, it is implicitly assumed that sensors exchange their measurements with a fusion center meaning that the estimation is centralized. The fusion center sends to the sensors detections of a target they track made by other sensors. This enables decentralized control of sensors. It should be mentioned that, whenever a sensor is used (no matter if

Chapter 3. Estimation Under Intermittent Information

the sensor is in the *idle* or *tracking* mode), newly obtained information regarding targets within its FOV are communicated to the fusion center. Although in simulations we assume no out-of-sequence measurements, if measurements have time stamps, out-of-sequence measurements do not represent a problem for our filter. However, out-of-sequence or late measurements might cause evaders to slip out of sensors' FOV.

A trade-off between energy consumption for motion on one side and the energy consumption for sensing and processing the acquired data on the other side is also taken into account when designing the Adaptive Filter. The user can put its own preferences and account for noise in the developed algorithm by adjusting the size of the following areas of FOVs. 'No Moving Area' is a central area (around the sensor's platform) of a FOV such that, if a detection is obtained within it, the pursuer will not move towards the observed target. With the size of 'No Moving Area', comprised in η , we control how much the pursuer moves while tracking. If sensing and processing of the gathered information require high energy consumption relative to energy consumption when moving, the size of 'No Moving Area' is decreased (decreased η) to accommodate for that. When a detection of \mathcal{T}_i is obtained by \mathcal{P}_j at time instant k outside of its 'No Moving Area', the following control policy is applied until the next detection:

$$u_j(t) = K(y_j^i(k) - q_{\mathcal{P}_i}(t)), \quad (3.6)$$

where K is a proportional gain and $K > 0$. Notice that such a control policy yields the exponentially decreasing distance between \mathcal{P}_j and $y_j^i(k)$ until the next detection, i.e., we have an exponentially stable sensor-target system with $y_j^i(k)$ as the equilibrium point. However, the exponential stability is replaced with asymptotic stability in realistic applications due to limited maximal velocities of pursuers (i.e., saturations of actuators).

'Escape Area' is an area closest to the boundary of a FOV. The area of 'Escape Area' is larger (larger ζ) if the measurement noise level is higher (i.e., greater variance of the noise) in order to prevent a target to escape from the FOV. Therefore, based on the previous detection, a new detection is obtained when there is a chance that the tracked target enters

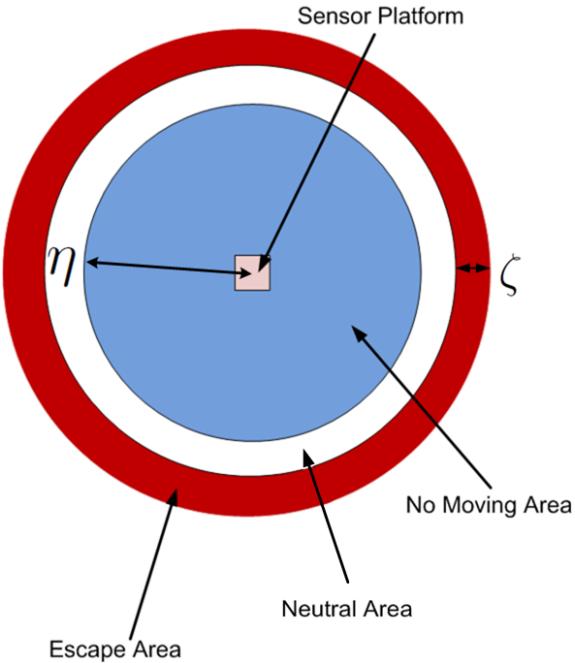


Figure 3.2: Parts of the Field-Of-View of an omnidirectional sensor.

'Escape Area' in order to slip out of the FOV (using expression analog to (3.4)). 'Escape Area' also accounts for the increased uncertainty of sensors towards the boundary of their FOVs. These two areas can differ from a pursuer to a pursuer, and depend on the target being tracked. Definitions of the 'No Moving Area' and 'Escape Area' for the j^{th} sensor are:

$$\mathcal{B}_j^{NM}(t) = \{q \in \mathcal{B}_j(t) : \|q - q_{\mathcal{P}_j}(t)\| \leq \eta_j\}, \quad (3.7)$$

$$\mathcal{B}_j^{EA}(t) = \{q \in \mathcal{B}_j(t) : \inf_{b \in \partial \mathcal{B}_j(t)} \|q - b\| \leq \zeta_j\}, \quad (3.8)$$

where η_j and ζ_j are real numbers determined by the user. Finally, the remaining part of the FOV is named 'Neutral Area' and does not have any functionality. Depending on η_j and ζ_j , 'Neutral Area' might not even exist. An illustration of the FOV areas for an omnidirectional sensor is given in Figure 3.2.

Now we have everything to define the Adaptive Filter's *decision* (or sampling policy)

whether to make a new observation in the next available time instant. The *decision* $\delta_j^i(k+1)$ is a non-analytic $\{0, 1\}$ -valued function. Hence, we are not able to show the *decision* in an analytic form but it is given by the algorithm in Figure 3.3. Notice that the algorithm includes the moving and information exchange policy of the sensors for completeness. The exposition of this section is summarized as a block diagram in Figure 3.4.

3.2.1 No-Escape Property of the Approach

When it comes to tracking properties of the adaptive scheme presented in Section 3.2, a target will almost surely stay in one of the sensors' FOV once the target enters an idle sensor's FOV provided that the sensors are faster than the target. The same is true for scenarios with multiple targets as long as the number of sensors is equal or greater than the number of targets. This property is valid because we create a worse-case detection obtaining policy. Almost surely is because the probability that the measurement noise is apart from the noise mean more than the width of 'Escape Area' is zero (of course, if the width is properly designed based on the finite variance of the noise). A slight technicality, that is always fulfilled in reality, has to be mentioned here. Sensors's FOVs must have the area big enough to allow for appropriate size of 'Escape Area'. The no-escape property is illustrated in Figures 3.7 and 3.9.

If we define the displacement from tracking sensors' 'No Moving Areas' (i.e., the set distance) to corresponding targets as states of the system and targets' positions as inputs to the system, we have a stable pursuers-targets system where new measurements induce jumps of the state. Let us define the set distance of a vector (or a point) x to a set \mathcal{C} as $|x|_{\mathcal{C}} = \inf_{y \in \mathcal{C}} \{|x - y|\}$. Recall that sensors are exponentially approaching detections obtained outside 'No Moving Area'. The region of attraction is the sensors' FOV. We model the state of such system in a hybrid manner and analyze its stability (for more see [55]).

Chapter 3. Estimation Under Intermittent Information

Theorem 1 Assume that pursuers are faster than targets, $N > M$ and the Adaptive Filter is implemented. Once a target enters an idle sensor's FOV, it will almost surely stay under sensors' surveillance (i.e., the sensors-targets system is stable almost surely).

Proof 2 We prove this theorem for a one-sensor-one-target case. A generalization to a multi-agent scenario is straightforward.

Consider a pursuer \mathcal{P} and a target \mathcal{T} , and let the target enter the pursuer's FOV at time k_0 , i.e., $y(k_0) \in \mathcal{B}_{\mathcal{P}}(k_0)$. Without loss of generality, take $\eta = 0$. Let us define the following system representing dynamics of the displacement between the sensor and the target using (3.6):

$$\begin{aligned}\dot{e}(t) &= -Ke(t), & t \in [k_i, k_i + 1], i \in \mathbb{N}_0 \\ e(t^+) &= q_{\mathcal{P}}(t) - y(t), & t \in \{k_0, k_1, \dots\},\end{aligned}\tag{3.9}$$

where $\{k_0, k_1, \dots\}$ is a set of detection times, $e(t) = q_{\mathcal{P}}(t) - y(t)$ is the displacement for $t \in [k_i, k_i + 1]$, and $y(t)$ is the system input perturbed with noise. From (3.9), $e(t)$ exponentially decreases to the origin between two consecutive detections. If the filter is properly designed and the pursuer is faster than the target, we have $\Pr\{y(k_i) \notin \mathcal{B}_{\mathcal{P}}(k_i) | y(k_0) \in \mathcal{B}_{\mathcal{P}}(k_0)\} = 0, \forall k_i \geq k_0$ where \Pr denotes probability. Now, take $\delta_v = \mathcal{B}_{\mathcal{P}}(k_0) - q_{\mathcal{P}}(k_0)$, where δ_v denotes δ -vicinity of the origin at time k_0 , and $\mathcal{B}_{\mathcal{P}}(k_0) - q_{\mathcal{P}}(k_0)$ is a translation of $\mathcal{B}_{\mathcal{P}}(k_0)$ to the origin. Also, take $\epsilon_v = \mathcal{B}_{\mathcal{P}}(t) - q_{\mathcal{P}}(t)$, where $t \geq k_0$, to be ϵ -vicinity of the origin. Now we conclude that once we have $e(k_0) \in \delta_v$, it will almost surely yield $e(t) \in \epsilon_v, \forall t \geq k_0$. Note that $\mathcal{B}_{\mathcal{P}}(t)$ can be an arbitrary vicinity of the pursuer. Using the definition of stability on the page 49. of [55], the stability of the sensors-targets system is proven.

Corollary 1 Assuming the conditions of Theorem 1 are met. If all targets are initially within FOVs of different sensors, then they will almost surely stay within sensors's surveillance (i.e., the sensors-targets system is stable almost surely).

3.3 Simulation Results

3.3.1 Estimation Performance of the Filters

The proposed tracking strategy has been implemented and applied to numerous case studies, and the estimation quality of targets' positions has been compared to Unscented Kalman Filter (UKF) and Sampling Importance Resampling PF (SIR PF) or bootstrap PF. In order to make it concise, this section includes only two case studies to convey our observations. SIR is not the most progressive of all PFs, but is well studied and frequently implemented. The implemented PF is capable of dealing with non-periodic observations. A basic study of PF dealing with intermittent observations as a consequence of having more targets than sensors could be found in [27]. Between two consecutive observations there is no information of the inputs. This lack of information is circumvented using the "zero-order-hold" strategy, i.e., the previous input is used until the new one comes (for a comprehensive discussion regarding this issue see [17]).

We start the analysis of the filters' performance with two one-sensor-one-target case studies. The first case study is simple and the target makes a circular path having almost constant inputs. The second case study includes a zig-zag path and is very challenging for estimation. The reasons are the complicated geometry of the path and abrupt changes of inputs including negative linear and angular velocities and accelerations while turning¹. Our filter is used in the version with periodic sampling and in the version with adaptive sampling. In all simulations, the minimal possible sampling time T_s is 0.07 s. This period is the minimal period needed to run simulations in real time. Several outcomes of these simulations are found in Figures 3.5 and 3.6. The sensor's radius is 2.3 m and the AOI is 15×15 m² in both case studies. In addition, the maximal linear velocity of the sensor is 2 m/s, while the target's linear velocity attains values between -0.7 and 0.7 m/s and angular velocity's values are between -1.5 and 1.5 rad/s.

¹Videos of simulations and experiments are at <http://marhes.ece.unm.edu/documents/Domagoj/>

Chapter 3. Estimation Under Intermittent Information

For KF and BF, the first detection is provided through the initialization of those filters, i.e., the initial state and prior distribution. Therefore, the first detection is not shown in the corresponding figures, but is counted as a detection for cases with KF and BF. Our filter initializes itself by making detections in the first three consecutive steps. Since the initialization is performed only once, this does not present a big increase in number of detections. Hence, the first three detections are counted as one detection when our filter is used.

The mean of Root Mean Square Errors (RMSEs) along with the corresponding variance of 100 Monte Carlo simulations using different filters are presented in Tables 3.1, 3.2, 3.3 and 3.4. Since the implemented noise is Gaussian, the performances of UKF and PF are quite similar. The performance of our filter is highly competitive with respect to PF. It should not be forgotten that our filter does not require availability of inputs and exact modeling, performs faster and skips the estimation process when possible (for more regarding the latter two properties see Subsection 3.3.3). Therefore, in certain applications the proposed tracking approach is more suitable than UKF and PF, especially in settings where modeling is difficult (e.g., due to lack of knowledge) and as the sampling period increases.

One of the reasons why our filter performs with such a good quality, despite being deprived of many information that UKF and SIR PF are provided with, is the fact that it uses Γ samples creating a Γ^{th} -order Markov process model of targets' maneuvers. This makes our method more robust to overcome measurement noise and missing detections. The effect of the noise is averaged out through Γ observations. However, our method brings a certain inertia in the estimate that is sometimes undesirable. This inertia can be diminished by decreasing Γ .

The sensor measures noisy target positions. The noise, for the sake of simplicity, is taken to be a white zero-mean Gaussian with the following diagonal covariance matrix $R = \text{diag}(r_1, r_2)$, where $r_1 = r_2 = 10^{-2}$. The use of lower noise levels (smaller r_1 and

Chapter 3. Estimation Under Intermittent Information

r_2) results in several times smaller RMSE of our filter comparing to UKF and PF for the circular path. For higher noise levels, our filter accurately estimates the geometry of the circular path, but the velocity estimate is significantly deteriorated. For the zig-zag path, a decrease in the noise level gives a smaller advantage to UKF and PF with respect to our filter taking RMSE as a criterion.

However, one thing should be pointed out when it comes to the actual implementation of UKF and PF and the modeled noise levels. As the sampling period increases, we have to introduce process noise with covariance matrix $Q = \text{diag}(q_1, q_2, q_3)$ in order for the filters to converge. This process noise needs to be increased in order to account for a higher uncertainty. For instance, in highly intermittent scenarios, we need to use even $Q = \text{diag}(10^{-1}, 10^{-1}, 10^{-1})$ for UKF and PF. By doing so, we can use a smaller number of particles for PF and their impoverishment is decreased (for more details about the particle impoverishment refer to [30]). An increase in the number of particles is computationally more demanding than an increase of the noise level. For example, more than 10000 particles have to be used to compensate for the increase in uncertainty without increasing the level of the modeled noise in order for PF to converge. More particles lead to an increased running time per second of the PF. When increasing modeled noise, we compensate for the increased uncertainty so that we can use a smaller number of particles (1000 particles) to obtain convergence. The Running Time Per Step (RTPS) is 3.5 ms in the case of 1000 particles, and in the case of 10 times more particles RTPS becomes almost 140 times greater. On the other hand, overly augmented noise levels lead to greater RMSEs. Hence, intermittent information have to be handled carefully and with a lot of tuning of model parameters in the case of UKF and PF. The case of PF with 1000 particles is elaborated thoroughly in Subsection 3.3.3. On the other hand, our filter does not have any convergence problems.

Since the simulations' duration is $T_{final} = 55$ s and $T_s = 0.07$ s, the maximal number of detections is 786. The statistics of the estimates when the number of observations is minimal (i.e., the maximal sampling period), based on the sensors' radius, are also provided

Table 3.1: A comparison of standard filters with the Periodic Filter for the circular path

Circular Path	UKF	SIR PF	Periodic
# detections		786	
RMSE	0.0960	0.0954	0.1546
Variance	$4.1 * 10^{-6}$	$3.6 * 10^{-6}$	$2.6 * 10^{-4}$
# detections		100	
RMSE	0.1060	0.1407	0.1405
Variance	$4.8 * 10^{-5}$	$6 * 10^{-5}$	$2 * 10^{-4}$
# detections		31	
RMSE	0.1480	0.2084	0.2031
Variance	$2.8 * 10^{-4}$	$2.6 * 10^{-4}$	0.0011
# detections		22	
RMSE	0.2080	0.2571	0.2697
Variance	0.0022	$4.8 * 10^{-4}$	0.0059
# detections		17	
RMSE	0.4623	0.3352	0.3785
Variance	0.0334	$7.4 * 10^{-4}$	0.0137

in Tables 3.1 and 3.3. As it can be seen, the Adaptive Filter has almost the minimal possible number of observations. The Adaptive Filter can approach the minimal possible number of observations even further, but the actual number depends on the moving-sensing trade-off determined by the user (and a realization of the noise, of course). The implemented size of the 'No Moving Area' puts a rather high cost to the movement. That is the reason for the above observation. In addition, the Periodic Filter does not give any cost to movement nor takes the noise into account. In the simulations, we use $\eta_j \in [1/4R_j, 3/4R_j]$ where R_j is a sensing radius of the j^{th} pursuer. In addition, simulations show that $\zeta_j \in [3T_s * \max\{|v_{i,\min}|, |v_{i,\max}|\}, 9T_s * \max\{|v_{i,\min}|, |v_{i,\max}|\}]$ is a valid choice when tracking the i^{th} target.

Table 3.2: Statistics of the Adaptive Filter for the circular path

Circular Path	# det. mean	# det. Var.	RMSE	Variance
Adaptive	22.14	0.9392	0.3262	0.0050

Table 3.3: A comparison of the standard filters with the Periodic Filter for the zig-zag path

Zig-Zag Path	UKF	SIR PF	Periodic
# detections	786		
RMSE	0.0966	0.0959	0.1459
Variance	$4.6 * 10^{-6}$	$3.6 * 10^{-6}$	$2.1 * 10^{-5}$
# detections	100		
RMSE	0.1170	0.1425	0.2687
Variance	$6.6 * 10^{-5}$	$4.3 * 10^{-5}$	$4.8 * 10^{-4}$
# detections	25		
RMSE	1.0728	0.8357	0.9156
Variance	0.0133	0.0114	0.0065
# detections	20		
RMSE	1.5459	1.1704	1.2350
Variance	0.0509	0.0724	0.0080

3.3.2 Multi-agent Tracking and Filtering Algorithms' Analysis

After verifying the proposed filter on one-sensor-one-target scenarios, a progressive multi-agent scenario is implemented. In this scenario, four sensors cooperatively track two targets whose behavior is purely stochastic. Each target has to visit 10 uniformly distributed points in the AOI. All targets start within one of the sensors' FOVs. Targets are not tracked

Table 3.4: Statistics of the Adaptive Filter for the zig-zag path

Zig-Zag Path	# det. mean	# det. Var.	RMSE	Variance
Adaptive	25.380	3.0976	1.0044	0.0306

by the same sensor during the simulation. Since sensors communicate and collaborate, any sensor can ‘look’ after any target depending on the game progression. However, one sensor is in charge of only one target at any time. Hence, the idle sensors are available to take care of upcoming targets. Simulations show that such approach does not allow targets to escape from the sensors’ FOVs. Centralized access to all observations is assumed in the implementation. Illustrations of this scenario are given in Figures 3.7. Notice that both sensors and targets form heterogeneous groups (refer to Subsection 2.2.1). For brevity, the properties of the sensors and targets are not included herein.

3.3.3 Complexity Analysis

When comparing algorithms, several criteria can be used. We use the Running Time Per Step (RTPS) and “big O” notation. The former informs about the actual implementation and is prone to one’s programming skills. Nevertheless, RTPS is useful when comparing execution times of algorithms applied on actual problems. The second criterion contains information regarding the potential of an algorithm, i.e., scalability of an algorithm. Results for both criteria are shown in Table 3.5. The simulations are performed on a computer with a 2.4GHz processor and 2GB of memory. Our implementation of UKF and PF is in accordance with the “big O” complexity of these filters in the literature (the literature for UKF is extensive, and for PF see [95]). According to [93], the number of operations for computing a fitting curve through Γ points is $(a*\Gamma+b*ord)*ord^2$ where ord is the order of the curve and a and b are constants. The order of a curve is the number of generators constituting a basis for the ideal as a vector space. When the curve is a 2D circle, the previous expression becomes the one in Table 3.5 with $ord = 4$. Apparently, scalability of the fitting circle algorithm can be determined with respect to Γ or ord . Scalability of the algorithm is linear in Γ and cubic in ord . It should be noted that Γ in our simulations is a number up to 20 and that the difference in RTPS when $\Gamma = 3$ and $\Gamma = 20$ is 0.1 ms. In addition, ord does not grow rapidly as the space, in which fitting takes place, grows (see [93]). For instance,

Table 3.5: A complexity comparison of the standard filters with the filter we developed. RTPSs are obtained with $\#particles = 1000$ for PF, and $\Gamma = 17$ for our filter.

Filter	RTPS [s]		“Big O”
	Mean	Variance	
UKF	$4.9 * 10^{-4}$	$1.7 * 10^{-8}$	$O((\#states)^3)$
SIR PF	0.0098	$5.6 * 10^{-7}$	$O(\#particles)$
Our Filter	0.0035	$4.3 * 10^{-7}$	$O((\Gamma + ord) * ord^2)$

“the curse of dimensionality” is a well-known problem for PFs ([96]) and a limiting factor in real-time implementations. Therefore, generalizations of our algorithm to 3D scenarios are less computationally demanding than generalizations utilizing a PF.

Moreover, our filter has the ability to decrease processing requirements when a new detection is anticipated with a certain ρ from previous detections. By choosing ρ , the user can accommodate the filter to the complexity of targets’ maneuvers and decrease computational requirements while controlling the estimation error. About 10 to 25 % of detections for the Adaptive Filter does not lead to a computation of a new fitting circle in the case of the zig-zag path (Figures 3.8(c) and 3.8(d)). For the case of the circular path and periodic version of the filter, it is between 10 and 35 % (Figures 3.8(a) and 3.8(b)). These percentages represent significant savings of the processing power needed for other tasks such as communication, signal processing, path planning, etc.

When it comes to the complexity of multi-agent estimation, the complexity is $N * O(filter)$. Each sensor has to obtain readings and targets’ positions are determined using one of the above filters for each target.

From Table 3.5 and the previous discussion, it can be concluded that our filter demands a smaller portion of the processing power than the standard filters and shows better scalability. Furthermore, our filter requires memory for only Γ (about 10) detections while PF has to keep in memory all particles (1000 in our simulations) and an importance density.

3.4 Improving the Marco Polo Game

When a sensor is switched to the pursuit mode in the Marco Polo game from Chapter 2, the Adaptive Filter is used. It should be mentioned that the capturing strategy from Chapter 2 does not have adaptive sampling and works only on average based on the probabilistic model of targets. The new approach presented herein provides estimates of each realization of Markov processes (i.e., targets). Hence, the work herein complements Chapter 2 by adding estimation and adaptive sampling. Consequently, the capturing strategy is improved meaning that once the capturing conditions are met and a sensor is deployed to capture a target, the target is almost always caught due to the no-escape property given in Subsection 3.2.1. The only situations when a target is not caught are when the target is close enough to the boundary of AOI at the moment the sensor is deployed for capturing, and slips out of the AOI. In addition, instead of keeping an evader within its FOV as in the previous sections, a sensor in the pursuit mode uses the prediction of target positions in order to capture the target. In order to accommodate the Adaptive Filter for the presence of obstacles, we modify its adaptive sampling policy. Besides the conditions stated in Section 3.2, a new detection is also made when the prediction of the evader position collides with an obstacle. The rationale is that, when facing an obstacle, the evader has to make abrupt changes in its maneuvers.

Snapshots of the improved Marco Polo game are provided in Figure 3.9.

3.5 Experimental Verification of Tracking Methodology

We create three different experimental setups². In the first setup, one P3-AT (or Pioneer) robot is the pursuer and the other P3-AT is the evader. This experimental setup is an example of a nonholonomic evader with the behavior quite similar to the targets used in

²Videos of the experiments are available at <http://marhes.ece.unm.edu/documents/Domagoj/>

Chapter 3. Estimation Under Intermittent Information

simulations in Section 3.3.1, i.e., it simply wanders in AOI. Due to the similarity of the maneuvers of P3-AT robots to the targets in Section 3.3.1, the performance of the Adaptive Filter is fairly similar to the performance obtained by simulation. Hence, the performance statistics of the experiments are not provided as they do not bring any additional insight into the Adaptive Filter performance. This can be inferred from the accompanying video. The P3-AT robots in all experiments are controlled through the Player/Stage/Gazebo interface.

In the second experimental setup, two P3-AT robots are pursuers and a Rovio robot is the evader. The pursuers have different linear and angular velocities, and different areas of FOVs. Rovio possesses both holonomic and nonholonomic behaviors. Since Rovio is a toy and not an autonomous agent for scientific research, it is controlled by a human operator. It is hard to control Rovio in a smooth manner and there is no speed control; hence, Rovio is an example of an evader with quite an erratic behavior. Because of the human operator and the lack of smooth controllability remotely, this setup presents a real challenge for our approach. Due to the erratic evader, the estimation quality is decreased as expected. For more details, refer to the accompanying video.

In the third experimental setup, we use an Ascending Technologies Hummingbird quadrotor and Rovio. The design and physical properties of a standard Hummingbird quadrotor are described in detail in [38]. The quadrotor is the pursuer, and Rovio is the evader. We implemented a PD low level attitude controller for waypoint tracking using Matlab Simulink and a VICON motion capturing system. Control commands from the attitude controller are sent to the quadrotor using wireless X-bee RF modules. Waypoints are sent via serial communication from the Adaptive Filter to the attitude controller. During the experiment, height and heading of the quadrotor are kept constant by the attitude controller. Upon receiving a new waypoint, a ramp is generated in order to keep the smoothness of the quadrotor trajectories (Figure 3.11).

In Figure 3.10 and 3.12 we provide snapshots of the experiments. In the upper/lower right corner of the snapshots, the Matlab plots reconstructed from the experimental data are

included. All experiments verify the no-escape property of the approach. To accommodate for nonholonomic sensors platforms, and the latency in the link between the computer running the Adaptive Filter and the Player client on the P3-AT robots, we use $\rho = 15T_s * \max\{|v_{i,\min}|, |v_{i,\max}|\}$ in order to assure the no-escape property. It is worth of mentioning that no other modifications to accommodate for experimental settings have been made. Finally, the VICON motion capturing system is used to emulate on-board sensors.

3.6 Conclusion

In this chapter, we propose a method for adaptive sampling of targets' positions in order to rationally use expensive resources in tracking problems. We also develop a *predictor-corrector* tracking filter, investigate its properties, and compare its performance under intermittent information and its complexity with the state of the art filters (UKF and PF). Our filter uses geometrical properties of targets' maneuvers for estimating targets' positions. Our filter yields better estimation than UKF as the sampling period increases, and its performance is comparable to the estimation quality of PFs. Moreover, our filter performs faster and demands less memory resources than PF; hence, it may be more suitable for certain on-line estimation problems. In addition, the developed filter requires substantially less information than the existing filters (no inputs of evaders, no exact model of evaders and noise). The proposed approach results in no-escaping targets that are being tracked. Furthermore, our latest results regarding the Marco Polo game are presented, and the tracking filter is utilized while capturing targets. At this stage, the Marco Polo game successfully brings together probabilistic modeling of targets, sensor placement, area surveillance, path planning in the presence of obstacles and other agents, false alarms, estimation of the targets' positions and efficient capturing policy. Finally, our filter is validated experimentally using heterogeneous agents.

In the future, we plan to improve the fitting circle algorithm. Improvements of the algo-

Chapter 3. Estimation Under Intermittent Information

rithm in order to estimate targets' velocities more accurately will lead to better estimation. An implementation of the tracking filter in 3D scenarios seems a natural way to extend our work. An experimental setup of the Marco Polo game that will show the full potential of our testbed and methodologies brought together to solve the problem of intermittent information in pursuit-evasion games is ongoing work. Future research regarding the Marco Polo game is oriented towards further decentralization of the developed methods.

```

1: Set parameters  $\Gamma_i$ ,  $\rho_j$ ,  $\eta_j$  and  $\zeta_j$  for all  $i \in \mathcal{I}_{\mathcal{T}}$  and  $j \in \mathcal{I}_{\mathcal{P}}$  based on targets', pursuers' and
   noise properties, preferences and  $T_s$ 
2: Set time:  $t = 0$ , i.e.,  $k = 0$ 
3: Obtain sensors' readings and make tracking pairs target-pursuer  $(i, j)$ 
4: For all pairs  $(i, j)$  set  $\delta_j^i(index) = 1$ ,  $index \in \{1, 2\}$ 
5: while  $t \leq T_{final}$  do
6:   for all Pursuers do
7:     if  $\mathcal{P}_j$  is idle then
8:       Scan  $\mathcal{B}_j^{EA}(k)$  for new targets
9:       Exchange obtained information with fusion center
10:      if New targets are detected then
11:        Randomly choose one target (say  $\mathcal{T}_i$ ) and start tracking
12:        Set  $\delta_j^i(index) = 1$ ,  $index \in \{k + 1, k + 2\}$ 
13:      end if
14:    else
15:      if  $\delta_j^i(k) = 1$  then
16:        Use corresponding sensor to scan FOV
17:        Exchange information (including  $y_j^i(k)$ ) with fusion center
18:        if  $||E[q_{\mathcal{T}_i}^p(k)] - y_j^i(k)|| \geq \rho_j$  then
19:          Recalculate a fitting circle using last  $\Gamma_i$  detections
20:          if  $||E[q_{\mathcal{T}_i}^p(k)] - y_j^i(k)|| \geq \rho_j$  then
21:            Set  $\delta_j^i(k + 1) = 1$ 
22:          end if
23:        end if
24:        if  $y_j^i(k) \notin \mathcal{B}_j^{NM}(k)$  then
25:          Apply  $u_j(t)$  given by (3.6)
26:        end if
27:        if  $d_{ij}(k) \leq \zeta_j$  then
28:          Set  $\delta_j^i(k + 1) = 1$ 
29:        end if
30:        Make prediction  $E[q_{\mathcal{T}_i}(k + 1)]$ 
31:      end if
32:    end if
33:  end for
34:   $t = t + T_s$ , i.e.,  $k = k + 1$ 
35: end while

```

Figure 3.3: The adaptive sampling rate algorithm developed in Chapter 3 for multi-agent tracking scenarios.

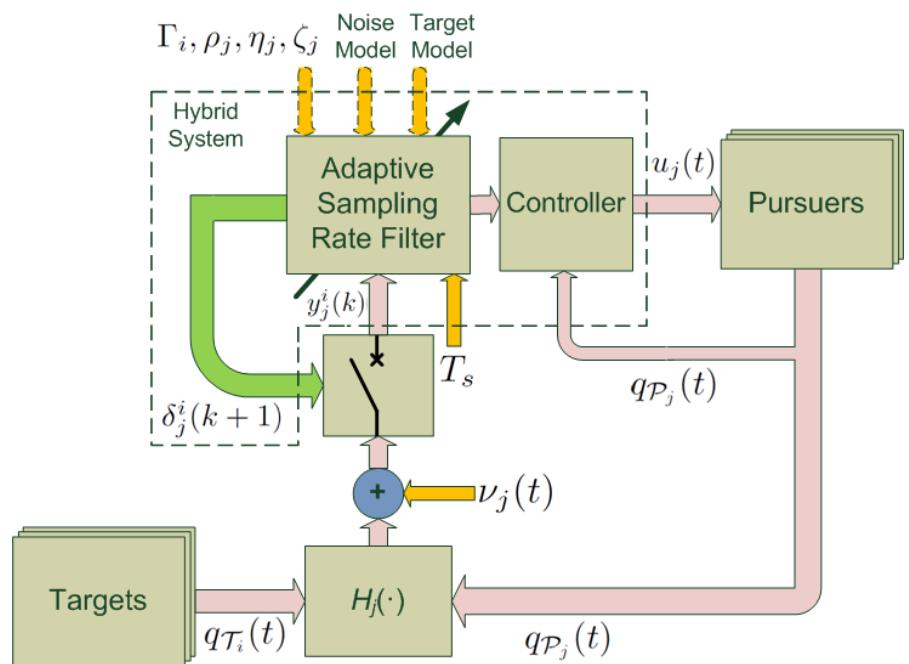


Figure 3.4: A block diagram of the methodology developed for multi-agent tracking scenarios.

Chapter 3. Estimation Under Intermittent Information

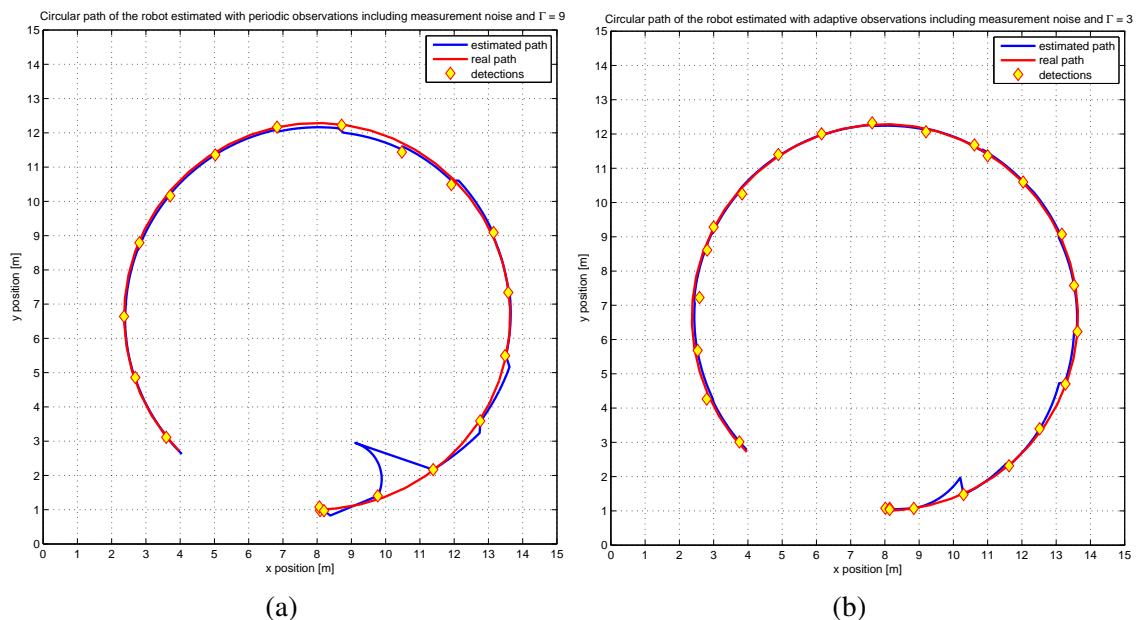


Figure 3.5: Estimates and actual paths using different filters: (a) An estimate of the circular path using the Periodic Filter (#det. = 17); and, (b) An estimate of the circular path using the Adaptive Filter (#det. = 23).

Chapter 3. Estimation Under Intermittent Information

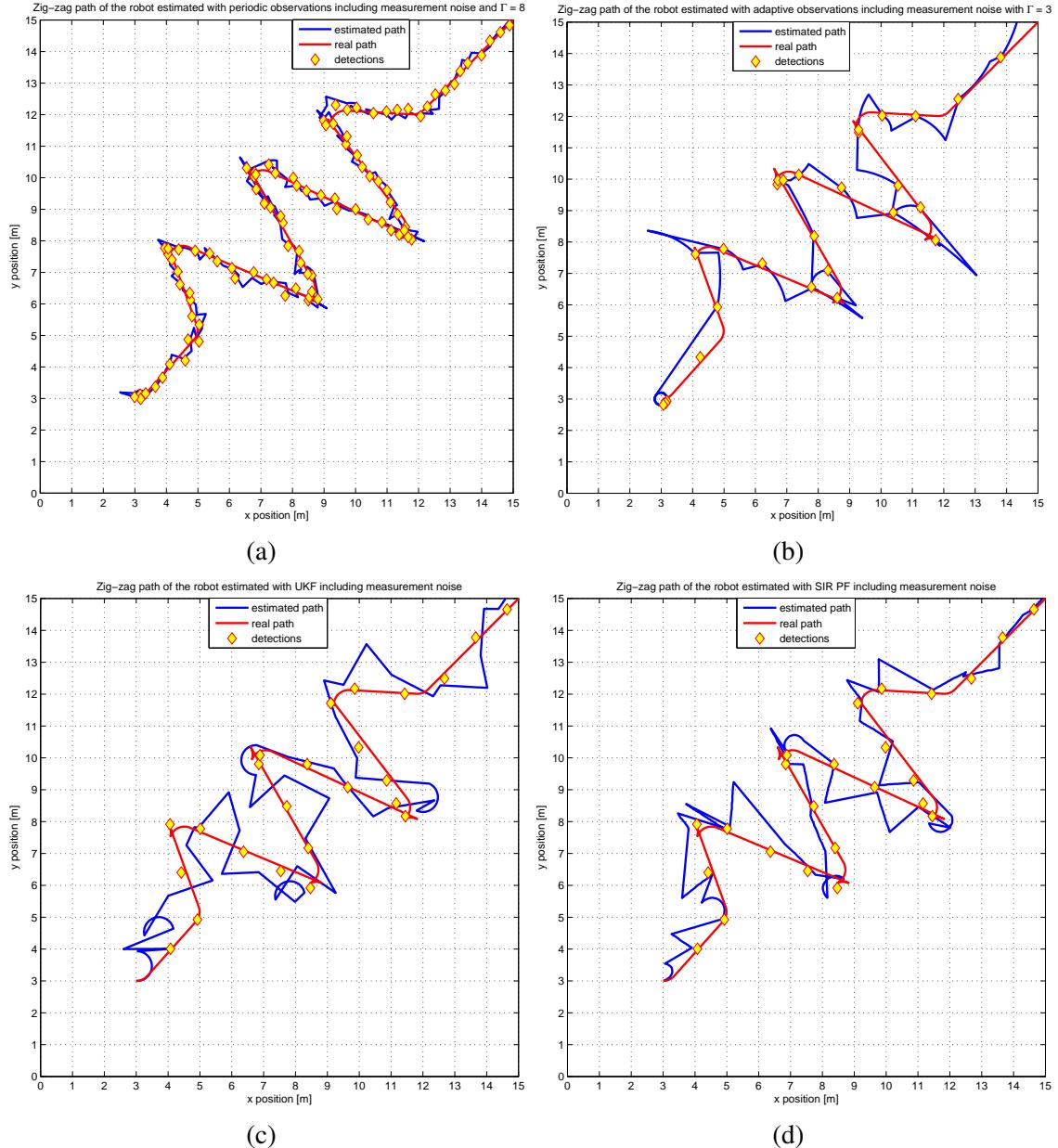


Figure 3.6: Estimates and actual paths using different filters: (a) An estimate of the zig-zag path using the Periodic Filter (#det. = 100); (b) An estimate of the zig-zag path using the Adaptive Filter (#det. = 25); (c) An estimate of the zig-zag path using a UKF filter (#det. = 25); and, (d) An estimate of the zig-zag path using a PF filter (#det. = 25).

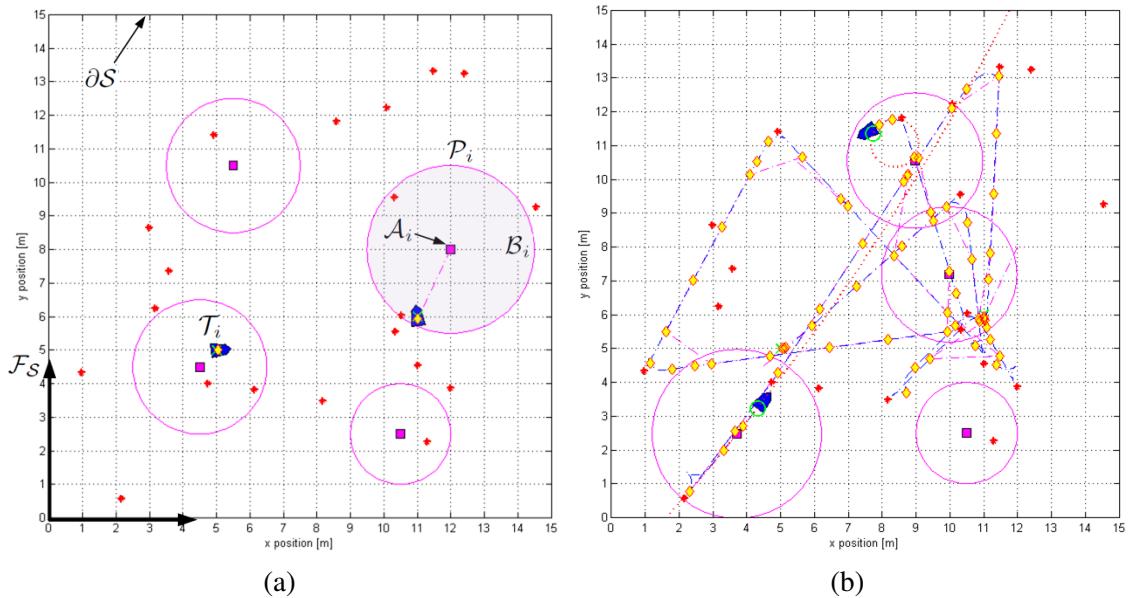


Figure 3.7: Snapshots of the multi-agent scenario: (a) The initial state of a cooperative scenario including 4 sensors and 2 targets. The simulation snapshot is edited in order to visualize the notation and terminology introduced in Section 3.1; and, (b) The final state of a cooperative scenario including 4 sensors and 2 targets. Purple squares represent sensor platforms, purple circles represent the sensors' FOV boundaries, blue polygons represent targets, while yellow diamonds represent detections. Red dotted circles (i.e., the circle and the line) represent fitting circles, red stars are the targets' waypoints, blue hashed curves are the targets' paths, purple hashed curves are the sensors' paths, green crosses are the initial positions of the targets and green circles are estimates of targets' positions.

Chapter 3. Estimation Under Intermittent Information

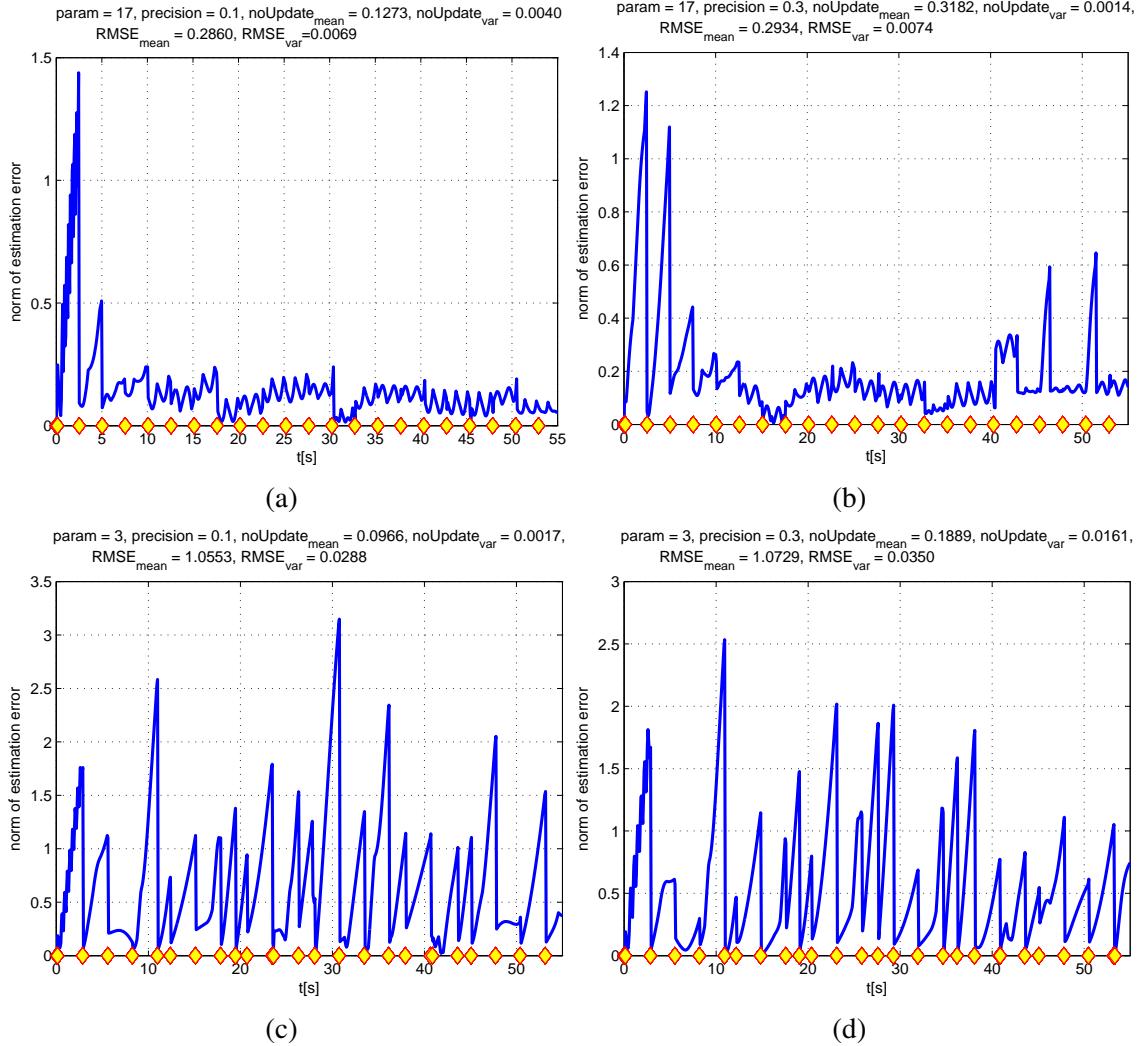


Figure 3.8: Estimation error in time: (a) A sample of the estimation error for the circular path using the Periodic Filter with $\rho = 0.1$ and $\Gamma = 17$. The title of the plot comprises statistics of 50 Monte Carlo simulations; (b) A sample of the estimation error for the circular path using the Periodic Filter with $\rho = 0.3$ and $\Gamma = 17$. The title of the plot comprises statistics of 50 Monte Carlo simulations; (c) A sample of the estimation error for the zig-zag path using the Adaptive Filter with $\rho = 0.1$ and $\Gamma = 3$. The title of the plot comprises statistics of 50 Monte Carlo simulations; and, (d) A sample of the estimation error for the zig-zag path using the Adaptive Filter with $\rho = 0.3$ and $\Gamma = 3$. The title of the plot comprises statistics of 50 Monte Carlo simulations.

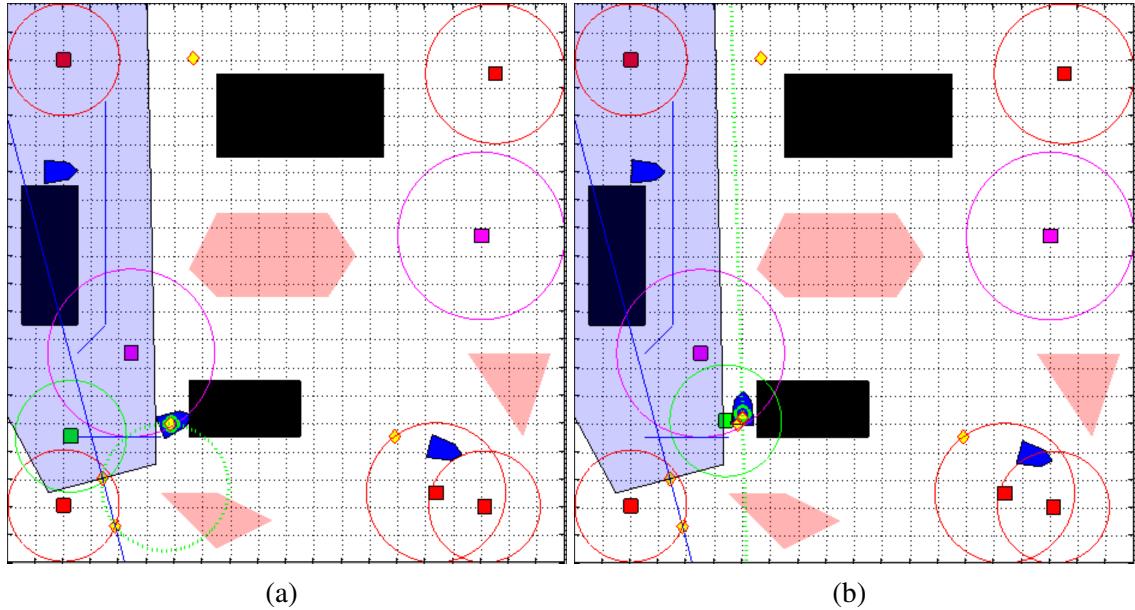


Figure 3.9: Snapshots of the implemented capturing strategy in the comprehensive Marco Polo scenario: (a) The UGV painted green is deployed to capture the target located in the lower-left part of the AOI; and, (b) A snapshot just before the target is captured.

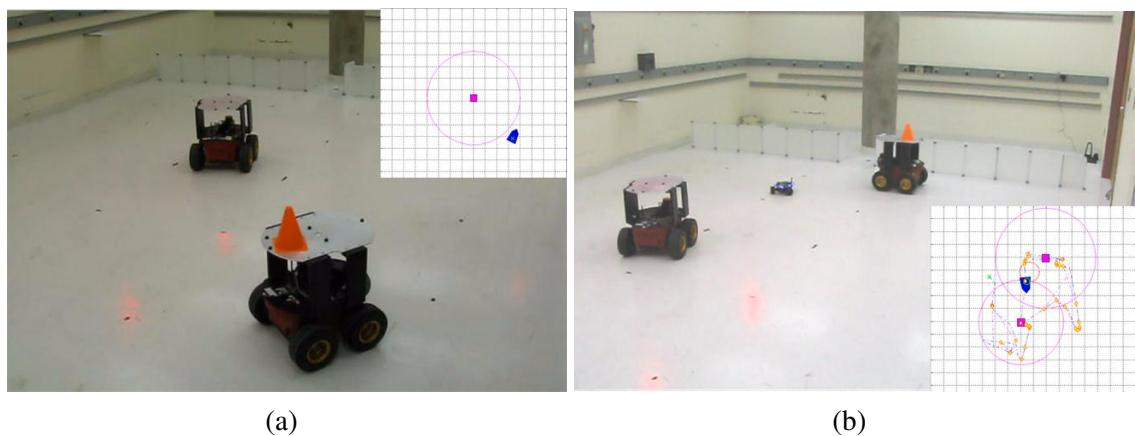


Figure 3.10: Snapshots of the experiments along with Matlab plots constructed from the experimental data. Notice the correspondence with the real picture. (a) A snapshot of the first experimental setup; and, (b) A snapshot of the second experimental setup.

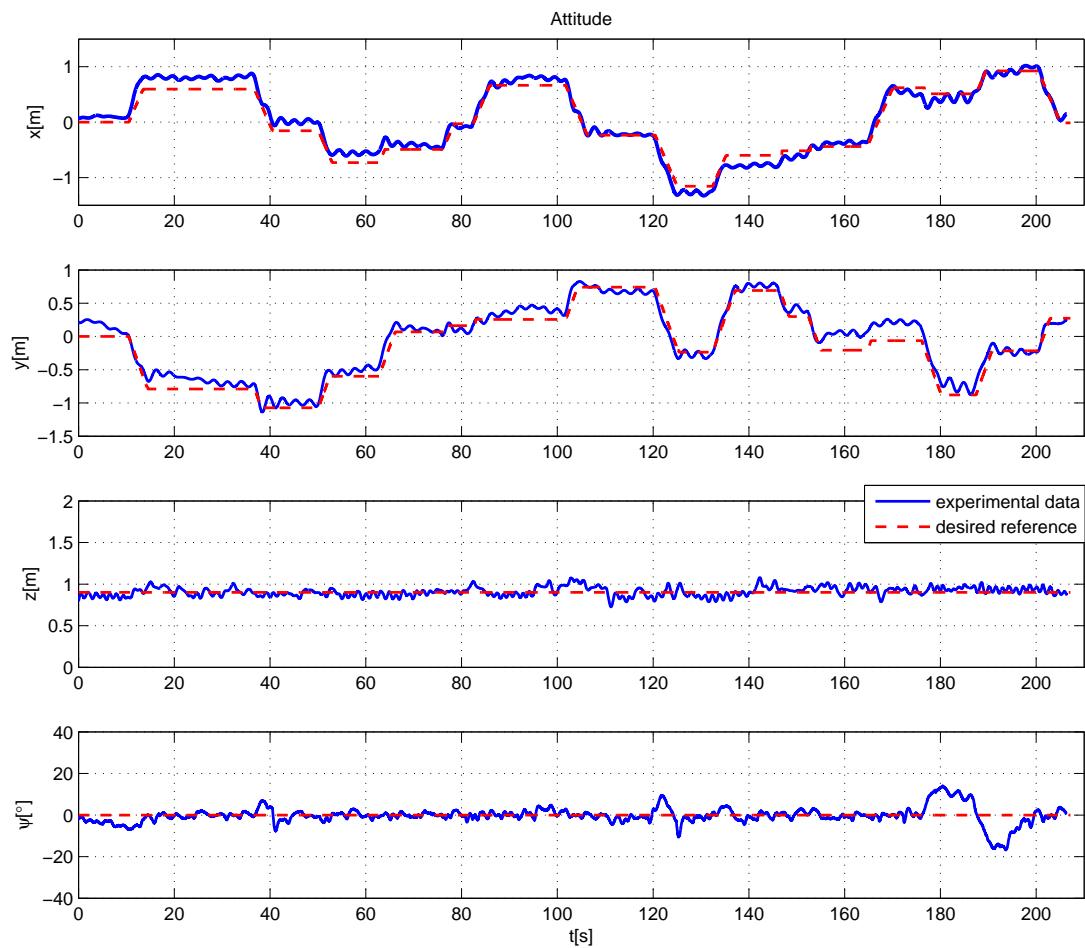


Figure 3.11: Experimental data of the quadrotor attitude while tracking waypoints provided by the Adaptive Filter.

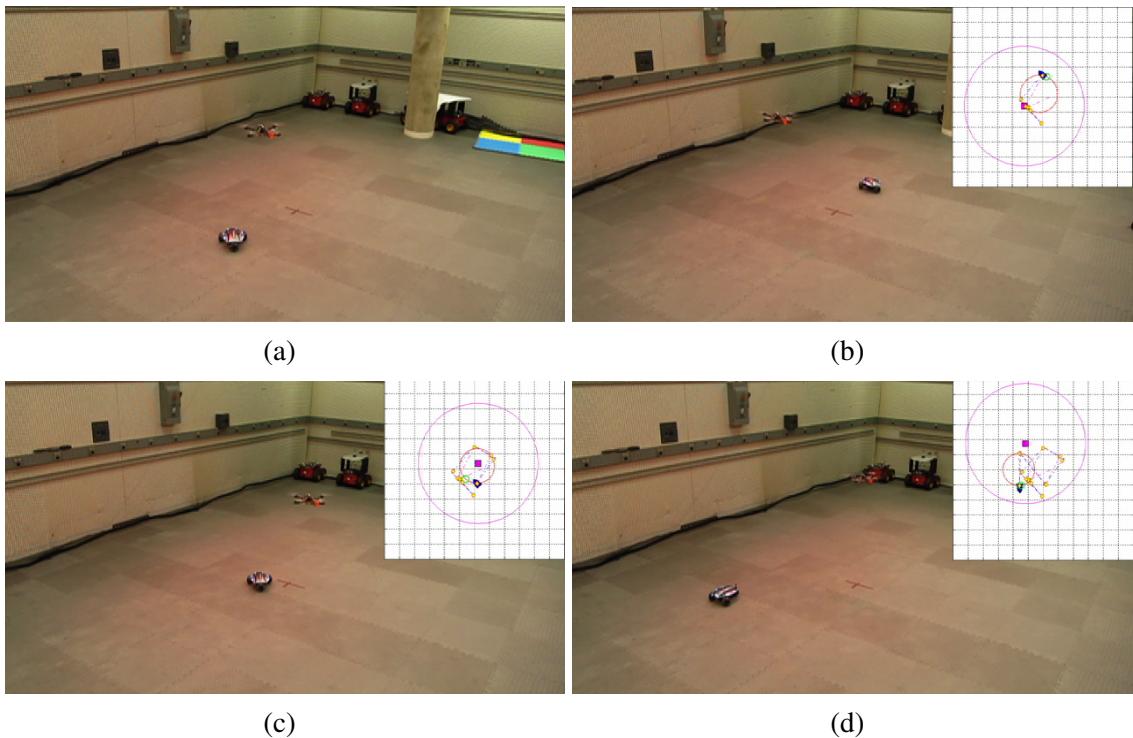


Figure 3.12: Snapshots of the experimental setup involving a quadrotor and Rovio. (a) The beginning of the experiment; (b) Rovio is driving backwards; (c) Rovio is turning; and, (d) The end of the experiment.

Chapter 4

Stability Under Intermittent Information

The main limitation of the approach in [39] and [40] is the periodicity of transmission instants inherited from the standard definition of \mathcal{L}_p -gain. Recall that the standard \mathcal{L}_p -gain is not a function of time (i.e., there is no prediction horizon) nor state. To circumvent this limitation, we devise an input-output triggered approach employing \mathcal{L}_p -gains over a finite horizon in the small gain theorem. Under the term *input-output triggering* we refer to self-triggering based on the value of the system's external input and output in the last feedback transmission. The triggering event in our approach is violation of the small gain condition. Notice that \mathcal{L}_p -gains over a finite horizon enable prediction of the triggering event. We point out that state-triggered implementations of the dissipativity-based and ISS approaches are facilitated by the requirement that storage and Lyapunov functions are known, respectively. Recall that storage and Lyapunov functions are state-dependent by definition.

The chapter is organized as follows. In Section 4.1, we provide an example that motivates the results and approach proposed in this chapter. Section 4.2 formulates the problem

of intermittent feedback under various assumptions. Section 4.3 presents the notation and definitions utilized in this chapter. The methodology brought together to solve the problem is presented in Section 4.4. The small gain theorem is employed in Section 4.5 to obtain an input-output triggered sampling policy. The proposed input-output triggered sampling policy is verified on a trajectory tracking problem in Section 4.6. Conclusions are in Section 4.7. Proofs and several technical results are included in the Appendix.

4.1 Motivational Example

Using laser-based or radar-based sensors, Autonomous Cruise Control (ACC) technology allows a vehicle to slow down when approaching another vehicle and accelerate to the desired speed when traffic allows. Besides reducing driver fatigue, improving comfort and fuel economy, ACC is also intended to keep cars from crashing [97]. The sampling periods of ACC loops are typically fixed and designed for the worst case scenario (e.g., fast and heavy traffic). Besides, these fixed sampling periods are often determined experimentally and are based on the traditional rules of thumb (e.g., 20 times the time constant of the dominant pole). Intuitively, the sampling periods of ACC loops should not remain constant as the desired speed, distance between the cars, the environment (urban on non-urban), and paths (straight or turns) change. The work presented herein quantifies this intuition.

Consider the *trajectory tracking* controller in [98] as an example of a simple ACC. In [98], a velocity-controlled unicycle robot R_1 given by

$$\dot{x}_{R1} = v_{R1} \cos \theta_{R1}, \quad \dot{y}_{R1} = v_{R1} \sin \theta_{R1}, \quad \dot{\theta}_{R1} = \omega_{R1} \quad (4.1)$$

tracks a trajectory generated by a virtual velocity-controlled unicycle robot R_2 with states x_{R2} , y_{R2} and θ_{R2} , and linear and angular velocities v_{R2} and ω_{R2} , respectively. See Figure

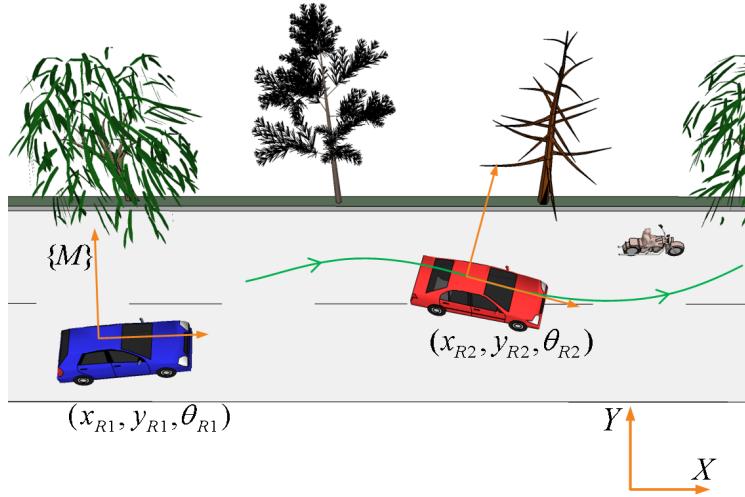


Figure 4.1: An illustration of the trajectory tracking problem considered in Chapter 4.

4.1 for an illustration. The tracking error x_p in the coordinate frame $\{M\}$ of robot R_1 is

$$x_p = \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \end{bmatrix} = \begin{bmatrix} \cos \theta_{R1} & \sin \theta_{R1} & 0 \\ -\sin \theta_{R1} & \cos \theta_{R1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{R2} - x_{R1} \\ y_{R2} - y_{R1} \\ \theta_{R2} - \theta_{R1} \end{bmatrix}. \quad (4.2)$$

After differentiating (4.2) we obtain:

$$\dot{x}_p = \begin{bmatrix} \omega_{R1} x_{p2} - v_{R1} + v_{R2} \cos x_{p3} \\ -\omega_{R1} x_{p1} + v_{R2} \sin x_{p3} \\ \omega_{R2} - \omega_{R1} \end{bmatrix}. \quad (4.3)$$

System (4.3) can be interpreted as a plant with state x_p and external inputs v_{R2} and ω_{R2} . Take the output of the plant to be $y = x_p$ and introduce $\omega_p := [v_{R2} \ \omega_{R2}]^\top$. The plant is controlled through control signals v_{R1} and ω_{R1} . In order to compute v_{R1} and ω_{R1} , and track an unknown trajectory (a trajectory is given by $v_{R2}(t)$, $\omega_{R2}(t)$ and initial conditions of x_{R2} , y_{R2} and θ_{R2}), robot R_1 needs to know the state of the plant x_p and the inputs to R_2 ,

Chapter 4. Stability Under Intermittent Information

i.e., v_{R2} and ω_{R2} . Following [98], choose the following control law:

$$\begin{aligned} v_{R1} &= v_{R2} \cos x_{p3} + k_1 x_{p1}, \\ \omega_{R1} &= \omega_{R2} + k_2 v_{R2} \frac{\sin x_{p3}}{x_{p3}} x_{p2} + k_3 x_{p3}, \end{aligned} \quad (4.4)$$

where k_1, k_2, k_3 are positive control gains. Let us introduce $u := [v_{R1} \ \omega_{R1}]^\top$. Proposition 3.1 in [98] shows that the control law (4.4) makes the origin $x_p = [0 \ 0 \ 0]^\top$ of the plant (4.3) globally asymptotically stable provided that $v_{R2}(t)$, $\omega_{R2}(t)$ and their derivatives are bounded for all times $t \geq 0$ and $\lim_{t \rightarrow \infty} v_{R2}(t) \neq 0$ or $\lim_{t \rightarrow \infty} \omega_{R2}(t) \neq 0$.

The above asymptotic stability result is obtained assuming instantaneous and continuous information. In real-life applications, continuous access to the values of y and ω_p is rarely achievable. In other words, the control signal u is typically computed using intermittent measurements corrupted by noise. The measurements of the outputs and external inputs of the plant are denoted \hat{y} and $\hat{\omega}_p$, respectively. In general, as new up-to-date values of \hat{y} and $\hat{\omega}_p$ arrive, the control law may change abruptly. Afterwards, the newly computed values u are sent to actuators. These values might be noisy and intermittently updated as well. Hence, the plant is not controlled by u but instead by \hat{u} . An illustration of such a control system is provided in Figure 4.2.

A goal of this chapter is to take advantage of the available information from the plant, i.e., \hat{y} and $\hat{\omega}_p$, and design sampling/control update instants $\mathcal{T} = \{t_1, t_2, \dots\}$ such that stability of the control system is preserved. As our intuition suggests, different \hat{y} and $\hat{\omega}_p$ may yield different time instants in \mathcal{T} . In fact, the intersampling intervals $\tau_1 = t_2 - t_1$, $\tau_2 = t_3 - t_2, \dots$, for R_1 are determined based on the distance from the desired trajectory (i.e., \hat{y}) and the nature of trajectory (i.e., $\hat{\omega}_p$). Driven by the desire to obtain intersampling instants τ_i 's as large as possible, we adopt a hybrid systems modeling formalism and analysis in this chapter. Hybrid modeling captures state jumps and permits the use of multiple models (i.e., switched systems) which in turn can be exploited to maximize intersampling intervals τ_i 's. Figure 4.3 contrasts different methods for computing τ_i 's. The solid blue line

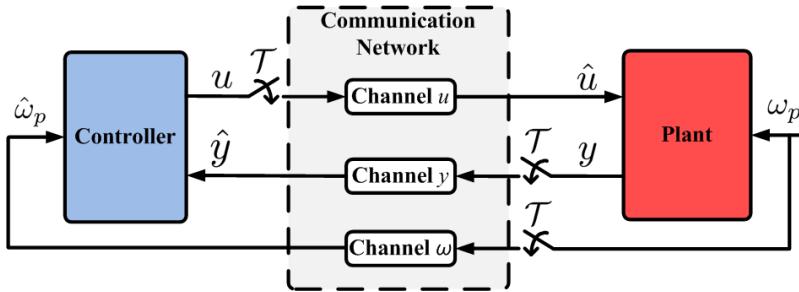


Figure 4.2: A diagram of a control system with the plant and controller interacting over a communication network with intermittent information updates. The three switches indicate that the information between the plant and controller are exchanged at discrete time instants belonging to a set \mathcal{T} .

in Figure 4.3 represents τ_i 's computed via the methodology devised in this chapter. Apparently, the use of finite horizon \mathcal{L}_p -gains (this notion somewhat corresponds to the notion of individual \mathcal{L}_p -gains considered in [99]) produces larger τ_i 's with respect to the use of unified gains. Unified gains are simply the maximum of all individual gains of a switched system. As discussed in [99], unified gains are a valid (although quite conservative) choice for the \mathcal{L}_p -gain of a switched system. However, even such conservative \mathcal{L}_p -gains of interconnected switched systems, when used in the small gain theorem, do not suffice to conclude stability of the closed-loop system. Therefore, one should use the finite horizon \mathcal{L}_p -gains of interconnected switched systems in order to decrease conservativeness, i.e., maximize τ_i 's, when applying the small gain theorem.

4.2 Problem Formulation and Assumptions

Consider a nonlinear feedback control system consisting of a plant

$$\begin{aligned}\dot{x}_p &= f_p(t, x_p, u, \omega_p), \\ y &= g_p(t, x_p),\end{aligned}\tag{4.5}$$

Chapter 4. Stability Under Intermittent Information

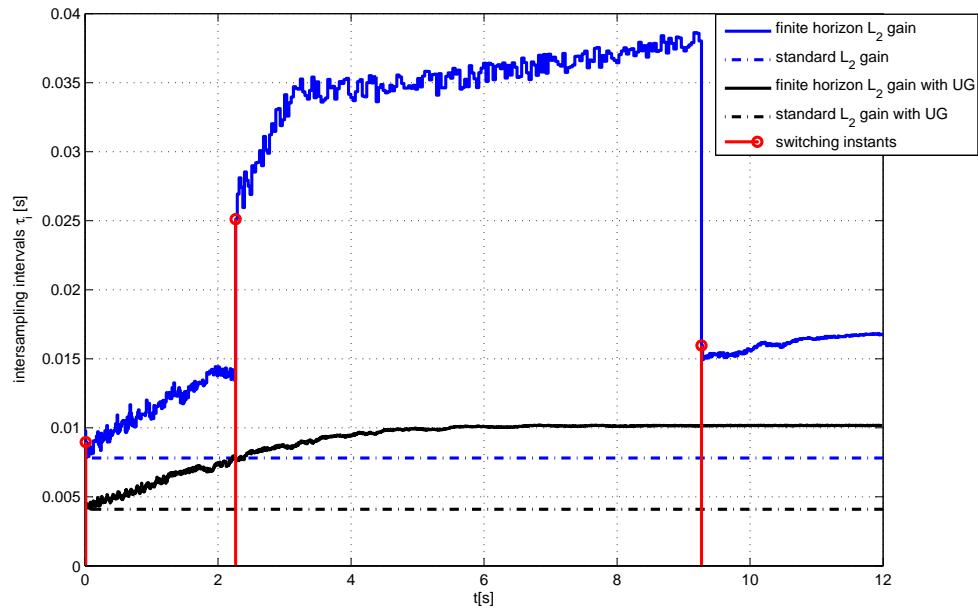


Figure 4.3: A comparison of the intersampling intervals τ_i 's obtained for different notions of \mathcal{L}_p -gains. The abbreviation UG stands for 'Unified Gain'. Red stems indicate time instants when changes in $\hat{\omega}_p$ occur. The solid blue line indicates τ_i 's generated via the methodology devised in this chapter.

and a controller

$$\begin{aligned} \dot{x}_c &= f_c(t, x_c, y, \omega_c), \\ u &= g_c(t, x_c), \end{aligned} \tag{4.6}$$

where $x_p \in \mathbb{R}^{n_p}$ and $x_c \in \mathbb{R}^{n_c}$ are the states, $y \in \mathbb{R}^{n_y}$ and $u \in \mathbb{R}^{n_u}$ are the outputs, and $\omega_p \in \mathbb{R}^{n_{\omega_p}}$ and $\omega_c \in \mathbb{R}^{n_{\omega_c}}$ are the external inputs or disturbances of the plant and controller, respectively. Notice that y is the input of the controller and u is the input of the plant. In the control systems such as the above one, one tacitly assumes that the controller is fed continuously and instantaneously by the output of the plant y , and that the control signal u continuously and instantaneously drives the plant. However, in real-life applications these assumptions are rarely fulfilled (see Figure 4.2), and sometimes excessively demanding since, as we will show here, stability of the closed-loop system can be achieved

via intermittent feedback.

In order to account for the intermittent knowledge of u by the plant, and of y and ω_p by the controller, we model the links between the plant and controller as communication networks that cause intermittent exchange of information. More precisely, we introduce the output error vector e as follows:

$$e := \begin{bmatrix} \hat{y} - y \\ \hat{u} - u \end{bmatrix} =: \begin{bmatrix} e_y \\ e_u \end{bmatrix}, \quad (4.7)$$

where \hat{y} (respectively, \hat{u}) is an estimate of y (respectively, u) computed at the controller end (respectively, the plant end). Hence, u on the right-hand side of (4.5) becomes \hat{u} while y on the right-hand side of (4.6) becomes \hat{y} . The input error vector e_ω is defined as follows:

$$e_\omega := \hat{\omega}_p - \omega_p, \quad (4.8)$$

where $\hat{\omega}_p$ is an estimate of ω_p at the controller end. We take this estimate $\hat{\omega}_p$ to be the external input to the controller, i.e., $\omega_c = \hat{\omega}_p$. In scenarios where no estimation is performed, $\hat{\omega}_p$, \hat{y} and \hat{u} are simply the most recently communicated values (or transmitted measurements) of the external input, output of the plant, and control signal. In this chapter, we assume that these values remain constant between two consecutive updates, i.e.,

$$\dot{\hat{\omega}}_p = 0, \quad \dot{\hat{y}} = 0, \quad \dot{\hat{u}} = 0, \quad (4.9)$$

which is known as the zero-order hold strategy [17]. Estimates \hat{y} and \hat{u} experience jumps when new (up-to-date) information arrives, i.e.,

$$\left. \begin{array}{l} \hat{y}(t^+) = y(t) + h_y(t, e(t)) \\ \hat{u}(t^+) = u(t) + h_u(t, e(t)) \end{array} \right\} \quad t \in \mathcal{T}. \quad (4.10)$$

It is assumed that the jump times at the controller and plant end coincide. In (4.10), h_y and h_u denote jump functions $h_y : \mathbb{R} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_y}$ and $h_u : \mathbb{R} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_u}$. Likewise, $\hat{\omega}_p$ may experience jumps when the most recent information arrives. Time instants when

Chapter 4. Stability Under Intermittent Information

jumps of $\hat{\omega}_p$ occur are denoted t_i^δ , $i \in \mathbb{N}$, and belong to a set \mathcal{T}^δ . Consequently, \mathcal{T}^δ is a subset of \mathcal{T} . Since $\hat{\omega}_p$ is not a state of the closed-loop system, the corresponding jump equations for $\hat{\omega}_p(t_i^\delta)$ are omitted.

The standing assumption in this chapter is as follows.

Assumption 1 (standing assumption) *The jump times at the controller and plant end coincide. The set of sampling instants \mathcal{T}^δ at which $\hat{\omega}_p$ changes its value satisfies $\mathcal{T}^\delta \subseteq \mathcal{T}$.*

The main problem considered herein can now be stated:

Problem 3 *Find the set of sampling instants $\mathcal{T} = \{t_1, t_2, \dots, t_i, \dots\}$ defining the intersampling intervals $\tau_1 = t_2 - t_1, \tau_2 = t_3 - t_2, \dots, \tau_i = t_{i+1} - t_i$, to update the values of \hat{y} , \hat{u} and $\hat{\omega}_p$ such that the closed-loop system (4.5)-(4.6) is stable in some appropriate sense.*

Based on the assumptions under which the Problem 3 is solved, different types of stability are achieved, e.g., stability, asymptotic stability, and \mathcal{L}_p -stability (with bias). The following cases are investigated:

Case 1 *The signals \hat{u} , \hat{y} and $\hat{\omega}_p$ are not corrupted by noise, and ω_p is constant between two consecutive t_i^δ 's belonging to \mathcal{T}^δ .*

Case 2 *The signal $\hat{\omega}_p$ is potentially corrupted by noise, and ω_p is arbitrary.*

Case 3 *The signals \hat{u} and \hat{y} are corrupted by noise.*

Case 1 represents an idealized environment. When considering more realistic environments, we expect that there exist τ_i 's yielding stability in some appropriate sense due to robustness properties of \mathcal{L}_p -stability.

4.3 Mathematical Preliminaries

4.3.1 Notation

To shorten the notation, we use $(x, y) := [x^\top \quad y^\top]^\top$. The dimension of a vector x is denoted n_x . Next, let $f : \mathbb{R} \rightarrow \mathbb{R}^n$ be a Lebesgue measurable function on $[a, b] \subset \mathbb{R}$. We use

$$\|f[a, b]\|_p := \left(\int_{[a, b]} \|f(s)\|^p ds \right)^{1/p}$$

to denote the \mathcal{L}_p norm of f when restricted to the interval $[a, b]$. If the corresponding norm is finite, we write $f \in \mathcal{L}_p[a, b]$. In the above expression, $\|\cdot\|$ refers to the Euclidean norm of a vector. If the argument of $\|\cdot\|$ is a matrix B , then it denotes the induced 2-norm of B . Eigenvalues and singular values of a matrix are denoted λ_i and σ_i , respectively. Given $x \in \mathbb{R}^n$, we define

$$\bar{x} = (|x_1|, |x_2|, \dots, |x_n|),$$

where $|\cdot|$ denotes the (scalar) absolute value function. Given $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n$, the partial order \preceq is defined as

$$x \preceq y \iff x_i \leq y_i \quad \forall i \in \{1, \dots, n\}.$$

An n -dimensional vector with all entries 1 is denoted $\mathbf{1}_n$. The set \mathcal{A}_n denotes the set of all $n \times n$ matrices and \mathcal{A}_n^+ denotes the subset of all matrices that are symmetric and have nonnegative entries. \mathbb{R}_+^n denotes the nonnegative orthant. The natural numbers are denoted \mathbb{N} or \mathbb{N}_0 when zero is included.

4.3.2 Hybrid Systems

Let $\{t_i\}_{i=0}^{\infty}$ be a sequence of increasing time instants such that $0 < t_{i+1} - t_i$ for all $i \in \mathbb{N}_0$, where t_0 is the initial time. Consider the hybrid system

$$\Sigma^{\delta} \left\{ \begin{array}{l} \dot{x} = f_h^{\delta}(t, x, \omega) \\ y = g_h^{\delta}(t, x, \omega) \\ x(t^+) = h_h^{\delta}(t, x(t)) \end{array} \right. \begin{array}{l} t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \\ t \in \mathcal{T} := \{t_i : i \in \mathbb{N}\}, \end{array} \quad (4.11)$$

with the input (or disturbance) ω , output y , and a piecewise constant and right-continuous function of time $\delta : [t_0, \infty) \rightarrow \mathcal{P}$ called a *switching signal*, where \mathcal{P} is an index set. Typically, \mathcal{P} is a bounded subset of a finite-dimensional linear vector space [100]. Even though our theory does not require countability nor finiteness of \mathcal{P} , one can always discretize a dense and uncountable \mathcal{P} for practical purposes. We assume enough regularity on f_h^{δ} and h_h^{δ} to guarantee existence of the solutions given by right-continuous functions $t \mapsto x(t)$ for a given switching signal $\delta : [t_0, \infty) \rightarrow \mathcal{P}$ starting from x_0 at $t = t_0$. Jumps of the state x occur at each $t \in \mathcal{T}$. The value of the state after the jump is given by $x(t^+) = \lim_{t' \searrow t} x(t')$ for each $t \in \mathcal{T}$. The set of switching times is denoted $\mathcal{T}^{\delta} := \{t_i^{\delta} : i \in \mathbb{N}\}$ and we define $t_0^{\delta} := t_0$ for notational convenience. In this chapter, we assume $\mathcal{T}^{\delta} \subseteq \mathcal{T}$. Similarly to τ_i 's, we define $\tau_i^{\delta} := t_{i+1}^{\delta} - t_i^{\delta}$. Notice that the above hybrid model does not prevent jump and/or switching times from accumulating in finite time, i.e., Zeno behavior. In fact, valid self-triggered control policies must guarantee absence of Zeno behavior and assure a lower bound on dwell times (see Remark 4 for more details).

4.3.3 Types of Stability

Definition 8 (stability) For $\omega \equiv 0$ and a switching signal δ , the equilibrium point $x = 0$ of Σ^{δ} is (locally) uniformly stable if there exists a class- \mathcal{K} function α and a positive constant c , independent of t_0 , such that $\|x(t)\| \leq \alpha(\|x(t_0)\|)$ for every $t \geq t_0$ and for every

Chapter 4. Stability Under Intermittent Information

$\|x(t_0)\| < c$. If this property holds for any initial state $x(t_0)$, then Σ^δ is globally uniformly stable.

Definition 9 (asymptotic stability) For $\omega \equiv 0$ and a switching signal δ , the equilibrium point $x = 0$ of Σ^δ is (locally) uniformly asymptotically stable if there exists a class- \mathcal{KL} function β and a positive constant c , independent of t_0 , such that $\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0)$ for every $t \geq t_0$ and for every $\|x(t_0)\| < c$. If this property holds for any initial state $x(t_0)$, then Σ^δ is globally uniformly asymptotically stable.

Definition 10 (\mathcal{L}_p -stability with bias b) Let $p \in [1, \infty]$. For a switching signal δ , the hybrid system Σ^δ is \mathcal{L}_p -stable with bias $b(t) \equiv b \geq 0$ from ω to y with (linear) gain $\gamma \geq 0$ if there exists $K \geq 0$ such that for all t_0 we have that $\|y[t_0, t]\|_p \leq K\|x_0\| + \gamma\|\omega[t_0, t]\|_p + \|b[t_0, t]\|_p$ for all $t \geq t_0$.

Definition 11 (\mathcal{L}_p -stability with bias b over a finite horizon τ) Let $p \in [1, \infty]$. For a switching signal δ , the hybrid system Σ^δ is \mathcal{L}_p -stable over a finite horizon $\tau \geq 0$ with bias $b(t) \equiv b \geq 0$ from ω to y with (linear) constant gain $\tilde{\gamma}(\tau) \geq 0$ if there exists a constant $\tilde{K}(\tau) \geq 0$ such that for all t_0 we have that $\|y[t_0, t]\|_p \leq \tilde{K}(\tau)\|x_0\| + \tilde{\gamma}(\tau)\|\omega[t_0, t]\|_p + \|b[t_0, t]\|_p$ for all $t \in [t_0, t_0 + \tau]$.

Definition 12 (detectability) Let $p, q \in [1, \infty]$. For a switching signal δ , the state x of Σ^δ is \mathcal{L}_p to \mathcal{L}_q detectable from (y, ω) to x with (linear) gain $\gamma \geq 0$ if there exists $K \geq 0$ such that for all t_0 we have that $\|x[t_0, t]\|_q \leq K\|x_0\| + \gamma\|y[t_0, t]\|_p + \gamma\|\omega[t_0, t]\|_p$ for all $t \geq t_0$.

Definitions 8 and 9 are taken from [56]. When $b(t) \equiv 0$, Definition 10 becomes the \mathcal{L}_p -stability definition found in [39]. In addition, Definition 12 is taken from [39]. Lastly, Definition 11 is motivated by the work in [101].

Proposition 2 \mathcal{L}_p -stability with bias $b(t) \equiv b \geq 0$ from ω to y with constant gain $\gamma \geq 0$ and \mathcal{L}_p to \mathcal{L}_p detectability of Σ^δ for a switching signal δ imply \mathcal{L}_p -stability with bias from ω to state x for the switching signal δ . The same holds when \mathcal{L}_p -stability with bias is replaced with \mathcal{L}_p -stability with bias over a finite horizon.

Proof 3 From the \mathcal{L}_p -stability with bias assumption, Definition 10 implies that there exists $K \geq 0$ such that for all t_0 we have

$$\|y[t_0, t]\|_p \leq K\|x_0\| + \gamma\|\omega[t_0, t]\|_p + \|b[t_0, t]\|_p,$$

while \mathcal{L}_p to \mathcal{L}_p detectability from (y, ω) to x with gain γ' implies that there exists $K' \geq 0$ such that

$$\|x[t_0, t]\|_p \leq K'\|x_0\| + \gamma'\|y[t_0, t]\|_p + \gamma'\|\omega[t_0, t]\|_p$$

for all $t \geq t_0$. Then, we obtain

$$\|x[t_0, t]\|_p \leq (K\gamma' + K')\|x_0\| + (\gamma\gamma' + \gamma')\|\omega[t_0, t]\|_p + \gamma'\|b[t_0, t]\|_p \quad (4.12)$$

for all $t \geq t_0$. This proves the claim.

4.4 Methodology

4.4.1 Modeling Approach

Along the lines of the approach from [39], we write the nonlinear feedback control system (4.5)-(4.6) as the following interconnected hybrid system:

$$\left. \begin{array}{l} \dot{x} = f(t, x, e, \hat{\omega}_p, e_\omega) \\ \dot{e} = g(t, x, e, \hat{\omega}_p, e_\omega) \end{array} \right\} \quad t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \quad (4.13a)$$

$$\left. \begin{array}{l} x(t^+) = x(t) \\ e(t^+) = h(t, e(t)) \end{array} \right\} \quad t \in \mathcal{T}, \quad (4.13b)$$

$$f(t, x, e, \hat{\omega}_p, e_\omega) := \begin{bmatrix} f_p(t, x_p, g_c(t, x_c) + e_u, \hat{\omega}_p - e_\omega) \\ f_c(t, x_c, g_p(t, x_p) + e_y, \hat{\omega}_p) \end{bmatrix}; \quad h(t_i, e) := \begin{bmatrix} h_y(t_i, e(t_i)) \\ h_u(t_i, e(t_i)) \end{bmatrix} \quad (4.14)$$

$$g(t, x, e, \hat{\omega}_p, e_\omega) := \begin{cases} \underbrace{\hat{f}_p(t, x_p, x_c, g_p(t, x_p) + e_y, g_c(t, x_c) + e_u, \hat{\omega}_p - e_\omega)}_{\equiv 0 \text{ for zero-order-hold estimation strategy}} - \frac{\partial g_p}{\partial t}(t, x_p) - \frac{\partial g_p}{\partial x_p}(t, x_p) f_p(t, x_p, g_c(t, x_c) + e_u, \hat{\omega}_p - e_\omega) \\ \underbrace{\hat{f}_c(t, x_p, x_c, g_p(t, x_p) + e_y, g_c(t, x_c) + e_u, \hat{\omega}_p)}_{-\frac{\partial g_c}{\partial t}(t, x_c) - \frac{\partial g_c}{\partial x_c}(t, x_c) f_c(t, x_p, g_p(t, x_p) + e_y, \hat{\omega}_p)} - \frac{\partial g_c}{\partial t}(t, x_c) - \frac{\partial g_c}{\partial x_c}(t, x_c) f_c(t, x_p, g_p(t, x_p) + e_y, \hat{\omega}_p) \end{cases} \quad (4.15)$$

where $x := (x_p, x_c)$, and functions f , g and h are given by (4.14) and (4.15). By inspecting the expression (4.15), one infers that g_p and g_c have to be piecewise continuously differentiable in order to write (4.13).

By identifying the switching signal δ with $\hat{\omega}_p$, we write (4.13) in the form of a hybrid system (4.11) as follows:

$$\begin{cases} \dot{x} = f^\delta(t, x, e, e_\omega) \\ \dot{e} = g^\delta(t, x, e, e_\omega) \end{cases} \quad t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \quad (4.16a)$$

$$\begin{cases} x(t^+) = x(t) \\ e(t^+) = h(t, e(t)) \end{cases} \quad t \in \mathcal{T}. \quad (4.16b)$$

The superscript of functions f^δ and g^δ accounts for the fact that, in general, properties of the system (4.13) change for different $\hat{\omega}_p$. In this chapter, we are interested in changes in \mathcal{L}_p -gains of (4.13) for different $\hat{\omega}_p$. The target-pursuit controller in [102] is an example where different external inputs lead to different types of stability – exponential, asymptotic and \mathcal{L}_p -stability. However, since [102] uses infinite horizon \mathcal{L}_p -gains, intersampling intervals τ_i 's in [102] are constant for a constant $\hat{\omega}_p$, i.e., τ_i 's do not adapt to changes in \hat{y} .

The form (4.16) of a closed-loop system is amenable for analysis with the small gain

Chapter 4. Stability Under Intermittent Information

theorem. To the best of our knowledge, the form (4.16) is an original modeling approach, and we are not aware of any application of the small gain theorem to systems that exhibit both switches and state jumps. For example, applications of the small gain theorem to jump systems are found in [39] and [103] while an application to switched systems is found in [99]. In the remainder of this section we prepare the terrain for Section 4.5 where τ_i 's are extracted from (4.16).

In Section 4.5, we present sufficient conditions for the interconnected system (4.16) to be \mathcal{L}_p -stable (with bias) from e_ω to (x, e) . Our approach utilizes $\hat{\omega}_p$, \hat{y} , standard \mathcal{L}_p -gains and \mathcal{L}_p -gains over a finite horizon to obtain \mathcal{L}_p -stability of the closed-loop system. We develop an approach that does not require availability of Lyapunov or storage functions. The problem of estimating \mathcal{L}_p -gains of nonlinear systems is reportedly a very hard one (see [101], [104], [40]). A number of results regarding \mathcal{L}_p -stability of systems are found in [105] and [56, Chapters 5 & 6]. However, only for special classes of systems (e.g., affine in control with a Lyapunov function that satisfies the Hamilton-Jacobi inequality) estimates of \mathcal{L}_p -gains are provided. In addition to presenting methods for calculating \mathcal{L}_p -gains from the exact model of a system (as in [105] and [56]), the authors in [106] and [107] present experimental and data-driven methods for estimating \mathcal{L}_p -gains when the exact model is not available. Even though the latter methods still lack theoretical proofs or guarantees, they show good performance in practice.

Based on the similarity between \mathcal{L}_2 -gain calculation problems and optimal control problems, [101] develops an algorithm for calculating \mathcal{L}_2 -gains over a finite horizon of nonlinear systems. Under relatively mild conditions on the system of interest (including continuity and existence of partial derivatives), the algorithm in [101] generates \mathcal{L}_2 -gains over a finite horizon that provably satisfy the necessary conditions for optimality. In Sub-sections 4.4.4 and 4.4.5, we devise a novel method to calculate \mathcal{L}_p -gains over a finite horizon.

4.4.2 Why \mathcal{L}_p -gains Over a Finite Horizon?

\mathcal{L}_p -gains over a finite horizon allow prediction of the triggering event in this chapter. In addition, as suggested by the example in Section 4.1, they produce less conservative intertransmission intervals τ_i 's than classical \mathcal{L}_p -gains when used in the small gain theorem. This is due to the fact that \mathcal{L}_p -gains over a finite horizon are monotonically nondecreasing in τ . To show this fact, we use the following characterization for $p \in [1, \infty)$ taken from [101], [108] and [109]:

$$[\tilde{\gamma}(\tau)]^p := \sup_{\omega \in \mathcal{L}_p[t_0, t_0 + \tau]} \left\{ \frac{\int_{t_0}^{t_0 + \tau} \|y(t)\|^p dt}{\int_{t_0}^{t_0 + \tau} \|\omega(t)\|^p dt} \right\}, \quad (4.17)$$

where $\|x(t_0)\| = 0$, $\|\omega[t_0, t_0 + \tau]\|_p \neq 0$, $b = 0$ and δ is fixed to be constant to generate an output y and solution x of Σ^δ . The case $p = \infty$ is similar.

Proposition 3 *The function $\tau \mapsto \tilde{\gamma}(\tau)$ is monotonically nondecreasing.*

Proof 4 Take $\tau > 0$ and choose any τ' such that $\tau' > \tau$. According to (4.17), for the horizon $[t_0, t_0 + \tau']$ we can write

$$[\tilde{\gamma}(\tau')]^p = \sup_{\omega \in \mathcal{L}_p[t_0, t_0 + \tau']} \left\{ \frac{\int_{t_0}^{t_0 + \tau} \|y(t)\|^p dt + \int_{t_0 + \tau}^{t_0 + \tau'} \|y(t)\|^p dt}{\int_{t_0}^{t_0 + \tau} \|\omega(t)\|^p dt + \int_{t_0 + \tau}^{t_0 + \tau'} \|\omega(t)\|^p dt} \right\}.$$

Now, choose $\omega \in \mathcal{L}_p[t_0, t_0 + \tau']$ such that $\omega(t) = 0$ for $t \in (t_0 + \tau, t_0 + \tau']$. This yields

$$\begin{aligned} [\tilde{\gamma}(\tau')]^p &= \sup_{\omega \in \mathcal{L}_p[t_0, t_0 + \tau']} \left\{ \frac{\int_{t_0}^{t_0 + \tau} \|y(t)\|^p dt + \int_{t_0 + \tau}^{t_0 + \tau'} \|y(t)\|^p dt}{\int_{t_0}^{t_0 + \tau} \|\omega(t)\|^p dt} \right\} \geq \\ &\geq \sup_{\omega \in \mathcal{L}_p[t_0, t_0 + \tau]} \left\{ \frac{\int_{t_0}^{t_0 + \tau} \|y(t)\|^p dt}{\int_{t_0}^{t_0 + \tau} \|\omega(t)\|^p dt} \right\} = [\tilde{\gamma}(\tau)]^p. \end{aligned}$$

Taking the p^{th} root of the above inequality shows the claim.

Since a standard (i.e., infinite horizon or classical) \mathcal{L}_p -gain γ can be defined as

$$\gamma := \sup_{\tau \geq 0} \tilde{\gamma}(\tau), \quad (4.18)$$

we conclude that $\tilde{\gamma}(\tau) \leq \gamma$ for all $\tau \geq 0$.

Recall that the small gain theorem requires $\gamma_1\gamma_2 < 1$ for stability, where γ_1 and γ_2 are the infinite horizon \mathcal{L}_p -gains of feedback interconnected systems [56]. Take

$$\gamma_1\gamma_2 \geq 1 \quad (4.19)$$

to be the triggering event that has to be precluded as it imperils closed-loop stability. In order to determine the time horizon when the triggering event might happen, we use gains over a finite horizon and trigger jumps in order to preclude the gains to satisfy

$$\tilde{\gamma}_1(\tau_i)\tilde{\gamma}_2(\tau_i) \geq 1. \quad (4.20)$$

Denoting the maximal such τ_i as τ_i^* , we want τ_i^* to be as large as possible. Due to the monotonicity property (4.18), we infer that \mathcal{L}_p -gains over a finite horizon yield larger τ_i^* 's than standard \mathcal{L}_p -gains. For example, the τ_i^* that satisfies $\tilde{\gamma}_1(\tau_i)\tilde{\gamma}_2(\tau_i) < 1$ is larger or equal than the τ_i^* that satisfies $\gamma_1\tilde{\gamma}_2(\tau_i) < 1$. Furthermore, some systems might only be \mathcal{L}_p -stable over a finite horizon and not \mathcal{L}_p -stable in the standard \mathcal{L}_p sense. For example, systems that are \mathcal{L}_p -stable over a finite horizon but not \mathcal{L}_p -stable in the standard sense are considered in Theorem 3. Hence, \mathcal{L}_p -gains over a finite horizon will be used instead of \mathcal{L}_p -gains in order to obtain less conservative (i.e., larger) τ_i in Problem 3. However, applying the small gain theorem to interconnected switched systems described by their \mathcal{L}_p -gains over a finite horizon is more involved and additional stability conditions have to be met. The following subsection provides conditions that guarantee \mathcal{L}_p -stability of such closed-loop systems via the small gain theorem when \mathcal{L}_p -gains over a finite horizon are employed.

4.4.3 \mathcal{L}_p -Stability of Hybrid Systems

The following theorem is the main result of this chapter and is used throughout the remainder of the chapter. It provides sufficient and necessary conditions for \mathcal{L}_p -stability of a system obtained by switching between systems that are \mathcal{L}_p -stable over a finite horizon. In

Chapter 4. Stability Under Intermittent Information

particular, the theorem holds when switching between \mathcal{L}_p -stable systems as well. Recall that switches (and possible state jumps) occur at time instants in \mathcal{T}^δ .

Theorem 2 Consider a hybrid system Σ^δ given by (4.11). Let $K \geq 0$ and $p \in [1, \infty)$. If δ is such that

(i) There exist constants $\tilde{K}(\tau_i^\delta), \tilde{\gamma}(\tau_i^\delta)$ such that

$$\|y[t_i^\delta, t']\|_p \leq \tilde{K}(\tau_i^\delta) \|x(t_i^{\delta+})\| + \tilde{\gamma}(\tau_i^\delta) \|\omega[t_i^\delta, t']\|_p. \quad (4.21)$$

for all $t' \in [t_i^\delta, t_{i+1}^\delta]$ and all $i \in \mathbb{N}_0$, where $\tau_i^\delta = t_{i+1}^\delta - t_i^\delta$, and such that

$$K_M := \sup_{i \in \mathbb{N}_0} \tilde{K}(\tau_i^\delta), \quad (4.22)$$

$$\gamma_M := \sup_{i \in \mathbb{N}_0} \tilde{\gamma}(\tau_i^\delta), \quad (4.23)$$

exist, and

(ii) The condition

$$\sum_{i=1}^{\infty} \|x(t_i^{\delta+})\| \leq K \|x(t_0)\|, \quad (4.24)$$

holds,

then Σ^δ is \mathcal{L}_p -stable from ω to y with constant $K_M(K+1)$ and gain γ_M for the given δ . For $p = \infty$, the same result holds with the constant $K_M K$ and gain γ_M when (4.24) is replaced with $\sup_{i \in \mathbb{N}} \|x(t_i^{\delta+})\| \leq K \|x(t_0)\|$. In addition, if the state x is \mathcal{L}_p to \mathcal{L}_p detectable, then conditions (i) and (ii) are both sufficient and necessary.

Proof 5 The proof is in the Appendix.

Remark 1 We point out that condition (i) in Theorem 2 does not simply mean that each individual system is \mathcal{L}_p -stable over τ_i^δ , $i \in \mathbb{N}_0$, as can be seen from the following. Take $\tilde{K}(\tau_i^\delta) = 0$ in (4.22) and $\tilde{\gamma}(\tau_i^\delta) = i$ in (4.23). Obviously, each of the individual systems is \mathcal{L}_p -stable over τ_i^δ , but Σ^δ is not \mathcal{L}_p -stable since $\gamma_M = \infty$.

Remark 2 Condition (4.24) is satisfied when, for example, $\|x(t_{i+1}^{\delta+})\| \leq \lambda \|x(t_i^{\delta+})\|$, where $\lambda \in [0, 1)$. This resembles the uniformly globally exponentially stable protocols from [39]. In scenarios with a finite number of time horizons (switches), λ can also be greater or equal to 1.

4.4.4 Extensions of Previous Work

This subsection relaxes assumptions in the results of [40] and computes \mathcal{L}_p -gain over a finite horizon for the hybrid system related to the output error vector e given by (4.16).

Theorem 3 Assume $\delta = \hat{\omega}_p$ is fixed to be constant. Suppose that there exist $A \in \mathcal{A}_{n_e}^+$ such that $\|A\| < \infty$, and a continuous function $\tilde{y} : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\omega} \times \mathbb{R}^{n_\omega} \rightarrow \mathbb{R}_+^{n_e}$ such that the output error dynamics in (4.16a) satisfies

$$\bar{e} = \overline{g(t, x, e, \hat{\omega}_p, e_\omega)} \preceq A\bar{e} + \tilde{y}(t, x, \hat{\omega}_p, e_\omega) \quad (4.25)$$

for all $e \in \mathbb{R}^{n_e}$ and all $(t, x(t), \hat{\omega}_p, e_\omega(t)) \in \mathcal{S}$ provided that $t \in [t_0, t_0 + \tau]$, where $\mathcal{S} \subseteq \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\omega} \times \mathbb{R}^{n_\omega}$. Then, the output error system is \mathcal{L}_p -stable from \tilde{y} to e over a finite horizon $\tau \geq 0$ for any $p \in [1, \infty]$, i.e.,

$$\|e[t_0, t_0 + \tau]\|_p = \tilde{K}_e(\tau)\|e(t_0)\| + \tilde{\gamma}_e(\tau)\|\tilde{y}[t_0, t_0 + \tau]\|_p, \quad (4.26)$$

where

$$\tilde{K}_e(\tau) = \left(\frac{\exp(\|A\|p\tau) - 1}{p\|A\|} \right)^{1/p}, \quad (4.27)$$

$$\tilde{\gamma}_e(\tau) = \frac{\exp(\|A\|\tau) - 1}{\|A\|}. \quad (4.28)$$

Proof 6 The proof is in the Appendix.

Remark 3 While [40, Section V] requires a constant A , i.e., A that is independent of $(t, x, \hat{\omega}_p, e_\omega)$, we allow A to change as \mathcal{S} changes in the subsequent time horizons (according to changes in δ and \hat{y}). This extension allows us to apply Theorem 3 to a broader class of functions g .

The above theorem relaxes the positive semidefiniteness requirement posed on A in [40, Theorem 5.1] leading to less conservative (i.e., larger) τ_i 's as shown in the remainder of this section. It also simplifies the problem of finding one such A since A 's now belong to the larger set $\mathcal{A}_{n_e}^+$. This initial A is then used as the initial point in the corresponding convex programming problem aimed at minimizing $\|A\|$ (see Subsection 4.4.5).

Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a continuous function. The work in [40] constructs a matrix A such that $\|f(A)\| = f(\|A\|)$ for any f by requiring that matrix A is symmetric, with nonnegative entries and positive semidefinite. While symmetry of the matrix and nonnegative entries are required throughout [40], positive semidefiniteness is required only in [40, Lemma 7.1]. Exploiting the fact that f in [40, Lemma 7.1] (i.e., in Theorem 3 herein) is the exponential function $f(\cdot) = \exp(\cdot)$ and A is a real symmetric matrix with nonnegative entries, the following lemmas show that the positive semidefiniteness requirement in [40, Theorem 5.1] can be relaxed for the settings herein.

Lemma 1 Suppose that $f(\cdot) = \exp(\cdot)$, and A is a real symmetric matrix. The eigenvalue of A with the largest absolute value is real and nonnegative if and only if $\|f(A)\| = f(\|A\|)$.

Proof 7 It is well known that real symmetric matrices can be diagonalized, i.e., $A = UDU^T$ where D is a diagonal matrix and U is an orthogonal matrix. Since the spectral norm is unitarily invariant [110], it is straightforward to show that the following equalities

hold

$$\begin{aligned}\|f(A)\| &= \|UDU^T\| = \|D\| = \|diag(f(\lambda_1(A)), f(\lambda_2(A)), \dots, f(\lambda_n(A))\| = \\ &= \max_k |f(\lambda_k(A))|.\end{aligned}\tag{4.29}$$

On the other hand, using definition of the induced matrix 2-norm $\|\cdot\|$, we have

$$f(\|A\|) = f(\max_k |\sigma_k|).\tag{4.30}$$

In addition, real symmetric matrices have the following property

$$\sigma_k = |\lambda_k|.\tag{4.31}$$

Using (4.31), and monotonicity and positivity of the exponential function, we conclude that expressions (4.29) and (4.30) are equal if and only if the eigenvalue of A with the largest absolute value is real and nonnegative.

Lemma 2 If A is a symmetric matrix with nonnegative entries, then the eigenvalue of A with the largest absolute value is real and nonnegative.

Proof 8 This lemma follows from the Perron-Frobenius theory of nonnegative matrices that can be found in, for example, [111, Chapter 8].

4.4.5 Minimization of $\|A\|$ is a Convex Problem

Our goal is to minimize $\|A\|$ in order to decrease $\tilde{\gamma}_e(\tau)$ in (4.28). This leads to less conservative τ_i 's in Problem 3. According to Theorem 3, we are solving the following optimization problem:

$$\text{minimize} \quad \|A\|\tag{4.32a}$$

$$\text{subject to} \quad A \in \mathcal{A}_{n_e}^+\tag{4.32b}$$

$$\text{and} \quad \overline{g(t, x, e, \hat{\omega}_p, e_\omega)} \preceq A\bar{e} + \tilde{y}(t, x, \hat{\omega}_p, e_\omega)\tag{4.32c}$$

Chapter 4. Stability Under Intermittent Information

for all $e \in \mathbb{R}^{n_e}$ and all $(t, x, \hat{\omega}_p, e_\omega) \in \mathcal{S}$. After analyzing the above optimization problem, the following proposition is obtained.

Proposition 4 *The optimization problem (4.32) is convex.*

Proof 9 It is well known that $\|A\|$ is a convex function of A (see [112, Chapter 3]). Now, let us prove that constraints (4.32b) and (4.32c) yield a convex set. First, a convex combination of two matrices in $\mathcal{A}_{n_e}^+$ is again in $\mathcal{A}_{n_e}^+$. This is due to the fact that symmetric matrices with nonnegative elements remain symmetric with nonnegative elements when multiplied with nonnegative scalars and when added together. Let us now show that inequality (4.32c) yields a convex set in A . Pick any $(t', x', \hat{\omega}_p, e'_\omega) \in \mathcal{S}$ (recall that $\delta = \hat{\omega}_p$ is constant by the assumption of Theorem 3). Now, let us introduce substitutions $E(\bar{e}) = \overline{g(t', x', e, \hat{\omega}_p, e'_\omega)}$ and $F = \tilde{y}(t', x', \hat{\omega}_p, e'_\omega)$. Our goal is to show that if

$$E(\bar{e}) \preceq A_1 \bar{e} + F, \quad (4.33)$$

$$E(\bar{e}) \preceq A_2 \bar{e} + F, \quad (4.34)$$

then

$$E(\bar{e}) \preceq [(1 - \alpha)A_1 + \alpha A_2]\bar{e} + F \quad (4.35)$$

where $\alpha \in [0, 1]$. Using (4.33) and (4.34), we obtain

$$(1 - \alpha)A_1 \bar{e} + \alpha A_2 \bar{e} \succeq (1 - \alpha)(E(\bar{e}) - F) + \alpha(E(\bar{e}) - F) = E(\bar{e}) - F$$

which is equivalent to (4.35). Since $(t', x', \hat{\omega}_p, e'_\omega)$ was picked arbitrarily from \mathcal{S} , therefore (4.35) holds for all $e \in \mathbb{R}^{n_e}$ and all $(t, x, \hat{\omega}_p, e_\omega) \in \mathcal{S}$. The fact that the intersection of convex sets is a convex set concludes the proof.

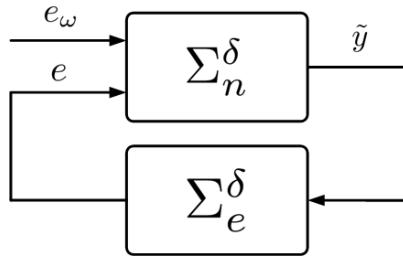


Figure 4.4: Interconnection of the nominal hybrid system Σ_n^δ and the output error hybrid system Σ_e^δ .

4.5 Input-Output Triggering

We now apply the small gain theorem to the closed-loop system (4.16) in order to solve Problem 3 for Cases 1, 2 and 3.

We treat (4.16) as a feedback interconnection of the nominal system Σ_n^δ given by the first equations in (4.16a) and (4.16b) with the output \tilde{y} obtained when applying Theorem 3, and the output error system Σ_e^δ given by the second equations in (4.16a) and (4.16b). The interconnection of Σ_n^δ and Σ_e^δ is illustrated in Figure 4.4. The nominal system Σ_n^δ and the input error vector e_ω is something that we cannot change, and the only information available to us are measurements $\hat{\omega}_p$ and \hat{y} . What we are able to design are transmission instants t_i 's. By designing t_i 's, we impact \mathcal{L}_p -gains over a finite horizon of the output error system Σ_e^δ . Recall that when $\hat{\omega}_p$ changes its value (i.e., switches) at time instant t_i , then, for the sake of convenience, we refer to such t_i as t_j^δ where $j \leq i$. The possible mismatch in indexes of t_i and t_j^δ is due to the fact that $\hat{\omega}_p$ does not necessarily change at each t_i , $i \in \mathbb{N}$. In what follows, we lay out the main ideas put together in order to devise input-output triggering.

Since we do not control values of the switching signal δ but merely switching instants, we assume that Σ_n^δ is \mathcal{L}_p -stable from (e_ω, e) to \tilde{y} for some $p \in [1, \infty]$ and all $\delta \equiv c$ where $c \in \mathcal{P}$. The corresponding \mathcal{L}_p -gains are denoted γ_n^δ .

Chapter 4. Stability Under Intermittent Information

Let us investigate what happens with \mathcal{L}_p -gains of Σ_n^δ and Σ_e^δ between two consecutive switching instants t_i^δ 's. Since δ is constant between consecutive t_i^δ 's, the \mathcal{L}_p -gain of Σ_n^δ is γ_n^δ . Notice that Theorem 2 is applicable to Σ_e^δ with t_i^δ 's substituted with t_i 's (this variant of Theorem 2 is provided in [113]). In case that jumps of e at t_i 's given by the second equation in (4.16b) satisfy condition (4.24), we proceed as follows. By making sure that $\tilde{\gamma}_e^\delta(\tau_i) < \kappa/\gamma_n^\delta$, where $\kappa \in (0, 1)$, over all τ_i 's until the next switch, the small gain condition $\gamma_n^\delta \gamma_e^\delta \leq \kappa$ is satisfied. Notice that γ_e^δ plays the same role as γ_M in Theorem 2. Since in this chapter we are interested in obtaining intersampling intervals τ_i as large as possible, one chooses κ as great as possible (e.g., $\kappa = 0.999$). Next, we assume that the second equation in (4.16a) can be written in the form of (4.25), apply Theorem 3 and obtain a stabilizing sampling policy $\tau_i \in (0, \tau_i^*]$ where

$$\tau_i^* = \frac{1}{\|A^\delta\|} \ln \left(\kappa \frac{\|A^\delta\|}{\gamma_n^\delta} + 1 \right). \quad (4.36)$$

This policy yields the closed-loop system (4.16) \mathcal{L}_p -stable over a horizon τ_i^δ . In the above expression we use A^δ instead of A to emphasize that Σ_e^δ , in general, exhibits switches.

Up to this point we did not take advantage of introducing \mathcal{S} in Theorem 3. Let us from now on require that the projection of \mathcal{S} on $\mathbb{R}^{n_x} \times \mathbb{R}^{n_\omega} \times \mathbb{R}^{n_\omega}$ for all $t \in [t_i, t_{i+1}]$ is a compact set \mathcal{C}_i . Likewise, the corresponding A^δ is denoted A_i^δ . It is well known that $\|A_i^\delta\|$ is a continuous function of the entries of A_i^δ . If the entries of A_i^δ are continuous functions of $(x, \hat{\omega}_p, e_\omega) \in \mathcal{C}_i$, then $\|A_i^\delta\|$ attains its maximum and minimum on any compact \mathcal{C}_i , i.e., $\|A_i^\delta\| < \infty$ on \mathcal{C}_i . Therefore, the assumption $\|A_i^\delta\| < \infty$ from Theorem 3 can be replaced by the following assumptions:

- i) $t \in [t_i, t_{i+1}]$,
- ii) $\mathcal{C}_i := \{(x(t), \hat{\omega}_p, e_\omega(t)) : t \in [t_i, t_{i+1}]\}$ is a compact set, and
- iii) $\|A_i^\delta\|$ is a continuous function on \mathcal{C}_i .

Chapter 4. Stability Under Intermittent Information

We now write (4.36) as

$$\tau_i^* = \frac{1}{\|A_i^\delta\|} \ln \left(\kappa \frac{\|A_i^\delta\|}{\gamma_n^\delta} + 1 \right). \quad (4.37)$$

Basically, at time instants t_i , $i \in \mathbb{N}$, we perform the following steps:

1. Obtain measurements $\hat{y}(t_i)$ and $\hat{\omega}_p(t_i)$,
2. Extract state estimate $\hat{x}(t_i)$ from the measurements,
3. Update the control law (4.6),
4. Actuate the plant with $\hat{u}(t_i)$,
5. Estimate \mathcal{C}_i from (4.5)-(4.6) using reachability analysis (for more refer to [52] and [53]),
6. Compute τ_i^* using (4.37), and
7. Pick $\tau_i \in (0, \tau_i^*]$ such that Zeno behavior is avoided.

Sets \mathcal{C}_i 's can be easily (over)estimated when the upper bound on $\bar{\omega}_p$ is known. The operator $\bar{\cdot}$ is defined in Subsection 4.3.1. Notice that a decrease in $\|A_i^\delta\|$ results in an increase in τ_i^* according to (4.37). Therefore, in cases where more conservative estimates of \mathcal{C}_i lead to larger $\|A_i^\delta\|$, we want the estimate of \mathcal{C}_i to be as accurate as possible. Notice that A_i^δ and γ_n^δ are precisely what gives rise to input-output triggering. In other words, A_i^δ changes as $\hat{\omega}_p(t_i)$ and $\hat{y}(t_i)$ change, and γ_n^δ changes as $\hat{\omega}_p(t_i)$ changes producing different τ_i^* 's in (4.37). A remark is in order regarding Step 2. Since Step 2 is essentially an estimation problem, we assume this problem is solvable. For example, assuming that the controller can access its state x_c , a bijective mapping from x_p to y given by (4.5) for any $t \geq t_0$ makes this estimation problem solvable.

Lastly, let us investigate conditions that yield absence of Zeno behavior and accumulation points of switching instants. In other words, let us show existence of a positive lower

Chapter 4. Stability Under Intermittent Information

bound on τ_i 's. Before we proceed further, notice that τ_i^* given by (4.37) attains values in \mathbb{R}^+ and is a monotonically decreasing function in both $\|A_i^\delta\|$ and γ_n^δ .

Assumption 2 Assume that $\sup_{\delta \equiv c, c \in \mathcal{P}} \gamma_n^\delta \leq Z$, where Z is a positive constant. In addition, assume that $\sup_{\mathcal{C}_i, i \in \mathbb{N}} \|A_i^\delta\| \leq V$, where V is a positive constant.

Remark 4 Under Assumption 2, there exists $\tau_{\min}^* = \frac{1}{V} \ln (\kappa \frac{V}{Z} + 1) > 0$ such that for every τ_i^* computed via (4.37) the following holds: $\tau_i^* \geq \tau_{\min}^*$. By choosing $\tau_i = \rho \tau_i^*$, $\rho \in (0, 1]$, in Step 7, we infer that intervals τ_i 's between two consecutive transmission instants are lower bounded by a strictly positive time $\rho \tau_{\min}^*$. Hence, the unwanted Zeno behavior [55] and accumulation of switching instants in finite time (refer to [114] and [100]) are avoided. In other words, the triggering condition (4.37) does not yield continuous feedback that might be impossible to achieve.

Existence of an upper bound on τ_i 's is shown under the following assumption.

Assumption 3 Assume that $\inf_{\delta \equiv c, c \in \mathcal{P}} \gamma_n^\delta \geq Z_l$, where Z_l is a positive constant.

Remark 5 Under Assumption 3, we infer that τ_i^* 's are upper bounded by

$$\tau_{\max}^* = \lim_{a \searrow 0} \frac{1}{a} \ln \left(\kappa \frac{a}{Z_l} + 1 \right) = \frac{\kappa}{Z_l}.$$

By choosing $\tau_i = \rho \tau_i^*$, $\rho \in (0, 1]$, in Step 7, we infer that intervals τ_i 's between two consecutive transmission instants are upper bounded by a strictly positive time $\rho \tau_{\max}^*$.

From a robustness viewpoint one typically prefers $\gamma_n^\delta = 0$ for some $\delta \equiv c, c \in \mathcal{P}$. However, notice that even when state x of Σ_n^δ is detectable from (\tilde{y}, e_ω, e) , this state x can grow unbounded on the infinite time horizon for some $\gamma_n^\delta = 0$ due to Proposition 2. In other words, \mathcal{C}_i 's expand unbounded which in turn violates the compactness requirement on \mathcal{C}_i .

4.5.1 Input-Output Triggering for Case 1

For Case 1, the hybrid system (4.16) becomes

$$\left. \begin{array}{l} \dot{x} = f^\delta(t, x, e, 0) \\ \dot{e} = g^\delta(t, x, e, 0) \end{array} \right\} \quad t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \quad (4.38a)$$

$$\left. \begin{array}{l} x(t^+) = x(t) \\ e(t^+) = 0 \end{array} \right\} \quad t \in \mathcal{T}. \quad (4.38b)$$

In other words, ω_p is known accurately at any time and the values of u and y are received without delays and distortions at transmission instants t_i 's.

Theorem 4 For $p \in [1, \infty)$ assume that

- (i) There exists $L \geq 0$ such that $\bar{\omega}_p \preceq L \mathbf{1}_{n_\omega}$,
- (ii) There exist $A^\delta \in \mathcal{A}_{n_e}^+$ such that $\|A^\delta\|$ is a continuous function of $(x, \hat{\omega}_p, e_\omega)$, and that there exists a continuous function \tilde{y} such that the output error dynamics (4.13a) satisfies (4.25) for all $e \in \mathbb{R}^{n_e}$ and all $(x, \hat{\omega}_p, e_\omega) \in \mathcal{C}_i$ on $t \in [t_i, t_{i+1}]$ where \mathcal{C}_i is a compact set,
- (iii) Σ_n^δ is \mathcal{L}_p -stable from (e_ω, e) to \tilde{y} for every $\delta \equiv c$, $c \in \mathcal{P}$, with gain γ_n^δ such that Z and Z_l in Assumptions 2 and 3 exist,
- (iv) Sampling policy (4.37) yields x that satisfies (4.24) for given $\delta : [t_0, \infty) \rightarrow \mathcal{P}$, and
- (v) x is \mathcal{L}_p to \mathcal{L}_p detectable from (\tilde{y}, e_ω, e) .

Then, the equilibrium point $(x, e) = 0$ of the closed-loop system (4.38) is globally uniformly asymptotically stable for given δ .

Proof 10 Notice that (i) defines \mathcal{P} used in (iii). The proof is in the Appendix.

Chapter 4. Stability Under Intermittent Information

Theorem 5 For $p = \infty$ replace condition (iv) of Theorem 4 with

(iv) Sampling policy (4.37) yields x that satisfies $\sup_{i \in \mathbb{N}} \|x(t_i^{\delta+})\| \leq K \|x(t_0)\|$ for given $\delta : [t_0, \infty) \rightarrow \mathcal{P}$.

Then, the equilibrium point $(x, e) = 0$ of the closed-loop system (4.38) is globally uniformly stable for given δ .

Proof 11 Similarly to the previous proof, the small gain theorem yields

$$\|(x[t_0, t], e[t_0, t])\|_\infty \leq K \|(x_0, e_0)\|$$

for a constant $K \geq 0$ and all $t \geq t_0$. Since $\|s[t_0, t]\|_\infty := \sup_{t' \in [t_0, t]} \|s(t')\|$ for a right-continuous signal $s(t)$, the proof is completed.

Remark 6 Let us consider the case where u or y (or both) is the output of a state observer. In other words, the plant or controller (or both) is fed with an estimate provided by an observer. Consequently, in (4.38b) we have $e(t^+) = e_o(t)$ for all $t \in \mathcal{T}$ where $e_o(t)$ is the observer error. For $p \in [1, \infty)$, if the observer error satisfies condition (4.24) (e.g., exponentially converging observers such as the Luenberger observer for linear systems), then Theorem 4 holds. For $p = \infty$, if the observer error is bounded, then Theorem 5 holds. In addition, let us consider the case of lossy communication channels. If there is an upper bound on the maximum number of successive dropouts, say $N^D \in \mathbb{N}$, simply use τ_i^*/N^D as intertransmission intervals in order for Theorem 4 to hold.

Remark 7 In order to account for possible delays introduced by the communication networks in Figure 4.2, one can use scattering transformation for the small gain theorem [115]. Provided that Σ_n^δ and Σ_e^δ are input-feedforward output-feedback passive systems satisfying certain conic relations, the work in [115] makes stability properties of (4.16) independent of constant time delays and Theorem 4 is applicable again. In light of [43], these constant time delays are allowed to be larger than the intersampling intervals τ_i 's.

4.5.2 Input-Output Triggering for Case 2

Let us rewrite (4.16) as follows:

$$\left. \begin{array}{l} \dot{x} = f^\delta(t, x, e, e_\omega) \\ \dot{e} = g^\delta(t, x, e, e_\omega) \end{array} \right\} t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \quad (4.39a)$$

$$\left. \begin{array}{l} x(t^+) = x(t) \\ e(t^+) = 0 \end{array} \right\} t \in \mathcal{T}. \quad (4.39b)$$

Following the same approach as in the previous subsection, we reach the next result:

Theorem 6 When $p \in [1, \infty)$ assume that the conditions of Theorem 4 are met, and when $p = \infty$ assume that the conditions of Theorem 5 are met. Then, the closed-loop system (4.39) is \mathcal{L}_p -stable from e_ω to (x, e) for given δ .

Proof 12 Following the proof of Theorem 4, we reach (A.22) which is \mathcal{L}_p -stability from e_ω to (x, e) .

4.5.3 Input-Output Triggering for both Case 2 and Case 3

Let us rewrite (4.16) as follows:

$$\left. \begin{array}{l} \dot{x} = f^\delta(t, x, e, e_\omega) \\ \dot{e} = g^\delta(t, x, e, e_\omega) \end{array} \right\} t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \quad (4.40a)$$

$$\left. \begin{array}{l} x(t^+) = x(t) \\ e(t^+) = \nu(t) \end{array} \right\} t \in \mathcal{T}, \quad (4.40b)$$

where $\nu \in \mathbb{R}^{n_e}$ models measurement noise and/or quantization error. Assume that $\|\nu\|$ has an upper bound ν_B . Using Theorem 3, i.e., expression (A.10), one can verify that Σ_e^δ is \mathcal{L}_p -stable with bias

$$b = \nu_B \exp \eta \quad (4.41)$$

Chapter 4. Stability Under Intermittent Information

from \tilde{y} to e where $\eta = \sup_{\delta, i \in \mathbb{N}_0} \|A_i^\delta\| \tau_i^*$. Under the hypotheses of Theorem 7, Assumptions 2 and 3 hold (see the proof of Theorem 2). Hence, this supremum exists. The following theorem concludes this subsection:

Theorem 7 *Assume that the assumptions of Theorem 6 hold. Then, the closed loop system (4.40) is \mathcal{L}_p -stable with bias from (e_ω) to (x, e) for given δ .*

Proof 13 *First, let us show that the small gain theorem holds for \mathcal{L}_p -stability with bias. Assume we have*

$$\|\tilde{y}[t_0, t]\|_p \leq K_n \|x(t_0)\| + \gamma_n \|e[t_0, t]\|_p + \gamma_n \|e_\omega[t_0, t]\|_p,$$

and

$$\|e[t_0, t]\|_p \leq K_e \|e(t_0)\| + \gamma_e \|\tilde{y}[t_0, t]\|_p + \|b[t_0, t]\|_p,$$

for all $t \geq t_0$ and such that $\gamma_n \gamma_e < 1$. Using the same steps as in the proof of the small gain theorem in [56], we obtain

$$\|\tilde{y}[t_0, t]\|_p \leq \frac{1}{1 - \gamma_n \gamma_e} [\gamma_n \|e_\omega[t_0, t]\|_p + K_n \|x(t_0)\| + \gamma_n K_e \|e(t_0)\| + \gamma_n \|b[t_0, t]\|_p]$$

for all $t \geq t_0$. For $\|(y, e)[t_0, t]\|_p$ we obtain an analogous inequality. Using the fact that $\|(\tilde{y}, e)[t_0, t]\|_p \leq \|\tilde{y}[t_0, t]\|_p + \|e[t_0, t]\|_p$, we obtain

$$\|(\tilde{y}, e)[t_0, t]\|_p \leq K \|x(t_0)\| + \gamma \|e_\omega[t_0, t]\|_p + K_1 \|b[t_0, t]\|_p$$

for all $t \geq t_0$ where $K, K_1, \gamma \geq 0$. Notice that the above expression is equivalent to (A.20). Therefore, the small gain theorem holds for \mathcal{L}_p -stable systems with bias. The remainder of the proof follows the proof of Theorem 4.

Remark 8 *Noisy measurements can be a consequence of quantization errors. According to [116], interconnections of systems with linear \mathcal{L}_p -gains prone to quantization errors do not yield closed-loop systems with linear \mathcal{L}_p -gains. Hence, \mathcal{L}_p -stability with bias in Theorem 7 cannot be relaxed without contradicting the work in [116].*

Remarks 6 and 7 are applicable to both Case 2 and Case 3.

4.6 Case Study - Trajectory Tracking

In this section, we apply the input-output triggered update policy (4.37) to the *trajectory tracking* controller presented in Section 4.1.

Since the controller (4.4) is not a dynamic controller, we have $f_c \equiv 0$. Next, we take $x_c = (v_{R1}, \omega_{R1})$ and

$$u = g_c(t, x_c) = x_c. \quad (4.42)$$

Recall that the states of the plant (4.3) are measured directly, i.e.,

$$y = g_p(t, x_p) = x_p, \quad (4.43)$$

and assume that the communication network for transmitting the control input (v_{R1}, ω_{R1}) to the actuators of R_1 can be neglected due to on-board controllers. Because of the absence of the communication network for transmitting u and $f_c \equiv 0$, we have that $e_u \equiv 0$. Consequently, we can exclude x_c from x and take $x = x_p$. Notice that f_p is given by (4.3). Recall that the external input is $\omega_p = (v_{R2}, \omega_{R2})$. Now we have

$$e = \hat{x} - x = [e_1 \ e_2 \ e_3]^\top, \quad (4.44)$$

and

$$e_\omega = \hat{\omega}_p - \omega_p = [e_{\omega,1} \ e_{\omega,2}]^\top. \quad (4.45)$$

After substituting (4.3), (4.4), (4.42), (4.43), (4.44) and (4.45) into (4.14) and (4.15), we obtain (compare with expression (4.16))

$$\dot{e} = -\dot{x} = \underbrace{\begin{bmatrix} -Q - P(x_2 + e_2)x_2 - k_3(x_3 + e_3)x_2 + R + k_1(x_1 + e_1) - S \\ \hat{\omega}_{R2}x_1 + P(x_2 + e_2)x_1 + k_3(x_3 + e_3)x_1 - T \\ e_{\omega,2} + P(x_2 + e_2) + k_3(x_3 + e_3) \end{bmatrix}}_{=g^\delta(t,x,e,e_\omega)=-f^\delta(t,x,e,e_\omega)} \quad (4.46)$$

and (compare with expression (4.25))

$$\bar{e} \preceq \underbrace{\begin{bmatrix} k_1 & k_2|\hat{v}_{R2}|M_2 & \max\{|\hat{v}_{R2}|, k_3 M_2\} \\ k_2|\hat{v}_{R2}|M_2 & k_2|\hat{v}_{R2}|M_1 & \max\{k_2|\hat{v}_{R2}|, k_3 M_1\} \\ \max\{|\hat{v}_{R2}|, k_3 M_2\} & \max\{k_2|\hat{v}_{R2}|, k_3 M_1\} & k_3 \end{bmatrix}}_{A_i^\delta = \text{initial point for the convex program (4.32)}} \bar{e} +$$

$$+ \underbrace{\begin{bmatrix} k_2|\hat{v}_{R2}|x_2^2 + |k_1 x_1 + e_{\omega,1} \cos x_3 - \hat{\omega}_{R2} x_2 - k_3 x_2 x_3| \\ k_2|\hat{v}_{R2} x_1 x_2| + |\hat{\omega}_{R2} x_1 + k_3 x_1 x_3 - T| \\ k_2|\hat{v}_{R2} x_2| + |e_{\omega,2} + k_3 x_3| \end{bmatrix}}_{\tilde{y}(t, x, \hat{\omega}_p, e_\omega)} \quad (4.47)$$

where

$$P = k_2 \hat{v}_{R2} \frac{\sin(x_3 + e_3)}{x_3 + e_3},$$

$$Q = \hat{\omega}_{R2} x_2,$$

$$R = \hat{v}_{R2} \cos(x_3 + e_3),$$

$$S = (\hat{v}_{R2} - e_{\omega,1}) \cos x_3,$$

$$T = (\hat{v}_{R2} - e_{\omega,1}) \sin x_3,$$

and $|x_1| \leq M_1$, $|x_2| \leq M_2$. Constants M_1 and M_2 are obtained from the compact sets \mathcal{C}_i 's (see the next paragraph for more details about computing M_1 and M_2). We choose $k_1 = 1.5$, $k_2 = 1.2$ and $k_3 = 1.1$. In addition, ω_p takes values in $[-3, 3] \times [-3, 3]$. Since scenarios with both Case 2 and 3 are more realistic, this section includes numerical results for such a scenario. When emulating noisy environments, we use $e_\omega \in U([-0.3, 0.3] \times [-0.3, 0.3])$ and $\nu \in U([-0.15, 0.15] \times [-0.15, 0.15] \times [-0.15, 0.15])$ where $U(\mathcal{C})$ denotes the uniform distribution over a compact set \mathcal{C} .

Before presenting numerical results, let us verify that the hypotheses of Theorem 7 hold and provide details for Steps 1-7 (these steps are found below expression (4.37)). Since the upper bound on ω_p is known, we confine measurements $\hat{\omega}_p$ to the same set, i.e., $\delta \in \mathcal{P} = [-3, 3] \times [-3, 3]$. Hence, hypothesis (i) holds. In other words, if we obtain $\hat{\omega}_p(t_i) \notin \mathcal{P}$ due to measurement noise, we use the closest value in \mathcal{P} (with respect to the Euclidean distance) for $\hat{\omega}_p(t_i)$. Using such $\hat{\omega}_p(t_i)$ and $\hat{y}(t_i)$, we update control signal $u(t_i)$. Notice that the second and the third component of \mathcal{C}_i belong to compact sets $[-3, 3] \times [-3, 3]$ and

Chapter 4. Stability Under Intermittent Information

$[-0.3, 0.3] \times [-0.3, 0.3]$, respectively. The first component of \mathcal{C}_i is computed as follows (i.e., this is Step 5). Utilizing $\hat{y}(t_i)$, we obtain $\hat{x}(t_i) \in \hat{y}(t_i) \pm [-0.15, 0.15] \times [-0.15, 0.15] \times [-0.15, 0.15]$. Starting from this $\hat{x}(t_i)$, and due to the fact that $u(t_i)$ is the linear and angular velocity of $R1$ and remains constant until t_{i+1} , we readily obtain reachable states $\hat{x}(t_i + \tau_i)$ for any $\tau_i \geq 0$ using $u(t_i)$ and the upper bounds on ω_p . Inspecting the form of A_i^δ in (4.47), we infer that the first two components of x are needed for (4.47) to hold on any \mathcal{C}_i . Due to the properties of \preceq , we choose the maximum values of $|x_1|$ and $|x_2|$ in \mathcal{C}_i denoted M_1 and M_2 , respectively. Since A_i^δ and \tilde{y} are continuous functions, we infer that hypothesis (ii) is fulfilled. Combining the approach of [101] and the power iterations method [106], we estimate \mathcal{L}_2 -gains Σ_n^δ over \mathcal{P} and obtain $Z = 96$ and $Z_l = 9$. Hence, hypothesis (iii) holds. Hypothesis (iv) is verified as the simulation progresses for given $\delta : [t_0, \infty) \rightarrow \mathcal{P}$. In case hypothesis (iv) may get violated, simply decrease τ_i 's in Step 7 by decreasing ρ (see Remark 4). We point out that $\rho = 1$ was used in the simulations in this section. Finally, hypothesis (v) is inferred from (4.46) and (4.47) as follows. It can be shown that

$$\|x\|^2 \leq k(\|\tilde{y}\|^2 + \|(e, e_\omega)\|^2)$$

for any $k \geq 2$. Integrating both sides of the last inequality over $[t_0, t]$ for any $t \geq t_0$ and taking the square root, yields

$$\|x[t_0, t]\|_2 \leq \sqrt{k}\|\tilde{y}[t_0, t]\|_2 + \sqrt{k}\|(e, e_\omega)[t_0, t]\|_2.$$

In other words, the state x of the system Σ_n^δ is \mathcal{L}_2 to \mathcal{L}_2 detectable from (e, e_ω, \tilde{y}) .

In the simulations, we use $\omega_p(t) = (1, 1)t_{[0, 2.26]} + (0.6, 0.15)t_{[2.26, 9.25]} + (2, 2)t_{[9.25, 12]}$, where $t_{\mathcal{I}}$ is the indicator function on an interval \mathcal{I} , i.e., $t_{\mathcal{I}} = t$ when $t \in \mathcal{I}$ and zero otherwise. The corresponding \mathcal{L}_2 -gains Σ_n^δ are as follows: $\gamma_n^{(0.6, 0.15)} = 22$, $\gamma_n^{(1, 1)} = 53$ and $\gamma_n^{(2, 2)} = 56$. In order to illustrate Theorem 7 and the mechanism behind (4.37), we superpose a continuous signal $e_\omega(t) \in [-0.3, 0.3] \times [-0.3, 0.3]$, where $t \in [0, 12]$, onto the above $\omega_p(t)$, and update the control law with $\hat{\omega}_p$ being $(0.6, 0.15)$, $(1, 1)$ or $(2, 2)$ accordingly. This way, we are able to use a fixed γ_n^δ between two switches so that the impact of

changes in \hat{y} on τ_i 's is easier to observe. The obtained numerical results are provided in Figure 4.5. As can be seen from Figure 4.5, intersampling intervals τ_i 's tend to increase as $\|x\|$ approaches the origin because M_1 and M_2 decrease. In addition, the abrupt changes of τ_i at 2.26 s and 9.25 s, visible in Figure 4.5(c), are the consequence of the abrupt changes in $\hat{\omega}_p$. In other words, τ_i 's adapt to the changes in $\hat{\omega}_p$. This adaptation of τ_i 's follows from (4.37) where individual gains are considered instead of the unified gain [99]. The simulation results obtained using the unified gain $\sup_{\delta \equiv c, c \in \mathcal{P}} \gamma_n^\delta = Z = 96$ achieved for $\delta \equiv (3, 3)$ and the corresponding $A_i^{(3,3)}$ in (4.37) are shown in Figure 4.6. Apparently, the use of the unified gains decreases τ_i 's, does not allow for adaptation of τ_i , and yet does not necessarily yield stability of the closed-loop system since (4.24) does not have to hold (a similar observation is found in [99]). Consequently, the number of transmissions in the scenario depicted in Figure 4.5 is 580 while in the scenario depicted in Figure 4.6 is 1377. Finally, it should be mentioned that oscillations of x in Figures 4.5(a) and 4.6(a) are an inherited property of the controller, and not a consequence of intermittent feedback.

4.7 Conclusion

In this chapter we present a methodology for input-output triggered control of nonlinear systems. Based on the currently available measurements of the output and external input of the plant, a sampling policy yielding the closed-loop system stable in some sense is devised. Using the formalism of \mathcal{L}_p -gains and \mathcal{L}_p -gains over a finite horizon, the small gain theorem is employed to prove stability, asymptotic, and \mathcal{L}_p -stability (with bias) of the closed-loop system. Different types of stability are a consequence of different assumptions on the noise environment causing the mismatch between the actual external input and output of the plant, and the measurements available to the controller via feedback. The closed-loop systems are modeled as hybrid systems, and a novel result regarding \mathcal{L}_p -stability of such systems is presented. Finally, our input-output triggered sampling policy

Chapter 4. Stability Under Intermittent Information

is exemplified on a trajectory tracking controller for velocity-controlled unicycles.

The future work is dedicated to applying scattering transformation between the controller and plant in order to eliminate detrimental effects of delays. Furthermore, actuators with saturation will be analyzed. In order to obtain larger intertransmission intervals, zero-order hold estimation strategies will be replaced with model-based estimation of control signals and plant outputs. Finally, we expect our results (with slight modifications) to hold for ISS of hybrid systems.

Chapter 4. Stability Under Intermittent Information

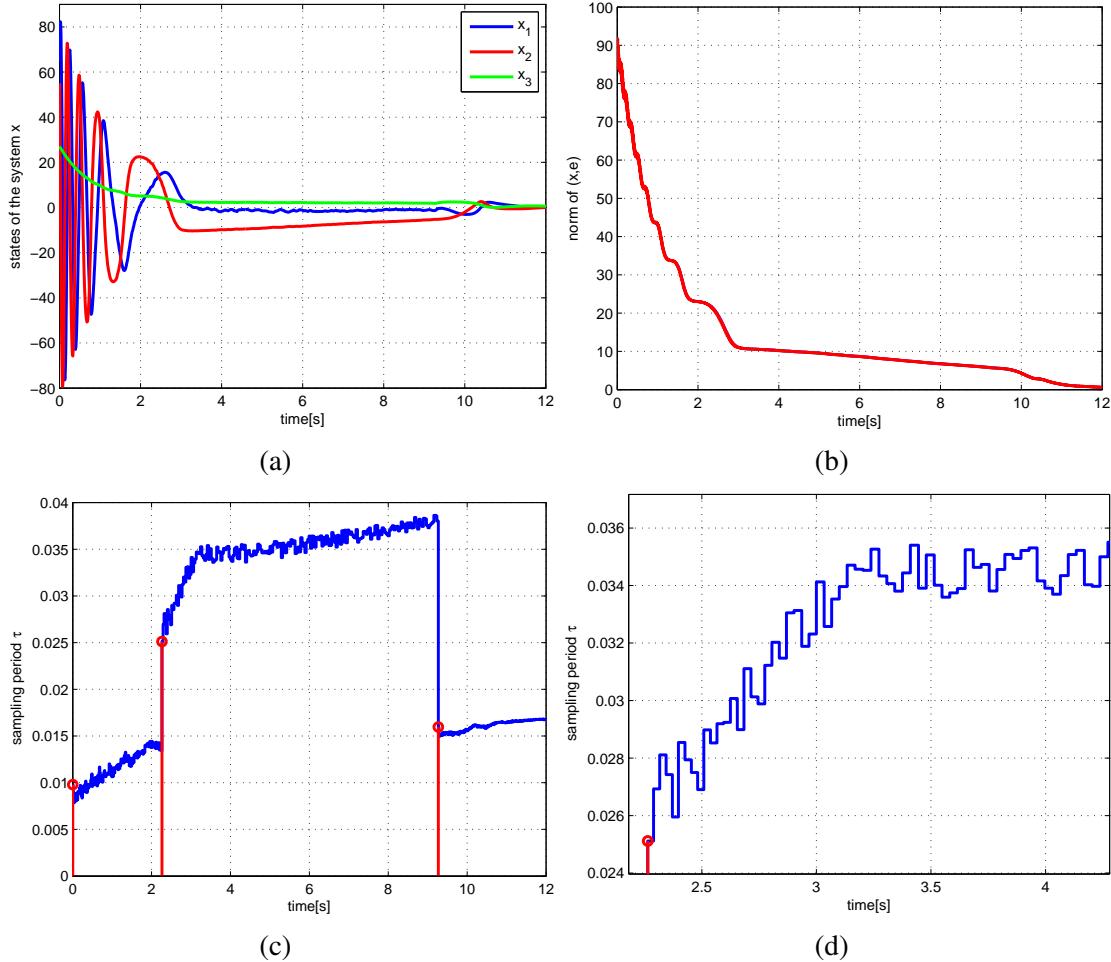


Figure 4.5: A realistic scenario illustrating input-output triggering: (a) States x of the tracking system; (b) Norm of (x, e) ; (c) Values of intersampling intervals τ_i 's between two consecutive transmissions. Red stems indicate time instants when changes in δ happen; and, (d) A detail from Figure 5.4(c).

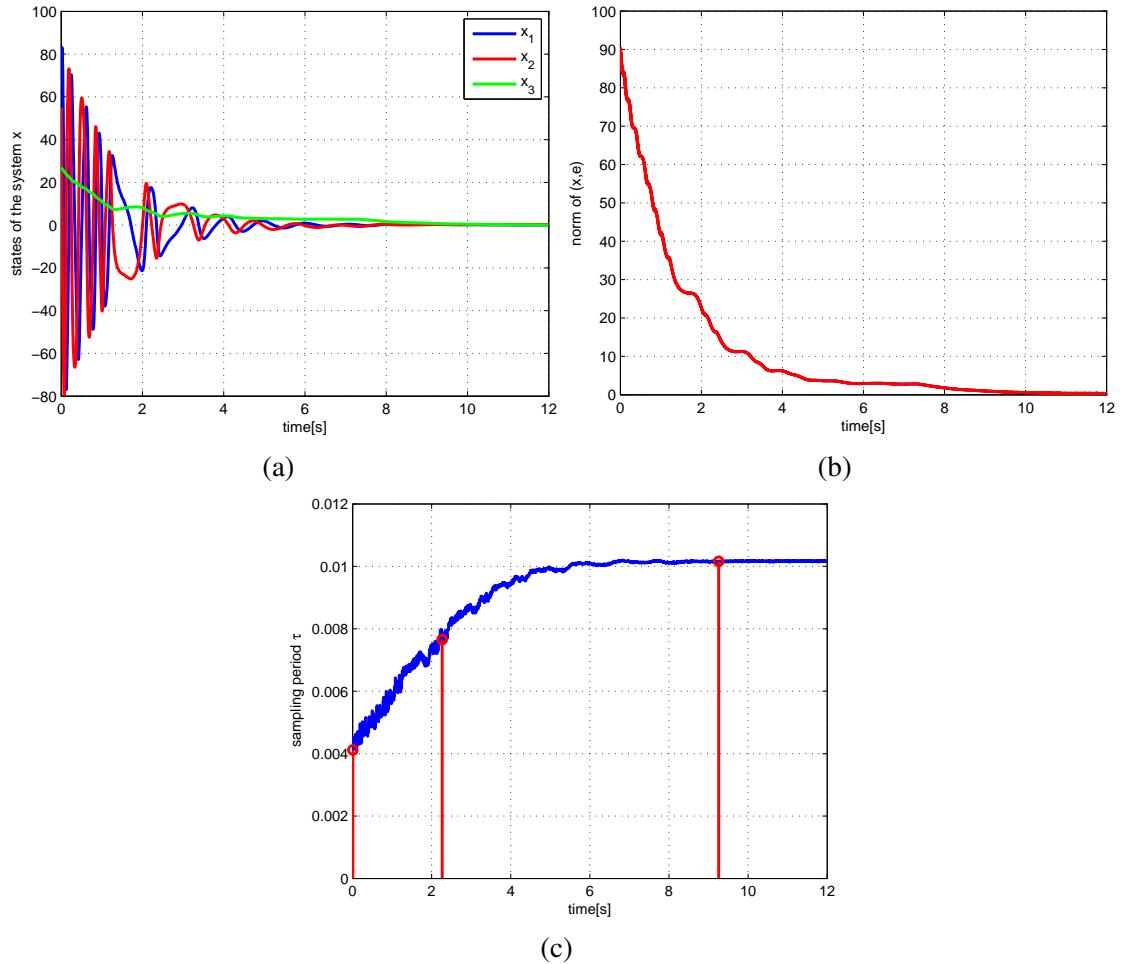


Figure 4.6: A realistic scenario illustrating input-output triggering using the unified gains: (a) States x of the tracking system; (b) Norm of (x, e) ; and (c) Values of intersampling intervals τ_i 's between two consecutive transmissions. Red stems indicate time instants when changes in δ happen.

Chapter 5

Optimal Self-Triggering

Recent years have witnessed an increasing interest in *event-triggered* implementations of control laws (refer to Chapter 4). Many works, such as [45], [46], [42], [47], [117] and [48], replace the traditional periodic paradigm, where up-to-date information are transmitted and control laws are executed in a periodic fashion, with the event-triggered paradigm. In the event-triggered paradigm, one defines a desired performance, and sampling (i.e., transmission of up-to-date information) is triggered when an event, called a *triggering event*, representing the unwanted performance occurs. The work in [46] successfully applies event-triggering to control, estimation and optimization tasks. A variant of event-triggering, known as *self-triggering*, uses the current sampling instance to predict and preclude an occurrence of the triggering event (refer to [46], [47] and [48]). In comparison with event-triggering, self-triggering decreases requirements posed on sensors and processors in embedded systems. There is also a *state-triggered* variant of self-triggering that exploits the value of the system state in the last feedback transmission [45]. In order to simplify the presentation of this chapter and improve readability, we refer to all these paradigms simply as *intermittent feedback*. This term is found in, for example, [3] and [4]. Intermittent feedback is motivated by the rational use of expensive resources at disposition in an effort to decrease energy consumption, processing and sensing requirements. Consequently,

Chapter 5. Optimal Self-Triggering

autonomy and life span of the components increase.

At the moment, the research community is interested in extending intersampling intervals as much as possible without taking into account a deterioration in the performance due to intermittent feedback. In cases where energy consumption for using sensors, transmitting the obtained information, and executing control laws is relatively inexpensive compared to the slower convergence and excessive use of control power, extending intersampling intervals is not desirable. For instance, think of an airplane driven by an autopilot system designed to follow the shortest path between two points. Any deviation from the shortest path caused by intermittent feedback increases total fuel consumption. This increase in fuel consumption is probably more costly than the cost of energy saved due to intermittent feedback (i.e., a decrease in energy consumption due to obtaining up-to-date measurements, transmitting them to controllers, and executing control laws less often). In this chapter, we encode these energy consumption trade-offs in a cost function, and design an Approximate Dynamic Programming (ADP) approach that yields optimal intertransmission intervals with respect to the cost function.

The main contributions of this chapter are threefold: a) formulation of the optimal self-triggering problem as a Dynamic Programming (DP) problem; b) employment of Particle Filters (PFs) fed by intermittent feedback to account for partially observable states; and c) formulation of properties that successful approximation architectures in ADP approaches satisfy. To the best of our knowledge, the problem of optimal intermittent feedback has yet to be addressed.

The remainder of the chapter is organized as follows. Section 5.1 presents the problem of optimal intermittent feedback and assumptions under which the problem is solved. The methodology brought together to solve the problem is presented in Section 5.2. The proposed methodology is verified on a trajectory tracking controller in Section 5.3. Finally, conclusions are drawn in Section 5.4.

5.1 Problem Statement and Assumptions

Consider a time-invariant nonlinear feedback control system consisting of a plant

$$\begin{aligned}\dot{x}_p &= f_p(x_p, u, \omega_p), \\ y &= g_p(x_p),\end{aligned}\tag{5.1}$$

and a controller

$$\begin{aligned}\dot{x}_c &= f_c(x_c, y, \omega_c), \\ u &= g_c(x_c)\end{aligned}\tag{5.2}$$

where $x_p \in \mathbb{R}^{n_p}$ and $x_c \in \mathbb{R}^{n_c}$ are the states, $y \in \mathbb{R}^{n_y}$ and $u \in \mathbb{R}^{n_u}$ are the outputs, and $\omega_p \in \mathbb{R}^{n_{\omega_p}}$ and $\omega_c \in \mathbb{R}^{n_{\omega_c}}$ are the external/exogenous inputs or disturbances of the plant and controller, respectively. Notice that y is the input of the controller, and u is the input of the plant. Let us denote the compound state of the closed-loop systems (5.1) and (5.2) by $x = (x_p, x_c)$ where $x \in \mathbb{R}^{n_x}$.

In the above control system, one tacitly assumes that the controller is fed continuously and instantaneously by the output y and the external input ω_p of the plant. However, in real-life applications this assumption is rarely fulfilled, and sometimes excessively demanding since stability of closed-loop systems can be achieved via intermittent feedback. For example, extremely fast processing units, sensors and communication devices are needed in order to emulate continuous and instantaneous feedback using digital technology. At the same time, intermittent feedback has detrimental effects on the performance of the control loop.

In order to account for the intermittent knowledge of y and ω_p by the controller, we model the links between the plant and controller as communication networks that cause intermittent exchange of information. More precisely, we introduce the output error vector e as follows:

$$e(t) := \hat{y}(t) - y(t)\tag{5.3}$$

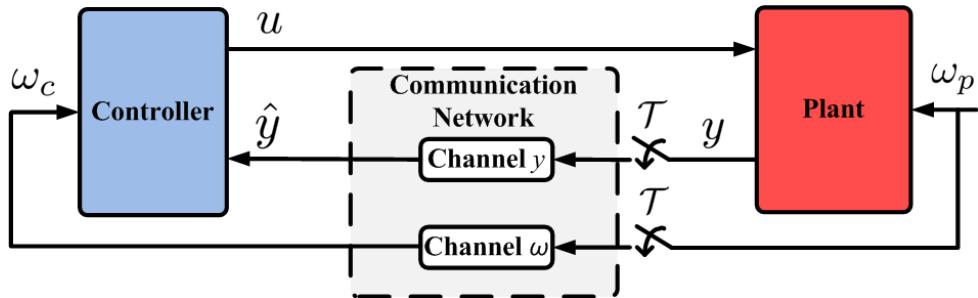


Figure 5.1: A diagram of a plant and controller with discrete transmission instants and communication channels giving rise to *intermittent feedback*.

where \hat{y} is an estimate of y performed from the perspective of the controller, and the input error vector e_ω as follows:

$$e_\omega(t) := \hat{\omega}_p(t) - \omega_p(t) \quad (5.4)$$

where $\hat{\omega}_p$ is an estimate of ω_p from the perspective of the controller. For the sake of simplicity, we take \hat{y} and $\hat{\omega}_p$ to be the most recently communicated values (or transmitted measurements) of the output and external input of the plant. This zero-order-hold strategy is commonly used for estimation and control under communication network constraints [17]. Now we introduce $\mathcal{T} := \{t_i : i \in \mathbb{N}_0\}$ as the set of time instants when outputs and externals inputs of the plant are transmitted over communication networks. Finally, many control laws are designed such that $\omega_c = \hat{\omega}_p$. Examples are trajectory tracking controllers as in [47], [117] and [98]. An illustration of a control system indicating the communication channels that cause intermittent information is provided in Figure 5.1.

Next, we want to minimize the following cost function $V : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ that captures *performance vs. energy* trade-offs

$$V_{\tau_i}(x_0) = \mathbb{E}_{e_\omega} \left\{ \sum_{i=1}^{\infty} \gamma^i \underbrace{\left[\int_{t_{i-1}}^{t_i} (x_p^T Q x_p + u^T R u) dt + S \right]}_{l(x_p, u, \tau_i)} \right\} \quad (5.5)$$

Chapter 5. Optimal Self-Triggering

over all sampling policies τ_i and for all initial conditions $x_0 \in \mathbb{R}^{n_x}$. In addition, $\gamma \in (0, 1)$ is a discount factor that makes the sum (5.5) finite provided that $l(x_p, u, \tau_i)$ is bounded over all $[t_{i-1}, t_i]$ where $t_i \in \mathcal{T}$ and

$$t_i = t_{i-1} + \tau_{i-1}. \quad (5.6)$$

For clarity, we use τ_i instead of $\tau(\hat{y}(t_i), \hat{\omega}_p(t_i))$, but one has to keep in mind that, in general, intersampling intervals τ_i 's depend on the most recently transmitted information from the plant, i.e., on $\hat{y}(t_i)$ and $\hat{\omega}_p(t_i)$. In addition, Q and R are positive definite matrices, and $S \in \mathbb{R}$ such that $S \geq 0$ is the cost incurred for sampling y and ω_p , transmitting \hat{y} and $\hat{\omega}_p$, and updating the control signal u . In (5.5), the conditional expectation over a stochastic signal e_ω is denoted \mathbb{E}_{e_ω} .

The main problem considered herein can now be stated:

Problem 4 For the system (5.1) and (5.2) with values of $\hat{\omega}_p$ and \hat{y} received at t_i , $i \in \mathbb{N}_0$, find time intervals τ_i 's until the next transmission instants such that (5.5) is minimized.

We solve the above problem under the following assumptions:

Assumption 4 \hat{y} is corrupted by measurement noise.

Assumption 5 $\hat{\omega}_p$ is corrupted by measurement noise, and ω_p is arbitrary between two consecutive t_i 's.

Due to these assumptions, we have to deal with partially observable states (see Subsection 5.2.5 for more details).

5.2 Methodology

This section presents the tools brought together to solve Problem 4 under Assumptions 4 and 5. Starting from the input-output triggering that provides maximal stabilizing intersampling intervals, we find optimal τ_i^* 's for the cost function (5.5) resorting to ADP and PF.

5.2.1 Input-Output Triggering via the Small Gain Theorem

Building on the small gain theorem, we develop *input-output triggering* in Chapter 4. In other words, based on the currently available but outdated measurements of the outputs and external inputs of a plant, a simple expression for when to obtain new up-to-date measurements and execute the control law is provided. The details of this approach are out of scope of this chapter, and not needed in order to follow the remainder of the chapter. In fact, with slight modifications, our ADP approach is applicable to any self-triggered sampling policy (e.g., [45], [46], [47] and [48]). Essentially, our input-output triggered sampling policy outputs maximal allowable intersampling intervals τ_i^{\max} 's that provably yield stable closed-loop system (5.1) and (5.2). Starting from these τ_i^{\max} 's, the work presented herein finds

$$\tau_i^* \in [0, \tau_i^{\max}] \quad (5.7)$$

that minimize (5.5). Because we know the upper bounds τ_i^{\max} 's of stabilizing sampling policies, τ_i^* 's obtained in this chapter provably stabilize the plant. A comprehensive treatment of the problem whether ADP solutions of optimal problems yield stability can be found in [118].

5.2.2 Dynamic Programming

Notice that the cost function (5.5) has the standard DP form. Let us now introduce a state transition function f that maps $x(t_{i-1})$, $u(t_{i-1})$ and $\hat{\omega}_p(t_{i-1})$ to $x(t_i)$ given some e_ω over $[t_{i-1}, t_i]$, i.e.,

$$x(t_i) = f(x(t_{i-1}), u(t_{i-1}), \tau_{i-1}, \hat{\omega}_p(t_{i-1}), e_\omega). \quad (5.8)$$

Due to intermittent feedback and presence of nonlinearities in the plant and controller, the state transition function over τ_i 's, in general, cannot be given in closed form with, for example, a difference equation [36]. This is a typical impediment one faces when analyzing nonlinear systems under intermittent feedback. Therefore, in general, the state transition function (5.8) needs to be simulated using (5.1), (5.2), $\hat{y}(t_{i-1})$, $\hat{\omega}_p(t_{i-1})$ and e_ω over time horizon τ_{i-1} .

Next, let us assume that e_ω is a stationary stochastic process. Consequently, since we consider the infinite horizon problem (5.5) and a time-invariant control system (5.1) and (5.2), τ_i is not a function of t_i . Hence, we simply write τ instead of τ_i in the remainder of the chapter. Solving the DP problem of minimizing (5.5) backwards through time is combinatorially impossible since the state space x in (5.5) is uncountable. Therefore, we write the stochastic control problem of minimizing (5.5) over τ in its equivalent form known as the Hamilton-Jacobi-Bellman equation

$$V^*(z) = \inf_{\tau \in [0, \tau^{\max}]} \left(l(z, u, \tau) + \gamma \mathbb{E}_{e_\omega} \{ V^*(f(z, u, \tau, \hat{\omega}_p, e_\omega)) \} \right) \quad (5.9)$$

where $V^*(z)$ is called the optimal value function (or optimal cost-to-go function), and represents the cost incurred by an optimal policy τ^* when the initial condition in (5.5) is z . It is well known that V^* is the unique fixed point of (5.9). Therefore, the problem of minimizing (5.5) boils down to finding V^* in (5.9).

For notational convenience, we introduce the Bellman operator \mathcal{M} as

$$\mathcal{M}g = (\mathcal{M}g)(z) = \inf_{\tau \in [0, \tau^{\max}]} \left(l(z, u, \tau) + \gamma \mathbb{E}_{e_\omega} \{ g(f(z, u, \tau, \hat{\omega}_p, e_\omega)) \} \right) \quad (5.10)$$

for any $g : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$. Since $\gamma \in (0, 1)$, it can be shown that \mathcal{M} is a contraction, i.e.,

$$\|\mathcal{M}u - \mathcal{M}v\|_s \leq \gamma \|u - v\|_s \quad (5.11)$$

where $\|v\|_s = \sup_{z \in \mathbb{R}^{n_x}} v(z)$. This contraction result is found in, among others, [64, Proposition 3.9.2] and [119]. The set \mathcal{B} of all bounded, real valued functions with the norm $\|\cdot\|_s$ is a Banach space. Therefore, for each initial $V^0 \in \mathcal{B}$, the sequence of value functions $V^{n+1} = \mathcal{M}V^n = \mathcal{M}^{n+1}V^0$ converges to V^* .

Due to the ‘‘curses of dimensionality’’, solving (5.9) for $V^*(z)$ or iterating an initial V^0 is deemed intractable for most of the problems of interest; hence, we employ ADP. In what follows, our goal is to find an approximation \hat{V}^* of V^* .

Two remarks are in order. First, it can be shown that the problem of finding an optimal τ^* for each state in (5.9) is non-convex. However, since τ is confined to a rather small compact set $[0, \tau^{\max}]$, we utilize gradient search methods with constraints starting from different initial points in order to obtain τ^* . Second, the conditional expectation \mathbb{E}_{e_ω} in (5.9) can be obtained in closed form only for special cases. Otherwise, it can be calculated numerically by replacing the integral with a sum using a quadrature approximation. In Section 5.3, we use the Simpson formula [120].

5.2.3 Approximate Dynamic Programming - Value Iteration

Among a number of methods in ADP, we choose the Value Iteration (VI) method for its simplicity and a wide spectrum of applications. Notice that \mathcal{B} is an infinite dimensional vector space, meaning that it takes infinitely many parameters to describe V^* . Therefore, one introduces an approximate value function \hat{V}^i of V^i where $i \in \mathbb{N}_0$. Then, the VI method iteratively applies \mathcal{M} to an approximate value function \hat{V}^0 until $\|\hat{V}^{i+1} - \hat{V}^i\|_s < \epsilon$ where $\epsilon > 0$. Approximate value functions \hat{V}^i , $i \in \mathbb{N}_0$, can be represented in finite parameter approximation architectures such as neural networks (NNs). Note that it is not possible to

obtain true value functions V^i 's but only their approximations; hence, we write \hat{V}^i instead of V^i . Basically, VI performs

$$\hat{V}^{i+1} = \mathcal{M}\hat{V}^i, \quad (5.12)$$

until

$$\|\hat{V}^{i+1} - \hat{V}^i\|_s < \epsilon. \quad (5.13)$$

In order to calculate \hat{V}^{i+1} in (5.12), we need to apply (5.10) over all $z \in \mathbb{R}^{n_x}$. Obviously, this is computationally impossible since \mathbb{R}^{n_x} contains uncountably many points. Therefore, many ADP approaches focus on a compact subset $\mathcal{C}_x \subset \mathbb{R}^{n_x}$, choose a finite set of points $\mathcal{X} \subset \mathcal{C}_x$, and calculate \hat{V}^{i+1} only for the points in \mathcal{X} . Afterwards, the values of \hat{V}^{i+1} for $\mathcal{C}_x \setminus \mathcal{X}$ are obtained via some kind of interpolation/generalization.

5.2.4 Approximation Architecture

The problem of choosing an approximation architecture that fits \hat{V}^{i+1} to $\hat{V}^{i+1}(\mathcal{X})$ and, at the same time, is able to interpolate/generalize for $\hat{V}^{i+1}(\mathcal{C}_x \setminus \mathcal{X})$ appears to be crucial in order for ADP to converge. It is considered that ADP is not converging when either the stopping criterion (5.13) is never reached (refer to [119] and [121]) or \hat{V}^* is not an accurate approximation of V^* [122]. The latter criterion is concerned with suboptimality of the obtained solution. In this chapter, we focus on the former deferring suboptimality analyses for future work.

It appears that the key property that has to be preserved by an approximation architecture is the contraction property (5.11) (refer to [119], [121] and [123]). In [119], the author classifies function approximators as expansion or contraction approximators. *Expansion approximators*, such as linear regressors and NNs, exaggerate changes on $\mathcal{C}_x \setminus \mathcal{X}$. *Contraction approximators* (or local averagers), such as k-nearest-neighbor, linear interpolation,

Chapter 5. Optimal Self-Triggering

grid methods and other state aggregations methods, conservatively respond to changes in \mathcal{X} . Therefore, on the one hand, a VI that includes a contraction approximator always converges, in the sense of (5.13), to the fixed point determined by the approximator, say \hat{V}_{ca}^* . However, not much can be said about the value $\|V^* - \hat{V}_{ca}^*\|_s$ (see [119] and [121]). On the other hand, a VI that includes an expansion approximator might diverge [121]. However, NNs are still a widely used approximation architecture due to their notable successes (for example, [124], [125], and [126]), adaptive architectures [127], performance guarantees under certain assumptions [123], and inventions of novel NN architectures. These novel NN architectures are also called nonparametric approximation [123] and they adapt to the training data. Examples are kernel-based NNs (refer to [128] and [129]) and recurrent NNs (refer to [130] and [129]). Almost all references in this subsection provide advantages and disadvantages of different approximation architectures. This fact shows the importance of the choice of approximation architectures.

A goal of this chapter is not to advocate certain architectures. Instead, based on our experience and the references above, we define properties that successful approximation architectures possess (e.g., contraction approximators and kernel-based NNs). Based on the specifics of the problem (dimensionality of the problem, availability and density of data, available processing power, memory requirements, etc.), one should choose a suitable architecture.

Desired Properties

Assume that $V^*(x)$ is a smooth function on \mathcal{C}_x , and choose a smooth function approximator. At the i^{th} step, where $i \in \mathbb{N}_0$, randomly pick any $x' \in \mathcal{C}_x$, calculate $(\mathcal{M}\hat{V}^i)(x')$, and fit $\hat{V}^i(x')$ to $(\mathcal{M}\hat{V}^i)(x')$ obtaining \hat{V}^{i+1} . We are looking for an approximation architecture that satisfies the following properties

- (i) $\hat{V}^{i+1}(x') = (\mathcal{M}\hat{V}^i)(x');$

- (ii) $\text{supp}(\hat{V}^{i+1} - \hat{V}^i) = \mathcal{C}_i$, where $\text{supp}(f) = \{x : f(x) \neq 0\}$ is the support of a function f , and $\mathcal{C}_i \subset \mathcal{C}_x$ is a convex and compact neighborhood of x' ; and
- (iii) for any $c \in \partial\mathcal{C}_i$, where $\partial\mathcal{C}_i$ denotes the boundary of \mathcal{C}_i , the following holds

$$\hat{V}^{i+1}[S] \subseteq [\hat{V}^{i+1}(c), \hat{V}^{i+1}(x')], \quad (5.14)$$

where $\hat{V}^{i+1}[S]$ is the image of the segment S connecting x' and c ,

in order to have $\|V^{i+1} - V^i\|_s \rightarrow 0$ as $i \rightarrow \infty$.

Remark 9 Let us consider two value functions \hat{u}^i and \hat{v}^i in the i^{th} step, and apply \mathcal{M} at a randomly chosen x'_i . Due to (5.11), we have $\|(\mathcal{M}\hat{u}^i)(x'_i) - (\mathcal{M}\hat{v}^i)(x'_i)\| \leq \gamma \|\hat{u}^i(x'_i) - \hat{v}^i(x'_i)\|$. From property (i), we conclude that $\|\hat{u}^{i+1}(x'_i) - \hat{v}^{i+1}(x'_i)\| \leq \gamma \|\hat{u}^i(x'_i) - \hat{v}^i(x'_i)\|$. Since the approximator is smooth, we know that there exists a neighborhood $\mathcal{C}'_i \subseteq \mathcal{C}_i$ of x'_i such that $\sup_{x \in \mathcal{C}'_i} \|\hat{u}^{i+1}(x) - \hat{v}^{i+1}(x)\| \leq \sup_{x \in \mathcal{C}'_i} \|\hat{u}^i(x) - \hat{v}^i(x)\|$. This means that the nonexpansion property required in [119] is obtained locally around x'_i . The nonexpansion property from [119] is basically (5.11) when γ is replaced with 1. Finally, property (iii) eliminates counterexamples in which the Lebesgue measures of \mathcal{C}'_i , $i \in \mathbb{N}_0$, tend to zero. Consequently, generalization of the approximation architecture is ensured.

Remark 10 Property (i) is the accuracy requirement in order to preserve (5.11). Property (ii) is the “local property” found in [119], [121] and [127]. This local property is built in the activation functions of the kernel-based NNs. Property (iii) is used to ensure that \mathcal{C}'_i ’s are not merely x'_i ’s. In addition, property (iii) curbs expansiveness on $\mathcal{C}_i \setminus \mathcal{C}'_i$.

Remark 11 Notice that Desired Properties imply online learning of NNs [129]. The motivation behind this choice lies in the fact that it is straightforward to check properties (i), (ii) and (iii) in online learning. Moreover, since we randomly pick points $x'_i \in \mathcal{C}_x$ in each step, we do not have to specify \mathcal{X} . By choosing random x'_i ’s, we also avoid the problem of

exploration vs. exploitation (see [64] and [65] for more). On the other side, when using batch learning, properties (i), (ii) and (iii) cannot be guaranteed since NNs are expansion approximators. In fact, not until we switched to online learning in the example from Section 5.3, convergence was obtained. An extension of Desired Properties for batch learning and the problem of choosing \mathcal{X} are left for the future work.

Remark 12 As the stopping criterion we use the following: when $\|\hat{V}^{i+1} - \hat{V}^i\|_s < \epsilon$ for $N \in \mathbb{N}$ consecutive steps, the value iteration method has converged.

5.2.5 Partially Observable States

Notice that the approximate value function $\hat{V}(x)$ is a function of state x . Up to this point we did not take into account that x is not available due to Assumptions 4 and 5. In other words, we are solving the DP problem (5.5) with partially observable states. More details about strategies for solving DP problems with partially observable states are found in Chapter 5 of [62].

Let us assume that the controller can access its state x_c . Consequently, the controller can calculate u at any given time. However, the controller does not have access to the state of the plant x_p but merely to $\hat{\omega}_p$ and \hat{y} . We circumvent this problem by introducing a particle filter (refer to [29] and [30]) that provides estimates \hat{x}_p of the actual state x_p . More details about the problem of estimation under intermittent information can be found in [2]. It is well known that particle filters are suitable for nonlinear processes, non-Gaussian and nonadditive measurement and process noise.

More precisely, we model the closed-loop system (5.1) and (5.2) as

$$\begin{aligned} x_p(t_i) &= f_p^d(x(t_{i-1}), u(t_{i-1}), \tau_{i-1}, \hat{\omega}_p(t_{i-1}), e_\omega), \\ \hat{y}(t_i) &= g(x_p(t_i), \nu), \end{aligned} \tag{5.15}$$

where f_p^d represents a discrete transition function of the plant obtained in similar fashion as (5.8), and statistics of the process noise e_ω and measurement noise ν are known and time invariant. Based on (5.15), we build a particle filter that extracts \hat{x}_p from $\hat{\omega}_p$ and \hat{y} and feeds the controller. Details of our particle filter implementation can be found in Chapter 3.

We deal with partially observable states by first obtaining $\hat{V}^*(x)$ for the case of perfect state information. Then, we employ particle filtering and iterate $\hat{V}^*(x)$ using (5.12) to obtain the approximation of $\hat{V}^*(\hat{x})$ for the case of partially observable states. The motivation behind obtaining $\hat{V}^*(\hat{x})$ from $\hat{V}^*(x)$ lies in the fact that particle filters need several steps to obtain a reliable estimate \hat{x} starting from a poor initial estimate. Basically, since $\hat{V}^*(x)$ is a close estimate of $\hat{V}^*(\hat{x})$ when \hat{x} is a close estimate of x , we exploit $\hat{V}^*(x)$ to fine tune $\hat{V}^*(\hat{x})$ without a need to explore.

5.3 Case Study - Trajectory Tracking

In this section, we apply the optimal self-triggered sampling policy to the *trajectory tracking* controller presented in [98]. Besides for trajectory tracking, this controller can be employed in leader-follower, target-pursuit, obstacle avoidance and waypoint following problems.

In [98], a velocity-controlled unicycle robot R_1 given by

$$\dot{x}_{R1} = v_{R1} \cos \theta_{R1}, \quad \dot{y}_{R1} = v_{R1} \sin \theta_{R1}, \quad \dot{\theta}_{R1} = \omega_{R1} \quad (5.16)$$

tracks a trajectory generated by a virtual velocity-controlled unicycle robot R_2 with states $(x_{R2}, y_{R2}, \theta_{R2})$, and linear and angular velocities v_{R2} and ω_{R2} , respectively. See Figure 5.2

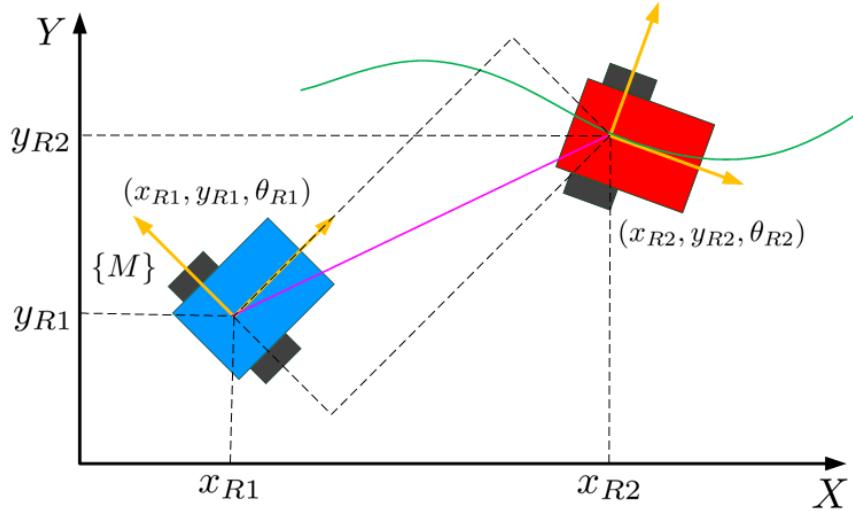


Figure 5.2: An illustration of the trajectory tracking problem considered in Chapter 5.

for an illustration. The tracking error x_p in the coordinate frame $\{M\}$ of robot R_1 becomes

$$x_p = \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \end{bmatrix} = \begin{bmatrix} \cos \theta_{R1} & \sin \theta_{R1} & 0 \\ -\sin \theta_{R1} & \cos \theta_{R1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{R2} - x_{R1} \\ y_{R2} - y_{R1} \\ \theta_{R2} - \theta_{R1} \end{bmatrix}. \quad (5.17)$$

Applying the following control law

$$\begin{aligned} v_{R1} &= v_{R2} \cos x_3 + k_1 x_1, \\ \omega_{R1} &= \omega_{R2} + k_2 v_{R2} \frac{\sin x_3}{x_3} x_2 + k_3 x_3 \end{aligned} \quad (5.18)$$

where k_1, k_2 and k_3 are positive control gains, [98] shows that control law (5.18) makes the origin $x_p = [0 \ 0 \ 0]^T$ globally asymptotically stable provided that $v_{R2}(t), \omega_{R2}(t)$ and their derivatives are bounded for all times $t \geq 0$ and $\lim_{t \rightarrow \infty} v_{R2}(t) \neq 0$ or $\lim_{t \rightarrow \infty} \omega_{R2}(t) \neq 0$.

Since the controller (5.18) is not a dynamic controller, we have that $x = x_p$. Next, for the sake of simplicity, we use the following measurement model

$$\hat{y}(t_i) = x(t_i) + \nu. \quad (5.19)$$

Chapter 5. Optimal Self-Triggering

In addition, the external input is $\omega_p = [v_{R2} \ \omega_{R2}]^T$. When emulating noise in (5.15), we use $e_\omega \in U([-0.3, 0.3] \times [-0.3, 0.3])$ and $\nu \in U([-0.15, 0.15] \times [-0.15, 0.15] \times [-0.15, 0.15])$ where $U(\mathcal{S})$ denotes the uniform distribution over a compact set \mathcal{S} .

The following coefficients were used in the cost function (5.5): $Q = 0.1I_3$, $R = 0.1I_2$, $S = 15$ and $\gamma = 0.96$ where I_n is the $n \times n$ identity matrix and $n \in \mathbb{N}$. A remark is in order regarding the choice of Q , R and S . On the one hand, as we decrease S and keep Q and R fixed, the obtained sampling policy τ approaches zero. In other words, as the energy consumption for sampling, transmitting and processing decreases, the optimal intermittent feedback turns into continuous feedback. On the other hand, as S becomes greater, τ approaches τ^{max} . The above choice of Q , R and S yields $\tau \in [0.6\tau^{max}, 0.9\tau^{max}]$.

As the approximation architecture we choose a Multilayer Perceptron (MLP) with 100 hidden neurons. In addition, we confine x to the set $\mathcal{C}_x = [-100, 100]^2 \times [-30\pi, 30\pi]$. Not until we used that many hidden neurons, properties (i), (ii) and (iii) were satisfied on \mathcal{C}_x . Even though activation functions in MLPs are not locally responsive, we were able to satisfy (i), (ii) and (iii). We presume the reason is low dimensionality of the considered tracking problem. For high dimensional problems, the kernel-based NNs appear to be more suitable. In the stopping criterion from Remark 12 we choose $\epsilon = 1$ and $N = 10$, and obtain $\hat{V}^*(x)$ in about 300 to 400 steps depending on the initial $\hat{V}^0(x)$ and ω_p . Afterwards, we obtain $\hat{V}^*(\hat{x})$ from $\hat{V}^*(x)$ using (5.12) and \hat{x} fed from the particle filter. With $\epsilon = 1$ and $N = 10$, it takes about 50 simulations for $\hat{V}^*(\hat{x})$ to converge starting from $\hat{V}^*(x)$. The obtained approximation $\hat{V}^*(\hat{x})$ of $V^*(x)$ for $\omega_p = (1, 1)$ is illustrated in Figure 5.3.

In the simulation included in this chapter, we choose $k_1 = 1.5$, $k_2 = 1.2$ and $k_3 = 1.1$. Figure 5.4 is obtained for the trajectory generated with

$$\omega_p(t) = [1 \ 1]^\top t_{[0,1.83)} + [0.6 \ 0.15]^\top t_{[1.83,8.8)} + [2 \ 2]^\top t_{[8.8,12]},$$

where $t_{\mathcal{I}}$ is the indicator function on an interval \mathcal{I} , i.e., $t_{\mathcal{I}} = t$ when $t \in \mathcal{I}$ and zero otherwise. Stability of the obtained sampling policy can be inferred from 5.4(a) and 5.4(b).

Chapter 5. Optimal Self-Triggering

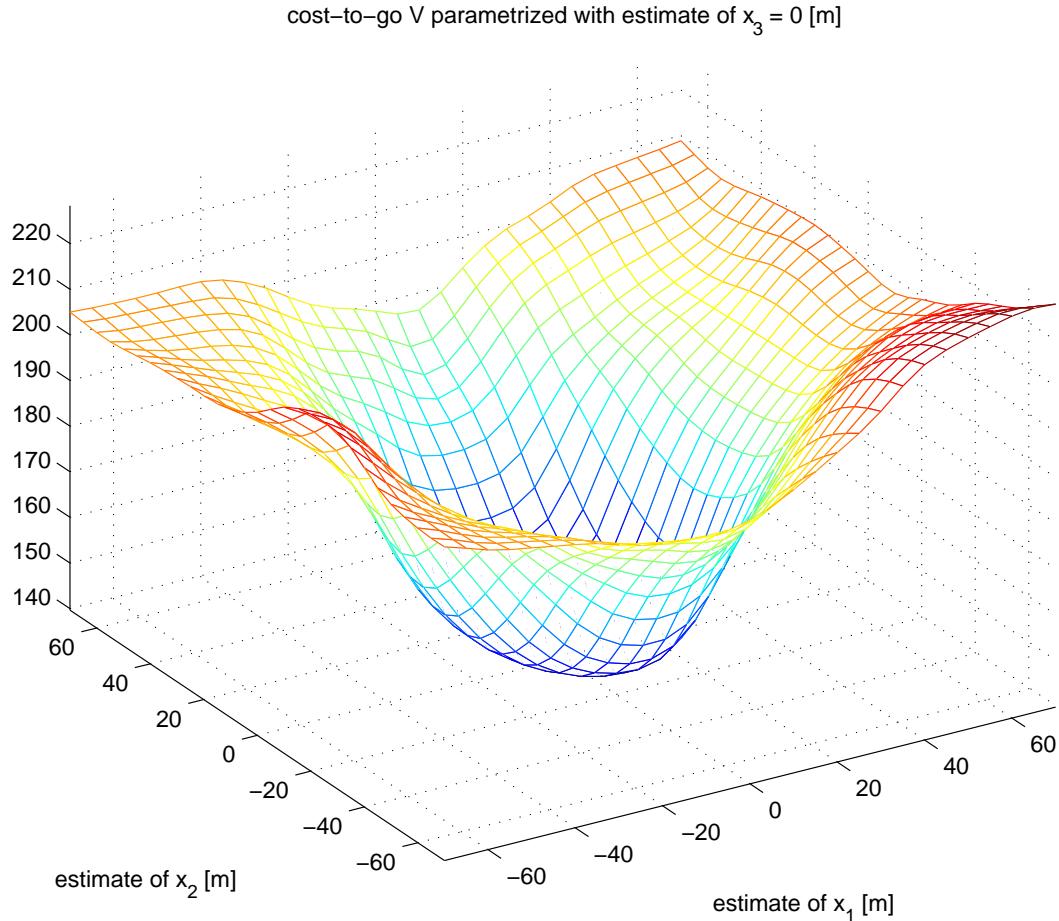


Figure 5.3: An approximation $\hat{V}^*(\hat{x})$ of the optimal value function $V^*(x)$ for $\omega_p = (1, 1)$ depicted as a function of $\hat{x}_1 \in [-70, 70]$ and $\hat{x}_2 \in [-70, 70]$ when $\hat{x}_3 = 0$.

In addition, Figure 5.4 shows that the intersampling interval τ tends to increase as $\|x\|$ approaches the origin. Finally, it should be mentioned that oscillations of x in Figure 5.4(a) are an inherited property of the controller, and not a consequence of intermittent feedback.

5.4 Conclusion

This chapter investigates the problem of optimal input-output triggering for nonlinear systems. We replace the traditional periodic paradigm, where up-to-date information are transmitted and control laws are executed in a periodic fashion, with optimal intermittent feedback. In other words, we develop a methodology that, based on the currently available but outdated measurements of the outputs and external inputs of a plant, provides time instants when to obtain new up-to-date measurements and execute the control law such that a given cost function is minimized. The optimization problem is formulated as a DP problem, and ADP is employed to solve it. In addition, because the investigated problems contain partially observable states, our methodology includes Particle Filtering under intermittent feedback. Furthermore, instead of advocating one approximation architecture over another in ADP, we formulate properties that successful approximation architectures satisfy. Finally, our approach is successfully applied to a trajectory tracking controller for velocity-controlled unicycles.

In the future, the main goal is to further investigate the properties of successful approximation architectures. In addition, we plan to estimate how suboptimal the methodology developed in this chapter is.

Chapter 5. Optimal Self-Triggering

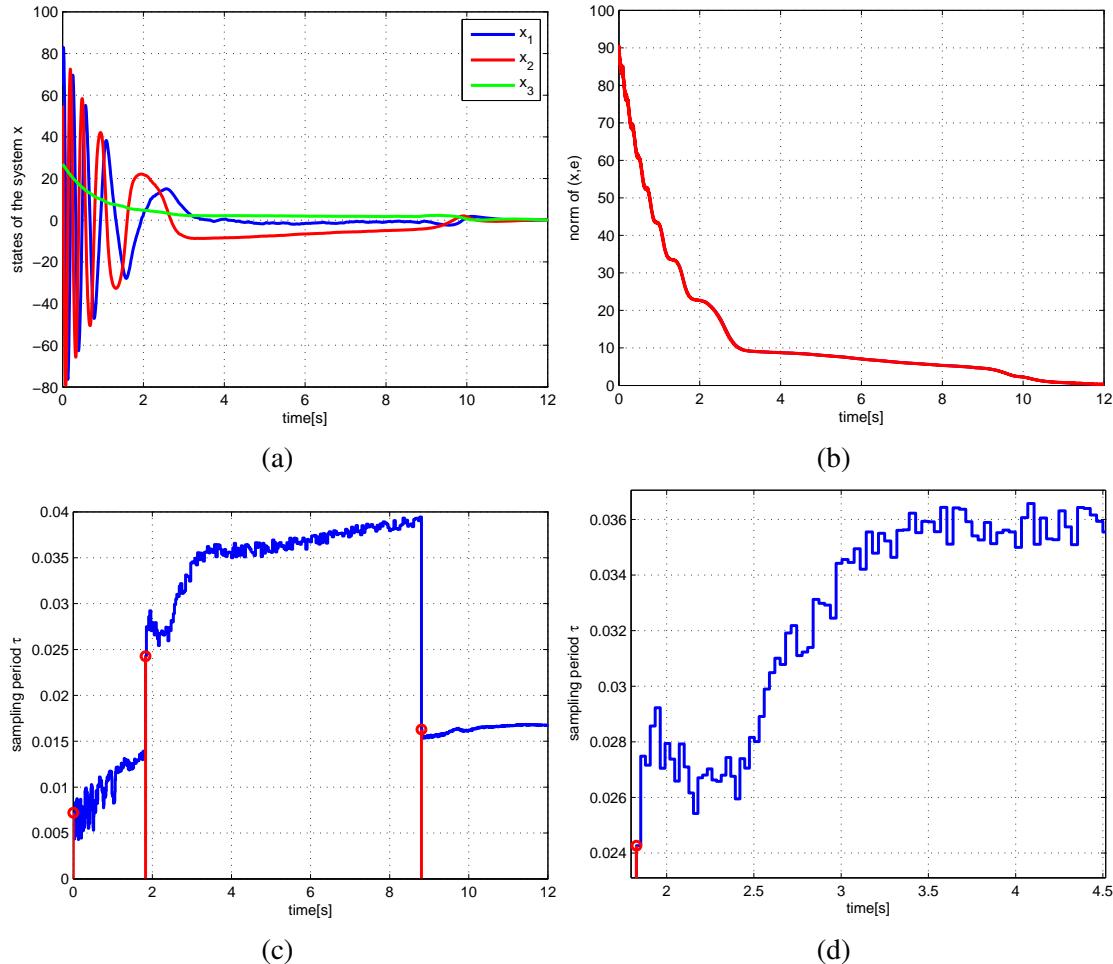


Figure 5.4: An illustration of the optimal input-output triggering: (a) State x of the tracking system; (b) Norm of (x, e) ; (c) Values of the sampling period τ between two consecutive transmissions. Red stems indicate time instants when changes in ω_p happen; and, (d) A detail from Figure 5.4(c).

Chapter 6

Decentralized Output Synchronization

In order to reconcile the *radio network* and *local* model of wireless networks (see subsection 1.2.5), we partition the set of agents in subsets with the following property: *when all agents in a subset broadcast simultaneously, the wireless network is collision free*. Basically, we do not allow agents, that belong to different partitions, to broadcast at the same time due to possibility of message collisions. Consequently, agents in the same partition synchronously broadcast their outputs via wireless. We take advantage of the predictability in synchronous wireless networks to detect possible changes in the communication topology among the agents. When a receiver does not receive a message in an allotted time interval, we say that an event has occurred and a decentralized topology discovery algorithm is triggered.

In order to determine when agents in different partitions should broadcast, we utilize *self-triggered* feedback developed in Chapter 4. Essentially, based on the current topology (captured in the graph Laplacian matrix) and dynamics of the agents, each agent computes when to broadcast its outputs such that output synchronization is achieved. In other words, the communication between agents is neither continuous nor periodic as in [67], [68] and [72], but adapts to changes in the topology. The motivation behind self-triggering is

to reduce communication and computational load without compromising stability. In addition, valid self-triggered broadcasting policies must guarantee that broadcasting instants do not accumulate in finite time which is known as Zeno behavior (see [100], [55] and Remark 15). Consequently, self-triggering eliminates the problem of arbitrary fast switching (refer to [68, Chapter 2], [131] and [100]) since changes in the communication topology between broadcasting instants do not impact stability. See Remark 18 for more details.

The remainder of the chapter is organized as follows. Section 6.1 presents the notation, concepts from graph theory and stability notions utilized in this chapter. In addition, the notion of *average dwell-time* for switched systems is presented. Section 6.2 formulates the problem of decentralized output synchronization with intermittent communication and switching topology. The methodology brought together to solve the problem is presented in Section 6.3. The case of switching topology is investigated in Section 6.4. The proposed methodology is verified using numerical simulations in Section 6.5. Conclusions and future challenges are in Section 6.6.

6.1 Mathematical Preliminaries

6.1.1 Notation

To shorten the notation, we use $(x, y) := [x^\top \quad y^\top]^\top$. The dimension of a vector x is denoted n_x . We use

$$\|f[a, b]\|_p := \left(\int_{[a, b]} \|f(s)\|^p ds \right)^{1/p} \quad (6.1)$$

to denote the \mathcal{L}_p norm of a Lebesgue measurable function f when restricted to the interval $[a, b] \subset \mathbb{R}$. In this chapter, $\|\cdot\|$ refers to the Euclidean norm of a vector. If the argument of $\|\cdot\|$ is a matrix, then it denotes the induced matrix 2-norm. The set of all eigenvalues of a

Chapter 6. Decentralized Output Synchronization

matrix A is denoted $\lambda(A)$. Given $x \in \mathbb{R}^n$, we define

$$\bar{x} = (|x_1|, |x_2|, \dots, |x_n|),$$

where $|\cdot|$ denotes the absolute value function. When the argument of $|\cdot|$ is a set, then it denotes the cardinality of the set. Given $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n$, the partial order \preceq is given as

$$x \preceq y \iff x_i \leq y_i \quad \forall i \in \{1, \dots, n\}.$$

An n -dimensional vector with all entries 0 is denoted $\mathbf{0}_n$. The set \mathcal{A}_n^+ denotes the subset of all $n \times n$ matrices that are symmetric and have nonnegative entries. Finally, let \mathbb{R}_+^n denote the nonnegative orthant.

6.1.2 Graph Theory

A *directed graph*, or digraph, is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, \dots, v_N\}$ is a nonempty set of *nodes* (or vertices) with unique ID numbers, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of the corresponding *edges*. When the edge (i, j) belongs to \mathcal{E} it means that there is an information flow from the node i to the node j . We do not allow *self-loops*, i.e., edges that connect a vertex to itself. When both (i, j) and (j, i) belong to \mathcal{E} , we say that the *link* between i and j is *bidirectional*. Otherwise, the link between i and j is *unidirectional*. The set of *neighbors* of the node v_i is $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ which is all nodes that the node v_i can obtain information from. A *path* in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. The *distance* between two vertices in a graph is the number of edges in a shortest path connecting them. The greatest distance between any pair of vertices is called the *diameter* of a graph and is denoted $\text{diam}(\mathcal{G})$. A *cycle* in \mathcal{G} is a directed path with distinct nodes except for the starting and ending node. An *inclusive cycle* for an edge is a cycle that contains the edge on its path.

Given a graph \mathcal{G} , the graph Laplacian matrix $L \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is defined as

$$L = [l_{ij}], \quad l_{ij} = \begin{cases} -1, & j \in \mathcal{N}_i \\ |\mathcal{N}_i|, & j = i \\ 0, & \text{otherwise} \end{cases}.$$

For more details about algebraic graph theory refer to [132].

6.1.3 Stability Notions

Consider a hybrid (or impulsive) system

$$\Sigma \quad \left\{ \begin{array}{l} \dot{x} = f(x, \omega) \\ y = g(x, \omega) \\ x(t^+) = h(x(t)) \end{array} \right. \quad \begin{array}{l} t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \\ t \in \mathcal{T}, \end{array} \quad (6.2)$$

with the state $x \in \mathbb{R}^{n_x}$, output $y \in \mathbb{R}^{n_y}$ and input $\omega \in \mathbb{R}^{n_\omega}$. We assume enough regularity on f and h to guarantee existence of the solutions given by right-continuous functions $t \mapsto x(t)$ on $[t_0, \infty)$ starting from x_0 at $t = t_0$. Jumps of the state x occur at each $t \in \mathcal{T} := \{t_i : i \in \mathbb{N}\}$. The value of the state after the jump is given by $x(t^+) = \lim_{t' \searrow t} x(t')$ for each $t \in \mathcal{T}$.

Definition 13 (global exponential stability) For $\omega \equiv 0$, the equilibrium point $x = 0$ of Σ is Globally Exponentially Stable (GES) if there exist $k, l \geq 0$ such that $\|x(t)\| \leq k \exp(-l(t - t_0)) \|x(t_0)\|$ for all $t \geq t_0$ and for any $x(t_0)$.

Definition 14 (input-to-state stability) The system Σ is input-to-state stable (ISS) if there exist a class- \mathcal{KL} function β and a class- \mathcal{K}_∞ function γ such that, for any $x(t_0)$ and every input ω , the corresponding solution $x(t)$ satisfies $\|x(t)\| \leq \beta(\|x_0\|, t - t_0) + \gamma(\|\omega[t_0, t]\|_\infty)$.

Definition 15 (uniform bounded-input bounded-output stability) *The system Σ is uniformly bounded-input bounded-output stable if there exists a finite constant η such that, for any t_0 and any input signal $\omega(t)$, the corresponding zero-state response (i.e., $x_0 = \mathbf{0}_{n_x}$) satisfies $\|x[t_0, t]\|_\infty \leq \eta \|\omega[t_0, t]\|_\infty$.*

Definitions 13 and 14 are taken from [56] while Definition 15 is taken from [133].

6.1.4 Switched Systems and Average Dwell-Time

Consider a family of systems (6.2) indexed by the parameter ρ taking values in a set $\mathcal{P} = \{1, 2, \dots, m\}$. Let us define a right-continuous and piecewise constant function $\sigma : [t_0, \infty) \rightarrow \mathcal{P}$ called a *switching signal* [100]. The role of σ is to specify which system is active at any time $t \geq t_0$. The resulting switched system is given by

$$\Sigma_\sigma \quad \left\{ \begin{array}{l} \dot{x} = f_\sigma(x, \omega) \\ y = g(x, \omega) \\ x(t^+) = h_\sigma(x(t)) \end{array} \right. \quad \begin{array}{l} t \in \bigcup_{i \in \mathbb{N}_0} [t_i, t_{i+1}), \\ t \in \mathcal{T}. \end{array} \quad (6.3)$$

For each switching signal σ and each $t \geq t_0$, let $N_\sigma(t, t_0)$ denote the number of discontinuities, called *switching times*, of σ on the open interval (t_0, t) . We say that σ has *average dwell-time* τ_a if there exist two positive numbers N_0 and τ_a such that

$$N_\sigma(t, t_0) \leq N_0 + \frac{t - t_0}{\tau_a} \quad (6.4)$$

for every $t \geq t_0$. For a comprehensive discussion refer to [100] and [131]. In this chapter, different values of σ correspond to different topologies L .

6.2 Problem Statement

Consider N linear systems, i.e., agents, given by

$$\begin{aligned}\dot{x}_i &= A_i x_i + B_i u_i, \\ y_i &= C_i x_i,\end{aligned}\tag{6.5}$$

where $x_i \in \mathbb{R}^{n_{x_i}}$ is the state, $u_i \in \mathbb{R}^{n_{u_i}}$ is the input, $y_i \in \mathbb{R}^{n_y}$ is the output of the i^{th} system, $i \in \{1, 2, \dots, N\}$, and A_i, B_i, C_i are matrices of appropriate dimensions. Since these agents are vertices of a communication graph, the set of all agents is denoted \mathcal{V} . Hence, $|\mathcal{V}| = N$. Motivated by [68, Chapter 2], we consider the following decentralized control policy

$$u_i = -K_i \sum_{j \in \mathcal{N}_i} [(y_j - y_i) - (d_j - d_i)] + \omega_i,\tag{6.6}$$

where K_i is a $n_{u_i} \times n_y$ matrix, \mathcal{N}_i denotes the set of neighbors of the i^{th} system, $d_i \in \mathbb{R}^{n_y}$ is the bias term, and $\omega_i \in \mathbb{R}^{n_y}$ is the disturbance term. Next, we define the following stack vectors $x := (x_1, x_2, \dots, x_N)$, $y := (y_1, y_2, \dots, y_N)$, $d := (d_1, d_2, \dots, d_N)$ and $\omega := (\omega_1, \omega_2, \dots, \omega_N)$. Knowing the Laplacian matrix L of a communication graph \mathcal{G} , the closed-loop dynamic equation of (6.5) given the control law (6.6) becomes

$$\begin{aligned}\dot{x} &= A^{\text{cl}} x - B^{\text{cl}} d + B^{\text{d}} \omega, \\ y &= C^{\text{cl}} x,\end{aligned}\tag{6.7}$$

where

$$A^{\text{cl}} = [A_{ij}^{\text{cl}}], \quad A_{ij}^{\text{cl}} = \begin{cases} A_i - l_{ii} B_i K_i C_i, & i = j \\ -l_{ij} B_i K_i C_j, & \text{otherwise} \end{cases},$$

$$B^{\text{cl}} = [B_{ij}^{\text{cl}}], \quad B_{ij}^{\text{cl}} = -l_{ij} B_i K_i,$$

$$C^{\text{cl}} = \text{diag}(C_1, C_2, \dots, C_N),$$

$$B^{\text{d}} = \text{diag}(B_1, B_2, \dots, B_N).$$

Chapter 6. Decentralized Output Synchronization

In the above expressions, A_{ij}^{cl} and B_{ij}^{cl} are matrix blocks while $\text{diag}(\cdot, \cdot, \dots, \cdot)$ denotes the block-diagonal matrix.

Assumption 6 A^{cl} is Hurwitz.

Remark 13 Using the Geršgorin circle theorem, the work in [69] provides sufficient conditions for A^{cl} to be Hurwitz. Applying these sufficient conditions to A^{cl} herein, we obtain that when

(i) $A_i - l_{ii}B_iK_iC_i$ is Hurwitz for all $i \in \{1, 2, \dots, N\}$, and

(ii) $\min_{\lambda_i \in \lambda\{A_i - l_{ii}B_iK_iC_i\}} \{|\lambda_i|\} \geq \sum_{j \in \mathcal{N}_i} \|B_iK_iC_j\|$,

are fulfilled, the matrix A^{cl} is Hurwitz. Thus, by changing K_i 's for different topologies, one can ensure that Assumption 6 is satisfied.

When $\omega \equiv \mathbf{0}_{n_y}$, the equilibrium of (6.7) is given by

$$x_{eq} = (A^{\text{cl}})^{-1}B^{\text{cl}}d \quad (6.8)$$

with the corresponding output

$$y_{eq} = C^{\text{cl}}x_{eq}. \quad (6.9)$$

Definition 16 Suppose we have a system of N agents given by (6.5). We say that the agents output synchronize if $\|y - y_{eq}\| \rightarrow 0$ as $t \rightarrow \infty$.

Remark 14 The above definition of output synchronization differs from the definition found in, for instance, [43] where it is required that $\|y_i - y_j\| \rightarrow 0$ as $t \rightarrow \infty$ for all $i, j \in \{1, 2, \dots, N\}$. Our definition aims at controlling the asymptotic values y_{eq} of the outputs y regardless of initial conditions. Problems in which the asymptotic values of y

Chapter 6. Decentralized Output Synchronization

depend on initial conditions will be considered in the future work. These problems are characterized by matrices A^{cl} that have an eigenvalue at the origin of the complex plane. In our definition, notice that one can change y_{eq} by changing d . For example, one can change formations by changing d .

It is well known that substitutions $x' = x - x_{eq}$ and $y' = y - y_{eq}$ transform (6.7) into the equivalent system

$$\begin{aligned}\dot{x}' &= A^{\text{cl}}x' + B^{\text{d}}\omega, \\ y' &= C^{\text{cl}}x',\end{aligned}\tag{6.10}$$

such that $x'_{eq} = 0$ is the equilibrium point when $\omega \equiv \mathbf{0}_{n_y}$. From Assumption 6 and [56, Corollary 5.2.], we infer that the closed-loop system (6.10) is \mathcal{L}_p -stable from ω to y' for each $p \in [1, \infty]$.

Since we do not consider continuous feedback in (6.6), the control signal becomes

$$u_i = -K_i \sum_{j \in \mathcal{N}_i} [(\hat{y}_i - \hat{y}_j) - (d_i - d_j)] + \omega_i,\tag{6.11}$$

where \hat{y}_j is the most recently transmitted value of the output of the j^{th} agent. Let $\mathcal{T}_i := \{t_i^j : j \in \mathbb{N}\}$ denote the set of broadcasting time instants of the i^{th} agent and $\mathcal{T} := \cup_{i=1}^N \mathcal{T}_i$. In order to account for the fact that outdated \hat{y}_i 's are used in control law (6.6) and not the actual outputs y_i 's, we introduce the output error vector

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} := \begin{bmatrix} \hat{y}_1 - y_1 \\ \hat{y}_2 - y_2 \\ \vdots \\ \hat{y}_N - y_N \end{bmatrix} = \hat{y} - y.\tag{6.12}$$

The above expression uses $\hat{y} := (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N)$. Taking e into account, the closed-loop dynamics (6.10) become

$$\dot{x}' = A^{\text{cl}}x' + B^{\text{cl}}e + B^{\text{d}}\omega.\tag{6.13}$$

Since $\dot{\hat{y}} = 0$ and $\dot{y}_{eq} = 0$, the corresponding output error dynamics are

$$\dot{e} = -C^{\text{cl}}\dot{x}'. \quad (6.14)$$

The following three problems are solved in the remainder of this chapter.

Problem 5 *Partition the set of agents \mathcal{V} into subsets \mathcal{P}_i 's with the following property: when all agents in each \mathcal{P}_i broadcast simultaneously, message collisions are avoided.*

Problem 6 *Given a fixed topology, design sets of broadcasting instants \mathcal{T}_i , where $i \in \{1, 2, \dots, N\}$, such that the outputs of agents synchronize in the sense of Definition 16.*

Problem 7 *Find conditions that preserve output synchronization under switching communication topology.*

6.3 Methodology

6.3.1 Decentralized Topology Discovery for Directed Graphs

In order to solve Problem 5, each agent has to know the communication topology, i.e., the graph Laplacian matrix L . This problem is known as topology discovery and has been an active area of research (e.g., [79], [80], [81] and [82]). In this chapter we implement the approach from [79] due to its applicability to directed graphs. The approach in [79] utilizes the flooding algorithm to propagate information about the graph topology. This approach converges in finite time Δ after the network topology stops changing. Δ is proportional to $\text{diam}(\mathcal{G})$. Since we consider a finite number of agents N , there exists an upper bound on Δ , denoted Δ_u , for all admissible topologies. The admissible topologies are those that satisfy Assumptions 6 and 7.

Assumption 7 All unidirectional links have an inclusive cycle.

Assumption 7 is the main assumption in [79]. In order to simplify the exposition of the chapter, we neglect Δ_u , i.e., we take $\Delta_u = 0$. In case $\Delta_u \neq 0$, one should take into account Δ_u when designing \mathcal{T}_i 's in Problem 6. Basically, broadcasting instants in \mathcal{T} should leave enough time for topology discovery to be completed without compromising output synchronization. In addition, by assuming $\Delta_u = 0$, the agents have the access to the up-to-date topology when designing $\mathcal{T}_i, i \in \{1, \dots, N\}$.

A remark is in order. Notice that leader-follower topologies do not satisfy Assumption 7. In case of fixed communication topologies, Assumption 7 can be omitted as this assumption is needed only for decentralized topology discovery. Thus, the work presented herein is applicable to time-invariant leader-follower topologies as well.

6.3.2 Partitioning the Agents

After obtaining L using the approach from [79], we have to partition the set \mathcal{V} in order to avoid message collisions according to the *radio network model* described in Introduction. In other words, we want to allow simultaneous broadcast of agents that have no common receivers and are not receivers themselves at that particular time instants. Notice that, if one is not concerned with message collisions, then there is no need to partition the agents.

Consider the graph depicted in Figure 6.1. The corresponding graph Laplacian matrix is

$$L_1 = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & -1 & 0 & 2 & 0 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}. \quad (6.15)$$

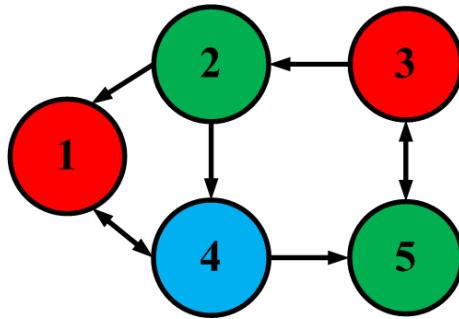


Figure 6.1: The graph partition $\mathcal{P}_1 = \{1, 3\}$, $\mathcal{P}_2 = \{2, 5\}$ and $\mathcal{P}_3 = \{4\}$ obtained via Algorithm 1. Nodes belonging to different partitions are colored differently.

Notice that this graph satisfies Assumption 7. Next, we partition $\mathcal{V} = \{1, 2, 3, 4, 5\}$ in Figure 6.1 using Algorithm 1. In Algorithm 1, the element-wise product of the i^{th} and the j^{th} column of L is denoted $L(i) \cdot L(j)$ while the N^{th} -dimensional vector of all zeros is denoted $\mathbf{0}_N$. The input to the algorithm is L , and the outputs are subsets \mathcal{P}_i . The number of nonempty \mathcal{P}_i 's is $T \leq N$, and we prune empty \mathcal{P}_i 's.

At this point, Problem 5 is solved. In the following subsection, we utilize L , \mathcal{P}_i , $i \in \{1, \dots, T\}$, and expressions (6.13) and (6.14) to solve Problem 6.

6.3.3 Designing Broadcasting Instants

In order to solve Problem 6, we use the extensions of [40] found in Chapter 4. In what follows, we provide only the scheduling protocol considered herein and adapt results from Chapter 4 to this specific protocol. In other words, even though the framework presented in this subsection is applicable to the larger group of *uniformly persistently exciting scheduling protocols* [40], we do not pursue that direction in this chapter.

Let us consider the following scheduling protocol:

Protocol 1 *The agents from $\mathcal{P}_{[(i+1) \bmod N]+1}$ broadcast their outputs τ seconds after the*

Algorithm 1 Algorithm developed for graph partitioning. Taking L as the input, the algorithm outputs $\{\mathcal{P}_1, \dots, \mathcal{P}_N\}$.

```

1:  $\mathcal{P}_i \leftarrow \{\emptyset\}$  for all  $i \in \{1, \dots, N\}$ ;  $k \leftarrow 0$ 
2: for  $i = 1$  to  $N$  do
3:   if  $i \notin \mathcal{P}_m$  for every  $m \in \{1, \dots, N\}$  then
4:      $k \leftarrow k + 1$ 
5:      $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup \{i\}$ 
6:     for  $j = i + 1$  to  $N$  do
7:       if  $L(i).L(j) = \mathbf{0}_N$  for all  $i \in \mathcal{P}_k$  then
8:          $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup \{j\}$ 
9:       end if
10:      end for
11:    end if
12:  end for

```

agents from $\mathcal{P}_{[i \bmod N]+1}$ have broadcast their outputs, where \bmod is the module operator.

Protocol 1 is a variant of round-robin scheduling [134]. Notice that elements of the set $\mathcal{T}_{[(i+1) \bmod N]+1}$ are equal to elements $\mathcal{T}_{[i \bmod N]+1}$ increased by τ . The impact of broadcasting agents' outputs is as follows:

Property 1 *If the i^{th} agent broadcasts at time t , the corresponding components of e reset to zero while other components remain unchanged, i.e.,*

$$\left. \begin{aligned} e_{(i-1)n_y+1}^+(t) &= \dots = e_{in_y}^+(t) = 0, \\ e_j^+(t) &= e_j(t), \end{aligned} \right\} \quad (6.16)$$

for all $j \in \{1, \dots, Nn_y\} \setminus \{(i-1)n_y + 1, \dots, in_y\}$, where the set difference is denoted \setminus .

Now, let us interconnect dynamics (6.13) and (6.14) and employ the small gain theorem

Chapter 6. Decentralized Output Synchronization

[56]. To this end, we upper bound the output error dynamics (6.14) as follows:

$$\bar{e} = \overline{-C^{\text{cl}}(A^{\text{cl}}x' + B^{\text{cl}}e + B^{\text{d}}\omega)} \preceq A^*\bar{e} + \tilde{y}(x', \omega), \quad (6.17)$$

where

$$A^* = [a_{ij}^*] := \max\{|c_{ij}^*|, |c_{ji}^*|\}, \quad (6.18)$$

$$\tilde{y}(x, \omega) := \overline{C^{\text{cl}}(A^{\text{cl}}x + B^{\text{d}}\omega)}. \quad (6.19)$$

In (6.18), we use $-C^{\text{cl}}B^{\text{cl}} = [c_{ij}^*]$. Notice that $A^* \in \mathcal{A}_{n_e}^+$ and $\tilde{y} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\omega} \rightarrow \mathbb{R}_+^{n_e}$ is a continuous function. With this choice of A^* and \tilde{y} , the upper bound (6.17) holds for all $(x', e, \omega) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_e} \times \mathbb{R}^{n_\omega}$ and all $t \in \mathbb{R}$. The above exposition brings us to the following theorem.

Theorem 8 Suppose that Protocol 1 is implemented. In addition, suppose that $\tau \in (0, \tau^*)$, where $\tau^* := \frac{\ln(2)}{\|A^*\|T}$. Then, the output error system (6.14) is \mathcal{L}_p -stable from \tilde{y} to e for any $p \in [1, \infty]$ with gain

$$\gamma_e = \frac{T \exp(\|A^*\|(T-1)\tau)(\exp(\|A^*\|\tau) - 1)}{\|A^*\|(2 - \exp(\|A^*\|T\tau))}. \quad (6.20)$$

Proof 14 This result is obtained by following the proof of [40, Theorem 5.1]. However, the assumptions of [40, Theorem 5.1] require A^* both to be positive semidefinite and to belong to $\mathcal{A}_{n_e}^+$. Positive definiteness of A^* together with the requirement that $A^* \in \mathcal{A}_{n_e}^+$ is needed in [40] to establish equality $\|\exp(A^*)\| = \exp(\|A^*\|)$. Using Lemma 1 and 2, we know that equality $\|\exp(A^*)\| = \exp(\|A^*\|)$ holds for matrices in $\mathcal{A}_{n_e}^+$. In other words, the requirement on A^* to be positive semidefinite is redundant. This concludes the proof.

Next, take (e, ω) to be the input and \tilde{y} , obtained in Theorem 8, to be the output of the dynamics (6.13). For given $p \in [1, \infty]$, the corresponding \mathcal{L}_p -gain from (e, ω) to \tilde{y} is

Chapter 6. Decentralized Output Synchronization

denoted γ . Hence, systems (6.13) and (6.14) are interconnected according to Figure 6.2. We point out that \tilde{y} is an auxiliary signal used to interconnect (6.13) and (6.14), but does not exist physically. According to the small gain theorem, the open loop gain $\gamma\gamma_e$ must be strictly less than one in order for this interconnection to be \mathcal{L}_p -stable from ω to (e, \tilde{y}) .

Theorem 9 *Suppose that Protocol 1 is implemented. If the interbroadcasting interval τ in (6.20) is such that $\gamma\gamma_e < 1$, then the interconnection in Figure 6.2 is \mathcal{L}_p -stable from ω to (e, \tilde{y}) for given $p \in [1, \infty]$.*

Proof 15 *From Assumption 6 and [56, Corollary 5.2.], we infer that (6.13) is \mathcal{L}_p -stable from (e, ω) to \tilde{y} with gain γ and (6.14) is \mathcal{L}_p -stable from \tilde{y} to e with gain γ_e given by (6.20). In other words, we have*

$$\|\tilde{y}[t_0, t]\|_p \leq K\|x'_0\| + \gamma\|(e, \omega)[t_0, t]\|_p,$$

$$\|e[t_0, t]\|_p \leq K_e\|e_0\| + \gamma_e\|\tilde{y}[t_0, t]\|_p.$$

After manipulating the above inequalities we obtain

$$\|\tilde{y}[t_0, t]\|_p \leq \frac{1}{1 - \gamma\gamma_e} [K\|x'_0\| + \gamma K_e\|e_0\| + \gamma\|\omega[t_0, t]\|_p], \quad (6.21)$$

$$\|e[t_0, t]\|_p \leq \frac{1}{1 - \gamma\gamma_e} [\gamma_e K\|x'_0\| + K_e\|e_0\| + \gamma\gamma_e\|\omega[t_0, t]\|_p]. \quad (6.22)$$

Using the fact that $\|(\tilde{y}, e)[t_0, t]\|_p \leq \|\tilde{y}[t_0, t]\|_p + \|e[t_0, t]\|_p$, we obtain

$$\begin{aligned} \|(e, \tilde{y})[t_0, t]\|_p &\leq \frac{1}{1 - \gamma\gamma_e} [(K + \gamma_e K)\|x'_0\| + \\ &\quad + (K_e + \gamma K_e)\|e_0\| + (\gamma + \gamma\gamma_e)\|\omega[t_0, t]\|_p], \end{aligned}$$

which proves the claim.

Remark 15 *Notice that $\gamma_e(\tau)$ in (6.20) is a monotonically increasing function of $\tau \in [0, \tau^*)$. In addition, notice that $\gamma_e(0) = 0$. By the assumption of Theorem 9, we know*

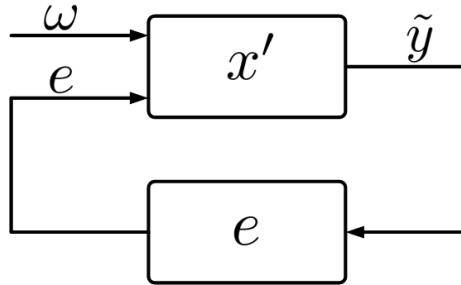


Figure 6.2: Interconnection of the nominal and the output error dynamics.

that $\gamma < \infty$. Since our goal it to design τ such that $\gamma\gamma_e(\tau) < 1$, we first find τ' such that $\gamma\gamma_e(\tau') = 1$, and then compute $\tau = \kappa\tau'$, where $\kappa \in (0, 1)$. Due to monotonicity of $\gamma_e(\tau)$, the obtained τ' is strictly positive; hence, $\tau = \kappa\tau'$ is strictly positive. Consequently, the unwanted Zeno behavior [55] is avoided. In other words, our approach does not yield continuous feedback that is impossible to achieve with digital technology. Since we are interested in obtaining the interbroadcasting interval τ as large as possible, we choose κ as great as possible (e.g., $\kappa = 0.999$).

Remark 16 Let us consider the case of lossy communication channels. If there is an upper bound on the maximum number of successive dropouts in the wireless network, say $N^D \in \mathbb{N}$, simply use τ/N^D as the interbroadcasting interval in order for Theorem 9 to hold.

Remark 17 Notice that $\|\bar{v}\| = \|v\|$ for any $v \in \mathbb{R}^{n_v}$. From (6.1) we infer that $\|\bar{v}[t_0, t]\|_p = \|v[t_0, t]\|_p$ for any $t \geq t_0$ and any $p \in [1, \infty]$, where $v(t) : \mathbb{R} \rightarrow \mathbb{R}^{n_v}$. Therefore, the \mathcal{L}_p -gain from (e, ω) to $C^{\text{cl}}(A^{\text{cl}}x + B^{\text{d}}\omega)$ equals the \mathcal{L}_p -gain from (e, ω) to \tilde{y} given by (6.19).

From Assumption 6 and [56, Corollary 5.2.], we infer that for (6.13) there exist $K_d \geq 0$ and $\gamma_d \geq 0$ such that $\|x'[t_0, t]\|_p \leq K_d\|x'_0\| + \gamma_d\|(e, \omega)[t_0, t]\|_p$ for any $p \in [1, \infty]$. Consequently, x' is \mathcal{L}_p to \mathcal{L}_p detectable from (e, ω, \tilde{y}) for any $p \in [1, \infty]$ (see inequality (6.23)).

Chapter 6. Decentralized Output Synchronization

Corollary 2 Assume that the conditions of Theorem 9 are met. Then, output synchronization of systems given by (6.5) is \mathcal{L}_p -stable from ω to (e, x') for given $p \in [1, \infty]$.

Proof 16 Detectability of x' from (e, ω, \tilde{y}) means

$$\|x'[t_0, t]\|_p \leq K_d \|x'_0\| + \gamma_d \|\tilde{y}[t_0, t]\|_p + \gamma_d \|(e, \omega)[t_0, t]\|_p. \quad (6.23)$$

Combining (6.21) and (6.23) yields

$$\begin{aligned} \|x'[t_0, t]\|_p &\leq (K_d + \frac{\gamma_d K}{1 - \gamma \gamma_e}) \|x'_0\| + \frac{\gamma_d \gamma K_e}{1 - \gamma \gamma_e} \|e_0\| + \\ &+ \gamma_d \|e[t_0, t]\|_p + (\gamma_d + \frac{\gamma d \gamma}{1 - \gamma \gamma_e}) \|\omega[t_0, t]\|_p. \end{aligned} \quad (6.24)$$

Combining (6.22) and (6.24), we obtain

$$\begin{aligned} \|x'[t_0, t]\|_p &\leq (K_d + \frac{\gamma_d K + \gamma_d \gamma_e K}{1 - \gamma \gamma_e}) \|x'_0\| + \\ &+ \frac{\gamma_d \gamma K_e + \gamma_d K_e}{1 - \gamma \gamma_e} \|e_0\| + (\gamma_d + \frac{\gamma_d \gamma + \gamma_d \gamma \gamma_e}{1 - \gamma \gamma_e}) \|\omega[t_0, t]\|_p. \end{aligned} \quad (6.25)$$

Using the fact that $\|(x', e)[t_0, t]\|_p \leq \|x'[t_0, t]\|_p + \|e[t_0, t]\|_p$, from (6.22) and (6.25) we obtain

$$\begin{aligned} \|(e, x')[t_0, t]\|_p &\leq (K_d + \frac{\gamma_d K + \gamma_d \gamma_e K + \gamma_e K}{1 - \gamma \gamma_e}) \|x'_0\| + \\ &+ \frac{\gamma_d \gamma K_e + \gamma_d K_e + K_e}{1 - \gamma \gamma_e} \|e_0\| + \frac{\gamma_d + \gamma_d \gamma + \gamma \gamma_e}{1 - \gamma \gamma_e} \|\omega[t_0, t]\|_p, \end{aligned} \quad (6.26)$$

which proves the claim.

Corollary 3 Assume that the conditions of Theorem 9 are met. Then, output synchronization of systems given by (6.5) is ISS from ω to (e, x') .

Proof 17 For $p \in [1, \infty)$ let us define

$$\begin{aligned} K_1 &:= \max \left\{ K_d + \frac{\gamma_d K + \gamma_d \gamma_e K + \gamma_e K}{1 - \gamma \gamma_e}, \frac{\gamma_d \gamma K_e + \gamma_d K_e + K_e}{1 - \gamma \gamma_e} \right\}, \\ \gamma_1 &:= \frac{\gamma_d + \gamma_d \gamma + \gamma \gamma_e}{1 - \gamma \gamma_e}. \end{aligned}$$

Chapter 6. Decentralized Output Synchronization

Recall that all norms on finite dimensional vector spaces are equivalent. In what follows, we use the following inequalities: $\sqrt{\sum_{i=1}^{n_w} |w_i|^2} \leq \sum_{i=1}^{n_w} |w_i| \leq \sqrt{n_w} \sqrt{\sum_{i=1}^{n_w} |w_i|^2}$ where $w = (w_1, \dots, w_{n_w}) \in \mathbb{R}^{n_w}$. Now, inequality (6.26) can be written as

$$\|(e, x')[t_0, t]\|_p \leq K_1 \sqrt{n_e + n_x} \|(e_0, x'_0)\| + \gamma_1 \|\omega[t_0, t]\|_p.$$

Now, raise both sides of the above inequality to the p^{th} power. Using the following inequality

$$(a + b)^p \leq (1 + \epsilon)^{p-1} a^p + \left(1 + \frac{1}{\epsilon}\right)^{p-1} b^p,$$

where $a, b \geq 0$ and $\epsilon > 0$, yields

$$\begin{aligned} \|(e, x')[t_0, t]\|_p^p &\leq (1 + \epsilon)^{p-1} K_1^p (n_e + n_x)^{p/2} \|(e_0, x'_0)\|^p + \\ &\quad + \left(1 + \frac{1}{\epsilon}\right)^{p-1} \gamma_1^p \|\omega[t_0, t]\|_p^p. \end{aligned}$$

Introducing $K_2 = (1 + \epsilon)^{p-1} K_1^p (n_e + n_x)^{p/2}$ and $\gamma_2 = \left(1 + \frac{1}{\epsilon}\right)^{p-1} \gamma_1^p$ yields

$$\int_{t_0}^t \|(e(s), x'(s))\|^p ds \leq K_2 \|(e_0, x'_0)\|^p + \gamma_2 \int_{t_0}^t \|\omega(s)\|^p ds.$$

The above inequality has the form of inequality (2) in [135] which is equivalent to ISS.

When $p = \infty$, follow the line of reasoning in Subsection 6.4.2. This proves the claim.

6.4 Stability under Switching Topology

The switched system with impulsive effects we are considering herein is

$$\begin{aligned} \begin{bmatrix} \dot{x}' \\ \dot{e} \end{bmatrix} &= \begin{bmatrix} A_\sigma^{cl} & B_\sigma^{cl} \\ -C^{cl} A_\sigma^{cl} & -C^{cl} B_\sigma^{cl} \end{bmatrix} \begin{bmatrix} x' \\ e \end{bmatrix} + \underbrace{\begin{bmatrix} B^d \\ -C^{cl} B^d \end{bmatrix}}_{B_\omega} \omega, \quad t \notin \mathcal{T}, \\ \left. \begin{aligned} x'^+(t) &= x'(t) \\ e^+(t) &= \Gamma_i e(t) \end{aligned} \right\} &t \in \mathcal{T}, \end{aligned} \tag{6.27}$$

where matrix Γ_i implements Property 1 at the i^{th} broadcasting instant. To shorten the notation, we use $z := (x', e)$.

6.4.1 Switching without Disturbances

After setting $\omega \equiv \mathbf{0}_{n_y}$ in (6.27), we obtain the following result:

Theorem 10 Suppose that the conditions of Corollary 2 hold and $\omega \equiv \mathbf{0}_{n_y}$. In addition, assume that L is fixed. Then, the equilibrium point $(e, x') = 0$ of the closed-loop system (6.13) and (6.14) is GES.

Proof 18 Let us show that (6.13) and (6.14) satisfy the assumptions of [40, Theorem 2.5]. In other words, we show that there exist nonnegative constants L_1, L_2, L_3 and L_4 such that

$$\left. \begin{array}{l} \|A^{\text{cl}}x' + B^{\text{cl}}e\| \leq L_1(\|x'\| + \|e\|) \\ \|C^{\text{cl}}A^{\text{cl}}x' + C^{\text{cl}}B^{\text{cl}}e\| \leq L_2(\|x'\| + \|e\|) \\ \|x'^+(t)\| \leq L_3\|x'(t)\| \\ \|e^+(t)\| \leq L_4\|e(t)\| \end{array} \right\} \quad (6.28)$$

for all $x' \in \mathbb{R}^{n_x}$ and all $e \in \mathbb{R}^{n_e}$. Notice that x' does not experience jumps when new information arrives; hence, we can take $L_3 = 1$. From (6.16) it follows that the last inequality is satisfied with $L_4 = 1$. Finally, it is straightforward to show that $L_1 = \max\{\|A^{\text{cl}}\|, \|B^{\text{cl}}\|\}$ and $L_2 = \max\{\|C^{\text{cl}}A^{\text{cl}}\|, \|C^{\text{cl}}B^{\text{cl}}\|\}$ satisfy above inequalities. The GES property follows from [40, Theorem 2.5].

Next, notice that the equilibrium point x_{eq} given by (6.8) is a function of L . In other words, different communication topologies result in different x_{eq} , i.e., different x' . In order to apply results from [131] and [100], x_{eq} must be the same for all admissible topologies. This can be achieved by adapting d in (6.8) such that x_{eq} is constant as the topology changes

Chapter 6. Decentralized Output Synchronization

or one can simply use $d = \mathbf{0}_{n_d}$ yielding $x_{eq} = \mathbf{0}_{n_x}$. For the sake of simplicity, we use $d = \mathbf{0}_{n_y}$ herein. Consequently, $x' = x$ holds so we use x instead of x' in the remainder of the chapter.

For a finite number of agents N , there can be at most 2^{N^2-N} different topologies as self-loops are not allowed. Hence, our switched system consists of a finite number of subsystems. The index set of these subsystems is $\mathcal{P} = \{1, 2, \dots, m\}$, where $m \leq 2^{N^2-N}$ is the number of admissible topologies.

According to Theorem 10, each subsystem in \mathcal{P} is GES. Let us now apply [136, Theorem 15.3.] to each subsystem in \mathcal{P} . From (6.28) we infer that the flow and jump maps are Lipschitz continuous and are zero at zero. In addition, jump times t_i 's are predefined (i.e., time-triggered and do not depend on the actual solution of the system), and such that $0 < t_1 < t_2 < \dots < t_i$ and $\lim_{i \rightarrow \infty} t_i = \infty$ hold. Consequently, all conditions of [136, Theorem 15.3.] are met. From [136, Theorem 15.3.], we know that there exist functions $V_\rho : \mathbb{R} \times \mathbb{R}^{n_x+n_e} \rightarrow \mathbb{R}$, $\rho \in \mathcal{P}$, that are right-continuous in t and Lipschitz continuous in z , and satisfy the following inequalities

$$c_{1,\rho} \|z\|^2 \leq V_\rho(t, z) \leq c_{2,\rho} \|z\|^2, \quad t \geq t_0, \quad (6.29)$$

$$D_\rho^+ V_\rho(t, z) \leq -c_{3,\rho} \|z\|^2, \quad t \notin \mathcal{T}, \quad (6.30)$$

$$V_\rho(t^+, z^+) \leq V_\rho(t, z), \quad t \in \mathcal{T}, \quad (6.31)$$

for all $z \in \mathbb{R}^{n_x+n_e}$, where $c_{1,\rho}$, $c_{2,\rho}$ and $c_{3,\rho}$ are positive constants. These constants are readily obtained once k and l from Definition 9 are known (see the proof of [136, Theorem 15.3.]). In the above inequalities, $D_\rho^+ V_\rho(t, z)$ denotes the upper right derivative of function V_ρ with respect to the solutions of the ρ^{th} system. The upper right derivative of V_ρ is given by

$$D_\rho^+ V_\rho(t, z) := \limsup_{h \rightarrow 0, h > 0} \left(\frac{1}{h} [V_\rho(t + h, z(t + h)) - V_\rho(t, z(t))] \right),$$

Chapter 6. Decentralized Output Synchronization

where $z(t)$, $t \geq t_0$, denotes the trajectory of the ρ^{th} system. We now rewrite (6.29) and (6.30) as follows

$$c_1 \|z\|^2 \leq V_\rho(t, z) \leq c_2 \|z\|^2, \quad t \geq t_0, \quad (6.32)$$

$$D_\rho^+ V_\rho(t, z) \leq -2\lambda_0 V_\rho(t, z), \quad t \notin \mathcal{T}, \quad (6.33)$$

$$V_\rho(t, z) \leq \mu V_\varrho(t, z), \quad t \geq t_0, \quad (6.34)$$

for all $z \in \mathbb{R}^{n_x+n_e}$ and all $\rho, \varrho \in \mathcal{P}$, where

$$\begin{aligned} c_1 &= \min_{\rho \in \mathcal{P}} c_{1,\rho} > 0, & c_2 &= \max_{\rho \in \mathcal{P}} c_{2,\rho} > 0, \\ \lambda_0 &= \min_{\rho \in \mathcal{P}} \frac{c_{3,\rho}}{2c_{1,\rho}} > 0, & \mu &= \max_{\rho, \varrho \in \mathcal{P}} \frac{c_{2,\rho}}{c_{1,\varrho}} > 0. \end{aligned}$$

Notice that $\mu > 1$ in the view of interchangeability of ρ and ϱ in (6.34). Following ideas from [100] and [131], we obtain the following result:

Theorem 11 *Consider the family of m systems for which (6.31), (6.32), (6.33) and (6.34) hold. Then the corresponding switched system (6.27) is GES uniformly in t_0 for every switching signal σ with average dwell-time*

$$\tau_a > \frac{\ln \mu}{2\lambda_0} \quad (6.35)$$

and N_0 arbitrary.

Proof 19 *This proof follows the proof of [100, Theorem 3.2]. Pick an arbitrary $T > 0$, let $t_0 := 0$, and denote the switching times on the interval $(0, T)$ by $t_1, \dots, t_{N_\sigma(T,0)}$. Consider the function $W(t) := \exp(2\lambda_0 t) V_{\sigma(t)}(t, z(t))$. On each interval $[t_j, t_{j+1})$ we have*

$$D_{\sigma(t_j)}^+ W = 2\lambda_0 W + \exp(2\lambda_0 t) D_{\sigma(t_j)}^+ V_{\sigma(t_j)}(t, z) \leq 0$$

due to (6.33). Due to (6.31), when state jumps occur we have $V_{\sigma(t_j)}(t^+, z^+) \leq V_{\sigma(t_j)}(t, z)$ no matter whether the jump times coincide with the switching times or not. Therefore, W

Chapter 6. Decentralized Output Synchronization

is nonincreasing between two switching times. This together with (6.31) and (6.34) yields

$$\begin{aligned} W(t_{j+1}) &= \exp(2\lambda_0 t_{j+1}) V_{\sigma(t_{j+1})}(t_{j+1}, z(t_{j+1})) \leq \\ &\leq \mu \exp(2\lambda_0 t_{j+1}) V_{\sigma(t_j)}(t_{j+1}, z(t_{j+1})) \leq \\ &\leq \mu \exp(2\lambda_0 t_{j+1}^-) V_{\sigma(t_j)}(t_{j+1}^-, z^-(t_{j+1})) = \mu W(t_{j+1}^-) \leq \mu W(t_j). \end{aligned}$$

In the above expressions, the time instant just before t is denoted t^- . When the functions of interest are continuous in t , then $t^- = t$, but we still write t^- instead of t for clarity. In addition, the left limit of a solution $z(t)$ at instant t is denoted $z^-(t)$. Iterating the last inequality from $j = 0$ to $j = N_\sigma(T, 0) - 1$, we obtain

$$W(T^-) \leq W(t_{N_\sigma(T, 0)}) \leq \mu^{N_\sigma(T, 0)} W(0).$$

Using the definition of W , the above inequality and (6.31) we have

$$\begin{aligned} \exp(2\lambda_0 T) V_{\sigma(T^-)}(T, z(T)) &\leq \exp(2\lambda_0 T^-) V_{\sigma(T^-)}(T^-, z(T^-)) \leq \\ &\leq \mu^{N_\sigma(T, 0)} V_{\sigma(0)}(0, z(0)). \end{aligned}$$

Now suppose that σ has the average dwell-time property (6.4). Hence, we can write

$$\begin{aligned} V_{\sigma(T^-)}(T, z(T)) &\leq \exp(-2\lambda_0 T + (N_0 + \frac{T}{\tau_a} \ln \mu) V_{\sigma(0)}(0, z(0)) = \\ &= \exp(N_0 \ln \mu) \exp((\frac{\ln \mu}{\tau_a} - 2\lambda) T) V_{\sigma(0)}(0, z(0)). \end{aligned}$$

From the above inequality, we infer that if τ_a satisfies (6.35), then $V_{\sigma(T^-)}(T, z(T))$ converges to zero exponentially as $T \rightarrow \infty$, i.e., it is upper-bounded by $\mu^{N_0} \exp(-2\lambda T) V_{\sigma(0)}(0, z(0))$ for some $\lambda \in (0, \lambda_0)$. From (6.32) we obtain $\|z(T)\| \leq \sqrt{\frac{c_2}{c_1} \mu^{N_0} \exp(-2\lambda T)} \|z(0)\|$. This proves GES.

Notice that the value of the initial time t_0 was fixed to 0 for convenience. In fact, (6.27) is GES for any t_0 , i.e., it is GES uniformly in t_0 .

Remark 18 Recall that changes of the topology in $[t_i, t_{i+1})$, where $t_i, t_{i+1} \in \mathcal{T}$, remain unnoticed until t_{i+1} (or even later). Therefore, if $\min_{i \in \mathcal{P}} \tau_i \geq \tau_a$, then we effectively have

that the switched system (6.27) is GES uniformly in t_0 for any switching signal. Obviously, we want to obtain τ_i 's in Subsection 6.3.3 as large as possible. This is yet another motivation for developing self-triggered control policies.

6.4.2 Switching with Disturbances

After proving that the outputs of (6.5) synchronize exponentially in the absence of disturbance ω provided that the switching signal σ has the average dwell-time τ_a given by (6.35), we now analyze output synchronization when ω is present. Basically, this subsection provides a result similar to Corollary 3 but for the switched system (6.27).

Notice that (6.27) can be interpreted as a linear time-varying impulsive system. From Theorem 11 we infer that the state transition matrix $\Phi(t, t_0)$ of (6.27) satisfies

$$\|\Phi(t, t_0)\| \leq k \exp(-l(t - t_0)), \quad (6.36)$$

where $k = \sqrt{\frac{c_2}{c_1} \mu^{N_0}}$ and $l = \lambda$ for some $\lambda \in (0, \lambda_0)$. For the explicit form of state transition matrices of linear time-varying impulsive systems refer to [136, Chapter 3]. From the corresponding variation of constants formula (see [136, Chapter 3])

$$z(t) = \Phi(t, t_0)z(0) + \int_{t_0}^t \Phi(t, s)B_\omega\omega(s)ds,$$

and (6.36), we obtain

$$\|z(t)\| \leq k \exp(-l(t - t_0))z(0) + bk \int_{t_0}^t \exp(-l(t - s))\|\omega(s)\|ds,$$

where $\|B_\omega\| \leq b$. Since $t \geq t_0$, therefore $\int_{t_0}^t \exp(-l(t-s))ds \leq 1/l$ for any t_0 . Using [133, Theorem 12.2], we infer that (6.27) is *uniformly bounded-input bounded-state* stable which in turn implies ISS (refer to [68, Theorem 2.35 & Remark 2.36] for more details). We conclude the above discussion in the following theorem.

Theorem 12 Assume that Theorem 11 holds so that the system (6.27) is GES uniformly in t_0 for every switching signal σ with average dwell-time (6.35). Then, output synchronization of systems given by (6.5) is ISS from ω to (e, x) .

6.5 Example

Consider the following five agents:

$$\begin{aligned} A_1 &= \begin{bmatrix} -3 & 1 \\ 2 & -1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} -4 \\ -4 \end{bmatrix}, \quad C_1 = \begin{bmatrix} -4 & 1 \end{bmatrix}, \quad K_1 = 2, \\ A_2 &= \begin{bmatrix} 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} -5 \end{bmatrix}, \quad K_2 = 4, \\ A_3 &= \begin{bmatrix} -2 & -5 \\ 1 & -1 \end{bmatrix}, \quad B_3 = \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \quad C_3 = \begin{bmatrix} -1 & -1 \end{bmatrix}, \quad K_3 = -2, \\ A_4 &= \begin{bmatrix} -3 & -1 \\ 5 & -3 \end{bmatrix}, \quad B_4 = \begin{bmatrix} -4 \\ -2 \end{bmatrix}, \quad C_4 = \begin{bmatrix} 0 & -4 \end{bmatrix}, \quad K_4 = 4, \\ A_5 &= \begin{bmatrix} 1 & 3 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad B_5 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}, \quad C_5 = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}, \quad K_5 = 5. \end{aligned}$$

In addition to topology L_1 given by (6.15), we consider another topology given by

$$L_2 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}. \quad (6.37)$$

One can easily verify that the corresponding A_1^{cl} and A_2^{cl} are Hurwitz even though condition (i) from Remark 13 is not fulfilled for $i = 3$. In addition, matrices A_2 and A_5 contain unstable eigenvalues. Notice that L_2 satisfies Assumption 7. Applying Algorithm 1 to L_2 ,

Chapter 6. Decentralized Output Synchronization

we obtain the partition $\mathcal{P}_1 = \{1, 3\}$, $\mathcal{P}_2 = \{2, 4\}$ and $\mathcal{P}_3 = \{5\}$. According to Section 6.4, let us choose $d = \mathbf{0}_{n_d}$. Consequently, $x' = x$.

Let us compute τ for the first topology given by L_1 . For L_1 , equations (6.13) and (6.14) become

$$\begin{aligned} \dot{x} &= \underbrace{\begin{bmatrix} -67 & 17 & 40 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\ -62 & 15 & 40 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\ 0 & 0 & -19 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & 0 & 0 & -4 & -4 & -4 \\ 0 & 0 & 0 & 3 & 1 & 0 & 0 & -2 & -2 & -2 \\ 64 & -16 & 80 & 0 & 0 & -3 & -129 & 0 & 0 & 0 \\ 32 & -8 & 40 & 0 & 0 & 5 & -67 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 5 & 0 & 20 & -9 & -7 & -9 \\ 0 & 0 & 0 & 5 & 5 & 0 & 20 & -10 & -11 & -9 \\ 0 & 0 & 0 & -5 & -5 & 0 & -20 & 11 & 10 & 10 \end{bmatrix}}_{A_1^{\text{cl}}} x + \\ &+ \underbrace{\begin{bmatrix} 16 & -8 & 0 & -8 & 0 \\ 16 & -8 & 0 & -8 & 0 \\ 0 & 4 & -4 & 0 & 0 \\ 0 & 0 & -4 & 0 & -4 \\ 0 & 0 & -2 & 0 & -2 \\ -16 & -16 & 0 & 32 & 0 \\ -8 & -8 & 0 & 16 & 0 \\ 0 & 0 & -5 & -5 & 10 \\ 0 & 0 & -5 & -5 & 10 \\ 0 & 0 & 5 & 5 & -10 \end{bmatrix}}_{B_1^{\text{cl}}} e + \underbrace{\begin{bmatrix} -4 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{B^d} \omega, \\ \dot{e} &= -\underbrace{\begin{bmatrix} -4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix}}_{C^{\text{cl}}} \dot{x}. \end{aligned}$$

Let us consider the case $p = 2$ and apply [56, Theorem 5.4]. State space matrices for [56, Theorem 5.4] are: $A = A_1^{\text{cl}}$, $B = [B_1^{\text{cl}} \ B_d]$, $C = C^{\text{cl}} A_1^{\text{cl}}$ and $D = C^{\text{cl}} [\mathbf{0} \ B_d]$, where $\mathbf{0}$ is the matrix with all zero entries and dimensions equal the dimensions of B_1^{cl} . The corresponding transfer function $G(s) = C(sI - A)^{-1}B + D$ is obtained in MATLAB. Using the MATLAB function `norm(G(s), inf)`, the corresponding \mathcal{L}_2 -gain is readily obtained: $\gamma_1 = 132$. From (6.18) we obtain

$$A_1^* = \begin{bmatrix} 48 & 24 & 0 & 32 & 0 \\ 24 & 20 & 20 & 32 & 0 \\ 0 & 20 & 6 & 0 & 6 \\ 32 & 32 & 0 & 64 & 5 \\ 0 & 0 & 6 & 5 & 10 \end{bmatrix}.$$

Chapter 6. Decentralized Output Synchronization

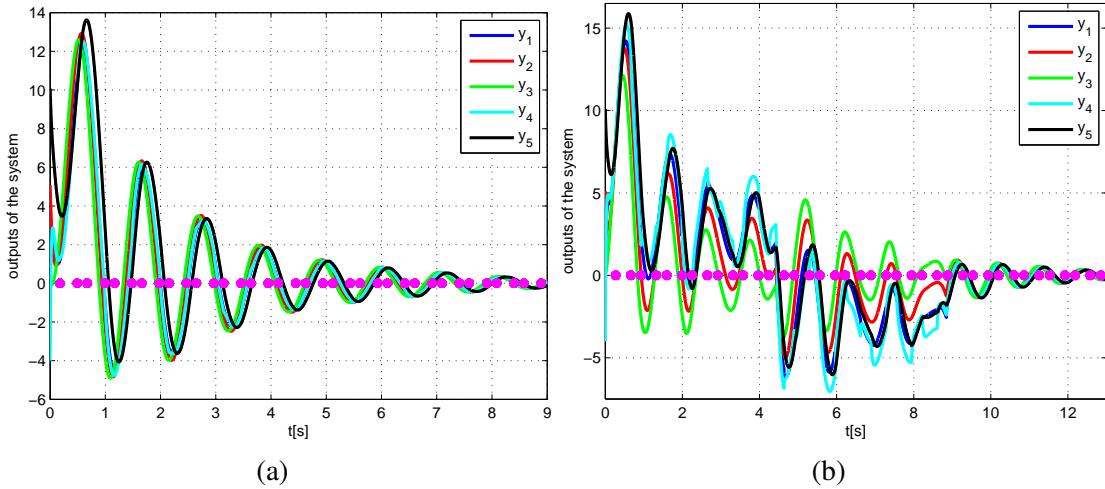


Figure 6.3: Outputs of the agents for: (a) Scenario without disturbance; and, (b) Scenario with disturbance. Magenta dots indicate switching instants.

Solving (6.20) with $T = 3$ such that $\gamma_1 \gamma_{e,1} < 1$ yields $\tau_1 = 1.1 * 10^{-3}$ s. This corresponds to broadcasting frequency of 307 Hz for each agent. The same steps for L_2 yield $\gamma_2 = 267$, $\tau_2 = 8.7 * 10^{-4}$ s and broadcasting frequency 383 Hz for each agent.

In order to verify results of Section 6.4, we toggle between topologies L_1 and L_2 . Numerical results for two scenarios are provided in Figure 6.3. In the first scenario, we choose $\omega \equiv \mathbf{0}_{n_y}$. In the second scenario, we choose $\omega_i(t) = 5t_{[0,4.4)} - 5t_{[4.4,8.8)} + 0t_{[8.8,13)}$, $i \in \{1, \dots, n_d\}$, where $t_{\mathcal{I}}$ is the indicator function on an interval \mathcal{I} . In other words, $t_{\mathcal{I}} = t$ when $t \in \mathcal{I}$ and zero otherwise. According to Theorem 10, we know that each topology is GES. At the moment, we are devising methods to obtain constants k and l from Definition 9 for these two topologies. Consequently, we are not able to explicitly compute the lower bound on τ_a in (6.35). However, in a number of performed simulations, we were not able to find a switching signal $\sigma(t)$ that would destabilize the system. This suggests that $\min\{\tau_1, \tau_2\} \geq \tau_a$ (see Remark 18).

6.6 Conclusions and Future Work

This chapter investigates a problem of decentralized output synchronization for heterogeneous linear systems. Motivated by the rational use of limited resources, we develop a self-triggered output broadcasting policy for the interconnected systems. In other words, each system broadcasts its outputs only when is needed in order for the outputs of all systems to synchronize. Thus, the control signal of each system is updated based on currently available but outdated outputs received from its neighbors. These broadcasting instants adapt to the current communication topology. For a fixed topology, we prove that our broadcasting policy yields global exponential output synchronization, and \mathcal{L}_p -stable output synchronization in the presence of disturbances. When switching topology is considered, we provide an average dwell-time condition that yields disturbance-to-state stable output synchronization. We point out that our approach is applicable to directed and unbalanced communication topologies.

The future work is dedicated to computing the average dwell-time explicitly. In addition, we plan to incorporate noisy measurements and delays. Taking into account our previous work, an extension to nonlinear systems seems straightforward.

Appendix A

Proofs of main results from Chapter 4

Lemma 3 For $p \in [1, \infty)$, \mathcal{L}_p -stability from input ω to state x implies ISS.

Proof 20 Take inequality

$$\|x[t_0, t]\|_p \leq K\|x_0\| + \gamma\|\omega[t_0, t]\|_p$$

and raise both sides to the p^{th} power. Using the following inequality found in [137]

$$(a + b)^p \leq (1 + \epsilon)^{p-1}a^p + \left(1 + \frac{1}{\epsilon}\right)^{p-1}b^p,$$

where $a, b \geq 0$ and $\epsilon > 0$, we obtain

$$\|x[t_0, t]\|_p^p \leq (1 + \epsilon)^{p-1}K^p\|x_0\|^p + \left(1 + \frac{1}{\epsilon}\right)^{p-1}\gamma^p\|\omega[t_0, t]\|_p^p.$$

Introducing $K_1 = (1 + \epsilon)^{p-1}K^p$ and $\gamma_1 = \left(1 + \frac{1}{\epsilon}\right)^{p-1}\gamma^p$ yields

$$\int_{t_0}^t \|x(s)\|^p ds \leq K_1\|x_0\|^p + \gamma_1 \int_{t_0}^t \|\omega(s)\|^p ds.$$

The above inequality has the form of inequality (2) in [135] which is equivalent to ISS. This proves the claim.

Appendix A. Proofs of main results from Chapter 4

Proof 21 (Proof of Theorem 2) We prove this theorem for the case $p \in [1, \infty)$. The proof for the case $p = \infty$ is similar.

(Sufficiency) Let us start from some initial condition $x(t_0)$ and apply input w to a hybrid system Σ given by (4.11) to obtain the state trajectory $t \mapsto x(t)$ and associated output $t \mapsto y(t)$. Now, we can write for every $t \geq t_0$

$$\begin{aligned} \|y[t_0, t]\|_p^p &= \int_{t_0}^t \|y(s)\|^p ds = \sum_{i=0}^{J-1} \int_{t_i^\delta}^{t_{i+1}^\delta} \|y(s)\|^p ds + \int_{t_J^\delta}^t \|y(s)\|^p ds \\ &= \sum_{i=0}^{J-1} \|y[t_i^\delta, t_{i+1}^\delta]\|_p^p + \|y[t_J^\delta, t]\|_p^p, \end{aligned} \quad (\text{A.1})$$

where $J = \arg \max\{j : t_j^\delta \leq t\}$. From (4.21) we obtain

$$\begin{aligned} \|y[t_0, t]\|_p^p &\leq \sum_{i=0}^{J-1} \left(\tilde{K}(\tau_i^\delta) \|x(t_i^{\delta+})\| + \tilde{\gamma}(\tau_i^\delta) \|\omega[t_i^\delta, t_{i+1}^\delta]\|_p \right)^p + \\ &\quad + \left(\tilde{K}(\tau_J^\delta) \|x(t_J^{\delta+})\| + \tilde{\gamma}(\tau_J^\delta) \|\omega[t_J^\delta, t]\|_p \right)^p. \end{aligned} \quad (\text{A.2})$$

Using (4.22) and (4.23) yields

$$\begin{aligned} \|y[t_0, t]\|_p^p &\leq \sum_{i=0}^{J-1} \left(K_M \|x(t_i^{\delta+})\| + \gamma_M \|\omega[t_i^\delta, t_{i+1}^\delta]\|_p \right)^p + \\ &\quad + \left(K_M \|x(t_J^{\delta+})\| + \gamma_M \|\omega[t_J^\delta, t]\|_p \right)^p. \end{aligned} \quad (\text{A.3})$$

In what follows we use the following version of the Minkowski inequality

$$\left(\sum_{i=1}^M (a_i + b_i)^p \right)^{1/p} \leq \left(\sum_{i=1}^M a_i^p \right)^{1/p} + \left(\sum_{i=1}^M b_i^p \right)^{1/p}, \quad (\text{A.4})$$

where $a_i, b_i \geq 0$ and $M \in \mathbb{N} \cup \{\infty\}$. Taking the p^{th} root of (A.3) yields

$$\begin{aligned} \|y[t_0, t]\|_p &\leq \left(\sum_{i=0}^{J-1} \left(K_M \|x(t_i^{\delta+})\| + \gamma_M \|\omega[t_i^\delta, t_{i+1}^\delta]\|_p \right)^p + \right. \\ &\quad \left. + \left(K_M \|x(t_J^{\delta+})\| + \gamma_M \|\omega[t_J^\delta, t]\|_p \right)^p \right)^{\frac{1}{p}}. \end{aligned} \quad (\text{A.5})$$

Appendix A. Proofs of main results from Chapter 4

Applying (A.4) to the right hand side of (A.5) leads to

$$\begin{aligned} \|y[t_0, t]\|_p \leq & K_M \left(\|x(t_0)\|^p + \sum_{i=1}^J \|x(t_i^+)\|^p \right)^{1/p} + \gamma_M \left(\sum_{i=0}^{J-1} \|\omega[t_i^\delta, t_{i+1}^\delta]\|_p^p + \right. \\ & \left. + \|\omega[t_J^\delta, t]\|_p^p \right)^{\frac{1}{p}}. \end{aligned} \quad (\text{A.6})$$

Let us now apply $(a+b)^{1/p} \leq a^{1/p} + b^{1/p}$ where $a, b \geq 0$ to the first term in (A.6), and notice that

$$\sum_{i=0}^{J-1} \|\omega[t_i^\delta, t_{i+1}^\delta]\|_p^p + \|\omega[t_J^\delta, t]\|_p^p = \|\omega[t_0, t]\|_p^p$$

as in (A.1). Hence,

$$\|y[t_0, t]\|_p \leq K_M \left(\|x(t_0)\| + \sum_{i=1}^J \|x(t_i^{\delta+})\| \right) + \gamma_M \|\omega[t_0, t]\|_p. \quad (\text{A.7})$$

Applying (4.24) we obtain

$$\|y[t_0, t]\|_p \leq K_M(K+1)\|x(t_0)\| + \gamma_M \|\omega[t_0, t]\|_p \quad (\text{A.8})$$

for all $t \geq t_0$.

(Necessity) Let us show that if at least one of the two conditions in the theorem is not met, then Σ^δ is not \mathcal{L}_p -stable with the constant $K_M(K+1)$ and gain γ_M for a given δ . Basically, we have to find an input ω and a sequence of $x(t_i^{\delta+})$'s that violate the constant $K_M(K+1)$ and/or gain γ_M .

First, assume that condition (i) is not satisfied, i.e., there exists at least one $j \in \mathbb{N}_0$ such that $\tilde{K}(\tau_j^\delta) > K_M$ or $\tilde{\gamma}(\tau_j^\delta) > \gamma_M$ or $K_M = \infty$ or $\gamma_M = \infty$. If either $K_M = \infty$ or $\gamma_M = \infty$, Σ^δ is not \mathcal{L}_p -stable by definition since $K_M \notin \mathbb{R}$ or $\gamma_M \notin \mathbb{R}$. Without loss of generality, let us assume that j is unique. If $\tilde{\gamma}(\tau_j^\delta) > \gamma_M$, take $x(t_0) = 0$ and $\omega \equiv 0$ everywhere but on the horizon τ_j^δ . On the horizon τ_j^δ , take $\omega[t_j^\delta, t_{j+1}^\delta]$ that yields gain $\tilde{\gamma}(\tau_j^\delta) > \gamma_M$. Obviously, Σ^δ is not \mathcal{L}_p -stable with gain γ_M . If $\tilde{K}(\tau_j^\delta) > K_M$, choose

Appendix A. Proofs of main results from Chapter 4

$x(t_i^{\delta+})$'s such that $\sum_{i=1}^j \|x(t_i^{\delta+})\| = K\|x(t_0)\|$ and $\|x(t_i^{\delta+})\| = 0$ for $i > j$. Since on the horizon τ_j we have that $\tilde{K}(\tau_j^\delta) > K_M$, Σ^δ is not \mathcal{L}_p -stable with the constant $K_M(K + 1)$.

If condition (ii) is violated, say $\|x(t_i^{\delta+})\| = \|x(t_0)\| \neq 0$ for all $i \in \{1, 2, \dots\}$, then Σ^δ is not \mathcal{L}_p -stable because, for example, the zero input $\omega \equiv 0 \in \mathcal{L}_p[t_0, \infty]$ yields the state $x \notin \mathcal{L}_p[t_0, \infty]$. From the detectability assumption and Definition 12, we conclude that $y \notin \mathcal{L}_p[t_0, \infty]$ even though $\|\omega[t_0, \infty]\|_p = 0$. This concludes the proof.

Let $Df(t)$ denote the left-handed derivative of $f : \mathbb{R} \rightarrow \mathbb{R}^n$, i.e.,

$$Df(t) = \lim_{h \rightarrow 0, h < 0} \frac{f(t+h) - f(t)}{h}.$$

The following two lemmas and theorem are taken from [40] and slightly modified.

Lemma 4 Let $I = [t_0, t_1]$, $v \in \mathbb{R}^n$ and consider

$$Dv \preceq Av + d(t), \quad v(t_0) = v_0, \quad \forall t \in I$$

where $A \in \mathcal{A}_{n_e}^+$, $\|A\| < \infty$, and $d(t) : I \rightarrow \mathbb{R}^n$ is continuous. Then, for all $t \in I$, $v(t)$ is bounded by

$$v(t) \preceq \exp(A(t-t_0))v_0 + \int_{t_0}^t \exp(A(t-s))d(s)ds.$$

Lemma 5 (Young's Inequality) Let $*$ denote convolution over the interval I , $f \in \mathcal{L}_p[I]$ and $f \in \mathcal{L}_q[I]$. The Young's inequality is $\|f * g\|_r \leq \|f\|_p \|g\|_q$ for $1/r = 1/p + 1/q - 1$ where $p, q, r > 0$.

Theorem 13 (Riesz-Thorin Interpolation Theorem) Let $F : \mathcal{A}_n^+ \rightarrow \mathcal{A}_n^+$ be a linear operator and suppose that $p_0, p_1, q_0, q_1 \in [1, \infty]$ satisfy $p_0 < p_1$ and $q_0 < q_1$. For any $t \in [0, 1]$ define p_t, q_t by $1/p_t = (1-t)/p_0 + t/p_1$ and $1/q_t = (1-t/q_0) + t/q_1$. Then, $\|F\|_{p_t \rightarrow q_t} \leq \|F\|_{p_0 \rightarrow q_0}^{1-t} \|F\|_{p_1 \rightarrow q_1}^t$. In particular, if $\|F\|_{p_0 \rightarrow q_0} \leq M_0$ and $\|F\|_{p_1 \rightarrow q_1} \leq M_1$, then $\|F\|_{p_t \rightarrow q_t} \leq M_0^{1-t} M_1^t$.

Appendix A. Proofs of main results from Chapter 4

Proof 22 (Proof of Theorem 3) By the hypotheses of the theorem, we have

$$\bar{e} = \overline{g(t, x, e, \hat{\omega}_p, e_\omega)} \preceq A\bar{e} + \tilde{y}(t, x, \hat{\omega}_p, e_\omega)$$

on each interval I , and the i^{th} component of \bar{e} is given by:

$$\left| \frac{d}{dt} e_i(t) \right| = \left| \lim_{h \rightarrow 0, h < 0} \frac{e_i(t+h) - e_i}{h} \right| \geq \lim_{h \rightarrow 0, h < 0} \frac{|e_i(t+h)| - |e_i|}{h} = D\bar{e}_i(t).$$

Therefore,

$$D\bar{e} \preceq A\bar{e} + \tilde{y}(t).$$

Using Lemma 4, we can write

$$\bar{e}(t) \preceq \exp(A(t - t_0))\bar{e}(t_0) + \int_{t_0}^t \exp(A(t-s))\tilde{y}(s)ds. \quad (\text{A.9})$$

Setting the input term $\tilde{y} \equiv 0$, we obtain

$$\bar{e}(t) \preceq \exp(A(t - t_0))\bar{e}(t_0).$$

Taking the norm of both sides of the this inequality and using Lemmas 1 and 2 we obtain:

$$\|\bar{e}(t)\| \leq \exp(\|A\|(t - t_0))\|\bar{e}(t_0)\|. \quad (\text{A.10})$$

Raising to the $p^{\text{th}} \in [1, \infty)$ power and integrating over $[t_0, t]$ yields

$$\|\bar{e}[t_0, t]\|_p^p \leq \frac{\exp(\|A\|p(t - t_0)) - 1}{p\|A\|}\|\bar{e}(t_0)\|^p.$$

Taking the p^{th} root yields

$$\|\bar{e}[t_0, t]\|_p \leq \left(\frac{\exp(\|A\|p(t - t_0)) - 1}{p\|A\|} \right)^{1/p} \|\bar{e}(t_0)\|, \quad p \in [1, \infty). \quad (\text{A.11})$$

The \mathcal{L}_∞ bound is easily obtained by taking $\lim_{p \rightarrow \infty} \|\bar{e}[t_0, t]\|_p$ obtaining

$$\|\bar{e}[t_0, t]\|_\infty \leq \exp(\|A\|(t - t_0)) - 1 \|\bar{e}(t_0)\|.$$

Appendix A. Proofs of main results from Chapter 4

Let us now set $\bar{e}(t_0) = 0$ and estimate the contribution from the input term. From (A.9) we have:

$$\bar{e}(t) \leq \int_{t_0}^t \exp(A(t-s))\tilde{y}(s)ds.$$

Using Lemmas 1 and 2 we obtain

$$\|\bar{e}(t)\| \leq \int_{t_0}^t \exp(\|A\|(t-s))\|\tilde{y}(s)\|ds. \quad (\text{A.12})$$

Let us denote $\phi(s) = \exp(\|A\|s)$. Integrating the previous inequality and using Lemma 5 with $p = q = r = 1$ yields the \mathcal{L}_1 -norm estimate:

$$\|\bar{e}[t_0, t]\|_1 \leq \|\phi[0, t-t_0]\|_1 \|\tilde{y}[t_0, t]\|_1. \quad (\text{A.13})$$

Applying the \mathcal{L}_∞ -norm to (A.12) and using Lemma 5 with $q = r = \infty$ and $p = 1$ yields the \mathcal{L}_∞ -norm estimate:

$$\|\bar{e}[t_0, t]\|_\infty \leq \|\phi[0, t-t_0]\|_1 \|\tilde{y}[t_0, t]\|_\infty. \quad (\text{A.14})$$

We can think of (A.9) as a linear operator G mapping \tilde{y} to \bar{e} with bound for the norms $\|G\|_1 \leq \|G\|_1^*$ and $\|G\|_\infty \leq \|G\|_\infty^*$ where $\|G\|_1^*$ and $\|G\|_\infty^*$ are given by (A.13) and (A.14), respectively. Because $\|G\|_1^* = \|G\|_\infty^*$, Theorem 13 gives that $\|G\|_p \leq \|G\|_1^* = \|G\|_\infty^*$ for all $p \in [1, \infty]$. This yields

$$\|\bar{e}[t_0, t]\|_p \leq \|\phi[0, t-t_0]\|_1 \|\tilde{y}[t_0, t]\|_p, \quad p \in [1, \infty].$$

Since $\|\phi[0, t-t_0]\|_1 = \frac{\exp(\|A\|(t-t_0))-1}{\|A\|}$, we obtain

$$\|\bar{e}[t_0, t]\|_p \leq \frac{\exp(\|A\|(t-t_0))-1}{\|A\|} \|\tilde{y}[t_0, t]\|_p, \quad p \in [1, \infty]. \quad (\text{A.15})$$

After summing the contributions of (A.11) and (A.15), the statement of the theorem follows.

Proof 23 (Proof of Theorem 4) The following proof (with slight modifications) is used in all stability theorems of Section 4.5. That is why we first derive general results, and later specify the obtained results for Case 1.

Appendix A. Proofs of main results from Chapter 4

Let us first analyze what happens on horizons τ_i^δ , $i \in \mathbb{N}_0$, in between two consecutive switching instants. Without loss of generality, let us analyze τ_0^δ . From assumption (iii) we have

$$\|\tilde{y}[t_0, t]\|_p \leq K_n^\delta \|x(t_0)\| + \gamma_n^\delta \|(e, e_\omega)[t_0, t]\|_p,$$

i.e.,

$$\|\tilde{y}[t_0, t]\|_p \leq K_n^\delta \|x(t_0)\| + \gamma_n^\delta \|e[t_0, t]\|_p + \gamma_n^\delta \|e_\omega[t_0, t]\|_p, \quad (\text{A.16})$$

for all $t \geq t_0$. Next, from assumption (ii) for the continuous dynamics of Σ_e^δ we have

$$\|e[t_i, t']\|_p \leq \tilde{K}_e^\delta(\tau_i) \|e(t_i^+)\| + \tilde{\gamma}_e^\delta(\tau_i) \|\tilde{y}[t_i, t']\|_p \quad (\text{A.17})$$

for all $t_i < t_0 + \tau_0^\delta$ and all $t' \in [t_i, t_{i+1}]$. Constants $\tilde{K}_e^\delta(\tau_i)$ and $\tilde{\gamma}_e^\delta(\tau_i)$ are given by (4.27) and (4.28). Due to the perfect resets of e given by (4.38b), condition (4.24) becomes $0 = \sum_{i=1}^{\infty} \|e(t_i^+)\| \leq K_e^\delta \|e(t_0)\|$ so we can choose $K_e = 0$ for Case 1.

From Figure 4.4 it can be concluded that the open loop gain is $\gamma_n^\delta \gamma_e^\delta$ where γ_n^δ and γ_e^δ are classical \mathcal{L}_p -gains. Recall that the small gain theorem requires $\gamma_n^\delta \gamma_e^\delta < 1$ (see [56]). Now, let us design τ_i 's in (A.17) such that γ_e^δ , obtained from $\tilde{\gamma}_e^\delta(\tau_i)$ using (4.23), satisfies $\gamma_e^\delta \gamma_n^\delta < 1$. Hence, we want all $\tilde{\gamma}_e^\delta(\tau_i)$ to satisfy $\tilde{\gamma}_e^\delta(\tau_i) \leq \kappa / \gamma_n^\delta$. Using (4.28), we obtain

$$\frac{\exp(\|A_i^\delta\| \tau_i) - 1}{\|A_i^\delta\|} \leq \frac{\kappa}{\gamma_n}. \quad (\text{A.18})$$

Solving the above inequalities for τ_i yields

$$\tau_i^* = \frac{1}{\|A_i^\delta\|} \ln \left(\kappa \frac{\|A_i^\delta\|}{\gamma_n} + 1 \right).$$

Using the above τ_i^* 's in (4.27), we calculate $K_{M,e}^\delta$ from $\tilde{K}_e^\delta(\tau_i)$'s via (4.22). This supremum exists since τ_i^* are upper bounded by Remark 5 (below we show that $\|A_i^\delta\|$ is bounded for all C_i 's). Now, all conditions of Theorem 2 are satisfied and we have

$$\|e[t_0, t]\|_p \leq K_{M,e}^\delta \|e(t_0)\| + \gamma_e^\delta \|\tilde{y}[t_0, t]\|_p \quad (\text{A.19})$$

Appendix A. Proofs of main results from Chapter 4

for any $t \geq t_0$.

Applying the small gain theorem with (A.16) and (A.19), we obtain:

$$\|(\tilde{y}, e)[t_0, t]\|_p \leq K_1^\delta \|(x_0, e_0)\| + \gamma_1^\delta \|e_\omega[t_0, t]\|_p \quad (\text{A.20})$$

for all $t \geq t_0$ where $K_1^\delta \geq 0$ and $\gamma_1^\delta \geq 0$ are functions of $K_{M,e}^\delta$, K_n^δ , γ_n^δ and γ_e^δ . For more details regarding the application of the small gain theorem in our settings, refer to the proof of Theorem 7. Using the detectability assumption (v) on x , from Proposition 2 and (A.20) we obtain

$$\|(x, e)[t_0, t]\|_p \leq K_2^\delta \|(x_0, e_0)\| + \gamma_2^\delta \|e_\omega[t_0, t]\|_p \quad (\text{A.21})$$

for all $t \geq t_0$ where $K_2^\delta \geq 0$ and $\gamma_2^\delta \geq 0$.

Let us now verify that condition $\|A_i^\delta\| < \infty$ of Theorem 3 follows from (i), (ii), (iii) and (v) and that a lower bound on dwell times is assured. In other words, we want to ensure existence of V in Assumption 2 using the assumptions of Theorem 4. The initial condition $(x_0, e_0) \in \mathbb{R}^{n_x \times n_e}$ implies $\|(x_0, e_0)\| < \infty$. Next, due to (i), we infer that $\bar{\omega}_p \preceq L_1 \mathbf{1}_{n_\omega}$ where $L_1 \geq L$ accounts for the noise in $\hat{\omega}_p$. Notice that (i) defines \mathcal{P} . Therefore, $\bar{e}_\omega \preceq (L + L_1) \mathbf{1}_{n_\omega}$. The right-continuity of solutions yields bounded $(x(t), e(t))$ on finite horizons (i.e., on bounded and closed intersampling intervals $[t_0, t_1]$) starting from (x_0, e_0) provided that $\gamma_n^\delta \gamma_e^\delta < 1$. What we have so far is $\mathcal{C}_0 = \{(x(t), \hat{\omega}_p, e_\omega(t)) \mid t \in [t_0, t_0 + \tau_0]\}$ where $\hat{\omega}_p$ and e_ω are bounded for any $t \geq t_0$ and the set of possible x is bounded and closed for a bounded τ_0 . The set of values that x can attain on finite horizons is calculated applying reachability analysis to (4.5)-(4.6) starting from t_0 . Notice that, as t grows, so does \mathcal{C}_0 and consequently $\|A_0^\delta\|$ (or stays the same). If we are able to choose a bounded τ_0 , then \mathcal{C}_0 is compact and $\|A_0^\delta\|$ is bounded. Notice that the left side of (A.18) is continuous in τ_0 , equals zero for $\tau_0 = 0$, and monotonically increases in both τ_0 and $\|A^\delta\|$. Due to (iii), there exists a bounded and closed τ_0^* such that (A.18) is satisfied. The same procedure is applied for the subsequent intersampling intervals τ_i 's. However, as t increases in the subsequent $[t_i, t_{i+1}]$'s, it might happen that $x \rightarrow \infty$ yielding $\|A_i^\delta\| \rightarrow \infty$ and $\tau_i^* \rightarrow 0$, i.e.,

Appendix A. Proofs of main results from Chapter 4

the undesired Zeno behavior may occur. Because \bar{e}_ω is bounded in (A.21), Lemma 3 from the Appendix yields that x is bounded on τ_0^δ , i.e., V from Assumption 2 exists. Hence, Zeno behavior and accumulation of switching instants in finite time are avoided.

So far we have constructed \mathcal{L}_p -stable systems when δ is constant. Now, we allow for switches and patch together systems on different τ_i^δ 's using Theorem 2 again. Since $\sup_{\delta \in c, c \in \mathcal{P}} \gamma_n^\delta < \infty$, $\gamma_n^\delta \gamma_e^\delta \leq \kappa$ on all τ_i^δ 's and all τ_i 's are bounded, it can be shown that suprema (4.22) and (4.23) exist. Using (iv) and applying Theorem 2, we conclude that there exist $K \geq 0$ and $\gamma \geq 0$ such that

$$\|(x, e)[t_0, t]\|_p \leq K\|(x_0, e_0)\| + \gamma\|e_\omega[t_0, t]\|_p \quad (\text{A.22})$$

for all $t \geq t_0$ given some δ .

For Case 1 we have $\|e_\omega[t_0, t]\|_p = 0$ yielding

$$\|(x, e)[t_0, t]\|_p \leq K\|(x_0, e_0)\|.$$

Using Lemma 3, we know that there exists a class- \mathcal{KL} function β such that $\|(x(t), e(t))\| \leq \beta(\|(x(t_0), e(t_0))\|, t - t_0)$. Consequently, the closed-loop system (4.38) is globally uniformly asymptotically stable. This concludes the proof.

References

- [1] R. Poovendran, “Cyber-physical systems: Close encounters between two worlds [point of view],” *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1363–1366, 2010.
- [2] D. Tolić and R. Fierro, “Adaptive sampling for tracking in pursuit-evasion games,” in *IEEE Multi-Conference on Systems and Control*, Denver, CO, September 2011, pp. 179 –184.
- [3] T. Estrada and P. J. Antsaklis, “Stability of model-based networked control systems with intermittent feedback,” in *Proc. of the 17th IFAC World Congress on Automatic Control*, July 2008, pp. 12 581–12 586.
- [4] C. Li, G. Feng, and X. Liao, “Stabilization of nonlinear systems via periodically intermittent control,” *IEEE Trans. on Circuits and Systems – II: Express Briefs*, vol. 54, no. 11, pp. 1019 – 1023, 2007.
- [5] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, September 2007.
- [6] J. Hespanha, P. Naghshtabrizi, and X. Yonggang, “A survey of recent results in Networked Control Systems,” *Proc. of IEEE Special Issue on Technology of Networked Control Systems*, vol. 95, no. 1, pp. 138 – 162, January 2007.
- [7] B. Triplet, D. Klein, and K. Morgansen, “Cooperative estimation for coordinated target tracking in a cluttered environment,” *ACM/Springer Mobile Networks and Applications (MONET)*, vol. 14, no. 3, pp. 336–349, 2009.
- [8] M. Bertogna, M. Cirinei, and G. Lipari, “Schedulability analysis of global scheduling algorithms on multiprocessor platforms,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 20, no. 4, pp. 553–566, April 2009.
- [9] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*. online book, 2008.

References

- [10] M. Zavlanos and G. Pappas, “Distributed hybrid control for multiple pursuer multiple evader games,” in *10th International Conference on Hybrid Systems: Computation and Control*, Pisa, Italy, April 2007, pp. 787–789.
- [11] C. Cassandras and J. Lygeros, *Stochastic Hybrid Systems*. CRC Press, 2006.
- [12] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [13] S. Ge and F. Lewis, Eds., *Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications*. Boca Raton, FL: CRC Press, Taylor and Francis Group, 2006, chapters 10., 11. and 13.
- [14] R. Sanfelice and E. Frazzoli, “A hybrid control framework for robust maneuver-based motion planning,” in *American Control Conference*, Seattle, WA, July 2008, pp. 2254–2259.
- [15] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [16] C. Cai and S. Ferrari, “Information-driven sensor path planning by approximate cell decomposition,” *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. in press, 2008. [Online]. Available: <http://fred.mems.duke.edu/silvia.ferrari/SMCDeminingarticle.pdf>
- [17] O. Imer, “Optimal estimation and control under communication network constraints,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, Illinois, 2005.
- [18] Z. Wang, D. Ho, and X. Liu, “Variance-constrained filtering for uncertain stochastic systems with missing measurements,” *IEEE Trans. on Automatic Control*, vol. 48, no. 7, pp. 1254 – 1258, July 2003.
- [19] Z. Jin, C.-K. Ko, and R. Murray, “Estimation for nonlinear dynamical systems over packet-dropping networks,” in *Proceedings of the American Control Conference*, New York, July 2007, pp. 5037 – 5042.
- [20] A. Tantawy, X. Koutsoukos, and G. Biswas, “Maximum likelihood detection with intermittent observations,” in *43rd Annual Conference on Information Sciences and Systems*, Baltimore, Maryland, March 2009, pp. 442 – 447.
- [21] M. Micheli and M. Jordan, “Random sampling of a continuous-time stochastic dynamical system,” in *Proc. of the 15th Int. Symposium on Mathematical Theory of Networks and Systems*, Indiana, USA, 2002.

References

- [22] S. C. Smith and P. Seiler, “Estimation with lossy measurements: Jump estimators for jump systems,” *IEEE Trans. on Automatic Control*, vol. 48, no. 12, pp. 2163 – 2171, December 2003.
- [23] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, “Kalman filtering with intermittent observations,” *IEEE Trans. on Aut. Control*, vol. 49, no. 9, pp. 1453 – 1464, Sept. 2004.
- [24] K. Plarre and F. Bullo, “On Kalman filtering for detectable systems with intermittent observations,” *IEEE Trans. on Automatic Control*, vol. 54, no. 2, pp. 386 – 390, February 2009.
- [25] V. Saligrama and D. Castanon, “Reliable tracking with intermittent communications,” in *ICASSP 2006 Proceedings*, vol. 5, May 2006, pp. 1141 – 1144.
- [26] M. Porfiri, D. Roberson, and D. Stilwell, “Tracking and formation control of multiple autonomous agents: A two-level consensus approach,” *Automatica*, vol. 43, no. 8, pp. 1318 – 1328, August 2007.
- [27] Z. Tang, “Information-theoretic management of mobile sensor agents,” Ph.D. dissertation, The Ohio State University, Ohio, 2005.
- [28] T. Chung and J. Burdick, “Multi-agent probabilistic search in a sequential decision-theoretic framework,” in *IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, 2008, pp. 146 – 151.
- [29] A. Haug, *A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes*, MITRE Corporation, January 2005, technical report.
- [30] A. Doucet and A. Johansen, *A tutorial on particle filtering and smoothing: fifteen years later*, Department of Statistics, University of British Columbia, December 2008, technical report.
- [31] H. A. P. Blom, “An efficient decision-making-free filter for processes with abrupt changes,” in *Proc. 7th IFAC Symp. Identification and System Parameter Estimation*, York, UK, July 1985.
- [32] A. Satish and R. L. Kashyap, “Multiple target tracking using maximum likelihood principle,” *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1677–1695, 1995.
- [33] J. Chen, Z. Qin, and X. L. R. Guo, “Interacting multiple model particle-type filtering approaches to ground target tracking,” *Journal of Computers*, vol. 3, no. 7, pp. 23–30, July 2008.

References

- [34] G. Hollinger and S. Singh, “Multi-robot coordination with periodic connectivity,” in *IEEE Int. Conf. on Robotics and Automation*, Anchorage, Alaska, May 2010, accepted for publication.
- [35] K. J. Åström and B. Wittenmark, *Computer Controlled Systems*. Englewood Cliffs, NJ: Prentice Hall, 1990.
- [36] D. Nešić, A. Teel, and P. Kokotović, “Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations,” *System and Control Letters*, vol. 38, no. 4-5, pp. 259–270, 1999.
- [37] M. Schwager, N. Michael, V. Kumar, and D. Rus, “Time scales and stability in networked multi-robot systems,” in *Proc. of the International Conference on Robotics and Automation (ICRA 11)*, May 2011, pp. 3855–3862.
- [38] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, “Energy-efficient autonomous four-rotor flying robot controlled at 1 khz,” in *Proceedings of IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007, pp. 361–366.
- [39] D. Nešić and A. R. Teel, “Input-output stability properties of Networked Control Systems,” *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1650–1667, October 2004.
- [40] M. Tabbara, D. Nešić, and A. R. Teel, “Stability of wireless and wireline networked control systems,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1615–1630, September 2007.
- [41] M. Tabbara and D. Nešić, “Input-output stability of Networked Control Systems with stochastic protocols and channels,” *IEEE Trans. Aut. Contr.*, vol. 53, no. 5, pp. 1160–1175, 2008.
- [42] H. Yu and P. Antsaklis, “Event-triggered real-time scheduling for stabilization of passive and output feedback passive systems,” in *Proceedings of the American Control Conference*, San Francisco, CA, June-July 2011, pp. 1674 – 1679.
- [43] H. Yu and P. J. Antsaklis, *Output Synchronization of Multi-Agent Systems with Event-Driven Communication: Communication Delay and Signal Quantization*, Department of Electrical Engineering, University of Notre Dame, July 2011, technical report. [Online]. Available: <http://www.nd.edu/~pantsakl/Publications/isis-2011-001.pdf>
- [44] A. Anta and P. Tabuada, “Isochronous manifolds in self-triggered control,” in *Proc. IEEE Conference for Decision and Control*, Shanghai, China, December 2009, pp. 3194 – 3199.

References

- [45] ——, “To sample or not to sample: Self-triggered control for nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030 – 2042, September 2010.
- [46] M. Lemmon, *Event-triggered Feedback in Control, Estimation, and Optimization*, ser. Lecture Notes in Control and Information Sciences, A. Bemporad, M. Heemels, and M. Johansson, Eds. Springer Verlag, 2010, vol. 405.
- [47] P. Tallapragada and N. Chopra, “On event triggered trajectory tracking for control affine nonlinear systems,” in *Proceedings of the IEEE Conference on Decision and Control*, December 2011, pp. 5377–5382.
- [48] C. Nowzari and J. Cortés, “Self-triggered coordination of robotic networks for optimal deployment,” in *American Control Conf.*, San Francisco, CA, June-July 2011, pp. 1039–1044.
- [49] J. Willems, “Dissipative dynamical systems, parts i and ii,” *Archive for Rational Mechanics and Analysis*, vol. 45, pp. 321–393, December 1972.
- [50] J. M. Esposito, “Simulation and control of hybrid systems with applications to mobile robotics,” Ph.D. dissertation, University of Pennsylvania, PA, 2002.
- [51] D. Copp and R. G. Sanfelice, “On the effect and robustness of zero-crossing detection algorithms in simulation of hybrid systems jumping on surfaces,” in *Proceedings of the American Control Conference*, Montreal, Canada, 2012, pp. 2449 – 2454.
- [52] M. M. K. Oishi, “User-interfaces for hybrid systems: Analysis and design through hybrid reachability,” Ph.D. dissertation, Stanford University, CA, November 2003.
- [53] J. H. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, “Applications of hybrid reachability analysis to robotic aerial vehicles,” *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 335–354, March 2011.
- [54] J. Lygeros, “Lecture notes on hybrid systems,” Department of Electrical and Computer Engineering, University of Patras, Greece, 2004.
- [55] R. Goebel, R. Sanfelice, and A. Teel, “Hybrid dynamical systems,” *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28 – 93, 2009.
- [56] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
- [57] P. Yang, R. Freeman, and K. Lynch, “A general stability condition for multi-agent coordination by coupled estimation and control,” in *Proceedings of the American Control Conference*, New York City, July 2007, pp. 723–728.

References

- [58] S. Hirche, C. Chen, and M. Buss, “Performance oriented control over networks switching controllers and switched time delay,” *Asian Journal of Control*, vol. 10, no. 1, pp. 24–33, 2008.
- [59] S. Hedlund and A. Rantzer, “Optimal control of hybrid systems,” in *Conference on Decision and Control*, 1999, pp. 3972–3977.
- [60] S. C. Bengea and R. A. DeCarlo, “Optimal control of switching systems,” *Automatica*, vol. 41, no. 1, pp. 11–27, 2005.
- [61] X. Xuping and P. Antsaklis, “Optimal control of switched systems based on parameterization of the switching instants,” *IEEE Trans. on Automatic Control*, vol. 49, no. 1, pp. 2–16, January 2004.
- [62] D. P. Bertsekas, *Dynamic Programming and optimal control, Vol. I*, 3rd ed. Belmont, Massachusetts: Athena Scientific, 2005.
- [63] ———, *Dynamic Programming and optimal control, Vol. II*, 3rd ed. Belmont, Massachusetts: Athena Scientific, 2007.
- [64] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, ser. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley and Sons, Inc., 2007.
- [65] R. Sutton and A. Barto, *Reinforcement Learning*. Cambridge, Massachusetts: The MIT Press, 1998.
- [66] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, Massachusetts: Athena Scientific, 1996.
- [67] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [68] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control - Theory and Applications*, ser. Communications and Control Engineering. Springer London, 2008.
- [69] M. Franceschelli, A. Gasparri, A. Giua, and G. Ulivi, “Decentralized stabilization of heterogeneous linear multi-agent systems,” in *IEEE Int. Conf. on Robotics and Automation*, May 2010, pp. 3556–3561.
- [70] L. Fang and P. J. Antsaklis, “On communication requirements for multi-agent consensus seeking,” in *Networked Embedded Sensing and Control, Proceedings of Workshop NESCO5: University of Notre Dame, USA, October 17-18, 2005*, ser.

References

- Lecture Notes in Control and Information Sciences, P. Antsaklis and P. Tabuada, Eds. Springer, 2006, vol. 331, pp. 133–147.
- [71] F. Xiao and L. Wang, “Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays,” *IEEE Trans. on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.
 - [72] M. Guinaldo, D. V. Dimarogonas, K. H. Johansson, J. Sánchez, and S. Dormido, “Distributed event-based control for interconnected linear systems,” in *Proceedings of the IEEE Conference on Decision and Control*, Orlando, FL, December 2011, pp. 2553–2558.
 - [73] J. Schneider and R. Wattenhofer, “What is the use of collision detection (in wireless networks)?” in *Distributed Computing*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2010, vol. 6343, pp. 133–147.
 - [74] SkyPilot, “Why synchronous for wireless,” 2008. [Online]. Available: http://skypilot.trilliantinc.com/pdf/wp_WhySynchronousForWireless.pdf
 - [75] DigiLink, “Business class synchronous wireless connectivity,” 2010. [Online]. Available: <http://www.digilink.net/business-wireless.shtml>
 - [76] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, “The challenges of building mobile underwater wireless networks for aquatic applications,” *IEEE Network*, vol. 20, no. 3, pp. 12–18, 2006.
 - [77] R. Obermaisser, *Time-Triggered Communication*. CRC Press, 2011.
 - [78] ———, *Event-Triggered and Time-Triggered Control Paradigms*, ser. Real-Time Systems Series. Springer US, 2005, vol. 22.
 - [79] L. Bao and J. Garcia-Luna-Acaves, “Link-state routing in networks with unidirectional links,” in *Eight International Conference in Computer Communications and Networks*, 1999, pp. 358–363.
 - [80] M. Harchol-Balter, T. Leighton, and D. Lewin, “Resource discovery in distributed networks,” in *8th Annual ACM-SIGACT/SIGOPS Symposium on Principles of Distributed Computing (PODC 1999)*, 1999, pp. 229–238.
 - [81] V. V. Dimakopoulos and E. Pitoura, “A peer-to-peer approach to resource discovery in multi-agent systems (extended version),” in *Proc. of CIA2003, Springer-Verlag, Lecture Notes on Computer Science*, 2003, pp. 62–77.

References

- [82] T. S. Hoang, H. Kuruma, D. Basin, and J.-R. Abrial, “Developing topology discovery in event-b,” *Science and Computer Programming*, vol. 74, no. 11-12, pp. 879–899, November 2009.
- [83] B. Perteet, J. McClintock, and R. Fierro, “A multi-vehicle framework for the development of robotic games: The Marco Polo case,” in *IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, April 10-14 2007, pp. 3717–3722.
- [84] S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. Baumgartner, “A geometric optimization approach to detecting and intercepting dynamic targets using a mobile sensor network,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 292–320, 2009.
- [85] D. Tolić, R. Fierro, and S. Ferrari, “Cooperative multi-target tracking via hybrid modeling and geometric optimization,” in *17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, June 24-26 2009, pp. 440–445.
- [86] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, “A vision-based formation control framework,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, October 2002.
- [87] K. Baumgartner, “Control and optimization of track coverage in underwater sensor networks,” Ph.D. dissertation, Duke University, North Carolina, December 2007.
- [88] A. Ghaffarkhah and Y. Mostofi, “Communication-aware target tracking using navigation functions - centralized case,” in *Int. Conf. on Robot Communication and Coordination*, Odense, Denmark, April 2009.
- [89] J. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006.
- [90] T. Wettergren, R. Streit, and J. Short, “Tracking with distributed sets of proximity sensors using geometric invariants,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1366 – 1374, 2004.
- [91] A. Gelman, “Objections to Bayesian statistics,” *Bayesian Analyses*, vol. 3, no. 3, pp. 445 – 450, 2008.
- [92] V. Isler and N. Karnad, “The role of information in the cop-robber game,” *Theoretical Computer Science*, vol. 3, no. 399, pp. 179 – 190, 2008, accepted to the Special Issue on Graph Searching.
- [93] V. Pratt, “Direct least-squares fitting of algebraic surfaces,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 145 – 152, 1987.

References

- [94] N. Chernov, *Circular and linear regression: Fitting circles and lines by least squares*, University of Alabama at Birmingham, 2009, manual. [Online]. Available: <http://www.math.uab.edu/chernov/cl/book.pdf>
- [95] H. Wang, A. Szabo, J. Bamberger, D. Brunn, and U. Hanebeck, “Performance comparison of nonlinear filters for indoor WLAN positioning,” in *11th International Conference on Information Fusion*, Cologne, June-July 2008, pp. 1 – 7.
- [96] F. Daum, “Nonlinear filters: beyond the Kalman filter,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57 – 69, August 2008.
- [97] W. D. Jones, “Keeping cars from crashing,” *IEEE Spectrum*, vol. 38, no. 9, pp. 40–45, September 2001.
- [98] C. C. de Wit, H. Khennouf, C. Samson, and O. Sordalen, “Nonlinear control design for mobile robots,” *Recent Trends in Mobile Robots*, pp. 121–156, 1993.
- [99] J. Zhao and D. J. Hill, “Vector \mathcal{L}_2 -gain and stability of feedback switched systems,” *Automatica*, vol. 45, no. 7, pp. 1703–1707, July 2009.
- [100] D. Liberzon, *Switching in Systems and Control*. Birkhauser Boston, 2003.
- [101] J. Imae, “L₂-gain computation for nonlinear systems using optimal control algorithms,” in *Proceedings of the IEEE Conference on Decision and Control*, Kobe, Japan, December 1996, pp. 547–551.
- [102] D. Tolić and R. Fierro, “Stability of feedback linearization under intermittent information: A target-pursuit case,” in *American Control Conference*, 2011, pp. 3184 –3190.
- [103] D. Liberzon and D. Nešić, “Stability analysis of hybrid systems via small-gain theorem,” in *Proceedings of the Ninth International Workshop on Hybrid Systems: Computation and Control, Santa Barbara, CA, March 2006*, pp. 421–435.
- [104] A. Megretski, “Lecture notes on Model Order Reduction,” Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2004.
- [105] A. J. van der Schaft, *\mathcal{L}_2 -Gain and Passivity Techniques in Nonlinear Control*, 2nd ed., E. Sontag and M. Thoma, Eds. Springer Verlag, 2000.
- [106] H. Hjalmarsson, “From experiment design to closed-loop control,” *Automatica*, vol. 41, no. 3, pp. 393–438, 2005.
- [107] M. B. Syberg, “Complexity issues, validation and input design for control in system identification,” Ph.D. dissertation, KTH School of Electrical Engineering, Stockholm, Sweden, 2008.

References

- [108] J. Imae and G. Wanyoike, “ H_∞ norm computation for LTV systems using nonlinear optimal control algorithms,” *International Journal of Control*, vol. 63, no. 1, pp. 161–182, 1996.
- [109] M. James and J. Baras, “Robust H_∞ output feedback control for nonlinear systems,” *IEEE Trans. on Automatic Control*, vol. 40, no. 6, pp. 1007–1017, 1995.
- [110] J. Gutiérrez-Gutiérrez and P. Crespo, “Asymptotically equivalent sequences of matrices and Hermitian block Toeplitz matrices with continuous symbols: Applications to MIMO systems,” *IEEE Trans. on Information Theory*, vol. 54, no. 12, pp. 5671–5680, December 2008.
- [111] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.
- [112] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [113] D. Tolić, R. G. Sanfelice, and R. Fierro, “Self-triggering in nonlinear systems: A small-gain theorem approach,” in *IEEE Mediterranean Conf. on Contr. and Automat.*, Barcelona, Spain, 2012, under review.
- [114] J. Hespanha and A. Tiwari, Eds., *Proceedings of the Ninth International Workshop on Hybrid Systems: Computation and Control, Santa Barbara, CA, March 2006*, ser. Lecture notes in Computer Science. Springer, Berlin, 2006, vol. 3927.
- [115] S. Hirche, T. Matiakis, and M. Buss, “A distributed controller approach for delay-independent stability of networked control systems,” *Automatica*, vol. 45, no. 5, pp. 1828–1836, August 2009.
- [116] N. Martins, “Finite gain \mathcal{L}_p stability requires analog control,” *Systems and Control Letters*, vol. 55, no. 11, pp. 949–954, November 2006.
- [117] D. Tolić and R. Fierro, “Stability of feedback linearization under intermittent information: A target-pursuit case,” in *American Control Conf.*, San Francisco, CA, 2011, pp. 3184 – 3190.
- [118] W. Zhang, “Controller synthesis for switched systems using approximate dynamic programming,” Ph.D. dissertation, Purdue University, Indiana, December 2009.
- [119] G. J. Gordon, *Stable function approximation in dynamic programming*, School of Computer Science, Carnegie Mellon University, 1995, technical report.
- [120] P. J. Davis and P. Rabinowitz, *Methods of numerical integration*, 2nd ed. Mineola, NY: Dover Publications, Inc., 2007.

References

- [121] J. M. Lee, N. S. Kaisare, and J. H. Lee, “Choice of approximator and design of penalty function for an approximate dynamic programming based control approach,” *Journal of Process Control*, vol. 16, no. 2, pp. 135 – 156, February 2006.
- [122] B. O’Donoghue, Y. Wang, and S. Boyd, “Min-max approximate dynamic programming,” in *IEEE Multi-Conference on Systems and Control*, Denver, CO, September 2011, pp. 424–431.
- [123] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*, ser. Automation and Control Engineering Series. CRC Press, 2010.
- [124] A. Samuels, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.
- [125] G. Tesauro, “Neurogammon: a neural network backgammon program,” in *IJNN Proceedings III*, 1990, pp. 33–39.
- [126] L. Lin, “Self-improving reactive agents based on reinforcement learning, planning and teaching,” *Machine Learning*, vol. 8, no. 3-4, pp. 293–322, 1992.
- [127] P. Vamplew and R. Ollington, “Global versus local constructive function approximation for on-line reinforcement learning,” in *Proceedings of the 18th Australian Joint conference on Advances in Artificial Intelligence*, ser. AI’05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 113–122.
- [128] B. M. Bethke, “Kernel-based approximate dynamic programming using bellman residual elimination,” Ph.D. dissertation, MIT, Massachusetts, February 2010.
- [129] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, November 2008.
- [130] I. Szita, “Rewarding excursions: Extending reinforcement learning to complex domains,” Ph.D. dissertation, Eötvös Loránd University, Budapest, Hungary, March 2007.
- [131] J. P. Hespanha and A. S. Morse, “Stability of switched systems with average dwell-time,” in *IEEE Conference on Decision and Control*, December 1999, pp. 2655–2660.
- [132] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, New York, 2001.
- [133] W. J. Rugh, *Linear System Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1996.

References

- [134] G. Walsch and H. Ye, “Scheduling of networked control systems,” *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 57–65, February 2001.
- [135] D. Angeli, E. Sontag, and Y. Wang, “Further equivalences and semiglobal versions of integral input to state stability,” *Dynam. Control*, vol. 10, no. 2, pp. 127–149, 2000.
- [136] D. Bainov and P. Simeonov, *Systems with Impulse Effects: Stability, Theory and Applications*. Ellis Horwood Limited, 1989.
- [137] R. A. Freeman, “On the relationship between induced l^∞ -gain and various notions of dissipativity,” in *Proceedings of the IEEE Conference on Decision and Control*, 2004, pp. 3430–3434.