ABSTRACT

Title of dissertation: CONTROLLER SYNTHESIS UNDER INFORMATION AND FINITE-TIME LOGICAL CONSTRAINTS

Dipankar Maity
Doctor of Philosophy, 2018

Dissertation directed by: Professor John S. Baras
Department of Electrical & Computer Engineering

In robotics, networks, and many related fields, a typical controller design problem needs to address both logical and informational constraints. The logical constraints may arise due to the complex task description or decision making process, while the information-related constraints emerge naturally as a consequence of the limitations on communication and computation capabilities.

In the first part of the thesis, we consider the problem of synthesizing an event-based controller to address the information-related constraints in the controller design. We consider dynamical systems that are operating under continuous state feedback. This assumes that the measurements are continuously transmitted to the controller in order to generate the input and thus, increases the cost of communication by requiring huge communication resources. In many situations, it so happens that the measurement does not change fast enough that continuous transmission is required. Taking motivation from this, we consider the case where

instead continuous feedback we seek an intermittent-feedback. As a result, the system trajectory will deviate from its ideal behavior. However, the question is how much would it deviate? Given the allowed bound on this deviation, can we design some controller that requires fewer measurements than the original controller and still manages to keep the deviation within this prescribed bound? Two important questions remain: 1) What will be the structure of the (optimal) controller? 2) How will the system know the (optimal) instances to transmit the measurement? When the system sends out measurement to controller, it is called as an "event". Thus, we are looking for an event-generator and a controller to perform event-based control under the constraints on the availability of the state information.

The next part focuses on controller synthesis problems that have logical, spatio-temporal constraints on the trajectory of the system; a robot motion planning problem fits as a good example of these kind of finite-time logically constrained problems. We adopt an automata-based approach to abstract the motion of the robot into an automata, and verify the satisfaction of the logical constraints on this automata. The abstraction of the dynamics of the robot into an automata is based on certain reachability guarantee of the robot's dynamics. The controller synthesis problem over the abstracted automata can be represented as a shortest-path-problem.

In part III, we consider the problem of jointly addressing the logical and information constraints. The problem is approached with the notion of robustness of logical constraints. We propose two different frameworks for this problem with two different notions of robustness and two different approaches for the controller synthesis. One framework relies on the abstraction of the dynamical systems into a finite transition system, whereas the other relies on tools and results from prescribed performance control to design continuous feedback control to satisfy the robust logical constraints. We adopt an hierarchical controller synthesis method where a continuous feedback controller is designed to satisfy the (robust) logical constraints, and later, that controller is replaced by a suitable event-triggered intermittent feedback controller to cope with informational constraints.

CONTROLLER SYNTHESIS UNDER INFORMATION AND FINITE-TIME LOGIC CONSTRAINTS

by

Dipankar Maity

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor André L. Tits
Professor Nuno C. Martins
Professor Perinkulam S. Krishnaprasad
Professor Nikhil Chopra

Dedication

To my parents.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor John Baras, for the opportunity to be a part of his excellent team and allowing me to work on such thought-provoking and relevant research topics. I thank him for his invaluable guidance, unstinting support, and insightful feedback that have helped me to become a more independent researcher. His emphasis on identifying and solving real-world problems, and tireless passion for conducting research are some of the things I aspire to instil in my professional endeavours. I would also like to take the opportunity to acknowledge the support of the following sources: US Air Force Office of Scientific Research MURI grant FA9550-09-1-0538, DARPA through ARO grants W911NF-14-1-0384, W911NF-15-1-0646, Office of Naval Research (ONR) grant N00014-17-1-2622, National Science Foundation (NSF) grants CNS-1035655, CNS-1544787, National Institute of Standards and Technology (NIST) grant 70NANB11H148, DARPA STTR contract with Boston Engineering Corporation, Knut and Alice Wallenberg Foundation by the Swedish Strategic Research Foundation, and by the Swedish Research Council.

I am thankful to Kimberly Edwards, Alexis Jenkins and Melanie Prange for helping me with the administrative matters.

I would like to thank Professor Andre Tits, Professor Perinkulam Krishnaprasad, Professor Nuno Martins, and Professor Nikhil Chopra for sparing their valuable time and agreeing to serve on my committee. I am also thankful to Prof. Tits and Prof. Krishnaprasad for the insightful discussions in class, in office, or even in the corridors.

Throughout my time at the University of Maryland, I have been fortunate to have wonderful friends and colleagues who deserve a special mention. Many thanks to my office-mates Evripidis Paraskevas, Ladan Rabiee, Yuchen Zhou, Peixin Gao and Lida Apergi for their unconditional support, honest feedback, and heartfelt friendship. Thanks to Aneesh Raghavan, Bhaskar Ramasubramanian, and Van Sy Mai for our hours-long discussions which have enriched my knowledge and strengthened my understanding. Thanks to Lars Lindemann and Vahid Mamduhi for making my visit memorable at KTH Royal Institute Technology.

My days at College Park would not have been more enjoyable without Udit, Soumyadip, Soham, Jiaul and Debdipta. Many thanks for making our apartment a "home" and providing me a "family" outside my family.

This would have not been possible without my family: my mother, my father and my sister for being my motivation, my inspiration, and my support throughout the highs and lows of my journey. Their constant encouragement and faith in me have made this journey smooth and seamless. Words, by no means, can express the gratitude I owe them.

Table of Contents

List of Figures

Chapter 1: Introduction

In recent years, systems have become more complex, distributed and interconnected. Moreover, the controllers for such systems are required to be designed for *higher level task descriptions* –which often times require satisfaction of logical constraints and complex decision making. With this increasing complexity and sophistication of the systems, new metrics for the design of the controllers are considered. These new metrics account for communication, sensing and computing overhead associated with the computation of the feedback control laws; and these metrics are much different from the classical $\mathcal{H}_2$ or $\mathcal{H}_\infty$ performance criteria. Although controller synthesis problems in the past have addressed instances with boolean decision variables, switched or hybrid systems, however, representing certain higher level specifications require special forms and tools of mathematics. Some of these specifications can be represented and formally verified by using *temporal logic*[1] [53]. Similarly, efforts have been made in reducing the communication and computational overhead associated with controller synthesis by using techniques such as low (or variable) precision quantization [100]. Recently, the focus has been shifted towards reducing the frequency of sensing rather than the quality of the sensed measure-

[1]Temporal logic was invented in the domain of computer science for purpose of automated formal testing and verification [78]

ments to deal with limited network bandwidth. These techniques adaptively change their sampling instances to assist the controller to ensure the performance criteria.

With the emergence of new topics such as Cyber-Physical-Systems(CPSs) or Internet-of-Things(IoT), systems are becoming more heterogeneous and resource constrained; and therefore they require systematic and careful modification to the existing theories in order to address the challenges and requirements in those systems. Needless to mention that these systems are expected to perform complex tasks with minimal information exchange so that communication bandwidth and sensing energy are not overused.

Information has manifold interpretations in different fields of studies. By information in this thesis we primarily mean the knowledge of state (or output) measurements, and by information constraints we mean the limited access of such information to the controller.

1.1 CPS and Emergence of Logical Constraints

Cyber-Physical Systems span a broad class of systems with multiple physical components –generally known as the plants– which are interconnected through certain cyber components. As CPSs encompass large number of systems, it is difficult to mathematically generalize the requirements and objectives associated with such systems. However, in many systems of our interests such as Robotics [15], Traffic Network [23], etc. logical decision making among the subsystems and temporal constraints on the actions of the subsystems are crucial factors in feasibility and optimality of the given task specification.

In control, the plants are generally represented with dynamical systems where the flow of the certain state (signal) is controlled by some actuator input. The logical constraints are used to evaluate certain higher level specifications of these signals; for example *signal* $x(t)$ *should reach the value* 5 *before* $x(t)$ *becomes negative.* Such a specification can be represented using the temporal logic as follows $(x(t) \geq 0)\mathbf{U}(x \geq 5)$, where $\mathbf{U}$ is the `Until` operator.

1.1.1 Temporal logic and dynamical systems

The behavior of the overall system is characterized by the evolution of the state trajectory of the underlying dynamical system associated with the CPS. Therefore, the temporal logical specifications are imposed on the trajectory of the states. The trajectories are controlled by operating the actuators and consequently, the control inputs are to be designed such that the trajectories are ensured to satisfy the temporal logical constraints.

Temporal logic was designed primarily with the focus of verifying temporal properties of Finite State Machines (FSMs) or systems with discrete states and discrete controlled transition from one state to another. However, a typical dynamical system in control have continuous state and control space. Therefore, initial techniques on this field relied on the abstracting a dynamical system into a finite state machine [40], [1], where the transitions are made by certain choice of particular control actions. The complexity of resulting FSM of a dynamical system is proportional to the accuracy of the abstraction. The constraint in constructing the FSM is that the original system should be able to mimic the behavior of the FSM. Finally, the temporal logical specification is evaluated on the FSM.

1.2 Information Constraints and Intermittent Feedback

Another aspect as highlighted in the beginning is the constraint in the communication and sensing resources in a CPS. In many cases, when the state signal does not vary much for an interval of time and one may reduce the amount of sensing and transmission of the sensed signal to the control with barely any degradation in the performance. Thus intermittent sampling may lead to significant reduction in the communication resources: especially in a CPS with multiple agents which incurs a huge amount of the intra-system communication. Periodic sampling possibly would be most intuitive approach towards an intermittent feedback controller synthesis. However, finding a suitable period is a challenging task [7]. In the recent years event-triggered and self-triggered control have gained popularity in this aspect where the system adaptive selects the sampling instances based on the performance criteria.

1.2.1 Event-triggered controller synthesis

In event-triggered control the sampling instances of state measurements are computed adaptively. The event-triggered system is composed of two parts: a controller and an event-generator [7]. The event generator determines the sampling instances which are also known as the ‘events’ based on the evolution a certain event signal (a function of the state signal).

Synthesis of an event-trigger system is a co-design of the controller and the event generator. The controller depends on the frequency of the events generated by the event-generator and the events are generated based on the the quality of some signal which depends on the control input. Most often, the event-generator follows a threshold based policy, that is, an sampling is done (and immediately sent to the controller) when some signal crosses a threshold. Also, to maintain simplicity, the controllers are chosen to be simple zero order hold [29], [28].

1.3 A framework to combine Temporal logic and event-triggered constraints for dynamical systems

Since in a CPS, the two types of constraints –logical and information– may arrive simultaneously, efforts must be taken to combine these two framework. However, the challenge lies in the fragile behavior of a logical constraint. Consider the example in Fig. 1.1 where in both the cases the continuous feedback trajectory (black curve with arrowhead) satisfies the task and hence the logical constraint evaluates to be `true`. Event-triggered trajectories generally deviate from the continuous feedback trajectory. Therefore, if we consider an ϵ tube around the continuous feedback trajectory as shown in Fig. 1.1, we see that not all trajectories (consider the red dotted line) can satisfy the logical task of eventually reaching A. Therefore, combining event-trigger framework with a temporal logical constrained problem is non-trivial.

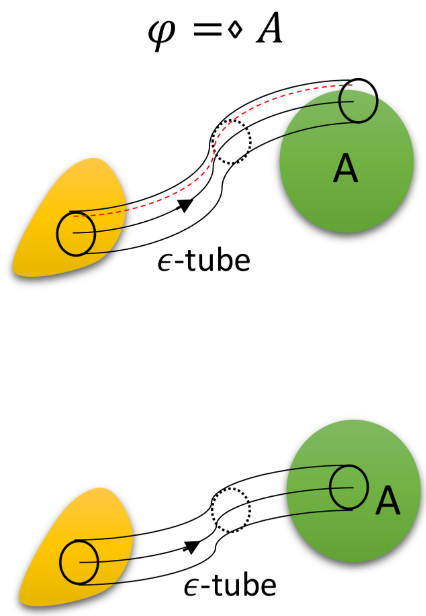


Figure 1.1: Task: trajectory should reach A eventually

However, this issue can be addressed by robustifying the logical constraints [37].

1.4 Contributions of the Thesis

The first problem of this thesis addresses the event-triggered controller synthesis problem of a nondeterministic linear system. The performance criterion is to mimic the continuous feedback trajectory with ϵ-precision using the event-triggered controller. Although, threshold based policies are ubiquitous in event-trigger literature, however, in this chapter we formally show that the threshold based policies are optimal for these systems. Also, we study the optimal controller associated with the threshold based optimal event-generator. We extend our results from finite time horizon to infinite time horizon as well. This is presented in **Chapter 2.** Relevant Publication:

D. Maity and John S. Baras, "Optimal Event-triggered Control of Non-deterministic

Linear Systems", IEEE Transactions on Automatic Control, (under Review).

In the next problem we focus on the class of control-affine nonlinear systems. The objective is similar to Chapter 1 in the sense that the event-triggered trajectory should mimic the continuous feedback trajectory with any given precision. We adopt a Lyapunov function based technique for the design of our threshold-based event-generator. For this problem, to retain simplicity, we adhere to the practice of choosing the controller from the zero-order-hold class. This is the content of **Chapter 3** of this thesis. Relevant Publication:

D. Maity and J. S. Baras, "Event Based control of Nonlinear Systems: A Lyapunov Function Based Approach", IEEE Conference on Decision and Control, 2015,

So far we have assumed the disturbance to be non-deterministic but its statistics are unknown. In the next segment we consider noise with known distributions (precisely, Wiener noise). We focus on the nonlinear stochastic optimal control problems with intermittent feedback. The results of this chapter is connected with the well celebrated Hamilton-Jacobi-Bellman (HJB) equations. We show an intermittent feedback controller might be designed from the corresponding continuous feedback controller. The results are presented in **Chapter 4**.

In **Chapter 5** we study an intermittent feedback based controller synthesis problem for multi-agent systems. We show that for Linear-Quadratic-Gaussian systems, the sub-game-perfect equilibrium (SPE) control policy for the agents does not depend on the event-triggering instance. Moreover, the SPE control policy and the SPE sampling policy can be found by solving two decoupled dynamic programming problems. Relevant Publication:

D. Maity, A. Anastasopoulos, and J. S. Baras, "Linear Quadratic Games with Costly Measurements", IEEE Conference on Decision and Control, 2017.

Next we study the controller synthesis problem under temporal logic specification in a robotics motion planning framework. We propose an automaton abstraction of the dynamics of the robot based on reachability aspects. We also focus on addressing finite time temporal logic tasks where we have time bounds on executing the tasks. This is presented in **chapter 6**. relevant publication:

D. Maity and J. S. Baras, "Motion Planning in Dynamic Environment with Bounded Time Temporal Logic Specifications", Mediterranean Conference in Control and Automation, IEEE, 2015.

In **Chapter 7** we propose a framework to combine the temporal logic constraints and information constraints in the controller synthesis. Our approach is compositional in nature, i.e., firstly a continuous feedback controller is designed using the method proposed in Chapter 6 and subsequently that controller is replaced with an event-triggered one using the results presented in Chapter 3. Furthermore, we address the effects of delay in the measurement transmission as well. Relevant Publication:

D. Maity and John S. Baras, "Event-Triggered Controller Synthesis for Dynamical Systems with Temporal Logic Constraints", American Control Conference, IEEE, 2018

Chapter 8 of the thesis also deals with jointly addressing the logical and information constraints. While in Chapter 7 we took an abstraction based approach, in this chapter we do not abstract the dynamics into an Finite State Machine (FSM) rather we adopt prescribed performance based control approach. Hence, in one hand, this method does not suffer from computational complexity issues associated with the increasing size of the FSM, however on the other hand, the class of temporal logical tasks that can be addressed in this framework is restricted compared to Chapter 7. However, the design of the event-trigger strategy is similar to Chapter 3, i.e., a continuous feedback control is approximated by a zero-order-hold controller and threshold-based event-generator. Relevant publication:

D. Maity, L. Lindeman, John S. Baras, and Dimos V. Dimarogonas "Event-triggered Feedback Control for Signal Temporal Logic Tasks", IEEE Conference on Decision and Control, 2018.

Chapter 2: Part I

Controller Synthesis under Resource Constraints: Event-triggered Control Approach

In this chapter, an event-triggered controller is sought to replace the continuous feedback policy with an intermittent feedback one for a non-deterministic linear system. The non-determinism arises due to the presence of an exogenous disturbance in the dynamics of the system. An event-triggered framework communicates the measurement to the controller only at certain discrete time instances which are generated by an event-generator. The objective of this new control architecture is to synthesize an optimal event-generator and controller such that the trajectory of the states of the event-triggered system mimics the same of the feedback system with prescribed precision. The optimality is in the sense that the least number of state measurements are sent to the controller.

The results of this chapter show that such an optimal event-triggered controller retains the linear structure when the continuous feedback controller is linear; and the optimal event-generator follows a threshold based policy, where the event-generator decides to send the state measurement to the controller every time a certain signal exceeds a certain threshold.

We start with a finite horizon design problem and the extend the results for infinite horizon controller synthesis. The structural properties of the optimal event-triggered controller and event-generator remain unchanged when extended to an infinite horizon.

2.1 Introduction

Consider the generic linear non-deterministic control system in $\mathbb{R}^n$ as given in (2.1).

$$\dot{x} = Ax + Bu + d$$
$$x(0) = x_0. \tag{2.1}$$

where d is a non-deterministic exogenous disturbance acting on the system. The initial state $x(0)$ is known.

Given a control law $u(t) = K(t, x(t))$, control of such a system described in (2.1) generally requires continuously reading the sensor measurements, transmitting it to the controller. In a centralized system, the performance depends on the continuity of the communication, and computing the control signal accurately. In a distributed system, although the controller is implemented distributively, however, it also requires continuous interaction among the subsystems. Sensing, communication and data handling are indispensable parts for networked control systems. As a result, their performance is generally determined by the available resources to perform sensing, transmission, and computation. Scarcity of such resources are the sources of performance degradation for large and interconnected systems.

In the recent past, researchers have proposed novel techniques to approximate the control law $u = K(t, x(t))$, for system (2.1), in such a way that requires only finite number of transmission (i.e. discrete-time transmission) to overcome the problem with continuous sensing, continuous transmitting and continuous computing [6] [44], [5]. Event-based control has been proved to be very effective in dealing with limited resources such as transmission bandwidth, sensing energy and computational resources. Clearly such an approximate control signal will only lead to a behavior of the state trajectory which is approximate to the trajectory obtained from continuous feedback. Such control scheme generally has two components: the *Controller* and the *Event-Generator*. The event-generator decides the discrete time instances when the state measurement is to be transmitted, and the controller computes the control based on the received measurements from the event-generator.

A great deal of research has been performed in the last few decades to improvise such frameworks and extend them to non-linear and stochastic systems. In [7], a comparison between the performance of event based control and periodic-sampling based control has shown that under some conditions the event based control performs better than periodic control. A simple PID controller is proposed in [5] for event based control which reduces large CPU computation at the cost of minor control performance degradation. The supremacy of event-based strategy over a periodic-sampling strategy is not only in reducing the number of transmission when there is not much variation in the measurements, but also in increased transmission when there is rapid variation. In periodic sampling, the challenge is to find the suitable period of transmission to guarantee a certain level of performance. In the recent studies, the foci mostly have been on finding a feasible controller and a compatible event-generator that together can approximate the continuous feedback trajectory with arbitrary precision. The aim of this chapter is to identify the optimal controller and event-generator pair which minimizes the total number of measurement transmissions, hence entailing minimalistic energy, bandwidth and computation resources.

In event-based control, self-triggered control or periodic control, the controller being unable to access the continuous state, it estimates the state and the estimated state is used to produce the control input. Since the generated control input is different from the actual (continuous) feedback input, the response of the system is not as it would have been if there were a continuous state feedback. Let the state of the continuous feedback system be denoted as $x_c(t)$ and the state of the event-based system as $x(t)$. The signal $e(t) = x(t) - x_c(t)$ denotes the deviation in trajectories for using the event-based controller. For a given ϵ, the aim is to find an event-generator and a corresponding controller such that $\|e(t)\| \leq \epsilon$ for all t with minimum transmission. The framework of our work is similar to [64] and [66], but none of them addresses the question of optimality. The framework is schematically represented in Figure 2.1 where the event-generator determines the triggering instances and consequently sends the state information to the controller through the switched communication link. The system is influenced by the exogenous disturbance $d(t)$. In [64], for a similar problem (without the optimality criterion) it was assumed that the closed loop plant dynamics with linear feedback is asymptotically stable (i.e.

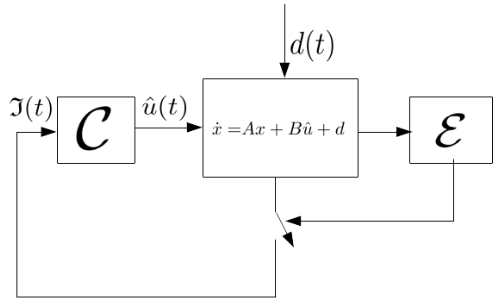


Figure 2.1: Event based control loop with three subsystems: control input generator, the plant and the event generator. The communication link is shown in dashed line which allows communication only in discrete manner.

$A - BK$ is Hurwitz) and in this chapter no such assumption is made. Similar to [64], we restrict ourselves to design event-based controller to replace the continuous controller which are linear feedback i.e $K(t, x(t)) = K(t)x(t)$. If the system is not asymptotically stable, the residual error, $e(t_k)$, after the k-th triggering persists, and for unstable systems it might increases exponentially. The proposed approach assures that the control input can be designed (by introducing an 'corrective' control component $\psi(t)$) in such a way that can mitigate this residual error.

The main contributions of this chapter are:

First, the optimal structure of the event-triggered controller is derived and it is found to be only dependent on the latest state information, and not on all the prior measurements. Thus, the controller does not need any (extra) memory (except latest measurement) to implement the control law. Further, it is found that the optimal controller is linear with respect to the latest state measurement. Moreover, we show uniqueness (and existence) of such optimal controller.

Second, we show that the controllability of the system is sufficient to ensure (by constructing additional corrective control) that there exists a event-based controller and an event-generator so that the norm of the error, $\|e(t)\|$, can be bounded by any given positive constant ϵ for all t. Thus, our approach is applicable to those systems where the closed-loop system is not Hurwitz; hence extending the applicability of event-triggered controllers to the class of problems that are not readily handled by existing techniques such as [64].

Third, we design our event-triggering mechanism that minimizes the total number of triggering under the worst case disturbance $(d(\cdot))$. It is shown by the study that such an event-triggering mechanism has a threshold based policy. This policy is found by solving certain dynamic programming, and the policy is unique. Such a triggering policy does not exhibit Zeno behavior, i.e., inter-triggering duration has a finite positive lower bound.

The rest of the chapter is organized as follows: Section 2.1.1 performs a comprehensive literature survey; Section 2.3 formulates the general problem that will be addressed; Section 2.4 provides the optimal controller synthesis; Section 2.5 describes the optimal event-triggering strategy; and Section 2.7 illustrates the framework with examples. Finally we conclude our chapter with a discussion in Section 2.8.

2.1.1 Literature Review

Event based controller synthesis is a well studied topic in control for more than a decade or so, and the literature is vast and enriched with various aspects of event-based frameworks on many different systems. This section provides a brief and concise representation of these works.

In present literature, various seemingly similar frameworks have been studied to reduce the communication overhead e.g. Event-Based control [8], self-triggered control [94], [59] and periodic-time control [75], [18], [106]. The essence behind these techniques is to transmit the measurements at discrete time instances rather than communicating continuously. Even comparisons among such methodologies have been performed to judge the effectiveness, see for example [7], [8]. In the literature, asynchronous control [42], event-based sampling [6], event-driven sampling [44], Lebesgue sampling [8], deadband sampling [76] have been proposed to carry out the idea of supplicating communication only when some event has been occurred. In [45], the authors analyzed event-based control in a stochastic setting. [25], [26], [101], took a different approach to reduce the information content rather than reducing the communication frequency. A state feedback approach for an event based system is considered in [64] where the feedback control is generated from another system which is updated every time a trigger is introduced. Later this framework was extended to consider the output feedback scenario in [57]. Output feedback based decentralized event-trigger control is also consider in [31], [104]. [104] studies the problem of multi-agent consensus in an output feedback event-based framework. Event-based control for distributed interconnected linear systems is proposed in [29] and [86].

In event-triggered framework, there are some works which aim to directly optimize certain cost function rather than penalizing the actual trajectory deviation. Such an event-based control of the standard LQG problem in lossy channel is studied in [73] and proposed a sub-optimal solution to it. An event triggered state estimation has been the focus of study in [74].

In parallel, event-based control for non-linear systems have been explored in the recent past. Asymptotic stability of an event-triggered non-linear system is studied in [95] with the assumption of input-state-stability. [91] studies a real-time scheduling and stabilizing control task under event-triggered framework. This work is extended for homogeneous and polynomial nonlinear systems in [3]. A feedback linearization based approach was taken in [90] to design an event-based controller for nonlinear systems. The nonlinear counterpart of [66] is investigated in [65] using a Lyapunov function based approach. An Lyapunov function based approach was also taken in [97] to study non-linear event based systems under delay and packet drop-outs.

2.2 Notation

$x(t)$: state of the event based system at time t, $x_c(t)$: state of the equivalent continuous feedback system at t, $e(t) = x(t) - x_c(t)$: error in the state trajectory. $d : \mathbb{R}_+ \to \mathbb{R}^n$: the exogenous disturbance, t_i: the i-th triggering instance, $x(t_i)$: value of the state at i-th triggering instance, $\theta(t)$: the latest triggering instance before time t, $N(t)$: number of measurements sent until time t, $\mathfrak{I}(t) = \{x(t_i)\}_{i=0}^{N(t)}$ (or $\{x(t_i)\}_{i=0}^{N(t)} \cup \{e(t_i)\}_{i=0}^{N(t)}$): the information available to controller at time t, $\mathcal{K}(t, \mathfrak{I}(t))$: the generic structure of the controller, $\mathcal{C}$: event-triggered controller, $\mathcal{E}$: event generator, $u(t)$: continuous feedback input, $\hat{u}(t)$: event-triggered (intermittent feedback) input, J: cost function of total number of transmissions, $\Phi(t,s), \tilde{\Phi}(t,s)$:

state transition matrices, $\|\cdot\|$: a norm in $\mathbb{R}^n$.

2.3 Problem Formulation

Let us consider the linear non-deterministic dynamics of a system to be given by (2.2)

$$\begin{aligned} \dot{x} =& Ax + Bu + d \\ x(0) =& x_0 \end{aligned} \tag{2.2}$$

where for all t, $x(t) \in \mathbb{R}^n$ is the state of the system and $u(t)$ is the control. $d(t)$ is an n dimensional exogenous disturbance to the system which is non-deterministic. We assume that $d : [0, \infty) \to \mathbb{R}^n$ is an Lebesgue integrable function so that (2.2) has a well defined solution for all $t \geq 0$; we also assume $d(\cdot) \in \mathcal{D}$ for some set $\mathcal{D}$ (in general, $\mathcal{D}$ is the set of Lebesgue integrable functions, sometimes we assume $\mathcal{D} = \mathcal{L}^\infty([0, T])$ i.e. the set of bounded Lebesgue integrable functions).

Let a feedback control $u(t) = -Kx(t)$ has been designed to achieve some desirable behavior on the trajectory $x(\cdot)$. In the absence of d, it is sufficient to know only the initial state x_0 to calculate $u(t)$ for all t; however, the presence of d makes it absolutely necessary to know $x(t)$ in order to calculate $u(t)$ precisely.

The closed-loop continuous feedback system has the state dynamics (2.3)

$$\dot{x}_c = \tilde{A}x_c + d \tag{2.3}$$

where $\tilde{A} = A - BK$.

The communication of the state measurement to the controller is done in a discrete time manner and on demand. Due to the availability of discrete measurements, $\{x(t_i)\}_{i\in\mathbb{N}}$, as opposed to continuous measurements $\{x(s)\}_{s\geq 0}$, the computation of the control $u(t)$ will not be accurate and hence the trajectory $x(\cdot)$ will deviate from its desired trajectory $x_c(\cdot)$. In this chapter our constraint on the controller $\mathcal{C}$ and event-generator $\mathcal{E}$ is to ensure $\|x(t) - x_c(t)\| \leq \epsilon$ for all t and for all realization of the disturbance $d(\cdot)$.

In this framework, since the communication is done in a discrete time manner, the exact state of the system, $x(t)$, is available to the controller only at those time instances, t_i. Let at any time t, $\theta(t)$ denote the latest instance ($t_i \leq t$) when the state value $(x(t_i))$ was communicated to the controller. Therefore $\theta(t)$ is a piecewise constant function and $\theta(t) \leq t$ where the equality holds at the triggering instances.

Therefore, the control has to be designed in a way such that it does not require the continuous measurement of the state of the system and, nonetheless, it drives the new system to approximate the closed loop system (2.3) within the given tolerance bound for any realization of the disturbance d. Let $\hat{u}(t) = \mathcal{K}(t, \{x(t_i)\}_{i=1}^{N(t)})$ where $t_{N(t)} = \theta(t)$; $N(t)$ denotes the total number of measurements sent until time t. The new system with $\hat{u}$ as control input has the dynamics

$$\dot{x} = Ax + B\hat{u} + d \tag{2.4}$$

$$x(0) = x_0.$$

The deviation of $x(t)$ from $x_c(t)$ will depend on the choice of $\{t_i\}_{i\geq 1}$, $N(t)$ and $\mathcal{K}(\cdot,\cdot)$; and these are our optimization variables. We divide these variables into two groups $\mathcal{E} = \{N(t), \{t_i\}\}$, and $\mathcal{C} = \mathcal{K}(\cdot,\cdot)$, where $\mathcal{E}$ will be referred as the event-generator which will decide the sampling instance t_i and send $x(t_i)$ to the controller, and $\mathcal{C}$ will be named as the event-triggered controller which will generate the input $\hat{u}(t)$ based on the measurements sent by $\mathcal{E}$.

Formally, we define that the control input is given by

$$\hat{u}(t) = \mathcal{K}\Big(t, \mathfrak{I}(t)\Big) \tag{2.5}$$

where from our previous discussion, $\mathfrak{I}(t) = \{x(t_i)\}_{i=1}^{N(t)}$, and will be called as the *information* available to $\mathcal{C}$ at time t. Later in Section 2.4, we will notice that for systems with non-Hurwitz $\tilde{A}$, the controller $\mathcal{C}$ needs more *information* than the state value $x(t_i)$ at the triggering instances to ensure $\|e\| \leq \epsilon$. In fact, we will notice that $\mathcal{E}$ needs to send the pair $(x(t_i), e(t_i))$ to $\mathcal{C}$ at each triggering instance t_i. Therefore, in that case $\mathfrak{I}(t) = \big\{\{x(t_i)\}_{i=1}^{N(t)}, \{e(t_i)\}_{i=1}^{N(t)}\big\}$.

The event-generator $\mathcal{E}$ is attached to the plant and makes its decision at time t based on the information $\mathcal{F}_t = \{x(s)\}_{0\leq s\leq t}$.

It is straightforward to notice that, the more measurements are acquired, the 'closer' $x(t)$ will be to $x_c(t)$. For given $T, and \epsilon > 0$, the requirement is to design $(\mathcal{E}, \mathcal{C})$ pair such that $\sup_{t\in[0,T]} \|x_c(t) - x(t)\| \leq \epsilon$ for every realization of the disturbance $d(\cdot)$ while minimizing the number of measurements sent by $\mathcal{E}$ to $\mathcal{C}$.

First, we will solve this problem for a finite horizon $[0, T]$ and later we take

$T \to \infty$ to study the infinite horizon behavior. Therefore for a finite horizon $[0, T]$, formally:

Problem 2.3.1. *For any given $\epsilon > 0$,*

$$\inf_{\mathcal{E},\mathcal{C}} \sup_{d(\cdot)\in\mathcal{D}} N(T) \tag{2.6}$$

$$s.t. \quad \sup_{t\in[0,T]} \|x_c(t) - x(t)\| \le \epsilon \qquad \forall d(\cdot) \in \mathcal{D}. \tag{2.7}$$

For an infinite horizon problem, we make the following assumptions that $\lim_{t\to\infty} \|x_c(t)\| < +\infty$, and $\int_0^\infty \|d(t)\| dt < +\infty$.

We consider the following problem (the formulation ensures the optimization problem attains a finite value when there is a feasible event-based $\mathcal{E}$ and $\mathcal{C}$):

Problem 2.3.2. *For any given $\epsilon > 0$, and some $T < +\infty$*

$$J_T^* = \inf_{\mathcal{E},\mathcal{C}} \sup_{d(\cdot)\in\mathcal{D}} \sum_{i=1}^{N(T)} e^{-t_i} \tag{2.8}$$

$$s.t. \quad \sup_{t\in[0,T]} \|x_c(t) - x(t)\| \le \epsilon, \qquad \forall d(\cdot) \in D. \tag{2.9}$$

Let $\mathcal{E}_T^*$ and $\mathcal{C}_T^*$ be the solution of the finite horizon problem with cost J_T^*. Then for infinite horizon:

$$J_\infty^* = \limsup_{T\to\infty} J_T^*, \tag{2.10}$$

$$\mathcal{E}_\infty^* = \limsup_{T\to\infty} \mathcal{E}_T^*, \tag{2.11}$$

$$\mathcal{C}_\infty^* = \limsup_{T\to\infty} \mathcal{C}_T^*. \tag{2.12}$$

We assume that problems 2.3.1 and 2.3.2 are feasible, i.e., for each problem there exists a pair $(\mathcal{E}, \mathcal{C})$ such that the problem has a finite value at the optimum.

In event-triggered control, the design of $\mathcal{E}$ needs to satisfy non-Zeno behavior, i.e., there should not be infinite number of state transmission within any finite time interval. Note that both problem formulation incurs $+\infty$ cost for any $\mathcal{E}$ with Zeno behavior.

2.3.1 Assumptions

The following assumptions are carried out throughout the chapter:

1) The communication link through which the event-generator sends the information to the controller is delay-free, noiseless, and none of the packets are dropped out.

2) The system parameters (A, B, K) are assumed to be time invariant.

3) The initial condition $x(0)$ is deterministic and the initial information to the controller is $\mathfrak{I}(0) = \{x(0)\}$.

2.4 Optimal Controller Synthesis

In this section we devout our attention to the effects of the controller $\mathcal{C}$ on the error signal $e(t)$.

For simplicity, we restrict ourselves to the space of control strategies which are affine in the latest measurement, i.e.

$$\hat{u}(t) = \mathcal{K}\Big(t, \{x(t_i)\}_{i=1}^{n(t)}\Big) = L(t)x(\theta(t)) + \psi(t) \tag{2.13}$$

where $L(t)$ and $\psi(t)$ characterize $\mathcal{K}(t, \cdot)$, and $\psi(t)$ does not depend on $x(\theta(t))$. Later we will remove this assumption and consider a general controller as proposed in (2.5) and show that the affinity assumption does not lose generality (see Theorem 2.4.6).

Let us note that with the control given in (2.13), the state dynamics evolve as:

$$\dot{x}(t) = Ax(t) + BL(t)x(\theta(t)) + B\psi(t) + d(t) \tag{2.14}$$

and the error $e = x - x_c$ has the dynamics

$$\begin{aligned}&\dot{e}(t) = \tilde{A}e(t) + BKx(t) + BL(t)x(\theta(t)) + B\psi(t)\\&e(0) = 0\end{aligned}$$

We will occasionally suppress the time argument, to maintain brevity, whenever the context is not ambiguous.

From (2.14), we can write:

$$x(t) = F(t)x(\theta(t)) + \int_{\theta(t)}^{t} \Phi(t, s)(B\psi(s) + d(s))ds$$

where

$$F(t) = \Phi(t, \theta(t)) + \int_{\theta(t)}^{t} \Phi(t, s) BL ds.$$

$\Phi(t, s)$ is the state transition matrix corresponding to drift matrix A, i.e. $\frac{\partial \Phi(t,s)}{\partial t} = A\Phi(t, s)$ and $\Phi(s, s) = I$ for all s, t; for the time invariant case, $\Phi(t, s) = e^{A(t-s)}$.

Thus,

$$\dot{e} = \tilde{A}e + B(KF + L)x(\theta(t)) + B\phi + d_1 \tag{2.15}$$

where $\phi(t) = \psi(t) + K \int_{\theta(t)}^{t} \Phi(t, s)B\psi(s)ds$ and $d_1(t) = BK \int_{\theta(t)}^{t} \Phi(t, s)d(s)ds$.

Solving the linear dynamics of e, we can write:

$$e(t) = \tilde{\Phi}(t, \theta(t))e(\theta(t)) + \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds + f(t)$$

where $\tilde{\Phi}(t, s) = e^{\tilde{A}(t-s)}$. By using the fact that for all $s \in [\theta(t), t]$, $\theta(s) = \theta(t)$, we obtain

$$f(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)B\big((KF(s) + L(s))x(\theta(t)) + \phi(s)\big)ds.$$

From the expression of e in (2.15),

$$\sup_{d \in \mathcal{D}} \|e(t)\| = \sup_{d \in \mathcal{D}} \left\| \tilde{\Phi}(t, \theta(t))e(\theta(t)) + f(t) + \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds \right\|.$$

One can note that $f(t)$ is the only part in $e(t)$ that does not depend on the disturbance d. $\tilde{\Phi}(t, \theta(t))e(\theta(t))$ depends on the disturbance until time $\theta(t)$ and the effect of the disturbance for the interval $[\theta(t), t]$ is only in the term $\int_{\theta(t)}^{t} \tilde{\Phi}(t,s)d_1(s)ds$. Since the disturbance signal could be any Lebesgue integrable function, and the controller wants to minimize the $\sup_d \|e(t)\|$, the optimal choice for $\mathcal{C}$ would be to minimize $\|f(t)\|$. Thus, in this case, the necessary and sufficient condition for an optimal $f(\cdot)$ is $f(t) = 0$ for all t. Since $\phi(s)$ does not depend on $x(\theta(t))$, then $f(t) \equiv 0$ is equivalent to:

$$KF(t) + L(t) = 0 \tag{2.16}$$

$$\phi(t) = 0$$

for all t. The following lemma characterizes the $L(t)$ that is able to satisfy (2.16).

Lemma 2.4.1. $L(t) = -K\tilde{\Phi}(t, \theta(t))$ *satisfies* $KF(t) + L(t) = 0$.

Proof. Let us substitute $L(t) = -K\tilde{\Phi}(t, \theta(t))$ in the expression on $F(t)$:

$$\begin{aligned} F(t) &= \Phi(t, \theta(t)) - \int_{\theta(t)}^{t} \Phi(t,s)BK\tilde{\Phi}(s, \theta(t))ds \\ &= \Phi(t, \theta(t)) + \int_{\theta(t)}^{t} \frac{d}{ds}(\Phi(t,s)\tilde{\Phi}(s, \theta(t)))ds \\ &= \tilde{\Phi}(t, \theta(t)). \end{aligned}$$

Hence, $KF(t) + L(t) = 0$ for all t. □

The following theorem characterizes the optimal controller structure and its behavior:

Theorem 2.4.2. *Under the event-triggering scheme where the only information sent by $\mathcal{E}$ to $\mathcal{C}$ is the sampled state value $x(t_i)$, and $\mathcal{C}$ uses an affine controller as given in (2.13), the optimal controller that tries to ensure $\sup_{d\in\mathcal{D}}\sup_{t\in[0,T]}\|e(t)\| \leq \epsilon$ has the following structure:*

$$\hat{u}(t) = -Kx_d \tag{2.17}$$

where for all $t \in [t_i, t_{i+1})$

$$\dot{x}_d = \tilde{A}x_d$$

$$x_d(t_i) = x(t_i)$$

Proof. The optimal controller will have the structure that satisfies the necessary conditions (2.16). Hence, $\psi(\cdot) \equiv 0$ and using Lemma 2.4.1 $L(t) = -K\tilde{\Phi}(t, \theta(t))$.

The theorem is proved by noting that $x_d(t) = \tilde{\Phi}(t, \theta(t))x(\theta(t))$ □

This structure for the controller, (2.17), was assumed without justification for optimality in earlier works [64], [66].

Remark 2.4.3. *Comparing the dynamics of x_c and x_d, we notice a 'certainty-equivalence' type property in the controller structure; i.e. x_d is an (worst case) estimate of x_c and the control $\hat{u}$ replaces $x_c(t)$ with that estimate.*

Therefore, using the optimal controller as described in Theorem 2.4.2,

$$\dot{e} = \tilde{A}e + d_1. \tag{2.18}$$

The stability and boundedness of e is totally determined by the matrix $\tilde{A}$ and the disturbance d_1 (or equivalently d), and not by any parameter of the controller $\mathcal{C}$ and $\mathcal{E}$. The vast majority of the past work is based on the assumption that $\tilde{A}$ is Hurwitz and $\mathcal{D} \subseteq \mathcal{L}^\infty([0,T])$. Since $\mathcal{D} \subseteq \mathcal{L}^\infty([0,T])$, therefore $\|d(t)\| \leq \bar{d}$ almost everywhere for some $\bar{d} \geq 0$. Therefore, with these assumptions, $\|d_1(t)\| \leq \bar{d}\|BK\| \int_{\theta(t)}^{t} \|\Phi(t,s)\| ds$.

Thus, it is trivially true that:

$$\|e(t)\| \leq \sup_t \|d_1(t)\| \int_0^t \tilde{\Phi}(t,s) ds \leq \beta \sup_t \|d_1(t)\|$$

where $\beta = \int_0^\infty \tilde{\Phi}(t,s) ds < \infty$. Maintaining an event-triggering scheme such that $\sup_t \|d_1(t)\| \leq \epsilon/\beta$ will ensure $\|e(t)\| \leq \epsilon$. By noticing that $\|d_1(t)\| \leq \bar{d}\|BK\| \int_{\theta(t)}^{t} \|\Phi(t,s)\| ds$, it is sufficient to ensure $\int_{\theta(t)}^{t} \|\Phi(t,s)\| ds \leq \epsilon/(\beta\bar{d}\|BK\|)$. Therefore, under the assumptions of bounded noise and Hurwitz closed-loop system, the triggering instances can be computed offline by solving $\int_{\theta(t)}^{t} \|\Phi(t,s)\| ds \leq \epsilon/(\beta\bar{d}\|BK\|)$. A event-triggered control problem with these assumptions has been studied in [64], and the conclusion was alike.We want to address the problem when $\tilde{A}$ is not Hurwitz and/or $\mathcal{D} \not\subseteq \mathcal{L}^\infty([0,T])$.

To maintain the continuity of the analysis, let us assume that $\mathcal{E}$ can also measure $e(t)$ for all t, and at each triggering instance t_i, $\mathcal{E}$ sends the information $(x(t_i), e(t_i))$ to $\mathcal{C}$. For now, this is an assumption that $\mathcal{E}$ knows $e(t)$ since in practice $\mathcal{E}$ has only the knowledge of $\mathcal{F}_t$ ($\mathcal{F}_t = \{x(s)\}_{0 \leq s \leq t}$) and 'somehow' it has to compute $e(t)$. Later in this chapter we will show how $\mathcal{E}$ can compute $e(t)$ for all t.

Here we restrict ourselves (without loss of generality, see Theorem 2.4.6 for the general result) to the controller structures:

$$\hat{u}(t) = L(t)x(\theta(t)) + \psi(t, e(\theta(t))). \tag{2.19}$$

Using this control input, and after some simplifications

$$e(t) = g_1(t, e(\theta(t))) + g_2(t, x(\theta(t))) + \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) d_1(s) ds.$$

where

$$g_1(t,x) = \tilde{\Phi}(t, \theta(t))x + \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) B\phi(s,x).$$

$\phi(t,x) = \psi(t,x) + K \int_{\theta(t)}^{t} \Phi(t,s) B\psi(s,x) ds$ and

$$g_2(t,x) = \Big[\int_{\theta(t)}^{t} \Phi(t,s)(B(KF(s) + L(s))) ds \Big] x.$$

The requirement is finding controller $\mathcal{C}$ such that $g_1(t, e(\theta(t))) = 0$ and $g_2(t, x(\theta(t))) = 0$ (or minimize $\|g_1(t, e(\theta(t)))\|$, $\|g_2(t, x(\theta(t)))\|$).

Notice that $g_2(t, x(\theta(t)))$ can be made equal to 0 for all t, if $L(t) = -K\tilde{\Phi}(t, \theta(t))$ as stated in Lemma 2.4.1.

From the expression of $g_1(t,x)$, for a fixed x, one can verify that:

$$\dot{g}_1(t,x) = \tilde{A}g_1(t,x) + B\phi(t,x) \tag{2.20}$$
$$g_1(\theta(t),x) = x$$

Therefore, $g_1(t,x)$ has a linear dynamics with $\phi(t,x)$ acting as a control input to that system. Therefore making $g(t,x) = 0$ for all t, x, is equivalent of asking whether (A,B) is a controllable pair. Since we have freedom in selecting $\psi(t,x)$, we can control the value of $\phi(t,x)$ by properly selecting $\psi(t,x)$. The following theorem formally states how to bring $g(t,x)$ to 0 by proper choice of $\phi(t,x)$.

Theorem 2.4.4. *If (A,B) is a controllable pair, then there exists a $\phi(t,x)$ such that $g_1(t,x) = 0$ for all $t > \theta(t)$. Moreover, such a $\phi(t,x)$ is linear in x.*

Proof. Controllability of (A,B) implies controllability of $(\tilde{A},B)$. For a controllable time invariant linear system, the state is controllable to the zero state in arbitrarily small time. Therefore, $\forall x$ and $\forall r \in (\theta(t),t]$ $\exists \phi(\cdot,x) : [\theta(t),r] \to \mathbb{R}^m$ such that

$$0 = g_1(r,x) = \tilde{\Phi}(r,\theta(t))x + \int_{\theta(t)}^{r} \tilde{\Phi}(r,s)B\phi(s,x),$$

which implies such a $\phi(\cdot,x)$ is linear in x, i.e. $\phi(s,x) = \gamma(s)x$ for all s, x. One can verify that

$$\gamma(s) = \begin{cases} K\Phi(s,\theta(t))\big(I - a(s)W\big) - B'\Phi(\theta(t),s)'W & \forall s \in [\theta(t), \theta(t)+\delta] \\ 0 & s > \theta(t)+\delta \end{cases}$$

can ensure $g_1(r,x) = 0$ for all $r \geq \theta(t) + \delta$, where

$$a(s) = \int_{\theta(t)}^{s} \Phi(\theta(t),\sigma)BB'\Phi(\theta(t),\sigma)'d\sigma,$$
$$W(\delta) = [a\big(\theta(t)+\delta\big)]^{-1}. \tag{2.21}$$

Since δ can be made arbitrarily small, we can have $g_1(t,x) = 0$ for all $t > \theta(t)$. □

Although, δ could be made arbitrarily small in Theorem 2.4.4, we must note that the gain of the proposed controller in Theorem 2.4.4 depends on $W(\delta)$. The eigenvalues of $W(\delta)$ increases (arbitrary high) as $\delta \to 0$. Thus, from an implementation point of view, δ should have some finite positive value, even though theoretically δ can be arbitrarily small.

As soon as we set δ to have some finite positive value, we have to ensure within the period $[t_i, t_i + \delta)$ (t_i is any triggering instance) no triggering occurs. This could be trickier since we need to ensure $\sup_t \|e(t)\| \leq \epsilon$ while we have $\|e(t_i)\| = \epsilon$ and $d(t)$ can take any realization within the period $[t_i, t_i + \delta)$. Thus it might cause a Zeno effect in the triggering system. A more detailed discussion to tackle this implementation issue is presented in Section 2.8. For the analysis further, we will assume that δ is chosen arbitrarily small.

Controller $\mathcal{C}$ has the freedom to select $\psi(t,x)$ and not $\phi(t,x)$ directly. Therefore, we need to ensure that there exists a $\psi(t,x)$ for the proposed $\phi(t,x)$ in Theorem 2.4.4.

Proposition 2.4.5. *For all* $t \in [\theta(t), \theta(t)+\delta]$,

$$\psi(t,x) = (K\tilde{\Phi}(t,\theta(t)) - B'\Phi(\theta(t),t)'W)x$$

and $\psi(t,x) = K\tilde{\Phi}(t,\theta(t))x \ \forall t > \theta(t)+\delta$ *achieves the* $\phi(t,x)$ *in Theorem 2.4.4. Where* W *is defined in (2.21).*

Proof. We start by using the definition of $\phi(t,x)$:

$$\phi(t,x) = \psi(t,x) + K\int_{\theta(t)}^{t} \Phi(t,s)B\psi(s,x)ds,$$

and let us choose $\psi(t,x) = K\tilde{\Phi}(t,\theta(t))x + \psi_1(t,x)$. Thus,

$$\phi(t,x) = K\Phi(t,\theta(t))x + \psi_1(t,x) + K\int_{\theta(t)}^{t} \Phi(t,s)B\psi_1(s,x)ds.$$

Let us now select,

$$\psi_1(t,x) = \begin{cases} -B'\Phi(\theta(t),t)'Wx, & \theta(t)+\delta \geq t \geq \theta(t) \\ 0 & t > \theta(t)+\delta \end{cases}$$

Thus, one can verify that for all $t \in [\theta(t), \theta(t) + \delta]$,

$$\begin{aligned}\phi(t,x) =& K\Phi(t,\theta(t))x - B'\Phi(\theta(t),t)'Wx - K\int_{\theta(t)}^{t}\Phi(t,s)BB'\Phi(\theta(t),s)'Wxds \\ =& K\Phi(t,\theta(t))\left(I - a(t)W\right)x - B'\Phi(\theta(t),t)'Wx \\ =& \gamma(t)x\end{aligned}$$

Similarly, for $t > \theta(t) + \delta$,

$$\begin{aligned}\phi(t,x) =& K\left(\Phi(t,\theta(t)) - \int_{\theta(t)}^{\theta(t)+\delta}\Phi(t,s)BB'\Phi(\theta(t),s)'dsW\right)x \\ =& K\Phi(t,\theta(t))\left(I - a(\theta(t)+\delta)W\right)x = 0\end{aligned}$$

as desired. □

Therefore, the corrective control input can be expressed as compactly

$$\psi(t, e(\theta(t))) = K\tilde{\Phi}(t,\theta(t))e(\theta(t)) - 1_{t<\theta(t)+\delta}B'\Phi(\theta(t),t)'We(\theta(t))$$

where

$$1_{x<y} = \begin{cases} 1 & x < y \\ 0 & x \geq y \end{cases}$$

is an indicator function.

Thus, under the assumption that (A, B) is controllable, we have proved that for any $t \geq \theta(t) + \delta$,

$$e(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) d_1(s) ds,$$

and since $\delta > 0$ can be made arbitrarily small, we can conclude that for all $t > \theta(t)$

$$e(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) d_1(s) ds. \tag{2.22}$$

In the analysis so far, we have restricted ourselves to controller of the form:

$$\begin{aligned}\mathcal{K}(t, \mathfrak{I}(t)) =& \mathcal{K}\Big(t, \{x(t_i)\}_{i=1}^{n(t)}, \{e(t_i)\}_{i=1}^{n(t)}\Big) \\ =& L(t)x(\theta(t)) + M(t)e(\theta(t))\end{aligned}$$

In the next theorem, we show that the optimal controller is indeed of this form and the restriction does not lose generality.

Theorem 2.4.6. *If (A, B) is a controllable pair and the controller $\mathcal{C}$ has information $\mathfrak{I}(t) = (\{x(t_i)\}_{i=1}^{n(t)}, \{e(t_i)\}_{i=1}^{n(t)})$, then the optimal controller has the following linear form:*

$$\hat{u} = L(t)x(\theta(t)) + M(t)e(\theta(t))$$

where $L(t) = -K\tilde{\Phi}(t, \theta(t))$, and $M(t) = K\tilde{\Phi}(t, \theta(t)) - 1_{t \leq \theta(t)+\delta} B'\Phi(\theta(t), t)W(\delta)$. $W(\delta)$ is defined in (2.21).

Moreover, with this controller, $e(t)$ *can be controlled to have the value:*

$$e(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t, s) d_1(s) ds$$

for all $t > \theta(t)$*.*

Proof. The proof of this Theorem is presented in Appendix 2.10.1. □

As a remark from Theorem 2.4.6, we obtain that the evolution of the error $e(t)$ is reset at each triggering instance, irrespective of whether $\tilde{A}$ is Hurwitz or not. This is only done through the appropriate construction of the corrective component (ψ) in the control, and without this component $e(t)$ will grow exponentially when $\tilde{A}$ is unstable and hence violate the requirement $\|e(t)\| \leq \epsilon$ for any Zeno effect free triggering strategy.

At this point, we have shown that under the assumption that $\mathcal{E}$ can measure and transmit both $x(\theta(t))$ and $e(\theta(t))$, the controller $\mathcal{C}$ can ensure that $\forall t > \theta(t)$, (2.22) holds. Therefore, the next step would be to determine triggering instances t_i (hence characterizing $\theta(\cdot)$) such that $\|e(t)\| \leq \epsilon$ is satisfied. Also, we need to ensure that $\mathcal{E}$ can precisely calculate $e(t_i)$ at each triggering instance so that it can communicate it to the controller.

At this point, we focus on how $\mathcal{E}$ would precisely calculate and send $e(t_i)$ to the controller at each triggering instance. In practice, $\mathcal{E}$ has the knowledge $\mathcal{F}_t$ and calculation of $e(t)$ requires the knowledge of x_c which is not available. From the dynamics of $x(t)$, if the controller's parameters $L(t)$ and ψ in (2.19) are known to

$\mathcal{E}$, then $\mathcal{E}$ can uniquely determine $e(t)$ by observing $x(t)$ only. To see this, let us define a dummy state $x_d(t)$ which follows the dynamics

$$\dot{x}_d = Ax_d + B\hat{u}$$
$$x_d(\theta(t)) = x(\theta(t))$$

where $\hat{u}$ is the input generated by the optimal controller $\mathcal{K}(t, \mathfrak{I}(t))$. If $\mathcal{E}$ knows the structure of the controller, then $\mathcal{E}$ can compute x_d precisely just by observing $x(t)$. Now, if we define a new variable $\Delta(t) = x(t) - x_d(t)$, then $\Delta(t)$ follows the dynamics:

$$\dot{\Delta} = A\Delta + d(t)$$
$$\Delta(\theta(t)) = 0.$$

From the definition of Δ, one can immediately verify that $BK\Delta(t) = d_1(t)$. Thus, in order to know $d_1(t)$, $\mathcal{E}$ needs to monitor $x(t)$ and compute $x_d(t)$. We notice that due to the third assumption that $\mathfrak{I}(0) = \{x(0)\}$, we have $e(0) = 0$. Hence, based on whether the corrective control ψ is used or not, we either have $e(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t,s)d_1(s)ds$ or $e(t) = \int_0^t \tilde{\Phi}(t,s)d_1(s)ds$ due to (2.22) and (2.18) respectively. Therefore, in either situation, $e(t)$ can be calculated by the event-generator $\mathcal{E}$. Hence, even in the absence of $x_c(t)$, $e(t)$ can be calculated precisely only from the knowledge of $x(t)$ and computing a dummy state $x_d(t)$.

Fortunately, the optimal structure of $L(t)$ is unique and therefore $\mathcal{C}$ does not need to communicate this information to the event-generator. Similarly ψ has a structure that is determined by the parameter δ. Therefore, the only information related to $\mathcal{C}$ that $\mathcal{E}$ needs to know is the value of δ chosen by $\mathcal{C}$. To resolve this issue, in the same framework, $\mathcal{E}$ can prescribe a $\delta(t_i)$ for the controller send the augmented information $(x(t_i), e(t_i), \delta(t_i))$ to $\mathcal{C}$. Otherwise if $\mathcal{C}$ selects $\delta(t_i)$ (for equation (2.21)) after receiving $(x(t_i), e(t_i))$, then that value of $\delta(t_i)$ needs to be communicated to $\mathcal{E}$; and this requires a bi-directional communication between the controller and event-generator. Whichever of these two methods are adopted, our next results are going to be invariant of this choice.

In the following analysis, without loss of generality we will assume that $\delta(t_i)$ is chosen arbitrarily small enough, either by $\mathcal{C}$ or by $\mathcal{E}$, at each triggering time t_i such that at the next triggering instance t_{i+1},

$$e(t) = \int_{t_i}^{t} \tilde{\Phi}(t,s) d_1(s) ds, \qquad \forall t \leq t_{i+1}.$$

Now that we have the optimal controller designed and event-generator having the precise knowledge of $e(t)$ for all t, we are ready to study the optimal event-generating scheme by solving Problem 2.3.1.

2.5 Optimal Event Generator Synthesis

In this section we will assume that the optimal $\mathcal{C}$ uses a $\psi(\cdot)$ (or $\mathcal{E}$ prescribes a δ arbitrarily small) such that for all $t > \theta(t)$, we have

$$e(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) d_1(s) ds.$$

Later we will remove that assumption that δ is arbitrarily small and consider a δ which is finite and bounded from below (see Section 2.8).

Definition 2.5.1. *Two optimization problems $P1$ and $P2$ are equivalent if the optimal solution of one is the optimal solution for the other and the corresponding optimal values for both the problems are the same.*

Let us formulate an unconstrained optimization problem that is equivalent to Problem 2.3.1.

Problem 2.5.2. *For any given $\epsilon > 0$,*

$$\inf_{\mathcal{E},\mathcal{C}} J_1(\mathcal{C},\mathcal{E}). \tag{2.23}$$

$$J_1(\mathcal{C},\mathcal{E}) = \sup_{d(\cdot)\in\mathcal{D}} (N(T) + \sup_{s\in[0,T]} c^{\epsilon}(\|e(s)\|)),$$

where

$$c^{\epsilon}(x) = \begin{cases} 0 & x \leq \epsilon \\ \\ +\infty & x > \epsilon. \end{cases}$$

Proposition 2.5.3. *Problem 2.3.1 is equivalent to Problem 2.5.2.*

Proof. Note that the optimal $\mathcal{C}$ for both the problems will be same as what discussed in Section 2.4. Therefore, we will focus on synthesising $\mathcal{E}$ only.

Let $\mathcal{E}_1$ be an optimal solution for Problem 2.3.1. By finiteness assumption, $J(\mathcal{C}, \mathcal{E}_1)$ is finite, and satisfies the constraint $\sup_{[0,T]} \|e(t)\| \leq \epsilon$. As a result, $J_1(\mathcal{C}, \mathcal{E}_1) = J(\mathcal{C}, \mathcal{E}_1)$.

Let us assume $\mathcal{E}_2$ be an optimal solution of Problem 2.5.2, then $J_1(\mathcal{C}, \mathcal{E}_2) \leq J_1(\mathcal{C}, \mathcal{E}_1) = J(\mathcal{C}, \mathcal{E}_1)$. Since $J_1(\mathcal{C}, \mathcal{E}_2)$ is finite, it satisfies the constraint $\sup_{[0,T]} \|e(t)\| \leq \epsilon$; and therefore it is a feasible solution for Problem 2.3.1, and further $J(\mathcal{C}, \mathcal{E}_2) = J_1(\mathcal{C}, \mathcal{E}_2)$.

Thus, $J(\mathcal{C}, \mathcal{E}_2) = J_1(\mathcal{C}, \mathcal{E}_2) \leq J(\mathcal{C}, \mathcal{E}_1) \leq J(\mathcal{C}, \mathcal{E}_2)$. Hence $J(\mathcal{C}, \mathcal{E}_1) = J(\mathcal{C}, \mathcal{E}_2) = J_1(\mathcal{C}, \mathcal{E}_1) = J_1(\mathcal{C}, \mathcal{E}_2)$. □

The construction of Problem 2.5.2 is followed by the well-known barrier function method in optimization [102], however, we do not construct a barrier function which is smooth and continuous such as log-barrier-functions. Instead of following a gradient based optimization here on the unconstrainted objective $J_1(\mathcal{C}, \mathcal{E})$, we will adopt a dynamic programming based approach, where the solution of the dynamic program will be the time instances to trigger the events.

Let us define the set

$$S = \{t_0, t_1, \cdots, t_l \mid \forall i, t_i < t_{i+1}, t_0 \geq 0, t_i < T, l \in \mathbb{N}\}.$$

S denotes the set of all possible event-triggering strategies. Analogous to S, let us also define:

$$S(t) = \{t_0, t_1, \cdots, t_l \mid \forall i, t_i < t_{i+1}, t_0 \geq t, t_i < T, l \in \mathbb{N}\}$$

which is the set of all feasible triggering instances after time t.

Since $\min_{\mathcal{E},\mathcal{C}} J_1(\mathcal{C}, \mathcal{E}) = \min_{\mathcal{E}} J_1(\mathcal{C}^*, \mathcal{E})$ where $\mathcal{C}^*$ is the optimal controller discussed previously, we will suppress the dependency of J_1 (or J) on $\mathcal{C}^*$ in the following analysis.

Let us denote a value function

$$\begin{aligned} V(t, e) &= \inf_{\mathcal{E} \in S(t), e(t) = e} \sup_{d(\cdot) \in \mathcal{D}} \{|\mathcal{E}| + \sup_{s \in [t,T]} c^{\epsilon}(\|e(s)\|)\} \\ V(T, e) &= 0, \end{aligned} \tag{2.24}$$

and $V(0, 0)$ will be the solution to Problem 2.5.2. Also, note that if $\|e(t)\| = \|e\| > \epsilon$, then $V(t, e) = +\infty$.

From the special structure of $c^{\epsilon}(\cdot)$, we can write $\sup_{s \in [t,T]} c^{\epsilon}(\|e(s)\|) = \sup_{s \in [t,r]} c^{\epsilon}(\|e(s)\|) + \sup_{s \in [r,T]} c^{\epsilon}(\|e(s)\|)$ for all $r \in [t, T]$.

Let us assume $\|e(t)\| \leq \epsilon$ and $\{t_0^*, t_1^*, \cdots, t_k^*\} \in S(t)$ be the optimal triggering instances starting at time t. Therefore, by optimality criterion,

$$V(t,e) = \inf_{t_0 \geq t, e(t)=e} \{1_{t_0<T} + \sup_{d(\cdot)\in\mathcal{D}} \sup_{s\in[t,t_0]} c^{\epsilon}(\|e(s)\|) + V(t_0,0)\}$$

$$=1_{t_0^*<T} + V(t_0^*,0).$$

Clearly for all $s > t$,

$$V(t,0) \geq V(s,0),$$

ensuring that $V(\cdot,0)$ is a non-increasing function.

Let us define:

$$\tau^*(t) = \inf_{s} \Big\{T > s \geq t \mid \|e(s)\| = \epsilon\Big\}$$

where in our convention inf over an empty set evaluates to be $+\infty$. Therefore $t_0^* \leq \tau^*(t)$. Also, we have for all $r \leq \tau^*(t)$, $V(r,0) \geq V(\tau^*(t),0)$. Thus,

$$1_{t_0^*<T} + V(t_0^*,0) \geq 1_{\tau^*(t)<T} + V(\tau^*(t),0) \tag{2.25}$$

However, since t_0^* is optimal, then for all $s \geq t$,

$$1_{t_0^*<T} + V(t_0^*,0) \leq 1_{s<T} + V(s,0) + \sup_{d(\cdot)\in\mathcal{D}} \sup_{r\in[t,s]} c^{\epsilon}(\|e(r)\|).$$

Substituting, $s = \tau^*(t)$,

$$1_{t_0^*<T} + V(t_0^*, 0) \leq 1_{\tau^*(t)<T} + V(\tau^*(t), 0). \tag{2.26}$$

Combining (2.25) and (2.26), we obtain:

$$1_{t_0^*<T} + V(t_0^*, 0) = 1_{\tau^*(t)<T} + V(\tau^*(t), 0). \tag{2.27}$$

Noticing that the function $1_{\cdot<T} + V(\cdot, 0) : \mathbb{R}_+ \to \mathbb{R}_+ \cup \{+\infty\}$ is non-increasing and $t_0^* \leq \tau^*(t)$, the condition for t_0^* to be optimal is $t_0^* = \tau^*(t)$.

Therefore,

$$V(t,e) = \begin{cases} 1 + V(\tau^*(t), 0) & \tau^*(t) < T \\ 0 & \tau^*(t) = +\infty \end{cases}$$

Thus, the triggering strategy is to wait until the error reaches to the value ϵ and then trigger an event in order to 'reset' the error.

Remark 2.5.4. *Formally, if t_i is the i-th triggering instance, then*

$$t_{i+1} = \inf_s \left\{ T > s > t_i \mid \|e(s)\| = \epsilon \right\} \tag{2.28}$$
$$t_0 = \inf_s \left\{ T > s > 0 \mid \|e(s)\| = \epsilon \right\}$$

The event triggering mechanism (2.28) is very common in literature e.g. [64], [66], [65], [29] , however, to the best of our knowledge, none of the past works formally states the optimality of such a strategy. In the following we show that the event-triggering strategy is unique and well defined, i.e., the strategy does not exhibit Zeno behavior.

Theorem 2.5.5. *The optimal triggering strategy* (2.28) *is unique.*

Proof. Let $\mathcal{E}_1 = \{t_0^1, t_2^1, .., t_{l_1}^1\}$ and $\mathcal{E}_2 = \{t_0^2, t_2^2, .., t_{l_2}^2\}$ be any two optimal solutions of the dynamic programming equation (2.25). Then by (2.28), for all $i = 1, 2, \cdots$

$$t_i^1 = \inf_{s \in (t_{i-1}^1, T)} \Big\{ s \geq 0 \mid \sup_{(t_{i-1}^1, s]} \|e(r)\| = \epsilon \Big\},$$

and $\sup_{r \in (t_{l_1}^1, T)} \|e(r)\| < \epsilon$. Similarly,

$$t_i^2 = \inf_{s \in (t_{i-1}^2, T)} \Big\{ s \geq 0 \mid \sup_{(t_{i-1}^2, s]} \|e(r)\| = \epsilon \Big\},$$

and $\sup_{r \in (t_{l_1}^1, T)} \|e(r)\| < \epsilon$.

$$t_0^1 = \inf_{s \in (0, T)} \Big\{ s \geq 0 \mid \sup_{(0, s]} \|e(r)\| = \epsilon \Big\} = t_0^2$$

Therefore using mathematical induction based argument, $t_i^1 = t_i^2$ for all i and $l_1 = l_2 = l$. □

Corollary 2.5.6. *The optimal cost* $J_1(\mathcal{C}^*, \mathcal{E}^*)$ *is finite iff there is no Zeno effect in the event triggering mechanism.*

Proof. First, let us assume that the optimal triggering mechanism $(\mathcal{E}^*)$ is free of Zeno behavior. Therefore, for the horizon $[0, T]$, there are only finite number of triggering instance, say $\{t_0, t_1, \cdots, t_l\}$. From (2.28), within each interval $[0, t_0], [t_0, t_1], \cdots, [t_l, T]$, $\|e(t)\| \leq \epsilon$. Hence, $J_1(\mathcal{C}^*, \mathcal{E}^*) = V(0,0) = l < +\infty$.

Now, let us assume that $J_1(\mathcal{C}^*, \mathcal{E}^*) = J < +\infty$. Thus, $\mathcal{E}^*$ and $\mathcal{C}^*$ ensure $\sup_{d(\cdot)\in\mathcal{D}} \|e(t)\| \leq \epsilon$ for all $t \in [0, T]$. Therefore from (2.5.2), $J = N(T) = |\mathcal{E}|$. Thus there are only finitely many triggering instances and hence, the triggering mechanism does not exhibit Zeno behavior. □

Now, we also claim that $J_1(\mathcal{C}^*, \mathcal{E}^*)$ is always finite, and the finiteness of $J_1(\mathcal{C}^*, \mathcal{E}^*)$ is ensured by showing that there exists a time interval of positive measure between any two triggerings.

Theorem 2.5.7. *The inter-event times are bounded from below.*

Proof. Let t_i be a triggering instance when $(x(t_i), e(t_i))$ was sent to the controller. Thus, for all $t > t_i$

$$e(t) = \int_{t_i}^{t} \tilde{\Phi}(t, s) d_1(s) ds.$$

If t_{i+1} is the next triggering instance, then

$$\epsilon = \Big\| \int_{t_i}^{t_{i+1}} \tilde{\Phi}(t_{i+1}, s) d_1(s) ds \Big\|.$$

Using $d_1(t) = BK \int_{t_i}^{t} \Phi(t, s) d(s) ds$,

$$\epsilon = \Big\| \int_{t_i}^{t_{i+1}} (\Phi(t_{i+1}, s) - \tilde{\Phi}(t_{i+1}, s)) d(s) ds \Big\|.$$

Thus,

$$\epsilon \leq \Delta(t_i, t_{i+1})$$

where for all $\sigma \geq \tau$

$$\Delta(\tau, \sigma) = \sup_{s \in [\tau, \sigma]} \|(\Phi(\sigma, s) - \tilde{\Phi}(\sigma, s))\| \int_{\tau}^{\sigma} \|d(s)\| ds.$$

Note that $\Delta(\tau, \tau) = 0$ and it can be verified that $\Delta(\cdot, \cdot)$ is an uniformly continuous function on a compact domain. Thus for all $\epsilon > 0$, $\exists \delta > 0$ [1], such that if $\|(\tau_1, \sigma_1) - (\tau_2, \sigma_2)\| < \delta$, then $\|\Delta(\tau_1, \sigma_1) - \Delta(\tau_2, \sigma_2)\| < \epsilon$ for all $(\tau_1, \sigma_1), (\tau_2, \sigma_2)$ such that $\sigma_i \geq \tau_i$, $i = 1, 2$.

Using $\tau_1 = \tau_2 = \sigma_1 = t_i$, for all σ_2 such that $|\sigma_2 - t_i| < \delta$, $\|\Delta(t_i, \sigma_2)\| < \epsilon$

However, $\epsilon \leq \Delta(t_i, t_{i+1})$. Thus, $|t_{i+1} - t_i| > \delta$. Therefore for each triggering instance t_i, the next triggering is atleast after δ amount of time. □

Remark 2.5.8. *Using the assumption that* $\int_0^\infty \|d(t)\| dt < +\infty$ *one can show that the inter-event times are bounded from below by* δ *for an infinite horizon problem as well.*

Lemma 2.5.9. *For all* $T > 0$, $\exists \rho > 0$ *such that* $\sup_t \|d(t)\| < \rho$ *implies there will be no triggering.*

[1] δ will depend on ϵ; it should be denote it by $\delta(\epsilon)$

Proof. Note that, for all $t \leq T$,

$$\|e(t)\| \leq \rho \int_0^T \|\Phi(T,s) - \tilde{\Phi}(T,s)\| ds.$$

Therefore, for all $\rho \leq \frac{\epsilon}{\int_0^T \|\Phi(T,s)-\tilde{\Phi}(T,s)\| ds}$, $\sup_t \|e(t)\| \leq \epsilon$ and hence there will be no triggering. □

Therefore, for the trivial case $d(t) \equiv 0$, there will be no triggering and moreover, $e(t) \equiv 0$. This shows the well known fact that for a deterministic system, any feedback law can be realized with the only information of the initial state.

To summarize, we have proved that for any time invariant controllable linear system, any linear feedback can be replaced by an event-triggered feedback by satisfying the constraint $\sup_t \|e(t)\| \leq \epsilon$ for any $\epsilon > 0$.

2.6 Infinite Horizon Design Problem

In this section, we visit the optimal $(\mathcal{E}, \mathcal{C})$ design problem for the infinite horizon problem as presented in Problem 2.3.2.

From Section 2.4, we notice that the controller synthesis does not depend on the horizon $[0, T]$, and neither does it depend on the cost function. Rather, the design is aimed to satisfy the constraint $\|e(t)\| \leq \epsilon$ for all t. Thus, for an infinite horizon problem, the optimal $\mathcal{C}$ will have the same structure. One can formally prove this statement by repeating the analysis done in Section 2.4, however, we do not present the analysis here to maintain brevity.

In order generate the optimal $\mathcal{E}$, let us construct the equivalent problem as it was done for the finite horizon problem.

Problem 2.6.1. *For any given $\epsilon > 0$,*

$$\inf_{\mathcal{E},\mathcal{C}} J_{2,T}(\mathcal{C},\mathcal{E}) \tag{2.29}$$

where

$$J_{2,T}(\mathcal{C},\mathcal{E}) = \sup_{d(\cdot)\in\mathcal{D}} \{\sum_{i=1}^{N(T)} e^{-t_i} + \sup_{s\in[0,T]} c^{\epsilon}(\|e(s)\|)\}.$$

If $\mathcal{E}_T^*$ is the solution of the above problem then $\mathcal{E}_\infty^* = \limsup_{T\to\infty} \mathcal{E}_T^*$.

For any finite horizon $[0,T]$, the optimal triggering instances $\{t_0^*, t_1^*, \cdots, t_l^*\}$ depends on the horizon T. Let us formally denote

$$\mathcal{T}(T) = \{t_0^*(T), t_1^*(T), \cdots, t_{N(T)}^*(T)\}. \tag{2.30}$$

With slight abuse of notation, by $\limsup_{T\to\infty} \mathcal{E}_T^*$, we basically want to compute $\mathcal{T}(\infty) = \limsup_{T\to\infty} \mathcal{T}(T)$; and then we want to characterize $\mathcal{E}_\infty^*$ by $\mathcal{T}(\infty)$. In general, for an infinite horizon problem, $\limsup_{T\to\infty} N(T)$ converges to infinity in (2.30). However, if $\limsup_{T\to\infty} N(T)$ is countable then $\limsup_{T\to\infty} \sum_{i=1}^{N(T)} e^{-t_i}$ is finite. On the other hand, if the triggering strategy forms a continuum of triggering instances (i.e. Zeno behavior), then $\limsup_{T\to\infty} \sum_{i=1}^{N(T)} e^{-t_i} = \infty$. Therefore, the optimal triggering instances –that achieve finite value for $J_{2,T}$– found by this formulation excludes Zeno behavior, but at the same time it allows for countable number of triggerings which are desired for an infinite horizon problem.

Proposition 2.6.2. *Problem 2.3.2 is equivalent to Problem 2.6.1.*

The proof of this proposition is very similar to the proof of Proposition 2.5.3 and hence we omit it.

Let us denote a value function for the infinite horizon problem for any arbitrary interval $[0, T]$:

$$V_T(t, e) = \inf_{\mathcal{E} \in S(t)} \sup_{d(\cdot) \in \mathcal{D}} \{ \sum_{i=1}^{|\mathcal{E}|} e^{-t_i} + \sup_{s \in [t,T]} c^{\epsilon}(\|e(s)\|) \}$$
$$V_T(T, e) = 0, \tag{2.31}$$

and $\limsup_{T \to \infty} V_T(0, 0)$ is what we are interested in.

Let $t_0 > t$ be the first element of $S(t)$, then by dynamic programming principle,

$$V_T(t, e) = \inf_{t_0 \geq t} \{ 1_{t_0 < T} e^{-t_0} + \sup_{d(\cdot) \in \mathcal{D}} \sup_{s \in [t, t_0]} c^{\epsilon}(\|e(s)\|) + V_T(t_0, 0) \} \tag{2.32}$$

Clearly, in this case as well, for all $s > t$

$$V_T(t, 0) \geq V_T(s, 0).$$

Let us define:

$$t_0^*(t) = \inf_{s\in[t,T]} \Big\{ s \geq t \mid \sup_{[t,s]} \|e(r)\| = \epsilon \Big\}.$$

Therefore by the same argument as for the finite horizon case,

$$V_T(t,e) = \begin{cases} e^{-t_0^*(t)} + V_T(t_0^*(t), 0) & t_0^*(t) < T \\ 0 & t_0^*(t) = +\infty \end{cases} \tag{2.33}$$

Thus, the triggering strategy is exactly same as what we had before and the strategy does not depend on the horizon T, although the output of the strategy (i.e. number of triggerings) varies with T. One property to note here is, if $\{t_0, t_1, \cdots, t_{N(T_1)}\}$ are the optimal time instances for triggering for a horizon $[0, T_1]$, and $\{s_0, s_1, \cdots, s_{N(T_2)}\}$ are the optimal time instances for triggering for a horizon $[0, T_2]$ ($T_2 > T_1$), then $N(T_2) \geq N(T_1)$ and $s_i = t_i$ for all $0 \leq i \leq N(T_1)$. This is nothing but the optimality principle.

From (2.33), one can obtain

$$V_\infty(t,e) \triangleq \limsup_{T\to\infty} V_T(t,e) = e^{-t_0^*(t)} + V_\infty(t_0^*(t), 0).$$

The value function $V_T(t,e)$ depends on T; and its value is non-decreasing as T increases. The following theorem ensures that for the optimal $\mathcal{E}_\infty^*$, the value function $V_\infty(t,e)$ attains a finite value for all t and $\|e\| \leq \epsilon$.

Theorem 2.6.3. *The optimal value of the asymptotic case ($T \to \infty$) of Problem 2.6.1 is finite at $\mathcal{E}^*_\infty$.*

Proof. First let us note that for the asymptotic case:

$$V_\infty(t,e) = \limsup_{T\to\infty} \inf_{\mathcal{E}\in S(t)} \sup_{d(\cdot)\in\mathcal{D}} \{\sum_{i=1}^{|\mathcal{E}|} e^{-t_i} + \sup_{s\in[t,T]} c^\epsilon(\|e(s)\|)\}$$

$$\limsup_{T\to\infty} V_\infty(T,e) = 0, \qquad \forall e \tag{2.34}$$

and the optimal value of the asymptotic problem is $V_\infty(0,0)$.

Due to Remark 2.5.8, one can show that for the infinite horizon case also there exists $\delta > 0$ such that the inter-event times are at least δ apart, i.e. $t_{i+1} - t_i > \delta$ for all i.

Therefore, using (2.34), one can write:

$$V_\infty(0,0) = \sum_{i=0}^{k-1} e^{-t_i^*} + V_\infty(t^*_{k-1}, 0)$$

where the first k optimal triggering instances are $t_0^*, t_1^*, t_2^*, \cdots, t^*_{k-1}$. Using the fact that $t^*_{i+1} - t^*_i > \delta$ for all $i > 0$ and $t^*_0 > \delta$, one can obtain $k\delta < t^*_{k-1}$. Hence,

$$V_\infty(0,0) \leq e^{-\delta}\frac{1-e^{-k\delta}}{1-e^{-\delta}} + V_\infty(t^*_{k-1}, 0).$$

Thus for any k-th triggering time t^*_{k-1},

$$V_\infty(0,0) \leq e^{-\delta}\frac{1-e^{-t^*_{k-1}}}{1-e^{-\delta}} + V_\infty(t^*_{k-1}, 0) \tag{2.35}$$

Hence taking $k \to +\infty$, or equivalently $t^*_{k-1} \to \infty$

$$V_\infty(0,0) \leq \frac{e^{-\delta}}{1-e^{-\delta}}.$$

□

Let $\mathcal{E}_\infty$ be an event-generator such that within some finite interval $[T_1, T_2]$, the are N number of generated events. Then, clearly by the definition of $J_{2,\infty}$ in (2.29), $J_{2,\infty}(\mathcal{C}, \mathcal{E}_\infty) \geq Ne^{-T_2}$ for any controller $\mathcal{C}$. On the other hand, by Theorem 2.6.3, we have $J^*_{2,\infty} = V_\infty(0,0) \leq \frac{e^{-\delta}}{1-e^{-\delta}}$. Thus, any optimal event-generating policy would have only finite number of triggerings within a finite interval. In fact, if N_{T_1,T_2} is the optimal number of triggering in any interval $[T_1, T_2]$, then

$$N_{T_1,T_2} \leq \frac{e^{T_2-T_1}}{e^{\delta}-1}.$$

Thus, to summarize, for an infinite horizon problem, the next triggering time at any time t is given by[2]:

$$t^*_0(t) = \inf_s \{s \geq t | \sup_{r\in[t,s]} \|e(r)\| = \epsilon\}.$$

The triggering strategy is not necessarily a periodic strategy over the infinite horizon.

[2]This could be formally proved by following the steps similar to the ones used to conclude Remark 2.5.4

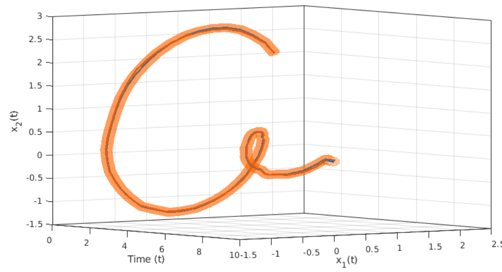


Figure 2.2: The behavior of the closed loop system in shown in red curve and the blue curve shows the same for event-based system. The orange tube has the tolerance radius of 0.1.

2.7 Simulation Results

In this section, we will illustrate our approach using a system evolving in $\mathbb{R}^2$ with the dynamics:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.75 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d.$$

The designed control is $u = -x_2$. For this simulation, we used d to be a bounded valued disturbance with values in $[-0.5, 0.5]$. The ϵ for this simulation was chosen to be 0.1, and we use $\|\cdot\|_2$ norm, i.e., the requirement is $\|e(t)\|_2 \leq \epsilon = 0.1$ for all t.

In Figure 2.2, we show the trajectory of the closed loop system, the trajectory of the optimal event-based system discussed here, and the orange tube has a radius of $\epsilon = 0.1$. One may visualize that the phase-plot (projection of the plot in the $x_1 x_2$ plane) is spiral due to the chosen parameters

In Figure 2.3, we show the optimal control $u(t)$, $\psi(t)$ and the triggering in-

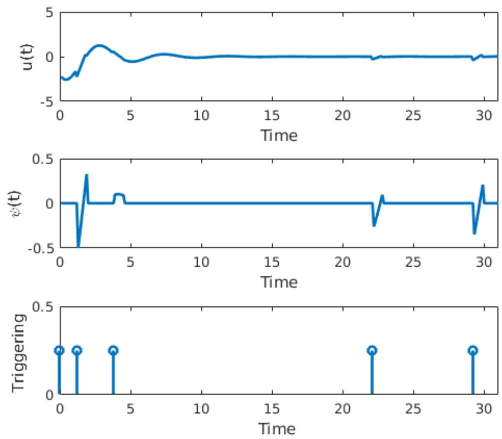


Figure 2.3: Top: control $u(t) = -K\tilde{\Phi}(t, \theta(t))x(\theta(t)) + \psi(t, e(\theta(t)))$, Middle: $\psi(t, e(\theta(t)))$ vs t, Down: The optimal triggering time instances.

stances. For the whole time interval of $[0, 35]$, only 4 (except the one at time 0) triggerings were initiated.

To see the effect of ψ we performed a simulation under same disturbance and selected $\psi = 0$. The state trajectory and the corresponding error norm is presented in Figures 2.4 and 2.5 respectively.

These figures support the fact that ψ plays a very important role in ensuring system's performance (and stability) for an event-based system when the closed loop dynamics is not Hurwitz.

2.8 Discussion

2.8.1 A note on the choice of δ in Theorem 2.4.4

In the analysis, we have theoretically shown that under the controllability assumption on (A, B), the effect of the residual error $e(\theta(t))$ can be nullified in

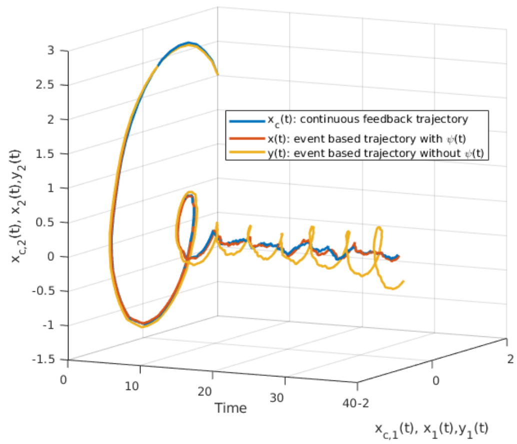


Figure 2.4: The state trajectories: closed loop system, optimal event-based system with information sharing $\mathfrak{I} = \{x(t_i), e(t_i)\}_{i=1,2,\ldots}$, and event-based system with information $\mathfrak{I} = \{x(t_i)\}_{i=1,2,\ldots}$

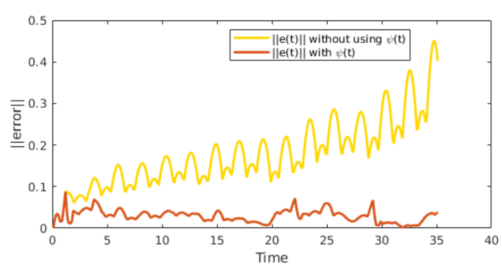


Figure 2.5: The norm of the error for event based systems under two different situations: using $\psi(t)$ (red) and without $\psi(t)$ (yellow).

arbitrary small time and hence for all $t > \theta(t)$, $e(t)$ does not depend on $e(\theta(t))$. However, as the permitted time (δ in Theorem 2.4.4) to mitigate the effect of $e(\theta(t))$ gets smaller and smaller, the amplitude of the corrective control gets larger and larger; and in the limit $\delta \to 0$, the corrective control component becomes a Dirac-delta distribution. In practice, the system might not be able to handle such an 'impulsive' nature of the controller. Therefore, in this section, we make an attempt to study the same problem while allowing the controller to have a certain positive amount of time to mitigate this error. The purpose of this subsection is primarily on the implementation aspect of such a controller, where we aim to show that even without using an 'impulsive' controller, the requirement $\|e(t)\| \leq \epsilon$ can be still achieved by *slightly* changing the threshold for the event-triggering strategy. In order to do so, we assume that the disturbance is bounded $\sup_t \|d(t)\| \leq D$. The study of the problem with arbitrary $d(\cdot)$ is beyond the scope of this chapter.

The performance of the (heuristic) method that we are going to propose, depends on a parameter $\alpha \in (0.5, 1)$.

Firstly, let us note that, by choosing the optimal linear controller we have ((2.42),(2.44))

$$e(t) = G(t, \mathfrak{I}) + \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) d_1(s) ds$$

$$\dot{G} = AG + Ba$$

$$G(\theta(t), \mathfrak{I}) = e(\theta(t))$$

where $a(t)$ can be chosen freely.

Since (A, B) is a controllable pair, by suitable pole placement $\|G(t, \mathfrak{I})\| \leq e^{-\lambda(t-\theta(t))}e(\theta(t))$ can be achieved for any $\lambda > 0$. Let the event-generator $\mathcal{E}$ triggers an event when $\|e(t)\| = \alpha\epsilon$ where $\alpha \in (0.5, 1)$. Let us also note that,

$$h(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds = \int_{\theta(t)}^{t} (\Phi(t, s) - \tilde{\Phi}(t, s))d(s)ds$$

is a differentiable function with $h(\theta(t)) = 0$ and $h'_+(\theta(t)) = 0$[3].

Thus, we can define

$$D_1 = \sup_s \{s \geq \theta(t) \mid \sup_{r \in [\theta(t), s]} \|h(r)\| \leq (1 - \alpha)\epsilon\}.$$

Due to the above mentioned properties of $h(t)$, $D_1 - \theta(t)$ is strictly positive, in fact, $D_1 - \theta(t) > \frac{(1-\alpha)\epsilon}{L_h}$ where L_h is the Lipschitz constant of hh.

Therefore, for all $t \in [\theta(t), D_1]$, $h(t) \leq (1 - \alpha)\epsilon$ and we have the following lemma:

Lemma 2.8.1. *If (A, B) is a controllable pair, then for all $\lambda > 0$ there exists control such that for all $t \in [\theta(t), D_1]$*

$$\|G(t, \mathfrak{I})\| \leq \|G(\theta(t), \mathfrak{I})\| e^{-\lambda(t-\theta(t))}$$

$$G(D_1, \mathfrak{I}) = 0.$$

Moreover, the controller that achieves the above requirement, produces control signal

[3] h'_+ is the upper-Dini-derivative of h

of bounded value.

These ensure $\forall t \leq D_1$, $\|e(t)\| \leq \epsilon$ and at time $t = D_1$,

$$\|e(D_1)\| \leq \int_{\theta(t)}^{D_1} \|\tilde{\Phi}(t,s)BKd_1(s)\|ds = (1-\alpha)\epsilon < \alpha\epsilon$$

Thus, the inter-triggering intervals are at least of $(1-\alpha)\epsilon/L_h$ duration and hence the controller has $(1-\alpha)\epsilon/L_h$ amount of time to mitigate the effect of residual error $e(\theta(t))$.

2.8.2 An approximate solution for optimal control problems

Let us consider the following optimal control problem:

Problem 2.8.2. *Find an event-generator and optimal controller pair* $(\mathcal{E}, \mathcal{C})$ *such that:*

$$\min_{\mathcal{E},\mathcal{C}} \sup_{d(\cdot)\in\mathcal{D}} \left[N(T) + \alpha \int_0^T l(s, x(s), \hat{u}(s))ds \right] \tag{2.36}$$

$$s.t. \qquad \dot{x} = Ax + B\hat{u} + d \tag{2.37}$$

where $\hat{u}$ *is an event-based control (linear in measurements) input generated by* $(\mathcal{E}, \mathcal{C})$*, and* $\alpha > 0$.

Solving this problem even for quadratic $l(s, x, u)$ is not trivial (as compared to the well celebrated LQ problems) due to the event-based structure of the controller. Such an optimization problem is common when there is communication cost associated with the transmission of the measurements. However, using our approach one can solve this problem approximately. In the first stage, let us solve Problem 2.8.3 to get a linear feedback controller.

Problem 2.8.3.

$$\min_{u} \sup_{d(\cdot)\in\mathcal{D}} \left[\int_0^T l(s, x(s), u(s))ds\right] \tag{2.38}$$

$$s.t. \qquad \dot{x} = Ax + Bu + d \tag{2.39}$$

where u is in the space of linear feedback controllers.

The above problem can be solved by assuming $u = Kx$ and then perform optimization in the space of $\mathbb{R}^{m\times n}$ matrices.

In the second stage the linear feedback control obtained by solving Problem 2.8.3 is approximated with an event-based control $\hat{u}$ that solves Problem 2.3.1.

Let us denote $u^*(t)$ and $x^*(t)$ to be the optimal control and the corresponding optimal trajectory for the Problem 2.8.3. If $\hat{u}(t)$ and $x(t)$ are the optimal approximation of u^* and x^* by soving Problem 2.3.1, then $\|x^*(t) - x(t)\| \leq \epsilon$, and one can also show that $\|u^*(t) - \hat{u}(t)\| \leq L_u(t)\epsilon$ for some function $L_u(t) > 0$. Thus,

$$\int_0^T l(s, x, \hat{u})ds = \int_0^T l(s, u^*, x^*)ds + O(\epsilon).$$

Therefore, the event-trigger controller generated in this two-step approach produces a cost which is $O(\epsilon)$ away from the optimal cost of the continuous feedback system. Taking $\epsilon \to 0$, we will have $\int_0^T l(s, x, \hat{u})ds \to \int_0^T l(s, u^*, x^*)ds$, however

$N(T)$ (the number of triggerings) will be higher as $\epsilon \to 0$.

The study of this optimal control problem is not the aim of this chapter, and this section is meant to demonstrate the applicability of this approach beyond the problems described in Problem 2.3.1–2.6.1.

2.9 Conclusion

In this chapter, we propose an optimal controller and event-generator pair to replace a linear continuous feedback controller with an event-triggered one. The deviation in trajectories between the continuous feedback system and the event-triggered system was considered to be a metric of performance. The analysis shows that for any given performance bound $\epsilon > 0$, there always exists a pair of event-triggered controller and event generator that bounds the trajectory deviation by ϵ, provided the system is controllable.

We show that the optimal controller is linear with respect to the latest information received. Moreover, the presence of corrective component $\psi(t, e(\theta(t)))$ is the crucial part of the controller when the closed-loop system is not Hurwitz. It is the $\psi(\cdot, \cdot)$ which ensures that the deviation in state trajectory does not grow unboundedly for a non-Hurwitz system. Without this component in the controller, the constraint $\|e(t)\| \leq \epsilon$ cannot be guaranteed (as illustrated in the simulation results).

The optimal event generator follows a threshold strategy where the threshold is the given performance metric ϵ. The event generator tracks the error $e(t)$ and transmits the state and error measurements to the controller whenever $\|e(t)\|$ reaches the threshold ϵ. Such a threshold based policy is ubiquitous and implementable easily. Further, such a threshold strategy does not exhibit Zeno behavior.

2.10 Appendix

2.10.1 Proof of Theorem 2.4.6

Let us consider the general form of the controller to be:

$$\hat{u}(t) = \mathcal{K}\Big(t, \{x(t_i)\}_{i=1}^{n(t)}, \{e(t_i)\}_{i=1}^{n(t)}\Big) \tag{2.40}$$

To maintain brevity we will use $\mathcal{K}(t, \mathfrak{I}_t)$ instead of $\mathcal{K}\Big(t, \{x(t_i)\}_{i=1}^{n(t)}, \{e(t_i)\}_{i=1}^{n(t)}\Big)$ where $\mathfrak{I}_t$ is the information related to the state measurement that is available to controller at time t.

Thus,

$$\dot{x} = Ax + B\mathcal{K}(t, I_t) + d \tag{2.41}$$

which leads to (using the fact that between $\theta(t)$ and t, no information arrives from $\mathcal{E}$ to $\mathcal{C}$, i.e. $\mathfrak{I}_t = \mathfrak{I}_s = \mathfrak{I}_{\theta(t)} = \mathfrak{I}$ (say) for all $s \in [\theta(t), t]$):

$$\begin{aligned}
x(t) =& \Phi(t, \theta(t))x(\theta(t)) + \int_{\theta(t)}^{t} \Phi(t, s)(B\mathcal{K}(s, \mathfrak{I}) + d(s))ds \\
x(t) =& F(t, \mathfrak{I}) + \int_{\theta(t)}^{t} \Phi(t, s)d(s)ds
\end{aligned}$$

where $F(t, \mathfrak{I}) = \Phi(t, \theta(t))x(\theta(t)) + \int_{\theta(t)}^{t} \Phi(t, s)B\mathcal{K}(s, \mathfrak{I})ds$

Therefore $e = x - x_c$ can be represented as:

$$\dot{e} = \tilde{A}e + BKx + B\mathcal{K}(t, \mathfrak{I})$$
$$\dot{e} = \tilde{A}e + BKF(t, \mathfrak{I}) + B\mathcal{K}(t, \mathfrak{I}) + BK \int_{\theta(t)}^{t} \Phi(t, s)d(s)ds.$$

Denoting $d_1(t) = BK \int_{\theta(t)}^{t} \Phi(t, s)d(s)ds$,

$$e(t) = G(t, \mathfrak{I}) + \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds. \tag{2.42}$$

where

$$G(t, \mathfrak{I}) = \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)B(KF(s, \mathfrak{I}) + \mathcal{K}(s, \mathfrak{I}))ds + \tilde{\Phi}(t, \theta(t))e(\theta(t)).$$

$$\sup_{d \in \mathcal{D}} \|e(t)\| = \sup_{d \in \mathcal{D}} (\|G(t, \mathfrak{I})\| + \| \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds\|)$$
$$= \sup_{d \in \mathcal{D}} \|G(t, \mathfrak{I})\| + \sup_{d \in \mathcal{D}} \| \int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds\|$$

All the equalities hold in the above derivation since we have no restriction on the disturbance. In fact, one can notice that $G(t, \mathfrak{I})$ depends on the realization of the disturbance until time $\theta(t)$ and the other term $\int_{\theta(t)}^{t} \tilde{\Phi}(t, s)d_1(s)ds$ depends on the realization of noise after time $\theta(t)$.

Therefore, in order to keep $\sup_{d \in \mathcal{D}} \|e(t)\|$ as low as possible, controller $\mathcal{C}$ is required to minimize $\|G(t, \mathfrak{I})\|$ –by properly selecting $\mathcal{K}(t, \mathfrak{I})$– since the other term,

$\| \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) d_1(s) ds \|$ is totally determined by the disturbance and hence it is left uncontrolled.

To further simplify $G(t, \mathfrak{I})$, we first note that

$$
\begin{aligned}
&\int_{\theta(t)}^{t} \tilde{\Phi}(t,s) BKF(s,\mathfrak{I}) ds \\
&= \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) BK\Phi(s,\theta(t)) x(\theta(t)) + \int_{\theta(t)}^{t} \tilde{\Phi}(t,s) BK \int_{\theta(t)}^{s} \Phi(s,r) B\mathcal{K}(r,\mathfrak{I}) dr ds \\
&= \int_{\theta(t)}^{t} \frac{d}{ds} (\tilde{\Phi}(t,s)\Phi(s,\theta(t))) x(\theta(t)) + \int_{r=\theta(t)}^{t} \Big[\int_{s=r}^{t} \tilde{\Phi}(t,s) BK\Phi(s,r) ds \Big] B\mathcal{K}(r,\mathfrak{I}) dr \\
&= \big(\Phi(t,\theta(t)) - \tilde{\Phi}(t,\theta(t))\big) x(\theta(t)) + \int_{\theta(t)}^{t} \big(\Phi(t,r) - \tilde{\Phi}(t,r)\big) B\mathcal{K}(r,\mathfrak{I}) dr
\end{aligned}
$$

Therefore,

$$
G(t,\mathfrak{I}) = \tilde{\Phi}(t,\theta(t)) e(\theta(t)) + \big(\Phi(t,\theta(t)) - \tilde{\Phi}(t,\theta(t))\big) x(\theta(t)) + \int_{\theta(t)}^{t} \Phi(t,r) B\mathcal{K}(r,\mathfrak{I}) dr \tag{2.43}
$$

Looking into the expression of $G(t, \mathfrak{I})$ in (2.43), one can guess that the $\mathcal{K}(t, \mathfrak{I})$ which aims to minimize $\|G(t, \mathfrak{I})\|$, is only a function of $x(\theta(t))$ and $e(\theta(t))$ and moreover $\mathcal{K}(t, \mathfrak{I})$ has to be linear with respect to $x(\theta(t))$ and $e(\theta(t))$.

Now, we want to check whether there exist matrix valued functions $M(t)$ and $L(t)$ such that $\mathcal{K}(t, \mathfrak{I}) = M(t)e(\theta(t)) + L(t)x(\theta(t))$ can make $\|G(t, \mathfrak{I})\| = 0$ (or arbitrary small).

Let us take $L(t) = -K\tilde{\Phi}(t, \theta(t))$, and this leads to

$$G(t, \mathfrak{I}) = \Big(\tilde{\Phi}(t, \theta(t)) + \int_{\theta(t)}^{t} \Phi(t, r)BM(r)dr\Big)e(\theta(t)).$$

This is a similar situation as we dealt with in Theorem 2.4.4. Under the assumption that (A, B) is a controllable pair, we can choose $M(t)$ such a way that $G(t, \mathfrak{I})$ converges to zero exponentially fast with the decay rate as fast as desired. Moreover as presented in Theorem 2.4.4, we can make $G(t, \mathfrak{I}) = 0$ for all $t > \theta(t)$. To see this choose $M(t) = K\tilde{\Phi}(t, \theta(t)) + M_1(t)$, and thus

$$\begin{aligned} G(t, \mathfrak{I}) =& \Big(\Phi(t, \theta(t)) + \int_{\theta(t)}^{t} \Phi(t, r)BM_1(r)dr\Big)e(\theta(t)) \\ =& \mathcal{G}(t)e(\theta(t)). \end{aligned}$$

where

$$\begin{aligned} \dot{\mathcal{G}} &= A\mathcal{G} + BM_1 \\ \mathcal{G}(\theta(t)) &= I \end{aligned} \tag{2.44}$$

Since (A, B) is controllable, we have the Grammian

$$W(\delta) = \int_{\theta(t)}^{\theta(t)+\delta} \Phi(\theta(t), s)BB'\Phi(\theta(t), s)'ds$$

is positive definite for all $\delta > 0$. Therefore, by selecting $M_1(t) = -1_{t \leq \theta(t)+\delta}B'\Phi(\theta(t), t)'W(\delta)^{-1}$ for all $t \geq \theta(t)$, one can verify that $\mathcal{G}(t) = 0$ for all $t \geq \theta(t) + \delta$. Since δ can be made arbitrarily small, one can conclude $\mathcal{G}(t) = 0$ for all $t > \theta(t)$; and hence $G(t, \mathfrak{I}) = 0$ for all $t > \theta(t)$

Thus, $\mathcal{K}(t, \mathfrak{I})$ is linear with respect to the elements of $\mathfrak{I}$, and furthermore, it only depends on the latest measurements. Therefore, as a result of using such $\mathcal{K}(t, \mathfrak{I})$, we can conclude that $\forall t > \theta(t)$

$$e(t) = \int_{\theta(t)}^{t} \tilde{\Phi}(t, s) d_1(s) ds.$$

Chapter 3: Event-Triggered Framework for Control Affine Nonlinear Systems: A Lyapunov Function Based Approach

In the previous chapter we focused on designing optimal controller and optimal event-generator for a linear system. In this chapter we focus on an event-triggered control strategy for control affine nonlinear systems. The proposed method ensures sufficient reduction in communication by only invoking a communication when some event has occurred. The error between the continuous state feedback nonlinear system and the event based system can be bounded in an invariant set. The upper bound of this error is derived which can be controlled by appropriately choosing the parameters for the event-triggering function.

3.1 Introduction & Literature Review

Computing the control law of a large system generally requires continuous reading from the sensors and transmitting these measurements to the control input generators. As a result, the performance depends on the continuous availability of the sensor measurements, and efficient and accurate computation of the control law. For a distributive system, although the computation is done distributively yet it requires continuous information exchange among the subsystems. Communicating the measurements obtained in a sensor network and propagating the data to the controller are necessary and important part for networked control systems. Consequently, the performance is generally restricted by the available network bandwidth and computing resources.

To overcome the limitation of available communication resources, researchers have come up with novel techniques such as event-based control [8], self triggered control [94], [59] and periodic time control [75], [18] that require only discrete-time communications. These methods ask for discrete communications between the sensor network and the controller, and as a result the controller can only generate a control that approximates the continuous state feedback control. The communication is done periodically, after T amount of time, in periodic control. Finding a suitable time period T to guarantee some level of performance is a main challenge for this approach. In self triggered and event based control, the communication is done only when some event has occurred.

Event based control has attracted a great deal of research in the recent past due to its effectiveness. A study that has been made in [7] on the performance of event based control and periodic control has revealed the fact that under some conditions the event based control performs better than periodic control. Interested readers are directed to confer [58] and [43] and the references therein to get a broad review on event-triggered and self-triggered control, estimation and optimization. Recent publications like [64] considered a state feedback approach for an event based system where the feedback control is generated from another subsystem and this subsystem is updated every time a trigger is generated. Event based control is proposed for distributed interconnected linear systems in [29] and [86].

Despite of the wide applicability of event based control, there is not much work, in literature, for nonlinear systems. In [95], the authors proposed an event based approach for nonlinear input-to-state-stable (ISS) systems. [96] and [97] studied the event based stabilization of nonlinear plants using a Lyapunov function based technique. [91] considered an event based approach for real time scheduling tasks. [3] extended the technique proposed in [91] for homogeneous and polynomial systems. Whereas the previous methods are Lyapunov function based approaches, [90] took a different formalism to study event based control for input-output linearizable systems with relative degree equal to the dimension (n) of the statespace. In [89], they refined the method for input-output linearizable input affine nonlinear systems with relative degree $r \leq n$. [90] and [89] focus on the deviation of the event based system from a continuous state feedback system and showed that this error is bounded. However, the rest of the past work mostly focus on the stabilizing behavior of the event based system rather than the actual error incurred due to the event based approach.

In this chapter, we also adopt a Lyapunov function based approach for an event based control strategy applied to input affine nonlinear control systems. In many cases, the control input is of state-feedback form to achieve optimality or some other desired performance and hence we consider that the controller is a state-feedback and known to us a priori. Therefore, it will be interesting to see how the system behaves if this control is approximated by a piecewise constant control. In the following sections we are going to explain on how to construct such a piecewise constant approximation of a given control input by using an event-based approach. We show that the error incurred by using this approximated input can be bounded within an invariant subset (Theorem 3.2.4) and moreover, the volume of this set can be controlled by the choice of some parameters related to the event-triggering function. We also show that under the adopted event-triggering strategy, there is a minimum time between two successive events (Theorem 3.3.1) which prohibits infinite triggerings in finite time.

3.2 Event-Based Nonlinear Control System

Let us consider the input-affine nonlinear dynamics as given in (3.1).

$$\dot{x} = f(t,x) + \sum_{i=1}^{m} g_i(t,x) \cdot u_i \tag{3.1}$$

where u_i is the control input and that input is of the form $\gamma_i(x)$ to achieve some desired behavior from the system.

The closed loop system is given in (3.2)

$$\dot{x}_c = f(t,x_c) + \sum_{i=1}^{m} g_i(t,x_c) \cdot \gamma_i(x_c) \tag{3.2}$$

We assume the following properties for the nonlinear systems considered in (3.1) and (3.2)

(A1) $\gamma_i(\cdot)$ and for all t, $f(t,\cdot)$ and $g_i(t,\cdot)$ are Lipschitz functions with Lipschitz constants L_γ^i, L_f and L_g^i respectively, for all $i = 1, 2, \cdots, m$.

(A2) $f_x(t,x) = \frac{\partial f(t,x)}{\partial x}$ and $((g\gamma)_i)_x(t,x) = \frac{\partial (g\gamma)_i(t,x)}{\partial x}$ are Lipschitz continuous with Lipschitz constants L_1 and L_2^i respectively, where $(g\gamma)_i(t,x) = g_i(t,x)\gamma_i(x)$.

(A3) The closed loop system with $u_i = \gamma_i(x)$ is exponentially stable.

Note that, we do not assume that the system (3.1) is ISS (Input to state stable), so the stability of (3.1) with any other input is not guaranteed.

Let us denote

$$F(t,x) = f(t,x) + \sum_{i=1}^{m} g_i(t,x) \cdot \gamma_i(x), \tag{3.3}$$

and the trajectory of the closed loop system (3.2) to be $x_c(t)$.

Our aim is to generate the controls u_i in such a way that does not require continuous availability of the state $x(t)$ and the deviation of the trajectory of this event based system from that of (3.2) is within some tolerance level. To design such a control we will consider $u_i(t) = \gamma_i(x(t_k)),\ \forall t \in [t_k, t_{k+1})$, where $x(t_k)$ is the value of the state at k-th triggering time t_k.

Theorem 3.2.1. *Let $x = 0$ be an equilibrium point for the nonlinear system $\dot{x} = h(t,x)$, where $h : [0,\infty) \times D \to \mathbb{R}^n$ is continuously differentiable, D is some domain in $\mathbb{R}^n$ that contains the origin, and the Jacobian matrix $\partial h/\partial x$ is bounded and Lipschitz on D, uniformly in t.*

Let

$$H(t) = \frac{\partial h}{\partial x}(t,x)\bigg|_{x=0}.$$

Then $x = 0$ is an exponentially stable equilibrium point for the nonlinear system if and only if it is an exponentially stable equilibrium point for the linear system $\dot{x} = H(t)x$.

Proof. For the proof of this theorem, the readers are directed to [51, Theorem 4.15].

□

Theorem 3.2.2. *The linear system*

$$\dot{p} = A(t)p$$

is exponentially stable, where $A(t) = \frac{\partial F}{\partial x}(t,x)\big|_{x=x_c(t)}$.

Proof. Let us consider the system,

$$(x_c \dot{-} p) = F(t, x_c - p) \tag{3.4}$$

By assumption (A3), (3.4) is an exponentially stable system and hence $\lim_{t\to\infty}(x_c(t) - p(t)) = 0$. We can write (3.4) in the following way as given in (3.5).

$$\dot{x_c} - \dot{p} = F(t, x_c) - \frac{\partial F}{\partial x}(t,x)\bigg|_{x=x_c(t)} p + O(\|p\|^2) \tag{3.5}$$

where $\lim_{\|p\|\to 0} O(\|p\|^2)/\|p\| = 0$. Since $x_c(t)$ satisfies (3.2) and by the definition of

$A(t)$, we obtain from (3.5)

$$\dot{p} = A(t)p + O(\|p\|^2). \tag{3.6}$$

By assumption (A3), both $x_c(t) \to 0$ and $x_c(t) - p(t) \to 0$ exponentially as $t \to \infty$ and as a consequence, $p(t) \to 0$ exponentially as $t \to \infty$. Therefore, $p = 0$ is an exponentially stable equilibrium point for (3.6). Theorem 3.2.1 ensures that the linearization of the nonlinear system (3.6) around $p = 0$ (i.e. $\dot{p} = A(t)p$) is exponentially stable. □

3.2.1 Event Based Closed Loop System and The Error Dynamics

The closed loop system with continuous state feedback is represented in (3.2). In the event based strategy, since we do not have continuous state feedback, the control law takes the form of $u_i = \gamma_i(x(t_k))$ where $x(t_k)$ is the value of the state at the previous triggering instance t_k. Therefore, basically u_i is a piecewise constant function. The event based closed loop system is obtained in (3.7).

$$\dot{x} = f(t,x) + \sum_{i=1}^{m} g_i(t,x)\gamma_i(x(t_k)) \quad \forall t \in [t_k, t_{k+1}). \tag{3.7}$$

The error e between the actual closed loop system (3.2) and the event based closed loop system (3.7) is defined to be $x_c - x$. e follows the nonlinear dynamics (3.8):

$$\dot{e} = F(t,x_c) - F(t,x) + \sum_{i=1}^{m} g_i(t,x)(\gamma_i(x) - \gamma_i(x(t_k))). \tag{3.8}$$

Our goal will be to keep this error bounded while only using the limited state measurements available at discrete time instances.

Proposition 3.2.3. *For all* $t \geq t_k$*, The dynamics of* $e(t)$ *can be written as,*

$$\dot{e} = A(t)e + \mu(t, x, x_c)e + \sum_{i=1}^{m} g_i(t, x)(\gamma_i(x) - \gamma_i(x(t_k))) \tag{3.9}$$

where

$\mu(t, x, x_c) = \int_0^1 \frac{\partial F}{\partial x}(t, x + se)ds - \frac{\partial F}{\partial x}(t, x)\big|_{x=x_c(t)}$.

Proof. By using Mean Value Theorem,

$$F(t, x_c) = F(t, x + e) = F(t, x) + \Big[\int_0^1 \frac{\partial F}{\partial x}(t, x + se)ds\Big]e.$$

Using (3.8) along with the application of Mean Value Theorem on $F(t, \cdot)$, we obtain,

$$\begin{aligned} \dot{e} = A(t)e + \Big(\int_0^1 \frac{\partial F}{\partial x}(t, x + se)ds - A(t)\Big)e \\ + \sum_{i=1}^{m} g_i(t, x)(\gamma_i(x) - \gamma_i(x(t_k))). \end{aligned} \tag{3.10}$$

Using the definition of $A(t)$ and defining $\mu(t, x, x_c) = \int_0^1 \frac{\partial F}{\partial x}(t, x + se)ds - A(t)$, we obtain (3.9). □

From the expression of $\mu(t, x, x_c)$, clearly $\mu(t, x, x_c)|_{e=0} = \mu(t, x_c, x_c) = 0$. Using this fact, the linearization of $\mu(t, x, x_c)e$ around $e = 0$ is:

$\frac{\partial(\mu e)}{\partial e}\big|_{e=0} = \big(\mu + \frac{\partial \mu}{\partial e}e\big)\Big|_{e=0} = 0.$

Theorem 3.2.4. *There exists $\epsilon > 0$ and a compact set $\Omega_\epsilon \subseteq \mathbb{R}^n$ such that for all t, x, y, if $\sum_{i=1}^{m} \|g_i(t,x)\gamma_i(y)\|_2 \leq \epsilon$ and $e(0) \in \Omega_\epsilon$, then the error $e(t) \in \Omega_\epsilon$ for all t.*

Proof. Let us first denote

$$\delta(t, x, \{t_l\}_{l=0}^{\infty}) = \sum_{i=1}^{m} g_i(t,x)(\gamma_i(x) - \gamma_i(x(t_k))),$$

where $t_k \leq t < t_{k+1}$. We can consider (3.9) to be a nonlinear system with perturbation, where the perturbation term is $\delta(t, x, \{t_l\}_{l=0}^{\infty})$ and the unperturbed nonlinear system is:

$$\dot{e} = A(t)e + \mu(t, x, x_c)e. \tag{3.11}$$

Theorems 3.2.1 and 3.2.2 along with the fact that linearization of $\mu(t, x, x_c)e$ around $e = 0$ is zero ensure that the unperturbed nonlinear system (3.11) is exponentially stable. Let $V(t, e) = e^T P(t) e$ be a Lyapunov function for the linear system $\dot{e} = A(t)e$. $P(t)$ satisfies the following properties:

1) $P(t)$ is continuously differentiable, symmetric, bounded, positive define matrix; that is, $0 < c_1 I \leq P(t) \leq c_2 I, \ \forall t$.

2) $P(t)$ satisfies the differential equation $\dot{P}(t) = -A^T(t)P(t) - P(t)A(t) - Q(t)$, where $Q(t)$ is continuous, symmetric, positive definite for all t, i.e. $Q(t) \geq c_3 I > 0$. Considering the same Lyapunov function for the unperturbed system will show that the unperturbed system (3.11) is exponentially stable if $\|e\| < r$ for some $r > 0$.

Let us consider the same Lyapunov function for the perturbed system and we obtain,

$$
\begin{aligned}
\dot{V}(t,e) &= \frac{d}{dt} e(t)^T P(t) e(t) \\
&= e^T (A^T(t)P(t) + P(t)A(t) + \dot{P}(t))e \\
&\quad + 2e^T P(t)(\mu(t,x,x_c)e + \delta(t,x,x(t_k))) \qquad (3.12) \\
&= -e^T Q(t) e + 2e^T P(t)(\mu(t,x,x_c)e + \delta(t,x,\{t_l\}_{l=0}^{\infty})).
\end{aligned}
$$

Using the definition of $\mu(t,x,x_c)$ given in Proposition 3.2.3 and the Lipschitz continuity assumption, (A2), on $F(t,x)$ we can write $\|\mu(t,x,x_c)\|_2 \leq (L_1 + \sum_{i=1}^{m} L_2^i)\|e\|_2 = L\|e\|_2$.

Therefore, from (3.12), we obtain,

$$
\begin{aligned}
\dot{V}(t,e) &\leq -c_3\|e\|_2^2 + 2Lc_2\|e\|_2^3 + 2c_2\|\delta\|_2\|e\|_2 \\
&= \|e\|_2(\|e\|_2 - \theta_1)(\|e\|_2 - \theta_2)
\end{aligned}
$$

where $\theta_1 = \frac{c_3 - \sqrt{c_3^2 - 16Lc_2^2\|\delta\|_2}}{4Lc_2}$, $\theta_2 = \frac{c_3 + \sqrt{c_3^2 - 16Lc_2^2\|\delta\|_2}}{4Lc_2}$.

If θ_1 and θ_2 are real and $\|e\|_2 \in [\theta_1,\ \theta_2]$, then $\dot{V}(t,e) \leq 0$ and hence $\Omega_\epsilon = \{e \in \mathbb{R}^n |\ \|e\|_2 \in [0,\ \theta_2]\}$ is an invariant set. To ensure θ_1, and θ_2 are real, we need $\|\delta\|_2 \leq \frac{c_3^2}{16Lc_2^2}$. By defining 2ϵ to be $\frac{c_3^2}{16Lc_2^2}$, we have $e(t) \in \Omega_\epsilon$ if $e(0) \in \Omega_\epsilon$ and $\|\delta\|_2 \leq 2\sum_{i=1}^{m} \|g_i(t,x)\gamma_i(y)\|_2 \leq 2\epsilon$. □

$\|e\|_2$ will be bounded from below by θ_1, however, θ_1 can be made arbitrarily small by controlling $\|\delta\|_2$.

Corollary 3.2.5. *Under the same hypothesis as in Theorem 3.2.4, the behavior of the event based closed loop system (3.7) remains in a bounded domain around the trajectory of the closed loop system (3.2).*

Corollary 3.2.5 follows from the fact that $x_c(t) - x(t) = e(t) \in \Omega_e$ and hence $\|x_c(t) - x(t)\|_2 = \|e\|_2 \in [0, \theta_2]$. This implies $x(t)$ remains is a domain $\Omega(x_c) = \{x \in \mathbb{R}^n | \ \|x_c - x\|_2 \in [0, \ \theta_2]\}$.

In Theorem 3.2.4, we have established an if-then relationship between the perturbation $\delta(t, x, \{t_l\}_{l=0}^{\infty}))$ and the error e. In the next theorem, we will state the exact relationship between the error $e(t)$ and the perturbation $\delta(t, x, \{t_l\}_{l=0}^{\infty})$.

Theorem 3.2.6. *Consider the dynamics given in (3.9) and suppose that we have a Lyapunov function $V(t, e)$ that satisfies*

$$c_1\|e\|_2^2 \leq V(t, e) \leq c_2\|e\|_2^2 \tag{3.13}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial e}(A(t)e + \mu(t, x, x_c)e) \leq -c_3\|e\|_2^2 \tag{3.14}$$

$$\left\| \frac{\partial V}{\partial e} \right\|_2 \leq c_4\|e\|_2 \tag{3.15}$$

for all $(t, e) \in [0, \infty) \times \mathbb{R}^n$ for some positive constants c_1, c_2, c_3 and c_4. Then,

$$\|e(t)\|_2 \leq \frac{c_4}{2c_1} \int_0^t e^{-(t-s)c_3/2c_2} \|\delta(s, x, \{t_l\}_{l=0}^{\infty})\|_2 ds \tag{3.16}$$

Proof. Let us consider the Lyapunov function described in the statement of this theorem and apply it for the system (3.9). We obtain,

$$\dot{V}(t,e) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial e}(A(t)e + \mu(t,x,x_c)e + \delta(t,x,\{t_l\}_{l=1}^m))$$
$$\leq -c_3\|e\|_2^2 + c_4\|e\|_2\|\delta\|_2 \tag{3.17}$$

Let $U(t) = \sqrt{V(t,e(t))}$ and whenever $V(t,e(t)) > 0$, we have $\dot{U}(t) = \frac{\dot{V}(t,e(t))}{2\sqrt{V(t,e(t))}}$ and using (3.13) and (3.17) we obtain,

$$\dot{U}(t) \leq -\frac{c_3}{2c_2}U(t) + \frac{c_4}{2\sqrt{c_1}}\|\delta\|_2 \tag{3.18}$$

We have, $V(t,x) = \int_0^x \frac{\partial V(t,y)}{\partial y}dy = \int_0^1 \frac{\partial V(t,sx)}{\partial x}dsx$ and hence using (3.15), $V(t,x) \leq c_4 \int_0^1 sds\|x\|_2^2 = \frac{c_4}{2}\|x\|_2^2$. This implies $c_4 \geq 2c_1$.
Therefore, when $V(t,e(t)) = 0$ (i.e. $e(t) = 0$), we have $V(t+h, e(t+h)) \leq \frac{c_4}{2}\|e(t+h)\|_2^2 = \frac{c_4}{2}\|\delta\|_2^2 h^2 + o(h^2)$; where $\lim_{h\to 0} o(h^2)/h^2 = 0$

$$D^+U(t) = \lim_{h\to 0^+} \frac{\sqrt{V(t+h,e(t+h))}}{h} \leq \frac{c_4}{2\sqrt{c_1}}\|\delta\|_2 \tag{3.19}$$

Therefore, using (3.18) and (3.19), we can write,

$$D^+U \leq -\frac{c_3}{2c_2}U(t) + \frac{c_4}{2\sqrt{c_1}}\|\delta\|_2 \tag{3.20}$$

Using Comparison Lemma [51, Section 3.4], we obtain,

$$U(t) \leq U(0)e^{-tc_3/2c_2} + \frac{c_4}{2\sqrt{c_1}} \int_0^t e^{(s-t)c_3/2c_2} \|\delta\|_2 ds \tag{3.21}$$

Since $e(0) = 0$, (3.21) and (3.13) ensure,

$$\|e(t)\|_2 \leq \frac{c_4}{2c_1} \int_0^t e^{-(t-s)c_3/2c_2} \|\delta(s, x, \{t_l\}_{l=0}^{\infty})\|_2 ds$$

□

3.3 Event-Triggering Strategy

Since piecewise constant control is used instead of continuous feedback to drive the system (3.1), the system will fluctuate from its expected behavior. We will implement an event-triggering strategy so that the system determines when the exact state $x(t)$ has to be transmitted to the control generator and the behavior of the system does not go beyond the tolerance level. Our goal is to keep $e(t)$ within the given tolerance level. Theorems 3.2.4 and 3.2.6 give explicit relation between the error $e(t)$ and the perturbation $\delta(t, x, \{t_l\}_{l=0}^{\infty})$ caused by control mismatch.

We consider a simple event-triggering function, based on the instantaneous value of $\delta(t, x, \{t_l\}_{l=0}^{\infty})$, given in (3.22):

$$f_{event}(\|\delta(t, x, \{t_l\}_{l=0}^{\infty})\|_2) = \epsilon - \|\delta\|_2 \tag{3.22}$$

where $\epsilon > 0$. Other variants of triggering functions are possible and we refer to some of them but due to space limitation, we proceed with our further analysis based on the stated triggering function given in (3.22). Analysis for other event triggering function such as (3.23) and (3.24) are similar and straight forward.

$$f^1_{event}(\|\delta\|_2) = \epsilon_1 + \epsilon_2 e^{-at} - \|\delta\|_2 \tag{3.23}$$

$$f^2_{event}(\|\delta\|_2) = \epsilon_3 - \int_{t_k}^{t} e^{-(t-s)c_3/2c_2}\|\delta\|_2 ds \tag{3.24}$$

where t_k is the time instance when the last event was triggered, $\epsilon_1, \epsilon_2, \epsilon_3$ and a are some design parameters which take nonnegative values.

The $k+1$-th event is generated when f_{event} (similarly f^1_{event} or f^2_{event}) attains a value of zero. The event-triggering mechanism (3.22) was used for event based control in several works like [64] and [41], whereas (3.23) was used in [41].

If the ϵ defined in (3.22) is same as that given in Theorem 3.2.4, we can guarantee that error will be bounded in a positive invariant set Ω_e or equivalently the event based state trajectory will be bounded in a domain around the closed loop state trajectory. For any other arbitrary ϵ, Theorem (3.2.6) ensures that $\|e(t)\|_2 \leq \frac{c_2 c_4}{c_1 c_3}\epsilon$ for all $t \in [0, \infty)$. Therefore, in any case, for all chosen ϵ there exist $r_2(\epsilon) > r_1(\epsilon) \geq 0$, such that $\Omega(x_c) = \{x \in \mathbb{R}^n | \; r_1 \leq \|x - x_c\|_2 \leq r_2\}$ is an invariant set. Similarly for the triggering mechanisms (3.23) and (3.24), there exist such invariant sets which depend on the choice of the design parameters $\epsilon_1, \epsilon_2, \epsilon_3$ and a.

Theorem 3.3.1. *Inter event time for the event-triggering mechanism(s) defined in (3.22) (or in (3.23) and (3.24)) is bounded from below.*

Proof. Clearly, due to the fact that $\gamma_i(x)$ is a Lipschitz continuous function, we have

$$\|\delta\|_2 \leq L_\gamma \sum_{i=0}^{m} \|g_i(t,x)\|_2 \|x(t) - x(t_k)\|_2$$

where $L_\gamma = \max\{L_\gamma^i |\ i = 1, 2, \cdots, m\}$. Since $g_i(t, \cdot)$ is a Lipschitz continuous function and $x(t)$ remains in a compact domain $(\Omega(x_c))$, $g_i(t, \cdot)$ is bounded for all t. Hence, we can write, $\sum\limits_{i=0}^{m} \|g_i(t,x)\|_2 \leq G_\infty < \infty$.

As a result, $\|x(t) - x(t_k)\|_2 \leq \frac{\epsilon}{G_\infty L_\gamma}$ ensures $\|\delta\|_2 \leq \epsilon$.

$$\begin{aligned} \|x(t) - x(t_k)\|_2 &\leq \left\| \int_{t_k}^{t} \left(f(s,x) + \sum_{i=1}^{m} g_i(s,x)\gamma_i(x(t_k)) \right) ds \right\|_2 \\ &\leq \int_{t_k}^{t} \left(\bar{L}\|x(s) - x(t_k)\|_2 + K(s) \right) ds \end{aligned} \tag{3.25}$$

where $\bar{L} = L_f + \sum\limits_{i=0}^{m} L_g^i \|\gamma_i(x(t_k))\|_2$ and $K(s) = \left\| f(s, x(t_k)) + \sum\limits_{i=0}^{m} \gamma_i(x(t_k)) g_i(s, x(t_k)) \right\|_2$.

Using Gröanwall-Bellman inequality for (3.25), we get,

$$\|x(t) - x(t_k)\|_2 \leq e^{\bar{L}(t-t_k)} \int_{t_k}^{t} K(s) ds \tag{3.26}$$

If $t_k + T$ is the time when the $k+1$-th event was triggered, then from (3.26)

$$e^{\bar{L}T} \int_{0}^{T} K(s + t_k) ds \geq \frac{\epsilon}{G_\infty L_\gamma} \tag{3.27}$$

Defining $\phi(T) = e^{\bar{L}T} \int\limits_{0}^{T} K(s + t_k) ds$, we have $\phi(0) = 0$ and $\phi(T)$ is continuous, increasing with finite $\dot{\phi}(T)$ for all $T \in [0, \infty)$. Therefore, there exists $T_{\min} > 0$ such that for all $T < T_{\min}$, $\phi(T) < \frac{\epsilon}{G_\infty L_\gamma}$ and hence, the inter event time is bound from below by $T_{\min}$, where $T_{\min}$ is such that $e^{\bar{L}T_{\min}} \int\limits_0^{T_{\min}} K(s + t_k)ds = \frac{\epsilon}{G_\infty L_\gamma}$. □

Theorem (3.3.1) suggests that the event-triggering mechanism does not exhibit Zeno behavior [105].

3.4 Simulation Results

3.4.1 Example 1: Interconnected Inverted Pendulums

The first example demonstrates the application of the event based nonlinear control scheme, presented in this chapter, on a network of inverted pendulums (Figure 3.1). The nonlinear dynamics for each pendulum is given in (3.28):

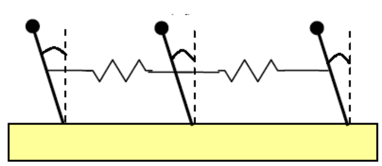


Figure 3.1: Three pendulums interconnected by springs. The angular positions are measured anticlockwise from the vertical axes. [95]

$$\dot{x}^i = \begin{bmatrix} x_2^i \\ \frac{g}{l}\sin(x_1^i) - \frac{a_i k}{ml^2} x_1^i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u^i + \begin{bmatrix} 0 \\ \sum_j \frac{h_{ij} k x_1^j}{ml^2} \end{bmatrix} \tag{3.28}$$

where x_1^i is the angular position of the i-th pendulum and x_2^i is the angular velocity. g is the acceleration due to gravity, l is the length and m is the mass of a pendulum, k is the spring constant and a_i is the number of springs attached to the i-th pendulum.

We consider the following parameter values to conduct the experiment: $g = 10, m = 1, l = 2$ and $k = 5$. The parameter values are taken from [95] and [41], where the authors considered the linearlized dynamics of the pendulums to study event based control of networked linear systems. $h_{ij} = 1$ if the i-th and j-th pendulums are connected by a spring, otherwise, $h_{ij} = 0$. Similar to the approach adopted in [95] and [41], we consider the control inputs to be as given below so that the poles of the closed loop system for each pendulum are at -1 and -2.

$$u^i = -(2ml^2 - k)x_1^i - mgl\sin(x_1^i) - 3ml^2x_2^i - kx_1^2$$

if $i = 1, 3$ and for $i = 2$

$$u^i = -2(ml^2 - k)x_1^i - mgl\sin(x_1^i) - 3ml^2x_2^i - k(x_1^1 + x_1^3).$$

These control laws linearize the system and decouple each subsystem. A candidate Lyapunov function of the form $(x^i)^T P^i x^i$ for each subsystem can be chosen independently. We choose $P^i = P = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ which gives $Q = \begin{bmatrix} -4 & -3 \\ -3 & -4 \end{bmatrix}$. Therefore, $c_1 = 0.38, c_2 = 2.62, c_3 = 1$ and $c_4 = 5.24$. We choose the value of ϵ to be 0.05. The initial condition for the system is chosen to be $[\pi/3, 0, -\pi/5, 0, -2\pi/3, 0]$. The behavior of the closed loop system and event based system are plotted in Figure 3.2.

The errors associated with each dimension are plotted in Figure 3.3 where the event-triggering instances are also shown. Total number of events triggered is 49 and the average interval between two events is 0.4083.

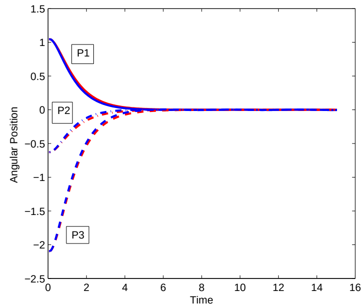


Figure 3.2: The red curves show the behavior under continuous feedback and the blue ones for event based feedback. P1, P2 and P3 correspond to pendulum 1, pendulum 2 and pendulum 3 respectively.

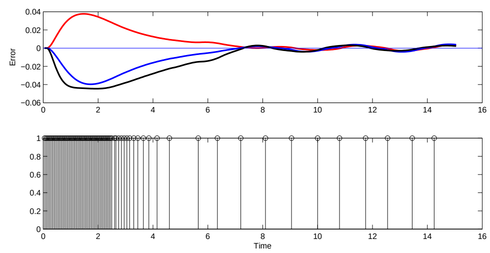


Figure 3.3: (upper) The red curve is for the error in angular position of pendulum 1, the blue and black ones are for the same for pendulum 2 and pendulum 3 respectively. (lower) The event-triggering profile.

3.4.2 Example 2

In the second experiment we consider the following nonlinear dynamics:

$$\dot{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} = \begin{bmatrix} -\sin(x_1) \\ -x_2 \end{bmatrix} + \begin{bmatrix} -x_2 \\ x_1 \end{bmatrix} u \tag{3.29}$$

This system cannot be linearized for any choice of the control input u but with $u = -x_2$, we can stabilize the system around the origin. A Lyapunov function $V(t,x) = x_1^2 + x_2^2$ proves that the closed loop system is exponentially stable. We choose different initial conditions for this system to observe how the event based system differs from the closed loop system. We choose twelve different initial conditions as shown in

Figure 3.4. The initial conditions are chosen in such a way that two of them lie on the $x_2 = 0$ line and five of them are the reflections of other five about the $x_2 = 0$ axis.

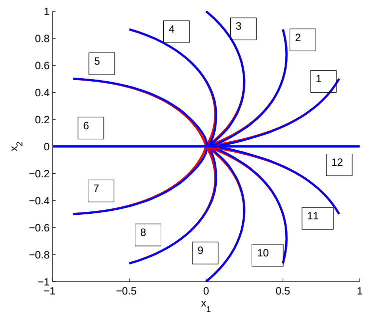


Figure 3.4: The red curves are for the continuous feedback system and the blue curves are for the event based system. All the trajectories converge to the equilibrium point at the origin. There are twelve different initial positions as numbered in the figure.

The event-triggering profile for the different initial positions are shown in Figure 3.5. The dynamics is symmetric about the $x_2 = 0$ axis with the chosen control. This symmetry is also reflected in the event-triggering pattern. So we only plot the event-triggering patterns for the first seven initial conditions shown in Figure 3.4. The triggering patterns for the 5-th and 7-th initial conditions are same since the initial conditions mirror each other. No further event is triggered after the first one at $t = 0$ for the 6-th and 12-th initial conditions.

3.5 Conclusions

In this chapter, we have proposed an event-based control strategy for an input affine nonlinear system. We use an event-triggering strategy to ensure that the error remains in a bounded domain, and as a consequence, the event based system

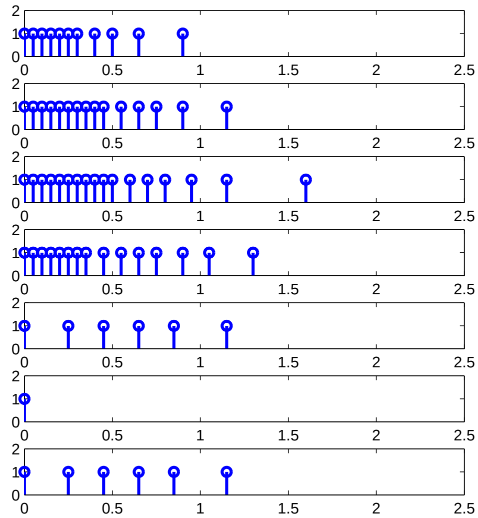


Figure 3.5: We only show event-triggering profile for the first seven intial conditions, rest of them are similar to their mirroring initial conditions. Intial conditions on the line $x_2 = 0$ do not require any triggering (expect the one at $t = 0$ to set the initial values) because $u \equiv 0$. Event-triggering profile is same for intial conditions 5 and 7 since they mirror each other.

approximates the behavior of the continuous state feedback system. The threshold based nature triggering strategy for this non-linear system is similar to the optimal triggering strategy for the linear systems proposed in the previous chapter.

Simulation results show the application of event based strategy on two input affine nonlinear systems. Theorem 3.2.6 gives the explicit expression on the boundedness of the error $e(t)$. Theorem 3.3.1 also shows a relation between the inter event time and the error bound. The optimal error bound can be selected based on the precision needed and the communication resources available for triggering.

Possible extensions could be to analyse event-triggering controller for a general nonlinear control systems of the form $\dot{x} = f(t, x, u)$, and include time delays and dropouts into the network.

Chapter 4: Event-Triggered Optimal Control of Nonlinear Stochastic Systems

In this chapter we study the classical stochastic nonlinear optimal control in an intermittent feedback setting. It is well known that the solution of such a problem is determined by solving the Hamilton-Jacobi-Bellman (HJB) partial deferential equation (PDE). Solution of the HJB results in a feedback control policy, and that is not desirable in this case as we hope for intermittent feedback policy. In this chapter we show that the optimal intermittent feedback policy needs a PDE similar to HJB to be solved. Furthermore, we show that there is a connection between the PDE corresponding to optimal intermittent feedback and the HJB equation for the continuous feedback counterpart of the same problem.

4.1 Introduction & Literature Review

In Chapter 2 we discussed an event-triggered control policy for a linear non-deterministic system and subsequently in Chapter 3 we extended the similar idea for a class of nonlinear systems, namely the control-affine class. Like Chapter 2, the focus is to design optimal control for a stochastic nonlinear system that entails only finite number of state measurements. As the disturbance in the dynamics for the problem in Chapter 2 is non-deterministic and no further information was available except the optimal strategy required active sensing to perform the event-triggering mechanism. In this situation, as we are interested in optimizing the expected cost and the dynamics is subjected to a Wiener noise with known distribution, the optimal instances for triggering the state sampling can be computed offline and hence it does not require an active sensing.

As we presented in the last two chapters, event triggered control have gained increasing popularity over the time due the qualities such as reduced sensing, reduced communication, and often times reduced computation. As presented in [3] and [30], the event triggered control framework has been adapted to the nonlinear control settings. Event-triggered control has also been combined with a model predictive control framework in [60]. Although, in those works, the focus was to develop an event-trigger strategy for certain class of nonlinear systems to achieve properties like stability and consensus; and the optimality of the control law or the event-triggering strategy was not the primary focus of these studies. Similarly, the large body of past works on event-trigger control do not deal with the optimality aspects: [89], [90], [91], [94], [95], [96] and [97].

The closest work on this topic that addresses the optimality aspects both in control and in the number of triggering is studied in [72]. In this work the authors considered an continuous time stochastic LQG optimal control problem and the measurements were obtained intermittently in a discrete-time manner. To an extent, the motivation of this chapter is obtained from this work. Since LQG is a special case of the stochastic nonlinear control problem, we reproduce the findings of [73] from our results. Another work that attempted a similar problem is [80], where the control had two values, and the optimal trigger instance decided the transition from one value to the other. There framework only dealt with one transition for the entire horizon of the problem.

The organization of this chapter is as follows: In Section 4.3 we formulate the general problem and whose solution is approached in Section 4.4. As the final optimization problem is not always possible to solve analytically, we consider some special cases in Section 4.5. Finally, we conclude this chapter in Section 8.6.

4.2 Notation

$x(t)$: the state of system, $dw(t)$: Brwonian motion Noise, t_i: event-triggering instance, $\mathcal{E}$: event generator (set of triggering times t_i), $\mathcal{C}$: controller, $\mathfrak{I}(t)$: σ-algebra generated by the measurements obtained until time t, $\pi(t,x)$: density of the state $x(t)$ given $\mathfrak{I}(t)$, $\tilde{\pi}(t,x,s)$: density of the state $x(t)$ given $\mathfrak{I}(s)$ where $s \leq t$. $\langle \cdot, \cdot \rangle$: standard inner product (in L_2).

4.3 Problem Formulation

Let us consider the following nonlinear stochastic dynamics:

$$dx = f(t,x,u)dt + \sigma(t,x,u)dw(t) \tag{4.1}$$

$dw(t)$ standard Brownian motion noise in $\mathbb{R}^r$ with $\mathbb{E}[dw(t)dw(t)'] = I_r dt$.

The finite horizon cost function that we are interested to minimize is:

$$J(0,x_0) = \mathbb{E}\Big[\int_0^T l(s,x(s),u(s))ds + g(T,x(T))\ |x(0)=x_0\Big].$$

The solution of this standard nonlinear stochastic optimal control is characterized by the value function that satisfies the Hamilton-Jacobi-Bellman (HJB) equation (4.2):

$$-\frac{\partial V}{\partial t} = \min_u \{l(t,x,u) + \frac{\partial V}{\partial x} f(t,x,u) + tr(\frac{\partial^2 V}{\partial x^2} a(t,x,u))\} \tag{4.2}$$

where $a = \sigma\sigma'$, $V(T,x) = 0$, and

$$V(t,x) = \mathbb{E}\left[\int_t^T l(s,x(s),u(s))ds + g(T,x(T)) \mid x(t) = x\right]$$

The optimal u^* is found by solving (4.2), and hence u^* is a feedback control law.

In this work, we address the above mentioned optimal control problem in an event-triggered framework. By event-trigger, we mean that that continuous feedback is not possible and only intermittent on-demand feedback is allowed. The schematic diagram is as presented in Figure 4.1 where, in contrast to classical control setup, we have two decisions to be made at the controller side: first, optimal time instances to ask for the measurements and second, the optimal control law to steer the dynamics (4.1). In this continuous time framework, we only allow instantaneous measurements. The controller needs to pay certain cost in order to receive

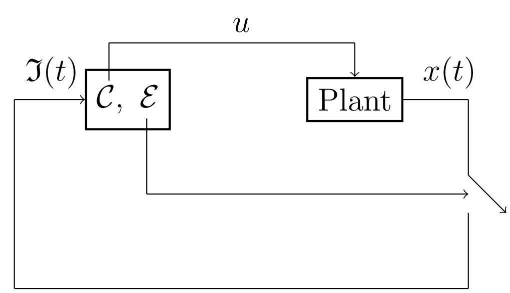


Figure 4.1: Schematic representation of the system.

each measurement. We assume that the communication link used to transmit the measurement from the sensors to the controller is noiseless and delay-free, i.e., whenever the controller receives the measurement it is a perfect copy of the state at that time instance.

The event-generator $\mathcal{E}$ generates the optimal time instances $\{t_i\}_{i\geq 1}$ at when the measurements are transmitted to the controller. In this work we assume[1] that the controller has the perfect knowledge on the initial state $x(0)$, and we denote $t_0 = 0$. Let at time t, $n(t)$ be the number of measurements that have been transmitted i.e. the controller $\mathcal{C}$ has the information $\{x(t_i)\}_{i=0}^{n(t)}$. Thus, the control $u(t)$ generated by the controller $\mathcal{C}$ is measurable with respect to the (σ)-algebra generated by the random variables $\{x(t_i)\}_{i=0}^{n(t)}$. Let us denote $\mathfrak{I}(t)$ to be the algebra generated by $\{x(t_i)\}_{i=0}^{n(t)}$. Similarly, the optimal time-instances generated by $\mathcal{E}$ at time t is also $\mathfrak{I}(t)$ measurable.

Therefore, the new optimization problem for our case is:

[1]This assumption can be easily relaxed when we have only the distribution $p_0(\cdot)$ as opposed to the actual value of $x(0)$.

$$\min_{\mathcal{E}=\{t_i\}_{i\geq 1},\mathcal{C}=u(\cdot)} \mathbb{E}\Big[\int_0^T l(s,x(s),u(s))ds + g(T,x(T)) + \lambda|\mathcal{E}| \;\Big|\mathfrak{I}(0) = x_0\Big]$$

where $|\mathcal{E}|$ denotes the cardinality of the set $\mathcal{E}$ i.e. the number of measurements transmitted. $\lambda > 0$ is the cost associated with each transmission of the measurements.

The optimal cost-to-go for any arbitrary time $t > 0$ can be written as:

$$\min_{\mathcal{E}(t),\mathcal{C}} \mathbb{E}\Big[\int_0^T l(s,x(s),u(s))ds + g(T,x(T)) + \lambda|\mathcal{E}| \;\Big|\mathfrak{I}(t)\Big]. \tag{4.3}$$

We restrict ourselves to the class of $\mathcal{E}$ that produces only finite number of triggerings in the interval $[0,T]$. We also assume that the joint minimization in (4.3) can be written as a nested optimization as shown in (4.4)

$$J^*(t) = \min_{\mathcal{E}(t)} \{\mathbb{E}\Big[\min_{\mathcal{C}} \mathbb{E}\Big[\int_t^T l(s,x(s),u(s))ds + g(T,x(T)) \;|\mathfrak{I}(t)\Big] + \lambda|\mathcal{E}(t)| \mid \mathfrak{I}(t)\Big]\} \tag{4.4}$$

where $\mathcal{E}(t) = \{t_1 < t_2 < \cdots \mid t_1 \geq t\}$.

In (4.4), we have separated the control cost and measurement acquisition costs, and we define the optimal control cost as:

$$J_c^*(t) = \min_u \mathbb{E}\Big[\int_t^T l(s, x(s), u(s))ds + g(T, x(T)) \; |\mathfrak{I}(t)\Big] \tag{4.5}$$

and the measurement cost as:

$$J_m(t) = \mathbb{E}\Big[\lambda|\mathcal{E}(t) \; |\mathfrak{I}(t)\Big]. \tag{4.6}$$

The optimal control cost $J_c^*(t)$ depends on $\mathcal{E}(t)$ since the optimal control u^* depends on the information $\mathfrak{I}$ which in turn is decided by $\mathcal{E}$.

4.4 Solution Approach

Since $x(t)$ is not available to synthesize control at time t, therefore it is customary in stochastic control to introduce *information state*, $\pi(t, \cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$, on which the control depends i.e. $u(t) = k(t, \pi(t, \cdot))$.

The information state is generally the probability density of the state given the information to the controller i.e. $\pi(t, x) = \mathbb{P}^u(x(t) = x \mid \mathfrak{I}(t))$[2]. Let us denote the quantity $\tilde{\pi}(t, x, s) = \mathbb{P}^u(x(t) = x \,|\mathfrak{I}(s))$ for all $s \le t$, and thus $\pi(t, x) = \tilde{\pi}(t, x; t)$ for all $x \in \mathbb{R}^{n_x}$, and $\tilde{\pi}(t, \cdot; s) = \mathbb{E}_{\sim\mathfrak{I}(t)}[\pi(t, \cdot) \; |\mathfrak{I}(s)]$[3].

Lemma 4.4.1. *The evolution of $\tilde{\pi}(t, x; s)$ for all $t \ge s$ can be characterized by the*

[2]Here we make an abuse of notation; precisely, $\pi(t, x)dx = \mathbb{P}^u(x(t) \in dx \mid \mathfrak{I}(t))$. Similar for the definition of $\tilde{\pi}(t, x, s)$

[3]The notation $\mathbb{E}_{\sim Y}[X]$ defines the expectation of the quantity X with respect to the random variable Y

following special form of Fokker-Planck equation:

$$\frac{\partial \tilde{\pi}(t,x;s)}{\partial t} = \mathcal{L}^u(\tilde{\pi}(t,x;s))$$
$$\tilde{\pi}(s,x,s) = \pi(s,x);$$

Moreover,

$$\frac{\partial \pi(t,x)}{\partial t} = \mathcal{L}^u(\pi(t,x)) \tag{4.7}$$
$$\pi(t_i,x) = {}^4\delta(x(t_i)-x);$$

where $\{t_i\}_{i\geq 0}$ *are the triggering time instances.*

Proof. If t_i be an instant when the measurement arrives to the controller, then trivially,

$$\pi(t_i,x) = \mathbb{P}^u(x(t_i) = x | X(t_i) = x(t_i)) = \delta(x(t_i) - x)$$

Let also that t_i and $t_{i+1} > t_i$ be two consecutive sampling instances. For all $t \in (t_i, t_{i+1})$, let us consider a function $h(x(t))$ where $x(t)$ satisfies the SDE 4.1. Therefore, by Itô's rule,

$$dh(x) = \nabla h(x)(f(t,x,u)dt + \sigma(t,x,u)dw(t)) + \frac{1}{2}tr\Big(a(t,x,u)\frac{\partial^2 h(x)}{\partial x^2}\Big)dt.$$

[4] $\tilde{\pi}(0,x,0) = p_0(x)$ when initial distribution $(p_0(\cdot))$ of $x(0)$ is available instead of the actual value.

Let us define an operator $\mathcal{L}_*^u$ such that:

$$\mathcal{L}_*^u\big(h(x)\big) = \nabla h(x) f(t,x,u) + \frac{1}{2} tr\big(a(t,x,u)\frac{\partial^2 h(x)}{\partial x^2}\big)$$

Thus,

$$h(x(t) - h(x(t_i)) = \int_{t_i}^{t} \nabla h(x)\sigma(s,x,u)dw(s) + \int_{t_i}^{t} \mathcal{L}_*^u(h(x))ds$$

Taking expectation on both sides:

$$\mathbb{E}[h(x(t)) - h(x(t_i)) \mid \mathfrak{I}(t)] = \int_{t_i}^{t} \mathbb{E}[\mathcal{L}_*^u(h(x))ds]$$

$$\int_{\mathbb{R}^n} h(y)\pi(t,y)dy - h(x(t_i)) = \int_{t_i}^{t}\int_{\mathbb{R}^n} \mathcal{L}_*^u(h(x))\pi(s,x)dxds \tag{4.8}$$

Let us denote the adjoint operator of $\mathcal{L}_*^u$ by $\mathcal{L}^u$ in the following way:

$$\int_{\mathbb{R}^n} \mathcal{L}^u(h_1(x))h_2(x)dx = \int_{\mathbb{R}^n} \mathcal{L}_*^u(h_2(x))h_1(x)dx$$

Therefore, from (4.8), one can write:

$$\int_{\mathbb{R}^n} h(y)\pi(t,y)dy - h(x(t_i)) = \int_{t_i}^{t}\int_{\mathbb{R}^n} h(x)\mathcal{L}^u\big(\pi(s,x)\big)dxds$$

Taking derivative w.r.t. t on both sides:

$$\int_{\mathbb{R}^n} h(y)\frac{\partial \pi(t,y)}{\partial t}dy = \int_{\mathbb{R}^n} h(x)\mathcal{L}^u\big(\pi(t,x)\big)dx$$

□

Since the above relation holds for any twice differentiable (L_2) function, we can conclude that $\pi(t,x)$ is a week solution of the PDE (4.7).

Similarly, one can the prove the PDE associated with $\tilde{\pi}(t,x,s)$.

Theorem 4.4.2 (Sufficiency for Optimal control)**.** *Let there exist a twice differentiable function $V(t,x):[0,T]\times\mathbb{R}^{n_x}\to\mathbb{R}$, such that for all $0\le t\le T$:*

$$
\begin{aligned}
-\Big\langle \frac{\partial V}{\partial t}(t,\cdot),\pi(t,\cdot)\Big\rangle &= \min_u \Big\langle l(t,\cdot,u)+\nabla V(t,\cdot)f(t,\cdot,u) \\
&\quad + \frac{1}{2}tr\big(a(t,\cdot,u)\frac{\partial^2 V}{\partial x^2}(t,\cdot)\big),\pi(t,\cdot)\Big\rangle \qquad (4.9)\\
V(T,x) &= g(T,x),
\end{aligned}
$$

and define:

$$
\mathcal{J}(t,s)=\langle V(t,\cdot),\tilde{\pi}(t,\cdot;s)\rangle.
$$

where $\langle f(\cdot),g(\cdot)\rangle=\int_{\mathbb{R}^n}f(x)g(x)dx$.

Then, $J_c^*(s)=\mathcal{J}(s,s)$ and the optimal u^* minimizes the right-hand-side of (4.9).

Proof. We start by differentiating $\mathcal{J}(t,s)$ w.r.t. first argument t:

$$
\frac{\partial\mathcal{J}}{\partial t}=\Big\langle\frac{\partial V(t,\cdot)}{\partial t},\tilde{\pi}(t,\cdot;s)\Big\rangle+\Big\langle V(t,\cdot),\frac{\partial\tilde{\pi}(t,\cdot;s)}{\partial t}\Big\rangle.
$$

From (4.9), we have:

$$-\mathbb{E}_{\sim\mathfrak{I}(t)}\Big[\Big\langle\frac{\partial V}{\partial t}(t,\cdot),\pi(t,\cdot)\Big\rangle\ |\mathfrak{I}(s)\Big] = \mathbb{E}_{\sim\mathfrak{I}(t)}\Big[\min_u\Big\langle l(t,\cdot,u)+\nabla V(t,\cdot)f(t,\cdot,u) + \frac{1}{2}tr\big(a(t,\cdot,u)\frac{\partial^2 V}{\partial x^2}(t,\cdot)\big),\pi(t,\cdot)\Big\rangle\ |\mathfrak{I}(s)\Big],$$

which leads to

$$\begin{aligned}-\Big\langle\frac{\partial V}{\partial t}(t,\cdot),\tilde{\pi}(t,\cdot;s)\Big\rangle &= \min_u\Big\langle l(t,\cdot,u)+\nabla V(t,\cdot)f(t,\cdot,u)\\ &\quad+\frac{1}{2}tr\big(a(t,\cdot,u)\frac{\partial^2 V}{\partial x^2}(t,\cdot)\big),\tilde{\pi}(t,\cdot;s)\Big\rangle\\ &= \min_u\Big\langle l(t,\cdot,u)+\mathcal{L}^u_*(V(t,\cdot)),\tilde{\pi}(t,\cdot;s)\Big\rangle.\end{aligned}$$

Using the fact that $\mathcal{L}^u$ and $\mathcal{L}^u_*$ are adjoint operators under the inner product $\langle\cdot,\cdot\rangle$, we can write:

$$\begin{aligned}\Big\langle V(t,\cdot),\frac{\partial\tilde{\pi}(t,\cdot;s)}{\partial t}\Big\rangle &= \Big\langle V(t,\cdot),\mathcal{L}^u(\tilde{\pi}(t,\cdot;s))\Big\rangle\\ &= \Big\langle\mathcal{L}^u_*(V(t,\cdot)),\tilde{\pi}(t,\cdot;s)\Big\rangle\end{aligned}$$

Thus, combining the two results,

$$\frac{\partial\mathcal{J}}{\partial t} = -\min_u\Big\langle l(t,\cdot,u)+\mathcal{L}^u_*(V(t,\cdot)),\tilde{\pi}(t,\cdot;s)\Big\rangle+\Big\langle\mathcal{L}^u_*(V(t,\cdot)),\tilde{\pi}(t,\cdot;s)\Big\rangle.$$

Let us denote:

$\mathcal{H}(t,u) = \Big\langle l(t,\cdot,u) + \mathcal{L}_*^u(V(t,\cdot)), \tilde{\pi}(t,\cdot,s)\Big\rangle$, and consequently,

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial t} + \Big\langle l(t,\cdot,u), \tilde{\pi}(t,\cdot;s)\Big\rangle &= \mathcal{H}(t,u) - \min_u \mathcal{H}(t,u) \\ &\geq 0 \end{aligned} \tag{4.10}$$

where the equality holds for $u \equiv u^*$ that minimizes the Hamiltonian $\mathcal{H}$.

Integrating (4.10) from $\sigma(\geq s)$ to T leads to:

$$\begin{aligned}\mathcal{J}(T,s) - \mathcal{J}(\sigma,s) + \int_\sigma^T \Big\langle l(t,\cdot,u), \tilde{\pi}(t,\cdot;s)\Big\rangle dt &\geq 0. \\ \mathcal{J}(\sigma,s) \leq \Big\langle g(T,x), \tilde{\pi}(T,\cdot;s)\Big\rangle + \int_\sigma^T \Big\langle l(t,\cdot,u), \tilde{\pi}(t,\cdot;s)\Big\rangle & \\ = \mathbb{E}\Big[\int_\sigma^T l(t,x,u)dt + g(T,x(T)) \ |\mathfrak{I}(s)\Big] & \\ \mathcal{J}(\sigma,s) \leq \min_u \mathbb{E}\Big[\int_\sigma^T l(t,x,u)dt + g(T,x(T)) \ |\mathfrak{I}(s)\Big]. &\end{aligned}$$

Furthermore, substituting $u = u^*$ (the optimal control minimizing $\mathcal{H}$) in (4.10) and integrating, one obtains:

$$\mathcal{J}(\sigma,s) = \mathbb{E}\Big[\int_\sigma^T l(t,x,u^*)dt + g(T,x(T)) \ |\mathfrak{I}(s)\Big].$$

Therefore, the control u that minimizes $\mathbb{E}\Big[\int_\sigma^T l(t,x,u)dt + g(T,x(T)) \ |\mathfrak{I}(s)\Big]$ must be equal to u^* with almost sure probability. Thus,

$$\begin{aligned} J_c^*(s) &= \min_u \mathbb{E}\Big[\int_s^T l(t,x,u)dt + g(T,x(T)) \; |\mathfrak{I}(s)\Big] \\ &= \mathcal{J}(s,s). \end{aligned}$$

and the optimal u^* is found by solving (4.9). □

The challenging part in order to compute optimal control is to solve the coupled PDEs of $V(t,x)$ and $\pi(t,x)$. However, this is a common fact in nonlinear stochastic optimal control and there is no-way around it.

Remark 4.4.3. *For the case when we have perfect state measurement $x(t)$ for all time t, then $\pi(t,x) = \delta(x(t)-x)$. Thus (4.9) can be written as:*

$$\begin{aligned} -\Big\langle \frac{\partial V}{\partial t}(t,\cdot), \delta(x(t)-\cdot)\Big\rangle &= \min_u \Big\langle l(t,\cdot,u) + \nabla V(t,\cdot) f(t,\cdot,u) \\ &\quad + \frac{1}{2} tr\Big(a(t,\cdot,u)\frac{\partial^2 V}{\partial x^2}(t,\cdot)\Big), \delta(x(t)-\cdot)\Big\rangle \\ -\frac{\partial V}{\partial t}(t,x(t)) &= \min_u \{ l(t,x(t),u) + \nabla V(t,x(t)) f(t,\cdot,u) \\ &\quad + \frac{1}{2} tr\Big(a(t,x(t),u)\frac{\partial^2 V}{\partial x^2}(t,x(t))\Big)\} \end{aligned}$$

which the classical HJB equation for the prefect measurement case.

Since $J_c^*(t) = \mathcal{J}(t,t) = \int V(t,x)\pi(t,x)$, we now have

$$J^*(t) = \min_{\mathcal{E}(t)} \mathbb{E}\big[V(t,x) + \lambda|\mathcal{E}| \mid \mathfrak{I}(t)\big] \tag{4.11}$$

where $V(t,x)$ satisfies the functional PDE

$$-\Big\langle \frac{\partial V}{\partial t}(t,\cdot), \pi(t,\cdot)\Big\rangle = \min_u \Big\langle l(t,\cdot,u) + \nabla V(t,\cdot) f(t,\cdot,u) \\ + \frac{1}{2} tr\Big(a(t,\cdot,u)\frac{\partial^2 V}{\partial x^2}(t,\cdot)\Big), \pi(t,\cdot)\Big\rangle. \qquad (4.12)$$

Solving (4.9) is non-trivial and even more challenging than solving a HJB for a general stochastic control problem under perfect state observation. In the following proposition we show that the solution of (4.9) can be constructed from the solution of an HJB for an equivalent perfectly observable stochastic control problem.

Lemma 4.4.4. *The solution of* (4.9) *can be written as* $V(t,x) = W(t,x) + \int_t^T \psi(s)ds$ *where* $\psi(t)$ *depends on* $\pi(t,\cdot)$ *and* $W(t,x)$*.* $W(t,x)$ *is the solution of the HJB equation under perfect observation, i.e.,*

$$-\frac{\partial W}{\partial t}(t,x) = \min_u \Big(l(t,x,u) + \nabla W(t,x) f(t,x,u) \\ + tr\Big(a(t,x,u)\frac{\partial^2 W}{\partial x^2}(t,x)\Big)\Big) \qquad (4.13)$$
$$W(T,x) = g(T,x).$$

Proof. It can trivially be checked that the terminal condition $V(T,x) = g(T,x)$ holds.

Let $u = k(x)$ achieves the minimization in (4.13). Moreover, let us define $\psi(t)$

as:

$$\psi(t) = \min_u \Big\langle l(t,\cdot,u) + \nabla W(t,\cdot) f(t,\cdot,u) + tr\Big(a(t,\cdot,u)\frac{\partial^2 W}{\partial x^2}(t,\cdot)\Big), \pi(t,\cdot)\Big\rangle$$
$$- \Big\langle l(t,\cdot,k(x)) + \nabla W(t,\cdot) f(t,\cdot,k(x)) + tr\Big(a(t,\cdot,k(x))\frac{\partial^2 W}{\partial x^2}(t,\cdot)\Big), \pi(t,\cdot)\Big\rangle$$

Considering $V(t,x) = W(t,x) + \int_t^T \psi(s)ds$, we obtain:

$$-\Big\langle \frac{\partial V}{\partial t}(t,\cdot), \pi(t,\cdot)\Big\rangle = -\Big\langle \frac{\partial W}{\partial t}(t,\cdot), \pi(t,\cdot)\Big\rangle + \psi(t)$$

Using the definition of $\psi(t)$, and substituting it in the above equation one obtains:

$$\begin{aligned} -\Big\langle \frac{\partial V}{\partial t}(t,\cdot), \pi(t,\cdot)\Big\rangle = & \min_u \Big\langle l(t,\cdot,u) + \nabla W(t,\cdot) f(t,\cdot,u) \\ & + tr\Big(a(t,\cdot,u)\frac{\partial^2 W}{\partial x^2}(t,\cdot)\Big), \pi(t,\cdot)\Big\rangle \\ & - \Big\langle \frac{\partial W}{\partial t}(t,\cdot) + l(t,\cdot,k(x)) \\ & + \nabla W(t,\cdot) f(t,\cdot,k(x)) \\ & + tr\Big(a(t,\cdot,k(x))\frac{\partial^2 W}{\partial x^2}(t,\cdot)\Big), \pi(t,\cdot)\Big\rangle \end{aligned}$$

Using (4.13) along with the facts that $\nabla W = \nabla V$, $\frac{\partial^2 W}{\partial x^2}(t,x) = \frac{\partial^2 V}{\partial x^2}(t,x)$, we can obtain:

$$\begin{aligned} -\Big\langle \frac{\partial V}{\partial t}(t,\cdot), \pi(t,\cdot)\Big\rangle = & \min_u \Big\langle l(t,\cdot,u) + \nabla V(t,\cdot) f(t,\cdot,u) \\ & + tr\Big(a(t,\cdot,u)\frac{\partial^2 V}{\partial x^2}(t,\cdot)\Big), \pi(t,\cdot)\Big\rangle \end{aligned}$$

which is precisely (4.9). ☐

Corollary 4.4.5. *For all* $t \in [0, T]$ *and for all* $x \in \mathbb{R}^n$, $V(t, x) \geq W(t, x)$

Proof. The corollary is proved by showing that $\psi(t) \geq 0$ for all t. For the sake of brevity, let us denote:

$$\phi(t, x, u) = l(t, x, u) + \nabla W(t, x) f(t, x, u) + tr\Big(a(t, x, u)\frac{\partial^2 W}{\partial x^2}(t, x)\Big)$$

Since $u = k(x)$ is an optimal control for (4.13),

$$\phi(t, x, k(x)) \leq \phi(t, x, u)$$

for all x and u. Thus, for all u

$$\begin{aligned}\big\langle \phi(t, \cdot, k(\cdot)), \pi(t, \cdot)\big\rangle &\leq \big\langle \phi(t, \cdot, u), \pi(t, \cdot)\big\rangle \\ &\leq \min_u \big\langle \phi(t, \cdot, u), \pi(t, \cdot)\big\rangle.\end{aligned}$$

Therefore,

$$\psi(t) = \min_u \big\langle \phi(t, \cdot, u), \pi(t, \cdot)\big\rangle - \big\langle \phi(t, \cdot, k(\cdot)), \pi(t, \cdot)\big\rangle \geq 0 \tag{4.14}$$

Thus $\int_t^T \psi(s)ds \geq 0$ for all t, and hence $V(t, x) \geq W(t, x)$ for all t, x. ☐

It should be noted that the choice of triggering instances t_i affects the value function $V(t, x)$ through $\psi(t)$ since $W(t, x)$ does not depend on the event-triggering strategy $\mathcal{E}(t)$. From the expression of $\psi(t)$, one can conclude is that $\psi(t)$ is right continuous[5] and $\psi(t_i) = 0$ at all triggering instances t_i. Recall from (4.11) that

$$\begin{aligned} J^*(t) &= \min_{\mathcal{E}(t)} \mathbb{E}\big[V(t,x) + \lambda|\mathcal{E}| \mid \mathfrak{I}(t)\big] \\ &= \min_{\mathcal{E}(t)} \mathbb{E}\big[W(t,x) + \int_t^T \psi(s)ds + \lambda|\mathcal{E}| \mid \mathfrak{I}(t)\big]. \end{aligned}$$

Thus the optimal cost

$$J^*(0) = \mathbb{E}[W(0,x) \mid \mathfrak{I}(0)] + \min_{\mathcal{E}} \mathbb{E}\big[\int_0^T \psi(s)ds + \lambda|\mathcal{E}| \mid \mathfrak{I}(0)\big]$$

Therefore, the optimization problem to find the optimal triggering policy $\mathcal{E}(t)$ can be written as:

Problem 4.4.6.

$$\begin{aligned} &\min_{\mathcal{E}} \mathbb{E}\big[\int_0^T \psi(s)ds + \lambda|\mathcal{E}| \mid \mathfrak{I}(0)\big] \\ s.t.\quad &\psi(t) + \big\langle \phi(t,\cdot,k(\cdot)), \pi(t,\cdot)\big\rangle - \min_u \big\langle \phi(t,\cdot,u), \pi(t,\cdot)\big\rangle = 0 \\ &\frac{\partial \pi(t,x)}{\partial t} = \mathcal{L}^u(\pi(t,x)) \\ &\pi(t_i,x) = \delta(x(t_i) - x). \end{aligned}$$

Solving Problem 4.4.6 is non-trivial, however, for certain class of problems (precisely Linear Quadratic Gaussian), the problem can be solved explicitly and it

[5]since $\pi(t,\cdot)$ is right continuous and $\phi(t,\cdot,\cdot)$ is continuous for all t

will be demonstrated in Section 4.5.

4.5 Special cases

4.5.1 Linear Quadratic Gaussian Control

In this section we illustrate the application of Theorem 4.4.2 for a linear quadratic Gaussian case as described below:

$$dx = Axdt + Budt + \sigma dw(t)$$
$$x(0) \sim \mathcal{N}(\mu, \Sigma)$$

and the cost function to be optimized is:

$$J = \mathbb{E}\left[\int_0^T (x'Lx + u'Ru)dt + x(T)'Qx(T) + \lambda|\mathcal{E}(t)|\right].$$

The probability density $\pi(t,x)$ will follow the Fokker-Planck equation (with resetting):

$$\begin{aligned}\frac{\partial \pi(t,x)}{\partial t} =& -\nabla\cdot(\pi(t,x)(Ax+Bu)) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2}{\partial x_i \partial x_j}(\pi(t,x)a_{ij})\\ \pi(t_i,x) =& \delta(x(t_i)-x)\end{aligned}$$

where $\nabla\cdot\varphi$ computes the divergence of the vector valued function $\varphi(x)$, and a_{ij} is the ij-th component of the matrix $\sigma\sigma'$. Since, it is well known that the density function will be Gaussian, we can represent the solution of the above PDE as:

$$\pi(t,x) = \alpha e^{\frac{-(x-m(t))'K(t)^{-1}(x-m(t))}{2}}$$

where the normalizing constant $\alpha = (2\pi)^{-n/2}(\det(K(t)))^{-1/2}$ where in this expression π is the mathematical constant π and not the distribution $\pi(t,x)$.

Furthermore, one can verify that guessed solution of $\pi(t,x)$ satisfy the PDE if the parameters $m(t), K(t)$ satisfy the following ordinary differential equations:

$$\begin{aligned} \dot{m} =& Am + Bu \\ \dot{K} =& AK + KA' + C \\ m(t_i) =& x(t_i),\ K(t_i) = 0 \end{aligned}$$

According to Theorem 4.4.2, the optimal u^* can be constructed from a value function $V(t,x)$ satisfying (4.9). Due to Proposition 4.4.6, $V(t,x) = W(t,x) + \int_t^T \psi(s)ds$. Where solving (4.13), one can show $W(t,x) = x'P(t)x + \eta(t)$ where

$$\begin{aligned} &\dot{P} + PA + A'P - PBR^{-1}B'P + L = 0 \\ &\dot{\eta} + tr(aP) = 0 \\ &P(T) = Q, \eta(T) = 0. \end{aligned}$$

The optimal feedback control that minimizes the RHS of (4.13) is $u(t) = k(x) =$

$-R^{-1}B'Px$. Thus,

$$\begin{aligned}\phi(t,x,u) =& x'Lx + u'Ru + (Ax+Bu)'Px + x'P(Ax+Bu)\\ &+ tr(aP)\end{aligned}$$

Thus, from (4.14),

$$\begin{aligned}\psi(t) =& \min_u \left\langle \phi(t,\cdot,u), \pi(t,\cdot)\right\rangle - \left\langle \phi(t,\cdot,k(\cdot)), \pi(t,\cdot)\right\rangle\\ =& \min_u \left\langle x'Lx + u'Ru + (Ax+Bu)'Px \right.\\ &\left. + x'P(Ax+Bu) + tr(aP), \pi(t,x)\right\rangle\\ &- \left\langle x'Lx + k(x)'Rk(x) + (Ax+Bk(x))'Px\right.\\ &\left. + x'P(Ax+Bk(x)) + tr(aP), \pi(t,x)\right\rangle\\ =& \min_u \left\langle u'Ru + u'B'Px + x'PBu, \pi(t,x)\right\rangle\\ &+ \left\langle x'PBR^{-1}B'Px, \pi(t,x)\right\rangle\\ =& tr(PBR^{-1}B'P\mathbb{E}[xx']) + \min_u\{u'Ru + u'B'Pm + m'PBu\}\\ =& tr(PBR^{-1}B'P\mathbb{E}[(x-m)(x-m)'])\\ =& tr(PBR^{-1}B'PK)\end{aligned}$$

For the event-triggered problem, the optimal u is found be $u(t) = -R^{-1}B'P(t)m(t)$ which minimizes $u'Ru + u'B'Pm + m'PBu$.

Thus Problem 4.4.6 simplifies into:

Problem 4.5.1.

$$\min_{N,t_1,t_2,\cdots,t_N} \int_0^T tr(P(t)BR^{-1}B'P(t)K(t))dt + \lambda N$$

$$s.t. \quad \dot{K} = AK + KA' + C$$

$$K(t_i) = \boldsymbol{0}.$$

Furthermore, one can combine the two constraints in the above optimization problem and express $K(t)$ as:

$$K(t) = \int_{t_i}^{t} e^{A(t-s)} C e^{A'(t-s)} ds \qquad \forall t \in [t_i, t_{i+1})$$

Thus, Problem 4.5.1 can be represented as:

$$\min_{N,t_1,\cdots,t_N} \sum_{k=0}^{N} \int_{t_k}^{t_{k+1}} \int_{t_k}^{t} tr(\tilde{R}(t) e^{A(t-s)} C e^{A'(t-s)}) ds + \lambda N$$

$$t_0 = 0, t_{N+1} = T,$$

where $\tilde{R}(t) = P(t)BR^{-1}B'P(t)$.

4.5.2 Linear Quadratic Control with State Dependent Noise

The dynamics of the states are given as:

$$dx = Axdt + Budt + \sum_{k=1}^{p} \sigma_k x dw_k(t) \tag{4.15}$$

$$x(0) \sim p_0(\cdot)$$

where each $dw_k(t)$ is a one dimensional Wiener process noise independent of $dw_l(t)$, and σ_k is the corresponding noise matrix, and $p_0(\cdot)$ is the initial state distribution. To maintain brevity, we will proceed with the case when $p = 1$, and accordingly $\sigma_1 = \sigma$. The cost function associated with this problem is same as in Section 4.5.1, i.e.,

$$J = \mathbb{E}\left[\int_0^T (x'Lx + u'Ru)dt + x(T)'Qx(T) + \lambda|\mathcal{E}(t)|\right].$$

Firstly, let us note that the distribution of the state $\pi(t,x)$ will not be Gaussian even when $p_0(x)$ is Gaussian. Moreover, it might be the case, that $\pi(t,x)$ might not even be represented in terms of some finite dimensional parameters.

Let us first characterize $W(t,x)$, as mentioned in Lemma 4.4.4, for this problem

$$-\frac{\partial W}{\partial t} = \min_u \Big(x'Lx + u'Ru + +\nabla W(t,x) f(t,x,u) \\ + tr\Big(a(t,x,u)\frac{\partial^2 W}{\partial x^2}(t,x)\Big)\Big)$$

Let us guess $W(t,x) = x'Px$, then

$$-\frac{\partial W}{\partial t} = \min_u \Big(x'Lx + u'Ru + x'P(Ax + Bu) + (Ax + Bu)'Px + tr\big(\sigma x x' \sigma' P\big)\Big)$$

$$-x'\dot{P}x = \min_u \Big(x'(L + A'P + PA + \sigma' P \sigma)x + u'Ru + x'PBu + u'B'Px\Big)$$

The optimal u^* that minimizes the above equation is given as $u^*(t) = k(x) = -R^{-1}B'Px$ which is exactly same as the linear quadratic case presented in Section 4.5.1. Thus, in order for our guess $x'P(t)x$ to be a valid value function, we require:

$$\dot{P} + L + A'P + PA + \sigma' P \sigma - PBR^{-1}B'P = 0$$

$$P(T) = Q$$

For this problem, $\phi(t,x,u)$ is given as:

$$\phi(t,x,u) = x'(L + A'P + PA + \sigma' P \sigma)x + u'Ru + u'B'Px + x'PBu$$

Thus one can show,

$$\begin{aligned}\psi(t) =& \big\langle x'PBR^{-1}B'Px, \pi(t,x)\big\rangle + \min_u \big\langle u'Ru + u'B'Px + x'PBu, \pi(t,x)\big\rangle \\ =& tr(PBR^{-1}B'P\mathbb{E}[xx']) + \min_u\{u'Ru + u'B'Pm + m'PBu\} \\ =& tr(PBR^{-1}B'P\mathbb{E}[(x-m)(x-m)']) \\ =& tr(PBR^{-1}B'PK)\end{aligned}$$

where $m(t) = \mathbb{E}[x(t)] = \langle x, \pi(t,x)\rangle$ and $K(t) = \mathbb{E}[(x-m)(x-m)'] = \langle xx', \pi(t,x)\rangle - mm'$.

Taking expectation on both sides of (4.15),

$$\begin{aligned}&\dot{m} = Am + Bu = (A - BR^{-1}B'P)m \\ &m(0) = \langle x, p_0(x)\rangle \\ &m(t_i) = x(t_i).\end{aligned}$$

Let us define $K_1(t) = x(t)x(t)'$ and thus, by Itô rule,

$$\begin{aligned}dK_1 =& (Axdt + Budt + \sigma xdw)x' + x(Axdt + Budt + \sigma xdw)' + \sigma xx'\sigma' dt \\ \mathbb{E}[\dot{K}_1] =& A\mathbb{E}[K_1] + \mathbb{E}[K_1]A' + \sigma\mathbb{E}[K_1]\sigma' + Bum' + mu'B'\end{aligned}$$

Since $\dot{m} = Am + Bu$, $K_2(t) \triangleq mm'$ satisfies $\dot{K}_2 = m(Am + Bu)' + (Am + Bu)m'$.

Thus, $K(t) = \mathbb{E}[K_1(t)] - K_2(t)$ satisfies,

$$\dot{K} = AK + KA' + \sigma K \sigma' + \sigma m m' \sigma'.$$

$$K(0) = \langle xx', p_0(x) \rangle - m(0)m(0)'$$

$$K(t_i) = \mathbf{0}.$$

The above matrix equation can be represented as a vector differential equation by defining $l(t) = vec(K(t))$.

$$\dot{l} = \tilde{A} l + vec(\sigma m m' \sigma')$$

$$l(0) = vec(K(0))$$

$$l(t_i) = 0$$

where $\tilde{A} = I \otimes A + A \otimes I + \sigma \otimes \sigma$. Since, $tr(\cdot)$ is an linear operator, we can represent $\bar{r}(t)' l(t) \triangleq tr(PBR^{-1}B'PK)$ for some vector $\bar{r}(t)$. Thus, the final optimization problem becomes:

$$\min_{N, t_1, \cdots, t_N} \sum_{k=0}^{N} \int_{t_k}^{t_{k+1}} \int_{t_k}^{t} \bar{r}(t) e^{\tilde{A}(t-s)} vec(\sigma m m' \sigma') ds dt + \lambda N$$

$$t_0 = 0, t_{N+1} = T,$$

which is very similar to the optimization problem we received for the standard LQG case.

4.6 Conclusion

In this chapter we considered the general nonlinear stochastic intermittent feedback control. This is not 'event-trigged' control per se, as there is no event to be triggered rather the triggering instances are found offline by optimizing Problem 4.4.6. We showed how the solution of this problem has a strong connection with the solution of the same problem with continuous feedback control.

For the special structures of the costs and dynamics we show the synthesis of the optimal controller and event triggering pairs.

Chapter 5: Linear Quadratic Games with Costly Measurements

While the past chapters are focus on a single controller or (equivalently) centralized structure of the controller, in this chapter we discuss a game theoretic formulation of intermittent feedback problem. In this chapter we consider a stochastic linear quadratic two-player[1] game. The state measurements are observed through a switched noiseless communication link. Each player incurs a finite cost every time the link is established to get measurements. Along with the usual control action, each player is equipped with a switching action to control the communication link. The measurements help to improve the estimate and hence reduce the quadratic cost but at the same time the cost is increased due to switching. We study the subgame perfect equilibrium control and switching strategies for the players. We show that the problem can be solved in a two-step process by solving two dynamic programming problems. The first step corresponds to solving a dynamic programming for the control strategy and the second step solves another dynamic programming for the switching strategy.

[1]The idea could be extended for a general N player game.

5.1 Introduction & Literature Review

Linear quadratic (LQ) stochastic games have attracted a great deal of attention in the control and related community due to its wide applicability in stochastic control, minimax control, multi-agent systems and economics [13], [34], [38], [98], [11], [24], [47]. There is a well established notion of (Nash) equilibrium (NE) strategies for static games, and in dynamic games there are refinements of NE known as subgame perfect equilibria (SPE). Closed form solutions for these NE (or SPE) may generally not exist or hard to compute if one such exists. Among the various classes of dynamic games, LQ games exhibit a closed form expression for SPE, and it is characterized by some Riccati equations. Necessary and sufficient conditions for the NE strategies of LQ games have been studied in [38], [17], [11]. Contrary to the prior belief, [12] shows existence of nonlinear control strategies for LQ games.

Amongst the vast majority of the prior works, the underlying assumption is the availability of free observations. Dynamic games are studied with either open-loop strategy (i.e. only measurement is the initial state) or feedback strategies where the observation is available freely at any time. Challenges emerge when the measurements are on demand, but costly. This adds an extra layer of decision making, for the players, because now they have to both, control the system and ask for measurements.

In this chapter, we consider a class of two-player linear quadratic stochastic games of finite horizon. The game dynamics are partially observable. Contrary to the existing literature, the observations are not freely available. Each observation

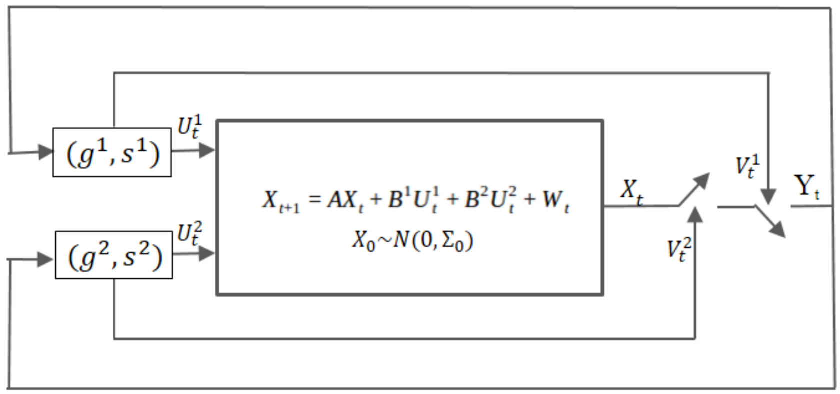


Figure 5.1: Schematic of the system. Each player has to select their controller strategy g^i and switching strategy s^i. All the links are noiseless and delay-free.

requires a finite cost for establishing a link for communication. The link through which the observations are communicated to the players (their controllers) is noiseless but operated by two switches (Figure 5.1), one for each player. The link is established only when both the players are willing for it, and they both get the actual state measurement at that time. Consequently, there is an apparent trade off between cost of obtaining state measurement and the estimation quality.

In this game, the players can make a precise estimate of the state if they establish the link at every time instance. However, since the link establishment is costly, they can compromise the estimation accuracy in exchange of the cost for accessing the measurement. Therefore, the problem is to optimally decide when to establish the link and how to use the acquired measurement in order to minimize their individual cost. Since, in general, the players will have different preferences over the time instances when they want to acquire the measurement, they have to come to an agreement when to actually establish the link.

The closest work on the similar game framework has been studied in [68] where the authors studied zero-sum stochastic differential LQ games. However, the selection for switching times were performed in an collaborative way rather than being the outcome of a strategic interaction. The major digression of the work presented in this chapter from [68] is that we consider an explicit game for the switching strategy as well. We express the switch as a Boolean control action and seek for SPE for both control and switching strategies.

The primary contributions of this chapter are as follows:

(a) We study the SPE of this dynamic game and show that they can be found through a two-step process. Specifically, in the first step we fix the switching strategy and study the SPE for control strategies. The study shows that the control strategy is linear in estimated state, where the gain is characterized with two backward Riccati equations which can be computed offline. Moreover, the Riccati equations do not depend on the switching strategy.

(b) Regarding the equilibrium switching strategy, we provide a backward recursive algorithm to find all SPE where value functions need only be computed over a finite and quadratically-sized (in the duration of the game) set.

(c) Regarding the equilibrium switching strategy, we show that there are many equilibria among which there is one that is strictly preferable by both users and has a Markovian structure. It is found in our study that a strictly preferable switching strategy for a player not only depends on their own cost-to-go, but also depends on the cost-to-go for the opponent.

The remaining of the chapter is organized as follows: The problem formulation is provided in Section 5.2, Section 5.3 contains the results on the SPE of the control strategy, SPE for the switching strategy and its offline computation are analyzed in Section 5.4. Finally we conclude this chapter in Section 5.6.

5.2 Problem Formulation

In the discrete time Gauss-Markov setting, we consider the following linear dynamics of the state X_t:

$$X_{t+1} = AX_t + B^1 U_t^1 + B^2 U_t^2 + W_t \tag{5.1}$$

where $X_t \in \mathcal{X} = \mathbb{R}^n$, and $U_t^i \in \mathcal{U}^i = \mathbb{R}^m$ denotes the action of player i. $W_t \in \mathbb{R}^n$ is a Gaussian noise with $\mathbb{E}[W_t] = 0$ and $\mathbb{E}[W_t W_s^{'}] = S\delta_{t-s}$ (δ_t is the Kronecker delta.), and $X_0 \sim \mathcal{N}(0, \Sigma_0)$.

There are two additional actions (switching actions) $V_t^1 \in \{0, 1\}$ and $V_t^2 \in \{0, 1\}$. These switching actions control a switch (switch closes if both are equal to 1) and the observation available to both users is $Y_t \in \mathcal{X} \cup \{e\}$ with

$$Y_t = \begin{cases} X_t & , V_t^1 = V_t^2 = 1 \\ e & , \text{else}, \end{cases} \tag{5.2}$$

where "e" denotes an erasure. The evolution of random variables in period t is assumed to be ...$X_t \to (V_t^1, V_t^2) \to Y_t \to (U_t^1, U_t^2)$...

The information available at time t to player i before she takes the switching action V_t^i is

$$I_t = (Y^{t-1}, U^{1,t-1}, U^{2,t-1}, V^{1,t-1}, V^{2,t-1}), \tag{5.3}$$

and the information available at time t to player i before she takes the control action U_t^i is

$$\bar{I}_t = (I_t, V_t^1, V_t^2, Y_t). \tag{5.4}$$

As a result, the actions have the functional form

$$V_t^i = s_t^i(I_t), \qquad i = 1, 2, \tag{5.5a}$$

$$U_t^i = g_t^i(\bar{I}_t), \qquad i = 1, 2, \tag{5.5b}$$

where by $g^i = (g_t^i)_{t=0}^{T-1}$, $s^i = (s_t^i)_{t=0}^{T-1}$, we denote the control and switching strategies of player i. $\forall t \in \{0, \cdots, T-1\}$, let us denote an I_t measurable random variable $\Delta_t = V_t^1 \cdot V_t^2$.

The individual cost that each player needs to minimize is quadratic in state and action, and it also depends on the switching actions V_t^1 and V_t^2. We consider a game for a finite duration ($\{0, \cdots, T\}$) and the per-stage costs are explicitly written as:

$$C_t^i(x_t, u_t^1, u_t^2, v_t^1, v_t^2) = \|x_t\|_{Q^i}^2 + \|u_t^i\|_{Q^{ii}}^2 + \|u_t^j\|_{Q^{ij}}^2 + \lambda_i (v_t^i \cdot v_t^j) \tag{5.6}$$

for all $t \in \{0, \cdots, T-1\}$ and

$$C_T^i(x_t, u_t^1, u_t^2, v_t^1, v_t^2) = \|x_T\|_{Q^i}^2. \tag{5.7}$$

The quantity $\lambda_i > 0$ is the cost paid by player i when both the players attempt to close the switch and they observe the state information X_t. Therefore the average cost over the time horizon $\{0, \cdots, T\}$ is represented as,

$$J^i(\sigma^1, \sigma^2) = \sum_{t=0}^{T} \mathbb{E}[C_t^i(X_t, U_t^1, U_t^2, V_t^1, V_t^2)] \tag{5.8}$$

where $\sigma^i = (g^i, s^i)$ denotes the strategy of the player i that corresponds to control strategy g^i and switching strategy s^i.

The objective of player i is:

$$\min_{\sigma^i} J^i(\sigma^1, \sigma^2) = \min_{s^i}\{\min_{g^i} J^i(\sigma^1, \sigma^2)\} \tag{5.9}$$

5.3 Subgame Perfect Control Strategy

For dynamic games with complete information the appropriate equilibrium concept is a refinement of Nash equilibrium (NE) called the subgame perfect equilibrium (SPE). A strategy profile (σ^1, σ^2) is a SPE if the restriction of (σ^1, σ^2) to any proper subgame of the original game constitutes a NE [39, pp. 94].

We seek to characterize the SPE $(\sigma^{1*}, \sigma^{2*})$ for this switched LQG game. Moreover, we will show that among the multiple SPE, there exists one that simultaneously minimizes the cost for both users among all SPE and thus it will be the preferable SPE solution of this game. In this section, we study the SPE control strategy for both the players.

Theorem 5.3.1. *For any switching profile (s^1, s^2) of the players, the SPE control strategy g^{i*} has the following structure:*

$$U_t^i = g_t^{i*}(\bar{I}_t) = -L_t^i \hat{X}_t, \tag{5.10}$$

where

$$\hat{X}_t = \begin{cases} A\hat{X}_{t-1} + B^1 U_{t-1}^1 + B^2 U_{t-1}^2, & V_t^1 \cdot V_t^2 = 0 \\ X_t, & V_t^1 \cdot V_t^2 = 1. \end{cases} \tag{5.11}$$

Furthermore, the cost-to-go incurred by player i under the SPE control strategy at any time step k is given by,

$$\mathbb{J}_k^{i*}(\bar{I}_k) = \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|E_t\|_{Q^i}^2 + \lambda_i \Delta_t) + \sum_{t=k}^{T-2} \Delta_{t+1} \|AE_t + W_t\|_{P_{t+1}^i}^2 + \|E_T\|_{Q^i}^2 \mid \bar{I}_k\Big] + \|\hat{X}_k\|_{P_k^i}^2 \tag{5.12}$$

where $E_t = X_t - \hat{X}_t$. The matrices L_t^i and P_t^i depend only on the game parameters A, B^i, Q^i, Q^{ii} and Q^{ij} (detailed expressions are in the proof of the theorem) and thus, can be calculated offline without the knowledge of the switching strategy profile.

proof The proof of this theorem is provided in Appendix 5.7.1.

To maintain brevity $\mathbb{J}_k^{i*}(\bar{I}_k)$ will be denoted as $\mathbb{J}_k^{i*}$. From this point onward we will set $\Delta_T = 0$ and write (5.12) in compact form as

$$\mathbb{J}_k^{i*}(\bar{I}_k) = \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|E_t\|_{Q^i}^2 + \Delta_{t+1}\|AE_t + W_t\|_{P_{t+1}^i}^2 + \lambda_i\Delta_t) + \|E_T\|_{Q^i}^2|\ \bar{I}_k\Big] + \|\hat{X}_k\|_{P_k^i}^2 \tag{5.13}$$

It should be noted that in Theorem 5.3.1, the g^{i*} depends on the given switching strategy (s^1, s^2) through the $\hat{X}_t$.

There are several remarks to be made at this point.

The stochastic control version of the same problem (i.e. single player single objective) is a modified Kalman filtering problem where the observations are available on demand after paying certain cost λ per observation. Therefore, the decision of switching will solely depend on the influence of switching on the error covariance matrix. This is a side result of our work and details will appear elsewhere.

From Theorem 5.3.1, $\min_{\sigma^i} J^i(\sigma^1, \sigma^2) = \min_{s^i} \mathbb{E}[\mathbb{J}_0^{i*}]$. Therefore, the total cost incurred by player i with control strategy profile (g^{1*}, g^{2*}) is $\mathbb{E}[\mathbb{J}_0^{i*}]$. Hence, the total cost incurred with the switching is:

$$\mathbb{E}[\mathbb{J}_0^{i*}] = \mathbb{E}[\|\hat{X}_0\|_{P_0^i}^2] + \mathbb{E}\Big[\sum_{t=0}^{T-1}\big(\|E_t\|_{Q^i}^2 + \Delta_{t+1}\|AE_t + W_t\|_{P_{t+1}^i}^2 + \lambda_i\Delta_t\big) + \|E_T\|_{Q^1}^2\Big]. \tag{5.14}$$

Another remark that is apparent from our result is that the SPE control strategy is completely characterized by the pair of matrices (P_t^1, P_t^2) which is uniquely determined by backward dynamic equations.

5.4 Subgame Perfect Switching Strategy

In this section we complete the procedure for finding the SPE of this game by focusing on the switching strategies. We will do that by considering the backward induction process for finding SPE and reduce the cost-to-go functions into a simpler and more tractable form (compared to the one in (5.13)).

In this problem the switching action is taken first at time k based on the knowledge I_k and then the augmented knowledge $\bar{I}_k = (I_k,\, V_k^1,\, V_k^2,\, Y_k)$ is used to select the control strategies U_k^i. In order to visualize it, one might break the time period $[k, k+1]$ into two halves where in the first half, switching action is performed and in the second half, control action is performed. In Theorem 5.3.1, $\mathbb{J}_k^{i*}$ is the optimal cost-to-go after the switching decision has been taken at time k.

The actual (before switching action is taken) cost-to-go at stage k is:

$$\mathbb{V}_k^i(I_k) = \mathbb{E}\Big[\sum_{t=k}^{T} C_t^i(X_t, U_t^1, U_t^2, V_t^1, V_t^2)|\ I_k\Big] \tag{5.15}$$

and the optimization (game) variables are control U_t^i and switching V_t^i for all $t \geq k$.

Due to the fact $\bar{I}_t \supseteq I_t$ for all t, we can write

$$\begin{aligned} \mathbb{V}_k^i(I_k) =& \mathbb{E}\Big[\mathbb{E}\Big[\sum_{t=k}^{T} C_t^i(X_t, U_t^1, U_t^2, V_t^1, V_t^2)|\ \bar{I}_k\Big]\ I_k\Big] \\ =& \mathbb{E}\big[\mathbb{J}_k^i(g^1, g^2)|\ I_k\big] \end{aligned} \tag{5.16}$$

where $\mathbb{J}_k^i(g^1, g^2) = \mathbb{E}\Big[\sum_{t=k}^{T} C_t^i(X_t, U_t^1, U_t^2, V_t^1, V_t^2)|\ \bar{I}_k\Big]$.

Since each player is interested in minimizing their cost, they are interested in $\min_{s^i,g^i} \mathbb{V}_k^i(I_k)$ at every stage k (finally they want to minimize $\mathbb{V}_0^i(I_0)$).

We can write,

$$\min_{s^i,g^i} \mathbb{V}_k^i(I_k) = \min_{s^i}\{\min_{g^i} \mathbb{V}_k^i(I_k)\} = \min_{s^i} \mathbb{E}[\mathbb{J}_k^{i*}|\ I_k]. \tag{5.17}$$

We substitute the expression of $\mathbb{J}_k^{i*}$ from Theorem 5.3.1 into (5.17), but before that, let us define,

$$M_t = \mathbb{E}[E_t E_t'|\ \bar{I}_t] = (1 - \Delta_t)(AM_{t-1}A' + S) \tag{5.18}$$

where $AM_{-1}A' + S = \Sigma_0$ (since $X_0 \sim \mathcal{N}(0, \Sigma_0)$).

We also define $M_{t|t-1} = AM_{t-1}A' + S$. Note that $M_{t|t-1}$ is I_t measurable whereas M_t is $\bar{I}_t$ measurable. M_t and $M_{t|t-1}$ are related as follows:

$$M_t = (1 - \Delta_t)M_{t|t-1} \tag{5.19}$$

Now let us consider the k-th stage cost $\mathbb{J}_k^{i*}$.

$$\begin{aligned}\mathbb{J}_k^{i*} =&\mathbb{E}\Big[\sum_{t=k}^{T-1}\big(\|E_t\|_{Q^i}^2 + \Delta_{t+1}\|AE_t + W_t\|_{P_{t+1}^i}^2 + \lambda_i\Delta_t\big) + \|E_T\|_{Q^i}^2 |\ \bar{I}_k\Big] + \|\hat{X}_k\|_{P_k^i}^2 \\ =&\mathbb{E}\Big[\sum_{t=k}^{T-1}\big(tr(Q^iM_t + \Delta_{t+1}(AM_tA' + S)P_{t+1}^i) + \lambda_i\Delta_t\big) + tr(Q^iM_T) |\ \bar{I}_k\Big] + \|\hat{X}_k\|_{P_k^i}^2 \\ =&\mathbb{E}\Big[\sum_{t=k}^{T-1}\big(tr((1-\Delta_t)Q^iM_{t|t-1} + \Delta_{t+1}M_{t+1|t}P_{t+1}^i) + \lambda_i\Delta_t\big) + tr(Q^iM_T) |\ \bar{I}_k\Big] \\ &+ \|\hat{X}_k\|_{P_k^i}^2 \end{aligned} \tag{5.20}$$

Let us define $\mathcal{V}_k^i(I_k) = \mathbb{E}\Big[\mathbb{J}_k^{i*} |\ I_k\Big]$ Therefore,

$$\begin{aligned}\mathcal{V}_k^i(I_k) =&\mathbb{E}\Big[\sum_{t=k}^{T-1}\big(tr((1-\Delta_t)Q^iM_{t|t-1}) + tr(\Delta_{t+1}M_{t+1|t}P_{t+1}^i) + \lambda_i\Delta_t\big) + tr(Q^iM_T) |\ I_k\Big] \\ &+ \mathbb{E}\Big[\|\hat{X}_k\|_{P_k^i}^2 |\ I_k\Big]\end{aligned} \tag{5.21}$$

Using (5.47), we get

$$\begin{aligned}\mathcal{V}_k^i(I_k) = \mathbb{E}\Big[\sum_{t=k}^{T-1}\big(tr((1-\Delta_t)Q^iM_{t|t-1}) + tr(\Delta_tM_{t|t-1}P_t^i) + \lambda_i\Delta_t\big) + tr(Q^iM_T) |\ I_k\Big] \\ + \|\hat{X}_{k|k-1}\|_{P_k^i}^2\end{aligned} \tag{5.22}$$

The selection of switching strategy $s_k^i(I_k)$ has no effect of $\hat{X}_{k|k-1}$ and hence it does not play any role in the game at stage k.

Let us define an instantaneous cost:

$$\bar{C}^i_t(M_{t|t-1}, \Delta_t) = (1-\Delta_t)tr(Q^i M_{t|t-1}) + \Delta_t tr(M_{t|t-1} P^i_t) + \lambda_i \Delta_t. \tag{5.23}$$

With slight abuse of notation, after neglecting the $\hat{X}_{k|k-1}$ term, we obtain,

$$\mathcal{V}^i_k(I_k) = \mathbb{E}\Big[\sum_{t=k}^{T} \bar{C}^i_t(M_{t|t-1}, \Delta_t)|\ I_k\Big]. \tag{5.24}$$

Therefore,

$$\begin{aligned}\min_{s^i, g^i} \mathbb{V}^i_k(I_k) &= \min_{s^i} \mathcal{V}^i_k(I_k) \\ &= \min_{s^i} \mathbb{E}\Big[\sum_{t=k}^{T} \bar{C}^i_t(M_{t|t-1}, \Delta_t)|\ I_k\Big]. \end{aligned} \tag{5.25}$$

Let us denote:

$$\mathcal{V}^{i*}_k = \min_{s^i} \mathcal{V}^i_k(I_k). \tag{5.26}$$

Let us perform the similar backward induction to find the SPE for the switching strategies. Note at time T, there is no action to optimize and

$$\mathcal{V}^i_T(I_T) = \mathbb{E}\Big[\bar{C}^i_T(M_{T|T-1}, \Delta_T)|\ I_T\Big] = \mathbb{E}\Big[tr(Q^i M_T)\ |\ I_T\Big]. \tag{5.27}$$

Let us define

$$\mathcal{V}^{i*}_T = \mathcal{V}^i_T(I_T) = \mathbb{E}\Big[tr(Q^i M_T)\ |\ I_T\Big]. \tag{5.28}$$

Similarly, at $T-1$,

$$\mathcal{V}^i_{T-1}(I_{T-1}) = \mathbb{E}\Big[\bar{C}^i_{T-1}(M_{T-1|T-2}, \Delta_{T-1}) + \bar{C}^i_T(M_T, 0)|\ I_{T-1}\Big]. \tag{5.29}$$

$$\mathcal{V}^i_{T-1}(I_{T-1}) = \mathbb{E}\Big[\bar{C}^i_{T-1}(M_{T-1|T-2}, \Delta_{T-1}) + tr(Q^i M_T)|\ I_{T-1}\Big]. \tag{5.30}$$

Using $M_{T-1} = (1-\Delta_{T-1})M_{T-1|T-2}$ and $M_T = AM_{T-1}A' + S$,

$$\begin{aligned}\mathcal{V}^i_{T-1}(I_{T-1}) =& \mathbb{E}\Big[\bar{C}^i_{T-1}(M_{T-1|T-2}, \Delta_{T-1}) + \\ & (1-\Delta_{T-1})tr(Q^i(AM_{T-1|T-2}A')) + tr(Q^i S)|\ I_{T-1}\Big]\end{aligned} \tag{5.31}$$

If $(s^{1*}_{T-1}, s^{2*}_{T-1})$ is a SPE strategy at time $T-1$ then

$$\mathcal{V}^i_{T-1}(I_{T-1})\big|_{(s^{i*}_{T-1}, s^{j*}_{T-1})} \leq \mathcal{V}^i_{T-1}(I_{T-1})\big|_{(s^{i}_{T-1}, s^{j*}_{T-1})} \tag{5.32}$$

$\forall s^i_{T-1}$ and for both $i = 1, 2$; $j = 1, 2$ and $i \neq j$.

Using the above definition of SPE, $s^i_{T-1}(I_{T-1}) = 0$ for $i = 1, 2$ is an equilibrium strategy since unilateral change from 0 to 1 does not change the cost for any player. However, there might be other equilibria (in this case only $(1, 1)$) which produces lower cost for the above cost function.

It is straightforward to show that the equilibrium strategy at $T-1$ is

$$s^{i*}_{T-1}(I_{T-1}) = \begin{cases} 1 & \text{if} \quad \bar{C}^i_{T-1}(M_{T-1|T-2}, 1) - \bar{C}^i_{T-1}(M_{T-1|T-2}, 0) \leq tr(Q^i(AM_{T-1|T-2}A')) \\ 0 & \text{otherwise} \end{cases} \tag{5.33}$$

From (5.33) we notice that $(1, 0)$ and $(0, 1)$ can also be an equilibrium strategy. However those equilibria are equivalent to $(0, 0)$ in the sense that they produce the same cost-to-go $\mathcal{V}^{i*}_{T-1}$ for both $i = 1, 2$. Therefore, we will restrict our attention on two equilibria $(0, 0)$ and $(1, 1)$

As a remark, it is pointed out that adding an infinitesimal switching cost ϵ_i for every time player i requests for a switching (irrespective of whether the switch was closed or not) will ensure that $(0, 1)$ and $(1, 0)$ is never an SPE.

Let us note when $\bar{C}^i_{T-1}(M_{T-1|T-2}, 1) - \bar{C}^i_{T-1}(M_{T-1|T-2}, 0) = tr(Q^i(AM_{T-1|T-2}A'))$, then $s^{i*}_{T-1} = 0$ or 1, both produces the same cost-to-go value. Under such situations, all possible switching actions are equivalent. In order to obliterate such instances we make the following assumption:

Assumption 5.4.1. *If* $\bar{C}^i_{T-1}(M_{T-1|T-2}, 1) - \bar{C}^i_{T-1}(M_{T-1|T-2}, 0) = tr(Q^i(AM_{T-1|T-2}A'))$, $s^{i*}(I_{T-1}) = 0$ *for all possible history* I_{T-1}. *Then, (5.33) is modified as follows:*

$$s^{i*}_{T-1}(I_{T-1}) = \begin{cases} 1 & \textit{if} \quad \bar{C}^i_{T-1}(M_{T-1|T-2}, 1) - \bar{C}^i_{T-1}(M_{T-1|T-2}, 0) < tr(Q^i(AM_{T-1|T-2}A')) \\ 0 & \textit{otherwise} \end{cases} \tag{5.34}$$

Irrespective of whether SPE $s^i_{T-1}(I_{T-1})$ is 0 or 1, the optimal cost-to-go $\mathcal{V}^{i*}_{T-1}$

depends only on M_{T-2} and also the best SPE strategy (that produces the least cost among all SPE) $s^{i*}_{T-1}(I_{T-1})$ depends only on M_{T-2} (or $M_{T-1|T-2}$).

Therefore, we hypothesize the following:

Claim 5.4.2. *For any k, there exists a $s^{i*}_k(I_k)$ that depends only on M_{k-1} and produces the least cost-to-go among all SPE. Hence $\mathcal{V}^{i*}_k \equiv \mathcal{V}^{i*}_k(M_{k-1})$ (i.e. $\mathcal{V}^{i*}_k$ only depends on M_{k-1}).*

Proof: The hypothesis is true for $k = T, T-1$. Let us assume it is true for some $k+1 \leq T$, i.e. $\mathcal{V}^{i*}_{k+1} \equiv \mathcal{V}^{i*}_{k+1}(M_k)$. Therefore,

$$\mathcal{V}^{i*}_k = \min_{s^i} \sum_{t=k}^{T} \mathbb{E}\Big[\bar{C}^i_t(M_{t|t-1}, \Delta_t)|\ I_k\Big]$$

Then using a dynamic programming argument,

$$\begin{aligned}
\mathcal{V}^{i*}_k &= \min_{s^i_k} \mathbb{E}\Big[\bar{C}^i_k(M_{k|k-1}, \Delta_k) + \mathcal{V}^{i*}_{k+1}(M_k)|\ I_k\Big] \\
&= \min_{s^i_k} \mathbb{E}\Big[\bar{C}^i_k(M_{k|k-1}, \Delta_k) + \mathcal{V}^{i*}_{k+1}\big((1-\Delta_k)M_{k|k-1}\big)|\ I_k\Big]
\end{aligned} \tag{5.35}$$

From (5.35), the best equilibrium strategy $s^{i*}_k(I_k) = 1$ if

$$\bar{C}^i_k(M_{k|k-1}, 1) + \mathcal{V}^{i*}_{k+1}(\mathbf{0}) < \bar{C}^i_k(M_{k|k-1}, 0) + \mathcal{V}^{i*}_{k+1}(M_{k|k-1})$$

(similar to assumption 5.4.1, we only consider the strict inequality), otherwise $s^i_k(I_k) = 0$.

Therefore $s_k^{i*}(I_k)$ requires only the knowledge of M_{k-1} and hence from (5.35), $\mathcal{V}_k^i(I_k) \equiv \mathcal{V}_k^{i*}(M_{k-1})$

For this class of games, there always exists a Markovian SPE switching strategy and a Markovian SPE control strategy which produce the least cost-to-go among all SPE. Though, there might be other non-Markovian SPE strategies which produce the same cost, however, due to the claim 5.4.2, it is sufficient to consider only the Markovian strategies to find the best SPE corresponding to the least cost-to-go.

5.4.1 Offline Calculation of $\mathcal{V}_k^{i*}(M_{k-1})$

In the following we define how the players can take the decision online by using some stored offline functions (value functions).

Let us define $\mathcal{V}_k^{i*}(M)$ in the following manner:

$$\mathcal{V}_T^{i*}(M) = \bar{C}_T^i(M, 0). \qquad \forall M \text{ and } i = 1, 2 \tag{5.36}$$

and

$$\mathcal{V}_k^{i*}(M) = \begin{cases} \bar{C}_k^i(M, 1) + \mathcal{V}_{k+1}^{i*}(\mathbf{0}) & \text{if } \vartheta(k, M) > 1, \\ \bar{C}_k^i(M, 0) + \mathcal{V}_{k+1}^{i*}(AMA' + S) & \text{otherwise.} \end{cases} \tag{5.37}$$

where $\vartheta(k, M) = \min\{\vartheta^1(k, M), \vartheta^2(k, M)\}$, and

$$\vartheta^i(k, M) = \frac{\bar{C}_k^i(M, 0) + \mathcal{V}_{k+1}^{i*}(AMA' + S)}{\bar{C}_k^i(M, 1) + \mathcal{V}_{k+1}^{i*}(\mathbf{0})} \tag{5.38}$$

By construction, if $\mathcal{V}^{i*}_{k+1}(\cdot)$ denotes the minimum cost-to-go (for the subgame starting at $k+1$) among the SPE, $\mathcal{V}^{i*}_{k}(\cdot)$ defined in (5.37) provides the minimum cost-to-go at stage k for player i. Therefore, by backward inductions, $\mathcal{V}^{i*}_{\cdot}(\cdot)$ denotes the cost-to-go function along an SPE that simultaneously minimizes the cost-to-go for both players.

Claim 5.4.3. *For any k, M and history I_k, the best switching strategy (SPE) is given by $s^{i*}_k(I_k) = 1$ for $i = 1, 2$ if and only if,*

$$\bar{C}^i_k(M, 1) + \mathcal{V}^{i*}_{k+1}(\boldsymbol{0}) < \bar{C}^i_k(M, 0) + \mathcal{V}^{i*}_{k+1}(AMA' + S). \tag{5.39}$$

Otherwise $s^{i*}_k(I_k) = 0$ for $i = 1, 2$.

proof $\Rightarrow$ is trivially true.

$\Leftarrow$: First, notice that we have established $s^i_k(I_k) = 0$ is an SPE strategy for all k, I_k. Now let us assume that at some k, M, (5.39) holds, then if player i selects a strategy such that $s^i_k(I_k) = 0$, then the cost-to-go for player i with any strategy profile $((s^1)^T_{k+1}, (s^2)^T_{k+1})$ from time $k+1$ onward is

$$\begin{aligned}
&\bar{C}^i_k(M, 0) + \mathcal{V}^i_{k+1}(AMA' + S, (s^1)^T_{k+1}, (s^2)^T_{k+1}) \\
&\geq \bar{C}^i_k(M, 0) + \mathcal{V}^{i*}_{k+1}(AMA' + S) \\
&> \bar{C}^i_k(M, 1) + \mathcal{V}^{i*}_{k+1}(\mathbf{0})
\end{aligned} \tag{5.40}$$

Therefore, unilateral deviation is harmful (strictly non-profitable) for the player i, and that allows us to conclude $s^i_k(I_k) = 1$ for $i = 1, 2$ is an equilibrium for (k, M).

Therefore $(s^1_k(I_k), s^2_k(I_k)) = (0,0)$ and $(1,1)$ both are equilibria. However, the cost-to-go by selecting $(1,1)$ is strictly lesser than selecting $(0,0)$, and this is, therefore, preferable by the players.

Note that, (5.37) can be calculated and stored offline and (5.39) can be evaluated online using the stored values.

Equation (5.39) is equivalent to:

$$\lambda_i < \mathcal{V}^{i*}_{k+1}(AMA' + S) - \mathcal{V}^{i*}_{k+1}(\mathbf{0}) - tr\big((P^i_k - Q^i)(AMA' + S)\big) \tag{5.41}$$

which shows a threshold policy for SPE switching.

We note that $M_{0|-1} = \Sigma_0$, therefore at time 0 we only need the value $\mathcal{V}^{i*}_0(\Sigma_0)$ not the function $\mathcal{V}^{i*}_0(\cdot)$ in the entire space of symmetric positive semidefinite matrices. In order to decide $(s^{1*}_0(I_0), s^{2*}_0(I_0))$ we need to know only four values $\mathcal{V}^{i*}_1(\mathbf{0})$, $\mathcal{V}^{i*}_1(A\Sigma_0 A' + S)$ for $i = 1, 2$. Therefore, given the variance of X_0, we need to store only finite number of values to characterize all the value functions for a finite duration game.

Claim 5.4.4. *The maximum number of values (value function evaluations) needed to be stored to calculate the switching strategies for entire game of duration* $[0, T]$ *is* $T(T+3)$.

proof Let at stage k, $M_{k|k-1}$ (or M_{k-1}) takes n_k number of possible distinct values based on all possible previous history I_k. Therefore to determine the switching at time k, we need to make n_k comparison tests (5.39) and for each test the $\mathcal{V}^{i*}_{k+1}(\mathbf{0})$ term is common. Therefore we need to evaluate the value function only at $n_k + 1$

number of points at time k.

For the switching pair $(1,1)$, $M_k = 0$ (or $M_{k+1|k} = S$) and for any other possible switching profile at stage k, $M_k = M_{k|k-1}$. Therefore at stage $k+1$, n_{k+1} will be at most $n_k + 1$ (and n_{k+1} possible values of M_k.) Therefore,

$$n_{k+1} \leq n_k + 1 \tag{5.42}$$

with $n_0 = 1$, we get $n_k \leq k+1$.

Total value function evaluations to be stored $=$

$$2 * \sum_{k=0}^{T-1} (n_k + 1) \leq T(T+3).$$

The factor 2 in above equation is due to the fact that we have to evaluate the value functions for both the players.

Remark 5.4.5. *A switching is performed only when it strictly reduces the cost-to-go for both users. Therefore, each switching minimizes the welfare cost-to-go. However, the converse is not necessarily true i.e. a switching with a potential to reduce the welfare cost-to-go may not always be performed.*

5.4.2 Centralized Optimization vs. Game Setup

The problem we consider here is a game theoretic setup between two players with their own optimization criterion with two actions (control and switch). While they can select their controllers independently, however, their individual switching action does not affect the system (and cost) unless they switch synchronously. A valid question to ask is how a centralized agent would select its action strategies in order to optimize the welfare cost (i.e. the sum of two individual players' cost).

We have shown in Theorem 5.3.1 that the control strategy is totally characterized by Riccati equations for two-player setup. Similar analysis would show that same characteristics for the control strategy are true for the centralized agent. However, it will have a single Riccati equation as opposed to two equations that we have. Similarly, the gain of the controller might change. Considering the symmetric case i.e. $B^1 = B^2$, $Q^1 = Q^2$, $Q^{12} = Q^{22} = Q^{11} = Q^{21}$ we can show that the control strategy for the centralized agent will be equivalent to the strategies of the two agents (i.e. $L_t = \begin{bmatrix} L_t^1 \\ L_t^2 \end{bmatrix}$), Therefore, for a fixed switching strategy, the optimal welfare cost is the same for both, the game setup and the centralized structure. However, the centralized switching strategy will be different from game switching if $\lambda_1 \neq \lambda_2$.

The above anomaly is seen since, in our model, the (selfish) players will not switch unless the switching strictly reduces their own cost, even though the switch might reduce the social welfare cost. However, the social welfare cost will always be minimized when we give the switching control to a centralized entity with the cost-to-go at stage k being the social welfare $\mathcal{V}_k = \mathcal{V}_k^1 + \mathcal{V}_k^2$. It is straightforward to notice $\mathcal{V}_k^* \leq \mathcal{V}_k^{1*} + \mathcal{V}_k^{2*}$.

The centralized switching strategy is given by

$$\bar{C}_k(M, 0) + \mathcal{V}_{k+1}^*(AMA' + S) > \bar{C}_k(M, 1) + \mathcal{V}_{k+1}^*(\mathbf{0}) \tag{5.43}$$

where $\bar{C}_K(\cdot,\cdot) = \bar{C}_k^1(\cdot,\cdot) + \bar{C}_k^2(\cdot,\cdot)$. An interesting study will be to characterize the social loss $l_k = \mathcal{V}_k^{1*} + \mathcal{V}_k^{2*} - \mathcal{V}_k^*$.

This is also known as price of anarchy and it will be studied elsewhere.

5.5 Simulation Results

We consider the following two-dimensional system to illustrate our analysis that has been carried out in the preceding sections.

$$X_{k+1} = \begin{bmatrix} 0.4 & 0.8 \\ -0.8 & 1 \end{bmatrix} X_k + U_k^1 - U_k^2 + W_k$$

where $X_k, U_k^1, U_k^2, W_k \in \mathbb{R}^2$ for all k. $W_k \sim \mathcal{N}(0, 0.25\mathbf{I})$. The observation cost parameters $\lambda_1 = 1$ and $\lambda_2 = 1.5$.

For the cost (5.6), the following parameters are taken:
$Q^1 = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.7 \end{bmatrix}$, $Q^2 = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.2 \end{bmatrix}$, $Q^{12} = Q^{21} = \mathbf{0}$ and $Q^{11} = Q^{22} = \mathbf{I}$.

One can show that $L_{t-1}^i = (-1)^{i-1} P_t^i (\mathbf{I} + P_t^1 + P_t^2)^{-1} A$. By denoting $P_t = \mathbf{I} + P_t^1 + P_t^2$, one can verify:

$$P_t^i = Q^i + A' P_{t+1}^{-1} P_{t+1}^i (P_{t+1}^i + 1) P_{t+1}^{-1} A$$

$$P_T^i = Q^i$$

We set the horizon of the game to be $T = 15$ and assume that X_0 is known to the players i.e. $M_0 = \mathbf{0}$.

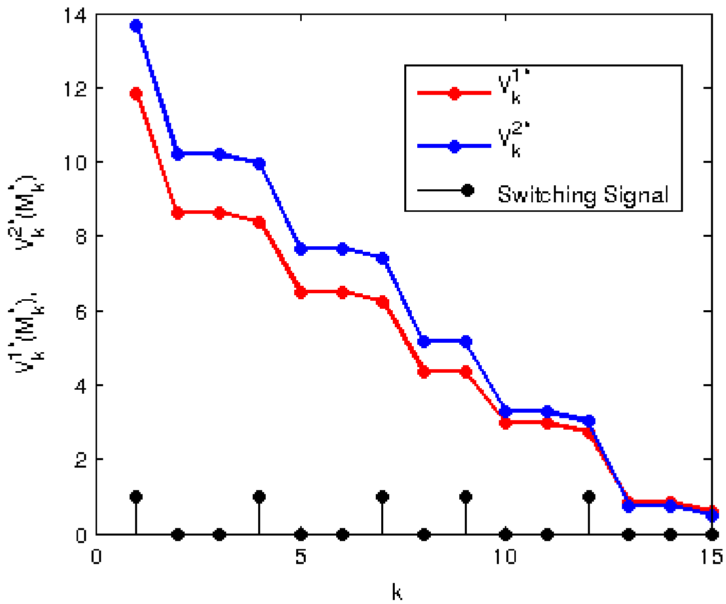


Figure 5.2: The red and blue lines plot $\mathcal{V}_k^{i*}(M^*)$ w.r.t k for $i = 1$ and 2 respectively. M^* is the optimal trajectory of M_k for the optimal switching strategy (s^{1*}, s^{2*}). The black dots show the behavior of the optimal switching signal.

In Figure 5.2, we show the optimal switching strategy $\Delta_k^*(\equiv V_k^{1*} \cdot V_k^{2*})$ in black dots. In a game with horizon 15, the switch was closed for 5 times. In red line, we plot the value function $\mathcal{V}_k^{1*}(M_k^*)$ along the optimal trajectory of M_k determined by (5.19) and the optimal Δ_k^*. Similarly, in blue lines we plot $\mathcal{V}_k^{2*}(M_k^*)$.

In Figure 5.3, we illustrate a comparative result for the cases when observation costs are finite ($\lambda_1 = 1, \lambda_2 = 1.5$) and when observation costs are infinite (so that no observation is practically acquired). In this figure we see that even with 5 observations (out of 15 possible), there are more than 50% reductions in costs. The dotted curves in this figure also indicate the envelop of $\mathcal{V}_k^{i*}$. In other words, all the graphs of $\mathcal{V}_k^{i*}(M_k^*)$ obtained by varying the pair (λ_1, λ_2) will remain below the dotted lines shown in Figure 5.3.

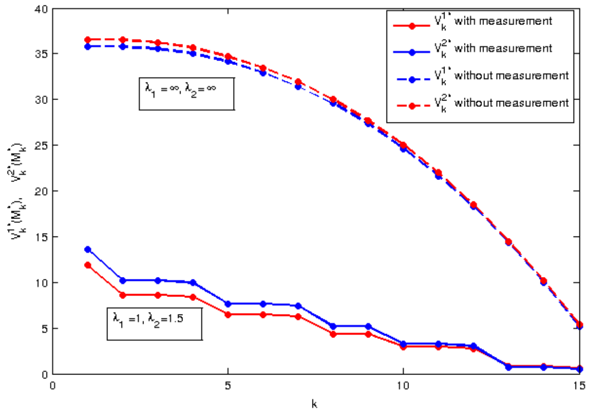


Figure 5.3: A comparison among the costs for the cases when costly measurements are available ($\lambda_1 = 1, \lambda_2 = 1.5$) and no measurements are available ($\lambda_1 = \lambda_2 = \infty$)

5.6 Conclusion

In this chapter, we have considered a switched stochastic LQ game where the switching carries a finite cost. We have characterized the SPE control and switching strategies for both the players. The SPE control strategy turns out to be a linear strategy characterized by Riccati equations which do not depend on the switching strategy. The quality of state estimation depends on the switching strategy and hence the switching cost-to-go function depends on the estimation error variance. We have shown that no-switch (open switch) is always a SPE. However, at certain time instances coordinated switching is also a SPE. Moreover when both no-switch and switch are SPE, then the cost-to-go with switching is lower than the same with no-switching for both players. We studied a two-player game, however similar analysis is easily carried out for a general n-player game.

5.7 Appendix

5.7.1 Proof of Theorem 5.3.1

The idea of the proof is based on backward induction. It should be noted that $\hat{X}_t$ satisfies the Kalman-filter like equations except the fact that the measurements are only available only through a switching and we always get noise free measurements whenever a switching is done.

We define the filtered variable as $\hat{X}_t = \mathbb{E}[X_t \mid \bar{I}_t]$ and the prediction variable as $\hat{X}_{t+1|t} = \mathbb{E}[X_{t+1}| \ I_t]$.

$$\hat{X}_{t|t-1} = A\hat{X}_{t-1} + B^1U^1_{t-1} + B^2U^2_{t-1} \tag{5.44}$$

Therefore,

$$\hat{X}_t = (1 - \Delta_t)\hat{X}_{t|t-1} + \Delta_t X_t$$

where $\Delta_t = V^1_t \cdot V^2_t$.

$\hat{X}_{t|t-1}$ satisfies the dynamics (5.11). In a compact form, one can check

$$\hat{X}_t = A\hat{X}_{t-1} + B^1U^1_{t-1} + B^2U^2_{t-1} + \Delta_t(AE_{t-1} + W_{t-1}) \tag{5.45}$$

where $E_t = X_t - \hat{X}_t$. Thus it satisfies the difference equation:

$$E_t = (1 - \Delta_t)(AE_{t-1} + W_{t-1}) \tag{5.46}$$

Therefore, we can write

$$\hat{X}_t = \hat{X}_{t|t-1} + \Delta_t(AE_{t-1} + W_{t-1}) \tag{5.47}$$

Let P_t^i satisfy the following backward equation for $i = 1, 2$:

$$\begin{aligned} P_t^i =& Q^i + L_t^{i\prime} Q^{ii} L_t^i + L_t^{j\prime} Q^{ij} L_t^j + (A - B^i L_t^i - B^j L_t^j)' P_{t+1}^i (A - B^i L_t^i - B^j L_t^j) \\ P_T^i =& Q^i, \end{aligned} \tag{5.48}$$

and L_t^i satisfies the relation:

$$\begin{aligned} &\Big(Q^{ii} + B^{i\prime} P_t^i \big(I - B^j (Q^{jj} + B^{j\prime} P_t^j B^j)^{-1} B^{j\prime} P_t^j\big) B^i\Big) L_{t-1}^i \\ &= B^{i\prime} P_t^i (I - B^j (Q^{jj} + B^{j\prime} P_t^j B^j)^{-1} B^{j\prime} P_t^j) A \end{aligned} \tag{5.49}$$

Let us consider the cost segment for player i for the given switching strategy profile (s^{1*}, s^{2*}):

$$\begin{aligned}\mathbb{J}_k^i(g^1, g^2) =& \mathbb{E}\Big[\sum_{t=k}^{T} C_t^i(X_t, U_t^1, U_t^2, V_t^1, V_t^2) |\ \bar{I}_k\Big] \\ =& \mathbb{E}\Big[\sum_{t=k}^{T} C_t^i(\hat{X}_t, U_t^1, U_t^2, V_t^1, V_t^2) + \|E_t\|_{Q^i}^2 |\ \bar{I}_k\Big] \\ =& \mathbb{E}\Big[\sum_{t=k}^{T-1} (\|\hat{X}_t\|_{Q^i}^2 + \|U_t^i\|_{Q^{ii}}^2 + \|U_t^j\|_{Q^{ij}}^2) + \|\hat{X}_T\|_{Q^i}^2 |\ \bar{I}_k\Big] + \\ & \mathbb{E}\Big[\sum_{t=k}^{T-1} (\|E_t\|_{Q^i}^2 + \lambda_i \Delta_t) + \|E_T\|_{Q^i}^2 |\ \bar{I}_k\Big]\end{aligned} \tag{5.50}$$

Let us denote $\mathbb{J}_k^{1*} = \min_{g^1} \mathbb{J}_k^1(g^1, g^{2*})$ and $g^{1*} = \arg\min_{g^1} \mathbb{J}_k^1(g^1, g^{2*})$. Similarly we define $\mathbb{J}_k^{2*}$ and g^{2*}.

Claim 5.7.1. *For all k,*

$$\mathbb{J}_k^{i*} = \mathbb{E}\Big[\sum_{t=k}^{T-1} (\|E_t\|_{Q^i}^2 + \Delta_{t+1}\|AE_t + W_t\|_{P_{t+1}^i}^2 + \lambda_i \Delta_t) + \|E_T\|_{Q^i}^2 |\ \bar{I}_k\Big] + \|\hat{X}_k\|_{P_k^i}^2 \tag{5.51}$$

where $\Delta_T = 0$.

proof This is proven by induction. It is easy to check that conditioned under $\bar{I}_t$, E_t and $\hat{X}_t$ are uncorrelated.

Hence $\mathbb{E}[\|X_t\|_{Q^i}^2 |\ \bar{I}_t] = \mathbb{E}[\|\hat{X}_t\|_{Q^i}^2\ |\ \bar{I}_t] + \mathbb{E}[\|E_t\|_{Q^i}^2 |\ \bar{I}_t]$.

Therefore, at $k = T$, the above claim is true. At $k = T - 1$,

$$\mathbb{J}^{i*}_{T-1} = \min_{g^i_{T-1}} \mathbb{E}\Big[\|\hat{X}_{T-1}\|^2_{Q^i} + \|U^i_{T-1}\|^2_{Q^{ii}} + \|U^j_{T-1}\|^2_{Q^{ij}} + \\ \|\hat{X}_T\|^2_{Q^i}|\ \bar{I}_{T-1}\Big] + \mathbb{E}\Big[\|E_{T-1}\|^2_{Q^i} + \lambda_i\Delta_{T-1} + \|E_T\|^2_{Q^i}|\ \bar{I}_{T-1}\Big] \quad (5.52)$$

Using $\hat{X}_T = A\hat{X}_{T-1} + B^1U^1_{T-1} + B^2U^2_{T-2}$, we obtain

$$g^{i*}_{T-1} = -L^i_{T-1}\hat{X}_{T-1}$$

Consequently, the claim holds for $k = T - 1$.

Let us assume that the claim holds for some $k + 1 \leq T$. Then,

$$\begin{aligned}\mathbb{J}^{i*}_k =& \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|E_t\|^2_{Q^i} + \lambda_i\Delta_t) + \|E_T\|^2_{Q^i}|\ \bar{I}_k\Big] + \\ & \min_{g^i} \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|\hat{X}_t\|^2_{Q^i} + \|U^i_t\|^2_{Q^{ii}} + \|U^j_t\|^2_{Q^{ij}}) + \|\hat{X}_T\|^2_{Q^i}|\ \bar{I}_k\Big] \\ =& \min_{g^i_k} \mathbb{E}\Big[\|\hat{X}_k\|^2_{Q^i} + \|U^i_k\|^2_{Q^{ii}} + \|U^j_k\|^2_{Q^{ij}} + J^{i*}(k+1)|\ \bar{I}_k\Big] + \mathbb{E}\Big[\|E_k\|^2_{Q^i} + \lambda_i\Delta_k|\ \bar{I}_k\Big]\end{aligned} \quad (5.53)$$

We used that fact that $\bar{I}_k \subset \bar{I}_{k+1}$ and hence

$$\mathbb{E}[\mathbb{E}[X|I_{k+1}]|I_k] = \mathbb{E}[X|I_k].$$

Therefore using the hypothesis that the claim holds for $k + 1$, we can write

(5.53) as:

$$\mathbb{J}_k^{i*} = \min_{g_k^i} \mathbb{E}\Big[\|\hat{X}_k\|^2_{Q^i} + \|U_k^i\|^2_{Q^{ii}} + \|U_k^j\|^2_{Q^{ij}} + \|\hat{X}_{k+1}\|^2_{P^i_{k+1}}|\ \bar{I}_k\Big]$$
$$+ \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|E_t\|^2_{Q^i} + \lambda_i\Delta_t) + \|E_T\|^2_{Q^i}|\ \bar{I}_k\Big] + \mathbb{E}\Big[\sum_{t=k+1}^{T-1}\Delta_{t+1}\|AE_t + W_t\|^2_{P^i_{t+1}}|\ \bar{I}_k\Big] \tag{5.54}$$

Note that,

$$\hat{X}_{k+1} = A\hat{X}_k + B^1U_k^1 + B^2U_k^2 + \Delta_{k+1}(AE_k + W_k)$$

and therefore,

$$\mathbb{E}[\|\hat{X}_{k+1}\|^2_{P^i_{k+1}}|\ \bar{I}_k] = \mathbb{E}[\|A\hat{X}_k + B^1U_k^1 + B^2U_k^2\|^2_{P^i_{k+1}} + \Delta_{k+1}\|(AE_k + W_k)\|^2_{P^i_{k+1}}|\ \bar{I}_k]$$

. As a result, we obtain,

$$\mathbb{J}_k^{i*} = \min_{g_k^i} \mathbb{E}\Big[\|\hat{X}_k\|^2_{Q^i} + \|U_k^i\|^2_{Q^{ii}} + \|U_k^j\|^2_{Q^{ij}} + \|A\hat{X}_k + B^1U_k^1 + B^2U_k^2\|^2_{P^i_{k+1}}|\ \bar{I}_k\Big]$$
$$+ \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|E_t\|^2_{Q^i} + \lambda_i\Delta_t) + \|E_T\|^2_{Q^i}|\ \bar{I}_k\Big] + \mathbb{E}\Big[\sum_{t=k}^{T-1}\Delta_{t+1}\|AE_t + W_t\|^2_{P^i_{t+1}}|\ \bar{I}_k\Big] \tag{5.55}$$

Note that $\hat{X}_t$ is $\bar{I}_t$ measurable for all t. Thus, we can say from (5.55) that the optimal U_k^1 for player 1 should be given by:

$$U_k^1 = -(Q^{11} + B^{1\prime} P_{k+1}^1 B^1)^{-1} B^{1\prime} P_{k+1}^1 (A\hat{X}_k + B^2 U_k^2) \tag{5.56}$$

Similarly, for player 2, it can be shown that the optimal U_k^2 will be:

$$U_k^2 = -(Q^{22} + B^{2\prime} P_{k+1}^2 B^2)^{-1} B^{2\prime} P_{k+1}^2 (A\hat{X}_k + B^1 U_k^1) \tag{5.57}$$

Comparing the expressions for optimal U_k^i and along with the definition of L_k^i matrices we obtain (basically solving the two linear equations in U_k^i):

$$U_k^i = g_k^{i*}(\bar{I}_k) = -L_k^i \hat{X}_k \tag{5.58}$$

Now substituting the optimal U_k^i in (5.55), and using the definition of P_k^i from (5.48) we get:

$$\mathbb{J}_k^{i*} = \mathbb{E}\Big[\sum_{t=k}^{T-1}(\|E_t\|_{Q^i}^2 + \Delta_{t+1}\|AE_t + W_t\|_{P_{t+1}^i}^2 + \lambda_i \Delta_t) + \|E_T\|_{Q^i}^2 |\ \bar{I}_k\Big] + \|\hat{X}_k\|_{P_k^i}^2$$

Chapter 6: Part II

Motion Planning in Dynamic Environments with Bounded Time Temporal Logic Specifications

In this segment of the thesis, we consider the problem of robotic motion planning that satisfies some bounded time high level specifications. in this chapter we will synthesize a continuous feedback control, and in the following chapters we will focus of synthesizing an event-triggered intermittent feedback control for similar problems.

Although temporal logic can efficiently express high level specifications such as coverage, obstacle avoidance, temporal ordering of tasks etc., it fails to address problems with explicit timing constraints. The inherent limitations of Linear Temporal Logic (LTL) to address problems with explicit timing constraints have been overcome by translating the planning problem from the workspace of the robot to a higher dimensional space called *spacetime* where the existing LTL semantics and grammar are sufficient to mathematically formulate the bounded time high level specifications. A discrete path will be generated, that will meet the specifications with all timing constraints and, at the same time, it will optimize some cost function. A continuous trajectory satisfying the continuous dynamics of the robot will be generated from the discrete path using proper control laws.

6.1 Introduction

Motion planning [54], [55] of a robot is mainly maneuvering the robot from its initial position and configuration to a final position and configuration while maintaining all physical and environmental constraints. This started with the focus on finding optimal trajectories to reach the goal while avoiding obstacles [19] and then evolved into generating plans or paths for complex planning objectives in dynamic or even more complex environments [87], [56]. Planning problems in dynamic or uncertain environments were approached mainly by velocity tuning method [48]. Studies have been also done on the complexities of planning problems in dynamic environments [81], [35]. Though these techniques along with the theories of traditional optimal control with artificial potential functions [19], [103] or cell decomposition or sampling based methods [55] served as promising approaches for robotic path planning, they failed to address problems with multiple goals or a particular sequence of goals.

Researchers have come up with novel formulations and efficient computational approaches to mathematically formulate specifications such as motion sequencing, synchronization etc. Temporal logics such as linear temporal logic (LTL), computational tree logic (CTL), developed for model checking, have been widely accepted by the robotics community for the purpose of motion planning [36], [92]. Development of sophisticated model checking tools such as SPIN [46] and NuSMV [20] made it easier to generate discrete robot paths satisfying the objectives or to produce counter examples proving that the objectives are not achievable. Though temporal logic can efficiently overcome the previous issues with motion sequencing, planning with multiple goals, however, they cannot mathematically formulate a planning problem with time bounded objectives.

In robotics and related fields, there is a class of problems where timing constraints are common, such as simple as “go to position 1 by time t_1 and eventually go to position 2 ”. Even in a simple navigation problem, if the environment is dynamic, we have to face timing constraints which cannot be handled by LTL. Planning in dynamic environments has been done in heuristic ways which do not necessarily give the optimal solution [48], [35]. Planning with time bounded objectives are hard because the discrete path has to be generated in such a way that one can find an equivalent continuous path respecting all the constraints in the dynamics of the robot and simultaneously satisfying the specifications. [9] considered planning in dynamic environments for time bounded temporal logic specifications, however, the dynamics of the robot were modeled using probabilistic Markov models. Mixed Integer Linear Programming (MILP) approaches have been also considered to solve planning problems by formulating the problems as mixed integer linear optimization problems [84], [99], [49].

Metric Temporal Logic (MTL) deals with model checking under timing constraints. The complexity of MTL model checking is undecided and MITL (Metric Interval Temporal logic), a subset of MTL, has the complexity of EXPSPACE-complete [77], whereas LTL is PSPACE-complete [77]. Signal Temporal Logic (STL), similar to MTL also performs model checking with timing constraints [33], [32]. Temporal logic has been widely used by the robotics community [36], [92], [9] to solve problems with complex tasks and that motivates us to use LTL for the planning problems with time constraints. Based on the facts that LTL is computationally less expensive and that there is good availability of tools to check LTL specifications, the goal in this chapter will be to translate a bounded time high level specification to a purely LTL specification.

So far the temporal logic planning problems, which generally do not include any explicit timing specification, have been solved in the workspace/configurationspace of the robot [36], [92], [9]. We will introduce time explicitly along with the workspace and plan in a higher dimensional space. Hereafter this higher dimensional space will be called *spacetime*. Any discrete trajectory generated in spacetime will be forced to meet the explicit timing constraints.

In this chapter, we will consider completing a task with multiple subtasks and some subtasks have to be finished within certain time bound while avoiding the moving and static obstacles in a dynamic environment. We are also interested in minimizing some cost function along the way of task completion. For this chapter, we consider the cost function to be the total time to complete the work. One can consider any cost function with the same framework that we are going to propose (as pointed out in section 6.5.1 after remark 6.5.13). To solve the problem, we will borrow some bounded time temporal operators from MTL and then translate them into usual LTL operators on spacetime. To accomplish the goals, we propose a framework to extend the LTL to incorporate both finite and infinite (e.g. periodic surveillance etc.) duration tasks. A method to synthesize the suitable control laws is proposed to steer the robot while obeying all constraints. An automata theoretic approach [21], [93] is adopted to check whether the problem specifications can be met. Finally, we are also interested in finding control laws that will generate a continuous trajectory which is equivalent to the discrete path generated by the automaton.

6.2 Problem Formulation

We consider a robot whose dynamics are given by (6.1).

$$\dot{x} = f(t, x, u),\ \ x(0) \in \mathcal{X}_0,\ \ x(t) \in \mathcal{X},\ \ u(t) \in \mathcal{U} \tag{6.1}$$

$x(t)$ is the position of the robot at time $t \geq t_0$, $\mathcal{X}_0 \subseteq \mathcal{X}$ is a compact set that represents the set of possible initial positions of the robot. The goal of this chapter is to find a control law $u(t) \in \mathcal{U}$ so that the trajectory generated by (6.1) follows some time specific high level requirements. We consider the presence of static as well as time varying objects within the environment where the robot stays for any time $t(\geq t_0)$. The environment is modeled as time varying and the dynamic properties of the environment can be used, for example, to describe the presence of moving obstacles or to describe the state of a door being open or closed at time t.

Let $\Pi = \{\pi_1, \pi_2, \cdots \pi_n\}$ be the set of atomic propositions which labels $\mathcal{X}$ as a collection of rooms, doors, free space, obstacles etc. As the environment is dynamic, there could be a moving body in some part of the free space making that part to be treated as obstacle for that time. The occupied place becomes free space as soon as the body moves to a new place. Thus, labeling the environment is time dependent. We define a map, F, to label the time varying environment.

$$F : \mathcal{X} \times \mathcal{I} \rightarrow 2^{\Pi} \tag{6.2}$$

where $\mathcal{I} = \{[a, b)| \ \ b \geq a \geq 0\}$ and throughout the chapter 2^{Ω} denotes the power set of a set Ω. We also define $\Pi_{\mathcal{I}} = \{\pi_{k,\mathbf{I}}| \ \pi_k \in \Pi, \mathbf{I} \in \mathcal{I}\}$ and a mapping $F_{\mathcal{I}} : \mathcal{X} \rightarrow 2^{\Pi_{\mathcal{I}}}$ s.t. $F_{\mathcal{I}}(x) = \pi_{k,\mathbf{I}}$ iff $F(x, \mathbf{I}) = \pi_k$.

The general problem we are interested in solving is:

Problem 6.2.1. *Given a workspace, a bounded time high level task (ϕ) and a cost function, find suitable control law ($u(t)$) such that the robot with dynamics (6.1) complete the given task while avoiding collision with all moving and static obstacles in the environment and minimizes the cost function.*

Given any environment, one can approximate and decompose it in polygonal cells [54], [55], [19] and obtain a cellularly decomposed environment similar to one in Fig. 6.1. Cellular decomposition of the workspace is a well studied problem and the interested readers are directed to [54], [55, Section 6.3] and the references therein. For the rest of the chapter we will consider our planning problem in the workspace shown in Fig. 6.2 where the spacetime is represented for the planning problem. The blue segments in Fig. 6.2 represent walls, the red segments represent doors and the black continuous curve is the trajectory of the moving obstacle. The doors may be closed for certain time period or open otherwise as shown in Fig. 6.2 with the black surface. The 2D projection of this spacetime on the workspace is given in Fig. 6.1. The initial position of the robot is I (the yellow cell). Only to illustrate the problem clearly, 2D workspace is considered and time is represented as the third dimension; otherwise, the same framework works for 3D workspace as well. We consider the following example problem to illustrate our method.

Example 6.2.2. *Starting from I, visit R_3 within the time interval $\mathbf{I}_1$, visit R_4 within time interval $\mathbf{I}_2$; before visiting R_3 or R_4, robot must visit R_2. Eventually visit R_1 and R_5, and complete the whole task in the least time.*

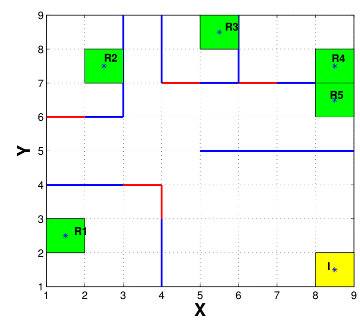


Figure 6.1: Rectangular decomposition on the workspace of the robot (numbers on the X and Y axes are only to uniquely identify a cell)

Like Problem 6.2.1, Example 6.2.2 has two aspects: first, it requires some jobs

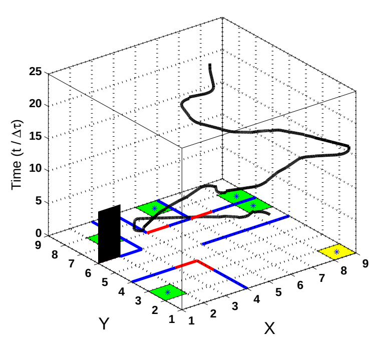


Figure 6.2: Discretized spacetime with obstacles (The black surface represents the fact that the door to that region is closed for that time duration. The black curve is the continuous trajectory of a moving obstacle.)

to be done within specific time interval and second, it has an optimization aspect of completing the whole task in the least time. The task specification (without the optimization part) itself cannot be expressed using LTL logic due to the explicit timing specifications. The existing LTL grammar and semantics have to be extended to capture these types of explicit timing specifications, and for this purpose, we will define some notations and operators for LTL.

6.3 Preliminaries

This section provides some concepts and notations on LTL, *Finite Transition System* (FTS) and *Büchi Automata* (BA) which will be used throughout the chapter. Let us denote the trajectory of the system (6.1) starting at t_0 as $x[t_0] = \{x(s)\ |s \geq t_0 \ \ \dot{x} = f(t,x,u),\ x(t_0) = x_0\}$.

Definition 6.3.1. *The syntax of LTL formulas are defined according to the following grammar rules:*

$$\phi ::= \top \mid \pi \mid \neg\phi \mid \phi \vee \phi \mid \phi \mathbf{U} \phi \mid \phi \mathbf{R} \phi$$

where $\pi \in \Pi$, $\top$ and $\bot (= \neg\top)$ are the Boolean constants `true` and `false` respectively. $\vee$ denotes the conjunction operator and $\neg$ denotes the negation operator. **U** and **R** symbolize the *Until* and *Release* operators respectively. Other temporal logic operators such as eventually ($\Diamond$), always ($\Box$) etc. can be represented using the grammar in definition 6.3.1.

Definition 6.3.2. *The semantics of any formula ϕ is recursively defined as:*

$x[t_0] \models \pi$ *iff* $\pi \in F(x(t_0), t_0)$

$x[t_0] \models \neg\pi$ *iff* $\pi \notin F(x(t_0), t_0)$

$x[t_0] \models \phi_1 \vee \phi_2$ *iff* $x[t_0] \models \phi_1$ *or* $x[t_0] \models \phi_2$

$x[t_0] \models \phi_1 \wedge \phi_2$ *iff* $x[t_0] \models \phi_1$ *and* $x[t_0] \models \phi_2$

$x[t_0] \models \phi_1 \mathbf{U} \phi_2$ *iff* $\exists s \geq t_0$ *s.t.* $x[s] \models \phi_2$ *and* $\forall\ t_0 \leq s' < s,\ x[s'] \models \phi_1$.

$x[t_0] \models \phi_1 \mathbf{R} \phi_2$ *iff* $\forall s \geq t_0\ x[s] \models \phi_2$ *or* $\exists s'$ *s.t.* $t_0 \leq s' < s,\ x[s'] \models \phi_1$.

More details on LTL grammar and semantics can be found in [10].

Definition 6.3.3. *A Finite Transition System (FTS) is a tuple* $\mathcal{E} = \{Q, Q_0, \Pi_{I_D}, \mathcal{A}, TS, \rightarrow_{\mathcal{E}}, h_{\mathcal{E}}\}$*, where:*

Q *is a set of states.*

$Q_0 \subseteq Q$ *is the set of possible initial states.*

Π_{I_D} *is the set of atomic propositions.*

$\mathcal{A}$ *is the set of all actions or policies.*

$TS : Q \times \mathcal{A} \rightarrow Q$ *is a mapping that dictates the transition from one state to another by applying some action.*

$\rightarrow_{\mathcal{E}} \subseteq Q \times Q$ *captures the transitional relationship between the states.* $(Q_i, Q_j) \in \rightarrow_{\mathcal{E}}$ *iff* $\exists \alpha \in \mathcal{A}$ *s.t.* $Q_j = TS(Q_i, \alpha)$.

$h_{\mathcal{E}} : Q \rightarrow 2^{\Pi_{I_D}}$ *is the map which assigns atomic propositions to the states where those propositions are satisfied.*

Definition 6.3.4. *A Büchi automaton (BA) is a tuple* $\mathcal{B} = \{S_B, S_{0B}, \Sigma_B, \delta_B, F_B\}$ *where:*

S_B *is a set of states and* S_{0B} *is the initial state.*

Σ_B *is the set of input alphabets.*

$\delta_B : S_B \times \Sigma_B \rightarrow 2^{S_B}$ is a transition relationship.

$F_B \subseteq S_B$ is a set of accepting states.

For each LTL formula ϕ, a corresponding Büchi automaton can be generated [21] that accepts the words which satisfy the specification ϕ. Generating a Büchi automaton from a given LTL formula is a well studied problem and the interested readers may confer [21].

6.4 Extended Linear Temporal Logic

Two new operators $\mathbf{U_I}$ and $\mathbf{R_I}$ are introduced as follows:

Definition 6.4.1. *The extension of the LTL grammar is given by: $\phi ::= \phi \mathbf{U_I} \phi \mid \phi \mathbf{R_I} \phi$*

where $\mathbf{I} \in \mathcal{I}$. The semantics $\phi_1 \mathbf{U_I} \phi_2$ means $\exists s \in \mathbf{I}$ s.t. $x[s] \models \phi_2$ and $\forall\ t_0 \leq s' < s,\ x[s'] \models \phi_1$. It should be noted that the expression $\phi_1 \mathbf{U}_{[t_0,\infty)} \phi_2$ is equivalent to $\phi_1 \mathbf{U} \phi_2$. Once we have $\mathbf{U_I}$ and $\mathbf{R_I}$, we can always define other temporal operators such as $\Diamond_{\mathbf{I}}$, $\Box_{\mathbf{I}}$ etc. Thus, this grammar can model both finite and infinite duration tasks.

Like the LTL grammar and semantics in [10], similar grammar and semantics can be defined over the atomic propositions set $\Pi_{\mathcal{I}}$ as follows:

Definition 6.4.2. *The syntax of LTL formulas over $\Pi_{\mathcal{I}}$ are defined according to the following grammar rules:*

$$\phi_I ::= \top_I \mid \pi_I \mid \neg \phi_I \mid \phi_I \vee \phi_I \mid \phi_I \mathbf{U} \phi_I \mid \phi_I \mathbf{R} \phi_I \mid (\phi_I)_I$$

where $\top_I = \pi_I \vee \neg \pi_I$ and $\pi_I \in \Pi_{\mathcal{I}}$.

Definition 6.4.3. *The semantics of any formula* ϕ_I *is recursively defined as:*

$x[t_0] \models \pi_I$ *iff* $t_0 \in I$ *and* $\pi \in F(x(t_0), t_0)$

$x[t_0] \models \neg\pi_I$ *iff either* $t_0 \notin I$ *or* $\pi \notin F(x(t_0), t_0)$

$x[t_0] \models \phi_I$ *iff* $t_0 \in I$ *and* $x[t_0] \models \phi$

$x[t_0] \models \phi_{1,I_1} \vee \phi_{2,I_2}$ *iff* $x[t_0] \models \phi_{1,I_1}$ *or* $x[t_0] \models \phi_{2,I_2}$

$x[t_0] \models \phi_{1,I_1} \wedge \phi_{2,I_2}$ *iff* $x[t_0] \models \phi_{1,I_1}$ *and* $x[t_0] \models \phi_{2,I_2}$

$x[t_0] \models \phi_{1,I_1}\mathbf{U}\phi_{2,I_2}$ *iff* $\exists s \geq t_0$ *s.t.* $x[s] \models \phi_{2,I_2}$ *and* $\forall\ t_0 \leq s' < s,\ x[s'] \models \phi_{1,I_1}$.

$x[t_0] \models \phi_{1,I_1}\mathbf{R}\phi_{2,I_2}$ *iff* $\forall s \geq t_0\ x[s] \models \phi_{2,I_2}$ *or* $\exists s'$ *s.t.* $t_0 \leq s' < s,\ x[s'] \models \phi_{1,I_1}$.

$x[t_0] \models (\phi_{I_1})_{I_2}$ *iff* $x[t_0] \models \phi_{I_1 \cap I_2}$

It is easy to notice that setting I, I_1, I_2 as $[t_0, \infty)$ the usual LTL semantics is obtained. Now we will try to express the operator defined in definition (6.4.1) in terms of the grammar in definition (6.4.3). Let us denote the lower and upper bound of an interval, I, as $\check{I}$ and $\hat{I}$ respectively, i.e. $I = [\check{I}, \hat{I})$.

Proposition 6.4.4. *LTL formula* $\phi_1\mathbf{U}_I\phi_2$ *is equivalent to* $\phi_{1,[t_0,\hat{I})}\mathbf{U}\phi_{2,I}$

Proof. Let $x[t_0] \models \phi_{1,[t_0,\hat{I})}\mathbf{U}\phi_{2,I}$ then $\exists s \geq t_0$ s.t. $x[s] \models \phi_{2,I}$ and $\forall\ t_0 \leq s' < s,\ x[s'] \models \phi_{1,[t_0,\hat{I})}$. Using definition 6.4.3, $x[s] \models \phi_{2,I}$ is rewritten as $s \in I$ and $x[s] \models \phi_2$. Since $s \in I$ and $s' \in [t_0, s)$, then always $s' \in [t_0, \hat{I})$. Therefore, $x[t_0] \models \phi_{1,[t_0,\hat{I})}\mathbf{U}\phi_{2,I}$ if $\exists s(\geq t_0) \in I$ s.t. $x[s] \models \phi_2$ and $\forall\ t_0 \leq s' < s,\ x[s'] \models \phi_1$ which is equivalent to say $x[t_0] \models \phi_1\mathbf{U}_I\phi_2$. Similarly it can be proved that $x[t_0] \models \phi_1\mathbf{U}_I\phi_2$ implies $x[t_0] \models \phi_{1,[t_0,\hat{I})}\mathbf{U}\phi_{2,I}$. □

Remark 6.4.5. *LTL formula* $\phi_1\mathbf{R}_I\phi_2$ *is equivalent to* $\phi_{1,I}\mathbf{R}\phi_{2,[t_0,\infty)}$.

Any bounded time high level specification can be represented by the usual and extended LTL grammars described in definitions (6.3.1) and (6.4.1) and can be converted to an equivalent formula of definition (6.4.2) which only contains the usual LTL operators. The advantage of having usual LTL operators is that we can easily translate the formula to a B$\ddot{u}$chi automaton which is an important and necessary step for model checking.

The explicit time dependent part of example 6.2.2 can be represented by the LTL formula $\phi = (\neg\mathcal{O}\mathbf{U}_{\mathbf{I}_1}R_3) \wedge (\neg\mathcal{O}\mathbf{U}_{\mathbf{I}_2}R_4)$; where $\mathcal{O}$ represents the set of obstacles (the walls, moving obstacles or the closed doors).

6.5 Robot Motion Planning

In this section a discrete path that will satisfy the requirements of example 6.2.2 for the robot will be generated.

6.5.1 Generating Discrete Path

To proceed with the formal verification, the workspace is discretized. There exists several techniques for cell decomposition in polygonal environments [55], [19]. We will divide our workspace, $\mathcal{X}$, in rectangles [1]. Let $Q = \{q_1, q_2, ..., q_m\}$ be a partition in the workspace. Let us define a map $T : \mathcal{X} \rightarrow Q$ to partition the continuous workspace $\mathcal{X}$. $\forall x, y \in \mathcal{X}$ and $I \in \mathcal{I}$, the map T has the following properties: a) if $T(x) = T(y)$, then $F(x, I) = F(y, I)$, and b) if $T(x) = T(y)$ and $x' \in q_j$ is reachable from $x \in q_i$, then $\exists y' \in q_j$ such that y' is reachable

[1] See Section 6.5.2 for details

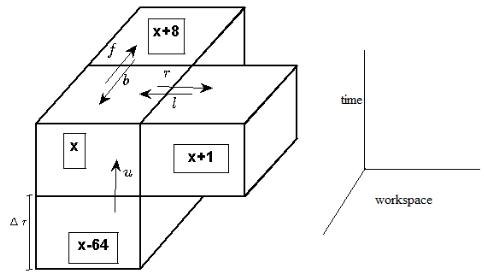


Figure 6.3: Transitional relationship among the blocks in discretized spacetime.

from y. The set Q_0 denotes the set of states where the robot can stay initially i.e. $\cup_{q_0 \in Q_0} T^{-1}(q_0) \subseteq \mathcal{X}_0$. Discretization on the set $\mathcal{I}$ is defined as $\mathcal{I}_D = \{[t_0 + n_1\Delta\tau, t_0 + n_2\Delta\tau)|\ n_2 > n_1 = 0, 1, \cdots\}$. Like the choice of cell size in cell decomposition, the choice of $\Delta\tau$ determines how accurately the dynamicity of the environment is captured.

Let us also define $\mathcal{Q} = \{(q, I_k)|\ q \in Q, I_k = (t_0 + k\Delta\tau, t_0 + (k+1)\Delta\tau] \in \mathcal{I}_D\}$ as the discretization of the continuous spacetime. The mapping $L : \mathcal{Q} \to 2^{\Pi_{\mathcal{I}_D}}$ is defined as $L(q, I_k) = \pi_I$ *iff* $F(x, I) = \pi$ s.t. $T(x) = q$ and $I_k \subseteq I,\ \forall I,\ I_k \in \mathcal{I}_D$. Let $\mathcal{A}$ denotes the set of actions the robot can take in this discretized spacetime. The set of actions available at any particular state $q \in \mathcal{Q}$ will be denoted by $\mathcal{A}_q\ (\subseteq \mathcal{A})$. The discretized spacetime for example 6.2.2, given in Fig. 6.2, contains sixty four blocks (8×8) in each time layer. Fig. 6.3 illustrates the numbering scheme and transitional relations that have been adopted. $x,\ x+1,\ x+8$ and $x-64$ in Fig. 6.3 denote the numbers of the blocks. The arrows show the possible transition from one block to another. The letters (f, b, l, r, u) represent the action to be taken in order to perform that transition. The discretized spacetime can be represented by an FTS($\mathcal{E}$) similar to that given in definition (6.3.3).

Definition 6.5.1. *The equivalent FTS($\mathcal{E}$) of the discretized spacetime is given by a* *tuple* $\{\mathcal{Q}, \mathcal{Q}_0, \Pi_{I_D}, \mathcal{A}, TS, \rightarrow_{\mathcal{E}}, h_{\mathcal{E}}\}$

where $\mathcal{Q}$ *is the set of states.*

$\mathcal{Q}_0 = \{(q_0, I_0) \mid q_0 \in Q_0\}$.

$\mathcal{A} = \{f, b, l, r, u\}$ *where* f, b, l, r, u *are the abbreviation of* $forward$*,* $backward$*,* $left$*,* $right$ *and up respectively.*

TS *denotes the possible transitions upon applying action* $\alpha \in \mathcal{A}$ *on a robot residing at block* q*. As the names suggest, applying* $\mathcal{A}_q = r$ *at a block* q *makes the robot to move to the block right to* q*, provided there is a block right to* q*. If there is no block right to* q*,* $r \notin \mathcal{A}_q$*.*

$\rightarrow_{\mathcal{E}} \subseteq \mathcal{Q} \times \mathcal{Q}$ *captures the topological relationship between the blocks.* $\big((q_i, I_{k_i}), (q_j, I_{k_j})\big) \in \rightarrow_{\mathcal{E}}$ *iff* $\big(k_i = k_j$ *and* q_i*,* q_j *share a common edge, i.e.* $(q_j, I_{k_j}) \in TS\big((q_i, I_{k_i}), \mathcal{A}_q\big)$ *with* $\mathcal{A}_q \in \{f, b, l, r\}\big)$ *or* $(q_i = q_j,\ k_i + 1 = k_j$ *i.e.* $\mathcal{A}_q = u)$*.*

$h_{\mathcal{E}} : \mathcal{Q} \rightarrow 2^{\Pi_{I_D}}$ *is the map same as* L*.*

The *up* action corresponds to movements in the time direction and since time is unidirectional, there is no *down* action present in the FTS.

Let $p : \mathbb{N} \rightarrow \mathcal{Q}$ be a path for the robot in the discretized spacetime ($\mathcal{Q}$) with $p(0) \in \mathcal{Q}_0$ and $(p(i), p(i+1)) \in \rightarrow_{\mathcal{E}}$. The sequence of actions taken is denoted by $\alpha = \alpha_0 \alpha_1 \cdots \alpha_n$ which satisfies the fact that $\forall i = 0, \cdots, n$, $p(i+1) = TS(p(i), \alpha_i)$.

Analogous to the continuous LTL formulas given in definitions (6.4.2) and (6.4.3), we can also have discrete LTL formulas and grammars, on the set of atomic propositions $\big(\Pi_{\mathcal{I}_D}\big)$, as given in definitions (6.5.2) and (6.5.3).

Definition 6.5.2. *The syntax of bounded time discrete LTL formulas are defined* *in the following grammar rules:*

$$\phi_{I_D} ::= \top_{I_D} \mid \pi_{I_D} \mid \neg\phi_{I_D} \mid \phi_{I_D} \vee \phi_{I_D} \mid \phi_{I_D}\mathbf{U}\phi_{I_D} \mid \phi_{I_D}\mathbf{R}\phi_{I_D} \mid \bigcirc\phi_{I_D} \mid (\phi_{I_D})_{I_D}$$

where $\pi_{I_D} \in \Pi_{\mathcal{I}_D}$,$\top_{I_D} = \pi_{I_D} \vee \neg\pi_{I_D}$ and $\bigcirc$ is the next operator.

We define two projection functions $pr_1 : p(i) \to Q$ and $pr_2 : p(i) \to \mathcal{I}_D$ on a path $p = p(0)p(1)\cdots p(l)$ in the discretized spacetime. Let $p(i) = (q_i, I_{k_i}) \in \mathcal{Q}$ then $pr_1(p(i)) = q_i$ and $pr_2(p(i)) = I_{k_i}$. We will use the short hand notation $p[i_0]$ to denote the portion of the path p that starts from $p(i_0)$ i.e. $p[i_0] = p(i_0)p(i_0{+}1)\cdots p(l)$ (with this notation, $p \equiv p[0]$).

Definition 6.5.3. *The semantics of any formula ϕ_{I_D} on a discrete path p is recursively defined as:*

$p \models \pi_{I_D}$ *iff* $pr_2(p(0)) \subseteq I_D$ *and* $\pi_{I_D} \in L(p(0))$

$p \models \neg\pi_{I_D}$ *iff either* $pr_2(p(0)) \nsubseteq I_D$ *or* $\pi_{I_D} \notin L(p(0))$

$p \models \phi_{I_D}$ *iff* $pr_2(p(0)) \subseteq I_D$ *and* $pr_1(p) \models \phi$*.*

$p \models \phi_{1,I_{D1}} \vee \phi_{2,I_{D2}}$ *iff* $p \models \phi_{1,I_{D1}}$ *or* $p \models \phi_{2,I_{D2}}$

$p \models \phi_{1,I_{D1}} \wedge \phi_{2,I_{D2}}$ *iff* $p \models \phi_{1,I_{D1}}$ *and* $p \models \phi_{2,I_{D2}}$

$p \models \bigcirc\phi_{I_D}$ *iff* $p[1] \models \phi_{I_D}$

$p \models \phi_{1,I_{D1}}\mathbf{U}\phi_{2,I_{D2}}$ *iff* $\exists i \geq 0$ *s.t.* $p[i] \models \phi_{2,I_{D2}}$ *and* $\forall\ 0 \leq j < i,\ p[j] \models \phi_{1,I_{D1}}$

$p \models \phi_{1,I_{D1}}\mathbf{R}\phi_{2,I_{D2}}$ *iff* $\forall i \geq 0\ p[i] \models \phi_{2,I_{D2}}$ *or* $\exists j$ *s.t.* $0 \leq j < i,\ p[j] \models \phi_{1,I_{D1}}$

$p \models (\phi_{I_{D1}})_{I_{D2}}$ *iff* $p \models \phi_{I_{D1} \cap I_{D2}}$*.*

The FTS($\mathcal{E}$) of the environment can be translated into an equivalent Büchi automaton as given in definition 6.5.4.

Definition 6.5.4. *The Büchi automaton ($\mathcal{E}'$) corresponding to the FTS($\mathcal{E}$) in definition (6.5.1) is a tuple* $\mathcal{E}' = \{\mathcal{Q}', q_e, 2^{\Pi_{I_D}}, \delta_{\mathcal{E}'}, F_{\mathcal{E}'}\}$ *where:*

$\mathcal{Q}' = \mathcal{Q} \cup q_e$ *for* $q_e \notin \mathcal{Q}$.

$\delta_{\mathcal{E}'} : \mathcal{Q}' \times 2^{\Pi_{I_D}} \rightarrow 2^{\mathcal{Q}'}$ *s.t.* $(q_i, I_{k_i}) \in \delta_{\mathcal{E}'}\big((q_j, I_{k_j}), \pi\big)$ *iff* $\big((q_i, I_{k_i}), (q_j, I_{k_j})\big) \in \rightarrow_{\mathcal{E}}$ *and* $\pi \in h_{\mathcal{E}}(q_j, I_{k_j})$, *and* $(q_0, I_{k_0}) \in \delta_{\mathcal{E}'}\big(q_e, \pi_0\big)$ *iff* $(q_0, I_{K_0}) \in \mathcal{Q}_0$ *and* $\pi_0 \in h_{\mathcal{E}}(q_0, I_{k_0})$

$F_{\mathcal{E}'}$ *is the set of accepting states.*

The aim is to find a path on $\mathcal{E}'$ that satisfies a given LTL formula ϕ_{I_D} (representing the time bounded task specification). That is to find the language, $\mathcal{L}$, which is accepted by both automata $\mathcal{B}_{\phi_{I_D}}$ and $\mathcal{E}'$. This can be done by constructing a product automaton $\mathcal{B}_{\phi_{I_D}} \times \mathcal{E}'$ whose language will be $\mathcal{L}(\mathcal{B}_{\phi_{I_D}}) \cap \mathcal{L}(\mathcal{E}')$.

Definition 6.5.5. *The product automaton* $\mathcal{P} = \{S_{\mathcal{P}}, S_{0\mathcal{P}},\, 2^{\Pi_{I_D}}, \delta_{\mathcal{P}}, F_{\mathcal{P}}\}$ *where:*

$S_{\mathcal{P}} = \mathcal{Q}' \times S_{\mathcal{B}_\phi}$ *and* $S_{0\mathcal{P}} = \{(q_e, S_{0B_\phi})\}$.

$\delta_{\mathcal{P}} : S_{\mathcal{P}} \times 2^{\Pi_{I_D}} \rightarrow 2^{S_{\mathcal{P}}}$ *s.t.* $\big((q_i, I_{k_i}), S_n\big) \in \delta_{\mathcal{P}}\big(((q_j, I_{k_j}), S_m), \pi\big)$ *iff* $(q_i, I_{k_i}) \in \delta_{\mathcal{E}'}((q_j, I_{k_j}), \pi)$ *and* $S_n \in \delta_{B_\phi}(S_m, \pi)$

$F_{\mathcal{P}} = F_{\mathcal{E}'} \times F_{\mathcal{B}_\phi}$

By construction, the language of this product automaton is the common language of both automata $\mathcal{E}'$ and $\mathcal{B}_\phi$, i.e. $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{E}') \cap \mathcal{L}(\mathcal{B}_\phi)$. A run, $r : \mathbb{N} \rightarrow S_{\mathcal{P}}$ of $\mathcal{P}$, is a sequence of states which is obtained by applying an input trace ω; i.e. $r(0) \in S_{0\mathcal{P}}$ and $\forall i \geq 0,\ r(i+1) \in \delta_{\mathcal{P}}(r(i), \omega(i))$. An accepting run of an automaton contains at least one final state. More precisely, a run r of $\mathcal{P}$ over an infinite trace ω is accepting if and only if $iof(r) \cap F_{\mathcal{P}} \neq \emptyset$, where $iof(r)$ is the function that returns the set of states that are encountered infinitely often in the run r. If $\mathcal{L}(\mathcal{P}) \neq \emptyset$, there exists accepting run(s) for $\mathcal{P}$. An infinite length accepting run consists of two parts a prefix and a periodic suffix.

Definition 6.5.6 (Accessible part ($Ac(\mathcal{A})$))**.** *This corresponds to the part of the automata that is reachable from the initial state.*

The accessible part is constructed by deleting all the states that are not reachable from the initial state

Definition 6.5.7 (Co-Accessible part $CoAc(\mathcal{A})$)**.** *This corresponds to the part of the automata that has a path to the accepting state of the automata. $CoAc(\mathcal{A})$ is constructed by deleting all the states of $\mathcal{A}$ from which there is no path to the accepting state(s).*

Definition 6.5.8 (Trim)**.** *The trimmed automata $Trim(\mathcal{A})$ is obtained by retaining only the accessible and co-accessible part.*

$Trim(\mathcal{A}) = CoAc(Ac(\mathcal{A})) = Ac(CoAc(\mathcal{A}))$.

Definition 6.5.9 (Redundant part)**.** *The redundant part of the automata consists of the states that are only accessible via an accepting state.*

Definition 6.5.10 (Pruning)**.** *The pruned automata is constructed by trimming and removing the redundant part of the automata ($\mathcal{A}$).*

Lemma 6.5.11. *Let $\mathcal{E}'$ be the pruned automata representation of the space-time workspace, and $\mathcal{B}_{\phi_{I_D}}$ be the automata representation of the LTL task. Then, by construction $\mathcal{P} = \mathcal{B}_{\phi_{I_D}} \times \mathcal{E}'$ is a sub-automata[2] of $\mathcal{E}'$.*

[2] A sub-automata is consists of the subset of the nodes and the corresponding transitions.

Thus, pruning reduces the complexity of the automata. Instead of being a lager automata, the product automata is a subset of the workspace automata $\mathcal{E}'$. The above Lemma holds true due to the unidirectionality of time associated in the space-time representation of the automata.

Remark 6.5.12. *Let* $pr : S_{\mathcal{P}} \rightarrow \mathcal{Q}'$ *be a projection function such that* $pr(q, s) = q$*; where* $q \in \mathcal{Q}'$ *and* $s \in S_{\mathcal{B}_\phi}$*. If* r *is an accepting run of* $\mathcal{P}$*,* p *is a path on* $\mathcal{E}$*; where* $p(i) = pr(r(i))$*.*

Let us define $\mathcal{R}_{\mathcal{P}} = \{r \mid r \text{ is an accepting run of } \mathcal{P}\}$. Note that the cardinality of $\mathcal{R}_{\mathcal{P}}$ can be more than one.

Remark 6.5.13. *Problem 6.2.2 is feasible iff* $\mathcal{R}_{\mathcal{P}} \neq \emptyset$*.*

We will convert this product automaton $\mathcal{P}$ to a directed weighted graph $\mathcal{G}(V, E, W)$. V is the set of nodes and E, the set of edges, is a binary relation on V, and W is the set of weights associated with the edges. Through this conversion process, the states of the automaton become the nodes of the graph and the transitional relation ($\delta_{\mathcal{P}}$) defines the set of the edges (E). Thus, $S_{\mathcal{P}}$ and V are basically the same set and hence can be related using a bijective mapping Z(say). $(V_i, V_j) \triangleq E_{ij} \in E$ *iff* $\exists \pi_{I_D}$ s.t. $Z^{-1}(V_i) \in \delta_{\mathcal{P}}(Z^{-1}(V_j), \pi_{I_D})$. Depending on the cost function to be minimized, the weights on the edges can be constructed accordingly. Since each edge represents a transition from one node (position and time) to another with the proper application of an action, the position, time and action information are available at each edge. Any cost that is a function of position, time and action can be calculated easily for the edges and can be put as a weight on the edge. For example 6.2.2 the weight w_{ij} associated with the edge E_{ij} is defined to be $w_{ij} = \big(pr_2 \circ pr \circ Z^{-1}(V_j) - pr_2 \circ pr \circ Z^{-1}(V_i)\big)/\Delta\tau$ [where $f \circ g(h) = f(g(h))$]. From the topological and transitional relation given in definition 6.5.1, one can easily check that $w_{ij} \in \{0, 1\}$, and $w_{ij} = 1$ only when $\mathcal{A}_q = u$. Let us denote $V_0 = \{v \mid Z^{-1}(v) \in S_{0\mathcal{P}}\}$ and $V_{\mathcal{P}} = \{v \mid Z^{-1}(v) \in F_{\mathcal{P}}\}$. Let $P_{\mathcal{G}}$ be the set of all paths on the graph $\mathcal{G}$ that start on a node $v \in V_0$ and end on a node $v \in V_{\mathcal{P}}$. $p_g : \mathbb{N} \to V$ be a path on $\mathcal{G}$ such that $p_g(0) \in V_0$, $(p_g(i), p_g(i+1)) \in E$ and $p_g(l) \in V_{\mathcal{P}}$ (l is the length of the path). Cost associated with the link $p_g(i) \to p_g(i+1)$ is $C_{i,i+1} = \big(pr_2 \circ pr \circ Z^{-1}(p_g(i+1)) - pr_2 \circ pr \circ Z^{-1}(p_g(i))\big)/\Delta\tau$. Therefore, a path with the least cumulative link cost will complete the given task in the least time. In other words, the solution of example 6.2.2 is the solution of the optimization problem 6.5.14 which can be solved efficiently by using a suitable graph search algorithm.

Problem 6.5.14.

$$\begin{aligned} \min_{p_g} \quad & \Sigma_{i=0}^{l-1} C_{i,i+1} \\ \textit{subject to} \quad & p_g \in P_{\mathcal{G}} \end{aligned}$$

Due to the equivalence between $\mathcal{P}$ and $\mathcal{G}$, an equivalent run r on $\mathcal{P}$ can be obtained for the path $p_g \in P_{\mathcal{G}}$. Let p be the projection of r on $\mathcal{Q}$ i.e. $p(i) = pr(r(i))$. The time and space sequence of p_g are $pr_2 \circ pr(r(i))$ and $pr_1 \circ pr(r(i))$ respectively. It is possible to find a path $p_g \in P_{\mathcal{G}}$ such that $\forall i = 0, 1, \cdots l$, $pr_2 \circ pr(r(i)) = I_0$. Such a path requires the whole task to be done in $\Delta\tau$ amount of time and hence the cost for that path will be zero. This discrepancy arises since the dynamics or the physical constraints of the robot have not been considered in the formulation of problem 6.5.14. Practically, it may not be plausible to complete the whole task

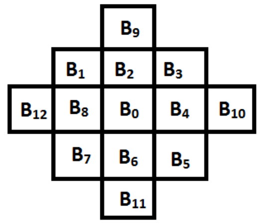


Figure 6.4: Set of reachable cells within time $\Delta\tau$ from cell B_0.

in that small time. The reachability property of the robot has to be incorporated in this planning problem. We will consider the reachable set from a cell $q_i \in Q$ to be the set of cells that can be reached from any point in q_i within $\Delta\tau$ amount of time. For this chapter, we consider $\Delta\tau$ to be the smallest time s.t. for every $x \in B_0$ in Fig. (6.4), $\forall i \in \{1, 2, \cdots, 12\}$, $\exists y \in B_i$ which is reachable from x within $\Delta\tau$ time. This particular reachability property enforces the requirement that between two successive applications of action u, there can be at most two actions from the set $\{f, b, l, r\}$. Let us define a new atomic proposition ξ_k such that $p[i] \models \xi_k$ *iff* $pr_2(p(i)) = I_k$. $\zeta_k = \Box\big((\xi_k \wedge \bigcirc\xi_k \wedge \bigcirc\bigcirc\xi_k) \Rightarrow \bigcirc\bigcirc\bigcirc(\neg\xi_k)\big)$ ensures no more than two transitions within the $k-th$ time layer. Thus, the equivalent LTL specification of the reachability constraint is given by $\phi_{reach} = \bigwedge_{\forall k=0,1,\cdots} \zeta_k$. If ϕ_1 is the LTL representation of example 6.2.2, $\phi = \phi_1 \wedge \phi_{reach}$ ensures that any path satisfying ϕ will solve example 6.2.2 with the reachability constraint.

In this chapter we assume that the reachability profile from the cell B_0 is what presented in Fig. (6.4). However, the actual shape of the reachable sets depends on the dynamics of the robot. In the following we briefly discuss some properties of the reachable sets and their construction.

6.5.2 Reachability Analysis

In the cell decomposition process, the cells are generated based on reachability analysis of the robot's dynamics.

Let us denote by $\mathcal{R}(t_0, X, U, \Delta\tau)$ the reachable set in $\Delta\tau$ amount of time of the robot with starting position $(x(t_0))$ inside X and the applied control law lies in the set U, i.e.,

$$\mathcal{R}(t_0, X, U, \Delta\tau) = \{x_1 \in \mathbb{R}^n \mid \exists u \in U, u : [t_0, t_0 + \Delta\tau] \to \mathcal{U}, \dot{x} = f(t, x, u), \quad x(t_0) \in X, x_1 = x(t_0 + \Delta\tau)\} \tag{6.3}$$

For a controllable LTI system with $\mathcal{U} = \mathbb{R}^m$,

$$\mathcal{R}(t_0, X, \mathbb{R}^m, \Delta\tau) = \mathbb{R}^n$$

for all X, t_0 and $\Delta\tau$, i.e. any point can be reached in arbitrary small time if the control inputs are not constrained.

Theorem 6.5.15. *Let $c_1, c_2, \cdots, c_d$ be a partition in $\mathcal{X}$, and $N_i \subseteq \{c_1, c_2, \cdots, c_d\}$ be the set of neighbor cells of the cell c_i. For a controllable linear system $\dot{x} = Ax + Bu$, any $c_j^i \in N_i$, $j = 1, 2, \cdots, |N_i|$ is reachable from c_i in arbitrary small time.*

The above theorem is trivial due to the controllability assumption of the LTI system.

Theorem 6.5.16. *Let $c_1, c_2, \cdots, c_d$ be a partition in $\mathcal{X}$, and $N_i \subseteq \{c_1, c_2, \cdots, c_d\}$*

be the set of neighbor cells of the cell c_i*. For a controllable nonlinear system of the form* $\dot{x} = G(x)u$*, any* $c^i_j \in N_i$*,* $j = 1, 2, \cdots, |N_i|$ *is reachable from* c_i *in arbitrary small time.*

Proof. Let us consider any to arbitrary points $q_0 \in c_i$ and $q_1 \in c^i_j \in N_i$. Since the nonlinear system is controllable, there exists a control $u(t) : [t_0, t_1] \to \mathbb{R}^m$ such that $x(t_1) = q_1$ where

$$\dot{x} = G(x)u$$

$$x(t_0) = q_0$$

Let us now consider the control:

$$v(t) = \beta u(\alpha + \beta t) \tag{6.4}$$

which drives the system $\dot{x} = G(x)v$.

Let us define $y(t) = x(\frac{t-\alpha}{\beta})$, then:

$$\dot{y} = \frac{1}{\beta}G(y)v(\frac{t-\alpha}{\beta}) = G(y)u(t)$$

Thus, $u(t) : [t_0, t_1] \to \mathbb{R}^m$ ensures $y(t_1) = q_1$, while $y(t_0) = q_0$.

Choosing $\beta = \frac{t_1-t_0}{\lambda}$ and $\alpha = t_0 - \frac{t_1-t_0}{\lambda}\tau_0$ ensures $v(t) : [\tau_0, \tau_0 + \lambda] \to \mathbb{R}^m$ as defined in (6.4) ensures for all τ_0 $\lambda > 0$, $x(\tau_0 + \lambda) = q_1$ if $x(\tau_0) = q_0$. □

6.5.3 From Discrete Path to Continuous Trajectory

The solution of problem 6.5.14 is a discrete path, on the discretized spacetime, that satisfies the temporal specification and minimizes the given cost function. The goal is to find some input $u(t) \in \mathcal{U}$ for all $t \geq t_0$ so that the robot can follow the discrete trajectory. In this chapter, the unicycle robot dynamics given in (6.5) are considered.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{6.5}$$

x and y are the position coordinates and θ is the orientation or heading angle. v and ω are the control inputs. The discrete path is essentially a sequence of blocks that has to be visited. While transitioning from one block to another, the robot must be confined inside those blocks. This imposes some extra conditions that have to be taken care of while generating the continuous trajectory. A state feedback technique is used to achieve this. Let the robot is currently at block q_1 with positions and orientation given by $[x_1, y_1, \theta_1]$ and it has to go to block q_2. For this transition we consider the final position, $[x_2, y_2]$, to be the center of the block q_2 and θ_2 is determined by the following rule:

$$\theta_2 = \begin{cases} 0^o & \text{if } |\Delta x| \geq |\Delta y| \text{ and } \Delta x \geq 0 \\ 180^o & \text{if } |\Delta x| \geq |\Delta y| \text{ and } \Delta x < 0 \\ 90^0 & \text{if } |\Delta x| < |\Delta y| \text{ and } \Delta y \geq 0 \\ -90^0 & \text{if } |\Delta x| < |\Delta y| \text{ and } \Delta y < 0 \end{cases}$$

where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. With these initial and final conditions, the robot starts moving and let at time t, it is at $[x(t), y(t), \theta(t)]$. At this time t, angular velocity, $\omega(t)$, is along $\theta_2 - \theta(t)$ and $|\omega(t)| \leq \omega_{\max}$. Linear velocity, $v(t)$, is chosen to be $min(dist(x(t), \delta X), dist(y(t), \delta Y), v_{\max})$. δX is the set of boundary points of the blocks q_1 and q_2 along the x-axis; similar definition for δY along y-axis as well. $dist(a, A) = inf\{\|x - a\|_2 \mid x \in A\}$ is a function that gives the minimum distance from a to the set A. These choices of inputs for the dynamics (6.5) ensure that the continuous trajectory never leaves the blocks q_1 and q_2. The continuous trajectory being confined within blocks q_1 and q_2, and the properties of the map T ensure that the continuous trajectory satisfies the temporal specification.

6.6 Simulations

We consider example 6.2.2 in a dynamic environment to test our approach. Noting that the choice of t_0 does not matter under this setting, we consider $t_0 = 0$. The dynamic environment contains a moving obstacle and the door to region R_2 is closed for the time interval $[0, 8\Delta\tau]$. The dynamic behaviors are incorporated in formulating the FTS ($\mathcal{E}$) and the Büchi automata ($\mathcal{E}'$). The reachability of the robot is considered to be the same as what given in Fig. 6.4. I_1 and I_2 in example 6.2.2 are considered to be $[14\Delta\tau, 17\Delta\tau]$ and $[16\Delta\tau, 21\Delta\tau]$ respectively. The discrete path is obtained by solving problem 6.5.14 using Dijkstra's algorithm [27]. The continuous path is generated by the controller described in section 6.5.3 with $\omega_{\max} = \pi/4\, rad/s$, $v_{\max} = 0.4\ m/s$ and the cellsize (Fig. 6.5) is $1 \times 1\ m^2$. The initial configuration of the robot is assumed to be $[x_0, y_0, \theta_0] = [8.5, 1.5, 135^o]$. The projected (in workspace) trajectories, both discrete and continuous, are shown in Fig. 6.5. The continuous trajectory in spacetime is shown in Fig. 6.6. For this example we considered total planning time upto $25\Delta\tau$ and hence the FTS $\mathcal{E}$ has 1600 states and the automata for the LTL specification consists of 128 states.

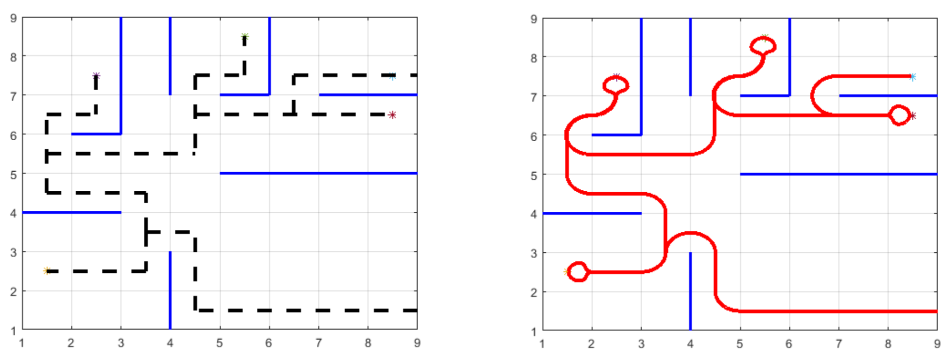


Figure 6.5: Projected continuous and discrete trajectories. (Dashed black line is the discrete path and the red curve is the corresponding continuous trajectory.)

6.7 Conclusion

In this chapter, we presented a method to generate continuous trajectories of a robot in a dynamic environment subject to some bounded time temporal logic specifications and optimization objective. We extended the rules of LTL to incor-

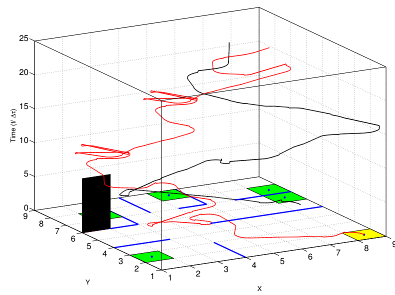


Figure 6.6: Continuous trajectory in spacetime. (Red curve is the trajectory generated for the robot and the black curve is the obstacle trajectory.)

porate timing constraints and we also projected the planning problem into a higher dimensional space to formally generate a path. minimize the task completion time but one can consider other objectives under the same framework. One possible way to find control inputs, for the continuous system (6.5), is proposed to make the robot follow the discrete optimal path obtained by solving problem 6.5.14. In this chapter we considered task of finite duration, however, task with infinite duration can also be formulated using the proposed framework. The generated continuous trajectory has high curvature at some points, solely because no constraint on the curvature of the trajectory was imposed while generating the continuous trajectory.

As a future direction, one might consider this framework for multi-robot planning probelms or planning in a dynamic environment to incrementally improve the solution while satisfying the timing constraints.

Chapter 7: Part III

Event-Triggered Controller Synthesis for Dynamical Systems with Temporal Logic Constraints

In the previous chapter we discussed controller synthesis under temporal logic constraints. The synthesized controller was feedback in nature and hence requires continuous sensor measurements to compute the control. In this chapter, we propose an event-triggered control framework for dynamical systems with temporal logical constraints. Event-triggered control methodologies have proven to be very efficient in reducing sensing, communication and computation costs. When a continuous feedback control is replaced with an event-triggered strategy, the corresponding state trajectories also differ. In a system with logical constraints, such small deviation in the trajectory might lead to unsatisfiability of the logical constraints. In this chapter, we develop an approach where we ensure that the event-triggered state trajectory is confined within an ϵ tube of the ideal trajectory associated with the continuous state feedback. At the same time, we will ensure satisfiability of the logical constraints as well.

7.1 Introduction

Present control systems are typically a large network of heterogeneous components sharing some common resources and information, and with coordinated cooperation, they aim to achieve desired performance. These kinds of highly complex systems are ubiquitous in cyber-physical-systems (CPS), and also referred as networked-CPS. In many CPS, the controller synthesis is subjected to many logical constraints that arise due to presence of logical variables and reasoning among the subsystems. Recent studies on controller synthesis with linear temporal logic (LTL) have paved a way to design controllers for large complex systems with safety, synchronisation, and other logical constraints [4, 16, 52].

Novel formulations and efficient computational approaches have been proposed to mathematically formulate specifications such as trajectory sequencing, synchronization etc. Temporal logics such as linear temporal logic (LTL), computational tree logic (CTL), developed for model checking, have been widely accepted by the robotics community for the purpose of motion planning [36], [92]. Development of sophisticated model checking tools such as SPIN and NuSMV made it easier to synthesize controllers for such systems. As an alternative approach controller synthesis has been done using mixed integer linear programming [16].

Another challenge for the large connected CPS is that computation of the control law requires continuous sensing (often times distributed) and transmitting the sensed signals to the controllers. Consequently, the performance of such systems is generally determined by the availability of sensing power, bandwidth for continuous transmission and resources for fast computation. Therefore it will be beneficial if the same (with little tolerance) performance can be achieved with lesser intensive sensing and computing tasks.

To circumvent the problem of limited communication bandwidth or computing resources or sensing capability, researchers have developed techniques that require intermittent communications only at certain discrete time instances to perform the same task with minor performance degradation. These control methodologies are known in many forms e.g. event-triggered, self-triggered or periodic control [43], [18]. These control strategies do not require the state information $x(t)$ for all time t, rather they sample $x(t)$ intermittently depending on the systems' performance criterion [66], [65]. These techniques have proven to be efficient for large scale interconnected systems to reduce communication and sensing operations.

In this work, we study the temporal logic based controller synthesis problem in an event-triggered framework. We consider a controller synthesis problem for a given control affine nonlinear system and the objective is to design an event triggered controller for that system with logical constraints. We assume the logical constraints can be represented using temporal logic and its propositional calculus.

Existing literature results show that the trajectory of an event triggered system deviates from the nominal system as a consequence of limited communication [65], [65]. Although, the continuous feedback system satisfies the logical constraints, now with an event triggered controller we have no guarantee that the logical constraint over the event-triggered trajectory will be satisfied as well.

We show that suitably modifying the given logical constraints, and creating stricter constraints will make the event-triggered trajectory satisfy the original logical constraint provided we can synthesize a continuous controller to satisfy the stricter constraint. The stricter logical constraint is often times known as robust logical constraint [37] since any perturbed trajectory (within some ϵ bound) will still satisfy the constraint. We adopt this notion of robustness in this chapter. In the next stage, we design an event-triggered controller ensuring that the event-triggered controller confines the trajectory within an ϵ-tube around the actual trajectory. Further, we show the effects of delay (in transmitting the measurement to the controller) on the performance. The analysis shows that if the delay is bounded by a certain quantity, which depends on the physical parameters of the plant and the controller, then the delayed system will be able to perform similar to the delay-free system without further modification in design.

In Section 7.2, we formally describe the problem and our two-step approach towards the problem. Section 7.3 provides preliminary background on the temporal logic and construction of ϵ-robust logic formulae. We design an event triggered controller for this problem in Section 7.4 and study the effects of delay on such an event triggered controller. Finally, we illustrate the application of our framework using two examples in Section 7.5.

7.2 Problem formulation

Let us consider the input-affine nonlinear state space model as given in (7.1).

$$\dot{x} = f_0(t,x) + \sum_{i=1}^{m} f_i(t,x) \cdot u_i \tag{7.1}$$
$$x(t_0) = x_0.$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, $u_i(t) \in \mathbb{R}$ is the i-th control input. The trajectory of the dynamical system starting at t_0 under application of some control $u = [u^1, \cdots, u^m]$ is denoted as $x^{u,x_0}[t_0]$. Similarly the trajectory starting from any arbitrary point (t,x) is represented as $x^{u,x}[t]$. The objective of this chapter is to design an event-triggered controller $u(\cdot)$ that ensures satisfiability of a temporal logical constraints (φ).

By $x^{u,x_0}[t_0] \models \varphi$ we denote that the trajectory of the dynamics (7.1) under input u satisfies the logical constraint φ. Similarly, $x^{u,x_0}[t_0] \not\models \varphi$ denotes that the trajectory does not satisfy the logical constraint. In this chapter we focus on the real time linear temporal logics [36], [82] which have been proven to be very effective for expressing logical constraints in dynamical systems [16]. In the following, we formally pose the problem that we aim to solve in this chapter.

Problem 7.2.1. *Given an input-affine dynamics (7.1) and a logical constraint (φ) on the trajectory of the system, design an event-triggered framework to generate the* *control* $u(t)$ *such that the event-triggered trajectory satisfies* φ*, i.e.*

$$\begin{aligned} \textit{find} \quad & u \\ \textit{subject to} \quad & u \in \mathcal{U}^e \\ & \dot{x} = f_0(t, x) + \sum_{i=1}^{m} f_i(t, x) \cdot u_i \\ & x^{u, x_0}[t_0] \models \varphi \end{aligned} \tag{7.2}$$

where $\mathcal{U}^e$ denotes the set of event based control strategies. In this chapter we do not impose any restriction on the event-triggered framework other than the exclusion of Zeno behavior [2]. We provide sufficient conditions which, if satisfied, ensure that the trajectory of the designed event-triggered system will satisfy the logical constraint φ.

While temporal logic can express various types of logical constraints (see Section 7.3), it comes with a cost that verifying whether a trajectory satisfies the logical constraints is PSPACE-complete [82]. Therefore synthesis of a controller is a hard problem in its own right, and synthesis of an event-based controller is harder for obvious reasons. However, there are some proposed techniques which can generate a (feedback) controller that satisfies the temporal logic constraints, see for example [15], [22].

Therefore, we divide the original problem into two subproblems: Problem 7.2.2 and Problem 7.2.3.

Problem 7.2.2. *Given the dynamics (7.1), design a feedback controller* $u_i(t) = \gamma_i(t, x(t))$ *such that* $x^{\gamma,x_0}[t_0] \models \varphi^\epsilon$.

where φ^ϵ is another logical constraint derived from φ. φ^ϵ is a stricter constraint than φ in the sense that $x^{\gamma,x_0}[t_0] \models \varphi^\epsilon$ implies $\xi[t_0] \models \varphi^\epsilon$ for all piecewise continuous curves $\xi(\cdot) : [t_0, +\infty) \to \mathbb{R}^n$ such that $\sup_{t\in[t_0,+\infty)} \|x(t) - \xi(t)\| \leq \epsilon$. In the following sections we will explicitly explain how φ^ϵ is related to φ for a given $\epsilon \geq 0$.

Problem 7.2.3. *For all* $\epsilon > 0$*, given the dynamics (7.1) and a feedback control* $\gamma(t, x(t))$*, design an event-triggered controller* $\gamma^e(t, x(\tau_k))$ *such that the trajectory of the event-triggered system* $x^{\gamma^e,x_0}[t_0]$ *remains within an* ϵ *neighborhood of the ideal trajectory associated with the feedback closed-loop system. As* $\epsilon \to 0$, $\gamma^e(t, \cdot) \to \gamma(t, \cdot)$ *pointwise* $\forall t$.

Therefore, in this two step approach, we first design a feedback controller for satisfying the ϵ-strict constraint φ^ϵ (for some $\epsilon > 0$). In the next stage we use an event-triggering mechanism which provides sufficient condition(s) for ensuring that the trajectory of the event-triggered system will be in an ϵ neighborhood of the actual feedback trajectory pointwise, i.e. $\|x_e(t) - x(t)\| \leq \epsilon$ (x_e is the event-triggered trajectory and x is the ideal feedback trajectory) for all t. In Figure 7.1, we present our schematic for event-triggered controller synthesis using the proposed two-step approach. From this point onward, we will suppress the control and initial state in denoting a trajectory when these are apparent from the context i.e. we will represent $x^{u,x_0}[t_0]$ as $x[t_0]$ etc.

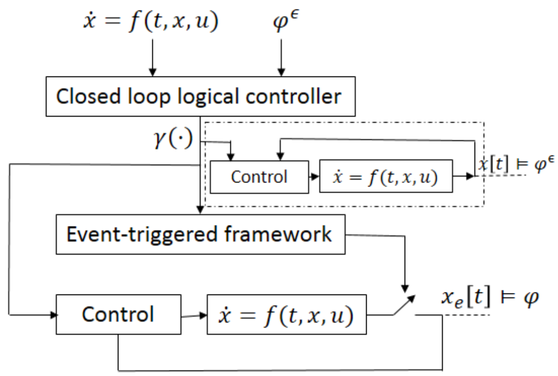


Figure 7.1: Schematic of two-step event-triggered controller synthesis with logical constraints

7.3 Propositional Temporal Logic

Like other families of propositional logic, temporal logic over the reals also requires a set of propositional variables $\Pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$. Associated with each propositional variable π_i, there is a labelling function $\mathcal{L}_i : \mathcal{X} \to \{0,1\}$ which denotes whether the proposition π_i is true at some point in $\mathcal{X}$. Therefore, Π divides $\mathcal{X}$ into subsets and assigns π_i with each of the subsets. A formula of a propositional logic is defined over a Boolean signal and in our case, $\mathcal{L}_i(\cdot)$ maps the $\mathbb{R}^n$ valued signal $(x(t))$ to a Boolean signal. For example, π_1 could be associated with the ball of radius 1 at the origin of $\mathcal{X} = \mathbb{R}^2$. Then $\mathcal{L}_1(s) = 1$ for all $x \in B_0(1)$ and 0 otherwise, where $B_x(\delta) = \{y \in \mathcal{X} \mid \|y - x\|_2 \leq \delta\}$ is a ball of radius δ centered at x. Note that it is not necessary that the regions associated with π_i are non-overlapping. We will use the notation $\pi_i \cong \mathcal{X}_i(\subseteq \mathcal{X})$ to denote $\mathcal{L}_i(x) = 1$ for all $x \in \mathcal{X}_i$ and $\mathcal{L}_i(x) = 0$ for all $x \in \mathcal{X} \setminus \mathcal{X}_i$ (basically the indicator function of the set $\mathcal{X}_i$). At this point, it should be noted that any algebraic constraint on x of the form $G(x) \leq 0$ could be associated with a proposition π such that $\pi \cong \{x \in \mathcal{X} \mid G(x) \leq 0\}$.

However, the power of temporal logic is beyond capturing these algebraic constraints. Let us first provide an informal overview of the capability of the logic and then formally state the syntax and semantics of the logic. The RTL (Temporal Logic over Reals) [83] formulae are built on the propositional variables Π with the use of usual logical operators $\neg$ (negation), $\vee$ (conjunction) and $\wedge$ (disjunction), and some special temporal operators e.g. $\mathbf{U}$ (until), $\Diamond$ (eventually), $\Box$ (always) and other operators that could be derived from the mentioned operators. For example, the formula $\Diamond\Box\pi$ (read as "Eventually Always in π") where $\pi \cong \mathcal{X}_i$, when satisfied by a trajectory of the dynamics (7.1), means that eventually the trajectory enter the region $\mathcal{X}_i$ and stay there for all future times. Similarly, $(\neg\pi_1 \wedge \neg\pi_2 \wedge \neg\pi_3)\mathbf{U}\pi_4$ states the rule that region $\mathcal{X}_4$ must be reached while avoiding regions $\mathcal{X}_i$ for $i = 1, 2, 3$ ($\pi_j \cong \mathcal{X}_j$). Although the satisfaction of the formula tells us that the state trajectory will reach $\mathcal{X}_4$, it does not provide any interval of time within which it will reach the destination. This limitation can easily be circumvented by the traditional augmentation of a new state $x_{n+1} = t$.

Adding an extra equation $\dot{x}_{n+1} = 1$ with $x_{n+1}(t_0) = t_0$ in the dynamics (7.1), we can pose time dependent constraints as well. In this case, the augmented space is $\mathcal{X} \times [t_0, T)$ (T could be $+\infty$ for an infinite horizon problem). The formula $\Box(\pi_1 \vee \pi_2)$ where $\pi_1 \cong \{(x, t) \mid x \in \mathcal{X}, t_0 \leq t < 6\}$ and $\pi_2 \cong \{(x, t) \mid x \in \mathcal{X}_i, 5 \leq t\}$ requires the trajectory to enter the region $\mathcal{X}_2$ no earlier than $t = 5$ and the trajectory should remain within $\mathcal{X}_2$ for all $t \in [6, \infty)$. The formula also mentions that the trajectory will be in $\mathcal{X}_1$ when it is not in $\mathcal{X}_2$. Therefore, with this state-space augmentation, all the properties related to the timing aspect of a trajectory of the system (7.1) can be expressed.

Definition 7.3.1. *The syntax of RTL formulas are defined according to the following grammar rules:*

$$\phi ::= \top \mid \pi \mid \neg\phi \mid \phi \vee \phi \mid \phi \mathbf{U} \phi \mid \phi \mathbf{R} \phi$$

where $\pi \in \Pi$, $\top$ and $\perp (= \neg\top)$ are the Boolean constants `true` and `false` respectively. $\mathbf{R}$ symbolizes the *Release* operator. Other temporal logic operators can be represented using the grammar in definition 7.3.1 e.g. eventually ($\Diamond\varphi = \top\mathbf{U}\varphi$), always ($\Box\varphi = \neg\Diamond(\neg\varphi)$) etc.

If $x[t_0]$ denotes a trajectory starting at time t_0, the semantics of the grammar in Definition 7.3.1 is given as follows:

Definition 7.3.2. *The semantics of any formula ϕ over the trajectory $x[t_0]$ is recursively defined as:*

$x[t_0] \models \pi$ *iff* $\mathcal{L}_\pi(x(t_0)) = 1$

$x[t_0] \models \neg\pi$ *iff* $\mathcal{L}_\pi(x(t_0)) = 0$

$x[t_0] \models \phi_1 \vee \phi_2$ *iff* $x[t_0] \models \phi_1$ *or* $x[t_0] \models \phi_2$

$x[t_0] \models \phi_1 \wedge \phi_2$ *iff* $x[t_0] \models \phi_1$ *and* $x[t_0] \models \phi_2$

$x[t_0] \models \phi_1 \mathbf{U} \phi_2$ *iff* $\exists s \geq t_0$ *s.t.* $x[s] \models \phi_2$ *and* $\forall\ t_0 \leq s' < s,\ x[s'] \models \phi_1$.

$x[t_0] \models \phi_1 \mathbf{R} \phi_2$ *iff* $\forall s \geq t_0$ $x[s] \models \phi_2$ *or* $\exists s'$ *s.t.* $t_0 \leq s' < s,\ x[s'] \models \phi_1$.

More details on RTL grammar and semantics can be found in [82], [36].

7.3.1 Construction of ϵ-Robust Formula

As described in Section 7.2, the motivation behind constructing an ϵ-robust formula φ^ϵ is that any trajectory satisfying the stricter formula φ^ϵ is robust in the sense that any perturbed trajectory with less than ϵ perturbation will also satisfy the original constraint φ.

The idea is as follows: if the trajectory needs to visit a region $\pi_i(\cong \mathcal{X}_i)$, then we push the boundary of $\mathcal{X}_i$ inwards by amount ϵ and denote this new set (and proposition) by $\mathcal{X}_i^\epsilon$ (π_i^ϵ). Similarly if the trajectory needs to avoid some region $\pi_j(\cong \mathcal{X}_j)$ then the boundary of $\mathcal{X}_j$ is expanded outwards by an amount ϵ.

The RTL syntax presented in Definition (7.3.1) is in negative normal form (NNF) [21], and this enables us to detect which regions must be avoided by noting the presence of the negation ($\neg$) operator immediately before the corresponding propositions π_i.

Note that, by our definition $\pi_i \cong \mathcal{X}_i$ and $\neg\pi_i \cong \mathcal{X} \setminus \mathcal{X}_i$.

Definition 7.3.3. *In a given metric space $(\mathcal{X}, \rho)$ the open ball centered at $x \in \mathcal{X}$ of radius r is defined as $B_x(r) = \{y \in \mathcal{X} \mid \rho(x, y) < r\}$. Let $\epsilon > 0$ be a given parameter, then the ϵ-contraction of the set $\mathcal{Y} \subseteq \mathcal{X}$ is denoted by $\mathcal{Y}^\epsilon = \{y \in \mathcal{Y} \mid B_y(\epsilon) \subseteq \mathcal{Y}\}$. Similarly, the ϵ-expansion of the set is denoted by $\mathcal{Y}^{-\epsilon} = \{x \in \mathcal{X} \mid \exists y \in \mathcal{Y}, x \in B_y(\epsilon)\}$.*

For a RTL formula φ, the ϵ-robust formula is constructed as follows:

1) replace each $\pi_i(\cong \mathcal{X}_i)$, which is not preceded by any negation, by π_i^ϵ where $\pi_i^\epsilon \cong \mathcal{X}_i^\epsilon$.

2) any π_j that is preceded by a negation ($\neg$) should be replaced by $\pi_j^{-\epsilon} \cong \mathcal{X}_j^{-\epsilon} \cup (\mathcal{X} \setminus \mathcal{X}^\epsilon)$. When $\mathcal{X} = \mathcal{X}^\epsilon$ for all finite $\epsilon > 0$ (e.g. $\mathcal{X} = \mathbb{R}^n$), $\pi_j^{-\epsilon} \cong \mathcal{X}_j^{-\epsilon}$.

In a similar way one can define ϵ-robust formulas over the space $\mathcal{X} \times [t_0, T)$. However, for this chapter we will restrict ourselves to the robustness only in $\mathcal{X}$ space.

Proposition 7.3.4. *Let the trajectory $x[t_0]$ satisfies the RTL formula φ^ϵ for some $\epsilon > 0$. Then for all $\delta \leq \epsilon$ and for any curve $y(\cdot) : [t_0, T) \to \mathcal{X}$ such that $\sup_t \rho(x(t), y(t)) \leq \delta$, $y[t_0] \models \varphi$.*

The above proposition can be proved inductively; interested readers may see [36] for a proof.

Remark 7.3.5. *In order to construct a π_i^ϵ from π_i, $\mathcal{X}_i$ must have a non-empty interior.*

Let us define the radius of a set $\mathcal{X}_i$ in the following way $r(\mathcal{X}_i) = \sup\{r \mid \exists x \in \mathcal{X}_i, B_x(r) \subseteq \mathcal{X}_i\}$. Therefore for each π_i, π_i^ϵ is well defined for $\epsilon \leq r(\mathcal{X}_i)$, for $\epsilon > r(\mathcal{X}_i)$, $\pi_i^\epsilon \cong \emptyset$. For the subsequent section we will implicitly assume that $\min_i\{r(\mathcal{X}_i)\} > 0$ and moreover, each of these sets, $\mathcal{X}_i$ is a polyhedron.

Therefore, to solve problem 7.2.2, our goal will be to design a controller that satisfies the robust RTL formula φ^ϵ. The power of using these temporal logic formulae is that each formula can be represented by an equivalent (Büchi) automaton. Satisfaction of an RTL formula is equivalent to finding a path from the initial state to one of the accepting states of the automaton. The construction of such automata can be done automatically using the available tools SPIN and NuSMV [46], [20]. The dynamics (7.1) can be represented as a finite transition system (FTS), where the transitions are performed by selecting the control u. Thus, the dynamics constraint and logical constraint in (7.2) can jointly be represented by forming a product of the automata and the FTS. More detail on such construction and controller synthesis can be found in our earlier work [67]. In this chapter, we spare the details on controller synthesis for temporal logic task, however, interested reader may see, for example, [62], [22]. Also, we dedicate the next chapter on designing a feedback (and subsequently event-triggered) controller for temporal logic tasks.

7.4 Event Triggered controller synthesis

In the previous section it is presented how the control inputs can be generated for the dynamics (7.1) so that the trajectory of the system satisfies the logical constraint φ^{ϵ} for some $\epsilon > 0$. This section will focus on designing an event-triggered controller that will replace the feedback controller designed to satisfy φ^{ϵ} in such a way that the trajectory of the event-triggered system will remain within ϵ distance of the ideal feedback trajectory.

Let us denote the controller $u(t) = \gamma(x(t))$ that achieves the satisfaction of φ^{ϵ}. The closed-loop dynamics are:

$$\begin{aligned} \dot{x} &= f_0(t,x) + \sum_{i=1}^{m} f_i(t,x)\gamma_i(x) \\ x(t_0) &= x_0 \end{aligned} \tag{7.3}$$

Let $x[t_0]$ denote the trajectory of the above closed loop system. We make the following assumptions on the system (7.3).

Assumptions 7.4.1. ***(A1)*** *$\gamma_i(\cdot)$ and for all t, $f_i(t,\cdot)$ are Lipschitz functions with Lipschitz constants L_γ^i, L_f^i respectively, for all $i = 0, 1, 2, \cdots, m$.*

(A2) *For all $i = 1, 2\cdots, m$ and $\forall t$, $f_i(t,x)\gamma_i(x)$ and $f_0(t,x)$ are continuously differentiable functions w.r.t x with continuous first derivative.*

In event-triggered framework, the controller is designed to be:

$$\gamma_i^e(t) = \gamma_i(x(t_k)) \quad \forall t \in [t_k, t_{k+1}) \tag{7.4}$$

where t_ks are the event-triggering times. An event-generator needs to be designed that will produce the t_k in certain way that is explained in the following.

Let us denote the event-triggered closed loop system as $x_e(t)$ and the corresponding trajectory as $x_e[t_0]$. Thus,

$$\dot{x}_e = f_0(t, x_e) + \sum_{i=1}^{m} f_i(t, x_e)\gamma_i^e(x_e(t_k)) \tag{7.5}$$

$$x_e(t_0) = x_0. \tag{7.6}$$

Let us define the error $e(t) = x(t) - x_e(t)$. Note that the event-triggered controller $\gamma_i^e(t,\cdot)$ for all t is an approximation of the ideal feedback controller $\gamma_i(t,\cdot)$ by piecewise constant functions. Therefore, designing the γ_i at first makes the problem tractable for generating the event-triggered controller.

The dynamics of $e(t)$ is given as:

$$\dot{e} = F(t,x) - F(t,x_e) + \sum_{i=0}^{m} f_i(t,x_e)(\gamma_i^e(x_e(t_k)) - \gamma_i(x_e(t)))$$
$$e(t_0) = 0, \tag{7.7}$$

where $F(t,x) = f_0(t,x) + \sum_{i=1}^{m} f_i(t,x)\gamma_i(x)$

Assumption 7.4.2. *$y(t)$ is exponentially stable with*

$$\dot{y} = A(t)y \tag{7.8}$$

where $A(t) = \frac{\partial F(t,x)}{\partial x}\Big|_{x=x(t)}$ and $x(t)$ is the trajectory of (7.3).

We can write,

$$\dot{e} = A(t)e + \tilde{f}(t,x_e,e)e + \delta(t) \tag{7.9}$$

where $\tilde{f}(t,x_e,e)e = F(t,x) - F(t,x_e) - A(t)e$ and $\delta(t) = \sum_{i=0}^{m} f_i(t,x_e)(\gamma_i^e(x_e(t_k)) - \gamma_i(x_e(t)))$

Using Assumption 7.4.1 (A2),

$$F(t,x+h) = F(t,x) + \int_{s=0}^{1} dF(t,x+sh)dsh$$

where $dF(t,\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is a linear map which is the derivative of the map $F(t,\cdot)$:

$\mathbb{R}^n \to \mathbb{R}^n$. Selecting $h = x_e - x = -e$, we obtain:

$$F(t, x_e) = F(t, x) - \int_0^1 dF(t, x - se)dse.$$

Therefore,

$$\tilde{f}(t, x_e, e) = \int_{s=0}^1 dF(t, x_e(t) + (1-s)e(t))ds - A(t).$$

Also $\tilde{f}(t, x_e, 0) = 0$.

Since $\dot{y} = A(t)y$ is the linearization of the system $\dot{e} = A(t)e + \tilde{f}(t, x_e, e)e$ around $e = 0$, we can say that $\dot{e} = A(t)e + \tilde{f}(t, x_e, e)e$ is locally exponentially stable due to Assumption 7.4.2. As a consequence of the Lyapunov converse theorem [51, Theorem 4.14], we have a (local) quadratic Lyapunov function that satisfies:

$$c_1\|e\|^2 \leq V(t, e) \leq c_2\|e\|^2 \tag{7.10}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial e}(A(t) + \tilde{f}(t, x_e, e))e \leq -c_3\|e\|^2 \tag{7.11}$$

$$\left\|\frac{\partial V}{\partial e}\right\| \leq c_4\|e\| \tag{7.12}$$

Proposition 7.4.3. *For all t,*

$$\|e(t)\|_2 \leq \frac{c_4}{2c_1}\int_{t_0}^{t} e^{-(t-s)c_3/2c_2}\|\delta(s)\|_2 ds \tag{7.13}$$

Proof. A detailed proof of this can be found in Theorem 3.2.6. The proposition is due to the BIBO (bounded input bounded output) stability of an exponential stable system. □

Note that at each t_k, $\delta(t_k) = 0$. We can bound $\|\delta(t)\|$ to ensure a bound on $\|e(t)\|$.

Remark 7.4.4. *Without computing $x(t)$ real-time, $e(t)$ can be bounded by observing the signal $\delta(t)$ which depends only on x_e. Moreover, due to the Lipschitz assumptions on $f_i(t,\cdot)$ and $\gamma_i(\cdot)$, it is sufficient to only monitor the difference signal $x(t) - x(t_k)$.*

From (7.13), we have the sufficiency condition that $\sup_t \|\delta(t)\| \leq \epsilon_1 = \frac{c_1 c_3}{c_2 c_4}\epsilon$ ensures $\sup_t \|e(t)\| \leq \epsilon$.

We propose the following event-trigger function:

$$g(t) = \frac{c_1 c_3}{c_2 c_4}\epsilon - \|\delta(t)\|. \tag{7.14}$$

An event is generated whenever $g(t) \leq 0$ and the state value at that time $(x(t_k))$ is sent to the controller. The set of triggering times is denoted by $\mathcal{T} = \{t_1, t_2, \cdots, t_k, \cdots\}$ such that $g(t_k) = 0$ and $g(t) < 0$ otherwise.

There could be other event-triggering functions that can also ensure bounded error $e(t)$. In this chapter, we consider (7.14) to carry out the analysis further and to perform the simulations.

Lemma 7.4.5. *For all t,*

$$\|\delta(t)\| \leq \alpha \|x_e(t) - x_e(t_k)\|^2 + \beta(t)\|x_e(t) - x_e(t_k)\|. \tag{7.15}$$

for some $\alpha, \beta(t) > 0$. t_k is the latest triggering time at time t.

Proof. We have $\delta(t) = \sum_{i=1}^{m} f_i(t, x_e)(\gamma_i^e(x_e(t_k)) - \gamma_i(x_e(t)))$. By rearranging,

$$\delta(t) = \sum_{i=1}^{m} f_i(t, x_e(t_k))(\gamma_i^e(x_e(t_k)) - \gamma_i(x_e(t))) + \sum_{i=1}^{m} (f_i(t, x_e(t)) - f_i(t, x_e(t_k)))(\gamma_i^e(x_e(t_k)) - \gamma_i(x_e(t)))$$

Using the Lipschitz continuity assumption in Assumption 7.4.1, we can write:

$$\begin{aligned} \|\delta\| \leq & \|(x_e(t_k)) - (x_e(t))\| \sum_{i=1}^{m} L_\gamma^i \|f_i(t, x_e(t_k))\| + \\ & \|(x_e(t_k)) - (x_e(t))\|^2 \sum_{i=1}^{m} L_\gamma^i L_f^i. \end{aligned} \tag{7.16}$$

Now we define:

$\alpha = \sum_{i=1}^{m} L_\gamma^i L_f^i$ and $\beta(t) = \sum_{i=1}^{m} L_\gamma^i \|f_i(t, x_e(t_k))\|$. □

In [65] a different bound on δ was derived. There it was shown that:

$$\|\delta(t)\| \leq \kappa(t) \|(x_e(t_k)) - (x_e(t))\|. \tag{7.17}$$

$\kappa(t) = \max_i \{L_\gamma^i\} \sup_{x \in \Omega_\epsilon} \sum_{i=1}^{m} \|f_i(t, x)\|$. Where the trajectory $x(t)$ of (7.3) is bounded in the domain Ω_ϵ.

Comparing (7.17) with (7.15), we notice that the former is bounded linearly w.r.t. $\|(x_e(t_k)) - (x_e(t))\|$ whereas the later is bounded by a quadratic form of $\|(x_e(t_k)) - (x_e(t))\|$. In most of the practical applications $\epsilon \ll 1$ and hence $\|(x_e(t_k)) - (x_e(t))\|$ is required to keep smaller than ϵ (see Proposition 7.4.6). Therefore, $\|(x_e(t_k)) - (x_e(t))\|^2$ can be bounded by $\|(x_e(t_k)) - (x_e(t))\|$ with proper coefficient. Furthermore, the presence of sup operator over the whole domain Ω_ϵ in (7.17) implies that $\kappa(t) \geq \beta(t)$ (in general $\kappa(t) \gg \beta(t)$). Therefore, (7.15) could be a better approximation of $\|\delta(t)\|$.

Proposition 7.4.6 (sufficiency)**.** *For all t,*

$$\|x_e(t) - x_e(t_k)\| \leq \frac{\epsilon_1}{\epsilon_1 + \frac{\beta^2(t)}{4\alpha}} \frac{\beta(t)}{4\alpha}$$

ensures $\|e(t)\| \leq \epsilon$, for $\epsilon_1 = \frac{c_2 c_4}{c_1 c_3}\epsilon$.

Proof. From (7.15), $\alpha\|x_e(t) - x_e(t_k)\|^2 + \beta(t)\|x_e(t) - x_e(t_k)\| \leq \epsilon_1$ implies $\|\delta\| \leq \epsilon_1$.

Therefore,

$$\|x_e(t) - x_e(t_k)\| \leq \frac{\sqrt{\beta(t)^2 + 4\alpha\epsilon_1} - \beta(t)}{2\alpha}$$

implies $\|\delta\| \leq \epsilon_1$. One can verify that

$$\sqrt{\beta(t)^2 + 4\alpha\epsilon_1} \geq \beta(t) + \frac{2\alpha\beta(t)\epsilon_1}{(\beta(t)^2 + 4\alpha\epsilon_1)}.$$

Thus,

$$\|x_e(t) - x_e(t_k)\| \leq \frac{\beta(t)\epsilon_1}{(\beta(t)^2 + 4\alpha\epsilon_1)}$$

ensures $\|\delta\| \leq \epsilon_1 = \frac{c_2 c_4}{c_1 c_3}\epsilon$.

Proposition 7.4.3 ensures that $\|\delta\| \leq \epsilon_1 = \frac{c_2 c_4}{c_1 c_3}\epsilon$ implies $\|e(t)\| \leq \epsilon$. □

As a comparison, in [65], the sufficient condition equivalent to Proposition

7.4.6 was $\|x_e(t) - x_e(t_k)\| \leq \frac{\epsilon_1}{\kappa(t)}$. If

$$\epsilon_1 < \frac{\beta(\kappa - \beta)}{4\alpha}$$

the sufficiency condition in Proposition 7.4.6 is relaxed than its counterpart in [65].

The following lemma ensures that the proposed event-triggering mechanism excludes Zeno behavior.

Lemma 7.4.7. *The inter-trigger time $\tau_k = t_k - t_{k-1}$ is bounded from below, i.e. $\inf_k \tau_k \geq \alpha > 0$. This ensures that within a finite interval $[t_0, T)$ there will be a finite number of triggerings.*

The lemma can be proved following the approach of 3.3.1; we omit it due to space limitation.

Theorem 7.4.8 (Main Result). *If there exists $\epsilon > 0$ and controllers $\gamma_i(\cdot)$ such that the closed-loop trajectory $x[t_0] \models \varphi^\epsilon$, then the event triggered trajectory $x_e[t_0] \models \varphi$ where events are generated whenever $g(t) \leq 0$.*

The proof follows directly from Proposition (7.3.4) where $y[t_0] = x_e[t_0]$ and we have ensured $\sup_t \rho(x(t), x_e(t)) = \sup_t \|e(t)\| \leq \epsilon$.

7.4.1 Implication of Delays

In this section we study the scenario when the sampled state $x(t_k)$ arrives to the controller at time $t_k + \Delta_k$ where Δ_k is the delay in the channel at time t_k. The aim of this section is to find a bound on the the delays so that the proposed event-triggered strategy still ensures that $\|e(t)\| \leq \epsilon$ for all time t.

In order to study that we start with the sufficiency condition $\|x_e(t)-x_e(t_k)\| \leq h(\epsilon)$ which ensures $\|e(t)\| \leq \epsilon$. Here $h(\epsilon)$ is $\frac{\beta(t)\epsilon_1}{(\beta(t)^2+4\alpha\epsilon_1)}$ (or $\frac{\epsilon_1}{\kappa(t)}$ by [65]).

Let the delays at triggering times t_{k-1} and t_k be Δ_{k-1}, Δ_k. Thus for all $t \in [t_{k-1}+\Delta_{k-1}, t_k+\Delta_k)$ the requirement is

$$\|x_e(t) - x_e(t_{k-1})\| \leq h(\epsilon)$$

Now,

$$\begin{aligned}\|x_e(t) - x_e(t_{k-1})\| &\leq \int_{t_{k-1}}^{t} \|f_0(s,x) + \sum_{i=1}^{m} f_i(s,x)\gamma_i^e(x(t_{k-1}))\|ds \\ &\leq \int_{t_{k-1}}^{t} (l\|x_e(s) - x_e(t_{k-1})\| + \|p(s)\|)ds\end{aligned}$$

where $l = L_f^0 + \sum_{i=1}^m L_f^i \gamma_i^e(x(t_{k-1}))$ and $p(s) = f_0(s, x(t_{k-1})) + \sum_{i=1}^m f_i(s, x(t_{k-1}))\gamma_i^e(x(t_{k-1}))$.

Therefore,

$$\|x_e(t) - x_e(t_{k-1})\| \leq \left(\int_{t_{k-1}}^{t} \|p(s)\|ds\right) e^{l(t-t_{k-1})}$$

Let $T_k = \inf_t\{t > t_{k-1} | \left(\int_{t_{k-1}}^{t} \|p(s)\|ds\right) e^{l(t-t_{k-1})} = h(\epsilon)\}$.

Therefore we must have $T_k \geq t_k + \Delta_k$ and $t_k \geq t_{k-1} + \Delta_{k-1}$. Thus,

$$\Delta_k + \Delta_{k-1} \leq T_k - t_{k-1}$$

From the definition of T_k,

$$\left(\int_0^{T_k-t_{k-1}} \|p(s+t_{k-1})\|ds\right) e^{l(T_k-t_{k-1})} = h(\epsilon) = \bar{\epsilon} \tag{7.18}$$

It is trivial to verify the above equation has unique solution for $T_k - t_{k-1}$ whenever $\bar{\epsilon} \geq 0$, and let us denote this solution by $T_k - t_{k-1} = \tilde{w}(\bar{\epsilon})$ for some function $\tilde{w}$ which satisfies the differential equation:

$$\begin{aligned} \frac{d\tilde{w}(r)}{dr} &= \frac{1}{rl + e^{l\tilde{w}}\|p(\tilde{w}+t_{k-1})\|} \\ \tilde{w}(0) &= 0. \end{aligned} \tag{7.19}$$

When f_i does not depend explicitly on time, then $p(s) = p$ and for this special case,

$$\tilde{w}(\bar{\epsilon}) = \frac{1}{l} W(l\bar{\epsilon}/p) \tag{7.20}$$

where $W(\cdot)$ is the Lambert W function.

From (7.19) one can verify that for all $r > 0$, $\tilde{w}(r) > 0$. Moreover using comparison lemma [51, Lemma 3.4], one can show for (7.19) that

$$\tilde{w}(r) \geq \frac{1}{l} W(lr/p_m)$$

where $p_m = \sup \|p(\tilde{w} + t_{k-1})\|$. Using the concavity property of $W(\cdot)$ along with

$W(0) = 0$, for all $0 \leq r \leq r_m$

$$\tilde{w}(r) \geq \frac{W(lr_m/p_m)}{lr_m} r$$

.

Thus, for $\bar{\epsilon}_m \geq \bar{\epsilon} = h(\epsilon)$,

$$\sup_k \{\Delta_k + \Delta_{k-1}\} \leq \frac{W(l\bar{\epsilon}_m/p_m)}{l\bar{\epsilon}_m} h(\epsilon) \tag{7.21}$$

ensures that $\|e(t)\| \leq \epsilon$. Thus, (7.21) states the sufficient condition for delays under which the proposed event trigger mechanism will be able to ensure $\|e(t)\| \leq \epsilon$.

7.5 Examples and Simulations

7.5.1 Example 1

Let us consider the following nonlinear system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} \sin(x_1) \\ x_2 \end{bmatrix} + \begin{bmatrix} -x_2 \\ x_1 \end{bmatrix} u \tag{7.22}$$

where $x_1(0) = 0, x_2(0) = 1$. We consider the region $\pi_1 \cong \mathcal{X}_1 = B_0(0.1) \subset \mathbb{R}^2$. The constraint (requirement) is to guide the state of the system within $B_0(0.1)$ and keep the trajectory within that ball for all future times. The logical constraint is represented by $\Diamond\Box\pi_1$. To proceed, we consider the ϵ-robust formula π_1^ϵ with $\epsilon = 0.05$.

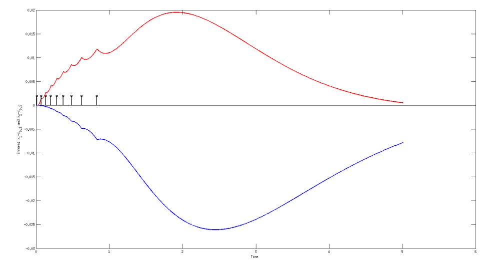


Figure 7.2: The red curve corresponds to the first component of the error $e = x_e - x$ and the blue one corresponds to the second component. The plot also shows the triggering instances. At each triggering instance, we notice corrective changes in the error components. Eventually the error components go to zero but is not shown here.

Therefore, $\mathcal{X}_1^\epsilon = B_0(0.05)$.

We chose the controller $u = -x_2$ that achieves the property that $x[0] \models \pi_1^\epsilon$ (use the Lyapunov function $V = x_1^2 + x_2^2$ to verify).

At this stage we need to design an event triggering mechanism that will ensure that the event triggered system will follow the actual trajectory $x[t_0]$. The event triggered system dynamics is given by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} \sin(x_1) \\ x_2 \end{bmatrix} - \begin{bmatrix} -x_2 \\ x_1 \end{bmatrix} x_2(t_k) \tag{7.23}$$

The initial condition is given as $(0, 1)$. One can check that $\bar{M}_i, L_\gamma^i$ defined in Proposition 7.4.6 have value 1. In Figure 7.2, the error signal is shown along with the triggering instances. From Figure 7.2 we note that only 9 samples are needed to achieve the task. This requires drastically reduced communication and sensing when compared to the continuous time feedback system. The trajectories of the ideal and even-triggered systems, and the ϵ-tube are shown in Figure 7.3.

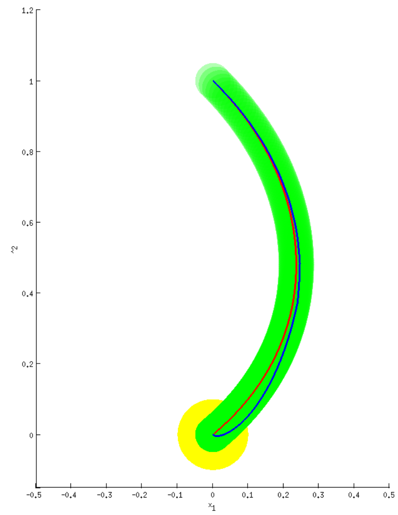


Figure 7.3: The red trajectory corresponds to continuous feedback control $x[0]$ and the blue trajectory corresponds to event triggered control. The green tube has a radius 0.05 and it shows that the proposed event triggered control ensures the trajectory is within that tube. The yellow circle corresponds to the given rule that the trajectory must be confined in there eventually.

7.5.2 Example 2

In this example we consider a robotic motion planning task with temporal sequencing and obstacle avoidance. The robot dynamics considered here is a unicycle model as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \tag{7.24}$$

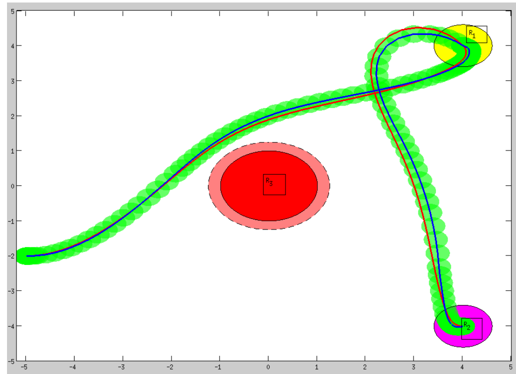


Figure 7.4: The closed loop trajectory is plotted using blue line and the event triggered trajectory with red line. The green tube around the nominal trajectory has radius 0.25. The initial position and orientation of the robot is $(-5, -2, 0)$

where $x, y \in \mathbb{R}^2$ is the physical position and $\theta \in [0, 360^o)$ is the heading angle. The task is given as follows:

$$\varphi \models \Diamond\pi_2 \wedge (\neg\pi_2 \mathbf{U} \pi_1) \wedge \Box\neg\pi_3 \tag{7.25}$$

where π_1, π_2 and π_3 corresponds to three circular regions as shown (denoted by R_1, R_2 and R_3) in Figure 7.4. The RTL formula defines the task of avoiding R_2 until reaching R_1 and eventually reaching R_2, and during the whole time the trajectory should avoid R_3. We adopt a potential function based approach [85] to generate the control laws for navigating the robot. As presented in previous sections, we expand and contract the appropriate regions while synthesizing the control. The region R_3 has been expanded by $\epsilon = 0.25$ (the dashed boundary around R_3 shows the expansion in Figure 7.4) while R_1 was contracted and R_2 has been both expanded and contracted (since both π_2 and $\neg\pi_2$ are present), however, we do not explicitly show them in Figure 7.4. In figure 7.5, we show the triggering instances for this

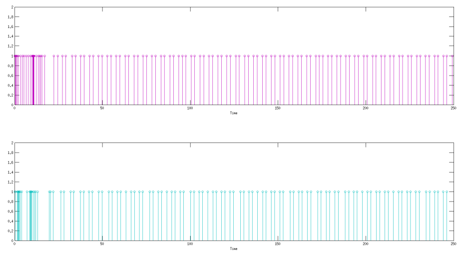


Figure 7.5: The upper graph shows the triggering instances for the trajectory from initial position to R_1. The lower graph shows the same for the other segment of the trajectory.

problem.

7.6 Conclusion

In this chapter, we have proposed a framework for integrating the event-triggered controller synthesis and logic based controller synthesis. Our solution is based on composition of two independent controller synthesis framework. It is also noteworthy that not any pair of logic-based-controller and event-trigger-controller has this unique composability property. We have derived an explicit event triggering mechanism to bound the trajectory within an ϵ-tube. With the notion of robust logic constraints, the resulting trajectory finally satisfies the logical constraint. Simulation results show the significant reduction in communicating the state value for updating the controller. This reduces the communication and computation costs.

Chapter 8: Event-triggered Feedback Control for Signal Temporal Logic Tasks

In this final chapter of the thesis we propose a framework for event-triggered control synthesis under signal temporal logic (STL) tasks. In the last chapter we adopted a compositional based approach which is retained here, i.e., replacing a continuous feedback controller with a intermittent event-triggered feedback. However, in this chapter the continuous feedback controller is not generated following an abstraction and cell decomposition based approach as presented in Chapter 6 and Chapter 7. The continuous feedback control law is designed, using the prescribed performance control technique, to satisfy the STL task; and in the sequel, the continuous feedback controller is replaced by an event-triggered controller. The event-triggering mechanism is based on a norm bound on the difference between the value of the current state and the value of the state at the last triggering instance. Simulations of a multi-agent system example quantitatively show the efficacy of using an event-triggered controller to reduce communication and computation efforts.

8.1 Introduction

Robot motion planning has traditionally been a challenging problem to the control community. Initially, the studies were primarily directed towards optimal navigation from an initial to a goal position while avoiding obstacles [50]. The focus, however, shifted over the years to integrate complex high-level task descriptions with the low-level dynamics of the robots. Consequently, temporal logics [71] were brought into the domain of motion planning to formally express and systematically address such complex behaviors in a generic way [15,36,52]. Temporal logics have a rich expressivity [10]; however, integrating the temporal logic descriptions with the dynamics of the robot is far from trivial. An appropriate abstraction of the dynamical system was needed in order to incorporate the high-level temporal logic task while not violating the physical constraints (dynamics) of the system. The derived methods are highly relying on automata theory and hence complexity problems arise as the size of the automata grows exponentially with the 'size of the task'. These complexity issues become even more severe when multi-agent systems are considered. Efficient techniques have been proposed in the context of temporal logic-based design. Nonetheless, integration of temporal logic tasks with a 'finer' abstraction of the dynamics is still computationally expensive. Moreover, temporal logic formulae expressing real-time constraints such as in signal temporal logic (STL) [70] are even more intricate to handle.

In this chapter, we take a different approach of integrating high level STL tasks with the dynamics of the robot. Instead of following the automata-based approach, we aim for a feedback control law that maximizes a robustness metric associated with the temporal logic formula. STL was introduced in [70], while space robustness [32] is the aforementioned robustness metric for STL. In a previous work [61], the authors leveraged ideas from prescribed performance control [14] to derive a feedback control law that satisfies the STL task under consideration. Prescribed performance control essentially allows to impose a transient behavior to the robustness metric, that, if properly designed, results in a satisfaction of the STL task. To implement such a feedback control law, we need continuous transmission of the measurements from the sensors to the controllers, and this can be a bottleneck in implementations. Mobile robots, e.g., operating in uncertain environments, have limited energy and bandwidth to transmit continuous measurements to the controllers. To alleviate this problem, we delve into synthesizing an event-triggered feedback control law in this chapter that will ensure the satisfaction of the STL task. Event-triggered control is an approach that has gained increasing attention recently [43, 91]. Extensions to multi-agent systems have appeared, e.g., in [28]. An overview of this topic in the setup of hybrid systems can be found in [79]. To the best of the knowledge, in [69], the authors made a first attempt in combining event-triggered control and temporal logic-based specifications. However, the synthesis of the control for the satisfaction of the temporal-logic formula is based on the automata theory and discretization methods, and hence it needs to deal with the high computational complexity issues associated with such methods.

The main contribution of this chapter is an event-triggering feedback control law for dynamic systems under STL task specifications. This event-triggered feedback control law is robust with respect to noise and the task satisfaction, while, at the same time, avoiding discretizations. We emphasize that the major difference compared with an existing work [61] is the event-based nature of the approach taken here due to which a drastic reduction in communication is observed.

The rest of the chapter is organized as follows: notations and preliminaries are provided in Section 8.2, the formal problem definition is given in Section 8.3, while Section 8.4 studies the control synthesis problem. Simulations are performed in Section 8.5, and finally we conclude the chapter in Section 8.6.

8.2 Notation and Preliminaries

Scalars are denoted by lowercase, non-bold letters x and column vectors are lowercase, bold letters $\boldsymbol{x}$. True and false are denoted by $\top$ and $\bot$ with $\mathbb{B} := \{\top, \bot\}$; $\mathbb{R}^n$ is the n-dimensional vector space over the real numbers $\mathbb{R}$. The natural, non-negative, and positive real numbers are $\mathbb{N}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$, respectively.

8.2.1 Signals and Systems

Let $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$, and $\boldsymbol{w} \in \mathcal{W} \subset \mathbb{R}^n$ be the state, input, and additive noise of a nonlinear system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{w}, \tag{8.1}$$

where $\mathcal{W}$ is a bounded set.

Assumptions 8.2.1. *The functions* $f : \mathbb{R}^n \to \mathbb{R}^n$ *and* $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ *are locally Lipschitz continuous, and* $g(\boldsymbol{x})g^T(\boldsymbol{x})$ *is positive definite and bounded for all* $\boldsymbol{x} \in \mathbb{R}^n$*, i.e.,* $\exists \lambda_{\min}, \lambda_{\max} \in \mathbb{R}_{>0}$ *with* $\lambda_{\min} < \lambda_{\max}$ *such that* $\lambda_{\min} < \|g(\boldsymbol{x})g^T(\boldsymbol{x})\| < \lambda_{\max}$ *for all* $\boldsymbol{x} \in \mathbb{R}^n$*.*

We next state two basic results regarding existence and uniqueness of solutions for the initial-value problem (IVP)

$$\dot{\boldsymbol{y}} := H(\boldsymbol{y}, t) \text{ with } \boldsymbol{y}(0) := \boldsymbol{y}_0 \in \Omega_{\boldsymbol{y}}, \tag{8.2}$$

where $\Omega_{\boldsymbol{y}} \in \mathbb{R}^{n+1}$ is a non-empty and open set and $H : \Omega_{\boldsymbol{y}} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{n+1}$. The signal $\boldsymbol{y} : \mathcal{J} \to \Omega_{\boldsymbol{y}}$ with $\mathcal{J} := [0, \tau_{max}) \subseteq \mathbb{R}_{\geq 0}$ is a maximal solution to (8.2) if and only if there is no other solution $\boldsymbol{y}' : [0, \tau') \to \Omega_{\boldsymbol{y}}$ to (8.2) with $\tau_{max} < \tau'$ such that $\boldsymbol{y}(t) = \boldsymbol{y}'(t)$ for all $t \in [0, \tau_{max})$. The maximal solution $\boldsymbol{y}$ is complete if and only if $\tau_{max} = \infty$.

Lemma 8.2.2. *[88, Theorem 54] Consider the IVP in (8.2). Assume that* $H : \Omega_{\boldsymbol{y}} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{n+1}$ *is: 1) locally Lipschitz continuous on* $\boldsymbol{y}$ *for each* $t \in \mathbb{R}_{\geq 0}$*; 2) piecewise continuous on* t *for each fixed* $\boldsymbol{y} \in \Omega_{\boldsymbol{y}}$*. Then, there exists a unique and maximal solution* $\boldsymbol{y} : \mathcal{J} \to \Omega_{\boldsymbol{y}}$ *with* $\mathcal{J} := [0, \tau_{max}) \subseteq \mathbb{R}_{\geq 0}$ *and* $\tau_{max} \in \mathbb{R}_{>0} \cup \{\infty\}$*.*

Lemma 8.2.3. *[88, Proposition C.3.6] Assume that the assumptions of Lemma 8.2.2 hold. For a maximal solution* $\boldsymbol{y} : \mathcal{J} \to \Omega_{\boldsymbol{y}}$ *with* $\tau_{max} < \infty$*, i.e.,* $\boldsymbol{y}$ *is not complete, and for any compact set* $\Omega'_{\boldsymbol{y}} \subset \Omega_{\boldsymbol{y}}$*, there exists* $t' \in \mathcal{J}$ *such that* $\boldsymbol{y}(t') \notin \Omega'_{\boldsymbol{y}}$*.*

8.2.2 Signal Temporal Logic (STL)

Signal temporal logic (STL) is a predicate logic based on continuous-time signals. STL consists of predicates μ that are obtained after evaluation of a predicate function $h : \mathbb{R}^n \to \mathbb{R}$ as follows

$$\mu := \begin{cases} \top \text{ if } h(\boldsymbol{x}) \geq 0 \\ \bot \text{ if } h(\boldsymbol{x}) < 0. \end{cases}$$

For instance, consider the predicate $\mu := (x \geq 1)$, which can be expressed by the predicate function $h(x) := x - 1$. Hence, h maps from $\mathbb{R}^n$ to $\mathbb{R}$, while μ maps from $\mathbb{R}^n$ to $\mathbb{B}$. Note that $\boldsymbol{x}$ in $h(\boldsymbol{x})$ is seen as a state measurement and not as a signal. In this chapter, we use $\boldsymbol{x}$ both to denote a state and a signal, i.e., a solution of (8.1) with $\boldsymbol{x}(0)$. It will, however, be clear from the context what $\boldsymbol{x}$ stands for.

If μ is a predicate, the STL syntax is given by

$$\phi \;::=\; \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \,\mathcal{U}_{[a,b]}\, \phi_2 \;,$$

where ϕ_1, ϕ_2 are STL formulas. The temporal until-operator $\mathcal{U}_{[a,b]}$ is time bounded within $[a, b]$ where $a, b \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ such that $a \leq b$. The semantics of STL are introduced in Definition 8.2.4 where the satisfaction relation $(\boldsymbol{x}, t) \models \phi$ denotes that the signal $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ satisfies ϕ at time t.

Definition 8.2.4. *[70, Definition 1] The STL semantics are recursively given by:*

$$
\begin{aligned}
(\boldsymbol{x}, t) \models \mu \quad & \Leftrightarrow \quad h(\boldsymbol{x}(t)) \geq 0 \\
(\boldsymbol{x}, t) \models \neg\mu \quad & \Leftrightarrow \quad \neg((\boldsymbol{x}, t) \models \mu) \\
(\boldsymbol{x}, t) \models \phi_1 \wedge \phi_2 \quad & \Leftrightarrow \quad (\boldsymbol{x}, t) \models \phi_1 \wedge (\boldsymbol{x}, t) \models \phi_2 \\
(\boldsymbol{x}, t) \models \phi_1 \, \mathcal{U}_{[a,b]} \, \phi_2 \quad & \Leftrightarrow \quad \exists t_1 \in [t+a, t+b] \ s.t. \ (\boldsymbol{x}, t_1) \models \phi_2 \\
& \qquad \wedge \forall t_2 \in [t, t_1], (\boldsymbol{x}, t_2) \models \phi_1
\end{aligned}
$$

Disjunction-, eventually-, and always-operator can be derived as $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$, $F_{[a,b]}\phi := \top \, \mathcal{U}_{[a,b]} \, \phi$, and $G_{[a,b]}\phi := \neg F_{[a,b]}\neg\phi$. Space robustness [32] are robust semantics for STL, which are given in Definition 8.2.5 and denoted by $\rho^{\phi}(\boldsymbol{x}, t)$. Space robustness determines how robustly a signal $\boldsymbol{x}$ satisfies the formula ϕ at time t. It holds that $(\boldsymbol{x}, t) \models \phi$ if $\rho^{\phi}(\boldsymbol{x}, t) > 0$ [37, Proposition 16].

Definition 8.2.5. *[32, Definition 3] The semantics of space robustness are recursively given by:*

$$
\begin{aligned}
\rho^{\mu}(\boldsymbol{x}, t) &:= h(\boldsymbol{x}(t)) \\
\rho^{\neg\phi}(\boldsymbol{x}, t) &:= -\rho^{\phi}(\boldsymbol{x}, t) \\
\rho^{\phi_1 \wedge \phi_2}(\boldsymbol{x}, t) &:= \min\left(\rho^{\phi_1}(\boldsymbol{x}, t), \rho^{\phi_2}(\boldsymbol{x}, t)\right)
\end{aligned}
$$

$$\rho^{\phi_1 \vee \phi_2}(\boldsymbol{x}, t) := \max\big(\rho^{\phi_1}(\boldsymbol{x}, t), \rho^{\phi_2}(\boldsymbol{x}, t)\big)$$

$$\rho^{\phi_1 \mathcal{U}_{[a,b]} \phi_2}(\boldsymbol{x}, t) := \max_{t_1 \in [t+a, t+b]} \Bigg(\min \Big(\rho^{\phi_2}(\boldsymbol{x}, t_1), \min_{t_2 \in [t, t_1)} \rho^{\phi_1}(\boldsymbol{x}, t_2) \Big) \Bigg)$$

$$\rho^{F_{[a,b]}\phi}(\boldsymbol{x}, t) := \max_{t_1 \in [t+a, t+b]} \rho^{\phi}(\boldsymbol{x}, t_1)$$

$$\rho^{G_{[a,b]}\phi}(\boldsymbol{x}, t) := \min_{t_1 \in [t+a, t+b]} \rho^{\phi}(\boldsymbol{x}, t_1).$$

We abuse the notation as $\rho^{\phi}(\boldsymbol{x}(t)) := \rho^{\phi}(\boldsymbol{x}, t)$ if t is not explicitly contained in $\rho^{\phi}(\boldsymbol{x}, t)$. For instance, $\rho^{\mu}(\boldsymbol{x}(t)) := \rho^{\mu}(\boldsymbol{x}, t) := h(\boldsymbol{x}(t))$ since $h(\boldsymbol{x}(t))$ does not contain t as an explicit parameter. However, t is explicitly contained in $\rho^{\phi}(\boldsymbol{x}, t)$ if temporal operators (eventually, always, or until) are contained in ϕ.

8.2.3 Event-Triggered Control

In contrast to continuous feedback control, event-triggered feedback control requires the state measurements intermittently while ensuring a performance arbitrarily close to the continuous feedback control. These controllers rely on an event generator that decides on the instances when the state measurements are sent to the controller. Thereby, the communication between the sensors and the controller can be reduced. Thus, in event-triggered control, the controller obtains new state information only at certain discrete time instances denoted by $t_1, t_2, \cdots, t_i, \cdots$.

Definition 8.2.6. *The information set, which is denoted by $\mathfrak{I}(t)$ and associated* *with an event-triggered controller, denotes the set of measurements available to the controller to compute its control value at time* t*. This information set* $\mathfrak{I}(t)$ *is updated at each triggering instance* t_i*.*

Let $N(t) := |\mathfrak{I}(t)|$ denote the number of state measurements available to the controller until time t, where $|\mathfrak{I}(t)|$ denotes the cardinality of $\mathfrak{I}(t)$. There are three possibilities for $\mathfrak{I}(t)$: first, the controller remembers all the past and the present measurements

$$\mathfrak{I}(t) := \{\boldsymbol{x}(t_i)\}_{i=1}^{N(t)}.$$

Second, the controller remembers only the present measurement i.e. $\mathfrak{I}(t) := \{\boldsymbol{x}(t_{N(t)})\}$; and third, the controller remembers the present and some of the past measurements.

Let us use $\hat{\tau}(t)$ to denote the latest time instance when a new state measurement was sent to the controller for any $t \in \mathbb{R}_{\geq 0}$. That is, if t_i, t_{i+1} are two successive triggering time instances, then $\hat{\tau}(t) := t_i$ for all $t \in [t_i, t_{i+1})$. Thus, $\hat{\tau}(t)$ is a piecewise constant function. For any given event-triggered controller synthesis problem, we need to answer two questions: first, how to construct the set $\mathfrak{I}(t)$, and second, how to use $\mathfrak{I}(t)$ to compute the control law. Another issue to consider is to guarantee the avoidance of Zeno behavior, i.e., the case of infinite switching in finite time. This will be explicitly shown in the design by guaranteeing a strictly positive minimum inter-triggering time.

8.2.4 Prescribed Performance Control (PPC)

Prescribed performance control (PPC) [14] constrains a generic error $\boldsymbol{e} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ to a user-designed funnel. For instance, consider $\boldsymbol{e}(t) := \begin{bmatrix} e_1(t) & \dots e_n(t) \end{bmatrix}^T := \boldsymbol{x}(t) - \boldsymbol{x}_d(t)$, where $\boldsymbol{x}_d$ is a desired trajectory. In order to prescribe transient and steady-state behavior to this error, let us define the performance function γ in Definition 8.2.7 as well as the transformation function S in Definition 8.2.8.

Definition 8.2.7. *[14] A performance function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ is a continuously differentiable, bounded, positive, and non-increasing function given by $\gamma(t) := (\gamma_0 - \gamma_\infty)\exp(-lt) + \gamma_\infty$ where $\gamma_0, \gamma_\infty \in \mathbb{R}_{>0}$ with $\gamma_0 \geq \gamma_\infty$ and $l \in \mathbb{R}_{\geq 0}$.*

Definition 8.2.8. *[14] A transformation function $S : (-1, M) \to \mathbb{R}$ with $M \in [0,1]$ is a smooth and strictly increasing function, hence admitting an inverse. In particular, let $S(\xi) := \ln\left(-\frac{\xi+1}{\xi-M}\right)$.*

Now assume that γ_i is a performance function in the sense of Definition 8.2.7. The task is to synthesize a continuous feedback control law such that each error e_i satisfies

$$-\gamma_i(t) < e_i(t) < M\gamma_i(t) \quad \forall t \in \mathbb{R}_{\geq 0}, \forall i \in \{1, \dots, n\} \tag{8.3}$$

given that $-\gamma_i(0) < e_i(0) < M\gamma_i(0)$; γ_i is a design parameter by which transient and steady-state behavior of e_i can be prescribed. Similar to M in the right inequality of (8.3), another constant could be added to the left inequality, which will, however, not be considered here. Note also that (8.3) is a constrained control problem with

n constraints subject to the dynamics in (8.1). Next, define the normalized error $\xi_i := \frac{e_i}{\gamma_i}$. Dividing (8.3) by γ_i and applying the transformation function S results in an unconstrained control problem $-\infty < S\big(\xi_i(t)\big) < \infty$ with the transformed error $\epsilon_i := S\big(\xi_i\big)$. If $\epsilon_i(t)$ is bounded for all $t \geq \mathbb{R}_{\geq 0}$, then e_i satisfies (8.3).

8.3 Problem Definition

In this chapter, we consider the following fragment of STL

$$\psi ::= \top \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2 \tag{8.4a}$$

$$\phi ::= G_{[a,b]}\psi \mid F_{[a,b]}\psi \tag{8.4b}$$

$$\theta^{s_1} ::= \bigwedge_{i=1}^{K} \phi_i \text{ with } b_k \leq a_{k+1},\ \forall k \in \{1, \ldots, K-1\} \tag{8.4c}$$

$$\theta^{s_2} ::= F_{[c_1,d_1]}\Big(\psi_1 \wedge F_{[c_2,d_2]}\big(\psi_2 \wedge F_{[c_3,d_3]}(\ldots \wedge \phi_K)\big)\Big) \tag{8.4d}$$

$$\theta ::= \theta^{s_1} \mid \theta^{s_2}, \tag{8.4e}$$

where μ is a predicate; ψ in (8.4b) and $\psi_1, \psi_2, \ldots$ in (8.4a) and (8.4d) are formulas of class ψ given in (8.4a), whereas ϕ_i with $i \in \{1, \ldots, K\}$ in (8.4c) and (8.4d) are formulas of class ϕ given in (8.4b) with time intervals $[a_i, b_i]$. Formulas of class θ given in (8.4e) either consist of (8.4c) or (8.4d). We refer to ψ as *non-temporal formulas*. Due to the previous discussion, we write $\rho^{\psi}\big(\boldsymbol{x}(t)\big) := \rho^{\psi}(\boldsymbol{x}, t)$ and sometimes even omit t resulting in $\rho^{\psi}(\boldsymbol{x})$. In contrast, ϕ and θ are referred to as *temporal formulas* due to the use of the always- and eventually-operators.

The control strategy that will be introduced in Section 8.4 requires three additional assumptions that will be explained in the sequel. First, for conjunctions of *non-temporal formulas* of class ψ given in (8.4a), e.g., $\psi := \psi_1 \wedge \psi_2$, we approximate the robust semantics from Definition 8.2.5, e.g., $\rho^{\psi_1 \wedge \psi_2}$, by a smooth function as follows.

Assumptions 8.3.1. *The robust semantics for a conjunction of* non-temporal formulas *of class ψ given in* (8.4a)*, i.e., $\rho^{\psi_1 \wedge \psi_2}(\boldsymbol{x}, t)$, are approximated by a smooth function as*

$$\rho^{\psi_1 \wedge \psi_2}(\boldsymbol{x}, t) \approx -\ln\big(\exp(-\rho^{\psi_1}(\boldsymbol{x}, t)) + \exp(-\rho^{\psi_2}(\boldsymbol{x}, t))\big).$$

From now on, when we write $\rho^{\psi}(\boldsymbol{x}, t)$, $\rho^{\phi}(\boldsymbol{x}, t)$, or $\rho^{\theta}(\boldsymbol{x}, t)$ for formulas of class ψ, ϕ, and θ, respectively, we mean the robust semantics including the smooth approximation in Assumption 8.3.1 unless stated otherwise. This approximation is an under-approximation of the robust semantics as remarked in [61], i.e., the property that $(\boldsymbol{x}, t) \models \theta$ if $\rho^{\theta}(\boldsymbol{x}, t) > 0$ is preserved. The next example illustrates the above and emphasizes that the smooth approximation is only used for conjunctions of *non-temporal formulas* ψ.

Example 8.3.2. *Assume the formula $\theta := F_{[5,15]}(\psi_1 \wedge \psi_2) \wedge G_{[20,30]}\psi_3$. Then the robust semantics at time $t := 0$ are $\rho^{\theta}(\boldsymbol{x}, 0) = \min\big(\max_{t\in[5,15]}(-\ln(\exp(-\rho^{\psi_1}(\boldsymbol{x}, t)) + \exp(-\rho^{\psi_2}(\boldsymbol{x}, t))), \min_{t\in[20,30]} \rho^{\psi_3}(\boldsymbol{x}, t)\big)$.*

Second, the next assumption restricts the class of ψ formulas given in (8.4a) that are contained in (8.4b) and (8.4e).

Assumptions 8.3.3. *Each formula of class* ψ *that is contained in* (8.4b) *and* (8.4e) *is: 1) such that* $\rho^{\psi}(\boldsymbol{x})$ *is concave; and 2) well-posed in the sense that* $(\boldsymbol{x}, 0) \models \psi$ *implies* $\|\boldsymbol{x}(0)\| \leq C < \infty$ *for some* $C \geq 0$.

Third, let the global optimum of $\rho^{\psi}(\boldsymbol{x})$ be

$$\rho^{\psi}_{opt} := \sup_{\boldsymbol{x} \in \mathbb{R}^n} \rho^{\psi}(\boldsymbol{x}) \tag{8.5}$$

with the function $\rho^{\psi}(\boldsymbol{x})$ being continuous and concave due to Assumption 8.3.1 and 8.3.3, which simplifies the calculation of ρ^{ψ}_{opt}. It holds that ϕ is feasible, i.e., $\exists \boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ s.t. $(\boldsymbol{x}, 0) \models \phi$, if $\rho^{\psi}_{opt} > 0$. To guarantee feasibility, we pose the next assumption.

Assumptions 8.3.4. *The maximum of* $\rho^{\psi}(\boldsymbol{x})$ *is s.t.* $\rho^{\psi}_{opt} > 0$.

If Assumption 8.3.4 does not hold, ϕ may not be feasible and a least violating solution can be found by setting $r \leq 0$; however, this case is not within the scope of this chapter. For further remarks regarding the aforementioned assumptions, we refer the reader to the article [61].

In [61], a continuous feedback control law was derived to satisfy formulas of class ϕ given in (8.4b). In this chapter, the focus is to derive an event-based feedback control law to satisfy ϕ. A hybrid control strategy, in the same vein as in [61], can then be used to satisfy formulae of class θ given in (8.4e). We now shortly summarize the main idea used to achieve $r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$, where $r \in \mathbb{R}_{>0}$ is a robustness measure and $\rho_{max} \in \mathbb{R}_{>0}$ with $r < \rho_{max}$ is a robustness delimiter. It then follows that $(\boldsymbol{x}, 0) \models \phi$ since $r > 0$; $r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$ is achieved by prescribing a temporal behavior to $\rho^{\psi}\big(\boldsymbol{x}(t)\big)$ through the design parameters γ and ρ_{max} as

$$-\gamma(t) + \rho_{max} < \rho^{\psi}\big(\boldsymbol{x}(t)\big) < \rho_{max}. \tag{8.6}$$

Note the use of $\rho^{\psi}\big(\boldsymbol{x}(t)\big)$ and not $\rho^{\phi}(\boldsymbol{x}, 0)$ itself. The connection between the non-temporal $\rho^{\psi}\big(\boldsymbol{x}(t)\big)$ and the temporal $\rho^{\phi}(\boldsymbol{x}, 0)$ is made by the choice of the performance function γ. The proposed solution in [61] consists of two steps. First, the control law $\boldsymbol{u}$ is designed such that (8.6) holds for all $t \in \mathbb{R}_{\geq 0}$. In a second step, γ is designed such that satisfaction of (8.6) for all $t \in \mathbb{R}_{\geq 0}$ implies $r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$. This second step results in selecting

$$t_* \in \begin{cases} a & \text{if } \phi = G_{[a,b]}\psi \\ [a, b] & \text{if } \phi = F_{[a,b]}\psi, \end{cases} \tag{8.7}$$

$$\rho_{max} \in \Big(\max\big(0, \rho^{\psi}(\boldsymbol{x}(0))\big), \rho^{\psi}_{opt} - \chi\Big] \tag{8.8}$$

$$r \in (0, \rho_{max}) \tag{8.9}$$

$$\gamma_0 \in \begin{cases} (\rho_{max} - \rho^{\psi}(\boldsymbol{x}(0)), \infty) & \text{if } t_* > 0 \\ (\rho_{max} - \rho^{\psi}(\boldsymbol{x}(0)), \rho_{max} - r] & \text{if } t_* = 0 \end{cases} \tag{8.10}$$

$$\gamma_\infty \in \Big(0, \min\big(\gamma_0, \rho_{max} - r\big)\Big] \tag{8.11}$$

$$l \in \begin{cases} \mathbb{R}_{\geq 0} & \text{if } -\gamma_0 + \rho_{max} \geq r \\ \frac{\ln\left(\frac{r + \gamma_\infty - \rho_{max}}{-(\gamma_0 - \gamma_\infty)}\right)}{-t_*} & \text{if } -\gamma_0 + \rho_{max} < r, \end{cases} \tag{8.12}$$

where $\chi > 0$ is a small constant that has to satisfy $\chi < \rho^{\psi}_{opt} - \max\big(0, \rho^{\psi}(\boldsymbol{x}(0))\big)$. Furthermore, it needs to hold that $\rho^{\psi}(\boldsymbol{x}(0)) > r$ if $t_* = 0$. This chapter, however, will only focus on the first step and derive an event-based feedback control law such that (8.6) holds for all $t \in \mathbb{R}_{\geq 0}$. Define now the one-dimensional error, the normalized error, and the transformed error as

$$e(\boldsymbol{x}) := \rho^{\psi}(\boldsymbol{x}) - \rho_{max} \tag{8.13}$$

$$\xi(\boldsymbol{x}, t) := \frac{e(\boldsymbol{x})}{\gamma(t)} \tag{8.14}$$

$$\epsilon(\boldsymbol{x}, t) := S\big(\xi(\boldsymbol{x}, t)\big) = \ln\Big(-\frac{\xi(\boldsymbol{x}, t) + 1}{\xi(\boldsymbol{x}, t)}\Big), \tag{8.15}$$

respectively. As a notational rule, when talking about the solution $\boldsymbol{x}$ of (8.1) at time t, we use $e(t)$, $\xi(t)$, and $\epsilon(t)$, while we use $e(\boldsymbol{x})$, $\xi(\boldsymbol{x}, t)$, and $\epsilon(\boldsymbol{x}, t)$ when we talk about $\boldsymbol{x}$ as a state. Equation (8.6) can now be written as $-\gamma(t) < e(t) < 0$, which resembles (8.3) by setting $M := 0$ and can further be written as $-1 < \xi(t) < 0$. Applying the function S results in $-\infty < \epsilon(t) < \infty$. If now $\epsilon(t)$ is bounded for all $t \geq 0$, then (8.6) holds for all for all $t \geq 0$. We remark that $\xi\big(\boldsymbol{x}(0), 0\big) \in \Omega_{\xi} := (-1, 0)$ needs to hold initally, which is ensured by the choice of γ_0. The formal problem definition is now given as follows.

Problem 8.3.5. *Consider the system in* (8.1) *and a STL formula of class* ϕ *given in* (8.4b). *Design an event-triggered feedback control law* $\boldsymbol{u}(\boldsymbol{x}, t)$ *such that* $0 < r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$, *i.e.,* $(\boldsymbol{x}, 0) \models \phi$.

8.4 Control Synthesis

In this section, the event-triggered feedback control law is derived in two steps. First, a continuous feedback control law $\boldsymbol{u}$ is derived such that $0 < r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$, which we will later alter to obtain an event-triggered control law $\hat{\boldsymbol{u}}$. We state the main result upfront in Theorem 8.4.1 which is proved in the subsequent section.

Theorem 8.4.1 (Main Result)**.** *The dynamical system (8.1), satisfying Assumption 8.2.1, along with the choice of PPC parameters as per equations* (8.8) *–* (8.12) *satisfies a STL formula ϕ of the form* (8.4b) *if Assumptions 8.3.1 – 8.3.4 are satisfied and if the event-triggered control law $\hat{\mathbf{u}}$ has the form*

$$\hat{\boldsymbol{u}}(t) := \boldsymbol{u}(\boldsymbol{x}(t_i), t) \qquad \forall t \in (t_i, t_{i+1}] \tag{8.16}$$

where the triggering instances $\{t_i\}$ are generated as:

$$\begin{aligned} t_0 &:= 0 \\ t_{i+1} &:= \inf\{t > t_i \mid \|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| > \delta_{\boldsymbol{x}}\} \quad i \geq 1, \end{aligned} \tag{8.17}$$

for some $\delta_{\boldsymbol{x}} > 0$ (obtained later in the chapter). The function $\mathbf{u}(\boldsymbol{x}, t)$ in (8.16) is chosen as

$$\boldsymbol{u}(\boldsymbol{x}, t) := -L(\boldsymbol{x})\Big(f(\boldsymbol{x}) - \frac{\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}}{\|\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}\|^2}(\xi\dot{\gamma} - k\epsilon)\Big) \tag{8.18}$$

where $k > 0$ *and* $L(\boldsymbol{x}) := g(\boldsymbol{x})^T\big(g(\boldsymbol{x})g(\boldsymbol{x})^T\big)^{-1}$.

The theorem is proved in several steps by using Theorems 8.4.2 and 8.4.3 that are proposed in the rest of this section. To start with, we propose a continuous feedback control law $\boldsymbol{u}$ to guarantee the satisfaction of the STL formula ϕ, and later we replace (approximately) the continuous feedback control law with an event-triggered version $\hat{\boldsymbol{u}}$, while still guaranteeing the satisfaction of the STL formula ϕ.

Theorem 8.4.2. *The dynamical system (8.1), satisfying Assumption 8.2.1, along with the choice of PPC parameters as per equations* (8.8) – (8.12) *satisfies a STL formula* ϕ *of the form* (8.4b) *if the continuous feedback control law* (8.18) *is applied and if Assumptions 8.3.1 – 8.3.4 are satisfied. It then holds that* $0 < r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$*, i.e.,* $(\boldsymbol{x}, 0) \models \phi$*, with all closed-loop signals being continuous and bounded.*

Proof. This proof follows the same line of proof as in [61, Theorem 1 and 2] and is hence only sketched due to space limitations. First, define the stacked vector $\boldsymbol{y} := \begin{bmatrix}\boldsymbol{x}^T & \xi\end{bmatrix}^T$ and the sets $\Omega_\xi := (-1, 0)$ and $\Omega_{\boldsymbol{x}} := \{\boldsymbol{x} \in \mathbb{R}^n | -1 < \xi(\boldsymbol{x}, 0) := \frac{\rho^{\psi}(\boldsymbol{x}) - \rho_{max}}{\gamma_0} < 0\}$. Note that $\xi(0) \in \Omega_\xi$ due to the choice of γ_0 and that hence $\boldsymbol{x}_0 \in \Omega_{\boldsymbol{x}}$. Furthermore, $L(\boldsymbol{x})$ is always well-defined due to Assumption 8.2.1. We remark already that $\|\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}\|$ is upper and lower (strictly positive) bounded for all $\xi \in \Omega_\xi$ and $\boldsymbol{x} \in \Omega_{\boldsymbol{x}}$, which will be shown in detail in the sequel of this proof. By the same arguments as in [61], it follows that the conditions in Lemma 8.2.2 hold. Consequently, for $\Omega_{\boldsymbol{y}} := \Omega_{\boldsymbol{x}} \times \Omega_\xi$, which is a bounded and open set as shown in [61], there exists a maximal solution $\boldsymbol{y} : \mathcal{J} \to \Omega_{\boldsymbol{y}}$ with $\mathcal{J} := [0, \tau_{max})$ and $\tau_{max} > 0$, i.e.,

$\xi(t) \in \Omega_\xi$ and $\boldsymbol{x}(t) \in \Omega_{\boldsymbol{x}}$ for all $t \in \mathcal{J}$.

We next show that $\boldsymbol{y}$ is complete, i.e., $\tau_{max} = \infty$, by contradiction of Lemma 8.2.3. Assume therefore $\tau_{max} < \infty$ and consider the Lyapunov function $V(\epsilon) := \frac{1}{2}\epsilon^2$ and define $\dot{V} := \frac{\partial V}{\partial \epsilon}\frac{d\epsilon}{dt}$. Thus, it holds that

$$\dot{V} = \epsilon\dot{\epsilon} = \epsilon\Big(- \frac{1}{\gamma\xi(1+\xi)}\big(\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \dot{\boldsymbol{x}} - \xi\dot{\gamma}\big)\Big). \tag{8.19}$$

Inserting (8.1) and (8.18) into (8.19) results in

$$\dot{V} = -\frac{\epsilon}{\gamma\xi(1+\xi)}\Big(\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{w} - k\epsilon\Big)$$

Define $\alpha := -\frac{1}{\gamma\xi(1+\xi)}$, which satisfies $\alpha(t) \in [\frac{4}{\gamma_0}, \infty) \in \mathbb{R}_{>0}$ for all $t \in \mathcal{J}$. This follows since $\frac{4}{\gamma_0} \leq -\frac{1}{\gamma_0\xi(1+\xi)} \leq -\frac{1}{\gamma\xi(1+\xi)} \leq -\frac{1}{\gamma_\infty\xi(1+\xi)} < \infty$ for $\xi \in \Omega_\xi$. Next, $\dot{V}$ can be upper bounded by

$$\dot{V} \leq |\epsilon|\alpha\Big\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{w}\Big\| - k\epsilon^2\alpha \leq |\epsilon|\alpha k_1 - k\epsilon^2\alpha \tag{8.20}$$

where the positive constant k_1 is derived as follows. By using the extreme value theorem, which states that a continuous function on a compact set is bounded, it holds that $\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}\|$ is upper bounded. Note therefore that $\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}$ is continuous on $\text{cl}(\Omega_{\boldsymbol{x}})$, where cl denotes the set closure, and that $\text{cl}(\Omega_{\boldsymbol{x}})$ is a compact set. Recall also that $\boldsymbol{x}(t) \in \Omega_{\boldsymbol{x}}$ for all $t \in \mathcal{J}$. Furthermore, $\boldsymbol{w} \in \mathcal{W}$ is bounded. Hence, it follows that there exists an upper bound $k_1 \in \mathbb{R}_{\geq 0}$ for $\Big\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{w}\Big\|$.

According to (8.20), $\dot{V} \leq 0$ if $\frac{k_1}{k} \leq |\epsilon|$ and it can be concluded that the transformed error $|\epsilon|$ will be upper bounded due to the level sets of $V(\epsilon)$ as $|\epsilon(t)| \leq \max\left(|\epsilon(0)|, \frac{k_1}{k}\right)$, i.e., $\epsilon(t)$ is lower and upper bounded and hence evolves in a compact set. Again, by the same arguments as in [61] it follows that there exist compact sets $\Omega'_\xi \subset \Omega_\xi$ and $\Omega'_{\boldsymbol{x}} \subset \Omega_{\boldsymbol{x}}$ such that $\xi(t) \in \Omega'_\xi$ and $\boldsymbol{x}(t) \in \Omega'_{\boldsymbol{x}}$ for all $t \in \mathcal{J}$. Involving Lemma 8.2.3, it follows by contradiction that $\tau = \infty$, i.e., $\mathcal{J} = \mathbb{R}_{\geq 0}$ and all solutions are complete. We can then conclude by the choice of γ and [61, Theorem 2] that $0 < r \leq \rho^\phi(\boldsymbol{x}, 0) \leq \rho_{max}$, which implies $(\boldsymbol{x}, 0) \models \phi$.

Furthermore, we now show that the control law $\boldsymbol{u}(\boldsymbol{x}, t)$ is well-posed, i.e., continuous and bounded. Recall that $\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}\|$ is upper bounded as discussed previously. A lower bound for $\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}\|$ is derived as follows. Since $\rho^\psi(\boldsymbol{x})$ is a smooth and concave function due to Assumption 8.3.1 and 8.3.3, we have $\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}\| \geq \frac{\rho^\psi_{opt} - \rho^\psi(\boldsymbol{x})}{\|\boldsymbol{x}^* - \boldsymbol{x}\|}$ where $\rho^\psi(\boldsymbol{x}^*) = \rho^\psi_{opt}$. It holds that $\rho^\psi_{opt} - \rho^\psi\big(\boldsymbol{x}(t)\big)$ is lower bounded since $\rho_{max} \leq \rho^\psi_{opt} - \chi < \rho^\psi_{opt}$ as by (8.8), which leads to $\rho^\psi\big(\boldsymbol{x}(t)\big) < \rho_{max} < \rho^\psi_{opt}$ since (8.6) holds for all $t \geq 0$. Hence, there exists $\kappa_1 \geq \chi > 0$ such that $\rho^\psi_{opt} - \rho^\psi\big(\boldsymbol{x}(t)\big) \geq \kappa_1$ for all $t \geq 0$. Furthermore, $\|\boldsymbol{x}^* - \boldsymbol{x}(t)\|$ is upper bounded since $\boldsymbol{x}^*$ is finite and since $\boldsymbol{x}(t) \in \Omega'_{\boldsymbol{x}}$, a compact set, for all $t \geq 0$ so that there exists a $\kappa_2 > 0$ such that $\|\boldsymbol{x}^* - \boldsymbol{x}(t)\| \leq \kappa_2$. Thus, $\|\frac{\partial \rho^\psi(\boldsymbol{x})}{\partial \boldsymbol{x}}\| \geq \frac{\kappa_1}{\kappa_2} > 0$. The functions $\epsilon(\boldsymbol{x}, t)$, $\xi(\boldsymbol{x}, t)$, $f(\boldsymbol{x})$, and $g(\boldsymbol{x})$ are locally Lipschitz continuous on $\boldsymbol{x}$ and continuous on t. Recall that γ is continuous and bounded with $0 < \gamma(t) < \infty$. Thus, $\boldsymbol{u}(\boldsymbol{x}, t)$ is well-posed and hence $\boldsymbol{x}$ and all the other functions of $\boldsymbol{x}$ are also well-posed. □

As a comparison with the earlier work [61], the control law (8.18) derived in Theorem 8.4.2 requires the exact knowledge of $f(\boldsymbol{x})$, whereas this was not needed in [61]. Hence, in [61] $f(\boldsymbol{x})$ was used for certain low-level control purposes, e.g., collision avoidance, which is not possible here without representing the collision avoidance as an STL formula.

The continuous feedback control law $\boldsymbol{u}$ proposed in Theorem 8.4.2 achieves the desired goal, i.e., $0 < r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$, which in turns leads to $(\boldsymbol{x}, 0) \models \phi$. However, the focus of this chapter is to design an event-based control law $\hat{\boldsymbol{u}}$. In the sequel, a simple event-based control law will be proposed based on the control law presented in Theorem 8.4.2. In fact, we will show that simply replacing the non-linear control law in (8.18) by its equivalent zero-order hold approximation will be sufficient for our purpose.

In this chapter, we will assume that the associated information structure is $\mathfrak{I}(t) := \{\boldsymbol{x}(\hat{\tau}(t))\}$[1] so that the controller does not need any extra memory. The next theorem presents sufficient conditions for replacing the continuous feedback control law $\boldsymbol{u}(\boldsymbol{x}, t)$ in (8.18) with an event-based one.

Theorem 8.4.3. *With the same assumptions as in Theorem 8.4.2, the event-triggered control law.*

$$\hat{\boldsymbol{u}}(\mathfrak{I}(t), t) := \boldsymbol{u}(\boldsymbol{x}(\hat{\tau}(t)), t) \tag{8.21}$$

guarantees $0 < r \leq \rho^{\phi}(\boldsymbol{x}, 0) \leq \rho_{max}$*, i.e.,* $(\boldsymbol{x}, 0) \models \phi$*, provided that* $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\mathfrak{I}(t), t)\| \leq \delta_{\boldsymbol{u}}$ *for all* $t \in \mathbb{R}_{\geq 0}$*, where* $\delta_{\boldsymbol{u}} > 0$ *is a design parameter.*

Proof. We proceed as in the proof of Theorem 8.4.2 and omit Step A, which can

[1] confer Section 8.2.3 for details

be performed similarly and guarantees a maximal solution $\boldsymbol{y} : \mathcal{J} \to \Omega_{\boldsymbol{y}}$ with $\mathcal{J} := [0, \tau_{max})$. For Step B, we directly start by considering the Lyapunov function $V(\epsilon) = \frac{1}{2}\epsilon^2$. Thus,

$$\dot{V} = -\frac{\epsilon}{\gamma\xi(1+\xi)}\Big(\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \big(f(\boldsymbol{x}) + g(\boldsymbol{x})\hat{\boldsymbol{u}} + \boldsymbol{w}\big) - \xi\dot{\gamma}\Big).$$

Again by denoting $\alpha := -\frac{1}{\gamma\xi(1+\xi)}$, we write $\dot{V}$ as

$$\begin{aligned}\dot{V} &= \alpha\epsilon\Big(\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \big(f(\boldsymbol{x}) + g(\boldsymbol{x})(\hat{\boldsymbol{u}} + \boldsymbol{u} - \boldsymbol{u}) + \boldsymbol{w}\big) - \xi\dot{\gamma}\Big)\\ &\le |\epsilon|\alpha k_1 - \epsilon^2\alpha k + |\epsilon\|\alpha\Big\|\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T g(\boldsymbol{x})\Big\|\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|,\end{aligned}$$

where again $\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ and $g(\boldsymbol{x})$ are continuous functions on the compact set $\mathrm{cl}(\Omega_{\boldsymbol{x}_0})$. Hence, we can write $\|\frac{\partial \rho^{\psi}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T g(\boldsymbol{x})\| \le k_2$ for some positive constant k_2 and finally obtain

$$\dot{V} \le |\epsilon|\alpha(k_1 - \epsilon k + k_2\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|)$$

Thus, $\|\boldsymbol{u} - \hat{\boldsymbol{u}}\| \le \delta_{\boldsymbol{u}}$ implies that $\|\epsilon(t)\| \le \max\{\epsilon(0), \frac{k_1 + k_2\delta_{\boldsymbol{u}}}{k}\}$. By the same arguments as before, we can conclude that $\tau_{max} = \infty$ and hence $0 < r \le \rho^{\phi}(\boldsymbol{x}, 0) \le \rho_{max}$, i.e., $(\boldsymbol{x}, 0) \models \phi$. □

Note that according to Theorem 8.4.3, there is no bound imposed on the value of $\delta_{\boldsymbol{u}}$. Larger values of $\delta_{\boldsymbol{u}}$ imply larger inter-event times, whereas smaller values of $\delta_{\boldsymbol{u}}$ imply more frequent triggering of the events. However, larger values of $\delta_{\boldsymbol{u}}$ also imply that $\epsilon(\boldsymbol{x}, t)$ can attain higher values, which implies (using (8.18)) larger magnitude in the control signal. Thus, the choice of a suitable $\delta_{\boldsymbol{u}}$ is a design parameter.

Theorem 8.4.3 may not be very useful in practice since $\boldsymbol{u}(\boldsymbol{x}, t)$ still needs to be computed continuously in order to ensure $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\mathfrak{I}(t), t)\| \leq \delta_{\boldsymbol{u}}$. The triggering instances t_i and hence the triggering rule in (8.17), which will be derived in the sequel, should be chosen in a way to ensure $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\mathfrak{I}(t), t)\| \leq \delta_{\boldsymbol{u}}$. We provide a sufficient condition based on the normed difference $\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\|$ that ensures $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\mathfrak{I}(t), t)\| \leq \delta_{\boldsymbol{u}}$, i.e., we show that $\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \delta_{\boldsymbol{x}}$ implies $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\mathfrak{I}(t), t)\| \leq \delta_{\boldsymbol{u}}$ (for some $\delta_{\boldsymbol{x}}$ which we compute explicitly). Consequently, the triggering condition is based on $\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \delta_{\boldsymbol{x}}$ and hence it circumvents the problem of the continuous computation of $\boldsymbol{u}(\boldsymbol{x}, t)$.

Due to Assumptions 8.2.1 and 8.3.3 one can show that $\boldsymbol{u}(\boldsymbol{x}, t)$ is a locally Lipschitz function on $\boldsymbol{x}$ for all t. Let us choose some $\delta_1 > 0$, and by $L(\delta_1)$ let us denote the Lipschitz constant of $\boldsymbol{u}$ over a closed ball $B_{\delta_1}(\boldsymbol{x}(t_i))$ centered at $\boldsymbol{x}(t_i)$ with radius δ_1, i.e., $\|\boldsymbol{u}(\boldsymbol{x}, t) - \boldsymbol{u}(\boldsymbol{y}, t)\| \leq L(\delta_1)\|\boldsymbol{x} - \boldsymbol{y}\|$. Clearly $L(\delta_1)$ is a non-decreasing function of δ_1[2]. Let us consider

$$\delta_{\boldsymbol{x}} := \min(\delta_u / L(\delta_1), \delta_1), \tag{8.22}$$

which implies $B_{\delta_{\boldsymbol{x}}}(\boldsymbol{x}(t_i)) \subseteq B_{\delta_1}(\boldsymbol{x}(t_i))$. Thus, for all $t_{i+1} \geq t > t_i$ and $\boldsymbol{x}(t) \in$

[2]Let $\delta_2 > \delta_1$ and $L(\delta_k)$ be the Lipschitz constant within the ball of $B_{\delta_k}(\boldsymbol{x}(t_i))$. Note that $L(\delta_k)$ is the smallest upper bound possible and that $B_{\delta_2}(\boldsymbol{x}(t_i)) \supset B_{\delta_1}(\boldsymbol{x}(t_i))$. Thus, with $L(\delta_2)$ being the Lipschitz constant of a larger set, we have $L(\delta_2) \geq L(\delta_1)$.

$B_{\delta_{\boldsymbol{x}}}(\boldsymbol{x}(t_i))$ and by using Lipschitz continuity of $\boldsymbol{u}$, we can obtain:

$$\|\boldsymbol{u}(\boldsymbol{x}(t), t) - \hat{\boldsymbol{u}}(\Im(t), t)\| \leq L(\delta_1)\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \delta_{\boldsymbol{u}}.$$

Therefore, $\boldsymbol{x}(t) \in B_{\delta_{\boldsymbol{x}}}(\boldsymbol{x}(t_i))$ for all $t \in (t_i, t_{i+1}]$ is a sufficient condition to ensure $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\Im(t), t)\| \leq \delta_{\boldsymbol{u}}$. Thus, we use $\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \delta_{\boldsymbol{x}}$ for the triggering generations, and the $(i+1)$-th triggering instance is

$$t_{i+1} = \inf\{t > t_i | \|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| > \delta_{\boldsymbol{x}}\},$$

which is the triggering rule given in (8.17).

Remark 8.4.4. *With the choice of* $\delta_{\boldsymbol{x}}$ *in* (8.22), $\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \delta_{\boldsymbol{x}}$ *is a sufficient condition for* $\|\boldsymbol{u}(\boldsymbol{x}, t) - \hat{\boldsymbol{u}}(\Im(t), t)\| \leq \delta_{\boldsymbol{u}}$. *Hence Theorem 8.4.1 is finally obtained by the choice of* $\delta_{\boldsymbol{x}}$ *(in* (8.22)*) in conjunction with Theorem 8.4.3.*

Thus, (8.17) is used as a triggering rule from this point onwards. Since the triggerings are generated by the rule (8.17), we now show that Zeno behavior is not exhibited by the proposed event-triggered control law.

Corollary 8.4.5. *The proposed event-triggered control law in Theorem 8.4.3 does not exhibit Zeno behavior, i.e., the inter-event times* $t_{i+1} - t_i$ *for all* $i \in \mathbb{N}$ *are lower bounded.*

Proof. Note that the triggering condition is whenever $t_{i+1} = \inf\{t > t_i | \|\boldsymbol{x}(t) -$

$\boldsymbol{x}(t_i)\| > \delta_{\boldsymbol{x}}\}$ for $\delta_{\boldsymbol{x}}$ as in (8.22). From the dynamics in (8.1) we have

$$\begin{aligned}\boldsymbol{x}(t) &= \boldsymbol{x}(t_i) + \int_{t_i}^{t} [f(\boldsymbol{x}(s)) + g(\boldsymbol{x}(s))\hat{\boldsymbol{u}}(\mathfrak{I}(s), s) + \boldsymbol{w}(s)]ds \\ \boldsymbol{x}(t) - \boldsymbol{x}(t_i) &= \int_{t_i}^{t} [f(\boldsymbol{x}(s)) - f(\boldsymbol{x}(t_i))]ds + \int_{t_i}^{t} K_1(s)ds \\ &\quad + \int_{t_i}^{t} [g(\boldsymbol{x}(s)) - g(\boldsymbol{x}(t_i))]\hat{\boldsymbol{u}}(\mathfrak{I}(s), s)ds\end{aligned}$$

so that

$$\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \int_{t_i}^{t} K(s)ds + \int_{t_i}^{t} L_0(s)\|\boldsymbol{x}(s) - \boldsymbol{x}(t_i)\|ds$$

where $K(s) = \|K_1(s)\|$,

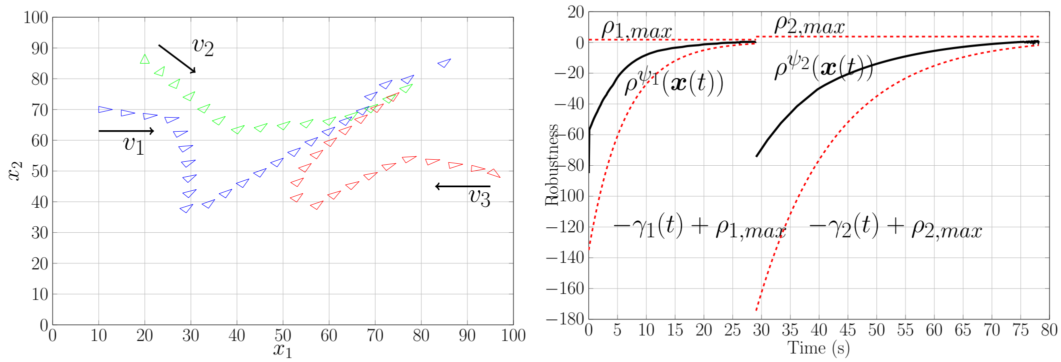
$$K_1(s) := f(\boldsymbol{x}(t_i)) + g(\boldsymbol{x}(t_i))\hat{\boldsymbol{u}}(\mathfrak{I}(t_i), s) + \boldsymbol{w}(s)$$

and

$$L_0(s) := L_f + L_g\hat{\boldsymbol{u}}(\mathfrak{I}(t_i), s)$$

with L_f and L_g being the Lipschitz constants of the functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ in the domain $B_{\delta_{\boldsymbol{x}}}(\boldsymbol{x}(t_i))$. Thus using the Grönwall-Bellman inequality,

$$\|\boldsymbol{x}(t) - \boldsymbol{x}(t_i)\| \leq \Big(\int_{t_i}^{t} K(s)ds\Big)e^{\int_{t_i}^{t} L_0(s)ds} \tag{8.23}$$



(a) Agent Trajectories (b) Robustness

Figure 8.1: Agent Trajectories and Robustness

In order for $\boldsymbol{x}(t)$ to leave the domain $B_{\delta_{\boldsymbol{x}}}(\boldsymbol{x}(t_i))$, it is necessary that

$$\zeta(t) := \Big(\int_{t_i}^{t} K(s)ds \Big) e^{\int_{t_i}^{t} L_0(s)ds} > \delta_{\boldsymbol{x}}$$

Clearly $\zeta(t_i) = 0$ and $\zeta(t)$ is differentiable everywhere for all $t > t_i$ with finite $\zeta'(t)$ and hence $\zeta(t)$ is Lipschitz continuous. Let us denote its Lipschitz constant by L_ζ.

Therefore at the next triggering instance t_{i+1} we have

$$L_\zeta(t_{i+1} - t_i) \geq \zeta(t_{i+1}) \geq \delta_{\boldsymbol{x}} \tag{8.24}$$

$$t_{i+1} - t_i \geq \delta_{\boldsymbol{x}} / L_\zeta \tag{8.25}$$

and thereby, Zeno behavior is excluded. □

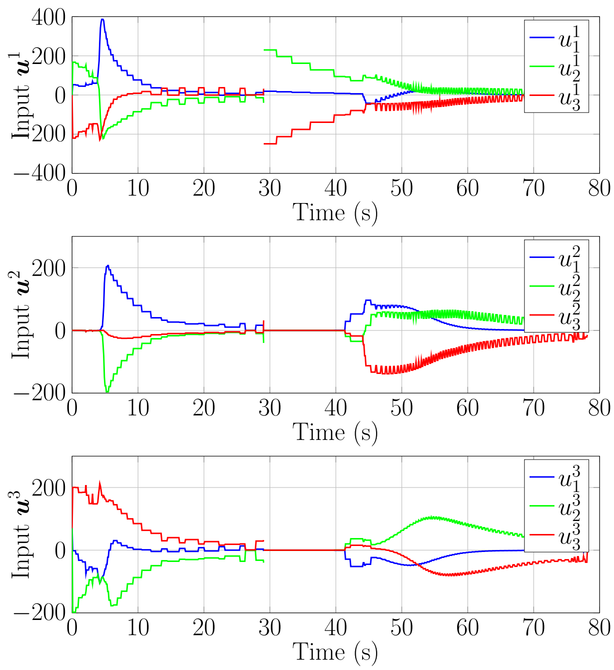


Figure 8.2: Inputs

8.5 Simulations

We consider a centralized multi-agent system consisting of three agents v_1, v_2, and v_3. Each agent is described by its two-dimensional position and its orientation. In figures, the orientation will be indicated by triangles. More specifically, each agent is a three-wheeled omni-directional mobile robot as in [63] with three states: two states x_1 and x_2 describing the robot's position and one state x_3 describing its orientation with respect to the x_1-axis. The states of each agent v_i with $i \in \{1, 2, 3\}$ are hence described by $\boldsymbol{x}^i := \begin{bmatrix} x_1^i & x_2^i & x_3^i \end{bmatrix}$, while the control input is $\boldsymbol{u}^i := \begin{bmatrix} u_1^i & u_2^i & u_3^i \end{bmatrix}$. The multi-agent system is described by the stacked state vector of all agents as $\boldsymbol{x} := \begin{bmatrix} \boldsymbol{x}^1 & \boldsymbol{x}^2 & \boldsymbol{x}^3 \end{bmatrix}^T$ and the stacked input vector is $\boldsymbol{u} := \begin{bmatrix} \boldsymbol{u}^1 & \boldsymbol{u}^2 & \boldsymbol{u}^3 \end{bmatrix}^T$.

The dynamics of each robot are

$$\dot{\boldsymbol{x}}^i = g_i(\boldsymbol{x}^i)\boldsymbol{u}^i = \begin{bmatrix} \cos(x_3^i) & -\sin(x_3^i) & 0 \\ \sin(x_3^i) & \cos(x_3^i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(B_i^T\right)^{-1} R_i \boldsymbol{u}^i,$$

where R_i is the wheel radius and

$$B_i := \begin{bmatrix} 0 & \cos(\pi/6) & -\cos(\pi/6) \\ -1 & \sin(\pi/6) & \sin(\pi/6) \\ L_i & L_i & L_i \end{bmatrix}$$

describes geometrical constraints with L_i as the radius of the robot body. We set $L_i := 0.2$ and $R_i := 0.02$ for $i \in \{1, 2, 3\}$. The overall dynamics are then given by

$$\dot{\boldsymbol{x}} := \operatorname{diag}\big(g_1(\boldsymbol{x}^1), g_2(\boldsymbol{x}^2), g_3(\boldsymbol{x}^3)\big)\boldsymbol{u}.$$

The STL task imposed on the multi-agent system is a formula $\theta := F_{[0,50]}\psi_1 \wedge F_{[50,100]}\psi_2$ where ψ_1 orders agent v_1, v_2, and v_3 to the positions $\begin{bmatrix} 20 & 30 \end{bmatrix}^T$, $\begin{bmatrix} 40 & 60 \end{bmatrix}^T$, and $\begin{bmatrix} 60 & 30 \end{bmatrix}^T$, respectively, while eventually all agents have an orientation of 45 degrees. Furthermore, agent v_1 and v_3 should stay close; ψ_2 orders agent v_1 to $\begin{bmatrix} 90 & 90 \end{bmatrix}^T$, while agent v_1 and v_2 and agent v_2 and v_3 should stay in proximity and while all agents remain the orientation of 45 degrees. In formulas, this can be expressed as $\psi_1 := (\| \begin{bmatrix} x_1^1 & x_2^1 \end{bmatrix}^T - \begin{bmatrix} 20 & 30 \end{bmatrix}^T \| < 10) \wedge (\| \begin{bmatrix} x_1^2 & x_2^2 \end{bmatrix}^T - \begin{bmatrix} 40 & 60 \end{bmatrix}^T \| <$

$10) \wedge (\| \begin{bmatrix} x_1^3 & x_2^3 \end{bmatrix}^T - \begin{bmatrix} 60 & 30 \end{bmatrix}^T \| < 10) \wedge (\| \begin{bmatrix} x_1^1 & x_2^1 \end{bmatrix}^T - \begin{bmatrix} x_1^3 & x_2^3 \end{bmatrix}^T \| < 30) \wedge (|x_3^1 - 45| < 5) \wedge (|x_3^2 - 45| < 5) \wedge (|x_3^3 - 45| < 5)$ and $\psi^2 := (\| \begin{bmatrix} x_1^1 & x_2^1 \end{bmatrix}^T - \begin{bmatrix} 90 & 90 \end{bmatrix}^T \| < 10) \wedge (\| \begin{bmatrix} x_1^1 & x_2^1 \end{bmatrix}^T - \begin{bmatrix} x_1^2 & x_2^2 \end{bmatrix}^T \| < 10) \wedge (\| \begin{bmatrix} x_1^2 & x_2^2 \end{bmatrix}^T - \begin{bmatrix} x_1^3 & x_2^3 \end{bmatrix}^T \| < 10) \wedge (|x_3^1 - 45| < 5) \wedge (|x_3^2 - 45| < 5) \wedge (|x_3^3 - 45| < 5)$. These formulas result in $\rho_{opt}^{\psi_1} = 1.86$ and $\rho_{opt}^{\psi_2} = 3.89$ so that $\rho_{max}^{\psi_1} := 1.8$, $\rho_{max}^{\psi_1} := 3.8$, $r_1 := 0.5$, and $r_2 := 1$ have been selected. The parameters k and δ have been set to $k := 0.75$ and $\delta := 50$.

All simulations have been performed in real-time on a two-core 1,8 GHz CPU with 4 GB of RAM. Computational complexity is not an issue due to the computationally efficient feedback control laws. The agent trajectories in the x_1-x_2 plane are displayed in Fig. 8.1(a), while the funnel (8.6), including $\rho^{\psi_1}(\boldsymbol{x}(t))$ and $\rho^{\psi_2}(\boldsymbol{x}(t))$, is shown in Fig. 8.1(b). The formula θ is satisfied, i.e, $(\boldsymbol{x}, 0) \models \theta$, and it holds that $\min(r_1, r_2) = 0.5 < \rho^{\theta}(\boldsymbol{x}, 0) < 1.8 = \min(\rho_{1,max}, \rho_{2,max})$. The control inputs are shown in Fig. 8.2 and it is visible that during the satisfaction of the first subformula $F_{[0,50]}\psi_1$ there are fewer control updates than for the second subformula $F_{[50,100]}\psi_2$. The coordination of agent v_1, v_2, and v_3 leads to an increase in control updates in the latter case. The sampling time has been 100 Hz, i.e., 0.01 seconds step length. In total, there were 7725 samples and 185 control updates, thereby reducing the communication load by a factor of 41.76.

8.6 Conclusion

In this chapter, we have derived an event-triggered feedback control law for dynamical systems under signal temporal logic tasks. The event-triggering mechanism is based on a norm bound on the difference between the event-triggered control law and a continuous feedback version of it. This leads to a significant decrease in communication between the sensors of the dynamical system and the actuators, which is of high interest in practical applications where communication is usually assumed to be costly. In this Chapter, we propose a method to overcome potential practical problems with high sampling times that may occur in existing works, where we derived a continuous feedback control law for dynamic systems under signal temporal logic tasks.

The event-triggering mechanism is based on a continuous feedback control law that, as opposed to the existing work, eliminates some terms in the system dynamics. These terms have, however, been proven to be useful to formulate, for instance, dynamic couplings in multi-agent systems that encode collision avoidance or consensus tasks. A research direction is hence to derive an event-triggering mechanism that does not eliminate these system dynamics. Another future direction is to derive a self-triggering mechanism to avoid continuous sensor readings.

Bibliography

[1] Rajeev Alur, Thomas A Henzinger, Gerardo Lafferriere, and George J Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[2] Aaron D Ames, Alessandro Abate, and Shankar Sastry. Sufficient conditions for the existence of Zeno behavior. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 696–701. IEEE, 2005.

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

[4] Marco Antoniotti and Bud Mishra. Discrete event models+ temporal logic= supervisory controller: Automatic synthesis of locomotion controllers. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 1441–1446. IEEE, 1995.

[5] Karl-Erik Årzén. A simple event-based PID controller. In *Proc. 14th IFAC World Congress*, volume 18, pages 423–428, 1999.

[6] Karl J Aström. Event based control. In *Analysis and design of nonlinear control systems*, pages 127–147. Springer, 2008.

[7] Karl Johan Åström and Bo Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of the 14th IFAC World congress*, volume 11, pages 301–306. Citeseer, 1999.

[8] Karl Johan Åström and Bo Bernhardsson. Comparison of Riemann and Lebesque sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, volume 2, pages 2011–2016. IEEE, 2002.

[9] AI Medina Ayala, Sean B Andersson, and Calin Belta. Probabilistic control from time-bounded temporal logic specifications in dynamic environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4705–4710. IEEE, 2012.

[10] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[11] T Basar. On the uniqueness of the Nash solution in linear-quadratic differential games. *International Journal of Game Theory*, 5(2-3):65–90, 1976.

[12] Tamer Basar. A counterexample in linear-quadratic games: Existence of nonlinear Nash solutions. *Journal of Optimization Theory and Applications*, 14(4):425–430, 1974.

[13] Tamer Basar and Geert Jan Olsder. *Dynamic noncooperative game theory*, volume 200. SIAM, 1995.

[14] Charalampos P Bechlioulis and George A Rovithakis. A low-complexity global approximation-free control scheme with prescribed performance for unknown pure feedback systems. *Automatica*, 50(4):1217–1226, 2014.

[15] Calin Belta, Antonio Bicchi, Magnus Egerstedt, Emilio Frazzoli, Eric Klavins, and George J Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):61–70, 2007.

[16] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[17] Pierre Bernhard. Linear-quadratic, two-person, zero-sum differential games: necessary and sufficient conditions. *Journal of Optimization Theory and Applications*, 27(1):51–69, 1979.

[18] Wenming Bian and Mark French. General fast sampling theorems for nonlinear systems. *Systems & control letters*, 54(11):1037–1050, 2005.

[19] Howie M Choset, Seth Hutchinson, Kevin M Lynch, George Kantor, Wolfram Burgard, Lydia E Kavraki, and Sebastian Thrun. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[20] Alessandro Cimatti, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *International Conference on Computer Aided Verification*, pages 359–364. Springer, 2002.

[21] Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model checking*. MIT press, 1999.

[22] David C Conner, Alfred A Rizzi, and Howie Choset. Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies. *Robotics Institute*, page 124, 2006.

[23] Samuel Coogan, Ebru Aydin Gol, Murat Arcak, and Calin Belta. Traffic network control from temporal logic specifications. *IEEE Transactions on Control of Network Systems*, 3(2):162–172, 2016.

[24] JB Cruz Jr and CI Chen. Series Nash solution of two-person, nonzero-sum, linear-quadratic differential games. *Journal of Optimization Theory and Applications*, 7(4):240–257, 1971.

[25] Claudio De Persis. Robust stabilization of nonlinear systems by quantized and ternary control. *Systems & Control Letters*, 58(8):602–608, 2009.

[26] Claudio De Persis and Alberto Isidori. Stabilizability by state feedback implies stabilizability by encoded state feedback. *Systems & Control Letters*, 53(3):249–258, 2004.

[27] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[28] Dimos V Dimarogonas, Emilio Frazzoli, and Karl H Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5):1291–1297, 2012.

[29] Dimos V Dimarogonas and Karl Henrik Johansson. Event-triggered control for multi-agent systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 7131–7136. IEEE, 2009.

[30] Derui Ding, Zidong Wang, Bo Shen, and Guoliang Wei. Event-triggered consensus control for discrete-time stochastic multi-agent systems: the input-to-state stability in probability. *Automatica*, 62:284–291, 2015.

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

[32] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 92–106. Springer, 2010.

[33] Alexandre Donzé, Oded Maler, Ezio Bartocci, Dejan Nickovic, Radu Grosu, and Scott Smolka. On temporal logic and signal processing. In *International Symposium on Automated Technology for Verification and Analysis*, pages 92–106. Springer, 2012.

[34] Jacob Engwerda. *LQ dynamic optimization and differential games*. John Wiley & Sons, 2005.

[35] Michael Erdmann and Tomas Lozano-Perez. On multiple moving objects. *Algorithmica*, 2(1-4):477, 1987.

[36] Georgios E Fainekos, Antoine Girard, Hadas Kress-Gazit, and George J Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, 2009.

[37] Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[38] MH Foley and WE Schmitendorf. On a class of nonzero-sum, linear-quadratic differential games. *Journal of Optimization Theory and Applications*, 7(5):357–377, 1971.

[39] Drew Fudenberg and Jean Tirole. *Game theory*. MIT Press, Cambridge, Massachusetts, 1991.

[40] Antoine Girard and George J Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.

[41] Maria Guinaldo, Dimos V Dimarogonas, Karl Henrik Johansson, J Sánchez, and Sebastián Dormido. Distributed event-based control for interconnected linear systems. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2553–2558. IEEE, 2011.

[42] WPMH Heemels, RJA Gorter, A van Zijl, PPJ Van den Bosch, Siep Weiland, WHA Hendrix, and MR Vonder. Asynchronous measurement and control: a case study on motor synchronization. *Control Engineering Practice*, 7(12):1467–1482, 1999.

[43] WPMH Heemels, Karl H Johansson, and Paulo Tabuada. An introduction to event-triggered and self-triggered control. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3270–3285. IEEE, 2012.

[44] WPMH Heemels, JH Sandee, and PPJ Van Den Bosch. Analysis of event-driven controllers for linear systems. *International journal of control*, 81(4):571–590, 2008.

[45] Toivo Henningsson, Erik Johannesson, and Anton Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, 2008.

[46] Gerard J. Holzmann. The model checker spin. *IEEE Transactions on software engineering*, 23(5):279–295, 1997.

[47] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[48] Kamal Kant and Steven W Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *The international journal of robotics research*, 5(3):72–89, 1986.

[49] Sertac Karaman and Emilio Frazzoli. Vehicle routing problem with metric temporal logic specifications. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3953–3958. IEEE, 2008.

[50] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[51] Hassan K Khalil and JW Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

[52] Marius Kloetzer and Calin Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008.

[53] Leslie Lamport. What good is temporal logic? In *IFIP congress*, volume 83, pages 657–668, 1983.

[54] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[55] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[56] Steven M LaValle and Rajeev Sharma. On motion planning in changing, partially predictable environments. *The International Journal of Robotics Research*, 16(6):775–805, 1997.

[57] Daniel Lehmann and Jan Lunze. Event-based output-feedback control. In *Control & Automation (MED), 2011 19th Mediterranean Conference on*, pages 982–987. IEEE, 2011.

[58] Michael Lemmon. Event-triggered feedback in control, estimation, and optimization. In *Networked Control Systems*, pages 293–358. Springer, 2010.

[59] Michael Lemmon, Thidapat Chantem, Xiaobo Sharon Hu, and Matthew Zyskowski. On self-triggered full-information H-infinity controllers. In *Hybrid Systems: computation and control*, pages 371–384. Springer, 2007.

[60] Huiping Li and Yang Shi. Event-triggered robust model predictive control of continuous-time nonlinear systems. *Automatica*, 50(5):1507–1513, 2014.

[61] Lars Lindemann, Christos K Verginis, and Dimos V Dimarogonas. Prescribed performance control for signal temporal logic specifications. In *Proceedings of the 2017 IEEE 56th Conference on Decision and Control (CDC)*, pages 2997–3002, Melbourne, Australia, December 2017.

[62] Stephen R Lindemann and Steven M LaValle. Smoothly blending vector fields for global robot navigation. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 3553–3559. IEEE, 2005.

[63] Yong Liu, J Jim Zhu, Robert L Williams, and Jianhua Wu. Omni-directional mobile robot controller based on trajectory linearization. *Robotics and Autonomous Systems*, 56(5):461–479, 2008.

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

[65] Dipankar Maity and John S Baras. Event based control for control affine nonlinear systems: A lyapunov function based approach. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 3767–3772. IEEE, 2015.

[66] Dipankar Maity and John S Baras. Event based control of stochastic linear systems. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on*, pages 1–8. IEEE, 2015.

[67] Dipankar Maity and John S Baras. Motion planning in dynamic environments with bounded time temporal logic specifications. In *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pages 940–946. IEEE, 2015.

[68] Dipankar Maity and John S Baras. Optimal strategies for stochastic linear quadratic differential games with costly information. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 276–282. IEEE, 2016.

[69] Dipankar Maity and John S Baras. Event-triggered controller synthesis for dynamical systems with temporal logic constraints. *arXiv preprint arXiv:1802.09657*, 2018.

[70] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Proceedings of the International Conference on FORMATS-FTRTFT*, pages 152–166, Grenoble, France, 2004.

[71] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag New York, Inc., New York, NY, 1 edition, 1992.

[72] Adam Molin and Sandra Hirche. Optimal event-triggered control under costly observations. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010)*, 2010.

[73] Adam Molin and Sandra Hirche. Suboptimal event-based control of linear systems over lossy channels. *IFAC Proceedings Volumes*, 43(19):55–60, 2010.

[74] Adam Molin and Sandra Hirche. Event-triggered state estimation: An iterative algorithm and optimality properties. *IEEE Transactions on Automatic Control*, 2017.

[75] Dragan Nešić, Andrew R Teel, and PV Kokotović. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters*, 38(4):259–270, 1999.

[76] Paul G Otanez, James R Moyne, and Dawn M Tilbury. Using deadbands to reduce communication in networked control systems. In *American Control Conference, 2002. Proceedings of the 2002*, volume 4, pages 3015–3020. IEEE, 2002.

[77] Joël Ouaknine and James Worrell. Some recent results in metric temporal logic. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 1–13. Springer, 2008.

[78] Amir Pnueli. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57. IEEE, 1977.

[79] Romain Postoyan, Paulo Tabuada, Dragan Nešić, and Adolfo Anta. A framework for the event-triggered stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 60(4):982–996, 2015.

[80] Maben Rabi, Karl H Johansson, and Mikael Johansson. Optimal stopping for event-triggered sensing and actuation. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3607–3612. IEEE, 2008.

[81] John Reif and Micha Sharir. Motion planning in the presence of moving obstacles. *Journal of the ACM (JACM)*, 41(4):764–790, 1994.

[82] Mark Reynolds. Continuous temporal models. In *Australian Joint Conference on Artificial Intelligence*, pages 414–425. Springer, 2001.

[83] Mark Reynolds. The complexity of temporal logic over the reals. *Annals of Pure and Applied Logic*, 161(8):1063–1096, 2010.

[84] Arthur Richards and Jonathan P How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference, 2002. Proceedings of the 2002*, volume 3, pages 1936–1941. IEEE, 2002.

[85] Elon Rimon and Daniel E Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on robotics and automation*, 8(5):501–518, 1992.

[86] Georg S Seyboth, Dimos V Dimarogonas, and Karl Henrik Johansson. Control of multi-agent systems via event-based communication. In *Proceedings of the 18th IFAC World Congress, 2011*, 2011.

[87] Rajeev Sharma. Locally efficient path planning in an uncertain, dynamic environment using a probabilistic model. *IEEE Transactions on Robotics and Automation*, 8(1):105–110, 1992.

[88] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Springer Science & Business Media, Berlin, Germany, 2 edition, 2013.

[89] Christian Stöcker and Jan Lunze. Event-based control of input-output linearizable systems. In *Proceedings of 18th IFAC World Congress*, 2011.

[90] Christian Stöcker and Jan Lunze. Event-based control of nonlinear systems: An input-output linearization approach. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2541–2546. IEEE, 2011.

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

[92] Alphan Ulusoy, Stephen L Smith, Xu Chu Ding, and Calin Belta. Robust multi-robot optimal path planning with temporal logic constraints. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4693–4698. IEEE, 2012.

[93] Moshe Y Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the First Symposium on Logic in Computer Science*, pages 322–331. IEEE Computer Society, 1986.

[94] Manel Velasco, Josep Fuertes, and Pau Marti. The self triggered task model for real-time control systems. In *Work-in-Progress Session of the 24th IEEE Real-Time Systems Symposium (RTSS03)*, volume 384, 2003.

[95] Xiaofeng Wang and Michael D Lemmon. Event design in event-triggered feedback control systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 2105–2110. IEEE, 2008.

[96] Xiaofeng Wang and Michael D Lemmon. Event-triggered broadcasting across distributed networked control systems. In *American Control Conference, 2008*, pages 3139–3144. IEEE, 2008.

[97] Xiaofeng Wang and Michael D Lemmon. Event-triggering in distributed networked systems with data dropouts and delays. In *HSCC*, volume 9, pages 366–380. Springer, 2009.

[98] AJTM Weeren, JM Schumacher, and JC Engwerda. Asymptotic analysis of linear feedback Nash equilibria in nonzero-sum linear-quadratic differential games. *Journal of Optimization Theory and Applications*, 101(3):693–722, 1999.

[99] Eric M Wolff, Ufuk Topcu, and Richard M Murray. Automaton-guided controller synthesis for nonlinear systems with temporal logic. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4332–4339. IEEE, 2013.

[100] Wing Shing Wong and Roger W Brockett. Systems with finite communication bandwidth constraints. I. state estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, 1997.

[101] Wing Shing Wong and Roger W Brockett. Systems with finite communication bandwidth constraints. II. Stabilization with limited information feedback. *Automatic Control, IEEE Transactions on*, 44(5):1049–1053, 1999.

[102] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[103] Wei Xi, Xiaobo Tan, and John S Baras. A hybrid scheme for distributed control of autonomous swarms. In *American Control Conference, 2005. Proceedings of the 2005*, pages 3486–3491. IEEE, 2005.

[104] Hao Zhang, Gang Feng, Huaicheng Yan, and Qijun Chen. Observer-based output feedback event-triggered control for consensus of multi-agent systems. *IEEE Transactions on Industrial Electronics*, 61(9):4885–4894, 2014.

[105] Jun Zhang, Karl Henrik Johansson, John Lygeros, and Shankar Sastry. Zeno hybrid system. *International Journal of Robust and Nonlinear Control*, 11(5):435–451, 2001.

[106] Y Zheng, DH Owens, and SA Billings. Fast sampling and stability of nonlinear sampled-data systems: Part 2. sampling rate estimations. *IMA Journal of Mathematical Control and Information*, 7(1):13–33, 1990.