Timing is everything: Analysis and synthesis of traffic patterns in event-triggered control

de Albuquerque Gleizer, G.

DOI
10.4233/uuid:dc2fe776-5da3-4894-bd09-1e62d0397c83

Publication date
2022

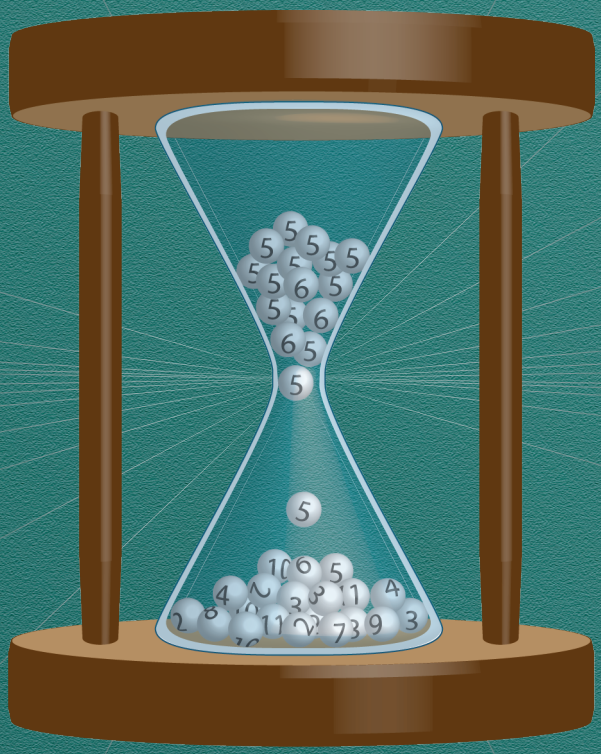
Document Version
Final published version

Citation (APA)
de Albuquerque Gleizer, G. (2022). *Timing is everything: Analysis and synthesis of traffic patterns in event-triggered control*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:dc2fe776-5da3-4894-bd09-1e62d0397c83

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Timing is Everything

Analysis and Synthesis of Traffic Patterns in Event-Triggered Control



Gabriel de Albuquerque Gleizer

Timing is everything: Analysis and synthesis of traffic patterns in event-triggered control

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board of Doctorates,
to be defended publicly on
Wednesday 29 June 2022 at 17:30 o'clock

by

Gabriel DE ALBUQUERQUE GLEIZER

Mestre em Ciências, em Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Brazil,
born in Rio de Janeiro, Brazil.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Dr. M. Mazo Espinosa,	Delft University of Technology, promotor
Prof. dr. ir. B. De Schutter,	Delft University of Technology, promotor
Independent members:	
Prof. dr. D. Nesic,	University of Melbourne, Australia
Prof. dr. P. Tabuada,	University of California, Los Angeles, USA
Prof. dr. ir. M. H. G. Verhaegen,	Delft University of Technology
Dr. D. G. T. Antunes,	Eindhoven University of Technology
Dr. R. Postoyan,	CNRS and Lorraine University, France
Prof. dr. ir. T. Keviczky,	Delft University of Technology

This research was funded by the European Research Council through the SENTIENT project, Grant No. 755953.

Keywords: Event-triggered control, self-triggered control, formal methods, abstraction, scheduling, networked control systems, cyber-physical systems, hybrid systems, sampled-data control, quantitative analysis and synthesis, chaos, combinatorial games, automata theory, bounded disturbances.

Printed by: Print Service Ede

Cover: Gabriel de Albuquerque Gleizer

Style: TU Delft House Style, with small modifications

The author has set this thesis in LaTeX using the Libertinus and Inconsolata fonts.

ISBN: 978-94-6384-347-8 e-ISBN: 978-94-6366-573-5

To Eliza and Liana.

Contents

Summary

Event-triggered control (ETC) and self-triggered control (STC) are sample-and-hold control paradigms in which sensor data is only updated to the controller when necessary, often aperiodically, in contrast to the well-established periodic sampling paradigm. In ETC, a state-dependent event triggers a transmission, while in STC the controller decides when to request the next sample. The main objective of ETC and STC is to reduce sampling and transmissions when either sampling/transmitting is costly or network resources are scarce. However, despite years of development in the ETC/STC field, little is known about their sampling performances or how to accommodate the generated aperiodic traffic of multiple ETC systems in a shared communication medium. This dissertation presents methods to (i) schedule multiple ETC systems in a shared network, (ii) evaluate ETC systems' sampling performance, and (iii) creating STC strategies that improve ETC systems' sampling performance. In particular, we focus for the most part on ETC applied to linear time-invariant (LTI) systems.

To solve these problems, we first model the timing behavior of ETC/STC systems, obtaining what we call *traffic models*. The states of a traffic model are the transmitted samples and its output is the elapsed time between consecutive transmissions, the *inter-sample time* (IST). These models are infinite-state systems that can exhibit very complex—even chaotic—behavior, as we demonstrate. To solve synthesis problems such as scheduling and optimal STC sampling strategies, we augment the models with early-sampling choices, which are guaranteed to preserve control stability and performance. The models are then abstracted into finite-state systems or timed automata, on which many of our problems can be computationally solved. Using these abstractions, the obtained schedulers are always valid for the real systems, and the obtained metrics are always formal bounds to the real system's performance.

Our abstraction method is based on quotient and l-complete systems. That is, we partition the state-space into regions, each region comprising all states whose next IST, or next sequence of l ISTs, is the same. This is made possible by observing that periodic ETC (PETC)—a practical version of ETC where events are checked periodically—has a finite output set, and that each obtained region is described by an intersection of finitely many quadratic cones. The abstraction transitions, which enable predicting how samples and their corresponding ISTs evolve over time, can be computed exactly using nonlinear satisfiability-modulo-theories solvers, or approximately through convex semi-definite relaxations. Infinite periodic IST patterns arising from these abstractions can be verified to exist in the real traffic model via an eigenvector problem, which is central for solving problem (ii) exactly.

Our methodology comprises a comprehensive framework for solving qualitative (scheduling) and quantitative (sampling performance) problems for ETC and STC, as well as a computational machinery that automates these processes, ultimately consolidated in the open-source tool `ETCetera`. With the developed methods, we can show cases where

ETC significantly outperforms periodic sampling in terms of average inter-sample time, and how to increase this performance further using look-ahead. We also manage to solve the ETC scheduling problem efficiently, which is helped by an abstraction minimization algorithm that we propose. In summary, this dissertation provides new tools to understand and manipulate ETC traffic, and ultimately casts new light on the practical relevance of ETC and STC.

Samenvatting

Event-triggered control (ETC, gebeurtenis getriggerde regeling) en *self-triggered control* (STC, zelf getriggerde regeling) zijn *sample-and-hold* (bemonsteren en vasthouden) regelparadigma's waarin sensordata alleen wordt geüpdate naar de regelaar indien noodzakelijk, vaak aperiodiek, in tegenstelling tot de goed gevestigde periodische bemonstering paradigma. Binnen ETC wordt een transmissie getriggerd door een toestandsafhankelijke gebeurtenis, terwijl in STC de regelaar beslist wanneer het volgende monster wordt opgevraagd. Het hoofddoel van ETC en STC is het reduceren van bemonsteren en transmissies wanneer ofwel bemonsteren/zenden duur is of netwerkbronnen schaars zijn. Ondanks jaren van ontwikkelingen binnen het veld van ETC/STC, is er echter weinig bekend over diens bemonsteringprestaties of hoe de gegenereerde aperiodieke verkeer van meerdere ETC-systemen in een gedeeld communicatiemedium geaccommodeerd moet worden. Deze dissertatie presenteert methodes om (i) meerdere ETC-systemen in een gedeeld netwerk te plannen, (ii) de bemonsteringsprestaties van ETC-systemen te evalueren, en (iii) STC-strategieën te creëren die de bemonsteringsprestaties van ETC-systemen verbeteren. In het bijzonder richten we ons voor het grootste deel op ETC toegepast op lineaire tijd-invariante (LTI) systemen.

Om deze problemen op te lossen, modelleren we eerst het timinggedrag van ETC/STC-systemen, waarbij we zogenaamde verkeersmodellen verkrijgen. De toestanden van een verkeersmodel zijn de verzonden monsters en de uitgang is de verstreken tijd tussen opeenvolgende transmissies, de *inter-sample time* (IST, bemonstering tussentijd). Deze modellen zijn oneindigetoestandssystemen die zeer complex — zelfs chaotisch — gedrag kunnen vertonen, zoals we demonstreren. Om syntheseproblemen zoals planning en optimale STC-bemonsteringstrategieën op te lossen, breiden we de modellen uit met vroegebemonstering keuzes, die gegarandeerd regelaar stabiliteit en prestaties behouden. De modellen worden vervolgens geabstraheerd in eindigetoestandssystemen of tijdsautomaten, waarop veel van onze problemen computationeel kunnen worden opgelost. Met behulp van deze abstracties zijn de verkregen planners altijd geldig voor de echte systemen, en de verkregen metrieken zijn altijd formele grenzen voor de prestaties van het echte systeem.

Onze abstractiemethode is gebaseerd op quotiënt- en l-complete systemen. Dat wil zeggen, we verdelen de toestandsruimte in regio's, waarbij elke regio alle toestanden omvat waarvan de volgende IST of de volgende reeks van l IST's hetzelfde is. Dit wordt mogelijk gemaakt door te observeren dat periodieke ETC (PETC) — een praktische versie van ETC waarbij gebeurtenissen periodiek worden gecontroleerd — een eindige uitgangsset heeft, en dat elke verkregen regio wordt beschreven door een snijpunt van eindige hoeveelheid kwadratische kegels. De abstractie-overgangen, die het mogelijk maken om te voorspellen hoe monsters en hun corresponderende IST's over de tijd evolueren, kunnen exact worden berekend met behulp van niet-lineaire *satisfiability-modulo-theories* oplossers, of bij benadering door convexe semidefiniet programmeeringsproblemen versoepelingen. Oneindige periodieke IST-patronen die voortkomen uit deze abstracties kunnen worden geverifieerd

of deze bestaan in het echte verkeersmodel via een eigenvectorprobleem, die centraal staan voor het exact oplossen van probleem (ii).

Onze methodologie omvat een uitgebreid raamwerk voor het oplossen van kwalitatieve (plannen) en kwantitatieve (bemonsteringsprestaties) problemen voor ETC en STC, evenals computationeel middelen die deze processen automatiseren, en zijn geconsolideerd in de open-source tool `ETCetera`. Met de ontwikkelde methoden kunnen we gevallen laten zien waarin ETC aanzienlijk beter presteert dan periodieke bemonstering in termen van gemiddelde inter-bemonsteringstijd, en hoe deze prestatie verder kan worden verbeterd met behulp van vooruitziendheid. We slagen er ook in om het ETC-planningsprobleem efficiënt op te lossen, middels een voorgesteld abstractieminimalisatie-algoritme. Samengevat biedt deze dissertatie nieuwe middelen om ETC-verkeer te begrijpen en te manipuleren, en werpt het uiteindelijk een nieuw licht op de praktische relevantie van ETC en STC.

Acknowledgments

A doctorate is a long and mostly lonesome journey. The solitude is exacerbated when you do half of your doctorate during a pandemic, working mostly from home. But it would be extremely naive to think it is an individual effort. The style of interactions is different from most types of work, as indeed you have to spend significant time thinking with yourself, writing, rewriting, double-checking, triple-checking, questioning, doubting... but how much can an aspiring scientist do without support from peers and predecessors? And perhaps most importantly, from family and friends, who lift the morale and fight the occasional lack of humor from the doctoral candidate?

First of all, I would like to thank my biggest partner and supporter throughout this journey, my loved wife Eliza. She was the first one to encourage me on this crazy idea of interrupting an industrial career and pursuing my dream of doing a doctorate... abroad. She came with me to the Netherlands with 10x the uncertainty I had, and endured a tough period, lonelier than mine, in a new country with a tough job market to immigrants. Nevertheless, she would always be the first to cheer me up and encourage me to give it my best at my doctorate. She was even brave enough to jump with me on an even greater journey, that of having a baby.

Talking about it, I would like to thank my daughter Liana. Despite how harder life gets with the sleepless nights and the extra responsibility, having such an adorable daughter gave me a whole new level of love and life enjoyment, which I never had thought to exist. She came at the right time, weeks before the pandemic crashed in, and turned otherwise boring into ever-shiny days, full of discoveries and excitement. She also, I confess, endured some monologues about abstractions, linear algebra, and chaos theory while she just wanted to relax and play with her favorite stuffed frog. Nevertheless, I must give big thanks to the daycare teachers that have been taking care of my girl while Eliza and I work. They undoubtedly have the most beautiful job there is.

I would also like to immensely thank my family, in particular my mother, Luciana. She supported my decision to come abroad for a doctorate from day one. I imagine how hard it is for a mother to do so; just thinking about the same with my daughter makes my eyes water... still there she was, rooting for me, making the decision much lighter. I also thank my brother and best friend Bernardo for the same reasons, plus keeping me up-to-date about Botafogo... and my mother-in-law Tânia for having helped us while we moved out of our flat in Rio. Lastly, special thanks to my cousins Orlandino and Matheus, who gave me the pleasure of several great family reunions.

If my family supported my decision, it would not even exist if someone did not bet on me as a doctoral candidate. Hence, I need to greatly thank my daily supervisor, Manuel: it is not every day that someone in Europe selects a candidate coming from 6 years of industry work, and not having studied in US or European schools. But more than accepting to interview me and offering me a job, he has been the best supervisor I could ask for. He has given the right mix of freedom and attention, always providing nice insights, always

pushing for excellence, always embarking on wild ideas. Not without many fun chats about rock music or parenting. Of course, I also thank my second promotor, Professor Bart, for all the interesting discussions, and for having helped in critical times.

Throughout these four years I had the luck of making good friends at work. I would like to thank Cees for the friendship, great conversations, *gezellige* walks, the mentoring in my first few years, and of course, translating my summary to Dutch. I would like to thank Giannis and Artemis for being such good friends, for the long discussions about every imaginable topic, the excellent times in Greece, and also of course for the work collaboration. In addition, I thank Daniel and Twan for the great times and conversations, good discussions and idea exchanges, Khushraj and Sonam for the friendship, for helping us in troublesome times, and for the prolific collaboration, and Rudi, Alexey, Andrea and Anton for the nice talks and good laughs. Finally, I thank all DCSC Ph.D. candidates, postdocs, professors and secretaries for the great times in and outside the office.

Part of my work was done in collaboration with researchers in the University of Trento and Bruno Kessler Foundation, namely Matteo Trobinger, Timofei Istomin, Amy Lynn Murphy and Prof. Gian Pietro Picco. I thank all of them for such a nice and fruitful collaboration, which gave a new twist to the project I have worked on.

I would like to also thank the former Master students I have supervised or just partially helped (and hence, was helped in return): Aleksandra Szymanek, Jacob Lont, Aniket Samant, and Bas Boot. Supervising was one of the parts of the job I enjoyed the most, and certainly having had bright, skillful, independent, and enthusiastic students contributed to that enjoyment.

If I could manage to conclude a dissertation, it is because I have drunk from a knowledge fountain far greater than my immediate colleagues. Each name that appears in the Bibliography list deserves gratitude of mine, but not only those. The reviewers of the papers I have submitted to have contributed greatly to the quality of this work. Furthermore, the several professors who go beyond their job duties to record lectures and write books, especially when they make them freely available on the internet, have been fundamental in the process. I would not make sufficient sense of chaos theory, topology, and algebraic geometry, if it were not for them. The same goes to research groups that develop software tools and make them available for other researches, especially when open-source. And, of course, all people that develop, maintain, and contribute to open-knowledge portals such as Wikipedia, who I cannot thank enough.

To finalize, I am greatly indebted to the European society to have funded my Ph.D., to the Brazilian society to have funded my education prior to that, and to TU Delft and the Dutch society to have received me so well. And to all the great educators that I have the luck to have had. Last but not least, immense thanks to my late father, Adilson, who was my first enthusiast on the path of math and engineering. See you on the other side, dad!

Gabriel
Delft, May 2022

1

Introduction

If any of us humans would describe how we control something that we are operating, it would be quite different than what computers do. Think about how you control the temperature of your shower (if you still have the old pair of hot and cold water valves), or how you control the cooking of your food.

In the first scenario, you set the valves to some level, wait for a while until it is pleasant enough, and start showering; at some point the water may feel too cold or too hot, and that *triggers* something inside you: you stop cleaning yourself for a moment and do a small adjustment in those valves. The process repeats until you are done with showering — and if you are lucky to live in a place where water supplies are steady, you only make a handful of adjustments.

In the second scenario, you are not constantly tasting how good the food is; hence you make a judgment, trying to *predict* how much seasoning is needed, or how much time is left before it is done. Then you add a teaspoon of salt, some 10 rounds of crushing black pepper, stir, wait a few minutes, and try the food. If it is still not quite there yet, you add another pinch of salt, try again, and maybe this time it is good enough.

Real-life automatic control applications are typically not like any of these. In modern days, a sensor collects the relevant data periodically, at some fast enough *sampling rate*; this data feeds a controller running on a computer, which immediately updates the control command to the actuator, that changes the operational setting of the process or machine. This is what we control engineers call *periodic sample-and-hold control*, and it is very successfully used in industry, robotics, home appliances, and many other applications.

So why do we not naturally apply periodic control on our daily activities? The most honest answer would perhaps be "because we do not have internal clocks dictating our actions," or simply "we are not computers". But if we reflect a little further, in none of the scenarios above it would be particularly efficient to do periodic control.

In the showering scenario, the luckiest among us have only a pair of actuators, both of our hands, to clean ourselves *and* regulate the shower temperature. And unless we are extremely skilled, we need both hands for the cleaning process, which is much more demanding to the actuators than regulating the shower temperature. Not surprisingly, then, the strategy that we all learn from experience to efficiently allocate these two tasks

to our limited set of actuators is to apply some sort of *event-triggered control* (ETC) to the temperature regulation problem.

The second scenario is slightly different. We still have the limited number of actuators for a multitude of tasks, but also the process of sensing is much more costly: we need to get a spoon, bring it to the mouth, and wait a few seconds to finish our tasting analysis. It is impractical to taste the food all the time, and in some cases it is even destructive or impossible to do so (think of a steak or a cake). Hence we rely on some *model* of the cooking process, typically a combination of recipe and experience, to predict when is the next time we want to perform the sample-compute-act process. Instead of a sensation triggering the action, *we decide* ourselves when to sense next, an approach called *self-triggered control* (STC).

Historically, automatic control systems are designed such that these resource constraints are absent, by having dedicated devices performing sensing and actuating functions. That is, we install a dedicated sensor to constantly monitor a variable of interest, and put enough actuators on the system such that they do not have to be divided into separate control tasks. However, more recently there is one resource that ceased to be dedicated to a single sensor-control-actuator triplet: the communication medium. Driven by the pressure costs in industrial applications, as well as the increase in monitoring and control functions in appliances or vehicles, the use of communication networks connecting multiple sensors, actuators and control devices became prevalent in the past several decades. The cost benefits range from reduction in wiring costs, to savings in maintenance and commissioning, to reduction of weight in vehicles and of footprint in general. In addition, computer networks, as opposed to old point-to-point 4–20 mA connections, allow for a myriad of different communication packages to be sent in the same network, such as device diagnostics and configuration.[1] Finally, some control applications are only possible when using wireless communication, such as drone swarms and car platooning.

With the communication medium becoming a shared resource, the control community had to break away from the old periodic control paradigm. Instead of treating sampling, transmitting, and actuating as free tasks, some cost needed to be attributed to those tasks. Hence, if the thought was “sample as much as possible”, now it is “sample only when needed”, or “act only when necessary”. This thought process, together with the right people and mathematical tools, originated the concepts of event-triggered and self-triggered control, which we introduce next.

1.1 Event-triggered and self-triggered control

While aperiodic sampling approaches for control have been studied since the 1950s (see, e.g., [2]), it regained renewed interest in the 2000s, probably due to the reasons described earlier. Inspired by the differences between Riemann and Lebesgue integration, an early form of event-triggered control was proposed by Åström and Bernhardsson [3] with the name of Lebesgue sampling. With this approach, a sample is only sent to the controller when the *sampling error*, i.e., the difference between the current measurement and the last sent measurement, exceeds a given threshold. Unlike a properly designed periodic sampling

[1] In fact, the HART protocol [1] allows for complex communications on top of the 4–20 mA standard connection, effectively making a computer network with dedicated point-to-point communication. Obviously, it still suffers from excessive wiring.

strategy, though, Lebesgue sampling does not guarantee global exponential stability (GES) to the origin, but instead a form of practical stability [4], i.e., convergence of the state to a ball around the origin. It was only with Tabuada in [5] that an event design that gives GES was achieved, based on the framework of input-to-state stability (ISS) [6]: a sample should be sent when the *fraction* between the sampling error and the actual state norms exceeds a threshold. Fig. 1.1 shows an example of ETC architecture, the one addressed in [5]. Note that the condition checker sits on the measurement side, which effectively requires this capability to be on the sensing node.

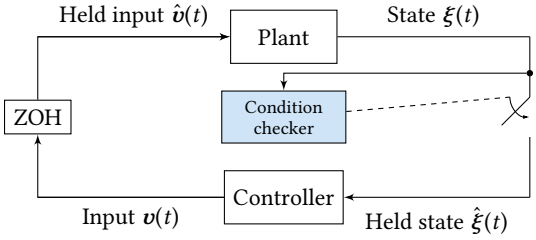


Figure 1.1: A basic ETC block diagram, where the state $\xi(t)$ is fully measured and the triggering condition is verified on the sensor size. ZOH stands for "zero-order hold".

After Tabuada's seminal paper, a myriad of new event designs (e.g., [7, 8]); different triggering architectures and communication architectures, such as decentralized triggering conditions (e.g., [9–12]); and applications to larger classes of control systems like output-feedback control (e.g., [4, 11, 13]) were proposed. In particular, Heemels et al. [11] propose periodic ETC (PETC), where the event is checked periodically instead of continuously. This is a much more practical implementation than continuous ETC (CETC), because smart sensors with condition checking capabilities are typically implemented in computing devices.

The core idea behind [5] is to design an event that is a surrogate of a Lyapunov condition. In essence, the triggering condition is such that, while it is not satisfied, a companion Lyapunov function $V : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}_+$ has negative derivative [5], or is bounded by a decaying function [7, 14] (Fig. 1.2 depicts an example), or is decreasing in average [8]. Hence, if one can predict a time instant such that this triggering condition has not yet turned true, sampling at this moment will ensure the same Lyapunov stability and performance criterion that is ensured by the ETC. This is fundamentally what STC does.

Self-triggered control is a close relative to ETC and followed a similar trajectory. One of the earliest proponents was Velasco et al. [15] in 2003, but the necessary mathematical formalism to establish a solid, general framework, was only given years later [9, 14, 16], again using the ISS formalism. In STC, the controller decides when to sample next given the most recent sample, typically by predicting when an ETC event would happen, as mentioned previously. Fig. 1.3 shows the STC embodiment of the ETC in Fig. 1.1.

STC has some advantages over ETC. The first is that it does not require the sensor to have smart capabilities other than being able to transmit a message upon request. The second is that the communication times are more predictable under STC than under ETC:

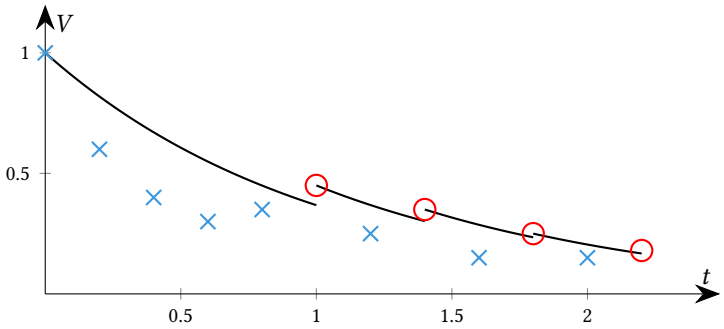


Figure 1.2: Lyapunov function V at states checked under PETC, where the triggering condition is based on bounding the Lyapunov function with an exponentially decaying function. Red circles indicate when samples are transmitted to the controller, while blue x's are the remaining checked states.

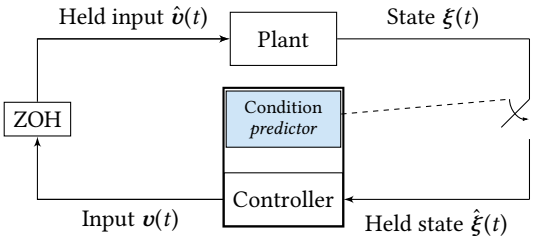


Figure 1.3: A basic STC block diagram, representing the self-triggered implementation of Fig. 1.1.

this can facilitate scheduling if the controller is requesting messages from multiple sensors, and also allows for energy savings on the sensor side. For example, if the controller only needs data from the sensor after 1 minute, the sensor can effectively sleep within this minute. Naturally, these benefits come at some costs. The first one is that extra computation is needed on the controller side. The second is the loss of simplicity: events in ETC are typically very simple formulas, while STC needs heavier mathematical machinery, often involving the model of the plant per se.

Still comparing ETC and STC, there is also an aspect that involves what trade-offs each strategy gives, which is common in comparisons between reactive and predictive methods in control. Due to its reactive nature, ETC can act faster in face of noise and disturbances, or also more wisely *avoid* reacting while disturbances are actually helping stabilization. In contrast, STC may consider a worst-case disturbance prediction analysis to determine sampling times, in an attempt to match ETC's performance while increasing the average sampling frequency; or to ignore the effects of disturbance altogether, providing more predictable sampling times at the cost of a loss in control performance.

1.2 Timing is everything

As argued before, the biggest practical incentive to move from the simple well-established periodic sampling towards ETC or STC is the reduction of necessary transmissions when control systems share a communication medium, i.e., in the context of networked control systems (NCSs). By reducing the communications, one could fit more controllers in the same network, reducing costs and potentially enabling applications with communication bandwidth constraints. In addition, in the particular case of wireless NCSs, less sampling can imply reduced energy consumption on radio usage. This can substantially increase the lifespan of wireless control applications in which wireless motes are battery powered, or make these motes fully energy-sustainable if they use some sort of energy harvesting.

Let us look at the benefit of enabling more controllers per network, hereafter called increasing *network capacity*, by considering a simple example: one controller can access the network at a time, and this communication takes some elapsed time Δ (known as *channel occupancy time*). For simplicity, during this time the whole control task is executed, i.e., sampling, transmissions, and control action calculation. Then, consider a case where three control loops have sampling period of 3Δ each: it is easy to come up with a recurring schedule that accommodates these three control loops, see the Gantt chart in Fig. 1.4 (left) for a depiction. Now, consider only two control loops using ETC in this same network: even if they produce much sparser traffic, the high variability of inter-sample times that they generate make collisions almost inevitable, see Fig. 1.4 (right). Clearly, without some sort of scheduling mechanism to adjust the transmission times, ETC's potential in increasing network capacity cannot be fulfilled.

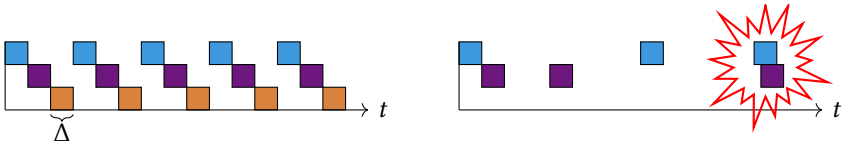


Figure 1.4: Gantt charts depicting network usage by different controllers. Periodic sampling (left) and ETC (right).

In STC, the communication is one-step predictable by design, hence it has advantages in what concerns scheduling. If two controllers would sample at conflicting times, one could anticipate one of these samplings to avoid the collision. Note that, based on how these events are constructed, sampling earlier is always safe from a Lyapunov stability perspective. Nonetheless, imagine you are implementing a network of self-triggered controllers: what is the minimal information you need to be sure that conflicts can always be avoided by sampling early? At least, the *minimum inter-sample time*, or MIST, of each controller needs to be known, and fortunately, methods to compute it are abundant in the ETC and STC literature. Now, suppose the control systems at hand have the same MIST of 3Δ. Clearly, one cannot install four of these control systems in the same network. Now, the MIST can be used as a fixed sampling period, since it is the time interval for which the Lyapunov conditions are satisfied for the whole state space. Hence, even STC cannot increase network capacity if all the information one has is the MISTs.

Let us look now at the second benefit, the mere average reduction in transmissions

that ETC and STC can provide. If one inspects the rich ETC and STC literature, it becomes evident that most of the evidence of superiority of ETC in comparison to periodic control in this aspect is only supported through numerical case studies. In some cases, such as PETC, one can obtain a straightforward *qualitative* assertion of non-inferiority by setting the *event-checking period* equal to the largest periodic sampling time one can obtain. Nonetheless, no *quantitative* measurement of this superiority has been established up until the work that we have conducted; e.g., in PETC, doing this strategy often leads to periodic triggering of the events, bringing no benefits at all.

Hopefully the reader is convinced at this point that a deeper look into these two aspects, scheduling and quantitative evaluation of ETC or STC systems, is paramount to their success in real applications. If the most relevant benefit they can offer is a change in timing of control tasks, then understanding this timing and being able to manipulate it is arguably as important as being able to determine stability and control performance.

Objective of this dissertation

The main objective of this dissertation is to present methods to (i) schedule multiple ETC systems, (ii) evaluate ETC systems' sampling performance, and (iii) if possible improve ETC systems' sampling peformance. As a building block to all these three problems, we need to *model the timing behavior* of ETC/STC systems.

1.3 Related work

Much of the work described in this dissertation is based on symbolic abstractions using the framework of Tabuada [17]. The modeling approaches are basically combinations of well-established quotient systems and also l-complete approximations, which were first introduced by Moor and Raisch [18]. When approaching quantitative problems, we rely on the framework of quantitative automata of Chatterjee et al. [19]. Finally, a significant part of our qualitative understanding of ETC's generated traffic, and the insight to solving several problems, is largely influenced by the behavioral approach to system modeling advocated by Willems [20].

The problem of scheduling for ETC has been investigated prior to the project that culminated in this dissertation. Notably, traffic models and their use for scheduling have been addressed in [21–23], and these works have been used as stepping stones for much that was done in this thesis.

The problem of formally quantifying ETC/STC performance was largely unaddressed until this project. An exception is on deadbeat ETC for discrete-time systems subject to Gaussian disturbances [24]. The subject of qualitatively understanding ETC's sampling behavior gained interest in the past years, during this project, with some interesting initial results on planar linear systems in [25, 26].

1.4 Contributions and outline of this dissertation

In the next chapter we present the mathematical notation used throughout this dissertation, and the necessary preliminaries: (weighted) transition systems and abstractions, ETC and STC, and the fundamentals of ETC and STC traffic modeling.

In Chapter 3 we present the first contribution of this dissertation: scalable abstractions of PETC systems. We develop abstractions whose size (in terms of number of states and transitions) is independent of the state-space dimension, while the computation needed to obtain them is only polynomially dependent on it. Moreover, these models are exact simulations of the original PETC system's traffic models, which gives them an advantage in terms of predictive power. We also propose slightly different ways of solving the scheduling problem; in particular, we notice that the abstractions we obtain allow us to solve the scheduling problem using only finite systems, as opposed to timed automata as proposed in previous work [22]. Hence, we achieve much higher scalability on the scheduling problem, both in the abstraction process and in the scheduling process.

Chapter 4 goes a step further in the modeling process, aiming at improving the long-term predictability of the models presented in Chapter 3. There we propose an abstraction refinement approach that ensures exact predictability for as much time as is need to attain a specified reduction in a reference Lyapunov function. We propose an alternative PETC called Mixed PETC, which switches to (a slow-rate) periodic sampling when the state is close to the origin, and show that we can obtain finite bisimilar traffic models to this Mixed PETC (roughly posed, models that have exact predictive accuracy in the infinite run, see Chapter 2 for a definition of bisimulation). Our proposed models can be used to compute tighter estimates of traffic usage and decay rate of the closed-loop system.

The problem of computing the traffic performance of PETC is addressed thoroughly in Chapter 5. There we show how one can build abstractions and their refinements to compute the precise average inter-sample time of a PETC system. We give conditions to when this precise computation is possible, and examples indicating when it is not. In particular, we show how one can verify a given periodic traffic pattern of the system by casting it as an eigenvalue problem. This work is the first to succeed in quantifying PETC sampling performance, casting new light on its practical importance.

We further study ETC traffic in Chapter 6. We set ourselves to understand why sometimes ETC has predictable traffic, and why in other times it seems to be essentially random. We find examples where in fact ETC exhibits chaotic behavior, which is more common the more aggressive the ETC is designed towards sampling reduction. We present several results related to how one can verify steady or periodic traffic patterns in CETC and PETC, and how this relates to metrics such as average and limit-inferior ISTs. We then establish notions of robust metrics, which are metrics that ignore exceptional initial states: measure-zero sets from which any small perturbation leads to different behaviors with different metrics. We show that these metrics can be computed exactly on PETC that does not exhibit chaotic traffic, how to compute upper bounds to its behavioral entropy (a measure of chaos observed from the traffic traces), and how to compute approximate robust metrics even in chaotic cases.

Chapter 7 moves from quantitative verification to quantitative synthesis. In this chapter we address the problem of improving the average sampling performance of PETC by using early samples that reap benefits in the long run. This is achieved again by employing finite-state abstractions, similarly to what is done for scheduling, but now solving a mean-payoff game [27] on the abstraction, which optimizes the sampling performance in it. We show how to compute the achieved performance by employing this strategy refined to the concrete system — effectively a looking-ahead STC — and how to estimate its optimality

gap. In a numerical example, it is shown that the ETC's sampling performance can be increased drastically by employing our approach.

In Chapter 8, we approach a different, but related, abstraction problem. Instead of abstracting an ETC traffic model, we want to further abstract its abstraction without losing any relevant information. In other words, we address the problem of obtaining reduced abstractions for the sake of improved scalability. While finite system minimization with respect to simulation or bisimulation is largely well-established, to our surprise the problem was unaddressed for minimization with respect to *alternating simulation* (see Section 2.2.3), a relation between systems that is suited for control applications. This is particularly relevant for the scheduling problem, but also for other synthesis problems as in Chapter 7. We provide an algorithm that minimizes finite systems modulo alternating simulation equivalence, proving its correctness and optimality, as well as showing that this system is unique up to a special type of isomorphism. Later we apply this algorithm to PETC traffic models, obtaining interesting general results about their schedulability and showcasing how this minimization can drastically reduce the computation needed to obtain schedulers (or unschedulability certificates).

Chapters 3–7 constitute the core of this dissertation. In all of them, the underlying problem is somehow modeling a nominal PETC traffic — nominal in the sense that the system is not subject to disturbances, noise, and full-state information is communicated to the controller. Chapter 9 is the exception, presenting our last contribution: predicting ISTs for systems under bounded external disturbances and noise, and with partial state information. In fact, this prediction constitutes an STC that can be applied in this more general control setting. We present an approach based on guaranteed state estimators, which rely on set-based representation of the uncertainty, reachability analysis, and set operations such as Minkowski sum and intersections. We design the algorithm in a way that most of the heavy computation can be done offline, and only simpler linear-algebraic operations are performed online. This is a stepping stone to understanding, predicting, quantifying, and scheduling ETC on a more general and challenging control setting.

Finally, in Chapter 10 we present our conclusions. We summarize our key findings, present opportunities for improvement, and discuss interesting future work directions.

1.5 Scope

Before entering the next chapter, it is important to comment on the scope of this work. This dissertation is dedicated to ETC/STC of linear time-invariant (LTI) systems. Some of the results we obtain are applicable for nonlinear systems, but for traffic modeling aimed at nonlinear systems the reader is referred to [28–30]. Similarly, except for Chapter 9, we have considered exclusively the nominal case where the system is not affected by disturbances. While the traffic models we create can be extended to perturbed systems, an available approach for linear systems with bounded disturbances is given in [23] and for nonlinear systems in [30], and for linear systems with stochastic disturbances in [31].

Another important aspect is that we consider the problem of analyzing and manipulating the traffic generated by a *given controller*. That is, we are not allowed to alter the controller or even the triggering condition. This is based on the principle of *separation of concerns*: the control design is a job to be executed regardless of the cyber-physical implementation, and it typically generates a continuous-time controller. Then, an event-

triggering mechanism is designed in order to ensure that a good enough fraction of the continuous-time performance is met, an approach called *emulation*. Finally, the sampling performance can be evaluated given the triplet plant–controller–event, or a scheduler can be designed for such triplets. Obviously the separation of concerns has the drawback of decentralized approaches in general, i.e., missing the opportunity of optimizing both control and sampling performance together. However, it allows that changes in the cyber-physical implementation do not require a full redesign of the controller, keeping the tasks somewhat independent. The problem of designing controllers together with events is called *co-design*, and has its own large body of literature, see, e.g., [4, 32, 33].

1.6 Other contributions

In addition to the work here reported, we have worked in two main fronts. The first is the development of a tool that automates much of the work needed to abstract ETC traffic models, evaluate their performance, and design schedulers and improved sampling strategies — all the work reported in Chapters 3–8. This open-source tool is called `ETCetera` [34] and is available in `https://gitlab.tudelft.nl/sync-lab/ETCetera`. This tool was a joint effort with the other researchers in the ERC project that funded this research, and further contains abstraction methods for nonlinear systems [28, 30], its own scheduling problem solver [35], interfaces with UPPAAL Tiga [36] for scheduling with timed-game automata (see Chapter 3), and simulation capabilities.

The second front is the design of specialized network stacks for ETC. In a collaboration with the University of Trento and the Bruno Kessler Foundation, we have developed the Wireless Control Bus [37], a network stack that utilizes concurrent transmissions to minimize energy usage in wireless control systems. Concurrent transmissions allow for constructive interference, which is particularly useful to turn otherwise competing sensors flagging their events into a fast flooding of events for triggering a sensor data dissemination phase. The stack is especially suitable for large-scale systems (in terms of both physical dimensions and numbers of sensors and actuators connected to the network), for which its tailor-made decentralized PETC protocol achieves massive reductions in radio energy usage.

2

Mathematical Notation and Preliminaries

In this chapter we provide the notation (Section 2.1) and mathematical preliminaries that are used across most of the chapters of this thesis. Section 2.2 presents the notion of generalized transition systems from [17], weighted systems for quantitative verification [19], and several definitions and fundamental results about abstractions. Then, we provide a succinct introduction to event-triggered control (ETC) and self-triggered control (STC) for linear time-invariant systems in Section 2.4. Finally, the traffic modeling of ETC systems using generalized transition systems is presented in Section 2.5.

Notions that are only used in one specific chapter are presented locally.

2.1 Notation

This thesis applies a standard for font usage in mathematical symbols, with a few exceptions to comply with tradition. Letters in normal italicized fonts (e.g., x) are used for scalar variables or parameters, or scalar-valued functions, or elements of a set in a more abstract way; bold lower-case letters (e.g., $\boldsymbol{x}$) are used for vectors or vector-valued functions, and bold upper-case letters (e.g., $\boldsymbol{X}$) for matrix and matrix-valued functions; and calligraphic letters (e.g., $\mathcal{X}$) are used for sets or set-valued functions. Signals of time (such as solutions to differential equations) are denoted with Greek letters, while points in some set are denoted with (when possible, corresponding) Roman letters (e.g., $\xi(t_0) = x$).

We denote by $\mathbb{N}_0$ the set of natural numbers including 0, $\mathbb{N} \coloneqq \mathbb{N}_0 \setminus \{0\}$, $\mathbb{N}_{\leq n} \coloneqq \{1, 2, ..., n\}$, by $\mathbb{Q}$ the set of rational numbers, by $\mathbb{R}$ the set of real numbers, with $\mathbb{R}_+ \coloneqq \{x \in \mathbb{R} \mid x \geq 0\}$, i.e., the set of non-negative reals, and by $\mathbb{C}$ the set of complex numbers. The function $\lfloor a \rfloor$ denotes the largest integer not larger than $a \in \mathbb{R}$, while $\lceil a \rceil$ denotes the smallest integer not smaller than a. For a complex number $z \in \mathbb{C}$, z^* denotes its complex conjugate, $\arg z$ denotes its argument, and $\Im(z)$ denotes its imaginary part.

For a vector $\boldsymbol{x} \in \mathbb{R}^n$ we denote by $|\boldsymbol{x}|$ its Euclidean norm. For a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we write $\mathrm{Tr}(\boldsymbol{A})$ to denote its trace, $|\boldsymbol{A}|$ to denote its 2-induced norm. The set of eigenvalues of $\boldsymbol{A}$ is denoted by $\lambda(\boldsymbol{A}) \subset \mathbb{C}^n$; $\lambda_{\max}(\boldsymbol{A}), \lambda_{\min}(\boldsymbol{A})$, and $\lambda_i(\boldsymbol{A})$ are its largest, smallest, and i-th largest-in-magnitude eigenvalues, respectively. The sets $\mathbb{S}^n, \mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ are the sets of

symmetric, positive definite, and positive semi-definite real n by n matrices, respectively. For $\boldsymbol{P} \in \mathbb{S}^n$, we write $\boldsymbol{P} \succ \boldsymbol{0}$ ($\boldsymbol{P} \succeq \boldsymbol{0}$) if $\boldsymbol{P}$ is positive definite (semi-definite).

For a set $\mathcal{X} \subseteq \Omega$, we denote by $|\mathcal{X}|$ its cardinality, by $\mathrm{cl}(\mathcal{X})$ its closure, by $\partial\mathcal{X}$ its boundary, and by $\bar{\mathcal{X}}$ its complement, i.e., $\Omega \setminus \mathcal{X}$. We apply a function $f : \mathcal{X} \to \mathcal{Y}$ to a set $\mathcal{A} \subseteq \mathcal{X}$ the usual way, $f(\mathcal{A}) \coloneqq \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{A}\}$. A set $\mathcal{Q} \subseteq \mathbb{R}^n$ is called *homogeneous* if $\boldsymbol{x} \in \mathcal{Q} \implies \lambda\boldsymbol{x} \in \mathcal{Q}, \forall \lambda \in \mathbb{R} \setminus \{0\}$. For a (binary) relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$, its inverse is denoted as $\mathcal{R}^{-1} = \{(x_b, x_a) \in \mathcal{X}_b \times \mathcal{X}_a \mid (x_a, x_b) \in \mathcal{R}\}$. Every function $f : \mathcal{X}_a \to \mathcal{X}_b$ can be read as a relation, namely $\{(x_a, x_b) \in \mathcal{X}_a \times \mathcal{X}_b \mid x_b = f(x_a)\}$. When $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{X}$ is an equivalence relation on $\mathcal{X}$, i.e., it is reflexive, symmetric and transitive, we denote by $[x] \coloneqq \{x' \mid (x, x') \in \mathcal{R}\}$ the equivalence class of $x \in \mathcal{X}$, and by $\mathcal{X}/\mathcal{R}$ the set of all equivalent classes. Finally, we denote by $\pi_{\mathcal{R}}(\mathcal{X}_a) \coloneqq \{x_b \in \mathcal{X}_b \mid (x_a, x_b) \in \mathcal{R} \text{ for some } x_b \in \mathcal{X}_b\}$ the natural projection of $\mathcal{X}_a$ onto $\mathcal{X}_b$.

For a sequence $\sigma \coloneqq \{x_i\}_{i=0}^{n}$, we denote by $|\sigma| = n + 1$ its length. We often use a string notation for sequences, e.g., $\sigma = abc$ reads $\sigma(0) = x_0 = a, \sigma(1) = x_1 = b, \sigma(2) = x_2 = c$. Powers and concatenations work as expected, e.g., $\sigma^2 = \sigma\sigma = abcabc$. In particular, σ^ω denotes the infinite repetition of σ. We denote by $\mathcal{X}^+$ (resp. $\mathcal{X}^\omega$) the sets of finite (resp. infinite) sequences with elements on $\mathcal{X}$. When the sub- and superscripts are absent, $\{a_i\} \coloneqq a_0, a_1, a_2, \ldots$ denotes an infinite (on the right) sequence.

We say that an autonomous time-invariant system $\dot{\xi}(t) = f(\xi(t))$ is globally exponentially stable (GES) if there exist $M < \infty$ and $b > 0$ such that every solution of the system satisfies $|\xi(t)| \leq M\mathrm{e}^{-bt}|\xi(0)|$ for every initial state $\xi(0)$; moreover, we call b a decay rate estimate of the system. When needed to avoid ambiguity, we use $\xi_{\boldsymbol{x}}(t)$ to denote a trajectory from initial state $\xi(0) = \boldsymbol{x}$.

2.2 Abstractions and transition Systems

Throughout this dissertation, we use the term *abstraction* to denote a simpler system (typically finite-state) that captures desired properties of the *concrete* system (typically infinite-state) we study. Finite-state abstractions are sometimes called finite-state approximations or symbolic models. The process of *abstracting* requires a formalism that allows one to establish a relation between the concrete and abstract systems. We choose to use the framework of [17], where the concept of Generalized Transition System is presented:

Definition 2.1 (Generalized Transition System [17])**.** *A generalized transition system $\mathcal{S}$ is a 6-tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, H)$ where:*

- *$\mathcal{X}$ is the set of states,*
- *$\mathcal{X}_0 \subseteq \mathcal{X}$ is the set of initial states,*
- *$\mathcal{U}$ is the set of inputs,*
- *$\mathcal{E} \subseteq \mathcal{X} \times \mathcal{U} \times \mathcal{X}$ is the set of edges (or transitions),*
- *$\mathcal{Y}$ is the set of outputs, and*
- *$H : \mathcal{X} \to \mathcal{Y}$ is the output map.*

We often refer to generalized transition systems simply as systems.

A system is said to be *finite- (infinite-) state* when the cardinality of $\mathcal{X}$ is finite (infinite), and it is said to be just *finite* if both $\mathcal{X}$ and $\mathcal{U}$ are finite. A transition in $\mathcal{E}$ is denoted by a triplet (x, u, x'), and we often use $x \xrightarrow{u} x'$ to denote $(x, u, x') \in \mathcal{E}$. The set $U(x) \coloneqq \{u \mid (x, u, x') \in \mathcal{E}\}\}$ denotes the actions available at state x. We define $\mathrm{Post}(x) \coloneqq \{x' \mid (x, u, x') \in$

$\mathcal{E}\}$ as the set of states that can be reached from x in one step, and by $\mathrm{Pre}(x,u) \coloneqq \{x' \in \mathcal{X} \mid x' \xrightarrow{u} x\}$ the states from which, by applying u, reach x in one step. When the system $\mathcal{S}$ is not clear from context, we use, respectively, $x \xrightarrow{u}_{\mathcal{S}} x', U^{\mathcal{S}}(x), \mathrm{Post}^{\mathcal{S}}(x,u)$, and $\mathrm{Pre}^{\mathcal{S}}(x,u)$.

System $\mathcal{S}$ is said to be *non-blocking* if $\forall x \in \mathcal{X}, \mathrm{Post}^{\mathcal{S}}(x) \neq \emptyset$, and *deterministic* if for every $x \in \mathcal{X}$ and $u \in U(x)$, we have $|\mathrm{Post}(x,u)| = 1$. For a state $x \in \mathcal{X}$, we denote by $\mathcal{S}(x) \coloneqq \{\mathcal{X}, \{x\}, \mathcal{U}, \mathcal{Y}, \mathcal{E}, H\}$ the system $\mathcal{S}$ initialized at x.

We call $x_0 u_0 x_1 u_1 x_2 \ldots$ an *infinite internal behavior* [20], or *run* of $\mathcal{S}$ if $x_0 \in \mathcal{X}_0$ and $(x_i, u_i, x_{i+1}) \in \mathcal{E}$ for all $i \in \mathbb{N}$, $y_0 y_1 \ldots$ its corresponding *infinite external behavior*, or *trace*, if $H(x_i) = y_i$ for all $i \in \mathbb{N}$. We denote by $B_{\mathcal{S}}(r)$ the external behavior from a run $r = x_0 u_0 x_1 u_1 x_2 \ldots$ (in the case above, $B_{\mathcal{S}}(r) = y_0 y_1 \ldots$), by $\mathcal{B}^l_x(\mathcal{S})$ (resp. $\mathcal{B}^+_x(\mathcal{S})$ and $\mathcal{B}^\omega_x(\mathcal{S})$) the set of all l-long (resp. finite and infinite) external behaviors of $\mathcal{S}$ starting from state x, and by $\mathcal{B}^l(\mathcal{S}) \coloneqq \bigcup_{x \in \mathcal{X}_0} \mathcal{B}^l_x(\mathcal{S})$ (resp. $\mathcal{B}^+(\mathcal{S}) \coloneqq \bigcup_{x \in \mathcal{X}_0} \mathcal{B}^+_x(\mathcal{S})$ and $\mathcal{B}^\omega(\mathcal{S}) \coloneqq \bigcup_{x \in \mathcal{X}_0} \mathcal{B}^\omega_x(\mathcal{S})$) the set of all l-long (resp. finite and infinite) external behaviors of $\mathcal{S}$. Finally, $\mathcal{B}^{\leq n}(\mathcal{S})$ is the set of all behaviors of length $\leq n$. A state x is called *reachable* if there exists a run r of $\mathcal{S}$ containing x; throughout this thesis, we assume every state is reachable, which does not affect generality as one can always remove unreachable states from a transition system without changing its behavior.

When studying behaviors of finite systems, we can distinguish their transient components: a finite sequence β is called *transient* if there exists a finite l such that $\gamma\beta\alpha \in \mathcal{B}^\omega(\mathcal{S})$ implies that $|\gamma| \leq l$ and β is not a subsequence of α; equivalently, β cannot occur infinitely often in any infinite behavior of $\mathcal{S}$.

A system is called *autonomous* if $U(x)$ is a singleton for all $x \in X$, in which case we shorten the notation of $\mathcal{S}$ by the 5-tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H)$; in this case $\mathcal{E}$ is a subset of $\mathcal{X} \times \mathcal{X}$, and $x \longrightarrow x'$ means $(x, x') \in \mathcal{E}$. Likewise, the notation for runs is simplified to $x_0 x_1 x_2 \ldots$.

If $\mathcal{S}$ is both finite-state and autonomous, we can associate a directed graph, or digraph, G with it, where states are nodes and an edge $x \longrightarrow x'$ exists if $(x, x') \in \mathcal{E}$. Every digraph has an associated $(0,1)$-matrix, the incidence matrix $\boldsymbol{T}$, obtained by attributing an index i to each node; then $T_{ij} = 1$ if $x_i \longrightarrow x_j$, $T_{ij} = 0$ otherwise. We say that $\boldsymbol{T}$ is *the incidence matrix of* $\mathcal{S}$. A digraph is said to be *strongly connected* if there is a path from every node to every other node. A *strongly connected component* (SSC) of G is a maximal subgraph of G that is strongly connected.

2.2.1 Simulation, behavioral equivalence, and abstractions

Before presenting abstraction methods, we introduce several formal notions of relation between two different systems. The first and perhaps most celebrated one is that of *simulation* [38, 39], which requires the definition of a binary relation among the state spaces of the two systems:

Definition 2.2 (Simulation Relation [17])**.** *Consider two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$. A relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$ if the following conditions are satisfied:*

i) for every $x_{a0} \in \mathcal{X}_{a0}$, there exists $x_{b0} \in \mathcal{X}_{b0}$ with $(x_{a0}, x_{b0}) \in \mathcal{R}$;
ii) for every $(x_a, x_b) \in \mathcal{R}, H_a(x_a) = H_b(x_b)$;
iii) for every $(x_a, x_b) \in \mathcal{R}$, we have that $(x_a, u_a, x'_a) \in \mathcal{E}_a$ implies the existence of $(x_b, u_b, x'_b) \in \mathcal{E}_b$ satisfying $(x'_a, x'_b) \in \mathcal{R}$.

A simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$ is denoted by $\mathcal{S}_a \preceq \mathcal{S}_b$. Essentially, a simulation relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$ captures which states of $\mathcal{S}_a$ are simulated by which states of $\mathcal{S}_b$: for the right state selection, their outputs are the same; and every transition in $\mathcal{S}_a$ leads to a state whose output can also be attained in $\mathcal{S}_b$ after a single transition. It is important to notice, however, that there might be transitions in $\mathcal{S}_b$ that lead to states that are not related to the ones attained in $\mathcal{S}_a$. When using simulation relations to model the behavior of a system, these transitions are called spurious transitions.

If $\mathcal{S}_a \preceq \mathcal{S}_b$, it becomes clear that any sequence of outputs from $\mathcal{S}_a$ can be generated by $\mathcal{S}_b$; the converse is not true, unless there is in fact a bisimulation:

Definition 2.3 (Bisimulation [17])**.** *Consider two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$. $\mathcal{S}_a$ is said to be bisimilar to $\mathcal{S}_b$, denoted $\mathcal{S}_a \cong \mathcal{S}_b$, if there exists a relation $\mathcal{R}$ such that:*

- *$\mathcal{R}$ is a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$;*
- *$\mathcal{R}^{-1}$ is a simulation relation from $\mathcal{S}_b$ to $\mathcal{S}_a$.*

Weaker but important relations associated with simulation and bisimulation are, respectively, *behavioral inclusion* and *behavioral equivalence*:

Definition 2.4 (Behavioral inclusion and equivalence [17])**.** *Consider two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$. We say that $\mathcal{S}_a$ is* behaviorally included *in $\mathcal{S}_b$, denoted by $\mathcal{S}_a \preceq_{\mathcal{B}} \mathcal{S}_b$, if $\mathcal{B}^\omega(\mathcal{S}_a) \subseteq \mathcal{B}^\omega(\mathcal{S}_b)$. In case $\mathcal{B}^\omega(\mathcal{S}_a) = \mathcal{B}^\omega(\mathcal{S}_b)$, we say that $\mathcal{S}_a$ and $\mathcal{S}_b$ are* behaviorally equivalent*, which is denoted by $\mathcal{S}_a \cong_{\mathcal{B}} \mathcal{S}_b$.*

(Bi)simulations imply behavioral inclusion (equivalence):

Theorem 2.1 ([17])**.** *Given two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$:*

- *$\mathcal{S}_a \preceq \mathcal{S}_b \implies \mathcal{S}_a \preceq_{\mathcal{B}} \mathcal{S}_b$;*
- *$\mathcal{S}_a \cong \mathcal{S}_b \implies \mathcal{S}_a \cong_{\mathcal{B}} \mathcal{S}_b$.*

The main difference between simulation and behavioral inclusion is that, in the former, a relationship between states must be established: every transition in the concrete system must have at least one matching transition in the abstraction leading to related states. Behavioral inclusion is oblivious to state-based descriptions of a system: all one needs is that all traces observed in the concrete system can also be observed in the abstraction.

Having the relation definitions in place, we can now present existing abstraction methods. The first is the construction of *quotient systems*:

Definition 2.5 (Quotient System [17])**.** *Consider a system $\mathcal{S} = (\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, H)$ and let $\mathcal{R}$ be an equivalence relation on $\mathcal{X}$ such that $(x, x') \in \mathcal{R} \implies H(x) = H(x')$. The quotient of $\mathcal{S}$ by $\mathcal{R}$, denoted by $\mathcal{S}_{/\mathcal{R}}$, is the system $(\mathcal{X}_{/\mathcal{R}}, \mathcal{X}_{/\mathcal{R}0}, \mathcal{U}, \mathcal{E}_{/\mathcal{R}}, \mathcal{Y}, H_{/\mathcal{R}})$ consisting of*

- *$\mathcal{X}_{/\mathcal{R}} = \mathcal{X}/\mathcal{R}$;*
- *$\mathcal{X}_{/\mathcal{R}0} = \{x_{/\mathcal{R}} \in \mathcal{X}_{/\mathcal{R}} \mid x_{/\mathcal{R}} \cap \mathcal{X}_0 \neq \emptyset\}$;*
- *$(x_{/\mathcal{R}}, u, x'_{/\mathcal{R}}) \in \mathcal{E}_{/\mathcal{R}}$ if there exists $(x, u, x') \in \mathcal{E}$ with $x \in x_{/\mathcal{R}}$ and $x' \in x'_{/\mathcal{R}}$;*
- *$H_{/\mathcal{R}}(x_{/\mathcal{R}}) = H(x)$ for some $x \in x_{/\mathcal{R}}$.*

Building a quotient system is fundamentally aggregating states of the original system that produce the same output, and then determining the transitions so that every possible transition of the original system is reproduced in the quotient (symbolic) system. By construction, $\mathcal{S} \preceq \mathcal{S}_{/\mathcal{R}}$.

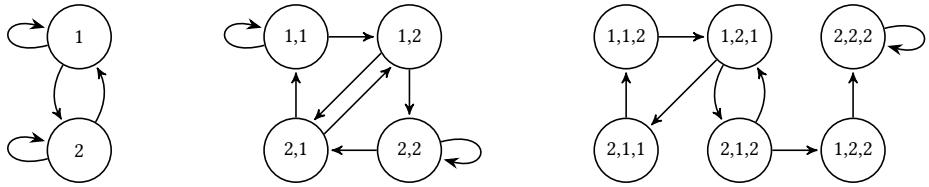


Figure 2.1: Example of l-complete PETC traffic models, for $l = 1$ (left), $l = 2$ (middle), and $l = 3$ (right).

A second way building an abstraction is based on behavioral inclusion, through a so-called *l-complete model*. Here we adapt the original definitions from [18, 40] for autonomous systems:

Definition 2.6 (Strongest l-complete abstraction (adapted from [18, 40]))**.** *Consider an autonomous transition system $\mathcal{S} := (\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H)$, and let $\mathcal{X}_l \subseteq \mathcal{Y}^l$ be the set of all l-long subsequences of all behaviors in $\mathcal{B}^\omega(\mathcal{S})$. Then, the system $\mathcal{S}_l = (\mathcal{X}_l, \mathcal{B}^l(\mathcal{S}), \mathcal{E}_l, \mathcal{Y}, H)$ is called the* strongest l-complete abstraction (SlCA) *of $\mathcal{S}$, where*

- $\mathcal{E}_l = \{(y\sigma, \sigma y') \mid y, y' \in \mathcal{Y}, \sigma \in \mathcal{Y}^{l-1}, y\sigma, \sigma y' \in \mathcal{X}_l\}$.
- $H(y\sigma) = y$.

The idea behind the SlCA is to encode the states as the l-long behavior fragments of the concrete system. The transitions follow the "*domino rule*": e.g., if the last 4 elements of the behavior up to a given time are $abcd$, after one step the first 3 elements must be bcd; thus, from having observed $abcd$ alone, a transition from state $abcd$ can lead to any state starting with bcd. Finally, the output of a state is its first element.

To illustrate how successive l-complete approximations of a system operate, consider a system $\mathcal{S}$ with behavior set $\{2^\omega, 12^\omega, 212^\omega, (112)^\omega, (121)^\omega, (211)^\omega\}$. Fig. 2.1 presents its 1-, 2-, and 3-complete abstractions; as one can see, $\mathcal{S}_1$ and $\mathcal{S}_2$ have the trivial set of all possible behaviors over the set $\mathcal{Y}$, but $\mathcal{B}^\omega(\mathcal{S}_3)$ is smaller, closer to the concrete behavior set. That is, we have $\mathcal{B}^\omega(\mathcal{S}) \subseteq \mathcal{B}^\omega(\mathcal{S}_3) \subseteq \mathcal{B}^\omega(\mathcal{S}_2) \subseteq \mathcal{B}^\omega(\mathcal{S}_1)$, and in this example $\mathcal{S}_3$ has fewer spurious behaviors than $\mathcal{S}_2$ and $\mathcal{S}_1$.

Remark 2.1. *Here we directly present a realization of the SlCA as a transition system according to Def. 2.1. Schmuck et al. [40] showed that different realizations exist for the SlCA of a system, depending on whether you encode states based on past, future, or a mix of past and future observations. In Def. 2.6, we pick the one based on future observations, which simplifies the encoding (all states are l-long sequences without the need for "no-output yet" characters, see [18]), and is the tightest abstraction from a simulation relation perspective (see [40, Thm. 5]).*

In [40, Theorem 9], it is concluded that a quotient-based approach as that of Def. 2.5 can create an abstraction bisimilar to the SlCA in case the concrete system is future-unique, which is the case of deterministic autonomous systems. Thus, we shall use the term l-complete for quotient-based abstractions whose states represent the next l outputs of their related concrete states. With this in mind, the following fact is a direct consequence of Theorems 6 and 7 from [40].

Proposition 2.1. *Consider a deterministic autonomous system $\mathcal{S}$ and its SlCA $\mathcal{S}_l$ from Definition 2.6, for some $l \geq 1$. Then, $\mathcal{S} \preceq \mathcal{S}_{l+1} \preceq \mathcal{S}_l$.*

Prop. 2.1 gives that l-complete abstractions provide a framework of obtaining simulations and their refinements. It is not a surprising result, since encoding states with more elements of the concrete system's behavior constrains the set of behaviors it can generate, even if it increases the number of states in the abstraction.

Remark 2.2. *Bisimulation is obtained when $\mathcal{S}_{l+1} = \mathcal{S}_l$ (modulo the names of the states); it is trivial to see that this only happens when abstracting an autonomous deterministic system if the abstraction is deterministic. In addition,* $\lim_{l\to\infty} \mathcal{S}_l \cong_{\mathcal{B}} \mathcal{S}$.

2.2.2 Quantitative automata

While much of the field of formal methods in control is concerned with qualitative analysis, such as establishing safety, stability, and reachability, often quantitative computations are of interest: examples are computing the decay rate, the maximum overshoot, or our case, the average sampling period of an ETC system. In [19], Chatterjee et al. establish a comprehensive framework for quantitative problems on finite-state systems, from which we borrow some definitions and results, while adjusting notation to keep consistency with the previous section.

Definition 2.7 (Weighted transition system (adapted from [19]))**.** *A* weighted transition system *(WTS) $\mathcal{S}$ is the tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, H, \gamma)$, where*

- $(\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, H)$ *is a* non-blocking *transition system;*
- $\gamma : \mathcal{E} \to \mathbb{Q}$ *is the* weight function.

The notation adjustment we have made is including outputs to comply with Tabuada's transition systems.

We denote by $\gamma_{\mathcal{S}}(r)$ the value trace of $r = x_0 u_0 x_1 \ldots$ (e.g., $\gamma_{\mathcal{S}}(r) = v_0 v_1 \ldots$); when the system is clear from context, we omit the subscript, abusing the notation of γ to sequences. We use $\gamma_i(r)$ for the i-th element of $\gamma(r)$. We denote by $\mathcal{V}_x^\omega(\mathcal{S})$ the set of all possible value traces of $\mathcal{S}$ starting from state x, and by $\mathcal{V}^\omega(\mathcal{S}) \coloneqq \bigcup_{x\in\mathcal{X}_0} \mathcal{V}_x^\omega(\mathcal{S})$ the set of all possible value traces of $\mathcal{S}$. Again abusing notation, we denote by $\gamma(\mathcal{E})$ the set of all possible weight valuations of $\mathcal{S}$, which is guaranteed to be finite if $\mathcal{S}$ is finite. In some cases, the weights are state-dependent instead of transition-dependent; we call these systems *simple* WTSs:

Definition 2.8 (Simple WTS)**.** *A WTS is called* simple *if for all $(x, u, x') \in \mathcal{E}$, we have that $\gamma(x, u, x') = H(x)$.*

If $\mathcal{S}$ is a simple WTS, it holds trivially that $\mathcal{V}(\mathcal{S}) = \mathcal{B}^\omega(\mathcal{S})$.

A *value function* $\mathrm{Val} : \mathbb{Q}^\omega \to \mathbb{R}$ attributes a value to an infinite sequence of weights $v \coloneqq v_0 v_1 \ldots$. Among the most well-studied value functions, the ones of our interest are

$$\mathrm{LimInf}(v) \coloneqq \liminf_{n\to\infty} v_i,$$

$$\mathrm{LimSup}(v) \coloneqq \limsup_{n\to\infty} v_i,$$

$$\mathrm{LimAvg}(v) \coloneqq \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} v_i.$$

Similarly, for a finite sequence $\sigma \coloneqq \{v_i\}_{i=0}^n$, let $\mathrm{Avg}(\sigma) \coloneqq \frac{1}{n+1}\sum_{i=0}^n v_i$ (the inf counterpart is simply $\min(\sigma)$. For each value function we can define the following *values* of a value trace set $\mathcal{V} \subseteq 2^{\mathbb{N}\to\mathbb{Q}}$,

$$\begin{aligned}
\mathrm{ILI}(\mathcal{V}) &\coloneqq \inf\left\{ \liminf_{i\to\infty} y_i \;\middle|\; \{y_i\} \in \mathcal{V} \right\},\\
\mathrm{SLS}(\mathcal{V}) &\coloneqq \sup\left\{ \limsup_{i\to\infty} y_i \;\middle|\; \{y_i\} \in \mathcal{V} \right\},\\
\mathrm{ILA}(\mathcal{V}) &\coloneqq \inf\left\{ \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^n y_i \;\middle|\; \{y_i\} \in \mathcal{V} \right\},\\
\mathrm{SLA}(\mathcal{V}) &\coloneqq \sup\left\{ \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^n y_i \;\middle|\; \{y_i\} \in \mathcal{V} \right\}.
\end{aligned} \tag{2.1}$$

For a value $V \in \{\mathrm{ILI}, \mathrm{SLS}, \mathrm{ILA}, \mathrm{SLA}\}$, we often use the shorthand notation $V(\mathcal{S}) \coloneqq V(\mathcal{V})$.

The following result, extracted from [19, Theorem 3] and its proof, gives that these values are computable on finite WTSs:

Theorem 2.2. *Given a finite-state WTS $\mathcal{S}$,*

1) $\mathrm{ILI}(\mathcal{S})$ *can be computed in $\mathcal{O}(|\mathcal{X}|+|\mathcal{E}|)$; moreover, there exists $x \in \mathcal{X}$ such that $H(x) = \mathrm{ILI}(\mathcal{S})$ and x belongs to an SCC of the graph defined by $\mathcal{S}$.*
2) $\mathrm{ILA}(\mathcal{S})$ *can be computed in $\mathcal{O}(|\mathcal{X}||\mathcal{E}|)$. Moreover, system $\mathcal{S}$ admits a cycle $x_0x_1\ldots x_k$ satisfying $x_i \longrightarrow x_{i+1}, i < k$, and $x_k \longrightarrow x_0$, s.t. the run $r = (x_0x_1\ldots x_k)^\omega$ satisfies*

$$\liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^n \gamma_i(r)) = \mathrm{ILA}(\mathcal{S}).$$

It is easy to see that, for a finite WTS $\mathcal{S}$, we have that $\mathrm{ILA}(\mathcal{S}) = -\mathrm{SLA}(-\mathcal{S})$, where we denote by $-\mathcal{S}$ the WTS $\mathcal{S}$ with all of its weights negated; hence the results of Theorem 2.2 apply equally to SLS and SLA.

Remark 2.3. *The algorithm for computing* $\mathrm{ILI}(\mathcal{S})$ *and* $\mathrm{SLS}(\mathcal{S})$ *is the same as the one to determine Büchi acceptance, and consists of computing SSCs and performing reachability to those [19]. The cycle mentioned in Theorem 2.2 is a* minimum average cycle *(MAC), or* smallest-in-average cycle *(SAC), of the weighted digraph defined by $\mathcal{S}$. The algorithm to compute the value is due to [41], which also detects reachable SCCs and employs dynamic programming on those. The cycle can be recovered in $\mathcal{O}(|\mathcal{X}|)$ using the algorithm in [42].*

2.2.3 Abstractions for control synthesis

Whenever the interest is in control design, normal simulation relations as in Section 2.2.1 are not sufficient. Instead, [17] proposes the use of *alternating simulation relations* [43] for this end.[1] Here we make a small adaptation of the definition of Tabuada:

[1] The concept of alternating simulations in [43] was proposed for multi-player games on structures called alternating transition systems. It was later simplified by Tabuada for a two-player game, where the *controller* chooses actions in $\mathcal{U}$ to meet some specification against an antagonist *environment* that chooses the transitions.

Definition 2.9 (Alternating simulation relation)**.** *Consider two weighted systems (Def. 2.7) $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{U}_a \subseteq \mathcal{U}_b$ and $\gamma_a(\mathcal{E}_a) = \gamma_b(\mathcal{E}_b)$. A relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is a* alternating simulation relation *(ASR) from $\mathcal{S}_a$ to $\mathcal{S}_b$ if the following conditions are satisfied:*

i) $\forall x_{b0} \in \mathcal{X}_{b0}\ \exists x_{a0} \in \mathcal{X}_{a0}$ *such that* $(x_{a0}, x_{b0}) \in \mathcal{R}$*;*

ii) $\forall (x_a, x_b) \in \mathcal{R}$*, it holds that* $H(x_a) = H(x_b)$*;*

iii) $\forall (x_a, x_b) \in \mathcal{R}, \forall u_a \in U_a(x_a)\ \exists u_b \in U_b(x_b)\ \forall x_b' \in \mathrm{Post}^{\mathcal{S}_b}(x_b, u_b), \exists x_a' \in \mathrm{Post}^{\mathcal{S}_a}(x_a, u_a)$ *such that* $(x_a', x_b') \in \mathcal{R}$*.*

When an ASR from $\mathcal{S}_a$ to $\mathcal{S}_b$ exists, we say that $\mathcal{S}_b$ is an alternating simulation *(AS) of $\mathcal{S}_a$, denoting it by $\mathcal{S}_a \preceq_{\mathrm{AS}} \mathcal{S}_b$. When using a specific relation $\mathcal{R}$, we use the notation $\mathcal{S}_a \preceq_{\mathcal{R}} \mathcal{S}_b$.*

It is easy to see that if two relations $\mathcal{R}_1$ and $\mathcal{R}_2$ satisfy $\mathcal{S}_a \preceq_{\mathcal{R}_1} \mathcal{S}_b$ and $\mathcal{S}_a \preceq_{\mathcal{R}_2} \mathcal{S}_b$, then $\mathcal{S}_a \preceq_{\mathcal{R}_1 \cup \mathcal{R}_2} \mathcal{S}_b$. The union of all ASRs from $\mathcal{S}_a$ to $\mathcal{S}_b$ is called the *maximal alternating simulation relation* from $\mathcal{S}_a$ to $\mathcal{S}_b$.

Intuitively, given $\mathcal{S}_a$ and $\mathcal{S}_b$, an ASR from a $\mathcal{S}_a$ to $\mathcal{S}_b$, implies that every controller move of $\mathcal{S}_a$ can be "replicated" by the controller of $\mathcal{S}_b$ and every environment move of $\mathcal{S}_b$ can be "replicated" by that of $\mathcal{S}_a$. Informally, this means that the controller of $\mathcal{S}_b$ is at least as powerful as that of $\mathcal{S}_a$ and the environment of $\mathcal{S}_a$ is at least as powerful as that of $\mathcal{S}_b$. This interpretation is also behind our modification of the definition w.r.t. [17], where condition (i) is reversed: in our definition, the "environment" picks the initial state, so every initial state in $\mathcal{S}_b$ must be matched in $\mathcal{S}_a$. [2]

Remark 2.4. *Unlike in the simulation relation, when $\mathcal{S}_a \preceq_{\mathrm{AS}} \mathcal{S}_b$, we say that $\mathcal{S}_a$ (the left-hand side) is the abstraction.*

The importance of alternating simulations for control stems from the following fact: given any temporal-logic specification ϕ over the alphabet $\mathcal{Y}$, if $\mathcal{S}_a \preceq_{\mathrm{AS}} \mathcal{S}_b$, then the existence of a controller for $\mathcal{S}_a$ such that the closed-loop system satisfies ϕ implies that there exists a controller for $\mathcal{S}_b$ meeting the same specification; in fact, the strategy for $\mathcal{S}_a$ can be refined for $\mathcal{S}_b$. Moreover, for any specification ϕ, if $(x, x') \in \mathcal{R}$ and the controller can ensure ϕ from x, then it can ensure ϕ from x'; the symmetric notion holds: if the controller cannot ensure ϕ from x', then it cannot ensure it from x. Alternating simulation commutes with composition (as normal simulations do), making this notion suitable for control design of a composition of systems, such as the scheduling problem we tackle in Chapter 3. For a thorough exposition about these facts and how to synthesize controllers for several types of specifications we refer the reader to [17].

Like in the case of simulation, AS also admits a bisimulation notion:

Definition 2.10 (Alternating bisimulation)**.** *Two transition systems $\mathcal{S}_a$ and $\mathcal{S}_b$ are said to be* alternatingly bisimilar*, denoted by $\mathcal{S}_a \cong_{AS} \mathcal{S}_b$, if there is an ASR $\mathcal{R}$ from $\mathcal{S}_a$ to $\mathcal{S}_b$ such that its inverse $\mathcal{R}^{-1}$ is an ASR from $\mathcal{S}_b$ to $\mathcal{S}_a$.*

A relaxed notion w.r.t. bisimulation is that of equivalence:

[2] Note that Tabuada's definition and ours are not fundamentally different. In both cases, one could have a single initial state, and condition (i) of Def. 2.9 would be a consequence of condition (iii) by adding silent transitions from the initial state to the "real" initial state set; for Tabuada's definition, condition (i) would be derived by (iii) if instead the controller would have a different action for each of these transitions.

Definition 2.11 (Alternating simulation equivalence (ASE))**.** *Two transition systems $\mathcal{S}_a$ and $\mathcal{S}_b$ are said to be* alternating-simulation equivalent*, denoted by $\mathcal{S}_a \simeq_{\text{AS}} \mathcal{S}_b$, if there is an ASR $\mathcal{R}$ from $\mathcal{S}_a$ to $\mathcal{S}_b$ and an ASR $\mathcal{R}'$ from $\mathcal{S}_b$ to $\mathcal{S}_a$.*

2.3 Timed automata

Another class of transition systems for which a wide range of verification and synthesis problems can be computationally solved is that of timed automata. Timed automata are regular automata equipped with clocks, which are resettable real-valued variables measuring the passage of time. Let C be a finite set of said clocks, and consider binary relations $\bowtie \in \{<,\leq,=,\geq,>\}$. A clock constraint g is a conjunctive formula of atomic constraints $c \bowtie k, c \in C, k \in \mathbb{N}$. We denote by $\mathcal{B}(C)$ the set of all clock constraints.

Definition 2.12. *(Timed safety automaton, [44]). A* timed safety automaton *(TSA) is a tuple $\mathcal{A} = (\mathcal{L}, \mathcal{L}_0, \mathcal{U}, C, \mathcal{E}, I)$ where:*

- *$\mathcal{L}$ is the finite set of locations (or discrete states),*
- *$\mathcal{L}_0 \subseteq \mathcal{L}$ is the set of initial locations,*
- *$\mathcal{U}$ is the finite set of actions,*
- *C is the finite set of clocks,*
- *$\mathcal{E} \subseteq \mathcal{L} \times \mathcal{B}(C) \times \mathcal{U} \times 2^C \times \mathcal{L}$ is the set of edges (or transitions), and*
- *$I : \mathcal{L} \longrightarrow \mathcal{B}(C)$ assigns invariants to locations.*

A TSA is a system with both discrete (the locations) and continuous states (the clocks). All clocks increase value at the same rate, but transitions can reset the value of certain clocks. The system can change locations through edges, depending on the action taken and the clocks' values. We denote by $l \xrightarrow{g,a,r} l'$ the transition from $l \in \mathcal{L}$ to $l' \in \mathcal{L}$ under action $a \in \mathcal{U}$, with $r \subseteq C$ as the set of clocks reset when this transition is taken, and g over C as the guards that enabled the transition. *Invariants* of a location are the sufficient clock conditions for a transition to happen; in other words, the system is forced to leave the place l if a clock c violates any invariant $I(l)$. Symmetrically, a *guard* is a necessary condition for a transition to occur.

Timed game automata (TGA) extend TSA by partitioning the set of actions into controllable and uncontrollable. Controllable actions are decisions that the system operator can choose, while uncontrollable actions are taken independently of the system operator (e.g., by the environment or an opponent).

Definition 2.13. *(Timed game automaton, [44]). A* timed game automaton *is a tuple $\mathcal{A} = (\mathcal{L}, \mathcal{L}_0, \mathcal{U}_\text{c}, \mathcal{U}_\text{u}, C, \mathcal{E}, I)$ where:*

- *$(\mathcal{L}, \mathcal{L}_0, \mathcal{U}_\text{c} \cup \mathcal{U}_\text{u}, C, \mathcal{E}, I)$ is a TSA,*
- *$\mathcal{U}_\text{c}$ is the set of controllable actions,*
- *$\mathcal{U}_\text{u}$ is the set of uncontrollable actions, and*
- *$\mathcal{U}_\text{c} \cap \mathcal{U}_\text{u} = \emptyset$.*

The distinction between controllable and uncontrollable is paramount in our case. The scheduler can control when to sample, but not how the system will react to this choice.

To define a strategy, let $\mathcal{A}$ be a TGA, and $\mathcal{L}_\text{c} \subseteq \mathcal{L}$ be its set of locations, for which a controllable action exists. A strategy $S : \mathcal{L}_\text{c} \times C \longrightarrow 2^{\mathcal{U}_\text{c}}$ determines which actions can be taken depending on the TGA states. A deterministic strategy outputs a single action.

Finally, TGAs can be combined into a *network of timed game automata* (NTGA), which allows for modularity [44]. An NTGA consists of n TGAs $\mathcal{A}_i = (\mathcal{L}_i, \mathcal{L}_{i0}, \mathcal{U}_c, \mathcal{U}_u, \mathcal{C}, \mathcal{E}_i, I_i)$, where 1) uncontrollable actions take precedence over controllable actions, and 2) a location of the network, denoted as $\bar{l} := (l_1, ..., l_n)$, has its invariant $I(\bar{l}) = \wedge_i I_i(l_i)$. Most importantly, TGAs within an NTGA can have transitions influence each other through *synchronization channels*: for a channel a, the initiating transition is labeled a! and, when fired, all transitions labeled a? have to fire simultaneously.

2.4 Event-triggered and self-triggered control

In most of the chapters ahead—the exception being Chapter 9—, we consider the basic ETC architecture of Fig. 1.1 applied to a linear time-invariant (LTI) system using static linear state-feedback and sample-and-hold control [45]:

$$\begin{aligned}\dot{\xi}(t) &= \boldsymbol{A}\xi(t) + \boldsymbol{B}\hat{\upsilon}(t),\\ \upsilon(t) &= \boldsymbol{K}\hat{\xi}(t),\\ \xi(0) &= \hat{\xi}(0) = \boldsymbol{x}_0,\end{aligned}$$

where the state of the plant $\xi(t) \in \mathbb{R}^{n_x}$ is sampled at instants t_i, $\forall i \in \mathbb{N}$, and held constant for the controller, which makes the state signal used for control $\hat{\xi}$ satisfy $\hat{\xi}(t) = \xi(t_i), \forall t \in [t_i, t_{i+1})$. Likewise, the control input $\upsilon(t) \in \mathbb{R}^{n_u}$ is held in-between sampling instants, giving $\hat{\upsilon}(t) = \upsilon(t_i), \forall t \in [t_i, t_{i+1})$, where $\hat{\upsilon}$ is the control input signal that is in fact received by the plant. The matrices $\boldsymbol{A}, \boldsymbol{B}$ are the plant matrices and $\boldsymbol{K}$ is the control gain, all of which have appropriate dimensions. The time instant zero is chosen to the be the first sampling moment, hence $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$ is the initial plant state and initial sample. Note that we have neglected possible delays in transmission or control computation. For this case, the equations can simplified further:

$$\begin{aligned}\dot{\xi}(t) &= \boldsymbol{A}\xi(t) + \boldsymbol{B}\boldsymbol{K}\hat{\xi}(t),\\ \xi(0) &= \hat{\xi}(0) = \boldsymbol{x}_0.\end{aligned} \tag{2.2}$$

In ETC, a *triggering condition* determines the sequence of times t_i. In PETC, this condition is checked only periodically, with a fundamental checking period h. The sampling time t_{i+1} hence assumes the following form:

$$t_{i+1} = \inf\{t \in \mathcal{T} \mid t > t_i \text{ and } c(t - t_i, \xi(t), \hat{\xi}(t))\}, \tag{2.3}$$

where $c : \mathcal{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \longrightarrow \{\text{true}, \text{false}\}$ is the *triggering condition*, and $\mathcal{T}$ is the set of checking times: the standard one is $\mathcal{T} = \mathbb{R}_+$, which is used in [5], and hereafter called continuous ETC (CETC); the other one we often consider is PETC, where $\mathcal{T} = h\mathbb{N}$, with h being the *checking period*.

In this dissertation, we consider the family of *quadratic triggering conditions* from [11] with an additional maximum inter-event time condition below:

$$c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) := \begin{bmatrix}\boldsymbol{x}\\ \hat{\boldsymbol{x}}\end{bmatrix}^{\mathsf{T}} \boldsymbol{Q}(s) \begin{bmatrix}\boldsymbol{x}\\ \hat{\boldsymbol{x}}\end{bmatrix} > 0 \text{ or } \; s \leq \bar{\tau} \tag{2.4}$$

where $\boldsymbol{Q} : \mathcal{T} \longrightarrow \mathbb{S}^{2n_x}$ is the designed triggering matrix function (possibly constant), and $\bar{\tau}$ is the chosen maximum inter-event time. When $\mathcal{T} = h\mathbb{N}$, we assume for consistency that $\bar{\tau}/h \in \mathbb{N}$.

Remark 2.5. *Often a maximum $\bar{\tau}$ naturally emerges form an ETC triggering condition. If it ensures GES of the origin, this holds if all eigenvalues of $\boldsymbol{A}$ have non-negative real part or, more generally, if the control action satisfies $\boldsymbol{BKx} \neq \boldsymbol{0}$ for all $\boldsymbol{x}$ in the asymptotically stable subspace of $\boldsymbol{A}$. In such situations, $\boldsymbol{\xi}(t)$ does not converge to the origin without intermittent resampling. Still, one may want to set a smaller maximum inter-event time so as to establish a "heart beat" of the system.*

Many of the triggering conditions available in the literature can be written as in Eq. (2.4); the interested reader may refer to [11] for a comprehensive list of triggering and stability conditions.

Remark 2.6. *In STC, the sampling time t_{i+1} is chosen at the time instant t_i based on available information. Most commonly, STC uses the currently available sample $\boldsymbol{\xi}(t_i)$ to determine the* inter-sample time *(IST) s_i such that $t_{i+1} = t_i + s_i$. Clearly, given an ETC triggering condition c, an STC implementation could obtain the exact same behavior by computing t_{i+1} using Equation* (2.3)*; while this is trivial for PETC, this is not so computationally acceptable for CETC, as this would essentially involve integrating the plant forward and detect a zero crossing. Moreover, if the system is affected by disturbances, a precise prediction is no longer possible. We address the latter case in Chapter 9.*

2.5 ETC TRAFFIC MODELS

Throughout this dissertation, we are interested in modeling the traffic generated by (P)ETC, i.e., understanding how the inter-sample times evolve from different initial conditions. We start by constructing the *concrete* traffic model, which is infinite-state. First, note that $\boldsymbol{\xi}(t)$ is a function of $\hat{\boldsymbol{\xi}}(t) = \boldsymbol{\xi}(t_i)$ and the elapsed time $s \coloneqq t - t_i$:

$$\boldsymbol{\xi}(t_i + s) = \boldsymbol{M}(s)\boldsymbol{\xi}(t_i), \tag{2.5}$$
$$\boldsymbol{M}(s) \coloneqq \boldsymbol{A}_{\mathrm{d}}(s) + \boldsymbol{B}_{\mathrm{d}}(s)\boldsymbol{K} \coloneqq \mathrm{e}^{\boldsymbol{A}s} + \int_0^s \mathrm{e}^{\boldsymbol{A}t}\mathrm{d}t\boldsymbol{BK}.$$

Applying (2.5) with $s = t_{i+1} - t_i$, the *inter-event time* (IET, or *inter-sample time*, IST) $t_{i+1} - t_i$ is solely a function of the i-th sample $\boldsymbol{\xi}(t_i)$. Hence, we can define an IET function $\tau : \mathbb{R}^{n_x} \longrightarrow (0, \bar{\tau}] \cap \mathcal{T}$ that returns the value of $t_{i+1} - t_i$ given a sampled state $\boldsymbol{x} \in \mathbb{R}^{n_x}$.[3] It follows from Eqs. (2.3)–(2.5) that

$$\tau(\boldsymbol{x}) = \inf\left\{ s \in \mathcal{T} \mid \boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(s)\boldsymbol{x} > 0 \text{ or } s = \bar{\tau} \right\},$$
$$\boldsymbol{N}(s) \coloneqq \begin{bmatrix} \boldsymbol{M}(s) \\ \mathbf{I} \end{bmatrix}^{\mathsf{T}} \boldsymbol{Q}(s) \begin{bmatrix} \boldsymbol{M}(s) \\ \mathbf{I} \end{bmatrix}, \tag{2.6}$$

where $\mathbf{I}$ denotes the identity matrix. Thus, the event-driven evolution of sampled states can be compactly described by the recurrence

$$\boldsymbol{\xi}(t_{i+1}) = \boldsymbol{M}(\tau(\boldsymbol{\xi}(t_i)))\boldsymbol{\xi}(t_i) =: f(\boldsymbol{\xi}(t_i)), \tag{2.7}$$

[3] We assume the triggering condition prevents Zeno behavior, which is standard in ETC design.

which finally gives the following discrete-time model:

$$\begin{aligned} \boldsymbol{x}_{i+1} &= f(\boldsymbol{x}_i), \\ y_i &= \tau(\boldsymbol{x}_i), \end{aligned} \tag{2.8}$$

where $\boldsymbol{x}_i \coloneqq \boldsymbol{\xi}(t_i)$ and $f(\boldsymbol{x}) \coloneqq \boldsymbol{M}(\tau(\boldsymbol{x}))\boldsymbol{x}$. We call f the *sample map* and (2.8) the *sample system*, which is equipped with an output y that gives the associated inter-event time of $\boldsymbol{x}_i$: for a traffic model, this is the output of interest. We shall denote the sequence of outputs from Eq. (2.8) for a given initial state $\boldsymbol{x}_0$ by $\{y_i(\boldsymbol{x}_0)\}$.

Referring to Remark 2.6, one can replace τ from (2.6) for any other function that can be computed on the controller, which makes the model in (2.8) essentially the same for STC.

Eq. (2.8) can be described as a weighted generalized transition system:

Definition 2.14 (ETC traffic model)**.** *The ETC traffic model is the WTS* $\mathcal{S} = (\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H, \gamma)$ *where*

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_0 = \mathbb{R}^{n_x}; \\ \mathcal{E} &= \{(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{X} \mid \boldsymbol{x}' = f(\boldsymbol{x})\}; \\ \mathcal{Y} &= (0, \bar{\tau}] \cap \mathcal{T}; \\ H &= \tau; \\ \gamma(\boldsymbol{x}, \boldsymbol{x}') &= H(\boldsymbol{x}). \end{aligned} \tag{2.9}$$

We also include a weight function, which is equal to the output of the outbound state. This will be useful in the chapters where we work with quantitative results, such as Chapters 5–7. In the other chapters, the system is referred to simply by the 5-tuple without γ. Clearly, the ETC traffic model is a simple WTS.

For PETC it is more convenient to deal with the discrete time $k \in \mathbb{N}$, such that $k = t/h, \forall t \in \mathcal{T}$. The analogous of the function τ in discrete time is the function $\kappa \coloneqq \tau/h$, which admits the following expression:

$$\kappa(\boldsymbol{x}) = \min \left\{ k \in \{1, 2, \ldots \bar{k}\} \;\middle|\; \boldsymbol{x}^{\mathsf{T}} \boldsymbol{N}(hk)\boldsymbol{x} > 0 \text{ or } k = \bar{k} \right\}, \tag{2.10}$$

where $\bar{k} \coloneqq \bar{\tau}/h$ is the maximal-in-discrete-time IST.

Clearly, the PETC traffic model is finite-output:

Definition 2.15 (PETC traffic model)**.** *The PETC traffic model is the system* $\mathcal{S} = (\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H, \gamma)$ *where*

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_0 = \mathbb{R}^{n_x}; \\ \mathcal{E} &= \{(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{X} \mid \boldsymbol{x}' = f(\boldsymbol{x})\}; \\ \mathcal{Y} &= \{1, 2, \ldots, \bar{k}\}; \\ H &= \kappa; \\ \gamma(\boldsymbol{x}, \boldsymbol{x}') &= H(\boldsymbol{x}). \end{aligned} \tag{2.11}$$

The models in Defs. 2.14 and 2.15 are autonomous, and hence they serve solely for the purpose of evaluating the traffic of ETC. If we want to be able to manipulate the ETC traffic pattern for scheduling or improving its performance, we need to have an action set.

Throughout this dissertation, we consider the possibility of *early triggering*, i.e., we allow a scheduler or any other sampling strategy to query for a sample before $\tau(\xi(t_i))$ time units have elapsed. Considering how ETC triggering conditions are designed—the latest moment such that some Lyapunov stability condition is sure to hold—this is a sound strategy in terms of control stability and performance.

Definition 2.16 (ETC traffic model with early triggering)**.** *The ETC traffic model with early triggering actions is the system* $\mathcal{S} = (\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, H, \gamma)$ *where*

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_0 = \mathbb{R}^{n_x}; \\ \mathcal{U} &= (0, \bar{\tau}] \cap \mathcal{T}; \\ \mathcal{E} &= \{(\boldsymbol{x}, s, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{U} \times \mathcal{X} \mid \boldsymbol{x}' = \boldsymbol{M}(s)\boldsymbol{x} \text{ and } s \leq \tau(\boldsymbol{x})\}; \\ \mathcal{Y} &= (0, \bar{\tau}] \cap \mathcal{T}; \\ H &= \tau; \\ \gamma(\boldsymbol{x}, s, \boldsymbol{x}') &= s. \end{aligned} \tag{2.12}$$

Notice that the action is precisely the IST to be chosen, and it also reflects on the weight function. The output map is still τ and it now gives the ETC-based *deadline*. Again, it is useful to specialize this model for PETC, which renders an infinite-state, but finite-input and -output model.

Definition 2.17 (PETC traffic model with early triggering)**.** *The PETC traffic model with early triggering actions is the system* $\mathcal{S} = (\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, H, \gamma)$ *where*

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_0 = \mathbb{R}^{n_x}; \\ \mathcal{U} &= \{1, 2, ..., \bar{k}\}; \\ \mathcal{E} &= \{(\boldsymbol{x}, s, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{U} \times \mathcal{X} \mid \boldsymbol{x}' = \boldsymbol{M}(hs)\boldsymbol{x} \text{ and } s \leq \tau(\boldsymbol{x})\}; \\ \mathcal{Y} &= \{1, 2, ..., \bar{k}\}; \\ H &= \kappa; \\ \gamma(\boldsymbol{x}, s, \boldsymbol{x}') &= s. \end{aligned} \tag{2.13}$$

3

Scalable Traffic Models for Scheduling

This chapter addresses the problem of modeling and scheduling the transmissions generated by multiple event-triggered control (ETC) loops sharing a network. We present a method to build a finite-state model of the traffic generated by PETC based on quotient systems, which by construction mitigates the combinatorial explosion that is typical of symbolic models; in addition, it is a simulation of the original traffic model. The model is then augmented with early sampling actions that can be used by a scheduler to adjust each system's traffic. The complete networked control system is then modeled both as a network of timed game automata or as a parallel composition of automata. In both cases, the scheduling problem is effectively solved, whilst the latter provides improved scalability at the cost of being more restrictive in application.

This chapter is based on *G. de A. Gleizer and M. Mazo Jr. "Scalable Traffic Models for Scheduling of Linear Periodic Event-Triggered Controllers", presented at IFAC World Congress 2020* [46]. Section 3.6 is partially based on *G. Delimpaltadakis, G. de A. Gleizer, I. van Straalen, and M. Mazo Jr., "ETCetera: beyond event-triggered control," in Proc. of the 25th Int'l Conf. on Hybrid Systems: Computation and Control (HSCC '22)* [34].

3.1 Introduction

As we have argued in Chapter 1, the reduction in communications achieved by ETC is of reduced importance if one cannot accommodate multiple ETC loops in a shared network, and if nothing is done on a scheduler level, packet collisions are bound to happen. In this chapter, we address the problem of collision-free scheduling of multiple ETC systems in a shared network. More specifically, we design a scheduler that can adjust the traffic of each system and prevent said collisions, while ensuring stability and performance of the individual plants. Figure 3.1 depicts a networked control system (NCS) with multiple ETC loops sharing a single communication channel, where all the individual plant controllers are assumed, for simplicity, to run on the same device. In an ETC context, plant i can decide (based on the event occurrence) when to send its state sample $\hat{x}_i$, or the controller (assumed to be collocated with the scheduler) can request it. Because ETC systems are better described as hybrid systems (see, e.g., [4, 11, 47]), this scheduling problem belongs to the class of control synthesis problems on hybrid systems, which is known to be undecidable in general [48, 49]. Therefore, we use symbolic abstractions to tackle the scheduling problem, which are particularly suitable as scheduling actions have an intrinsically discrete nature, that of ordering tasks in the timeline.

This work is a follow up on [21–23, 50], which use timed game automata (TGA, Def. 2.13) to approximately simulate ETC traffic models. Doing so, they demonstrate that a scheduling strategy can be computed by composing multiple traffic TGAs with a network TGA and solving a safety game. The major drawback of the abstractions presented therein is the curse of dimensionality: their proposed isotropic partitioning creates a model where the number of states in the abstraction increases exponentially with the state-space dimension of the concrete plant.

Here we propose a different way of creating the traffic models: instead of partitioning space, we partition time, and determine the states associated with a given triggering time *a posteriori*. For PETC this allows to construct a quotient model (Def. 2.5), which simulates (Def. 2.2) the concrete traffic model. The resulting regions are intersections of non-convex quadratic cones that, despite being easy to check membership online, make the problem of computing transitions a non-convex quadratic constraint satisfaction problem, which is in general NP-hard [51]. We propose using semidefinite relaxations [51, 52], which are fast and reliable to solve, but add extra conservativeness to the resulting abstraction. After having constructed the traffic model, we augment it to allow for controllable early triggers, which can be used by the scheduler to avoid conflicts.

To solve the scheduling problem we apply two approaches. In the first, we follow the steps in [22] using timed game automata, with some minor modifications to keep the number and earliness of scheduling interventions small. To test this, we generate strategies using UPPAAL Tiga [36] and provide simulation results for an NCS with two ETC loops. In the second approach, we form the scheduling problem using finite transition systems, which requires additional assumptions concerning the checking periods of the individual PETC systems. The advantage is increased scalability with respect to the number of control loops in the shared network. Both approaches are implemented in `ETCetera` [34].

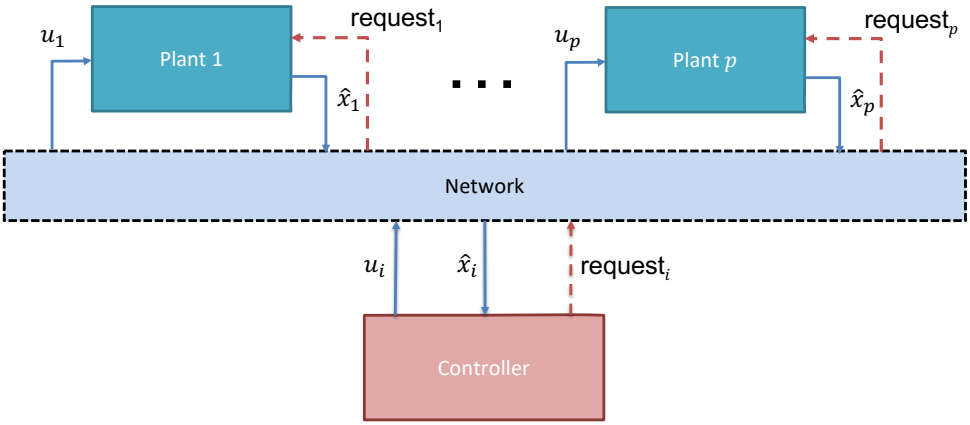


Figure 3.1: A network of p ETC systems.

3.2 Problem Formulation

The starting point for scheduling ETC traffic is modeling it, for which we use symbolic abstractions as in [22, 53]; however, we aim to build a quotient model, obtaining an exact simulation relation. More than that, we want to mitigate the curse of dimensionality that is typical of such abstractions:

Problem 3.1. *Build a quotient model $S_{/\mathcal{R}}$ for the traffic generated by system* (2.2) *using triggering condition* (2.4) *such that the cardinality of $\mathcal{X}_{/\mathcal{R}}$ does not directly depend on n_x.*

A traffic model alone is not sufficient for scheduling. System (2.2) is autonomous, and a scheduler needs to be able to alter the traffic pattern in some way to avoid communication conflicts. We choose to allow the scheduler to request data before the ETC triggers. Thus, we need to enrich the traffic model with controllable actions that represent this early triggering:

Problem 3.2. *Enhance $S_{/\mathcal{R}}$ with transitions that capture the evolution of system* (2.2) *when inter-event times smaller than $\kappa(x)$ are chosen.*

Finally, we need to pose and solve the scheduling problem:

Problem 3.3. *Design an NTGA that forms the scheduling problem, for which a strategy serves as a scheduler for the NCS with multiple event-triggered loops. In doing so, try to keep the number of communications to a small level.*

3.3 PETC Traffic Model

Constructing a quotient model of the PETC traffic model in Def. 2.15 requires two steps: 1) gathering the states that share the same output in a single quotient state, and 2) computing the transition relations between them.

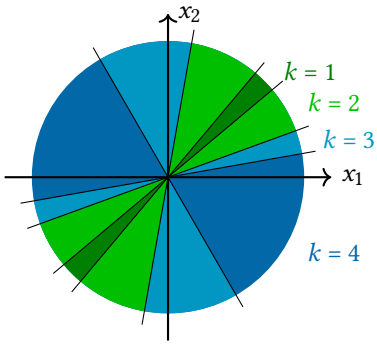


Figure 3.2: An example of the quotient-induced partition of $\mathbb{R}^{n_x}$ on a two-dimensional linear PETC system, with $\bar{k} = 4$. Each color represents a set $\mathcal{Q}_k$, which is a conjunction of quadratic cones.

3.3.1 Quotient state set

Gathering states that share the same output is in a sense straightforward in PETC. From Eq. (2.10), we can determine the set $\mathcal{K}_k \subseteq \mathbb{R}^{n_x}$ of states that will certainly have triggered by time k:

$$\mathcal{K}_k = \begin{cases} \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{x}^\top \boldsymbol{N}(hk)\boldsymbol{x} > 0\}, & k < \bar{k}, \\ \mathbb{R}^{n_x}, & k = \bar{k}. \end{cases} \tag{3.1}$$

Recall that $\bar{k}$ is the imposed maximal-in-discrete-time IST. To determine the state set whose output k is the *minimum* that satisfies $\boldsymbol{x}^\top \boldsymbol{N}(hk)\boldsymbol{x} > 0$, one must remove from $\mathcal{K}_k$ all states that could have triggered before, i.e., that belong to some $\mathcal{K}_j$ with $j < k$. This is expressed as

$$\mathcal{Q}_k = \mathcal{K}_k \cap \bigcap_{j=1}^{k-1} \bar{\mathcal{K}}_j. \tag{3.2}$$

By construction, $\mathcal{Q}_k, k \in \{1, 2, ..., \bar{k}\}$ constitutes a partition of $\mathbb{R}^{n_x}$ (see Fig. 3.2 for a depiction); also, $H(\boldsymbol{x}) = k, \forall \boldsymbol{x} \in \mathcal{Q}_k$. Therefore, $\mathcal{X}_{/\mathcal{R}} = \{\mathcal{Q}_1, \mathcal{Q}_2, ...\}$ is a good candidate for a quotient state set of the system $\mathcal{S}$. Finally, different from [21], we have that $|\mathcal{X}_{/\mathcal{R}}| = \bar{k}$, i.e., the cardinality of the quotient state space does not depend explicitly on n_x. This in part accomplishes solving Problem 3.1; however, for completing the model, we need to establish the transitions between these quotient states.

Remark 3.1. *Matrices $\boldsymbol{N}(hk)$ can be computed offline. Online determination of which region the current state $\boldsymbol{x}$ belongs to requires at most $\bar{k}$ quadratic operations: for each k from 1 to $\bar{k} - 1$, if $\boldsymbol{x}^\top \boldsymbol{N}(hk)\boldsymbol{x} > 0$, return $\mathcal{Q}_k$; if by the end of the iteration no such inequality was positive, return $\mathcal{Q}_{\bar{k}}$.*

Remark 3.2. *Unperturbed state-feedback ETC has an intrinsic positive minimum inter-sample time (MIST), which, in the case of PETC, can be bigger than $k = 1$. In this case, for all $k < \underline{k}$, where $\underline{k}$ is such MIST, it holds that $\boldsymbol{N}(hk) \preceq \boldsymbol{0}$. This can be checked offline, and the corresponding matrices may be discarded. Likewise, a maximum inter-event time $\bar{k}$ can naturally show up if, for example, $\boldsymbol{N}(hk^*) \succ \boldsymbol{0}$ for some k^*, which can also be checked offline. In this case, take $\bar{k} = k^*$.*

3.3.2 Quotient transition relations

The problem of determining the transition relation between two quotient states $\mathcal{Q}_i$ and $\mathcal{Q}_j$ is, from Eq. (2.9),

$$\exists \boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}} \,:\, \boldsymbol{x} \in \mathcal{Q}_i, \xi_{\boldsymbol{x}}(hi) = \boldsymbol{M}(hi)\boldsymbol{x} \in \mathcal{Q}_j, \tag{3.3}$$

where the last equality uses Eq. (2.5). Expanding $\mathcal{Q}_i, \mathcal{Q}_j$ with Eqs. (3.2) and (3.1) arrives in the following *non-convex quadratic constraint satisfaction problem*:

$$\begin{aligned}
\exists\ & \boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}} \\
\text{s.t. } & \boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(hi)\boldsymbol{x} > 0, \\
& \boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(hi')\boldsymbol{x} \le 0, \forall i' \in \{1, ..., i-1\}, \\
& \boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(hi)^{\mathsf{T}}\boldsymbol{N}(hj)\boldsymbol{M}(hi)\boldsymbol{x} > 0, \\
& \boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(hi)^{\mathsf{T}}\boldsymbol{N}(hj')\boldsymbol{M}(hi)\boldsymbol{x} \le 0, \forall j' \in \{1, ..., j-1\}.
\end{aligned} \tag{3.4}$$

The non-convexity of this problem can be easily checked using the facts that both $>$ and $\le$ inequalities are present, and that the matrices $\boldsymbol{N}(hi)$ are non-definite.[1] We solve it by means of semi-definite relaxations [52, SDR,],[2] which take the form

$$\begin{aligned}
\exists\ & \boldsymbol{X} \in \mathbb{S}^{n_{\mathrm{x}}}_{+} \\
\text{s.t. } & \mathrm{Tr}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{N}(hi)) \ge 0, \\
& \mathrm{Tr}(\boldsymbol{X}\boldsymbol{N}(hi')) \le 0, \forall i' \in \{1, ..., i-1\}, \\
& \mathrm{Tr}(\boldsymbol{X}\boldsymbol{M}(hi)^{\mathsf{T}}\boldsymbol{N}(hj)\boldsymbol{M}(hi)) \ge 0, \\
& \mathrm{Tr}(\boldsymbol{X}\boldsymbol{M}(hi)^{\mathsf{T}}\boldsymbol{N}(hj')\boldsymbol{M}(hi)) \le 0, \forall j' \in \{1, ..., j-1\}, \\
& \mathrm{Tr}(\boldsymbol{X}) = 1,
\end{aligned} \tag{3.5}$$

where the last equation was added to avoid the trivial solution $\boldsymbol{X} = 0$; the value 1 was chosen arbitrarily, since Eq. (3.4) is homogeneous. To determine (offline) the complete transition set $\mathcal{E}_{/\mathcal{R}}$, one requires solving $\bar{k}^2$ semidefinite problems. The final model follows:

Model 3.1 (Quotient PETC Traffic Model)**.** *The quotient model of the PETC traffic in Def. 2.15 is the system* $\mathcal{S}_{/\mathcal{R}} =: (\mathcal{X}_{/\mathcal{R}}, \mathcal{X}_{/\mathcal{R}0}, \mathcal{E}_{/\mathcal{R}}, \mathcal{Y}, H_{/\mathcal{R}})$ *with*

- $\mathcal{X}_{/\mathcal{R}} = \mathcal{X}_{/\mathcal{R}0} = \{\mathcal{Q}_1, \mathcal{Q}_2, ..., \mathcal{Q}_{\bar{k}}\}$;
- $\mathcal{E}_{/\mathcal{R}} = \{(\mathcal{Q}_i, \mathcal{Q}_j) \mid$ *Eq.* (3.5) *is satisfied*$\}$;
- $H_{/\mathcal{R}}(\mathcal{Q}_k) = k$.

A typical PETC traffic model is depicted in Fig. 3.3 (left). By construction, we obtain the following result:

Proposition 3.1. *Model 3.1 is a quotient system of* $\mathcal{S}$ *from Def. 2.15, and, therefore,* $\mathcal{S}_{/\mathcal{R}}$ *simulates* $\mathcal{S}$*.*

As a consequence, all sequences of triggering times generated by system (2.2)–(2.10) can be generated by our model $\mathcal{S}_{/\mathcal{R}}$. This solves Problem 3.1.

[1] See Remark 3.2: the definite cases are discarded.

[2] Additionally, we relax the strict inequalities with non-strict ones, so that it can fit the semi-definite programming formulation.

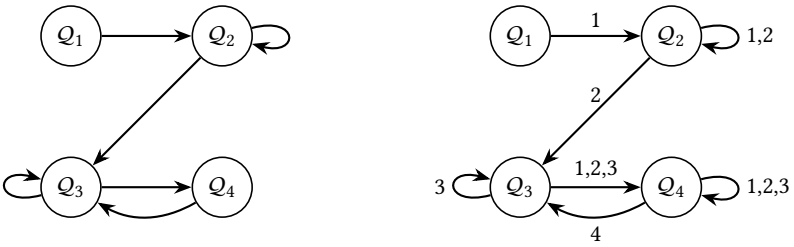


Figure 3.3: An example of quotient PETC traffic model (left) and the same system enhanced with early triggering actions (right).

Remark 3.3. *A relaxation generally provides conservative solutions. In our case, it may generate spurious transitions. If such transitions do occur, this does not change the fact that the constructed symbolic model simulates $\mathcal{S}$.*

3.3.3 Early triggering

As stated earlier, for the traffic model to be applicable for scheduling, we need to augment it with controllable transitions that correspond to early triggering. From a quotient state $\mathcal{Q}_i$, one can allow early triggers for any $k \in \mathbb{N} : k < i$; for simplicity we choose to label the corresponding actions by k. It remains necessary to verify which transitions exist for such actions. The problem is essentially the same as (3.3):

$$\exists \boldsymbol{x} \in \mathbb{R}^{n_x} : \boldsymbol{x} \in \mathcal{Q}_i, \xi_{\boldsymbol{x}}(hk) = \boldsymbol{M}(hi)\boldsymbol{x} \in \mathcal{Q}_j. \tag{3.6}$$

Eq. (3.6) can also be relaxed into a semidefinite problem like (3.5), by replacing j by k. This approach gives the following model:

Model 3.2 (Quotient PETC traffic model with early triggering)**.** *The quotient model of the PETC traffic with early triggering is the system $\mathcal{S}^*_{/\mathcal{R}} =: (\mathcal{X}_{/\mathcal{R}}, \mathcal{X}_{/\mathcal{R}0}, \mathcal{U}, \mathcal{E}^*_{/\mathcal{R}}, \mathcal{Y}, H_{/\mathcal{R}})$, where*

- $\mathcal{U} = \{1, 2, ..., \bar{k}\}$;
- $\mathcal{E}^*_{/\mathcal{R}} = \{(\mathcal{Q}_i, k, \mathcal{Q}_j) \mid$ *the SDR of Eq.* (3.6) *is satisfied*$\}$.

Computing all of its transitions requires solving $\bar{k} + 2\bar{k} + ... + \bar{k}(\bar{k}-1) = \bar{k}^2(\bar{k}-1)/2 \in \mathcal{O}(\bar{k}^3)$ semidefinite problems. A depiction of the quotient model in Fig. 3.3 (left) enhanced with early triggering is seen on the right side of the same figure. Model 3.2 solves Problem 3.2.

3.4 Scheduling of PETC systems

3.4.1 Early triggering and TGA

Now we can transform the quotient system $\mathcal{S}^*_{/\mathcal{R}}$ into a TGA. For the game part, we set the early triggering actions in $\mathcal{S}^*_{/\mathcal{R}}$ as controllable, and the event triggers as uncontrollable. All that is left is defining the clock set, the guards, and the invariants, resulting in the following TGA:

Model 3.3 (PETC Traffic Timed Game)**.** *The model is the TGA $\mathcal{A} = (\mathcal{X}_{/\mathcal{R}}, \mathcal{X}_{/\mathcal{R}0}, \mathcal{U}_c, \mathcal{U}_u, C, \mathcal{E}_c \cup \mathcal{E}_u, I)$ where*

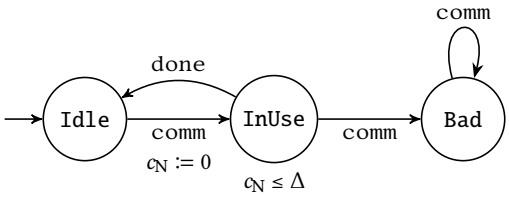


Figure 3.4: TGA of a shared network.

- $\mathcal{U}_c = \{\texttt{early}\}$;
- $\mathcal{U}_u = \{\texttt{trigger}\}$;
- $C = \{c\}$;
- $\mathcal{E}_c = \{(Q_i, c = k, \texttt{early}, \{c\}, Q_j) \mid (Q_i, k, Q_j) \in \mathcal{E}^*\}$;
- $\mathcal{E}_u = \{(Q_i, c = i, \texttt{trigger}, \{c\}, Q_j) \mid (Q_i, Q_j) \in \mathcal{E}_{/\mathcal{R}}\}$;
- $I(Q_i) = (c \leq i)$.

Model 3.3 uses one clock, that is reset at every transition. The invariant of a quotient state Q_i is $c \leq i$, because i is the time that a trigger is sure to occur; hence $c = i$ is the clock constraint associated with this uncontrolled action. For the controlled, early triggering actions, the transition is enabled at discrete instants satisfying $c = k$, for $k < i$.

3.4.2 Network and NCS models

For scheduling, we follow the same strategy as described in [22], using the same network model as theirs, with a minor technical change[3]:

Model 3.4. *(Network TGA, adapted from [22]). The network model is the TGA* $\mathcal{N} = (\mathcal{L}, l_0, \mathcal{U}_{cN}, \{*\}, C_N, \mathcal{E}_N, I_N)$ *where*

- $\mathcal{L} = \{\texttt{Idle}, \texttt{InUse}, \texttt{Bad}\}$;
- $\mathcal{U}_{cN} = \{\texttt{comm}, \texttt{done}\}$;
- $C = \{c_N\}$;
- $\mathcal{E}_N = \{(\texttt{Idle}, \texttt{true}, \texttt{comm}, \{c_N\}, \texttt{InUse}),$
  $(\texttt{InUse}, c_N = \Delta, \texttt{done}, \emptyset, \texttt{Idle}),$
  $(\texttt{InUse}, \texttt{true}, \texttt{comm}, \emptyset, \texttt{Bad}),$
  $(\texttt{Bad}, \texttt{true}, \texttt{comm}, \emptyset, \texttt{Bad})\}$;
- $I_N(\texttt{InUse}) = (c_N \leq \Delta)$,

where $*$ *denotes a "do-not-care" action and* Δ *is the maximum channel occupancy time.*

Model 3.4 is represented in Fig. 3.4. The state `Bad` is reached if a second communication happens while the channel is still occupied by the first.

To model the NCS, we build an NTGA of the two or more traffic models $\mathcal{A}_i$ with the network model $\mathcal{N}$. What remains to be done is synchronizing the correct actions. For this, we add a synchronization channel called up, which is used as follows:

[3]The difference of this model with respect to [22] is that, here, all actions are controlled. We do this because of how NTGA are composed in UPPAAL Tiga: if an uncontrolled edge is synchronized with a controlled edge, the composed edge is uncontrolled. When we compose the traffic models with the network model, we want the `early` communications to be controlled, and the `trigger` ones not to.

- every `early` and `trigger` actions of each traffic model $\mathcal{A}_i$ fires the synchronizing action `up!`;
- every `comm` action of the network model $\mathcal{N}$ takes the synchronizing action `up?`.

While avoiding the `Bad` state is necessary, we also want that the number of early triggers is small, so as to benefit from the communication savings of ETC. For that, we introduce an integer variable $e, 0 \le e \le E$, representing an accumulated "earliness" of communications, with E as the maximum allowed earliness. It is essentially a bounded integrator that increases every time an early trigger is done and decreases when a natural trigger happens. It starts at zero and is updated as

$$e \longleftarrow \max(0, \min(E, e + r(k - i) - \bar{e})) \tag{3.7}$$

for every `trigger` or `early` transition from any traffic model, from quotient state $\mathcal{Q}_i$ when $c = k$. The parameters $r \in \mathbb{N}_+$ and $\bar{e} \in \mathbb{N}_+$ represent the cost of a time unit and a reference value for e, respectively. The earlier the trigger is, the higher the cost incurred. Parameter $\bar{e}$ is necessarily positive so that natural triggers discount e. Like any arithmetic on bounded integers, the evolution of e can be represented as an automaton itself.[4]

As a final note, remember that the time in model $\mathcal{A}$ is normalized w.r.t. the check time h. When composing the NTGA, one needs to put the clocks and their constraints in the same time scale.

3.4.3 Strategies for schedulers

In UPPAAL Tiga, strategies can be generated so as to guarantee certain specifications. We refer the reader to the manual of UPPAAL Tiga [36] for the complete list. In our case, we want that the NTGA never enters state `Bad` of $\mathcal{N}$, while keeping the earliness below a certain threshold E. This can be achieved by setting the specification `strategy safe = control: A[] not network.Bad and e < E`. The resulting strategy maps the locations of each automaton and their clock valuations into the decision of whether to trigger early or not. Therefore, a scheduler that implements such strategy needs to determine online the regions $\mathcal{Q}_i$ that the state of each system belongs to, and keep track of how much time elapsed since the last communication of each plant.

3.5 Numerical Results using TGA

Consider two copies of a linearized batch reactor, taken from [54], of the form (2.2) with

$$\begin{aligned} \boldsymbol{A}_i &= \begin{bmatrix} 1.38 & -0.208 & 6.715 & -5.676 \\ -0.581 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix}, \\ \boldsymbol{B}_i &= \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}, \quad \forall i \in \{1, 2\}. \end{aligned} \tag{3.8}$$

[4] UPPAAL Tiga allows one to use integer variables, and it performs the necessary operations automatically.

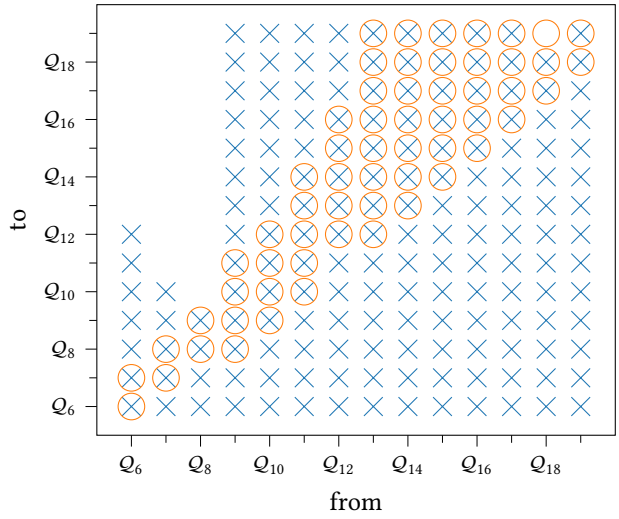


Figure 3.5: Transition relations of $\mathcal{S}^{\star}_{/\mathcal{R}}$ of loop 1, for `trigger` actions (x) and `early` actions (o) with $k = 1$.

Two different controllers $\boldsymbol{K}_i$ were designed for this plant using LQR with matrices $\boldsymbol{Q}_{\mathrm{LQR},1} = \boldsymbol{Q}_{\mathrm{LQR},2} = \mathbf{I}$ and $\boldsymbol{R}_1 = 0.1\mathbf{I}, \boldsymbol{R}_2 = 0.05\mathbf{I}$. The Lyapunov function chosen was the LQ cost, that is, setting $\boldsymbol{Q}_{\mathrm{lyap},i} = \boldsymbol{Q}_{\mathrm{LQR},i} + \boldsymbol{K}_i^{\mathsf{T}}\boldsymbol{R}_i\boldsymbol{K}_i$ and solving the continuous-time Lyapunov equation for $\boldsymbol{P}_i$. We used a triggering condition based on the Lyapunov function, so as to guarantee that

$$\dot{V}_i(t) \leq -\rho_i \boldsymbol{\xi}_i(t)^{\mathsf{T}} \boldsymbol{P}_i \boldsymbol{\xi}_i(t),$$

for some $0 < \rho_i < 1$. We set $\rho_1 = \rho_2 = 0.8$. This triggering condition can be expressed in quadratic form (2.4) with

$$\boldsymbol{Q}_i = \begin{bmatrix} \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{P}_i + \boldsymbol{P}_i\boldsymbol{A}_i + \rho_i \boldsymbol{Q}_{\mathrm{lyap},i} & \boldsymbol{P}_i\boldsymbol{B}_i\boldsymbol{K}_i \\ \boldsymbol{K}_i^{\mathsf{T}}\boldsymbol{B}_i^{\mathsf{T}}\boldsymbol{P}_i & \mathbf{0} \end{bmatrix}.$$

In both cases, $h_1 = h_2 = h = 0.01$; following Remark 3.2, we obtained natural maximum inter-event times at $\bar{k}_1 = 19$ and $\bar{k}_2 = 16$ by imposing that $\boldsymbol{N}(hk)$ have its largest eigenvalue bigger than 10^{-3}. Likewise, both have MISTs greater than 1: $\underline{k}_1 = 6$, $\underline{k}_2 = 4$.

To build Model 3.3 for each control loop, we used Python with Numpy, Scipy and control packages, and CVXPY [55] with solver SCS [56] to solve the semidefinite problems involved. The whole process of computing matrices $\boldsymbol{N}(hk)$ and solving the semidefinite problems took 46.64 seconds for loop 1 and 31.51 seconds for loop 2. The computer used is a MacBook Pro with a 3.1 GHz Intel Core i5 CPU and memory of 8 GB, 2133 MHz LPDDR3. The resulting transition relation for closed-loop system 1 is represented in Figure 3.5. As one can see, there is a significant amount of nondeterminism introduced by this model, especially for high triggering times.

A series of scripts was used to generate the XML files that are used for TGA models in UPPAAL Tiga. We used all times in the NTGA relative to h, and set $\Delta = 1$. The earliness

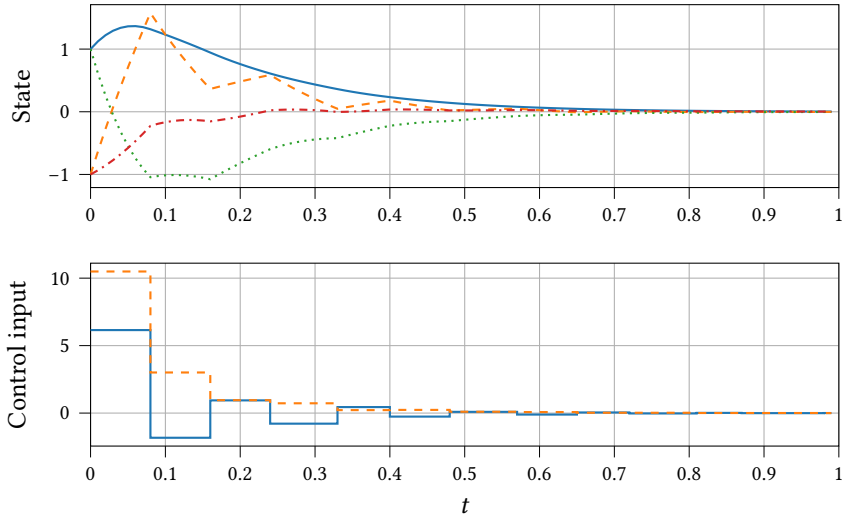


Figure 3.6: Trajectories of $\xi_1(t)$ (top) and $K_1\hat{\xi}_1(t)$ (bottom).

parameters for Eq. (3.7) were $r = 2, \bar{e} = 1, E = 2$. These parameters allow the scheduler to trigger one step earlier at every two communications.

The strategy was solved in UPPAAL STRATEGO [57] version 4.1.20-5, which includes all functionalities of UPPAAL Tiga. It took 0.864 s to find a solution. The generated strategy is too long to be reproduced in this paper, but we give below one example of when an early trigger has to occur:

$$\begin{aligned}&\text{If System 1 is in } \mathcal{Q}_6, \text{ System 2 is in } \mathcal{Q}_4, \text{ and } e = 0,\\ &\quad \text{when } c_1 = 5 \text{ and } c_2 \in \{1,2,3\}, \text{ do } \texttt{early} \text{ on System 1;}\\ &\quad \text{when } c_2 = 3 \text{ and } c_1 \in \{3,4,5\}, \text{ do } \texttt{early} \text{ on System 2,}\end{aligned}$$

where c_i represents the clock valuation of system i. As one can see, the strategy is not deterministic. In the example above, the early trigger can be executed on any of the loops when $(c_1, c_2) = (5, 3)$. In such case, the scheduler must arbitrate who triggers.

Figures 3.6 and 3.7 show the results of a simulation of the two control loops executing in parallel with the communication managed by the synthesized scheduler. The initial conditions are $\xi_1(0) = \begin{bmatrix}1 & -1 & 1 & -1\end{bmatrix}^\mathsf{T}$ and $\xi_2(0) = \begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix}^\mathsf{T}$. The first pair of communications was arbitrated on a round-robin fashion. Figure 3.8 shows the communication pattern of the NCS. As we can see, both systems' states converge to zero, while there is no conflict in communications. As designed through the earliness mechanism, about half of the communications are early triggers, and half are natural, event triggers.

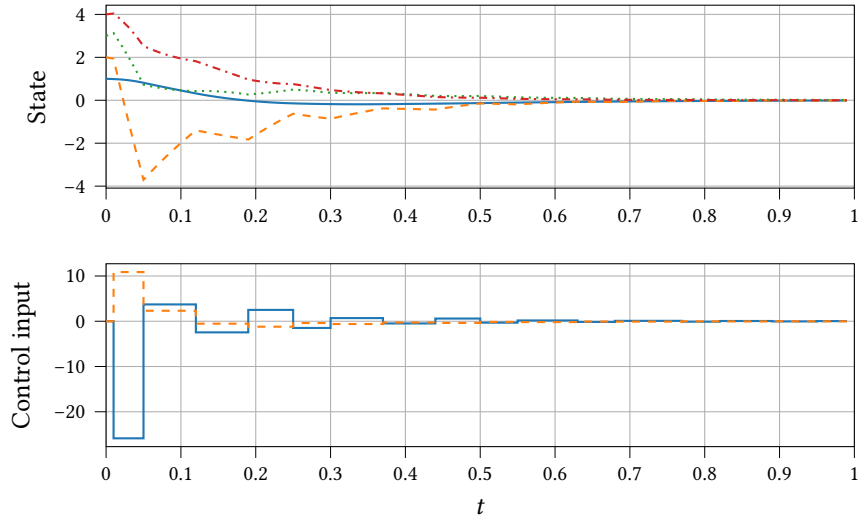


Figure 3.7: Trajectories of $\xi_2(t)$ (top) and $K_2\hat{\xi}_2(t)$ (bottom).

Figure 3.8: Communication pattern of the simulated NCS: 'x' marks represent event triggers, while 'o' marks represent early triggers.

3.6 Scheduling with finite-state systems

Even though our models have good scalability w.r.t. the plant's state-space dimension, an issue remains with the scalability w.r.t. the number of control loops to be scheduled. The size of the obtained NTGA state space grows exponentially with the number of control loops, and solving strategies for TGA is EXPTIME-complete [58]. According to experiments in [59], solving the scheduling problem using UPPAAL Tiga on a system with 4 control loops did not finish after five days. The conclusion is that scheduling ETC control loops with TGA seems to be only viable with up to three systems, or if some other type of multiplexing can be done to logically isolate the communications of small groups of control loops.

In case the checking times of each plant and the channel occupancy time have a common divisor, one can use finite-state systems instead of TGA to solve the scheduling problem, by employing discrete clocks instead of continuous ones. In [35], such a method was developed for PETC systems assuming, for simplicity, that all control loops share the same checking time h and that $\Delta = h$.

First, Model 3.2 is modified such that it acts on a per-sample basis. Instead of having actions corresponding to the next inter-event time $\mathcal{U} = \mathcal{Y} = \{1, 2, \ldots, \bar{k}\}$, a system can either wait (`w`) , or trigger (`t`) at the current sampling time. The result is a new transition system:

Model 3.5 (Discrete-clock PETC traffic model)**.** *Given Model 3.2 and a maximum initial*

discrete-time delay $d \in \mathbb{N}$, the discrete-clock PETC traffic model with early triggers is the system $\hat{\mathcal{S}} = (\hat{\mathcal{X}}, \hat{\mathcal{X}}_0, \hat{\mathcal{U}}, \hat{\mathcal{E}}, \hat{\mathcal{Y}}, \hat{H})$, where:

- $\hat{\mathcal{X}} := \{T_i \mid \mathcal{Q}_i \in \mathcal{X}_{/\mathcal{R}}\} \cup \{W_{i,k} \mid 1 \le k < H(\mathcal{Q}_i), \mathcal{Q}_i \in \mathcal{X}_{/\mathcal{R}}\} \cup \{I_1, I_2, \dots, I_d\}$,
- $\hat{\mathcal{X}}_0 := \{I_1\}$,
- $\hat{\mathcal{U}} := \{\mathtt{w}, \mathtt{t}\}$,
- $\hat{\mathcal{E}} := \hat{\mathcal{E}}_{\mathtt{t}} \cup \hat{\mathcal{E}}_{\mathtt{w}} \cup \hat{\mathcal{E}}_0$, *where*
  - $\hat{\mathcal{E}}_{\mathtt{t}} := \{(T_i, \mathtt{t}, T_j) \mid (\mathcal{Q}_i, 1, \mathcal{Q}_j) \in \mathcal{E}^*_{/\mathcal{R}}\} \cup \{(W_{i,k}, \mathtt{t}, T_j) \mid (\mathcal{Q}_i, k+1, \mathcal{Q}_j) \in \mathcal{E}^*_{/\mathcal{R}}\}$,
  - $\hat{\mathcal{E}}_{\mathtt{w}} := \{(W_{i,k}, \mathtt{w}, W_{i,k+1}) \mid 1 \le k < H(\mathcal{Q}_i) - 1\}$,
  - $\hat{\mathcal{E}}_0 := \{I_i, \mathtt{t}, T_j \mid i \in \mathbb{N}_{\le d}, \mathcal{Q}_j \in \mathcal{X}_{/\mathcal{R}}\} \cup \{I_i, \mathtt{w}, I_{i+1} \mid i \in \mathbb{N}_{<d-1}\}$,
- $\hat{\mathcal{Y}} := \{\mathtt{T}, \mathtt{W}\}$,
- $\hat{H}(T_i) = \mathtt{T}$, $\hat{H}(W_{i,j}) = \hat{H}(I_k) = \mathtt{W}$.

The state space of Model 3.5 is composed by "triggered" states T_i, one for each quotient state $\mathcal{Q}_i$; "waited" states $W_{i,k}$, representing a state in which k time units have elapsed since the last trigger, starting from T_i; and initialization states I_j. The outputs indicate whether the system has just triggered (T) or waited (W). In this transition system, triggering after k samples becomes a sequence of $k-1$ w actions followed by a single t action. The scheduling problem now becomes avoiding that two systems enter a T state simultaneously. The first sampling upon initialization of the system is dictated by the states I_j: each system starts at I_1 and has d time units to trigger its first sample, from which it can go to any T_i state (effectively initializing the traffic model). This approach also gives an extra degree of freedom when it comes to initialization, and is more realistic as the state of the plant is normally not known until the first sample is taken. Another reason for the initialization states is that the output is defined with respect to what have just happened; hence, the scheduling problem would start being infeasible if it initialized at a T_i state. The discrete-clock version of the system in Fig. 3.3 is depicted in Fig. 3.9.

For scheduling, we perform a *parallel composition* of each PETC system. When scheduling p control loops, denote the discrete-clock traffic model of the i-th loop by $\hat{\mathcal{S}}_i = (\hat{\mathcal{X}}_i, \hat{\mathcal{X}}_{i;0}, \hat{\mathcal{U}}_i, \hat{\mathcal{E}}_i, \hat{\mathcal{Y}}_i, \hat{H}_i)$. The resulting composition is given by the transition system $\mathcal{S}_\times = (\mathcal{X}_\times, \mathcal{X}_{\times;0}, \mathcal{U}_\times, \mathcal{E}_\times, \mathcal{Y}_\times, H_\times)$, where

- $\mathcal{X}_\times := \hat{\mathcal{X}}_1 \times \cdots \times \hat{\mathcal{X}}_p$,
- $\mathcal{X}_{\times;0} := \hat{\mathcal{X}}_{1;0} \times \cdots \times \hat{\mathcal{X}}_{p;0}$,
- $\mathcal{U}_\times := \hat{\mathcal{U}}_1 \times \cdots \times \hat{\mathcal{U}}_p$,
- $\mathcal{E}_\times := \{(x_1, \dots, x_p), (u_1, \dots, u_p), (x'_1, \dots, x'_p) \mid \forall i \in \mathbb{N}_{\le p}, (x_i, u_i, x'_i) \in \hat{\mathcal{E}}_i\}$,
- $\mathcal{Y}_\times := \hat{\mathcal{Y}}_0 \times \cdots \times \hat{\mathcal{Y}}_n$,
- $H_\times(x_0, \dots, x_n) := (\hat{H}_0(x_0), \dots, \hat{H}_n(x_n))$.

The job of the scheduler is to create a strategy satisfying the following safety specification: "*always avoid states in $\mathcal{S}_\times$ whose output contains more than one* T". Creating a controller that is able to satisfy such a specification can be done effectively for finite transition systems by solving a safety game [17, Chap. 6], where the maximal fixed point of the operator

$$F_{\mathcal{W}}(\mathcal{Z}) = \{x \in \mathcal{Z} \mid x \in \mathcal{W} \text{ and } \exists u \in U(x) : \emptyset \neq \mathrm{Post}_u(x) \subseteq \mathcal{Z}\}, \tag{3.9}$$

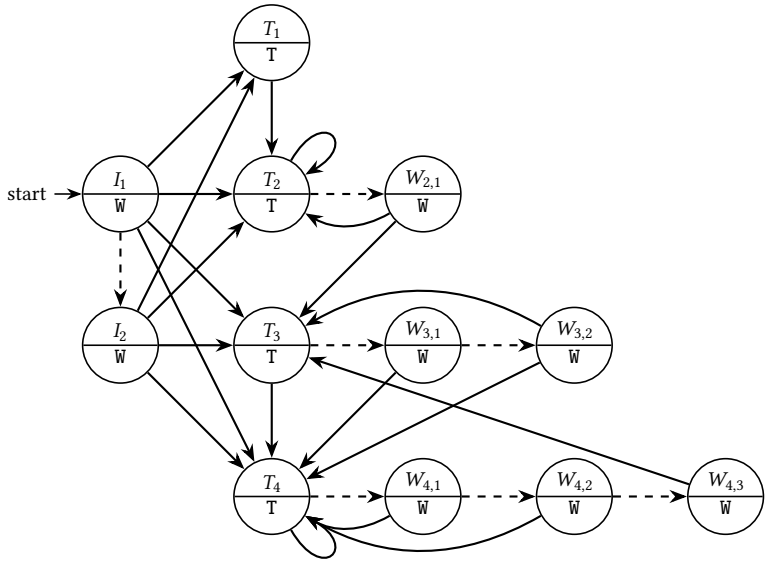


Figure 3.9: The discrete-clock model $\hat{S}$ of the system depicted in Fig. 3.3, with $d = 2$. Dashed arrows represent action `w`, while full arrows represent `t`.

is found by iterating over its output (starting with $\mathcal{Z}_0 := \mathcal{X}_\times$). The set $\mathcal{W}$ is the *safe set* which in this case is given by:

$$\mathcal{W} = \left\{ (x_1, \ldots, x_n) \in \mathcal{X}_\times \;\middle|\; \left| \{ i \in \mathbb{N}_{\leq p} \mid \hat{H}_i(x_i) = \mathtt{T} \} \right| \leq 1 \right\}. \tag{3.10}$$

The iteration finishes at some finite i s.t. $\mathcal{Z}_i = \mathcal{Z}_{i+1} =: \mathcal{Z}^*$, which contains all states of $\mathcal{S}_\times$ from which there exists at least one action guaranteeing that bad states (a collision of transmissions) can be avoided. If $\mathcal{Z}^* = \emptyset$, the set of control loops is not schedulable (given their abstractions); likewise if $\mathcal{Z}^* \neq \emptyset$ but $\mathcal{Z}^* \not\ni \mathcal{X}_{\times;0}$, which indicates that the system is not schedulable from its initial condition, but it may be if the systems are initialized in a more favorable way. The scheduler strategy can subsequently be defined as:

$$U_c(x) = \{ u \in \mathcal{U}_\times \mid \emptyset \neq \mathrm{Post}_u(x) \subseteq \mathcal{Z}^* \}. \tag{3.11}$$

This is a function that returns a set of *safe actions* given some combined state $x \in \mathcal{X}_\times$. This combined state, just like in the scheduling problem with TGA, can be obtained from the last sampled states of each of the control systems, plus a clock keeping track of the elapsed time since the previous transmission.

Solving safety games in finite transition systems is more computationally efficient than solving with TGA because it does not need the additional step of abstracting (even if on-the-fly) the infinite TGA into a finite system. For additional efficiency (especially when it comes to memory), this scheduling problem has been implemented in `ETCetera` [34] using binary decision diagrams (BDDs) [60]. For details of this implementation, the reader is kindly referred to [35].

A numerical test. To exemplify the gain in scalability when using finite state systems for the scheduling problem, we have solved it for the control loops presented in Section 3.5; but instead of only two loops, we ran the problem with five loops: four copies of the system with matrices $\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{K}_1, \boldsymbol{Q}_1$ and one copy of the system with matrices $\boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{K}_2, \boldsymbol{Q}_2$. All systems were set with $d = 10$ maximum initial delay. The scheduling problem using `ETCetera` with BDDs took only 663 ms to synthesize a scheduler. When adding a sixth system to the network, another copy of the $(\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{K}_1, \boldsymbol{Q}_1)$ loop, the algorithm took significantly longer—6 minutes and 41 seconds—to return an infeasibility answer.[5]

3.7 Conclusions

In this chapter, we have presented a method to build a quotient model of the traffic generated by PETC, and how to augment it with early sampling actions for scheduling of multiple PETC loops. The quotient model has many advantages with respect to previous work: first, it is a (exact) simulation instead of an approximate simulation;[6] and second, it avoids the combinatorial explosion created by isotropic partitioning of the state space. The state space and output map of the quotient model can be easily created straight from the PETC and system matrices, requiring no solution of LMIs or other optimization problems. The transition relations do require semidefinite problems to be solved, but only one per transition, with no reachability tools required. It is relatively fast to compute, and the models generated are reasonably small. The use of TGA models for scheduling of ETC had already been demonstrated in [22]; here, we demonstrate that they can also be done for PETC, and argue that it is in fact simpler to do so.

As we have seen, using TGA can make scheduling more than a few controllers in the same network prohibitive. Under additional assumptions, we could pose the scheduling problem with finite transition systems and, harnessing the power of BDDs, showed that it can scale significantly better. However, it seems that solution times can vary significantly, especially when the loops are in fact unschedulable. In Chapter 8, we address this issue by minimizing the individual transition systems modulo alternating simulation equivalence.

Among the disadvantages of our solution is the high nondeterminism of the generated models. The state-space partitions are based solely on the output function, and each region seems to be large enough that, after some time, many regions can be reached. A highly nondeterministic traffic model can hamper the generation of strategies, as the predictability of the model after multiple steps gets smaller. In Chapter 4, we address this problem by developing a framework of abstraction refinements for PETC.

A second point of attention is addressing optimality of these schedulers. Parameterizing the earliness function (3.7) is not always trivial. Even so, finding a scheduler that minimizes the interventions is still an open problem. Priced TGA could be used, but their undecidability for games with three clocks has been proven by [61], putting a roadblock in that direction. Approximate solutions using stochastic priced TGA [57] have been briefly explored (experimentally), and did not result in improved performance for the control loops, perhaps because it is difficult to characterize probabilities in the transition systems

[5] These results can be reproduced within `ETCetera` by running the script in `examples/scheduling_safetygame_5sys_4d.py`.

[6] In approximate simulations, item (ii) of Def. 2.2 is replaced with "for every $(x_a, x_b) \in \mathcal{R}, |H_a(x_a) - H_b(x_b)| \leq \epsilon$" for some $\epsilon > 0$.

when the concrete systems are deterministic. In Chapter 7, we address sampling optimality for one system, aiming at increasing its average inter-sample time. A similar approach could be used for multiple systems.

The conflict-free scheduling problem with early triggers that we have presented has broad applicability, but it may be too restrictive if the network protocol in use has ways to recover from conflicts. For example, in Controller Area Networks (CAN bus), the node with highest hard-coded priority successfully sends its message, hence conflicts are not totally destructive. ETC scheduling for CAN networks has been addressed in [62], based on the traffic models we introduced here. In addition, occasional late samples can not be detrimental to stability, and allowing some can make the schedulability problem much less stringent. Late triggers have been partially investigated in [35, 62] and shown to enable schedulability when it is not possible otherwise. However, the different ways to address the problem of late sampling in a way that their impact in control performance is formally bounded or minimized is subject of current investigation.

4

Bisimilar traffic models for a modified PETC

We provide a method to construct finite abstractions exactly bisimilar to linear systems under a modified periodic event-triggered control, when considering as output the inter-event times they generate. Assuming that the initial state lies on a known compact set, these finite-state models can exactly predict all sequences of sampling times until a specified Lyapunov sublevel set is reached. Based on these results, we provide a way to build tight models simulating the traffic of conventional PETC. These models allow computing tight bounds on the PETC average sampling frequency and global exponential stability (GES) decay rate.

This chapter is based on *G. de A. Gleizer and M. Mazo Jr. "Towards Traffic Bisimulation of Linear Periodic Event-Triggered Controllers", IEEE Control Systems Letters, vol. 5, no. 1, pp. 25–30, 2021.* [63].

4.1 Introduction

The models presented in Section 3 have great scalability properties, but at the same time they exhibit severe non-determinism, likely due to the small number of states of the abstraction and the relaxations used when computing the transition relation. Therefore, even though their one-step ahead predictions are exact, after a couple of steps its prediction capability is severely limited. For example, revisit Fig. 3.5: The state Q_8 is the one that exhibits the least non-determinism, having only 4 successor states after a `trigger` transition. At the next step, it can reach Q_9, which has `trigger` transitions to all states. Hence, the prediction of the IST after two samples from the current state is trvially "any".

This chapter tackles precisely this longer-term predictability issue. Building upon the quotient model from the previous chapter, we develop new abstractions that enumerate all possible sequences of inter-event times until a Lyapunov sublevel set is reached. Based on this, we propose a modified PETC mixed with periodic control, hereafter denoted MPETC, that initiates with PETC sampling and switches to periodic sampling when the states lie inside the aforementioned sublevel set. The MPETC retains the practical benefits from PETC, while improving traffic predictability; for it, our abstraction constitutes a bisimulation. The abstraction is computable because the number of possible sampling sequences generated during the PETC phase is finite, and checking whether PETC generates a given sampling sequence is decidable: it is equivalent to the satisfiability of a conjunction of non-convex quadratic inequalities, to which solvers exist (e.g., the satisfiability-modulo-theories (SMT) solver Z3 [64]). In our symbolic model, each state is associated with a sequence of inter-event times, which is similar in spirit to [65]. When generating these finitely many discrete states, exhaustive search can be avoided by employing a recursive algorithm.

The clear advantages of our new model are the exact enumeration of sampling sequences that can be generated by PETC on a significant future horizon and the establishment of tight bounds on the Lyapunov function convergence speed. Naturally, a disadvantage of our presented method is a substantial growth in the number of discrete states when compared to Chapter 3. Finally, we show how to modify our bisimilar model to obtain a tight traffic model simulating an unmodified PETC system, presenting two derived results: tighter decay rate estimation (compared to, e.g., [11]), and maximum average triggering frequency computation.

4.2 Problem Formulation

Consider system (2.2)–(2.4) with $\mathcal{T} = h\mathbb{N}$ (PETC), a quadratic Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^\mathsf{T}\boldsymbol{P}\boldsymbol{x}, \boldsymbol{P} \succ \boldsymbol{0}$, and the following assumptions:

Assumption 4.1. *System* (2.2)–(2.4) *is GES, and there exists a known constant* $0 \leq a < 1$ *such that every solution of the system satisfies* $V(\xi(t_{i+1})) \leq aV(\xi(t_i))$.

Remark 4.1. *To compute a, one can verify the implication*

$$\forall \boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}} (\forall i \in \mathbb{N}_{<k}\, \boldsymbol{x}^\mathsf{T}\boldsymbol{N}(hi)\boldsymbol{x} \leq 0) \text{ and } (\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(hk)\boldsymbol{x} > 0) \implies V(\boldsymbol{M}(hk)\boldsymbol{x}) \leq aV(\boldsymbol{x})$$

for every $k \in \{1, \ldots \bar{k}\}$. *This can be cast as a set of LMIs through the S-procedure.*

Assumption 4.2. *For system* (2.2)*, there exists some* $h_{\mathrm{P}} > 0$ *such that the periodic sampling sequence with* $t_{i+1} = t_i + h_{\mathrm{P}}$ *ensures* $V(\xi(t_{i+1}) \leq V(\xi(t_i))$.

This will not necessarily follow from Assumption 4.1; however, ETC is typically designed based on a continuous-time Lyapunov function, and for small enough values of h, the same Lyapunov function will work for periodic control.[1]

Assumption 4.3. *A value* $V_0 > 0$ *is known such that* $\xi(0) \in \mathcal{X}_0 = \{\boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{p}}} : V(\boldsymbol{x}) \leq V_0\}$.

Now let us propose a modification to the PETC system. Since ETC can reduce communication frequency while ensuring a fast decay rate, it makes practical sense to focus on ETC during the transient period. However, once states are close enough to the origin, decay rates have disputable practical relevance. Therefore, we admit that, when $\hat{\xi}(t)$ enters a small sublevel set $\mathcal{X}_{\mathrm{P}} \coloneqq \{x \in \mathbb{R}^{n_{\mathrm{p}}} | V(x) \leq rV_0\}, r < 1$, the controller can switch to periodic sampling, with h_{P} significantly bigger than h; in fact, it can be as big as possible, provided that it preserves Assumption 4.2. This results in more predictable (hence schedulable) traffic while retaining a reduction of traffic. Let us denote by $t_{i+1}(t_i, \xi(t_i))|_{\mathrm{PETC}}$ the solution of Eq. (2.4). Mathematically, the mixed sampling strategy, hereafter denoted MPETC, dictates the sampling times as follows:

$$\begin{aligned} t_{i+1} &= t_{i+1}(t_i, \xi(t_i))|_{\mathrm{PETC}}, & V(\xi(t_i)) > rV_0 \\ t_{i+1} &= t_i + h_{\mathrm{P}}, & V(\xi(t_i)) \leq rV_0. \end{aligned} \tag{4.1}$$

Hereafter, let $\mathcal{X}_0 \coloneqq \{\boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}} | V(\boldsymbol{x}) \leq V_0\}$ and $\mathcal{X}_{\mathrm{P}} \coloneqq \{\boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}} | V(\boldsymbol{x}) \leq rV_0\} = r\mathcal{X}_0$. This system has the following infinite-state traffic model: $\mathcal{S}_{\mathrm{E}} \coloneqq (\mathcal{X}, \mathcal{X}_0, \mathcal{E}_{\mathrm{PETC}} \cup \mathcal{E}_{\mathrm{P}}, \mathcal{Y}_{\mathrm{E}}, H_{\mathrm{E}})$ where

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_0; \\ \mathcal{E}_{\mathrm{PETC}} &= \{(\boldsymbol{x}, \boldsymbol{x}') \in (\mathcal{X} \setminus \mathcal{X}_{\mathrm{P}}) \times \mathcal{X} \mid \boldsymbol{x}' = \xi_{\boldsymbol{x}}(h\kappa(\boldsymbol{x}))\}; \\ \mathcal{E}_{\mathrm{P}} &= \{(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}_{\mathrm{P}} \times \mathcal{X}_{\mathrm{P}} \mid \boldsymbol{x}' = \xi_{\boldsymbol{x}}(h_{\mathrm{P}})\}; \\ \mathcal{Y}_{\mathrm{E}} &= \{h, 2h, ..., \bar{k}h\} \cup \{h_{\mathrm{P}}\}; \\ H_{\mathrm{E}}(\boldsymbol{x}) &= \begin{cases} h\kappa(x), & \boldsymbol{x} \in \mathcal{X} \setminus \mathcal{X}_{\mathrm{P}}, \\ h_{\mathrm{P}}, & \boldsymbol{x} \in \mathcal{X}_{\mathrm{P}}. \end{cases} \end{aligned} \tag{4.2}$$

For states starting outside $\mathcal{X}_{\mathrm{P}}$, transitions and outputs (the inter-sample times) are dictated by the PETC strategy; for states inside $\mathcal{X}_{\mathrm{P}}$, transitions and outputs are dictated by periodic sampling. Note that $\mathcal{E}_{\mathrm{P}}$ is defined over $\mathcal{X}_{\mathrm{P}} \times \mathcal{X}_{\mathrm{P}}$, i.e., states starting in $\mathcal{X}_{\mathrm{P}}$ always land in $\mathcal{X}_{\mathrm{P}}$: this comes from the forward invariance of sublevel sets of V due to Assumption 4.2. We are ready to define our main problem:

Problem 4.1. *Considering Assumptions 4.1–4.3, determine if* $\mathcal{S}_{\mathrm{E}}$ *admits a computable finite-state bisimulation. If so, provide an algorithm to compute it.*

[1] This is easy to see when one considers the first order approximation of the discrete-time transition matrix $\mathrm{e}^{Ah} \cong \mathbf{I} + Ah$. If the continuous-time Lyapunov inequality $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} \preceq -\epsilon\mathbf{I}$ holds for some $\epsilon > 0$ then: $h(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}) \preceq -h\epsilon\mathbf{I} \iff (\mathbf{I} + \boldsymbol{A}h)^{\mathsf{T}}\boldsymbol{P}(\mathbf{I} + \boldsymbol{A}h) - \boldsymbol{P} \preceq -h\epsilon\mathbf{I} + h^2\boldsymbol{A}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{A}$, which for sufficiently small h results in $\mathrm{e}^{\boldsymbol{A}^{\mathsf{T}}h}\boldsymbol{P}\mathrm{e}^{\boldsymbol{A}h} - \boldsymbol{P} \preceq -h\epsilon\mathbf{I} \prec \mathbf{0}$, i.e. the discrete-time Lyapunov inequality.

4.3 Main Result

To build a bisimilar model of $\mathcal{S}_\mathrm{E}$, the main observation is that eventually all trajectories of the system (2.2), (4.1) enter $\mathcal{X}_\mathrm{P}$, which follows from Assumption 4.1. Clearly, when in $\mathcal{X}_\mathrm{P}$ the system admits a trivial, single-state traffic bisimulation:

Proposition 4.1. *Define $H_\mathrm{P} : \mathbb{R}^{n_\mathrm{x}} \longrightarrow \mathbb{R}$ such that $H_\mathrm{P} \equiv h_\mathrm{P}$. The system*

$$\mathcal{S}_\mathrm{P}^\mathrm{B} = (\{\mathcal{X}_\mathrm{P}\}, \{\mathcal{X}_\mathrm{P}\}, \{(\mathcal{X}_\mathrm{P}, \mathcal{X}_\mathrm{P})\}, \{h_\mathrm{P}\}, H_\mathrm{P})$$

is a bisimilar quotient system of

$$(\mathcal{X}_\mathrm{P}, \mathcal{X}_\mathrm{P}, \mathcal{E}_\mathrm{PETC} \cup \mathcal{E}_\mathrm{P}, \mathcal{Y}_\mathrm{E}, H_\mathrm{E}).$$

Another important observation is that, since the PETC system is asymptotically stable (Assumption 4.1), states from $\mathcal{X}_0$ reach $\mathcal{X}_\mathrm{P}$ in finite time. Thus, for any state in $\mathcal{X}_0$, there is a finite number of PETC-generated samples, after which all samples are periodically taken. Let $K \coloneqq \{\underline{k}, \underline{k}+1, ..., \bar{k}\}$; since at each step there are finitely many ($|K|$) inter-sample time possibilities, we can state the following:

Lemma 4.1. *Let Assumptions 4.1–4.3 hold, define $N \coloneqq \lceil \log_a(r) \rceil$. Then system* (4.2) *can produce at most $|K|((|K|-1)^N - 1)(|K|-1)^{-1}$ different traces.*

Proof. Using Assumption 4.1, recursively apply $V(\xi(t_{i+1}) \leq aV(\xi(t_i))$ to get $V(\xi(t_N) \leq a^N V(\boldsymbol{x}_0) \leq a^N V_0$. Then, $N > \log_a(r)$ implies $a^N V_0 \leq rV_0$; thus, it takes at most N steps to enter $\mathcal{X}_\mathrm{P}$. After this, from Proposition 4.1, the remaining trace is $h_\mathrm{P}, h_\mathrm{P}, ...$ This is the trace if $\boldsymbol{x}_0 \in \mathcal{X}_\mathrm{P}$, which accounts for one trace; $\mathcal{S}_\mathrm{E}$ has at most $|K|$ traces for which it takes one step to reach $\mathcal{X}_\mathrm{P}$ from $\boldsymbol{x}_0$, at most $|K|^2$ traces for which it takes two steps to reach $\mathcal{X}_\mathrm{P}$, and etc., up to $|K|^N$ for the maximum number of steps. Summing up this geometric series gives $|K|((|K|-1)^N - 1)(|K|-1)^{-1}$. □

Lemma 4.1 permits the construction of a rather straightforward finite-state model similar to $\mathcal{S}_\mathrm{E}$. Denote by K^m the set of all sequences of length m of the form $(k_i)_{i=0}^m, k_i \in K$. We create one state for each sequence in K^m. The state $k_1 k_2 ... k_m$ is associated with the trace $hk_1, hk_2, ..., hk_m, h_\mathrm{P}, h_\mathrm{P}, ...$, thus taking m samples to enter the periodic sampling region. By definition, its successor must be $k_2 ... k_m$. Finally, let ε denote the empty sequence; a state k generates the trace $hk, h_\mathrm{P}, h_\mathrm{P}, ...$, and thus its successor is ε, associated with the periodic sampling region. Hence, $\mathrm{Post}(\varepsilon) = \varepsilon$ and $H^\mathrm{S}(\varepsilon) = h_\mathrm{P}$. Let $\mathbb{K}_N \coloneqq \cup_{i=1}^N K^i \cup \{\varepsilon\}$; we consolidate this modeling strategy with the following result:

Proposition 4.2. *Let Assumptions 4.1–4.3 hold and $N \coloneqq \lceil \log_a(r) \rceil$. Consider the transition system $\mathcal{S}^\mathrm{S} \coloneqq \left(\mathbb{K}_N, \mathbb{K}_N, \mathcal{E}^\mathrm{S}, \mathcal{Y}_\mathrm{E}, H^\mathrm{S}\right)$ with*

- *$\mathcal{E}^\mathrm{S} = \{(k\sigma, \sigma) | k\sigma \in \mathbb{K}_N\} \cup \{(\varepsilon, \varepsilon)\}$;*
- *$H^\mathrm{S}(k\sigma) = hk$ and $H^\mathrm{S}(\varepsilon) = h_\mathrm{P}$.*

Then $\mathcal{S}^\mathrm{S} \succeq \mathcal{S}_\mathrm{E}$.

Proof. System $\mathcal{S}^\mathrm{S}$ generates all possible traces of type $hk_1, hk_2, ..., hk_m, h^*, h^*, ...$, for $0 \leq m \leq N$, which, according to Lemma 4.1, include all possible traces of $\mathcal{S}_\mathrm{E}$; thus, the behavior of $\mathcal{S}^\mathrm{S}$ contains that of $\mathcal{S}_\mathrm{E}$. Because both systems $\mathcal{S}_\mathrm{E}$ and $\mathcal{S}^\mathrm{S}$ are deterministic and non-blocking, this implies that $\mathcal{S}^\mathrm{S} \succeq \mathcal{S}_\mathrm{E}$ [17, Proposition 4.11]. □

The set $\mathbb{K}_N$ includes sequences that may not be generated by the PETC opereation. To trim off these spurious sequences, let us define the following relation:

Definition 4.1 (MPETC inter-sample sequence relation)**.** *We denote by* $\mathcal{R}_{\mathrm{B}} \subseteq \mathcal{X} \times \mathbb{K}_N$ *the relation satisfying*

$$(\boldsymbol{x}, \varepsilon) \in \mathcal{R}_{\mathrm{B}} \text{ iff } \boldsymbol{x} \in \mathcal{X}_{\mathrm{P}}, \tag{4.3}$$

and $(\boldsymbol{x}, k_1 k_2 ... k_m) \in \mathcal{R}_{\mathrm{B}}$ *if and only if*

$$\begin{aligned}
\boldsymbol{x} &\in \mathcal{X}_0, && \text{(4.4a)}\\
\boldsymbol{x} &\in \mathcal{Q}_{k_1}, && \text{(4.4b)}\\
\boldsymbol{M}(hk_1)\boldsymbol{x} &\in \mathcal{Q}_{k_2}, && \text{(4.4c)}\\
\boldsymbol{M}(hk_2)\boldsymbol{M}(k_1)\boldsymbol{x} &\in \mathcal{Q}_{k_3}, && \text{(4.4d)}\\
&\vdots\\
\boldsymbol{M}(hk_{m-1})...\boldsymbol{M}(hk_1)\boldsymbol{x} &\in \mathcal{Q}_{k_m}, && \text{(4.4e)}\\
\boldsymbol{x} &\notin \mathcal{X}_{\mathrm{P}}, && \text{(4.4f)}\\
\boldsymbol{M}(hk_1)\boldsymbol{x} &\notin \mathcal{X}_{\mathrm{P}}, && \text{(4.4g)}\\
&\vdots\\
\boldsymbol{M}(hk_{m-1})...\boldsymbol{M}(hk_1)\boldsymbol{x} &\notin \mathcal{X}_{\mathrm{P}}, && \text{(4.4h)}\\
\boldsymbol{M}(hk_m)...\boldsymbol{M}(hk_1)\boldsymbol{x} &\in \mathcal{X}_{\mathrm{P}}, && \text{(4.4i)}
\end{aligned}$$

where the sets $\mathcal{Q}_k$ *are defined in* (3.2).

Eq. (4.3) determines that states $\boldsymbol{x} \in \mathcal{X}_{\mathrm{P}}$ are related to the state ε. Finally, a state $\boldsymbol{x} \in \mathbb{R}^n$ is related to a state $k_1 k_2 ... k_m$ of the abstraction if the following are satisfied: 1) it belongs to the compact set of interest (Eq. (4.4a)), 2) the inter-sample time sequence that it generates up until it enters $\mathcal{X}_{\mathrm{P}}$ is $hk_1, hk_2, ..., kh_m$ (Eqs. (4.4b)–(4.4e)), and 3) the sampled states $\xi_{\boldsymbol{x}}(k_1 h), \xi_{\boldsymbol{x}}((k_1 + k_2)h), ...$ of the trajectory starting from $\boldsymbol{x}$ do not belong to $\mathcal{X}_{\mathrm{P}}$ (Eqs. (4.4f)–(4.4h)), while the m-th sampled state does belong to $\mathcal{X}_{\mathrm{P}}$ (Eq. (4.4i)).

We now employ the relation $\mathcal{R}_{\mathrm{B}}$ to derive a finite model bisimilar to $\mathcal{S}_{\mathrm{E}}$ as follows:

Definition 4.2. *The* MPETC finite traffic model *is the system* $\mathcal{S}^{\mathrm{B}} := \left(\mathcal{X}^{\mathrm{B}}, \mathcal{X}^{\mathrm{B}}, \mathcal{E}^{\mathrm{S}}, \mathcal{Y}_{\mathrm{E}}, H^{\mathrm{S}}\right)$ *with* $\mathcal{X}^{\mathrm{B}} := \pi_{\mathcal{R}_{\mathrm{B}}}(\mathcal{X})$.

This model is a subset of $\mathcal{S}^{\mathrm{S}}$, generating only inter-sample sequences that can be produced by the concrete system $\mathcal{S}_{\mathrm{E}}$. Topologically, it is still a tree, such as $\mathcal{S}^{\mathrm{S}}$, but with fewer states (see Figure 4.1). Our main result follows:

Theorem 4.1. *Let Assumptions 4.1–4.3 hold and* $N := \lceil \log_a(r) \rceil$. *Then,* $\mathcal{S}^{\mathrm{E}} \approxeq \mathcal{S}^{\mathrm{B}}$.

Proof. We show that $\mathcal{R}_{\mathrm{B}}$ is a simulation relation from $\mathcal{S}_{\mathrm{E}}$ to $\mathcal{S}^{\mathrm{B}}$ and $\mathcal{R}_{\mathrm{B}}^{-1}$ is a simulation relation from $\mathcal{S}^{\mathrm{B}}$ to $\mathcal{S}_{\mathrm{E}}$, checking each of the conditions of Definition 2.2.

Step 1: $\mathcal{R}_{\mathrm{B}}$ is a simulation relation from $\mathcal{S}_{\mathrm{E}}$ to $\mathcal{S}^{\mathrm{B}}$.

For condition (i), take a point $\boldsymbol{x}_0 \in \mathcal{X}_0 = \mathcal{X}$. It either belongs to $\mathcal{X}_{\mathrm{P}}$, for which Eq. (4.3) provides its related state; or it takes m PETC steps to reach $\mathcal{X}_{\mathrm{P}}$. In this latter case, it generates some trace $hk_1, hk_2, ..., hk_m, h^*, h^*, ...$ and therefore, by definition, it satisfies Eq. (4.4). Hence,

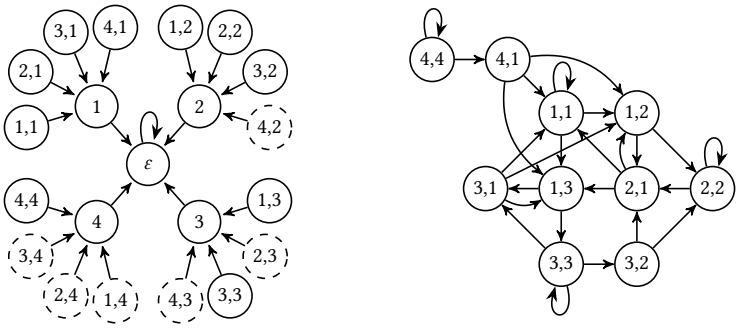


Figure 4.1: On the left, an illustration of $\mathcal{S}^{\mathrm{S}}$ (all states) and $\mathcal{S}^{\mathrm{B}}$ (only solid-line states). On the right, a depiction of $\mathcal{S}^{\mathcal{S}'}$ with $\mathcal{X}^{\mathcal{S}'} = \{\sigma \in \mathcal{X}^{\mathrm{B}} : |\sigma| = 2\}$.

4

the related state $k_1k_2...k_m$ belongs to $\mathcal{X}^{\mathrm{B}}$. Condition (ii) trivially holds by the definition of $\mathcal{R}_{\mathrm{B}}$, and in particular by (4.3) and (4.4b).

Finally, for condition (iii), take $(\boldsymbol{x}, \sigma) \in \mathcal{R}_{\mathrm{B}}$. If $\boldsymbol{x} \in \mathcal{X}_{\mathrm{P}}$, then $\sigma = \varepsilon$. From Assumption 4.2, $\mathrm{Post}_{\mathcal{S}_{\mathrm{E}}}(\boldsymbol{x}) \in \mathcal{X}_{\mathrm{P}}$, which is related to $\varepsilon = \mathrm{Post}_{\mathcal{S}^{\mathrm{B}}}(\varepsilon)$. If $\boldsymbol{x} \notin \mathcal{X}_{\mathrm{P}}$, then $\sigma = k_1\sigma' \in \mathbb{K}_N$. Therefore, $\mathrm{Post}_{\mathcal{S}_{\mathrm{E}}}(\boldsymbol{x}) = \boldsymbol{M}(k_1)\boldsymbol{x}$. From Assumption 4.1, $\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{X}_0$; also, by inspecting Eq. (4.4), $\boldsymbol{M}(k_1)\boldsymbol{x}$ satisfies Eqs. (4.4c)–(4.4i) and Eqs. (4.4g)–(4.4h): this implies that $k_2...k_m = \sigma' = \mathrm{Post}_{\mathcal{S}^{\mathrm{B}}}(k_1\sigma')$ is related to $\boldsymbol{M}(k_1)\boldsymbol{x}$.

Step 2: $\mathcal{R}_{\mathrm{B}}^{-1}$ is a simulation relation from $\mathcal{S}^{\mathrm{B}}$ to $\mathcal{S}_{\mathrm{E}}$.

For condition (i), if $k_1k_2...k_m \in \mathcal{X}_0^{\mathrm{B}}$, then there exists a related initial state $\boldsymbol{x}_0$ which satisfies Eq. (4.4); hence, from Eq. (4.4a), $\boldsymbol{x}_0 \in \mathcal{X}_0$. For ε, any related state $\boldsymbol{x}_0$ belongs to $\mathcal{X}_{\mathrm{P}} \subset \mathcal{X} = \mathcal{X}_0$. Condition (ii) is the same as in Step 1. Finally, condition (iii) is verified because the reasoning in Step 1 applies to every $\boldsymbol{x} \in \mathcal{X}$ satisfying $(\sigma, \boldsymbol{x}) \in \mathcal{R}_{\mathrm{B}}^{-1}$. □

Remark 4.2. *Determining if there exists $\boldsymbol{x}$ satisfying Eq.* (4.4) *is a problem of checking non-emptiness of a semi-algebraic set, which has been proven to be decidable [66]. One tool that can be used to solve it is the SMT solver Z3 [64]. This requires unfolding the memberships of Eq.* (4.4) *into the conjunctions of quadratic inequalities by applying Eq.* (3.2).

For example, suppose we want to verify the sequence $\sigma = (3, 2)$. First we convert $\boldsymbol{x} \in \mathcal{Q}_3$, which is equivalent to $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(1)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(2)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(3)\boldsymbol{x} > 0$ if $3 < \bar{k}$, or just $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(1)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(2)\boldsymbol{x} \leq 0$ if $3 = \bar{k}$. Then we add constraints associated to $\boldsymbol{M}(3)\boldsymbol{x} \in \mathcal{Q}_2$, which are $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{N}(1)\boldsymbol{M}(3)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{N}(2)\boldsymbol{M}(3)\boldsymbol{x} > 0$, to the constraint set. The next step is to add equations (4.4f)–(4.4i)*, which translate to $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{x} > r, \boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{P}\boldsymbol{M}(3)\boldsymbol{x} > r$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{M}(2)^{\mathsf{T}}\boldsymbol{P}\boldsymbol{M}(2)\boldsymbol{M}(3)\boldsymbol{x} \leq r$. Finally, we add the condition* (4.4a)*, which is $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{x} \leq 1$. The final SMT query to check whether* $(3, 2)$ *is a behavior of the system then becomes "$\exists \boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}}$ such that $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(1)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(2)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}(3)\boldsymbol{x} > 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{N}(1)\boldsymbol{M}(3)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{N}(2)\boldsymbol{M}(3)\boldsymbol{x} > 0$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{x} > r$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{P}\boldsymbol{M}(3)\boldsymbol{x} > r$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}(3)^{\mathsf{T}}\boldsymbol{M}(2)^{\mathsf{T}}\boldsymbol{P}\boldsymbol{M}(2)\boldsymbol{M}(3)\boldsymbol{x} \leq r$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{x} \leq 1$."*

Proposition 4.3 (Complexity)**.** *The set $\mathcal{X}^{\mathrm{B}}$ can be computed with $\mathcal{O}\left(|K|^N(N|K|)^{n_{\mathrm{x}}}\right) 2^{\mathcal{O}(n_{\mathrm{x}})}$ operations.*

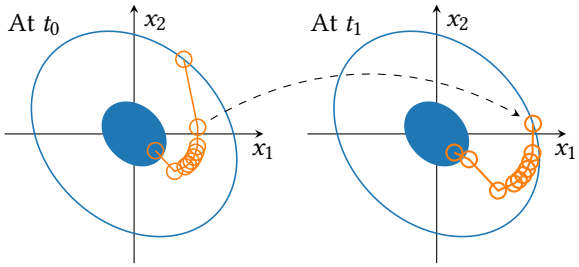


Figure 4.2: Depiction of the strategy used to build the PETC traffic abstraction: the trajectory $\hat{\xi}(t)$ is in orange with samples marked, and $\mathcal{X}_\mathrm{P}$ is the blue ellipse. The tail of the sequence from $t_0 = 0$ is the head of the sequence from the following sample t_1.

Proof. From Lemma 4.1, we have seen that there can be at most $|K|((|K|-1)^N - 1)(|K|-1)^{-1} \in \mathcal{O}(|K|^N)$ sampling sequences. Determining the state set $\mathcal{X}^\mathrm{B}$ requires, in the worst case, checking the existence of all of those sequences. For a sequence of length m, Eq. (4.4) has one membership in $\mathcal{X}_0$ and m memberships in $\mathcal{X}_\mathrm{P}$, each corresponding to one quadratic inequality; and m memberships in $\mathcal{Q}_k$, each corresponding to $k - \underline{k} + 1$ quadratic inequalities. Therefore, in the worst case, Eq. (4.4) has $m + 1 + m|K|$ inequalities, or $1 + N + N|K| \in \mathcal{O}(N|K|)$ for the longest sequence. The best known bound for deciding the existence of a real solution to a conjunction of s polynomial inequalities of n_x variables and maximum degree d is $s^{n_\mathrm{x}+1} d^{\mathcal{O}(n_\mathrm{x})}$ [67]. Replacing s by $1 + N + N|K|$ and d by 2, multiplying by the number of checks and working out the limits for big-O notation concludes the proof. □

Remark 4.3. *Whilst all sequences of length N must be checked in the worst case, it is generally more efficient to employ a recursive algorithm, i.e., verifying Eq.* (4.4) *for sequences from length 1 to N. If a sequence σ shorter than N does not verify Eq.* (4.4)*, then no sequence $k\sigma$ can do. Hence, many checks can be discarded using this simple observation.*

Remark 4.4. *Due to characteristics of the inequalities associated to Definition 4.1, one can set $V_0 = 1$ without loss of generality, with the only input to the model being the ratio of contraction r. For $V_0 = c > 0, c \neq 1$, the model is the same: replace $\boldsymbol{x}$ by $\sqrt{c}\boldsymbol{x}$ in Eq.* (4.4)*, and $\sqrt{c}$ is canceled out in all inequalities.*

4.3.1 Derived results for the original PETC

With a few changes to $\mathcal{S}^\mathrm{B}$, we can build a similar model of the PETC traffic that generates fewer spurious traces than the quotient model of Chap. 3. This is because the PETC section of the MPETC trace is of course generated by the pure PETC system (2.2)–(2.10). Hence, to simulate the PETC traffic, one could do the following: for a given state $\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}}$, take $V_0 = V(\boldsymbol{x})$ and determine its related state $k\sigma$ from Eq. (4.4). Now take its successor $\boldsymbol{M}(k)\boldsymbol{x}$. Again, set $V_0 = V(\boldsymbol{M}(k)\boldsymbol{x})$ and determine its related state: it has to take the form $\sigma\sigma'$, i.e., its first inter-sample times should be all but the first inter-sample times of its predecessor. This idea is depicted in Fig. 4.2. Let us formalize this procedure.

Definition 4.3 (PETC inter-sample sequence relation)**.** *Let $V_0 = 1$. The relation $\mathcal{R}_{\mathcal{S}'} \subseteq \mathbb{R}^{n_\mathrm{x}} \times \mathbb{K}_N$ is given by: $(\boldsymbol{x}, k_1 k_2 \ldots k_m) \in \mathcal{R}_{\mathcal{S}'}$ iff $\boldsymbol{x}/\sqrt{V(\boldsymbol{x})}$ satisfies Eq.* (4.4)*.*

Theorem 4.2. *Let Assumption 4.1 hold. Then, the system* $\mathcal{S}_{S'} := (\mathcal{X}^{S'}, \mathcal{X}^{S'}, \mathcal{E}^{S'}, \mathcal{Y}_E, H^S)$*, with* $\mathcal{X}^{S'} := \pi_{\mathcal{R}_{S'}}(\mathbb{R}^{n_x})$ *and* $\mathcal{E}^{S'} = \{(k\sigma, \sigma\sigma') \mid k\sigma, \sigma\sigma' \in \mathcal{X}^{S'}\}$*, simulates the traffic generated by System* (2.2)–(2.4).

Proof. Take an initial state $\boldsymbol{x} \in \mathbb{R}^{n_x}$, a PETC trajectory $\xi_{\boldsymbol{x}}(t)$ and its associated inter-sample sequence $k_1k_2...k_m$, after which $V(\xi_{\boldsymbol{x}}(t_m)) \le rV(\boldsymbol{x})$ but $V(\xi_{\boldsymbol{x}}(t_{m-1})) > rV(\boldsymbol{x})$. This implies that $\boldsymbol{x}/\sqrt{V(\boldsymbol{x})}$ satisfies Eq. (4.4). Hence, $(\boldsymbol{x}/\sqrt{V(\boldsymbol{x})}, k_1k_2...k_m) \in \mathcal{R}_{S'}$, and condition (i) of Def. 2.2 holds. Condition (ii) is trivially satisfied, as $H(\boldsymbol{x}) = hk_1 = H^S(k_1k_2...k_m)$. For condition (iii), take its related sequence $k_1k_2...k_m$. The successor of $\boldsymbol{x}$ is $\boldsymbol{x}' := \xi_{\boldsymbol{x}}(hk_1) = \boldsymbol{M}(k_1)\boldsymbol{x}$, which satisfies Eqs. (4.4c)–(4.4i) and Eqs. (4.4g)–(4.4h); from homogeneity of $\mathcal{Q}_k$, $\boldsymbol{M}(k_1)\boldsymbol{x}/\sqrt{V(\boldsymbol{M}(k_1)\boldsymbol{x})}$ also satisfies Eqs. (4.4c)–(4.4e). Additionally, because of Assumption 4.1, we have that $V(\boldsymbol{x}') < V(\boldsymbol{x})$; hence, Eqs. (4.4g)–(4.4h) holding for $\boldsymbol{x}/\sqrt{V(\boldsymbol{x})}$ imply that $V(\xi_{\boldsymbol{x}}(t_i)) > rV(\boldsymbol{x}) > rV(\boldsymbol{x}')$ for all $1 \le i \le m$, and therefore Eqs. (4.4g)–(4.4h) also hold for $\boldsymbol{x}'/\sqrt{V(\boldsymbol{x}')}$. This shows that the prefix of the sequence related to $\boldsymbol{x}'$ is $k_2...k_m$. Finally, $\boldsymbol{x}'/\sqrt{V(\boldsymbol{x}')}$ satisfies Eq. (4.4) for some sequence in $\mathcal{S}_{S'}$; combining with the conclusion about the prefix above, $(\boldsymbol{x}', \sigma\sigma') \in \mathcal{R}_{S'}$. The related transition exists because $(k\sigma, \sigma\sigma') \in \mathcal{E}^{S'}$ for every $\sigma\sigma' \in \mathcal{X}^{S'}$. □

Remark 4.5. *Effectively, to verify if* $\boldsymbol{x}/\sqrt{V(\boldsymbol{x})}$ *satisfies Eq.* (4.4) *using SMT solvers, one only needs to replace* $\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} \le 1$ *with the equality* $\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} = 1$.

Note that $\mathcal{S}_{S'}$ is, in general, nondeterministic. A depiction of such construction is seen in Fig. 4.1. Some useful verification applications can be derived from the model $\mathcal{S}_{S'}$:

Proposition 4.4. *An upper bound for the average triggering frequency of system* (2.2), (2.4) *is* $f^* = \max_{\sigma \in \mathcal{X}^{S'}}(|\sigma|/(h\sum_{k_i \in \sigma} k_i))$.

Proof. In the worst case, $\mathcal{S}_{S'}$ generates $\sigma^* := \operatorname{argmax}_{\sigma \in \mathcal{X}^{S'}}(|\sigma|/(h\sum_{k_i \in \sigma} k_i))$ repeatedly. □

Proposition 4.5. *Let* $T^* = h\max_{\sigma \in \mathcal{X}^{S'}}(\sum_{k_i \in \sigma}(k_i))$ *be the longest (time-wise) sequence in* $\mathcal{X}^{S'}$*. Then* $b^* = H$ *is an upper bound for the GES decay rate of system* (2.2),(2.4).

Proof. Take an initial state $\boldsymbol{x} \in \mathbb{R}^{n_x}$, its related sequence $\sigma \in \mathcal{X}^{S'}$, and set $T = h\sum_{k_i \in \sigma}(k_i)$. Consider two cases:

Case 1: $t < T$. From GES of the PETC (Assumption 4.1), $V(t) \le Me^{-2bt}V(0)$ for some $b > 0$ and $M < \infty$. Then,

$$V(t) \le Me^{-2bt}V(0) = Me^{-2bt+2b^*t-2b^*t}V(0) = e^{-2b^*t}Me^{-2(b-b^*)t}V(0) \le M'e^{-2b^*t}V(0), \quad (4.5)$$

where $M' := M\max(1, e^{2(b^*-b)T^*}) \ge M\max(1, e^{2(b^*-b)T})$.

Case 2: $t > T$. From Def. 4.3, $V(T) = V(\xi_{\boldsymbol{x}}(T)) \le rV_0 = e^{\log(r)}V(0) = e^{-2b^*T}V(0)$. Partition the trajectory $\xi_{\boldsymbol{x}}(t)$ in intervals $[0, t_{m_1}], [t_{m_1}, t_{m_2}], ..., [t_{m_n}, t]$ satisfying $V(t_{m_i}) \le rV(t_{m_{i-1}})$ and $V(t_{m_i}) > rV(t_{m_{i-1}-1})$. Each interval but the last is associated with a sequence $\sigma_i \in \mathcal{X}^{S'}$, and therefore its duration is $T_i \le T^*$. Thus, with $t' = t - t_{m_n}$,

$$\begin{aligned} V(t) = V\left(\textstyle\sum_{i=1}^{n} T_i + t'\right) &\le r^n V(t') \\ &\overset{\text{Eq. (4.5)}}{\le} r^n M' e^{-2b^*t'}V(0) = M'e^{-2b^*nT^*}e^{-2b^*t'}V(0) \\ &= M'e^{-2b^*(nT^*+t')}V(0) \le M'e^{-2b^*t}V(0). \end{aligned}$$

In the two cases, we have $V(t) \leq M' e^{-2b^* t} V(0)$, which implies

$$|\xi(t)| \leq \sqrt{M' \lambda_{\max}(\boldsymbol{P})/\lambda_{\min}(\boldsymbol{P})} e^{-b^* t} |\xi(0)|.$$

□

We conjecture that Proposition 4.5 provides a better estimate of the convergence rate of System (2.2)–(2.4) than what can be obtained by, e.g., the theorems in [11]. The reason behind this conjecture is that, as $N \longrightarrow \infty$ (or $r \longrightarrow 0$), our bound approaches what would be the joint spectral radius of the associated discrete-time system (see, e.g., [68]).

4.4 Numerical Results

Consider a plant and controller of the form (2.2) from [5], and the Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{P}_{\text{Lyap}} \boldsymbol{x}$ such that the continuous-time closed-loop system satisfies $\mathrm{d}V(\xi(t))/\mathrm{d}t = -\xi(t)^{\mathsf{T}} \boldsymbol{Q}_{\text{Lyap}} \xi(t)$, determined by the following matrices:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \boldsymbol{K} = \begin{bmatrix} 1 & -4 \end{bmatrix},$$

$$\boldsymbol{P}_{\text{Lyap}} = \begin{bmatrix} 1 & 0.25 \\ 0.25 & 1 \end{bmatrix}, \ \boldsymbol{Q}_{\text{Lyap}} = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 1.5 \end{bmatrix}.$$

For the PETC implementation, we use a predictive Lyapunov-based triggering condition of the form $V(\zeta(t)) > -\rho \zeta(t)^{\mathsf{T}} \boldsymbol{Q}_{\text{Lyap}} \zeta(t)$, where $\zeta(t) \coloneqq \boldsymbol{A}_{\mathrm{d}}(1)\xi(t) + \boldsymbol{B}_{\mathrm{d}}(1)\boldsymbol{K}\hat{\xi}(t)$ is the next-sample prediction of the state and $0 < \rho < 1$ is the triggering parameter. Setting $h = 0.1$ and $\bar{k} = 6$, we put it in the form (2.4), and obtained $a = 0.952$ using LMIs based on Remark 4.1. For the periodic sampling, the maximum period that verifies Assumption 4.2 is $h_{\mathrm{P}} = 0.4$ (with resolution of 0.01). Finally, we verified that $\underline{k} = 1$ and set $r = 0.1$. Lemma 4.1 gives $N = 47$ and a worst-case value of $8.5 \cdot 10^{32}$ bisimulation states. We computed the bisimilar state set $\mathcal{X}^{\mathrm{B}}$ implementing a recursive algorithm (as discussed in Remark 4.3), obtaining a total of only 219 states, out of which 84 belong to $\mathcal{X}^{\mathrm{S}'}$. The Python implementation, using Z3 to solve Eq. (4.4), took 66 seconds to generate these state sets.

Using MPETC, the maximum time it takes for $\mathcal{X}_{\mathrm{P}}$ to be reached from any initial state $\boldsymbol{x} \in \mathcal{X}_0$ is $T^* = h \max_{\sigma \in \mathcal{X}^{\mathrm{B}}} (\sum_{k_i \in \sigma} (k_i)) = 2.3$. This is highlighted in Fig. 4.3, which shows

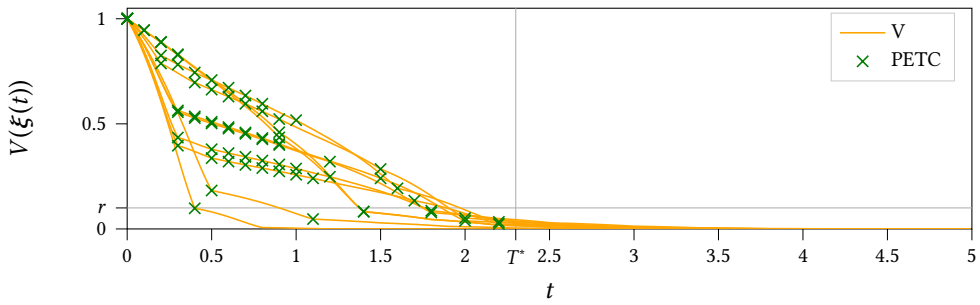


Figure 4.3: Trajectory of the Lyapunov function for 10 different initial conditions under MPETC, with PETC samples marked. The maximum time T^* it takes to reach $\mathcal{X}_{\mathrm{P}} = r\mathcal{X}_0$ is highlighted.

simulations from 10 different initial conditions. For PETC, applying Proposition 4.5 gives $b^* = 0.5$, while the best GES rate that can be obtained using the LMI approaches from [11] is $b = 0.23$, using Theorem III.4. For the average PETC sampling frequency, Proposition 4.4 gives $f^* = 20/3$ (much lower than the MIST value of $1/h = 10$), corresponding to the sequence $\sigma = (4, 1, 1, 1, 1, 1)$.

Reproducing these results. The algorithm to generate MPETC bisimulations $\mathcal{S}^{\mathrm{E}}$ and their derived PETC abstractions $\mathcal{S}_{S'}$ has been implemented in `ETCetera`. To reproduce the results above, the following scripts within `ETCetera` can be used:

- `examples/mpetc_simulations.py` for Figs. 4.2 and 4.3;
- `examples/mpetc_bisim.py` for abstraction and the remaining numbers presented herein.

4

4.5 Conclusions

We have presented a practical alternative to ETC, the MPETC, which provides the benefits of PETC during transients and the traffic predictability of periodic sampling when close to steady state. Furthermore, we have presented a method to compute a bisimilar traffic model for MPETC. In addition, we have presented some verification applications of the (bi)similar models that can be used for both PETC and MPETC. This is an important step towards understanding traffic characteristics of ETC, and it may support its applicability in real NCSs, since the traffic benefits are among the main motivations for ETC use.

Extending the models we have presented with early triggers for scheduling is easy. If one denotes by $\mathcal{Q}_\sigma$ the set of points in $\mathbb{R}^{n_x}$ such that Eq. (4.4) is satisfied, where $\sigma := k_1 k_2 ... k_m$, then checking if a transition $\sigma \xrightarrow{i} \sigma'$ exists amounts to answering whether there exists $\boldsymbol{x} \in \mathcal{Q}_\sigma$ such that $\boldsymbol{M}(hi)\boldsymbol{x} \in \mathcal{Q}_{\sigma'}$. This is also a problem of checking the existence of a real solution to a conjunction of quadratic inequalities.

Among the drawbacks of the presented approach compared to Chap. 3 is the increased computational complexity, putting at risk the scalability of the abstraction process. One could use SDRs to solve Eq. (4.4), but it is unclear whether such relaxations can be too permissive, to the point that most transitions cannot be ruled out, even when they do not exist in fact. This can also have negative effects on the practical number of computations necessary to finalize the model, as can be hinted by Remark 4.3: the number of m-long sequences to be checked is $\mathcal{O}(K)N_{m-1}$, where N_{m-1} is the number of $(m-1)$-long sequences. The smaller N_{m-1} is, the smaller the number of checks to determine the m-long sequences is. Using relaxations can only increase N_m, for all m, and hence the number of satisfiability checks can increase exponentially. A better alternative to SDR may be δ-SMT [69], which allows for better control of the precision error.

5

Computing the sampling performance of PETC

We propose an approach through finite-state abstractions to do formal quantification of the traffic generated by ETC of linear systems, in particular aiming at computing its smallest average inter-sample time (SAIST). The method involves abstracting the traffic model through l-complete abstractions, finding the cycle of minimum average length in the graph associated to it, and verifying whether this cycle is an infinitely recurring traffic pattern. Based on this, we present a semi-algorithm that can compute the exact SAIST of PETC, and conditions for its termination. The method is proven to be robust to sufficiently small model uncertainties, which allows its application to compute the SAIST of ETC of nonlinear systems.

This chapter is based on *G. de A. Gleizer and M. Mazo Jr. "Computing the sampling performance of event-triggered control", in Proc. of the 24th Int'l Conf. on Hybrid Systems: Computation and Control (HSCC '21).* [70] and *G. de A. Gleizer and M. Mazo Jr. Computing the average inter-sample time of event-triggered control using quantitative automata, under review at Nonlinear Analysis: Hybrid Systems.* [71].

5.1 Introduction

Even though the major benefit of ETC compared to traditional periodic sampling is the reduction in resource usage, until recently little was formally known about its traffic-related performance. Existing work can be divided in two methods: analytical vs. abstraction-based.[1]

The earliest work is due to Demirel et al. [24], and can be seen as a mix of abstraction-based and analytical, where they compute the average inter-sample time (IST) of linear event-triggered *deadbeat* controllers subject to Gaussian disturbances. The deadbeat nature of the controller makes the system cease to be history-dependent after finite time, which allows for the sampling behavior to be abstracted as a simple Markov chain. The sampling performance can then be obtained analytically from this construction. Unfortunately, the whole methodology is critically dependent on the deadbeat assumption, and cannot be extended to a general class of linear controllers. The remaining papers that use analytical methods [25, 26, 73] are dedicated to two-dimensional linear time-invariant (LTI) systems, and some conditions are given to show when traffic converges to periodic sampling or oscillatory patterns. While they are mostly interested in this *qualitative* understanding, [73] allows to approximately compute average inter-sample for such planar systems when the triggering parameters are sufficiently small.

The abstraction-based methods, such as the ones presented in Chapters 3 and 4, aim majorly at ETC scheduling, with Props. 4.4 and 4.5 being the only results in this category to give some *quantitative* performance indicator. Prop. 4.4 is the one aimed at sampling performance, and essentially chooses the highest-in-frequency sequence of inter-sample times as an upper bound to the average frequency that ETC exhibits. However, it is clear that this bound may be too conservative: consider the numerical example of Section 4.4; the sequence that maximizes frequency according to Prop. 4.4 is $(4,1,1,1,1,1)$. But what if this sequence only occurs transiently? E.g., if $(4,1,1,1,1,1)^\omega$ is not a behavior of the PETC system, it might be that the average frequency exhibited by the system is in fact lower.

This chapter tackles the precise computation of the smallest (across initial states) average inter-sample time (SAIST) of LTI systems under PETC. The SAIST constitutes a natural metric which directly translates into average resource utilization in a network. Our approach is based on the abstraction of a closed-loop PETC system into a weighted transition system (WTS, Def. 2.7), where the weight of a transition is the IST generated by the state. Instead of using the smallest-in-average sequence as in Prop. 4.4, we use the smallest-in-average *cycle* (SAC) of the weighted graph associated with the abstraction using Karp's algorithm (Theorem 2.2), which we prove to be a lower bound of the PETC's SAIST. If $\sigma := k_1 k_2 ... k_m$ is the SAC of the abstraction and σ^ω is a behavior of the concrete system, the lower bound is proven to be the exact SAIST of the PETC system. This observation gives rise to the concept of *smallest-average-cycle-equivalent simulation (SACE simulation)*. If the smallest cycle is not exhibited by the PETC system, the abstraction can be further refined until the cycle breaks, providing tighter bounds. This gives rise to a semi-algorithm to compute average metrics through abstractions. This algorithmic framework can in fact

[1] It is also worth mentioning the approach of [72], which proposed an event-triggering mechanism that ensures given traffic criteria in terms of a token bucket model. Despite very interesting, we veer away from this approach because it is unclear whether adding conditions to enforce traffic patterns could actually degrade the sampling performance of the original mechanism.

be applied to the calculation of limit average metrics of general infinite systems, as long as the weights belong to a finite set. When specialized to PETC SAIST, this semi-algorithm is shown to be robust to sufficiently small model uncertainties, which allows its application to PETC of smooth nonlinear systems.

This chapter follows the following structure: The main problem is stated in Section 5.2. The concept of SACE simulation and its associated basic results are shown in Section 5.3. Then, a general semi-algorithm to compute limit average metrics of infinite systems is presented in Section 5.4, while its specialization for PETC SAIST computation is presented in Section 5.5. Finally, numerical examples are given in Section 5.6, and conclusions and future work are discussed in 5.7.

5.2 Problem Statement

Consider a PETC system (2.2)–(2.4) with $\mathcal{T} = h\mathbb{N}$. Throughout this chapter, we assume the time units have been scaled so that $h = 1$.[2] Recall that the samples evolved according to the discrete time model

$$\begin{aligned} \boldsymbol{x}_{i+1} &= f(\boldsymbol{x}_i), \\ y_i &= \tau(\boldsymbol{x}_i). \end{aligned} \tag{2.8 revisited}$$

Because the system is deterministic, there is a unique sequence of inter-event times from a given initial state $\boldsymbol{x}_0$. Let us denote it by $\{y_i(\boldsymbol{x}_0)\}$. With this, we can attribute an *average inter-sample time* (AIST) to every initial state:

$$\mathrm{AIST}(\boldsymbol{x}) \coloneqq \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} y_i(\boldsymbol{x}).$$

Using lim inf instead of lim lets us use the limit lower bound in case the regular limit does not exist, making the AIST metric well-defined.

Objective of this chapter. We want to devise a method to compute the exact *smallest average inter-sample time* (SAIST) of the PETC system (2.2)–(2.4); i.e., the infimal AIST across all possible initial states:

$$\mathrm{SAIST} \coloneqq \inf_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} y_i(\boldsymbol{x}). \tag{5.1}$$

Furthermore, we want to understand the cases where the exact SAIST computation is not possible, and quantify the estimation error if the best we can obtain is an approximation.

The mere application of Eq. (5.1) is largely unpromising: how can one choose a sufficiently large n, or how can one exhaustively search for initial states to obtain one that yields the SAIST? For this reason, we approach the SAIST computation problem through finite-state abstractions.

Remark 5.1. *The way we define SAIST implies that we do not expect that a system's AIST is irrespective of its initial conditions; as we shall see later in Section 5.6, it is possible that multiple AISTs are observed. Hence, in these cases, we conservatively take the smallest.*

[2] This time re-scaling can be achieved by simply multiplying $\boldsymbol{A}$ and $\boldsymbol{B}$ with h.

5.3 SACE simulations and error bounds

From Theorem 2.2, we have an indication that it would be relatively straightforward to compute the minimum average inter-sample time of an infinite system if we could represent it as a weighted automaton, such as an abstraction. The question is how meaningful the value obtained from the abstraction is for the concrete system. First, recall that ETC traffic models are simple WTS (Defs. 2.14 and 2.15); as such, we can characterize weight sequences, hence run values, exclusively by external behaviors. Considering the notion of behavioral inclusion, this gives a straightforward result:[3]

Proposition 5.1. *If two simple WTSs $\mathcal{S}_a$ and $\mathcal{S}_b$ satisfy $\mathcal{S}_a \preceq_{\mathcal{B}} (\cong_{\mathcal{B}}) \mathcal{S}_b$, then* $\mathrm{ILA}(\mathcal{S}_a) \geq (=) \mathrm{ILA}(\mathcal{S}_b)$ *and* $\mathrm{SLA}(\mathcal{S}_a) \leq (=) \mathrm{SLA}(\mathcal{S}_b)$.

Proof. Since the systems are simple, $\mathcal{V}(\mathcal{S}_s) = \mathcal{B}^\omega(\mathcal{S}_s), s \in \{a, b\}$. By definition, $\mathrm{SLA}(\mathcal{S}_s) = \inf\{\mathrm{LimAvg}(\{y_i\}) \mid \{y_i\} \in \mathcal{V}(\mathcal{S}_s)\} = \inf\{\mathrm{LimAvg}(\{y_i\}) \mid \{y_i\} \in \mathcal{B}^\omega(\mathcal{S}_s)\}$. Since $\mathcal{B}^\omega(\mathcal{S}_a) \subseteq (=) \mathcal{B}^\omega(\mathcal{S}_b)$, and the inferior of the function on a set can only be smaller than that of a set contained in it, the desired result follows for ILA. For SLA, the same reasoning is applied symmetrically. □

5

Proposition 5.1 hints that obtaining a finite-state (bi)simulation of a PETC system provides means to compute a lower bound (or the actual value) for its SAIST. Hence we could use the abstractions from previous chapters to compute the metrics of interest. However, on the one hand, a simulation alone does not provide how conservative the lower bound may be; on the other hand, a finite-state bisimulation of an infinite system is often impossible to be obtained. In fact, bisimulation is excessively strong, in the sense that all behaviors and their fragments are exactly captured. As hinted by Theorem 2.2, the ILA is determined by a minimum average cycle of the system. If one such cycle happens to have a correspondence with the concrete system, this is sufficient to obtain the exact value for the SAIST. This suggests the following abstraction:

Definition 5.1 (Smallest-average-cycle-equivalent simulation)**.** *Consider two simple WTSs $\mathcal{S}_a$ and $\mathcal{S}_b$ satisfying $\mathcal{S}_a \preceq \mathcal{S}_b$. Let* $\mathrm{SAC}(\mathcal{S}_b)$ *be the set of smallest-in-average cycles of $\mathcal{S}_b$. If there exists a behavior of the form $dc^\omega \in \mathcal{B}^\omega(\mathcal{S}_a)$ where d is finite and $c \in \mathrm{SAC}(\mathcal{S}_b)$, then $\mathcal{S}_b$ is a* smallest-average-cycle-equivalent (SACE) simulation *of $\mathcal{S}_a$.*

A SACE simulation is a normal simulation with the added requirement that at least one of the SACs of the abstraction is an actual recurrent behavior of the concrete system, after some finite transient. Clearly, SACE simulation is stronger than simulation but significantly weaker than bisimulation. Equivalently, a *largest-average-cycle-equivalent simulation*, or LACE simulation, can be defined using the maximum average cycle instead. The following result is a straightforward conclusion from Proposition 5.1 and Theorem 2.2.

Proposition 5.2. *Consider two simple WTSs $\mathcal{S}_a$ and $\mathcal{S}_b$; if $\mathcal{S}_b$ is a finite-state SACE simulation of $\mathcal{S}_a$, then* $\mathrm{ILA}(\mathcal{S}_a) = \mathrm{ILA}(\mathcal{S}_b)$.

[3] Recall from (2.1) that $\mathrm{ILA}(\mathcal{V}) \coloneqq \inf\{\liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} y_i \mid \{y_i\} \in \mathcal{V}(\mathcal{S})\}$ is the infimal limit average of $\mathcal{S}$.

Proof. From Def. 5.1, take $dc^\omega \in \mathcal{B}^\omega(\mathcal{S}_a)$ for some finite-length sequence d with $c = k_1 ... k_N \in \mathrm{SAC}(\mathcal{S}_b)$. The associated LimAvg is

$$v := \mathrm{LimAvg}(dc^\omega) = \mathrm{LimAvg}(c^\omega) = \frac{1}{N}\sum_1^N k_i.$$

From Theorem 2.2, $v = \mathrm{ILA}(\mathcal{S}_b)$. As v is also the value of a behavior from $\mathcal{S}_a$, it holds that $v \geq \mathrm{ILA}(\mathcal{S}_a)$. Since Prop. 5.1 gives that $\mathrm{ILA}(\mathcal{S}_a) \geq \mathrm{ILA}(\mathcal{S}_b)$, we have that

$$\mathrm{ILA}(\mathcal{S}_b) = v \geq \mathrm{ILA}(\mathcal{S}_a) \geq \mathrm{ILA}(\mathcal{S}_b),$$

and thus $\mathrm{ILA}(\mathcal{S}_b) = \mathrm{ILA}(\mathcal{S}_a)$. □

Again, an equivalent result can be used for the computation of $\mathrm{SLA}(\mathcal{S}_a)$.

Remark 5.2. *In fact, to use Def. 5.1 and Prop. 5.2, it is not needed that the WTSs are simple. One can always turn a WTS into an equivalent simple one by adding artificial states: suppose that (x, y) and (x, z) belong to $\mathcal{E}$ and $\gamma(x, y) = a \neq \gamma(x, z) = b$. Add artificial states y' and z' and replace the aforementioned transitions with $(x, y'), (x, z'), (y', y), (z', z)$, setting $\gamma(x, y') = \gamma(x, z') = 0$, $\gamma(y', y) = a$ and $\gamma(z', z) = b$. Applying this procedure to the whole system gives a simple WTS, and again behaviors are equal to sequences of weights. The LimAvg value of any run of this modified system is half of the value of the original equivalent run (since we are adding zeros at every other transition).*

For the cases where obtaining a SACE simulation of $\mathrm{ILA}(\mathcal{S}_a)$ is not possible, one may still be interested in computing an estimate of the error $\mathrm{ILA}(\mathcal{S}_a) - \mathrm{ILA}(\mathcal{S}_b)$. A trivial estimate would be $\mathrm{SLA}(\mathcal{S}_b) - \mathrm{SLA}(\mathcal{S}_b)$, but a better approximation can be found by inspecting the maximal average cycle of the attractors of $\mathcal{S}_b$.

Proposition 5.3. *Let $\mathcal{S}_a := (\mathcal{X}_a, \mathcal{X}_a, \mathcal{E}_a, \mathcal{Y}, H_a)$ and $\mathcal{S}_b := (\mathcal{X}_b, \mathcal{X}_b, \mathcal{E}_b, \mathcal{Y}, H_b)$ be two simple autonomous WTSs, $\mathcal{R}$ be a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$, and $\mathcal{A} \subset \mathcal{X}_b$ be a strongly forward invariant set [4] of $\mathcal{S}_b$. If there exists $x_b \in \mathcal{A}$ such that $(x_a, x_b) \in \mathcal{R}$ for some $x_a \in \mathcal{X}_a$, then $\mathrm{ILA}(\mathcal{S}_a) \leq \mathrm{SLA}((\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b)) \leq \mathrm{SLA}(\mathcal{S}_b)$.*

Proof. First, it is a simple exercise to see that $(\mathcal{X}, \mathcal{X}', \mathcal{E}, \mathcal{Y}, H) \preceq (\mathcal{X}, \mathcal{X}, \mathcal{E}, \mathcal{Y}, H)$ if $\mathcal{X}' \subseteq \mathcal{X}$. Now, take $(x_a, x_b) \in \mathcal{R}$ where $x_b \in \mathcal{A}$. Then, $(\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a) \preceq \mathcal{S}_a$. At the same time, with the same relation $\mathcal{R}$ we can verify that $(\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a) \preceq (\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)$. Therefore, by Prop. 5.1, $\mathrm{ILA}((\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a)) \geq \mathrm{ILA}(\mathcal{S}_a)$, and $\mathrm{SLA}((\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)) \geq \mathrm{SLA}((\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a))$. Because $\mathrm{SLA}(\cdot) \geq \mathrm{ILA}(\cdot)$, we get that $\mathrm{SLA}((\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)) \geq \mathrm{ILA}(\mathcal{S}_a)$.

Now, because $\mathcal{A}$ is strongly forward invariant, every run of $(\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)$ contains only states in $\mathcal{A}$. Thus, $(\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b) \cong_{\mathcal{B}} (\mathcal{A}, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b) \preceq (\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b)$. Then, applying Prop. 5.1 again gives $\mathrm{SLA}((\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b)) \geq \mathrm{ILA}(\mathcal{S}_a)$. This proves the left-hand-side inequality.

The right-hand-side inequality is trivial: because $(\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b) \preceq \mathcal{S}_b$ (as we only remove states), Proposition 5.1 also gives that $\mathrm{SLA}(\mathcal{S}_b) \geq \mathrm{SLA}((\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b))$. □

[4] A strongly forward invariant set $\mathcal{A} \subseteq \mathcal{X}$ is a set that satisfies $\forall x \in \mathcal{A}, (x, x') \in \mathcal{E} \implies x' \in \mathcal{A}$.

When the abstraction $\mathcal{S}_b$ is finite, its smallest strongly invariant sets are simply the attractive strongly connected components (SCCs) of the graph associated with $\mathcal{S}_b$. Obtaining the SCCs of a graph with n vertices and m edges has complexity $\mathcal{O}(n+m)$ [41] and in fact is part of the steps to compute its smallest (or largest) average cycle.

5.4 Limit Average from l-complete abstractions

In this section we provide some results on the computation of the infimal limit average of a simple WTS $\mathcal{S}$ through the use of its SlCA $\mathcal{S}_l$. The first result is an obvious conclusion from combining Prop. 2.1 with 5.1:

Proposition 5.4. *Consider a simple WTS $\mathcal{S}$ and its SlCA $\mathcal{S}_l$ (Def. 2.6), for some $l \geq 1$. It holds that* $\mathrm{ILA}(\mathcal{S}_l) \leq \mathrm{ILA}(\mathcal{S})$.

Considering the idea of SACE simulation, a simple conceptual algorithm that can compute the exact value of $\mathrm{ILA}(\mathcal{S})$ is given in Alg. 1. The idea is to increment l until the smallest-in-average cycle of $\mathcal{S}_l$ is verified in the concrete system. The algorithm requires one to be able to compute the SlCA of a given system (line 3) and to verify the existence of periodic behavior (line 5); these steps will be discussed for PETC traffic on Section 5.5. As we will see now, Alg. 1 is in fact a semi-algorithm; depending on the behavior of $\mathcal{S}$, it may not terminate. The following result shows under which conditions there is a finite l such that $\mathrm{ILA}(\mathcal{S}_l) = \mathrm{ILA}(\mathcal{S})$.

Theorem 5.1. *Consider a simple finite WTS $\mathcal{S}$ and assume that there exists a finite $m \in \mathbb{N}$ such that every infinite behavior $\alpha \in \mathcal{B}^\omega(\mathcal{S})$ satisfies* $\mathrm{Avg}(\beta) \geq \mathrm{ILA}(\mathcal{S})$*, for every non-transient subsequence β of α with $|\beta| = m$. Then there exists a finite l such that the l-complete simulation $\mathcal{S}_l$ of $\mathcal{S}$ satisfies* $\mathrm{ILA}(\mathcal{S}_l) = \mathrm{ILA}(\mathcal{S})$.

Proof. First we prove that, if β is transient, then there exists l large enough such that β cannot be a subsequence of σ^ω for any cycle σ of $\mathcal{S}_l$. For that, suppose by contradiction that, $\forall L, \exists l \geq L$ for which a cycle σ of $\mathcal{S}_l$ exists s.t. β *is* a subsequence of σ^ω; w.l.o.g., assume that $l > m$. Then, there exists a word $\gamma\beta$ of length l that is a subsequence of σ^ω; hence, $|\gamma| = l - m$. This holds because for any natural number p, $\sigma^p\beta$ is a subsequence of $\sigma^p\sigma^\omega = \sigma^\omega$. Now, by definition of $\mathcal{S}_l$, $\gamma\beta \in \mathcal{B}^l(\mathcal{S})$. Since l can be chosen arbitrarily large, β can occur arbitrarily late in a behavior of $\mathcal{S}$, thus contradicting the fact that it is transient.

Therefore, there exists l large enough such that, for every cycle σ of $\mathcal{S}_l$, every m-long subsequence β of σ^ω is non-transient. From Theorem 2.2, one such cycle satisfies $\mathrm{ILA}(\mathcal{S}_l) = \mathrm{LimAvg}(\sigma^\omega)$. Let $p \coloneqq |\sigma|$. Then, σ^m has length pm and as such it can be divided in p non-transient subsequences β_i, not necessarily distinct, of length m. Now,

$$\mathrm{ILA}(\mathcal{S}_l) = \mathrm{LimAvg}(\sigma^\omega) = \mathrm{LimAvg}((\sigma^m)^\omega)) = \mathrm{Avg}(\sigma^m) = \frac{1}{p}\sum_{i=1}^{p}\mathrm{Avg}(\beta_i) \geq \mathrm{ILA}(\mathcal{S}).$$

Since, by Prop. 5.4, $\mathrm{ILA}(\mathcal{S}_l) \leq \mathrm{ILA}(\mathcal{S})$, it holds that $\mathrm{ILA}(\mathcal{S}_l) = \mathrm{ILA}(\mathcal{S})$. □

Theorem 5.1 states that it is sufficient for it to exist an m large enough such that every "persistent" m-long behavior fragment β of $\mathcal{S}$ has higher or equal average than $\mathrm{ILA}(\mathcal{S})$. Intuitively, constraining the assumption of β occurring infinitely often has the idea of

Algorithm 1 Computation of ILA($\mathcal{S}$)

Input: A simple WTS $\mathcal{S}$ with $\mathcal{Y} \subset \mathbb{Q}, |\mathcal{Y}| < \infty$
Output: $l, \mathcal{S}_l, \mathtt{V}, \sigma$

```
1: l ← 1
2: while true do
3:     Build S_l                                   ▷ (Def. 2.6)
4:     V ← ILA(S_l), σ ← SAC(S_l)                  ▷ [41, 42]
5:     if σ^ω ∈ B^ω(S) then
6:         return
7:     end if
8:     l ← l + 1
9: end while
```

excluding transient behaviors β, which do not affect the LimAvg value. For cases where β can occur infinitely often in some behavior, but β^ω is not a behavior of $\mathcal{S}$, one can construct counterexamples in which $\mathrm{ILA}(\mathcal{S}_l) < \mathrm{ILA}(\mathcal{S})$ for all l:

Example 5.1. *Consider a system $\mathcal{S}$ with behavior set $\mathcal{B}^\omega(\mathcal{S}) = \{(1^n 2^n)^\omega \mid n \in \mathbb{N}\}$. Obviously,* $\mathrm{ILA}(\mathcal{S}) = 1.5$. *However, for any l, $(1^l)^\omega \in \mathcal{B}^\omega(\mathcal{S}_l)$, hence* $\mathrm{ILA}(\mathcal{S}_l) = 1$ *for any l.*

Example 5.2. *Consider the system $\mathcal{S} = ([0,1],[0,1],\mathcal{E},\{0,1\},H)$ where $\mathcal{E} = \{(x, x+a \bmod 1)\}$ and $H(x) = 1$ if $x < a$ and 0 otherwise. When a is irrational, $\mathcal{S}$ is called an irrational rotation. Because it is ergodic with respect to the Lebesgue measure [74],* $\mathrm{LimAvg}(\alpha) = a$ *for any $\alpha \in \mathcal{B}^\omega(\mathcal{S})$. Thus,* $\mathrm{ILA}(\mathcal{S}) = a$ *is irrational. Since for every finite l,* $\mathrm{ILA}(\mathcal{S}_l)$ *is a rational number (as a consequence of Theorem 2.2 and the fact that $\mathcal{S}_l$ is finite),* $\mathrm{ILA}(\mathcal{S}_l) \neq \mathrm{ILA}(\mathcal{S})$. *Finally, from Prop. 5.4,* $\mathrm{ILA}(\mathcal{S}_l) \leq \mathrm{ILA}(\mathcal{S})$, *thus* $\mathrm{ILA}(\mathcal{S}_l) < \mathrm{ILA}(\mathcal{S})$ *for all finite l.*

Note that, for Ex. 5.2, the minimum number of 1s in a behavior fragment of length n is $\lfloor na \rfloor$, hence $\mathrm{ILA}(\mathcal{S}_l) = \frac{\lfloor la \rfloor}{l}$, which asymptotically approaches a as l goes to infinity. For Ex. 5.1, we cannot obtain this asymptotic approximation.

The conditions in Theorem 5.1 do not imply that the SAC σ of $\mathcal{S}_l$ satisfies $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$; thus, we may have equality of LimAvg values without a SACE simulation. Therefore, under these conditions, Alg. 1 can be interrupted with the exact value, but with no certificate that this is the case. Its termination is guaranteed when there is a cyclic minimizing behavior, and additionally that the other behaviors have limit average values strictly larger than that of the cycle:

Theorem 5.2. *Consider a simple WTS $\mathcal{S}$, and suppose $\mathcal{S}$ satisfies the premises of Theorem 5.1. Furthermore, assume there exists an m-long sequence σ such that $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$, and that every non-transient subsequence β, $|\beta| = m$ of every behavior $\alpha \in \mathcal{B}^\omega(\mathcal{S})$ satisfies* $\mathrm{LimAvg}(\beta^\omega) > \mathrm{ILA}(\mathcal{S})$ *if β is not a subsequence of σ^ω. Then Alg. 1 terminates with* $\mathtt{V} = \mathrm{ILA}(\mathcal{S})$.

The proof requires some technical results on cyclic permutations of sequences and we leave it for the appendix. The main insight is that the conditions of Theorem 5.2 imply that, for sufficiently large l, $\mathcal{S}_l$ has only one SAC σ, modulo cyclic permutations, which

attains the minimum value; at the same time, for large enough l, this σ satisfies $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$. Hereafter, we say that a system satisfying the premises of Theorem 5.2 has an *isolated SAC*. This does not mean that the behavior of $\mathcal{S}$ is simple, or that a finite-state bisimulation of it exists:

Example 5.3. *Consider the* doubling map *system* $\mathcal{S} = ([0,1],[0,1],\mathcal{E},\{0,1\},H)$ *where* $\mathcal{E} = \{(x, 2x \bmod 1) \mid x \in [0,1]\}$ *and* $H(x) = 0$ *if* $x < 1/2$ *and 1 otherwise. The behavior of this system is* $(0^+1^+)^\omega$*, its smallest cycle is* 0^ω *with value zero (obtained with* $x_0 = 0$*). This system does not admit a finite-state bisimulation, but its 1-complete abstraction is* $\mathcal{S}_1 = \{\{0,1\},\{0,1\},\{(0,0),(0,1),(1,0),(1,1)\},\{0,1\},\mathrm{Id}\}$*, where* Id *is the identity operator. Clearly,* $\mathcal{S}_1$ *is a SACE simulation of* $\mathcal{S}$ *(in fact, it is behaviorally equivalent, but not bisimilar). The system* $\mathcal{S}$ *satisfies the premises of Theorem 5.2 with* $m = 1$.

Now that we have the general framework for the computation of ILA($\mathcal{S}$), we see how to apply it for PETC traffic.

5.5 Computing the SAIST of PETC

Consider now the PETC traffic model $\mathcal{S}$ from Def. 2.15 as the concrete infinite-state system for which we want develop an algorithm like Alg. 1. For this we need to be able to (i) build an l-complete abstraction of the system, (ii) compute its SAC, and (iii) check if its minimum mean cycle exists in the concrete system. Naturally, Karp's algorithm [41, 42] constitute the tool for task (ii). In the next section we present how to obtain l-complete abstractions of $\mathcal{S}$. Then, in Section 5.5.2, we show how can a cyclic behavior be verified to be trace of $\mathcal{S}$. Finally, we present the full algorithm and discuss its robustness and applicability in subsequent subsections.

5.5.1 l-complete PETC traffic models

In this section, we fix $\mathcal{S}$ as a PETC traffic model (2.15). Obtaining its strongest l-complete abstraction is very similar to the models proposed in Chapter 4; however, instead of partitioning the state-space $\mathcal{X}$ into regions associated to different sequences that lead to some given reduction in a chosen Lyapunov function, we divide $\mathcal{X}$ into regions $\mathcal{X}_{y_1 y_2 \dots y_l}$, where the first l elements of any behavior in $\mathcal{B}^\omega_x(\mathcal{S})$, for any $x \in \mathcal{X}_{y_1 y_2 \dots y_l}$, are exactly $y_1, y_2, \dots, y_l$. If $\mathcal{S}$ is deterministic, this division generates a partition, as from one state x there exists only one infinite behavior. The relation is thus a simplified version of Def. 4.3:

Definition 5.2 (PETC inter-sample sequence relation)**.** *Given a sequence length* l*, we denote by* $\mathcal{R}_l \subseteq \mathbb{R}^{n_x} \times \mathcal{Y}^l$ *the relation satisfying* $(\boldsymbol{x}, k_1 k_2 \dots k_l) \in \mathcal{R}_l$ *if and only if*

$$\boldsymbol{x} \in \mathcal{Q}_{k_1}, \tag{5.2a}$$
$$\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_2}, \tag{5.2b}$$
$$\boldsymbol{M}(k_2)\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_3}, \tag{5.2c}$$
$$\vdots$$
$$\boldsymbol{M}(k_{l-1})\dots\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_l}, \tag{5.2d}$$

where the sets $\mathcal{Q}_k$ *are defined in* (3.2).

By construction, a state $\boldsymbol{x} \in \mathbb{R}^n$ of the concrete PETC traffic model is related to a state $k_1 k_2 ... k_l$ of the abstraction if its generated inter-sample time sequence for the next l samples is $k_1, k_2, ..., k_l$.

Remark 5.3. *Setting $l = 1$ gives a normal quotient state-set, similar to the one presented in Chapter 3.*

Definition 5.3. *Given an integer $l \geq 1$, the* l-complete PETC traffic model *is the system $S_l := (\mathcal{X}_l, \mathcal{X}_l, \mathcal{E}_l, \mathcal{Y}, H_l)$, with*

- $\mathcal{X}_l := \pi_{\mathcal{R}_l}(\mathcal{X})$,
- $\mathcal{E}_l = \{(k\sigma, \sigma k') \mid k, k' \in \mathcal{Y}, \sigma \in \mathcal{Y}^{l-1}, k\sigma, \sigma k' \in \mathcal{X}_l\}$,
- $H_l(k_1 k_2 ... k_l) = k_1$.

The model above partitions the state-space $\mathbb{R}^{n_x}$ of the PETC into subsets associated with the next l inter-sample times these states generate, i.e., it is an l-complete abstraction, but also a quotient-based model. Computing the state set, $\pi_{\mathcal{R}_l}(\mathcal{X})$, requires determining whether or not, for each $k_1 k_2 ... k_l \in \mathcal{Y}^l$, its associated conjunction of quadratic inequalities in Eq. (5.2) admits a solution $\boldsymbol{x} \in \mathbb{R}^{n_x}$; only if it does, then $\sigma \in \mathcal{X}_l$. This can be determined using a nonlinear satisfiability-modulo-theories (SMT) solver such as Z3 [64]: for that, the variable is $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and the query is $\exists \boldsymbol{x} \in \mathbb{R}^{n_x}$: Eq. (5.2) holds.[5] The output map H_l is the next sample alone, and the transition relation is based on the domino rule, as in Def. 2.6.

5.5.2 Verifying SACE equivalence

In this subsection, we are interested in determining whether a sequence of outputs $(k_1 k_2 ... k_m)^\omega =: \sigma^\omega$ is a possible behavior of $\boldsymbol{S}$. This is equivalent to finding a run $\{\boldsymbol{x}_i\}$ whose trace is σ^ω. From now on, we denote by $\mathcal{Q}_\sigma$, or σ-*cone*, the set of all points $\boldsymbol{x} \in \mathbb{R}^{n_x}$ satisfying Eq. (5.2) with $l = m$ and by $\boldsymbol{M}_\sigma := \boldsymbol{M}(k_m)\boldsymbol{M}(k_{m-1}) \cdots \boldsymbol{M}(k_1)$. For the formal results, consider the following classes of square matrices:

Definition 5.4 (Mixed matrix)**.** *Consider a matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ and let $\lambda_i, i \in \mathbb{N}_{\leq n}$ be its eigenvalues sorted such that $|\lambda_i| \geq |\lambda_{i+1}|$ for all i. We say that $\boldsymbol{M}$ is* mixed *if, for all $i < n$, $|\lambda_i| = |\lambda_{i+1}|$ implies that $\Im(\lambda_i) \neq 0$ and $\lambda_i = \lambda_{i+1}^*$.*

Remark 5.4. *Mixed matrices cannot have eigenvalues with the same magnitude, except for complex conjugate pairs. Every mixed matrix is diagonalizable, but the converse does not hold (e.g., the identity is not mixed). The set of mixed matrices is full Lebesgue measure. With a non-pathological choice of h,[6] the matrices $\boldsymbol{M}(1), \boldsymbol{M}(2), ...\boldsymbol{M}(\bar{k})$ from Eq. (2.10) are all mixed, even if $\boldsymbol{K}$ is chosen to place poles of $\boldsymbol{A} + \boldsymbol{BK}$ in the same point of the complex plane; it is sensible (but not guaranteed) to expect that their products are also mixed. From a linear systems perspective, all modes of a mixed matrix have different speeds.*

Definition 5.5 (Matrix of irrational rotations)**.** *A matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is said to be of* irrational rotations *if the arguments of all of its complex eigenvalues are irrational multiples of π.*

[5] Alternatively, this query may be solved approximately through convex relaxations as proposed in Chapter 3. Using relaxations implies finding inter-sample sequences that may not be exhibited by the real system. This still generates a simulation relation, but containing more spurious behaviors.

[6] Typically, only countably many values of h will render $\boldsymbol{M}(k)$ non-mixed for a given k.

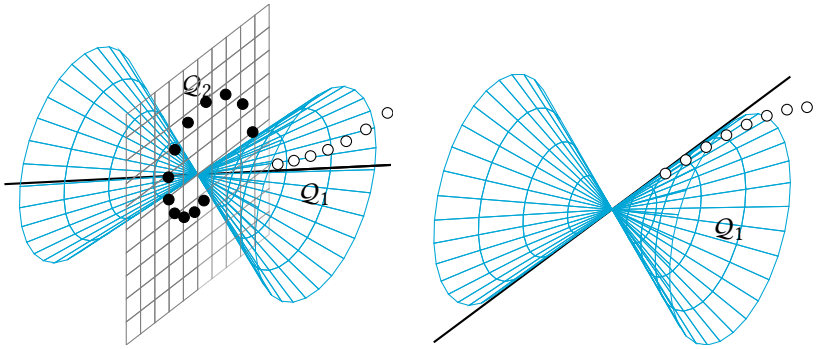


Figure 5.1: Illustration of Theorem 5.3 in $\mathbb{R}^3$. The blue cone splits $\mathbb{R}^3$ into $\mathcal{Q}_1$ and $\mathcal{Q}_2$ the line is an invariant of $\boldsymbol{M}(1)$ and the plane is an invariant of $\boldsymbol{M}(2)$. Points indicate distinct sample trajectories $\{\boldsymbol{x}_i\}$.

5

Remark 5.5. *If $\boldsymbol{M}$ has a pair of complex conjugate eigenvalues whose argument is a rational multiple of π, i.e., $p\pi/q$, where $p, q \in \mathbb{N}$, then the corresponding eigenvalues of $\boldsymbol{M}^q$ are real. The set of real matrices of* rational *rotations is Lebesgue-measure zero but dense in $\mathbb{R}^{n\times n}$.*

If $\boldsymbol{M}_\sigma$ is mixed and of irrational rotations, one can verify if σ^ω is a behavior of $\mathcal{S}$ by checking the linear invariants of $\boldsymbol{M}_\sigma$:

Theorem 5.3. *Consider the PETC system $\mathcal{S}$ (Def. 2.15) and let $\sigma \in \mathcal{Y}^m$, $m \in \mathbb{N}$, be a sequence of outputs. (i) If $\boldsymbol{M}_\sigma$ is nonsingular and there exists a linear invariant $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ such that $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_\sigma$, then $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$. Moreover, if (ii) $\boldsymbol{M}_\sigma$ is additionally mixed and of irrational rotations, then $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$ implies that there exists a linear invariant $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ such that $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$.*

To avoid a long detour in our exposition, we leave the proof to the appendix, instead providing here a depiction of the idea behind it: In Fig. 5.1, we have $m = 1$ and $\mathcal{Y} = \{1, 2\}$, and the blue cone splits $\mathbb{R}^3$, the state space, in $\mathcal{Q}_1$ and $\mathcal{Q}_2$; the two plots have different matrices $\boldsymbol{M}(1)$. Runs $\{\boldsymbol{x}_i\}$ that generate the trace 1^ω are solutions of the linear system $\boldsymbol{x}_{i+1} = \boldsymbol{M}(1)\boldsymbol{x}_i$, one such example being depicted with white dots. Likewise, black dots show a run generating the trace 2^ω, and it has to be a solution of $\boldsymbol{x}_{i+1} = \boldsymbol{M}(2)\boldsymbol{x}_i$. In the example on the left, the black line is supported by one real eigenvector of $\boldsymbol{M}(1)$ and, as it belongs to $\mathcal{Q}_1$, at least solutions on top of this eigendirection are runs of the PETC system $\mathcal{S}$. In our example, this eigenvector is associated with a dominant mode of $\boldsymbol{M}(1)$, so solutions starting close to it converge towards it. The plane depicted on the left of Fig. 5.1 is an invariant of $\boldsymbol{M}(2)$ associated to complex conjugate eigenvalues. Solutions starting in this plane stay in this plane, spiraling towards the origin (in case the PETC implementation is stabilizing), which confirms that 2^ω is also a behavior of $\mathcal{S}$. The example on the right shows the defective case where the converse does not hold: for that, assume that $\mathcal{Q}_1$ does not include its depicted blue boundary; however, the black line representing an eigendirection of $\boldsymbol{M}(1)$ runs precisely on this boundary. In this example, the white dots represent a run $\{\boldsymbol{x}_i\}$ in $\mathcal{Q}_1$, thus generating the trace 1^ω, but no invariant of $\boldsymbol{M}(1)$ is a subset of $\mathcal{Q}_1$. Because the depicted mode of $\boldsymbol{M}(1)$ is dominant, there are solutions that start close to its associated eigendirection that stay in $\mathcal{Q}_1$ forever.

Based on Theorem 5.3, in the non-defective cases we can verify a cyclic behavior σ^ω by taking the finitely many linear invariants $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ and checking if $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_\sigma$, or, more explicitly, taking $\sigma = k_1 k_2 ... k_m$,

$$\begin{aligned} \mathcal{A} \setminus \{\boldsymbol{0}\} &\subseteq \mathcal{Q}_{k_1}, \\ \boldsymbol{M}(k_1)\mathcal{A} \setminus \{\boldsymbol{0}\} &\subseteq \mathcal{Q}_{k_2}, \\ &\vdots \\ \boldsymbol{M}(k_{m-1})...\boldsymbol{M}(k_1)\mathcal{A} \setminus \{\boldsymbol{0}\} &\subseteq \mathcal{Q}_{k_m}. \end{aligned} \tag{5.3}$$

Because each $\mathcal{Q}_k$ is an intersection of quadratic sets (see Eq. (3.2)), we must be able to check whether a linear space is a subset of a given quadratic set, which is nothing but a positive-(semi)definiteness check:

Proposition 5.5. *Let $\mathcal{A}$ be a linear subspace with basis $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_m$, and let $\boldsymbol{V}$ be the matrix composed of the vectors $\boldsymbol{v}_i$ as columns. Let $\boldsymbol{Q} \in \mathbb{S}^n$ be a symmetric matrix and define $\mathcal{Q}_n \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} \geq 0\}$ and $\mathcal{Q}_s \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} > 0\}$. Then, $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_n$ (resp. $\mathcal{Q}_s$) if and only if $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}\boldsymbol{V} \succeq \boldsymbol{0}$ (resp. $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}\boldsymbol{V} \succ \boldsymbol{0}$).*

Proof. For brevity, let us consider the strict inequality case (the other is analogous). First, note that $\mathcal{A} = \{\boldsymbol{V}\boldsymbol{a} \mid \boldsymbol{a} \in \mathbb{R}^m\}$. Hence, if we want all points in $\mathcal{A}$ to belong to $\mathcal{Q}_s$, we need that

$$\forall \boldsymbol{a} \in \mathbb{R}^m \setminus \{\boldsymbol{0}\},\ \boldsymbol{a}^\mathsf{T}\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}\boldsymbol{V}\boldsymbol{a} > 0,$$

which is exactly the definition of $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}\boldsymbol{V} \succ \boldsymbol{0}$. □

5.5.3 SACE simulation algorithm

Combining the l-complete traffic models from Section 5.5.1 with the stopping criterion based on checking linear invariants from Section 5.5.2, we specialize Algorithm 1 into Algorithm 2 to generate a finite-state SACE simulation of the PETC traffic model $\mathcal{S}$, together with the computation of its SAIST ILA($\mathcal{S}$). In the outer loop, the relation $\mathcal{R}_l$ and corresponding finite-state system $\mathcal{S}_l$ are built, followed by the computation of one of its SACs σ. Then, an inner loop looks for linear subspaces $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ satisfying $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_\sigma$ (Theorem 5.3); because $\boldsymbol{M}_\sigma$ is assumed to be mixed and of irrational rotations[7], it suffices to verify 1-dimensional subspaces for real eigenvectors and 2-dimensional subspaces for complex conjugate ones[8]; if one is found, the algorithm terminates. Otherwise, l is incremented and the main loop is repeated. Hereafter, we say that a linear invariant subspace of a mixed matrix is *basic* if it is the span of a real eigenvector or of a pair of complex conjugate eigenvectors.

In order to state formal results about the correctness of Algorithm 2, we need to account for the conditions in Theorem 5.3.

Definition 5.6 (Normalized distance)**.** *The* normalized distance *between a point $\boldsymbol{x} \in \mathbb{R}^n$ and a set $\mathcal{A} \subseteq \mathbb{R}^n$, denoted by $d_{\mathrm{n}}(\boldsymbol{x}, \mathcal{A})$ is defined as $\inf_{\boldsymbol{l} \in \mathcal{A}} \left(1 - \frac{\boldsymbol{l}^\mathsf{T}\boldsymbol{x}}{|\boldsymbol{l}||\boldsymbol{x}|}\right)$. The normalized distance between two sets is $d_{\mathrm{n}}(\mathcal{A}, \mathcal{A}') \coloneqq \inf_{\boldsymbol{l} \in \mathcal{A}} d_{\mathrm{n}}(\boldsymbol{l}, \mathcal{A}')$.*

[7] Any matrix is arbitrarily close to a mixed matrix of irrational rotations; numerically checking if it is otherwise is not robust. A more thorough discussion about this is available in Section 5.5.4

[8] If a larger dimensional subspace $\mathcal{A}'$ is a subset of $\mathcal{Q}_\sigma$, any smaller dimensional subspace $\mathcal{A} \subset \mathcal{A}'$ will also be. Thus, there is no benefit in verifying subspaces that are combinations of smaller real linear subspaces

Algorithm 2 PETC SAIST computation algorithm

Input: $\mathcal{Y}$ and $\boldsymbol{M}(k), \mathcal{Q}_k, \forall k \in \mathcal{Y}$
Output: $l, \mathcal{S}_l, \sigma, \texttt{SAIST}$

$l \leftarrow 1$
while true **do**
  Build $\mathcal{R}_l$ and $\mathcal{S}_l$ ▷ (Defs. 5.2 and 5.3)
  $\texttt{SAIST} \leftarrow \text{ILA}(\mathcal{S}_l)$, $\sigma \leftarrow \text{SAC}(\mathcal{S}_l)$ ▷ [41, 42]
  for all $\mathcal{A} \in \text{BILS}(\boldsymbol{M}_\sigma)$ **do** ▷ BILS = basic invariant linear subspaces
    if $\mathcal{A}$ satisfies Eq. (5.3) with $k_1, k_2, ..., k_m = \sigma$ **then**
      return
    end if
  end for
  $l \leftarrow l + 1$
end while

As the quantity $\frac{\boldsymbol{l}^\mathsf{T}\boldsymbol{x}}{|\boldsymbol{l}||\boldsymbol{x}|}$ is the cosine of the angle between the vectors $\boldsymbol{l}$ and $\boldsymbol{x}$, the normalized distance varies between 0 and 1, measuring how close $\boldsymbol{x}$ is, modulo magnitude, to the set $\mathcal{A}$. It is a more sensible choice of distance when dealing with homogeneous sets than the Euclidean distance, which would be zero as the origin is always in or arbitrarily close to such sets. This distance is needed for some technical results that come later, as well as for the following definition.

Definition 5.7 (Regularity)**.** *A sequence of ISTs $\sigma := k_1 k_2 ... k_m$ is said to be* regular *if (i) $\boldsymbol{M}_\sigma$ is nonsingular, mixed, and of irrational rotations, and (ii) for every invariant linear subspace $\mathcal{A}$ of $\boldsymbol{M}_\sigma$, we have that $d_\text{n}(\mathcal{A}, \partial\mathcal{Q}_\sigma) \geq \epsilon$ for some $\epsilon > 0$.*

Regularity of a sequence σ prevents that one of the invariants of $\boldsymbol{M}_\sigma$ intersect $\partial\mathcal{Q}_\sigma$ (the case in the right of Fig. 5.1), requiring a minimal ϵ clearance to its boundary. The following result establishes conditions for the termination of Alg. 2; the proof is in the Appendix.

Theorem 5.4. *Suppose that $\mathcal{S}$ (Def. 2.15) has an isolated smallest-in-average cycle σ that is regular. Then, Alg. 2 terminates with* $\texttt{SAIST} = \text{ILA}(\mathcal{S})$.

The conditions of Theorem 5.4 are the same behavioral conditions as in Theorem 5.2: the system must exhibit a minimizing periodic behavior, and competing infinite behaviors must be composed of subsequences that have average value strictly larger than the minimal value. Additionally, the smallest cycle must be regular, which is not a limiting assumption. Therefore, the algorithm may not terminate when, for example, a minimizing behavior is aperiodic. In this case, we may still expect increasingly better estimates of $\text{ILA}(\mathcal{S})$ with larger values of l.

5.5.4 Robustness and computability

Algorithm 2 relies on the matrices $\boldsymbol{M}(k)$ from Eq. (2.10), whose elements are typically transcendental. Therefore, one may wonder if the algorithm, or more generically a given

l-complete SACE traffic model, is robust to small round-off errors when computing those matrices, as well as other small model mismatches. In this section, we are going to see that this is true given that some mild assumptions are satisfied. For this, we need proper definitions.

Definition 5.8 (Perturbed PETC system)**.** *Given a PETC system* (2.2)–(2.4) *and its data* $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}, \boldsymbol{Q}, \bar{k}$*, the PETC system with data* $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{K}}, \tilde{\boldsymbol{Q}}, \bar{k}$ *is called a* δ*-perturbation of the former if* $\left|\boldsymbol{A} - \tilde{\boldsymbol{A}}\right| \leq \delta, \left|\boldsymbol{B}\boldsymbol{K} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right| \leq \delta$*, and* $\left|\boldsymbol{Q} - \tilde{\boldsymbol{Q}}\right| \leq \delta$*. Furthermore, the traffic model* $\tilde{\mathcal{S}}$ *cf. Def. 2.15 of a* δ*-perturbation of system* (2.2)–(2.4) *is denoted a* δ*-perturbation of* $\mathcal{S}$*.*

Remark 5.6. *Considering Footnote 2, Def. 5.8 also encompasses variations in the actual checking period* h*.*

Definition 5.9 (ϵ-inflation)**.** *The* ϵ-inflation *of a quadratic cone* $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{x} \geq (>)\, 0\}$ *is the set* $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathsf{T}}(\boldsymbol{Q} + \epsilon\mathbf{I})\boldsymbol{x} \geq (>)\, 0\}$*, for* $\epsilon > 0$*. An* ϵ*-inflation of the intersection of quadratic cones is defined as the intersection of the* ϵ*-inflations.*

Let $\mathcal{P}_\delta(\mathcal{S})$ be the set of all δ-perturbations of $\mathcal{S}$. We have the following results.

Proposition 5.6. *If* $\mathcal{S}_l$ *is an* l*-complete model (Def. 5.3) of* $\mathcal{S}$*, then there exists* $\delta > 0$ *such that* $\mathcal{S}_l$ *is an* l*-complete model of every* $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$ *if there exists an* $\epsilon > 0$ *such that the following conditions hold:*

- *For every* $\sigma \in \mathcal{B}^l(\mathcal{S})$*, there exists* $\boldsymbol{x} \in \mathcal{Q}_\sigma$ *s.t.* $d_{\mathrm{n}}(\boldsymbol{x}, \partial\mathcal{Q}_\sigma) > \epsilon$*; and*
- *for every* $\sigma \notin \mathcal{B}^l(\mathcal{S})$*, every* ϵ*-inflation of* $\mathcal{Q}_\sigma$ *is empty.*

Proof. By Definition 5.3, $\mathcal{S}_l$ is an SlCA of every $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$ if

1) $\sigma \in \mathcal{B}^l(\mathcal{S}) \implies \sigma \in \mathcal{B}^l(\tilde{\mathcal{S}}), \forall \tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$, and
2) $\sigma \notin \mathcal{B}^l(\mathcal{S}) \implies \sigma \in \mathcal{B}^l(\tilde{\mathcal{S}}), \forall \tilde{\mathcal{S}} \notin \mathcal{P}_\delta(\mathcal{S})$.

For item 1, we must have a non-zero vector $\boldsymbol{x} \in \tilde{\mathcal{Q}}_\sigma$, where $\tilde{\mathcal{Q}}_\sigma$ is the σ-cone of the δ-perturbation $\tilde{\mathcal{S}}$. Because $d_{\mathrm{n}}(\boldsymbol{x}, \partial\mathcal{Q}_\sigma) > \epsilon$, we have that the normalized distance to the complement of $\mathcal{Q}_\sigma$ satisfies $d_{\mathrm{n}}(\boldsymbol{x}, \bar{\mathcal{Q}}_\sigma) > \epsilon$. By continuity, this implies that $d_{\mathrm{n}}(\boldsymbol{x}, \tilde{\bar{\mathcal{Q}}}_\sigma) > 0$ for small enough δ, and hence $\boldsymbol{x} \in \tilde{\mathcal{Q}} \implies \sigma \in \mathcal{B}^l(\tilde{\mathcal{S}})$. Likewise, for item 2, we cannot have a vector $\boldsymbol{x} \in \bar{\mathcal{Q}}_\sigma$; by continuity, for small enough δ, $\tilde{\mathcal{Q}}_\sigma$ is a subset of the ϵ-inflation of $\mathcal{Q}_\sigma$, which is empty, and therefore $\sigma \notin \mathcal{B}^l(\tilde{\mathcal{S}})$. □

The conditions in Prop. 5.6 rule out marginal cases of degeneracy, and are expected to hold in general for sufficiently small ϵ. I.e., if σ is a behavior of $\mathcal{S}$ and $\mathcal{Q}_\sigma$ has a non-empty interior (equivalent to $n_{\mathrm{d}}(\boldsymbol{x}, \partial\mathcal{Q}_\sigma) > \epsilon$ for some $\boldsymbol{x}$ and ϵ), then sufficiently small perturbations to the sets whose intersection gives $\mathcal{Q}_\sigma$ do not render it empty; symmetrically, if σ is *not* a behavior of $\mathcal{S}$, not only $\mathcal{Q}_\sigma$ must be empty, but small perturbations on the sets whose intersection composes $\mathcal{Q}_\sigma$ must retain its emptiness, thus not creating a new behavior.

Proposition 5.7. *Let* σ^ω *be a cyclic behavior of* $\mathcal{S}$*. Then, if* σ *is regular, there exists some* $\delta > 0$ *such that* $\sigma^\omega \in \mathcal{B}^\omega(\tilde{\mathcal{S}})$*, for all* $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$*.*

Proof. From Theorem 5.3, we have that $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S}) \implies \mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$ for a basic linear invariant subspace $\mathcal{A}$ of $\boldsymbol{M}_\sigma$. From regularity of σ, $d_{\mathrm{n}}(\mathcal{A}, \partial\mathcal{Q}_\sigma) > \epsilon$. Together with $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$, we have that $d_{\mathrm{n}}(\mathcal{A}, \bar{\mathcal{Q}}_\sigma) > \epsilon$. Since σ is regular, $\boldsymbol{M}_\sigma$ is mixed by definition. Then, by

continuity of eigenvalues and eigenvectors, for small enough δ, the perturbed eigenvalues $\tilde{\lambda}_i$ are qualitatively unchanged: $\lambda_i \in \mathbb{R} \implies \tilde{\lambda}_i \in \mathbb{R}$, $\Im(\lambda_i) \neq 0 \implies \Im(\tilde{\lambda}_i) \neq 0$, and $|\lambda_i| > |\lambda_{i+1}| \implies |\tilde{\lambda}_i| > |\tilde{\lambda}_{i+1}|$. Thus, if $\mathcal{A}$ is a line associated to a real eigenvalue, so is the corresponding basic linear subspace $\tilde{\mathcal{A}}$ of $\tilde{\boldsymbol{M}}_\sigma$; and likewise if $\mathcal{A}$ is a plane corresponding to complex conjugate eigenvalues of irrational rotations: even if $\tilde{\boldsymbol{M}}_\sigma$ is not of irrational rotations, the plane $\tilde{\mathcal{A}}$ is one of its invariants. In addition, $d_{\mathrm{n}}(\mathcal{A}, \tilde{\mathcal{A}}) < d$, where d diminishes with δ. Hence, for small enough δ we have that $d_{\mathrm{n}}(\mathcal{A}, \bar{\mathcal{Q}}_\sigma) > \epsilon \implies d_{\mathrm{n}}(\tilde{\mathcal{A}}, \bar{\tilde{\mathcal{Q}}}_\sigma) > 0 \implies \tilde{\mathcal{A}} \setminus \{\mathbf{0}\} \subseteq \tilde{\mathcal{Q}}_\sigma$. Therefore, applying again Theorem 5.3, we conclude that $\sigma^\omega \in \mathcal{B}^\omega(\tilde{\mathcal{S}}), \forall \tilde{\mathcal{S}} \in \mathcal{P}_\epsilon(\mathcal{S})$. □

These two propositions combined give the following result:

Theorem 5.5. *Let $\mathcal{S}_l$ be the SACE simulation of a PETC traffic model $\mathcal{S}$. If its smallest-in-average cycle σ is regular and $\mathcal{S}_l$ satisfies the premises of Prop. 5.6, then there exists $\delta > 0$ such that $\mathcal{S}_l$ is SACE simulation of every $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$.*

Theorem 5.5 has two interesting implications. The first is that sufficiently small round-off errors on the matrices $\boldsymbol{M}(k)$ and $\boldsymbol{Q}$ of Eq. (2.10) do not affect the correct computation of ILA($\mathcal{S}$); hence, ILA($\mathcal{S}$) is computable for a class of linear systems, even though $\boldsymbol{M}(k)$ typically contains transcendental numbers. The second implication is that, informally, we can apply our method to nonlinear systems, as long as the closed loop ETC system is asymptotically stable and the involved functions are sufficiently smooth. Asymptotic stability implies that the state converges to a ball of any radius, no matter how small, in finite time; therefore, the sequence of sampling times up to this point do not affect the system's SAIST. Inside a sufficiently small ball, the nonlinear flow belongs to a convex combination of δ-perturbations of its linearization about the equilibrium. If the linearized system $\mathcal{S}$ satisfies the premises of Theorem 5.5, the SAIST of the nonlinear system is equal to ILA($\mathcal{S}$).

5.5.5 An improved algorithm

Verifying the existence of each l-long behavior to obtain $\mathcal{R}_l$ and $\mathcal{S}_l$ (line 3 of Alg. 2) has exponential complexity on the number of variables [63, 67]. To reduce the number of times these problems are solved, we propose a more efficient refinement approach than performing the full $(l+1)$-complete abstraction. At every iteration of Alg. 2, we only refine the states of the abstraction associated with the previous SAC. This procedure is explained in Algorithm 3, where $\mathcal{X}_\sigma$ is the set of states that compose the SAC.[9]

To illustrate this approach, see Fig. 2.1, where three steps of this refinement approach are executed in the example of Fig. 5.2: in depth 3, the SAC is already $(1,1,2)^\omega$, but it requires only 6 verification procedures: 1, 2, $(1,1)$, $(1,2)$, $(1,1,1)$ (disproved) and $(1,1,2)$; the 3-complete model would require up to $2+4+8 = 14$ verification procedures to obtain the same SAC. The disadvantage of this approach is that the obtained graph is more connected (as we have fewer states but more behaviors), and thus the computation of an upper bound using Prop. 5.3 often gives too distant values.

[9] In fact, the corrected Karp's algorithm in [42] returns sequence of states that generate the SAC, from which determining the SAC is trivial. Algorithm 3 needs the states, hence we use the function SAC* which returns both the states $\mathcal{X}_\sigma$ and the behavioral cycle σ itself.

Algorithm 3 Fast PETC SAIST computation algorithm

Input: $\mathcal{S}_1$ and $\boldsymbol{M}(k), \mathcal{Q}_k, \forall k \in \mathcal{Y}$
Output: $d, \mathcal{S}_d, \sigma, \texttt{SAIST}$

$d \leftarrow 1$ ▷ d is the depth of the algorithm
while true **do**
  $\texttt{SAIST} \leftarrow \text{ILA}(\mathcal{S}_l)$, $(\mathcal{X}_\sigma, \sigma) \leftarrow \text{SAC}^*(\mathcal{S}_l)$ ▷ [41, 42]
  for all $\mathcal{A} \in \text{BILS}(\boldsymbol{M}_\sigma)$ **do** ▷ BILS = basic invariant linear subspaces
    if $\mathcal{A}$ satisfies Eq. (5.3) with $k_1, k_2, ..., k_m =: \sigma$ **then**
      return
    end if
  end for
  $\mathcal{X}_{d+1} \leftarrow \mathcal{X}_d \setminus \mathcal{X}_\sigma$ ▷ Remove states to be refined
  for all $(\alpha, \beta) \in \mathcal{E}_d$ such that $\alpha \in \mathcal{X}_\sigma$ and $|\alpha| \leq |\beta|$ **do** ▷ If $|\alpha| > |\beta|$ no new candidate is generated via the domino rule
    $k \leftarrow \beta(|\alpha|)$ ▷ The $|\alpha|$-th element of β
    $\gamma \leftarrow \alpha k$ ▷ New candidate: sequence of length $|\alpha| + 1$ given the Domino rule
    if $\exists \boldsymbol{x} \in \mathbb{R}^{n_x}$ such that $k_1 k_2 ... k_l =: \gamma$ satisfies Eqs. (5.2)–(3.2) **then** ▷ Nonlinear SMT
      $\mathcal{X}_{d+1} \leftarrow \mathcal{X}_{d+1} \cup \{\gamma\}$
    end if
  end for
  $\mathcal{E}_{d+1} \leftarrow \{(k\sigma, \sigma k') \mid k, k' \in \mathcal{Y}, k\sigma, \sigma k' \in \mathcal{X}_{d+1}\}$ ▷ Domino rule
  $H_{d+1}(k\sigma) \leftarrow k$ for all $k\sigma \in \mathcal{X}_{d+1}$
  $\mathcal{S}_{d+1} \leftarrow (\mathcal{X}_{d+1}, \mathcal{X}_{d+1}, \mathcal{E}_{d+1}, \mathcal{Y}, H_{d+1})$
  $d \leftarrow d + 1$
end while

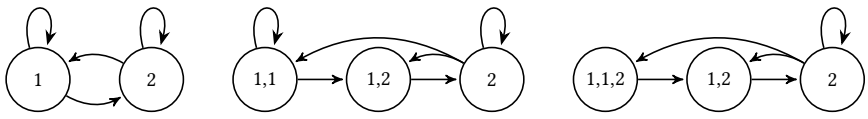


Figure 5.2: Illustration of Alg. 3 on the system $\mathcal{S}$ of Fig. 2.1.

5.6 Numerical Examples

In what follows we present three different numerical examples. They can be reproduced by using ETCetera, with an implementation of Algorithms 2 and 3, including an interface with Z3 to solve the nonlinear SMT problems associated with verifying a sequence σ. To reproduce the results of this section, the following scripts within ETCetera can be used:

- `examples/nahs_example1_traj.py` for Fig. 5.3;
- `examples/nahs_example1_traj2.py` for Fig. 5.4;
- `examples/nahs_example1_table.py` for Table 5.1 and other data presented in Section 5.6.1;
- `examples/nahs_example2_table.py` for Table 5.2;
- `examples/nahs_example3.py` for the SAIST bounds in Section 5.6.3 and Fig. 5.5.

5.6.1 A two-dimensional linear system

We start by considering the example from [70]: the system (2.2) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{K} = \begin{bmatrix} 0 & -5 \end{bmatrix},$$

and the triggering condition of [5], $|\boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t)| > a|\boldsymbol{\xi}(t)|$ for some $0 < a < 1$, which can be put in the form Eq. (2.4). Checking time was set to $h = 0.05$, and maximum inter-sample time to $\bar{k} = 20$. Using a Python implementation of Algorithm 2 with Z3 [64] to solve Eq. (5.2), we attempted to compute its SAIST through a SACE simulation for $a \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Table 5.1 presents the SAIST for each a, as well as the l value (Def. 2.6) where it was obtained. Only for $a = 0.1$ the algorithm did not terminate before $l = 50$: for this case, the actual $\bar{k}$ of the system was 3, and all $\boldsymbol{M}(k), k \leq 3$, have complex eigenvalues. Thus, it is possible that it does not have periodic behaviors, similarly to the irrational rotation of Example 5.2. Nonetheless, applying Prop. 5.3 gives an upper bound for $\mathrm{ILA}(\mathcal{S})$ of 1.596; hence, we know that $1.572 \leq \mathrm{ILA}(\mathcal{S}) \leq 1.596$, giving an uncertainty of only 0.024. For the other cases, trivial cycles were found for $a = 0.4$ (5^ω) and $a = 0.5$ (6^ω), but it took a few iterations to break, e.g., the 2^ω loop. Interestingly, the simplest cycles for $a = 0.2$ and $a = 0.3$ had length, respectively, 27 and 28, showing that PETC can often lead to very complex recurring patterns (see an example in Fig. 5.3). In addition, the case of $a = 0.4$ has two verified cyclic behaviors, 5^ω and 6^ω (see Fig. 5.4), while with $a = 0.5$ three cycles are obtained: $6^\omega, 7^\omega$ and 8^ω: this confirms that a single PETC system can exhibit multiple different periodic behaviors.

The results were generated on a MacBook Pro 2017 using a single processor. As Table 5.1 shows, even for $l = 50$ the CPU time was kept under 10 minutes.

Table 5.1: SAIST values for the example of Section 5.6.1

a	0.1	0.2	0.3	0.4	0.5
l	50*	15	26	12	10
SAIST	1.572	2.74	3.42	5	6
CPU time [s]	327	41	147	29	45

* Algorithm interrupted before finding a verified cycle.

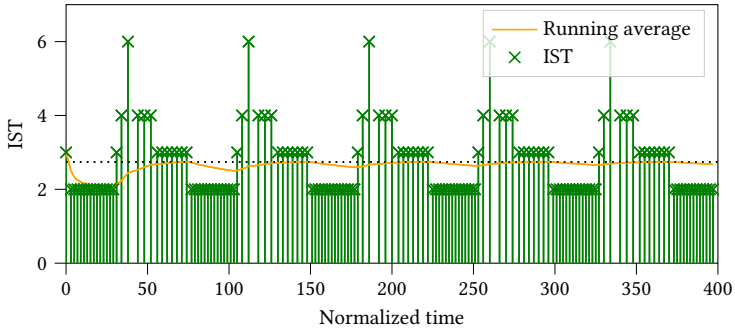


Figure 5.3: ISTs and their running average for the example of §5.6.1 with $a = 0.2$ from a pseudo-randomly generated initial state. The dashed black line represents the computed SAIST.

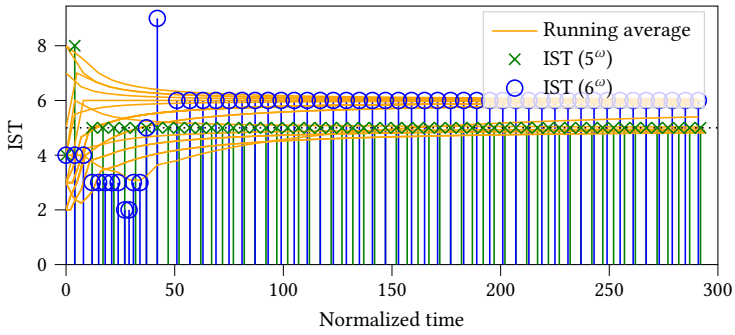


Figure 5.4: IST running averages for the example of §5.6.1 with $a = 0.4$ from 20 pseudo-randomly generated initial states, two of which with their raw ISTs depicted.The dashed black line represents the computed SAIST. In this example, there are two different periodic patterns that the system may converge to, depending on the initial state.

5.6.2 A three-dimensional linear system

With $n_x = 3$, the computational time involved in solving the existence problem of Eq. (5.2) increased significantly. Therefore we applied Alg. 3 to system (2.2)–(2.4) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{K} = \begin{bmatrix} -2 & -1 & -1 \end{bmatrix},$$

with $h = 0.1, \bar{k} = 20$ and the triggering condition $|\xi(t) - \hat{\xi}(t)| > a|\xi(t)|$. This time, some parallelization was also applied: at most 10 threads of an Intel® Xeon® W-2145 CPU were used, solving multiple instances of Eq. (5.2) in parallel whenever possible. Table 5.2 shows the results for multiple choices of a, where l now is the largest length of any state in the abstraction. The algorithm was set to a maximum depth of 200, which was only reached for $a = 0.2$. The CPU times varied dramatically, in some cases taking less than a minute, whilst in others reaching an hour. The most interesting thing we observe is that, even though the SAIST never decreases with a as expected, there is not a consistent increase on its values after $a = 0.3$. This is reasonable considering the results of §5.5.4: for small

Table 5.2: SAIST values for the example of Section 5.6.2

a	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
l	1	18*	14	8	6	7	6	5	9
SAIST	1	1.921	3	3	3	4	4	4	9.5
CPU time [s]	2	3056	1551	95	185	236	153	40	2955

enough perturbations of the ETC system's parameters, the same cycle may still be present (Prop. 5.7). Interestingly, for $a = 0.9$ there is a substantial jump in the SAIST value.

5.6.3 A nonlinear system

Consider now the PETC triggering rule $|\xi(t) - \hat{\xi}(t)| > a|\xi(t)|$ with $h = 0.05, a = 0.452$ applied to the following nonlinear jet engine system [29]:

$$\begin{aligned}\dot{\xi}_1(t) &= -\xi_2(t) - 1.5\xi_1(t)^2 - 0.5\xi_1(t)^3\\ \dot{\xi}_2(t) &= \upsilon(t),\\ \upsilon(t) &= \hat{\xi}_1(t) - 0.5(\hat{\xi}_1(t)^2 + 1)(y(t) + \hat{\xi}_1(t)^2 y(t) + \hat{\xi}_1(t) y(t)^2),\end{aligned}$$

where $y(t) = (\hat{\xi}_1(t)^2 + \hat{\xi}_2(t))/(\hat{\xi}_1(t)^2 + 1)$. The origin of the closed-loop system is asymptotically stable[10], therefore we can obtain its SAIST through its linearized model around the origin, which is of the form (2.2) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{K} = \begin{bmatrix} 1 & -0.5 \end{bmatrix}.$$

We ran Alg. 2 and stopped it with $l = 100$, obtaining an approximate value of $\mathrm{ILA}(\mathcal{S}) = 8.882$. Using Prop. 5.3, an upper bound of 8.892 was obtained, thus giving an error of 0.01. Figure 5.5 shows ISTs and their running averages for five PETC simulations starting each from a different pseudo-randomly generated initial state, for both the nonlinear model and the linearized model. It can be seen that the running averages in both cases converge to the predicted SAIST value, even though the averages are significantly different in the beginning of the simulation. The right plot shows how the difference between ISTs based on the nonlinear model and the linear model diminish as the state norm approaches zero: in the plotted simulation there is no error after the state norm is below 0.03 (around time instant 400).

5.6.4 Section 4.4 revisited

Before finishing the numerical examples section, it is worth revisiting the example of the previous chapter. Recall the discussion of the introduction: the lower bound to the SAIST of that system was 1.5, witnessed by the sequence $(4, 1, 1, 1, 1, 1)$; and while this is 50% higher than the trivial value of 1 given by the MIST (minimum inter-sample time), it is far from

[10] For stability analysis of PETC of nonlinear systems, see, e.g., [47].

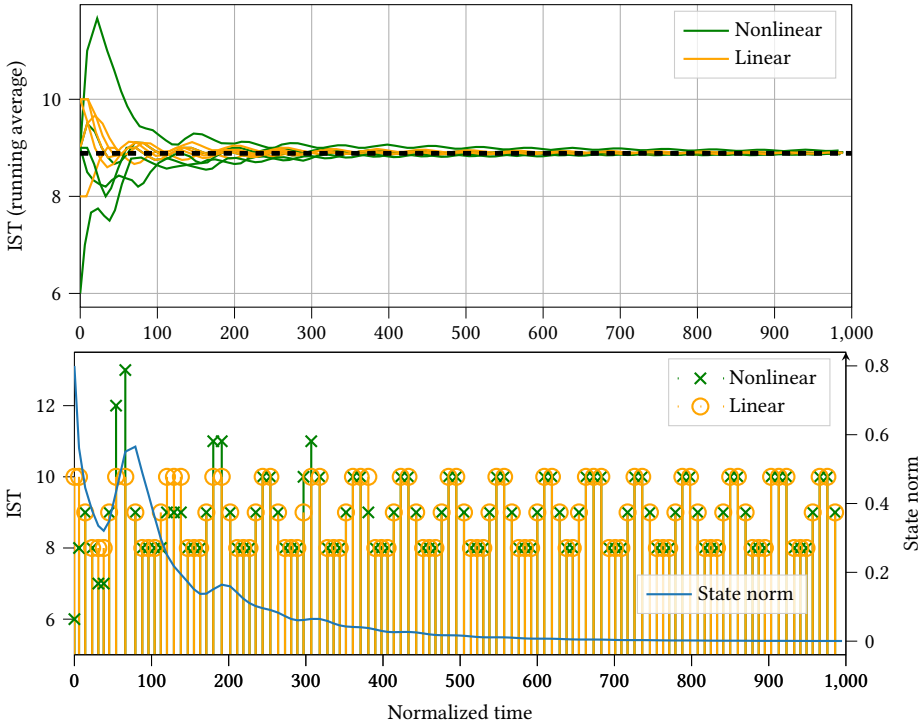


Figure 5.5: Top: running average of ISTs of five nonlinear PETC simulations and of five corresponding linear PETC simulations, with the dashed black line representing the estimated SAIST. Bottom: ISTs for one nonlinear PETC simulation and the corresponding ISTs predicted by the linear PETC model, with the state norm overlaid on a secondary axis.

the value of 4 that can still stabilize the system through periodic sampling (and somehow justifies the MPETC approach of said chapter). However, as we have seen, Prop. 4.4 can be extremely conservative as it does not consider which behaviors are transient and which are persistent. Using the improved version of Algorithm 2, as described in Sec. 5.6.2, we obtain after only 20 seconds of computation the exact SAIST of 3.97, witnessed by the behavior $(2,6,6,1,6,5,6,6,6,6,6,1,1,1,1,2,6,6,1)^\omega$. This is nearly the same SAIST of the best periodic sampling that can be found for that controller satisfying Assumption 4.2, and 4x higher than the MIST.

5.7 Conclusions

We have presented a method to compute the sampling performance of PETC, namely its smallest average inter-sample time, by means of an abstraction called SACE simulation. For this we rely on methods of abstracting and refining to obtain tighter simulations, and getting their smallest-in-average cycle through Karp's algorithm. A SACE simulation requires that this cycle, repeated *ad infinitum*, is a behavior of the concrete system; for this, we need to find an invariant of the system, which is possible for PETC of linear systems through the inspection of linear invariants of an associated discrete-time linear system. In the generic case — quotient sets with non-empty interior and linear invariants not touching

the boundary of the cones they belong to — a SACE simulation is proven to be robust to small model uncertainties, which allows us to use the presented method to a large class of nonlinear systems. Even if an exact SACE simulation is not obtained, every simulation provides a lower bound to the SAIST, and upper bounds can also be computed from the abstractions. Our numerical results indicate that these bounds can be very close after sufficiently many refinements.

As with most applications of finite-state abstractions, our approach suffers from the "curse of dimensionality": with a three-dimensional system the computation can reach nearly an hour to complete. In fact, it can be argued that this curse is more severe in our case than in most control and verification applications, since we rely on *strongest* l-complete abstractions, which require no spurious behavior fragments of length up to l. This may prevent the usage of most reachability tools to this end, as over- or under-approximations can create such spurious behaviors or remove potentially important ones. This is one of the reasons why we have used Z3 for our implementation, as it is one of the few exact nonlinear SAT solvers available. Nevertheless, the robustness results we have presented indicate that exactness may not be necessary in most cases. This is a stronger reason to use approximate nonlinear SMT solvers such as dReal [69] in order to start addressing the issue of dimensionality, as discussed in the previous chapter.

It is interesting to observe that the problem of computing the (smallest) limit average metric of an infinite system is highly dependent on its infinite behavior properties: systems with aperiodic behavior can make it impossible to obtain a SACE simulation, but other pathological behaviors can be even worse, such as the infamous $(1^n 2^n)^\omega$, where not even a good approximation can be achieved. Better behavioral understanding of systems is crucial for the further development of quantitative verification methods. This understanding is the main subject of Chapter 6, which has implications in metric computation.

Finally, natural extensions of this line of work are extending it to systems with disturbances, in particular stochastic noise, which is addressed in [31], as well as the usage of abstractions for *synthesis* of sampling strategies that maximize the closed-loop SAIST, which we discuss in Chapter 7.

5.A Proofs

5.A.1 Proof of Theorem 5.2

The proof relies on the notion of *cyclic permutations*. A word σ' is called a cyclic permutation of $\sigma \coloneqq a_0 a_1 ... a_n$ if $\sigma' = a_i a_{i+1} ... a_n a_0 a_1 ... a_{i-1}$ for some $i \leq n$. For example, the cyclic permutations of 1234 are 1234, 2341, 3412, and 4123. Clearly, all n-long subsequences of σ^ω are precisely the cyclic permutations of σ. Now we introduce the following Lemmas:

Lemma 5.1. *Let $\sigma \in \mathcal{Y}^n$ and $\sigma' \in \mathcal{Y}^n$ be cyclic permutations of each other. If $\sigma = \alpha a$ and $\sigma' = \alpha b$, where $\alpha \in \mathcal{Y}^{n-1}$ and $a, b \in \mathcal{Y}$, then $a = b$ and thus $\sigma = \sigma'$.*

Proof. Let $\sigma = a_0 a_1 ... a_{n-1}$. Then $\sigma' = a_i a_{i+1} ... a_{n-1} a_0 ... a_{i-1}$ for some $i > 0$ (if $i = 0$ the result is trivial). If their $(n-1)$-long prefixes are equal, then $a_j = a_{j+i \bmod n}$ for all $j < n-1$. In particular, take $j = i-1$; then $a_{i-1} = a_{2i-1 \bmod n} = a_{3i-1 \bmod n} = ... = a_{ki-1 \bmod n}$, where k is the smallest number such that $ki - 1 \bmod n = n-1$ (in the worst case, $k = n$, for i and n coprime). Thus, $a_{i-1} = a_{ki-1 \bmod n} = a_{n-1}$, concluding the proof. □

Lemma 5.2. *Let $\sigma \in \mathcal{Y}^n$ and $\sigma' \in \mathcal{Y}^n$ be cyclic permutations of each other. If $\sigma \neq \sigma'$, then there is a subsequence α of length n of $\sigma\sigma'$ that is not a cyclic permutation of σ.*

Proof. Let $\sigma = a_0 a_1 ... a_{n-1}$. Then $\sigma' = a_i a_{i+1} ... a_{n-1} a_0 ... a_{i-1}$ for some $i > 0$. We have $\sigma\sigma' = a_0 a_1 ... a_{n-1} a_i a_{i+1} ... a_{n-1} a_0 ... a_{i-1}$.

Suppose, for contradiction, that every n-long subsequence of $\sigma\sigma'$ is a cyclic permutation of σ. Let us look at the first nontrivial subsequence, $\sigma_1 \coloneqq a_1 ... a_{n-1} a_i$. Because $a_1 ... a_{n-1} a_0$ is a cyclic permutation of σ, from Lemma 5.1 we get that $a_0 = a_i$. Now let us apply induction: suppose that for some $J < n$, $a_j = a_{i+j \bmod n}$ for all $j < J$; we are going to show that this also holds for $j = J$. First, suppose that $J < n - i$; then $\sigma_J = a_J a_{J+1} ... a_{n-1} a_i a_{i+1} ... a_{i+J-2} a_{i+J-1} = a_J a_{J+1} ... a_{n-1} a_0 a_1 ... a_{J-2} a_{i+J-1}$. Again, because $a_J a_{J+1} ... a_{n-1} a_0 a_1 ... a_{J-1}$ is a cyclic permutation of σ, apply Lemma 5.1 to obtain $a_{i+J-1} = a_{J-1}$. Second, suppose that $J \geq n - i$. Then, $\sigma_J = a_J ... a_{n-1} a_i .. a_{n-1} a_0 a_1 ... a_{i+J-n-1} = a_J ... a_{n-1} a_0 .. a_{n-i-1} a_0 a_1 ... a_{i+J-n-1}$. Note that $a_k = a_{k+n \bmod n} = a_{k+n-i}$ as long as $k + n - i < J$, i.e., $k < i + J - n$. Thus, we arrive at $\sigma_J = a_J ... a_{n-1} a_0 .. a_{n-i-1} a_{n-i} ... a_{J-2} a_{i+J-n-1}$. Again, apply Lemma 5.1 to get that $a_{i+J-n-1} = a_{J-1}$. We have that $J - 1 + i \bmod n = i + J - n - 1$, since $n > J \geq n - i$; our hypothesis is thus confirmed. The fact that $a_j = a_{i+j \bmod n}$ for all $j < n$ implies that $\sigma' = a_i a_{i+1} ... a_{n-1} a_0 ... a_{i-1} = a_0 a_1 ... a_{n-1-i} a_{n-i} ... a_{n-1} = \sigma$, which contradicts the fact that $\sigma \neq \sigma'$. □

Proof of Theorem 5.2. From Theorem 5.1, there is an l large enough such that $\mathrm{ILA}(\mathcal{S}_l) = \mathrm{ILA}(\mathcal{S})$. It is easy to see that taking $l \geq m$ ensures that σ is a cycle of the graph associated to $\mathcal{S}_l$.

We prove that, because now $\mathrm{LimAvg}(\beta^\omega) > \mathrm{ILA}(\mathcal{S})$ for every β that is not a subsequence of σ^ω (thus not a cyclic permutation of σ), the SAC of $\mathcal{S}_l$ is unique up to cyclic permutations. Suppose, for contradiction, that another cycle σ' is a SAC of $\mathcal{S}_l$, with $|\sigma'| = p$. As in the proof of Theorem 5.1, we divide $(\sigma')^m$ into p subsequences of length m, obtaining

$$\mathrm{ILA}(\mathcal{S}_l) = \mathrm{LimAvg}((\sigma')^\omega) = \mathrm{LimAvg}(((\sigma')^m)^\omega)) = \mathrm{Avg}((\sigma')^m) = \frac{1}{p}\sum_{i=1}^{p} \mathrm{Avg}(\beta_i).$$

5

If (i) some β_i is not a cyclic permutation of σ, $\frac{1}{p}\sum_{i=1}^{p} \mathrm{Avg}(\beta_i) > \mathrm{ILA}(S_l)$, which yields the contradiction. Now, suppose (ii) that every β_i is a cyclic permutation of σ; since σ' is not the same cycle as σ, it cannot be that $\beta_i = \beta_j$ for all $i, j \le p$. If $\beta_i \neq \beta_j$ for some i, j, suppose without loss of generality that they are adjacent in $(\sigma')^\omega$, i.e., either $j = i+1$ or $i = p$ and $j = 1$. Then we have from Lemma 5.2 that there exists an m-long subsequence of $\beta_i\beta_j$ that is not a cyclic permutation of σ. Thus, σ' has at least one subsequence β' with average larger than $\mathrm{ILA}(S)$, which brings us back to case (i). The contradiction is thus achieved in all cases.

Concluding, S_l has only one cycle σ (modulo cyclic permutations) that attains its minimum value. Hence, running Karp's algorithm (Theorem 2.2) retrieves it; by assumption, $\sigma^\omega \in \mathcal{B}^\omega(S)$, thus the algorithm terminates at line 6. □

5.A.2 Proof of Theorem 5.3

Before the main proof, we need some definitions. Given a map $f : X \to X$ and the discrete-time autonomous system defined by $x_{i+1} = f(x_i)$, we call the *forward orbit* of x the set $\mathcal{O}(x) := \{f^n(x) \mid n \in \mathbb{N}\}$. The ω-limit set of x, denoted by $\omega(x)$ is the set of cluster points of $\mathcal{O}(x)$, or alternatively,

$$\omega(x) = \bigcap_{n\in\mathbb{N}} \mathrm{cl}(\{f^k(x) \mid k > n\}).$$

By definition of closure, if $\mathcal{O}(x) \subset \mathcal{A} \subset X$, then $\omega(x) \subset \mathrm{cl}(\mathcal{A})$.

We introduce the following Lemma.

Lemma 5.3. *Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a nonsingular mixed matrix and $\mathcal{Q} \subseteq \mathbb{R}^n$ be a homogeneous set, i.e., it satisfies $\boldsymbol{x} \in \mathcal{Q} \implies \lambda\boldsymbol{x} \in \mathcal{Q}, \forall \lambda \in \mathbb{R}\setminus\{0\}$. If there exists a trajectory $\boldsymbol{\xi} : \mathbb{N} \to \mathbb{R}^{n_x}$ satisfying $\boldsymbol{\xi}(k+1) = \boldsymbol{M}\boldsymbol{\xi}(k)$ and $\boldsymbol{\xi}(k) \in \mathcal{Q}\ \forall k \in \mathbb{N}$, then there exists a linear subspace $\mathcal{A}$ that is an invariant of $\boldsymbol{M}^q$ and satisfies $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q})$, where $q \in \mathbb{N}$. Furthermore, $q = 1$ if $\boldsymbol{M}$ is of irrational rotations.*

Proof. Because $\mathcal{Q}$ is homogeneous, $\boldsymbol{\xi}(k) \in \mathcal{Q}$ for all k implies that the normalized trajectory $\boldsymbol{\xi}(k)/|\boldsymbol{\xi}(k)| \in \mathcal{Q}$ for all k; likewise, for any constant $c \neq 0$, we have that $c\boldsymbol{\xi}(k)/|\boldsymbol{\xi}(k)| \in \mathcal{Q}$. Therefore, let us investigate the "normalized" version of the iteration $\boldsymbol{x}_{i+1} = \boldsymbol{M}\boldsymbol{x}_i$: this is defined by the map $f : B^n \to B^n$, where B^n is the unit ball in $\mathbb{R}^n$ and $f(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x}/|\boldsymbol{M}\boldsymbol{x}|$. Our strategy is to first determine what is $\omega(\boldsymbol{x})$; then, we will prove that the set $\{c\omega(\boldsymbol{x}) \mid c \in \mathbb{R}\setminus\{0\}\}$, a radial expansion of $\omega(\boldsymbol{x})$, is a linear subspace of $\boldsymbol{M}$. Because $\omega(\boldsymbol{x}) \subseteq \mathrm{cl}(\mathcal{Q})$ and $\boldsymbol{x} \in \mathcal{Q} \implies c\boldsymbol{x} \in \mathcal{Q}$, we conclude that $\{c\omega(\boldsymbol{x}) \mid c \in \mathbb{R}\setminus\{0\}\} \subseteq \mathrm{cl}(\mathcal{Q})$.

Now we investigate case by case depending on the eigenvalues λ_i of $\boldsymbol{M}$. Since $\boldsymbol{M}$ is mixed, it is diagonalizable, and hence the trajectory $\boldsymbol{\xi}(k)$ can be decomposed as $\sum_{i=1}^{n} a_i\boldsymbol{v}_i\lambda_i^k$, where $\boldsymbol{v}_i$ are the eigenvectors of $\boldsymbol{M}$ satisfying $|\boldsymbol{v}_i| = 1$, and the coefficients a_i are chosen such that $\boldsymbol{\xi}(0) = \sum_{i=1}^{n} a_i\boldsymbol{v}_i$. Let $m \le n$ such that $a_i = 0$ for $i < m$, hence λ_m is the dominant eigenvalue for this initial condition. Throughout, let $\boldsymbol{x} := \boldsymbol{\xi}(0)/|\boldsymbol{\xi}(0)|$.

Case 1: λ_m is real. Then

$$\lim_{k\to\infty}\frac{\boldsymbol{\xi}(k)}{|\boldsymbol{\xi}(k)|}=\lim_{k\to\infty}\frac{a_m\boldsymbol{v}_m\lambda_m^k+\ldots+a_n\boldsymbol{v}_n\lambda_n^k}{|a_m\boldsymbol{v}_m\lambda_m^k+\ldots+a_n\boldsymbol{v}_n\lambda_n^k|}$$
$$=\lim_{k\to\infty}\frac{a_m\boldsymbol{v}_m+\ldots+a_n\boldsymbol{v}_n\left(\frac{\lambda_n}{\lambda_m}\right)^k}{\left|a_m\boldsymbol{v}_m+\ldots+a_n\boldsymbol{v}_n\left(\frac{\lambda_n}{\lambda_m}\right)^k\right|}=\lim_{k\to\infty}\frac{a_m\boldsymbol{v}_m}{|a_m\boldsymbol{v}_m|}=\pm a_m\boldsymbol{v}_m.$$

Hence, the set $\{c\omega(\boldsymbol{x})\mid c\in\mathbb{R}\setminus\{0\}\}$ is the line $\{\pm c\boldsymbol{v}_m\mid c\in\mathbb{R}\setminus\{0\}\}=\{c\boldsymbol{v}_m\mid c\in\mathbb{R}\setminus\{0\}\}$, which is an invariant of $\boldsymbol{M}$.

For the next cases, λ_m and λ_{m+1} form a complex conjugate pair, thus $\boldsymbol{v}_{i+1}=\boldsymbol{v}_i^*$. Denote by $\theta:=\arg\lambda_m$.

Case 2: $\theta/\pi\notin\mathbb{Q}$. Using a similar approach as Case 1, we get $\lim_{k\to\infty}\boldsymbol{\xi}(k)/|\boldsymbol{\xi}(k)|=\pm(\boldsymbol{v}_m\mathrm{e}^{\mathrm{i}\theta k}+\boldsymbol{v}_{m+1}\mathrm{e}^{-\mathrm{i}\theta k})$. Because θ is not a rational multiple of π, $\{k\theta\mid k\in\mathbb{N}\}$ is a dense subset of $[0,2\pi]$ and, therefore, $\omega(\boldsymbol{x})=\mathrm{cl}(\{\pm(\boldsymbol{v}_m\mathrm{e}^{\mathrm{i}\theta k}+\boldsymbol{v}_{m+1}\mathrm{e}^{-\mathrm{i}\theta k}\mid\theta\in k\in\mathbb{N}\}$ which is equal to the ellipse $\mathcal{B}:=\{\boldsymbol{v}_m\mathrm{e}^{\mathrm{i}\alpha}+\boldsymbol{v}_{m+1}\mathrm{e}^{-\mathrm{i}\alpha}\mid\alpha\in[0,2\pi)\}$. The set $\{c\boldsymbol{x}\mid\boldsymbol{x}\in\mathcal{B},c\in\mathbb{R}\setminus\{0\}\}$ is the unique plane supported by $\boldsymbol{v}_m$ and $\boldsymbol{v}_{m+1}$, and as such is an invariant of $\boldsymbol{M}$.

Case 3: $\theta/\pi=p/q$, where $p,q\in\mathbb{N}$ are co-prime. The m-th and $(m+1)$-th eigenvalues of $\boldsymbol{M}$ have the form $r\mathrm{e}^{\pm\mathrm{i}p\pi/q}$, and as a consequence the corresponding eigenvalues of $\boldsymbol{M}^q$ are $\lambda_m^q=\lambda_{m+1}^q=r^q\in\mathbb{R}$. The geometric multiplicity of λ_m^q is 2, since $\boldsymbol{M}^q$ is also diagonalizable. Thus, we have that $\lim_{k\to\infty}\boldsymbol{\xi}(qk)/|\boldsymbol{\xi}(qk)|=a_m\boldsymbol{v}_i+a_{m+1}\boldsymbol{v}_{i+1}=:\boldsymbol{z}$. Hence, we have $\omega(\boldsymbol{x})\supseteq\{c\boldsymbol{z}\mid c\in\mathbb{R}\setminus\{0\}\}$, a line that is an invariant of $\boldsymbol{M}^q$. Finally, this line is a subset of $\mathrm{cl}(\mathcal{Q})$, since $\{c\boldsymbol{z}\mid c\in\mathbb{R}\setminus\{0\}\}\subseteq\omega(\boldsymbol{x})\subseteq\mathrm{cl}(\mathcal{Q})$. □

Proof of Theorem 5.3. Statement (i), $\mathcal{A}\setminus\{\boldsymbol{0}\}\subseteq\mathcal{Q}_\sigma$ implies $\sigma^\omega\in\mathcal{B}(\mathcal{S})$, is straightforward. Take any point $\boldsymbol{x}\in\mathcal{A}\subseteq\mathcal{Q}_\sigma$. By definition of $\mathcal{Q}_\sigma$, we have that $\boldsymbol{x}\in\mathcal{Q}_{k_1},\boldsymbol{M}(k_1)\boldsymbol{x}\in\mathcal{Q}_{k_2},\ldots$, and $\boldsymbol{M}(k_{m-1})\cdots\boldsymbol{M}(k_1)\boldsymbol{x}\in\mathcal{Q}_{k_m}$. The $(m+1)$-th element of the run starting from initial state $\boldsymbol{x}$ is $\boldsymbol{x}'=\boldsymbol{M}(k_m)\boldsymbol{M}(k_{m-1})\cdots\boldsymbol{M}(k_1)\boldsymbol{x}=\boldsymbol{M}_\sigma\boldsymbol{x}$. Since $\mathcal{A}$ is an invariant of $\boldsymbol{M}_\sigma$ and this matrix is nonsingular, $\boldsymbol{x}'\in\mathcal{A}\setminus\{\boldsymbol{0}\}$. Thus, the behavior from $\boldsymbol{x}$ is $\sigma\mathcal{B}_{\boldsymbol{x}'}(\mathcal{S})$. Applying the same reasoning recursively with $\boldsymbol{x}'$ in the place of $\boldsymbol{x}$, we conclude that $\mathcal{B}_{\boldsymbol{x}}(\mathcal{S})=\sigma^\omega$.

Statement (ii) follows from Lemma 5.3, by applying it with $\mathcal{Q}=\mathcal{Q}_\sigma$ and $\boldsymbol{M}=\boldsymbol{M}_\sigma$, and using the fact that $\mathcal{Q}_\sigma$ is an homogeneous set.

□

6

Chaos and Order in Event-Triggered Control

This chapter addresses robustness of the ETC traffic patterns that attain minimal average IST, and how this depends on the qualitative properties of the traffic. First, we notice that often these minimizing traffic patterns are fragile in the sense that they only occur for a measure-zero set of initial conditions, and small perturbations lead to completely different traffic patterns. Thus, we define and address the problem of computing robust traffic metrics of PETC. To compute such robust limit metrics, we characterize limit traffic patterns by observing invariant lines and planes through the origin, as well as their attractivity. We show that ETC can exhibit very complex, even chaotic traffic, especially when the triggering condition is aggressive in reducing communications. Then, we present abstraction-based methods to compute robust limit metrics such as limit average and limit inferior IST of PETC, as well as measuring the emergence of chaos. The proposed methodology and tools allow us to find ETC examples that provably outperform periodic sampling in terms of average IST. In particular for PETC, we prove that this requires aperiodic or chaotic traffic.

This chapter is based on *G. de A. Gleizer and M. Mazo Jr, "Chaos and order in event-triggered control," arXiv preprint arXiv:2201.04462* [75].

6.1 Introduction

While executing the algorithms of the previous chapter on a number of PETC systems and simulating them, we occasionally noticed that it was difficult to obtain, by randomly choosing the system's initial state, the metric that we had computed. Interestingly, in several cases the minimizing periodic patterns would only happen for very specific initial states; any small perturbation on those would eventually lead to a totally different traffic pattern. This could be seen a strength of our algorithm: it can find patterns that are difficult to find through mere simulation. But how useful in practice is a metric that only occurs from a measure-zero set of initial conditions?

The observation of these fleeting behaviors had already been made in [25, 26] for 2-dimensional linear systems. In fact, in most cases we would observe something similar to what was reported therein: a system has two limit fixed IST patterns, one "stable" and one "unstable", or it converges to quasi-periodic behaviors. However, as we tune the triggering parameters to make it trigger less frequently, different things start occurring. For example, as seen in Sec. 5.6.1, with $a = 0.2, 0.3$, we obtained periodic patterns that, by inspection, are "stable". This phenomenon was not observed in [25, 26], and it could perhaps be specific to PETC. But it could also be that more complex traffic patterns are only seen when the triggering condition is tuned to generate sufficiently sparse inter-sample times. We confirm this in the present chapter; in fact, ETC systems can even exhibit chaotic traffic.

The present chapter makes an attempt to expand the qualitative understanding of ETC's asymptotic traffic patterns and bridge it to the quantitative approach of Chapter 5. For that, we first characterize limit metrics of interest, such as limit inferior and limit average, and observe that they are related to the asymptotic properties of the traffic. This is the starting point for our main contributions: (i) presenting limit behaviors of LTI ETC systems and methods to compute them, not limited to $\mathbb{R}^2$; (ii) classifying limit behaviors in terms of stable vs. unstable, periodic vs. aperiodic, orderly vs. chaotic; and (iii) based on this classification, expanding the results from Chapter 5 for PETC to compute *robust* metrics. We propose auxiliary concepts and obtain results that may be useful on their own right: (i) we show that the law of evolution of state samples can be regarded as a map of the projective space to itself, which allows us to conclude that stationary traffic patterns are always exhibited in CETC for odd-dimensional systems (Theorem 6.2); (ii) we show that if a PETC that renders the origin globally asymptotically stable (GES) converges to a periodic traffic pattern, then this traffic pattern can be used as a (multi-rate) periodic sampling schedule (Prop. 6.8) — this does not necessarily happen to CETC; (iii) we provide a stability characterization for outputs of a system, when these outputs come from a finite set; and (iv) we present the notion of behavioral entropy (Def. 6.10) as a measure of chaos of a system's set of output trajectories, how to compute this quantity in an abstraction (Theorem 6.5), and show that this quantity is an upper bound of the concrete system's (Prop. 6.10).

This chapter is organized as follows: Section 6.2 is dedicated to some basic preliminaries on chaos in dynamical systems. Section 6.3 presents how ISTs can be computed, and the main problem statement. The qualitative side of the work, presenting limiting behaviors and their general properties, is given in Section 6.4, where we are able to establish conditions for which periodic patterns occur and the associated states that generate them. In doing that, we explore their local attractivity and the emergence of chaotic invariant sets. This paves

the way for the quantitative side of this work in Section 6.5 using symbolic abstractions, where we properly define robust limit metrics for PETC taking chaos into consideration, provide methods to estimate PETC's behavioral entropy, establish when traffic patterns are not involved in chaos, and describe how to estimate the desired robust limit metrics. Numerical examples are given in Section 6.6, and conclusions in Section 6.7.

6.2 Mathematical Preliminaries on Chaos

Consider the map $f : \mathcal{X} \to \mathcal{X}$ and the discrete-time system (or recursion) $\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k))$. A set $\mathcal{Y} \subset \mathcal{X}$ is said to *fixed* or *invariant* if $f(\mathcal{Y}) = \mathcal{Y}$, *forward invariant* if $f(\mathcal{Y}) \subseteq \mathcal{Y}$, and *periodic* if there is some $m \in \mathbb{N}$ such that $f^m(\mathcal{Y}) = \mathcal{Y}$. The *forward orbit* of a point $\boldsymbol{x}$ is $\mathcal{O}(\boldsymbol{x}) \coloneqq \{f^k(\boldsymbol{x}) \mid k \in \mathbb{N}_0\}$. Obviously, every forward orbit is forward invariant. Whilst there are multiple slightly different definitions of chaos, we use the concept of [76], which relies on the notions of *transitivity* and *sensitivity to initial conditions.*

Definition 6.1 (Transitivity [76, Sec. 2.5])**.** *A map $f : \mathcal{X} \to \mathcal{X}$ is said to be* (topologically) transitive *on an invariant set $\mathcal{Y}$ if the forward orbit of some point $p \in \mathcal{X}$ is dense in $\mathcal{Y}$. From the Birkhoff Transitivity Theorem, this is equivalent to the following property: for every two open subsets $\mathcal{U}$ and $\mathcal{V}$ of $\mathcal{Y}$, there is a positive integer n such that $f^n(\mathcal{U}) \cap \mathcal{V} \neq \emptyset$.*

If f is transitive, points starting arbitrarily close to each other can drift away but will come arbitrarily close back to each other after enough iterations.

Definition 6.2 (Sensitivity to initial conditions[76, Sec. 3.5])**.** *A map $f : \mathcal{X} \to \mathcal{X}$, $\mathcal{X}$ being a metric space, is said to be* sensitive to initial conditions *on an invariant set $\mathcal{Y} \subseteq \mathcal{X}$ if there is an $r > 0$ such that, for each point $\boldsymbol{x} \in \mathcal{Y}$ and for each $\epsilon > 0$, there exists a point $\boldsymbol{y} \in \mathcal{Y}$ satisfying $d(\boldsymbol{x}, \boldsymbol{y}) < \epsilon$ and a $k \in \mathbb{N}$ with $d(f^k(\boldsymbol{x}), f^k(\boldsymbol{y})) \geq r$.*

Definition 6.3 (Chaos[76, Sec. 3.5])**.** *A map $f : \mathcal{X} \to \mathcal{X}$, $\mathcal{X}$ being a metric space, is said to be* chaotic on an invariant set $\mathcal{Y}$ *provided (i) f is transitive on $\mathcal{Y}$, and (ii) f is sensitive to initial conditions on $\mathcal{Y}$.*

In case a chaotic system is additionally *ergodic*[1] the celebrated Birkhoff Ergodic Theorem is particularly useful when one is interested in limit average metrics:

Theorem 6.1 (Birkhoff Ergodic Theorem [76])**.** *Assume $f : \mathcal{X} \to \mathcal{X}$ is an ergodic function with ergodic measure μ, and let $g : \mathcal{X} \to \mathbb{R}$ be a μ-integrable function. Then,*

$$\lim_{n \to \infty} \frac{1}{n+1} \sum_{i=1}^{n} g \circ f^i(x) = \int_{\mathcal{X}} g(x) \mathrm{d}\mu(x)$$

for μ-almost every x.

As a consequence, if f is ergodic and μ is well-behaved[2], the time-average (left-hand side of the expression above) converges to the same value from almost every initial condition.

[1] See [76] for a rigorous definition of ergodicity. We skip the definition and present a simplified version of the Birkhoff Ergodic Theorem due to readability considerations.

[2] That is, $0 < \mu(x) < \infty$ for all x in the invariant set of interest; this prevents considering Dirac deltas or points out of the ergodic invariant set; as a consequence μ-almost all is the same as almost all.

6.3 Event-triggered control and its traffic

Consider the traffic model of an ETC system (2.2)–(2.4) through its sample map

$$\begin{aligned} \boldsymbol{x}_{i+1} &= f(\boldsymbol{x}_i), \\ y_i &= \tau(\boldsymbol{x}_i). \end{aligned} \tag{2.8 revisited}$$

As in the previous chapter, we shall denote the sequence of outputs from Eq. (2.8) for a given initial state $\boldsymbol{x}_0$ by $\{y_i(\boldsymbol{x}_0)\}$. Let us investigate the possible inter-sample behaviors of an ETC system.

6.3.1 Isochronous subsets in ETC

We start our analysis of inter-sample behaviors of ETC by studying the subsets of $\mathbb{R}^{n_x}$ that generate the same inter-sample time. The first characteristic to be highlighted is that inter-sample times are insensitive to magnitude.

Proposition 6.1 (Adapted from [14])**.** *The sample system* (2.8) *is homogeneous; more specifically, for all* $\lambda \in \mathbb{R} \setminus \{0\}$, $\tau(\lambda \boldsymbol{x}) = \tau(\boldsymbol{x})$ *and* $f(\lambda \boldsymbol{x}) = \lambda f(\boldsymbol{x})$.

Proof. With respect to Eq. (2.6), $\operatorname{sign}((\lambda \boldsymbol{x})^{\mathsf{T}} \boldsymbol{Q}(s)(\lambda \boldsymbol{x})) = \operatorname{sign}(\lambda^2 \boldsymbol{x}^{\mathsf{T}} \boldsymbol{Q}(s) \boldsymbol{x}) = \operatorname{sign}(\boldsymbol{x}^{\mathsf{T}} \boldsymbol{Q}(s) \boldsymbol{x})$, hence $\tau(\lambda \boldsymbol{x}) = \tau(\boldsymbol{x})$. With this, $f(\lambda \boldsymbol{x}) = \boldsymbol{M}(\tau(\lambda \boldsymbol{x}))\lambda \boldsymbol{x} = \lambda \boldsymbol{M}(\tau(\boldsymbol{x}))\boldsymbol{x} = \lambda f(\boldsymbol{x})$. □

This fact implies that the sequence $\{y_i(\boldsymbol{x})\}$ is equal to $\{y_i(\lambda \boldsymbol{x})\}$, for any $\lambda \neq 0$. Hence, to determine whether ETC exhibits fixed (periodic) behavior, we need to verify which lines passing thorough the origin, or collections of lines, are invariant under f or under a finite iterate of f. Hereafter we shall refer to lines that pass through the origin as *o-lines*.

Let us first look in detail what are the subsets of $\mathbb{R}^{n_x}$ which share the same inter-event time:

Definition 6.4. *Consider system* (2.8). *We denote by* $\mathcal{Q}_s \subset \mathbb{R}^{n_x}$*, the set of all states which trigger after* s *time units, i.e.,*

$$\mathcal{Q}_s \coloneqq \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \tau(\boldsymbol{x}) = s\}.$$

We call $\mathcal{Q}_s$ *an* isochronous subset.[3]

Proposition 6.2. *Consider system* (2.2)–(2.4). *An isochronous subset* $\mathcal{Q}_s$ *can be characterized as*

i) If $\mathcal{T} = \mathbb{R}_+$ *(CETC) and* $s < \bar{\tau}$, $\mathcal{Q}_s = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathsf{T}} \boldsymbol{N}(s) \boldsymbol{x} = 0$ *and* $\boldsymbol{x}^{\mathsf{T}} \boldsymbol{N}(s') \boldsymbol{x} \leq 0, \forall s' < s$ *and* $\boldsymbol{x}^{\mathsf{T}} \dot{\boldsymbol{N}}(s) \boldsymbol{x} > 0\}$.

ii) If $\mathcal{T} = h\mathbb{N}$ *(PETC) and* $s < \bar{\tau}$, $\mathcal{Q}_s = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathsf{T}} \boldsymbol{N}(s) \boldsymbol{x} > 0$ *and* $\boldsymbol{x}^{\mathsf{T}} \boldsymbol{N}(s') \boldsymbol{x} \leq 0, \forall s' < s, s' \in h\mathbb{N}\}$.

iii) $\mathcal{Q}_{\bar{\tau}} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathsf{T}} \boldsymbol{N}(s') \boldsymbol{x} \leq 0, \forall s' < \bar{\tau}, s' \in \mathcal{T}\}$.

Proof. This is a trivial manipulation of Eq. (2.6), where in (i) we use the fact that $\boldsymbol{N}(s)$ is differentiable over $[0, \bar{\tau})$. □

[3] The concept of isochronous manifolds was introduced in ETC for nonlinear homogeneous systems in [77]. In our case, the isochronous subsets of CETC are (in general) $(n_x - 1)$-dimensional subsets, but may not be manifolds. Whether they are manifolds or not is not relevant to our results.

The isochronous subset $\mathcal{Q}_{\bar{\tau}}$ is the intersection of an algebraic set with infinitely many semialgebraic sets for CETC; for PETC, it is the intersection of finitely many semialgebraic sets. We can extend the definition of isochronous subset to a sequence of inter-sample times:

Definition 6.5 (Isosequential subset)**.** *Consider the system* (2.8)*. The set* $\mathcal{Q}_{y\sigma}$*,* $y \in \mathcal{T}, \sigma \in \mathcal{T}^{m-1}$ *for some* $m \in \mathbb{N}$*, is defined recursively as the set of states* $\boldsymbol{x} \in \mathbb{R}^{n_x}$ *such that* $\boldsymbol{x} \in \mathcal{Q}_{y_1}$*, and* $\boldsymbol{M}(y_1)\boldsymbol{x} \in \mathcal{Q}_\sigma$*. By convention,* $\mathcal{Q}_\epsilon = \mathbb{R}^{n_x}$*, where* ϵ *denotes the empty sequence.*

As we can see, the set $\mathcal{Q}_\sigma$ is also the intersection of (semi)algebraic sets as in the singleton case. We end this section with a result that simplifies the analysis for CETC under some special conditions.

Proposition 6.3. *Consider system* (2.8) *and* $\mathcal{T} = \mathbb{R}_+$ *(CETC). If* $\bar{\tau} = \inf\{s > 0 \mid \boldsymbol{N}(s) \succ \boldsymbol{0}\} < \infty$ *and*

$$\forall \boldsymbol{x} \in \mathbb{R}^{n_x} \setminus \{\boldsymbol{0}\}, s \in (0, \bar{\tau}], \boldsymbol{x}^\mathsf{T}\boldsymbol{N}(s)\boldsymbol{x} = 0 \implies \boldsymbol{x}^\mathsf{T}\dot{\boldsymbol{N}}(s)\boldsymbol{x} > 0$$

then

i) f *and* τ *are differentiable;*

ii) $\mathcal{Q}_s = \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{x}^\mathsf{T}\boldsymbol{N}(s)\boldsymbol{x} = 0\}, \forall s \in (0, \bar{\tau})$.

Proof. We first prove (ii), which is a lemma to (i).

ii) Consider the function $\phi_{\boldsymbol{x}}(s) = \boldsymbol{x}^\mathsf{T}\boldsymbol{N}(s)\boldsymbol{x}$, which is differentiable. We want to prove that, if for some s, $\phi_{\boldsymbol{x}}(s) = 0$, then all conditions from Prop. 6.2 (i) are satisfied; since $\dot{\phi}_{\boldsymbol{x}}(s) > 0$ by assumption, we need to prove that $\phi_{\boldsymbol{x}}(s') \le 0, \forall s' < s$. Now, since $\phi_{\boldsymbol{x}}(s) = 0 \implies \dot{\phi}_{\boldsymbol{x}}(s) > 0$, from continuity, it holds that $\phi_{\boldsymbol{x}}(s^-) < 0$ for some $s^- < s$. For contradiction, assume $\phi_{\boldsymbol{x}}(s') > 0$ for some $s' < s^-$. Then, from Bolzano's theorem there is some point $s'' \in (s', s^-)$ such that that $\phi_{\boldsymbol{x}}(s'') = 0$. One such s'' must have $\phi_{\boldsymbol{x}}(s'')$ cross zero from positive to negative, which implies $\dot{\phi}_{\boldsymbol{x}}(s'') \le 0$, leading to a contradiction.

i) Now $\tau(\boldsymbol{x})$ is characterized by the implicit equation $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(\tau)\boldsymbol{x} = 0$. Therefore we can simply apply the implicit function theorem, whose condition ($\phi_{\boldsymbol{x}}(s) = 0 \Rightarrow \dot{\phi}_{\boldsymbol{x}}(s) \neq 0$) is satisfied by ours. □

Remark 6.1. *The condition in Proposition 6.3 is equivalent, by the s-procedure, to the linear matrix inequality* $\exists \lambda \in \mathbb{R} : \lambda \boldsymbol{N}(s) + \dot{\boldsymbol{N}}(s) \succ \boldsymbol{0}$*. Note that it is trivially satisfied if* $\dot{\boldsymbol{N}}(s) \succ \boldsymbol{0}$ *for all* $s \in [0, \bar{\tau}]$*, which holds when the triggering function* $\phi_{\boldsymbol{x}}$ *is monotonically increasing for all* $\boldsymbol{x}$*.*

The condition in Proposition 6.3 ensures that the triggering function crosses zero only once for each initial condition $\boldsymbol{x}$, which in turn simplifies the isochronous subset description to a simple quadratic form and renders f and τ continuous. As we will see, even when this continuity is observed, the behaviors generated by ETC can be extremely rich.

6.3.2 Problem statement

Expanding on Chapter 5, we are interested in quantifying the traffic usage of system (2.2)–(2.4), which involves studying the sample system (2.8). Some candidate metrics are the following:

- $\mathrm{Inf} := \inf_{\boldsymbol{x} \in \mathbb{R}^{n_x}} \tau(\boldsymbol{x})$;

- $\mathrm{Sup} \coloneqq \sup_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \tau(\boldsymbol{x})$;
- $\mathrm{InfLimInf} \coloneqq \inf_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \liminf_{i\to\infty} y_i(\boldsymbol{x})$;
- $\mathrm{SupLimSup} \coloneqq \sup_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \limsup_{i\to\infty} y_i(\boldsymbol{x})$;
- $\mathrm{InfLimAvg} \coloneqq \inf_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} y_i(\boldsymbol{x})$.
- $\mathrm{SupLimAvg} \coloneqq \sup_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \limsup_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} y_i(\boldsymbol{x})$.

The first two metrics are simply the minimal and maximal inter-sample times that can be exhibited. The minimal is the one that has received most attention in the literature, mainly to prove absence of Zeno behavior for different triggering conditions. These metrics serve as worst- and best-case inter-event times and provide a basic information about how sample-efficient one given ETC system is. Inf is trivially calculated as $\mathrm{Inf} = \inf\{s \in \mathcal{T} \mid \boldsymbol{N}(s) \prec \boldsymbol{0}\}$, while Sup is a bit more complicated: $\mathrm{Sup} = \min(\bar{\tau}, \inf\{s \in \mathcal{T} \mid \forall \boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}} \exists s' < s\ \boldsymbol{x}^\mathsf{T}\boldsymbol{N}(s')\boldsymbol{x} > \boldsymbol{0}\})$. The last four metrics concern limit behaviors of the system. InfLimInf gives what is the minimal inter-sample time the system can exhibit as the number of samples goes to infinity: in other words, after transients on the sequence y_i vanish. The symmetric case value is given by SupLimSup. Finally, InfLimAvg (SupLimAvg) gives the minimum (maximum) among initial states of average inter-sample time. Here, lim inf (lim sup) is used to ensure that the value exists even if the sequence of averages does not converge.

We argue that the limit metrics are more informative to determine the performance of a sampling mechanism than the simpler Inf and Sup metrics. For instance, if the states $\boldsymbol{x}$ associated to Inf are transient, in the sense that from almost all other initial states they are never visited, the Inf metric turns out to be very conservative; after a few samples, the typical inter-sample time of the system will be higher. InfLimInf gives the complementary information of what minimal inter-sample time can appear infinitely often. InfLimAvg informs about the average utilization rate. A disadvantage of these two metrics is that they can still capture exceptional behavior: suppose, for example, that a measure-zero set $\mathcal{X} \subset \mathbb{R}^{n_\mathrm{x}}$ is invariant under (2.8) and it is associated to the InfLimInf or InfLimAvg of the system; moreover, suppose for every state $\boldsymbol{x} \notin \mathcal{X}$, the trajectories $\xi_{\boldsymbol{x}}(t_i), i \in \mathbb{N}$, never enter $\mathcal{X}$, but instead converge to some other subset with higher values of InfLimInf or InfLimAvg. Then, the metric will not reflect the dominant performance of the system. This information might still be useful, but a more robust version of these metrics is of interest. In any case, robust or not, we need a hint of how one could compute these metrics. This will allow us to properly define what robust should be in this context.

Problem Statement. Given an ETC system, (i) identify its limit traffic patterns, (ii) characterize their robustness w.r.t. small perturbations in the initial state, and (iii) compute the system's robust limit metrics.

6.4 Qualitative analysis: limit behaviors in ETC

In this section we investigate the limit behaviors of the traffic generated by ETC. We first see that limit metrics are insensitive to transient behavior; then we look at some examples to classify the different limit behaviors that can be exhibited. In several cases, ETC traffic converges to a periodic sampling pattern, which is shown to be characterized by linear invariants. This characterization allows us to show that, if PETC stabilizes a periodic traffic pattern, then this traffic pattern can be used as a sampling schedule that guarantees GES of the system.

6.4.1 Properties of limit metrics

The following trivial result shows that limit metrics are insensitive to transient behavior. We focus on inferior metrics, as the superior counterparts follow similar reasoning.

Proposition 6.4. *Let* $\{k_i\}$ *be a sequence of real numbers and decompose it as* $k_i = a_i + b_i$*, where* b_i *is the* transient component*, i.e., it satisfies* $\lim_{i\to\infty} b_i = 0$*. Then,*

(i) $\liminf_{i\to\infty} k_i = \liminf_{i\to\infty} a_i$,

(ii) $\liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} k_i = \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} a_i$.

Proof. It is a property of liminf that $\liminf_{i\to\infty}(a_i + b_i) = \liminf_{i\to\infty} a_i + \liminf_{i\to\infty} b_i$ if either $\{a_i\}$ or $\{b_i\}$ converge. Thus, result (i) trivially holds. For item (ii), we only need to prove that the sequence $\{\frac{1}{n+1}\sum_{i=0}^{n} b_i\}$ converges and is equal to zero. For this, we apply the Stolz–Cesàro theorem:

$$\liminf_{n\to\infty} b_i = 0 \le \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} b_i \le \limsup_{n\to\infty} b_i = 0,$$

which concludes the proof. □

Corollary 6.1. *Let* $\{k_i\}$ *be* ultimately periodic*, i.e.,* $k_i = a_i + b_i$*,* $\lim_{i\to\infty} b_i = 0$ *and* $a_{i+M} = a_i$ *for some* $M \in \mathbb{N}_+$ *and all* i*. Then,*

(i) $\liminf_{i\to\infty} k_i = \min_{i<M} a_i$,

(ii) $\liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} k_i = \frac{1}{M}\sum_{i=0}^{M-1} a_i$.

Proposition 6.4 implies that computing limit metrics of ETC is fundamentally a problem of finding its limit behaviors, ignoring transients. In particular, given Corollary 6.1, if a sequence of inter-event times y_i converges to a periodic pattern, then the limit metrics are solely functions of the periodic component. This motivates us to study fixed and periodic solutions of (2.8); for example, if some y is a recurring pattern of (2.8), then there must be a subset of $\mathcal{Q}_y$ that is invariant. This is done in Section 6.4.3. Before that, we investigate some examples to understand what are the possible limit behaviors exhibited by ETC.

6.4.2 An illustrative example

Consider system (2.2)–(2.4), i.e.,

$$\begin{aligned}
\dot{\boldsymbol{\xi}}(t) &= \boldsymbol{A}\boldsymbol{\xi}(t) + \boldsymbol{B}\boldsymbol{K}\hat{\boldsymbol{\xi}}(t),\\
t_{i+1} &= \inf\{t \in \mathcal{T} \mid t > t_i \text{ and } c(t - t_i, \boldsymbol{\xi}(t), \hat{\boldsymbol{\xi}}(t))\},\\
c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) &\coloneqq \begin{bmatrix}\boldsymbol{x}\\ \hat{\boldsymbol{x}}\end{bmatrix}^{\mathsf{T}} \boldsymbol{Q}(s) \begin{bmatrix}\boldsymbol{x}\\ \hat{\boldsymbol{x}}\end{bmatrix} > 0 \text{ or } \; s \le \bar{\tau},
\end{aligned}$$

with state-space dimension $n_{\mathrm{x}} = 2$. In this case, an o-line of the corresponding sample map, Eq. (2.8), is uniquely defined by the angle $\theta \coloneqq \mathrm{angle}(\boldsymbol{x}) \coloneqq \arctan x_1/x_2 \in [-\pi/2, \pi/2)$. Using the coordinate θ and identifying points along an o-line (that is, regarding any point along an o-line as the same), the sample system (2.8) becomes

$$\begin{aligned}
\theta_{i+1} &= \tilde{f}(\theta_i) \coloneqq \mathrm{angle}\left(f(\begin{bmatrix}\sin\theta & \cos\theta\end{bmatrix}^{\mathsf{T}})\right),\\
y_i &= \tilde{\tau}(\theta) \coloneqq \tau\big(\begin{bmatrix}\sin\theta & \cos\theta\end{bmatrix}^{\mathsf{T}}\big).
\end{aligned} \tag{6.1}$$

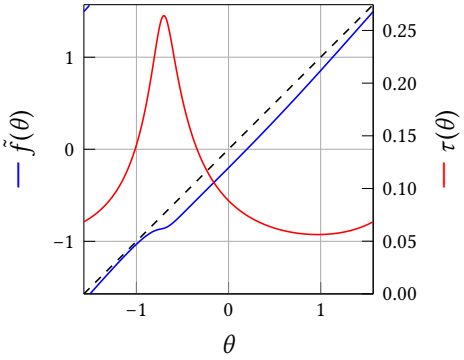


Figure 6.1: Map $\tilde{f}$ and inter-event time τ for case 1 of Example 6.1.

The map $\tilde{f}$ can be seen as a map on the unit circle. An analysis of system (6.1) has been conducted in [26], aiming at finding fixed points or the absence of them. In the cases studied in [26], when there was a fixed point, there was always a stable fixed point. In the next example we show that this is not always true, and investigate the many possible behaviors that ETC traffic exhibits.

6

Example 6.1. *Consider system* (2.2)–(2.4) *with*

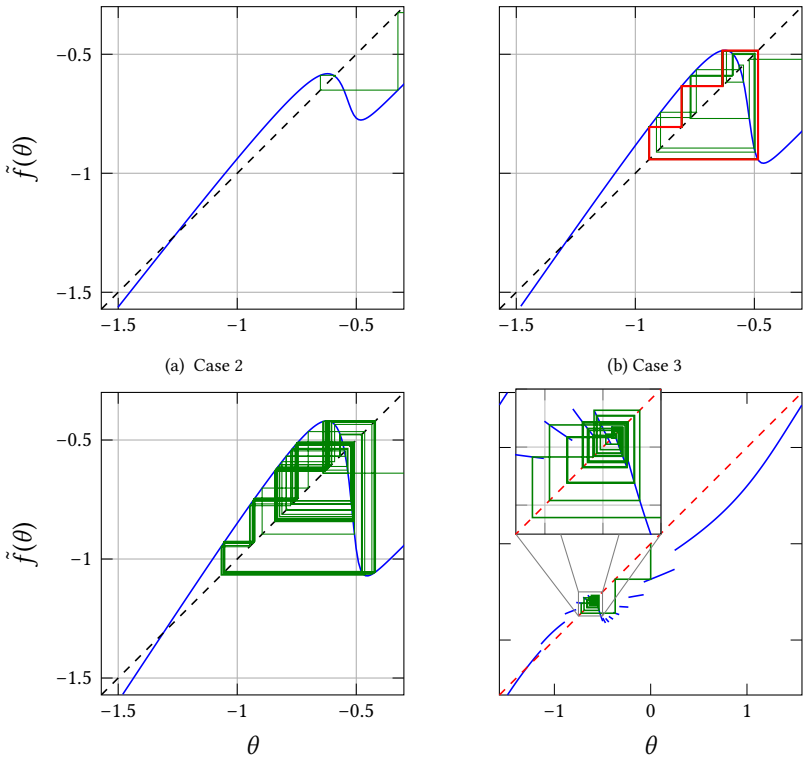
$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) = |\boldsymbol{x} - \hat{\boldsymbol{x}}| > a|\boldsymbol{x}|, \tag{6.2}$$

where $a \in (0,1)$ *is the triggering parameter. This is the seminal triggering condition of [5], which can be put in the form* (2.4) *with sufficiently large* $\bar{\tau}$*. The graphs of* $\tilde{f}$ *and* $\tau(\cdot)$*, for CETC* ($\mathcal{T} = \mathbb{R}_+$) *are given for four cases:*

1) $\boldsymbol{K} = \begin{bmatrix} 0 & -5 \end{bmatrix}$, $a = 0.2$*: Fig. 6.1. The map* $\tilde{f}$ *is invertible, orientation-preserving*[4]*, and has no fixed points.*
2) $\boldsymbol{K} = \begin{bmatrix} 0 & -6 \end{bmatrix}$, $a = 0.32$*: Fig. 6.2a. The map* $\tilde{f}$ *is no longer invertible. It has one unstable fixed point near* $\theta = -1.3$ *and one stable fixed point near* $\theta = -0.6$*.*
3) $\boldsymbol{K} = \begin{bmatrix} 0 & -6 \end{bmatrix}$, $a = 0.5$*: Fig. 6.2b. The map* $\tilde{f}$ *has two unstable fixed points, but a stable period-4 solution as indicated by the cobweb diagram.*
4) $\boldsymbol{K} = \begin{bmatrix} 0 & -6 \end{bmatrix}$, $a = 0.6$*: Fig. 6.2c. The map* $\tilde{f}$ *has no stable fixed points or orbits, and exhibits chaotic behavior. By inspection of the graph, the system has as a minimal set*[5] *the interval* $[-1.07, -0.42]$*, which contains the maximum inter-sample time* $\bar{\tau} \cong 0.76$*, so SupLimSup = Sup* $\cong$ *0.76.*

Finally, notice that all these maps are differentiable, but this is not always the case, as has been observed in [26]. In particular, it is almost never the case for PETC ($\mathcal{T} = h\mathbb{N}$)*. One example is shown in Fig. 6.2d, for* $\boldsymbol{K} = \begin{bmatrix} 0 & -6 \end{bmatrix}$, $a = 0.32$ *(like case 2) and* $h = 0.05$*. Different from the CETC case, its fixed points are unstable and it exhibits chaos.*

[4] A map $f : \mathcal{X} \to \mathcal{X}$ is said to be *orientation-preserving* if its Jacobian J_f satisfies $\det(J_f(\boldsymbol{x})) > 0$ for all $\boldsymbol{x} \in \mathcal{X}$.
[5] A minimal set is an invariant set which contains no proper subsets that are also invariant.



(a) Case 2

(b) Case 3

(c) Case 4

(d) Case 2, PETC ($h = 0.05$) implementation.

Figure 6.2: Maps $\tilde{f}$ for Example 6.1, along with cobweb diagrams of solutions of (6.1) starting from $\theta_0 = 0$. A stable orbit for Case 3 is highlighted in red.

Remark 6.2. *Invertible orientation-preserving maps on the circle have been extensively studied in the field of dynamical systems [74], and they have an attribute called* rotation number. *When the rotation number is rational p/q, p and q coprime, all solutions converge to a periodic orbit of period q. When it is irrational, all solutions are quasi-periodic: oscillatory, but the same point is never visited twice. In the latter case, if $\tilde{f}$ is twice continuously differentiable, it is topologically conjugate to an irrational rotation $g(\theta) = (\theta + r\pi \mod \pi) - \pi/2$, which is ergodic and its orbit is dense in $[-\pi/2, \pi/2)$. Hence, InfLimInf = Inf, and InfLimAvg = SupLimAvg can be obtained to arbitrary precision through simulations from any initial condition.*

6.4.3 Invariant isosequential sets in ETC

Example 6.1 illustrates the complex behavior that can emerge in ETC traffic. Nonetheless it becomes apparent that obtaining fixed or periodic patterns is a fundamental step in the traffic characterization. The first thing we want is a computational or analytical method to determine fixed and periodic patterns. Then, we want to characterize their local stability.

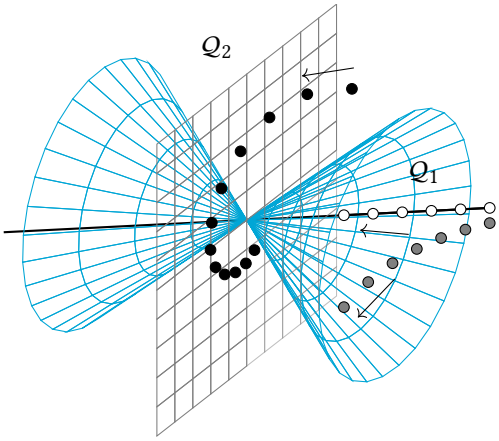


Figure 6.3: Illustration of Theorem 5.3 in $\mathbb{R}^3$. The blue cone splits $\mathbb{R}^3$ into $\mathcal{Q}_1$ and $\mathcal{Q}_2$ the line is an invariant of $\boldsymbol{M}(1)$ and the plane is an invariant of $\boldsymbol{M}(2)$. Points indicate distinct sample trajectories $\{\boldsymbol{x}_i\}$, with the arrows indicating progress of time.

In Theorem 5.3, we have seen that periodic patterns can be characterized by linear invariants. According to this Theorem, ETC exhibits a periodic sampling pattern whenever a linear invariant set $\mathcal{A}$ of the corresponding linear system is contained in the associated isosequential subset; in fact, the set $\mathcal{A} \setminus \{\mathbf{0}\}$ is a periodic set (with period m) of f. To better illustrate Theorem 5.3 and how we can further investigate these linear invariants to characterize their stability, consider a modified version of Fig. 5.1 for a PETC system with $n_{\mathrm{x}} = 3$ and $\bar{k} = 2$ in Fig. 6.3. The same behaviors as therein are present: because an invariant of $\boldsymbol{M}(1)$ is a subset of $\mathcal{Q}_1$, we know that 1^ω is a sampling pattern exhibited by the system; likewise with $\boldsymbol{M}(2)$. The corollary given below (see the proof in the Appendix) states that in general this invariant is an o-line (or *o-plane*, a plane through the origin), and we have an if-and-only-if condition.

Corollary 6.2. *Given the premises of Theorem 5.3, assume (i) $\boldsymbol{M}_\sigma$ is nonsingular, mixed, and of irrational rotations, and that (ii) for every linear invariant $\mathcal{A}$ of $\boldsymbol{M}_\sigma$, $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma) \implies \mathcal{A} \setminus \{\mathbf{0}\} \subseteq \mathcal{Q}_\sigma$. Then σ^ω is a possible output sequence of system* (2.8) *if and only if there exists an o-line or o-plane $\mathcal{A}$ invariant of $\boldsymbol{M}_\sigma$ such that $\mathcal{A} \setminus \{\mathbf{0}\} \subseteq \mathcal{Q}_\sigma$.*

Condition (ii) is satisfied in the illustrative example of Fig. 6.3, as the invariants lie in the interior of the corresponding isochronous sets; this prevents the pathological cases described in the previous chapter (Fig. 5.1, right). In addition, the conditions in this corollary simplify the analysis to o-lines and o-planes, giving that higher dimensional subspaces are not needed for investigation.

Remark 6.3. *Using Corollary 6.2, one can find fixed inter-sample patterns $(t)^\omega$ by searching over $t \in [\underline{\tau}, \bar{\tau}]$ for an $\boldsymbol{M}(t)$ with a linear subspace belonging to $\mathcal{Q}_t$, which can be checked using Prop. 5.5. This search is one-dimensional, in contrast to the search for invariants of system* (2.8) *over $\mathbb{R}^{n_{\mathrm{x}}}$.*

The following lemma is useful when dealing with fixed o-lines.

Lemma 6.1. *Let $\boldsymbol{l}$ be a fixed o-line of f in system* (2.8)*, i.e., $\boldsymbol{x} \in \boldsymbol{l} \implies f(\boldsymbol{x}) \in \boldsymbol{l}$. Then, there exists a real λ such that $f(\boldsymbol{x}) = \lambda \boldsymbol{x}$ for all $\boldsymbol{x} \in \boldsymbol{l}$.*

Proof. By Prop. 6.1, every $\boldsymbol{x} \in \boldsymbol{l}$ shares the same inter-sample time τ. Then, $f(\boldsymbol{x}) = \boldsymbol{M}(\tau)\boldsymbol{x} = a(\boldsymbol{x})\boldsymbol{x}$ since $f(\boldsymbol{x}) \in \boldsymbol{l}$. Hence, by definition of eigenvalues, $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{M}(\tau)$ and $a(\boldsymbol{x}) = \lambda$ is the corresponding eigenvalue. □

For some classic triggering conditions, we can get some interesting specialized results:

Proposition 6.5. *Consider system* (2.2)–(2.4) *with $c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) \equiv |\boldsymbol{x} - \hat{\boldsymbol{x}}| > a|\boldsymbol{x}|$, $\mathcal{T} = \mathbb{R}_+$, and assume $0 < a < 1$ is designed rendering the closed-loop system GES. A fixed o-line with inter-sample time τ exists iff $1/(1+a) \in \lambda(\boldsymbol{M}(\tau))$.*

Proof. By Lemma 6.1, the points in the fixed o-line satisfy $\hat{\boldsymbol{x}}(t_{i+1}) = \lambda \hat{\boldsymbol{x}}(t_i)$, where λ is a real eigenvalue of $\boldsymbol{M}(\tau)$. From the triggering condition, it then holds that $|\lambda \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}| = a|\lambda \hat{\boldsymbol{x}}|$. Hence, $|\lambda - 1| = a|\lambda| \therefore \lambda = 1/(1 \pm a)$. Since $|\lambda| < 1$ for GES, $\lambda = 1/(1+a) < 1$. □

Proposition 6.6. *Consider system* (2.2)–(2.4) *with $c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) \equiv \boldsymbol{x}^\mathsf{T}\boldsymbol{P}\boldsymbol{x} > \mathrm{e}^{-2\rho s}\hat{\boldsymbol{x}}^\mathsf{T}\boldsymbol{P}\hat{\boldsymbol{x}}$, $\mathcal{T} = \mathbb{R}_+$, with $0 < \rho < 1$ and $\boldsymbol{P} > \boldsymbol{0}$.[6] A fixed o-line with inter-sample time τ exists iff either $\mathrm{e}^{-\rho\tau}$ or $-\mathrm{e}^{-\rho\tau}$ is an eigenvalue of $\boldsymbol{M}(\tau)$.*

Proof. Using the same arguments as in Prop. 6.5, we have that $\hat{\boldsymbol{x}}(t_{i+1}) = a\hat{\boldsymbol{x}}(t_i)$. Let $\boldsymbol{z} \coloneqq \sqrt{\boldsymbol{P}}\hat{\boldsymbol{x}}$. An invariant o-line then satisfies $a^2\boldsymbol{z}^\mathsf{T}\boldsymbol{z} = \mathrm{e}^{-2\rho\tau}\boldsymbol{z}^\mathsf{T}\boldsymbol{z} \therefore a = \pm\mathrm{e}^{-\rho\tau}$. Now, $\sqrt{\boldsymbol{P}}^{-1}\boldsymbol{M}(\tau)\sqrt{\boldsymbol{P}}\hat{\boldsymbol{x}} = \sqrt{\boldsymbol{P}}^{-1}\boldsymbol{M}(\tau)\boldsymbol{z} = a\sqrt{\boldsymbol{P}}^{-1}\boldsymbol{z} = a\hat{\boldsymbol{x}}$. Since $\boldsymbol{M}(\tau)$ is similar to $\sqrt{\boldsymbol{P}}^{-1}\boldsymbol{M}(\tau)\sqrt{\boldsymbol{P}}$, $a \in \lambda(\boldsymbol{M}(\tau))$. □

A more general result can be obtained by invoking a result from topology (see the proof in the Appendix) to conclude about which cases a fixed o-line certainly exists, only by knowing the state-space dimension n_x;

Theorem 6.2. *Consider the system* (2.8) *and assume f is continuous and $f(\boldsymbol{x}) \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{0}$. If n_x is odd, then f has a fixed o-line.*

Apart from o-lines, it is also interesting to know when can o-planes be fixed. A PETC example where this happens is illustrated in Fig. 6.3. The next result presents for which dimensions this can generally hold (the proof is also in the Appendix).

Theorem 6.3. *System* (2.8) *can only exhibit a fixed o-plane $\mathcal{P}$ that is isochronous (i.e., $\forall \boldsymbol{x} \in \mathcal{P} \setminus \{\boldsymbol{0}\}, \tau(\boldsymbol{x}) = y$ for some y) if $\boldsymbol{N}(y)$ is singular or one of the following hold.*

(i) $n_\mathrm{x} = 2$ *and* $\underline{\tau} = \bar{\tau}$ *(periodic sampling, trivial);*
(ii) $n_\mathrm{x} = 3$ *and* $\mathcal{T} = h\mathbb{N}$ *(PETC);*
(iii) $n_\mathrm{x} \geq 4$.

After having determined the fixed (or periodic) o-lines and o-planes of system (2.8), the next step is to characterize their (local) attractivity. We say that an o-line $\boldsymbol{l} \subset \mathbb{R}^{n_\mathrm{x}}$ is attractive if for any other o-line $\boldsymbol{l}'$ close enough to $\boldsymbol{l}$, $\lim_{n\to\infty} f^n(\boldsymbol{l}') = \boldsymbol{l}$. The following can be applied for fixed o-lines (see proof in the Appendix.)

[6] This is the triggering condition initially used for STC in [14].

Proposition 6.7. *Let $\boldsymbol{l} \coloneqq \{a\boldsymbol{x} \mid a \in \mathbb{R} \setminus \{0\}\}$ be a fixed o-line of system* (2.8)*, and suppose f is differentiable at $\boldsymbol{x}$, with $J_f(\boldsymbol{x})$ being the corresponding Jacobian matrix. Take λ as the real s.t. $f(\boldsymbol{x}) = \lambda\boldsymbol{x}$ (Lemma 6.1), and let $\boldsymbol{O_x}$ be an orthonormal basis for the orthogonal complement of $\boldsymbol{x}$. Then, if $\frac{1}{\lambda}\boldsymbol{O_x}^\mathsf{T} J_f(\boldsymbol{x})\boldsymbol{O_x}$ is Schur, then $\boldsymbol{l}$ is locally attractive.*

The Jacobian matrix can be expressed as $J_f = \partial(\boldsymbol{M}(\tau(\boldsymbol{x}))\boldsymbol{x})/\partial\boldsymbol{x} = \partial(\boldsymbol{M}(\tau(\boldsymbol{x}))/\partial\boldsymbol{x})\boldsymbol{x} + \boldsymbol{M}(\tau(\boldsymbol{x}))) =$

$$\frac{-2}{\boldsymbol{x}^\mathsf{T}\dot{\boldsymbol{N}}(\tau(\boldsymbol{x}))\boldsymbol{x}}\dot{\boldsymbol{M}}(\tau(\boldsymbol{x}))\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(\tau(\boldsymbol{x})) + \boldsymbol{M}(\tau(\boldsymbol{x})). \tag{6.3}$$

The matrix $\boldsymbol{O_x}^\mathsf{T} J_f(\boldsymbol{x})\boldsymbol{O_x}$ is the Jacobian of f w.r.t. the non-radial directions and projected onto those. It is easy to see that the eigenvalues of $\boldsymbol{O_x}^\mathsf{T} J_f(\boldsymbol{x})\boldsymbol{O_x}$ are the same as those of J_f except the one associated with the eigenvector $\boldsymbol{x}$, while λ is precisely the eigenvalue associated with $\boldsymbol{x}$; hence Prop. 6.7 gives a condition on the ratio between the largest-in-magnitude eigenvalue of J_f and that of the fixed o-line in consideration. For fixed planes, this analysis may require more sophisticated analyses of orbital stability, such as Poincaré return maps.

As we see next, the case of PETC is revealing thanks to the fact that $\boldsymbol{M}$ is constant by parts and, thus, $J_f = \boldsymbol{M}(\tau(\boldsymbol{x}))$ almost everywhere. Because PETC exhibits a discrete set of outputs, a proper definition of stability of an infinite sequence is necessary.

Definition 6.6. *Consider system* (2.8) *with $\mathcal{T} = h\mathbb{N}$ (PETC). An infinite sequence of outputs $\{y_i\}$ is said to be stable if there exists $\boldsymbol{x} \in \mathbb{R}^{n_x}$ with a neighborhood $\mathcal{U}$ such that every $\boldsymbol{x}' \in \mathcal{U}$ satisfies $y_i(\boldsymbol{x}') = y_i(\boldsymbol{x}) = y_i, \forall i \in \mathbb{N}$.*

Proposition 6.8. *Consider system* (2.8) *with $\mathcal{T} = h\mathbb{N}$ (PETC) and assume it is GES. Let $\{y_i\}$ be a p-periodic output trajectory associated with it, and let $\boldsymbol{M} \coloneqq \boldsymbol{M}(y_{p-1})\cdots\boldsymbol{M}(y_1)\boldsymbol{M}(y_0)$. If $\{y_i\}$ is stable, then $\boldsymbol{M}$ is Schur.*

Proof. Every trajectory $\{\boldsymbol{x}_i\}$ of (2.8) that generates $\{y_i\}$ satisfies $\boldsymbol{x}_{i+p} = \boldsymbol{M}\boldsymbol{x}_i$. If $\boldsymbol{M}$ is not Schur, then from almost every $\boldsymbol{x}_0$, (and hence for any point's neighborhood) there are no $M > 0, 0 < a < 1$ such that $|\boldsymbol{x}_{mp}| \leq Ma^m|\boldsymbol{x}_0|$ which implies that the PETC system is not GES. This is a contradiction. □

Proposition 6.8 implies that stable fixed or periodic sampling patterns generated by a PETC system can be used in a multi-rate periodically sampled system, which will also render the origin GES. Note that the existence of such a stable periodic sampling pattern does not imply that the PETC generates that pattern everywhere; as a matter of fact, it may generate sequences that converge to this stable sequence. In these cases, the PETC has a rival periodic sampling schedule which also achieves GES. [7] This is not necessarily true if no stable periodic pattern is exhibited, i.e., when PETC exhibits chaotic or aperiodic traffic.

Remark 6.4. *Proposition 6.8 and its associated conclusion are not true for CETC. For example, consider the case 2 from Ex. 6.1: its stable fixed point occurs for the inter-event time $y \approx 0.3903$; the eigenvalues of $\boldsymbol{M}(y)$ are 0.757 (which is $1/(1+a)$ as expected from Prop. 6.5) and*

[7] While both approaches stabilize the system with equal limit average sampling performances, their transients should be different. It remains to be investigated if their asymptotic performance properties, i.e., GES decay rates, are the same.

-1.33, *hence* $\boldsymbol{M}(y)$ *is not Schur. Given Prop. 6.8, it is now not surprising that case 2's PETC implementation (Fig. 6.2d) does not exhibit an asymptotically stable inter-event time trajectory. More interestingly, this stays true regardless of how small* h *is.*

This Section has presented many properties of fixed and periodic subsets of ETC, such as dimensional conditions for fixed o-lines and o-planes to exist, how to find them, and how to characterize their attractivity. However, it has not yet provided a means to compute the limit metrics or their robust versions. Looking again at Example 6.1, it is clear that several challenges remain:

1) If a stable fixed or periodic pattern is found, can we ensure that it is almost globally attractive? (Here, almost is used to exclude the finitely many unstable fixed or periodic patterns, in case these exist.)
2) If f has fixed or periodic patterns, how can we obtain some information about the limit metrics?
3) If multiple fixed or periodic patterns are found, but inside a chaotic invariant set, how to compute robust limit metrics?

The next Section provides (partial) answers to these questions for PETC using a symbolic approach.

6.5 Quantitative analysis: a symbolic approach

This Section expands the results of Chapter 5 by considering a distinction between robust and fragile limit behaviors. Here we use the same traffic modeling framework as done there: hence, consider a PETC system (2.2)–(2.4) with $\mathcal{T} = h\mathbb{N}$, and denote hereafter its traffic model from Def. 2.15 by $\mathcal{S}$. Let $\mathcal{S}_l$ be its l-complete abstraction (Def. 5.3). Let us see how we can use the insights from the previous section to understand how can we further detect if a cycle satisfying Theorem 5.3 is robust or not by inspecting in Fig. 6.4 the 1- and 2-complete abstractions of the illustrative example in Fig. 6.3.

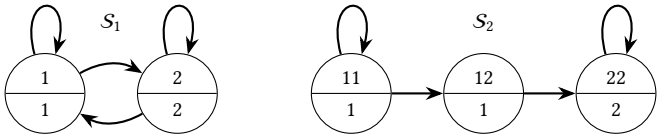


Figure 6.4: l-complete models of the illustrative PETC system of Fig. 6.3, for $l = 1$ (left) and $l = 2$ (right). Each node represents a state, with the top label being the state label and the bottom being its output.

By observing Fig. 6.4, note that $(1,2) \in \mathcal{X}_2$ means that there are points in $\mathbb{R}^3$ that belong to $\mathcal{Q}_1$, but the next sample would belong to $\mathcal{Q}_2$; at the same time, $(2,1) \notin \mathcal{X}_2$, which implies that no points leave $\mathcal{Q}_2$ after sampling, i.e., $\mathcal{Q}_2$ is forward-invariant. Hence, it appears that 2^ω is robust, while 1^ω is not. This kind of observation is central when using abstractions to differentiate robust from fragile behaviors.

6.5.1 Robust limit metrics

Fragile cyclic behaviors that minimize the limit average metrics happen in the real examples we have show: revisiting cases 2 and 3 of Example 6.1, we have 1 and 2 unstable fixed

points, respectively. In both cases, the unstable fixed point near $\theta = -1.3$ gives the value of InfLimInf and InfLimAvg; but for all other initial conditions θ_0, trajectories $\{\theta_i\}$ are attracted to the stable fixed point in case 2 and the stable period-4 orbit in case 3. Thus, *robust limit metrics should be oblivious to unstable orbits.* Let us properly define what stable and unstable behaviors are for systems with a finite output set:

Definition 6.7 (Stable behaviors)**.** *Consider a deterministic WTS $\mathcal{S}$ where $\mathcal{X}$ is a metric space and $\mathcal{Y}$ is finite. A periodic behavior $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$ is said to be* stable *if there exists $x \in \mathcal{X}$ with a neighborhood $\mathcal{U}$ such that every $x' \in \mathcal{U}$ satisfies $\mathcal{B}^\omega_x = \mathcal{B}^\omega_{x'} = \sigma^\omega$ (as in Def. 6.6).*

Definition 6.8 (Robust limit metrics)**.** *Let $\mathcal{S}$ be a simple WTS, $\mathcal{B}^\omega_{\mathrm{u}}(\mathcal{S})$ be the set of its unstable behaviors, and V be a system limit metric (*ILA *or* ILI*). Then the robust version of the metric is* $\mathrm{Rob}V(\mathcal{S}) \coloneqq V(\mathcal{B}^\omega(\mathcal{S}) \setminus \mathcal{B}^\omega_{\mathrm{u}}(\mathcal{S}))$.

Removing unstable behaviors as the ones discussed above is safe in that small perturbations in the initial state lead to distant behaviors. However, consider case 4 of Example 6.1 and its chaotic invariant set: it has infinitely many unstable orbits, and almost every orbit comes arbitrarily close to those orbits. In fact, due to transitivity, *every initial solution starting on the chaotic invariant set will come arbitrarily close to any unstable orbit within it.* Thus, the infimum of a set of metrics on behaviors on a chaotic set, even when excluding the unstable ones, can be equal to one of its unstable behaviors. This deserves a further distinction between unstable behaviors.

Definition 6.9 (Absolutely unstable behaviors)**.** *Consider a deterministic WTS $\mathcal{S}$ where $\mathcal{X}$ is a metric space and $\mathcal{Y}$ is finite. A periodic behavior $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$ is said to be* absolutely unstable *(a.u.) if it is unstable and for almost all x there exists $L \in \mathbb{N}$ such that $\forall l > L$, σ^l is not a subsequence of $\mathcal{B}^\omega_x(\mathcal{S})$. The set of a.u. behaviors of $\mathcal{S}$ is denoted by $\mathcal{B}^\omega_{\mathrm{au}}(\mathcal{S})$.*

A.u. behaviors are fragile in the sense that small perturbations to initial states lead to substantially different behaviors.

Periodic behaviors of a PETC system $\mathcal{S}$ that occur in an abstraction $\mathcal{S}_l$ can be verified to be (absolutely) unstable (see proof in the Appendix).

Proposition 6.9. *Consider a PETC traffic model $\mathcal{S}$ and let $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$. Assume $\boldsymbol{M}_k$ is nonsingular for all $k \in \{1, ..., \bar{k}\}$. Further, assume $\boldsymbol{M}_\sigma$ is mixed, and let $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ be the unitary eigenvectors of $\boldsymbol{M}$ ordered from largest-in-magnitude corresponding eigenvalue to smallest. Denote by $\mathcal{A}$ any linear invariant of $\boldsymbol{M}_\sigma$ containing $\boldsymbol{v}_1$. (I) If $\mathcal{A} \not\subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$, then σ^ω is an unstable behavior. (II) If additionally the cycle $x_1 x_2 ... x_c$ in $\mathcal{S}_l$ that generates σ^ω (i.e., $B_{\mathcal{S}_l}(\{x_1 x_2 ... x_c\}^\omega) = \sigma^\omega$) is the only cycle of its SCC, then σ^ω is absolutely unstable in $\mathcal{S}$.*

Hereafter we shall denote a linear invariant $\mathcal{A}$ containing $\boldsymbol{v}_1$ as in Prop. 6.9 a *dominant* linear invariant, after the concept of dominant modes in linear systems. Referring again to Fig. 6.3 and the corresponding 2-complete model (Fig. 6.4), we see two periodic behaviors, 1^ω and 2^ω. The illustrated o-line is an invariant of $\boldsymbol{M}(1)$ that is not dominant (as can be inferred by the trajectory of gray points that diverge from the line); moreover, the cycle of $\mathcal{S}_2$ that generates 1^ω is a simple cycle, the node 11 with a self loop. This implies that 1^ω is absolutely unstable. Note that this conclusion could not be obtained by inspecting $\mathcal{S}_1$, which is a complete graph without simple cycles. The behavior 2^ω, on the other hand, is stable.

Clearly, removing only a.u. behaviors is safe to give a lower bound estimate to $\mathrm{Rob}V(\mathcal{S})$, i.e., $V(\mathcal{B}^\omega(\mathcal{S})\setminus\mathcal{B}^\omega_{\mathrm{au}}(\mathcal{S})) \le V(\mathcal{B}^\omega(\mathcal{S})\setminus\mathcal{B}^\omega_{\mathrm{u}}(\mathcal{S}))$. An equality holds when $\mathcal{S}$ is not chaotic, since all unstable behaviors are also absolutely unstable. Therefore, determining when $\mathcal{S}$ is or is not chaotic is critical to compute the exact value of $\mathrm{Rob}V(\mathcal{S})$. As we see next, chaos on $\mathcal{S}$ can be estimated from the abstraction $\mathcal{S}_l$.

6.5.2 Estimating chaos in abstractions

In this section we show how to detect (and quantify) chaos on a PETC traffic model $\mathcal{S}$, and when one can conclude that $\mathcal{S}$ is not chaotic. A commonly used measure of chaos is the topological entropy $h(\mathcal{S})$ [76], satisfying $h(\mathcal{S}) \ge 0$, with $h(\mathcal{S}) = 0$ implying there is no chaos. However, instead of a topological measure, we are interested in a measure of chaos of the output of the system: if the state is behaving chaotically but this is not reflected in the output, it does not interfere in the metrics we are interested. Therefore, we shall introduce here a notion called *behavioral entropy*, which is a natural extension of the original concept.

Definition 6.10 (Behavioral entropy)**.** *Consider a system $\mathcal{S}$ and equip $\mathcal{Y}$ with a metric d. A set $\mathcal{W} \subset \mathcal{B}^\omega(\mathcal{S})$ is called (n,ϵ)-separated if for all behaviors $\mathbf{y}, \mathbf{y}' \in \mathcal{W}$, where $\mathbf{y} = y_0 y_1 ... y_i ...$ and $\mathbf{y}' = y'_0 y'_1 ... y'_i ...$, we have $d(y_i, y'_i) > \epsilon$ for all $i \le n$. Let $s(n,\epsilon,\mathcal{S})$ be the maximum cardinality of any (n,ϵ)-separated set. The behavioral entropy is the quantity*

$$h(\mathcal{S}) := \lim_{\epsilon\to 0} \limsup_{n\to\infty} \frac{\log(s(n,\epsilon,\mathcal{S}))}{n}. \tag{6.4}$$

In particular, if $|\mathcal{Y}| < \infty$ and the distance metric is $d(y,y') = 0$ if $y = y'$ and $d(y,y') = 1$ otherwise, we can ignore the ϵ component, and it turns out that

$$h(\mathcal{S}) = \limsup_{n\to\infty} \frac{\log(N(n,\mathcal{S}))}{n}, \tag{6.5}$$

where $N(n,\mathcal{S})$ is the number of different words of length n over the alphabet $\mathcal{Y}$ that are possible trace segments of $\mathcal{S}$.

A system is called behaviorally chaotic *whenever its behavioral entropy is positive.*

Remark 6.5. *The topological entropy also takes the form in Eq.* (6.5) *for* subshifts of finite type*, an abstraction used for autonomous dynamical systems to study their topological properties (see [76]).*

Definition 6.10 takes a behavioral approach [20] to extend the original definition [76] for systems that are possibly nondeterministic and have output maps. If $H = \mathrm{Id}$ and $\mathrm{Post}(x) = \{f(x)\}$ for some continuous map $f : \mathcal{X} \to \mathcal{X}$, we recover the original notion. It may seem unproductive to extend a measure of chaos to non-deterministic systems, as these should all be chaotic in some sense; however, this is not always the case. For example, consider $\mathcal{S}_2$ of Fig. 6.4: it is easy to see that $N(n,\mathcal{S}_2) = n+1$: 11...11, 11...12, ..., 122...22, 22...22. Hence, $h(\mathcal{S}_2) = \lim_{n\to\infty}(\log(n+1)/n) = 0$, and this system is not (behaviorally) chaotic.

Proposition 6.10. *Consider two transition systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b = \mathcal{Y}$ s.t. $\mathcal{S}_a \preceq_{\mathcal{B}} \mathcal{S}_b$. If $|\mathcal{Y}| < \infty$, then $h(\mathcal{S}_a) \le h(\mathcal{S}_b)$.*

Proof. Trivially, from behavioral inclusion [17], $\forall n \in \mathbb{N}, N(n, \mathcal{S}_a) \leq N(n, \mathcal{S}_b)$. The result follows from monotonicity of the log function. □

The question now is how to compute the behavioral entropy of a finite-state system. This result is known for topological entropy of subshifts of finite type, which is the same as a finite-state transition system with $H = \mathrm{Id}$:

Theorem 6.4 ([76, Theorem IX.1.9])**.** [8] *Let $\mathcal{S}$ be a finite system and $N'(n, \mathcal{S})$ be the number of different n-length words over the alphabet $\mathcal{X}$ generated by $\mathcal{S}$ (note that this reflects the* internal *behavior of $\mathcal{S}$). Then,*

$$\limsup_{n\to\infty} \frac{\log(N'(n, \mathcal{S}))}{n} = \log \lambda_1(\boldsymbol{T}),$$

where $\boldsymbol{T}$ is the incidence matrix of $\mathcal{S}$.

Under a detectability condition of $\mathcal{S}$, the same result holds for behavioral entropy:

Definition 6.11 (Detectability)**.** *A transition system $\mathcal{S}$ is said to be l-detectable if there exists a finite $l \in \mathbb{N}$ such that, for each word $w \in \mathcal{B}^+(\mathcal{S}), |w| \geq l$, there exists a unique $x \in \mathcal{X}$ such that $w \in \mathcal{B}^+_x(\mathcal{S})$.*

Theorem 6.5. *Let $\mathcal{S}$ be an l-detectable finite-state system for some finite $l \in \mathbb{N}$, and let $\boldsymbol{T}$ be its incidence matrix. Then,*

$$h(\mathcal{S}) = \log \lambda_1(\boldsymbol{T}). \tag{6.6}$$

Proof. Let $s := |\mathcal{Y}|$. Because of l-detectability, every $(n+l)$-long external behavior of $\mathcal{S}$ gives a unique n internal behavior, hence $N(n+l, \mathcal{S}) \geq N'(n, \mathcal{S})$. From every external behavior of length n, there can be at most s^l external behaviors of length $n+l$ (simply concatenate every possible word in $\mathcal{Y}^l$ to complete the length). Thus, $s^l N(n, \mathcal{S}) \geq N(n+l, \mathcal{S})$. Finally, since the output map H is single-valued, the number of external behaviors can never be bigger than the number of different internal behaviors: $N(n, \mathcal{S}) \leq N'(n, \mathcal{S})$. Combining these inequalities, the following holds for all $n > l$:

$$N(n, \mathcal{S}) \leq N'(n, \mathcal{S}) \leq s^l N(n, \mathcal{S}).$$

Now,

$$\limsup_{n\to\infty} \frac{\log(s^l N(n, \mathcal{S}))}{n} = \limsup_{n\to\infty} \left(\frac{\log(s^l)}{n} + \frac{\log(N(n, \mathcal{S}))}{n} \right) = \limsup_{n\to\infty} \frac{\log(N(n, \mathcal{S}))}{n}.$$

The sandwich rule and Theorem 6.4 conclude the proof. □

The following results help us apply Theorem 6.5 to the PETC traffic model.

Proposition 6.11. *A non-blocking finite-state l-detectable autonomous transition system $\mathcal{S}$ has zero behavioral entropy if and only if all the strongly connected components (SCCs) of its associated graph are isolated nodes or simple cycles.*

[8] In [76], the internal behavior from an initial state is called *itinerary*. The original Theorem states that this quantity is also the topological entropy of the subshift $\mathcal{S}$, but here we only need the formula relating the limit to the spectral radius of $\boldsymbol{T}$.

Proof. The spectrum of a digraph is the union of the spectra of its SCCs [78]. Because $\mathcal{S}$ is non-blocking, it must have at least one cycle. The adjacency matrix of an isolated node is $[0]$, thus its spectrum is $\{0\}$. Further, all vertices of a simple cycle have only one outgoing edge, hence the corresponding SCC has a constant outdegree of 1. From [78, Theorem 2.1], the spectral radius of an SCC is 1 iff it has constant outdegree 1. Hence, the spectral radius of the whole graph is $\max(1,0) = 1$, whose log is 0. □

Remark 6.6. *The l-complete PETC traffic model $\mathcal{S}_l$ is l-detectable because, by definition, each $k_1k_2...k_l \in \mathcal{X}_l$ is the unique state that generates the finite behavior $hk_1, hk_2, ...hk_l$.*

Theorem 6.6. *Consider the PETC system* (2.2)–(2.4) *($\mathcal{T} = h\mathbb{N}$), its traffic model $\mathcal{S}$ and its l-complete traffic model $\mathcal{S}_l$, with $l \in \mathbb{N}$. The following assertions are true:*

i) $h(\mathcal{S}) \leq h(\mathcal{S}_l)$;

ii) If all SCCs of $\mathcal{S}_l$ are simple cycles, then $h(\mathcal{S}) = h(\mathcal{S}_l) = 0$, i.e., $\mathcal{S}$ is not chaotic.

Proof. Assertion (i): Prop. 2.1 gives that $\mathcal{S} \preceq \mathcal{S}_l$; then, from Theorem 2.1, $\mathcal{S} \preceq_{\mathcal{B}} \mathcal{S}_l$; finally, Prop. 6.10 concludes the proof. Assertion (ii): $\mathcal{S}_l$ satisfies the premises of Prop. 6.11. Hence, $h(\mathcal{S}_l) = 0$. Using assertion (i) and the fact that $h(\mathcal{S}) \geq 0$, we conclude that $h(\mathcal{S}) = 0$. □

Revisiting Fig. 6.4, it is easy to see that $h(\mathcal{S}_1) = \log(2) = 1$ bit (base 2), while $h(\mathcal{S}_2) = 0$, which implies that the example of Fig. 6.3 is not chaotic.

6.5.3 Estimating and computing robust metrics

Now we are equipped with the necessary tools to estimate robust limit metrics using an abstraction and determine when they are equal to the concrete system's or simply a lower bound. Based on the discussion in Section 6.5.1, we define the following robust limit metric for the abstraction:

Definition 6.12 (Robust metric for $\mathcal{S}_l$)**.** *Consider a PETC traffic model $\mathcal{S}$ and its l-complete model $\mathcal{S}_l$. Let $\tilde{\mathcal{B}}^{\omega}_{\mathrm{au}}(\mathcal{S}_l)$ be the set of behaviors of $\mathcal{S}_l$ that are are simple cycles in $\mathcal{S}_l$ and are absolutely unstable in $\mathcal{S}$. We define* $\mathrm{Rob}V(\mathcal{S}_l)$ *as $V(\mathcal{B}^{\omega}(\mathcal{S}_l) \setminus \tilde{\mathcal{B}}^{\omega}_{\mathrm{au}}(\mathcal{S}_l))$.*

Theorem 6.7. *Consider a PETC traffic model $\mathcal{S}$ and its l-complete model $\mathcal{S}_l$. Consider $V \in \{\mathrm{ILI}, \mathrm{ILA}\}$; then* $\mathrm{Rob}V(\mathcal{S}'_l) \leq \mathrm{Rob}V(\mathcal{S})$*. Moreover, if all SCCs of $\mathcal{S}_l$ are simple cycles, and the minimizing cycle σ satisfies $\sigma^{\omega} \in \mathcal{B}^{\omega}(\mathcal{S})$, then* $\mathrm{Rob}V(\mathcal{S}_l) = \mathrm{Rob}V(\mathcal{S})$*.*

Proof. Because all behaviors in $\tilde{\mathcal{B}}_{\mathrm{au}}$ are absolutely unstable in $\mathcal{S}$, we have $\tilde{\mathcal{B}}^{\omega}_{\mathrm{au}}(\mathcal{S}_l) \subseteq \mathcal{B}^{\omega}_{\mathrm{au}}(\mathcal{S})$, and thus $\tilde{\mathcal{B}}^{\omega}_{\mathrm{au}}(\mathcal{S}_l) \subseteq \mathcal{B}^{\omega}_{\mathrm{u}}(\mathcal{S})$. From Prop. 2.1, $\mathcal{B}^{\omega}(\mathcal{S}_l) \supseteq \mathcal{B}^{\omega}(\mathcal{S})$; hence $\mathcal{B}^{\omega}(\mathcal{S}_l) \setminus \tilde{\mathcal{B}}^{\omega}_{\mathrm{au}}(\mathcal{S}_l) \supseteq \mathcal{B}^{\omega}(\mathcal{S}) \setminus \mathcal{B}^{\omega}_{\mathrm{u}}(\mathcal{S})$. Now, for any behavior set $\mathcal{B}$, $V(\mathcal{B}) = \inf\{f(y_i) \mid \{y_i\} \in \mathcal{B}\} = \inf\{F(\{y_i\}) \mid \{y_i\} \in \mathcal{B}\}$, where $F(\{y_i\})$ is either $\liminf_{i\to\infty} y_i$ (ILI) or $\liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} y_i$ (ILA). Hence, $\mathcal{B}_a \subseteq \mathcal{B}_b$ implies $V(\mathcal{B}_a) \geq V(\mathcal{B}_b)$, and the inequality $\mathrm{Rob}V(\mathcal{S}) \geq \mathrm{Rob}V(\mathcal{S}_l)$ follows.

For the equality: if $\mathcal{S}_l$ contains only simple cycles, then $\mathcal{S}$ is not behaviorally chaotic (Theorem 6.6), and thus $\mathcal{B}^{\omega}_{\mathrm{u}}(\mathcal{S}) = \mathcal{B}^{\omega}_{\mathrm{au}}(\mathcal{S})$ (all unstable cycles are absolutely unstable). Then, the minimizing cycle σ of $\mathcal{S}_l$ is by exclusion a stable cycle of $\mathcal{S}$. Since $\sigma^{\omega} \in \mathcal{B}^{\omega}(\mathcal{S}) \setminus \mathcal{B}^{\omega}_{\mathrm{u}}(\mathcal{S})$, we have that $\mathrm{Rob}V(\mathcal{S}_l) = F(\sigma^{\omega}) \geq \inf\{F(\{y_i\}) \mid \{y_i\} \in \mathcal{B}^{\omega}(\mathcal{S}) \setminus \mathcal{B}^{\omega}_{\mathrm{u}}(\mathcal{S})\} = \mathrm{Rob}V(\mathcal{S})$. Hence, $\mathrm{Rob}V(\mathcal{S}_l) = \mathrm{Rob}V(\mathcal{S})$. □

Revisiting Figs. 6.3 and 6.4 one last time, we have trivially that $\mathrm{ILI}(\mathcal{S}) = \mathrm{ILA}(\mathcal{S}) = 1$, but using Theorem 6.7 on $\mathcal{S}_2$ we conclude that $\mathrm{RobILI}(\mathcal{S}) = \mathrm{RobILA}(\mathcal{S}) = 2$. Nevertheless, by Prop. 6.8, $\boldsymbol{M}(2)$ must be Schur, and hence a periodic sampling of $2h$ would also stabilize the system with the same traffic performance.

Remark 6.7. *In the case of* RobILI*, if the invariant associated to the minimizing cycle σ can be verified to belong to a chaotic invariant set, under mild assumptions it holds that* $\mathrm{RobILI}(\mathcal{S}_l) = \mathrm{RobILI}(\mathcal{S})$*. To see this, first note that* $\mathrm{RobILI} = \min(\sigma) =: y$*; denoting by $\mathcal{X}_c$ the chaotic invariant set, if $\mathcal{Q}_y \cap \mathcal{X}_c$ has non-empty interior, by the Birkhoff Transitivity Theorem (see Def. 6.1) almost every solution starting in $\mathcal{X}_c$ visits $\mathcal{Q}_y$ infinitely often.*

Remark 6.8. *In case a chaotic invariant set is ergodic, the infinimal limit average is the same almost everywhere (when restricted to the set), i.e., it is independent of the initial condition (as a consequence of Birkhoff Ergodic Theorem). As a matter of fact, almost everywhere means everywhere except the union of periodic orbits. Thus,* $\mathrm{RobILA}(\mathcal{S}_l)$ *can then be a conservative estimate. Nevertheless, the associated* RobILA *can be estimated through simulations. Ergodicity can be statistically tested using the approach of [79], where one tests whether the two initially different distributions on $\mathcal{X}$ converge to an equal one upon the repeated application of the map f by using a non-parametric hypothesis test such as the Kolmogorov–Smirnov (KS) test. Alternatively, the test can be performed on the distributions of outputs; because $\mathcal{Y}$ is discrete, an hypothesis test appropriate for discrete supports, such as the Cramér–von Mises (CvM) test [80]. For this approach to succeed, it is important that the initial distribution contains only points that are in or lead to the chaotic invariant. The abstraction $\mathcal{S}_l$ can be used as an approximate selector of points on the chaotic invariant set, as its SCCs that are not simple cycles are related to over-approximations of potential chaotic invariants on the concrete system $\mathcal{S}$.*

6.6 Numerical examples

We have implemented the methods to compute behaviorial entropy and $\mathrm{RobILA}(\mathcal{S})$ using Theorem 6.7 in `ETCetera`. We now revisit Example 6.1 to compute robust metrics using the abstraction approach devised in Section 6.5.

Example 6.2. *Consider system* (2.2)–(2.4) *with*

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) = |\boldsymbol{x} - \hat{\boldsymbol{x}}| > a|\boldsymbol{x}|, \tag{6.2 revisited}$$

as in Example 6.1. Now we use PETC with $h = 0.05$ and check the following cases:

1) $\boldsymbol{K} = \begin{bmatrix} 0 & -5 \end{bmatrix}$, $a = 0.2$*, as in Ex. 6.1 case 1*
2) $\boldsymbol{K} = \begin{bmatrix} 0 & -6 \end{bmatrix}$, $a = 0.2$.
3) $\boldsymbol{K} = \begin{bmatrix} 0 & -6 \end{bmatrix}$, $a = 0.32$*, as in Ex. 6.1 case 2, and Fig. 6.2d.*

Table 6.1 shows the values of ILA and RobILA for each case, as well as the l value at which the algorithms were terminated (or interrupted) and CPU times. Case 1 shows a periodic sequence σ^ω with $|\sigma| = 27$ that is stable and attains both the ILA and the RobILA,

Table 6.1: ILA values for Example 6.2

Case	1	2	3
l (robust)	16 (16)	10 (10)	1 (10*)
ILA (RobILA)	0.137 (0.137)	0.1 (0.25)	0.1 (0.4)
CPU time (robust) [s]	50 (49)	23 (19)	0.81 (5655)

* Algorithm interrupted before finding a verified cycle.

as well as ILI = RobILI = 0.1. In fact, case 1 exhibits only this cycle, and a bisimulation is found with $l = 27$. Case 2 is different in that an a.u. cycle is attained at $y = 0.1$, but a stable cycle has $y = 0.25$ (stationary). Upon inspection of $\mathcal{S}_l$, there is another stable cycle at $y = 0.3$. Unsurprisingly, we also obtain $\mathrm{ILI}(\mathcal{S}) = 0.1$ and $\mathrm{RobILI}(\mathcal{S}) = 0.25$, which happen at the same cycles. Finally, Case 3 is a chaotic example; the ILA is found at $y = \underline{\tau} = 0.1$ in the first iteration, but RobILA is never confirmed, although a lower bound of 0.4 is obtained, related to two unstable cycles, $(0.4)^\omega$ and $(0.35, 0.45)^\omega$. However, note the CPU time for obtaining the $\mathcal{S}_{10}$ abstraction of approximately 1.5 hour (compare with the others of less than a minute): this is the effect of chaos on the refinements: as indicated by the entropy formula, Eq. (6.4), the number of l-sized sequences grows exponentially with l. In fact, $\mathcal{S}_{10}$ has 9271 states, and an entropy of 1.14 bits. The SCC at which the two cycles belong has 7767 states, a strong indicative of a chaotic invariant set. Figure 6.5 shows the evolution of $h(\mathcal{S}_l)$ as a function of l for the three cases, where it is clear that the entropy seems to stabilize at a high value in Case 3, whereas it descends to zero in the other cases. By applying Remark 6.8, two different initial distributions on states related to the large SCC of $\mathcal{S}_{10}$ where generated with 1000 points each, and after 9 iterations they converged to the same distribution (CvM test, $p = 0.998$), a good indicative that the chaotic invariant set is ergodic. The average of the obtained ensemble, which by Birkhoff Ergodic Theorem is approximately equal to the limit average of any run starting in the invariant, is 0.417, slightly higher than the 0.4 using Theorem 6.7. It is interesting to see that 0.4 is a slightly higher limit average than what was obtained in the CETC implementation (Ex. 6.1 Case 3, and Remark 6.4), of 0.39; more interestingly, $\boldsymbol{M}(0.4)$ is not Schur, which highlights that the PETC has a larger average sampling period than any stabilizing periodic sampling, at the cost of seemingly unpredictable traffic. Finally, while $\mathrm{ILI}(\mathcal{S}) = \mathrm{ILA}(\mathcal{S}) = 0.1$ (at the same unstable cycle 0.1^ω) the best lower bound for RobILI is found to be 0.3, which is witnessed by the unstable cycle $(0.3, 0.45, 0.4, 0.5)^\omega$. By inspection, the associated o-line belongs to the chaotic invariant, thus by Remark 6.7 this is the correct value of $\mathrm{RobILI}(\mathcal{S})$.

Reproducing the results of this chapter. The abstractions for cases 1, 2, and 3, the data in Table 6.1, the bisimulation case, and Figure 6.5 can be obtained by running `examples/chaos_abstractions.py` in `ETCetera`. The sample-based method to detect ergodicity and compute RobILA for case 3 can be reproduced by running `examples/chaos_sample_based.py`. ****ILI*** Finally, Figures 6.1–6.2d can be obtained by running `examples/chaos_examples.py`.

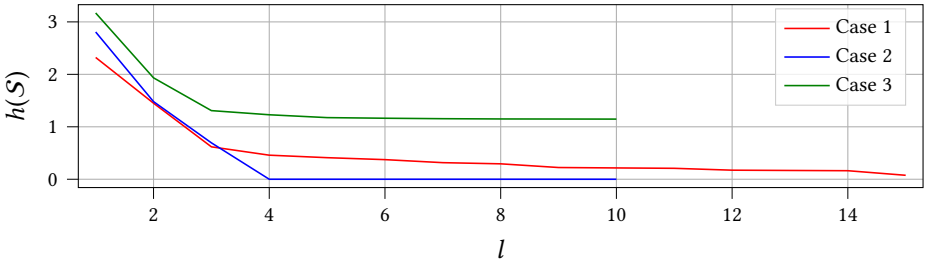


Figure 6.5: Entropy $h(S_l)$ as a function of l for Example 6.2.

6.7 Discussion and Conclusions

Event-triggered control can exhibit very complex traffic patterns, and this seems to be more common the more "aggressive" the triggering mechanism towards sampling reduction. Simple traffic is observed on the opposite case. This is in line with the findings on [25, 73] for $\mathbb{R}^2$, in which for small enough triggering parameters the states behave essentially like linear systems: two asymptotes, the "slowest" one attracting trajectories from the other, or a spiral towards the origin when eigenvalues are complex conjugate. This seems to be the case whenever f of the sample system (2.8), projected onto the projective space, is invertible, which poses the following conjecture:

Conjecture 6.1. *Consider system* (2.2)–(2.4) *with CETC* ($\mathcal{T} = \mathbb{R}_+$) *and its sample system* (2.8). *Let* $h : \mathbb{R}^{n_x} \longrightarrow \mathbb{P}^{n_x-1}$ *be the natural projection onto the projective space. If* $g = h \circ f \circ h^{-1}$ *is continuous and invertible, then it is topologically conjugate with* $g' = h \circ f' \circ h^{-1}$, *where* $f'(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x}$ *for some* $\boldsymbol{M} \in \mathbb{R}^{n_x \times n_x}$.

If proven, this would imply that all periodic o-lines or o-planes of the CETC have period one and can be obtained using Remark 6.3, enabling one to obtain (robust) limit metrics in the continuous case. The ETC sample map would be topologically equivalent to a (discrete-time) linear system, having modes of different speeds, that can be classified according to the eigenvalues of $\boldsymbol{M}$.

A symbolic method for computing robust metrics for PETC was presented in Section 6.5, and while it is an important first step, it suffers from the curse of dimensionality, particularly when chaotic behaviors are present; to address these issues, one could use alternative solvers to Z3 as suggested in Chapter 5, but also different abstractions that help pinpoint the existence of chaotic invariant sets. For the latter, an approach such as in [81] may be interesting, which can be seen in the framework of [5] as finding an abstraction that is *backwards* simulated by the concrete system.

We have also seen an example of CETC whose robust infimal limit average is higher than any stable periodic sampling strategy, whereas the same cannot happen with PETC under some generic assumptions. The first case *is a concrete example where ETC is more sampling-efficient than any periodic implementation,* but at the same time any practical implementation of it must rely on periodic checking of triggering times, thus becoming a PETC, and as such any stable inter-event time sequence it exhibits is stabilizing as a periodic sampling strategy; the only option for PETC to beat the most sampling-efficient

periodic implementation would involve chaotic traffic. This is not a problem per se, and one could speculate that chaotic traffic could help in cyber-security aspects, but it may nonetheless make scheduling of multiple ETC loops in a network even more challenging.

Finally, it is worth noting two important practical observations, one positive and one negative, about the work in this chapter. The positive one is that it is not actually limited to linear systems, using the same arguments as in Chapter 5: limit behavioral properties such as our metrics, including entropy, are preserved for the linearized model around the equilibrium. The negative aspect is that we have considered a very simple case, of state-feedback without disturbances. It is known that doing output-feedback or having disturbances can severely alter the inter-sample behavior of the closed-loop system, in some cases leading to Zeno behavior [82], and practical modifications to the triggering condition are often necessary. It is an open question whether adding these imperfections change our conclusions drastically, or if there are simple adjustments for these cases. Still, it is not difficult to extend the symbolic approach to perturbed systems, following the steps in [30]. Nonetheless, all the conclusions we have obtained for the nominal system bring new insight and understanding about ETC traffic patterns and its sampling performance.

6.A Proofs

6.A.1 Proof of Corollary 6.2

Proof. The if and only if statement is a straightforward combination of items (i) and (ii) of Theorem 5.3 and assumption (ii) of this corollary.

To see that $\mathcal{A}$ is either an o-line (1-dimensional) or an o-plane (2-dimensional), assume that it is higher dimensional. By assumption, $\boldsymbol{M}_\sigma$ is mixed, therefore $\mathcal{A}$ is spanned by o-lines (associated with real eigenvalues) and o-planes (associated with complex conjugate pairs). Let $\boldsymbol{V} \in \mathbb{R}^{n_\mathrm{x} \times m}$ be a basis for $\mathcal{A}$ with $m > 2$, where the i-th column of $\boldsymbol{V}$ is a real eigenvector of $\boldsymbol{M}_\sigma$ or, in case of a complex eigenvector pair $\boldsymbol{v}, \boldsymbol{v}^*$, the i-th and $(i+1)$-th columns are $\boldsymbol{v} + \boldsymbol{v}^*$ and $\mathrm{i}\boldsymbol{v} - \mathrm{i}\boldsymbol{v}^*$ respectively; these two columns correspond to an invariant plane of $\boldsymbol{M}_\sigma$. In the former case, we have

$$\boldsymbol{V}\boldsymbol{E}_i = \boldsymbol{v}, \tag{6.7}$$

and in the latter

$$\boldsymbol{V}\boldsymbol{E}_{i,i+1} = \begin{bmatrix} \boldsymbol{v} + \boldsymbol{v}^* & \mathrm{i}\boldsymbol{v} - \mathrm{i}\boldsymbol{v}^* \end{bmatrix}, \tag{6.8}$$

where $\boldsymbol{E}_i$ is a row matrix with the i-th element being 1 and the rest zero, and $\boldsymbol{E}_{i,i+1} \in \mathbb{R}^{2 \times m}$ has the entries $(1, i)$ and $(2, i+1)$ equal to 1, the rest being zero. These are nothing but selection matrices.

Since $\mathcal{Q}_\sigma$ is composed of an intersection of sets of the form $\{\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}} \mid \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_i\boldsymbol{x} \sim 0\}$, where $\sim \in \{=, \geq, >\}$, by Prop. 5.5, $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}_i\boldsymbol{V} \approx \boldsymbol{0}$ for every such $\boldsymbol{Q}_i$ determining $\mathcal{Q}_\sigma$, where $\approx \in \{=, \geq, >\}$, respectively. Since $\boldsymbol{A} \approx \boldsymbol{0} \implies \boldsymbol{B}^\mathsf{T}\boldsymbol{A}\boldsymbol{B} \approx \boldsymbol{0}$ for any non-singular $\boldsymbol{B}$, we can conclude that $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}_i\boldsymbol{V} \approx \boldsymbol{0}$ implies $(\boldsymbol{V}\boldsymbol{E}_i)^\mathsf{T}\boldsymbol{Q}_i(\boldsymbol{V}\boldsymbol{E}_i) \approx \boldsymbol{0}$ and $(\boldsymbol{V}\boldsymbol{E}_{i,i+1})^\mathsf{T}\boldsymbol{Q}_i(\boldsymbol{V}\boldsymbol{E}_{i,i+1}) \approx \boldsymbol{0}$, which imply that the corresponding o-line or o-plane is also a subset of $\mathcal{Q}_\sigma$. □

6.A.2 Proofs of Theorem 6.2 and Proposition 6.7

In these proofs, if f is not invertible in the pointwise sense, we treat its inverse in a set-based manner: $f^{-1} : \mathcal{Y} \rightrightarrows \mathcal{X}$, $f^{-1}(y) = \{x \in \mathcal{X} \mid f(x) = y\}$. In addition, here we work on the real projective space $\mathbb{P}^{n_\mathrm{x}-1}$, the space of all o-lines in $\mathbb{R}^{n_\mathrm{x}}$. The real projective space is the quotient of $\mathbb{R}^n \setminus \{\boldsymbol{0}\}$ by the relation $\boldsymbol{x} \sim \lambda\boldsymbol{x}, \lambda \in \mathbb{R} \setminus \{0\}$. Therefore, $\boldsymbol{x}$ and $\lambda\boldsymbol{x}$ are *the same point* $\boldsymbol{p} \in \mathbb{P}^{n_\mathrm{x}-1}$. We denote the natural projection of a point in $\mathbb{R}^{n_\mathrm{x}}$ onto $\mathbb{P}^{n_\mathrm{x}-1}$ by $h : \mathbb{R}^{n_\mathrm{x}} \setminus \{\boldsymbol{0}\} \longrightarrow \mathbb{P}^{n_\mathrm{x}-1}$.

Lemma 6.2. *Consider system* (2.8) *and assume $f(\boldsymbol{x}) \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{0}$. Then $g \coloneqq h \circ f \circ h^{-1}$ is a well-defined function. Moreover, if f is continuous, then g is also continuous.*

Proof. For any $\boldsymbol{p} \in \mathbb{P}^{n_\mathrm{x}-1}$, $h^{-1}(\boldsymbol{p})$ gives a whole o-line $\boldsymbol{l} \subset \mathbb{R}^{n_\mathrm{x}}$. From Prop. 6.1, it holds that $f(\boldsymbol{l}) = \boldsymbol{l}'$, where $\boldsymbol{l}'$ is also an o-line. Hence $h(\boldsymbol{l}') = \boldsymbol{p}' \in \mathbb{P}^{n_\mathrm{x}-1}$, so g is well defined. Continuity is then given by the fact that f is also continuous for o-lines, i.e., if $\boldsymbol{l}$ is an o-line, $\lim_{\boldsymbol{l}' \to \boldsymbol{l}} f(\boldsymbol{l}') = \boldsymbol{l}$; hence $\lim_{\boldsymbol{p}' \to \boldsymbol{p}} g(\boldsymbol{p}') = \boldsymbol{p}$. □

Proof of Theorem 6.2. Every continuous map from the real projective space to itself has a fixed point if its dimension is even [83, p. 109]. From Lemma 6.2, $g : \mathbb{P}^{n_\mathrm{x}-1} \longrightarrow \mathbb{P}^{n_\mathrm{x}-1}$ is a well-defined continuous function; thus, g has a fixed point if n_x is *odd*.

Now we need to show that, if g has a fixed point, then f has a fixed o-line. If $\boldsymbol{p}$ is a fixed point of g, then take a point $\boldsymbol{x} \in h^{-1}(\boldsymbol{p})$. Then, $h(f(\boldsymbol{x})) = g(h(\boldsymbol{x})) = \boldsymbol{p} \therefore f(\boldsymbol{x}) \in h^{-1}(\boldsymbol{p})$

Hence, there exists $\boldsymbol{x}' \in h^{-1}(\boldsymbol{p})$ satisfying $\boldsymbol{x}' = f(\boldsymbol{x})$, where $\boldsymbol{x}' = \lambda \boldsymbol{x}$, for some λ. Since f is homogeneous as per Prop. 6.1, $\boldsymbol{x}' = f(\boldsymbol{x})$ is true for any $\boldsymbol{x}$ in the o-line containing it. Hence, this line is fixed by f, and the proof is complete. □

Proof of Prop. 6.7. From Lemma 6.2, $g : \mathbb{P}^{n_\mathrm{x}-1} \longrightarrow \mathbb{P}^{n_\mathrm{x}-1}$ is well defined. Let $\boldsymbol{p} = h(\boldsymbol{x})$ for any $\boldsymbol{x} \in \boldsymbol{l}$. We want to show that there is a coordinate system for the tangent space of g at $\boldsymbol{p}$ such that the Jacobian of g at $\boldsymbol{p}$ is equal to $\frac{1}{\lambda}\boldsymbol{O_x}^\mathsf{T} J_f(\boldsymbol{x})\boldsymbol{O_x}$.

First, note that the real projective space is locally equal to the unit sphere, hence we can use the orthogonal subspace to a unitary $\boldsymbol{x}$ within $\boldsymbol{l}$ as the tangent subspace of $\boldsymbol{p}$ embedded in $\mathbb{R}^{n_\mathrm{x}}$. Denote it as $\mathcal{T}(\boldsymbol{x})$. Let $\boldsymbol{d}$ be a unitary vector orthogonal to $\boldsymbol{x}$. Any point in $\mathcal{T}(\boldsymbol{x})$ can be described as $\boldsymbol{x} + a\boldsymbol{d}$. To get the Jacobian of g, we apply f to $\boldsymbol{x} + h\boldsymbol{d}$ and project the result back to $\mathcal{T}(\boldsymbol{x})$: $f(\boldsymbol{x} + h\boldsymbol{d}) = f(\boldsymbol{x}) + hJ_f(\boldsymbol{x})\boldsymbol{d} + \mathcal{O}(h^2) = \lambda\boldsymbol{x} + hJ_f(\boldsymbol{x})\boldsymbol{d} + \mathcal{O}(h^2)$, whose projection back to $\mathcal{T}(\boldsymbol{x})$ is simply $\boldsymbol{x} + h/\lambda \cdot J_f(\boldsymbol{x})\boldsymbol{d} + \mathcal{O}(h^2)$. Thus, the vector of variation of g w.r.t. $\boldsymbol{d}$ embedded in $\mathbb{R}^{n_\mathrm{x}}$ is

$$\lim_{h \longrightarrow 0} \frac{\boldsymbol{x} + h/\lambda \cdot J_f(\boldsymbol{x})\boldsymbol{d} + \mathcal{O}(h^2) - \boldsymbol{x}}{h} = \frac{1}{\lambda} J_f(\boldsymbol{x})\boldsymbol{d}.$$

Now let $\boldsymbol{d}_i$ be the i-th column of $\boldsymbol{O_x}$. Every $\boldsymbol{d}_i$ is unitary and orthogonal to $\boldsymbol{x}$. Setting $\boldsymbol{d}_1, \boldsymbol{d}_2, ...\boldsymbol{d}_{n_\mathrm{x}-1}$ as a coordinate system for the tangent space of g at $\boldsymbol{p}$, the component of the derivative of g on $\boldsymbol{d}_j$ from a variation in $\boldsymbol{d}_i$ is $\boldsymbol{d}_j^\mathsf{T} \frac{1}{\lambda} J_f(\boldsymbol{x})\boldsymbol{d}_i$; putting in matrix form, we arrive at

$$J_g(\boldsymbol{p}) = \frac{1}{\lambda}\boldsymbol{O_x}^\mathsf{T} J_f(\boldsymbol{x})\boldsymbol{O_x},$$

which implies local attractivity if Schur. □

6.A.3 Proof of Theorem 6.3

We start by introducting the following lemma.

Lemma 6.3. *Let $\boldsymbol{N} \in \mathbb{S}^n$ be a nonsingular symmetric matrix. The following holds:*

1) *There is a plane through the origin $\mathcal{P}$ such that $\boldsymbol{x} \in \mathcal{P} \setminus \{0\} \implies \boldsymbol{x}^\mathsf{T}\boldsymbol{N}\boldsymbol{x} > 0$ if and only if $\boldsymbol{N}$ has at least two positive eigenvalues.*
2) *There is a plane through the origin $\mathcal{P}$ such that $\boldsymbol{x} \in \mathcal{P} \implies \boldsymbol{x}^\mathsf{T}\boldsymbol{N}\boldsymbol{x} = 0$ if and only if $n \geq 4$, and $\boldsymbol{N}$ has at least two positive and two negative eigenvalues.*

Proof. (1) This is a trivial consequence of Sylvester's law of inertia (see [84, Chap. XV.4]).
(2) Based on Prop. 5.5, this is equivalent to $\boldsymbol{V}^\mathsf{T}\boldsymbol{N}\boldsymbol{V} = \boldsymbol{0}$, for some $\boldsymbol{V} \in \mathbb{R}^{n\times 2}$.

Proof of necessity: We assume that some full-rank $\boldsymbol{V} \in \mathbb{R}^{n\times 2}$ satisfies $\boldsymbol{V}^\mathsf{T}\boldsymbol{N}\boldsymbol{V} = \boldsymbol{0}$ and prove that $\boldsymbol{N}$ has at least two positive and two negative eigenvalues (thus $n \geq 4$). Using Sylvester's law of inertia, we can write $\boldsymbol{N} = \boldsymbol{T}^\mathsf{T}\boldsymbol{S}\boldsymbol{T}$, where $\boldsymbol{S}$ is diagonal containing only 1 and -1 entries in the diagonal, and $\boldsymbol{T}$ is invertible. Thus, $\boldsymbol{V}^\mathsf{T}\boldsymbol{N}\boldsymbol{V} = \boldsymbol{W}_1^\mathsf{T}\boldsymbol{S}\boldsymbol{W}_1$, where $\boldsymbol{W}_1 = \boldsymbol{T}\boldsymbol{V}$ also has rank 2. Let $\boldsymbol{W}_2 \coloneqq \boldsymbol{S}\boldsymbol{W}_1$. Since $\boldsymbol{S}$ is invertible, $\boldsymbol{W}_2$ has rank 2 as well. Because $\boldsymbol{W}_1^\mathsf{T}\boldsymbol{W}_2 = \boldsymbol{0}$, the columns of $\boldsymbol{W}_1$ are orthogonal to the columns of $\boldsymbol{W}_2$, hence $\boldsymbol{W} \coloneqq \begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{W}_2\end{bmatrix}$ has rank 4, which implies that $n \geq 4$. Pre-multiplying $\boldsymbol{W}_2 = \boldsymbol{S}\boldsymbol{W}_1$ by $\boldsymbol{S}$, we get $\boldsymbol{S}\boldsymbol{W}_2 = \boldsymbol{S}^2\boldsymbol{W}_1 = \boldsymbol{W}_1$ (note that $\boldsymbol{S}^2 = \mathbf{I}$). Thus, we can write

$$\boldsymbol{S}\begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{W}_2\end{bmatrix} = \boldsymbol{S}\boldsymbol{W} = \begin{bmatrix}\boldsymbol{W}_2 & \boldsymbol{W}_1\end{bmatrix} = \boldsymbol{W}\boldsymbol{P},$$

where $\boldsymbol{P} \coloneqq \left[\begin{smallmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{smallmatrix}\right]$ is a permutation matrix. This matrix has two eigenvalues in 1 and two eigenvalues in -1. Now, take one pair $(\lambda, \boldsymbol{x})$ such that $\boldsymbol{P}\boldsymbol{x} = \lambda\boldsymbol{x}$. Then, $\boldsymbol{S}\boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{P}\boldsymbol{x} = \lambda\boldsymbol{W}\boldsymbol{x}$, so λ is also an eigenvalue of $\boldsymbol{S}$. Thus, $\boldsymbol{S}$ has at least two eigenvalues equal to 1 and two equal to -1.

Proof of sufficiency: now we start with a nonsingular matrix $\boldsymbol{N}$ with two positive and two negative eigenvalues, and then construct $\boldsymbol{V} \in \mathbb{R}^{n\times 2}$ such that $\boldsymbol{V}^{\mathsf{T}}\boldsymbol{N}\boldsymbol{V} = \mathbf{0}$. Take the Sylvester matrix $\boldsymbol{S}$ of $\boldsymbol{N}$ and select 4 rows and columns such that the corresponding submatrix has exactly two values of 1 and two of -1. Denote by this submatrix $\boldsymbol{S}_4$. We have that that there exists $\boldsymbol{W}_4 \in \mathbb{R}^{4\times 4}$ such that $\boldsymbol{W}_4^{-1}\boldsymbol{S}_4\boldsymbol{W}_4 = \boldsymbol{P}$, where $\boldsymbol{P}$ the same permutation matrix as in the proof of necessity, so $\boldsymbol{W}_4$ can be determined by the eigendecomposition of $\boldsymbol{P}$. Complete $\boldsymbol{W} \in \mathbb{R}^{n\times 4}$ from $\boldsymbol{W}_4$ by padding the remaining rows with zeros, and denote the first two columns of $\boldsymbol{W}$ by $\boldsymbol{W}_1$. Using the same arguments as in the proof of necessity, the matrix $\boldsymbol{V} = \boldsymbol{T}^{-1}\boldsymbol{W}_1$ satisfies $\boldsymbol{V}^{\mathsf{T}}\boldsymbol{N}\boldsymbol{V} = \mathbf{0}$. □

Proof of Theorem 6.3. Case (i) is trivial, since $\underline{\tau} = \bar{\tau}$ implies periodic sampling, so the whole $\mathbb{R}^{n_{\mathrm{x}}}$ is fixed and isochronous.

For case (ii), suppose $n_{\mathrm{x}} < 3$; then every isochronous set is composed by quadratic sets of the form $\cap_i\{\boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}} \mid \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}_i\boldsymbol{x} >(\geq) 0\}$, but by Lemma 6.3 (1) every such $\boldsymbol{Q}_i$ must have two positive eigenvalues; thus, every $\boldsymbol{Q}_i \succ \mathbf{0}$ which implies that $\mathcal{Q}_\tau = \mathbb{R}^{n_{\mathrm{x}}}$ for all k, hence every state samples at every possible inter-sample time. Because a state can only sample at one inter-sample time, this implies τ is unique, hence $\underline{\tau} = \bar{\tau}$, which is a contradiction.

For case (iii), in the CETC case every set $\mathcal{Q}_\tau$ is an interesecion of sets including the set $\{\boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{x}}} \mid \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}_\tau\boldsymbol{x} = 0\}$. Hence, by Lemma 6.3 (2), if $n_{\mathrm{x}} < 4$ then no plane can belong to $\mathcal{Q}_\tau$; thus $n_{\mathrm{x}} \geq 4$. □

6.A.4 Proof of Proposition 6.9

First we introduce the following Lemma.

Lemma 6.4. *Let $\xi(k+1) = \boldsymbol{M}\xi(k)$ be a linear autonomous system, $\boldsymbol{M}$ mixed, and let $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ be the unitary eigenvectors of $\boldsymbol{M}$ ordered from largest-in-magnitude corresponding eigenvalue to smallest. Denote by $\mathcal{A}$ any linear invariant of $\boldsymbol{M}$ containing $\boldsymbol{v}_1$. Then, for every initial state $\xi(0) = a_1\boldsymbol{v}_1 + \cdots = a_n\boldsymbol{v}_n$ where $a_1 \neq 0$, it holds that*

$$\lim_{k\to\infty} \frac{\xi(k)}{|\xi(k)|} \in \mathcal{A}.$$

Proof. This is consequence of the proof of Lemma 5.3 when $a_1 \neq 0$. □

Lemma 6.4 paraphrases the known fact that almost every trajectory of a linear system converges to its dominant mode.

Proof of Proposition 6.9. **Item (I):** For contradiction, assume that σ^ω is stable, and let $m \coloneqq |\sigma|$. First, we check $\boldsymbol{x}_0 \in \mathcal{Q}_\sigma$. In this case, the samples $\boldsymbol{x}_i$ evolve according to $\boldsymbol{x}_{i+m} = \boldsymbol{M}_\sigma\boldsymbol{x}_i$. Let $\boldsymbol{x}_0 = a_1\boldsymbol{v}_1 + \cdots = a_n\boldsymbol{v}_n$ where $\boldsymbol{v}_j$ are the eigenvectors of $\boldsymbol{M}_\sigma$ ordered as in Lemma 6.4. For any $\boldsymbol{x}_0 \in \mathbb{R}^n$ almost all points in its neighborhood satisfy $a_1 \neq 0$. Hence, by Lemma 6.4, $\lim_{i\to\infty} \frac{\boldsymbol{x}_{mi}}{|\boldsymbol{x}_{mi}|} \in \mathcal{A}$, but $\mathcal{A} \not\subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$. Thus, $\{\boldsymbol{x}_{mi}\}$ escapes $\mathcal{Q}_\sigma$ at a some finite i, hence $\{y(\boldsymbol{x}_{mi})\} \neq \sigma^\omega$. This contradicts the assumption that σ^ω is stable.

We now see that the set of states $\boldsymbol{x}$ such that $\mathcal{B}^{\omega}_{\boldsymbol{x}}(\mathcal{S}) = \alpha\sigma^{\omega}$, $|\alpha| < \infty$ is measure zero. This is done by induction on the length of α. Let $\mathcal{X}_m$ be the set of states whose behavior is if $\alpha\sigma^{\omega}$ with $|\alpha| = m$. If $m = 0$, we have already seen that $\mathcal{X}_0$ is a linear invariant of $\boldsymbol{M}_{\sigma}$; because this linear subspace does not contain $\boldsymbol{v}_1$, it has zero measure. Now assume $\mathcal{X}_m$ has zero measure. The set $\mathcal{X}_{m+1}$ is the pre-image of $\mathcal{X}_m$, hence, $\mathcal{X}_{m+1} \subseteq \cup_{k=1}^{\bar{k}} \boldsymbol{M}_k^{-1}\mathcal{X}_m$. Because $\boldsymbol{M}_k$ is nonsingular and the union is finite, $\mathcal{X}_{m+1}$ is also measure zero. This concludes the proof that σ^{ω} is unstable.

Item (II): First, note that c must be a multiple of $|\sigma|$. Let α be any l-long subsequence of σ^{ω}. We have already seen that for almost every $\boldsymbol{x} \in \mathcal{Q}_{\alpha}$ there exists a finite k such that the solution to (2.8) $\boldsymbol{\xi}_{\boldsymbol{x}}(ck-1) \notin \mathcal{Q}_{\alpha}$. To prove absolutely instability, it suffices to check the behavior from any such $\boldsymbol{\xi}_{\boldsymbol{x}}(ck-1)$ does not contain σ^L for some L large enough. We show that it is true with $L = c/|\sigma|$.

Let $\mathcal{C}$ be the simple-cycle SCC formed by $\{x_1, x_2, ..., x_c\}$. Since $\alpha \in \mathcal{C}$, w.l.o.g., let $x_1 = \alpha$, which is related to $\boldsymbol{x}$. Let $\boldsymbol{x}' := \boldsymbol{\xi}_{\boldsymbol{x}}(ck)$ and take x' as the unique state in $\mathcal{S}_l$ related to $\boldsymbol{x}'$, respectively. We first show that $x' \notin \mathcal{C}$: since there is a run from $\boldsymbol{x}$ to $\boldsymbol{x}'$ with length ck, there must be a run segment of length ck from x_1 to x'. Because $\mathcal{C}$ is strongly connected, if $x' \in \mathcal{C}$, the only path would be $(x_1...x_c)^k x_1$, hence $x' = x_1$. But this is a contradiction because $x' \neq \alpha$ since $\boldsymbol{\xi}_{\boldsymbol{x}}(ck-1) \notin \mathcal{Q}_{\alpha}$. Thus, $x' \notin \mathcal{C}$.

Now, there is no path in the abstraction connecting x' to $\mathcal{C}$ (otherwise $\mathcal{C}$ would not be a simple cycle). Therefore, because $\mathcal{S}_l \succeq \mathcal{S}$, it is trivial to see that there is also no path from $\boldsymbol{x}'$ back to $\mathcal{Q}_{\alpha}$, i.e., $\boldsymbol{\xi}_{\boldsymbol{x}'}(k) \notin \mathcal{Q}_{\alpha}, \forall k \in \mathbb{N}$. Thus, $\mathcal{B}^{\omega}_{\boldsymbol{x}'}(\mathcal{S})$ does not contain σ^L as a subsequence, concluding the proof. □

7

Optimizing sampling performance

ETC is a greedy sampling strategy in the sense that it always maximizes the lateness of the next sample, irrespective of long term impacts of this choice. We show that PETC is not optimal in terms of maximizing the smallest average inter-sample time (SAIST) while satisfying its own triggering condition. For that, we devise an STC based on the same triggering condition that can sample earlier than the ETC would. We abstract this system and solve a mean-payoff game to maximize the SAIST, which we show in a numerical example to substantially increase the performance w.r.t. the reference PETC system. We provide optimality bounds, and how to further improve the results through abstraction refinement techniques.

This chapter is based on *G. de A. Gleizer, K. Madnani and M. Mazo Jr. "Self-Triggered Control for Near-Maximal Average Inter-Sample Time", in 60th IEEE Conference on Decision and Control (CDC'21) [85].*

7.1 Introduction

Among nearly all existing work on ETC and STC, there is an underlying philosophy of sampling: it must be as late as possible. Quite often, the *triggering condition* is a surrogate of an underlying Lyapunov function condition, either ensuring its decrease in continuous time [5], its boundedness with respect to a decaying function [7, 14, 86], or its decrease in average [8]. The sampling happens as soon as the condition is violated, that is, as late as possible. Posing this sampling strategy as a mechanism to maximize SAIST, this corresponds to a *greedy* approach for optimization: one maximizes the short-term reward hoping that this brings long-term maximization. It is not surprising that this approach is very rarely optimal, and we have no reason to believe this would be the case with sampling for control.

Some exceptions to the aforementioned rule exist. Aiming at performance improvements, Antunes et al [87] addressed the problem of co-designing controllers and sampling strategies that minimize a quadratic control performance while ensuring that the SAIST is not smaller than some reference periodic controller. Before the ETC era, the problem of minimizing a weighted sum of quadratic performance and AIST for linear controllers with random disturbances was addressed in [88], using an elegant dynamic programming approach; unfortunately, this approach lacks the practicality of its STC successors, as the mere decision to sample or not requires solving an online recursion involving complicated operators. Therefore, we choose to focus on the more practical Lyapunov-based ETC and STC strategies.

In this chapter, we address the problem of synthesizing a self-triggered mechanism for state feedback of linear systems to maximize its SAIST, while ensuring the same control performance as a reference PETC. The improvement over PETC's SAIST is attained by considering predicted PETC triggering times as *deadlines*: sampling earlier always ensures equal or better control performance, and done in the right way it can provide long-term benefits in terms of average sampling. To determine when these long-term rewards are attained, we build on the weighted abstractions of Chapter 5, augmented with early sampling actions as done for scheduling in Chapter 3. In this case, the weights become the sampling actions, but instead of maximizing the next IST as ETC and STC do, we solve a a *mean-payoff game* on the abstraction to optimize the average weight, getting a strategy that approximately maximizes the concrete system's SAIST; the value of the abstraction game is in fact a lower bound to the optimal early-triggering strategy, while computing a modified cooperative maximum SAIST of the abstraction gives an upper bound. Our approach is based on l-complete abstractions, which gives a state-dependent sampling strategy (SDSS) that requires predicting the next l inter-sample times that the reference PETC would generate. As we will see in a numerical example, even $l = 1$ can provide massive improvements to the closed-loop system's AIST by simply using the strategy obtained from the mean-payoff game. The computational cost of this prediction is proportional to l, making our approach implementable on hardware with limited capabilities or time-critical applications. In addition, a major benefit of using abstractions is that the methodology is general in the sense that it can be used for more complex control specifications; and while we focus on the control of a single linear system, extensions to multiple controllers sharing a network (Chapter 3) and nonlinear systems [30] are possible using existing abstraction methods.

7.2 Problem Formulation

Consider a system (2.2) with sample-and-hold as described in Section 2.4. We are interested in constructing an STC strategy, i.e., using available information at time t_i to determine the next sampling time t_{i+1}. Because the system of interest is time-invariant, it suffices to determine the next inter-sample time (IST) $\tau_i := t_{i+1} - t_i$. In general terms, when full-state information is available, STC is a controller composed with a *state-dependent sampling strategy* (SDSS) $s : (\mathbb{R}^{n_x})^i \to \mathbb{R}_+$ that can consider the first i state samples to determine τ_i. As we consider mean-payoff objectives, which admit *positional* (or *static*) strategies [27], we focus on the particular case of *static* state-dependent sampling strategies, which have the form $s : R^{n_x} \to \mathbb{R}_+$. The algorithm that dictates τ_i must guarantee given performance specifications while being fast enough to execute in real time, depending on the application.

Given system (2.2), and an initial state $\boldsymbol{x} := \boldsymbol{\xi}(0)$, an SDSS determines a unique state trajectory $\boldsymbol{\xi}_{\boldsymbol{x}}(t)$, as well as the sequence of ISTs τ_i, since $\tau_0 = s(\boldsymbol{x})$ and the following recursion on i holds, similarly to the sample map (2.8):

$$\begin{aligned} \boldsymbol{\xi}_{\boldsymbol{x}}(t) &= \boldsymbol{M}(t - t_i)\boldsymbol{\xi}_{\boldsymbol{x}}(t_i),\ \forall t \in [t_i, t_{i+1}] \\ t_{i+1} &= t_i + \tau_i, \\ \tau_i &= s(\boldsymbol{\xi}_{\boldsymbol{x}}(t_i)), \end{aligned} \tag{7.1}$$

where as usual $\boldsymbol{M}(t) := \boldsymbol{A}_{\mathrm{d}}(t) + \boldsymbol{B}_{\mathrm{d}}(t)\boldsymbol{K} := \mathrm{e}^{\boldsymbol{A}t} + \int_0^t \mathrm{e}^{\boldsymbol{A}\tau}\mathrm{d}\tau\boldsymbol{B}\boldsymbol{K}$ is the state transition matrix under the held control input. Thus, for convenience, we denote the unique IST sequence of a given SDSS s from a given initial condition $\boldsymbol{x}$ by $\{\tau_i(\boldsymbol{x}; s)\}$. With this, we can proceed in a similar way to Chapter 5 to define the average IST (AIST) from $\boldsymbol{x}$ as

$$\mathrm{AIST}(\boldsymbol{x}; s) := \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \tau_i(\boldsymbol{x}; s),$$

and the smallest AIST (SAIST) across all initial states as

$$\mathrm{SAIST}(s) := \inf_{\boldsymbol{x}\in\mathbb{R}^{n_x}} \mathrm{AIST}(\boldsymbol{x}; s). \tag{7.2}$$

The SAIST of a given SDSS gives its sampling performance, which we are interested in maximizing.

A standard approach to SDSS design is predicting the last moment in time such that the a companion ETC triggering condition (2.3) is still met:

$$t_{i+1} = \sup\{t \in h\mathbb{N} \mid t > t_i \text{ and } c(t - t_i, \boldsymbol{\xi}(t), \hat{\boldsymbol{\xi}}(t))\}, \tag{7.3}$$

where h is the resolution of the line search involved in this check.

Clearly, the ETC and STC strategies described in (2.3) and (7.3), respectively, are greedy sampling strategies: they maximize the next IST while ensuring the triggering condition is not violated.[1] Therefore, we can use the ETC-generated IST as a state-dependent *deadline* $d : \mathbb{R}^{n_x} \to \mathcal{T}$ of an SDSS:

$$d(\boldsymbol{x}) := \max\{\tau \in \mathcal{T} \mid c(\tau, \boldsymbol{\xi}_{\boldsymbol{x}}(\tau), \boldsymbol{x})\}, \tag{7.4}$$

[1] In the PETC case, the triggering condition can be violated for over up to h time units, but one can use a predictive approach as in [86] to prevent violations.

where $\mathcal{T} = \{h, 2h, ..., \bar{\tau}\}$ is the set of possible inter-sample times, and $\bar{\tau} = hK$, with $K \in \mathbb{N}$. The question that naturally arises is whether sampling earlier than this deadline can provide long-term benefits in terms of average inter-sample time. This possibility is exactly what we exploit in this chapter. Hereafter, we shall refer to an SDSS $s^* : \mathbb{R}^{n_x} \longrightarrow \mathcal{T}$ that respects the deadlines in Eq. (7.4) as an *early-triggering SDSS*.

Objective of this chapter. For a given system (7.1), denote the reference PETC strategy by s; our main objective is to design an early-triggering SDSS s^* that ensures a strict improvement in SAIST; i.e., $\text{SAIST}(s^*) \geq \text{SAIST}(s) + v$, where $v > 0$. In addition, we want to estimate how far s^* is from the optimal strategy, i.e., find ϵ such that $\text{SAIST}(s^*) \geq \text{SAIST}(s') - \epsilon$ for all early triggering SDSSs s'.

7.3 Finding an SDSS through abstractions

As we have stated, our approach is to abstract a reference PETC system, compute a strategy for the abstraction aiming at maximizing SAIST, and verifying how this strategy performs as a concrete SDSS for an LTI system. Since the strategy design problem is a quantitative one, we need some adaptations to the framework presented in Section 2.2.3.

7.3.1 Weight-based abstractions

Because we are interested in a control design problem, normal simulation relations as in Section 2.2.1 are not sufficient. Instead, [17] proposes the use of *alternating simulation relations* [43] for this end. Here we adapt the definition of Tabuada for a relation that implies matching weights:

Definition 7.1 (Weight-based alternating simulation relation)**.** *Consider two weighted systems (Def. 2.7) $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{U}_a \subseteq \mathcal{U}_b$ and $\gamma_a(\mathcal{E}_a) = \gamma_b(\mathcal{E}_b)$. A relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is a* weight-based alternating simulation relation *(WBASR) from $\mathcal{S}_a$ to $\mathcal{S}_b$ if the following conditions are satisfied:*

i) for every $x_{b0} \in \mathcal{X}_{b0}$, there exists $x_{a0} \in \mathcal{X}_{a0}$ with $(x_{a0}, x_{b0}) \in \mathcal{R}$;
ii) for every $(x_a, x_b) \in \mathcal{R}$, it holds that $U_a(x_a) \subseteq U_b(x_b)$;
iii) for every $(x_a, x_b) \in \mathcal{R}, u \in U_a(x_a), x'_b \in \text{Post}_u(x_b)$, there exists $x'_a \in \text{Post}_u(x_a)$ such that $\gamma_a(x_a, u, x'_a) = \gamma_b(x_b, u, x'_b)$ and $(x'_a, x'_b) \in R$.

This definition is useful for using strategies for $\mathcal{S}_a$ on $\mathcal{S}_b$: for any initial state in $\mathcal{X}_{b0}$, we can initialize the abstraction $\mathcal{S}_a$ with a related state. Then, any action u available at x_{a0} is also available in $\mathcal{S}_b$ thanks to condition (ii). Finally, whatever transition (x_b, u, x'_b) system $\mathcal{S}_b$ takes, we can pick a transition with same weight in $\mathcal{S}_a$ that again leads to related states (x'_a, x'_b), and both systems can continue progressing. The main difference in our definitions from those in [17], apart from the weighted aspect, is that we are concerned with an "output map" on transitions rather than on states; this is more convenient for our intended application, but not fundamentally different as one could convert a system with weights on transitions to one with weights on states. In this chapter we focus on Def. 7.1 instead of Def. 2.9. Hence we shall call a WBASR simply ASR; also, when a ASR from $\mathcal{S}_a$ to $\mathcal{S}_b$ exists, we say that $\mathcal{S}_b$ is simply an alternating simulation of $\mathcal{S}_a$, denoting it by $\mathcal{S}_a \preceq_{\text{AS}} \mathcal{S}_b$. The following results are useful for our application:

Proposition 7.1. *Consider two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ and an alternating simulation relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$ from $\mathcal{S}_a$ to $\mathcal{S}_b$. If $\forall (x_a, x_b) \in \mathcal{R}, U_a(x_a) = U_b(x_b)$, then $\mathcal{R}^{-1}$ is a simulation relation from $\mathcal{S}_b$ to $\mathcal{S}_a$.*

Proof. Condition (i) of Def. 2.2 is the flipped version of Def. 7.1 (i). Now assume Def. 2.2 (ii) is false for some $(x_b, x_a) \in \mathcal{R}^{-1}$; then there is (x_b, u_b, x_b') without a matching transition in $\mathcal{S}_a$. However, by assumption, $u_b \in U_a(x_a)$; hence, from Def. 7.1 (iii), every $x_b' \in \mathrm{Post}_{u_b}(x_b)$ has a matching $x_a' \in \mathrm{Post}_{u_b}(x_a)$ s.t. $(x_a', x_b') \in \mathcal{R}$, which is a contradiction. □

Now, denote by $\mathcal{S}|s$ the system $\mathcal{S}$ controlled by strategy s; the following result is very similar to other results on alternating simulations (e.g., [17]):

Proposition 7.2. *Let $\mathcal{S}_a$ and $\mathcal{S}_b$ be two non-blocking systems such that $\mathcal{S}_a \preceq_{\mathrm{AS}} \mathcal{S}_b$. Let $\mathcal{V}^* \subseteq \mathcal{V}^\omega(\mathcal{S}_b)$ be a set of value traces. If there exists a strategy $s : X_b^+ \to \mathcal{U}$ such that $\mathcal{V}^\omega(\mathcal{S}_b|s) \subseteq \mathcal{V}^*$, then there exists a strategy $s' : X_a^+ \to \mathcal{U}$ such that $\mathcal{V}^\omega(\mathcal{S}_a|s') \subseteq \mathcal{V}^*$.*

Proof. (Sketch) Consider the alternating simulation relation $\mathcal{R} \subseteq (X_a, X_b)$. Take any state $x_b \in X_{b0}$: we can pick a related x_a (Def. 7.1 (i)). Choose $u = s(x_a)$; then $\forall (x_a, u, x_a') \in \mathcal{E}_a$, $\gamma_a(x_a, u, x_a')$ is the first element of some sequence in $\mathcal{V}^*$. From Def. 7.1 (ii), we can use u from x_b. From (iii), for any $(x_b, u, x_b') \in \mathcal{E}_b$ there exists $(x_a, u, x_a') \in \mathcal{E}_a$ with $(x_a', x_b') \in \mathcal{R}$, with matching weights. Since all such (x_a, u, x_a') have valid weights, so does any (x_b, u, x_b'). Hence, from any x_b' we can pick a related x_a' and choose $u = s(x_a x_a')$ for the next iteration. Repeating the same arguments recursively concludes the proof. □

The proof of Prop. 7.2 gives a way to use a strategy from an abstraction on the concrete system: run the abstraction in parallel with the concrete system: from any state x_b, pick the winning action u from a related state x_a and move the abstraction forward. If the strategy s is static, running the abstraction in parallel is unnecessary: the alternating simulation relation alone suffices, as long as the relation satisfies $|\{x_a \mid (x_a, x_b) \in \mathcal{R}\}| = 1$ for all $x_b \in \mathcal{X}_b$, i.e., every concrete state has a single related abstract state. This is the case for quotient systems, which we employ here. Hereafter we assume this is true; then, given a static strategy $s_a : \mathcal{X}_a \to \mathcal{U}_a \subseteq \mathcal{U}_b$, we call $s_b : \mathcal{X}_b \to \mathcal{U}_b$ a *refined strategy to* $\mathcal{S}_b$, or simply a *refinement of* s_a, by setting $s_b(x_b) = s_a(x_a)$, x_a being the unique state satisfying $(x_a, x_b) \in \mathcal{R}$.

7.3.2 Abstractions for optimal average weight

A transition system according to Def. 2.7 can be regarded as a game, where player 0 picks the action and player 1 antagonistically picks the transition. The problem of finding a strategy for a finite-state weighted transition system in order to maximize the average weight is known as mean-payoff game, a quantitative game. Quantitative games are games on weighted transition systems where a value function $\mathrm{Val} : \mathcal{V}^\omega(\mathcal{S}) \to \mathbb{R}$ is defined on runs, and one wants to find a strategy s for player 0 that

i) ensures a minimum value v, i.e, $\mathrm{Val}(\nu) \geq v$ for all $\nu \in \mathcal{V}^\omega(\mathcal{S}|s)$, or, if possible,
ii) maximizes the value, i.e., finds $\bar{v}$ such that $\mathrm{Val}(\nu) \geq \bar{v}$ for all $\nu \in \mathcal{V}^\omega(\mathcal{S}|s)$ and that, for any $v > \bar{v}$, there is no strategy s' s.t. $\mathrm{Val}(\nu') \geq v$ for all $\nu' \in \mathcal{V}^\omega(\mathcal{S}|s')$.

In the case of mean-payoff objectives, the value function is $\mathrm{LimAvg}(\nu)$. There exist positional (i.e., static) optimal strategies for mean-payoff games [27],[2] which implies that every $x \in \mathcal{X}$

[2]Solving the optimal value and strategy of a mean-payoff game has pseudo-polynomial complexity, with the best known bound of $\mathcal{O}(|\mathcal{X}|^2|\mathcal{E}|W)$ due to [89], where W is the maximum weight. Weights are assumed to be

admits an *optimal value* $V(x)$ that is the smallest value obtained from every run starting at x when both players 0 and 1 play optimally. By considering that we cannot control the initial state, i.e., player 1 picks it, we have that the *game value* is $V(\mathcal{S}) := \bar{v} = \inf\{V(x) \mid x \in \mathcal{X}_0\}$. Once a strategy s has been decided for a system $\mathcal{S}$, $\mathcal{S}|s$ becomes a 1-player game, containing only the (antagonistic) environment. In the case of a 1-player game, where the environment chooses initial states and transitions, the value of the game can be taken as a function of its weight behaviors, i.e., $V_{\text{adv}}(\mathcal{S}|s) = \inf\{\text{Val}(\nu) \mid \nu \in \mathcal{V}^{\omega}(\mathcal{S}|s)\}$. We also define a cooperative value from a given initial condition $V^{x}_{\text{coop}}(\mathcal{S}) := \sup\{\text{Val}(\nu) \mid \nu \in \mathcal{V}^{\omega}_{x}(\mathcal{S})\}$, where player 0 has control of transitions and actions for any given initial state x; furthermore, let $V_{\text{U}}(\mathcal{S}) := \inf_{x\in\mathcal{X}_0} V^{x}_{\text{coop}}(\mathcal{S})$: essentially, player 1 picks the initial state, but player 0 can choose the run from it. The following result, very similar to what was done in [90] for time-optimal control, gives that a strategy from an abstraction, refined to the concrete system, ensures that at least the abstraction value is attained in the concrete case, while an upper bound can be obtained from V_{U}:

Proposition 7.3. *Let $\mathcal{S}_a \preceq_{\text{AS}} \mathcal{S}_b \preceq \mathcal{S}_a$. Then, $V(\mathcal{S}_b) \leq V_{\text{U}}(\mathcal{S}_a)$. Moreover, let s_a be a strategy for $\mathcal{S}_a$ such that $V_{\text{adv}}(\mathcal{S}_a|s_a) \geq v$. Then, if s_b is a refinement of s_a, it holds that $V_{\text{adv}}(\mathcal{S}_b|s_b) \geq v$.*

Proof. (Sketch) The first inequality comes from $V(\mathcal{S}_b) \leq V_{\text{U}}(\mathcal{S}_b) \leq V_{\text{U}}(\mathcal{S}_a)$; the first holds by definition of V_{U}, where player 1 is more powerful than in the original game; and the second inequality holds from (weight) behavioral inclusion. For the last statement, it can be seen using similar arguments as [17, Sec. 8.2] that $\mathcal{S}_b|s_b \preceq \mathcal{S}_a|s_a$; hence, $\mathcal{V}^{\omega}(\mathcal{S}_b|s) \subseteq \mathcal{V}^{\omega}(\mathcal{S}_a|s)$, which implies that $V_{\text{adv}}(\mathcal{S}_a|s_a) \geq v \implies V_{\text{adv}}(\mathcal{S}_b|s_b) \geq v$. □

Prop. 7.3 implies that one can use an abstraction $\mathcal{S}_a$ to find a near-optimal strategy for the concrete system $\mathcal{S}_b$, then estimate its optimality gap by computing $\epsilon = V_{\text{U}}(\mathcal{S}_a) - V_{\text{adv}}(\mathcal{S}_a|s_a)$.[3]

7.3.3 SDSS design

Now that we have the relevant notions of abstraction in place, we can proceed to apply them towards an SDSS design for System (2.2). For that, we fix the PETC traffic model early triggering of Def. 2.17 as $\mathcal{S}$. Note the peculiarities of this transition system: (i) the input set of a given state is determined by the state's output, which is its deadline; and, (ii) the weight is solely a function of the chosen input. These characteristics simplify the job of finding an abstraction that is alternatingly weight simulated by $\mathcal{S}$. As we show, this can be achieved by simply enhancing the l-complete models from Def. 5.3 with early-sampling transitions. To do so, note that a transition from some state in $\mathcal{Q}_\sigma$ to a state in $\mathcal{Q}_{\sigma'}$ exists if

$$\exists \boldsymbol{x} \in \mathbb{R}^{n_{\text{x}}} \text{ such that } \boldsymbol{x} \in \mathcal{Q}_\sigma \text{ and } \boldsymbol{M}(hu)\boldsymbol{x} \in \mathcal{Q}_{\sigma'}, \tag{7.5}$$

for every u respecting the deadline, i.e., $u \leq \sigma(1)$. We can now define the following abstraction:

integers for complexity analysis; since we assume weights to be in $\mathbb{Q}$, this can always be done by appropriate normalization.

[3] The 1-player-game values can be obtained for finite systems using Karp's algorithm [41], whose complexity, $\mathcal{O}(|\mathcal{X}||\mathcal{E}|)$, is much smaller than that for optimal mean-payoff games; for V_{U}, a combination of Karp's algorithm and reachability on graphs can be used, retaining the same complexity.

Definition 7.2 (l-predictive ETC traffic model with early sampling)**.** *Given an integer $l \geq 1$, the* l-predictive traffic model with early sampling *of $\mathcal{S}$ is $\mathcal{S}_l := (\mathcal{X}_l, \mathcal{X}_l, \mathcal{U}, \mathcal{E}_l, \mathcal{Y}, H_l, \gamma_l)$, where*

- $\mathcal{X}_l := \pi_{\mathcal{R}_l}(\mathcal{X})$,
- $\mathcal{E}_l := \{(\sigma, u, \sigma') \in \mathcal{X}_l \times \mathcal{U} \times \mathcal{X}_l \mid u \leq \sigma(1), \exists \boldsymbol{x} \in \mathbb{R}^{n_x} : \ Eq.\ (7.5)\ holds\}$,
- $H_l(k_1 k_2 ... k_l) := k_1$.
- $\gamma_l(\sigma, u, \sigma') := hu$.

Proposition 7.4. *The relation $\mathcal{R}_l$ from Def. 5.2 is a simulation relation from $\mathcal{S}$ to $\mathcal{S}_l$, and $\mathcal{R}_l^{-1}$ is a alternating simulation relation from $\mathcal{S}_l$ to $\mathcal{S}$.*

Proof. The proof of alternating simulation is obtained by checking the conditions of Def 7.1: (i) is trivially satisfied, and so is (ii) with $U_l(k_1 k_2 ... k_l) = U(\boldsymbol{x}) = \{1, 2, ..., k_l\}$. Condition (iii) is ensured by construction of $\mathcal{E}_l$ and thanks to the fact that $\gamma(\boldsymbol{x}, u, \boldsymbol{x'}) = \gamma_l(\sigma, u, \sigma') = hu$. The simulation then follows from Prop. 7.1. □

Obtaining the strategy and ϵ. Now that we have a method to abstract $\mathcal{S}$ into a finite system, we can use the methods from Section 7.3.2 to build a near-optimal SDSS for the abstraction $\mathcal{S}_l$, then refine it for $\mathcal{S}$. The main question is how to define l. Given the results in [70], we suggest the following approach: (i) use the methods of Chapter 5 to compute the exact PETC SAIST, or a close enough under-approximation of it; denote this value by $V(\mathcal{S}_l | s_{\text{PETC}})$. Set $l = 1$; Then, (ii) compute $\mathcal{S}_l$ (Def. 7.2) and solve the mean-payoff game for it, obtaining the strategy s_l and the game value estimate v_l. After that, with s'_l being a refinement of s_l to $\mathcal{S}$, (iii) compute $V_{\text{U}}(\mathcal{S}_l)$ and verify, using a similar approach to Chapter 5, (a sufficiently close under-approximation of) $V_{\text{adv}}(\mathcal{S} | s'_l)$ [4] Finally, (iv) compute $\epsilon = V_{\text{U}}(\mathcal{S}_l) - V(\mathcal{S}_l)$ and (v) if the improvement $V_{\text{adv}}(\mathcal{S} | s'_l) - V(\mathcal{S}_l | s_{\text{PETC}})$ is large enough, ϵ is small enough, or l is too large,[5] stop; otherwise, increment l and redo steps (ii) to (v).

7.4 Numerical example

Let us revisit the numerical example of Chapter 4, which has a triggering condition that is very aggressive in sampling reductions, but consider a small modification: instead of $\bar{k} = 6$, which could arguably limit the SAIST of the PETC, we set $\bar{k} = 20$, which is high enough so that the system in fact exhibits a maximal IST of 11.

First, we compute the PETC's nominal performance using Algorithm 3: a SACE simulation is found with $l = 8$, giving a SAIST of approximately 0.233. Interestingly, this is smaller than the value of 0.395 that was obtained when we had limited $\bar{k}$ to 6 — already showing how sub-optimal ETC can be. Second, we compute a strategy following the steps in Section 7.3.3 with $l = 1$, which already gives massive improvements: it increases SAIST to $V_{\text{adv}}(\mathcal{S} | s'_1) = 0.5$, which is more than twice the PETC's SAIST. Essentially, the obtained

[4] Even though Chapter 5 was proposed for the PETC strategy, the same approach can be used for any fixed sampling strategy, as its essential feature is verifying cycles in the concrete system. We have implemented this modified algorithm in `ETCetera` [34].

[5] The complexity of the online part of the algorithm value is proportional to l, as the controller must predict the next l deadlines of the current state $\boldsymbol{x}$ under PETC. This involves simulating the PETC forward l steps, which takes at most lK operations of quadratic inequalities. Hence, a maximum l may be needed given online computational constraints.

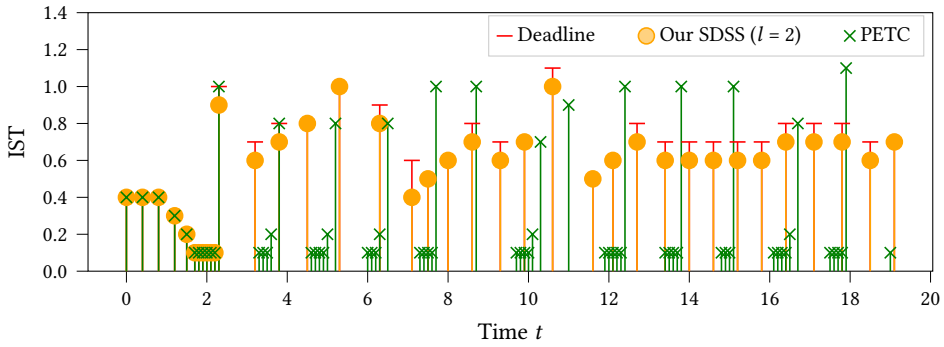


Figure 7.1: Comparison between simulated traces of our SDSS (l = 2) and of the PETC, both with the same initial state.

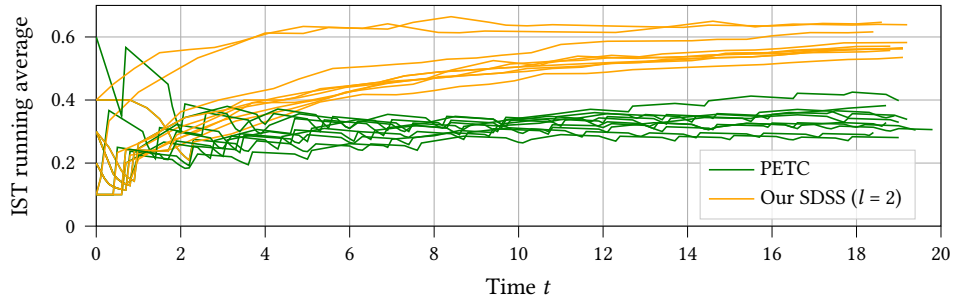


Figure 7.2: Running average of the ISTs generated from 10 different initial conditions under PETC and our near-optimal SDSS using l = 2.

strategy simply limits $s(\boldsymbol{x})$ to 5 for any $k \geq 5$ satisfying $(\boldsymbol{x}, k) \in \mathcal{R}_1$. This is yet another example on how limiting the maximum IST can actually have long-term benefits. With $l = 2$, the SAIST is improved further to 0.6; a simulation comparing this strategy and PETC is depicted in Fig. 7.1. One can see that only around $t = 2.5$ the SDSS samples before the deadline for the first time; doing so prevents the bursts of IST equal to 0.1 that happen recurrently with the PETC. The difference in (simulated) running averages between PETC and our SDSS is displayed in Fig. 7.2. The SAIST for $l = 3$ does not improve, and the upper bounds are $V_{\mathrm{U}}(S_1) = V_{\mathrm{U}}(S_2) = V_{\mathrm{U}}(S_3) = \bar{\tau} = 1.1$.

Reproducing the results of this chapter. The algorithms proposed here have been implemented in `ETCetera`. To reproduce the results above, run the following scripts:

- `examples/optsample_abstractions.py` to compute the abstractions needed to generate the SAIST of the nominal PETC system and strategies, SAISTs and V_{U} values for $l = 1, 2, 3$. This is a time consuming process that saves the obtained abstractions in files for later use;
- `examples/optsample_plots.py` for Figures 7.1 and 7.2, as well as reporting the numbers presented herein.

7.5 Discussion and conclusions

In this chapter we have presented an abstraction-based approach to build aperiodic sampling strategies for LTI systems in order to maximize their average inter-sample time. For this we rely on the properties of ETC strategies, which ensure stability and performance of the closed loop whenever their "deadlines" are respected. This makes our abstraction inherently safe from a control perspective, but one could relax this condition by allowing "late" samplings as long as the abstraction can incorporate some information about the control performance. Multiple ideas can be explored in this direction: e.g, further partitioning the state-space with Lyapunov level sets and add reachability and safety specifications; or adding a Lyapunov-based cost to the transitions of the abstraction. In the latter case, one can either have a safety or reachability objective to the Lyapunov function (e.g., maximize AIST subject to $V(\xi(t)) \leq r$ for all $t > T$), or play a multi-objective quantitative game [91] (e.g., find the Pareto-optimal set of strategies that maximize AIST and control performance).

One open question is whether our methodology gives $V(S_l) \to V(S)$ as $l \to \infty$, or if this is possible to obtain in general at all. We speculate that our method does not attain such asymptotic property, because of our abstraction approach: the state-space is partitioned based on a quotient-based method designed for verification, and later we enrich it with early actions. This is not how the standard bisimulation refinement in games operates (see [17, 92]). Abstracting the way we have proposed has benefits for the complexity of online operations, as to find the related abstract state one needs to simulate an autonomous system (the reference PETC) instead of a tree exploration involving all possible inter-sample times. However, this approach cannot (generally) eliminate the nondeterminism associated with sampling earlier than the deadline.

As seen in our numerical example, the gap ϵ is still very large after a few refinements. We have discussed that our method may give far-from-optimal strategies, but it can also be that V_{U} provides an excessively conservative upper bound. In fact, the example we have considered exhibits chaotic traffic, and upon inspection of its traffic model, it does have an unstable cycle $(11)^{\omega}$, which is exhibited in a behaviorally chaotic invariant set. Hence, no matter how large l is, it always holds that $V_{\mathrm{U}}(S_l) = 11$. At the same time, given the results from Chapter 6, it is impossible to obtain a strategy that gives $V(S|s) = 11$; that would require by Proposition 6.8 that $\boldsymbol{M}_{11}$ is Schur, which is not true. It may be possible, we speculate, to obtain a SAIST arbitrarily close to 11.

Finally, our method suffers (as in all previous chapters) from the curse of dimensionality. In the specific case of this chapter, solving the mean-payoff game is itself also a very computationally intensive process: obtaining the strategy for S_3 took a couple of hours, even though the system has only 334 states and 1591 transitions. Being able to reduce abstractions getting a similar (alternating-simulation-wise) system would be important for the applicability of this method, as well as for other related control-design problems such as scheduling. This is the topic of the next chapter.

8

Minimizing transition systems modulo alternating simulation equivalence

This chapter studies the reduction of finite-state transition systems for control synthesis problems. We revisit the notion of alternating simulation equivalence (ASE), a more relaxed condition than alternating bisimulations, to relate systems and their abstractions. As with alternating bisimulations, ASE preserves the property that the existence of a controller for the abstraction is necessary and sufficient for a controller to exist for the original system. Moreover, being a less stringent condition, ASE allows for further reduction of systems to produce smaller abstractions. We provide an algorithm that produces minimal AS equivalent abstractions. The theoretical results are then applied to obtain (un)schedulability certificates of periodic event-triggered control systems sharing a communication channel. A numerical example illustrates the results.

This chapter is based on 🗎 *G. de A. Gleizer, K. Madnani and M. Mazo Jr. "A simpler alternative: minimizing transition systems modulo alternating simulation equivalence", in Proc. of the 25th Int'l Conf. on Hybrid Systems: Computation and Control (HSCC '22)* [93].

8.1 Introduction

Control synthesis for finite transition systems (FTS), the problem of finding a controller (or strategy) that enforces specifications on a closed-loop system, is a long investigated problem [94]. Supervisory control, as it is often also referred to, has many applications in e.g. automation of manufacturing plants, traffic control, scheduling and planning, and control of dynamical and hybrid systems [17, 95]. The clearest advantage of using finite transition systems to model a control problem is that a large class of control problems in finite transition systems are *decidable*, meaning that the controller can be obtained automatically through an algorithm, or that a definitive answer that no controller can enforce the specifications is obtained. The disadvantage is often a very practical one: the problem may be too large to be solved with existing computational resources, owing to the large number of states and transitions the control problem may have. In particular, this is the case of scheduling the transmissions of ETC, the problem we have addressed in Chapter 3. As we have seen, it is often impossible to synthesize schedulers for more than a handful of ETC systems, due to the state explosion of the composed system. This state-space explosion problem is pervasive in the formal methods literature, and thus significant attention has been devoted to reducing transition systems. The reduction requires a formal relation between original and reduced system; for verification purposes, the most well-known relation is that of *simulation*. Algorithms to reduce systems modulo simulations soon emerged: the first being a reduction modulo *bisimulation*, where algorithms using *quotient systems* are often used [39]; later, minimization modulo *simulation equivalence* was devised in [96]. Simulation equivalence is a weaker relation than bisimulation but allows to verify most of the same properties; in particular, any linear temporal logic (LTL) property that can be verified on a system also holds for a simulation-equivalent system.[1]

For control synthesis, reducing the system using mere simulation notions is not enough. As we have seen in Chapters 2 and 7, the natural notion for control synthesis is that of alternating simulations. In fact, control synthesis amounts to solving a game where the controller plays against an antagonistic environment, and standard simulations preserve all possible moves from both players, *including moves that are irrational* for the game. Surprisingly, though, there has been little investigation of the problem of reducing systems modulo *alternating bisimulations* or *alternating simulation equivalence*. Reducing systems using alternating simulation notions has many practical benefits: not only the synthesis problems become smaller, and by extension the obtained controllers, making them easier to implement in limited hardware; but it becomes even more important, we argue, when solving control synthesis problems on a parallel composition of systems, one classic example being scheduling. In this case, the size of the game grows exponentially with the number of systems to be scheduled, hence *any reduction on the individual systems results in an exponential reduction of the size of the composed game.*

In this chapter we present a novel algorithm to reduce systems w.r.t. alternating-simulation equivalence (ASE), a different and relaxed notion than the more popular relative, alternating bisimulation. ASE is stronger than alternating simulation relations, as it guarantees not only that controllers can be transferred from abstraction to concrete system, but also that non-existence of a controller in the abstraction implies non-existence of a

[1] Larger classes of logic properties can be verified, such ACTL*, ECTL*, ECTL, ACTL as its sublogics, see [96]. For control, we are typically interested in LTL specifications.

controller for the concrete system. Hence the reduction via ASE is sound and complete for control synthesis. We prove that our algorithm in fact obtains a minimal system that is alternating-simulation equivalent to the original, and this minimal system is unique up to a special type of isomorphism. The algorithm is composed of five steps: (i) computing the maximal alternating simulation relation from the system to itself; (ii) forming the quotient system; (iii) eliminating irrational and/or redundant actions from the controller; (iv) eliminating irrational transitions from the environment; and (v) deleting states which are inaccessible from any of the initial states. The complexity of the algorithm is $O(m^2)$, where m is the number of transitions in the system to be reduced. This result is a very interesting theoretical contribution on its own right, generalizing the results in [96]. Because these simulation relations are closed under composition, the presented algorithm has a strong practical relevance for synthesis over composed systems. We demonstrate these benefits in the ETC scheduling problem (Chapter 3) as a case study. The insights from our algorithm allow to prove that, under some conditions, ETC and STC are equally schedulable. Additionally, we use our algorithm on a numerical case study, obtaining in the best case a system 50x smaller than the original one. This resulted in a reduction in CPU time of the scheduling problem of several orders of magnitude in some cases. Furthermore, the reduced systems also provide important insights to the user, as the reduced system somehow indicates the bottlenecks that must be addressed to improve schedulability.

8.1.1 Related Work

Algorithms for reducing state space preserving bisimulation using quotient systems have been extensively studied [97, 98], see [39, 99] for an overview. For many practical results, simulation equivalence, a coarser equivalence relation, is preferable. Various algorithms to obtain quotients based on simulation equivalence have been proposed,e.g., [100, 101], as well as their associated quotients [102]. However, unlike bisimulation, creating quotients based on simulation equivalence does not result in minimization [96]. Our algorithm and results are akin to those of [96]; we have here a generalization of its results, as alternating simulation reduces to simulations if one of the players has only one choice in every state.

The reduction of systems using alternating simulation equivalence has been addressed in [92, 103]. Different from the current work, Majumdar et al. propose a semi-algorithm that aims at reducing infinite systems into finite systems (not necessarily minimal); instead, here we want to minimize finite systems by reducing the number of states and transitions. These two approaches are complimentary and can be used in combination to obtain minimal finite realizations of certain classes of infinite systems (namely, class 2 systems as per [92]).

Reduction of other types of finite transition systems has been addressed, as in, e.g., [104] for alternating Büchi automata modulo different notions of simulations, namely direct, fair, and delayed simulations. Although such automata also represent games, they are defined differently than what is usual for control: an alternating Büchi automaton accepts a word if the controller can ensure it by playing against the environment; every such word forms the language of the automaton, and simulations must preserve this language in some sense. This is fundamentally different than most control problems, where one is not interested in specific words, but rather that the set of all words generated by the system satisfies some specifications. In addition, [104] does not contain results on minimality.

8.1.2 A case for alternating simulation equivalence

Let us revisit the definitions of alternating bisimulation and alternating simulation equivalence (ASE), from Section 2.2.3:

Definition (Alternating bisimulation)**.** *Two transition systems $\mathcal{S}_a$ and $\mathcal{S}_b$ are said to be* alternatingly bisimilar*, denoted by $\mathcal{S}_a \cong_{\text{AS}} \mathcal{S}_b$, if there is an alternating simulation relation (ASR) $\mathcal{R}$ from $\mathcal{S}_a$ to $\mathcal{S}_b$ such that its inverse $\mathcal{R}^{-1}$ is an ASR from $\mathcal{S}_b$ to $\mathcal{S}_a$.*

Definition (Alternating simulation equivalence (ASE))**.** *Two transition systems $\mathcal{S}_a$ and $\mathcal{S}_b$ are said to be* alternating-simulation equivalent*, denoted by $\mathcal{S}_a \simeq_{\text{AS}} \mathcal{S}_b$, if there is an ASR $\mathcal{R}$ from $\mathcal{S}_a$ to $\mathcal{S}_b$ and an ASR $\mathcal{R}'$ from $\mathcal{S}_b$ to $\mathcal{S}_a$.*

Clearly, ASE reduces to bisimulation when $\mathcal{R}' = \mathcal{R}^{-1}$; nevertheless, it preserves by definition the if-and-only-if property we are interested in: if a controller for $\mathcal{S}_a$ exists, then it exists for $\mathcal{S}_b$ (from $\mathcal{S}_a \preceq_{\text{AS}} \mathcal{S}_b$); and if a controller for $\mathcal{S}_b$ exists, then it exists for $\mathcal{S}_a$ (from $\mathcal{S}_b \preceq_{\text{AS}} \mathcal{S}_a$). Moreover, a second relation $\mathcal{R}'$ is an extra degree of freedom to find a reduced system that is ASE to the original—in other words, there exist more systems ASE to a given $\mathcal{S}$ than alternatingly bisimilar to it. There is a price to pay for this freedom: the controller designed for the reduced system will not be as *permissive* as the best controller that could be created by the original system; that is, it may contain fewer actions available to pick from at any point in the system's run. Nonetheless, this can be regarded as a benefit, considering the sheer size that the strategies for large FTSs can have; in addition, in practical implementations the strategies need to be determinized anyway.

8.2 Main result

Let the *size* of a finite transition system (FTS), denoted by $|\mathcal{S}|$, be given by the triplet $(|\mathcal{X}|, |\mathcal{X}_0|, |\mathcal{E}|)$. This induces a partial order amongst systems sizes using the natural extension of $\leq$ on numbers, i.e., $|(\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{Y}, \mathcal{E}, H)| \leq |(\mathcal{X}', \mathcal{X}_0', \mathcal{U}', \mathcal{Y}', \mathcal{E}', H')|$ iff $|\mathcal{X}| \leq |\mathcal{X}'|$, $|\mathcal{X}_0| \leq |\mathcal{X}_0'|$, and $|\mathcal{E}| \leq |\mathcal{E}'|$. In this section, we present our main result:[2] given an FTS $\mathcal{S}$, there exists a polynomial time algorithm that constructs a minimal FTS $\mathcal{S}_{\text{min}}$ equivalent to $\mathcal{S}$ modulo alternating simulation (AS). That is, $\mathcal{S}_{\text{min}} \simeq_{\text{AS}} \mathcal{S}$ and $|\mathcal{S}_{\text{min}}| \leq |\mathcal{S}'|$ for any $\mathcal{S}'$ satisfying $\mathcal{S}' \simeq_{\text{AS}} \mathcal{S}$. (i) We first provide an overview of the algorithm to obtain such a minimal system. (ii) We then provide the details of the each step of the algorithm and prove its correctness by showing that all steps preserve alternating simulation equivalence. (iii) We show that the output of the algorithm is indeed the unique minimum FTS (up to isomorphism) alternating-simulation equivalent to the input FTS $\mathcal{S}$. This, in turn, implies that for every FTS there is a unique minimum FTS equivalent modulo AS system that can be constructed using our algorithm. As we explain the algorithm, we use as a running example the FTS from Fig. 8.1, which is a simple discrete-clock PETC traffic model (see Model 3.5).

8.2.1 Overview of the algorithm

The algorithm can be summarized as follows. For a system $\mathcal{S} := (\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{Y}, \mathcal{E}, H)$, we denote by $\mathsf{TranSize}(\mathcal{S}) := |\mathcal{E}| + |\mathcal{X}_0|$, a measure for number of transitions in the system [3].

[2] When a proof is not right after the result statement, see it in the Appendix.

[3] We add the cardinality of $\mathcal{X}_0$ to total number of transitions because in principle the results we use from [105] assumes that there is a unique initial state. Multiple initial states can be simulated by adding silent transitions

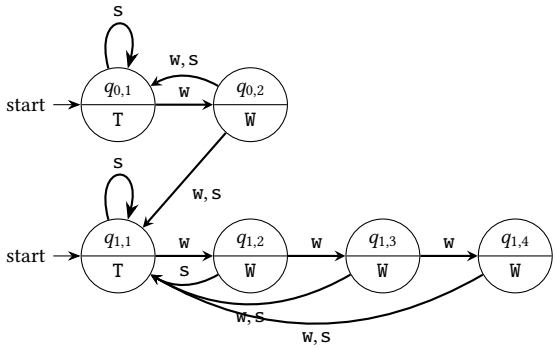


Figure 8.1: A finite LTS representing a PETC traffic model with scheduler actions. Node labels are states (top) and their outputs (bottom), and edge labels are actions.

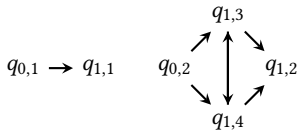


Figure 8.2: Maximal alternating simulation relation $\mathcal{R}^{\max}$ for the system in Fig. 8.1: $q \rightarrow q'$ means that $(q, q') \in \mathcal{R}^{\max}$. Self-loops and relations implied from transitivity are omitted.

Let $|\mathcal{X}| = n$ and $\mathsf{TranSize}(\mathcal{S}) = m$.

Step 0: Construct the maximal alternating simulation relation, denoted by $\mathcal{R}^{\max}$, from $\mathcal{S}$ to itself. This can be constructed using fixed-point algorithms as in [43] or the more efficient algorithm presented in [105], whose complexity is $O(m^2)$.

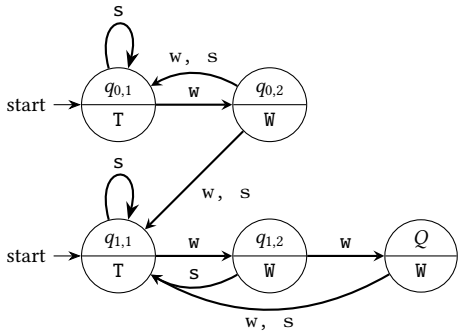
Step 1: Create a quotient system using $\mathcal{R}^{\max}$ of $\mathcal{S}$ by combining all the equivalent states (and hence all their incoming transitions and outgoing transitions) to get a quotient of the system modulo AS. This requires $O(n+m)$ computations given the partition $\mathcal{P}$ (which can be constructed while building $\mathcal{R}^{\max}$) as constructing the quotient transition relation from $\mathcal{E}$ requires taking the union of all the outgoing transitions from any state in the given partition.

Recall that, if $(q, q') \in \mathcal{R}^{\max}$ then if the controller can meet a specification from the state q then it will definitely meet it from state q'. Moreover, if the controller fails to meet the specification from q' it will definitely fail from q. In other words, q' (resp. q) is more advantageous position for the controller (resp. environment) as compared to q (resp. q'). This intuition is central to the next two steps.

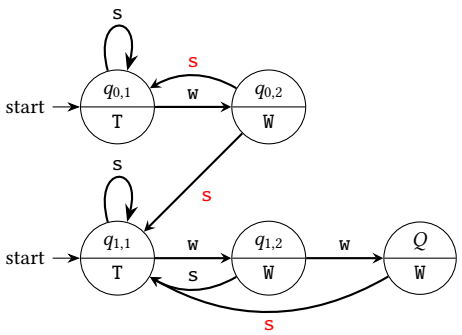
Step 2: Remove irrational choices and redundant choices for the controller: For every $x \in \mathcal{X}$ and every $a, b \in U(x), a \neq b$ if for every $x_b \in \mathrm{Post}(x, b)$ there exists an $x_a \in \mathrm{Post}(x, a)$ such that $(x_a, x_b) \in \mathcal{R}^{\max}$, then delete all transitions from x on a. In other words, remove a from $U(x)$. This is because, for every possible environment move on taking an action b leads to a more (or equally) advantageous state for the controller as compared to any possible state the system can end up on action a by controller. To check this, every transition is compared with every other transition at most once. Hence, the complexity of this step in the worst case is bounded by $O(m^2)$.

Step 3: Remove sub-optimal irrational choices for the environment: For every pair $x_1, x_2 \in \mathcal{X}_0$, if $(x_1, x_2) \in \mathcal{R}^{\max}$, then the choice of environment to start from x_2 will be irrational as x_1 is more advantageous position for the environment to start with. Hence, we remove x_2 from the initial state set (which is clearly an irrational move for the environment). Similarly, if $(x_1, x_2) \in \mathcal{R}^{\max}$, then for every $a \in \mathcal{U}$ if $x' \in \mathrm{Pre}(x_1, a) \cap \mathrm{Pre}(x_2, a)$, remove

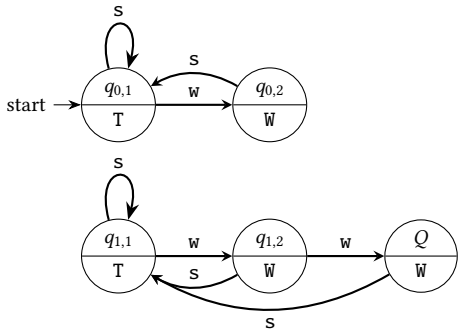
from a dummy initial state to all the states in $\mathcal{X}_0$ which requires $|\mathcal{X}_0|$ extra transitions



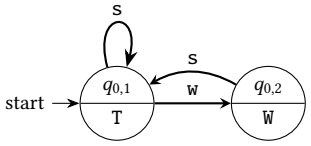
(a) Step 1: quotient system.



(b) Step 2: redundant actions removed (affected transitions in red).



(c) Step 3: irrational transitions removed.



(d) Step 4: inaccessible states removed. This is a minimal system modulo ASE.

Figure 8.3: System of Fig. 8.1 after steps 1 (top left), 2 (top right), 3 (bottom left), and 4 (bottom right), where $Q = \{(q_{1,3}), (q_{1,4})\}$.

8

transition (x', a, x_2) from $\mathcal{E}$. This is because, if the system is at x', and if the controller chooses an action a, the choice of moving to x_2 instead of x_1 is irrational for the environment as x_1 is more advantageous state for the environment. hence, we delete the transition (x', a, x_2). Similarly to step 3, before its deletion (or not), any transition is compared with all other transitions at most once. Hence, the worst case complexity is bounded by $O(m^2)$.

Step 4: Remove inaccessible states: Finally remove all the states that are not accessible from any initial state. This is a routine step with complexity $O(n + m)$. Note that while it seems that Steps 3 and 4 only remove transitions, this does not mean that they do not contribute in the reduction of number of states. Due to the removal of transitions, it could happen that a large fraction of the graph becomes unreachable. This is the step that cashes in the benefit of steps 3 and 4 in terms of reduction in state size.

The maximal alternating simulation relation from our working example (Fig. 8.1) is depicted in Fig. 8.2. Figure 8.3 illustrate the successive application of each step 1–4 on it.

8.2.2 Preserving equivalence modulo AS: correctness results

In this section, we formally present the construction/reduction mentioned in each step 1–4 and show that those reductions preserve equivalence modulo AS. These proofs are

available in the appendix for readability reasons. We also present results on the dimension reduction resulting from each step. We fix $\mathcal{S} := (\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{Y}, \mathcal{E}, H)$ for this section as a given FTS and apply our reduction steps. For any $i \in \{1,2,3,4\}$ the system resulting of applying step i, $S^i(\mathcal{S})$, is denoted by $\mathcal{S}_i$.

Step 1: Creating a quotient system. First, a quotient system $\mathcal{S}_1$ of $\mathcal{S}$ is created using $\mathcal{R}^{\max}$ as follows.

Definition 8.1 (Alternating simulation quotient)**.** *The system $\mathcal{S}_1 := (\mathcal{X}_1, \mathcal{X}_{0,1}, \mathcal{U}_1, \mathcal{E}_1, \mathcal{Y}, H_1)$ is called the* alternating simulation quotient *of $\mathcal{S}$ w.r.t. $\mathcal{R}^{\max}$ iff $\mathcal{X}_1 = \mathcal{P}$, $\mathcal{X}_{0,1} = \{Q \mid Q \in \mathcal{P} \wedge \exists q \in Q.\ q \in \mathcal{X}_0\}$, $\mathcal{U}_1 = \mathcal{U}$, $\mathcal{E}_1 = \{(Q,u,Q') \mid \exists q \in Q.\ \exists q' \in Q.\ (q,u,q)' \in \mathcal{E}\}$, $\forall Q \in \mathcal{X}_1.H_1(Q) = H(q)$ for any $q \in Q$ (H_1 is well-defined as $\forall q, q' \in Q.\ H(q) = H(q')$).*

This construction is similar to the celebrated quotient systems used for simulation and bisimulation; here we just make use of the already existing $\mathcal{R}^{\max}$instead of performing a refinement algorithm, like it has been done for simulation equivalence [96]. Step 1 preserves equivalence modulo AS:

Lemma 8.1. $\mathcal{S} \simeq_{\text{AS}} \mathcal{S}_1$.

Let $\mathsf{Part} : \mathcal{X} \longrightarrow \mathcal{X}_1$ be the function that maps every state to its corresponding partition, and $\mathcal{R}_1^{\max} \subseteq \mathcal{X}_1 \times \mathcal{X}_1$ be the smallest relation satisfying (I) $\forall (p,q) \in \mathcal{R}^{\max}.(\mathsf{Part}(p), \mathsf{Part}(q)) \in \mathcal{R}_1^{\max}$ and, (II) $\forall (P,Q) \in \mathcal{R}_1^{\max}.\exists p \in P.\exists q \in Q.(p,q) \in \mathcal{R}^{\max}$. Note that $\forall P, Q \in \mathcal{X}_1.\exists p \in P.\exists q \in Q.(p,q) \in \mathcal{R}^{\max} \Longrightarrow \forall p' \in P.\forall q' \in Q.(p',q') \in \mathcal{R}^{\max}$. This is because every $P, Q \in \mathcal{X}_1$ are sets containing states of $\mathcal{S}$ which are equivalent modulo $\mathcal{R}^{\max}$. Hence, if any element of P is related to any element Q with respect to $\mathcal{R}^{\max}$, then by transitivity of $\mathcal{R}^{\max}$ all elements of P are related to all elements of Q. Hence, (II) implies (III) $\forall (P,Q) \in \mathcal{R}_1^{\max}.\forall p \in P.\forall q \in Q.(p,q) \in \mathcal{R}^{\max}$. The following fact holds:

Lemma 8.2. *(1) $\mathcal{R}_1^{\max}$ is the maximal ASR from $\mathcal{S}_1$ to itself. Moreover, (2) $\mathcal{R}_1^{\max}$ is a partial order.*

In fact, if $\mathcal{R}^{\max}$ is a partial order (i.e., $(p,q) \in \mathcal{R}^{\max} \implies (q,p) \notin \mathcal{R}^{\max}$ for every $p \neq q$), then step 1 does not affect $\mathcal{S}$.

Proposition 8.1. *$|\mathcal{X}_1| \leq |\mathcal{X}|$ and $\mathsf{TranSize}(\mathcal{S}_1) \leq \mathsf{TranSize}(\mathcal{S})$, and no pair P, Q of $\mathcal{X}_1$ is equivalent modulo AS. Moreover, if $\mathcal{R}^{\max}$ is not antisymmetric, then $|\mathcal{X}_1| < |\mathcal{X}|$.*

Step 2: Removing irrational and redundant controller choices. We construct $\mathcal{S}_2 := (\mathcal{X}_2, \mathcal{X}_{0,2}, \mathcal{U}_2, \mathcal{E}_2, \mathcal{Y}_2, H_2)$ from $\mathcal{S}$ as follows. $\mathcal{X}_2 = \mathcal{X}$, $\mathcal{X}_{0,2} = \mathcal{X}_0$, $\mathcal{U}_2 = \mathcal{U}$, $\mathcal{Y}_2 = \mathcal{Y}$, $H_2 = H$. Before defining $\mathcal{E}_2$, we define an ordering $\sqsubseteq_{\mathcal{S}}$ on elements of $\mathcal{X} \times \mathcal{U}$: $(p',u') \sqsubseteq_{\mathcal{S}} (p,u) \iff u \in U(p) \wedge \forall (p,u,q) \in \mathcal{E}.\exists (p',u',q') \in \mathcal{E}.(q',q) \in \mathcal{R}^{\max}$. Note that $\sqsubseteq$ is a transitive relation. We say that an action u' is an *irrational move at a state p* of an FTS $\mathcal{S}$ iff $\exists u.(p,u) \sqsubseteq_{\mathcal{S}} (p,u') \wedge \neg((p,u') \sqsubseteq_{\mathcal{S}} (p,u))$. State p in this case is said to have *irrational moves*. Similarly, we say that u, u' are *equally rational* at a state p of an FTS iff $(p,u) \sqsubseteq_{\mathcal{S}} (p,u') \wedge ((p,u') \sqsubseteq_{\mathcal{S}} (p,u))$. Moreover, if u and u' are distinct then the state p, in this case, is said to have *redundant moves*. We construct $\mathcal{E}_2$ by removing all the transitions on irrational actions at p. Followed by this, we make available only one of the equally rational actions. This procedure preserves equivalence modulo AS. Let $\mathcal{I} : \mathcal{X} \longrightarrow \mathcal{X}$ be the identity function.

Lemma 8.3. *$\mathcal{S}_2 \preceq_{\mathcal{I}} \mathcal{S} \preceq_{\mathcal{R}^{\max}} \mathcal{S}_2$. Hence, $\mathcal{S} \simeq_{\mathrm{AS}} \mathcal{S}_2$.*

Proposition 8.2. *$|\mathcal{X}_2| = |\mathcal{X}|$, $\mathsf{TranSize}(\mathcal{S}_2) \leq \mathsf{TranSize}(\mathcal{S})$, and for every state $q \in \mathcal{X}_2$, $\mathcal{U}_2(q)$ only contains non-redundant rational actions. Moreover, if there are irrational or redundant actions available from any state q in $\mathcal{S}$, then $\mathsf{TranSize}(\mathcal{S}_2) < \mathsf{TranSize}(\mathcal{S})$.*

Step 3: Eliminating Irrational Choices for Environment. We construct $\mathcal{S}_3 := (\mathcal{X}_3, \mathcal{X}_{0,3}, \mathcal{U}_3, \mathcal{E}_3, \mathcal{Y}_3, H_3)$ from $\mathcal{S}$ as follows. $\mathcal{X}_3 = \mathcal{X}$, $\mathcal{U}_3 = \mathcal{U}$, $\mathcal{Y}_3 = \mathcal{Y}$, $H_3 = H$. Before the construction of $\mathcal{X}_{0,3}$ and $\mathcal{E}_3$ we define a new relation amongst transitions: any transition (p, u, q') in $\mathcal{E}$ is called a *younger sibling* of a transition (p, u, q) in $\mathcal{E}$ with respect to $\mathcal{S}$ iff $(q, q') \in \mathcal{R}^{\max} \wedge (q', q) \notin \mathcal{R}^{\max}$. Similarly, an initial state q_0' is called a younger sibling of yet another initial state q_0 with respect to $\mathcal{S}$ iff $(q_0, q_0') \in \mathcal{R}^{\max} \wedge (q_0', q_0) \notin \mathcal{R}^{\max}$. Then, $\mathcal{X}_{0,3}$ and $\mathcal{E}_3$ are constructed from $\mathcal{X}_0$ and $\mathcal{E}$ by deleting all the younger siblings. In other words, given any state p and $u \in U(q)$, if there are two transitions (p, u, q') and (p, u, q) in $\mathcal{E}$ and if $q \preceq_{\mathrm{AS}} q'$ but not vice-versa (i.e., q is strictly more advantageous position for the environment as compared to q') then delete the transition (p, u, q') from $\mathcal{S}$, as the environment has no reason to choose q' over q. Note that this definition is similar to the *younger brother* definition of [96], but here we need to take the label of the transitions into account while defining the "sibling" relationship due to the definition of AS.

Lemma 8.4. *$\mathcal{S} \preceq_{\mathcal{I}} \mathcal{S}_3 \preceq_{\mathcal{R}^{\max}} \mathcal{S}$. Thus, $\mathcal{S}_3 \simeq_{\mathrm{AS}} \mathcal{S}$.*

Proposition 8.3. *$|\mathcal{X}_3| = |\mathcal{X}|$, $\mathsf{TranSize}(\mathcal{S}_3) \leq \mathsf{TranSize}(\mathcal{S})$, and $\mathcal{S}_3$ contains no transitions or initial states that are younger siblings of another transition or initial state, respectively. Moreover, if there is any younger sibling transition or initial state in $\mathcal{S}$, then $\mathsf{TranSize}(\mathcal{S}_3) < \mathsf{TranSize}(\mathcal{S})$.*

Step 4: Removing states inaccessible from initial state set $\mathcal{X}_0$ in $\mathcal{S}$. Let $\mathcal{X}_\infty$ be the set of such states inaccessible from any initial state in $\mathcal{X}_0$. Then $\mathcal{S}_4 := (\mathcal{X}_4, \mathcal{X}_0, \mathcal{U}, \mathcal{E}_4, \mathcal{Y}, H)$, where $\mathcal{X}_4 = \mathcal{X} \setminus \mathcal{X}_\infty$, $\mathcal{E}_4 = \mathcal{E} \cap (\mathcal{X}_4 \times \mathcal{U}_4 \times \mathcal{X}_4)$.

Lemma 8.5. *$\mathcal{S}_4 \cong_{AS} \mathcal{S}$*

Proposition 8.4. *$|\mathcal{X}_4| \leq |\mathcal{X}|$, $\mathsf{TranSize}(\mathcal{S}_4) \leq \mathsf{TranSize}(\mathcal{S})$, and all states in $\mathcal{S}_4$ are accessible from $\mathcal{X}_{0,4}$. Moreover, if $\mathcal{X}_\infty$ is non-empty then $|\mathcal{X}_4| < |\mathcal{X}|$.*

The combination of Lemmas 8.1, 8.3, 8.4 and 8.5 gives our main correctness result:

Theorem 8.1. *$\mathcal{S} \simeq_{\mathrm{AS}} S^4(S^3(S^2(S^1(\mathcal{S}))))$.*

8.2.3 Optimality results

We have seen that our algorithm generates a system that is equivalente modulo ASE to its input system. Here we present the results showing that its output has minimal size, and in fact this minimal system is unique up to a special notion of isomorphism.

Theorem 8.2 (Necessary condition for minimality modulo ASE)**.** *Given any FTS $\mathcal{S}$, a minimal FTS $\mathcal{S}_{\mathsf{min}} := (\mathcal{X}_{min}, \mathcal{X}_{0,min}, \mathcal{U}_{min}\mathcal{E}_{min}, \mathcal{Y}_{min}, H_{min})$ equivalent to the former modulo AS necessarily satisfies the following conditions:*

N_1 *For any* $p, q \in \mathcal{X}_{min}$, $(\mathcal{S}_{\mathsf{min}}(p) \simeq_{\mathrm{AS}} \mathcal{S}_{\mathsf{min}}(q)) \implies p = q$. *That is, no two distinct states are equivalent modulo AS to each other.*

N_2 *For any* $p \in \mathcal{X}_{min}$, p *does not have any irrational or redundant moves.*

N_3 $\nexists t_1, t_2 \in \mathcal{E}_{min}, x_1, x_2 \in \mathcal{X}_{0,min}$ *such that* t_1 *is a younger sibling of* t_2 *or* x_1 *is a younger sibling of* x_2.

N_4 *All the states in* $\mathcal{X}_{min}$ *are connected from some* $x_0 \in \mathcal{X}_{0,min}$.

Proof. This theorem is a consequence of Propositions 8.1, 8.2, 8.3, 8.4. If any condition N_i, $i \in \{1,2,3,4\}$, is violated by $\mathcal{S}_{\mathsf{min}}$, Step i can be applied to get a strictly smaller system preserving equivalence modulo AS which contradicts that $\mathcal{S}_{\mathsf{min}}$ is minimal. □

We call any FTS satisfying the conditions in Theorem 8.2 as *potentially minimal systems*. Our algorithm satisfies such conditions:

Lemma 8.6. $\mathcal{S}_{out} = S^4(S^3(S^2(S^1(\mathcal{S}))))$, *satisfies the necessary conditions in Theorem 8.2.*

By Proposition 8.1, we know that after step 1 we get a $S^1(\mathcal{S})$ that satisfies N_1. The proof (see Section 8.A) then shows that after performing each step i, we get a system satisfying N_i. Moreover, if the input to the system satisfied any of the previous properties, they will continue to respect them.

In the following we show that the conditions in Theorem 8.2 are also sufficient for minimality modulo ASE. In fact, we prove something stronger: such a minimal system is unique up to a variant of isomorphism which we introduce as bijective alternating bisimulation isomorphism (BABI). We show this by proving that any two potentially minimum systems $\mathcal{S}_1$ and $\mathcal{S}_2$ such that $\mathcal{S}_1 \simeq_{\mathrm{AS}} \mathcal{S}_2$ implies that they are BABI to each other. It is important to note that for two structures to be connected via a BABI implies the existence of a bijective alternating bisimulation relation, but the converse is not necessarily true. Hence, the former is stricter than the latter. In fact, the existence of a bijective alternating bisimulation does not necessarily preserve the transition size [4].

Definition 8.2 (Bijective alternating bisimulation isomorphism)**.** *Given any two systems* $\mathcal{S}_j := (\mathcal{X}_j, \mathcal{X}_{0,j}, \mathcal{U}_j \mathcal{E}_j, \mathcal{Y}_j, H_j)$, $j \in \{1,2\}$, *we say that* $\mathcal{S}_1 \cong_{\mathsf{is}} \mathcal{S}_2$ *iff there exists a bijective function* $\mathcal{A} : \mathcal{X}_1 \to \mathcal{X}_2$ *such that* $\forall p \in \mathcal{X}_1 . \mathcal{A}(p) = q$ *implies:*

(1) $p \in \mathcal{X}_{0,1} \iff q \in \mathcal{X}_{0,2}$.

(2) $H_1(p) = H_2(q)$. *Output maps are preserved.*

(3) There exists a bijection $G_{p,q} : \mathcal{U}_1(p) \to \mathcal{U}_2(q)$ *such that* $\forall a \in \mathcal{U}_1(p). \mathrm{Post}^{\mathcal{S}_2}(q, b) = \{\mathcal{A}(p') \mid p' \in \mathrm{Post}^{\mathcal{S}_1}(p, a)\}$ *where* $b = G_{p,q}(a)$. *That is, input labels can be renamed on a per-state basis, as long as the renaming is one-to-one and reversible.*

Clearly, $\mathcal{S}_1 \cong_{\mathsf{is}} \mathcal{S}_2$ implies $|\mathcal{X}_0| = |\mathcal{X}_1|$ (implied by the existence of the bijection $\mathcal{A}$), $|\mathcal{X}_{0,1}| = |\mathcal{X}_{0,2}|$ (implied by item (1) and bijectivity of $\mathcal{A}$), total number of transitions are equal in both $\mathcal{S}_1, \mathcal{S}_2$ (implied by item (3)). Hence, $|\mathcal{X}_1| = |\mathcal{X}_2|$ and $\mathsf{TranSize}(\mathcal{S}_1) = \mathsf{TranSize}(\mathcal{S}_2)$.

Lemma 8.7. *Let* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *be any potentially minimal systems. Then,* $\mathcal{S}_1 \simeq_{\mathrm{AS}} \mathcal{S}_2$ *implies* $\mathcal{S}_1 \cong_{\mathsf{is}} \mathcal{S}_2$.

[4] As a counterexample, consider two single state systems, one with self loop on a and other with two self loops each on a and $\tilde{a}$.

Lemmas 8.6 and 8.7 imply our main optimality result:

Theorem 8.3. *The system $\mathcal{S}_{out} = S^4(S^3(S^2(S^1(\mathcal{S}))))$ is the unique (up to BABI) minimal system that is ASE to $\mathcal{S}$.*

8.3 Case Study: scheduling PETC systems

Let us now revisit the problem of scheduling multiple PETC systems in a shared network using finite transition systems (Section 3.6), taking a slightly more general version of it. We consider a network containing p PETC control-loops and $C < p$ communication channels with channel occupancy time h, which is also the shared checking time of all PETC loops in the network. Our main goal here is to determine whether there exists a strategy that, at every time $t \in h\mathbb{N}$, given the history of sampled states from all plants, determines which (if any) loops must send their samples to the controller. The number of loops sending their samples must be no greater than C, and, as usual, the PETC triggering times must be respected (only early triggers are allowed to be requested). That is, no controller can miss its PETC-given *deadline*. If a scheduler can be found, we also want to retrieve one such scheduling strategy for real-time implementation. For simplicity and without loss of generality, we again assume for the rest of this chapter that the time units are selected such that $h = 1$.

In the models from the previous chapters, we have relied of the fact that the ETC system is applied on a disturbance-free LTI system; this allows the quotient models (and refinements thereof) that we have proposed. However, when disturbances are present [23] or when considering nonlinear systems [30], it is not possible or computationally feasible to obtain such quotient systems. As, a state in the abstraction q is associated to a region in the state-space $\mathcal{R}_q \subseteq \mathbb{R}^{n_x}$ whose inter-sample time lies in a known interval $\{\tau_q^{\text{low}}, \tau_q^{\text{low}}+1, \ldots \tau_q^{\text{high}}\}$. This effectively constitutes a power simulation, which attains the same properties as approximate simulations (see [21] for details). In the cases presented in the previous chapters, it turns out that we can build the abstraction in such a way that $\tau_q^{\text{low}} = \tau_q^{\text{high}}$ for all q. Nonetheless, like in our case, the abstraction process returns the state set $\mathcal{X}_{\text{abs}}$ and a transition relation $\mathcal{E}_{\text{abs}} \subset \mathcal{X}_{\text{abs}} \times \mathcal{U}_{\text{abs}} \times \mathcal{X}_{\text{abs}}$, where $\mathcal{U}_{\text{abs}} = \{1, 2, ..., \tilde{k}\}$, where $(q, \tau, q') \in \mathcal{E}_{\text{abs}}$ means that q' can be reached from q if the chosen inter-sample time is τ. From this, we derive the following generalized definition of PETC traffic model with discrete clocks, after Model 3.5:

Definition 8.3 (Generalized PETC traffic model with discrete clocks)**.** *Given a maximum initial discrete-time delay $d \in \mathbb{N}$, the generalized discrete-clock PETC traffic model with early triggers is the transition system $\mathcal{S}_{\text{PETC}} := (\mathcal{X}, \mathcal{X}_0, \{\mathtt{w}, \mathtt{s}\}, \mathcal{E}_{\text{wait}} \cup \mathcal{E}_{\text{sched}} \cup \mathcal{E}_{\text{trigger}} \cup \mathcal{E}_0, H)$ where*

- $\mathcal{X} = \{(q, c) \mid q \in \mathcal{X}_{\text{abs}}, c \in \{1, 2, ..., \tau_q^{\text{high}}\}\} \cup \{I_1, I_2, \ldots, I_d\}$,
- $\mathcal{X}_0 = \{I_1\}$,
- $\mathcal{E}_{\text{wait}} = \{(q, c), \mathtt{w}, (q, c+1) \mid (q, c) \in \mathcal{X}_{\text{abs}} \text{ and } c < \tau_q^{\text{high}}\}$,
- $\mathcal{E}_{\text{sched}} = \{(q, c), \mathtt{s}, (q', 0) \mid (q, c) \in \mathcal{X}_{\text{abs}} \text{ and } (q, c, q') \in \mathcal{E}_{\text{abs}}\}$,
- $\mathcal{E}_{\text{trigger}} = \{(q, c), \mathtt{w}, (q', 0) \mid (q, c) \in \mathcal{X}_{\text{abs}} \text{ and } c \geq \tau_q^{\text{low}} \text{ and } (q, c, q') \in \mathcal{E}_{\text{abs}}\}$,
- $\mathcal{E}_0 := \{I_i, \mathtt{s}, (q, 0) \mid i \in \mathbb{N}_{\leq d}, (q, 0) \in \mathcal{X}\} \cup \{I_i, \mathtt{w}, I_{i+1} \mid i \in \mathbb{N}_{<d-1}\}$,
- $H(q, c) = \mathtt{T}$ *if* $c = 0$, *or* $\mathtt{W}$ *otherwise.*

8

As in Section 3.6, the actions w (for wait) and s (for sample) are the scheduler actions; because the spontaneous trigger of a given loop is out of the control of the scheduler, these transitions are considered (adversarial) nondeterminism for the scheduler. This is why the set $\mathcal{E}_{\text{trigger}}$ is a set of sampling transitions, but they occur when the action w is chosen. The output T represents when a transmission has just occurred, while W means that the loop has instead waited. The states I_i represent initialization states.

Our running example, Fig. 8.1, depicts such a simple PETC traffic model with only two regions, disregarding the initial states.[5] This example contains only two regions $\mathcal{R}_0$ and $\mathcal{R}_1$, mapped into q_0 and q_1, respectively, with $\tau_{q_0}^{\text{low}} = \tau_{q_0}^{\text{high}} = 2$, and $\tau_{q_1}^{\text{low}} = 3$ and $\tau_{q_1}^{\text{high}} = 4$. The states $(q_1, 3)$ and $(q_1, 4)$ represent the *triggering phase* of place q_1: even if the scheduler decide to wait, the sampling can occur in any of these states. This reflects the added uncertainty of these models when a quotient-based abstracion is not possible.

8.3.1 A general result on ETC scheduling

Using the reduction in Section 8.2, a first general result can be derived for scheduling of PETC.

Definition 8.4 (Reduced PETC traffic model)**.** *The reduced PETC traffic model of $\mathcal{S}$ in Def. 8.3 is the transition system $\mathcal{S}'_{\text{PETC}} := (\mathcal{X}', \mathcal{X}_0, \{\mathtt{w}, \mathtt{s}\}, \mathcal{E}'_{wait} \cup \mathcal{E}'_{sched} \cup \mathcal{E}'_0, H)$ where*

- $\mathcal{X}' := \{(q, c) \mid q \in Q, c \in \{1, 2, ..., \tau_q^{\text{low}}\}\}$,
- $\mathtt{E}'_{wait} := \{(q, c), \mathtt{w}, (q, c+1) \mid (q, c) \in X' \text{ and } c < \tau_q^{\text{low}}\}$,
- $\mathtt{E}'_{sched} := \{(q, c), \mathtt{s}, (q', 0) \mid (q, c) \in X' \text{ and } (q, c, q') \in \Delta\}$,
- $\mathcal{E}'_0\{I_i, \mathtt{s}, (q, 0) \mid i \in \mathbb{N}_{\leq d}, (q, 0) \in \mathcal{X}'\} \cup \{I_i, \mathtt{w}, I_{i+1} \mid i \in \mathbb{N}_{<d-1}\}$,
- $H(q, c) = \mathtt{T}$ *if* $c = 0$, *or* $\mathtt{W}$ *otherwise.*

The difference between Def. 8.4 and Def. 8.3 is that, in the former, the sampling always happens at most at τ_q^{low} for every q, and that this point in time it is a scheduled sampling. In other words, there is no event-based sampling anymore, but the scheduler may decide to sample at the first moment in which it knows that an event trigger could occur. This is very similar in spirit to *self-triggered control*, where the controller chooses the sampling time by predicting a worst-case situation in which the event-triggered control would occur. Thus, Def. 8.4 can also be regarded as a traffic model for STC systems, again with the addition of optional early sampling. Fig. 8.4 shows the reduced model from Fig. 8.1. The interesting fact is that these two approaches are equivalent from a schedulability perspective:

Proposition 8.5. *The PETC traffic model from Def. 8.3 and its reduced model from Def. 8.4 are alternating-simulation equivalent.*

Proof. Consider the identity relation $\mathcal{I} := \{(x, x) \mid x \in \mathcal{X}\}$ as a trivial ASR from $\mathcal{S}_{\text{PETC}}$ to itself. Take any state $x = (q, \tau_q^{\text{low}})$. Then $\text{Post}(x, \mathtt{s}) \subset \text{Post}(x, \mathtt{w})$, where this subset relation is strict. Therefore, the action w is an irrational action on x, thus we can remove it from x, preserving ASE by Lemma 8.3. This removal renders (q, c) inaccessible for all $c > \tau_q^{\text{low}}$.

[5] All states of the form $(q, 0)$ are unavoidable by the controller from the initial state I_1. That is, after d steps, the controller cannot prevent that one such state $(q, 0)$ has been visited; hence, for all effects of system reduction, the system as described in Def. 8.3 is the same as the examples depicted in Fig. 8.1 and subsquent figures, where all states of the form $(q, 0)$ are marked as initial.

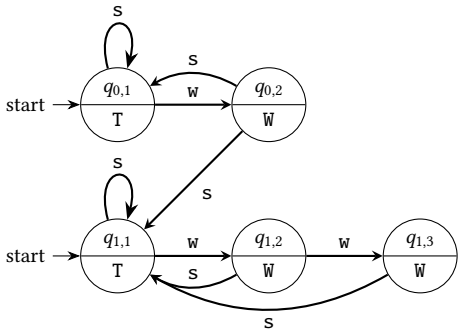


Figure 8.4: Reduced PETC traffic model of Fig. 8.1.

Thus, these states are removed by applying step 4, again preserving ASE by Lemma 8.5. The obtained system is as in Def. 8.4. □

The interpretation of this result is simple: the choice of waiting at time τ_q^{low} has no advantage over sampling, because in the worst case the environment may choose to sample anyway. Hence, from a schedulability perspective, *ETC brings no benefit over a STC-like sampling strategy that chooses to trigger on the earliest ETC triggering time.* Naturally, this general result does not give the minimal system, which depends on the structure of the particular abstraction, as will be illustrated in the next section.

Remark 8.1. *Prop. 8.5 can also be extended to CETC traffic models as those of [21, 30], where the times associated to a region $\mathcal{R}_q$ are an interval subset of $\mathbb{R}_+$ of the form $[\tau_q^{\text{low}}, \tau_q^{\text{high}}]$. The reason is that the results on correctness from Section 8.2 do not rely on finiteness of the transition system (only the ones on optimality do).*

8

8.3.2 Numerical example

Consider p PETC systems of the form (2.2)–(2.4) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} 1 & -4 \end{bmatrix}$$
$$c(s, \boldsymbol{x}, \hat{\boldsymbol{x}}) = |\boldsymbol{x} - \hat{\boldsymbol{x}}| > \sigma |\boldsymbol{x}|, \tag{8.1}$$

with $h = 0.05, \bar{k} = 50$ and $\sigma = 0.7$. Since all systems have the same model, only one traffic abstraction is needed. We use the l-complete models from Chapter 5 for increasing values of l and enhance them with early triggers as in Chapter 7. For a given l, denote the resulting PETC traffic model (Def. 8.3) by $\mathcal{S}_l$. We consider the problem of scheduling on a single channel, and set for each plant $d = 10$ initial time steps for the first sample to be requested.

We implemented our minimization algorithm in Python and ran it on $\mathcal{S}_l, l = 1, 2, 3$. The statistics of the traffic model before and after minimization modulo ASE are displayed in Table 8.1. The additional reduction w.r.t. only step 1 (alternating quotient system) is evident in all cases. The most interesting phenomenon is the striking reduction of the traffic models for $l = 1, 2$ to a system with only 11 states and 21 transitions, which is depicted

Table 8.1: Size of abstractions before and after minimization, and CPU time to minimize the system (in all cases $|X_0| = 1$).

		Original		Quotient		Minimal		CPU
		$\lvert X\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert X\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert X\rvert$	$\lvert\mathcal{E}\rvert$	time
l	1	153	832	118	571	**11**	**21**	657 ms
	2	518	1879	405	1566	**11**	**21**	8.24 s
	3	683	2412	604	2262	587	2126	15 s

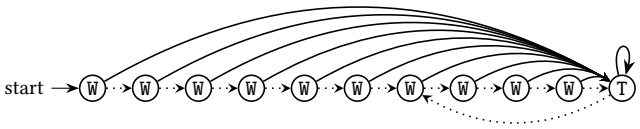


Figure 8.5: Minimized system for the numerical example, $l \in \{1,2\}$. State labels are their outputs, dashed lines are `w` actions and full lines are `s` actions.

in Fig. 8.5. Not only this is a massive reduction which greatly simplifies the scheduling problem, it also informs the user that refining the traffic model by increasing l from 1 to 2 is *irrelevant when it comes to schedulability.* As Fig. 8.5 suggests, these traffic models reduce to a single task with recurring deadline of five steps, after the initial phase. Only with $l = 3$ more complex behavior can be enforced by the scheduler, which becomes apparent by the fact that the minimization is not so impacting: 14% in states and 12% in transitions. This is to be expected because the original systems we abstract are deterministic, and higher values of l reduce the nondeterminism of the abstraction, giving less room for transition elimination in our algorithm. In all cases, the CPU times are within seconds, with an approximately quadratic dependence on the size of the original system. It is worth noting that our Python implementation uses the naive fixed-point algorithm to get the MAS relation, and this step dominated the CPU time of the reduction. Since the times were satisfactory, no performance optimizations were attempted.

Because of the refinement properties of the abstractions $\mathcal{S}_l$ (namely $\mathcal{S}_{l+1} \preceq_{\text{AS}} \mathcal{S}_l \preceq \mathcal{S}_{l+1}$), scheduling with these abstractions is sound but not complete: if p ETC plants are detected to be unschedulable for l, one may still find a schedule using a higher value of l. Thus, we employed the following scenario: first, set $l = 1$ and $p = 2$ and increase p until the systems are unschedulable; then, increase l and try again. We used `ETCetera` to solve the scheduling problem.

Our first attempt used a Python implementation of the composition and safety game solution, where the transitions are encoded with Python dictionaries. Without minimization and with $p = 2$, the scheduling problem took only 801 ms to be concluded, a number close to the 657 ms taken to minimize each system; this is expected, given the quadratic complexity of the minimization algorithm. Impressively, with only $p = 3$ *the scheduling problem without reduction crashed due to memory overflow.*[6] *After performing the minimization, we were able to compute a scheduler for* $p = 5$, a process that took 28.7 min to conclude. With $p = 6$, memory overflow also occurred with the minimal systems.

[6]The experiments were performed in a Intel(R) Xeon(R) W-2145 CPU @ 3.70GHz with 31 GB RAM.

Table 8.2: Scheduler size and CPU time using BDDs.

		Original		Minimal	
p	l	Schedule size	CPU time	Schedule size	CPU time
2	1	3 kB	3 ms	894 B	498 µs
3	1	7.6 kB	8 ms	1.9 kB	783 µs
4	1	19 kB	16 ms	4.2 kB	1.4 ms
5	1	47 kB	36 ms	9.5 kB	2.5 ms
6	1	None	**7.35 s**	None	101 ms
6	2	None	**15.4 min**	None	**84 ms**
6	3	None	35.5 min	None	28.1 min

Our second attempt to solve the scheduling problem used BDDs to encode the transition systems. Table 8.2 summarizes the results of this experiment. As expected, for all cases in $l \in \{1,2\}$ the problem was solved significantly faster with the minimized systems. The difference is much more significant in the non-schedulable cases, which is to be expected because it often requires more iterations in the fixed-point algorithm to detect that no schedule is viable. The difference is particularly massive for $l = 2$, owing to the immense reduction of the system dimensions in this case. For the case with $l = 3$ the time reduction was not as significant as in the aforementioned cases, which is in par with the smaller system size reduction that was obtained in this case.

Reproducing these results. The following Python scripts within `ETCetera` can be executed to reproduce the results presented above:

- `examples/altsim_casestudy_abstractions.py` to generate the traffic abstractions, which are saved as "`pickle`" files in the folder `/saves/`; this process takes about 30 minutes;
- `examples/altsim_casestudy_reductions.py` to run the minimization algorithm on the different abstractions ($l = 1,2,3$) and generate Table 8.1;
- `examples/altsim_casestudy_scheduling_fpiter_min.py` and `examples/altsim_casestudy_scheduling_fpiter_min.py` to generate schedulers on original and minimal systems, respectively, using the naive fixed point iteration; this can take hours and both scripts should be terminated by the operating system due to excessive memory usage as reported above;
- `examples/altsim_casestudy_scheduling_bdd.py` and `examples/altsim_casestudy_scheduling_bdd_min.py` to generate schedulers on original and minimal systems, respectively, using BDDs, and generating the data reported in Table 8.2.

8.4 Conclusion and Future Work

We have revisited the notion of alternating simulation equivalence, and argued about the benefits it can bring for size reduction of finite transition systems in the context of controller synthesis. An algorithm was devised to produce minimal abstractions modulo alternating simulation equivalence. The applicability of these theoretical developments

was then illustrated in the context of scheduling, providing interesting insights for the analysis of schedulability of event triggered systems.

This work opens the door to several further investigations, in particular: (i) extending the ASE notion to weighted transition systems to produce abstractions preserving quantitative properties; (ii) extensions of these same ideas to timed games; (iii) designing on-the-fly versions of the proposed reduction algorithm; and (iv) implementing symbolically the abstraction algorithm employing binary decision diagrams.

8.A Correctness and reduction proofs

Proof of Lemma 8.1. We first observe that if any pair of states $(p,q) \in \mathcal{R}^{\max}$ then for all states p' equivalent to p and q' equivalent to q modulo AS, we have that $(p',q') \in \mathcal{R}^{\max}$ (by transitivity of $\mathcal{R}^{\max}$). Hence $\forall (p,q) \in \mathcal{R}^{\max} \iff \forall p' \in \mathsf{Part}(p), q' \in \mathsf{Part}(q)(p',q') \in \mathcal{R}^{\max}$, where $\mathsf{Part} : \mathcal{X} \to \mathcal{X}_1$ is the function that maps every state to its corresponding partition.

To prove the lemma we show that (A) $\mathcal{R}_1 := \{(p, \mathsf{Part}(q)) \mid (p,q) \in \mathcal{R}^{\max}\}$ is an ASR from $\mathcal{S}$ to $\mathcal{S}_1$ and (B) $\mathcal{R}_2 := \{(\mathsf{Part}(p), q) \mid (p,q) \in \mathcal{R}^{\max}\}$ is an ASR from $\mathcal{S}_1$ to $\mathcal{S}$.

For (A) to be true, $\mathcal{R}_1$ must satisfy requirements (i), (ii) and (iii) from Def. 2.9. Note that $\mathcal{R}_1$ contains the set $\{(p, \mathsf{Part}(p) \mid p \in \mathcal{X}\}$. By construction, for every state $q \in \mathcal{X}_0$, $\mathsf{Part}(q) \in \mathcal{X}_{0,1}$ and for every $q \in \mathcal{X}$, $H(q) = H_1(\mathsf{Part}(q))$, so requirements (i) and (ii) are satisfied. Assume that $\mathcal{R}_1$ does not satisfy requirement (iii), i.e., $\psi := \exists q \in \mathcal{X}.\ \exists u \in U(q).\ \varphi(q,u)$ where $\varphi(q,u) := \forall u_1 \in U_1(\mathsf{Part}(q)).\ \exists Q' \in \mathrm{Post}^{\mathcal{S}_1}(\mathsf{Part}(q), u_1).\ \forall q' \in \mathrm{Post}^{\mathcal{S}}(q,u).\ (q', Q') \notin \mathcal{R}_1$. Now, for any such Q', note that by construction of $\mathcal{S}_1$, $(\mathsf{Part}(q),\ u_1, Q') \in \mathcal{E}_1$ implies that $\exists p \in \mathsf{Part}(q), p' \in Q'.\ (p, u_1, p') \in \mathcal{E}$. However, notice that $(q', Q') \notin \mathcal{R}_1 \implies \forall p' \in Q'\,(q', p') \notin \mathcal{R}^{\max}$. Therefore, $\varphi(q,u)$ implies $\varphi'(q,u) := \forall u_1 \in U_1(\mathsf{Part}(q)).\ \exists Q' \in \mathrm{Post}^{\mathcal{S}_1}(\mathsf{Part}(q), u_1), \exists p' \in Q'.\ \forall q' \in \mathrm{Post}^{\mathcal{S}}(q,u).\ (q', p') \notin \mathcal{R}^{\max}$.

Now let us inspect the set $U_1(\mathsf{Part}(q))$. By definition of $\mathcal{E}_1$, $U_1(\mathsf{Part}(q)) := \{u \mid u \in U(p), (q,p) \in \mathcal{R}^{\max}, (p,q) \in \mathcal{R}^{\max}\}$ Hence, $\forall u_1 \in U_1(\mathsf{Part}(q)) \exists Q' \in \mathrm{Post}^{\mathcal{S}_1}(\mathsf{Part}(q), u_1), \exists p' \in Q'$ can be replaced by $\forall p.(p,q)(q,p) \in \mathcal{R}^{\max} \forall u_1 \in U(p). \exists p' \in \mathrm{Post}^{\mathcal{S}}(p, u_1)$. Applying this and φ' we obtain $\varphi'' := \forall p.(p,q),(q,p) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(p).\ \exists p' \in \mathrm{Post}^{\mathcal{S}}(p,u_1) \forall q' \in \mathrm{Post}^{\mathcal{S}}(q,u).\ (q', p') \notin \mathcal{R}^{\max}$. The formula ψ then implies $\exists q \in \mathcal{X}.\ \exists u \in U(q). \forall p.(p,q),(q,p) \in \mathcal{R}^{\max} \forall u_1 \in U(p).\ \exists p' \in \mathrm{Post}^{\mathcal{S}}(p,u_1).\ \forall q' \in \mathrm{Post}^{\mathcal{S}}(q,u)\ (q', p') \notin \mathcal{R}^{\max}$. Finally, we particularize $\forall p$ to $p = q$ (which is a sound step as $\mathcal{R}^{\max}$ has all reflexive entries, i.e., $(q,q) \in MAS$), change $\mathsf{Part}(q)$ to q (this is implied by transitivity of $\preceq_{\mathrm{AS}}$ and $\simeq_{\mathrm{AS}}$) and rearrange the initial existential quantifiers to get $\exists q.\ \exists u \in U(q). \forall u_1 \in U(q).\ \exists p' \in \mathrm{Post}^{\mathcal{S}}(q,u_1).\ \forall q' \in \mathrm{Post}^{\mathcal{S}}(q,u).\ (q', p') \notin \mathcal{R}^{\max}$. This implies that $\mathcal{R}^{\max}$(that contains all the reflexive pairs) fails to satisfy requirement (iii) of for system $\mathcal{S}$ to itself, which is a contradiction.

For (B), conditions (i) and (ii) also hold trivially. We use contradiction again for condition (iii): suppose $\phi := \exists (P,q) \in \mathcal{R}_2. \exists u_1 \in U_1(P) \forall u \in U(q) \exists q' \in \mathrm{Post}^{\mathcal{S}}(q) \underline{\forall P' \in \mathrm{Post}^{\mathcal{S}_1}(P, u_1).\ (P', q') \notin \mathcal{R}_2}$. By construction of $\mathcal{E}_1$, the underlined subformula is equivalent to $\forall p^* \in P.\, u_1 \in U(p^*) \forall p' \in \mathrm{Post}^{\mathcal{S}}(p^*, u_1).\ (\mathsf{Part}(p'), q') \notin \mathcal{R}_2$. By construction of $\mathcal{R}_2$, and because all states in $\mathsf{Part}(p')$ are equivalent, $(\mathsf{Part}(p'), q') \notin \mathcal{R}_2 \implies (p', q') \notin \mathcal{R}^{\max}$. Hence, the underlined subformula implies $\forall p^* \in P.\, u_1 \in U(p^*) \forall p' \in \mathrm{Post}^{\mathcal{S}}(p^*, u_1).\ (p', q') \notin \mathcal{R}^{\max}$. Substituting in ϕ, it implies $\phi' := \exists (P,q) \in \mathcal{R}_2. \exists u_1 \in U_1(P) \forall u \in U(q) \exists q' \in \mathrm{Post}^{\mathcal{S}}(q) \forall p^* \in P.\, u_1 \in U(p^*) \forall p' \in \mathrm{Post}^{\mathcal{S}}(p^*, u_1).\ (p', q') \notin \mathcal{R}^{\max}$.

Now, note that by construction of $\mathcal{R}_2$, $(P,q) \in \mathcal{R}_2 \implies \exists p \in P.\, (p,q) \in \mathcal{R}^{\max}$. Using this fact and particularizing $\forall p^*.\, u_1 \in U(p^*)$ by p, (which is sound because $u_1 \in U(p)$) ϕ' implies $\exists (p,q) \in \mathcal{R}^{\max}. \exists u_1 \in U_1(P) \forall u \in U(q) \exists q' \in \mathrm{Post}^{\mathcal{S}}(q) \forall p' \in \mathrm{Post}^{\mathcal{S}}(p, u_1)\ .(p', q') \notin \mathcal{R}^{\max}$. This is a contradiction to the fact that $\mathcal{R}^{\max}$ is a maximal ASR from $\mathcal{S}$ to itself. □

Proof of Lemma 8.2. (1) $\mathcal{R}_1^{\max}$ **is an ASR from $\mathcal{S}_1$ to itself**: We show that $\mathcal{R}_1^{\max}$ satisfies all 3 requirements to be an ASR from $\mathcal{S}_1$ to itself. By definition, $\forall P \in \mathcal{X}_1$, $(P,P) \in \mathcal{R}_1^{\max}$. Hence, Requirement 1, trivially holds. As $\mathcal{R}^{\max}$is an AS, $\forall (p,q) \in \mathcal{R}^{\max}.H(p) = H(q)$. Moreover, by definition of $\mathcal{S}_1$, $\forall P \in \mathcal{X}_1. \forall p \in P. H(p) = H'(P)$. By definition (III), $(P,Q) \in \mathcal{R}_1^{\max}$ implies $\forall p \in P, q \in Q. (p,q) \in \mathcal{R}_1^{\max}$, which implies $H'(P) = H'(Q)$ (Requirement 2). As $\mathcal{R}^{\max}$ is an

AS from $\mathcal{S}$ to itself. Hence, $\forall(p,q) \in \mathcal{R}^{\max}.\forall u_1 \in U(p).\exists u_2 \in U(q).\forall(q,u_2,q') \in \mathcal{E}.\exists(p,u_1,p') \in \mathcal{E}.(p',q') \in \mathcal{R}^{\max}$. This along with (I), (II) and (III) implies $\forall(p,q).(\mathsf{Part}(p),\mathsf{Part}(q)) \in \mathcal{R}_1^{\max} \implies [\forall u_1 \in U_1(\mathsf{Part}(p)).\exists u_2 \in U_1(\mathsf{Part}(q))\forall q'.(\mathsf{Part}(q),u_2,\mathsf{Part}(q')) \in \mathcal{E}\exists p'.(\mathsf{Part}(p),u_1,\mathsf{Part}(p')) \in \mathcal{E}_1.(\mathsf{Part}(p'),\mathsf{Part}(q')) \in \mathcal{R}_1^{\max}$ which is equivalent to requirement 3.

$\mathcal{R}_1^{\max}$ is Maximal: Suppose $\mathcal{R}_1^{\max}$ is not maximal. Hence, $\exists P,Q \in \mathcal{X}_1$ such that $\mathcal{S}_1(P) \preceq_{\mathrm{AS}} \mathcal{S}_1(Q)$ but $(P,Q) \notin \mathcal{R}_1^{\max}$. Note that $(P,Q) \notin \mathcal{R}_1^{\max}$ implies $\exists p \in P.\exists q \in Q(p,q) \notin \mathcal{R}^{\max}$. But $\mathcal{S}_1(P) \preceq_{\mathrm{AS}} \mathcal{S}(p)$ and $\mathcal{S}_1(Q) \preceq_{\mathrm{AS}} \mathcal{S}(q)$. By transitivity of $\preceq_{\mathrm{AS}}$, $\mathcal{S}(p) \preceq_{\mathrm{AS}} \mathcal{S}(q)$. This implies, $\mathcal{R}^{\max}$ is not a maximal AS from $\mathcal{S}$ to itself which is a contradiction. (2) As $\mathcal{R}_1^{\max}$ is a maximal alternating simulation relation from $\mathcal{S}_1$ to itself, $\forall P,Q \in \mathcal{X}_1 \mathcal{S}_1(P) \preceq_{\mathrm{AS}} \mathcal{S}_1(Q)$ implies $(P,Q) \in \mathcal{R}_1^{\max}$. The relation $\preceq_{\mathrm{AS}}$ is reflexive and transitive which implies $\mathcal{R}_1^{\max}$ is also reflexive and transitive. Suppose $\mathcal{R}_1^{\max}$ is not anti-symmetric. There exists a distinct pair of states $P,Q \in \mathcal{X}_1$ such that $(P,Q) \in \mathcal{R}_1^{\max} \wedge (Q,P) \in \mathcal{R}_1^{\max}$. This, along with the definition of $\mathcal{R}_1^{\max}$ implies, $\exists p \in P.\exists q \in Q.(p,q) \in \mathcal{R}^{\max} \wedge (q,p) \in \mathcal{R}^{\max}$. But this implies, p and q lie in the same partition. Hence, $P = Q$, which is a contradiction. □

Proof of Prop. 8.1. If $\mathcal{R}^{\max}$ is not anti-symmetric, it has at least one pair of states p,q such that $(p,q) \in \mathcal{R}^{\max}$ and $(q,p) \in \mathcal{R}^{\max}$. While creating the quotient, these states are combined together reducing the number of states by at least 1. Moreover, we never add a new transition in this reduction. □

Proof of Lemma 8.3. To prove (A) the identity map $\mathcal{I} : \mathcal{X} \to \mathcal{X}$ is an ASR from $\mathcal{S}_2$ to $\mathcal{S}$ and (B) $\mathcal{R}^{\max}$ is an ASR from $\mathcal{S}$ to $\mathcal{S}_2$.

(A) Requirements (i) and (ii) are trivially satisfied as $\mathcal{X}_0 = \mathcal{X}_{0,2}$, and $H = H_2$. To prove that requirement (iii). Suppose it does not. Then, $\exists q.\ \exists u \in U_2(q).\ \forall u' \in U(q).\exists q'' \in \mathrm{Post}^{\mathcal{S}}(q,u').\forall(q' \in \mathrm{Post}^{\mathcal{S}_1}(q,u)\ q' \neq q''$ holds. By definition of $\mathcal{E}_2$, for all $q \in \mathcal{X}$, $U(q) \supseteq U_2(q)$ and $\forall u \in U_2(q).q' \in \mathcal{X}\ \mathrm{Post}^{\mathcal{S}_2}(q,u) = \mathrm{Post}^{\mathcal{S}_2}(q,u)$. Just substituting $u' = u$ (in which case we can substitute $\mathrm{Post}^{\mathcal{S}_1} = \mathrm{Post}^{\mathcal{S}}$) (iii) implies **I** is not an AS from S_2 to itself which is a contradiction.

(B) Requirements (i) and (ii) are trivially satisfied as $\mathcal{X}_0 = \mathcal{X}_{0,2}$, and $H = H_2$. To show that requirement (iii) is satisfied, we need to show that $\forall(p,q).\ \in \mathcal{R}^{\max}.\ \forall u_1 \in U(P).\ \exists u_2 \in U_2(q).\ \forall(q,u_2,q') \in \mathcal{E}_2.\ \exists(p,u_1,p') \in \mathcal{E}.\ (p',q') \in \mathcal{R}^{\max}$. Note that $(q,u_2,q') \in \mathcal{E}_2$ condition is within the scope of $\exists u_2.u_2 \in U_2(q)$. By construction, $(q,u_2,q') \in \mathcal{E}_2$ is equivalent to $(q,u_2,q') \in \mathcal{E}$ when $u_2 \in U_2(q)$. Hence, we need to show $\forall(p,q).\ \in \mathcal{R}^{\max}.\ \forall u_1 \in U(P).\ \exists u_2 \in U_2(q).\ \forall(q,u_2,q') \in \mathcal{E}.\ \exists(p,u_1,p') \in \mathcal{E}.(p',q') \in \mathcal{R}^{\max}$. In other words, we need to show $\kappa := \forall(p,q) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(P).\ \exists u_2 \in U_2(q).(p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2)$. We know that $\mathcal{R}^{\max}$ is an ASR from $\mathcal{S}$ to itself. Hence, it satisfies $\phi := \forall(p,q).\ \in \mathcal{R}^{\max}\forall u_1 \in U(p).\ \exists u_2 \in U(q).\ \wedge(p,u_1) \sqsubseteq \mathcal{S}(q,u_2)$. Now observe that $u_2 \in U(q) \iff u_2 \in U_2(q) \vee u_2 \in U(q) \setminus U_2(q)$. By using this identity in the formula ϕ, we get $\gamma = \forall(p,q) \in \mathcal{R}^{\max}.\forall u_1 \in U(p).\exists u_2.[u_2 \in U_2(q) \wedge (p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2)] \vee \underline{[u_2 \in U(q) \setminus U_2(q) \wedge (p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2)]}$. We analyze the underlined sub formula. Note that, by construction of $\mathcal{E}_2$, $u_2 \in U(q) \setminus U_2(q)$ implies $\exists u_2' \in U_2(q)(q,u_2) \sqsubseteq_{\mathcal{S}} (q,u_2')$. Using this implication in the underlined formula, we get $\varphi(p,u_1,q,u_2) = [\exists u_2' \in U_2(q).\ (q,u_2) \sqsubseteq_{\mathcal{S}} (q,u_2') \wedge (p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2)]$. Recall that $\sqsubseteq_{\mathcal{S}}$ is a transitive relation. Hence, $\varphi \equiv \varphi'(p,u_1,q) = [\exists u_2' \in U_2(q).(p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2')]$. Note that φ' no longer contains u_2 as free variable. Hence, substituting the underlined formula with $\varphi'(p,u_1,q)$ we get: $\forall(p,q) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(p).\ \exists u_2.\ [u_2 \in U_2(q) \wedge (p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2)] \vee \underline{[\exists u_2'.u_2' \in U_2(q)(p,u_1) \sqsubseteq_{\mathcal{S}} (q,u_2')]}$. Renaming u_2' as u_2, we get κ. Hence, $\phi \implies \kappa$, which proves the result. □

Proof of Prop. 8.2. If there exists a state $q \in \mathcal{X}$ and distinct $u, u' \in \mathcal{U}$ such that $(q,u) \preceq \mathcal{R}^{\max}(q,u')$ then either u is an irrational action or, both u and u' are equally rational at state q. In both the cases the transitions outgoing from q labelled u' will be removed. Hence $|\mathcal{E}'| < |\mathcal{E}|$. Note that removing irrational and redundant actions do not introduce new irrational or redundant actions. As $\mathcal{E}_2$ is constructed from $\mathcal{E}$ by removing all the transitions corresponding to irrational and redundant actions the former contain only rational moves at every state. □

Proof of Lemma 8.4. Let Q_{ys} and $\mathcal{E}_{ys}$ be set of all the initial states and transitions, respectively, which are younger siblings. Hence, $\mathcal{X}_{0,2} = \mathcal{X}_0 \setminus Q_{ys}$ and $\mathcal{E}_2 = \mathcal{E} \setminus \mathcal{E}_{ys}$. We show (A)Identity map $\mathcal{I} : \mathcal{X} \longrightarrow \mathcal{X}$ is an AS from $\mathcal{S}$ to $\mathcal{S}_2$ (B)$\mathcal{R}^{\max}$ is an AS from $\mathcal{S}_2$ to $\mathcal{S}$. (A) As $\mathcal{X}_{0,2} \subseteq \mathcal{X}_0$, Requirement 1 is trivially satisfied. The state set $\mathcal{X}$ and the output map H are same in both $\mathcal{S}_2$ and $\mathcal{S}$ implying satisfaction of requirement 2. Suppose requirement 3 is not satisfied. $\exists (p,p) \in \mathcal{I}.\ \exists u_1 \in U(p).\ \forall u_2 \in U(p).\ \exists (p,u_2,p') \in \mathcal{E}_2.\ \forall (p,u_1,p'') \in \mathcal{E}.\ p' \neq p''$. But $\mathcal{E}_2 \subseteq \mathcal{E}$. This leads to a contradiction (as it implies that a transition exists in $\mathcal{E}_2$ but not in $\mathcal{E} \supseteq \mathcal{E}_2$).

(B) We now show that $\mathcal{R}^{\max}$ is an AS from $\mathcal{S}_2$ to $\mathcal{S}$. We need to show that for every initial state $q \in \mathcal{X}_0$ there exists an initial state $q' \in \mathcal{X}_0$ such that $(q',q) \in \mathcal{R}^{\max}$. As $\mathcal{R}^{\max}$ is reflexive, we have that for every initial state $q \in \mathcal{X}_0 \setminus Q_{ys}$ there exists an initial state $q \in \mathcal{X}_0$ such that $q,q \in \mathcal{R}^{\max}$. For $q \in Q_{ys}$ there exists a $q' \in \mathcal{X}_0$ (the "elder sibling") such that $(q',q) \in \mathcal{R}^{\max}$. Requirement 2 trivially holds as the set of states and output map is identical in both $\mathcal{S}_2$ and $\mathcal{S}$. For proving requirement 3, recall that $\mathcal{R}^{\max}$ is an AS from $\mathcal{S}$ to itself. Hence, $\varphi = \forall (p,q) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(p).\ \exists u_2 \in U(q).\ \forall q'.\ (q,u_2,q') \in \mathcal{E} \longrightarrow\ \exists p'.(p,u_1,p') \in \mathcal{E} \wedge (p',q') \in \mathcal{R}^{\max}$. As $\mathcal{E}_2 = \mathcal{E} \setminus \mathcal{E}_{ys}$, $\exists p'.\ (p,u_1,p') \in \mathcal{E}$ is equivalent to $\{\exists p'.\ (p,u_1,p') \in \mathcal{E}_{ys}\} \vee \{\exists p'.\ (p,u_1,p') \in \mathcal{E}_2\}$. Thus, we get, $\varphi' = \forall (p,q) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(p).\ \exists u_2 \in U(q).\ \forall q'.\ (q,u_2,q') \in \mathcal{E} \longrightarrow [\{\exists p'.\ (p,u_1,p') \in \mathcal{E}_2 \wedge (p',q') \in \mathcal{R}^{\max}\} \vee \{\exists p'.\ (p,u_1,p') \in \mathcal{E}_{ys} \wedge (p',q') \in \mathcal{R}^{\max})\}]$. $(p,u_1,p') \in \mathcal{E}_{ys} \wedge (p',q') \in \mathcal{R}^{\max}$ implies(by existence of "elder brother" for p,u_1,p') $\exists p''.\ (p,u_1,p'') \in \mathcal{E}_2 \wedge (p,u_1,p') \in \mathcal{E}_{ys} \wedge (p'',p') \wedge (p',q')$ implies(by transitivity of $\preceq_{\mathrm{AS}}$) $\{\exists p''.(p,u_1,p'') \in \mathcal{E}_2 \wedge (p'',q')\}$. Finally substituting this implication in φ' we get:

$$\forall (p,q) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(p).\ \exists u_2 \in U(q).\ \forall q'.(q,u_2,q') \in \mathcal{E}.\ \ [\{\exists p'.\ (p,u_1,p') \in \mathcal{E}_2 \wedge (p',q') \in \mathcal{R}^{\max}\} \vee \underline{\{\exists p'.\ \exists p''.\ (p,u_1,p'') \in \mathcal{E}_2 \wedge (p'',q')\}}].$$

Removing $\exists p'$ from the underlined formula as there is no reference to p' in the formula within this quantifier,

$$\forall (p,q) \in \mathcal{R}^{\max}.\forall u_1 \in U(p).\ \exists u_2 \in U(q).\ \forall q'.\ (q,u_2,q') \in \mathcal{E} \longrightarrow [\{\exists p'.\ (p,u_1,p') \in \mathcal{E}_2 \wedge (p',q') \in \mathcal{R}^{\max}\} \vee \underline{\{\exists p''.\ (p,u_1,p'') \in \mathcal{E}_2 \wedge (p'',q') \in \mathcal{R}^{\max})\}}].$$

Renaming p'' to p' in the underlined subformula and applying idempotent ($\phi \vee \phi \equiv \phi$), we get

$$\forall (p,q) \in \mathcal{R}^{\max}.\ \forall u_1 \in U(p).\ \exists u_2 \in U(q).\ \forall q'.\ (q,u_2,q') \in \mathcal{E} \longrightarrow \exists p'.\ (p,u_1,p') \in \mathcal{E}_2 \wedge (p',q') \in \mathcal{R}^{\max},$$

which is Requirement 3. □

Proof of Prop. 8.3. If there exists a transition (or an initial state) that is a younger sibling, we remove that transition from $\mathcal{S}$ (or make the state non-initial) reducing $\mathsf{TranSize}(S)$ by at least 1. Note that we do not add or remove any states. Hence, the state size is not affected. By construction, we eliminate all the transitions or initial states which are younger siblings of another transition or initial state, respectively. Note that, removing a younger sibling does not add new younger siblings. □

Proof of Lemma 8.5. We show that the identity map $\mathcal{I} : \mathcal{X}_4 \to \mathcal{X}_4$ satisfies all the requirements for being an ASR from $\mathcal{S}$ to $\mathcal{S}_4$ and vice-versa (note that $\mathcal{I} = \{(x,x) \mid x \in \mathcal{X}_4\}$ is a subset of both $\mathcal{X} \times \mathcal{X}_4$ and $\mathcal{X}_4 \times \mathcal{X}$, hence it is a valid relation in both directions). Requirement (i) trivially holds (for both directions) as the initial state sets are the same. Requirement (ii) holds (for both directions) as the output map is the same and $\mathcal{I}$ is an identity function. Suppose requirement (iii) does not hold. That is, $\exists (p,p) \in \mathcal{I}.\exists u_1 \in U(p).\forall u_2 \in U_4(p).\exists (p,u_2,p') \in \mathcal{E}_4.\forall (p,u_1,p'') \in \mathcal{E}.p' \neq p''$. This is a contradiction since $\mathcal{E}_4 \subseteq \mathcal{E}$ (by construction) and the statement implies that there exists a transition in $\mathcal{E}_4$ not present in $\mathcal{E}$. For the inverse direction, assume again that requirement (iii) does not hold. That is, $\exists (p,p) \in \mathcal{I}.\ \exists u_1 \in U_4(p).\ \forall u_2 \in U(p).\ \exists (p,u_2,p') \in \mathcal{E}.\ \forall (p,u_1,p'') \in \mathcal{E}_4.\ p' \neq p''$. If $p' \in \mathcal{X}_4$, it leads to a contradiction (with $u_2 = u_1$) since $\mathcal{E}_4$ contains all elements of $\mathcal{E}$ where the source and the target states are in $\mathcal{X}_4$. If $p' \notin \mathcal{X}_4$, then by definition p is not reachable from any state in $\mathcal{X}_4$ which again is a contradiction as $p \in \mathcal{X}_4$. □

Lemma 8.8. *Let $\mathcal{S}, \mathcal{S}'$ be any two FTSs with the same state set $\mathcal{X}$. Let $\mathcal{R}^{\max}$ be the maximal ASR from $\mathcal{S}$ to itself, and $\mathcal{I} : \mathcal{X} \to \mathcal{X}$ be an identity function. If $\mathcal{S}' \preceq_{\mathcal{I}} \mathcal{S} \preceq_{\mathcal{R}^{\max}} \mathcal{S}'$ (or $\mathcal{S} \preceq_{\mathcal{I}} \mathcal{S}' \preceq_{MAS} \mathcal{S}$), then $\mathcal{R}^{\max}$ is the maximal ASR from $\mathcal{S}'$ to itself.*

Proof. Given $\mathcal{S}' \preceq_{\mathcal{I}} \mathcal{S} \preceq_{MAS} \mathcal{S}'$. Hence, $\forall (p,q) \in MAS.\mathcal{S}'(P) \preceq_{\mathrm{AS}} \mathcal{S}(P) \preceq_{\mathrm{AS}} \mathcal{S}(Q)$. This implies (1) $\mathcal{S}' \preceq_{\mathcal{R}_1^{\max}} \mathcal{S}'$ (by transitivity). As $\mathcal{R}^{\max}$ is also a maximal relation from $\mathcal{S}$ to itself, it contains all the reflexive pairs i.e. pairs of the form (p,p). Hence, $\forall p \in \mathcal{X}.\mathcal{S}'(p) \preceq_{\mathrm{AS}} \mathcal{S}(p) \preceq_{\mathrm{AS}} \mathcal{S}'(p)$. Hence, (2) $\forall p \in \mathcal{X}.\mathcal{S}(p) \simeq_{\mathrm{AS}} \mathcal{S}'(p)$. We now show that $\mathcal{R}^{\max}$ is the maximal relation satisfying (1). Suppose it is not. Then, there exists $(p',q') \notin \mathcal{R}_1^{\max}$ such that $\mathcal{S}'(p') \preceq_{\mathrm{AS}} \mathcal{S}'(q')$. By (2) $\mathcal{S}(p') \simeq_{\mathrm{AS}} \mathcal{S}'(p') \preceq_{\mathrm{AS}} \mathcal{S}'(q') \simeq_{\mathrm{AS}} \mathcal{S}(q')$. This implies $\mathcal{S}(p') \preceq_{\mathrm{AS}} \mathcal{S}(q')$. But this is a contradiction as $(p',q') \notin \mathcal{R}^{\max}$ and $\mathcal{R}^{\max}$ is a maximal ASR from $\mathcal{S}$ to itself. (3) Hence, $\mathcal{R}^{\max}$ is the maximal ASR from $\mathcal{S}'$ to itself.

Similar arguments as above prove that, if $\mathcal{S} \preceq_{\mathcal{I}} \mathcal{S}' \preceq_{MAS} \mathcal{S}$ is satisfied, $\mathcal{R}^{\max}$ is a maximal alternating simulation from $\mathcal{S}'$ to itself. □

Proof of Lemma 8.6. Let $\mathcal{T}_0 = \mathcal{S}$ and $\mathcal{T}_i = S^i(\mathcal{T}_{i-1})$. Let $\mathcal{T}_i := (\mathcal{X}_i, \mathcal{X}_{0,i}, \mathcal{U}_i, \mathcal{Y}_i, \mathcal{E}_i, H_i)$. By construction of $\mathcal{T}_2, \mathcal{T}_3$. $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}_3$. Let $\mathcal{X} = \mathcal{X}_1$. Moreover, recall that $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}_3$.

By Proposition 8.1, we know that $S^1(\mathcal{S})$ satisfies N_1 and has a maximal alternating simulation relation $\mathcal{R}_1^{\max}$ to itself which is anti-symmetric. Note that former is equivalent to latter.

By Proposition 8.1 $S^2(S^1(\mathcal{S}))$ satisfies N_2 and By Lemma 8.3, $S^2(S^1(\mathcal{S})) \preceq_{\mathcal{I}} S^1(\mathcal{S}) \prec_{\mathcal{R}_1^{\max}} S^2(S^1(\mathcal{S}))$. By Lemma 8.8, latter implies $\mathcal{R}_1^{\max}$ is the maximal alternating relation from $S^2(S^1(\mathcal{S}))$ to itself. This implies that it satisfies condition N_1 too.

By Prop. 8.3, $\mathcal{T}_3$ satisfies N_3. By Lemma 8.4, $S^2(S^1(\mathcal{S})) \preceq_{\mathcal{I}} S^3(S^2(S^1(\mathcal{S})) \preceq_{\mathcal{R}_1^{\max}} S^2(S^1(\mathcal{S}))$. By Lemma 8.8, $\mathcal{R}_1^{\max}$ is a maximal alternating simulation from $S^3(S^2(S^1(\mathcal{S}))$ to itself;

moreover, $\mathcal{R}_1^{\max}$ is already shown to be anti-symmetric. Hence, $\mathcal{T}_3 = S^3(S^2(S^1(\mathcal{S}))$ satisfies N_1. We now show that $\mathcal{T}_3$ satisfies N_2. We only delete non-deterministic transitions on each action to get $\mathcal{T}_3$ from $\mathcal{T}_2$ Hence, (1) $\forall p \in \mathcal{X}\ U_2(p) = U_3(p)$. Moreover, to get $\mathcal{T}_3$ we only delete transitions which are younger siblings. Hence, (2) $\forall p \in \mathcal{X}.\forall a \in U_2(p)$ every state $p' \in \text{Post}_{\mathcal{T}_2}(p,a)$ alternately simulates some state $p'' \in \text{Post}_{\mathcal{T}_3}(p,a)$ (where $p'' = p'$ if (p,a,p'') was not deleted in $\mathcal{T}_3$ else p'' is such that (p,a,p'') is an elder sibling to (p,a,p')). Suppose $\mathcal{T}_3$ has either an irrational action or redundant action at state p. This implies (3) there exist distinct u, u' such that $(p,u') \sqsubseteq_{\mathcal{T}_3} (p,u)$. In other words, every state $q \in \text{Post}_{\mathcal{T}_3}(p,u)$ alternately simulates a state $q'' \in \text{Post}_{\mathcal{T}_3}(p,u')$. Moreover, by construction, (4)$\forall p \in \mathcal{X}.\forall b \in U_3(p)$ every state $p' \in \text{Post}_{\mathcal{T}_3}(p,b)$ alternately simulates some state $p'' \in \text{Post}_{\mathcal{T}_3}(p,b)$. Combining (2),(3), (4) by substituting $a = u$ in (2) and $b = u'$ in (4), distinct u, u' imply $(p,u') \sqsubseteq_{\mathcal{T}_2} (p,u)$. This implies $\mathcal{T}_2$ does not satisfy N_2 which results in a contradiction.

Finally, note that, deletion of inaccessible states cannot affect the equivalence modulo alternating simulation equivalence. Moreover, trivially, removing these states do not add new states equivalent to an existing state, add irrational or redundant moves or, add younger siblings to the transition system, hence preserving N_1, N_2, N_3. And by construction, $\mathcal{S}_{out} = S^4(S^3(S^2(S^1(\mathcal{S}))))$ satisfies N_4. Hence, $\mathcal{S}_{out}$, the output of our algorithm, satisfies all the above mentioned conditions. □

Proof of Lemma 8.7. Given potentially minimal systems $\mathcal{S}_j := (\mathcal{X}_j, \mathcal{X}_{0,j},, \mathcal{U}_j, \mathcal{E}_j, \mathcal{Y}_j, H_j)$, $j \in \{1,2\}$, such that $\mathcal{S}_1 \simeq_{\text{AS}} \mathcal{S}_2$ we show that $\mathcal{S}_1 \cong_{\text{is}} \mathcal{S}_2$. As $\mathcal{S}_1 \simeq_{\text{AS}} \mathcal{S}_2$, denote the maximal ASR from $\mathcal{S}_1$ to $\mathcal{S}_2$ by $\mathcal{R}_1^{\max}$ and that from $\mathcal{S}_2$ to $\mathcal{S}_1$ by $\mathcal{R}_2^{\max}$. Let $\mathcal{A} \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ such that $\mathcal{A} := \{(p,q) \mid (p,q) \in \mathcal{R}_1^{\max} \wedge (q,p) \in \mathcal{R}_2^{\max}\} = \mathcal{R}_1^{\max} \cap (\mathcal{R}_2^{\max})^{-1}$. Note that any pair $(p,q) \in \mathcal{A}$ iff $\mathcal{S}_1(p) \simeq_{\text{AS}} \mathcal{S}_2(q)$. We prove the result by showing that $\mathcal{A}$ is a bijection satisfying all the 3 conditions of the Def. 8.2. Condition 2 is straightforward: every pair of states occurring in $\mathcal{A}$ are equivalent modulo Alternating Simulation and hence have identical labels.

Now let us focus on Condition 3. We show that $\mathcal{A}$ is a relation satisfying condition 3 of Def. 8.2. For that, we construct a relation $G_{p,q}$ satisfying condition 3; then we see it is a bijection. Note that any $(p,q) \in \mathcal{A}$ implies (C1) $(p,q) \in \mathcal{R}_1^{\max} \wedge$ (C2) $(q,p) \in \mathcal{R}_2^{\max}$.

Construct a candidate relation $G'_{p,q}$ satisfying the consequent of condition 3 of Def. 8.2. The former implies (C1.1) for every $a \in U_1(p)$ we can *choose* a $b \in U_2(q)$ such that for every state $q' \in \text{Post}^{\mathcal{S}_2}(q,b)$ we can find a state $p' \in \text{Post}^{\mathcal{S}_1}(p,a)$ such that $p' \preceq_{\text{AS}} q'$ $((p',q') \in \mathcal{R}_1^{\max})$. (C2) and (C1.1) together imply (C2.1) for the b chosen in previous step (C1.1) we can *find* an $a' \in U_1(p)$ such that for every state $p'' \in \text{Post}^{\mathcal{S}_1}(p,a')$ we can find a state $q'' \in \text{Post}^{\mathcal{S}_2}(q,b)$ such that $q'' \preceq_{\text{AS}} p''$ $((q'',p'') \in \mathcal{R}_2^{\max}$.

Combining (C1.1 and C2.1) we get (C3.1)$\forall a \in U_1(p).\exists b \in U_2(q).$ $\exists a' \in U_1(p)$ such that for every state $p'' \in \text{Post}^{\mathcal{S}_1}(p,a')$ there exists a state $q'' \in \text{Post}^{\mathcal{S}_2}(q,b)$ such that $(q'',p'') \in \mathcal{R}_2^{\max}$. Moreover, for this q'' we can find a state $p' \in \text{Post}^{\mathcal{S}_1}(p,a)$ such that $(p',q'') \in \mathcal{R}_1^{\max}(p' \preceq_{\text{AS}} q'')$. Hence, by transitivity of alternating simulation pre-order, for every state $p'' \in \text{Post}^{\mathcal{S}_1}(p,a')$ there exists a state $p' \in \text{Post}^{\mathcal{S}_1}(p,a)$ such that $p'' \preceq_{\text{AS}} p'$. Hence, $(p,a) \sqsubseteq_{\mathcal{S}_1} (p,a')$. Thus, if $a \neq a'$ then a is either redundant or an irrational choice for the controller at state p in FTS $\mathcal{S}_1$. This contradicts the assumption that $\mathcal{S}_1$ satisfies condition N_2. Hence, (C4)$a = a'$.

Thus combining (C3.1) and (C4) we get (C3) for any $a \in U_1(p)$ we can find $b \in U_2(q)$ such that for every state in $p' \in \text{Post}^{\mathcal{S}_1}(p,a)$ we can find $q' \in \text{Post}^{\mathcal{S}_2}(q,b)$ such. that $q' \preceq_{\text{AS}} p'$

$((q', p') \in \mathcal{R}_2^{\max})$; at the same time, by (C1.1), for every state $q'' \in \text{Post}^{\mathcal{S}_2}(q, b)$ there exists a state in $p'' \in \text{Post}^{\mathcal{S}_1}(p, a)$ such that $p'' \preceq_{\text{AS}} q''$ $((p'', q'') \in \mathcal{R}_1^{\max})$.

Note that (C3) is equivalent to $\psi(p, q) \coloneqq \forall a \in U_1(p).\ \exists b \in U_2(q).\ \varphi(p, q, a, b)$, where $\varphi(p, q, a, b) \coloneqq \varphi_1(p, q, a, b) \wedge \varphi_2(p, q, a, b)$, $\varphi_1 \coloneqq \forall p' \in \text{Post}^{\mathcal{S}_1}(p, a).\exists q' \in \text{Post}^{\mathcal{S}_2}(q, b).(q', p') \in \mathcal{R}_2^{\max}$, and $\varphi_2 \coloneqq \forall q'' \in \text{Post}^{\mathcal{S}_2}(q, b).\exists p'' \in \text{Post}^{\mathcal{S}_1}(p, a).(p'', q'') \in \mathcal{R}_1^{\max}$. Let $G'_{p,q} \subseteq U_1(p) \times U_2(q)$ such that $(a, b) \in G'_{p,q}$ iff $\varphi(p, q, a, b)$ holds.

Verify that $G'_{p,q}$ satisfies the consequent of condition 3. Note that for every $(a, b) \in G'_{p,q}$ we have that every state in $p' \in \text{Post}^{\mathcal{S}_1}(p, a)$ some state in $q' \in \text{Post}^{\mathcal{S}_2}(q, b)$ such that $q' \preceq_{\text{AS}} p'$ $((q', p') \in \mathcal{R}_2^{\max}$, due to $\varphi_1)$, which in turn, due to φ_2, satisfies $q' \preceq_{\text{AS}} p''$ for some state $p'' \in \text{Post}^{\mathcal{S}_1}(p, a)$ (i.e, $(p'', q') \in \mathcal{R}_1^{\max})$.

Now we prove that $p' = p''$ by contradiction. Suppose that $p' \neq p''$. Then, by transitivity of alternating simulation, $p' \preceq_{\text{AS}} p''$. Hence, transition (p, a, p'') is a younger sibling of transition (p, a, p') which contradicts the assumption that N_3 is satisfied by $\mathcal{S}_1$. Hence (C5) $p' = p''$.

Thus, (C5.1) for any $(a, b) \in G'_{p,q}$, for each $p' \in \text{Post}^{\mathcal{S}_1}(p, a)$ there is a state $q' \in \text{Post}^{\mathcal{S}_2}(q, b)$ such that $p' \preceq_{\text{AS}} q'$ (by φ_1). Moreover, this q' is in turn alternately simulates p' (by C5 and φ_2). Hence, $(p', q') \in \mathcal{A}$. Now we prove that there is a unique q' such that $(p', q') \in \mathcal{A}$. (C5.2) Suppose there exists a $p' \in \text{Post}^{\mathcal{S}_1}(p, a)$ that $\geq$ two distinct states $q', q'' \in \text{Post}^{\mathcal{S}_2}(q, b)$, then by (C5.1) $(p', q') \in \mathcal{A}$ and $(p', q'') \in \mathcal{A}$. This would imply that q' and q'' are equivalent modulo AS. This contradicts the assumption that $\mathcal{S}_2$ satisfies N_1. Hence, for every $p' \in \text{Post}^{\mathcal{S}_1}(p, a)$ there exists a unique $q' \in \text{Post}^{\mathcal{S}_2}(q, b)$ such that $(p', q') \in \mathcal{A}$. By symmetry of condition φ, for every $q' \in \text{Post}^{\mathcal{S}_2}(q, b)$ there exists a unique $p' \in \text{Post}^{\mathcal{S}_1}(p, a)$ such that $(p', q') \in \mathcal{A}$.

This implies (C6) $\text{Post}^{\mathcal{S}_2}(q, b) = \{q' \mid (p', q') \in \mathcal{A} \text{ and } p' \in \text{Post}^{\mathcal{S}_1}(p, a)\}$. Hence, by $\psi(p, q)$ we have (C7) i.e.For any $a \in U_1(p)$ we can find $b \in U_2(q)$ such that $(a, b) \in G'_{p,q}$.

By symmetry, repeating all steps starting from (C2), we get (C8) for any $(p, q) \in \mathcal{A}$ we can construct a relation $G''_{q,p} \subseteq U_2(q) \times U_1(p)$ such that $\text{Post}^{\mathcal{S}_1}(p, a) = \{p' | (q', p') \in \mathcal{A}^{-1} \wedge q' \in \text{Post}^{\mathcal{S}_1}(q, a)\}$, reading (9) $\forall b \in U_2(q).\exists a \in U_1(p).(b, a) \in G''_{q,p}$.

Building the bijection $G_{p,q}$. We now prove that $G_{p,q} \coloneqq G'_{p,q} \cap G''^{-1}_{q,p}$ is a well-defined bijective function such that for any $a \in U_1(p), b \in U_2(q)$, $b = G_{p,q}(a)$ implies $\text{Post}^{\mathcal{S}_2}(q, b) = \{q' \mid (p', q') \in \mathcal{A} \text{ and } p' \in \text{Post}^{\mathcal{S}_1}(p, a)\}$. This proves that $\mathcal{A}$ satisfies the required condition 3.

(10) For $G_{p,q}$ to be a well-defined function, we need to show that for any $a \in U_1(p)$, there is (A) at least 1 and (B) at most 1 $b \in U_2(q)$ such that $(a, b) \in G_{p,q}$; (A) is implied by (C7).

For (B), assume that for distinct $b_1, b_2 \in U_2(q)$ $(a, b_1), (a, b_2) \in G_{p,q}$. By (C6), we get that $\text{Post}^{\mathcal{S}_2}(q, b_1) = \{q' \mid (p', q') \in \mathcal{A} \text{ and } p \in \text{Post}^{\mathcal{S}_1}(p, a)\} = \text{Post}^{\mathcal{S}_2}(q, b_2)$. But this implies that b_1 is a redundant controller choice at state q in FTS $\mathcal{S}_2$ which contradicts N_2 for system $\mathcal{S}_2$. Hence, $G_{p,q}$ is a well-defined function. Applying the same reasoning on $G^{-1}_{q,p} = G^{-1}_{p,q} \cap G''_{q,p}$, we get that $G^{-1}_{q,p}$ is also a well-defined function, proving that $G_{p,q}$ is a bijection.

As $G_{p,q}$ contains elements from $G'_{p,q}$, any $(a, b) \in G_{p,q}$ satisfies (C6). Hence $G_{p,q}$ is the required bijection for condition 3 in Def. 8.2.

$\mathcal{A}$ is a bijection and satisfies condition 1: First we show that every initial state is related to a unique initial state. That is, (C11) $\mathcal{A}_0 \coloneqq \mathcal{A} \cap (\mathcal{X}_{0,1} \times \mathcal{X}_{0,2})$ is a bijection between

$\mathcal{X}_{0,1}$ and $\mathcal{X}_{0,2}$. We first show by contradiction that (C11.1) $\mathcal{A}_0$ is a well-defined function. If it is not, then there exists a state $p \in \mathcal{X}_{0,1}$ such that (C11.2) either p is not related to any state q in $\mathcal{A}_0$ or, (C11.3) $\exists q, q' \in \mathcal{X}_{0,2}.(p,q) \in \mathcal{A}_0 \wedge (p,q') \in \mathcal{A}_0 \wedge q \neq q'$. Note that $\mathcal{R}_2^{\max}$ is an ASR from $\mathcal{S}_2$ to $\mathcal{S}_1$, hence from condition (1) of Def. 2.9, p being an initial state of $\mathcal{S}_1$ implies $\exists q'.(q',p) \in \mathcal{R}_2^{\max}$. Now, (due to similar restrictions imposed by condition (1) for $\mathcal{R}_1^{\max}$ being an ASR from $\mathcal{S}_1$ to $\mathcal{S}_2$) this q' is related with some initial state p' of $\mathcal{S}_1$. Hence, $\exists p'.(p',q') \in \mathcal{R}_1^{\max}$. Now note that if $p' = p$, then (p,q') should be in $\mathcal{A}_0$ (by definition) which contradicts the assumption that (C11.2) holds. If $p' \neq p$, we have $\mathcal{S}_1(p) \preceq_{\text{AS}} \mathcal{S}_2(q')$ $\wedge$ $\mathcal{S}_1(q') \preceq_{\text{AS}} \mathcal{S}_2, (p) \wedge p \neq p'$. Hence, by transitivity of $\preceq_{\text{AS}}$, $p \preceq_{\text{AS}} p'$, $p \neq p'$ and both are initial states. This implies that p is an initial state which is younger sibling of p', which contradicts the assumption that $\mathcal{S}_1$ satisfies condition N_3. Note that to prove $\mathcal{A}_0$ is a bijection, it suffices to show that $\mathcal{A}_0^{-1}$ is a well-defined function, which is a symmetrical proof to that of $\mathcal{A}_0$.

Now we show that (C12) $\mathcal{A}$ is a partial function. That is, every $p \in \mathcal{X}_1$ is mapped to a unique $q \in \mathcal{X}_2$ via $\mathcal{A}$. Suppose it is not, i.e., there exists a state $p \in \mathcal{X}_1$ which is related to two distinct states $q, q' \in \mathcal{X}_2$. Hence, $(p,q), (p,q') \in \mathcal{A}$. By definition of $\mathcal{A}$, we have that $(q,p) \in \mathcal{R}_2^{\max}$ and $(p,q') \in \mathcal{R}_1^{\max}$, implying (by transitivity) that $q \preceq_{\text{AS}} q'$; symmetrically, $(p,q) \in \mathcal{R}_1^{\max}$ and $(q',p) \in \mathcal{R}_2^{\max}$, implying that $q' \preceq_{\text{AS}} q$. Thus, q and q' are equivalent modulo AS which is a contradiction as $\mathcal{S}_2$ satisfies N_1. Symmetrically, $\mathcal{A}^{-1}$ is a partial function relation.

Note that by (C11.1) every initial state is mapped to some initial state. By (C12), every state is mapped to a unique state. Hence, every initial state can only be mapped to a *unique* initial state.

We now show that $\mathcal{A}$ (and by symmetry $\mathcal{A}^{-1}$) is a well-defined function. We already showed that $\mathcal{A}$ (and $\mathcal{A}^{-1}$) are partial functions (C12). It remains to be proved that a state in $\mathcal{X}_1$ can be mapped to at least one state in $\mathcal{X}_2$ under $\mathcal{A}$ (and vice-versa under $\mathcal{A}^{-1}$). We already showed the latter for states in $\mathcal{X}_0$; we now show it for the remaining states. We prove this using contradiction. Assume that there exists a state in $\mathcal{X}_1$ that is not mapped to any state in $\mathcal{X}_2$ under $\mathcal{A}$. Let $\mathcal{P}$ be the set of all such states. As $\mathcal{X}_1$ is a finite set, so is $\mathcal{P}$. Note that by assumption N_4, $\mathcal{S}_1$ does not contain any inaccessible state. Hence, every state in $p \in \mathcal{X}_1$ can be reached from some initial state in $p_0 \in \mathcal{X}_{0,1}$ in $|\mathcal{X}_1|$ or less steps. Let c be the minimum number of steps required to reach the state $p' \in \mathcal{P}$ that is the nearest to the initial state set. That is, no state in $\mathcal{P}$ can be reached in $c-1$ or less steps and there is at least 1 state $p' \in \mathcal{P}$ that is reachable from initial state in c steps. Consider a state $p \in \text{Pre}(p', a)$ for some $a \in U_1$. Because p is reachable in $c-1$ steps, there exists a $q \in \mathcal{X}_2$ such that $(p,q) \in \mathcal{A}$. Now we recover (C5.2): for every $a \in U_1(p).\exists b \in U_2(q).\forall p'' \in \text{Post}^{\mathcal{S}_1}(p,a)$ there exists a unique $q'' \in \text{Post}^{\mathcal{S}_2}(q,b)$ such that $(p'',q'') \in \mathcal{A}$. This implies that for p' too there exists a unique $q' \in \text{Post}^{\mathcal{S}_2}(q,b)$ such that $(p',q') \in \mathcal{A}$. This leads to the contradiction, thus $\mathcal{A}$ is a well-defined function. By symmetry, the same holds for $\mathcal{A}^{-1}$. This implies $\mathcal{A}$ is a bijection. □

9

Self-triggered output-feedback control of LTI systems

This work addresses the problem of predicting the ISTs on a more general control setting, where the controller has partial state information and the system is subject to disturbances. First, we prove that additive noise does not hinder stability of output-feedback PETC of linear time-invariant (LTI) systems. Then we build an STC strategy that estimates PETC's worst-case triggering times. To accomplish this, we use set-based methods, more specifically ellipsoidal sets, which describe uncertainties on state, disturbances and noise. Ellipsoidal reachability is then used to predict worst-case triggering condition violations, ultimately determining the next communication time. The ellipsoidal state estimate is recursively updated using guaranteed state estimation (GSE) methods. The proposed STC is designed to be computationally tractable at the expense of some added conservatism. It is expected to be a practical STC implementation for a broad range of applications; in addition, it is a stepping stone to problems involving ETC traffic prediction, such as scheduling.

This chapter is based on *G. de A. Gleizer and M. Mazo Jr. "Self-triggered output-feedback control of LTI systems subject to disturbance and noise", in Automatica vol. 120, 2020* [106], and partially on *G. de A. Gleizer and M. Mazo Jr. "Self-triggered output feedback control for perturbed linear systems", presented in IFAC NecSys 2018* [107].

9.1 Introduction

Most of this dissertation is dedicated to understanding, measuring, and controlling the timing of ETC systems in the ideal case where disturbances are absent and there is perfect state information available. This chapter is the exception. Here we want to predict the ETC timing behavior of an LTI system in a much more general and realistic setting, where the system is subject to disturbances, measurement noise, and the controller is dynamic, input-feedback. The first step to predicting ISTs of an ETC system is simply one-step-ahead: given the existing information (history of plant's inputs and outputs), when can a given triggering mechanism fire? In particular, what is the *earliest* it can fire? As argued in Chapters 1 and 2, early sampling with respect to ETC is typically a safe choice for control stability and performance. If we are able to predict a worst-case IST from a given ETC, we can build an STC mechanism. This is what is presented in this chapter.

The effects of disturbances, noise, and presence of dynamic controllers with incomplete state information are well studied in ETC [4, 11, 13, 82, 108], but the combined effect of all these characteristics, in particular measurement noise, has received little attention. For STC, these effects are largely unaddressed. One of the earliest works in STC is [14], where perfect state feedback is considered but exogenous disturbances may be present. Cleverly, in [14] Manuel et al do not consider the disturbance in the event prediction, and show that doing so guarantees stability and a finite $\mathcal{L}_\infty$-gain; however, its disturbance rejection is poorer than ETC's, since event-triggering naturally takes disturbances into account. When the control system is not assumed to be a static state-feedback regulator, but take output feedback form, the literature is scarcer. In [109], an observer was developed for self-triggered state-feedback control of LTI systems. For general dynamic output-feedback controllers, the problem was unaddressed until our contribution.

This work has two main contributions: first, we prove that, if a PETC or STC closed-loop LTI system is globally exponentially stable, then it is input-to-state stable with respect to disturbances, measurement noise, and additive perturbations in the triggering condition; second, we devise a method to build self-triggered implementations of controllers subject to unknown but bounded disturbances and measurement noise. The stability results make use of the notion of homogeneous hybrid systems from [110]. The STC design consists of computing a lower bound to the triggering times of the PETC strategy from [11]. Here we use set-theoretic methods for control, namely set-valued reachability (SVR) and guaranteed state estimation (GSE). The state estimator keeps track of a set that contains all possible states in which the plant and controller could be. Reachable sets from the observer state set are then computed for a given sequence of elapsed time instants. At each of these instants, an algorithm checks if there is a point in the reachable set that violates a designed triggering condition. Such a check is conservative but computationally efficient. We hereafter refer to this method as preventive self-triggered control (PSTC), since it is designed to prevent triggers later than the reference PETC. The separation properties of linear systems allow for most of the computations to be carried out offline. We choose ellipsoids for the description of sets, even though other descriptions have been shown to be more effective for general-purpose SVR and GSE (e.g., constrained zonotopes in [111]). One reason for our choice is that the considered triggering functions are quadratic, which simplifies computations when ellipsoids are used. In any case, efficient ellipsoidal SVR and GSE methods are available for linear systems: for SVR we use [112] and [113]; for GSE, we adapt the results from

[114], [115] and [111]. The final algorithm attains similar control performance as PETC, while keeping the advantages of STC and reasonably small computational costs; thus, it is likely to fit most linear control applications. In addition, it is an important first step to the abstraction of ETC/STC systems under this more general case.

Related work. Set-based methods have also been employed for ETC and STC on recent works, such as observer-based state feedback ETC in [116], and ETC and STC for discrete-time systems subject to disturbances and noise in [117]. Conceptually, the latter is the most similar to our work, because of the usage of set-based methods for the disturbance reachability. The major differences are the following: (i) their stability results are for discrete-time systems, which do not immediately provide guarantees for continuous-time systems; (ii) they invoke the novel notion of θ-uniform global asymptotic stability (θ-UGAS), a system theoretic property weaker than input-to-state stability, which is what we use in this paper; (iii) their output-feedback controllers are restricted to observer-based state feedback; and (iv) they introduce new set-based events, while in this work we employ well-established event-triggering mechanisms. In addition, our work is particularly focused on implementation and computational efficiency, aspects that are very briefly touched upon in [117]. In summary, to the best of our knowledge, no available STC strategy takes measurement noise into account for continuous-time systems, nor is it prepared for general forms of output-feedback controllers.

Notation. We denote $A|_{\mathcal{I},\mathcal{J}}$ the sub-matrix of A indexed by the row index set $\mathcal{I} \subseteq \{1,...,n\}$ and the column index set $\mathcal{J} \subseteq \{1,...,m\}$. If $\mathcal{I} = \{1,\dots,n\}$ or $\mathcal{J} = \{1,\dots,m\}$ we use $A|_{\cdot,\mathcal{J}}$ or $A|_{\mathcal{I},\cdot}$, respectively. The set $\mathcal{B}(r)$ denotes a ball of radius $r \geq 0$. For two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ we denote their Minkowski sum as $\mathcal{X}_1 + \mathcal{X}_2 := \{x_1 + x_2 \mid x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2\}$. The (Minkowski) sum of a set $\mathcal{X}$ and a point x is naturally $\mathcal{X} + \{x\}$.

9.2 Preliminaries

9.2.1 Hybrid Dynamical Systems

For stability results, we will model the STC closed-loop system as a hybrid system under the framework of jump–flow systems [118, 119], which allows states to flow on continuous time and/or to jump instantly. In this modeling framework, solutions are defined on the *hybrid time domain*, which is a subset of $\mathbb{R}_+ \times \mathbb{N}_0$ that can be written as $\cup_{i\in\{0,\dots,J\}}([t_i, t_{i+1}] \times \{i\})$, where $J \in \mathbb{N}$ and $0 = t_0 \leq t_1 \leq \dots \leq t_{J+1}$, with J and/or t_{J+1} possibly ∞. A *hybrid signal* χ is a function defined on a hybrid domain. A *hybrid system* is described as follows:

$$\begin{cases} \dot{\chi} = f(\chi, \delta), & (\chi(t,j), \delta(t,j)) \in C \\ \chi^+ = g(\chi, \delta), & (\chi(t,j), \delta(t,j)) \in D \\ \psi = h(\chi, \delta), \end{cases} \tag{9.1}$$

where $\chi(t,j) \in \mathbb{R}^n$ is the state vector, $\delta(t,j) \in \mathbb{R}^{n_\mathrm{d}}$ is an exogenous input, $\psi(t,j) \in \mathbb{R}^{n_\mathrm{y}}$ is the output vector, f, g_i and h are continuous functions with inputs and outputs of appropriate dimensions, and $C \subseteq \mathbb{R}^{n+n_\mathrm{d}}$ and $D_i \subseteq \mathbb{R}^{n+n_\mathrm{d}}$ are closed sets. Following [120] and [110], we say that a pair (χ, δ) is a solution to (9.1) if $\operatorname{dom}\chi = \operatorname{dom}\delta$ and

- for all $j \in \mathbb{N}$ and almost all t such that $(t,j) \in \operatorname{dom}\chi$, the pair satisfies $(\chi(t,j), \delta(t,j)) \in C$ and $\dot{\chi}(t,j) = f(\chi(t,j), \delta(t,j))$;

- for all $(t,j) \in \operatorname{dom} \boldsymbol{\chi}$ such that $(t,j+1) \in \operatorname{dom} \boldsymbol{\chi}$, the pair satisfies $(\boldsymbol{\chi}(t,j), \boldsymbol{\delta}(t,j)) \in \mathcal{D}_i$ and $\boldsymbol{\chi}(t,j+1) = \boldsymbol{g}_i(\boldsymbol{\chi}(t,j), \boldsymbol{\delta}(t,j))$.

Definition 9.1 ($\mathcal{L}_p$ norm, [110]). *For a hybrid signal $\boldsymbol{\psi}$, with domain $\operatorname{dom} \boldsymbol{\psi}$, and a scalar $T \in \mathbb{R}_+$, the T-truncated $\mathcal{L}_p$-norm of $\boldsymbol{\psi}$ is given by*[1]

$$\|\boldsymbol{\psi}_{[T]}\|_p := \left(\sum_{i=1}^{j(T)} |\boldsymbol{\psi}(t_i, i-1)|^p + \sum_{i=0}^{j(T)} \int_{t_i}^{\sigma_i} |\boldsymbol{\psi}(s,i)|^p \, \mathrm{d}s \right)^{\frac{1}{p}}, \tag{9.2}$$

where $j(T) := \max\{k : (t,k) \in \operatorname{dom} \boldsymbol{\psi}, t+k \le T\}$, and $\sigma_i := \min(t_{i+1}, T-i)$. From (9.2), the $\mathcal{L}_p$-norm of $\boldsymbol{\psi}$ is defined as

$$\|\boldsymbol{\psi}\|_p := \lim_{T \to T^*} \|\boldsymbol{\psi}_{[T]}\|_p, \tag{9.3}$$

where $T^* = \sup\{t+j : (t,j) \in \operatorname{dom} \boldsymbol{\psi}\}$ (possibly infinity). The $\mathcal{L}_\infty$ norm is taken by replacing the sums (integrals) in (9.2) by the (essential) suprema.

Definition 9.2. (Global Exponential ISS, [110]) *System (9.1) is exponentially finite-gain input-to-state stable from $\boldsymbol{\delta}$ if there exist positive scalars k, a, and γ such that, for any initial condition $\boldsymbol{x}$ and any $\boldsymbol{\delta} \in \mathcal{L}_\infty$, all solutions to (9.1) satisfy*

$$|\boldsymbol{\chi}(t,j)| \le \max\left\{ k\mathrm{e}^{-a(t+j)}|\boldsymbol{x}|, \gamma \|\boldsymbol{\delta}\|_\infty \right\} \tag{9.4}$$

for all $(t,j) \in \operatorname{dom} \boldsymbol{\chi}$. Moreover, the origin is globally exponentially stable (GES) if (9.4) holds with $\boldsymbol{\delta} \equiv \mathbf{0}$.

Definition 9.3 ($\mathcal{L}_p$ stability, [110]). *Given $p \in [1, +\infty)$, system (9.1) is $\mathcal{L}_p$ stable from $\boldsymbol{\delta}$ to $\boldsymbol{\psi}$ with gain (upper bounded by) $k_p \ge 0$ if there exists a scalar $\beta \ge 0$ such that any solution to (9.1) satisfies*

$$\|\boldsymbol{\psi}\|_p \le \beta|\boldsymbol{x}| + k_p \|\boldsymbol{\delta}\|_p \tag{9.5}$$

for any initial condition $\boldsymbol{x} \in \mathbb{R}^n$ and any $\boldsymbol{\delta} \in \mathcal{L}_p$.

The last definition we need is that of homogeneous hybrid systems of degree zero:

Definition 9.4. (Homogeneous hybrid system, [110]) *The system (9.1) is homogeneous of degree zero if, for any scalar $\lambda > 0$, we have*

$$\begin{aligned} \boldsymbol{f}(\lambda\boldsymbol{\chi}, \mathbf{0}) &= \lambda \boldsymbol{f}(\boldsymbol{\chi}, \mathbf{0}), \forall \boldsymbol{\chi}(t,j) \in C_0, \\ \boldsymbol{g}(\lambda\boldsymbol{\chi}, \mathbf{0}) &= \lambda \boldsymbol{g}(\boldsymbol{\chi}, \mathbf{0}), \forall \boldsymbol{\chi}(t,j) \in \mathcal{D}_0, \end{aligned} \tag{9.6}$$

$$\begin{aligned} \boldsymbol{\chi} \in C_0 &\implies \lambda\boldsymbol{\chi} \in C_0, \\ \boldsymbol{\chi} \in \mathcal{D}_0 &\implies \lambda\boldsymbol{\chi} \in \mathcal{D}_0, \end{aligned} \tag{9.7}$$

where closed sets $C_0, \mathcal{D}_0$ are the projections of C and $\mathcal{D}$ when $\boldsymbol{\delta} \equiv \mathbf{0}$.

We are particularly interested in homogeneous systems that satisfy the following assumption:

[1] As a convention, $\sum_{i=1}^{0} f(i) = 0$.

Assumption 9.1. (Flow and jump sets, [110]) *For system (9.1), there exist scalars L_C and L_D such that, for all $(\boldsymbol{x}, \boldsymbol{d}) \in \mathbb{R}^{n+n_d}$,*

$$(\boldsymbol{x}, \boldsymbol{d}) \in C \implies \boldsymbol{x} \in C_0 + L_C \mathcal{B}(|\boldsymbol{d}|) \tag{9.8a}$$

$$(\boldsymbol{x}, \boldsymbol{d}) \in \mathcal{D} \implies \boldsymbol{x} \in \mathcal{D}_0 + L_D \mathcal{B}(|\boldsymbol{d}|). \tag{9.8b}$$

Homogeneous systems satisfying Assumption 9.1 have a powerful stability property:

Theorem 9.1 ([110])**.** *Let system (9.1) be homogeneous in the sense of Definition 9.4 and Assumption 9.1 hold; then, the following statements are equivalent:*

- *the origin of system (9.1) is GES if $\boldsymbol{\delta} \equiv \mathbf{0}$;*
- *system (9.1) is globally exponentially ISS;*
- *system (9.1) is $\mathcal{L}_p$ stable from $\boldsymbol{\delta}$ to $\boldsymbol{\psi}$.*

9.2.2 Recursive Guaranteed State Estimator

Consider an LTI system of the form:

$$\begin{aligned} \dot{\boldsymbol{\xi}}_p(t) &= \boldsymbol{A}_p \boldsymbol{\xi}_p(t) + \boldsymbol{B}_p \hat{\boldsymbol{v}}(t) + \boldsymbol{E}\boldsymbol{\omega}(t), \\ \boldsymbol{\psi}(t) &= \boldsymbol{C}_p \boldsymbol{\xi}_p(t) + \boldsymbol{\nu}(t), \\ \boldsymbol{\xi}_p(0) &= \boldsymbol{x}_p, \end{aligned} \tag{9.9}$$

where the sub-index p is used to denote plant variables, with $\boldsymbol{\xi}_p(t) \in \mathbb{R}^{n_p}$ as its state, $\hat{\boldsymbol{v}}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}$ as its received control input, $\boldsymbol{\omega}(t) \in \mathcal{W} \subset \mathbb{R}^{n_w}$ as the unknown disturbances, $\boldsymbol{\psi}(t) \in \mathbb{R}^{n_y}$ as the measured output, $\boldsymbol{\nu}(t) \in \mathcal{V} \subset \mathbb{R}^{n_y}$ as the unknown measurement noise, and $\boldsymbol{x}_p \in \mathcal{X}_0 \subset \mathbb{R}^{n_p}$ as its initial state. The following assumptions hold:

Assumption 9.2. *Sets $\mathcal{U}, \mathcal{W}$, and $\mathcal{V}$ are compact, and the pair $(\boldsymbol{A}_p, \boldsymbol{C}_p)$ is observable.*[2]

Let $\mathcal{F}_{\mathcal{U}}$ (resp. $\mathcal{F}_{\mathcal{W}}$) be the set of essentially bounded piecewise continuous functions from $\mathbb{R}_+$ to $\mathcal{U}$ (resp. $\mathcal{W}$). We denote a solution of system (9.9) for initial state $\boldsymbol{x}_p$, input $\hat{\boldsymbol{v}} \in \mathcal{F}_{\mathcal{U}}$, and disturbance $\boldsymbol{\omega} \in \mathcal{F}_{\mathcal{W}}$ by

$$\boldsymbol{\xi}_{\boldsymbol{x}_p \hat{\boldsymbol{v}} \boldsymbol{\omega}}(t) = \mathrm{e}^{\boldsymbol{A}_p t}\boldsymbol{x}_p + \int_0^t \mathrm{e}^{\boldsymbol{A}_p (t-\tau)}(\boldsymbol{B}_p \hat{\boldsymbol{v}}(\tau) + \boldsymbol{E}\boldsymbol{\omega}(\tau))\mathrm{d}\tau.$$

We are interested in computing the set of possible solutions to system (9.9) for sets of initial states, control input trajectories and disturbance trajectories. For that, the following definitions are necessary.

Definition 9.5 (Reachability operator)**.** *Given an initial time t_1, a final time t_2, an initial state set $\mathcal{X}$ and the sets $\mathcal{U}$ and $\mathcal{W}$, the reachability operator* reach$(\cdot)$ *is defined as* $\text{reach}(t_1, t_2, \mathcal{X}, \mathcal{U}, \mathcal{W}) := \{\boldsymbol{\xi}_{\boldsymbol{x}_p \hat{\boldsymbol{v}} \boldsymbol{\omega}}(t_2) : \boldsymbol{\xi}_{\boldsymbol{x}_p \hat{\boldsymbol{v}} \boldsymbol{\omega}}(t_1) \in \mathcal{X}, \boldsymbol{v} \in \mathcal{F}_{\mathcal{U}}, \boldsymbol{\omega} \in \mathcal{F}_{\mathcal{W}}\}$*. Moreover, the output of this operator is denoted as the reachable set.*

A recursive GSE is a set-valued version of a general recursive state estimation and, as such, it follows the same principles. A GSE requires that bounds to input, disturbance and noise signals are known in the form of sets:

[2] $(\boldsymbol{A}_p, \boldsymbol{C}_p)$ could be relaxed to be detectable. The unobservable but stable subspace does not affect the controller, thus one should only consider the observable subspace when implementing the results in this paper.

Assumption 9.3. *There exist known compact sets $\tilde{\mathcal{U}}, \tilde{\mathcal{W}}$ and $\tilde{\mathcal{V}}$ such that $\mathcal{U} \subseteq \tilde{\mathcal{U}}, \mathcal{W} \subseteq \tilde{\mathcal{W}}$ and $\mathcal{V} \subseteq \tilde{\mathcal{V}}$.*

Definition 9.6. (Recursive GSE, [121, Chap. 11]) *Let $\tilde{\mathcal{X}}(t_1|t_1) \ni \xi_{\mathrm{p}}(t_1)$ be an available set estimate of the current state at time t_1. Let $\boldsymbol{y} \coloneqq \boldsymbol{\psi}(t_2)$ be an output measurement obtained at t_2. A recursive GSE has the form*

$$\tilde{\mathcal{X}}(t_2|t_1) = \mathrm{reach}(t_1, t_2, \tilde{\mathcal{X}}(t_1|t_1), \tilde{\mathcal{U}}, \tilde{\mathcal{W}}), \tag{9.10a}$$

$$\mathcal{X}_{\mathrm{y}}(t_2) = \{\boldsymbol{x}_{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{p}}} \mid \exists \boldsymbol{v} \in \tilde{\mathcal{V}} : \boldsymbol{C}_{\mathrm{p}}\boldsymbol{x}_{\mathrm{p}} + \boldsymbol{v} = \boldsymbol{y}\}, \tag{9.10b}$$

$$\tilde{\mathcal{X}}(t_2|t_2) = \tilde{\mathcal{X}}(t_2|t_1) \cap \mathcal{X}_{\mathrm{y}}(t_2). \tag{9.10c}$$

Eq. (9.10a) is the prediction step, simply a reachability operation. Eq. (9.10c) is the update step, where the predicted set is intersected with $\mathcal{X}_{\mathrm{y}}(t_2)$, the set of all possible states that are coherent with the measurement. By construction, $\tilde{\mathcal{X}}(t_2|t_2) \ni \xi_{\mathrm{p}}(t_2)$. The sets above can have arbitrary complexity. Hence, it is common to replace the equations above with inclusions, then restricting the set families to computationally tractable ones.

Throughout this paper, the aforementioned sets will be (outer-approximated by) ellipsoids. This idea dates back to 1968 [114], when possibly the first GSE was proposed. Ellipsoids are described by few parameters – one vector and one symmetric matrix – and are bounded. Since they may be described as quadratic inequalities, they also harmonize well with the quadratic triggering functions generally employed for ETC of LTI systems. Some definitions follow:

Definition 9.7. (Ellipsoid, [112, Chap. 2]) *Let $\boldsymbol{m} \in \mathbb{R}^n$ and $\boldsymbol{M} \in \mathbb{S}^n_+$. An ellipsoid is defined in terms of its support function:*

$$\mathcal{E}(\boldsymbol{m}, \boldsymbol{M}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{l}^{\mathsf{T}}\boldsymbol{x} \leq \boldsymbol{l}^{\mathsf{T}}\boldsymbol{m} + (\boldsymbol{l}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{l})^{1/2}, \forall \boldsymbol{l} \in \mathbb{R}^n\}.$$

Remark 9.1. *In case the ellipsoid is not degenerate ($\boldsymbol{M} \succ \boldsymbol{0}$), it can be described in the well-known inequality form $\mathcal{E}(\boldsymbol{m}, \boldsymbol{M}) = \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x} - \boldsymbol{m})^{\mathsf{T}}\boldsymbol{M}^{-1}(\boldsymbol{x} - \boldsymbol{m}) \leq 1\}$. The degenerate case is flat on some of its semi-axes.*

A closely related set is the elliptical cylinder. The following definition comes from [115], with a small change in notation:

Definition 9.8 (Elliptical Cylinder)**.** *Let $\boldsymbol{M} \in \mathbb{S}^m_{++}$, $\boldsymbol{C} \in \mathbb{R}^{m \times n}$, $m \leq n$, and $\mathrm{rank}(\boldsymbol{C}) = m$. An Elliptical Cylinder is defined as*

$$\mathcal{K}(\boldsymbol{y}, \boldsymbol{M}, \boldsymbol{C}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid (\boldsymbol{C}\boldsymbol{x} - \boldsymbol{y})^{\mathsf{T}}\boldsymbol{M}^{-1}(\boldsymbol{C}\boldsymbol{x} - \boldsymbol{y}) \leq 1\}.$$

Remark 9.2. *If $m < n$, the elliptical cylinder is unbounded. If $m = n$, it trivially resolves to the ellipsoid $\mathcal{E}(\boldsymbol{C}^{-1}\boldsymbol{y}, \boldsymbol{C}^{-1}\boldsymbol{M}\boldsymbol{C}^{-\mathsf{T}})$.*

We use some operations on ellipsoids, namely affine transformations, intersections and Minkowski sums. An affine transformation on an ellipsoid is also an ellipsoid: $\boldsymbol{A}\mathcal{E}(\boldsymbol{m}, \boldsymbol{M}) + \boldsymbol{b} = \mathcal{E}(\boldsymbol{A}\boldsymbol{m} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{M}\boldsymbol{A}^{\mathsf{T}})$. Even though ellipsoids are not closed under Minkowski sums and

intersections, there are methods to tightly outer-approximate them with ellipsoids. Here we use trace-optimal outer-approximations. For the Minkowski sum, one has [112, Chap. 2]:

$$\begin{aligned} \mathcal{E}(\boldsymbol{m}^*, \boldsymbol{M}^*) &\supseteq \mathcal{E}(\boldsymbol{m}_1, \boldsymbol{M}_1) + \mathcal{E}(\boldsymbol{m}_2, \boldsymbol{M}_2) \\ \boldsymbol{m}^* &\coloneqq \boldsymbol{m}_1 + \boldsymbol{m}_2 \\ \boldsymbol{M}^* &\coloneqq (1 + p^{-1})\boldsymbol{M}_1 + (1 + p)\boldsymbol{M}_2 \\ p &\coloneqq \sqrt{\operatorname{Tr}(\boldsymbol{M}_1)\operatorname{Tr}(\boldsymbol{M}_2)^{-1}}. \end{aligned} \tag{9.11}$$

If not empty, the intersection may be outer-approximated by a fusion (see below). We particularly need to compute the intersection between an ellipsoid and an elliptical cylinder.

Definition 9.9 (Fusion). *(Adapted from [115]) A fusion between the ellipsoid $\mathcal{E}(\boldsymbol{m}_1, \boldsymbol{M}_1)$ and the elliptical cylinder $\mathcal{K}(\boldsymbol{y}, \boldsymbol{M}_2, \boldsymbol{C})$ is the ellipsoid $\mathcal{E}_\lambda(\boldsymbol{m}, \boldsymbol{M})$ defined over a parameter $\lambda \in [0, 1)$, such that:*

$$\begin{aligned} &\mathcal{E}_\lambda(\boldsymbol{m}, \boldsymbol{M}) \supseteq \mathcal{E}(\boldsymbol{m}_1, \boldsymbol{M}_1) \cap \mathcal{K}(\boldsymbol{y}, \boldsymbol{M}_2, \boldsymbol{C}) \\ &\boldsymbol{M} = z\boldsymbol{Z}^{-1} \\ &\boldsymbol{Z} = \lambda\boldsymbol{M}_1^{-1} + (1-\lambda)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{M}_2^{-1}\boldsymbol{C} \\ &\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{C}\boldsymbol{m}_1 \\ &z = 1 - \lambda(1-\lambda)\boldsymbol{e}^{\mathsf{T}}(\lambda\boldsymbol{M}_2 + (1-\lambda)\boldsymbol{C}\boldsymbol{M}_1\boldsymbol{C}^{\mathsf{T}})^{-1}\boldsymbol{e} \\ &\boldsymbol{m} = \boldsymbol{Z}^{-1}(\lambda\boldsymbol{M}_1^{-1}\boldsymbol{m}_1 + (1-\lambda)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{M}_2^{-1}\boldsymbol{y}). \end{aligned} \tag{9.12}$$

The parameter λ controls how close the output ellipsoid is to either of its inputs. For $\lambda = 1$, $\mathcal{E}_0(\boldsymbol{m}, \boldsymbol{M}) = \mathcal{E}(\boldsymbol{m}_1, \boldsymbol{M}_1)$; when λ gets close to 0, the output tends to be close to $\mathcal{K}(\boldsymbol{y}, \boldsymbol{M}_2, \boldsymbol{C})$.

Remark 9.3. *The trace of the matrix $\boldsymbol{M}$ is convex over λ, since the trace of the inverse is a convex function [52] and $z \in [0, 1]$ provided the intersection is not empty [115]. This allows the use of bisection or golden search methods to compute λ that minimizes the fusion trace.*

Ellipsoidal reachability

For linear systems with ellipsoidal descriptions of $\mathcal{X}, \mathcal{U}, \mathcal{W}$, and $\mathcal{V}$, ellipsoidal reachability can be used. The concept and techniques are thoroughly explained in [112, Chap. 3]. Its authors developed the Ellipsoidal Toolbox [113], which contains operations to compute reachable sets. In this paper we use the reachable set for the disturbance response $\mathcal{X}_{\mathrm{w}}(t) \coloneqq \operatorname{reach}(0, t, \mathbf{0}, \mathbf{0}, \mathcal{W})$. The Ellipsoidal Toolbox has the tools to compute outer-approximations of $\mathcal{X}_{\mathrm{w}}(t)$, denoted by $\bar{\mathcal{X}}_{\mathrm{w}}(t, \boldsymbol{l})$, that are tight along the ray supported by a given vector $\boldsymbol{l} \in \mathbb{R}^{n_{\mathrm{p}}}$, i.e., $\forall \alpha \in \mathbb{R}, \alpha\boldsymbol{l} \in \bar{\mathcal{X}}_{\mathrm{w}}(t, \boldsymbol{l}) \iff \alpha\boldsymbol{l} \in \mathcal{X}_{\mathrm{w}}(t)$. Overall tighter over-approximations can be obtained by computing $\bar{\mathcal{X}}_{\mathrm{w}}(t, \boldsymbol{l}_i)$ for different input vectors $\boldsymbol{l}_i$ and taking an ellipsoidal outer-approximation of the intersection. Let $\mathcal{L}$ be a pre-specified set of the said vectors. We denote by $\tilde{\mathcal{X}}_{\mathrm{w}}(t)$ such an outer-approximation satisfying $\tilde{\mathcal{X}}_{\mathrm{w}}(t) \supseteq \cap_{\boldsymbol{l} \in \mathcal{L}} \bar{\mathcal{X}}_{\mathrm{w}}(t, \boldsymbol{l})$. Figure 9.1 depicts the sets $\mathcal{X}_{\mathrm{w}}(t)$ and $\tilde{\mathcal{X}}_{\mathrm{w}}(t)$ for a given instant. Methods such as the ones available in the Ellipsoidal Toolbox can be used to compute the intersection outer-approximation.

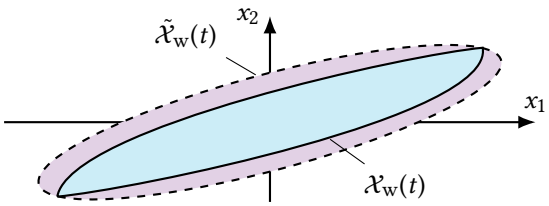


Figure 9.1: Illustration of a reachable set of the disturbance response $\mathcal{X}_{\mathrm{w}}(t)$ and an ellipsoidal outer-approximation $\tilde{\mathcal{X}}_{\mathrm{w}}(t)$.

9.3 Problem definition and stability results

Consider a controller for system (9.9) of the form

$$\begin{aligned} \boldsymbol{\xi}_{\mathrm{c}}(k+1) &= \boldsymbol{A}_{\mathrm{c}}\boldsymbol{\xi}_{\mathrm{c}}(k) + \boldsymbol{B}_{\mathrm{c}}\hat{\boldsymbol{\psi}}(k), \\ \boldsymbol{\upsilon}(k) &= \boldsymbol{C}_{\mathrm{c}}\boldsymbol{\xi}_{\mathrm{c}}(k) + \boldsymbol{D}_{\mathrm{c}}\hat{\boldsymbol{\psi}}(k), \end{aligned} \tag{9.13}$$

where $\boldsymbol{\xi}_{\mathrm{c}}(k) \in \mathbb{R}^{n_{\mathrm{c}}}$ is the controller state, $\boldsymbol{\upsilon}(k) \in \mathbb{R}^{n_{\mathrm{u}}}$ is the computed control command and $\hat{\boldsymbol{\psi}}(k) \in \mathbb{R}^{n_{\mathrm{y}}}$ is the available plant output measurement. The controller runs with period h, so that $t = hk$. The feedback loop is of sample-and-hold form. For two consecutive sampling times k_b and k_{b+1}, $\hat{\boldsymbol{\upsilon}}(t) = \boldsymbol{\upsilon}(k_b), \forall t \in [hk_b, hk_{b+1})$ and $\hat{\boldsymbol{\psi}}(k) = \boldsymbol{\psi}(hk_b), \forall k \in \{k_b, k_b + 1, \ldots, k_{b+1} - 1\}$. The closed-loop system is depicted in Fig. 9.2. We pose the PSTC problem as follows:

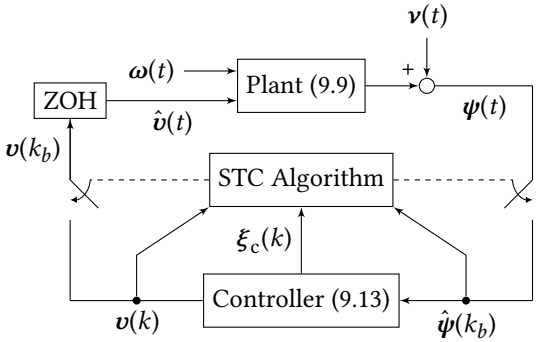


Figure 9.2: Block diagram of a plant controlled with STC. ZOH stands for zero-order hold.

Problem 9.1. *Let the plant (9.9) and controller (9.13) models be known and suppose that (conservative estimates of) the sets $\mathcal{X}_0, \mathcal{W}, \mathcal{V}$ are known. Design an algorithm that computes $\kappa_b := k_{b+1} - k_b$ at time k_b based on (historical values of) $\hat{\boldsymbol{\upsilon}}, \hat{\boldsymbol{\psi}}$ and other available information, e.g., $\boldsymbol{\xi}_{\mathrm{c}}(k_b)$. The closed-loop system must be globally exponentially ISS w.r.t. bounded disturbances and noise.*

Remark 9.4. *A compact set $\mathcal{X}_0$ is required for the STC strategy we develop in Sec. 9.4. A large enough set may be easily estimated in most applications. For $\mathcal{X}_0 = \mathbb{R}^{n_p}$, we provide an initialization algorithm in the Appendix, Section 9.C.*

9.3.1 Triggering mechanism and stability results

Our PSTC approach is to design an algorithm that computes worst-case triggering times of PETC. For compactness of expressions, denote the auxiliary vectors

$$\zeta(t) \coloneqq \begin{bmatrix} \boldsymbol{\psi}(t) \\ \boldsymbol{\upsilon}(\lfloor t/h \rfloor) \end{bmatrix} \text{ and } \hat{\zeta}(t) \coloneqq \begin{bmatrix} \hat{\boldsymbol{\psi}}(\lfloor t/h \rfloor) \\ \hat{\boldsymbol{\upsilon}}(t) \end{bmatrix}$$

as the updated output/input and the held output/input, respectively. We start with a centralized output-based PETC triggering mechanism from [11], which for STC means that all inputs and outputs are updated at the same time:

$$t_{b+1} = \inf_{t \in \mathcal{T}_b} \eta(\zeta(t), \hat{\zeta}(t)) > \epsilon^2, \tag{9.14a}$$

$$\eta(\zeta(t), \hat{\zeta}(t)) \coloneqq |\zeta(t) - \hat{\zeta}(t)|^2 - \sigma^2 |\zeta(t)|^2, \tag{9.14b}$$

where $\mathcal{T}_b = \{t_b + hk, t_b + 2hk, ..., t_b + h\bar{\kappa}\}$, $0 \le \sigma < 1$ is the designed triggering parameter, $\bar{\kappa}$ is a specified maximum inter-event discrete time, and $\epsilon \ge 0$ is a margin parameter.[3]

Unfortunately, there are no results in the literature for whether the closed-loop PETC system is ISS w.r.t. measurement noise or a positive value of ϵ. Thus, first we prove that this is the case; i.e., when the PETC (or any mechanism that triggers earlier) is GES, then it is ISS and $\mathcal{L}_p$ stable w.r.t. additive disturbances, measurement noise, and the ϵ parameter. These results are relevant not only for the current STC work, but also for PETC.

We first model the plant (9.9) controlled with (9.13) under the PETC triggering rule (9.14) with $\bar{\kappa} = \infty$ as a hybrid system (9.1) equipped with a timer, with $\boldsymbol{\chi}^\mathsf{T} \coloneqq [\xi_p^\mathsf{T} \ \xi_c^\mathsf{T} \ \hat{\boldsymbol{\psi}}^\mathsf{T} \ \hat{\boldsymbol{\upsilon}}^\mathsf{T}]$ and $\boldsymbol{\delta}^\mathsf{T} \coloneqq [\boldsymbol{\omega}^\mathsf{T} \ \boldsymbol{\nu}^\mathsf{T} \ \epsilon]$; the model is

$$\begin{bmatrix} \dot{\boldsymbol{\chi}} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} \bar{A}\boldsymbol{\chi} + \bar{B}\boldsymbol{\omega} \\ 1 \end{bmatrix}, \quad \tau \in [0, h], \tag{9.15a}$$

$$\begin{bmatrix} \boldsymbol{\chi}^+ \\ \tau^+ \end{bmatrix} = \begin{cases} \begin{bmatrix} J_1\boldsymbol{\chi} + L\boldsymbol{\nu} \\ 0 \end{bmatrix}, & \tau = h, \quad (9.15b) \\ & (\bar{F}\boldsymbol{\chi} + \bar{G}\boldsymbol{\nu})^\mathsf{T}\bar{Q}(\bar{F}\boldsymbol{\chi} + \bar{G}\boldsymbol{\nu}) \ge \epsilon^2 \quad (9.15c) \\ \begin{bmatrix} J_2\boldsymbol{\chi} \\ 0 \end{bmatrix}, & \tau = h, \quad (9.15d) \\ & (\bar{F}\boldsymbol{\chi} + \bar{G}\boldsymbol{\nu})^\mathsf{T}\bar{Q}(\bar{F}\boldsymbol{\chi} + \bar{G}\boldsymbol{\nu}) \le \epsilon^2 \quad (9.15e) \end{cases}$$

$$\boldsymbol{\psi} = \bar{C}\boldsymbol{\chi} + \boldsymbol{\nu}, \tag{9.15f}$$

[3] When $\epsilon > 0$, Eq. (9.14) is called mixed-triggering [82], which is often used in practice to improve sampling performance around the origin. When $\sigma = 0$, it is known as Lebesgue sampling [3].

where

$$\bar{A} = \begin{bmatrix} A_\mathrm{p} & \mathbf{0} & \mathbf{0} & B_\mathrm{p} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \bar{B} = \begin{bmatrix} E \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \bar{C} = \begin{bmatrix} C_\mathrm{p} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{9.16}$$

$$J_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ B_\mathrm{c} C_\mathrm{p} & A_\mathrm{c} & \mathbf{0} & \mathbf{0} \\ C_\mathrm{p} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ D_\mathrm{c} C_\mathrm{p} & C_\mathrm{c} & \mathbf{0} & \mathbf{0} \end{bmatrix}, J_2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_\mathrm{c} & B_\mathrm{c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, L = \begin{bmatrix} \mathbf{0} \\ B_\mathrm{c} \\ \mathbf{I} \\ D_\mathrm{c} \end{bmatrix}$$

$$\bar{Q} = \begin{bmatrix} (1-\sigma^2)\mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}, \bar{F} = \begin{bmatrix} C_\mathrm{p} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ D_\mathrm{c} C_\mathrm{p} & C_\mathrm{c} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \bar{G} = \begin{bmatrix} \mathbf{I} \\ D_\mathrm{c} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

where $\bar{Q}$ is partitioned according to $(\zeta, \hat{\zeta})$. The jump map matrices represent the update of input and output (J_1) or no update except for the controller state (J_2). The quadratic inequalities represent the triggering condition (9.14a), where condition (9.15c) is present for the PETC, but absent for an STC that triggers no later than PETC.

Remark 9.5. *The choice of non-strict inequalities in Eq. (9.15c) and Eq. (9.15e) renders the system non-deterministic. This choice was made for mathematical convenience: the proofs using Eq.* (9.15) *are valid across the non-determinism, and thus cover both choices of making strict either inequality. This approach has also benefits with respect to robustness, see [118].*

If we design an STC logic that triggers no later than the PETC, we need a slight modification on the stability theorems of [11] for PETC to prove that stability also holds when sampling earlier than the moment the triggering is fired.[4] For absent noise ($\boldsymbol{\nu} \equiv \mathbf{0}$) and $\epsilon = 0$, we can adapt the LMI conditions for verifying stability in [11] for PETC for STC. The following is an adaptation of [11, Theorem V.2]:

Theorem 9.2. *Consider the Hamiltonian matrix and its matrix exponential below*

$$\begin{aligned} H &\coloneqq \begin{bmatrix} \bar{A} + \rho \mathbf{I} & \bar{B}\bar{B}^\mathsf{T} \\ -\gamma^2 \bar{C}^\mathsf{T}\bar{C} & -(\bar{A} + \rho \mathbf{I})^\mathsf{T} \end{bmatrix}, \\ G(t) &\coloneqq \mathrm{e}^{-Ht} = \begin{bmatrix} G_{11}(t) & G_{12}(t) \\ G_{21}(t) & G_{22}(t) \end{bmatrix}. \end{aligned} \tag{9.17}$$

Assume $G_{11}(t)$ in invertible for all $t \in [0, h]$, which always holds for sufficiently small h [11]. Let $\bar{G}_{11} \coloneqq G_{11}(h)$, $\bar{G}_{12} \coloneqq G_{12}(h)$ and $\bar{G}_{21} \coloneqq G_{21}(h)$. Suppose there exists a symmetric matrix

[4] This is often not needed for CETC as argued in the previous chapters, and some special designs of PETC such as [86] prevent by design the violation of the triggering condition, hence also not requiring special stability results. However, the best techniques to establish stability and performance of PETC controllers are available in [11] and derived work, hence we demonstrate here that a small adaptation of their LMI conditions ensures the same properties for early-sampling approaches.

$P_h \succ 0$ *and a scalar* $\mu \geq 0$ *such that*

$$\begin{bmatrix} P_h & J_1^\mathsf{T}\bar{G}_{11}^{-\mathsf{T}}P_h\bar{S} & J_1^\mathsf{T}\Phi(P_h) \\ \star & \mathbf{I} - \bar{S}^\mathsf{T}P_h\bar{S} & \mathbf{0} \\ \star & \star & \Phi(P_h) \end{bmatrix} \succ \mathbf{0}, \\ \begin{bmatrix} P_h + \mu\bar{F}^\mathsf{T}\bar{Q}\bar{F} & J_2^\mathsf{T}\bar{G}_{11}^{-\mathsf{T}}P_h\bar{S} & J_2^\mathsf{T}\Phi(P_h) \\ \star & \mathbf{I} - \bar{S}^\mathsf{T}P_h\bar{S} & \mathbf{0} \\ \star & \star & \Phi(P_h) \end{bmatrix} \succ \mathbf{0}, \tag{9.18}$$

with $\Phi(P_h) \coloneqq \bar{G}_{11}^{-\mathsf{T}}P_h\bar{G}_{11}^{-1} + \bar{G}_{21}\bar{G}_{11}^{-1}$ *and* $\bar{S}$ *satisfying* $\bar{S}\bar{S}^\mathsf{T} = -\bar{G}_{11}^{-\mathsf{T}}\bar{G}_{12}$. *Then the set* $\{\begin{bmatrix}\chi^\mathsf{T} & \tau\end{bmatrix} \mid \chi = 0\}$ *is GES with decay rate* ρ *when* $\omega(t) \equiv 0$ *and has an* $\mathcal{L}_2$*-gain from* ω *to* ψ *smaller or equal than* γ. *Moreover,* any logic that triggers no later than such a PETC logic has the same stability and $\mathcal{L}_2$-gain properties.

Proof. (Sketch) We kindly refer the reader to the proof of the original theorem in [11, Theorem V.2], where a similar set of linear matrix inequalities (LMIs) are derived as sufficient conditions for the stability and $\mathcal{L}_2$-gain properties of the PETC. From that, consider a centralized version of the output feedback, constraining the J matrices to J_1 and J_2 in (9.16). The main change with respect to the original Theorem is to consider that triggering may happen earlier, so the triggering function $\chi^\mathsf{T}(t)\bar{F}^\mathsf{T}\bar{Q}\bar{F}\chi(t)$ may still be smaller than zero. Therefore, we allow the jumps in (9.15) according to J_1 at any situation, not only when the triggering function is positive. Then, while the second LMI in (9.18) uses the S-procedure to encode the triggering function being non-positive, the first LMI assumes nothing related to the triggering function, and the term P_h comes alone in the upper-left block.[5] □

For analysis purposes, even though ϵ is typically a design parameter, we can treat it as an external disturbance on the jump set. The main result of this Section is that the PETC system (9.15) is homogeneous in the sense of Definition 9.4, which implies that it is input-to-state and $\mathcal{L}_p$ stable w.r.t. noise and the ϵ parameter.

Lemma 9.1. *System (9.15) is homogeneous in the sense of Definition 9.4 and satisfies Assumption 9.1.*

The proof is found in Appendix, Section 9.A. The following result follows from Theorem 9.1 and Lemma 9.1.

Theorem 9.3. *If the system (9.9) with controller (9.13), using triggering mechanism (9.14) (or triggering earlier) is GES when* $\omega \equiv \mathbf{0}, \nu \equiv \mathbf{0}$ *and* $\epsilon = 0$, *then it is ISS and* $\mathcal{L}_p$*-stable if* $\omega \neq \mathbf{0}, \nu \neq \mathbf{0}$ *and* $\epsilon \neq 0$.

Remark 9.6. *Lemma 9.1 and Theorem 9.3 are valid for any quadratic triggering function of the form* $\eta(\zeta(t), \hat{\zeta}(t)) = \begin{bmatrix}\zeta(t)^\mathsf{T} & \hat{\zeta}(t)^\mathsf{T}\end{bmatrix}\bar{Q}\begin{bmatrix}\zeta(t) \\ \hat{\zeta}(t)\end{bmatrix}$, *as long as* $\bar{Q}$ *renders the closed-loop GES. We focus on the triggering function* (9.14b) *because for this case there are design procedures available (e.g., [11]).*

[5] Note that the first LMI is a sufficient condition for the periodic controller to have such stability and $\mathcal{L}_2$-gain properties; it is also stricter than the first LMI in [11, Theorem V.2]; hence this theorem gives that the STC ensures the performance of both PETC and the periodic implementation, which is to be expected.

9.4 Self-triggered control implementation

In this section, we devise a method to compute a lower bound of the PETC triggering time t_{b+1} from the available information at t_b. This lower bound becomes the STC triggering time. Throughout this section, we denote $\boldsymbol{z} \coloneqq \boldsymbol{\zeta}(t_b)$ and $\boldsymbol{u} \coloneqq \boldsymbol{v}(t_b)$. A way of computing such worst-case (earliest) time is by checking, for increasing values of $\kappa \in \mathbb{N}, \kappa \leq \bar{\kappa}$, whether $\eta(\boldsymbol{\zeta}(t_b + h\kappa), \boldsymbol{z})$ can be greater than ϵ^2 given the available information. This leads to the following subproblem:

Subproblem 9.1. *Let (supersets of) $\mathcal{X}(t_b)$ and $\mathcal{W}$ be known. For a given $\kappa \in \{1, ..., \bar{\kappa}\}$, determine, in a conservative but computationally efficient way, if there exist $\boldsymbol{x}'_{\mathrm{p}} \in \mathrm{reach}(t_b, t_b + h\kappa, \mathcal{X}(t_b), \boldsymbol{u}, \mathcal{W})$ and $\boldsymbol{v} \in \mathcal{E}(\boldsymbol{0}, \boldsymbol{V})$ such that $\eta\left(\begin{bmatrix} \boldsymbol{C}_{\mathrm{p}}\boldsymbol{x}'_{\mathrm{p}} + \boldsymbol{v} & \boldsymbol{v}(t_b + h\kappa) \end{bmatrix}^{\mathsf{T}}, \boldsymbol{z}\right) > \epsilon^2$.*

In the subproblem above, conservative means that, if the exact answer cannot be established, the answer is assumed to be true. Note that it requires the state set $\mathcal{X}(t_b)$, which ideally would be a single point. The larger this set is, the more conservative our solution is. This brings us the following subproblem:

Subproblem 9.2. *Given a superset of $\mathcal{X}_0$, historical values of $\hat{\boldsymbol{\zeta}}$, and $\boldsymbol{\xi}_{\mathrm{c}}(k)$, determine a small outer-approximation of $\mathcal{X}(t_b)$.*

In order to use ellipsoidal methods, we assume initial set estimates to be ellipsoids:

Assumption 9.4. *Matrices $\boldsymbol{X}_0 \in \mathbb{S}^{n_{\mathrm{p}}}_{++}$, $\bar{\boldsymbol{W}} \in \mathbb{S}^{n_{\mathrm{w}}}_{++}$, and $\boldsymbol{V} \in \mathbb{S}^{n_{\mathrm{y}}}_{++}$ are known, such that $\tilde{\mathcal{X}}_0 = \mathcal{E}(\boldsymbol{0}, \boldsymbol{X}_0) \supseteq \mathcal{X}_0$, $\tilde{\mathcal{W}} = \mathcal{E}(\boldsymbol{0}, \bar{\boldsymbol{W}}) \supseteq \mathcal{W}$, and $\tilde{\mathcal{V}} = \mathcal{E}(\boldsymbol{0}, \boldsymbol{V}) \supseteq \mathcal{V}$.*

Let us solve Subproblems 9.1 and 9.2 recursively. Suppose that, at time k_b, an ellipsoid $\tilde{\mathcal{X}}(k_b|k_{b-1}) \coloneqq \mathcal{E}(\tilde{\boldsymbol{\xi}}_{\mathrm{p}}(k_{b-1}), \boldsymbol{X}_{b|b-1}) \ni \boldsymbol{\xi}_p(hk_b)$ is known. First the state estimate $\tilde{\mathcal{X}}$ is updated with the newly acquired information $\boldsymbol{y}$. That is achieved through the intersection operation in (9.10c), which returns $\tilde{\mathcal{X}}(k_b|k_b)$: in this case, $\mathcal{X}_{\mathrm{y}}(t_b) = \mathcal{K}(\boldsymbol{y}, \boldsymbol{V}, \boldsymbol{C}_{\mathrm{p}})$ and therefore the trace-optimal Fusion in Eq. (9.12) is used.[6] From this point, denote the center of the state estimate as $\tilde{\boldsymbol{x}}_{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{p}}}$ and its shape matrix as $\boldsymbol{X} \in \mathbb{S}^{n_{\mathrm{p}}}_{++}$; thus, $\tilde{\mathcal{X}}(k_b|k_b) = \mathcal{E}(\tilde{\boldsymbol{x}}_{\mathrm{p}}, \boldsymbol{X})$.

We can now compute the reachable sets for the controller and plant states. First define the transition matrices:

$$\boldsymbol{\Phi}_{\mathrm{p}}(\kappa) \coloneqq \mathrm{e}^{\boldsymbol{A}_{\mathrm{p}} h\kappa}, \qquad \boldsymbol{\Gamma}_{\mathrm{p}}(\kappa) \coloneqq \int_0^{h\kappa} \mathrm{e}^{\boldsymbol{A}_{\mathrm{p}} s} \boldsymbol{B}_{\mathrm{p}} \mathrm{d}s, \tag{9.19a}$$

$$\boldsymbol{\Phi}_{\mathrm{c}}(\kappa) \coloneqq \boldsymbol{A}_{\mathrm{c}}^{\kappa}, \qquad \boldsymbol{\Gamma}_{\mathrm{c}}(\kappa) \coloneqq \sum_0^{\kappa-1} \boldsymbol{A}_{\mathrm{c}}^{\kappa} \boldsymbol{B}_{\mathrm{c}}, \tag{9.19b}$$

[6] Only a scalar parameter needs to be optimized and, since the function is convex, a golden search can be used up to a given precision. Nonetheless, this may be computationally too expensive depending on the application. In that case, a fixed λ can be picked, improving computation speed at the expense of larger ellipsoids and more frequent triggering.

Due to linearity, we can separate the reachable set $\mathcal{X}(t_b + h\kappa|t_b)$ between the contribution of state and control input, and that of the unknown disturbances:

$$\tilde{\mathcal{X}}(t_b+h\kappa|t_b) = \Phi_{\mathrm{p}}(\kappa)\tilde{\mathcal{X}}(k_b|k_b)+\Gamma_{\mathrm{p}}(\kappa)\boldsymbol{u}+\tilde{\mathcal{X}}_{\mathrm{w}}(\kappa), \tag{9.20a}$$

$$\tilde{\mathcal{X}}_{\mathrm{w}}(\kappa) \supseteq \mathcal{X}_{\mathrm{w}}(\kappa) = \bigcup_{\boldsymbol{\omega}\in\mathcal{F}_{\mathcal{W}}} \int_0^{h\kappa} \mathrm{e}^{A_{\mathrm{p}}(h\kappa-s)}\boldsymbol{E}\boldsymbol{\omega}(s)\mathrm{d}s. \tag{9.20b}$$

Remark 9.7. *The computation of supersets $\tilde{\mathcal{X}}_{\mathrm{w}}(\kappa) \supseteq \mathcal{X}_{\mathrm{w}}(\kappa)$ can be done off-line for all $\kappa \in \{1, ..., \bar{\kappa}\}$ using the method described in Section 9.2.2.*

We are ready to solve Subproblem 9.1. Denote $\boldsymbol{W}(\kappa)$ as the shape matrix of $\tilde{\mathcal{X}}_{\mathrm{w}}(\kappa)$, i.e., $\tilde{\mathcal{X}}_{\mathrm{w}}(\kappa) \coloneqq \mathcal{E}(\mathbf{0}, \boldsymbol{W}(\kappa))$; also, let $\boldsymbol{p}^{\mathsf{T}} \coloneqq [\boldsymbol{x}_{\mathrm{p}}^{\mathsf{T}} \;\; \boldsymbol{x}_{\mathrm{c}}^{\mathsf{T}} \;\; \boldsymbol{y}^{\mathsf{T}}]$ and

$$\boldsymbol{C}_{\mathrm{E}} \coloneqq \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \boldsymbol{C}_{\mathrm{c}} & \boldsymbol{D}_{\mathrm{c}} \end{bmatrix},$$

$$\boldsymbol{N}_\kappa \coloneqq \begin{bmatrix} \boldsymbol{C}_{\mathrm{p}}\Phi_{\mathrm{p}}(\kappa) & \boldsymbol{C}_{\mathrm{p}}\Gamma_{\mathrm{p}}(\kappa)\boldsymbol{C}_{\mathrm{c}} & \boldsymbol{C}_{\mathrm{p}}\Gamma_{\mathrm{p}}(\kappa)\boldsymbol{D}_{\mathrm{c}} \\ \mathbf{0} & \boldsymbol{C}_{\mathrm{c}}\Phi_{\mathrm{c}}(\kappa) & \boldsymbol{C}_{\mathrm{c}}\Gamma_{\mathrm{c}}(\kappa) + \boldsymbol{D}_{\mathrm{c}} \end{bmatrix}.$$

Note that, if there exists $\boldsymbol{z}'$ yielding $\eta(\boldsymbol{z}', \boldsymbol{z}) > \epsilon^2$, then $\max_{\boldsymbol{z}'} \eta(\boldsymbol{z}', \boldsymbol{z}) > \epsilon^2$. This means that we can pose Subproblem 9.1 as an optimization problem:

Subproblem 9.3. *From existing information on the controller, determine the worst-case triggering function value at a given time instant. That is, given $\tilde{\boldsymbol{x}}_{\mathrm{p}}, \boldsymbol{X}, \boldsymbol{x}_{\mathrm{c}}$ and $\boldsymbol{y}$, determine, for a given κ,*

$$\max_{\boldsymbol{z}', \boldsymbol{z}, \boldsymbol{x}_{\mathrm{p}}, \boldsymbol{d}, \boldsymbol{v}'} \quad \eta(\boldsymbol{z}', \boldsymbol{z}) = [\boldsymbol{z}'^{\mathsf{T}} \;\; \boldsymbol{z}^{\mathsf{T}}]\bar{\boldsymbol{Q}}\begin{bmatrix} \boldsymbol{z}' \\ \boldsymbol{z} \end{bmatrix} \tag{9.21a}$$

$$\text{subject to} \quad \boldsymbol{z}' = \boldsymbol{N}_\kappa \boldsymbol{p} + \begin{bmatrix} \boldsymbol{v}' \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{\mathrm{p}}\boldsymbol{d} \\ \mathbf{0} \end{bmatrix}, \tag{9.21b}$$

$$\boldsymbol{z} = \boldsymbol{C}_{\mathrm{E}}\boldsymbol{p}, \tag{9.21c}$$

$$(\boldsymbol{x}_{\mathrm{p}} - \tilde{\boldsymbol{x}}_{\mathrm{p}})^{\mathsf{T}}\boldsymbol{X}^{-1}(\boldsymbol{x}_{\mathrm{p}} - \tilde{\boldsymbol{x}}_{\mathrm{p}}) \leq 1, \tag{9.21d}$$

$$\boldsymbol{d}^{\mathsf{T}}\boldsymbol{W}(\kappa)^{-1}\boldsymbol{d} \leq 1, \tag{9.21e}$$

$$\boldsymbol{v}'^{\mathsf{T}}\boldsymbol{V}^{-1}\boldsymbol{v}' \leq 1, \tag{9.21f}$$

The decision variables are $\boldsymbol{z}'$ representing the possible values of $\boldsymbol{\zeta}(t_b + h\kappa)$; $\boldsymbol{z}$; $\boldsymbol{x}_{\mathrm{p}}$ which is the unknown value of $\boldsymbol{\xi}_{\mathrm{p}}(t_b)$; $\boldsymbol{d}$ as the contribution from the unknown disturbances to states at $t_b + h\kappa$; and $\boldsymbol{v}'$ as the unknown future noise $\boldsymbol{\nu}(t_b + h\kappa)$. The objective function (9.21a) is the triggering function and the constraints are: (9.21b) for the dynamics of $\boldsymbol{\zeta}$; (9.21c) as its initial condition; and (9.21d), (9.21e) and (9.21f) as the ellipsoidal constraints for the state estimate, $\boldsymbol{d}$ and $\boldsymbol{v}'$, respectively. This problem is solved for increasing values of $\kappa \in \{1, ..., \bar{\kappa}\}$, until one yields a value greater than ϵ.

Remark 9.8. *Subproblem 9.3 is a non-convex quadratically constrained quadratic programming (QCQP) problem. Its constraints are convex but the objective function is non-convex since $\bar{\boldsymbol{Q}}$ is not definite. Nevertheless, it is always feasible: one solution is obtained by taking $\boldsymbol{d} = \mathbf{0}, \boldsymbol{v} = \mathbf{0}, \boldsymbol{x}_{\mathrm{p}} = \tilde{\boldsymbol{x}}_{\mathrm{p}}$, and using these values to determine $\boldsymbol{z}'$ and $\boldsymbol{z}$ in Eqs.* (9.21b) *and* (9.21c).

The remark above discourages solving the actual optimization problem. Instead, we propose computing a conservative upper bound for it. Let $\tilde{\boldsymbol{p}}^\mathsf{T} \coloneqq [\tilde{\boldsymbol{x}}_\mathrm{p}^\mathsf{T} \ \boldsymbol{x}_\mathrm{c}^\mathsf{T} \ \boldsymbol{y}^\mathsf{T}]$ be the vector of available information, $\mathcal{N} \coloneqq \{1, 2, ..., n_\mathrm{p}\}$, and

$$
\begin{aligned}
\boldsymbol{Q}(\kappa) &\coloneqq \begin{bmatrix} \boldsymbol{N}_\kappa \\ \boldsymbol{C}_\mathrm{E} \end{bmatrix}^\mathsf{T} \bar{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{N}_\kappa \\ \boldsymbol{C}_\mathrm{E} \end{bmatrix}, \ \boldsymbol{C}_\mathrm{w} \coloneqq \begin{bmatrix} \boldsymbol{C}_\mathrm{p} \\ \boldsymbol{0} \end{bmatrix}, \ \boldsymbol{C}_\mathrm{v} \coloneqq \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix}, \\
\boldsymbol{F}_\mathrm{w}(\kappa) &\coloneqq \begin{bmatrix} \boldsymbol{N}_\kappa^\mathsf{T} & \boldsymbol{C}_\mathrm{E}^\mathsf{T} \end{bmatrix} \bar{\boldsymbol{Q}} \boldsymbol{C}_\mathrm{w}, \ \boldsymbol{F}_\mathrm{v}(\kappa) \coloneqq \begin{bmatrix} \boldsymbol{N}_\kappa^\mathsf{T} & \boldsymbol{C}_\mathrm{E}^\mathsf{T} \end{bmatrix} \bar{\boldsymbol{Q}} \boldsymbol{C}_\mathrm{v}, \\
\boldsymbol{R}_\mathrm{w}(\kappa) &\coloneqq \boldsymbol{F}_\mathrm{w}(\kappa) \boldsymbol{W}(\kappa) \boldsymbol{F}_\mathrm{w}(\kappa)^\mathsf{T}, \ \boldsymbol{R}_\mathrm{v}(\kappa) \coloneqq \boldsymbol{F}_\mathrm{v}(\kappa) \boldsymbol{V} \boldsymbol{F}_\mathrm{v}(\kappa)^\mathsf{T}, \\
\boldsymbol{Q}_\mathrm{w} &\coloneqq \boldsymbol{C}_\mathrm{w}^\mathsf{T} \bar{\boldsymbol{Q}} \boldsymbol{C}_\mathrm{w}, \ \boldsymbol{Q}_\mathrm{v} \coloneqq \boldsymbol{C}_\mathrm{v}^\mathsf{T} \bar{\boldsymbol{Q}} \boldsymbol{C}_\mathrm{v}, \ c_\mathrm{v} \coloneqq \lambda_{\max}(\boldsymbol{V} \boldsymbol{Q}_\mathrm{v}), \\
c_{\mathrm{vw}}(\kappa) &\coloneqq \sqrt{\lambda_{\max}(\boldsymbol{C}_\mathrm{v}^\mathsf{T} \bar{\boldsymbol{Q}} \boldsymbol{C}_\mathrm{w} \boldsymbol{W}(\kappa) \boldsymbol{C}_\mathrm{w}^\mathsf{T} \bar{\boldsymbol{Q}} \boldsymbol{C}_\mathrm{v} \boldsymbol{V})}.
\end{aligned} \tag{9.22}
$$

Note that all of the matrices and scalars above can be computed off-line for $\kappa \in \{1, ..., \bar{\kappa}\}$. Define the estimate of the triggering function

$$
\begin{aligned}
\bar{\eta}(\kappa, \tilde{\boldsymbol{p}}, \boldsymbol{X}) \coloneqq \tilde{\boldsymbol{p}}^\mathsf{T} \boldsymbol{Q}(\kappa) \tilde{\boldsymbol{p}} + 2 \sqrt{\tilde{\boldsymbol{p}}^\mathsf{T} \boldsymbol{Q}(\kappa)|_{\bullet, \mathcal{N}} \boldsymbol{X} \boldsymbol{Q}(\kappa)|_{\bullet, \mathcal{N}}^\mathsf{T} \tilde{\boldsymbol{p}}} + \lambda_{\max}(\boldsymbol{X} \boldsymbol{Q}(\kappa)|_{\mathcal{N}, \mathcal{N}}) + 2 \sqrt{\tilde{\boldsymbol{p}}^\mathsf{T} \boldsymbol{R}_\mathrm{v}(\kappa) \tilde{\boldsymbol{p}}} \\
+ 2 \sqrt{\lambda_{\max}(\boldsymbol{R}_\mathrm{v}(\kappa)|_{\mathcal{N}, \mathcal{N}} \boldsymbol{X})} + 2 \sqrt{\tilde{\boldsymbol{p}}^\mathsf{T} \boldsymbol{R}_\mathrm{w}(\kappa) \tilde{\boldsymbol{p}}} + 2 \sqrt{\lambda_{\max}(\boldsymbol{R}_\mathrm{w}(\kappa)|_{\mathcal{N}, \mathcal{N}} \boldsymbol{X})} \\
+ 2 c_{\mathrm{vw}}(\kappa) + c_\mathrm{v} + \lambda_{\max}(\boldsymbol{W}(\kappa) \boldsymbol{Q}_\mathrm{w}).
\end{aligned}
$$

All eigenvalues in Eq. (9.22) and in $\bar{\eta}$ are real, because their arguments are either symmetric matrices or products of symmetric matrices. We have the following result, whose proof is found in the appendix, Section 9.B.

Theorem 9.4. *$\bar{\eta}(\kappa, \tilde{\boldsymbol{p}}, \boldsymbol{X})$ provides an upper bound for the solution of Subproblem 9.3. That is,*

$$
\bar{\eta}(\kappa, \tilde{\boldsymbol{p}}, \boldsymbol{X}) \geq \eta(\boldsymbol{z}', \boldsymbol{z})
$$

for all $\boldsymbol{z}', \boldsymbol{z}, \boldsymbol{x}_\mathrm{p}, \boldsymbol{d}, \boldsymbol{v}'$ satisfying constraints (9.21b)–(9.21f).

The controller selects $\kappa^* = \inf_\kappa \bar{\eta}(\kappa, \tilde{\boldsymbol{p}},) > \epsilon^2$, if $\bar{\eta} > \epsilon^2$ for some $\kappa \leq \bar{\kappa}$, or $\kappa^* = \bar{\kappa}$ otherwise. Finally, step (9.10a) of the observer is executed using Eq. (9.20a). Its operations are the affine transformation $\boldsymbol{\Gamma}_\mathrm{p}(\kappa^*) \tilde{\mathcal{X}}(t_b|t_b) + \boldsymbol{\Phi}_\mathrm{p}(\kappa^*) \boldsymbol{u}$ followed by a Minkowski sum with $\tilde{\mathcal{X}}_\mathrm{w}(\kappa^*)$, which is outer-approximated through Eq. (9.11).

Algorithm 4 summarizes the steps performed at every instant k_b for both updating the state estimate and computing κ^*. The operations "fusion" and "minksum" represent the ellipsoidal outer-approximations from Eqs. (9.12) and (9.11), respectively. The ellipsoidal GSE (steps 2, 11 and 12) is depicted in Fig. 9.3.

Remark 9.9. *For the noiseless case ($\boldsymbol{V} = \boldsymbol{0}$), we need to modify step 2 of Alg. 4, because in this case the elliptical cylinder $\mathcal{K}(\boldsymbol{y}, \boldsymbol{V}, \boldsymbol{C}_\mathrm{p})$ degenerates to a hyperplane. The intersection between an ellipsoid and a hyperplane has an exact ellipsoidal solution (see [114, Appendix IV]).*

Remark 9.10. *The complexity of Algorithm 4 is $\mathcal{O}(\bar{\kappa} \max(n_\mathrm{p}, n_\mathrm{w}, n_\mathrm{y})^3)$. It is dominated by the iterative procedure to compute $\bar{\eta}$ (line 6), which involves matrix multiplications and eigenvalue computations on matrices whose sizes depend on $n_\mathrm{p}, n_\mathrm{w}$ and n_w.*[7]

[7] Computing eigenvalues has been proven to have the same big-O complexity as matrix multiplication in [122]. The actual complexity of the matrix multiplication is unknown, the best known being $\mathcal{O}(n^{2.37})$. We chose to use the exponent of 3 because most practical algorithms for small matrices have this complexity.

Algorithm 4 PSTC Algorithm

Input: $\boldsymbol{x}_\mathrm{c}, \boldsymbol{y}$
Output: $\boldsymbol{u}, \kappa^*$

1: $\boldsymbol{u} \leftarrow \boldsymbol{C}_\mathrm{c}\boldsymbol{x}_\mathrm{c} + \boldsymbol{D}_\mathrm{c}\boldsymbol{y}$
2: $\mathcal{E}(\tilde{\boldsymbol{x}}_\mathrm{p}, \boldsymbol{X}) \leftarrow \text{fusion}\left(\mathcal{E}(\tilde{\boldsymbol{x}}_\mathrm{p}, \boldsymbol{X}), \mathcal{K}(\boldsymbol{y}, \boldsymbol{V}, \boldsymbol{C}_\mathrm{p})\right)$ (Eq. 9.12)
3: $\tilde{\boldsymbol{p}} \leftarrow [\tilde{\boldsymbol{x}}_\mathrm{p}^\mathsf{T} \ \boldsymbol{x}_\mathrm{c}^\mathsf{T} \ \boldsymbol{y}^\mathsf{T}]^\mathsf{T}$
4: $\kappa^* \leftarrow 1$
5: **while** $\kappa^* < \bar{\kappa}$ **do**
6: **if** $\bar{\eta}(\kappa^*, \tilde{\boldsymbol{p}}, \boldsymbol{X}) > \epsilon^2$ **then**
7: break
8: **end if**
9: $\kappa^* \leftarrow \kappa^* + 1$
10: **end while**
11: $\mathcal{E}(\tilde{\boldsymbol{x}}_\mathrm{p}, \boldsymbol{X}) \leftarrow \boldsymbol{\Phi}_\mathrm{p}(\kappa^*)\mathcal{E}(\tilde{\boldsymbol{x}}_\mathrm{p}, \boldsymbol{X}) + \boldsymbol{\Gamma}_\mathrm{p}(\kappa^*)\boldsymbol{u}$
12: $\mathcal{E}(\tilde{\boldsymbol{x}}_\mathrm{p}, \boldsymbol{X}) \leftarrow \text{minksum}(\mathcal{E}(\tilde{\boldsymbol{x}}_\mathrm{p}, \boldsymbol{X}), \mathcal{E}(\boldsymbol{0}, \boldsymbol{W}_{\kappa^*}))$ (Eq. 9.11)

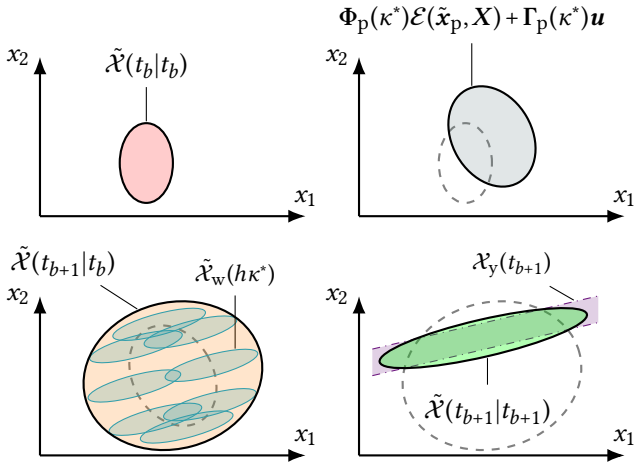


Figure 9.3: Steps of the ellipsoidal GSE in Alg. 4: step 11 (top right), step 12 (bottom left) and step 2 (bottom right).

9.5 Numerical example

In the following we describe a numerical example to illustrate our approach, which can be reproduced using the code in `https://github.com/ggleizer/pstc`. Consider the perturbed, unstable linearized batch plant with a PI controller taken from [123][8]:

$$A_{\mathrm{p}} = \begin{bmatrix} 1.38 & -0.208 & 6.715 & -5.676 \\ -0.581 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$B_{\mathrm{p}} = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix},\ C_{\mathrm{c}} = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix},\ E = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$A_{\mathrm{c}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ B_{\mathrm{c}} = \begin{bmatrix} 0 & h \\ h & 0 \end{bmatrix},\ C_{\mathrm{c}} = \begin{bmatrix} -2 & 0 \\ 0 & 8 \end{bmatrix},\ D_{\mathrm{c}} = \begin{bmatrix} 0 & -2 \\ 5 & 0 \end{bmatrix},$$

with $h = 0.01$, $\xi_{\mathrm{p}}(0) = 10[1\ -1\ -1\ 1]^{\mathsf{T}}$ and $\xi_{\mathrm{c}}(0) = \mathbf{0}$. The triggering parameter was set to $\sigma = 0.1$. We set $\bar{\kappa} = 25$ and computed $\boldsymbol{W}(\kappa)$ using the procedure described in Sec. 9.2.2, with $\mathcal{X}_{\mathrm{w}}(0) = \mathcal{E}(\mathbf{0}, 10^{-4}\mathbf{I})$ and $\mathcal{L} = \{\boldsymbol{c}_i | i \in \{1, 2, ..., n_{\mathrm{p}}\}\}$. The simulated disturbance was $\omega(t) = 0.1$, if $t \leq 5$; 0 otherwise. Simulations were run using Matlab R2018a on a MacBook Pro with a 3.1 GHz Intel Core i5 and 8 GB, 2133 MHz LPDDR memory. Noise was simulated through pseudo-random numbers between -0.01 and 0.01, which were pre-generated for all simulation steps with seed 1907. The optimal fusions from Eq. (9.12) were computed with the function `fminbnd` with default options. We set $\boldsymbol{W} = 0.1^2$ and $\boldsymbol{V} = 2 \cdot 0.011^2\mathbf{I}$, with the observer starting with $\tilde{\mathcal{X}}_0 = \mathbb{R}^{n_{\mathrm{p}}}$.

We first simulated the closed-loop STC without noise with $\epsilon = 0$, comparing its control and sampling performances PETC (Fig. 9.4). The state norms of both cases converge to zero at virtually the same rate, while, during the first five time units, PSTC gives similar inter-sample times as PETC. However, the PSTC triggering times tend to 1 as the state approaches the origin because $\bar{\eta}(\kappa, \mathbf{0}, \boldsymbol{X}) > 0$ for any $\kappa, \boldsymbol{X}$.

We then simulated a scenario with measurement noise: Fig. 9.5 (top) displays the evolution of the state norm under PSTC, while the middle and bottom plots show the inter-sample times from the PSTC. To inspect whether our PSTC indeed computes lower bounds of PETC ISTs, we compare the PSTC-generated ISTs to those obtained by using the PETC logic (9.14b) at each PSTC step — this way we ensure that both triggering mechanisms are applied from the same initial conditions at all times. As seen in Fig. 9.5, indeed the PSTC ISTs succeed in being lower bounds for the PETC ones. It is also clear how the sampling performances of both PSTC and PETC are affected by the noise: with $\epsilon = 0$, as the inputs get close enough to zero, noise alone can provoke a trigger. With $\epsilon = 0.1$, depicted in the bottom plot of Fig. 9.5, the resulting ISTs got significantly higher at a small cost in steady state error (as highlighted in Fig. 9.5, top).

The on-line CPU time statistics of Alg. 4 are displayed in Table 9.1. These numbers were obtained for the case with noise with $\epsilon = 0$, after ten consecutive runs of the main script to

[8] The controller was discretized using forward-Euler.

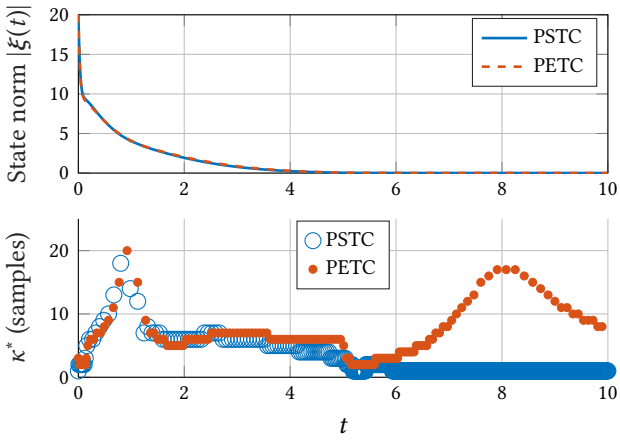


Figure 9.4: Simulation results without noise for PSTC and PETC: state norm $|\xi(t)|$ (top) and inter-event times κ^* (bottom).

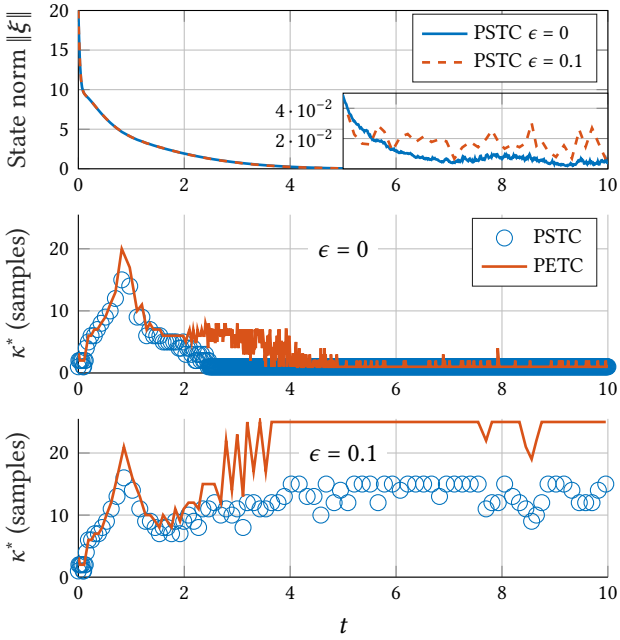


Figure 9.5: Simulation results with noise. State norm $|\xi(t)|$ with PSTC with $\epsilon \in \{0, 0.1\}$ (top); inter-event times κ^* from PSTC and PETC with $\epsilon = 0$ (middle) and $\epsilon = 0.1$ (bottom).

Table 9.1: CPU times of Alg. 4 for the numerical example.

	Time (ms)		
Phase (line(s) in Alg. 4)	**Min.**	**Mean**	**Max.**
Fusion (line 2)	0.39	0.49	1.71
Calculation of $\bar{\eta}$ (line 6)	0.50	0.60	1.90
Prediction (lines 11 and 12)	0.02	0.02	0.08
Full PSTC cycle	**1.01**	**1.27**	**8.49**

mitigate the overhead from, e.g., just-in-time compilation and process management of the operating system. The initialization step time (Section 9.C) was 0.03 ms. The figures show that the computations are fast, despite involving an optimization step for the fusion. The most expensive step was the calculation of $\bar{\eta}$, mainly due to the computation of eigenvalues and matrix multiplications. The off-line computations totaled 623.46 ms, out of which 609.26 ms were spent on the reachability ($\boldsymbol{W}(\kappa)$) and 14.19 ms on the remaining matrices and scalars ((9.22) and the operations in Section 9.C).

Remark 9.11. *Qualitatively comparing with [117], the issue of eventually triggering always when $\epsilon = 0$ also happens with $\theta = 1$ and $\gamma = 1$ in their STC. In this setting, one would achieve UGAS of the minimal robustly positive invariant subset associated with periodic control with disturbances. Increasing θ and γ enlarges the terminal set, in a similar way $\epsilon > 0$ does.*

9.6 Conclusions

We have presented a self-triggered strategy for output-feedback control of linear systems subject to bounded disturbances and noise, named PSTC. It is, to our knowledge, the first self-triggered implementation of such a general control structure. We first proved that the introduction of noise or mixed triggering does not hinder stability of neither PETC nor PSTC, then developed an algorithm that uses set-based methods for a viable self-triggered implementation. PSTC achieves virtually the same control performance as PETC, with slightly smaller inter-sample times. It is expected to be fast enough for most applications, as each step CPU time averaged 1 ms for the simulated four-state plant; and it scales well with the state-space dimension, since the few online optimization and line search operations are done on scalars, while higher-dimension computations are handled with basic linear algebra.

PSTC was developed for linear plants with linear controllers, which presents a limitation to its applicability. Some classes of nonlinearities could be handled by considering them as disturbances; since we assume that they are bounded, one would have to determine a compact set on which the states lie in order to compute the proper bounds. For locally linearizable systems, other types of unknown-but-bounded uncertainty descriptions are more suitable, such as parametric model uncertainty. In this case, the ellipsoidal estimator in [124] could be used as a starting point. There are also opportunities for improving the PSTC performance for linear systems. Aiming at a small computational complexity, we chose ellipsoids as set descriptors and devised simple upper bounds to the solution of online non-convex QCQP problems; however, these choices probably bring additional conservatism and hence increased communication frequency. From our simulations, this seems to

be particularly relevant when the state approaches the origin and when disturbances are significantly smaller than their estimated bounds. A few alternatives might reduce conservativeness: for example, (constrained) zonotopes [111] could replace ellipsoids; note, however, that this would require reformulating the optimization problem. Another possibility would be deriving tighter bounds for the non-convex QCQP.

Finally, the prediction of inter-sample times is a fundamental step for other problems that have been addressed in this dissertation, such as scheduling and formal sampling performance computation. The use of the methods presented in this chapter for these problems is an interesting direction for future work.

9.A Proof of Lemma 9.1

Before approaching the proof, one remark must be made: system (9.15) is equipped with a timer, with jumps only occurring after a certain time; this specializes it to what is defined in [110, Section 5] as a system with average dwell time, with $N = 1$, $\delta = 1/h$, and ζ arbitrarily small. This actually relaxes the Lyapunov stability conditions presented therein [110, Proposition 2]. Theorem 2 of [110] states that homogeneous systems with average dwell time satisfy Theorem 9.1 with $\boldsymbol{\psi} = \boldsymbol{\chi}$. Remark 16 of [110] argues that the same Propositions that build the proof of [110, Theorem 2] can be derived with $\boldsymbol{\psi} \neq \boldsymbol{\chi}$. Thus, the timer does not play a significant role in our proofs; as an additional benefit, the results without timer can be applied to continuous ETC.

Let $n \coloneqq n_\mathrm{p} + n_\mathrm{c} + n_\mathrm{u} + n_\mathrm{y}$ and the collected vector of exogenous signals $\boldsymbol{d}^\mathsf{T} \coloneqq \begin{bmatrix} \boldsymbol{w}^\mathsf{T} & \boldsymbol{v}^\mathsf{T} & \epsilon \end{bmatrix}$, giving $n_\mathrm{d} \coloneqq n_\mathrm{w} + n_\mathrm{y} + 1$. The flow sets is $C = \mathbb{R}^{n+n_\mathrm{d}}$ with projection $C_0 = \mathbb{R}^n$, and the jump set is $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ where

$$\mathcal{D}_1 = \{\boldsymbol{x} \in \mathbb{R}^n, [\boldsymbol{w}^\mathsf{T}\ \boldsymbol{v}^\mathsf{T}\ \epsilon]^\mathsf{T} \in \mathbb{R}^{n_\mathrm{d}} \mid (\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v})^\mathsf{T}\bar{\boldsymbol{Q}}(\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v}) \geq \epsilon^2\}, \tag{9.23}$$

$$\mathcal{D}_2 = \{\boldsymbol{x} \in \mathbb{R}^n, [\boldsymbol{w}^\mathsf{T}\ \boldsymbol{v}^\mathsf{T}\ \epsilon]^\mathsf{T} \in \mathbb{R}^{n_\mathrm{d}} \mid (\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v})^\mathsf{T}\bar{\boldsymbol{Q}}(\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v}) \leq \epsilon^2\}, \tag{9.24}$$

and their projections with $\boldsymbol{d} = \boldsymbol{0}$ are

$$\mathcal{D}_{10} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{x} \geq 0\},$$
$$\mathcal{D}_{20} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{x} \leq 0\}.$$

Since the flow map in (9.15) is linear, properties (9.6) and (9.7) hold for flows; also, condition (9.8a) is trivially satisfied because C and C_0 are the entire Euclidean space. In addition, since sets $\mathcal{D}_{i0}$ are conic and $\boldsymbol{g}$ is piecewise linear, it is easy to see that $\boldsymbol{g}(\lambda\boldsymbol{\chi}, \boldsymbol{0}) = \lambda\boldsymbol{g}(\boldsymbol{\chi}, \boldsymbol{0}), \forall \boldsymbol{\chi}(t,j) \in \mathcal{D}_{0i}, \forall i \in \{1,2\}$. Hence, homogeneity of $\boldsymbol{g}$ is also verified.

What remains to be verified is condition (9.8b). Note that the only components of $\boldsymbol{d}$ that enter the jump sets are $\boldsymbol{v}$ and ϵ. Rewriting the set sum on the LHS of (9.8b) gives

$$\begin{aligned}\mathcal{D}_{i0} + L_\mathrm{D}\mathcal{B}(|\boldsymbol{d}|) &= \{\boldsymbol{x}' + \boldsymbol{x}'' \mid \boldsymbol{x}' \in \mathcal{D}_{i0}, \boldsymbol{x}'' \in L_\mathrm{D}\mathcal{B}(|\boldsymbol{d}|)\} \\ &= \{\boldsymbol{x} \mid (\boldsymbol{x} - \boldsymbol{x}'')^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}(\boldsymbol{x} - \boldsymbol{x}'') \sim_i 0, \boldsymbol{x}''^\mathsf{T}\boldsymbol{x}'' \leq L_\mathrm{D}^2(\boldsymbol{v}^\mathsf{T}\boldsymbol{v} + \epsilon^2)\},\end{aligned}$$

where $\sim_1$ is $\geq$ and $\sim_2$ is $\leq$. Thus, (9.8b) can be restated as

$$\begin{aligned}&\forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{v} \in \mathbb{R}^{n_\mathrm{v}}, \epsilon \in \mathbb{R} : (\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v})^\mathsf{T}\bar{\boldsymbol{Q}}(\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v}) \sim_i \epsilon^2, \\ &\qquad \exists \boldsymbol{x}'' \in \mathbb{R}^n : (\boldsymbol{x} - \boldsymbol{x}'')^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}(\boldsymbol{x} - \boldsymbol{x}'') \sim_i 0, \boldsymbol{x}''^\mathsf{T}\boldsymbol{x}'' \leq L_\mathrm{D}^2(\boldsymbol{v}^\mathsf{T}\boldsymbol{v} + \epsilon^2).\end{aligned} \tag{9.25}$$

Since the pair $(\boldsymbol{A}_\mathrm{p}, \boldsymbol{C}_\mathrm{p})$ is observable, we can assume system (9.9) is in its canonical observable form; thus, taking $\boldsymbol{C}_\mathrm{p} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$, we can partition $\bar{\boldsymbol{F}}$ as

$$\bar{\boldsymbol{F}} = \left[\begin{array}{c|cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ D_\mathrm{c} & \mathbf{0} & C_\mathrm{c} & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array}\right] = \left[\begin{array}{c|c} \bar{\boldsymbol{G}} & \bar{\boldsymbol{H}} \end{array}\right],$$

where $\boldsymbol{x}^\mathsf{T}$ is partitioned accordingly as $\begin{bmatrix}\boldsymbol{y}^\mathsf{T} & \bar{\boldsymbol{x}}^\mathsf{T}\end{bmatrix}$, with $\bar{\boldsymbol{x}}$ containing all the remaining state components, obtaining $\bar{\boldsymbol{F}}\boldsymbol{x} = \bar{\boldsymbol{G}}\boldsymbol{y} + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}}$. We now divide the proof in two parts: $i = 1$ and $i = 2$.

To show (9.25) for $i = 1$, let us construct one $\boldsymbol{x}''$ that satisfies it for every $\boldsymbol{x}, \boldsymbol{v}, \epsilon$: take $\boldsymbol{x}''^\mathsf{T} = \begin{bmatrix}-\boldsymbol{v}^\mathsf{T} & \boldsymbol{0}\end{bmatrix}$. Then obviously $\boldsymbol{x}''^\mathsf{T}\boldsymbol{x}'' = \boldsymbol{v}^\mathsf{T}\boldsymbol{v} \le L_\mathrm{D}^2(\boldsymbol{v}^\mathsf{T}\boldsymbol{v} + \epsilon^2)$ with $L_D = 1$ and

$$\begin{aligned}(\boldsymbol{x} - \boldsymbol{x}'')^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}(\boldsymbol{x} - \boldsymbol{x}'') &= (\bar{\boldsymbol{F}}(\boldsymbol{x} - \boldsymbol{x}''))^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}(\boldsymbol{x} - \boldsymbol{x}'') = \left(\bar{\boldsymbol{F}}\begin{bmatrix}\boldsymbol{y}+\boldsymbol{v}\\ \bar{\boldsymbol{x}}\end{bmatrix}\right)^\mathsf{T}\bar{\boldsymbol{Q}}\left(\bar{\boldsymbol{F}}\begin{bmatrix}\boldsymbol{y}+\boldsymbol{v}\\ \bar{\boldsymbol{x}}\end{bmatrix}\right)\\ &= (\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}})^\mathsf{T}\bar{\boldsymbol{Q}}(\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}}) = (\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v})^\mathsf{T}\bar{\boldsymbol{Q}}(\bar{\boldsymbol{F}}\boldsymbol{x} + \bar{\boldsymbol{G}}\boldsymbol{v}) \ge \epsilon^2 \ge 0.\end{aligned}$$

Showing (9.25) for $i = 2$ is slightly more involved. First, notice the following fact:

$$\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}) < 0. \tag{9.26}$$

This is true because $\boldsymbol{x}^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{x}$ is just another representation of the triggering function (9.14b); thus, it can be expressed as $|\boldsymbol{z} - \hat{\boldsymbol{z}}|^2 - \sigma^2|\boldsymbol{z}|^2$ for some $\boldsymbol{z}, \hat{\boldsymbol{z}} \in \mathbb{R}^{n_\mathrm{y}+n_\mathrm{u}}$. This expression is negative if, e.g., $\boldsymbol{z} = \hat{\boldsymbol{z}} \neq \boldsymbol{0}$. Thus, $\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}$ cannot be positive semi-definite, hence its minimal eigenvalue is negative.

Again, let us construct one $\boldsymbol{x}''$ that satisfies (9.25) for every $\boldsymbol{x}, \boldsymbol{v}, \epsilon$. This is $\boldsymbol{x}''^\mathsf{T} = \begin{bmatrix}-\boldsymbol{v}^\mathsf{T} & \boldsymbol{0}\end{bmatrix} + \boldsymbol{q}^\mathsf{T}$, where $\boldsymbol{q}$ is the vector along the eigendirection corresponding to $\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})$, i.e. , $\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q} = \lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})\boldsymbol{q}$, satisfying

$$(\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}})^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q} \ge 0, \tag{9.27}$$

$$|\boldsymbol{q}|^2 = \left|\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})\right|^{-1}\epsilon^2. \tag{9.28}$$

One can always find such $\boldsymbol{q}$: (9.28) determines its norm; and, if (9.27) is not satisfied, $-\boldsymbol{q}$ satisfies it. This gives

$$\boldsymbol{q}^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q} = \frac{\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})}{\left|\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})\right|}\epsilon^2 = -\epsilon^2, \tag{9.29}$$

where the negative sign comes from Eq. (9.26). Therefore, the second inequality in (9.25) satisfies

$$\begin{aligned}&(\boldsymbol{x} - \boldsymbol{x}'')^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}(\boldsymbol{x} - \boldsymbol{x}'')\\ &\quad= (\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}})^\mathsf{T}\bar{\boldsymbol{Q}}(\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}}) - 2(\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}})^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q} + \boldsymbol{q}^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q}\\ &\quad\overset{(9.24)}{\le} \epsilon^2 - 2(\bar{\boldsymbol{G}}(\boldsymbol{y}+\boldsymbol{v}) + \bar{\boldsymbol{H}}\bar{\boldsymbol{x}})^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q} + \boldsymbol{q}^\mathsf{T}\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}}\boldsymbol{q} \overset{(9.27),(9.29)}{\le} \epsilon^2 - \epsilon^2 = 0.\end{aligned}$$

Additionally, the norm of $\boldsymbol{x}''$ satisfies

$$\left|\boldsymbol{x}''\right| \le |\boldsymbol{v}| + |\boldsymbol{q}| = |\boldsymbol{v}| + \left|\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})^{-\frac{1}{2}}\right||\epsilon| \le L(|\boldsymbol{v}| + |\epsilon|),$$

for $L \coloneqq \max\left(1, \left|\lambda_{\min}(\bar{\boldsymbol{F}}^\mathsf{T}\bar{\boldsymbol{Q}}\bar{\boldsymbol{F}})\right|^{-\frac{1}{2}}\right)$. Now, it is easy to see that

$$(|\boldsymbol{v}| + |\epsilon|)^2 \le 2\boldsymbol{v}^\mathsf{T}\boldsymbol{v} + 2\epsilon^2.$$

Hence, $\boldsymbol{x}''^\mathsf{T}\boldsymbol{x}'' \le L_\mathrm{D}^2(\boldsymbol{v}^\mathsf{T}\boldsymbol{v} + \epsilon^2)$ holds with $L_D = \sqrt{2}L$.

9.B Proof of Theorem 9.4

First, we introduce the following Lemma:

Lemma 9.2. *Let $\boldsymbol{M} \in \mathbb{S}_+^n$. For any $\boldsymbol{x} \in \mathbb{R}^n$ such that $\boldsymbol{x} \in \mathcal{E}(\boldsymbol{0}, \boldsymbol{M})$, there exist a vector $\boldsymbol{s}$ with $|\boldsymbol{s}| \leq 1$ and a matrix $\boldsymbol{S}$ such that $\boldsymbol{x} = \boldsymbol{S}\boldsymbol{s}$ and $\boldsymbol{S}\boldsymbol{S}^\mathsf{T} = \boldsymbol{M}$.*

Proof. Since $\boldsymbol{M}$ is symmetric, it admits the singular value decomposition

$$\boldsymbol{M} = \boldsymbol{U}^\mathsf{T} \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{U},$$

with $\boldsymbol{U}$ invertible and $\boldsymbol{D} \in \mathbb{S}_{++}$ diagonal. From Definition 9.7, it must hold that, for all $\boldsymbol{l} \in \mathbb{R}^n$,

$$\boldsymbol{l}^\mathsf{T}\boldsymbol{x} \leq (\boldsymbol{l}^\mathsf{T}\boldsymbol{M}\boldsymbol{l})^{1/2} = \left(\boldsymbol{l}^\mathsf{T}\boldsymbol{U}^\mathsf{T} \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{U}\boldsymbol{l} \right)^{1/2}. \tag{9.30}$$

Take $\boldsymbol{l}' \coloneqq \boldsymbol{U}\boldsymbol{l}$ and $\boldsymbol{s}' \coloneqq \boldsymbol{U}^{-\mathsf{T}}\boldsymbol{x}$. Then, (9.30) becomes

$$\boldsymbol{l}'^\mathsf{T}\boldsymbol{s}' \leq \left(\boldsymbol{l}'^\mathsf{T} \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{l}' \right)^{1/2} = (\boldsymbol{l}_1'^\mathsf{T}\boldsymbol{D}\boldsymbol{l}_1')^{1/2}, \tag{9.31}$$

where $\boldsymbol{l}'$ is partitioned into $\begin{bmatrix} \boldsymbol{l}_1'^\mathsf{T} & \boldsymbol{l}_2'^\mathsf{T} \end{bmatrix}^\mathsf{T}$ according to $\left[\begin{smallmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{smallmatrix}\right]$. Likewise, partition $\boldsymbol{s}'$ into $\begin{bmatrix} \boldsymbol{s}_1'^\mathsf{T} & \boldsymbol{s}_2'^\mathsf{T} \end{bmatrix}^\mathsf{T}$. Then, (9.31) becomes

$$\boldsymbol{l}_1'^\mathsf{T}\boldsymbol{s}_1' + \boldsymbol{l}_2'^\mathsf{T}\boldsymbol{s}_2' \leq (\boldsymbol{l}_1'^\mathsf{T}\boldsymbol{D}\boldsymbol{l}_1')^{1/2},$$

which, to hold for all $\boldsymbol{l}_1'$ and$\boldsymbol{l}_2'$, requires that $\boldsymbol{s}_2' = 0$. As $\boldsymbol{l}_1'^\mathsf{T}\boldsymbol{s}_1' \leq (\boldsymbol{l}_1'^\mathsf{T}\boldsymbol{D}\boldsymbol{l}_1')^{1/2}$ is the definition of the ellipsoid $\mathcal{E}(\boldsymbol{0}, \boldsymbol{D})$, we also conclude that $\boldsymbol{s}_1'^\mathsf{T}\boldsymbol{D}^{-1}\boldsymbol{s}_1' \leq 1$. Finally, the choice $\boldsymbol{s} = \boldsymbol{D}^{-1/2}\boldsymbol{s}_1'$ satisfies $\boldsymbol{s}^\mathsf{T}\boldsymbol{s} \leq 1$. Moreover,

$$\boldsymbol{U}^{-\mathsf{T}}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{s}_1' \\ \boldsymbol{s}_2' \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}^{1/2} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s} \iff \boldsymbol{x} = \boldsymbol{U}^\mathsf{T} \begin{bmatrix} \boldsymbol{D}^{1/2} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s},$$

so, $\boldsymbol{S} = \boldsymbol{U}^\mathsf{T} \begin{bmatrix} \boldsymbol{D}^{1/2} \\ \boldsymbol{0} \end{bmatrix}$ gives $\boldsymbol{x} = \boldsymbol{S}\boldsymbol{s}$ and $\boldsymbol{S}\boldsymbol{S}^\mathsf{T} = \boldsymbol{M}$. □

With the result above, the following Lemma introduces some useful bounds:

Lemma 9.3. *Let $\boldsymbol{M}_i \in \mathbb{S}_+^n, i \in \{1, 2\}$, $\boldsymbol{p} \in \mathbb{R}^m, \boldsymbol{F} \in \mathbb{R}^{n \times m}$, and $\boldsymbol{Q} \in \mathbb{S}^n$. For any $\boldsymbol{x}_i \in \mathbb{R}^n$ such that $\boldsymbol{x}_i \in \mathcal{E}(\boldsymbol{0}, \boldsymbol{M}_i)$, the following inequalities hold:*

$$\boldsymbol{p}^\mathsf{T}\boldsymbol{F}\boldsymbol{x}_i \leq \sqrt{\boldsymbol{p}^\mathsf{T}\boldsymbol{F}\boldsymbol{M}_i\boldsymbol{F}^\mathsf{T}\boldsymbol{p}}, \tag{9.32a}$$

$$\boldsymbol{x}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}_i \leq \lambda_{\max}(\boldsymbol{M}_i\boldsymbol{Q}), \tag{9.32b}$$

$$\boldsymbol{x}_1^\mathsf{T}\boldsymbol{F}\boldsymbol{x}_2 \leq \sqrt{\lambda_{\max}(\boldsymbol{F}\boldsymbol{M}_2\boldsymbol{F}^\mathsf{T}\boldsymbol{M}_1)}. \tag{9.32c}$$

Proof. Using Lemma 9.2, take $\boldsymbol{s}_i, \boldsymbol{S}_i$ satisfying $\boldsymbol{S}_i\boldsymbol{S}_i^\mathsf{T} = \boldsymbol{M}_i$ and $\boldsymbol{x}_i = \boldsymbol{S}_i\boldsymbol{s}_i$ such that $|\boldsymbol{s}_i| \leq 1$. Thus, $\boldsymbol{p}^\mathsf{T}\boldsymbol{F}\boldsymbol{x}_i = \boldsymbol{p}^\mathsf{T}\boldsymbol{F}\boldsymbol{S}_i\boldsymbol{s}_i \leq \left|\boldsymbol{p}^\mathsf{T}\boldsymbol{F}\boldsymbol{S}_i\right|$; $\boldsymbol{x}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}_i = \boldsymbol{s}_i^\mathsf{T}\boldsymbol{S}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{S}_i\boldsymbol{s}_i \leq \lambda_{\max}(\boldsymbol{S}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{S}_i)$; and $\boldsymbol{x}_1^\mathsf{T}\boldsymbol{F}\boldsymbol{x}_2 = \boldsymbol{s}_1^\mathsf{T}\boldsymbol{S}_1^\mathsf{T}\boldsymbol{F}\boldsymbol{S}_2\boldsymbol{s}_2 \leq |\boldsymbol{S}_1^\mathsf{T}\boldsymbol{F}\boldsymbol{S}_2| = \sqrt{\lambda_{\max}(\boldsymbol{S}_1^\mathsf{T}\boldsymbol{F}\boldsymbol{S}_2\boldsymbol{S}_2^\mathsf{T}\boldsymbol{F}^\mathsf{T}\boldsymbol{S}_1)}$. Using the fact that $\lambda(\boldsymbol{A}\boldsymbol{B}) = \lambda(\boldsymbol{B}\boldsymbol{A})$ for any $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$ and replacing $\boldsymbol{S}_i\boldsymbol{S}_i^\mathsf{T}$ with $\boldsymbol{M}_i$ provides (9.32). □

Now we can proceed to the proof of Theorem 9.4:

Proof of Theorem 9.4. Let $\boldsymbol{e} \coloneqq \boldsymbol{x}_\mathrm{p} - \tilde{\boldsymbol{x}}_\mathrm{p}$. Hence,

$$\boldsymbol{p} = \tilde{\boldsymbol{p}} + [\boldsymbol{e}^\mathsf{T} \ \mathbf{0} \ \mathbf{0}]^\mathsf{T} \tag{9.33}$$

and, from (9.21d), $\boldsymbol{e}^\mathsf{T}\boldsymbol{X}^{-1}\boldsymbol{e} \leq 1$. Rewrite Eq. (9.21a) as a function of $\tilde{\boldsymbol{p}}, \boldsymbol{e}, \boldsymbol{d}$, and $\boldsymbol{v}'$ by replacing $\boldsymbol{z}', \boldsymbol{z}$ and $\boldsymbol{p}$ from Eqs. (9.21b), (9.21c) and (9.33):

$$\begin{aligned}\eta(\boldsymbol{z}', \boldsymbol{z}) = \eta'(\kappa, \tilde{\boldsymbol{p}}, \boldsymbol{e}, \boldsymbol{v}', \boldsymbol{d}) = \\ \left(\tilde{\boldsymbol{p}} + \begin{bmatrix}\boldsymbol{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right)^\mathsf{T} Q(\kappa) \left(\tilde{\boldsymbol{p}} + \begin{bmatrix}\boldsymbol{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right) + 2\left(\tilde{\boldsymbol{p}} + \begin{bmatrix}\boldsymbol{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right)^\mathsf{T} F_\mathrm{v}(\kappa)\boldsymbol{v}' + 2\left(\tilde{\boldsymbol{p}} + \begin{bmatrix}\boldsymbol{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right)^\mathsf{T} F_\mathrm{w}(\kappa)\boldsymbol{d} \\ + 2\boldsymbol{v}'^\mathsf{T} C_\mathrm{v}^\mathsf{T} \bar{Q} C_\mathrm{w} \boldsymbol{d} + \boldsymbol{v}'^\mathsf{T} Q_\mathrm{v} \boldsymbol{v}' + \boldsymbol{d}^\mathsf{T} Q_\mathrm{w} \boldsymbol{d},\end{aligned}$$

which results in

$$\begin{aligned}\eta'(\kappa, \tilde{\boldsymbol{p}}, \boldsymbol{e}, \boldsymbol{v}', \boldsymbol{d}) = \tilde{\boldsymbol{p}}^\mathsf{T} Q(\kappa)\tilde{\boldsymbol{p}} + 2\tilde{\boldsymbol{p}}^\mathsf{T} Q(\kappa)|_{\cdot,\mathcal{N}}\boldsymbol{e} \\ + \boldsymbol{e}^\mathsf{T} Q(\kappa)|_{\mathcal{N},\mathcal{N}}\boldsymbol{e} + 2\tilde{\boldsymbol{p}}^\mathsf{T} F_\mathrm{v}(\kappa)\boldsymbol{v}' + 2\boldsymbol{e}^\mathsf{T} F_\mathrm{v}(\kappa)|_{\mathcal{N},\cdot}\boldsymbol{v}' + 2\tilde{\boldsymbol{p}}^\mathsf{T} F_\mathrm{w}(\kappa)\boldsymbol{d} + 2\boldsymbol{e}^\mathsf{T} F_\mathrm{w}(\kappa)|_{\mathcal{N},\cdot}\boldsymbol{d} \\ + 2\boldsymbol{v}'^\mathsf{T} C_\mathrm{v}^\mathsf{T} \bar{Q} C_\mathrm{w} \boldsymbol{d} + \boldsymbol{v}'^\mathsf{T} Q_\mathrm{v} \boldsymbol{v}' + \boldsymbol{d}^\mathsf{T} Q_\mathrm{w} \boldsymbol{d}.\end{aligned} \tag{9.34}$$

Now Lemma 9.3 is used. The only known term in Eq. (9.34) is the first. Eq. (9.32a) is used for second, fourth and sixth terms; Eq. (9.32b) for the third, ninth and tenth; and Eq. (9.32c) for the fifth, seventh and eighth terms. Mere replacement provides $\bar{\eta}(\kappa, \tilde{\boldsymbol{p}}, \boldsymbol{X})$. □

9.C Observer Initialization

For Assumption 9.4 to hold, we need to construct a bounded set $\tilde{\mathcal{X}}$ containing the initial state. Fortunately, this can be achieved for our class of systems in a finite number of steps, as detailed in this Section. During these first few steps, the PSTC must trigger periodically with $\kappa^* = 1$. The construction of $\tilde{\mathcal{X}}$ requires the following:

Assumption 9.5. *The matrix $\Phi_\mathrm{p}(1)$ is invertible and the pair $(\Phi_\mathrm{p}(1), C_\mathrm{p})$ is observable.*

This is not a limiting assumption: one can always find h such that $\Phi_\mathrm{p}(1) = \mathrm{e}^{A_\mathrm{p}h}$ is invertible.[9] Likewise, since the pair $(A_\mathrm{p}, C_\mathrm{p})$ is observable, so is $(\Phi_\mathrm{p}(1), C_\mathrm{p})$ with the proper selection of h.[10] For compactness of expressions, denote $\Phi_\mathrm{p}(1)$ as Φ_p and $\Gamma_\mathrm{p}(1)$ as Γ_p throughout the rest of this Appendix.

Instead of following the standard recursive GSE, which would require Minkowski sums of unbounded sets,[11] we collect sets relating the current state to each specific measurement

[9] For $h = 0$, $\mathrm{e}^{A_\mathrm{p}h} = \mathbf{I}$; from continuity, $\mathrm{e}^{A_\mathrm{p}h} \approx \mathbf{I}$ for small enough values of h, hence it is invertible.

[10] See [125, Sec. 6.8] for the pathological selections of h for which it does not hold.

[11] There are tools for that, but it is both unnecessary and computationally inefficient to do so. During the initialization, the STC has to trigger periodically, hence there is no advantage in keeping track of the best state estimate.

up to a certain instant, then compute an intersection outer-approximation. Let $\boldsymbol{O}(k)$ be the observability matrix for $k+1$ instants:

$$\boldsymbol{O}(k) := \begin{bmatrix} C_\mathrm{p} \\ C_\mathrm{p}\Phi_\mathrm{p} \\ \vdots \\ C_\mathrm{p}\Phi_\mathrm{p}^k \end{bmatrix}.$$

Denote $\bar{k} := \inf_{k\in\mathbb{N}_0} \operatorname{rank}(\boldsymbol{O}(k)) = n_\mathrm{p}$. This is the number of steps needed to reconstruct the initial state on linear systems. We will see that it is also the minimum number of steps for getting a bounded set estimate from measurements with bounded noise. For now, denote $\boldsymbol{\delta}(k_1,k_2) := \int_{hk_1}^{hk_2} \mathrm{e}^{A_\mathrm{p}(hk_2-s)}\boldsymbol{E}\boldsymbol{\omega}(s)\mathrm{d}s$ as the contribution of disturbances to state from k_1 to k_2, and let $\tilde{\boldsymbol{\psi}}(k,\bar{k}) := \boldsymbol{\psi}(hk) + C_\mathrm{p}\sum_{j=k}^{\bar{k}-1}\Phi_\mathrm{p}^{k-1-j}\Gamma_\mathrm{p}\hat{\boldsymbol{\upsilon}}(hj)$ be the prediction of the output at time $\bar{k}$ from the output at k and inputs from k to $\bar{k}-1$. The following holds:

Lemma 9.4. *Consider system* (9.9),(9.13) *with $b = k$ (periodic triggering), and let Assumption 9.4 hold. Then, for all $k \le \bar{k}$,*

$$C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}\xi_\mathrm{p}(h\bar{k}) \in \mathcal{E}(\tilde{\boldsymbol{\psi}}(k,\bar{k}),\boldsymbol{V}) + C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}\tilde{\mathcal{X}}_\mathrm{w}(\bar{k}-k).$$

Proof. We can assess the contribution of the information $\boldsymbol{\psi}(hk), k \le \bar{k}$ to the instant $\bar{k}$ in a similar manner to Eq.(9.20):

$$\xi_\mathrm{p}(h\bar{k}) = \Phi_\mathrm{p}^{\bar{k}-k}\xi_\mathrm{p}(hk) + \sum_{j=k}^{\bar{k}-1}\Phi_\mathrm{p}^{\bar{k}-1-j}\Gamma_\mathrm{p}\hat{\boldsymbol{\upsilon}}(hj) + \boldsymbol{\delta}(k,\bar{k}),$$

which implies, if Φ_p is invertible,

$$C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}\xi_\mathrm{p}(h\bar{k}) = C_\mathrm{p}\xi_\mathrm{p}(hk) + C_\mathrm{p}\sum_{j=k}^{\bar{k}-1}\Phi_\mathrm{p}^{k-1-j}\Gamma_\mathrm{p}\hat{\boldsymbol{\upsilon}}(hj) + C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}\boldsymbol{\delta}(k,\bar{k}). \tag{9.35}$$

Since $C_\mathrm{p}\xi_\mathrm{p}(hk) = \boldsymbol{\psi}(hk) - \boldsymbol{\nu}(hk)$, it belongs to the input uncertainty set $\mathcal{E}(\boldsymbol{\psi}(kh),\boldsymbol{V})$, which after summing with the contribution from inputs $C_\mathrm{p}\sum_{j=k}^{\bar{k}-1}\Phi_\mathrm{p}^{k-1-j}\Gamma_\mathrm{p}\hat{\boldsymbol{\upsilon}}(hj)$ yields $\mathcal{E}(\tilde{\boldsymbol{\psi}}(k,\bar{k}),\boldsymbol{V})$. The remaining term is the contribution from disturbances after $\bar{k}-k$ steps, which belongs to $\tilde{\mathcal{X}}_\mathrm{w}(\bar{k}-k)$, followed by the linear transformation through $C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}$. □

Denote the outer-approximation (Eq. 9.11) of the Minkowski sum in Lemma 9.4 as $\mathcal{E}(\tilde{\boldsymbol{\psi}}(k,\bar{k}),\tilde{\boldsymbol{V}}(k))$. From Definition 9.8, if $C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}\xi_\mathrm{p}(h\bar{k}) \in \mathcal{E}(\tilde{\boldsymbol{\psi}}(k,\bar{k}),\tilde{\boldsymbol{V}}(k))$, then $\xi_\mathrm{p}(h\bar{k}) \in \mathcal{K}(\tilde{\boldsymbol{\psi}}(k,\bar{k}),\tilde{\boldsymbol{V}}(k),C_\mathrm{p}\Phi_\mathrm{p}^{k-\bar{k}}) = \tilde{\mathcal{X}}(\bar{k}|k)$. That is, we have found the elliptical cylinder that contains $\xi_\mathrm{p}(h\bar{k})$ given information at k. Since this is true for all $k \in \{0,1,..\bar{k}\}$, we have that

$$\xi_\mathrm{p}(h\bar{k}) \in \bigcap_{k=0}^{k\le\bar{k}}\tilde{\mathcal{X}}(\bar{k}|k). \tag{9.36}$$

An ellipsoidal outer-approximation of this intersection of elliptical cylinders can be derived with the following:

Lemma 9.5. *Let* $\boldsymbol{C}_i \in \mathbb{R}^{m\times n}, \boldsymbol{M}_i \in \mathbb{S}^n_{++}, \boldsymbol{y}_i \in \mathbb{R}^m, i \in \{1,...,q\}$. *Denote* $\bar{\boldsymbol{C}} \coloneqq [\boldsymbol{C}_1^\mathsf{T} \quad \boldsymbol{C}_2^\mathsf{T} \quad \cdots \quad \boldsymbol{C}_q^\mathsf{T}]$ *and assume* $\operatorname{rank}(\bar{\boldsymbol{C}}) = n$. *Denote* $\bar{\boldsymbol{y}}^\mathsf{T} \coloneqq [\boldsymbol{y}_1^\mathsf{T} \quad \boldsymbol{y}_2^\mathsf{T} \quad \cdots \quad \boldsymbol{y}_q^\mathsf{T}]$ *and*

$$\bar{\boldsymbol{M}} \coloneqq \begin{bmatrix} \frac{1}{\mu_1}\boldsymbol{M}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{\mu_2}\boldsymbol{M}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \frac{1}{\mu_q}\boldsymbol{M}_q \end{bmatrix},$$

with $\sum_{i=1}^q \mu_i = 1$. *Then,*

$$\cap_i \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{M}_i, \boldsymbol{C}_i) \subseteq \mathcal{E}(\bar{\boldsymbol{C}}^\dagger \bar{\boldsymbol{y}}, \bar{\boldsymbol{C}}^\dagger \bar{\boldsymbol{M}} \bar{\boldsymbol{C}}^{\dagger\mathsf{T}}).$$

Proof. The intersection means that $(\boldsymbol{C}_i\boldsymbol{x} - \boldsymbol{y}_i)^\mathsf{T}\boldsymbol{M}_i^{-1}(\boldsymbol{C}_i\boldsymbol{x} - \boldsymbol{y}_i) \le 1$ for all i; thus, it holds that $\sum_{i=1}^q \lambda_i(\boldsymbol{C}_i\boldsymbol{x} - \boldsymbol{y}_i)^\mathsf{T}\boldsymbol{M}_i^{-1}(\boldsymbol{C}_i\boldsymbol{x} - \boldsymbol{y}_i) \le \sum_{i=1}^q \lambda_i$ for any $\lambda_i > 0$. Divide both sides by $\sum_{i=1}^q \lambda_i$ and denote $\mu_i = \lambda_i/(\sum_{i=1}^q \lambda_i)$. Putting in matrix form,

$$(\bar{\boldsymbol{C}}\boldsymbol{x} - \bar{\boldsymbol{y}})^\mathsf{T} \begin{bmatrix} \mu_1\boldsymbol{M}_1^{-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \mu_2\boldsymbol{M}_2^{-1} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \mu_q\boldsymbol{M}_q^{-1} \end{bmatrix} (\bar{\boldsymbol{C}}\boldsymbol{x} - \bar{\boldsymbol{y}}) \le 1.$$

The middle matrix is $\bar{\boldsymbol{M}}^{-1}$. Hence, $\bar{\boldsymbol{C}}\boldsymbol{x} \in \mathcal{E}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{M}})$. Since $\bar{\boldsymbol{C}}$ is full rank, then $mq \ge n$, which implies that $\bar{\boldsymbol{C}}^\dagger\bar{\boldsymbol{C}} = \mathbf{I}$. Therefore, $\boldsymbol{x} = \bar{\boldsymbol{C}}^\dagger\bar{\boldsymbol{C}}\boldsymbol{x} \in \bar{\boldsymbol{C}}^\dagger\mathcal{E}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{M}})$. Finally, applying the linear transformation on the latter ellipsoid gives $\boldsymbol{x} \in \mathcal{E}(\bar{\boldsymbol{C}}^\dagger\bar{\boldsymbol{y}}, \bar{\boldsymbol{C}}^\dagger\bar{\boldsymbol{M}}\bar{\boldsymbol{C}}^{\dagger\mathsf{T}})$. □

Finally, using the fact that $\boldsymbol{O}(\bar{k})$ is full-rank, we apply Lemma 9.5 with $\mu_i = \bar{k} + 1$ to Eq. (9.36), obtaining the main initialization step:

Theorem 9.5. *Let* $\bar{\boldsymbol{O}}(\bar{k}) \coloneqq \boldsymbol{O}(\bar{k})\Phi_\mathrm{p}^{-\bar{k}}$ *and*

$$\bar{\boldsymbol{\psi}}(\bar{k})^\mathsf{T} \coloneqq \begin{bmatrix} \tilde{\boldsymbol{\psi}}(0,\bar{k})^\mathsf{T} & \tilde{\boldsymbol{\psi}}(1,\bar{k})^\mathsf{T} & \cdots & \boldsymbol{\psi}(\bar{k})^\mathsf{T} \end{bmatrix},$$

$$\bar{\boldsymbol{V}}(\bar{k}) \coloneqq \begin{bmatrix} (\bar{k}+1)\tilde{\boldsymbol{V}}(0) & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & (\bar{k}+1)\tilde{\boldsymbol{V}}(1) & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & (\bar{k}+1)\tilde{\boldsymbol{V}}(\bar{k}) \end{bmatrix}.$$

Then $\boldsymbol{\xi}_\mathrm{p}(h\bar{k}) \in \mathcal{E}\left(\bar{\boldsymbol{O}}(\bar{k})^\dagger\bar{\boldsymbol{\psi}}(\bar{k}), \bar{\boldsymbol{O}}(\bar{k})^\dagger\bar{\boldsymbol{V}}(\bar{k})\bar{\boldsymbol{O}}(\bar{k})^{\dagger\mathsf{T}}\right)$.

Matrices $\bar{\boldsymbol{O}}(\bar{k})^\dagger, \bar{\boldsymbol{V}}(\bar{k}), \bar{\boldsymbol{O}}(\bar{k})^\dagger\bar{\boldsymbol{V}}(\bar{k})\bar{\boldsymbol{O}}(\bar{k})^{\dagger\mathsf{T}}$ and $\Phi_\mathrm{p}^{-k}, k \in \{1,...,\bar{k}\}$ can be computed offline. Online, $\tilde{\psi}(k,\bar{k})$ are calculated and, at $k = \bar{k}$, the center of the state estimate $\tilde{\mathcal{X}}$, $\bar{\boldsymbol{O}}(\bar{k})^\dagger\bar{\boldsymbol{\psi}}(\bar{k})$ is computed. The main loop with Algorithm 4 then follows.

10

Conclusion

Throughout this disseration, we have addressed a wide range of topics concerning the modeling of the timing of transmissions in event-triggered control systems, as well as their applications on scheduling, evaluating, and improving such ETC systems. The methods presented herein have strong relevance to the ETC community, as they shed light on surprisingly overlooked aspects of ETC: how relevant ETC communications savings actually are, and how to benefit from this reduction when implementing ETC systems in shared networks. Answering these questions is paramount in determining ETC's practical relevance over the well-established periodic sampling, and this dissertation provides preliminary answers, but perhaps most importantly it suggests a methodology to address them.

As a positive byproduct of this research, there were many interesting insights that we have gained about ETC/STC systems, their sampling behavior and their potential on real-life cyber–physical applications. We summarize these key learning points in Section 10.1. Nevertheless, despite how much we have learned, there is much work to be done before having more concrete and general answers to our main questions. Some of these answers may benefit from possible improvements to this work, which we discuss in Section 10.2. Finally, some interesting directions for future work in connected topics are discussed in Section 10.3.

10.1 Key learning points

Nominal linear PETC traffic models admit exact abstractions. As we have seen in Chapters 3–5, every linear PETC system of the form (2.2)–(2.4) (i.e., static linear state feedback, zero-order control, quadratic triggering conditions) admits l-complete abstractions that are exact in the sense that every abstract state is related to a concrete state and vice-versa, and their outputs are the same. This is not true for CETC systems, because their output space is infinite, and it is unclear whether it is true for systems subject to bounded disturbances (see Section 10.2).

The "computability" of our exact abstractions rely on the decidability of the first-order theory over the reals (known as Tarksi's Theorem [67, 126]), which allows one to decide

whether or not a boolean formula over polynomial inequalities admits a real solution. The reason why we put quotes around the word computability is because, to arrive at the main quadratic constraint problems (e.g., (5.2)), we first compute the matrix exponential, which (i) typically has transcendental numbers as entries and (ii) is not exactly computable for a dimension larger than 4.[1] Hence, the precise answer of computability of such models is more complex than what we have discussed; nevertheless, as we have seen in Chapter 5, l-complete PETC abstractions are in general robust to sufficiently small perturbations in the system's matrices, hence exact computability may play a lesser role in this theoretical inquiry. Clearly, computability of l-complete ETC traffic models of *discrete-time* linear systems has a simpler positive answer, as long as the matrices have algebraic entries, because matrix powers are used instead of matrix exponentials.

It is easy to see that l-complete traffic models can be obtained if one considers (i) other control update methods instead of sample-and-hold, such as to-zero [127] and model-based [128]; (ii) other polynomial triggering conditions (including those based on 1-norm and infinity-norm of the state); and (iii) linear dynamic controllers, even output-based. Of course, for the latter case the abstraction relies on full-state information.

ETC scheduling is hard when ETC is needed. The heavy machinery required to build traffic abstractions and solve the scheduling problems in Chapters 3 and 8 may give the wrong impression that ETC scheduling is always hard. In fact this is not the case: consider a network in which each transmission occupies the channel for Δ time units with p ETC systems whose MISTs are equal or larger than Δp. In this case, one can see that Round Robin effectively ensures that all systems meet their ETC-related deadlines, and it is by design conflict-free. However, this strategy executes the transmissions of each system periodically with period Δp each. Hence, one could argue that ETC is not necessary at all given the network capacity in this scenario. Only if the user is further interested in reducing the number of transmissions (see next learning point below) it is worth using our machinery; or if the network capacity does not actually allow for such a trivial scheduling solution. We argue that these situations are the ones that ETC is in fact needed.

Still, without sufficient refinement, it seems that our the abstractions are often reducible (as per our minimization algorithm in Chapter 8) to simple periodic traffic models admitting early sampling as Fig. 8.5. In fact, one can see that this reduction always happens if the abstraction has a state $\underline{k}\sigma$ such that $(\underline{k}\sigma, i, \underline{k}\sigma) \in \mathcal{E}$ for all $i \in \{1, 2, ...\underline{k}\}$, where $\underline{k}$ is the MIST of the system. In other words, there is a state whose deadline is the MIST and sampling before or at the deadline leads back to this state. If this holds, then the abstraction includes in it the model of a periodic task with deadline $\underline{k}$ admitting early executions $\mathcal{S}^{\mathrm{P}} = (\{x\}, \{x\}, \{1, 2, ...\underline{k}\}, \{(x, 1, x), (x, 2, x), ..., (x, \underline{k}, x)\}, \{\underline{k}\}, x \mapsto \underline{k})$. Intuitively, this state becomes the bottleneck for scheduling: no strategy can leave it, and the choice of sampling actions is the most restrictive of all. Unsurprisingly, applying the algorithm of Chapter 8 to a system with these conditions ends up with the W-T representation of $\mathcal{S}^{\mathrm{P}}$ alone. This would also be the case for the numerical example of Chapter 3.

ETC and STC are particularly relevant when sampling or transmitting is costly. As we have discussed in the introduction, reducing the number of samples required for

[1] This is because its exact methods rely on diagonalization or Jordan factorization, which relies on finding its eigenvalues, which needs computing the exact roots of a polynomial of degree equal to the matrix dimension, which is famously impossible in general for polynomials of degree 5 or more.

control is not only important to accommodate more control loops in a shared network. Reducing the number of transmissions in a network is useful to, e.g., accommodate control-unrelated traffic such as configuration and diagnostics data, or reduce radio energy in wireless networks. With the results we have obtained in Chapters 5, ETC can substantially reduce the average IST compared to periodic control, and this performance can be further improved using early-sampling and look-ahead (Chapter 7). Nonetheless, in the particular case where radio energy reduction is aimed at, this reduction in transmissions has little value if the controller has to continuously listen for sensor packets. Our traffic models can be used in this context to inform the controller of the next transmission time(s), finally realizing this energy reduction. In practice, though, with perturbed systems, this prediction can be very inaccurate as the system approaches equilibrium (Chapter 9). This is one of the reasons why we have in parallel worked on a tailor-made network stack, the Wireless Control Bus [37], which schedules listening times in very short time windows leveraging the power of concurrent transmissions, making it particularly useful for turning PETC transmission savings into effective energy savings.

ETC can exhibit complex traffic behavior. While most of the ETC literature has observed traffic patterns that eventually converge to some fixed value or exhibit well-behaved oscillatory, quasi-periodic patterns, we have shown in Chapter 6 that ETC's traffic pattern is much richer than this. In fact, it seems that complex and chaotic patterns arise when the triggering condition is more relaxed, i.e., it is designed to significantly reduce the amount of transmissions.

ETC is by far not optimal in resource usage minimization. As we have seen in Chapter 7, ETC is a greedy approach to sampling reduction, and by employing look ahead and early triggers one can come up with substantially better sampling strategies; one way of obtaining them is by solving mean-payoff games on our traffic abstractions.

10.2 Opportunities for improvements

10.2.1 Computational complexity

Throughout the discussion sections of each chapter in this dissertation, computational complexity was a recurring topic. To recap, there are two main layers of computational complexity: generating the abstractions and solving the scheduling or optimal strategy synthesis problem. We believe we have made a significant contribution to the latter with Chapter 8, although extensions for quantitative automata are yet to be explored. For the former, let us recap the complexities of our abstraction methods.

The number of states $|\mathcal{X}_l|$ of an l-complete PETC abstraction is in $\mathcal{O}(\bar{k}^l)$, but as we have seen in Chapter 6, this exponential increase in the abstraction state-space is only observed in behaviorally chaotic systems. The size of the transition set is in $\mathcal{O}(|\mathcal{X}_l|^2)$ for verification and in $\mathcal{O}(|\mathcal{X}_l|^2\bar{k})$ for synthesis, i.e., when augmenting the models with early triggers. The overall computational complexity of building an l-complete abstraction is thus $\mathcal{O}(\bar{k}|\mathcal{X}|^2)C$, where C is the complexity of verifying a single transition (x, u, x'). C is polynomial in n_x when using semi-definite relaxations (Chapter 3) and exponential in n_x when using an exact nonlinear SMT solver such as z3 — more precisely, $(l|\bar{k}|)^{n_x}2^{\mathcal{O}(n_x)}$ by adapting the result from Prop. 4.3. Because checking each transition from an l-complete model is independent of checking any other transition, the abstraction process is highly

parallelizable. The dominating factor, one could argue, is thus the complexity of verifying a single transition, C.

Clearly, the SDR approach from Chapter 3 is on one end of the computational complexity spectrum, and the Z3 approach is on the other end (with the respective trade-offs in precision). Exploiting other approaches in the middle may be worthwhile, such as sum-of-squares programming and δ-SMT [69]. The latter seems particularly promising given the general robustness of l-complete models to small errors in the system model presented in Chapter 5. If one can work the actual bounds better, e.g., what is the value of $\delta(l)$ such that a $\delta(l)$ perturbation in the system's matrices retain the same l-complete model, this may be a very promising approach.

In addition, as discussed in previous chapters, other triggering conditions may allow for the use of specialized solvers. With triggering conditions based on 1-norm or infinity-norm, the isosequential subsets are finite unions and intersections of polyhedra; therefore, the satisfiability problem involved in verifying an inter-sample sequence or transition relation may be cast as, e.g., a mixed-integer linear programming (MILP) problem. While MILPs are also known to have exponential complexity on the number of variables, there is a multitude of well-established solvers that perform very well in practice.

Finally, notice the strong dependence of the complexity numbers on $\bar{k}$ and l. For $\bar{k}$, this basically implies that it is easier and faster to abstract systems with large checking times h, but there is not much to be done on that front. When it comes to l, the best approach we have found for improving computations on a practical side is to (i) avoid constructing an l-complete abstraction starting at a large value of l, but instead incrementing l starting at 1 and only verifying the domino-consistent sequences of length $l+1$, and (ii) for SAIST computation, avoiding the full l-complete model by using some sort of lazy abstraction technique [129] that was briefly discussed in Section 5.6.2. This is what we have implemented in `ETCetera`. This lazy abstraction approach could further be explored for schedulability purposes, by iterating over an abstract-minimize-refine cycle using the minimization technique of Chapter 8.

10.2.2 Specialized abstractions for perturbed systems

When external bounded disturbances are present, the IST from a given state generally depends on the disturbance realization (Chapter 9). Approximate abstractions using power simulations [22] can be obtained using reachability analysis as we do in Chapter 9 for STC and was already done for general nonlinear systems in [30]. Specializing [30] for linear systems has the main advantage that the partitioning of the state-space can be better guided by, e.g., combining the cones obtained in the nominal quotient/l-complete systems with the proximity of the state to the origin (similarly to [23]).

To obtain exact abstractions, one could consider a traffic model in which the state is the pair $(\boldsymbol{x}_i, k_{i-1})$, i.e., the current sample and the previously observed inter-sample time, and the output is k_{i-1} instead of k_i — this trick was applied for abstracting linear systems subject to stochastic disturbances in [31]. For bounded non-deterministic disturbances, though, it is unclear at this point whether constructing the quotient model of such a system without spurious states is possible; for continuous-time systems, this requires verifying whether there is a point in $\mathbb{R}^{n_x}$ such that there is a disturbance realization rendering its IST equal to k_{i-1} and its reached point being $\boldsymbol{x}_i$. This depends in part on the decidability

of time-bounded, input-bounded reachability for continuous-time linear systems, which is just recently being investigated [130]: what is known so far is that it is decidable for hyper-cube-bounded inputs depending on the truth of Schanuel's conjecture, a conjecture in transcendence theory. For energy-bounded ($\mathcal{L}_2$) disturbances, ellipsoidal reachability may again be the best tool to use, as the reachable set is also an (approximately computable, at least) ellipsoid.

10.2.3 Other opportunities

Improved abstractions for synthesis. Our refinement approach (moving from an l to an $(l+1)$-complete model) is based on abstraction and refinement methods used for verification problems. In essence, our approach is not fundamentally different from the bisimulation algorithm from [17] for verification, where a region $\mathcal{R} \subset \mathcal{X}$ is split into $\mathcal{R} \cap \mathrm{Pre}(\mathcal{R}')$ and $\mathcal{R} \setminus \mathrm{Pre}(\mathcal{R}')$, with $\mathrm{Pre}(\mathcal{R}) := \{x \in \mathcal{X} \mid (x, x') \in \mathcal{E} \text{ for some } x' \in \mathcal{R}\}$. Only after the refinements are concluded we augment them with the early-sampling actions. This is different than the bisimulation algorithm for control, in which the Pre operator is $\mathrm{CPre}(\mathcal{R}) := \{x \in \mathcal{X} \mid (x, u, x') \in \mathcal{E} \text{ for some } u \in \mathcal{U}, x' \in \mathcal{R}\}$ (sometimes called controlled Pre). Note that using CPre changes how the regions are refined: in fact, CPre gives a larger set than using Pre on the autonomous system. The results based on existence of finite alternating bisimulation or simulation equivalence for infinite systems rely on the CPre operator (see, e.g., [92]); thus, there is no guarantee that our refinement approach can generate finite bisimulations of the control system — it may end with a finite bisimulation for the autonomous system, but after early-sample actions are added this property may be lost.

One reason we did not use CPre is that, for typical control synthesis, the chosen control action is irrelevant as long as it ensures the trace satisfies the given specifications. This is not the case for our scheduling / sampling strategy problems: the control action (when to trigger) matters; in fact, it would be more natural to define an output map on the transition rather than on the state, which is a common thing in the automata literature (notably the difference between Mealy and Moore machines). We did this somewhat implicitly in Chapter 7, but admittedly we only noticed this nuance late in the project. A natural next step is thus to investigate how to either adapt the traffic model so that the actual chosen inter-sample time is the output map, rather than a deadline, or using a version of the bisimulation algorithm for systems with output maps on transitions. The main operations involved in the abstraction process, namely verifying conjunctions of quadratic inequalities, should remain the same, and this approach should yield tighter abstractions for synthesis.

Lower entropy bounds. While we have provided a method to compute upper bounds on a PETC system's behavioral entropy, getting lower bounds would be very useful. Not only it provides the theoretically interesting answer that a system is in fact chaotic (when the lower bound is positive), it also informs the abstraction process that it is in some sense worthless to keep refining the abstraction further. In Chapter 6 we have discussed some ideas on how to do it, but the vast recent literature on abstractions can be further explored. One example is [131], which provides a bisimulation-like algorithm whose output abstraction's infinite traces are equivalent to those of the concrete system in some cases, even when a finite bisimulation is not available.

Linear CETC abstractions. Our abstractions have been mostly focused on PETC, both due to its practicality and because it is more amenable to abstracting thanks to its finite output space. Nevertheless, a similar approach to what we use in PETC can be done to create the quotient state-space of CETC in the special case where the CETC system satisfies the conditions of Proposition 6.3, i.e., its associated f map is continuous. One can see that, in this case, the set $\mathcal{Q}_{[\tau_1,\tau_2]}$ of points whose IST is in $[\tau_1, \tau_2]$ is the set $\{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{x}^T\boldsymbol{N}(\tau_1)\boldsymbol{x} \geq 0, \boldsymbol{x}^T\boldsymbol{N}(\tau_2)\boldsymbol{x} \leq 0\}$, i.e., it is defined by a simple pair of quadratic inequalities. This effectively creates the power quotient state-set, which can be defined with arbitrary precision (depending on the division of the time interval $[\underline{\tau}, \bar{\tau}]$. Building the transition set, however, requires more sophisticated reachability analysis than our approach for PETC, but a natural first step is to use the same LMI approach as in [21].

The scheduling problem without late triggers is too stringent. Take a PETC system with MIST equal to k and take $\mathcal{Q}_k$ as its corresponding isochronous set. It is often the case that, using only early triggers, it takes many samples before the system leaves $\mathcal{Q}_k$ if $\boldsymbol{x}_0 \in \mathcal{Q}_k$. Suppose the maximum number of samples is L. This means that for any $l < L$, the transitions $k^l \xrightarrow{i} k^l$, for all $i \in \mathbb{N}_{\leq k}$, will be present in the corresponding l-complete PETC traffic model with early triggering. It is possible to see that, regardless of the other states in the abstraction, applying the minimization algorithm of Chapter 8 on the derived discrete-clock model gives a trivial solution as in Fig. 8.5: a system with a recurring deadline of k time units. This reasoning not only indicates that often one needs to have sufficiently refined abstractions to obtain scheduling problems that do not reduce to simple periodic deadlines (equivalent as such to periodic sampling), but also that PETC systems may indeed persist in behaving like periodically sampled systems until sufficient time has passed. In these situations, PETC is indeed not much better to liberate network capacity as its periodic sampling counterpart, *if* all the scheduler can do is trigger early. Allowing occasional late samples, however, may allow for a quicker escape from $\mathcal{Q}_k$, getting faster to a region of the state space where PETC performs sufficiently better.

However, as we have seen, early samples guarantee by design control stability and performance given by the triggering condition. This is not the case with late triggers, hence they must be used wisely. One approach that ensures stability but ignores transient behaviors is to allow finitely many late triggers in every infinite run, turning the scheduling problem in essence a Büchi safety game. Other approaches may have higher computational complexity but attain better control performance, such as posing the scheduling problem as a quantitative safety game, where the weight map on a transition represents the Lyapunov function decay (or increase) by taking that choice. These approaches are currently subject of investigation in the group.

10

10.3 Future research directions

Can ETC actually increase control network capacity? In the examples we have explored for PETC scheduling, in all cases where the safety game had a winning strategy, there was also a trivial periodic schedule available (this was verified case by case, but made clearer after the case study of Chapter 8). In all cases where we added an extra control loop such that this trivial solution was not possible, we failed to obtain a scheduler. We have no reason to believe that this is always true, and it would be interesting to find a

counterexample; this would show that the transmission reductions provided by PETC can indeed enable larger networks of control systems. We suspect that one can find such an example with the scheduling approach we proposed, but this will certainly be easier when allowing late triggers.

Solving the scheduling problem (and others) with disturbances and partial information. The combination of our scheduling methods for the nominal case and the observer-based STC of Chapter 9 requires overcoming many challenges, beyond the abstraction-related discussed in Section 10.2. As we have seen in Chapter 9, when measurement noise is present, it is never possible to have the exact state information, but instead a bounded set containing it. If this set overlaps two different abstract states, the state-based scheduling strategies we design may not be able to find a coherent action. In fact, the scheduling problem becomes a partial-information safety game, which is in general undecidable. Nevertheless, there are many approaches in the literature of discrete-event systems dealing with partial information synthesis which can be explored (e.g., see [95]). The full solution envisioned, involving set-based observers combined with automata-based observers, is substantially more complex than the already not simple ETC scheduling problem we have addressed, but definitely possible to devise.

Other metrics derived from the traffic models. Even though we have built our abstractions for predicting, evaluating, and manipulating PETC traffic, they contain much more information than the mere IST behaviors of the system. One also gets how trajectories can traverse the state space, which gives information about control performance as well. One can leverage those models to derive improved Lyapunov functions yielding tigher bounds on the GES decay rate of the system, expanding on what we have briefly done in Chapter 4 — a potential way to do so is exploring path-complete Lyapunov functions [68]. Alternatively, other metrics can be obtained by choosing the appropriate weight function in the abstractions, such as Lyapunov decay rates; these metrics could be combinations of traffic and control performance indicators.

Connection with piecewise-affine systems. The PETC traffic model from Definition 2.15 can be seen as a piecewise-affine (PWA) discrete-time system, with the particularity that the regions for each affine mode are complicated cone intersections as opposed to the typically convex polyhedra often consider in the PWA literature. This connection has not been explored in this dissertation, but it may be a fruitful one.

Exploiting chaotic traffic. Chaotic traffic patterns may be bad from a predictability standpoint, with potential negative effects in schedulability, but they may be interesting from a cyber-security perspective: it may become more difficult for an external observer to detect whether the seemingly random traffic is of important control data or not, or to inject destructive information if the listening times from the controller / actuator are timed precisely (as in the case of STC).

ETC for shared actuators. This dissertation started with an admittedly awkward introduction to event-based mechanisms for control, by explaining how we manage multiple control tasks with our pair of actuators called arms. Still, it is interesting to see that the main practical motivation for ETC has been reduction in computation and network resources, but the problem of shared actuators has not been discussed (to the best of the author's knowledge). The most likely reason is that engineered control systems are often

over-actuated — not in the classical control-theoretic notion of over- and under-actuation — but in the sense that every component of the input space has a dedicated actuator. However, one could envision a scenario where, for example, a single robotic arm is responsible for operating several valves (like in our cooking scenario); this is particularly interesting as we consider control applications where the whole system cannot be simply overhauled with smart actuators, either because its too costly or impractical, hence requiring a moving robot to execute the several control tasks. One clear example is agriculture, where an autonomous vehicle may be responsible for controlling crops in a large area.

Bibliography

References

[1] "HART Technology Explained | FieldComm." [Online]. Available: https://www.fieldcommgroup.org/technologies/hart/hart-technology-explained

[2] R. E. Hufnagel, "Analysis of cyclic-rate sampled-data feedback-control systems," *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 77, no. 5, pp. 421–425, 1958.

[3] K. J. Åström and B. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, vol. 2. IEEE, 2002, pp. 2011–2016.

[4] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 3270–3285.

[5] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[6] E. Sontag, "Input to state stability," in *The Control Systems Handbook: Control System Advanced Methods, Second Edition.*, W. S. Levine, Ed. Boca Raton: CRC Press, 2011, pp. 45.1–45.21 (1034–1054).

[7] X. Wang and M. D. Lemmon, "Event design in event-triggered feedback control systems," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 2105–2110.

[8] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1992–1997, 2015.

[9] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 435–440.

[10] M. Mazo Jr. and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *Automatic Control, IEEE Transactions on*, vol. 56, no. 10, pp. 2456–2461, 2011.

[11] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, 2013.

[12] M. Mazo Jr and A. Fu, "Decentralized event-triggered controller implementations," *Event-Based Control and Signal Processing*, p. 121, 2015.

[13] V. Dolk, D. P. Borgers, and W. Heemels, "Output-based and decentralized dynamic event-triggered control with guaranteed lp-gain performance and zeno-freeness," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 34–49, 2017.

[14] M. Mazo Jr., A. Anta, and P. Tabuada, "An ISS self-triggered implementation of linear controllers," *Automatica*, vol. 46, no. 8, pp. 1310–1314, 2010.

[15] M. Velasco, J. Fuertes, and P. 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)*, vol. 384, 2003.

[16] A. Anta and P. Tabuada, "Self-triggered stabilization of homogeneous control systems," in *American Control Conference, 2008.* IEEE, 2008, pp. 4129–4134.

[17] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach.* Springer Science & Business Media, 2009.

[18] T. Moor and J. Raisch, "Supervisory control of hybrid systems within a behavioural framework," *Systems & control letters*, vol. 38, no. 3, pp. 157–166, 1999.

[19] K. Chatterjee, L. Doyen, and T. A. Henzinger, "Quantitative languages," *ACM Transactions on Computational Logic (TOCL)*, vol. 11, no. 4, pp. 1–38, 2010.

[20] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Transactions on automatic control*, vol. 36, no. 3, pp. 259–294, 1991.

[21] A. S. Kolarijani and M. Mazo Jr, "A formal traffic characterization of LTI event-triggered control systems," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 274–283, 2016.

[22] M. Mazo Jr, A. S. Kolarijani, D. Adzkiya, and C. Hop, "Abstracted models for scheduling of event-triggered control data traffic," in *Control Subject to Computational and Communication Constraints.* Springer, 2018, pp. 197–217.

[23] A. Fu and M. Mazo Jr., "Traffic models of periodic event-triggered control systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3453–3460, 2018.

[24] B. Demirel, V. Gupta, D. E. Quevedo, and M. Johansson, "On the trade-off between communication and control cost in event-triggered dead-beat control," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2973–2980, 2017.

[25] R. Postoyan, R. G. Sanfelice, and W. P. M. H. Heemels, "Inter-event times analysis for planar linear event-triggered controlled systems," in *Decision and Control, 2019. CDC 2019. 58th IEEE Conference on*, 2019.

[26] A. Rajan and P. Tallapragada, "Analysis of inter-event times for planar linear systems under a general class of event triggering rules," in *2020 59th IEEE Conference on Decision and Control (CDC).* IEEE, 2020, pp. 5206–5211.

[27] A. Ehrenfeucht and J. Mycielski, "Positional strategies for mean payoff games," *International Journal of Game Theory*, vol. 8, no. 2, pp. 109–113, 1979.

[28] G. Delimpaltadakis and M. Mazo Jr, "Traffic abstractions of nonlinear homogeneous event-triggered control systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 4991–4998.

[29] G. Delimpaltadakis and M. Mazo Jr., "Isochronous partitions for region-based self-triggered control," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1160–1173, 2020.

[30] G. Delimpaltadakis and M. Mazo Jr., "Abstracting the traffic of nonlinear event-triggered control systems," 2021.

[31] G. Delimpaltadakis, L. Laurenti, and M. Mazo Jr., "Abstracting the sampling behaviour of stochastic linear periodic event-triggered control systems," in *60th IEEE Conference on Decision and Control (accepted)*, 2021, https://arxiv.org/abs/2109.14391.

[32] C. Peng and T. C. Yang, "Event-triggered communication and h-∞ control co-design for networked control systems," *Automatica*, vol. 49, no. 5, pp. 1326–1332, 2013.

[33] M. Donkers, P. Tabuada, and W. Heemels, "Minimum attention control for linear systems," *Discrete Event Dynamic Systems*, vol. 24, no. 2, pp. 199–218, 2014.

[34] G. Delimpaltadakis, G. de A. Gleizer, I. van Straalen, and M. Mazo Jr., "ETCetera: beyond event-triggered control," in *25th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3501710.3519523

[35] I. van Straalen, "Efficient scheduler synthesis for periodic event triggered control systems: An approach with binary decision diagrams," Master's thesis, Delft University of Technology, 2021. [Online]. Available: http://resolver.tudelft.nl/uuid:f9764019-e908-45e7-a053-0f6fbc8a7792

[36] G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime, "Uppaal tiga user-manual," *Aalborg University*, 2007.

[37] M. Trobinger, G. A. Gleizer, T. Istomin, M. Mazo Jr, A. L. Murphy, and G. P. Picco, "The wireless control bus: Enabling efficient multi-hop event-triggered control with concurrent transmissions," *ACM Transactions on Cyber-Physical Systems (TCPS)*, vol. 6, no. 1, pp. 1–29, 2021.

[38] R. Milner, "An algebraic definition of simulation between programs," in *Proceedings of the 2nd international joint conference on Artificial intelligence*, 1971, pp. 481–489.

[39] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[40] A.-K. Schmuck, P. Tabuada, and J. Raisch, "Comparing asynchronous l-complete approximations and quotient based abstractions," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 6823–6829.

[41] R. M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete mathematics*, vol. 23, no. 3, pp. 309–311, 1978.

[42] M. Chaturvedi and R. M. McConnell, "A note on finding minimum mean cycle," *Information Processing Letters*, vol. 127, pp. 21–22, 2017.

[43] R. Alur, T. A. Henzinger, O. Kupferman, and M. Y. Vardi, "Alternating refinement relations," in *International Conference on Concurrency Theory*. Springer, 1998, pp. 163–178.

[44] J. Bengtsson and W. Yi, "Timed automata: Semantics, algorithms and tools," in *Advanced Course on Petri Nets*. Springer, 2004, pp. 87–124.

[45] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*. Courier Corporation, 2013.

[46] G. de A. Gleizer and M. Mazo Jr., "Scalable traffic models for scheduling of linear periodic event-triggered controllers," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2726–2732, 2020.

[47] R. Postoyan, A. Anta, W. P. M. H. Heemels, P. Tabuada, and D. Nešić, "Periodic event-triggered control for nonlinear systems," in *52nd IEEE conference on decision and control*. IEEE, 2013, pp. 7397–7402.

[48] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical computer science*, vol. 138, no. 1, pp. 3–34, 1995.

[49] T. A. Henzinger and P. W. Kopke, "Undecidability results for hybrid systems," Cornell University, Tech. Rep., 1995.

[50] A. S. Kolarijani, D. Adzkiya, and M. Mazo Jr., "Symbolic abstractions for the scheduling of event-triggered control systems," in *IEEE 54th Annual Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 6153–6158.

[51] J. Park and S. Boyd, "General heuristics for nonconvex quadratically constrained quadratic programming," *arXiv preprint arXiv:1703.07870*, 2017.

[52] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[53] A. S. Kolarijani and M. Mazo Jr, "Traffic characterization of lti event-triggered control systems: a formal approach," *arXiv preprint arXiv:1503.05816*, 2015.

[54] M. Donkers, "Networked and event-triggered control systems," Ph.D. dissertation, TU Eindhoven, 2011.

[55] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[56] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "SCS: Splitting conic solver, version 2.1.1," https://github.com/cvxgrp/scs, Nov. 2017.

[57] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, and J. H. Taankvist, "Uppaal stratego," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2015, pp. 206–211.

[58] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis, "Controller synthesis for timed automata," *IFAC Proceedings Volumes*, vol. 31, no. 18, pp. 447–452, 1998.

[59] P. Schalkwijk, "Automating scheduler design for networked control systems with event-based control: An approach with timed automata," Master's thesis, Delft University of Technology, 2019. [Online]. Available: http://resolver.tudelft.nl/uuid:6ae619f2-9247-4c30-9710-b1ddf362896d

[60] R. E. Bryant, "Symbolic boolean manipulation with ordered binary-decision diagrams," *ACM Computing Surveys (CSUR)*, vol. 24, no. 3, pp. 293–318, 1992.

[61] P. Bouyer, T. Brihaye, and N. Markey, "Improved undecidability results on weighted timed automata," *Information Processing Letters*, vol. 98, no. 5, pp. 188–194, 2006.

[62] A. A. Samant, "Scheduling strategies for event-triggered control using timed game automata over CAN networks," Master's thesis, Delft University of Technology, 2020. [Online]. Available: http://resolver.tudelft.nl/uuid:2dccaa3b-dbff-428e-a5d3-d46ada57504d

[63] G. de A. Gleizer and M. Mazo Jr., "Towards traffic bisimulation of linear periodic event-triggered controllers," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 25–30, 2021.

[64] L. De Moura and N. Bjørner, "Z3: An efficient SMT solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[65] E. Le Corronc, A. Girard, and G. Goessler, "Mode sequences as symbolic states in abstractions of incrementally stable switched systems," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 3225–3230.

[66] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in real algebraic geometry*. Springer, 2006.

[67] ——, "On the combinatorial and algebraic complexity of quantifier elimination," *Journal of the ACM (JACM)*, vol. 43, no. 6, pp. 1002–1045, 1996.

[68] A. A. Ahmadi, R. M. Jungers, P. A. Parrilo, and M. Roozbehani, "Joint spectral radius and path-complete graph Lyapunov functions," *SIAM Journal on Control and Optimization*, vol. 52, no. 1, pp. 687–717, 2014.

[69] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *International conference on automated deduction*. Springer, 2013, pp. 208–214.

[70] G. de A. Gleizer and M. Mazo Jr., "Computing the sampling performance of event-triggered control," in *Proc. of the 24th Int'l Conf. on Hybrid Systems: Computation and Control*, ser. HSCC '21. ACM, 2021.

[71] G. A. Gleizer and M. Mazo Jr., "Computing the average inter-sample time of event-triggered control using quantitative automata," 2021, https://arxiv.org/abs/2109.14391. Under review at Nonlinear Analysis: Hybrid Systems.

[72] S. Linsenmayer and F. Allgöwer, "Performance oriented triggering mechanisms with guaranteed traffic characterization for linear discrete-time systems," in *2018 European Control Conference (ECC)*, 2018, pp. 1474–1479.

[73] R. Postoyan, R. G. Sanfelice, and W. Heemels, "Explaining the "mystery" of periodicity in inter-transmission times in two-dimensional event-triggered controlled system," *IEEE Transactions on Automatic Control*, 2022, early access. DOI: 10.1109/TAC.2022.3147009.

[74] W. de Melo and S. van Strien, *One-Dimensional Dynamics*, ser. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics. Springer Berlin Heidelberg, 2012.

[75] G. A. Gleizer and M. Mazo Jr, "Chaos and order in event-triggered control," *arXiv preprint arXiv:2201.04462*, 2022.

[76] C. Robinson, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*, ser. Studies in Advanced Mathematics. CRC-Press, 1999. [Online]. Available: https://books.google.nl/books?id=px5gUpPfww8C

[77] A. Anta and P. Tabuada, "Exploiting isochrony in self-triggered control," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 950–962, 2011.

[78] R. A. Brualdi, "Spectra of digraphs," *Linear Algebra and its Applications*, vol. 432, no. 9, pp. 2181 – 2213, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379509001232

[79] I. Domowitz and M. A. El-Gamal, "A consistent test of stationary-ergodicity," *Econometric Theory*, vol. 9, no. 4, pp. 589–601, 1993.

[80] T. B. Arnold and J. W. Emerson, "Nonparametric goodness-of-fit tests for discrete null distributions." *R Journal*, vol. 3, no. 2, 2011.

[81] S. Day, R. Frongillo, and R. Trevino, "Algorithms for rigorous entropy bounds and symbolic dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 7, no. 4, pp. 1477–1506, 2008.

[82] D. Borgers and W. Heemels, "Event-separation properties of event-triggered control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2644–2656, 2014.

[83] A. Granas and J. Dugundji, *Fixed Point Theory*, ser. Monographs in Mathematics. Springer, 2003. [Online]. Available: https://books.google.nl/books?id=4_iJAoLSq3cC

[84] S. Lang, *Algebra*, ser. Graduate Texts in Mathematics. Springer New York, 2005. [Online]. Available: https://books.google.nl/books?id=Fge-BwqhqIYC

[85] G. de A. Gleizer, K. Madnani, and M. Mazo Jr., "Self-triggered control for near-maximal average inter-sample time," in *60th IEEE Conference on Decision and Control (accepted)*, 2021.

[86] A. Szymanek, G. A. Gleizer, and M. Mazo Jr., "Periodic event-triggered control with a relaxed triggering condition," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 1656–1661.

[87] D. Antunes, W. Heemels, and P. Tabuada, "Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design," in *2012 IEEE 51st IEEE conference on decision and control (CDC)*. IEEE, 2012, pp. 7212–7217.

[88] Y. Xu and J. P. Hespanha, "Optimal communication logics in networked control systems," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 4. IEEE, 2004, pp. 3527–3532.

[89] C. Comin and R. Rizzi, "Improved pseudo-polynomial bound for the value problem and optimal strategy synthesis in mean payoff games," *Algorithmica*, vol. 77, no. 4, pp. 995–1021, 2017.

[90] M. Mazo Jr. and P. Tabuada, "Symbolic approximate time-optimal control," *Systems & Control Letters*, vol. 60, no. 4, pp. 256–263, 2011.

[91] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin, "The complexity of multi-mean-payoff and multi-energy games," *Information and Computation*, vol. 241, pp. 177–196, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0890540115000164

[92] R. Majumdar, *Symbolic algorithms for verification and control*. University of California, Berkeley, 2003.

[93] G. A. Gleizer, K. Madnani, and M. Mazo Jr., "A simpler alternative: Minimizing transition systems modulo alternating simulation equivalence," in *25th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3501710.3519534

[94] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[95] C. G. Cassandras, S. Lafortune *et al.*, *Introduction to discrete event systems*. Springer, 2008, vol. 2.

[96] D. Bustan and O. Grumberg, "Simulation-based minimization," *ACM Transactions on Computational Logic (TOCL)*, vol. 4, no. 2, pp. 181–206, 2003.

[97] P. C. Kanellakis and S. A. Smolka, "Ccs expressions, finite state processes, and three problems of equivalence," *Information and computation*, vol. 86, no. 1, pp. 43–68, 1990.

[98] D. Lee and M. Yannakakis, "Online minimization of transition systems," in *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, 1992, pp. 264–274.

[99] J. A. Bergstra, A. Ponse, and S. A. Smolka, *Handbook of process algebra*. Elsevier, 2001.

[100] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke, "Computing simulations on finite and infinite graphs," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*. IEEE, 1995, pp. 453–462.

[101] F. Ranzato and F. Tapparo, "A new efficient simulation equivalence algorithm," in *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, 2007, pp. 171–180.

[102] R. Cleaveland and O. Sokolsky, "Equivalence and preorder checking for finite-state systems," *Handbook of Process Algebra*, pp. 391–424, 2001.

[103] T. A. Henzinger, R. Majumdar, and J.-F. Raskin, "A classification of symbolic transition systems," *ACM Transactions on Computational Logic (TOCL)*, vol. 6, no. 1, pp. 1–32, 2005.

[104] C. Fritz and T. Wilke, "State space reductions for alternating büchi automata quotienting by simulation equivalences," in *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 2002, pp. 157–168.

[105] K. Chatterjee, S. Chaubal, and P. Kamath, "Faster algorithms for alternating refinement relations," *Computer Science Logic 2012*, p. 167, 2012.

[106] G. de A. Gleizer and M. Mazo Jr., "Self-triggered output-feedback control of lti systems subject to disturbances and noise," *Automatica*, vol. 120, 2020, article no. 109129.

[107] ——, "Self-triggered output feedback control for perturbed linear systems," *IFAC-PapersOnLine*, vol. 51, no. 23, pp. 248–253, 2018.

[108] M. Donkers and W. Heemels, "Output-based event-triggered control with guaranteed-gain and improved and decentralized event-triggering," *Automatic Control, IEEE Transactions on*, vol. 57, no. 6, pp. 1362–1376, 2012.

[109] J. Almeida, C. Silvestre, and A. M. Pascoal, "Self-triggered output feedback control of linear plants in the presence of unknown disturbances." *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3040–3045, 2014.

[110] D. Nešić, A. R. Teel, G. Valmorbida, and L. Zaccarian, "Finite-gain lp stability for hybrid dynamical systems," *Automatica*, vol. 49, no. 8, pp. 2384–2396, 2013.

[111] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, "Constrained zonotopes: A new tool for set-based estimation and fault detection," *Automatica*, vol. 69, pp. 126–136, 2016.

[112] A. B. Kurzhanskiĭ and I. Vályi, *Ellipsoidal calculus for estimation and control*. Birkhäuser, 1997.

[113] ——, "Ellipsoidal toolbox," 2006, technical Report.

[114] F. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 22–28, 1968.

[115] L. Ros, A. Sabater, and F. Thomas, "An ellipsoidal calculus based on propagation and fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 4, pp. 430–442, 2002.

[116] L. G. Moreira, S. Tarbouriech, A. Seuret, and J. M. G. da Silva Jr, "Observer-based event-triggered control in the presence of cone-bounded nonlinear inputs," *Nonlinear Analysis: Hybrid Systems*, vol. 33, pp. 17–32, 2019.

[117] F. D. Brunner, W. P. M. H. Heemels, and F. Allgöwer, "Event-triggered and self-triggered control for linear systems based on reachable sets," *Automatica*, vol. 101, pp. 15–26, 2019.

[118] R. Goebel, R. G. Sanfelice, and A. Teel, "Hybrid dynamical systems," *Control Systems, IEEE*, vol. 29, no. 2, pp. 28–93, 2009.

[119] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[120] C. Cai and A. R. Teel, "Characterizations of input-to-state stability for hybrid systems," *Systems & Control Letters*, vol. 58, no. 1, pp. 47–53, 2009.

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

[122] J. Demmel, I. Dumitriu, and O. Holtz, "Fast linear algebra is stable," *Numerische Mathematik*, vol. 108, no. 1, pp. 59–91, 2007.

[123] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *Control Systems, IEEE*, vol. 21, no. 1, pp. 57–65, 2001.

[124] L. El Ghaoui and G. Calafiore, "Robust filtering for discrete-time systems with bounded noise and parametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 46, no. 7, pp. 1084–1089, 2001.

[125] M. Gopal, *Modern control system theory*. New Age International, 1993.

[126] A. Tarski, "A decision method for elementary algebra and geometry," *Journal of Symbolic Logic*, vol. 17, no. 3, 1952.

[127] L. Schenato, "To zero or to hold control inputs with lossy links?" *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1093–1099, 2009.

[128] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 422–434, 2012.

[129] T. A. Henzinger, R. Jhala, R. Majumdar, and G. Sutre, "Lazy abstraction," in *Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 2002, pp. 58–70.

[130] M. Dantam and A. Pouly, "On the decidability of reachability in continuous time linear time-invariant systems," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3447928.3456705

[131] A. J. Wagenmaker and N. Ozay, "A bisimulation-like algorithm for abstracting control systems," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 569–576.

Glossary

AIST Average inter-sample time (Section 5.2).

AS Alternating simulation, a relation between transition systems for control (Definition 2.9).

ASE Alternating simulation equivalence, an equivalence relation between two transition systems (Definition 2.11).

ASR Alternating simulation relation (Definition 2.9).

BABI Bijective alternating bisimulation isomorphism, an isomorphism on labeled graphs (or finite transition systems) allowing renaming of labels following certain rules (Def. 8.2).

CETC Continuous event-triggered control: ETC in which the triggering condition is checked in continuous time.

ETC Event-triggered control.

FTS Finite transition system, a generalized transition system (Def. 2.1) with finite state, input, output, and transition sets.

GSE Guaranteed state estimator, a set-based state estimation method for bounded uncertainties (Section 9.2.2).

ISS Input-to-state stability.

IST Inter-sample time, the time between two consecutive sampling instants in a sample-based control system.

LMI Linear matrix inequalities.

MIST Minimum inter-sample time, the smallest time between two consecutive samples being transmitted in a sampled-based controller.

MPETC Mixed periodic-event-triggered control, a modified PETC proposed in Chapter 4.

NCS Networked control system, a system where control devices, sensors, and actuators share information via computer networks.

NTGA Network of timed game automata.

PETC Periodic event-triggered control: ETC in which the triggering condition is checked periodically.

PSTC Preventive self-triggered control, an STC strategy where the IST is calculated to be early enough to prevent an associated ETC condition of triggering (Chapter 9).

QCQP Quadratically constrained quadratic programming, a class of mathematical programming where the objective function and constraints are quadratic functions.

SAC Smallest-in-average cycle, a cycle that attains minimal average weight among all cycles of a weighted graph.

SACE Smallest average-cycle equivalence, when two systems share a cyclic behavior in common that minimal average value (Definition 5.1).

SAIST Smallest average inter-sample time, smallest AIST across initial states, Eq. (5.1).

SCC Strongly connected component.

SDR Semi-definite relaxation.

SDSS State-dependent sampling strategy, a type of STC in which the sampling time is a function of the previously sampled state of the plant (Section 7.2).

SMT Satisfiability modulo theories.

STC Self-triggered control.

TGA Timed game automaton (Definition 2.13).

TSA Timed safety automaton (Definition 2.12).

WTS Weighted transition system, Definition 2.7.

ZOH Zero-order hold.

Curriculum Vitæ

Gabriel de Albuquerque Gleizer

05-10-1987 Born in Rio de Janeiro, Brazil.

Education

2005–2010 Control and Automation Engineering
Federal University of Rio de Janeiro
Grade: 9.1/10.0
Thesis: Multi-objective optimization applied to the mobility control of an articulated robot on irregular terrains
Supervisors: Prof. dr. Alexandre P. A. da Silva and Dr. Fernando Lizarralde

2010–2012 M.Sc. Electrical Engineering — Control, Automation and Robotics
Federal University of Rio de Janeiro
Grade: 3.0/3.0
Thesis: Optimal task scheduling for fieldbuses applied to FOUNDATION™ Fieldbus
Supervisors: Prof. dr. Liu Hsu and Dr. Danielle Zyngier

2018–2022 Ph.D. Systems and Control
Delft University of Technology
Grade: 9.6/10.0 (DISC average grade, certificate awarded)
Thesis: Timing is everything: Analysis and synthesis of traffic patterns in event-triggered control
Promotor: dr. Manuel Mazo Jr.
Promotor: Prof. dr. ir. Bart De Schutter

Awards

2010 A3P UFRJ Best Student Award

Work Experience

2010–2012	Control and Instrumentation Engineer LEAD – Federal University of Rio de Janeiro Rio de Janeiro, Brazil
2012–2017	Control Engineer / Lead Control Engineer GE Global Research Center Rio de Janeiro, Brazil

List of Publications

As a doctoral candidate

1. *G. de A. Gleizer* and M. Mazo Jr., "Self-triggered output feedback control for perturbed linear systems," *IFAC-PapersOnLine*, vol. 51, no. 23, pp. 248–253, 2018.
2. A. Szymanek, *G. de A. Gleizer*, and M. Mazo Jr., "Periodic event-triggered control with a relaxed triggering condition," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 1656–1661.
3. *G. de A. Gleizer* and M. Mazo Jr., "Self-triggered output-feedback control of lti systems subject to disturbances and noise," *Automatica*, vol. 120, 2020, article no. 109129.
4. *G. de A. Gleizer* and M. Mazo Jr., "Scalable traffic models for scheduling of linear periodic event-triggered controllers," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2726–2732, 2020.
5. *G. de A. Gleizer* and M. Mazo Jr., "Towards traffic bisimulation of linear periodic event-triggered controllers," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 25–30, 2021.
6. *G. de A. Gleizer* and M. Mazo Jr., "Computing the sampling performance of event-triggered control," in *Proc. of the 24th Int'l Conf. on Hybrid Systems: Computation and Control*, ser. HSCC '21. ACM, 2021.
7. M. Trobinger, *G. de A. Gleizer*, T. Istomin, M. Mazo Jr, A. L. Murphy, and G. P. Picco, "The wireless control bus: Enabling efficient multi-hop event-triggered control with concurrent transmissions," *ACM Transactions on Cyber-Physical Systems (TCPS)*, vol. 6, no. 1, pp. 1–29,2021.
8. *G. de A. Gleizer*, K. Madnani, and M. Mazo Jr., "Self-triggered control for near-maximal average inter-sample time," in *60th IEEE Conference on Decision and Control (accepted)*, 2021.
9. *G. de A. Gleizer* and M. Mazo Jr., "Computing the average inter-sample time of event-triggered control using quantitative automata," 2021, `https://arxiv.org/abs/2109.14391`. Under review at Nonlinear Analysis: Hybrid Systems.
10. G. Delimpaltadakis, *G. de A. Gleizer*, I. van Straalen, and M. Mazo Jr., "ETCetera: beyond event-triggered control," in *Proc. of the 25th Int'l Conf. on Hybrid Systems: Computation and Control*, ser. HSCC '22. ACM, 2022.
11. *G. de A. Gleizer*, K. Madnani, and M. Mazo Jr., "A simpler alternative: Minimizing transition systems modulo alternating simulation equivalence," in *Proc. of the 25th Int'l Conf. on Hybrid Systems: Computation and Control*, ser. HSCC '22. ACM, 2022.
12. *G. de A. Gleizer* and M. Mazo Jr, "Chaos and order in event-triggered control," *arXiv preprint arXiv:2201.04462*, 2022.

Included in this thesis.

Previous work

1. *G. de A. Gleizer*, C. Gonzaga, L. Vargas, A. Menon, D. Dai, and H. K. Matthews Jr. "System for controlling or monitoring a vehicle system along a route", November 27 2018. US Patent 10,137,912.

2. J. D. Brooks, H. K. Mathews Jr., D. Dai, R. S. Chandra, *G. de A. Gleizer*, B. P. Leao, and C. Gonzaga. "System and method for determining operational restrictions for vehicle control", June 13 2017. US Patent 9,676,403.

3. J. D. Brooks, H. K. Mathews Jr., *G. de A. Gleizer*, and L. Vargas. "System and method for asynchronously controlling brakes of vehicles in a vehicle system", June 6 2017. US Patent 9,669,811.

4. L. Vargas, H. K. Mathews Jr., B. N. Meyer, *G. de A. Gleizer*, and C. Gonzaga. "Power control system for a vehicle system", March 17 2017. US Patent App. 15/461,548.

5. J. D. Brooks, H. K. Mathews Jr., and *G. de A. Gleizer*. "System and method for monitoring coupler fatigue", November 30 2015. US Patent 9,937,936.

6. B. P. Leao, J. K. Klooster, *G. de A. Gleizer*. "Method and apparatus for generating or updating an aviation simulation scenario", July 9 2013. US Patent App. 13/938,034.

7. *G. de A. Gleizer*. "Optimal task scheduling for fieldbuses applied to foundation™ fieldbus" (in portuguese). Master's thesis, Universidade Federal do Rio de Janeiro, 2013.

8. A. L. Silva, L. P. Orenstein, F. Lizarralde, A. C. Leite, and *G. de A. Gleizer*. "Hardware and software update for an industrial robot (in portuguese)." In *Brazilian Conference of Automatics*, volume 18, pages 4403–4410, 2010.

9. G. Freitas, *G. de A. Gleizer*, F. Lizarralde, and L. Hsu. "Multi-objective optimization for kinematic reconfiguration of mobile robots." In *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pages 686–691. IEEE, 2010.

10. G. Freitas, *G. de A. Gleizer*, F. Lizarralde, and L. Hsu. "Reconfiguration of mobile robots on irregular terrains based on multiple objective optimization." In *Brazilian Conference of Automatics*, 2010.

11. G. Freitas, *G. de A. Gleizer*, F. Lizarralde, L. Hsu, and N. R. S. dos Reis. "Kinematic reconfigurability control for an environmental mobile robot operating in the amazon rain forest." *Journal of Field Robotics*, 27(2):197–216, 2010.