Title
Detection and Optimization Algorithms for Cyber-Physical Systems

Author
Bastos Hespanhol, Pedro Ivo

Publication Date
2020

Peer reviewed|Thesis/dissertation

Detection and Optimization Algorithms for Cyber-Physical Systems

by

Pedro Ivo Bastos Hespanhol

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering- Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Anil Jayanti Aswani, Chair
Professor Zuo-Jun (Max) Shen
Professor Prasad Raghavendra

Spring 2020

Detection and Optimization Algorithms for Cyber-Physical Systems

Abstract

Detection and Optimization Algorithms for Cyber-Physical Systems

by

Pedro Ivo Bastos Hespanhol

Doctor of Philosophy in Engineering- Industrial Engineering and Operations Research

University of California, Berkeley

Professor Anil Jayanti Aswani, Chair

Cyber-Physical Systems (CPS) play an ubiquitous role in operation and control in many different domains: power systems, finance, robotics, and automation. The complex interplay between cyber components such as software, communication protocols, computer servers and physical components, such as sensors and pieces of dedicated hardware, requires advanced and sophisticated methods and algorithms that ensure safe and efficient operation. In this thesis we tackle both safety and efficiency: We develop novel detection algorithms that are able to identify malicious attacks, sensor corruption and faulty measurements. Our detection mechanisms have provable guarantees based on rigorous asymptotic and non-asymptotic statistical analysis and can be readily implemented in CPS, such as robotic systems and autonomous vehicles. In addition, we developed collusion detection mechanisms that can be used to identify whether two or more CPS are colluding or not. We also design a mechanism that is able to induce selfish systems/agents to behave cooperatively. We showcase the performance of our algorithms with several different case studies. In our analyses, we place emphasis on algorithms that can be implemented in real-time, that is can be used while the system is under operation in the real-world. On the efficiency side, we developed real-time non-linear Model Predictive Control (MPC) Methods that can provide optimal solutions to the Optimal Control problem faced by the CPS during operation. Our algorithm exploits the control structure and is tailored for implementation in embedded hardware and can operate both with memory and computation time constraints. We showcase the performance of our algorithm with a C/C++ implementation and we compare to several current state-of-the-art Optimal Control solvers. We also extend our methodology to be used together with Pseudo-spectral Methods and Hybrid Systems, developing an integrated Mixed-Integer MPC algorithm that can handle complex non-linear dynamics and both continuous and discrete variables. With this thesis, our goal is to provide real-time practical algorithms that have provable guarantees in performance both in the detection task and in the optimal control task. Our algorithms are based on rigorous theoretical analysis and display very good performance and can be readily implemented in practical Cyber-Physical Systems.

To my parents, Luiza and Jorge

Contents

List of Figures

List of Tables

Acknowledgments

In this thesis, I present the work I have develop at UC Berkeley in the last five years in the Industrial Engineering and Operations Research (IEOR) Department. It was a long journey, often times arduous, but now I am sure that it was worth it and, most importantly, it was satisfying and it made me feel confident, not only as a researcher, but also as a human being. The experiences and the knowledge I have gained in those years are irreplaceable and they make what I am today.

I would like to express my gratitude to my advisor Professor Anil Aswani. Professor Aswani was one of the first people I ever had contact with at Berkeley and he was the one who sparked on me the desire to pursue a Ph.D. degree in IEOR. Ever since that summer back in 2014 where he hired me as a research assistant during my time as an exchange student, he has been a great source of knowledge and motivation, never failing to support me and offering advice. Our discussions, throughout the entire Ph.D. program at Berkeley, were fundamental in my development as a researcher and as a person. I will always be grateful for your help and for the cooperation we had during those five years.

I would like to thank Professor Ram Vasudevan and Matthew Porter from University of Michigan, with whom I had many productive collaborations. The research materials in Chapter 2-4 were developed with their help: Our joint effort across teams were very productive and we were able to develop new and interesting ideas. I would like to thank you both for your comments, suggestions, and discussions across the several papers we wrote together.

I would like to thank Researchers Rien Quirynen and Stefano Di Cairano and the entire Mistubishi Electric Research Laboratories (MERL) team for their collaboration with the research materials in Chapters 5-6: the experience I had working with MERL was invaluable and I would like to acknowledge their support and partnership with my research projects. The research environment of MERL is of utmost excellency and I felt confortable expressing my ideas and proposing new solutions and algorithms with the research team.

I would like to thank Professor Max Shen and Professor Prasad Raghavendra for being in my dissertation committee and for being always open and helpful to discussions and questions. I would like to thank Professor Alper Atamtürk, for giving me very good advice, support and for allowing me to be the teaching instructor of IEOR 262A (Math Programming I), which was one of my personal goals ever since I sat down on Etcheverry Hall to learning Linear Programming and Farkas's Lemma.

I also would like to thank professor Max Shen, and by extension the entire IEOR department staff and faculty for allowing me to teach a course (IEOR 265, Learning and Optimization) on my last semester, as a Ph.D. student. Lecturing a course was challenging, but a very rewarding experience. I would like to thank the department for putting their trust in me to teach this graduate course in behalf of the IEOR faculty.

Lastly, but definitely not least, I would like to thank my Ph.D. friends: the ones who started on the Ph.D. program with me and that have been a source of strength and encouragement since the very first day. The Farkas's Boys will always be together.

Chapter 1

Introduction

Cyber-Physical Systems (CPS) are ubiquitous in modern day operation of power systems, finance, robotics, and communication networks. The interplay between cyber components, that is software, computer servers, communication protocols, and physical components, hardware, actuators, and sensors, has become complex and it's analysis of utmost importance in designing reliable, efficient, and safe systems. With the technological improvement of the previous decade, the operation of physical systems are increasingly reliant on fast real-time sensor measurements, information-sharing and fast-response to breaches in security and in performance. In power systems, country-wide power grids are operated minute-by-minute by using information systems to best control the grid. In addition, advanced algorithms are implemented in specialized software in order to provide operational policies that are resilient to uncertainty and future potential disturbances. On finance, more and more quantitative firms rely on high-frequency trading algorithms, that trade large volumes of assets in real-time, using complex Machine Learning and Statistical Inference tools in order to provide them with leverage at the time of trade. These algorithms can only perform at a high level if implemented in specific hardware that is able to provide extremely fast computation power and communication speed. This is a very good example where both cyber and physical components need to be top-notch and need to be resilient and reliable. Lastly, the exciting area of autonomous vehicles and autonomous drones flight, represent the current frontier of challenges in integrating software, parallelizable algorithms, efficient communication protocols, with the latest state-of-the-art hardware architectures, such as massive GPU/FPGA frameworks, distributed LIDAR sensing and Machine Learning based prediction algorithms.

In all aforementioned areas the widespread use of those new technologies bring with them challenges in both the security and in the performance side. Security (or in other words, safety in operation) is central when automated systems are involved (this is evident in media discussion about self-driving cars and their role in pedestrian and driver safety). The discussion often revolves about the question of how to "prove" that a given CPS is safe and resilient, to say, other systems (for example like other vehicles on the street) or sensor failure due to unforeseen circumstances (like abrupt rain or snow). Another area of discussion lies in how to mitigate external threats to the system, for example due to malicious agents

tampering with sensor measurements or the communication channels, and how to ensure that the effects of these attacks are not only detected quickly but also mitigated effectively. Real-time operation of CPS requires real-time detection methods that can be carried in an automated way by the system and are amenable to remote supervision and control. As technology improves, the complexity and sophistication of external attacks and malicious disturbances increases and the game of "cat-and-mouse" between security methods and attacks that are designed to bypass such methods, reaches high-level of technical sophistication and thus requires advanced theoretical and practical tools in order to analyze the performance of defense mechanism, in order to provide provable guarantees on their efficacy in face of the potential threats the CPS might face. Lastly we state another thread of discussion which lies on the problem of collusion and competition between CPS. These systems often rely on algorithms that leverage both computational power and inference systems in order to make control decisions. When two or more of such systems interact the outcome can be unpredictable. A significant body of work in the literature focuses on studying such interactions and try to design detection methods that are able to indicate whether these automated systems are colluding or not. The main practical examples lie in automated pricing between airline companies, which are increasingly reliant on Machine Learning aided method in order to set their prices for travel flights. Recently, it has been found out that these algorithms are implicitly colluding in order to set prices higher that "fair" competition prices. This is done often without knowledge of the companies deploying such algorithms. This example illustrates the necessity of advanced detection and inference algorithms that are able to reliably detect and mitigate the collusion phenomenon when it occurs.

On the performance side, the challenge lies in guaranteeing performance in face of unforeseen disturbances and uncertain future scenarios, a key example being how autonomous vehicles should behave when predicting the movement of pedestrians and other vehicles. The introduction of uncertainty makes the control problem faced by a CPS extremely challenging. In addition, in practice there often are regulatory and technical constraints that introduce limitations in designing an efficient control policies. State-of-the art algorithms that rely on Constrained Optimization and Optimal Control need to be implemented in specific hardware, such as FPGA's or GPU's, in other to output control solutions in real-time in face of such uncertainties. In practice, such control decisions need to be updated every a couple hundred of milliseconds as new information arrives to the system, via the sensor hardware or the communication channels. Another key aspect lies in handling discrete set of events (such as the decision whether to lane change or not by an autonomous vehicle). These events need to be modelled using adequate techniques such as Integer Programming or Hybrid Systems formulations and require dedicated software that is able to find the best possible solution in real-time.

In this thesis we develop novel advances in the area of security and efficiency of CPS. In the area of security, we focus our attention in how to provide resilient and robust techniques that can identify external threats to the system and can provide effective mitigation measures against them. These threats can take the form of third-party malicious agents attempting to breach security protocols, DDOS-type of attacks on the communication channels, or even

inside-personnel tampering with sensor measurements. In a world that is increasingly reliant on CPS, such as drone surveillance, automated manufacturing and sensor-based power systems operations, providing provable effective algorithms with detection guarantees is essential in justifying the employment of this systems.

On this regard we build upon the large body of literature of attack detection mechanisms and safe system identification. Often, attack detection schemes heavily rely on estimation procedures, usually computing quantitative indices in order to claim that an attack or system corruption is under way. These procedures are called "passive", because the controller, that is the agent responsible for the safe operation of the CPS, can only collect data and does not actively interfere in the operation in order to detect or to mitigate the impact of the attack. In our work, we go in a different direction: even though we heavily rely on statistical properties and probabilistic guarantees, we provide an active defense mechanism where the controller has the power and the agency to interfere, in a non-disruptive manner, in the system in order to detect attacks and check for faulty behavior. This defense mechanism is called Dynamic Watermarking and it is one of the foundations of this thesis work.

The idea of Watermarking comes from the Compute Science literature in order to increase security in file-sharing channels: documents, or files, would be watermarked with an encrypted code, which only the intended receiver would be able to remove it without destroying the contents of the document. This idea provides an additional, and effective, layer of security in sharing sensitise information via channels that may be compromised. Translating this idea to CPS, Dynamic Watermarking is based on introducing a mark (which is essentially a small disturbance or perturbation) to the system, be it by injecting some noise in the cyber components, or by injecting some perturbation on the actuators in the physical components. This mark needs to be carefully designed in order to not degrade the system performance by a critical amount. However, the key benefit is that it provides the controller agency in using such designed watermark in order to make inferences in the system behavior and it allows the operators, be them engineers in a control room, or an AI that operates the system, to answer questions such as: is the system being attacked? are the sensor measurements being corrupted? Is the environment affecting any information gathered by the system? In our work on the subsequent chapters we provide algorithms and techniques that use Dynamic Watermarking as their foundation to answer those and many more questions. We justify our methods by rigorous statistical analysis and asymptotic and non-asymptotic probability guarantees. The analysis presented in the subsequent chapters include Hypothesis Testing, Finite-time Concentration Bounds, System Identification, and more. We illustrate our methodology with a vast array of examples, which range from simple autonomous robots maneuvers, to self-driving vehicles applications; ranging from single-agent systems to distributed systems.

We also analyze the interaction between CPS: namely we provide detection algorithms that are able detect collusion between the operating systems in an efficient fashion. These algorithms are based on rigorous statistical properties and are given based some solution concept that characterizes the agent's behavior. Often, this solution concept boils down to a Nash Equilibrium of a game that is induced by the agents behavior. We present our detection

algorithms under this notion of equilibrium and we derive key results regarding detection and testing of collusion on this environment. This line o research tries to address key questions posed in the recent literature regarding collusion between AI-controlled system, an we see our contributions as a significant step in direction of developing provably efficient detection mechanism. We also study the opposite problem, where instead of detecting a potential collusion between agents, we establish mechanisms that induce cooperation between selfish agents. We differentiate our contributions from the body of work of Mechanism Design by focusing our analysis on low-communication mechanisms which are amenable to real-time implementation and can be leverage by the hardware typically used in CPS.

The second body of contributions of these thesis lies in developing new real-time optimal control algorithms that are capable to provide the operator of CPS with high-quality solutions for the control problem. We frame the decision-making problem faced by the CPS as an Optimal Control problem, which is an non-linear constrained optimization problem with a particular structure. Our algorithms actively exploit this structure in order to produce optimal solutions in a memory efficient fashion in very fast computation times. By "real-time algorithms" we explicitly mean algorithms that are meant to be use as the system is under operation in real time. Typically, the system is allotted a couple of hundred milliseconds to output a solution that needs to executed immediately after. This real-time nature of the operation falls naturally under the Model-Predictive Control (MPC) framework, which is an advanced control technique that relies on Approximate Dynamic Programming ideas to compute optimal control policies in a successive fashion, warm-starting every iteration with information computed on the previous iterations. All algorithms we study fall under this framework and can be readily applied to CPS which requires real-time computations. We illustrate our methodologies again with a series of experiments and test cases, from simple to complex examples that showcase the algorithms performance. We present rigorous (local) convergence analyses of the algorithms and prove their consistency and establish their rates of convergence. Model-Predictive Control is a well-established control technique and our contribution builds on top of the existing literature by expanding and generalizing existing methods as well establishing new ones. As with the detection schemes, our focus is also on real-time implementation which is of utmost importance if any MPC-type method is to be practically useful.

We also present an algorithm that is able to handle discrete set of events and Hybrid Systems (that is systems with many discrete modes of operation). We leverage our Operations Research and Integer Programming background to equip our developed real-time optimal control algorithm with a framework, similar to the classical Branch-and-Bound algorithm, that is able to handle discrete decision variables. We embed this framework in the MPC-style of computation, and we provide a series of examples of how the integrated framework can be used in relevant applications, ranging from self-driving vehicles to automated satellite orbital control.

All of our contributions share the core thread of real-time practical applicability: Our methods and algorithms are developed with the goal of practical use in an environment where decisions and information flow in real-time across different pieces of software and hardware.

We base our developments in rigorous proofs of correctness and justify their application by showcasing computational experiments. In the modern day, Cyber-Physical Systems require efficient and reliable algorithms in order to provide the best quality of service to humanity.

Summary

On Chapter 2 we begin our analysis of detection algorithms that rely on Dynamic Watermarking. On this chapter we lay the foundations of the methodology and analyze their application on both single-agent systems and distributed systems. Our analyses start with general linear time-invariant (LTI) systems. We illustrate the application of our detection algorithms on both types of systems with a set of test case studies which showcase their numerical performance and compare our approach with other state-of-the-art detection mechanisms. The contents of this chapter are based on our work in [132, 133].

On Chapter 3 we generalize our analyses by focusing on non-asymptotic variations of the Dynamic Watermarking detection algorithms. The non-asymptotic detection schemes are of crucial important when we cannot safely rely on the asymptotic behavior of the underling probability distributions to "kick in". We focus on the key example of Cyber-Physical Systems with sensor switching, where the automated system is equipped with more than one sensor and can switch between them. Sensor switching is used together with Dynamic Watermarking to provide a provably safe and reliable detection mechanisms that can not only identify when an attack or sensor corruption has taken place, but can also design a policy to mitigate their effect by smartly using the sensor switching. We illustrate our methodology with a numerical case study that showcases the role of both sensor switching and Dynamic Watermarking in the detection and mitigation tasks of the CPS. We conclude the chapter with a related problem of establishing estimation consistency of switched linear systems, and we provide a set-membership estimator that is consistent when used to identify different modes of the system. The developments in this chapter follows our work in [131, 125]

On Chapter 4 we turn our focus to the interaction between CPS's. We provide a detection mechanism that is able to detect collusion among systems. Our mechanism is based on the notions of Nash Equilibrium and Variational Inequalities. We present the foundations for both and present the statistical properties of our algorithm. We also study the opposite problem: we also design an algorithm that induce selfish agents to act cooperatively. This time, our analyses is based on Mechanism Design literature and we focus on providing a low-dimension communication protocol that can be implemented efficiently in real-time applications of the subsequent chapters. This chapter is based on our work presented in [124, 126]

On Chapter 5 we study real-time Model Predictive Control Algorithms for CPS. In particular we provide a an efficient structure-exploiting algorithm that is capable of handling constrained convex Optimal Control problems. Our adjoint-based block-structure algorithm leads to a Solver that can be implemented in embedded hardware, such as FPGA, or specific GPU's architecture, to achieve real-time computation speeds. We provide full proofs and convergence analysis, based on convex set analysis and non-linear constrained optimization.

We illustrate the performance of this algorithm based on a C/C++ implementation on embedded hardware and report the numerical performance against state-of-the-art Optimal Control Solvers. This chapter materials are based on our work in [127, 121]

On Chapter 6 we generalize our real-time MPC Algorithm to different applications: On the first we apply algorithm in conjunction with Pseudospectral methods, which are methods that are able to efficiently handle complex set of non-linear dynamics to a high degree of accuracy. Pseudospectral methods can be naturally combined with non-linear MPC and our novel algorithm handles it in a efficient manner producing a highly competitive Optimal Control Algorithm for complex Cyber-Physical Systems. Lastly, we incorporate Hybrid Systems on our analyses, which introduce discrete set of controls and state variables. The integer nature of those variables makes to Optimal Control problem considerably more difficult. We use Branch-and-Bound techniques in conjunction with our structure-exploiting MPC solver to establish a real-time Mixed-Integer MPC solver, that can handle both continuous and discrete variables by exploiting the problem structure together with the Branch-and-Bound tree in order to achieve real-time computations We illustrate both lines of extension with numerical case studies that highlight the applicability of our proposed algorithm for both types of environments. The materials presented in this chapter are based on our work in [129, 128]

Lastly on Chapter 7, we present a discussion and outlook of future direction of research and how do we envision the field of safety and performance of CPS for the future and how we can contribute more in the design and analysis of probably safe and efficient Cyber-Physical Systems.

Chapter 2

Dynamic Watermarking in Cyber-Physical Systems

The secure and resilient control of cyber-physical systems (CPS) requires safe operation in the face of malicious attacks that can occur on either the physical layer (e.g., sensors and actuators) or the cyber layer (e.g., communication and computation capabilities)[226]. Real-life incidents like the Maroochy-Shire incident [3], the Stuxnet worm [149], and others [63] illustrate the importance of concerns about CPS security.

One approach to secure control has been to focus on cybersecurity of CPS [192, 145, 251, 141], but this does not fully exploit the physical aspects of CPS. An alternative is attack identification and detection considering the interplay between the cyber and physical parts of CPS [12, 63, 64, 193]. More specifically, these methods focus on monitoring measurements sent to controllers at the physical layer. While mitigating the effects of denial of service (DOS) attacks poses several challenges [12], detecting such attacks is not an issue for relatively reliable networks. However, false data injection (FDI) attacks, which aim to alter measurements, can be made especially stealthy by replaying recorded measurements [172, 130, 92, 215] or exploiting vulnerable subspaces of the physical layer's dynamics [238]. Many approaches for detecting FDI attacks are static (i.e., do not consider system dynamics) [106] or passive (i.e., do not actively control system to identify malicious nodes and sensors) [26, 90, 91].

In contrast, *dynamic watermarking* is an active defense technique that injects perturbations into the system control in order to detect attacks [252, 172, 171, 98]. More specifically, this method applies a *private excitation* to the system, which is a disturbance only known to the controller. Then it uses consistency tests to detect attacks by checking for correlation between sensor measurements and the private excitation. The goal is to be able to detect all sensor attacks whose magnitude exceeds some prespecified amount. More recently, the work done in [223] and [130] attempts to bridge this gap by providing statistical guarantees for complex types of attacks for general LTI systems.

While both papers address a general MIMO LTI system, the set of assumptions are somewhat different: the former assumes open-loop stability of the LTI system, and the

latter restricts the attack form. In particular, in [130], the tests provided are able to detect if a general MIMO LTI system is under a fairly general type of attack. In particular, it considers additive attacks that can dampen/amplify the system measurements, can replay the system from a different initial condition, or can do both. This form of attack, while arguably simple, encompasses many of the types of attacks reported in real-life incidents (e.g., replay attacks [149]) as well as compensate for external disturbances not accounted by the system model (e.g., wind when represented via internal model principle [130]).

Research on dynamic watermarking can be divided into two main areas of contribution: The first is the development of statistical hypothesis testing that tries to detect corrupted measurements by observing correlations between sensor outputs and the dynamic watermark [172, 174, 171, 252, 173]. This set of techniques apply to general LTI systems, but cannot ensure the zero-average-power property for general attack models. The second line of work [222, 143] considers general attack models and develop tests able to ensure that only attacks which add a zero-average-power signal to the sensor measurements can remain undetected, but constrain their analysis to LTI systems with specific structure on their dynamics. Our first contribution in this chapter is to partially bridge the gap between these two techniques by developing a method that applies to general LTI systems under specific attack models and that ensures the zero-average-power property for attacks.

Detecting attacks in control systems is an important aspect of designing secure and resilient control systems. Recently, a dynamic watermarking approach was proposed for detecting malicious sensor attacks for SISO LTI systems with partial state observations and MIMO LTI systems with a full rank input matrix and full state observations; however, these previous approaches cannot be applied to general LTI systems that are MIMO and have partial state observations. This paper designs a dynamic watermarking approach for detecting malicious sensor attacks for general LTI systems, and we provide a new set of asymptotic and statistical tests. We prove these tests can detect attacks that follow a specified attack model (more general than replay attacks), and we also show that these tests simplify to existing tests when the system is SISO or has full rank input matrix and full state observations. The benefit of our approach is demonstrated with a simulation analysis of detecting sensor attacks in autonomous vehicles. Our approach can distinguish between sensor attacks and wind disturbance (through an internal model principle framework), whereas improperly designed tests cannot distinguish between sensor attacks and wind disturbance.

On section 2.1 we start our analysis with the general LTI system model (i.e., MIMO systems with partial observations) and specifies our attack model. We provide intuition on why existing dynamic watermarking approaches cannot by used on a general LTI system. We construct a detection consistent dynamic watermarking approach for general LTI systems under our attack model, and our term *detection consistent* test is used to refer to a test that ensures the zero-average-power property (described above) for attacks. Next, we describe how our asymptotic tests can be converted into statistical tests, and we show how our tests are special cases of those in [222] for the SISO case or the MIMO case with full rank input matrix and full state observations. On section 2.2 we conduct simulations of an autonomous vehicle: Our tests are able to distinguish between sensor attacks and wind disturbances when

including wind disturbance in the system dynamics using the internal model principal, while improperly designed tests cannot distinguish between attacks and wind.

On section 2.3 we extend the Dynamic Watermarking methodology to distributed systems. Often in practice, the entire cyber-physical system is neither controllable nor observable by a single subcontroller, communication of sensor measurements is required to ensure closed-loop stability. The possibility of attacking the communication channel has not been explicitly considered by previous watermarking schemes, and requires a new design. In 2.3, we derive a statistical watermarking test that can detect both sensor and communication attacks. A unique (compared to the non-networked case) aspect of the implementing this test is the state-feedback controller must be designed so that the closed-loop system is controllable by each sub-controller, and we provide two approaches to design such a controller using Heymann's lemma and a multi-input generalization of Heymann's lemma. The usefulness of our approach is demonstrated in 2.4 with a simulation of detecting attacks in a platoon of autonomous vehicles. Our test allows each vehicle to independently detect attacks on both the communication channel between vehicles and on the sensor measurements. We show that our approach is able to detect the presence or absence of sensor attacks and attacks on the communication channel between vehicles.

2.1 Dynamic Watermarking in MIMO LTI Systems

On this section, we design a dynamic watermarking approach for detecting malicious sensor attacks for general LTI systems, and has two main contributions: First, we generalize the watermarking approach developed in [222] for SISO LTI systems with partial state observations and MIMO LTI systems with a full rank input matrix and full state observations under an arbitrary attack, and our generalization applies to general LTI systems under a specific attack model that is more general than replay attacks [252].

The design of intelligent transportation systems (ITS) is receiving increased attention [143, 107, 15, 255, 243, 175, 69], and one significant area for further study is the design of methods to ensure the safe and resilient operation of ITS. One recent work [143] considered the use of dynamic watermarking to detect sensor attacks in a network of autonomous vehicles coordinated by a supervisory controller; the watermarking approach was successfully able to detect attacks. However, large-scale deployments of ITS must be resilient in the face of persistent disturbances from environmental and human factors. Wind is an example of such a persistent disturbance. A second contribution of this work is from the perspective of modeling: We show that persistent disturbances such as those from wind can invalidate watermarking approaches, and we propose an internal model principle-based approach to handle persistent disturbances. This motivates our generalization of dynamic watermarking to general MIMO LTI systems with partial observations, since internal model states are not directly observed.

LTI System and Attack Model

Let $[r] = \{1, \dots, r\}$, and consider a MIMO LTI system $x_{n+1} = Ax_n + Bu_n + w_n$ with partial observations $y_n = Cx_n + z_n + v_n$, where $x \in \mathbb{R}^p$, $u \in \mathbb{R}^q$, and $y, z, v \in \mathbb{R}^m$. The v_n should be interpreted as an additive measurement disturbance added by an attacker, while w_n represents zero mean i.i.d. process noise with a jointly Gaussian distribution and covariance Σ_W, and z_n represents zero mean i.i.d. measurement noise with a jointly Gaussian distribution and covariance Σ_Z. We further assume the process noise is independent of the measurement noise, that is w_n for $n \geq 0$ is independent of z_n for $n \geq 0$.

If (A, B) is stabilizable and (A, C) is detectable, then a stabilizing output-feedback controller can be designed when $v_n \equiv 0$ using an observer and the separation principle. Let K be a constant state-feedback gain matrix such that $A + BK$ is Schur stable, and let L be a constant observer gain matrix such that $A + LC$ is Schur stable. The idea of dynamic watermarking in this context will be to superimpose a private (and random) excitation signal e_n known in value to the controller but unknown in value to the attacker. As a result, we will apply the control input $u_n = K\hat{x}_n + e_n$, where $\hat{x}_n$ is the observer-estimated state and e_n are i.i.d. Gaussian with zero mean and constant variance Σ_E fixed by the controller.

Let $\tilde{x}^\mathsf{T} = \begin{bmatrix} x^\mathsf{T} & \hat{x}^\mathsf{T} \end{bmatrix}$, and define $\underline{B}^\mathsf{T} = \begin{bmatrix} B^\mathsf{T} & B^\mathsf{T} \end{bmatrix}$, $\underline{C} = \begin{bmatrix} C & 0 \end{bmatrix}$, $\underline{D}^\mathsf{T} = \begin{bmatrix} \mathbb{I} & 0 \end{bmatrix}$, $\underline{L}^\mathsf{T} = \begin{bmatrix} 0 & -L^\mathsf{T} \end{bmatrix}$, and

$$\underline{A} = \begin{bmatrix} A & BK \\ -LC & A + BK + LC \end{bmatrix} \tag{2.1}$$

Then the closed-loop system with private excitation is given by $\tilde{x}_{n+1} = \underline{A}\tilde{x}_n + \underline{B}e_n + \underline{D}w_n + \underline{L}(z_n + v_n)$. If we define the observation error $\delta = \hat{x} - x$, then with the change of variables $\check{x}^\mathsf{T} = \begin{bmatrix} x^\mathsf{T} & \delta^\mathsf{T} \end{bmatrix}$ we have the dynamics $\check{x}_{n+1} = \underline{\underline{A}}\check{x}_n + \underline{\underline{B}}e_n + \underline{\underline{D}}w_n + \underline{\underline{L}}(z_n + v_n)$, where $\underline{\underline{B}}^\mathsf{T} = \begin{bmatrix} B^\mathsf{T} & 0 \end{bmatrix}$, $\underline{\underline{D}}^\mathsf{T} = \begin{bmatrix} \mathbb{I} & -\mathbb{I} \end{bmatrix}$, $\underline{\underline{L}} = \underline{L}$, and

$$\underline{\underline{A}} = \begin{bmatrix} A + BK & BK \\ 0 & A + LC \end{bmatrix}. \tag{2.2}$$

Recall that $\underline{\underline{A}}$ is Schur stable whenever $A + BK$ and $A + LC$ are both Schur stable.

Since the controller is fixed, we can suppose the attacker chooses $v_n = \alpha(Cx_n + z_n) + C\xi_n + \zeta_n$ for some fixed $\alpha \in \mathbb{R}$, where $\xi_{n+1} = (A + BK)\xi_n + \omega_n$, ζ_n are i.i.d. Gaussian with zero mean and constant variance Σ_S fixed by the attacker, and ω_n are i.i.d. Gaussian with zero mean and constant variance Σ_O fixed by the attacker. The idea underlying this attack model is that the attacker allows some fraction of the true output $Cx_n + z_n$ to be measured by the controller, and at the same time also incorporates the measurement of a false state ξ_n that evolves according the dynamics that would be expected under the controller.

Intuition for Designing a New Test

To better understand how to design a new test, it is instructive to apply existing dynamic watermarking schemes and the associated tests [222] to particular LTI systems. Such an

exercise provides intuition that we use to design new tests. Our main example is an LTI system with

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{2.3}$$

Suppose the attacker chooses $v_n = -(Cx_n + z_n) + C\xi_n + \zeta_n$ with $\Sigma_S = \Sigma_Z$ and $\Sigma_O = \Sigma_W$, meaning the output measurement $y_n = C\xi_n + \zeta_n$ has no component from the actual system. This is a SISO (i.e., $m = q = 1$) system with partial state measurement, and the tests in [222] pass for this example, even though the sensor has been compromised by an attacker. The problem in this example is that the test

$$\text{as-lim}_N \tfrac{1}{N} \textstyle\sum_{n=0}^{N-1} L(C\hat{x}_n - y_n)e_{n-1}^{\mathsf{T}} = 0 \tag{2.4}$$

from [222] correlates the innovations process $L(C\hat{x}_n - y_n)$ with the private excitation only one step back in time e_{n-1}; however, it takes two time steps for the control input to enter into the output in this example. And so when designing a new test for general LTI systems, we need to take into consideration that there is generally some delay between when some private excitation is applied to when it is observed.

Detection Consistent Test

Now let Σ_X be the positive semidefinite matrix that solves the following

$$\Sigma_X = \underline{\underline{A}}\Sigma_X\underline{\underline{A}}^{\mathsf{T}} + \underline{\underline{B}}\Sigma_E\underline{\underline{B}}^{\mathsf{T}} + \underline{\underline{D}}\Sigma_W\underline{\underline{D}}^{\mathsf{T}} + \underline{\underline{L}}\Sigma_Z\underline{\underline{L}}^{\mathsf{T}}. \tag{2.5}$$

Note that $\Sigma_X = \text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1} \check{x}_n\check{x}_n^{\mathsf{T}}$. Similarly let Σ_Δ be the positive semidefinite matrix that solves the following

$$\Sigma_\Delta = (A + LC)\Sigma_\Delta(A + LC)^{\mathsf{T}} + \Sigma_W + L\Sigma_Z L^{\mathsf{T}}. \tag{2.6}$$

Note $\Sigma_\Delta = \text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1} \delta_n\delta_n^{\mathsf{T}}$ and $\Sigma_\Delta = \underline{\underline{M}}\Sigma_X\underline{\underline{M}}^{\mathsf{T}}$, where $\underline{\underline{M}} = \begin{bmatrix} 0 & \mathbb{I} \end{bmatrix}$. Recall that Σ_X and Σ_Δ exist because the above are Lyapunov equations with matrices $\underline{\underline{A}}, (A+LC)$ that are Schur stable.

Lemma 2.1.1. *We have that*

$$\underline{\underline{A}}^r\underline{\underline{B}} = \begin{bmatrix} (A + BK)^r B \\ (A + BK)^r B \end{bmatrix} \tag{2.7}$$

for all $r \geq 0$

Proof. The result holds for $r = 0$ since $\underline{\underline{A}}^0 = \mathbb{I}$ and $(A + BK)^0 = \mathbb{I}$. Now suppose the result holds for r: We prove that it holds for $r + 1$. In particular, note that

$$\underline{\underline{A}}^{r+1}\underline{\underline{B}} = \underline{\underline{A}} \begin{bmatrix} (A + BK)^r B \\ (A + BK)^r B \end{bmatrix} = \begin{bmatrix} (A + BK)^{r+1} B \\ (A + BK)^{r+1} B \end{bmatrix}, \tag{2.8}$$

where the first equality holds by the inductive hypothesis, and the second equality follows by calculation of the matrix multiplication. Hence the result follows by induction. □

Proposition 2.1.2. *Let* $\underline{A}(\alpha) = \underline{A} + \alpha\underline{H}$ *with*

$$\underline{H} = \begin{bmatrix} 0 & 0 \\ -LC & 0 \end{bmatrix}, \tag{2.9}$$

and define $k' = \min\{k \geq 0 \mid C(A+BK)^k B \neq 0\}$. *Then we have that* $\underline{A}(\alpha)^k \underline{B} = \underline{A}^k \underline{B}$ *for* $0 \leq k \leq k'$.

Proof. If $k' = 0$, then the result holds trivially. So assume $k' \geq 1$. We have that $\underline{A}(\alpha)^0 \underline{B} = \underline{A}^0 \underline{B} = \underline{B}$ since $\underline{A}(\alpha)^0 = \underline{A}^0 = \mathbb{I}$. Now suppose $\underline{A}(\alpha)^k \underline{B} = \underline{A}^k \underline{B}$ for $0 \leq k \leq k'-1$. But using Lemma 2.1.1 implies that

$$\underline{A}(\alpha)^{k+1}\underline{B} = \underline{A}^{k+1}\underline{B} + \alpha\underline{H}\begin{bmatrix}(A+BK)^k B \\ (A+BK)^k B\end{bmatrix} = \underline{A}^{k+1}\underline{B} + \alpha\begin{bmatrix}0 \\ -LC(A+BK)^k B\end{bmatrix} = \underline{A}^{k+1}\underline{B}, \tag{2.10}$$

where we have used that $LC(A+BK)^k B = 0$ since $k < k'$. And so the result follows by induction. □

Now let $k' = \min\{k \geq 0 \mid C(A+BK)^k B \neq 0\}$, and consider the following tests

$$\text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^{\mathsf{T}} = C\Sigma_\Delta C^{\mathsf{T}} + \Sigma_Z \tag{2.11}$$

$$\text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)e_{n-k'-1}^{\mathsf{T}} = 0 \tag{2.12}$$

Theorem 2.1.3. *Suppose* (A, B) *is stabilizable,* (A, C) *is detectable,* Σ_E *is full rank, and* $k' = \min\{k \geq 0 \mid C(A+BK)^k B \neq 0\}$ *exists. If the test (2.11)–(2.12) holds, then*

$$\text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1} v_n^{\mathsf{T}} v_n = 0, \tag{2.13}$$

meaning that v_n *asymptotically has zero power.*

Proof. Observe that the dynamics for $\tilde{x}$ are given by

$$\tilde{x}_{n+1} = \underline{A}(\alpha)\cdot\tilde{x}_n + \underline{B}e_n + \underline{D}w_n + \underline{L}((1+\alpha)z_n + C\xi_n + \zeta_n), \tag{2.14}$$

where $\underline{A}(\alpha) = \underline{A} + \alpha\underline{H}$ with $\underline{H}$ given in (2.9). Next note that a basic calculation gives

$$\tilde{x}_n = \underline{A}(\alpha)^k\tilde{x}_{n-k} + \sum_{k'=0}^{k-1}\underline{A}(\alpha)^{k-k'-1}\big(\underline{B}e_{n+k'-k} + \underline{D}w_{n+k'-k} + (1+\alpha)\cdot\underline{L}z_{n+k'-k} + \underline{L}C\xi_{n+k'-k} + \underline{L}\zeta_{n+k'-k}\big) \tag{2.15}$$

If we define $\underline{\underline{C}} = \begin{bmatrix} -C & C \end{bmatrix}$, then $C\hat{x}_n - y_n = \underline{\underline{C}}\tilde{x}_n - \alpha \cdot \underline{C}\tilde{x}_n - (1+\alpha) \cdot z_n - C\xi_n - \zeta_n$, and so for $k \in [p]$ we have

$$\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\big((C\hat{x}_n - y_n)e_{n-k}^{\mathsf{T}}\big) = (\underline{\underline{C}} - \alpha \cdot \underline{C}) \cdot \underline{A}(\alpha)^{k-1}\underline{B}\Sigma_E. \tag{2.16}$$

Note that $k' \leq p-1$ by the Cayley-Hamilton theorem. So combining Proposition 2.1.2 with (2.16) implies

$$\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\big((C\hat{x}_n - y_n)e_{n-k'-1}^{\mathsf{T}}\big) = (\underline{\underline{C}} - \alpha \cdot \underline{C}) \cdot \underline{A}^{k'}\underline{B}\Sigma_E = -\alpha \cdot \underline{C} \cdot \underline{A}^{k'}\underline{B}\Sigma_E \tag{2.17}$$

where the second equality holds by by Lemma 2.1.1 and the definition of $\underline{\underline{C}}$. Because the test (2.12) holds, the quantity (2.17) should equal 0. But since Σ_E is full rank by assumption, Sylvester's rank inequality implies $\underline{C} \cdot \underline{A}^{k'}\underline{B}\Sigma_E \neq 0$ since

$$\underline{C} \cdot \underline{A}^{k'}\underline{B} = \begin{bmatrix} 0 \\ C(A+BK)^{k'}B \end{bmatrix} \neq 0 \tag{2.18}$$

where the first equality holds by Lemma 2.1.1 and the definition of $\underline{C}$. Thus we must have $\alpha = 0$.

Next consider the expression

$$\frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^{\mathsf{T}} = \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - (1+\alpha)\cdot(Cx_n + z_n) - C\xi_n - \zeta_n)\times$$
$$(C\hat{x}_n - (1+\alpha)\cdot(Cx_n + z_n) - C\xi_n - \zeta_n)^{\mathsf{T}} \tag{2.19}$$

We showed above that $\alpha = 0$, and so the expectation of the above expression is

$$C\Sigma_\Delta C^{\mathsf{T}} + \Sigma_Z + \Sigma_S + \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}\big(C\xi_n\xi_n^{\mathsf{T}}C^{\mathsf{T}}\big) + \frac{1}{N}\sum_{n=0}^{N-1} C(A+BK)^{N-1}x_0(C(A+BK)^{N-1}\xi_0)^{\mathsf{T}}. \tag{2.20}$$

Since $(A+BK)$ is Schur stable, the associated property of exponential stability implies

$$\lim_N \frac{1}{N}\sum_{n=0}^{N-1} C(A+BK)^{N-1}x_0(C(A+BK)^{N-1}\xi_0)^{\mathsf{T}} = 0 \tag{2.21}$$

by combining the Cauchy-Schwartz inequality with the exponential stability. However from the test (2.11), the expectation must equal $C\Sigma_\Delta C^{\mathsf{T}} + \Sigma_Z$ in the limit. Since all the terms in the above expectation (2.20) are positive semidefinite or have zero limit, this implies

$$\Sigma_S + \text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}\big(C\xi_n\xi_n^{\mathsf{T}}C^{\mathsf{T}}\big) = 0. \tag{2.22}$$

Finally, consider the expression

$$\frac{1}{N}\sum_{n=0}^{N-1} v_n v_n^{\mathsf{T}} = \frac{1}{N}\sum_{n=0}^{N-1} \big((\alpha(Cx_n + z_n) + C\xi_n + \zeta_n\big) \times \big(\alpha(Cx_n + z_n) + C\xi_n + \zeta_n)^{\mathsf{T}}\big). \quad (2.23)$$

Since $\alpha = 0$, the expectation of the above expression is

$$\Sigma_S + \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\big(C\xi_n\xi_n^{\mathsf{T}}C^{\mathsf{T}}\big) + \frac{1}{N}\sum_{n=0}^{N-1} C(A+BK)^{N-1}x_0(C(A+BK)^{N-1}\xi_0)^{\mathsf{T}}. \quad (2.24)$$

Combining (2.21)–(2.24) implies as-lim$_N \frac{1}{N}\sum_{n=0}^{N-1} v_n v_n^{\mathsf{T}} = 0$. However, $v_n^{\mathsf{T}} v_n$ equals the sum of the diagonal entries of $v_n v_n^{\mathsf{T}}$. Thus we have as-lim$_N \frac{1}{N}\sum_{n=0}^{N-1} v_n^{\mathsf{T}} v_n = 0$. □

Existence of $k' = \min\{k \geq 0 \mid C(A+BK)^k B \neq 0\}$ is easy to verify because Cayley-Hamilton implies $k' \leq p-1$ or it does not exist, but we also give sufficient conditions.

Corollary 2.1.4. *Suppose (A, B) is controllable, (A, C) is observable, and Σ_E is full rank. If the test (2.11)–(2.12) holds, then*

$$\text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1} v_n^{\mathsf{T}} v_n = 0, \quad (2.25)$$

meaning that v_n asymptotically has zero power.

Proof. We claim that, under the conditions stated, $k' = \min\{k \geq 0 \mid C(A+BK)^k B \neq 0\} \leq p-1$ exists. Indeed, since (A, B) is controllable we have that: $(A+BK, B)$ is controllable, and the controllability matrix

$$\mathfrak{C} = \begin{bmatrix} B & (A+BK)B & \dots & (A+BK)^{p-1}B \end{bmatrix} \quad (2.26)$$

has rank$(\mathfrak{C}) = p$. And so by Sylvester's rank inequality, we have rank$(C\mathfrak{C}) \geq$ rank(C) + rank$(\mathfrak{C}) - p =$ rank(C). But (A, C) is observable, and so the observability matrix

$$\mathfrak{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix} = \text{diag}(C, \dots, C) \begin{bmatrix} \mathbb{I} \\ A \\ \vdots \\ A^{p-1} \end{bmatrix} \quad (2.27)$$

has rank$(\mathfrak{O}) = p$. Again applying Sylvester's rank inequality implies $p\,$rank$(C) \geq$ rank$(\mathfrak{O}) = p$, or equivalently that rank$(C) \geq 1$. Combining this with the earlier inequality gives rank$(C\mathfrak{C}) \geq 1$, and so $C\mathfrak{C} \neq 0$. This means $k' \leq p-1$ exists since $C\mathfrak{C}$ is a block matrix consisting of the blocks $C(A+BK)^k B$. Thus the result follows by Theorem 2.1.3. □

Statistical Version of Test

For the purpose of implementation, we can also construct a statistical version of our test (2.11)–(2.12). Our approach is similar to [222] in that we construct a hypothesis test by thresholding the negative log-likelihood. Before defining the test, we make the following useful observation:

Proposition 2.1.5. *Let* $\psi_n^{\mathsf{T}} = \begin{bmatrix}(C\hat{x}_n - y_n)^{\mathsf{T}} & e_{n-k'-1}^{\mathsf{T}}\end{bmatrix}$. *The test (2.11)–(2.12) holds if and only if the following test holds:*

$$\text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1}\psi_n\psi_n^{\mathsf{T}} = \begin{bmatrix} C\Sigma_\Delta C^{\mathsf{T}} + \Sigma_Z & 0 \\ 0 & \Sigma_E \end{bmatrix}. \tag{2.28}$$

Moreover, if the test (2.11)–(2.12) holds or equivalently the test (2.28) holds, then we have that $\text{as-lim}_n \mathbb{E}(\psi_n) = 0$.

Proof. The equivalence between (2.11)–(2.12) and (2.28) follows from the definition of ψ_n and of the tests. Next suppose either (equivalent) test holds: Using the dynamics on $\check{x}$ we have $\mathbb{E}(\check{x}_{n+1}) = \underline{\underline{A}}\mathbb{E}(\check{x}_n) + \underline{\underline{L}}\mathbb{E}(v_n)$. But we have $v_n = C(A+BK)^n\xi_0 + C\sum_{k=0}^{n-1}(A+BK)^{n-k-1}\omega_k + \zeta_n$ since $\alpha = 0$ as shown in the proof of Theorem 2.1.3, and so $\mathbb{E}(v_n) = C(A+BK)^n\xi_0$. Since $(A+BK)$ is Schur stable, we have $\lim_n \mathbb{E}(v_n) = 0$ and hence $\lim_n \mathbb{E}(\check{x}_n) = \underline{\underline{A}}\lim_n \mathbb{E}(\check{x}_n)$. This means that $\lim_n \mathbb{E}(\check{x}_n) = 0$ since $\mathbb{I} - \underline{\underline{A}}$ is full rank (which can be seen by recalling that $\underline{\underline{A}}$ is Schur stable, so cannot have any eigenvalue of exactly one, and thus $\det(s\mathbb{I} - A) \neq 0$ for $s = 1$). Since $C\hat{x}_n - y_n = \begin{bmatrix}0 & C\end{bmatrix}\check{x}_n$, we have that $\mathbb{E}(C\hat{x}_n - y_n) = 0$. This implies $\mathbb{E}(\psi_n) = 0$ since $\mathbb{E}(e_{n-k'-1}) = 0$ by construction. □

This result implies that asymptotically the summation $S_n = \frac{1}{\ell}\sum_{n+1}^{n+\ell}\psi_n\psi_n^{\mathsf{T}}$ with $\ell \geq m+q$ has a Wishart distribution with ℓ degrees of freedom and a scale matrix that matches (2.28), and we use this observation to define a statistical test. In particular, we check if the negative log-likelihood

$$\mathcal{L}(S_n) = (m+q+1-\ell)\cdot\log\det S_n + \text{trace}\left(\begin{bmatrix}(C\Sigma_\Delta C^{\mathsf{T}} + \Sigma_Z)^{-1} & 0 \\ 0 & \Sigma_E^{-1}\end{bmatrix} \times S_n\right) \tag{2.29}$$

corresponding to this Wishart distribution and the summation S_n is large by conducting the hypothesis test

$$\begin{cases}\text{reject}, & \text{if } \mathcal{L}(S_n) > \tau(\alpha) \\ \text{accept}, & \text{if } \mathcal{L}(S_n) \leq \tau(\alpha)\end{cases} \tag{2.30}$$

where $\tau(\alpha)$ is a threshold that controls the false error rate α. A rejection corresponds to the detection of an attack, while an acceptance corresponds to the lack of detection of an attack. This notation emphasizes the fact that achieving a specified false error rate α (a false error in our context corresponds to detecting an attack when there is no attack occurring) requires changing the threshold $\tau(\alpha)$.

Relationship to Existing Tests

It is interesting to compare our test (2.11)–(2.12) to those tests designed in [222]. More specifically, [222] designed a related sequence of tests adapted to different (and less complex) assumptions about the model dynamics. We will show that our test is closely related to (and generalizes) these previous tests developed under assumptions of less complex dynamics.

The simplest test in [222] was designed for systems with direct state measurement (i.e., $C = \mathbb{I}$), no measurement error (i.e, $z_n \equiv 0$), and full rank input matrix (i.e, $\text{rank}(B) = p$). The SISO (i.e., $m = p = q = 1$) and MIMO cases were considered separately in [222], though the SISO case is a special case of the MIMO case. If we choose $L = -A$, then we have that: $y_n = x_n + v_n$, $x_{n+1} = Ax_n + BK\hat{x}_n + Be_n + w_n$, $\hat{x}_{n+1} = Ay_n + BK\hat{x}_n + Be_n$, $\Sigma_\Delta = \Sigma_W$, and $k' = 1$ since $\text{rank}(CB) = p$. So our test (2.11)–(2.12) simplifies to

$$\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} (y_{n+1} - Ay_n - BK\hat{x}_n - Be_n) \times (y_{n+1} - Ay_n - BK\hat{x}_n - Be_n)^\mathsf{T} = \Sigma_W \quad (2.31)$$

$$\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} (y_{n+1} - Ay_n - BK\hat{x}_n - Be_n) \times e_n^\mathsf{T} = 0. \quad (2.32)$$

This exactly matches the test designed in [222] for LTI systems with the above described properties.

A more complex test in [222] was designed for SISO (i.e., $m = q = 1$) systems with partial state measurement. In our notation, the tests in [222] for this case simplify to

$$\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} L(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\mathsf{T} L^\mathsf{T} = L\big(C\Sigma_\Delta C^\mathsf{T} + \Sigma_Z\big)L^\mathsf{T} \quad (2.33)$$

$$\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} L(C\hat{x}_n - y_n)e_{n-1}^\mathsf{T} = 0. \quad (2.34)$$

But $k' = 1$ since B is a nonzero vector and $\text{rank}(CB) = 1$ in this case. So the test (2.33)–(2.34) from [222] essentially matches our test (2.11)–(2.12), but with the difference that the test in [222] considers quantities with $L(C\hat{x}_n - y_n)$, while our test directly considers quantities with $C\hat{x}_n - y_n$; this is a negligible difference since $C\hat{x}_n - y_n$ is a scalar in this SISO case.

2.2 Simulations: Dynamic Watermarkig for Autonomous Vehicle

A standard model [241] for error kinematics of lane keeping and speed control has $x^\mathsf{T} = \begin{bmatrix}\psi & y & s & \gamma & v\end{bmatrix}$ and $u^\mathsf{T} = \begin{bmatrix}r & a\end{bmatrix}$, where ψ is heading error, y is lateral error, s is trajectory distance, γ is vehicle angle, v is vehicle velocity, r is steering, and a is acceleration.

Linearizing about a straight trajectory and constant velocity $v_0 = 10$, and then performing exact discretization with sampling period $t_s = 0.05$ yields

$$A = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{10} & 0 \\ \frac{1}{2} & 1 & 0 & \frac{1}{40} & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \frac{1}{400} & 0 \\ \frac{1}{2400} & 0 \\ 0 & \frac{1}{800} \\ \frac{1}{20} & 0 \\ 0 & \frac{1}{20} \end{bmatrix} \tag{2.35}$$

with $C = \begin{bmatrix} I & 0 \end{bmatrix} \in \mathbb{R}^{3\times 5}$. We used process and measurement noise with $\Sigma_W = 10^{-8}$ and $\Sigma_W = 10^{-5}$, respectively. Our simulations used the wind model: $d_{n+1} = 0.9d_n + \chi_n$, where χ_n are i.i.d. zero mean Gaussians with $\sigma_\chi^2 = 2 \times 10^{-6}$, and the wind state d entered additively into the y dynamics.

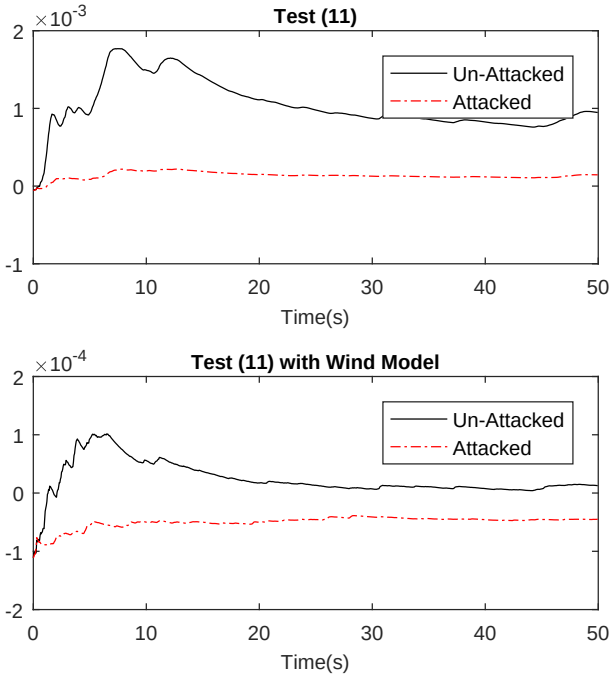

Figure 2.1: Deviation of (2.11) in Simulation of Autonomous Vehicle

We applied our tests using a dynamic watermark with variance $\Sigma_E = \frac{1}{2}\mathbb{I}$, where K and L were chosen to stabilize the closed-loop system without an attack. We conducted four simulations: Un-attacked and attacked simulations were conducted with a test computed without wind in the system model, and un-attacked and attacked simulations were conducted with a test computed with wind in the system model.

In both attack simulations, we chose an attacker with $\alpha = -0.6$, $\xi_0 = 0$, $\Sigma_O = 10^{-8}$, and $\Sigma_S = 10^{-8}$. Fig. 2.1 shows $\|\frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\mathsf{T} - C\Sigma_\Delta C^\mathsf{T} - \Sigma_Z\|$, and Fig. 2.2

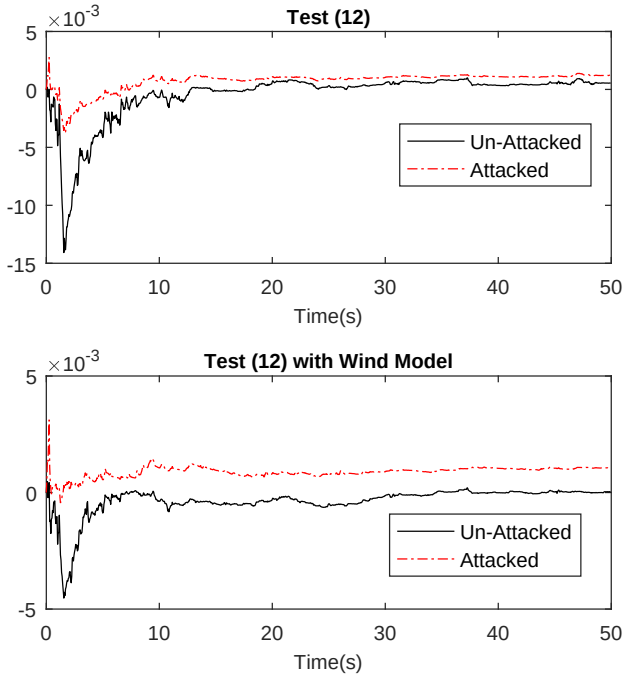

Figure 2.2: Deviation of (2.12) in Simulation of Autonomous Vehicle

shows $\|\frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)e_{n-k'-1}^{\mathsf{T}}\|$. If the test is detection consistent, then these values go to zero. The plots show dynamic watermarking cannot detect the presence or absence of an attack when wind affects the system dynamics but is not included in the test, while our test sdetect the presence or absence of an attack when a model of wind is included in the test. Fig. 2.3 shows the results of applying our statistical test (2.29), and the same behavior is seen.

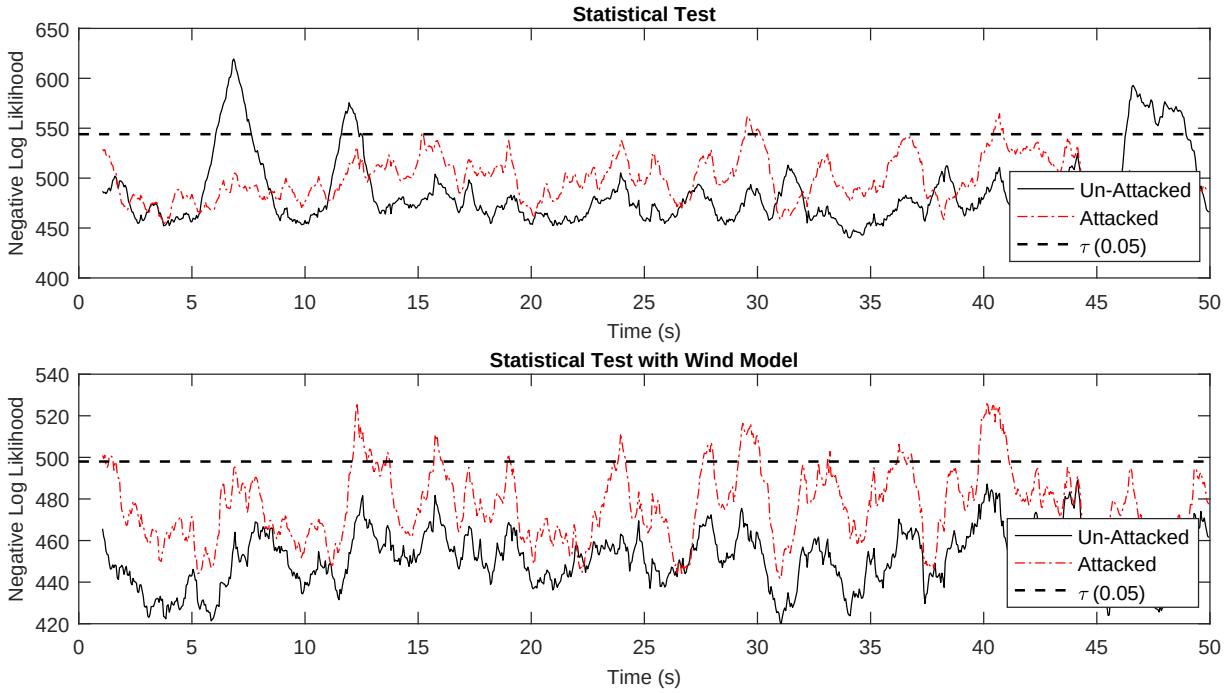

Figure 2.3: Value of (2.29) for Simulation of Autonomous Vehicle, with a Negative Log-Likelihood Threshold for $\alpha = 0.05$ False Detection Error Rate

2.3 Statistical Watermarking for Networked Control Systems

On this section, we develop a statistical watermarking approach for detecting malicious sensor and communication attacks on networked LTI systems. Our first contribution is to design a watermarking test using null hypothesis testing [172, 174, 171, 252, 173], and this requires characterizing the statistics of states and private watermarking signals under the dynamics of multiple subcontrollers within the networked system. A unique feature (as compared to the non-networked setting) of the watermarking scheme is it requires the state-feedback control to be such that the closed-loop system is controllable by each subcontroller, because otherwise a subcontroller could not independently verify the lack of an attack. Our second contribution is to provide two approaches to constructing such a state-feedback control, and this partly involves deriving a multi-input generalization of Heymann's lemma [134, 120].

Most watermarking [172, 174, 171, 252, 173, 130] is for LTI systems with a centralized controller, and only the approaches of [222, 143] can be used with networked LTI systems; however, the approaches [222, 143] require full state observation, which is not the case for many systems.

Our first contribution is to develop watermarking for networked LTI systems with partial state observation.[143, 107, 15, 255, 243, 175, 69] may benefit from watermarking. For instance, [143] considered the use of dynamic watermarking to detect sensor attacks in a network of autonomous vehicles coordinated by a supervisory controller; the watermarking approach was successfully able to detect attacks. However, large-scale deployments of ITS must be resilient in the face of partial state observations and partially distributed control structures. For example, vehicle platoons are susceptible to malicious interference of GPS and the communication channel between vehicles [228, 185, 54]. Our third contribution is to conduct a simulation that shows the efficacy of our watermarking scheme in detecting attacks on a vehicle platoon.

We first provide a model of the networked LTI system we consider, and specifies a model for communication and sensor attacks. Next, we present an example to give intuition about the new challenges with designing watermarking for networked systems. We construct a statistical watermarking test which allows each subcontroller to independently check for the presence of communication or sensor attacks. Our tests require a state-feedback controller such that the closed-loop system is controllable by each subcontroller, and we provide two methods for constructing such a controller.

Networked LTI System and Attack Modalities

We study a setting with κ subcontrollers. The subscripts i or j denote the i-th or j-th subcontroller, and the subscript n indicates time. Consider the LTI system with dynamics

$$x_{n+1} = Ax_n + \textstyle\sum_{i=1}^{\kappa} B_i u_{i,n} + w_n, \tag{2.36}$$

where $x \in \mathbb{R}^p$ is the state, $u_i \in \mathbb{R}^{q_i}$ is the input of the i-th subcontroller, and $w \in \mathbb{R}^p$ is a zero mean i.i.d. process noise with a jointly Gaussian distribution and covariance Σ_W. Each subcontroller steers a subset of the actuators, and each subcontroller makes the partial state observations

$$y_{i,n} = C_i x_n + z_{i,n} + v_{i,n}, \tag{2.37}$$

where $y_i \in \mathbb{R}^{m_i}$ is the observation of the i-th subcontroller, $z_i \in \mathbb{R}^{m_i}$ is zero mean i.i.d. measurement noise with a jointly Gaussian distribution and covariance $\Sigma_{Z,I}$, and $v_i \in \mathbb{R}^{m_i}$ should be interpreted as an additive measurement disturbance that is added by an attacker.

Network Communication Model

The LTI system here is networked in the following sense: The dynamics and partial observations are such that for

$$B = \begin{bmatrix} B_1 & \cdots & B_\kappa \end{bmatrix} \tag{2.38}$$

$$C^{\mathsf{T}} = \begin{bmatrix} C_1^{\mathsf{T}} & \cdots & C_\kappa^{\mathsf{T}} \end{bmatrix} \tag{2.39}$$

we have that (A, B) is stabilizable and (A, C) is detectable. In general, (A, B_i) is not stabilizable for some (or all) i, and similarly (A, C_i) is not detectable for some (or all) i. Thus coordination is required between subcontrollers to ensure closed-loop stability, and networking arises because we assume each subcontroller communicates its own partial state observations to all other subcontrollers. (Our setting assumes communication has zero cost.) Consider the values

$$s_{i,j,n} = y_{j,n} + \nu_{i,j,n}, \tag{2.40}$$

where $s_{i,j} \in \mathbb{R}^{m_j}$ is the value communicated to subcontroller i of the measurement made by subcontroller j, and $\nu_{i,j,n} \in \mathbb{R}^{m_j}$ should be interpreted as an additive communication disturbance added by an attacker. Clearly $\nu_{i,i,n} \equiv 0$ for all i, since the i-th subcontroller already has its own measurement.

Controller and Observer Structure

The idea of statistical watermarking in this context will be to superimpose a private (and random) excitation signal $e_{i,n}$ known in value to only the i-th subcontroller but unknown in value to the attacker or to the other subcontrollers. We will apply the control input $u_{i,n} = K_i\hat{x}_{i,n} + e_{i,n}$, where $\hat{x}_{i,n}$ is the observer-estimated state (the subscript i here indicates that each subcontroller operates its own observer, and that $\hat{x}_{i,n}$ is the state estimated by the observer of the i-th subcontroller) and $e_{i,n}$ are i.i.d. Gaussian with zero mean and constant variance $\Sigma_{E,I}$ fixed by the subcontrollers.

Now let K_i be constant state-feedback gain matrices such that $A + \sum_{i=1}^{\kappa} B_i K_i$ is Schur stable, and let L_i be constant observer gain matrices. It will be useful to define

$$K^{\mathsf{T}} = \begin{bmatrix} K_1^{\mathsf{T}} & \cdots & K_\kappa^{\mathsf{T}} \end{bmatrix}, \quad L = \begin{bmatrix} L_1 & \cdots & L_\kappa \end{bmatrix} \tag{2.41}$$

Then the closed-loop system with private excitation is

$$\begin{aligned} x_{n+1} &= Ax_n + \sum_{j=1}^{\kappa} B_j(K_j\hat{x}_{j,n} + e_{j,n}) + w_n \\ \hat{x}_{i,n+1} &= (A + \sum_{j=1}^{\kappa} B_jK_j + \sum_{j=1}^{\kappa} L_jC_j)\hat{x}_{i,n} - \sum_{j=1}^{\kappa} L_jC_jx_n + B_ie_{i,n} - \sum_{j=1}^{\kappa} L_j(z_{j,n} + v_{j,n} + \nu_{i,j,n}) \end{aligned} \tag{2.42}$$

These equations represent the fact that each subcontroller has its own observer using the measurements that it has received. It is not clear *a priori* that this closed-loop system is stable since each observer may start at a different initial condition. This concern is resolved by the following result:

Proposition 2.3.1. *Let K_i and L_i be constant state-feedback and observer gains such that $A + BK$, $A + LC$, and $A + BK + LC$ are Schur stable. The closed-loop system (2.42) is Schur stable with no private excitation $e_{j,n} \equiv 0$, process noise $w_n \equiv 0$, measurement noise $z_{j,n} \equiv 0$, measurement attack $v_{j,n} \equiv 0$, and communication attack $\nu_{i,j,n} \equiv 0$.*

Proof. Consider the change of variables from the states $x, \hat{x}_i$ to the states x, δ_1, d_i where $\delta_1 = \hat{x}_1 - x$ and $d_i = \hat{x}_i - \hat{x}_1$ for $i = 2, \dots, \kappa$. Then inserting this change of variables into (2.42) gives

$$\begin{aligned} x_{n+1} &= (A + \sum_{j=1}^{\kappa} B_j K_j) x_n + (\sum_{j=1}^{\kappa} B_j K_j)\delta_{1,n} + \sum_{j=2}^{\kappa} B_j K_j d_{j,n} \\ \delta_{1,n+1} &= (A + \sum_{j=1}^{\kappa} L_j C_j)\delta_{1,n} - \sum_{j=2}^{\kappa} B_j K_j d_{j,n} \\ d_{i,n+1} &= (A + \sum_{j=1}^{\kappa} B_j K_j + \sum_{j=1}^{\kappa} L_j C_j) d_{i,n}, \qquad \text{for } i = 2, \dots, \kappa \end{aligned} \tag{2.43}$$

If we put x, δ_i into a single vector $\check{x}$, then the dynamics $\check{x}_{n+1} = \check{A}\check{x}_n$ are such that $\check{A}$ is a block upper-triangular matrix with $A + BK$, $A + LC$, and $A + BK + LC$ on the diagonal. This means $\check{A}$ is Schur stable since we assumed $A + BK$, $A + LC$, and $A + BK + LC$ are Schur stable. □

Remark 2.3.2. This result implies that the separation principle does not hold. Fortunately, this is not a substantial impediment from the standpoint of design. Given a K such that $A + BK$ is stable, we can solve an LMI formulation [56, 187] to choose (when feasible) an L such that both $A + LC$ and $A + BK + LC$ are Schur stable. In particular, suppose there exists a positive definite matrix $Q \succ 0$ and general matrix R such that the following two LMI's

$$\begin{aligned} &\begin{bmatrix} Q & A^{\mathsf{T}}Q + C^{\mathsf{T}}R \\ Q^{\mathsf{T}}A + R^{\mathsf{T}}C & Q \end{bmatrix} \succ 0 \\ &\begin{bmatrix} Q & (A+BK)^{\mathsf{T}}Q + C^{\mathsf{T}}R \\ Q^{\mathsf{T}}(A+BK) + R^{\mathsf{T}}C & Q \end{bmatrix} \succ 0 \end{aligned} \tag{2.44}$$

are satisfied. Then choosing $L = Q^{-1}R^{\mathsf{T}}$ ensures that $A + LC$ and $A + BK + LC$ are Schur stable. Convex optimization can be used to determine if these LMI's have a solution, and compute a solution if possible.

For the purpose of designing our test, it will be useful to define another change of variables on the states. Consider the change of variables from the states $x, \hat{x}_i$ to the states x, δ_i where $\delta_i = \hat{x}_i - x$ for $i = 2, \dots, \kappa$. If there is no measurement attack $v_{j,n} \equiv 0$ and no communication attack $\nu_{i,j,n} \equiv 0$, then a straightforward calculation gives

$$\begin{aligned} x_{n+1} &= (A + BK)x_n + \sum_{j=1}^{\kappa} B_j(K_j\delta_{j,n} + e_{j,n}) + w_n \\ \delta_{i,n+1} &= (A + BK + LC)\delta_{i,n} + B_i e_{i,n} - \sum_{j=1}^{\kappa} L_j z_{j,n} - \sum_{j=1}^{\kappa} B_j(K_j\delta_{j,n} + e_{j,n}) - w_n \end{aligned} \tag{2.45}$$

If we define $\Delta^\mathsf{T} = \begin{bmatrix}\delta_1^\mathsf{T} & \cdots & \delta_\kappa^\mathsf{T}\end{bmatrix}$ and $E^\mathsf{T} = \begin{bmatrix}e_1^\mathsf{T} & \cdots & e_\kappa^\mathsf{T}\end{bmatrix}$, then the above dynamics for the δ_i can be written as

$$\Delta_{n+1} = \underline{A}\Delta_n + \text{blkdiag}(B_1, \ldots, B_\kappa)E_n - \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^\mathsf{T} \otimes \left(-w_n + \sum_{j=1}^{\kappa} L_j z_{j,n} + B_j e_{j,n}\right) \tag{2.46}$$

where $\otimes$ is the Kronecker product, $\text{blkdiag}(B_1, \ldots, B_\kappa)$ is the block diagonal matrix with $B_1, \ldots, B_\kappa$ on the diagonals, and $\underline{A}$ is the corresponding matrix defined to make the above equivalent to (2.45). This will be used to define our test.

Intuition for Watermarking Design

Watermarking for networked systems faces new challenges not encountered in the non-networked setting. To illustrate the new difficulty, consider the networked LTI system with

$$A = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix} \quad B_1 = \begin{bmatrix}1\\ 0\end{bmatrix} \quad B_2 = \begin{bmatrix}0\\ 1\end{bmatrix} \quad \begin{matrix}C_1 = \begin{bmatrix}1 & 0\end{bmatrix}\\ C_2 = \begin{bmatrix}0 & 1\end{bmatrix}\end{matrix} \tag{2.47}$$

In this example, (A, B_1) is not stabilizable and (A, C_2) is not detectable. And so coordination is required between the subcontrollers to stabilize the system.

For instance, the choice $K = -\frac{1}{2}\mathbb{I}$ makes $A + BK$ Schur stable, and implementing the corresponding output-feedback controller requires communication of partial observations between the two subcontrollers. In this case, the design is such that the first subcontroller cannot inject any watermarking signal into the second state, while the second subcontroller cannot inject any watermarking signal into the first state. This is problematic because this means each subcontroller cannot verify the accuracy of the communicated state information.

However, suppose we instead choose

$$K = -\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} \tag{2.48}$$

Then $A + BK$ is Schur stable. More importantly, $(A + BK, B_1)$ and $(A + BK, B_2)$ are controllable with this K. Thus each subcontroller can inject a private watermarking signal known only to the subcontroller, and such that this signal can be used to verify the accuracy of the communicated state information and of the partial observations.

This example shows that designing watermarking differs in the networked and non-networked cases. The networked case requires designing both the state-feedback controller and the corresponding tests to detect attacks; whereas watermarking in the non-networked case only requires designing the the corresponding tests to detect attacks [172, 174, 171, 252, 173, 130].

Specification of Statistical Test

Though watermarking for networked systems requires designing both the state-feedback controller and watermarking tests, we first focus on the latter. We construct a statistical test using the framework of null hypothesis testing, after assuming the existence of a state-feedback controller satisfying:

Condition 1. Let $k'_{i,j} = \min\{k \geq 0 \mid C_j(A+BK)^k B_i \neq 0\}$. For each i and j, there exists a $k'_{i,j} \leq p-1$

This condition is itself nontrivial because it may be that $C_j(A+BK)^k B_i \equiv 0$ for all $k \geq 0$. Approaches to synthesize a state-feedback controller K to ensure the above condition holds will be shown in the next section. This property is important because it means the watermarking signal of the i-th subcontroller is seen in the j-th output when the system is controlled by perfect-information state-feedback.

Variable Definitions

Now before specifying the test, it is useful to define some variables. Suppose we have K, L such that $A+BK$, $A+LC$, and $A+BK+LC$ are Schur stable. Let Σ_Δ be the positive semidefinite matrix that solves the Lyapunov equation

$$\Sigma_\Delta = \underline{A}\Sigma_\Delta \underline{A}^\mathsf{T} + \mathrm{blkdiag}(B_1,\ldots,B_\kappa)\Sigma_E \mathrm{blkdiag}(B_1,\ldots,B_\kappa)^\mathsf{T} + \\ - \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}^\mathsf{T} \otimes \left(\Sigma_W + \sum_{j=1}^{\kappa} L_j \Sigma_{Z,J} (L_j)^\mathsf{T} + B_j \Sigma_{E,J} (B_j)^\mathsf{T}\right) \quad (2.49)$$

where $\Sigma_E = \mathrm{blkdiag}(\Sigma_{E,1},\ldots,\Sigma_{E,\kappa})$. A solution exists because the above is a Lyapunov equation and since Proposition 2.3.1 ensured stability. Note by construction

$$\Sigma_\Delta = \text{as-lim}_N \frac{1}{N}\sum_{n=0}^{N-1} \Delta_n \Delta_n^\mathsf{T} \quad (2.50)$$

when there is no attack (i.e., $v_{i,n} \equiv 0$ and $\nu_{i,j,n} \equiv 0$ for all i,j,n). If we divide $\Sigma_\Delta \in \mathbb{R}^{\kappa p \times \kappa p}$ into sub-matrices with dimension $p \times p$, then define $D_I \in \mathbb{R}^{p\times p}$ to be the $i \times i$-th sub-matrix of Σ_Δ.

Lastly, we consider the matrix dynamics

$$\mathbb{E}(\Delta_{n+1} e_{i,t}^\mathsf{T}) = \underline{A}\mathbb{E}(\Delta_n e_{i,t}^\mathsf{T})) + f_i \otimes B_i \Sigma_{E,I} \cdot \mathbf{1}(t=n) - \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\mathsf{T} \otimes (B_i \Sigma_{E,I}) \cdot \mathbf{1}(t=n), \quad (2.51)$$

where $\mathbf{1}(\cdot)$ is an indictor function, and the vector f_i has a one in the i-th position and is zero otherwise. This means

$$\Sigma_{\Delta,I,k} := \mathbb{E}(\Delta_n e_{i,n-k-1}^\mathsf{T}) = \underline{A}^k f_i \otimes B_i \Sigma_{E,I} - \underline{A}^k \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\mathsf{T} \otimes (B_i \Sigma_{E,I}) \quad (2.52)$$

If we divide $\Sigma_{\Delta,I,k} \in \mathbb{R}^{\kappa p \times q_i}$ in sub-matrices of size $p \times q_i$, then let $Q_{I,J,k} \in \mathbb{R}^{p \times q_i}$ be the j-th sub-matrix of $\Sigma_{\Delta,I,k}$.

Definition of Test

Our statistical watermarking test will involve the (second-order) statistical characterization of the vectors defined as

$$\psi_{n,i,j} = \begin{bmatrix} e_{i,n-k'_{i,j}-1} \\ C_j \hat{x}_{i,n} - s_{i,j,n} \end{bmatrix}, \tag{2.53}$$

and this characterization will be used to specify the distribution corresponding to the null hypothesis of no attack.

Theorem 2.3.3. *If we have that $v_{i,n} \equiv 0$ and $\nu_{i,j,n} \equiv 0$, then* $\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} \psi_n \psi_n^T = R_{I,J}$, *where*

$$R_{I,J} = \begin{bmatrix} \Sigma_{E,I} & Q_{I,J,k'_{i,j}}^T C_j^T \\ C_j Q_{I,J,k'_{i,j}} & C_j D_i C_j^T + \Sigma_{Z,J} \end{bmatrix}. \tag{2.54}$$

Moreover, we have that $\text{as-lim}_n \mathbb{E}(\psi_n) = 0$.

Proof. First note that we have $\text{as-lim}_n \mathbb{E}(\delta_{i,n}) = 0$ by the stability from Proposition 2.3.1. But $C_j \hat{x}_{i,n} - s_{i,j,n} = C_j \delta_{i,n} - z_{j,n}$, and so $\mathbb{E}(C_j \hat{x}_{i,n} - s_{i,j,n}) = C_j \mathbb{E}(\delta_{i,n})$. This implies $\text{as-lim}_n \mathbb{E}(C_j \hat{x}_{i,n} - s_{i,j,n}) = 0$, which proves $\text{as-lim}_n \mathbb{E}(\psi_n) = 0$ since the e_i have zero mean.

Next observe the upper block triangle of (2.54) is correct by construction of $Q_{I,J,k'_{i,j}}$ and by definition of the e_i, and so we only have to prove that the lower-right block is correct. In particular, note that $\mathbb{E}((C_j \delta_{i,n} - z_n)(C_j \delta_{i,n} - z_n)^\mathsf{T}) = \mathbb{E}((C_j \delta_{i,n})(C_j \delta_{i,n})^\mathsf{T}) + \Sigma_{Z,I}$ since $z_{i,n}$ is independent of $\delta_{i,n}$ by (2.45). This implies that we have that $\text{as-lim}_n \mathbb{E}((C_j \delta_{i,n} - z_n)(C_j \delta_{i,n} - z_n)^\mathsf{T}) = C_j D_i C_j^\mathsf{T} + \Sigma_{Z,I}$. □

This result means that asymptotically the summation $S_{i,j} = \frac{1}{\ell} \sum_{n+1}^{n+\ell} \psi_{n,i,j} \psi_{n,i,j}^\mathsf{T}$ with $\ell \geq (m_i + q_i)$ has a Wishart distribution with ℓ degrees of freedom and a scale matrix that matches (2.54), and we use this to define a statistical test. In particular, we check if the negative log-likelihood

$$\mathcal{L} = \sum_{j=1}^{\kappa} (1 - \ell + m_i + q_i) \cdot \log\det S_{n,i,j} + \sum_{j=1}^{\kappa} \text{trace}\left(R_{I,J}^{-1} \cdot S_{n,i,j}\right) \tag{2.55}$$

corresponding to this Wishart distribution and the summations of $S_{n,i,j}$ is large by conducting the hypothesis test

$$\begin{cases} \text{reject}, & \text{if } \mathcal{L}(S_n) > \tau(\alpha) \\ \text{accept}, & \text{if } \mathcal{L}(S_n) \leq \tau(\alpha) \end{cases} \tag{2.56}$$

where $\tau(\alpha)$ is a threshold that controls the false error rate α. A rejection corresponds to the detection of an attack, while an acceptance corresponds to the lack of detection of an attack.

This notation emphasizes the fact that achieving a specified false error rate α (a false error in our context corresponds to detecting an attack when there is no attack occurring) requires changing the threshold $\tau(\alpha)$.

Designing the State-Feedback

We provide two approaches for designing a state-feedback controller that satisfies Condition 1. The first applies when B is square (i.e., $\sum_{i=1}^{k} q_i = p$); though it generalizes to skinny B (i.e., $\sum_{i=1}^{k} q_i < p$) in some cases, we do not prove this. The first approach relies upon a multi-input generalization (which we construct and prove) of Heymann's lemma [134, 120]. The second approach applies to B of arbitrary size where the range spaces of B_i have a nonempty intersection.

Algorithm 1 Compute State-Feedback K for Proposition 2.3.5

$x_1 := \frac{b_1}{\|b_1\|}$
for all $k \in \{1, \ldots, p-1\}$ **do**
  $x_{k+1} := \lambda^k \frac{b_{k+1}}{\|b_{k+1}\|}$
  $u_k := B^{-1}\big(x_{k+1} - Ax_k\big)$
end for
$x_{p+1} := \lambda^p \frac{b_1}{\|b_1\|}$
$u_p := B^{-1}\big(x_{p+1} - Ax_p\big)$
$X := \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}, \quad U := \begin{bmatrix} u_1 & \ldots & u_p \end{bmatrix}, \quad K := UX^{-1}$

Multiple Input Heymann's Lemma

Heymann's lemma [134, 120] is used to prove arbitrary pole placement of controllable, multiple input LTI systems by allowing a reduction to the case of arbitrary pole placement of a controllable, single input LTI system. Formally, it says

Lemma 2.3.4 (Heymann's Lemma)**.** *If (A, B) is controllable, then for any $b = Bv \neq 0$ there exists K (that depends on b) such that $(A + BK, b)$ is controllable.*

We need a multiple input generalization of Heymann's Lemma. Let b_i denote the i-th column of the matrix B. Then

Proposition 2.3.5. *If B is full rank and square-shaped (i.e., $B \in \mathbb{R}^{p\times p}$); then there exists a single K such that $A + BK$ is Schur stable and $(A + BK, b_i)$ is controllable for all i.*

Proof. We prove this result stepwise. Since B is full rank and square-shaped, its columns are linearly independent. Consider any λ with $0 < |\lambda| < 1$, and define $x_1 = \frac{b_1}{\|b_1\|}$ and

$x_{n+1} = Ax_n + Bu_n$. Now suppose there exists $u_1, \ldots, u_{k-1}$ such that $x_i = \lambda^{i-1}\frac{b_i}{\|b_i\|}$ for $i = 1, \ldots, k$. If $k < p$, then there exists a u_k satisfying $\lambda^k \frac{b_{k+1}}{\|b_{k+1}\|} - \lambda^{k-1} A \frac{b_k}{\|b_k\|} = Bu_k$ since B is full rank. Hence by definition of the dynamics on x there exists u_k such that $x_{k+1} = \lambda^k \frac{b_{k+1}}{\|b_{k+1}\|}$. If $k = p$, then there exists a u_p satisfying $\lambda^p \frac{b_1}{\|b_1\|} - \lambda^{p-1} A \frac{b_p}{\|b_p\|} = Bu_p$ since B is full rank. So by definition of the dynamics on x there exists u_p such that $x_{p+1} = \lambda^p \frac{b_1}{\|b_1\|}$.

Next define the matrices

$$U = \begin{bmatrix} u_1 & \cdots & u_p \end{bmatrix} \tag{2.57}$$

$$R = \begin{bmatrix} \frac{b_1}{\|b_1\|} & \cdots & \frac{b_p}{\|b_p\|} \end{bmatrix} \tag{2.58}$$

$$\Lambda = \operatorname{diag}\left(1, \lambda, \ldots, \lambda^{p-1}\right) \tag{2.59}$$

and $K = UR^{-1}\Lambda^{-1}$. The matrices Λ and R are invertible by construction since $0 < |\lambda| < 1$ and B is invertible. Note that by definition $U = K\Lambda B$. Finally, note for any i we have

$$\|b_i\| \cdot \begin{bmatrix} \frac{b_i}{\|b_i\|} & \cdots & \frac{b_p}{\|b_p\|} & \frac{b_1}{\|b_1\|} & \cdots & \frac{b_{i-1}}{\|b_{i-1}\|} \end{bmatrix} \cdot \Lambda = \begin{bmatrix} b_i & (A+BK)b_i & \cdots & (A+BK)^{p-1}b_i \end{bmatrix} \tag{2.60}$$

The left side has full rank by the assumptions on B, and the right side is the observability matrix for the $(A + BK, b_i)$. This proves $(A + BK, b_i)$ is observable for all i since we have shown that the observability matrix has full rank.

We conclude by proving that the above designed K makes $A+BK$ Schur stable. Consider any $x \in \mathbb{R}^p$, and observe that by the assumptions on B there exists $z \in \mathbb{R}^p$ such that $x = Rz$. But, as in (2.60), by construction $(A + BK)^p \frac{b_i}{\|b_i\|} = \lambda^p \frac{b_i}{\|b_i\|}$ for all i. Hence $(A + BK)^p x = (A + BK)^p Rz = \lambda^p Rz = \lambda^p x$ for all x. This means all the eigenvalues of $(A + BK)^p$ are λ^p, and using the spectral mapping theorem implies the eigenvalues of $A + BK$ are roots of λ^p. Thus the magnitude of the eigenvalues of $A + BK$ are $|\lambda|$, which means that $A + BK$ is Schur stable since $0 < |\lambda| < 1$. □

Though the above is an existence result, a state-feedback matrix K satisfying Proposition 2.3.5 can be computed using Algorithm 1. The correctness of this algorithm follows from the construction used in the proof of Proposition 2.3.5. Also, the next result proves that this K satisfies Condition 1.

Corollary 2.3.6. *Suppose $C_j \neq 0$ for all j. If B is full rank and square-shaped (i.e., $B \in \mathbb{R}^{p\times p}$); then there exists a K such that $A + BK$ is Schur stable and that Condition 1 holds.*

Proof. Consider any $i \in \{1, \ldots, \kappa\}$, and choose s to be any index such that the s-th column in B belongs to B_i. Proposition 2.3.5 says $(A + BK, b_s)$ is controllable. This means the controllability matrix

$$\mathfrak{C}' = \begin{bmatrix} b_s & (A+BK)b_s & \ldots & (A+BK)^{p-1}b_s \end{bmatrix} \tag{2.61}$$

has $\text{rank}(\mathfrak{C}') = p$, and so the controllability matrix

$$\mathfrak{C} = \begin{bmatrix} B_i & (A+BK)B_i & \dots & (A+BK)^{p-1}B_i \end{bmatrix} \tag{2.62}$$

also has $\text{rank}(\mathfrak{C}) = p$ since the columns of $\mathfrak{C}$ are a superset of the columns of $\mathfrak{C}'$. Thus by Sylvester's rank inequality, we have $\text{rank}(C_j\mathfrak{C}) \geq \text{rank}(C_j) + \text{rank}(\mathfrak{C}) - p = \text{rank}(C)$. But $\text{rank}(C_j) \geq 1$ since $C_j \neq 0$. Combining this with the earlier inequality gives $\text{rank}(C_j\mathfrak{C}) \geq 1$, and so $C_j\mathfrak{C} \neq 0$. This means $k'_{i,j} \leq p-1$ exists since $C_j\mathfrak{C}$ is a block matrix consisting of the blocks $C_j(A+BK)^kB_i$. □

Algorithm 2 Compute State-Feedback K for Proposition 2.3.7

choose any v such that $Bv \in \cap_{i=1}^{\kappa} \text{range}(B_i)$
compute K' satisfying Heymann's lemma for Bv
compute G such that $A + BK' + BvG$ is Schur stable
$K := K' + vG$

Nonempty Intersection of Inputs

We next consider B with arbitrary shape, such that the range spaces of B_i have a nonempty intersection. Our Algorithm 2 designs a K for this case, and it uses Heymann's lemma [134, 120]. The next result proves its correctness.

Proposition 2.3.7. *Suppose $C_j \neq 0$ for all j and that we have $\cap_{i=1}^{\kappa} \text{range}(B_i) \neq \emptyset$. If (A, B) is controllable, then Algorithm 2 computes a K such that $A + BK$ is Schur stable and that Condition 1 is satisfied.*

Proof. First note that v exists by assumption, and so we can compute K' by Heymann's lemma. This means $(A + BK', Bv)$ is controllable, which implies that we can compute G such that $A + BK' + BvG$ is Schur stable. This proves that $A + BK$ is Schur stable since $K = K' + vG$. Next consider any $i \in \{1, \dots, \kappa\}$. Since $(A + BK', Bv)$ is controllable, this means that $(A + BK' + BvG, Bv)$ is controllable. (This uses the well known fact that state-feedback does not affect controllability.) As a result, we have

$$\mathfrak{C}' = \begin{bmatrix} Bv & (A+BK)Bv & \dots & (A+BK)^{p-1}Bv \end{bmatrix} \tag{2.63}$$

has $\text{rank}(\mathfrak{C}') = p$. But by assumption, there exists v_i such that $B_iv_i = Bv$. So if we define

$$\mathfrak{C} = \begin{bmatrix} B_i & (A+BK)B_i & \dots & (A+BK)^{p-1}B_i \end{bmatrix}, \tag{2.64}$$

then we have that $\mathfrak{C}' = \mathfrak{C}\,\text{blkdiag}(v_i, \dots, v_i)$. Thus $p \geq \text{rank}(\mathfrak{C}) \geq \text{rank}(\mathfrak{C}') = p$. Thus by Sylvester's rank inequality, we have $\text{rank}(C_j\mathfrak{C}) \geq \text{rank}(C_j) + \text{rank}(\mathfrak{C}) - p = \text{rank}(C)$. But $\text{rank}(C_j) \geq 1$ since $C_j \neq 0$. Combining this with the earlier inequality gives $\text{rank}(C_j\mathfrak{C}) \geq 1$, and so $C_j\mathfrak{C} \neq 0$. This means $k'_{i,j} \leq p-1$ exists since $C_j\mathfrak{C}$ is a block matrix consisting of the blocks $C_j(A+BK)^kB_i$. □

2.4 Simulation: Autonomous Vehicle Platooning

We apply a standard model [242] for error kinematics of speed control of vehicles to generate a model for a three car platoon. The state vector for the three car platoon is $x^\mathsf{T} = \begin{bmatrix} e_1 & d_1 & e_2 & d_2 & e_3 & d_3 \end{bmatrix}$, where e_i is deviation from the desired velocity for car i, and d_i is deviation from the desired following distance between car i and car $i+1$. The discretized dynamics for a timestep of 0.05 seconds are then:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{20} & 1 & \frac{1}{20} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{20} & 1 & \frac{1}{20} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.65}$$

and

$$B_1 = \begin{bmatrix} \frac{1}{20} \\ -\frac{1}{800} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ \frac{1}{800} \\ \frac{1}{20} \\ -\frac{1}{800} \\ 0 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{800} \\ \frac{1}{20} \end{bmatrix}. \tag{2.66}$$

Assuming each car measures its own velocity and the distance to the car in front of it, we have

$$\begin{aligned} C_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ C_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\ C_2 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.67}$$

We assume that the process and measurement noise had variance $\Sigma_W = 5 \times 10^{-5} \cdot \mathbb{I}$ and $\Sigma_{Z,I} = 10^{-3} \cdot \mathbb{I}$, respectively.

We applied our statistical test (2.55) with each car using a watermarking signal with variance $\Sigma_{E,I} = 0.2$, where

$$\begin{aligned} K_1 &= \begin{bmatrix} -1 & 0.1 & 0 & 0 & 0 \end{bmatrix} \\ K_2 &= \begin{bmatrix} 1 & -1 & -2 & 0.1 & 0 \end{bmatrix} \\ K_3 &= \begin{bmatrix} 0.5 & -0.5 & 0.5 & -1 & -2 \end{bmatrix} \end{aligned} \tag{2.68}$$

and

$$L_1 = \begin{bmatrix} -0.5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \; L_2 = \begin{bmatrix} 0.05 & 0 \\ -0.5 & 0 \\ 0 & -0.5 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \; L_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.05 & 0 \\ -0.5 & 0 \\ 0 & -0.5 \end{bmatrix} \tag{2.69}$$

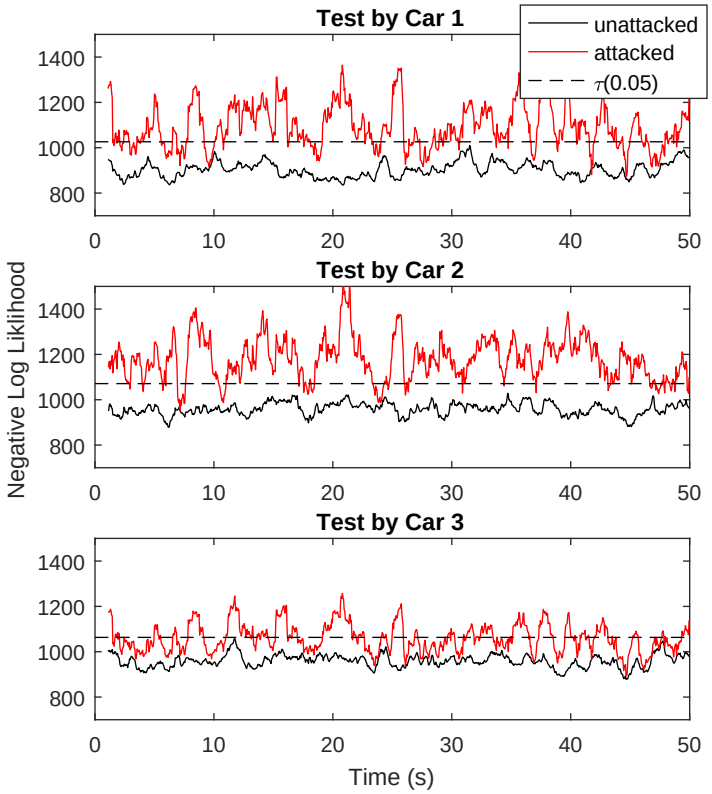

Figure 2.4: Value of (2.55) for Simulation of Vehicle Platoon, with a Negative Log-Likelihood Threshold for $\alpha = 0.05$ False Detection Error Rate

We conducted two simulations, where the platoon was un-attacked and attacked. In the attack simulations, the attacker chose $v_{1,n}$ and $\nu_{2,3,n}$ to be zero mean i.i.d. jointly Gaussian random variables with variance 0.5 and $0.2 \cdot \mathbb{I}$, respectively. Fig. 2.4 shows the results of applying our statistical test (2.55), and the plots show that our statistical watermarking test can detect the presence or absence of an attack.

Chapter 3

Switching in Cyber-Physical Systems: Finite-time consistency tests and estimation

Control system security is enhanced by the ability to detect malicious attacks on sensor measurements. As presented in the previous Chapter, Dynamic watermarking can detect such attacks on linear time-invariant (LTI) systems. However, existing theory focuses on attack detection and not on the use of watermarking in conjunction with attack mitigation strategies. In this chapter, we study the problem of switching between two sets of sensors: One set of sensors has high accuracy but is vulnerable to attack, while the second set of sensors has low accuracy but cannot be attacked. Though the design and analysis of intelligent transportation systems (ITS) has drawn renewed interest [143, 107, 15, 255, 243, 175, 69], there has been less work on secure control of ITS. One recent work considered the use of dynamic watermarking to detect sensor attacks in a network of autonomous vehicles coordinated by a supervisory controller[143], while [132] considered a platoon of vehicles where attacks happen not only on the sensors but also on the communication channel.

A particular feature of ITS is the possibility of redundancy in sensing. For instance, one can use a highly accurate satellite-based sensor (susceptible to external attack) and an on-board infrared sensor (*not* susceptible to external attack) in order to obtain spatial data. Then, one way of safeguarding a system susceptible to attacks is to switch from the high accuracy sensor to the on-board sensor when an attack is detected [170]. This approach naturally leads to systems with distributed observers with dynamic switching decision rules [36, 169]. In this scenario, it is crucial to design hypothesis tests that are able to detect attacks while having a decision rule that correctly selects which observer is to be used. Because control switching occurs at finite instances in time, the previous asymptotic results of dynamic watermarking cannot be used for this purpose. The reason, which is subtle, is that hypothesis tests based on characterization of asymptotic distributions will not have the correct theoretical properties in order to ensure proper control of the false alarm rate. Consequently, new finite sample hypothesis tests need to be constructed.

The problem is then to design a sensor switching strategy based on attack detection by dynamic watermarking. This requires new theory because existing results are not adequate to control or bound the behavior of sensor switching strategies that use finite data. To overcome this, we develop new finite sample hypothesis tests for dynamic watermarking in the case of bounded disturbances, using the modern theory of concentration of measure for random matrices. Our resulting switching strategy is validated with a simulation analysis in an autonomous driving setting, which demonstrates the strong performance of our proposed policy.

In Sect. 3.1, analyze Dynamic Watermarking applied to systems with switching. We also present the random matrix concentration inequalities that we use to perform our finite sample analysis. Next, we present our general LTI framework with switching observers. Then, we apply the concentration inequalities to the LTI setting in order to obtain appropriate concentration for the matrices involved. We present the finite sample consistency tests and a simple threshold that relates the attack magnitude to the power of our test. Next, in Sect. 3.2 we provide some numerical results demonstrating our approach on an autonomous vehicle application.

Consistency in Switched Linear Systems

Learning-based control has seen a resurgence in the past few years [16, 166, 189, 1] because of recent advances in system identification using machine learning and artificial intelligence. When the system has unknown dynamics, it becomes paramount to identify the underlying dynamics so that an appropriate controller can be computed in order to make the system stable [156]. System identification has become a central field of research lying in between control and statistics.

On section 3.3, we consider a fully observed switched autonomous linear system with bounded process noise. Each linear system is unknown to us, but we control switching between different linear dynamics. This departs from the existing literature on switched system identification, where the switching control is fixed and must also be estimated, such as in the work by [247, 99, 150]. Here, a control decision needs to be chosen together with the system identification. The dynamics for a single system may have a mix of stable or unstable modes and repeated eigenvalues. Identification can be done via estimation of the transition matrices [144, 231], and identification of transition matrices for *stable* systems has been studied [156, 232, 31, 256].

The identification problem in our setup is particularly challenging because the switching can cause stability/instability independent of the eigenvalues of each linear system [35]. The study of system identification for unstable systems is not as prolific as work on the stable case. Existing work for the unstable case of identification of a single linear system requires strong assumptions on repeated eigenvalues in order to prove asymptotic convergence [147], derive associated limiting distributions of the estimates of the model parameters [59, 60], and in order to generalize the result to other classes of transition matrices [146, 184, 182].

Recent work [86, 230, 229, 190] has shown the difficulty of identification for unstable linear systems when state observations are restricted to a single trajectory: Ordinary least squares (OLS) is statistically inconsistent when the dynamics have repeated unstable dynamics [183, 195], and this causes poor estimation when the dynamics have unstable modes with close eigenvalues. This can be partly overcome using instrumental variables, but this cannot handle systems matrices with eigenvalues both inside and outside the unit circle [195].

The set-membership estimator [39, 165, 17] exploits boundedness of the noise vector. This estimator has been studied in [73, 155] which provided a bounding ellipsoidal algorithm to obtain consistent estimators. Our work is related to previous studies where such estimators are applied, as in fault detection tests [47], regularized regression [32], robust estimation [100, 239], and kernel-based methods [66]. The work in [191] provides a greedy algorithm that uses a set-membership estimator to identify input-output models.

We defines our problem setup in refsec:Wald. Then we provide our proposed estimator and prove its statistical consistency. Next we consider an application of stabilizing a fully observed switched autonomous linear system with bounded process noise and unknown (to us) dynamics in each linear system.

3.1 Sensor Switching Control Under Attacks Detectable by Finite Sample Dynamic Watermarking Tests

As we saw in Chapter 1, we designed a dynamic watermarking consistency test for the MIMO LTI system with partial observations

$$\begin{aligned} x_{n+1} &= Ax_n + Bu_n + w_n \\ y_n &= Cx_n + z_n + v_n \end{aligned} \tag{3.1}$$

for some measurement noise z_n, system disturbance w_n, and attack vector v_n and where (A, B) is stabilizable, (A, C) is detectable. Where is the following holds

$$\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} \psi_n \psi_n^\top = \begin{bmatrix} C\Sigma_\Delta C^\top + \Sigma_Z & 0 \\ 0 & \Sigma_E \end{bmatrix} \tag{3.2}$$

where

$$\psi_n^\top = \begin{bmatrix} (C\hat{x}_n - y_n)^\top & e_{n-k'-1}^\top \end{bmatrix}, \tag{3.3}$$

for some specific matrices $\Sigma_\Delta, \Sigma_E, \Sigma_Z$, then all attack vectors v_n following a particular model [130] are constrained in power

$$\text{as-lim}_N \frac{1}{N} \sum_{n=0}^{N-1} v_n^\top v_n = 0. \tag{3.4}$$

where we use *as-lim* to denote the "almost sure limit", meaning that the given limit as N goes to infinity holds with probability one. Though these tests only provide asymptotic guarantees, that is enough to construct a statistical version of the test, similar to [222] where a hypothesis test is constructed by thresholding the negative log-likelihood. It follows that under a Gaussianity assumption for process and sensor noise, the matrix in (3.2) follows a well-behaved Wishart distribution. While that approach allows us to construct hypothesis tests using known distributions, the dependency of subsequent samples make finite sums display more complex behavior. Then it is up to the designer of the watermark to specify a threshold that controls the false error rate. In this framework a rejection of the hypothesis test corresponds to detection of an attack, while an acceptance corresponds to the lack of detection of an attack. This notation emphasizes the fact that achieving a specified false error rate requires changing the threshold.

The first contribution of this work is to provide finite-time guarantees on attack detection via dynamic watermarking, which to the best of our knowledge has not been done before. Namely, we provide statistical tests that provide finite-time guarantees on attack detection, instead of relying of asymptotic behavior of sums of random matrices. We also relate the magnitude of an attack to our test power, by describing the inherent trade-off between the test capability of triggering true detection, and the magnitude of the attacks that are allowed to remain undetected in the long run. The finite sample analysis of dynamic watermarking requires the use of random matrix concentration inequalities, which are useful in analyzing the matrices involved in the evolution of LTI system dynamics. The second major contribution of this paper is to provide finite sample concentration-based tests, which allow us to detect attacks and allow switching decisions based on such tests to correctly report attack detection infinitely often. Namely, if there is no attack, we develop a finite sample test that falsely reports attacks only a finite number of times. This is a crucial feature because it also implies that in the long-run the switching rule based on such a test is correctly selecting which observer is active infinitely often.

Lastly, we highlight the fact our switching rule provide a layer of mitigation of the attacks: By appropriately switching sensors, when an attack is detected, the system is still able to behave properly, until the attack presence is no longer detected. This measure of protection is key when coupled with the finite-time detection strategy, which is a novel contribution, to the best of our knowledge.

Preliminaries

In this section, we define all relevant notation concerning the random matrix analysis done throughout the paper. We also define the key concepts of Stein's Method [234, 159] applied to matrices and the relevant matrix concentration inequalities that will be used. This method turns out to be key to our finite sample analysis of dynamic watermarking, as it involves analyzing sums of inter-temporal dependent matrices. The importance of Stein's method lies precisely in obtaining finite-time bounds for a sequence of dependent random matrices, when the dependency follows a certain structure. The intuition behind it lies in the

"concentration of randomness", where the "average" of random matrices has probability tail behavior that can be bounded by appropriate expressions. The Stein's method for random matrices provide exactly such expressions in order to bound those summations.

We use the symbol $\|\cdot\|$ for the spectral norm of a matrix, which is the largest singular value of a general matrix. The space of $d\times d$ Hermitian real-valued matrices is denoted by $\mathcal{H}^d$. Moreover, the symbols $\lambda_{\max}(A), \lambda_{\min}(A)$ are respectively the maximum and the minimum eigenvalues of a Hermitian matrix $A \in \mathcal{H}^d$. The symbol $\preceq$ refers to the semidefinite partial order, namely $A \preceq B$ if and only if $B - A$ is positive semi-definite (p.s.d). For a matrix A, we let $(A)_{ij}$ denote the (ij)-th element of A. We let $\mathrm{tr}(\cdot)$ denote the trace operator.

We also define a master probability space $(\Omega, \mathcal{F}, \mathrm{P})$ and a filtration $\{\mathcal{F}_k\}$ contained in the master sigma algebra:

$$\mathcal{F}_k \subset \mathcal{F}_{k+1} \text{ and } \mathcal{F}_k \subset \mathcal{F}, \forall k \geq 0. \tag{3.5}$$

Note that the non-italicized $\mathrm{P}(\cdot)$ will refer to the probability measure. Given such filtration we also define the conditional expectation $\mathbb{E}_k[\cdot]$. We also let ϵ denote a Radamacher random variable, that takes values in $\{-1, 1\}$ with equal probability. The random matrix concentration inequalities involved in this work are derived using the method of exchangeable pairs based on the Stein's Method [234]. Let Z and Z' be random vectors taking values in a space $\mathbb{R}^d$. We say that (Z, Z') is an *exchangeable pair* if it has the same distribution as (Z', Z). Next, we define a matrix Stein pair:

Definition 3.1.1. Let Z and Z' be an exchangeable pair of random vectors taking values in a space $\mathcal{Z}$, and let $\psi : \mathcal{Z} \to \mathcal{H}^d$ be a measurable function. Define the random Hermitian matrices

$$X = \psi(Z) \text{ and } X' = \psi(Z'). \tag{3.6}$$

We say that (X, X') is a matrix Stein pair if there is a constant $\beta \in (0, 1]$ for which $\mathbb{E}[X - X'|Z] = \beta X$ *a.s.*.

Note it follows from the above definition that $\mathbb{E}[X] = 0$. Also, β is called the *scale factor* of the pair (X, X').

Lastly, we present the concept of dilations, which are used to derive our results. A symmetric dilation of a real-valued rectangular matrix B is

$$\mathcal{D}(B) = \begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix} \tag{3.7}$$

Note that $\mathcal{D}(B)$ is always symmetric, and it satisfies the following useful property:

$$\mathcal{D}(B)^2 = \begin{bmatrix} 0 & BB^\top \\ B^\top B & 0 \end{bmatrix} \tag{3.8}$$

Moreover, observe that the norm of the symmetric dilation has a useful relationship with the norm of the original matrix $\lambda_{\max}(\mathcal{D}(B)) = \|\mathcal{D}(B)\| = \|B\|$. We will construct bounds for symmetric matrices and then we will extend those bounds to non-symmetric matrices by using dilations.

Matrix Concentration Inequalities

In order for us to develop finite sample tests we require matrix concentration inequalities. The random matrices involved in this paper are not independent in the general case. We first present a version of matrix Hoeffding inequality for conditionally independent sums of random matrices, that is random matrices that become independent after conditioning on another matrix. This theorem, and the following theorems about concentrations, were first introduced by [159], as generalizations of the (respective) independent cases.

Proposition 3.1.2. *[159] Consider a finite sequence $(Y_k)_{(k\geq 1)}$ of random matrices in $\mathcal{H}^d$ that are conditionally independent given an auxiliary random matrix Z and finite sequences $(P_k)_{k\geq 1}$ and $(Q_k)_{k\geq 1}$ of deterministic matrices in $\mathcal{H}^d$. Assume that*

$$\mathbb{E}[Y_k|Z] = 0,\ Y_k^2 \preceq P_k^2,\ \mathbb{E}[Y_k^2|(Y_j)_{j\neq k}] \preceq Q_k^2 \ a.s. \forall k, \tag{3.9}$$

then for all $t \geq 0$ we have

$$\mathrm{P}\left(\lambda_{\max}\left(\sum_{k=0} Y_k\right) \geq t\right) \leq d \cdot e^{-t^2/2\sigma^2} \tag{3.10}$$

where $\sigma^2 = \frac{1}{2}\|\sum_k P_k^2 + Q_k^2\|$.

Next we present a version of the McDiarmid inequality for self-reproducing random matrices.

Proposition 3.1.3. *[159] Let $z = (Z_1, ..., Z_n)$ be a random vector taking values in a space $\mathcal{Z}$, and, for each index k, let Z_k' and Z_k be conditionally i.i.d. given $(Z_j)_{j\neq k}$. Suppose that $H : \mathcal{Z} \to \mathcal{H}^d$ is a function that satisfies the self-reproducing property*

$$\sum_{k=1}^{n}(H(z) - \mathbb{E}[H(z)|(Z_j)_{j\neq k}]) = s \cdot (H(z) - \mathbb{E}[H(z)])\ a.s. \tag{3.11}$$

for a parameter $s > 0$, as well as the bounded difference property

$$\mathbb{E}\left[(H(z) - H(Z_1, ..., Z_k', ..., Z_n))^2|z\right] \preceq P_k^2 \tag{3.12}$$

for each index k a.s., where P_k is a deterministic matrix in $\mathcal{H}^d$. Then, for all $t \geq 0$,

$$\mathrm{P}(\lambda_{\max}(H(z) - \mathbb{E}[H(z)]) \geq t) \leq d \cdot e^{-st^2/L} \tag{3.13}$$

for $L = \|\sum_{k=1}^n P_k^2\|$.

Now we provide an essential property that is called symmetrization, which is a generalization for summation of the symmetrization property presented in [159] for a single matrix:

Lemma 3.1.4. *Let $\{X_i\}_{i=1}^n$ be a sequence of random Hermitian matrices with $\mathbb{E}[X_i] = 0$. Then*

$$\mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n X_i}\right)\right] \leq \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{2\sum_{i=1}^n \epsilon_i X_i}\right)\right] \tag{3.14}$$

where $\{\epsilon_i\}_{i=1}^n$ are i.i.d. Radamacher random variables.

Proof. First, we construct a sequence of copies $\{X_i'\}_{i=1}^n$ independent from $\{X_i\}_{i=1}^n$, and let $\mathbb{E}'$ denote the expectation with respect to $\{X_i'\}_{i=1}^n$. So we have

$$\mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n X_i}\right)\right] = \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n X_i - \mathbb{E}'[X_i']}\right)\right] \leq \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n X_i - X_i'}\right)\right] = \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n \epsilon_i (X_i - X_i')}\right)\right] \tag{3.15}$$

where we have sequentially used Jensen's inequality and then the symmetry of $(X_i - X_i')$. Now we finish the proof by noting

$$\mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n X_i}\right)\right] \leq \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n \epsilon_i (X_i - X_i')}\right)\right] \leq \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{\sum_{i=1}^n \epsilon_i X_i} \mathrm{e}^{-\sum_{i=1}^n \epsilon_i X_i'}\right)\right] \leq \tag{3.16}$$

$$\mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{2\sum_{i=1}^n \epsilon_i X_i}\right)^{1/2} \operatorname{tr}\left(\mathrm{e}^{-2\sum_{i=1}^n \epsilon_i X_i'}\right)^{1/2}\right] = \mathbb{E}\left[\operatorname{tr}\left(\mathrm{e}^{2\sum_{i=1}^n \epsilon_i X_i}\right)\right] \tag{3.17}$$

where we have sequentially used the Golden-Thompson inequality, the Cauchy-Schwartz inequality two times, and the fact that both factors are identically distributed. (See [42] for the definition of those properties.) □

LTI System with Switching

We consider a MIMO LTI system that allows the controller to switch between two sets of sensors, and we will assume that both the measurement and process noise have stochastic distributions with a bounded support. Namely, we will assume that the noise vectors have bounded norm almost surely.

Consider a MIMO LTI system with partial observations and switching in the sensing

$$\begin{aligned} x_{n+1} &= Ax_n + Bu_n + w_n \\ y_n &= C(\alpha_n)x_n + z_n(\alpha_n) + \alpha_n v_n \end{aligned} \tag{3.18}$$

where $x \in \mathbb{R}^p$, $u \in \mathbb{R}^q$, $y, z, v \in \mathbb{R}^m$, and $\alpha_n \in \{0, 1\}$. The w_n represents zero mean i.i.d. process noise with covariance Σ_W. Moreover, we have

$$\begin{aligned} C_n = C(\alpha_n) &= \alpha_n C_1 + (1 - \alpha_n) C_2 \\ z_n(\alpha_n) &= \alpha_n \zeta_n + (1 - \alpha_n)\eta_n \end{aligned} \tag{3.19}$$

where ζ_n and η_n represent zero mean i.i.d. measurement noise with covariance matrices $\Sigma_\zeta \preceq \Sigma_\eta$, respectively. Note that $\alpha_n \in \{0, 1\}$ should be interpreted as the switching control action that selects between the observability matrices C_1 or C_2. The v_n is as an additive

measurement disturbance added by an attacker, which can only affect the observations made when the mode $\alpha = 1$ is selected. The idea of this model is that C_1 corresponds to a more accurate set of sensors than C_2, but conversely that some subset of sensors within C_1 are susceptible to an attack whereas the set of all sensors within C_2 are *not* susceptible to an attack. (Our results also apply when the sensors within C_2 are a strict subset of the sensors in C_1, with the only change being that $\Sigma_\eta \preceq \Sigma_\zeta$.)

We further assume the process noise is independent of the measurement noise, that is w_n for $n \geq 0$ is independent of ζ_n, η_n for $n \geq 0$. Lastly we assume both measurement and disturbance noises are bounded in magnitude. Namely, we assume that both measurement noise and systems disturbances are given by i.i.d. bounded random vectors: $\|w_k\| \leq K_w$ and $\|z_k\| \leq K_z, \forall k \geq 0$.

If (A, B) is stabilizable and both (A, C_1) and (A, C_2) are detectable, then an output-feedback controller can be designed when $v_n \equiv 0$ using an observer and the separation principle; the stability of this scheme is proved in Proposition 3.1.5. Let K be a constant state-feedback gain matrix such that $A + BK$ is Schur stable, and let L_i be a constant observer gain matrix such that $A + L_i C_i$ is Schur stable for $i \in \{1, 2\}$. The idea of dynamic watermarking in this context will be to superimpose a private (and random) excitation signal e_n known in value to the controller but unknown in value to the attacker. As a result, we will apply the control input $u_n = Kx'_n + e_n$, where x'_n is the observer-estimated state and e_n are i.i.d. random vectors on a bounded support, such that $\|e_k\| \leq K_e, \forall k \geq 0$, with zero mean and constant variance Σ_E fixed by the controller. Let

$$\begin{aligned} L(\alpha) &= \alpha L_1 + (1-\alpha) L_2 \\ L_n &= L(\alpha_n) \\ \underline{L}(\alpha)^\top &= \begin{bmatrix} 0 & -L(\alpha)^\top \end{bmatrix} \end{aligned} \tag{3.20}$$

Moreover, let $\tilde{x}^\top = \begin{bmatrix} x^\top & x'^\top \end{bmatrix}$, and define:

$$\begin{aligned} \underline{B}^\top &= \begin{bmatrix} B^\top & B^\top \end{bmatrix}, \underline{D}^\top = \begin{bmatrix} \mathbb{I} & 0 \end{bmatrix}, \text{ and} \\ \underline{A}(\alpha) &= \begin{bmatrix} A & BK \\ -L(\alpha)C(\alpha) & A + BK + L(\alpha)C(\alpha) \end{bmatrix}. \end{aligned} \tag{3.21}$$

Then the closed-loop system with private excitation is given by:

$$\tilde{x}_{n+1} = \underline{A}(\alpha_n)\tilde{x}_n + \underline{B}e_n + \underline{D}w_n + \underline{L}(\alpha_n)(z_n(\alpha_n) + \alpha_n v_n). \tag{3.22}$$

If we define the observation error $\delta' = x' - x$, then with the change of variables $\check{x}^\top = \begin{bmatrix} x^\top & \delta'^\top \end{bmatrix}$ we have the dynamics

$$\check{x}_{n+1} = \underline{\underline{A}}(\alpha_n)\check{x}_n + \underline{\underline{B}}e_n + \underline{\underline{D}}w_n + \underline{\underline{L}}(\alpha_n)(z_n(\alpha) + \alpha v_n) \tag{3.23}$$

where we further define the following matrices

$$\begin{aligned} \underline{\underline{B}}^\top &= \begin{bmatrix} B^\top & 0 \end{bmatrix}, \underline{\underline{D}}^\top = \begin{bmatrix} \mathbb{I} & -\mathbb{I} \end{bmatrix}, \underline{\underline{L}}(\alpha) = \underline{L}(\alpha), \\ \text{and } \underline{\underline{A}}(\alpha) &= \begin{bmatrix} A + BK & BK \\ 0 & A + L(\alpha)C(\alpha) \end{bmatrix}. \end{aligned} \tag{3.24}$$

Recall that $\underline{\underline{A}}(\alpha)$ is Schur stable whenever $A+BK$ and $A+L(\alpha)C(\alpha)$ are both Schur stable.

We present an schematic representation of the LTI system with switching in figure 1, highlighting the presence of the switching decision and the attack presence which affects only the sensor with low variance.

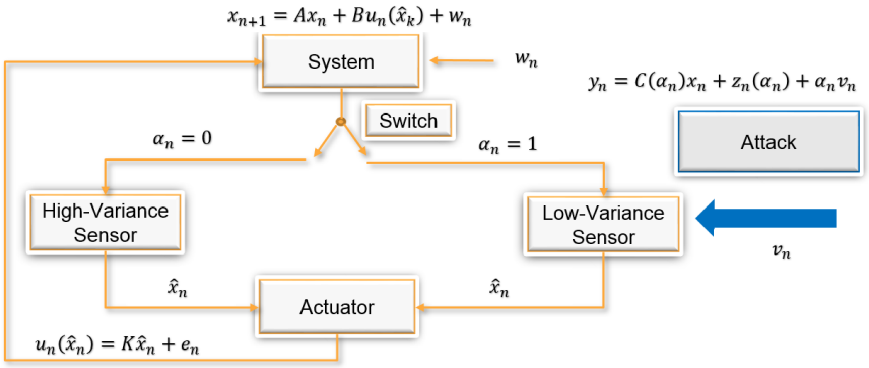

Figure 3.1: Schematic representation of the LTI system with switching: At every stage n, the controller can opt between using the high-variance sensor ($\alpha_n = 0$) or using the low-variance sensor ($\alpha_n = 1$). If they opt for the low-variance sensor, then there is a possibility than an attacker will affect the sensor measurement by adding an additive disturbance vector v_n. Afterwards, a system state estimate $\hat{x}_n$ is formed and used by the actuator to compute the control input u_n.

There is one technical point that needs to be addressed before proceeding: Since there is switching between observers, the closed-loop system will not necessarily be stable even though $A + BK$ and $A + L(\alpha)C(\alpha)$ are both Schur stable. One approach to resolving this issue is limiting the rate of switching. The intuition is that if we wait a certain number of time periods before performing a switch, that will be enough to make sure the "energy" of the system decreases. We make this state formal in Proposition 3.1.5.

Proposition 3.1.5. *Let P be the positive definite solution of the Lyapunov equation*

$$\underline{\underline{A}}(1)^\top P \underline{\underline{A}}(1) - P = -\mathbb{I}, \tag{3.25}$$

where $\mathbb{I}$ is the identity matrix. Then there exists a smallest positive integer τ such that

$$(\underline{\underline{A}}(0)^t)^\top P \underline{\underline{A}}(0)^t - P \preceq -\mathbb{I}, \text{ for all } t \geq \tau. \tag{3.26}$$

And the closed-loop system with no process noise (i.e. $w_k = 0, \forall k \geq 0$) is asymptotically stable under switching policies where: whenever we switch from $\alpha = 1$ to $\alpha = 0$ we maintain $\alpha = 0$ for at least τ time steps before any possible switching occurs to $\alpha = 1$.

Proof. A proof for stability of general discrete-time nonlinear and switched systems has been given in [8] using non-monotonic Lyapunov Functions. We provide a different proof more tailored to our problem.

We first shows that τ exists. Since the matrix $\underline{\underline{A}}(0)$ is Schur stable it follows that for any $\check{x} \in \mathbb{R}^{2p}$, $\underline{\underline{A}}(0)^k \check{x} \to 0$ as $k \to \infty$. Hence it follows that $\check{x}^\top (\underline{\underline{A}}(0)^k)^\top P \underline{\underline{A}}(0)^k \check{x} \to 0$ as $k \to \infty$, by the continuity of the function $f(y) = y^\top P y$. Now let $\bar{\lambda}$ be the smallest eigenvalue of $\underline{\underline{A}}(1)^\top P \underline{\underline{A}}(1)$ and λ_k be the largest eigenvalue of $(\underline{\underline{A}}(0)^k)^\top P \underline{\underline{A}}(0)^k$. We observe that $\lambda_k \to 0$ as $k \to \infty$, again by Schur stability of $\underline{\underline{A}}(0)$. Then let $\tau \geq 0$ be the scalar where $\lambda_t \leq \bar{\lambda}$, for all $t \geq \tau$. For this choice of τ the following inequality holds:

$$(\underline{\underline{A}}(0)^t)^\top P \underline{\underline{A}}(0)^t \preceq \underline{\underline{A}}(1)^\top P \underline{\underline{A}}(1), \text{ for all } t \geq \tau. \tag{3.27}$$

Then using (24) we obtain

$$(\underline{\underline{A}}(0)^t)^\top P \underline{\underline{A}}(0)^t - P \preceq -\mathbb{I}, \text{ for all } t \geq \tau, \tag{3.28}$$

which is exactly (25).

Now let P and τ be as defined in the hypothesis. Consider the function $V(\check{x}) = \check{x}^\top P \check{x}$. We first analyze the behavior of $V(\cdot)$ for a given pairs of states, given different switching decisions. If the switching decision $\alpha_k = 1$, then we can write

$$V(\check{x}_{k+1}) = \check{x}_{k+1}^\top P \check{x}_{k+1} = \check{x}_k^\top \underline{\underline{A}}(1)^\top P \underline{\underline{A}}(1) \check{x}_k. \tag{3.29}$$

Using (1), for any $\check{x}_k \neq 0$, we have

$$V(\check{x}_{k+1}) - V(\check{x}_k) = \tag{3.30}$$

$$\check{x}_k^\top \underline{\underline{A}}(1)^\top P \underline{\underline{A}}(1) \check{x}_k - \check{x}_k^\top P \check{x}_k = -||\check{x}_k||_2^2 < 0 \tag{3.31}$$

So for the decision $\alpha_k = 1$, the function $V(\check{x})$ decreases after one time step. Now let's suppose $\alpha_k = 0$, and we do not switch for $t \geq \tau$ time periods (that is $\alpha_j = 0$ for $j \in \{k, ..., k+t\}$). Then we can write

$$V(\check{x}_{k+t}) = \check{x}_{k+t}^\top P \check{x}_{k+t} = \check{x}_k^\top (\underline{\underline{A}}(0)^\tau)^\top P (\underline{\underline{A}}(0)^\tau) \check{x}_k, \tag{3.32}$$

and using (25), for any $\check{x}_k \neq 0$ we have

$$V(\check{x}_{k+t}) - V(\check{x}_k) = \tag{3.33}$$

$$\check{x}_k^\top (\underline{\underline{A}}(0)^t)^\top P (\underline{\underline{A}}(0)^t) \check{x}_k - \check{x}_k^\top P \check{x}_k \leq -||\check{x}_k||_2^2 < 0 \tag{3.34}$$

So under this switching decision rule the function $V(\check{x})$ decreases after t steps.

Now consider the sequence $\{V(\check{x}_k)\}_{k=0}^\infty$, under arbitrary switching generated by a policy where whenever we switch from $\alpha = 1$ to $\alpha = 0$ we maintain $\alpha = 0$ for at least τ time steps before any possible switching occurs to $\alpha = 1$. Consider subsequence $\{V_1, V_2, ...\} = \{V_j \text{ for } j \in J\}$, where

$$J = \begin{cases} \{k : \alpha_{k-1} = 1, \alpha_k = 0\} \bigcup \\ \{k : \alpha_{k-1} = 1, \alpha_k = 1\} \bigcup \\ \{k : \alpha_{k-1} = 0, \alpha_k = 1\} \end{cases} \tag{3.35}$$

We observe that this subsequence is decreasing. To see this note that

$$V_{j+1} - V_j < 0 \tag{3.36}$$

holds for the above five possible cases:

$$(j,(j+1)) : \begin{cases} \text{if } j \text{ and } (j+1) \text{ both belong to} \\ \quad \{k : \alpha_{k-1} = 1, \alpha_k = 1\}, \\ \text{if } j \in \{k : \alpha_{k-1} = 1, \alpha_k = 1\} \text{ and} \\ \quad (j+1) \in \{k : \alpha_{k-1} = 1, \alpha_k = 0\}, \\ \text{if } j \in \{k : \alpha_{k-1} = 0, \alpha_k = 1\} \text{ and} \\ \quad (j+1) \in \{k : \alpha_{k-1} = 1, \alpha_k = 0\}, \\ \text{if } j \in \{k : \alpha_{k-1} = 0, \alpha_k = 1\} \text{ and} \\ \quad (j+1) \in \{k : \alpha_{k-1} = 1, \alpha_k = 1\} \\ \text{if } j \in \{k : \alpha_{k-1} = 1, \alpha_k = 0\} \text{ and} \\ \quad (j+1) \in \{k : \alpha_{k-1} = 0, \alpha_k = 1\} \end{cases} \tag{3.37}$$

where the first four cases hold due to (24) and the last case holds due to (25). That covers all possible cases of pairs of elements in the subsequence. Hence the subsequence is decreasing. We observe that if J is a finite set, that means that there exists an index $\bar{k}$ such that $\alpha_k = 0$ for $k \geq \bar{k}$, then in this case the whole sequence $\{V(\check{x}_k)\}$ will converge to zero, since $\underline{\underline{A}}(0)$ is Schur stable. Therefore we will focus on the case that the set J is infinite.

Now, since each $V(\check{x}) > 0$ for $\|\check{x}\|_2 > 0$, then this subsequence converge to some constant $c \geq 0$. By continuity of $V(\check{x})$, it follows that $c = 0$. We show this by contradiction: Suppose that $c \neq 0$ and consider any $\Delta = \{\check{x} \in \mathbb{R}^{2p} : d \leq ||\check{x}||_2 \leq r\}$, with, $0 < d < r$ such that $\{\check{x} \in \mathbb{R}^{2p} : V(\check{x}) = c\} \subseteq \Delta$. Such d and r exist because $V(\check{x}) = \check{x}^\top P \check{x}$ is a strictly convex function, for positive definite matrix P. And we note that we can pick an r such that every state trajectory of the subsequence lies in the the ball $B_r = \{\check{x} \in \mathbb{R}^{2p} : \|\check{x}\|_2 < r\}$ The set Δ is a compact set and contains $\{\check{x} \in \mathbb{R}^{2p} : V(\check{x}) = c\}$. Now consider the following

$$\gamma^0 = \min_{\check{x} \in \Delta} \sup_{t \geq \tau} V(\check{x}) - V(\underline{\underline{A}}(0)^t \check{x}) \tag{3.38}$$

$$\gamma^1 = \min_{\check{x} \in \Delta} V(\check{x}) - V(\underline{\underline{A}}(1)\check{x}) \tag{3.39}$$

where both γ^1 and γ^0 exists, again by positive definiteness of matrix P and $V(\check{x}) = \check{x}^\top P \check{x}$. Now since $\lim_{j \to \infty} V_j = c$ and V is continuous then there exists an index $\bar{j}$ such that for all $j > \bar{j}$ and indices $q(j)$ such that $V_j = V(\check{x}_{q(j)}) \leq c + \gamma$, with $\gamma < \min\{\gamma^0, \gamma^1\}$ and $\check{x}_{q(j)} \in \Delta$. But then it must hold that

$$V(\check{x}_{q(j)}) - V(\underline{\underline{A}}(0)^t \check{x}_{q(j)}) \geq \gamma^0, \text{ for all } t \geq \tau \tag{3.40}$$

$$V(\check{x}_{q(j)}) - V(\underline{\underline{A}}(1)\check{x}_{q(j)}) \geq \gamma^1 \tag{3.41}$$

But then it follows that $V(\underline{\underline{A}}(0)^t \check{x}_{q(j)}) \leq c + \gamma - \gamma^0 < 0$ and $V(\underline{\underline{A}}(1)\check{x}_{q(j)}) \leq c + \gamma - \gamma^1 < 0$. This implies that the next point in the series after $\bar{j}$ will bring the function value of V strictly below c, which is a contradiction. Hence we conclude that $V_j \to 0$.

Now observe that the set of indices $\{k : \alpha_{k-1} = 0, \alpha_k = 0\}$ is the only remaining set not included in (36). We proceed to show that those elements also converge to zero, hence showing that the entire sequence $\{V(\check{x}_k)\}$ converge to zero. Since we found a convergent subsequence of V_j for $j \in J$, there exists an index j and an index $q(j)$ and a scalar $\epsilon > 0$ such that:

$$V_j = V(\check{x}_{q(j)}) \leq \epsilon \tag{3.42}$$

Now for every $l = 1, ..., \tau$, we define $\delta_l(\epsilon)$:

$$\delta_l(\epsilon) = \max_{\check{x}^\top P \check{x} \leq \epsilon} \left\{ \check{x}^\top (A(0)^l)^\top P (A(0)^l) \check{x} \right\}. \tag{3.43}$$

Observe that $\delta_l(\epsilon) \to 0$ as $\epsilon \to 0$. Moreover it follows that for all indices $k \in \{k : \alpha_{k-1} = 0, \alpha_k = 0\}$ that come after index $q(j)$, it holds that

$$V(\check{x}_k) \leq \max_{l \in \{1,...,\tau\}} \left\{ \delta_l(\epsilon) \right\} \text{ for } k \in \{k : \alpha_{k-1} = 0, \alpha_k = 0, k \geq q(j)\}. \tag{3.44}$$

Then it follows that those elements will go to zero as ϵ goes to zero. Hence it follows that

$$V(\check{x}_k) \leq \max \left\{ \delta_1(\epsilon), ..., \delta_\tau(\epsilon), \epsilon \right\} \text{ for } k \geq q(j) \tag{3.45}$$

Then elements of the sequence $\{V(\check{x}_k)\}$ will go to zero as $\epsilon \to 0$. Hence the function $V(\cdot)$ is a proper Lyapunov function for the system, as established in [117], under the desired switching rule, which implies that the closed-loop system is asymptotically stable. □

Lastly, we note that such a τ exists because $\underline{\underline{A}}(0)$ is Schur stable.

Matrix Inequalities for General LTI Systems

We will now apply the abstract concentration inequalities presented in Sect. 3.1 to our LTI setting with switching. We will begin our analysis considering that the system is under no attack. Under no attack we would like to keep using the most accurate sensor – that is keeping our switching control $\alpha_n \equiv 1$ for all $n \geq 0$. However, as it is usually observed for any kind of tests based on random quantities, we are susceptible to commit what is commonly known as false positive or type I errors. Hence our goal is to provide finite sample tests based on matrix concentration of measure such that type I errors happen only a finite number of times throughout the evolution of the system. This would imply that those tests report correctly that there is no attack infinitely often. To that end, we will utilize two observers: The first observer obtain system measurements from the switched system, using $C(\alpha_n)$; The second observer never switches and keeps measuring the system using the vulnerable sensor, using C_1. The finite-time statistical tests and the concentration inequalities analysis

presented in this section are referring to quantities associated with the second observer. For ease of notation and presentation we drop the subscript of the analysis define $C = C_1$ and $L = L_1$. Moreover, for the second observer we define: $\hat{x}_n$ and δ_n to denote the estimate state and observation error:

$$\hat{x}_{n+1} = (A + BK)\hat{x}_n + LC(\hat{x}_n - x_n) + Be_n - Lz_n \tag{3.46}$$

and $\delta_n = \hat{x}_n - x_n$. Then by the same type of variable substitution, using 3.23:

$$\delta_{n+1} = (A + LC)\delta_n - w_n + -Lz_n \tag{3.47}$$

We will start by bounding the vector $C\delta_n - z_n$:

Theorem 3.1.6. *Let $\delta_n = \hat{x}_n - x_n$. Assume that both measurement noise and systems disturbances are given by i.i.d. bounded random vectors: $\|w_k\| \leq K_w$ and $\|z_k\| \leq K_z, \forall k \geq 0$. Then when $v_n \equiv 0$ for all $n \geq 0$ we have*

$$\|C\delta_n - z_n\| \leq \bar{K}_n \tag{3.48}$$

where $\bar{K}_n = K_z + \sum_{k=0}^{n-1} \left\|C\bar{D}_k\right\| K_w + \left\|C\bar{L}_k\right\| K_z$ and

$$(C\delta_n - z_n)(C\delta_n - z_n)^\top \preceq \bar{K}_n^2 \mathbb{I}. \tag{3.49}$$

Moreover, it follows that

$$\mathbb{E}[(C\delta_n - z_n)(C\delta_n - z_n)^\top] = C\left(\sum_{k=0}^{n-1} \bar{D}_k \Sigma_w \bar{D}_k^\top + \bar{L}_k \Sigma_z \bar{L}_k^\top\right) C^\top + \Sigma_z \tag{3.50}$$

where

$$\begin{aligned} \bar{D}_k &= -(A + LC)^{n-1-k} \\ \bar{L}_k &= -(A + LC)^{n-1-k} L^\top. \end{aligned} \tag{3.51}$$

Proof. Recall our definition of δ_n (3.47) we can write

$$\delta_n = (A + LC)^n \delta_0 - \sum_{k=0}^{n-1} (A + LC)^{n-1-k} (\mathbb{I} w_k + L^\top z_k). \tag{3.52}$$

Assuming $\delta_0 = 0$, we have that

$$\delta_n = \sum_{k=0}^{n-1} \bar{D}_k w_k + \bar{L}_k z_k. \tag{3.53}$$

Now, we can define the following:

$$C\delta_n \delta_n^\top C^\top = \left(\sum_{k=0}^{n-1} \bar{D}_k w_k + \bar{L}_k z_k\right) \left(\sum_{k=0}^{n-1} \bar{D}_k w_k + \bar{L}_k z_k\right)^\top C^\top, \tag{3.54}$$

and obtain the expectation directly:

$$\mathbb{E}[(C\delta_n - z_n)(C\delta_n - z_n)^\top] = C\left(\sum_{k=0}^{n-1} \bar{D}_k \Sigma_w \bar{D}_k^\top + \bar{L}_k \Sigma_z \bar{L}_k^\top\right) C^\top + \Sigma_z, \tag{3.55}$$

since z_n and δ_n are independent for all n. Moreover, both system disturbances and measurement noise are independent. Under our *key* assumption that both measurement noise and systems disturbances are given by i.i.d. bounded random vectors we have that

$$\|\delta_n\| \leq \sum_{k=0}^{n-1} \left\|\bar{D}_k\right\| K_w + \left\|\bar{L}_k\right\| K_z, \tag{3.56}$$

and that

$$\|C\delta_n - z_n\| \leq K_z + \sum_{k=0}^{n-1} \left\|C\bar{D}_k\right\| K_w + \left\|C\bar{L}_k\right\| K_z = \bar{K}_n. \tag{3.57}$$

So we have $(C\delta_n - z_n)(C\delta_n - z_n)^\top \preceq \bar{K}_n^2 \mathbb{I}$. □

Now consider the matrix (3.2) that was used in the introduction to define the asymptotic tests. But now, instead of letting n go to infinity, we keep it finite and then analyze the finite summation of matrices. Let $k' = \min\{k \geq 0 \mid C(A+BK)^k B \neq 0\}$. The existence of such k' is guaranteed (see [130]). Moreover, define

$$\psi_n^\top = \begin{bmatrix}(C\hat{x}_n - y_n)^\top & e_{n-k'-1}^\top\end{bmatrix}. \tag{3.58}$$

Then we have:

$$\frac{1}{N}\sum_{n=0}^{N-1} \psi_n \psi_n^\top = \frac{1}{N}\begin{bmatrix}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top & \sum_{n=0}^{N-1}(C\hat{x}_n - y_n)e_{n-k'-1}^\top \\ \sum_{n=0}^{N-1} e_{n-k'-1}(C\hat{x}_n - y_n)^\top & \sum_{n=0}^{N-1} e_{n-k'-1}e_{n-k'-1}^\top\end{bmatrix} \tag{3.59}$$

It suits our purposes to make sure that the above matrix is centered (that is have zero expected value). In order to achieve this, we construct the matrix

$$\frac{1}{N}\sum_{n=0}^{N-1} \Psi_n = \frac{1}{N}\sum_{n=0}^{N-1} \psi_n \psi_n^\top - \frac{1}{N}\begin{bmatrix}\sum_{n=0}^{N-1}\mathbb{E}[(C\delta_n - z_n)(C\delta_n - z_n)^\top] & 0 \\ 0 & N\Sigma_e\end{bmatrix} \tag{3.60}$$

Note that it follows that: $\mathbb{E}[\Psi_n] = 0, \forall n \geq 0$, since $C\hat{x}_n - y_n = C\delta_n - z_n$. We wish to control the singular values of the above matrix. We will do so by analyzing each individual block. To ease the notation we define

$$\Phi_N = \frac{1}{N}\sum_{n=0}^{N-1} \Psi_n \tag{3.61}$$

and we define each submatrix

$$\Phi_N^{(1)} = \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top - \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top] \tag{3.62}$$

$$\Phi_N^{(2)} = \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)e_{n-k'-1}^\top \tag{3.63}$$

$$\Phi_N^{(3)} = \frac{1}{N}\sum_{n=0}^{N-1}(e_{n-k'-1}e_{n-k'-1}^\top - \Sigma_e) \tag{3.64}$$

such that

$$\Phi_N = \begin{bmatrix} \Phi_N^{(1)} & \Phi_N^{(2)} \\ (\Phi_N^{(2)})^\top & \Phi_N^{(3)} \end{bmatrix}. \tag{3.65}$$

Our next step is to bound the norm of $\Phi_N^{(1)}$.

Theorem 3.1.7. *If $v_n \equiv 0$ for all $n \geq 0$, then the following concentration inequality holds for all $N \geq 1$ and all t:*

$$\mathrm{P}\left(\left\|\Phi_N^{(1)}\right\| \geq t \right) \leq m \cdot e^{-N^2t^2/c_N^{(1)}} \tag{3.66}$$

where $c_N^{(1)} = 8\left\|\sum_{k=0}^{N-1}\left(\bar{K}_k^4\mathbb{I}\right)\right\|$.

Proof. We start by defining the matrix Y_n as

$$Y_n = (C\delta_n - z_n)(C\delta_n - z_n)^\top - \mathbb{E}[(C\delta_n - z_n)(C\delta_n - z_n)^\top]. \tag{3.67}$$

Now define a vector of independent i.i.d. Radamacher random variables $\{\epsilon_n\}_{n=0}^{N-1}$. Now we define a filtration $Z = (Y_n)_{n\geq 1}$ where $W_n = \epsilon_n Y_n, n \geq 1$. Then we see that each summand W_n is conditionally independent given Z, because the Radamacher random variables are all i.i.d. This allows us to use the Hoeffding Bound for conditionally independent sums to obtain

$$\mathrm{P}\left(\left\| \frac{1}{N}\sum_{n=0}^{N-1} Y_n \right\| \geq t \right) \leq d \cdot e^{-N^2t^2/8\sigma^2} \tag{3.68}$$

for $\sigma^2 = \left\|\sum_{k=0}^{N-1}\left(\bar{K}_k^4\mathbb{I}\right)\right\|$. The inequality follows from applying the Laplace transform method and using the symmetrization property (Lemma 1):

$$\mathbb{E}\left[\mathrm{tr}\left(\mathrm{e}^{\frac{1}{N}\sum_{n=0}^{N} Y_n}\right)\right] \leq \mathbb{E}\left[\mathrm{tr}\left(\mathrm{e}^{\frac{2}{N}\sum_{n=0}^{N-1}\epsilon_i Y_n}\right)\right] \tag{3.69}$$

In addition, we have used the fact $W_k^2 \preceq \bar{K}_k^4\mathbb{I}$ for all k, and the fact that

$$\frac{1}{2}\left\| \sum_k \bar{K}_k^4\mathbb{I} + E[W_k^2|(W_j)_{j\neq k}] \right\| \leq \left\| \sum_k \left(\bar{K}_k^4\mathbb{I}\right) \right\| \tag{3.70}$$

since $\mathbb{E}[W_k^2|(W_j)_{j\neq k}] = \mathbb{E}[Y_k^2|(W_j)_{j\neq k}] \preceq \bar{K}_k^4 \mathbb{I}$. □

Next, we provide a bound on the norm of $\Phi_N^{(2)}$. But before that we need the following proposition:

Proposition 3.1.8. *Let $e = (e_1, ..., e_k, ..., e_n)$ be a sequence of random vectors taking values in a space $\mathcal{Z}$. Now construct an exchangeable pair $e' = (e_1, ..., e'_k, ..., e_n)$ where e_k and e'_k are conditionally i.i.d. given $(e_j)_{j\neq k}$ and k is an independent coordinate drawn uniformly from $\{1, ..., n\}$. We define*

$$H(e) = \begin{bmatrix} 0 & \sum_{n=0}^{N-1}(d_n)e_{n-k'-1}^\top \\ \sum_{n=0}^{N-1} e_{n-k'-1}(d_n)^\top & 0 \end{bmatrix} \tag{3.71}$$

where $d_n = (C\delta_n - z_n)$. If $v_n \equiv 0$ for all $n \geq 0$, then the function $H(e)$ satisfies the bounded differences property

$$\mathbb{E}[(H(e) - H(e'))^2|e] \preceq \bar{P}_n^2 \tag{3.72}$$

for $\bar{P}_n^2 = \max\{P_n^2, P_n'^2\}\mathbb{I}$ with positive constants $P_n^2, P_n'^2$:

$$P_n'^2 = \bar{K}_n^2(K_e^2 + \|\Sigma_E\|) \tag{3.73}$$

$$P_n^2 = \left(K_e^2 + \mathrm{tr}(\Sigma_E)\right) \times \left\| C\left(\sum_{k=1}^{n-1} \bar{D}_k\Sigma_w\bar{D}_k^\top + \bar{L}_k\Sigma_z\bar{L}_k^\top\right)C^\top + \Sigma_z \right\| \tag{3.74}$$

Proof. Let $q_n = d_n e_{n-k'-1}^\top - d_n e_{n-k'-1}^{'\top}$ and observe that

$$\mathbb{E}[(H(e) - H(e'))^2|e] = \mathbb{E}\left[\begin{bmatrix} 0 & q_n \\ q_n^\top & 0 \end{bmatrix}^2 |e \right] = \mathbb{E}\left[\begin{bmatrix} Q_n & 0 \\ 0 & Q'_n \end{bmatrix} |e \right] \tag{3.75}$$

where we have defined

$$\begin{aligned} Q_n &= d_n e_{n-k'-1}^\top e_{n-k'-1} d_n^\top + d_n e_{n-k'-1}^{'\top} e_{n-k'-1}^{'} d_n^\top \\ Q'_n &= e_{n-k'-1} d_n^\top d_n e_{n-k'-1}^\top + e_{n-k'-1}^{'} d_n^\top d_n e_{n-k'-1}^{'\top} \end{aligned} \tag{3.76}$$

Now we have

$$\mathbb{E}[Q_n|e] = \mathbb{E}[d_n e_{n-k'-1}^\top e_{n-k'-1} d_n^\top + d_n e_{n-k'-1}^{'\top} e_{n-k'-1}^{'} d_n^\top|e] = \tag{3.77}$$

$$(e_{n-k'-1}^\top e_{n-k'-1})\mathbb{E}[d_n d_n^\top|e] + \mathbb{E}[e_{n-k'-1}^{'\top} e_{n-k'-1}^{'}|e]\mathbb{E}[d_n d_n^\top|e] \tag{3.78}$$

Recalling that $\|e_k\| \leq K_e \ \forall k \geq 0$ and (3.55), it follows that

$$\|\mathbb{E}[Q_n|e]\| \leq \left(K_e^2 + \mathrm{tr}(\Sigma_E)\right) \times \left\| C\left(\sum_{k=1}^{n-1} \bar{D}_k\Sigma_w\bar{D}_k^\top + \bar{L}_k\Sigma_z\bar{L}_k^\top\right)C^\top + \Sigma_z \right\| = P_n^2. \tag{3.79}$$

Moreover, it follows that

$$\|\mathbb{E}[Q'_n|e]\| = e_{n-k'-1}d_n^\top d_n e_{n-k'-1}^\top + e'_{n-k'-1}d_n^\top d_n e_{n-k'-1}^{'\top} = (\mathbb{E}[(d_n^\top d_n)|e])e_{n-k'-1}e_{n-k'-1}^\top + (\mathbb{E}[d_n^\top d_n|e])\mathbb{E}[e'_{n-k'-1}e_{n-k'-1}^{'\top}|e] \tag{3.80}$$

So we get

$$\|\mathbb{E}[Q'_n|e]\| \leq \bar{K}_n^2(K_e^2 + \|\Sigma_E\|) = P_n^{'2} \tag{3.81}$$

Hence it follows that

$$\left\|\mathbb{E}[(H(e) - H(e'))^2|e]\right\| \leq \max\{P_n^2, P_n^{'2}\} \tag{3.82}$$

So it follows that

$$\mathbb{E}[(H(e) - H(e'))^2|e] \preceq \bar{P}_n^2 \tag{3.83}$$

where $\bar{P}_n^2 = \max\{P_n^2, P_n^{'2}\}\mathbb{I}$. □

Now we are ready to provide our theorem.

Theorem 3.1.9. *If $v_n \equiv 0$ for all $n \geq 0$, then the following concentration inequality holds for all $N \geq 1$ and all t:*

$$\mathrm{P}\left(\left\|\Phi_N^{(2)}\right\| \geq t\right) \leq (m+p)\cdot e^{-N^2t^2/c_N^{(2)}} \tag{3.84}$$

where $c_N^{(2)} = \left\|\sum_{n=0}^{N-1}\bar{P}_n^2\right\|$ for $\bar{P}_n^2 = \max\{P_n^2, P_n^{'2}\}\mathbb{I}$, where

$$P_n^2 = \left(K_e^2 + \mathrm{tr}(\Sigma_E)\right) \times \left\|C\left(\sum_{k=1}^{n-1}\bar{D}_k\Sigma_w\bar{D}_k^\top + \bar{L}_k\Sigma_z\bar{L}_k^\top\right)C^\top + \Sigma_z\right\| \tag{3.85}$$

$$P_n^{'2} = \bar{K}_n^2(K_e^2 + \|\Sigma_E\|). \tag{3.86}$$

Proof. We wish to provide bounds on the operator norm of

$$\Phi_N^{(2)} = \frac{1}{N}\sum_{n=0}^{N-1}(C\delta_n - z_n)e_{n-k'-1}^\top \tag{3.87}$$

To achieve that, we will use the concept of matrix Stein pairs as defined previously. Let $E = (e_1, ..., e_k, ..., e_n)$ be a sequence of random vectors taking values in a space $\mathcal{Z}$. Now construct an exchangeable pair $E' = (e_1, ..., e'_k, ..., e_n)$ where e_k and e'_k are conditionally i.i.d. given $(e_j)_{j\neq k}$ and k is an independent coordinate drawn uniformly from $\{1, ..., n\}$. We define $H(e)$ as in Proposition 3.1.8:

$$H(e) = \begin{bmatrix} 0 & \sum_{n=0}^{N-1} b_n e_{n-k'-1}^\top \\ \sum_{n=0}^{N-1} e_{n-k'-1}b_n^\top & 0 \end{bmatrix} \tag{3.88}$$

where $b_n = C\delta_n - z_n$. Since $\mathbb{E}(H(e)) = 0$, this means $H(e)$ satisfies the self-reproducing property

$$\sum_{n=1}^{N} H(e) - \mathbb{E}[H(e)|(e_j)_{j\neq(n-k'-1)}] = H(e) \tag{3.89}$$

for the choice of parameter $s = 1$ (see (3.11) for the definition of s), since for all $n \in \{1, ..., N\}$ we have

$$H(e) - \mathbb{E}[H(e)|(e_j)_{j\neq(n-k'-1)}] = \begin{bmatrix} 0 & (C\delta_n - z_n)e_{n-k'-1}^\top \\ e_{n-k'-1}(C\delta_n - z_n)^\top & 0 \end{bmatrix} \tag{3.90}$$

Next, we use Proposition 3.1.8 to state that $H(e)$ also satisfies the bounded differences property. So we have

$$\mathbb{E}[(H(e) - H(e')^2|e] \preceq \bar{P}_n^2 \tag{3.91}$$

for $\bar{P}_n^2 = \max\{P_n^2, P_n'^2\}\mathbb{I}$. Hence, we apply the McDiarmid inequality to the dilation $H(e) \in \mathcal{H}^{m+p}$ to obtain

$$\mathrm{P}\left(\left\| \frac{1}{N} H(e) \right\| \geq t \right) = \mathrm{P}\left(\left\| \frac{1}{N} \sum_{n=0}^{N-1} (C\delta_n - z_n)e_{n-k'-1}^\top \right\| \geq t \right) \leq (m+p) \cdot e^{-N^2t^2/L} \tag{3.92}$$

for $L = \left\| \sum_{n=0}^{N-1} \bar{P}_n^2 \right\|$. □

Now we focus on bounding the last submatrix $\Phi_N^{(3)}$ (3.64), which is related only to the watermark vector.

Theorem 3.1.10. *The following concentration inequality holds for all $N \geq 1$ and all t:*

$$\mathrm{P}\left(\left\| \Phi_N^{(3)} \right\| \geq t \right) \leq 2q \cdot e^{-N^2t^2/c_N^{(3)}} \tag{3.93}$$

where $c_N^{(3)} = \left\| \sum_{k=0}^{N-1} (\bar{K}_e^2\mathbb{I} - \Sigma_e)^2 + \mathbb{E}[(e_n e_n^\top)^4] - \Sigma_e^2 \right\|$.

Proof. We wish to provide a bound on the norm of

$$\Phi_N^{(3)} = \frac{1}{N} \sum_{n=0}^{N-1} (e_{n-k'-1}e_{n-k'-1}^\top - \Sigma_e) \tag{3.94}$$

Define $\bar{E}_n = e_{n-k'-1}e_{n-k'-1}^\top - \Sigma_e$. We apply the Hoeffding bound for the independent sum to obtain

$$\mathrm{P}\left(\left\| \frac{1}{N} \sum_{n=0}^{N-1} \bar{E}_n \right\| \geq t \right) \leq d \cdot e^{-N^2t^2/2\sigma^2} \tag{3.95}$$

for $\sigma^2 = \frac{1}{2}\left\|\sum_{k=0}^{N-1}\left(\bar{K}_e^2\mathbb{I} - \Sigma_e\right)^2 + \mathbb{E}[(e_n e_n^\top)^4] - \Sigma_e^2\right\|$, since

$$\bar{E}_n^2 \preceq \sum_{k=0}^{N-1}\left(\bar{K}_e^2\mathbb{I} - \Sigma_e\right)^2 \tag{3.96}$$

and by the definition of expectation we have that $\mathbb{E}\left[\bar{E}_n^2\right] = \mathbb{E}[(e_{n-k'-1}e_{n-k'-1}^\top)^4] - \Sigma_e^2$. □

Finite Sample Tests for General LTI Systems

In this section, we provide our finite sample tests based on dynamic watermarking for general LTI Systems with switching. In the previous section, we obtained concentration inequalities for each of the submatrices of Φ_N (3.65). Note $\Phi_N^{(3)}$ is the private excitation matrix we get to design, and so it is in our power to choose the dynamic watermark to display a desired concentration behavior.

We are now ready to state the main theorem of this work, which characterizes the behavior of a switching rule based on the finite-time concentration inequalities. Our switching rule is constructed by thresholding the block submatrices $\Phi_N^{(1)}$ and $\Phi_N^{(2)}$ using the measurements of the second observer ((3.46) and (3.47)) and applying the switch on the first observer once those thresholds are violated, and then we switch back when violations disappear. We remark that the second observer is assumed to be not susceptible to attacks, and hence its measurements can be used to constructed the required matrices and also enjoy the guarantees provided in the previous section. Let S be a positive constant such that $\max\{c_N^{(1)}, c_N^{(2)}, c_N^{(3)}\} \leq NS$; such an S exists when $(A + BK)$ and $(A + L_nC_n)$ are Schur stable provided that the switching rule satisfies the condition specified in Proposition 3.1.5.

Theorem 3.1.11. *Recall the closed-loop MIMO LTI system (3.18) with α_n being our switching control action that chooses between two different observation matrices. Define the threshold $t_N = \sqrt{(1+\rho)S\log N/N}$, where $\rho > 0$. Let $\Phi_N^{(1)}$ and $\Phi_N^{(2)}$ be defined using the measurements from (3.46) and (3.47). Let α_N be the switching decision rule with*

- *we choose the switching input $\alpha_N = 0$ when we have $\left\|\Phi_N^{(1)}\right\| < t_N$ or $\left\|\Phi_N^{(2)}\right\| < t_N$*
- *we switch from $\alpha_{N-1} = 0$ to $\alpha_N = 1$ when $\alpha_{N-i} = 0$ for $i \in \{1, \ldots, \tau\}$ and $\left\|\Phi_N^{(1)}\right\| \geq t_N$ and $\left\|\Phi_N^{(2)}\right\| \geq t_N$.*

Moreover, let E_N for all $N \geq 1$ denote the event

$$E_N = \left[\left\|\Phi_N^{(1)}\right\| > t_N \bigcup \left\|\Phi_N^{(2)}\right\| > t_N\right] \tag{3.97}$$

Then if $v_N \equiv 0$ for all $N \geq 0$, we have that

$$\mathrm{P}(\limsup_{N\to\infty} E_N) = 0. \tag{3.98}$$

That is, under no attacks our switching rule incorrectly switches the system only a finite number of times.

Proof. Recall that we previously proved the following matrix concentration inequalities for each submatrix:

$$\mathrm{P}\left(\left\| \Phi_N^{(1)} \right\| \geq t_N \right) \leq m \cdot e^{-N^2 t_N^2 / c_N^{(1)}} \tag{3.99}$$

$$\mathrm{P}\left(\left\| \Phi_N^{(2)} \right\| \geq t_N \right) \leq (m+p) \cdot e^{-N^2 t_N^2 / c_N^{(2)}} \tag{3.100}$$

for the constants $c_N^{(1)}$ and $c_N^{(2)}$. Summing over all N, we have

$$\sum_{k=1}^{\infty} \mathrm{P}\left(\left\| \Phi_k^{(j)} \right\| \geq t_k \right) \leq (m+p) \int_1^{\infty} \frac{1}{k^{1+\rho}} dk < \infty. \tag{3.101}$$

Hence the Borel-Cantelli Lemma implies that for the event

$$E_N^{(j)} = \left[\left\| \Phi_N^{(j)} \right\| \geq t_N \right] \tag{3.102}$$

we have

$$\mathrm{P}(\limsup_{N \to \infty} E_N^{(j)}) = 0, \ \forall j = \{1, 2, 3\}. \tag{3.103}$$

Now, if we define the event

$$E_N = \left[\left\| \Phi_N^{(1)} \right\| > t_N \bigcup \left\| \Phi_N^{(2)} \right\| > t_N \right], \ N \geq 1, \tag{3.104}$$

then it follows that

$$\mathrm{P}(E_N) \leq \mathrm{P}\left(\bigcup_{j=1}^{2} E_N^{(j)} \right) \leq \sum_{j=1}^{2} \mathrm{P}\left(E_N^{(j)} \right), \ N \geq 1. \tag{3.105}$$

So summing once more for all N gives

$$\sum_{k=1}^{\infty} \mathrm{P}(E_k) \leq \sum_{k=1}^{\infty} \sum_{j=1}^{2} \mathrm{P}\left(E_N^{(j)} \right) < 2(m+p) \int_1^{\infty} \frac{1}{k^{1+\rho}} dk < \infty. \tag{3.106}$$

We obtain by applying Borel-Cantelli lemma that

$$\mathrm{P}\left(\limsup_{N \to \infty} E_N \right) = 0, \tag{3.107}$$

which is the desired result. □

The result of this theorem implies that if there is no attack to the system, the operator norm of the matrices involved can have "large" deviations only a finite number of times, hence we obtain that a switching rule based on tests derived from the concentration inequalities defined previously will trigger attack alerts only a finite number of times. In addition, we note that the having a second observer to compute the finite tests is the key to ensure that the concentration inequalities are consistent with the obtained measurements. On the other hand, the first observer measurements with switching plays the role in the control synthesis. Lastly, we observe that we do not need to enforce the test on $\left\|\Phi_3^{(3)}\right\|$ since this submatrix is only composed of the watermarking signal, and the attacks do not have the power to affect the watermarking imposed by the controller.

Attack Magnitude Thresholding

The previous section gives a finite sample test that works properly when there is no attack. Our goal here is to determine the trade-off between our test's statistical power and the attack magnitude. Intuitively, the power of the statistical test is related to the capability of detecting attacks, and a test with higher power can detect attacks with smaller magnitude. The trade-off is that by increasing the test's statistical power we also increase the rate of false-positives in our detection strategy. Namely, the power of our test is directly related to right-hand side of our finite sample tests, the threshold quantities t_N and we analyze their relation to the magnitude of the attack vectors. To do so, we consider the first observation matrix under a FDI attack $y_n = Cx_n + z_n + v_n$, where v_n is chosen by the attacker. First we consider the case where v_n is a small perturbation that does not necessarily follow any particular form. Then we consider a more structured case where v_n takes the form explored in [130]. Note we have again omitted the subscript of the observation matrix for clarity.

Perturbation Attacks

The first attack we analyze is when v_n consists of a small perturbation that could be determinstic and/or stochastic. To begin our analysis, let $\bar{\delta}_n$ be the measurement error when the system is under attack, and observe that

$$\bar{\delta}_{n+1} = (A + LC)\bar{\delta}_n - \mathbb{I}w_n - L^\top z_n - L^\top v_n. \tag{3.108}$$

Expanding this expression gives that

$$\bar{\delta}_n = (A + LC)^n \bar{\delta}_0 - \sum_{k=0}^{n-1} (A + LC)^{n-1-k}(\mathbb{I}w_k + L^\top z_k + L^\top v_k) \tag{3.109}$$

where $\bar{\delta}_0 = \delta_0 = 0$. So we can rewrite the above as

$$\bar{\delta}_n = \sum_{k=0}^{n-1} \bar{D}_k w_k + \bar{L}_k z_k + \bar{L}_k v_k = \delta_n + \sum_{k=0}^{n-1} \bar{L}_k v_k. \tag{3.110}$$

Next we define $V_n = C\sum_{k=0}^{n-1}\bar{L}_k v_k - v_n$, and observe that the quantity V_n is determined by the attacker since it depends upon the values of v_k. Qualitatively, we note that the magnitude of V_n is related to the attack magnitude, since if there is no attack then $V_n \equiv 0$ for all n. Intuitively V_n is an "accumulation" of the attack from time period 0 to period n, and the accumulation is done via the filters $\bar{L}_k, k \in \{0, ..., n-1\}$. We proceed to analyze the quantity V_n instead of each individual vector v_n since it is not possible to decouple the effects of each v_k by the detection system.

Theorem 3.1.12. *Consider the closed-loop MIMO LTI system (3.18) with α_N, t_N, E_N as defined in Theorem 3.1.11, and suppose the attacker chooses the perturbation attack described above. If the attack values v_k are such that there exists a positive constant G with*

$$\frac{1}{N}\sum_{k=0}^{N-1}\|V_k\| \leq \frac{G}{N}. \tag{3.111}$$

then we have that $\mathrm{P}(\limsup_{N\to\infty} E_N) = 0$. *That is, under a perturbation attack with the above specifications the attack is detected only a finite number of times.*

Proof. We begin by considering

$$\begin{aligned}
&\Phi_N^{(1)} = \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top - \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top] = \\
&\frac{1}{N}\sum_{n=0}^{N-1}(C\bar{\delta}_n - z_n - v_n)(C\bar{\delta}_n - z_n - v_n)^\top - \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top] = \\
&\frac{1}{N}\sum_{n=0}^{N-1}(C\delta_n - z_n)(C\delta_n - z_n)^\top + D_n + D_n^\top + M_n - \frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top]
\end{aligned} \tag{3.112}$$

where δ_n is the measurement error under no attack, and

$$\begin{aligned}
D_n &= \frac{1}{N}\sum_{n=0}^{N-1}(C\delta_n - z_n)V_n^\top \\
M_n &= \frac{1}{N}\sum_{n=0}^{N-1}V_n V_n^\top
\end{aligned} \tag{3.113}$$

Now using Theorem 3.1.7, we have that

$$\begin{aligned}
&\mathrm{P}(\left\|\Phi_N^{(1)}\right\| \geq t_N) \leq \mathrm{P}\Bigg(\|\frac{1}{N}\sum_{n=0}^{N-1}(C\delta_n - z_n)(C\delta_n - z_n)^\top - \\
&\frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[(C\hat{x}_n - y_n)(C\hat{x}_n - y_n)^\top]\| \geq t_N - 2\left\|D_N\right\| - \left\|M_N\right\|\Bigg) \leq m\, e^{\frac{-N^2(t_N - 2\|D_N\| - \|M_N\|)^2}{c_N^{(1)}}}
\end{aligned} \tag{3.114}$$

Next observe that

$$2\left\|\bar{D}_N\right\| + \|M_N\| \leq \frac{2\bar{K}_N}{N}\sum_{n=0}^{N-1}\|V_n\| + \frac{1}{N}\sum_{n=0}^{N-1}\|V_n\|^2 \leq \frac{2\bar{K}_N G}{N} + \frac{G^2}{N} \tag{3.115}$$

Since $(A+LC)$ is Schur stable, then from the definition of $\bar{K}_N$ we immediately get that there exists a positive constant $\bar{S}$ such that $\bar{K}_N \leq \bar{S}$ for all $N \geq 1$. Combining this with the above implies that

$$\sum_{k=1}^{\infty} \mathrm{P}(\left\|\Phi_k^{(1)}\right\| \geq t_k^{(1)}) < \infty, \tag{3.116}$$

and so the Borel-Cantelli lemma implies that $\left\|\Phi_N^{(1)}\right\| \geq t_N^{(1)}$ only finitely many times. Our next step considers

$$\Phi_N^{(2)} = \frac{1}{N}\sum_{n=0}^{N-1}(C\hat{x}_n - y_n)e_{n-k'-1}^{\top} = \frac{1}{N}\sum_{n=0}^{N-1}(C\bar{\delta}_n - z_n - v_n)e_{n-k'-1}^{\top} =$$
$$\frac{1}{N}\sum_{n=0}^{N-1}(C\delta_n - z_n)e_{n-k'-1}^{\top} + H_n \tag{3.117}$$

where

$$H_N = \frac{1}{N}\sum_{n=0}^{N-1} V_n e_{n-k'-1}^{\top}. \tag{3.118}$$

Now using Theorem 3.1.9, we have that

$$\mathrm{P}(\left\|\Phi_N^{(2)}\right\| \geq t_N) \leq \mathrm{P}\left(\|\frac{1}{N}\sum_{n=0}^{N-1}(C\delta_n - z_n)e_{n-k'-1}^{\top}\| \geq t_N - \|H_N\|\right) \leq 2m\, e^{\frac{-N^2(t_N - \|H_N\|)^2}{c_N^{(2)}}}. \tag{3.119}$$

Next observe that

$$\|H_N\| \leq \frac{K_e}{N}\sum_{k=0}^{N-1}\|V_N\| \leq \frac{K_e G}{N}. \tag{3.120}$$

Combining this with the above implies that

$$\sum_{k=1}^{\infty} \mathrm{P}(\left\|\Phi_k^{(2)}\right\| \geq t_k) < \infty, \tag{3.121}$$

and so the Borel-Cantelli lemma implies that $\left\|\Phi_N^{(2)}\right\| \geq t_N$ only finitely many times. The remainder of the proof follows similarly to that of the last steps of Theorem 3.1.11. □

Our analysis in this subsection is capable of only providing a simple relation between the power of our detection scheme and the magnitude of V_n. An analysis that translates to the bounds of each individual v_n is more involved because it depends explicitly on the structure/behavior of the matrix $(A+LC)$.

Replay Attacks

The second attack type we analyze is when

$$v_n = C\xi_n + \zeta_n - (Cx_n + z_n) \tag{3.122}$$

where $\xi_{n+1} = (A + BK)\xi_n + \omega_n$ and ω_n is a bounded disturbance. This is a *replay attack* [149], since it subtracts the real sensor measurements and substitutes these with a replay of the dynamics starting from a different initial condition. In fact, we will perform our analysis for a more general attack

$$v_n = C\xi_n + \zeta_n - \gamma \cdot (Cx_n + z_n), \tag{3.123}$$

where $\gamma \in \mathbb{R}$. This attack also allows for dampening or amplifying the true sensor measurements $(Cx_n + z_n)$.

Theorem 3.1.13. *Consider the closed-loop MIMO LTI system (3.18) with α_N, t_N, E_N as defined in Theorem 3.1.11, and suppose the attacker chooses the attack (3.123). If the attack is not trivial (i.e., a trivial attack has $v_N \equiv 0$ for all $N \geq 0$), then we have that* $\mathrm{P}(\limsup_{N\to\infty} \neg E_N) = 0$. *That is, under the attack with the above specifications the attack is* not *detected only a finite number of times.*

Proof. Suppose $\gamma \neq 0$. Then the proof of Theorem 1 in [130] shows that $\lim_{N\to\infty} \Phi_N^{(2)}$ exists almost surely and is not equal to 0. This means that $\mathrm{P}(\limsup_{n\to\infty} \neg E_N^{(2)}) = 0$. Now consider the case $\gamma = 0$. Then the proof of Theorem 1 in [130] shows that $\lim_{N\to\infty} \Phi_N^{(1)}$ exists almost surely and is not equal to 0. This means that $\mathrm{P}(\limsup_{n\to\infty} \neg E_N^{(1)}) = 0$. The remainder of the proof by repeating the last steps of Theorem 3.1.11 for the two cases, after noting that $\neg E_N = \neg E_N^{(1)} \vee \neg E_N^{(2)}$ by De Morgan's laws. □

This result is stronger than Theorem 3.1.12 in that it says all replay attacks, and more generally attacks of the form (3.123), will *not* be detected by the finite sample tests only a finite number of times. In fact, this result is analagous to the zero-average-power results (3.4) of past work on dynamic watermarking for LTI systems with general structure [130], since this result says that only (trivial) replay attacks with zero-average-power cannot be detected.

3.2 Experimental Results: Finite-time switching with Autonomous vehicles

To further demonstrate the effectiveness of this method, we return to the lane keeping example used in [130] which is based off of the standard model for lane keeping and speed control [241]. In this model the state vector takes the form $x^T = [\psi\ y\ s\ \gamma\ v]$ and input vector $u^T = [r\ a]$, where ψ is heading error, y is lateral error, s is trajectory distance, γ is vehicle

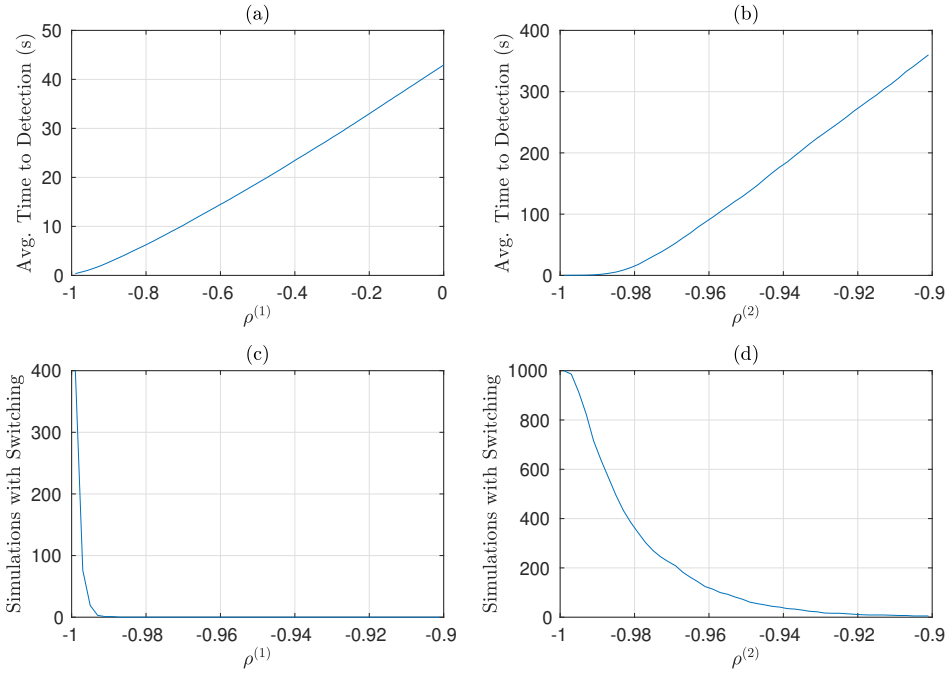

Figure 3.2: Average Time to Detect Replay Attack using Submatrix $\left\|\Phi_n^{(1)}\right\|$ with Threshold Parameter $\rho^{(1)}$ (a) and Submatrix $\left\|\Phi_n^{(2)}\right\|$ and Threshold Parameter $\rho^{(2)}$ (b); Number of Un-Attacked Trials that Result in Switching using Submatrix $\left\|\Phi_n^{(1)}\right\|$ with Threshold Parameter $\rho^{(1)}$ (c) and Submatrix $\left\|\Phi_n^{(2)}\right\|$ and Threshold Parameter $\rho^{(2)}$ (d)

angle, v is vehicle velocity, r is steering, and a is acceleration. Linearizing about a straight trajectory at a velocity of 10 m/s and step size of 0.05 seconds gives us an LTI system:

$$A = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{10} & 0 \\ \frac{1}{2} & 1 & 0 & \frac{1}{40} & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{400} & 0 \\ \frac{1}{2400} & 0 \\ 0 & \frac{1}{800} \\ \frac{1}{20} & 0 \\ 0 & \frac{1}{20} \end{bmatrix} \tag{3.124}$$

with $C_1 = C_2 = [I, 0] \in \mathbb{R}^{3\times 5}$. The process noise and watermark take the form of uniform random variables such that $w \in [-2.5 \times 10^{-4},\ 2.5 \times 10^{-4}]^5$ and $e \in [-2,\ 2]^2$. Similarly the measurement noise for each sensor is also estimated as uniform random variables where $\zeta \in [-1 \times 10^{-2},\ 1 \times 10^{-2}]^3$ and $\eta \in [-2 \times 10^{-2},\ 2 \times 10^{-2}]^3$. For this example we can think of the ζ measurements as localization using visual or lidar based localization with high definition mapping, and η as GPS localization. Finally controller and observer gains K and $L_1 = L_2$ were chosen to stabilize the closed loop system.

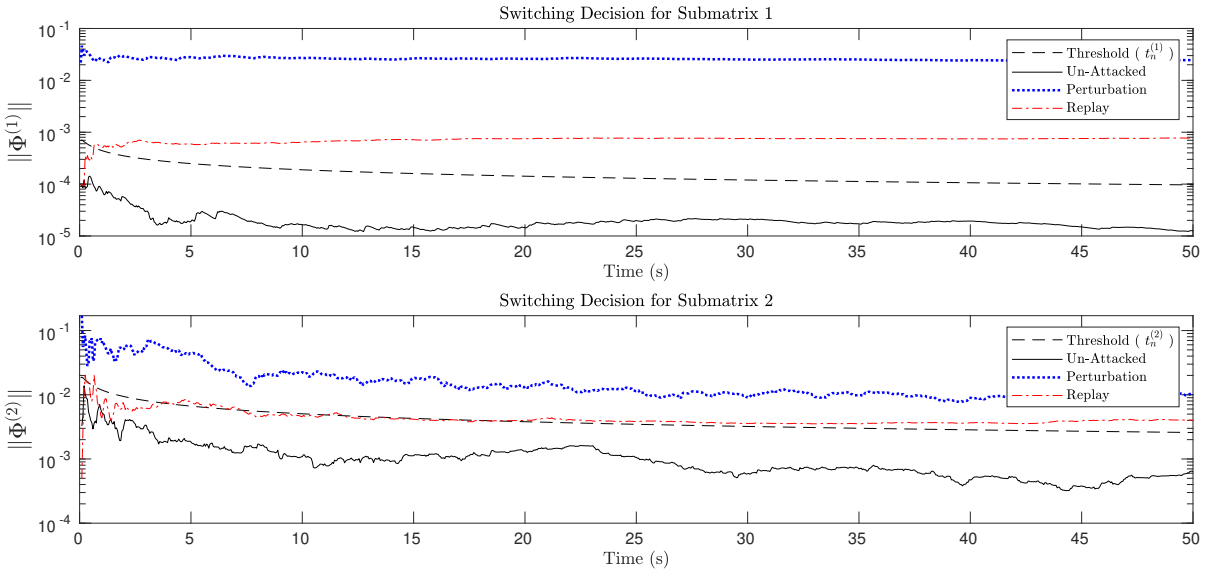

Figure 3.3: Switching Decision Values Based on the Submatrices $\|\Phi_n^{(1)}\|$ and $\|\Phi_n^{(2)}\|$ in Simulation of Autonomous Vehicle with Switching Disabled

For this system it was found that

$$c_n^{(1)} \leq 6.7502 \times 10^{-5} n \tag{3.125}$$

$$c_n^{(2)} \leq 0.0968 n. \tag{3.126}$$

Using the threshold structure defined in Theorem 3.1.11 results in

$$\tau_n^{(1)} = \sqrt{(1+\rho^{(1)})(6.7502 \times 10^{-5}) \log(n)/n} \tag{3.127}$$

$$\tau_n^{(2)} = \sqrt{(1+\rho^{(2)})(0.968) \log(n)/n}. \tag{3.128}$$

While the finite switching guarantee given by Theorem 3.1.11 only applies for $\rho^{(1)}, \rho^{(2)} > 0$, due to the conservative nature of the bounds in (3.125)-(3.126) in addition to the desire to also maintain a sufficiently quick detection we instead heuristically tune these values to find the desired balance.

For our analysis of this system, we once again consider the two forms of attack discussed previously. The perturbation attack takes the form of random noise pulled from a uniform distribution such that $v_n \in [-0.150.15]^3$ The replay attack is described in (3.122) where $\xi_0 = 0$ and ζ and ω are uniformly distributed such that $\zeta \in [-2.5 \times 10^{-4},\ 2.5 \times 10^{-4}]^3$ and $\omega \in [-2.5 \times 10^{-4},\ 2.5 \times 10^{-4}]^5$.

Each attacked system, along with an un-attacked system were simulated 1000 times for 10,000 discrete time steps. While the perturbation attack is detected and switching occurs

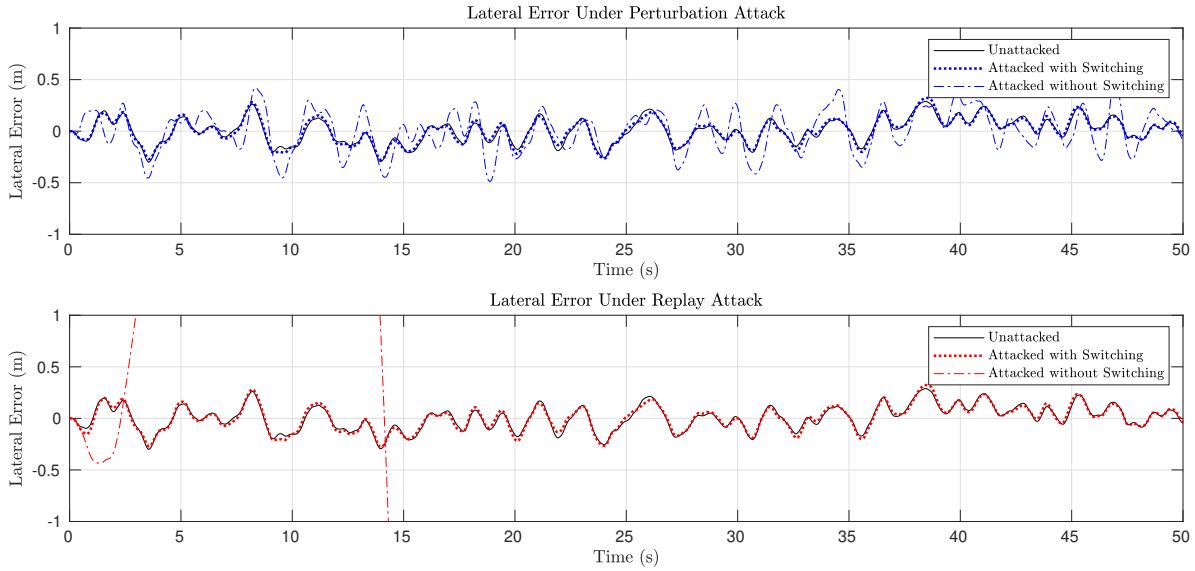

Figure 3.4: Performance Comparison of Simulated Autonomous Vehicle Lane Keeping with and without Switching Policy for Perturbation Attack (top) and Replay Attack (bottom)

almost immediately for $\rho^{(1)}, \rho^{(2)} < 1$, the replay attack can take a much longer time to be detected. Figure 3.2 shows the average time to detection for each of our switching conditions in addition to the number of trials that result in switching for the un-attacked case plotted against the corresponding value of $\rho^{(1)}$ or $\rho^{(2)}$. While the number of switching simulations for the un-attacked system under switching condition 2 appear to be quite large even when the average time to detect is relatively large, it is important to note that many of the unwanted switches occur in the first four discrete steps which can be mitigated in practice by ignoring the first four values.

Choosing values of $\rho^{(1)} = \rho^{(2)} = -0.98$ to balance the possibility of false alarms while maintaining the ability to quickly detect attacks, each attack was again simulated this time for 1000 discrete time steps both with and without the switching policy. Figure 3.3 shows the value of $\|\Phi_n^{(1)}\|$ and $\|\Phi_n^{(2)}\|$ for both normal operation and under each of the attacks when the switching policy is not being used. The plot shows that for both attack 1 and attack 2 the switching policy will result in an almost immediate and consistent transfer from the attacked sensor to the protected sensor. Furthermore, when the system is un-attacked the values of $\|\Phi_n^{(1)}\|$ and $\|\Phi_n^{(2)}\|$ remain below the switching threshold. Figure 3.4 compares the performance of the lane keeping algorithm for each attack with respect to the un-attacked performance both with and without the switching policy. This plot shows that for both attacks the switching policy is able to transfer to the protected sensor before significant deviation can occur. This switch allows the vehicles performance to gracefully degrade while avoiding total failure.

3.3 Statistical Consistency of Set-Membership Estimator for Linear Systems

Learning-based control has seen a resurgence in the past few years [16, 166, 189, 1] because of recent advances in system identification using machine learning and artificial intelligence. When the system has unknown dynamics, it becomes paramount to identify the underlying dynamics so that an appropriate controller can be computed in order to make the system stable [156]. System identification has become a central field of research lying in between control and statistics.

Here, we consider a fully observed switched autonomous linear system with bounded process noise. Each linear system is unknown to us, but we control switching between different linear dynamics. This departs from the existing literature on switched system identification, where the switching control is fixed and must also be estimated [247, 99, 150]. Here, a control decision needs to be chosen together with the system identification. The dynamics for a single system may have a mix of stable or unstable modes and repeated eigenvalues. Identification can be done via estimation of the transition matrices [144, 231], and identification of transition matrices for *stable* systems has been studied [156, 232, 31, 256].

The identification problem in our setup is particularly challenging because the switching can cause stability/instability independent of the eigenvalues of each linear system [35]. The study of system identification for unstable systems is not as prolific as work on the stable case. Existing work for the unstable case of identification of a single linear system requires strong assumptions on repeated eigenvalues in order to prove asymptotic convergence [147], derive associated limiting distributions of the estimates of the model parameters [59, 60], and in order to generalize the result to other classes of transition matrices [146, 184, 182].

Recent work [86, 230, 229, 190] has shown the difficulty of identification for unstable linear systems when state observations are restricted to a single trajectory: Ordinary least squares (OLS) is statistically inconsistent when the dynamics have repeated unstable dynamics [183, 195], and this causes poor estimation when the dynamics have unstable modes with close eigenvalues. This can be partly overcome using instrumental variables, but this cannot handle systems matrices with eigenvalues both inside and outside the unit circle [195].

The set-membership estimator [39, 165, 17] exploits boundedness of the noise vector. This estimator has been studied in [73, 155] which provided a bounding ellipsoidal algorithm to obtain consistent estimators. Our work is related to previous studies where such estimators are applied, as in fault detection tests [47], regularized regression [32], robust estimation [100, 239], and kernel-based methods [66]. The work in [191] provides a greedy algorithm that uses a set-membership estimator to identify input-output models.

Our main contribution is to prove (strong) statistical consistency of the set-membership estimator for switched linear autonomous systems, where measurements are not sequential and the system modes may be unstable. In past work, either the measurements were assumed to be sequential or statistical consistency was not proved. We use the idea behind Wald's

Theorem [249] to develop a novel consistency proof, in a way not done in other works [165, 32]; however, Wald's Theorem itself does not apply to set-membership estimation, which imposes one constraint for each measurement, and only holds for estimators that minimize a lower semicontinuous loss.

To show a setting where the set-membership estimator is useful, we present a control policy that uses this estimator on a switched linear system. Our policy is a greedy bandit algorithm that uses the set-membership estimator to identify in finite-time the stable mode of the linear system. Our analysis is similar to recent work on greedy bandits [86, 230]. The key difference in our setting is the state observations for each controller are not sequential, and so this means that OLS is not consistent for the matrix estimates in this setting.

Preliminaries and Problem Setup

Throughout this section, we use $\|\cdot\|$ to denote the spectral norm of a matrix, which is the largest singular value of a matrix. We use the function $\rho(A)$ to denote the spectral radius of a matrix A. For a matrix A we let $(A)_{ij}$ denote the (ij)-element of A. For two sets A and B, we denote their Minkowski sum by $A \oplus B$. Furthermore, the volume of set A is $\text{vol}(A)$.

For matrix $A \in \mathbb{R}^{d\times d}$, let $v(A) \in \mathbb{R}^{d^2}$ be a vectorization that stacks elements of A into a vector. For vector $u \in \mathbb{R}^{d^2}$, let $m(u) \in \mathbb{R}^{d\times d}$ be a matricization that folds elements of u into a matrix. We assume that $m \circ v(A) = A$ and $v \circ m(u) = u$. Let $\overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{+\infty\}$ be the extended nonnegative real line. A function $f : \mathcal{D} \to \overline{\mathbb{R}}$ is lower semicontinuous (lsc) at $\overline{x}$ if and only if $\liminf_{x\to\overline{x}} f(x) \geq f(\overline{x})$.

Next, we construct a compactification of $\mathbb{R}^n$ by defining $\mathbf{A}^n = \mathbb{S}^{n-1} \times \overline{\mathbb{R}}_+$, which directly compactifies $\mathbb{S}^{n-1} \times \mathbb{R}_+$. Note $\mathbf{A}^n$ can be shown to be equivalent to the *cosmic closure* of $\mathbb{R}^n$, as defined in [214]. To see why $\mathbf{A}^n$ is a compactification, note we can think of the $\mathbb{S}^{n-1} = \{v \in \mathbb{R}^n : \|v\|_2 = 1\}$ component as a direction of a vector and the $\overline{\mathbb{R}}_+$ component as a length of the vector. Thus our idea is to formally use $\{\lambda v : (v, \lambda) \in \mathbf{A}^n\}$ as a compactification of $\mathbb{R}^n$. We define the expectation $\mathbb{E}[\cdot]$. For a given probability event G, we let $\mathbb{I}_G \in \{0, 1\}$ be an indicator random variable associated with the event G. We use a.s. to denote "almost surely", and we use i.i.d. to denote "independent and identically distributed".

Consider a fully observed switched linear system

$$X_{t+1} = A_{\alpha_t} X_t + w_t \tag{3.129}$$

where $X_t \in \mathbb{R}^d$ is the state, $w_t \in \mathbb{R}^d$ is the i.i.d. process noise, and $\alpha_t \in \{1, ..., q\}$ is the control input that selects one of the (unknown to us) state dynamics matrices $A_1, ..., A_q$. We assume w_t lies in a (known to us) compact, convex set $\mathbf{W} \subset \mathbb{R}^d$ that has a strict interior. Also, the w_t has a (potentially unknown to us) p.d.f $w_t \sim f(w)$, where $\mathbb{E}[w_t] = 0$ and $f(w) > 0$ for all $w \in \mathbf{W}$; this assumption is mild for set-based estimation [17] and ensures the existence of a nonzero lower bound on the p.d.f.

Our goal is to estimate the matrices $A_1, ..., A_p$, and we consider the situation where a subset of the matrices is unstable. In practical control applications, it is important to be able

to precisely characterize the dynamics of each matrix so as to be able to design a stabilizing controller. Moreover, we wish to do the estimation without resetting the system (i.e., using a single state trajectory) and be able to do so given any arbitrary switching control input sequence $\{\alpha_t\}_{t\geq 0}$.

Given an arbitrary (known to us) sequence of switching control inputs $\{\alpha_0, ..., \alpha_{T-1}\}$ of length T, we collect the state measurements $\{x_0, x_1, ..., x_T\}$. In order for the problem to be well-posed, we assume each linear system is selected at least d times. Notationally, we organize measurements into groups where measurement pairs from the same linear system are grouped together: For each system p, we define the sequence of measurement pairs $\{(Y_i^{(p)}, X_i^{(p)})\}_{i=1}^{n_p}$, where n_p is the number of measurement pairs associated with system p. It is essential to note that for any p, a pair $(Y_i^{(p)}, X_i^{(p)})$ is composed of successive observations of the system

$$Y_i^{(p)} = A_p X_i^{(p)} + w_i. \tag{3.130}$$

Note $\{(Y_i^{(p)}, X_i^{(p)})\}_{i=1}^{n_p}$ are generally not successive since $X_{i+1}^{(p)}, Y_i^{(p)}$ are usually not the same because there may be an arbitrary number of switches between observations. Past consistency proofs for unstable systems (see for instance [146]) require sequential measurements: These proofs separate the state dynamics into stable and unstable modes and then invert the unstable modes so that all the necessary quantities in the proof remain finite. When there is arbitrary switching, it is no longer possible to separate stable and unstable modes.

Proposed Estimator and Consistency Proof

Nonsequential observations makes system identification more challenging than estimation of autoregressive models. One naive approach is to use OLS for each group of data. This approach is inconsistent for general A matrices [183, 195], specifically A with multiple geometric roots in the eigenvalue structure of the unstable matrix. Here, we provide an estimator that uses the boundedness of the disturbance vectors to overcome past issues. We prove consistency by adapting a celebrated argument by Wald [249], which is substantially different than typical analysis [86, 230, 229].

Set-Membership Estimator

We focus our analysis on a single group p, and so we drop the superscript for ease of notation. Let $\{(Y_i, X_i)\}_{i=1}^{n}$ be our sequence of measurements We let the associated true dynamics matrix A_p be labeled as A_0. Hence it follows that

$$Y_i = A_0 X_i + w_i, \text{ for } i \in \{1, ..., n\}. \tag{3.131}$$

Once again, we note the measurements pairs (Y_i, X_i) and (Y_{i+1}, X_{i+1}) are neither independent nor consecutive (in time) for any i, in general. We propose to estimate A_0 by the minimizer

to

$$\begin{aligned} \widehat{A} \in \arg \min_{A \in \mathbb{R}^{d \times d}} \ & \frac{1}{n} \sum_{i=1}^{n} l(X_i, Y_i, A) \\ & \text{s.t. } Y_i - AX_i \in \mathbf{W}, \text{ for } i \in \{1, ..., n\} \end{aligned} \tag{3.132}$$

where $l(\cdot)$ is a loss function. For example, we may choose $l(X_i, Y_i, A) = \|Y_i - AX_i\|_2^2$. Observe that when $l(\cdot) \equiv 0$ this simply becomes a feasibility problem. We will first prove consistency of the feasibility version of this problem, which will imply consistency for well-behaved loss functions.

This is a *set-membership estimator* and has been studied from the deterministic perspective [39, 165]. It uses the *a priori* knowledge that process noise belongs to a compact convex set, in order to enforce constraints associated with each measurement pair. Here, we prove statistical consistency for this estimator when applied to this general setting of nonsequential and non-independent sequence of measurements pairs. In particular, by compactifying the domain of the optimization problem we are able to analyze the estimator by considering the statistics at only a finite number of points.

Local Identifiability of Problem Setup

We begin by explicitly writing the feasibility version of the estimation but over a compactified domain:

$$\begin{aligned} \widehat{A} \in \arg\min_{A} \ & \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{W}}(Y_i - AX_i) \\ & \text{s.t. } A \in \{\lambda \cdot m(v) : (v, \lambda) \in \mathbf{A}^{d^2}\} \end{aligned} \tag{3.133}$$

where we define $\delta_{\mathbf{W}} : \mathbb{R}^n \to \overline{\mathbb{R}}$ to be the indicator

$$\delta_{\mathbf{W}}(u) = \begin{cases} 0, & \text{if } u \in \mathbf{W} \\ +\infty, & \text{otherwise} \end{cases} \tag{3.134}$$

As discussed in the next subsection, this compactification is required for the proof technique we use. For notation, let $L(X_i, Y_i, A) = \delta_{\mathbf{W}}(Y_i - AX_i)$. We also need to specify arithmetic [214] for points $(v, +\infty) \in \mathbf{A}^{d^2}$. For any $(\overline{X}, \overline{Y})$ and $\overline{A} \in \{\lambda \cdot m(v) : (v, \lambda) \in \mathbb{S}^{d^2-1} \times \{+\infty\}\}$, define

$$L(\overline{X}, \overline{Y}, \overline{A}) = \liminf_{(X,Y,A) \to (\overline{X}, \overline{Y}, \overline{A})} L(X, Y, A). \tag{3.135}$$

Next, for each subset $S \subseteq \mathbf{A}^{d^2}$ we define

$$\begin{aligned} h(X, Y, S) = \inf \ & L(X, Y, A) \\ & \text{s.t. } A \in \{\lambda \cdot m(v) : (v, \lambda) \in S\} \end{aligned} \tag{3.136}$$

We begin by characterizing the function $L(X, Y, A)$.

Lemma 3.3.1. *Function $L(X, Y, A)$ is lower semicontinuous.*

Proof. Fix $(\overline{X}, \overline{Y})$ and choose $\overline{A} \in \mathbb{R}^{d\times d}$. The function $Y - AX$ is continuous, and $\delta_{\mathbf{W}}(u)$ is lower semicontinuous [214]. Thus $L(\cdot)$ is lower semicontinuous at $(\overline{X}, \overline{Y}, \overline{A})$ since

$$L(X, Y, A) = \delta_{\mathbf{W}} \circ (Y - AX). \tag{3.137}$$

Next fix $(\overline{X}, \overline{Y})$ and choose any $\overline{A} \in \{\lambda \cdot m(v) : (v, \lambda) \in \mathbb{S}^{d^2-1} \times \{+\infty\}\}$. Lower semicontinuity holds at this point by the definition (3.135). □

Next define the extended real-valued function

$$V(A) = \begin{cases} 0, & \text{if } A = A_0 \\ +\infty, & \text{otherwise} \end{cases} \tag{3.138}$$

and define $E(S) = \inf_{(v,\lambda)\in S} V(\lambda \cdot m(v))$. Proving statistical consistency requires verifying that some *identifiability condition* holds [43], which means the underlying distributions are such that incorrect estimates are detected by measurements. If we define the mapping

$$\begin{aligned} B_n(A) &= \frac{1}{n}\sum_{i=1}^{n} L(X_i, Y_i, A) \\ H_n(S) &= \frac{1}{n}\sum_{i=1}^{n} h(X_i, Y_i, S) \end{aligned} \tag{3.139}$$

then we can prove a local identifiability condition holds.

Proposition 3.3.2. *For any A there is an open neighborhood $O(v, \lambda) \subset \mathbf{A}^{d^2}$, where $(v, \lambda) \in \mathbf{A}^{d^2}$ satisfies $A = \lambda \cdot m(v)$, such that $\lim_{n\to\infty} H_n(O(v, \lambda)) = E(O(v, \lambda))$ a.s.*

Proof. Let $(v_0, \lambda_0) \in \mathbf{A}^{d^2}$ be such that $\lambda_0 \cdot m(v_0) = A_0$. Then $h(X_i, Y_i, O(v_0, \lambda_0)) \equiv 0$ for any open neighborhood $O(v_0, \lambda_0)$. This means that we immediately get that

$$\lim_{n\to\infty} H_n(O(v_0, \lambda_0)) = 0 = E(O(v_0, \lambda_0)) \text{ a.s.} \tag{3.140}$$

Now consider any $A \neq A_0$, and let t_i be the time of measurement i for $i \geq 2$. Note $Y_i - AX_i = (A_0 - A)X_{t_i} + w_{t_i}$, and $X_{t_i} = A_{\alpha_{t_i-1}} X_{t_i-1} + w_{t_i-1}$. Thus

$$Y_i - AX_i = (A_0 - A)(A_{\alpha_{t_i-1}} X_{t_i-1} + w_{t_i-1}) + w_{t_i}. \tag{3.141}$$

Let $\kappa = \min_{w\in\mathbf{W}} f(w)$, and note $\kappa > 0$. The distribution of $Y_i - AX_i$ has support $\mathbf{W} \oplus (A_0 - A)\mathbf{W} \oplus Z_i$ for $Z_i = (A_0 - A)A_{\alpha_{t_i-1}} X_{t_i-1}$. The key observation is that $\oplus Z_i$ translates the set $\mathbf{W} \oplus (A_0 - A)\mathbf{W}$. Let $\mathbf{N} = \{(u, v) \in \mathbb{R}^d \times \mathbb{R}^d : u + (A_0 - A)v = 0\}$, and define

$\mathbf{V}(\mathbf{S}) = (\mathbf{S} \oplus \mathbf{N}) \cap (\mathbf{W} \times \mathbf{W})$. Thus we have

$$\mathbb{P}[L(X_i, Y_i, A) = +\infty | T] = \int_x \mathbb{P}[L(X_i, Y_i, A) = +\infty | X_{t_i - 1} = x, T] g(x) dx \geq$$
$$\int_x \left[\int_{(u,v) \in \mathbf{V}_i(A)} \kappa^2 du dv \right] g(x) dx \geq \int_x \left[\int_{(u,v) \in \mathbf{V}(\mathbf{J}(A))} \kappa^2 du dv \right] g(x) dx \geq$$
$$\int_{(u,v) \in \mathbf{V}(\mathbf{J}(A))} \kappa^2 du dv := c(A) \tag{3.142}$$

for any event T independent of (w_{t-1}, w_t), where

$$\mathbf{V}_i(A) = \{(u, v) \in \mathbf{W} \times \mathbf{W} : u + (A_0 - A)v + Z_i \notin \mathbf{W}\} \tag{3.143}$$

and

$$\mathbf{J}(A) \in \arg \min_{\mathbf{S} \subseteq \mathbf{W} \times \mathbf{W}} \{\text{vol}(\mathbf{V}(\mathbf{S})) \mid \text{vol}(\begin{bmatrix} \mathbb{I} & (A_0 - A) \end{bmatrix} \mathbf{S}) = \text{vol}(\mathbf{W} \oplus (A_0 - A)\mathbf{W}) - \text{vol}(\mathbf{W})\} \tag{3.144}$$

and $g(\cdot)$ is the p.d.f. of $X_{t_i - 1}$ conditioned on T. Now define

$$B'_n(A) = \frac{1}{n/2 - 1} \sum_{k=2}^{n/2} L(X_{2k-1}, Y_{2k-1}, A) \tag{3.145}$$

and note that by construction $(w_{t_{2k}-1}, w_{t_{2k}})$ is independent of all $(X_{2k'-1}, Y_{2k'-1})$ for $k' < k$. Thus for $n \geq 2$ we have

$$\mathbb{P}(B_n(A) = 0) \leq \mathbb{P}(B'_n(A) = 0) =$$
$$\mathbb{P}[L(X_{2\lfloor n/2 \rfloor - 1}, Y_{2\lfloor n/2 \rfloor - 1}, A) = 0 | B'_{n-1}(A) = 0] \times \mathbb{P}(B'_{n-1}(A) = 0) \leq$$
$$(1 - c(A)) \cdot \mathbb{P}(B'_{n-1}(A) = 0) \leq \ldots \leq (1 - c(A))^{\lfloor n/2 \rfloor - 1} \tag{3.146}$$

Noting $\text{vol}(\mathbf{W} \oplus (A_0 - A)\mathbf{W}) > \text{vol}(\mathbf{W})$, since $A \neq A_0$ and $\mathbf{W}$ has a strict interior, then Fredholm's theorm for linear algebra implies $\text{vol}(\mathbf{V}(\mathbf{J}(A))) > 0$. Hence $c(A) > 0$ and $\sum_{n=2}^{\infty} (1 - c(A))^{\lfloor n/2 \rfloor - 1} < +\infty$. Thus the Borel-Cantelli lemma implies $B_n(A) = 0$ only finitely often. This proves $\lim_{n \to \infty} B_n(A) = V(A) = +\infty$ a.s.

Consider the same $A \neq A_0$, and define (v, λ) so $\lambda \cdot m(v) = A$. Then for an open neighborhood $O(v, \lambda)$ we have $\mathbf{Z}(v, \lambda) = \cap_{(u,\mu) \in O(v,\lambda)} \mathbf{V}(\mathbf{J}(\mu \cdot m(u)))$ and

$$\mathbb{P}[h(X_i, Y_i, O(v, \lambda)) = +\infty | T] \geq \int_{(u,v) \in \mathbf{Z}(v,\lambda)} \kappa^2 du dv := d(O(v, \lambda)). \tag{3.147}$$

By the Monotone Convergence Theorem, the open neighborhood $O(v, \lambda)$ can be chosen so $(v_0, \lambda_0) \notin O(v, \lambda)$ and so $d(O(v, \lambda)) > 0$. By a similar argument as before we have that $\mathbb{P}(H_n(O(v, \lambda)) = 0) \leq (1 - d(A))^{\lfloor n/2 \rfloor - 1}$. So since $\sum_{n=2}^{\infty} (1 - d(A))^{\lfloor n/2 \rfloor - 1} < +\infty$, the Borel-Cantelli lemma implies $H_n(O(v, \lambda)) = 0$ only finitely often. This proves that

$$\lim_{n \to \infty} H_n(O(v, \lambda)) = E(O(v, \lambda)) = +\infty \text{ a.s.} \tag{3.148}$$

□

The above proposition establishes a local identifiability condition for our setup, namely a setting with nonsequential measurements and linear dynamics. The key intuition is that for any matrix A the sample average $H_n(\cdot)$ converges to its "expectation" $E(\cdot)$ on some open neighborhood of A.

Strong Statistical Consistency

We are now in a position to prove our main theorem, which adapts the argument from the classical Wald Consistency Theorem [249] and relies on the compactification of the feasible region. To understand the intuiton of why we compactify, recall that one definition of a compact set is a set where each of its open covers has a finite subcover. This is important for proving statistical consistency because, when parameters being estimated belong to a compact set, it allows us to perform an analysis only at a finite number of points in order to understand the global behavior. Compactification is important because it enables us to exploit this insight.

Theorem 3.3.3. *The feasibility estimator (3.133) is strongly consistent, meaning*

$$\lim_{n\to\infty} \widehat{A} = A_0 \ \ a.s. \tag{3.149}$$

or equivalently that $\mathbb{P}(\lim_{n\to\infty} \widehat{A} = A_0) = 1$.

Proof. Fix an open neighborhood U around the matrix A_0. Because $A_0 \in \mathbb{R}^{d\times d}$, the set U can be represented as $U = \{\lambda \cdot m(v) : (v,\lambda) \in S\}$ for some $S \subset \mathbb{S}^{d^2-1} \times \mathbb{R}_+$. Recalling the definition of $V(\cdot)$, we know there exists $\epsilon > 0$ such that $V(A) \geq 3\epsilon + V(A_0)$ for $A \in \mathbf{C}(S)$, where

$$\mathbf{C}(S) = \{\lambda \cdot m(v) : (v,\lambda) \in \mathbf{A}^{d^2} \setminus S\}. \tag{3.150}$$

For the next step, consider any fixed point (v,λ) in $\mathbf{A}^{d^2} \setminus S$. Let $\{N_k(v,\lambda)\}_{k\geq 1}$ be a sequence of open balls that shrink to (v,λ) as $k \to \infty$. Since $L(X,Y,A)$ is lower semicontinuous, it follows from the definition of $h(\cdot)$ that $\lim_{k\to\infty} h(X,Y,N_k(v,\lambda)) = L(X,Y,\lambda\cdot m(v))$. Since $A \in \mathbf{A}^{d^2} \setminus S$, the Monotone Convergence Theorem says there is an open neighborhood $N(v,\lambda) \subseteq O(v,\lambda)$ with

$$E(N(v,\lambda)) \geq V(A) - \epsilon \geq V(A_0) + 2\epsilon. \tag{3.151}$$

Now, since $\mathbf{A}^{d^2} \setminus S$ has been compactified then by one definition of a compact set there exists a finite subcover $\mathcal{B}_1, \ldots, \mathcal{B}_z$ of neighborhoods $N(v,\lambda)$ centered around $(v_1,\lambda_1), \ldots, (v_z,\lambda_z)$. This means $\mathbf{A}^{d^2} \setminus S \subseteq \bigcup_{k=1}^{z} \mathcal{B}_k$, and

$$e \inf_{A\in\mathbf{C}(S)} B_n(A) \geq \min_k \frac{1}{n} \sum_{i=1}^{n} h(X_i, Y_i, \mathcal{B}_k). \tag{3.152}$$

Using Proposition 3.3.2 with (3.151) and (3.152) implies

$$\lim_{n\to\infty} \inf_{A\in\mathbf{C}(S)} B_n(A) \geq V(A_0) + 2\epsilon \text{ a.s.} \tag{3.153}$$

By definition of (3.133) and $B_n(\cdot)$, $\widehat{A}$ minimizes $B_n(\cdot)$; hence, for almost all sample paths ω it follows that there exists N such that for all $n > N$ we have

$$B_n(\widehat{A}) \leq B_n(A_0) < V(A_0) + \epsilon < \inf_{A \in \mathbf{C}(S)} B_n(A). \tag{3.154}$$

This implies that $\widehat{A} \in U$ for all $n > N$. We complete the proof by letting the neighborhood U shrink to $\{A_0\}$. □

The above theorem proves consistency of the feasibility estimator (3.133). Consistency of the general estimator (3.132) follows as a direct corollary for well-behaved loss functions.

Corollary 3.3.4. *Suppose the loss function $l(X, Y, A)$ is continuous. Then the general estimator (3.132) is strongly consistent, meaning $\lim_{n\to\infty} \widehat{A} = A_0$ a.s..*

Proof. Since $l(X, Y, A)$ is continuous, any A feasible for (3.132) is feasible for (3.133). Also, A_0 is feasible for (3.132). □

3.4 Numerical Experiments: Set-membership estimator

We demonstrate consistency of our estimator (3.132) through two experiments. The first compares (3.132) to OLS on identification for a dynamics matrix where OLS is inconsistent. The second uses (3.132) to construct a switching control policy that identifies the stable mode of a switched linear system.

Comparison to OLS

Our first numerical experiment uses a single (i.e., no switching) state dynamics matrix that is given by

$$A_2 = \begin{bmatrix} 0 & 1.1 & 0 & 0 \\ 1.1 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 0 \\ 0 & 0 & 0 & 1.1 \end{bmatrix} \tag{3.155}$$

This matrix is unstable since it has $\rho(A_2) = 1.1$. Moreover, the eigenvalue 1.1 has a geometric multiplicity of three. This means OLS is inconsistent when estimating A_2 from X_t even in the absence of switching [183, 195]. In contrast, our estimator (3.132) is consistent by Corollary 3.3.4. This is verified by Fig. 3.5, which shows results of a simulation with process noise that has uniform distribution with support $\mathbf{W} = [-1, 1]^4$. The estimation error of OLS remains nonzero, whereas the estimation error of (3.132) using the loss function $l(X_i, Y_i, A) = \|Y_i - AX_i\|_2^2$ rapidly converges towards zero.

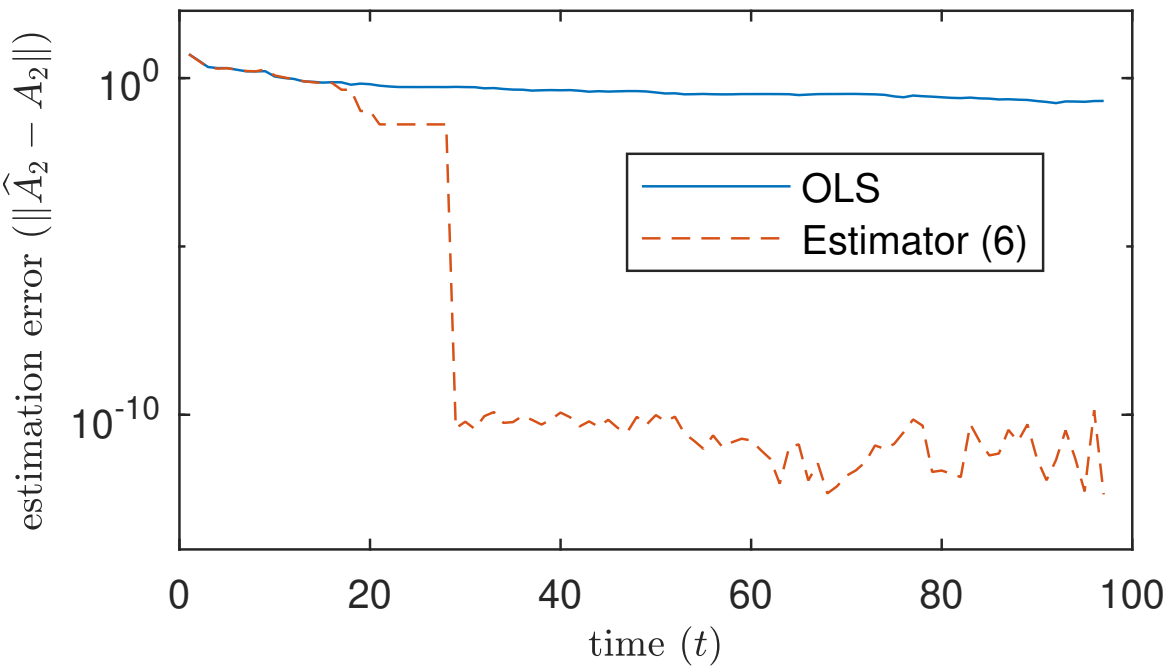

Figure 3.5: Estimation Error From Trajectory by A_2 Without Switching

Greedy Bandit Policy

We next consider the setup in the beginning of this section, constrained so that there exists $s \in \{1, \ldots, q\}$ with $\rho(A_s) < 1$ and $\rho(A_p) > 1$ for all $p \in \{1, \ldots, q\} \setminus \{s\}$. We specifically exclude the case $\rho(A_p) = 1$. Though (3.132) is consistent when $\rho(A_p) = 1$, the policy we construct requires this assumption. We construct a policy that inputs the sequence $X_0, \ldots, X_t$ and $\alpha_0, \ldots, \alpha_{t-1}$ and chooses a control action $\alpha_t \in \{1, \ldots, q\}$ that identifies the stable mode while maintaining stability of the closed-loop system. This problem can be interpreted as a multi-armed bandit [148, 2, 168], which involves a tradeoff between choices that: explore to learn more about the relevant distributions, and exploit by choosing the optimal (according to current estimates) actions. However, under specific assumptions a greedy algorithm can be (asymptotically) optimal [105, 164].

Our procedure is Algorithm 3, and we use the loss function $l(X_i, Y_i, A) = \|Y_i - AX_i\|_2^2$ for (3.132). We wish to identify the stable dynamics in finite time, because then the system can be brought to a stochastic equilibrium by selecting only the stable dynamics. The key idea is to use our estimator, which is consistent for all possible structures of A, once we group measurements as discussed previously. Note this algorithm greedily selects an arm with estimated spectral radius strictly smaller than 1. If at any given time t, no such arm exists, then we randomly select an arm and update the estimates. We can prove this algorithm maintains closed-loop stability:

Algorithm 3 Greedy Bandit Algorithm

Input: set $\{1, ..., q\}$ of candidate systems. initial state X_0

1: **for** systems $p \in \{1, ..., q\}$: **do**
2: select system p
3: obtain new measurement $X^{(p)}_{(1)}$
4: set $n_p \leftarrow 1$
5: compute estimate $\widehat{A}_p$ using (3.132)
6: compute estimate of spectral radius: $\hat{\rho}_p = \rho(\widehat{A}_p)$
7: **end for**
8: **for** each time instant $t > q$: **do**
9: **if** $\min_p\{\hat{\rho}_p\} \geq 1$ **then**
10: randomly select a system p
11: obtain new measurement $X^{(p)}_{(n_p+1)}$
12: set $n_p \leftarrow n_p + 1$
13: compute estimate $\widehat{A}_p$ using (3.132)
14: compute estimate of spectral radius: $\hat{\rho}_p = \rho(\widehat{A}_p)$
15: **else**
16: select any system p such that $\hat{\rho}_p < 1$.
17: obtain new measurement $X^{(p)}_{(n_p+1)}$
18: set $n_p \leftarrow n_p + 1$
19: compute estimate $\widehat{A}_p$ using (3.132)
20: compute estimate of spectral radius: $\hat{\rho}_p = \rho(\widehat{A}_p)$
21: **end if**
22: **end for**

Proposition 3.4.1. *Algorithm 3 chooses the dynamics matrix A_s infinitely many times and chooses the dynamics matrices A_p for $p \in \{1, \ldots, q\} \setminus \{s\}$ only finitely many times.*

Proof. We prove this by contradiction. Suppose there is a $p \in \{1, \ldots, q\} \setminus \{s\}$ such that the unstable dynamics A_p is chosen infinitely many times. Since spectral radius is a continuous function [138], combining the Continuous Mapping Theorem [43, 17] with Corollary 3.3.4 implies $\lim_{n\to\infty} \hat{\rho}_p = \rho_p > 1$ a.s.; hence $\hat{\rho}_p < 1$ only finitely many times. Thus by construction of the algorithm, this means A_p can be chosen by line 16 of the algorithm only finitely many times. So if A_p is chosen infinitely often, this means it must be chosen by line 10 infinitely often. However, if this occurs then we must have that $\hat{\rho}_s > 1$ infinitely often. However, again combining the Continuous Mapping Theorem with Corollary 3.3.4 implies $\lim_{n\to\infty} \hat{\rho}_s = \rho_s < 1$ a.s. This is a contradiction. □

We conducted a numerical simulation to demonstrate the stabilizing behavior of our Algorithm 3. In the scenario we simulated, the process noise had a uniform distribution with support $\mathbf{W} = [-1, 1]^4$. In addition to A_2 as defined in (3.155), we used the state

dynamics matrices

$$A_1 = \begin{bmatrix} 0.76 & 0 & 1.6 & 1.6 \\ 0 & 0.78 & 0 & 1.6 \\ 0 & 0 & 0.79 & 0 \\ 0 & 0 & 0 & 0.79 \end{bmatrix} \tag{3.156}$$

$$A_3 = \begin{bmatrix} 0.91 & 0.7 & 0 & 0 \\ 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0.28 & 0 \\ 0 & 0 & 0 & 1.05 \end{bmatrix} \tag{3.157}$$

$$A_4 = \begin{bmatrix} 0 & 0 & 0.98 & 0 \\ 0 & 0 & 0 & 0.77 \\ 0.98 & 0 & 0.56 & 0 \\ 0 & 0.84 & 0 & 0.14 \end{bmatrix} \tag{3.158}$$

Note $\rho(\bar{A}_1) = 0.7900$, $\rho(\bar{A}_2) = 1.1000$, $\rho(\bar{A}_3) = 1.2899$, and $\rho(\bar{A}_4) = 1.2992$. This means A_1 is Schur stable while the other matrices A_2, A_3, A_4 are not Schur stable. However, $\|\bar{A}_1\| = 2.9136$, whereas $\|\bar{A}_2\| = 1.1000$, $\|\bar{A}_3\| = 1.2899$, and $\|\bar{A}_4\| = 1.2992$. This shows the importance of working with the spectral radius rather than using the spectral norm.

Numerical results of one simulation run are shown in Figures3.6-3.8. Our other simulation runs had behavior that was qualitatively similar to the results we present here. At the beginning, the algorithm tries different arms. After a certain amount of tries of the different arms, the algorithm is able to identify which arm corresponds to the stabilizing mode. When the algorithm is trying different arms, the state grows at an exponential rate; however, once the stabilizing arm is found then the state fluctuates about the origin because of the process noise and the stabilizing action of that arm.

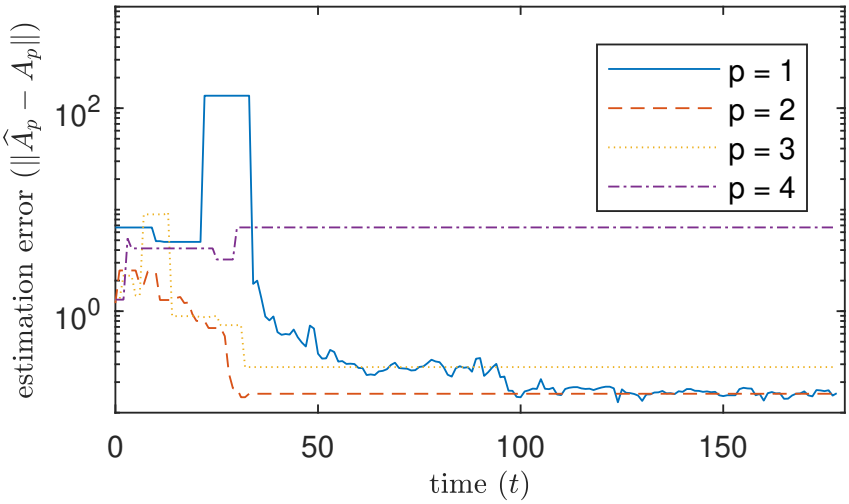

Figure 3.6: Estimation Error Using Our Estimator (3.132)

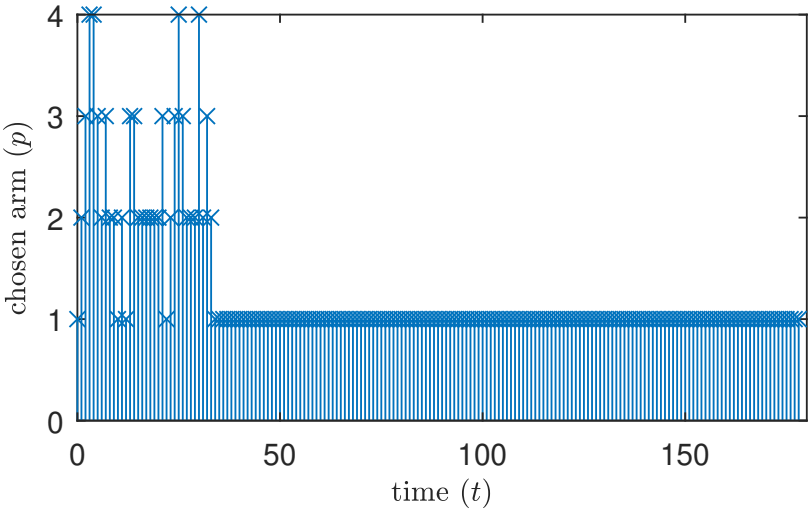

Figure 3.7: Arm p Chosen by Algorithm

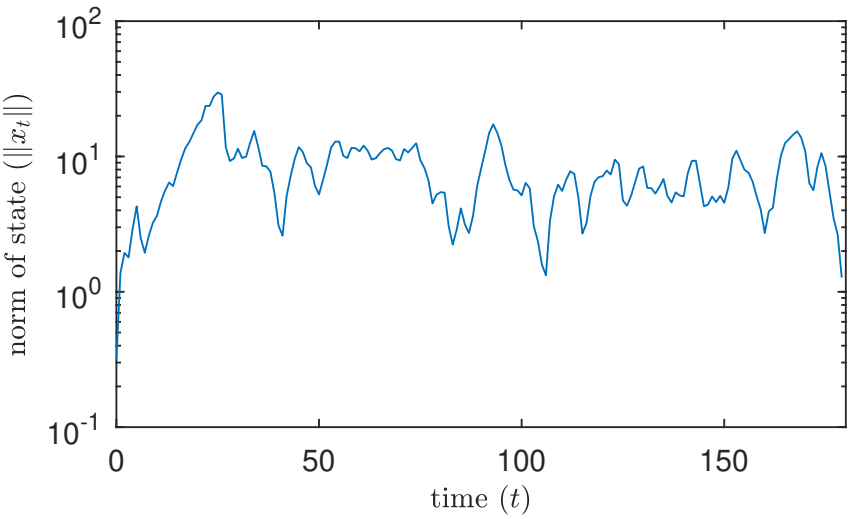

Figure 3.8: Norm of System State

Chapter 4

Detection Algorithm in Competitive Environments

In this chapter we study another type of inference problem that ties hypothesis testing and decision making analysis. In particular we consider a Cyber-Physical System (CPS) where agent's are strategic and can behave selfishly to detriment of the whole system. We also study the opposite problem, where agents who are supposed to be competitive, are in fact colluding.

In section 4.1 we study the latter problem first. We consider a platform where agent's interact and we focus on the question how can we detect whether the agents are competing or colluding. The problem is particularly challenging when agent's do not have complete information on their environment and even when competing agent's may not behave exactly according to some equilibrium concept. Our analysis focus in establishing and proving the efficacy of hypothesis testing in this type of problems.

On section 4.2 we provide a computational case study that illustrate our proposed hypothesis testing methodology in a special type of Bertrand-type environment. In this setting, agents are utility-maximizing and competitive, making decisions that jointly affect their environment, for example, by setting prices of items, which in turn jointly affect the overall demand. A regulator, with no knowledge of the agent's utility function, has access only to the agents' strategies (i.e., pricing decisions) and external shock values in order to decide if agents are behaving in competition according to some equilibrium problem. We leverage the formulation of such a problem as an inverse variational inequality and design a hypothesis test under a minimal set of assumptions.

On the following section, 4.3, we study the former problem: agent's who should be cooperating are in fact competing. Instead of a hypothesis testing framework, we provide a Mechanism Design framework that is able to provide appropriate incentives to the agent's in order to induce an efficient solution as a Nash Equilibrium of a particular game. In this setting, only the coordinating platform applies control inputs, however, it must do so based on information provided by the agents. One major challenge is that if the platform is not

correctly designed then the agents may provide false information to the coordinator in order to achieve improved outcomes for themselves at the expense of the overall system efficiency. Here, we design an interaction mechanism between the agents and the coordinator such that the mechanism: ensures agents truthfully report their information, has low communication requirements, and leads to a control action that achieves efficiency by achieving a Nash equilibrium. We illustrate our proposed mechanism in a model predictive control (MPC) application involving heating, ventilation, air-conditioning (HVAC) control by a building manager of an apartment building on section 4.4.

4.1 Hypothesis Testing Approach to Detecting Collusion in Competitive Environments

There is growing concern about the possibility for tacit collusion using algorithmic pricing, and regulators need tools to help detect the possibility of such collusion. Algorithmic pricing [55, 65] is increasingly used in many cyber-physical domains due to the growth of internet sales channels, but there is concern that the use of such algorithms will lead to tacit collusion that harms consumers [221, 84, 62, 61]. The current situation is unique in that, though it poses challenges for regulators because of the difficulty in detecting tacit collusion by algorithms, there is a large amount of real-time pricing and purchase data available for analysis by regulators. On this section we attempt to answer this pressing issue, where detection algorithm need to be implemented to detect irregularities not from external attackers (Chapter 1), but from the agent's behaviors themselves. As the focal case study, we apply our methodology to a type of Bertrand competition game, where agent's interact via some platform by setting prices. Our results are a initial step towards answering the question of how a regulator may be able to detect algorithmic collusion from a large corpus of pricing and purchase data.

The most closely related literature comes from economics and looks at collusion within auction bidding [198, 30, 194, 28]. One line of work [198, 194] conducts a statistical analysis of bidding data from situations where collusion is known to have occurred, and their results find that collusion (relative to competitive bidding) leads to less aggressive bidding, higher prices for consumers, and increased correlation in bids. Another line of work [198, 30, 28] uses econometrics to detect collusion, and is derived using analyses that assume (perfect) equilibrium behavior. However, assuming perfect equilibrium behavior is too stringent and so these approaches generally lead to too many false positives when trying to detect collusion. Such an assumption is too strong because it requires no model mismatch between the econometrics method and the bidders, and it requires bidders to be exactly accurate in the optimality of their bids.

The significant difference in our work is that we allow the agents to not be in perfect equilibrium. Instead, we presuppose that agents typically choose actions close (but not exactly equal) to equilibrium but that they will also occasionally choose actions far from

equilibrium. This weaker assumption partially mitigates model mismatch because it does not require any data without collusion to exactly match the equilibrium, and it also eliminates the need for agents to be exactly accurate in the optimality of their strategies. These ideas will be made more precise when we present our mathematical model.

Estimation in Equilibrium Models and Coalition Detection

Many competitive and cooperative environments where agents interact can be analyzed using equilibrium models, where solution concepts such as the Nash equilibrium are commonly used to study the agents' strategic behavior [179, 89]. Those models typically contain primitives – such as agents' private information, utility functions, and strategy spaces – that often are not known to an outside regulator or designer who then needs to estimate those elements in order to, for example, design a mechanism of interaction in order to induce a particular behavior or outcome. Throughout this paper we use strategy space and action space interchangeably. A common goal in the *mechanism design* literature is to precisely design such systems where the induced equilibrium maximizes social welfare [139], and a well known example is the VCG-mechanism [246, 67, 115].

However, the estimation of such primitives in equilibrium problems is quite challenging. In this work, we will only consider the case where the agents' actions are observed by the regulator and their strategy spaces are known. Hence the estimation problem lies solely on the agents' utility functions and personal information. One line of work is of structural estimation methods [11, 27, 217], which seeks to estimate parametric utility functions by observing agents acting in equilibrium. In those approaches, it is common to assume a "ground-truth" form of the utility function, and the methods derive necessary conditions based on constrained optimization in order to formally derive estimators of the parameters. Another related line of work is of using surrogate methods [254, 137, 87] that, while not strictly estimation methods, elicit information from the agents themselves about sensitivities over strategies (that is about derivatives of their utility functions with respect to strategies) by providing the agents with a common surrogate function. Those methods are able to induce the appropriate equilibrium behavior even though the agents' utility functions are not known.

Lastly, the estimation problem can be formulated as an inverse variational inequality problem [119, 40]. Such methods leverage the fact that equilibrium problems are a special case of variational inequalities, in order to pose an inverse optimization problem [9] where the solutions (i.e., the equilibrium strategies) are provided via samples, and the goal is to estimate the problem's parameters that generate such solutions. This approach is powerful as it does not require a "ground-truth" model, and under some technical conditions are shown to produce good approximate equilibrium behavior even when the parametric form of the utility functions is misspecified. However, these methods based on inverse optimization can encounter difficulties when the data gathered is noisy [19, 18], which is what happens in most applications.

We build on the framework of [40]. However, we will not assume that the equilibrium actions are provided to the regulator via samples. Instead, we develop a method that given some arbitrary samples of actions is able to identify whether these samples came from agents acting in (approximate) equilibrium or not. Hence our work is more tied to coalition-detection in equilibrium games, in the field of economics, and to hypothesis testing in statistics.

It is common to assume that agents behave in equilibrium, either in total cooperation or in total competition; however this is not what is observed in several applications [37]. In fact, agents often collude, exchange information, and form sub-groups – instead of cooperating as a whole [233, 108]. This poses a serious problem in estimation methods, which often assume the observed strategies come from an specific type of behavior (e.g., total cooperation or competition). The work done in [219, 218] provides a formal characterization of coalitions in games in a series of different environments and provides conditions where the establishment or not of coalitions can be tested. However the problem of coalition formation can also appear where agents play imperfectly and are learning the primitives of the environment while acting on it. Work done by [153] gives evidence that when agents employ learning algorithms in a competitive environment, such algorithms "learn" to implicitly collude, even if collusion is not part of the agents' plans.

In the following, we formulate the problem of identifying whether or not the observed actions come from agents in total competition or not. Instead of identifying the coalition itself based on structural properties of the game, we instead use a data-driven approach where a Kolmogorov-Smirnov hypothesis test [162, 154] is used in order to accept or reject the hypothesis that the strategies observed by the regulator come from agents in equilibrium. This novel approach is powerful in the sense that is independent of the type of utility functions considered and makes very mild assumptions on the *a priori* behavior of the agents, leveraging both the variational inequality formulation and the power of hypothesis testing.

Problem Setting and Model Description

We start by describing the agents' model under consideration and the estimation setting that the regulator faces as it observes the actions taken by the agents. We also present our hypothesis testing framework for detecting collusion.

We will focus on a setting where two agents are engaged in a Bertrand-type game, competing over prices of certain items (e.g., products or airline tickets). In this scenario, we let $p_i \in \mathbb{R}^q$ be the price vectors of agent $i \in \{1, 2\}$ over $\{1, ..., q\}$ items. Throughout the paper, we will utilize the terms prices and strategies interchangeably, as the strategy of each agent consist solely on the prices over the items. We assume the strategy space of each agent is denoted $\mathcal{P}_i$ and has the form

$$\mathcal{P}_i = \{p \in \mathbb{R}^q : Ap = b, p \in K\} \tag{4.1}$$

where A is a $m \times q$ matrix, b is a m-vector, and K is a closed convex cone. Hence the strategy space is a cone given in standard form. In addition, each agent has their own utility function

$$\mathbf{U}_i(p_1, p_2, \mu; \theta_i) = p_i D_i(p_1, p_2, \mu; \theta_i) \tag{4.2}$$

where $D_i(p_1, p_2, \mu; \theta_i)$ is the agent's demand function, which depends on both price vectors and is parametrized by the vector $\theta_i \in \mathbb{R}^d$, which we assume to be the private information of each agent. Lastly, the demand function also depends on μ, which is a common shock value disturbance that we assume to be a bounded disturbance with small magnitude (to be made precise in next subsection). The goal of this shock vector is to represent uncertainties that may affect the demand and are outside of the agents' control.

The goal of each agent is to select a feasible price $p_i \in \mathcal{P}_i$ such that its utility function is maximized. We focus our analysis to the Nash equilibrium of the resulting game:

Definition 4.1.1. A strategy profile (p_1^*, p_2^*) is a Nash equilibrium if each agent plays the "best-response" to the other, namely if

$$p_i^* \in \arg\max_{p_i \in \mathcal{P}_i} \mathbf{U}_i(p_1, p_2^*, \mu; \theta_i), \text{for } i \in \{1, 2\} \tag{4.3}$$

The (pure-strategy) Nash Equilibrium may not be an adequate solution concept for this constrained environment of the Betrand-type game, as it may not exist [83]. However, we will focus on the case where each agent plays imperfectly. Namely, we assume that before playing the game, a gap value ϵ is sampled from a (known to the agents and regulator) parametric distribution $\mathcal{D}(\phi)$ with (unknown to the regulator but known to the agents) parameter ϕ. Next, both agents pick a strategy vector (p_1, p_2) that is an ϵ-approximate Nash equilibrium of the Bertrand game. To formalize this notion of approximate equilibrium, we will use the characterization based on variational inequalities presented in [40].

Definition 4.1.2. Given a function $\mathbf{f} : \mathbb{R}^q \to: \mathbb{R}^q$ and a non-empty set $\mathcal{F} \in \mathbb{R}^q$, the problem of finding the point p^* such that

$$\mathbf{f}(p^*)^\top (p - p^*) \geq 0, \text{for } p \in \mathcal{F} \tag{4.4}$$

is called the variational inequality problem $VI(\mathbf{f}, \mathcal{F})$.

It turns out that several problems can be formulated as variational inequalities (We refer to [119] for an in-depth characterization). In particular, if we let $\mathcal{F} = \mathcal{P}_1 \times \mathcal{P}_2$ and we let

$$\mathbf{f}(p) = \begin{bmatrix} \mathbf{f}_1(p_1, p_2) \\ \mathbf{f}_2(p_1, p_2) \end{bmatrix} = \begin{bmatrix} -\nabla_1 U_1(p_1, p_2, \mu; \theta_1) \\ -\nabla_2 U_2(p_1, p_2, \mu; \theta_2) \end{bmatrix} \tag{4.5}$$

where ∇_i is the gradient w.r.t. p_i, then solving $VI(\mathbf{f}, \mathcal{F})$ from (4.4) is equivalent to finding the Nash equilibrium (4.3). With this is mind, we can establish the following definition for approximate Nash equilibrium:

Definition 4.1.3. A strategy profile $(\bar{p}_1, \bar{p}_2)$ is an ϵ-approximate Nash equilibrium if and only if

$$\mathbf{f}(\bar{p})^\top (p - \bar{p}) \geq -\epsilon, \text{for } p \in \mathcal{F}. \tag{4.6}$$

It will suit our purposes to formulate the above approximate variational inequality problem as a (convex) optimization problem. This can be done under technical regularity conditions that ensure constraint qualification holds (e.g., Slater's condition).

Theorem 4.1.4. *[40] Let $\mathcal{F} = \mathcal{P}_1 \times \mathcal{P}_2$, where $\mathcal{P}_i$ is given by (4.1), for $i \in \{1, 2\}$. Let $\boldsymbol{f}$ be given by (4.5). If $\mathcal{F}$ satisfies constraint qualification (e.g., Slater's condition), then a strategy profile $(\bar{p}_1, \bar{p}_2)$ is an ϵ-approximate Nash equilibrium if and only if*

$$\exists y_1, y_2 \in \mathbb{R}^m : \begin{cases} A_i^\top y_i \leq_C \boldsymbol{f}_i(\bar{p}_1, \bar{p}_2), \text{for } i \in \{1, 2\} \\ \sum_{i=1}^2 \boldsymbol{f}_i(\bar{p}_1, \bar{p}_2)^\top \bar{p}_i - b_i^\top y_i \leq \epsilon \end{cases} \tag{4.7}$$

where we use the symbol "$\leq_C$" to denote conic inequalities.

Next we assume that given some $\epsilon \sim \mathcal{D}(\phi)$, the agents solve the above feasibility problem in order to select the prices. In particular, we assume that the agents solve the above problem where the second inequality is replaced by an equality constraint – that is the selected strategies satisfy condition (4.6) with equality. We will not focus on how such prices are achieved, that is, how the agents learn to play the ϵ-approximate Nash Equilibrium strategies (we refer to [153] for a discussion about learning in cooperative games). Instead, we will focus on the following estimation problem faced by an external regulator: Given a sequence of observed prices and shocks $\{(p_1^j, p_2^j, \mu^j)\}_{j=1}^N$, the regulator would like to ascertain whether or not agents are playing according to ϵ-approximate Nash Equilibrium or not. In this setup, the private information vectors (θ_1, θ_2) of each agent are so-called *nuisance parameters* for the regulator (i.e., they require estimation even though they are not of primary interest). To that end, the regulator will construct estimates $(\hat{\theta}_1, \hat{\theta}_2)$ of the private information vectors and residual estimates $\hat{\epsilon}_j$ for each observation tuple $j \in \{1, ..., N\}$ by solving the inverse variational problem given by

$$\min_{\hat{\theta}, y, \hat{\epsilon}} \; L(\hat{\epsilon}^1, ..., \hat{\epsilon}^N) \tag{4.8}$$

$$\text{s.t. } A_i^\top y_i^j \leq_C \mathbf{f}_i(p_1^j, p_2^j), \text{for } i \in \{1, 2\}, j \in \{1, ..., N\} \tag{4.9}$$

$$\sum_{i=1}^2 \mathbf{f}_i(p_1^j, p_2^j)^\top p_i^j - b_i^\top y_i^j = \hat{\epsilon}^j, \text{for } j \in \{1, ..., N\} \tag{4.10}$$

where $L(\hat{\epsilon}^1, ..., \hat{\epsilon}^N)$ is some loss function over the residual estimates. We assumed that the regulator knows the distribution $\mathcal{D}(\phi)$, but does not know ϕ. Hence the loss function can be written, for example, as the negative log-likelihood as a function of ϕ [225]. We note in this optimization problem, the prices are given by our N samples, and we seek to select a $\hat{\theta}$ such that the resulting utilities form an approximate Nash equilibrium for every sample collected, where the computed $\hat{\epsilon}_j$ are our residual estimates of ϵ.

Lastly, in order to make a decision as to whether or not the observed prices are in approximate Nash equilibrium, the regulator will formulate a hypothesis test over the computed residuals.

Hypothesis Testing Framework

In order to formalize the hypothesis testing framework, we begin by describing the temporal sequence of events under consideration:

1. Both agents and regulator observe μ, the shock variable.
2. The agents solve the feasibility problem (4.7) for some $\epsilon \sim \mathcal{D}(\phi)$. The strategies (p_1, p_2) are selected to exactly be an ϵ-approximate Nash Equilibrium.
3. The regulator observes the strategies (p_1, p_2) and records it.
4. Steps 1-3 are repeated N times, on which the regulator collects the sample tuples $\{(p_1^j, p_2^j, \mu^j)\}_{j=1}^N$.
5. The regulator solves the inverse variational problem (4.8) for some parametric utility functions and computes the estimated residuals $\hat{\epsilon}^1, ..., \hat{\epsilon}^N$.
6. The regulator uses those residuals to perform a Kolmogorov-Smirnov test (to be defined next).

We note that the regulator does not know the true utility functions of the agents. Importantly, our approach is partially amenable to parametric form misspecification because non-colluding agents are not required to be in perfect equilibrium. In other words, some amount of the ϵ^j are meant to capture model misspecification.

Step 6 is conducted as follows: The regulator will use the computed estimated residuals to perform a hypothesis test to determine if the $\hat{\epsilon}^1, ..., \hat{\epsilon}^N$ come from the distribution $\mathcal{D}(\phi)$. However, even though the regulator knows the distribution's parametric form, they do not know the underlying parameter ϕ. Hence, hypothesis tests such as the standard Kolmogorov-Smirnov test are not applicable since they require knowing the true underlying parameters of the distribution under the null hypothesis. Therefore, we we will resort to the Lilliefors variation of the Kolmogorov-Smirnov test [154]. We first compute the empirical cumulative distribution function

$$\hat{F}_N(d) = \frac{1}{N} \sum_{j=1}^{N} \mathbb{I}(\hat{\epsilon}_j \leq d) \tag{4.11}$$

where $\mathbb{I}(\cdot)$ is an indicator function. Then the regulator computes some estimate $\hat{\phi} = g(\hat{\epsilon}^1, ..., \hat{\epsilon}^N)$ and computes the cumulative distribution function

$$\bar{F}_N(d) = F_{\mathcal{D}(\hat{\phi})}(d) \tag{4.12}$$

where $F_{\mathcal{D}(\hat{\phi})}(d)$ is the cumulative distribution function of a random variable of distribution $\mathcal{D}(\hat{\phi})$. Lastly, the regulator computes the test statistic

$$D^* = \max_d |\hat{F}_N(d) - \bar{F}_N(d)| \tag{4.13}$$

The null H_0 and alternative H_1 hypotheses for our test are

$$\mathcal{H}: \begin{cases} H_0: & \text{The agents are behaving in an } \epsilon\text{-approximate} \\ & \text{equilibrium where } \epsilon \sim \mathcal{D}(\phi) \\ H_1: & \text{Otherwise} \end{cases} \tag{4.14}$$

And the decision of whether to accept or reject the null hypothesis is made using the decision-rule

$$\begin{cases} \text{reject } H_0: & \text{if } D^* \geq \tau(N) \\ \text{accept } H_0: & \text{if } D^* < \tau(N) \end{cases} \tag{4.15}$$

where $\tau(N)$ is some threshold from the Lilliefors variation of the Kolmogorov-Smirnov test [154] and which is based on the number of samples collected and the desired significance level α.

Primal-Dual Algorithm to Generate Approximate Equilibrium Prices

A key part of our numerical simulations is to generate prices that are ϵ-approximate equilibrium. In the general case, we need to solve the variational inequality formulation in 4.7. That problem is hard to solve in general, but tailored algorithms do exist (we refer to[103] for an overview of such methods). However, for our setting the feasible region $\mathcal{P}$ contains only bounds on the prices. Hence the problem becomes to find prices (p_1, p_2) such that

$$\exists y_1, y_2 \geq 0: \begin{cases} y_i \geq D_i(p_1, p_2.\mu, \bar{\theta}_i) + p_i\theta_{i,i}, \text{for } i \in \{1,2\} \\ \sum_{i=1}^{2} \bar{p} y_i - p_i(D_i(p_1, p_2.\mu, \bar{\theta}_i) + p_i\theta_{i,i}) = \epsilon \end{cases} \tag{4.16}$$

With that in mind, we are able to generate samples of (p_1, p_2) by designing an acceptation/rejection of samples based on the shock values μ and nuisance parameters (η_1, η_2). First, we sample μ and η_1, η_2 according to their specified distributions. Then we solve the following system of nonlinear equations (via, for example, Newton's Method):

$$p_i D_i(p_1, p_2, \mu, \bar{\theta}_i) + (p_i)^2\theta_{i,i} = \frac{-\epsilon}{2}, \text{for } i \in \{1,2\}. \tag{4.17}$$

After solving this system, if $(p_1, p_2) \in \mathcal{P}$ then it means they are ϵ-approximate solution to the variational inequality problem (since we can set both y_1 and y_2 to zero), and we accept the sample (p_1, p_2, μ). If $p_1 < 0$ or $p_2 < 0$, then we reject the sample. Now without loss of generality, suppose that $p_1 > \bar{p}$. Then we can set $p_1 = \bar{p}$ and let $y_1 = D_1(\bar{p}, p_2.\mu, \bar{\theta}_1) + \bar{p}\theta_{1,1}$. Then by letting $y_2 = 0$ we solve for p_2

$$p_2 D_2(\bar{p}, p_2, \mu, \bar{\theta}_i) + (p_2)^2\theta_{2,2} = -\epsilon. \tag{4.18}$$

Lastly if $p_2 \geq 0$ and $y_1 \leq 0$, then we accept the sample (p_1, p_2, μ). In all other cases, we reject the sample. With this simple method, we can generate sample prices that are ϵ-approximate equilibrium. By repeating the above N times for each sampled ϵ_j, we can generate all the samples necessary for the numerical simulation.

4.2 Computational Experiments for Inverse Hyposthesis Testing

We analyze the performance of our approach in a Bertrand competition environment. We first detail the experiment setting and then proceed to the numerical experiments and analysis. We showcase our method in a setting where two agents compete over a single item and need to set their respective prices in the Bertrand-game environment. Each agent's true demand function has the following form:

$$\bar{D}_i(p_1, p_2, \mu; \bar{\theta}_i) = \bar{\theta}_{0,j} + \sum_{j=1}^{2} p_j \bar{\theta}_{i,j} + \bar{\theta}_{i,3}\mu + \eta_i \tag{4.19}$$

where $\bar{\theta}_i$ is the agent's private information vector, and we use the term η_i to encompass unmodeled terms of the dynamics. Furthermore, we assume the set of feasible price vectors belong to the polyhedral set

$$\mathcal{P} = \{(p_1, p_2) \in \mathbb{R}^2 : \ 0 \leq p_1 \leq \bar{p}, 0 \leq p_2 \leq \bar{p}\} \tag{4.20}$$

where $\bar{p}$ is an upper-bound on each price. We consider the case where ϵ is drawn from an exponential distribution $\epsilon \sim \exp(\bar{\lambda})$.

We assume that the regulator observes the shock μ but does not observe η_i. Hence, the regulator forms the following demand estimate given some estimate $\hat{\theta}$:

$$\mathbf{D}_i(p_1, p_2, \mu; \hat{\theta}) = \hat{\theta}_{0,j} + \sum_{j=1}^{2} p_j \hat{\theta}_{i,j} + \hat{\theta}_{i,3}\mu \tag{4.21}$$

Following the steps described in the previous section, the regulator collects the sample tuples $\{(p_1^j, p_2^j, \mu^j)\}_{j=1}^N$ and forms the optimization problem (4.8) with the loss function $L(\hat{\epsilon}^1, ..., \hat{\epsilon}^N)$ being the negative log-likelihood of the underlying exponential distribution. Note that in the negative log-likelihood the λ term is decoupled from the other terms because of the particular mathematical form of the density of an exponential distribution. As a result, we do not need to include λ in the inverse variational problem.

In order to make the presentation of the final optimization problem clear, we define the marginal utility function for each agent (as considered by the regulator) to be

$$m_i(p_1, p_2, \mu; \hat{\theta}_i) = p_i \frac{\partial}{\partial p_i} \mathbf{D}_i(p_1, p_2, \mu; \theta_i) + \mathbf{D}_i(p_1, p_2, \mu; \theta_i) = p_i \hat{\theta}_{i,i} + \hat{\theta}_{0,j} + \sum_{j=1}^{2} p_j \hat{\theta}_{i,j} + \hat{\theta}_{i,3}\mu. \tag{4.22}$$

In addition, we impose some structure to the fitted utility functions: (1) we normalize the fitted utility functions; (2) we enforce that the marginal utilities of each agent decrease as they increase their own prices (on the observed data); and (3) we enforce an additional

constraint that sets the dual variable y_i^j to zero if the observed price p_i^j is strictly less than the upper bound $\bar{p}$. Recalling the definition of $\mathbf{f}(p)$ in (4.4), the optimization problem becomes

$$\min_{\hat{\epsilon},y,\theta_1,\theta_2} \sum_{j=1}^{N} \hat{\epsilon_j} \tag{4.23}$$

$$\text{s.t. } y_i^j \geq m_i(p_1^j, p_2^j, \mu^j, \theta_1), \text{for } i \in \{1,2\}, j \in \{1,...,N\} \tag{4.24}$$

$$\bar{p}\sum_{i=1}^{2}(y_i^j) - \sum_{i=1}^{2} p_i^j m_i(p_1^j, p_2^j, \mu^j, \theta_i) = \hat{\epsilon}_j, \qquad \text{for } j \in \{1,...,N\} \tag{4.25}$$

$$m_i(1,1,0,\theta_i) = m_i(1,1,0,\bar{\theta}_i), \text{for } i \in \{1,2\} \tag{4.26}$$

$$y_i^j = 0, \text{for } i \in \{1,2\}, j \in \{1,...,N\} \text{ s.t. } p_i^j < \bar{p} \tag{4.27}$$

$$\theta_{i,i} \leq 0, \text{for } i \in \{1,2\} \tag{4.28}$$

$$\hat{\epsilon}^j \geq 0, \text{for } j \in \{1,...,N\} \tag{4.29}$$

$$y^j = (y_1^j, y_2^j) \geq 0, \text{for } \forall j \in \{1,...,N\} \tag{4.30}$$

where (4.26) are the normalization constraints, in which the marginal utility of both agents when there is no external shock and the prices are set to unity is equal to the true marginal at that point. (Note we could have set these normalization constraints to any other suitable positive value without affecting the results). Different normalization may yield different models that can be used to explain the same observed data. This phenomenon is common in inverse optimization problems, as discussed in detail in [40, 19].) Equation (4.28) ensures that the fitted marginal functions decrease as the agents increase their own prices. (This constraint is obtained after some arithmetic by requiring that $m_1(p_1, \cdot, \cdot; \theta_i)$ and $m_2(\cdot, p_2, \cdot; \theta_i)$ decrease as p_1 and p_2 increase, respectively, on the observed data.) The "dual" vector y is associated with the constraints 4.20), and (4.29) ensures that if the the observed prices are not on the boundary of the feasible region $\mathcal{P}$ then the associated dual variable is set to zero. We note that (4.29) has a very subtle implication in the optimization problem above: The very natural notion that dual variables are zero once their associated constraint is non-binding is not enforced at all by the original formulation in (4.8). If the prices are sampled in perfect Nash equilibrium (that is $\epsilon = 0$), then as argued in [40] the formulation in (4.8) is able to recover exactly the true parameters θ_i and the computed residuals are exactly zero. However, in our scenario prices are obtained in approximate equilibrium (i.e., $\epsilon > 0$). Hence if complimentary slackness (4.29) is not enforced explicitly then the computed residuals will present bias – namely they will be "shrunk" since the formulation (4.8) could achieve smaller values for the residuals by setting the dual variables to be positive, even though the sampled prices are in the interior of the feasible region.

Recall that the objective function follows from the negative log-likelihood of exponential distribution, where we dropped the term N/λ since it does not impact the optimization.

After solving the problem above, we compute the MLE estimate of λ

$$\hat{\lambda}_{MLE} = \frac{N}{\sum_{j=1}^{N} \hat{\epsilon}_j} \tag{4.31}$$

Then we let $\bar{F}_N(d)$ be defined as

$$\bar{F}_N(d) = F_{\exp(\hat{\lambda}_{MLE})}(d) \tag{4.32}$$

and conduct the Lilliefors hypothesis test (4.15). To illustrate the performance of the hypothesis testing we will simulate the process under two scenarios:

Scenario 1: Agents are competing over prices, i.e.: they solve the feasibility problem (4.7) after observing the shock variable μ and the value of ϵ.

Scenario 2: Agents are colluding, i.e.: instead of solving the feasibility problem (4.7), they maximize the sum of both utility functions up to a ϵ optimality gap.

Hence for Scenario 2, prices are generated after solving the following optimization problem:

$$(p_1^j, p_2^j) = \arg \max_{(p_1,p_2)\in\mathcal{P}} \sum_{i=1}^{2} p_i D_i(p_1, p_2, \theta_i, \mu_j), \text{for } j \in \{1, ..., N\} \tag{4.33}$$

In the next subsection, we present numerical simulations of these two scenarios and show how the regulator rejects/does not reject the null hypothesis as the agents change their behavior from competition to collusion.

Computational results

For the numerical simulations, we let $\bar{\lambda} = 20$. We chose $\bar{\theta}_1 = [10, -1, 0.5, 1]$ and $\bar{\theta}_2 = [8, 0.4, -3.0, 1]$ to be the agents' true private information vectors. The shock values were generated according to $\mathcal{N}(5, 1)$, and we fix the upper-bound $\bar{p} = 8.0$ on the prices. Furthermore, we fix our significance level $\alpha = 0.05$. The threshold $\tau(N)$ for the hypothesis testing is obtained by the table presented in [154]. Lastly, we let η_j for $j = \{1, 2\}$ be sampled from $\mathcal{N}(0, 1)$. For the first scenario, the approximate equilibrium prices need to be generated by solving (4.7). That is hard problem in general, but in our test case we are able to generate approximate equilibrium prices via a primal-dual algorithm described in the appendix. The results for Scenario 1 are summarized in Table 4.1.

It can be observed that when agents are competing (i.e., acting under the specifications of the null hypothesis), a false positive (i.e., decision of collusion occurring) was not seen in the experiments. This is not surprising because we set $\alpha = 0.05$ and each row in the table corresponds to a single numerical experiment. If we were to run a large number of repeated experiments, we would expect to see a close to α fraction of them report a false positive. In addition, we can observe that we are able to recover the correct estimate of λ for the

Table 4.1: Numerical Results for Scenario 1 (Competing)

N	D^*	$\tau(N)$	$\hat{\lambda}$	Decision
10	0.317	0.325	33.7	Competing
20	0.206	0.234	27.93	Competing
30	0.120	0.192	20.83	Competing
40	0.069	0.168	21.43	Competing
50	0.089	0.150	19.99	Competing
100	0.070	0.106	18.80	Competing
200	0.031	0.075	18.62	Competing
500	0.022	0.047	20.01	Competing

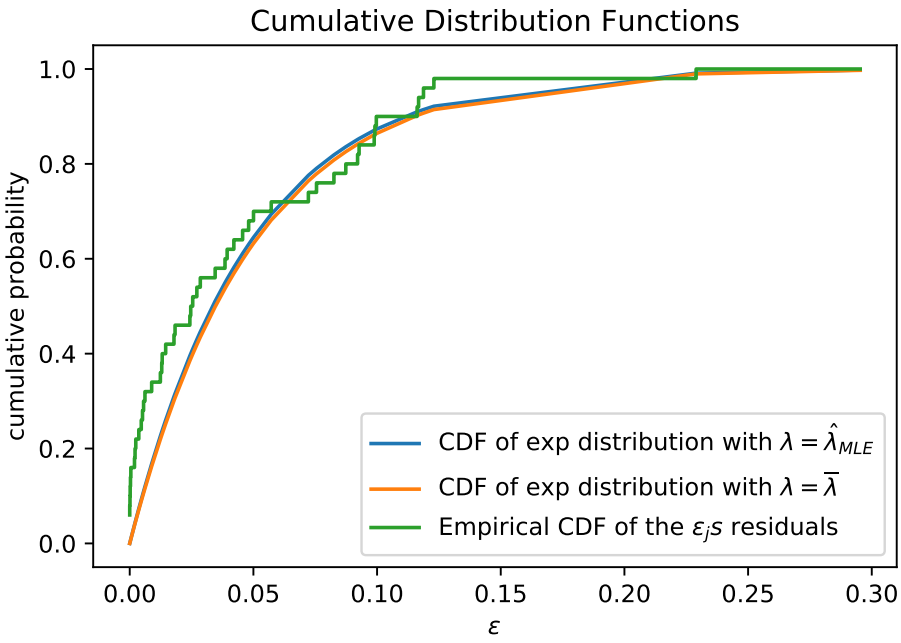

Figure 4.1: Comparing CDF's of Residuals For Scenario 1

underlying distribution generating the residuals. This is highlighted in Figure 4.1, where we plot the $\epsilon_j's$ samples from exp(20) and the computed residual estimates $\hat{\epsilon}_j's$ by the regulator after solving the optimization problem, for sample size equal to 50.

Now for the second scenario, we generate the prices by solving an aggregate problem where we sum both agents' utilities in order to compute the prices. In Table 4.2, we can see that the null hypothesis is rejected (i.e., decision of collusion occurring) for moderate

Table 4.2: Numerical Results for Scenario 2 (Colluding)

N	D^*	$\tau(N)$	$\hat{\lambda}$	Decision
10	0.261	0.325	0.12	Competing
20	0.263	0.234	0.11	Colluding
30	0.222	0.192	0.22	Colluding
40	0.300	0.168	0.26	Colluding
50	0.301	0.150	0.22	Colluding
100	0.322	0.106	0.28	Colluding
200	0.301	0.075	0.30	Colluding
500	0.335	0.047	0.30	Colluding

and large sample sizes. In addition, the MLE estimate of $\bar{\lambda}$ is inaccurate as well since the agents are not behaving in approximate equilibrium. In Figure 4.2, for $N = 50$ we plot the empirical CDF of residual estimates $\hat{\epsilon}'_j s$ by the regulator in this scenario. We can observe that when agents are cooperating instead of competing, the computed residuals are vastly different then their true values (we omit plotting the true cdf of exp(20) since the computed residuals are very large for this scenario). The null hypothesis that the agents are competing in equilibrium is rejected for almost all sample sizes, indicating that our method is able to identify when agents are not behaving in competition. We stress that rejecting the null hypothesis is not proof that agents are colluding, but rather gives some statistical evidence that suggests collusion is occurring.

4.3 Surrogate Optimal Control for Strategic Multi-Agent Systems

On his section we study how to design a platform to optimally control constrained multi-agent systems with a single coordinator and multiple strategic agents. Many systems have dynamics influenced by agents, including power systems [245], communication networks [140], water systems [180], and heating, ventilation, and air-conditioning (HVAC) automation [20]. These systems are characterized by information flows and the order of computations. For cooperative agents, various distributed model predictive control (MPC) schemes have been designed. A system-level control policy was obtained by aggregating locally-computed inputs [88, 10], and central platforms that compute a control based upon information sent by agents have also been designed [93].

Distributed control with strategic agents is less well-studied. The competitive nature of agents and asymmetries of information reward tactical behavior, ultimately leading to instability or poor performance [244, 212, 167, 136]. We focus our attention on the case

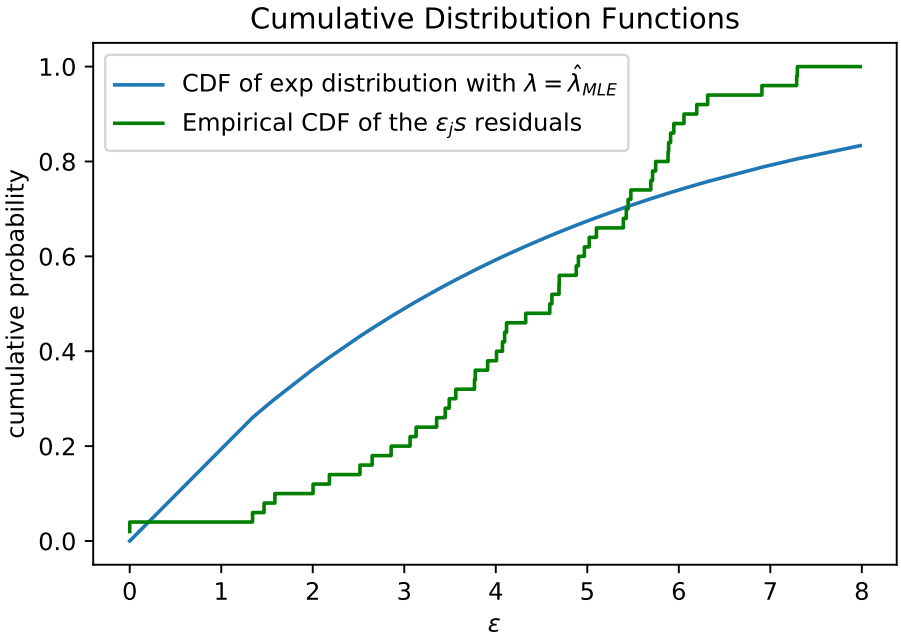

Figure 4.2: Comparing CDF's of Residuals For Scenario 2

where equilibrium behavior can be described as a Nash equilibrium of some non-cooperative game [179] that may be inefficient [68]. A common way to overcome such inefficiencies is to force agents to coordinate their goals with the system-wide goal [161], [152]. However, this approach requires strong assumptions that the agents' utility functions are common knowledge and/or agents are honest when transmitting information [237]. Another line of work [210, 72] provides pricing schemes to induce or manage agents' behavior in the equilibrium.

Low-dimensional communication in Mechanism Design

In this section, we study the case of strategic agents under weaker assumptions than past work like [227, 237]. In particular, the agents exchange information only with a central platform that is responsible for the control decision. Our goal is to design the interaction mechanism to ensure not only efficiency of the resulting control policy but also honest reporting from the agents. Originally, the study of such mechanisms [139] was concerned with the design of incentives to ensure efficient allocation of commodities amongst market participants, whilst ensuring truthfulness. The classical VCG mechanisms [246, 67, 115] are an example of such. Our first contribution lies in providing a mechanism that enjoy those properties when applied to an optimal control setting.

A major hurdle in implementing such mechanisms is their steep communication needs [254, 136, 87]. But, minimal strategy spaces that elicit efficient Nash equilibrium in convex

environments have been developed [213]. A second contribution of our work is to provide communication protocols that are of low complexity order: We avoid communicating the entire utility function by the agents and instead resort to vector-valued messages inspired by surrogate optimization [254], [137]. Hence, our goal in this paper is to provide a platform where (i) agents provide low-dimensional information, (ii) agents are honest, and (iii) an efficient control policy is implemented in the Nash equilibrium.

Lastly, we demonstrate the practical usefulness of our designed platform by conducting a simulation analysis of HVAC automation [158]. The situation we consider involves an apartment building where each apartment has its own preferences on desired room temperature versus the amount of energy consumption. The thermal dynamics of each apartment are coupled, and more efficient control is possible through coordination. Our simulations quantify the performance improvement possible through the use of our central platform in coordinating agents. In fact, this HVAC setup is similar to the setup in [72]. However, a major difference is that in [72] the central platform knows each agents' utility function and can set prices on the control inputs to induce agents' behavior. In contrast, we allow the agents to be strategic with respect to how they communicate information about their utility function to the central platform.

We first define the system model and the mechanism and how agents interact with it. Next, we provide a Nash equilibrium characterization of the agents' equilibrium behavior, rigorously establishing the existence of the Equilibrium solution and how it can be obtained.

System Model

Consider a system which obeys linear dynamics

$$x_{k+1} = Ax_k + Bu_k \tag{4.34}$$

where $x_k \in \mathbb{R}^n$ is the state vector, and $u_k \in \mathbb{R}^m$ is the input signal. Suppose this system is composed of I interconnected and non-overlapping subsystems that are each associated to an agent. Let $[I]$ denote the set $\{0, ..., I\}$. We let $x_k^{(i)} \in \mathbb{R}^{n_i}$ denote the state vector of subsystem i at period k. Then we have $x_k = (x_k^{(1)}, ..., x_k^{(I)})$ and $\sum_{i=1}^I n_i = n$. In addition, we can also partition the inputs where $u_k^{(i)} \in \mathbb{R}^{m_i}$. Note it follows that $u_k = (u_k^{(1)}, ..., u_k^{(I)})$ and $\sum_{i=1}^I m_i = m$.

The diagonal block A_{ii} of A gives the subsystem dynamics for the i-th agent. Influence by other agents is described by off-diagonal blocks A_{ij} of A when subsystem j impacts i. We assume agent i's input only affects states in their subsystem; hence, the input matrix $B = \text{diag}(B_1, ..., B_I)$ is block-diagonal. Let $\mathcal{N}_i$ be the set of neighboring subsystems of subsystem i. Then the dynamics for the i-th subsystem is

$$x_{k+1}^{(i)} = A_{ii}x_k^{(i)} + B_i u_k^{(i)} + \sum_{j \in \mathcal{N}_i} A_{ij}x_k^{(j)}. \tag{4.35}$$

So we recover Eq.(4.34) by stacking Eq.(4.35) for all agents. We assume each agent only knows their own local dynamics A_{ii} and B_i. Since agents do not know the $\sum_{j \in \mathcal{N}_i} A_{ij}x_k^{(j)}$

part of their dynamics, a central platform is needed through which each agent can receive this information.

Each subsystem has state $\mathbb{X}_i = \{G_x^{(i)} x^{(i)} \leq g_x^{(i)}\}$ and input constraints $\mathbb{U}_i = \{G_u^{(i)} u(i) \leq g_u^{(i)}\}$ that are polytopes containing the origin. Here, $\{G_x^{(i)}, G_u^{(i)}\}_{i=1}^I, \{g_x^{(i)}, g_u^{(i)}\}_{i=1}^I$ are matrices and vectors with appropriate dimensions, respectively. Lastly, each agent i has their own cost function

$$\mathbf{V}_i(x^{(i)}, u^{(i)})) = g_i(x_T^{(i)}) + \sum_{k=0}^{T-1} l_i(x_k^{(i)}, u_k^{(i)}) \tag{4.36}$$

where $l_i(\cdot,\cdot)$ and $g_i(\cdot)$ are stage and terminal costs, and T is control horizon. We assume each agent's stage and terminal costs are strictly convex, differentiable, and take their minimum at the origin. The cost function is the agents private information, and their goal is to minimize it. Next we define the Principal, which is the plataform/regulator with which the agents interact and communicate.

Principal Model

The central platform is operated by a coordinator that we call the *principal*. We assume the principal has complete knowledge about the dynamics of the system (i.e., matrices A and B) and constraints (i.e., the sets $\mathbb{X}_i, \mathbb{U}_i, \forall i \in [I]$), and importantly the principal is who gets to apply a control input to the entire system (restated, the agents do not directly provide control inputs). In this framework, if the principal knew the objective function of each agent, then they could compute a control sequence by solving the following convex optimal control problem (OCP-T):

$$\begin{aligned}
&\min_{x,u} \ \textstyle\sum_{i=1}^{M} (g_i(x_T^{(i)}) + \sum_{k=0}^{T-1} l_i(x_k^{(i)}, u_k^{(i)})) \\
&\text{s.t. } x_{k+1} = Ax_k + Bu_k, \quad \forall k \in [T-1] \\
&\qquad x_k^{(i)} \in \mathbb{X}_i, u_k^{(i)} \in \mathbb{U}_i, \quad \forall i \in [I], k \in [T] \\
&\qquad x_0^{(i)} = \bar{x}_0^{(i)}, \qquad\qquad \forall i \in [I]
\end{aligned} \tag{4.37}$$

Throughout the paper we let (x^*, u^*) denote the optimal solution of (OCP-T), which we call the *efficient trajectory*. However, solving this problem is not possible for the principal, since it does not know the objective functions of each agent. It then needs to elicit information from each agent. The need of information gives birth to two major issues, which are the central focus of this work: (1) The agents may not be able/not desire to transmit their entire cost functions to the principal, as each cost function is infinite-dimensional and their private information; (2) The agents are strategic and may not tell the truth. Therefore in order for the principal to solve (OCP-T) it also needs to design a mechanism that provides incentives to each agent to tell the truth. Hence, the principal is faced with both an optimal control problem and a mechanism design problem.

Mechanism Specification

As described in the previous section, the principal's goal is to solve (OCP-T). Towards that goal, the principal resorts to approximate the objective function based on a finite number of parameters that the agents can report, and then the principal will minimize this approximated function.

Let a *mechanism* $\mathcal{M}$ be a tuple $(M_1, ..., M_I, z, p)$, where M_i is the set of allowable messages agent i can send to the principal, and $z(\cdot)$ is the outcome function that determines the outcome $z(m)$ for any message profile $m = (m_1, ..., m_I) \in M_1 \times ... \times M_I$. Here, the outcome function maps a message profile m to a state/input trajectory (x, u):

$$z(m) : (M_1 \times ... \times M_I) \to \mathbb{R}^{n\times(T+1)} \times \mathbb{R}^{m\times T} \tag{4.38}$$

where $z_i(m)$ refers to the state/input trajectory associated with agent's i subsystem. Next, we define $p(m)$ to be a non-negative vector of "fees" for each agent.

The mechanism $\mathcal{M}$ together with the cost functions of each agents $(\mathbf{V}_i)_{i=1}^I$ induce a game $\mathbf{N} = (\mathcal{M}, (\mathbf{V}_i)_{i=1}^I)$ among the agents. We define the Nash equilibrium (NE) of this game as a message profile m^* such that

$$\mathbf{V}_i(z_i(m_i^*, m_{-i}^*)) + p_i(m_i^*, m_{-i}^*) \leq \mathbf{V}_i(z_i(m_i, m_{-i}^*)) + p_i(m_i, m_{-i}^*), \tag{4.39}$$

for all $m_i \in M_i$ and $i \in [I]$, where the compact notation m_{-i}^* denotes the vector of messages from all agents except i. The fee p_i increases costs for agent i, which is undesirable since agents are minimizing. The goal of the principal is to design the mechanism such that the efficient trajectory (x^*, u^*) can be implemented as the Nash equilibrium of the game $\mathbf{N}$. Implementation means that the trajectory corresponding to the Nash equilibrium of the game induced by the mechanism is equal to the efficient trajectory.

Low-Communication Mechanism

We specify our low-communication mechanism as follows: Each agent i reports messages $m_i \in M_i$ of the form

$$m_i = (v^{(i)}, w^{(i)}, \tilde{\lambda}^i, \tilde{J}^{(i)}, \tilde{x}^{(i)}) \tag{4.40}$$

where $v^{(i)} \in \mathbb{R}^{n_i\times(T+1)}$ are weights for every state of subsystem i for each stage; $w^{(i)} \in \mathbb{R}^{m_i\times T}$ are weights for every control input of subsystem i for each stage; $\tilde{\lambda}^i \in \mathbb{R}^{n_i\times T}$ are weights representing the "sensitivity" of agent i dynamics in cost function for each stage; $\tilde{J}^{(i)} = (\{\underline{x}_k^{(i)}, \bar{x}_k^{(i)}\}_{k=0}^T, \{\underline{u}_k^{(i)}, \bar{u}_k^{(i)}\}_{k=0}^T)$ is vector of bounds for states/inputs; and $\tilde{x}^{(i)} = (\tilde{x}_0^{(i)}, ..., \tilde{x}_T^{(i)})$ is a reference trajectory for the states of subsystem i. Restated, each agent provides some open-loop trajectory coupled with state and input bounds, as well as scalars measuring the "impact" of states, inputs, and dynamics in its cost function. We highlight the fact that the message sent by the agents is a vector of finite dimension instead of a function, which is in line with previous work on low-communication mechanisms [87].

In addition, the principal announces a single real-valued function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to all agents to be used as a surrogate function for their cost functions. Namely, for each agent i the principal forms the surrogate function

$$f_i(v_k^{(i)}, w_k^{(i)}, x_k^{(i)}, u_k^{(i)}) = \sum_{j=1}^{m_i} (f(x_{j,k}^{(i)}; v_{j,k}^{(i)}) + f(u_{j,k}^{(i)}; w_{j,k}^{(i)})) \text{ for } k \in [T-1] \tag{4.41}$$

as an approximation of the agent's stage cost $l_i(\cdot, \cdot)$. The notation $f(\cdot; \cdot)$ indicates the second argument is a parameter of the function and not a variable. We further only consider functions $f(\cdot; v)$ that are strictly convex for all possible parameters $v \in \mathbb{R}$. Lastly, for simplicity we let the principal announce the same function for both states and inputs. But one could consider different functions for states and inputs – the key property being that it is the same function for all agents. Then the principal forms the surrogate function

$$F_i(v(i)_T, x_T^{(i)}) = \sum_{j=1}^{m_i} f(x_{j,T}^{(i)}; v_{j,T}^{(i)}) \tag{4.42}$$

as an approximation of agent's terminal cost $g_i(\cdot)$. Based on a message profile m, the principal formulates the following surrogate optimal control problem (OCP-S):

$$\begin{aligned} \min_{x,u} \ & \sum_{i=1}^{I} (F_i(v(i)_T, x_T^{(i)}) + \sum_{k=0}^{T-1} f_i(v_k^{(i)}, w_k^{(i)}, x_k^{(i)}, u_k^{(i)})) \\ \text{s.t. } & x_{k+1} = Ax_k + Bu_k, && \forall k \in [T-1] \\ & x_k^{(i)} \in \mathbb{X}_i, u_k^{(i)} \in \mathbb{U}_i, && \forall i \in [I], k \in [T] \\ & (x_k^{(i)}, u_k^{(i)}) \in \tilde{J}_k^{(i)}, && \forall k \in [T] \\ & x_0^{(i)} = \bar{x}_0^{(i)}, && \forall i \in [I] \end{aligned} \tag{4.43}$$

where we explicitly consider the desired operational bounds reported by each agent $\tilde{J}_k^{(i)}$ for every stage k.

Since f is strictly convex, this optimization problem has an unique solution that we call $(x(y^*), y^*)$. Note that this notation means the y^* are the optimal inputs for OCP-S. Then, given a message profile m, we have that $z(m) = (x(y^*), y^*)$. That is, the outcome function of the mechanism $z(m)$ outputs exactly the state/input trajectory of the optimal solution of OCP-S. We also define $\lambda^{*(i)}$ to be the optimal lagrange multipliers associated with Eq.(4.35) for every agent i. Now, suppose the principal solved OCP-S once successfully. The principal sends the following reference trajectory $c^{(i)}$ to agent i:

$$c_k^{(i)} = \sum_{j \in \mathcal{N}_i} A_{ij} \tilde{x}_k^{(j)}, \forall k \in [T-1] \tag{4.44}$$

where $\tilde{x}^{(j)}$ is part of the message m_j as per (4.40). Observe that the reference trajectory sent to agent i does not depend upon solving OCP-S. Moreover the principal will assign the following fees to each agent:

$$p_i = \sum_{k=0}^{T-1} \Lambda_{-i,k}^{\top}(x_k^{(i)}(y^*) - \hat{x}_k^{(i)}(c^{(i)})) + ||\tilde{x}^{(i)} - x^{(i)}(y^*)||_2^2 + ||\tilde{\lambda}^{(i)} - \lambda^{*(i)}||_2^2 \tag{4.45}$$

where $\hat{x}^{(i)}(c^{(i)})$ is a state reference trajectory computed by the principal for agent i given that the other agents behave according to $c^{(i)}$. For example, the principal can solve another round of OCP-S but now excluding agent i's contribution to the objective function in order to obtain $\hat{x}^{(i)}(c^{(i)})$ (in a way akin to VCG mechanisms [115]). The key observation here is that the reference trajectory $\hat{x}^{(i)}(c^{(i)})$ does not depend on the message sent by agent i. The first term of the fee penalizes deviations of the computed optimal state trajectory $x^{(i)}(y^*)$ from $\hat{x}^{(i)}(c^{(i)})$. The second term penalizes mismatches between the reported $\tilde{x}^{(i)}$ and the optimal state trajectory $x^{(i)}(y^*)$. The third term penalizes deviations from the reported sensitivity vector $\tilde{\lambda}^{(i)}$ and the optimal lagrange multipliers $\lambda^{*(i)}$ of OCP-S associated with the dynamics of agent i. Lastly the vectors $\Lambda_{-i,k}$ are computed by the principal as follows:

$$\Lambda_{-i,k}^{\top} = \sum_{j:i\in\mathcal{N}_j} \tilde{\lambda}_k^{(j)\top} A_{ji},\ \forall k \in [T-1] \tag{4.46}$$

where we, once again, note that this vector does not depend on the message sent by agent i. We highlight the special purpose of the terms $c^{(i)}$: The only way agent i can infer the impact of it's neighbours in their subsystem is by the reported $c^{(i)}$ via the principal. Likewise that information is used by the principal in computing the fees imputed to agent i. The passing of not only "monetary transfers" but also "trajectory information" by the principal to the agents is essential in proving the implementability of the Nash equilibrium in the next section. For clarity, we stress that the reference trajectoty $c^{(i)}$ sent to agent i does **not** depend on the message sent by agent i.

Equilibrium Characterization

With the mechanism defined, we can now characterize the equilibrium behavior of agents interacting via this mechanism. The goal of this section is to characterize the Nash equilibrium (NE) of the resulting game, which is a message profile m^*. Our analysis begins by showing that in a NE m^*, each agent i reports a specific type of state reference trajectories to the principal.

Lemma 4.3.1. *Let $m^* = (v^*, w^*, \tilde{\lambda}^*, \tilde{J}^*, \tilde{x}^*)$ be a NE of the game induced by the mechanism. Then every agent $i \in I$ reports $\tilde{x}^{(i)*} = x^{(i)}(y^*)$ and $\tilde{\lambda}^{(i)*} = \lambda^{(i)*}$. In addition, the principal sends the following references to the agents:*

$$c_k^{*(i)} = \sum_{j\in\mathcal{N}_i} A_{ij} x_k^{(j)}(y^*),\ \forall k \in [T-1] \tag{4.47}$$

Proof. Suppose all agents adhere to the message profile m^*, except agent i which reports some message $m_i = (v^{(i)}, w^{*(i)}, \tilde{\lambda}^{(i)}, J^{*(i)}, \tilde{x}^{(i)})$. Since m^* is a NE, this deviation should give a higher cost for agent i, that is:

$$\mathbf{V}_i(z_i(m_i, m^*_{-i})) + p_i(m_i, m^*_{-i}) \geq \mathbf{V}_i(z_i(m^*_i, m^*_{-i})) + p_i(m^*_i, m^*_{-i}) \tag{4.48}$$

Now, observe that the outcome function $z(m)$ only depends on the $(v^*, w^*, \tilde{J}^*)$ components of the message m^*. Then substituting into (4.45) gives that

$$||\tilde{\lambda}^{(i)} - \lambda^{*(i)}||_2^2 + ||\tilde{x}^{(i)} - x^{(i)}(y^*))||_2^2 \geq ||\tilde{\lambda}^{*(i)} - \lambda^{*(i)}||_2^2 + ||\tilde{x}^{*(i)} - x^{(i)}(y^*))||_2^2$$

for all possible sensitivities and state trajectory reports $(\tilde{\lambda}^{(i)}, \tilde{x}^{(i)})$. Hence $(\tilde{\lambda}^{*(i)}, \tilde{x}^{*(i)})$ is the solution of the following minimization problem:

$$\min_{(\tilde{\lambda}^{*(i)}, \tilde{x}^{*(i)})} \{||\tilde{\lambda}^{(i)} - \lambda^{*(i)}||_2^2 + ||\tilde{x}^{(i)} - x^{(i)}(y^*)||_2^2\} \tag{4.49}$$

which achieves the minimum when $(\tilde{\lambda}^{*(i)}, \tilde{x}^{*(i)}) = (\lambda^{*(i)} x^{(i)}(y^*))$. Then by definition of $c^{(i)}$ it directly follows that

$$c_k^{*(i)} = \sum_{j \in \mathcal{N}_i} A_{ij} x_k^{(j)}(y^*), \, \forall k \in [T-1] \tag{4.50}$$

□

Next, observe that each agent can only "measure" the impact of other subsystems in its dynamics via the reference signal $c^{(i)}$ that is sent by the principal. We say an state/input sequence $(\check{x}^{(i)}, \check{u}^{(i)})$ is feasible for agent i if it is feasible for the agent's subsystem given the reference $c^{(i)}$. We proceed to show that given the reference $c^{(i)}$, any feasible state/input sequence $(\check{x}^{(i)}, \check{u}^{(i)})$ can be achieved by agent i. That is, agent i can send a message that makes the principal compute the input $y^{*(i)} = \check{u}^{(i)}$ and $x^{(i)}(y^*) = \check{x}^{(i)}$ as it solves the problem OCP-S, given that OCP-S is feasible.

Lemma 4.3.2. *For any agent i, given a feasible state/input sequence $(\check{x}^{(i)}, \check{u}^{(i)})$ there exists a message $\bar{m}_i$ such that $z_i(\bar{m}_i, m_{-i}) = (\check{x}^{(i)}, \check{u}^{(i)})$ for all possible messages of the other agents m_{-i}, given that the resulting optimization problem (OCP-S) is feasible for $(\bar{m}_i, m_{-i})$.*

Proof. Fix some agent i and a feasible state/input sequence $(\check{x}^{(i)}, \check{u}^{(i)})$. We prove this lemma by constructing the message $\bar{m}_i$. Specifically, suppose agent i chooses $\underline{x}_k^{(i)} = \bar{x}_k^{(i)} = \check{x}_k^{(i)}$ and $\underline{u}_k^{(i)} = \bar{u}_k^{(i)} = \check{u}_k^{(i)}$. This choice constrains OCP-S to require that $y^*(i) = \check{u}^{(i)}$ and $x^{(i)}(y^*) = \check{x}^{(i)}$. Then for any message m_{-i}, OCP-S is either infeasible or returns the desired solution for agent i, regardless of the message of other agents. □

What this lemma implies is that given the Nash equilibrium message profile m^*, agent i can unilaterally deviate in such a way that the principal will compute $(\check{x}^{(i)}, \check{u}^{(i)})$ as part of

the optimal solution, as long as OCP-S is feasible. We proceed in writing the agent's optimal control problem (OCP-A) in equilibrium:

$$\begin{aligned}
&\min_{x^{(i)},u^{(i)}} g_i(x_T^{(i)}) + \textstyle\sum_{k=0}^{T-1} l_i(x_k^{(i)}, u_k^{(i)}) + p_i^*(x^{(i)}) \\
&\text{s.t. } x_{k+1}^{(i)} = A_{ii}x_k^{(i)} + B_i u_k^{(i)} + c_k^{*(i)}, \quad \forall k \in [T-1] \\
&\qquad x_k^{(i)} \in \mathbb{X}_i, u_k^{(i)} \in \mathbb{U}_i, \qquad\qquad \forall k \in [T] \\
&\qquad x_0^{(i)} = \bar{x}_0^{(i)}
\end{aligned} \tag{4.51}$$

where the equilibrium fee, according to Lemmas 1 and 2 is given by:

$$p_i^*(x^{(i)}) = \sum_{k=0}^{T-1} \Lambda_{-i,k}^{*\top}(x_k^{(i)} - \hat{x}_k^{(i)}(c^{(i)})) \tag{4.52}$$

where $\Lambda_{-i,k}^{*\top} = \sum_{j:i\in\mathcal{N}_j} \lambda_k^{*(j)\top} A_{ji}$, $\forall k \in [T-1]$.

Thus in order for a message profile m^* to be a NE, we must have that the optimal solution $(x(y^*), y^*)$ for OCP-S must also be an optimal solution for each agents' OCP-A. Since both OCP-S and OCP-A are convex problems, it is enough to require that $(x^{*(i)}(y^*), y^{*(i)})$ satisfy the KKT conditions for OCP-A for every agent i. Next we present our main theorem, which shows that the efficient trajectory (x^*, u^*) can be implemented as a Nash equilibrium of the game induced by the mechanism:

Theorem 4.3.3. ***(Implementability):*** *Let (x^*, u^*) be the unique efficient trajectory. Let m^* be a message satisfying the following:*

$$\begin{aligned}
&f'(x_{j,k}^{*(i)}; v_{j,k}^{*(i)}) = \frac{\partial l_i(x_k^{*(i)}, u_k^{*(i)})}{\partial x_{j,k}^{(i)}}, && \forall i,j,k \\
&f'(x_{j,T}^{*(i)}; v_{j,T}^{*(i)}) = \frac{\partial g_i(x_T^{*(i)})}{\partial x_{j,T}^{(i)}}, && \forall i,j \\
&f'(u_{h,k}^{*(i)}; w_{h,k}^{*(i)}) = \frac{\partial l_i(x_k^{*(i)}, u_k^{*(i)})}{\partial u_{h,k}^{(i)}}, && \forall i,h,k \\
&\tilde{x}^{(i)} = x^{*(i)},\ \tilde{\lambda}^{(i)} = \lambda^{*(i)} && \forall i \\
&\tilde{J}^{(i)} = (\{-\infty,+\infty\}_{k=0}^T, \{-\infty,+\infty\}_{k=0}^T), && \forall i
\end{aligned} \tag{4.53}$$

where $f'(\cdot)$ denotes the derivative of $f(\cdot)$. Then (x^*, u^*) can be supported as a Nash equilibrium of the game induced by the mechanism, that is $(x(y^*), y^*) = (x^*, u^*)$. In addition m^* is the equilibrium message sent by the agents.

Proof. First, note OCP-T is an "aggregation" of each agent's problem: Instead of optimizing each agent separately with references $c^{(i)}$ for the neighbors, we optimize all agents at once.

The KKT stationarity conditions for multipliers $(\nu, \gamma_x, \gamma_u)$ of OCP-T, associated with the dynamics, state and input constraints respectively, are

$$\begin{aligned}
&\frac{\partial l_i(x_k^{*(i)}, u_k^{*(i)})}{\partial x_{j,k}^{(i)}} + \nu_k^{(i)\top} A_{ii,j} - \nu_{k-1,j}^{(i)} + \gamma_x^{(i)\top} G_{x,j,k}^{(i)} + \sum_{\ell \in \mathcal{N}_i} \nu_k^{(\ell)\top} A_{\ell i,j} = 0 \\
&\frac{\partial g_i(x_T^{*(i)})}{\partial x_{j,k}^{(i)}} - \nu_{T-1,j}^{(i)} + \gamma_x^{(i)\top} G_{x,j,T}^{(i)} = 0 \\
&\frac{\partial l_i(x_k^{*(i)}, u_k^{*(i)})}{\partial u_{h,k}^{(i)}} + \nu_k^{(i)\top} B_{ii,h} + \gamma_u^{(i)\top} G_{u,h,k}^{(i)} = 0
\end{aligned} \tag{4.54}$$

for all $j \in \{1, \ldots, n_i\}$, $h \in \{1, \ldots, m_i\}$ and $k \in \{1, ..., T-1\}$. On the above we use the notation $A_{li,j}$ to denote the column j of matrix A_{li}. In addition we let $B_{ii,h}$ denote the column h of B_{ii}. Similarly, $G_{\cdot,j,k}$ represents the column j of the stage k constraints matrix $G_{\cdot,k}$. Now, if the messages follow (4.53), then it is easy to see that (x^*, u^*) satisfy the KKT conditions of OCP-S:

$$\begin{aligned}
&f'(x_{j,k}^{*(i)}; v_{j,k}^{*(i)}) + \lambda_k^{(i)\top} A_{ii,j} - \lambda_{k-1,j}^{(i)\top} + \beta_x^{(i)\top} G_{x,j,k}^{(i)} + \sum_{\ell \in \mathcal{N}_i} \lambda_k^{(\ell)\top} A_{\ell i,j} = 0 \\
&f'(x_{j,T}^{*(i)}; v_{j,T}^{*(i)}) - \lambda_{T-1,j}^{(i)} + \beta_x^{(i)\top} G_{x,j,T}^{(i)} = 0 \\
&f'(u_{h,k}^{*(i)}; w_{h,k}^{*(i)}) + \lambda_k^{(i)\top} B_{ii,h} + \beta_u^{(i)\top} G_{u,h,k}^{(i)} = 0
\end{aligned} \tag{4.55}$$

for all $j \in \{1, \ldots, n_i\}$, $h \in \{1, \ldots, m_i\}$, $k \in \{1, ..., T-1\}$ and for $\lambda = \nu$, $\gamma_x = \beta_x$ and $\gamma_u = \beta_u$. But in the equilibrium, Lemma 1 says that the reference trajectory $c^{*(i)}$ sent to each agent is exactly the one that would be obtained if each agent applied the input sequent $y^{*(i)}$. Hence (x^*, u^*) solves, not only OCP-S, but also each agent's problem when $c^{*(i)}$ is sent to the agents (OCP-A). This can be seen directly by using the multipliers $\lambda^{(i)}$, $\gamma_x^{(i)}$ and $\gamma_u^{(i)}$ for every agent's subproblem and verifying that $(x^{*(i)}, u^{*(i)})$ solves the KKT conditions of OCP-A. As a result, no agent has incentive to deviate from m_i^*. Hence m^* will be a Nash equilibrium of the game induced by the mechanism. □

The above theorem can be viewed as follows: The mechanism is nash incentive compatible, as truthful reporting of the agent's own subsystem trajectory and marginal utility information is a nash equilibrium strategy. We finish this section with some remarks on Theorem 1:

At equilibrium, each agent reports the largest possible bounds $\tilde{J}^{*(i)}$ so that OCP-S is always feasible at equilibrium. One may argue why do we include such reports in the message vector? Their presence is key to establishing Lemma 2, as they provide a "credible threat" to the mechanism (and thus to other agents). This forces that the solution $(x(y^*), y^*)$ of OCP-S must solve each agent's subproblem at equilibrium. A similar argument with a numerical example is given in [87] in the context of routing.

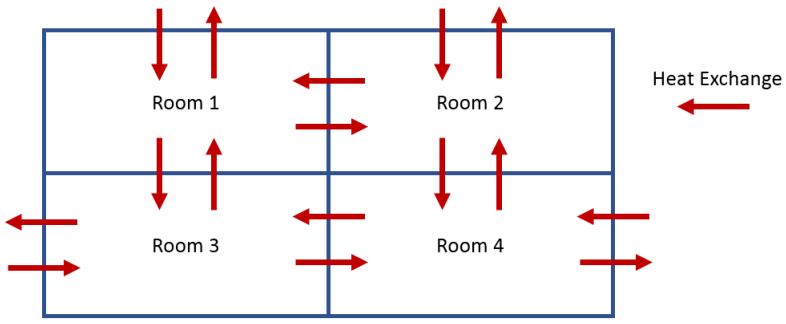

Figure 4.3: Room Configuration with Heat Exchange Vectors highlighted

Lastly, OCP-S may be infeasible outside of equilibrium, since an agent could report an infeasible operational range. This issue can be overcome by assuming that the principal may apply some feasible control input if OCP-S ends up being infeasible. More importantly, in order for the agents to behave according to the equilibrium strategies, they need to know the optimal solution (x^*, u^*) for OCP-T. This means that the agents need to "learn" the equilibrium by replaying the game and refining their messages. In the next section, we will provide one such simple learning process and, instead of theoretically proving its convergence to the Nash equilibrium defined in Theorem 1, we will present a test case on HVAC control in an MPC setting, where the game is replayed consecutively, but at each time, the initial condition $\bar{x}_0$ is different. This showcases the potential use of our mechanism when a learning protocol is used within the MPC framework.

4.4 HVAC Control Case Study

Consider a building manager who controls the HVAC system for four rooms. Each room occupant is an agent. Let $T_k = [T_k^1, T_k^2, T_k^3, T_k^4]^\top$ the state be the room temperatures. The building manager can heat/cool each individual room: Let $u_k = [u_k^1, u_k^2, u_k^3, u_k^4]^\top$ be the inputs in each room. Fig. 4.3 shows the layout of the rooms with respect to each other. Using standard HVAC models [21], the dynamics are

$$T_{k+1} = \begin{bmatrix} \rho_1 & -\beta & -\gamma & 0 \\ -\beta & \rho_2 & 0 & \eta \\ -\gamma & 0 & \rho_3 & -\nu \\ 0 & -\eta & -\nu & \rho_4 \end{bmatrix} T_k + \mu u_k - \alpha \begin{bmatrix} T_k^{out} \\ T_k^{out} \\ T_k^{out} \\ T_k^{out} \end{bmatrix} \tag{4.56}$$

where $\rho_1 = 1+\alpha+\beta+\gamma$; $\rho_2 = 1+\alpha+\beta+\eta$; $\rho_3 = 1+\alpha+\gamma+\nu$; $\rho_4 = 1+\alpha+\eta+\nu$; and β, γ, η, ν are the heat transmission coefficients between rooms; and α is the heat coefficient with the outside. In addition, μ is the heat coefficient between the HVAC and each room. Note we treat the outside temperature as an exogenous disturbance vector.

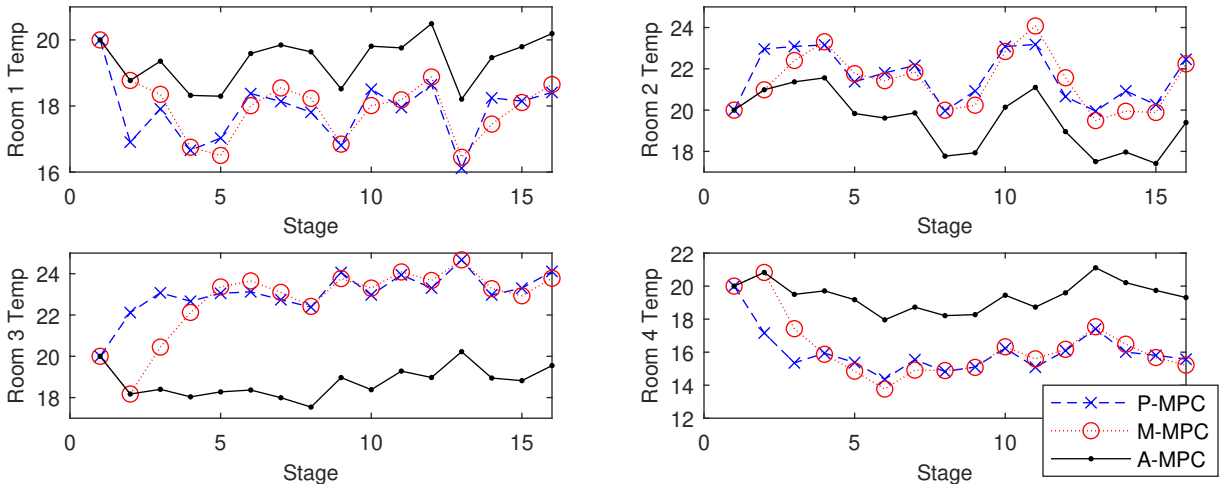

Figure 4.4: Closed-Loop State Trajectories for P-MPC, M-MPC, and A-MPC

Now suppose each agent has the private cost function

$$\mathbf{V}_i(T^{(i)}, u^{(i)}) = \frac{\lambda^{(i)}}{2}\sum_{k=0}^{N-1}(T_k^{(i)} - T_d^{(i)})^2 + \frac{(1-\lambda^{(i)})}{2}e^{(\gamma^{(i)}u_k^{(i)})^2} \tag{4.57}$$

where the tuple $(T_d^{(i)}, \lambda^{(i)}, \gamma^{(i)})$ is the agent's private information, namely: their desired room temperate and two scalars regulating the trade-off between comfort and energy usage. Following the setup of our mechanism, the building manager does not know the agent's private information nor the shape of their objective functions. The manager broadcasts the function $f(T, u; v, w) = \frac{1}{2}(e - vT_r)^2 + \frac{1}{2}(wu)^2$, where T_r is a reference temperature for the building manager.

We consider an MPC setting, where the principal's receding horizon OCP-S at stage t is given by

$$\begin{aligned}
\min\ & \frac{1}{2}\sum_{k=0}^{N-1}\sum_{i=1}^{I}(T_{t+k|t}^{(i)} - v_{t+k|t}^{(i)}T_r)^2 + (w_{t+k|t}^{(i)}u_{t+k|t}^{(i)})^2 \\
\text{s.t. } & T_{t+k+1|t} = AT_{t+k|t} + Bu_{t+k|t} + b_{t+k|t},\ \forall k \in [T-1] \\
& (T_{t+k|t}^{(i)}u_{t+k|t}^{(i)}) \in \tilde{J}_t^{(i)}, \forall k \in [N-1], i \in [I] \\
& T_{t|t}^{(i)} = T_t\ ,\ \forall i \in [I]
\end{aligned} \tag{4.58}$$

where A, B are given in (4.56) and $b_{t+k|t}$ is a prediction of $-\alpha T_{t+k}^{out}$ made at time t. Also, we use $u_{t+k|t}^{(i)}$ to denote the open-loop control input computed at stage t. Let (T_t^*, u_t^*) be the optimal solution of (4.58). The manager uses the current open-loop trajectories sent by

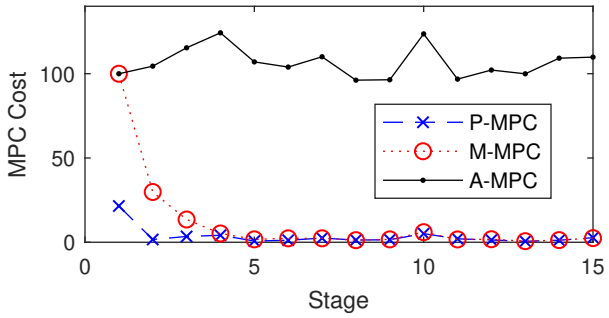

Figure 4.5: MPC Aggregated Stage Cost with Agents' True Utility Functions

agents to compute references

$$c^{(i)}_{t+k|t} = \sum_{i \in \mathcal{N}_i} A_{ij} \tilde{T}^{(j)}_{t+k|t}, \; \forall k \in [T-1] \tag{4.59}$$

where the neighborhoods $\mathcal{N}_i$ match the room configurations. The principal also uses $T^{*(i)}_t$ in order to compute the reference trajectory in the fee p_i. After receiving such references, each agent solves their own OCP-A with the computed fees p_i in the objective, obtaining a private solution vector $(\hat{T}^{(i)}_t, \hat{u}^{(i)}_t)$ and setting $\tilde{\lambda}^{(i)}_t$ to be the lagrange multipliers associated with the dynamics. Then each agent updates the remaining according to (4.53), which in our case reduces to

$$\begin{aligned} v^{(i)}_{t+k|t} &= ((1-\lambda^{(i)})\hat{T}^{(i)}_{t+k|t} + T^{(i)}_d)/T_r \\ w^{(i)}_{t+k|t} &= \sqrt{(1-\lambda^{(i)})(\gamma^{(i)})^2 e^{(\gamma^{(i)}\hat{u}^{(i)}_{t+k|t})^2}} \end{aligned} \tag{4.60}$$

after taking the associated derivatives. Lastly, $\tilde{J}^{(i)}_t = (\{-\infty, +\infty\}^T_{k=0}, \{-\infty, +\infty\}^T_{k=0})$ and $\tilde{T}^{(i)}_{t+1} = \hat{T}^{(i)}_t$. When $t = 0$, all weights are initialized to unit values. All optimization problems were solved using the optimization solver MOSEK [176]. We consider a optimal control length $T = 5$ and an MPC horizon of $N = 15$. We compare our mechanism-based MPC (M-MPC) with the perfect-information case (P-MPC), where the principal knows the exact form of each $\mathbf{V}_i$. We also consider a "consensus"-type case, where no weights are updated and T_r is set to the average of the desired temperatures (A-MPC). Fig. 4.4 shows the closed-loop state trajectory of the three approaches. It shows that our M-MPC closely tracks the P-MPC trajectory.

Fig. 4.5 shows M-MPC recovers the P-MPC cost after a few time steps. Since we used true costs to compute P-MPC, this shows our mechanism recovers the efficient trajectory. In contrast, the case without information exchange behaves poorly. This example shows our mechanism can be used with MPC: at each stage an optimal control problem is solved, the first-stage control is applied, and agents update their messages based on knowledge received from the principal.

Chapter 5

Real-time MPC for Cyber-Physical Systems

On this Chapter we will study algorithms that are used for the actual operation of Cyber-Physical Systems (CPS). The decision-making problem faced by the CPS can be often framed as an Optimal Control problem, where control decisions need to be computed in for a horizon a few seconds into the future. The nature of this decision-making problem, as "decisions taken across multiple time periods" lies in the center of the operation task of CPS. These Optimal Control problems need to be solve in a timely fashion in order to make sure the CPS can operate in real-time. We will develop optimization algorithms to solve the Optimal Control that is faced by the CPS. In our setting, this amounts to solving non-linear constrained optimization problem on a few hundred milliseconds intervals, for example in operating autonomous vehicles or orbital maneuvers of satellite. We will focus on Nonlinear model predictive control (NMPC) algorithm, which is an increasingly popular advanced control method for CPS. Nonlinear model predictive control (NMPC) generally requires the solution of a non-convex dynamic optimization problem at each sampling instant under strict timing constraints, based on a set of differential equations that can often be stiff and/or that may include implicit algebraic equations. We provide a local convergence analysis for the recently proposed adjoint-based sequential quadratic programming (SQP) algorithm that is based on a block-structured variant of the two-sided rank-one (TR1) quasi-Newton update formula to efficiently compute Jacobian matrix approximations in a sparsity preserving fashion. A particularly efficient algorithm implementation is proposed in case an implicit integration scheme is used for discretization of the optimal control problem, in which matrix factorization and matrix-matrix operations can be avoided entirely. The convergence analysis results as well as the computational performance of the proposed optimization algorithm are illustrated for two simulation case studies of NMPC.

We begin our analyses by introducing the direct multiple shooting based OCP problem formulation as well as the proposed adjoint-based inexact SQP method that is based on block-wise TR1 Jacobian updates in section 5.1. Section 5.2 presents the detailed convergence analysis for the optimization method and contains the main theoretical results of the

present paper. A numerically efficient implementation of the block-TR1 update formula in combination with a lifted Newton-type method for direct optimal control with implicit integration schemes such as, e.g., collocation methods, is then proposed and analyzed in Section 5.3. Finally, Section 5.4 presents numerical results of the NMPC case studies which illustrates the numerical performance of our proposed block-structure algorithm.

5.1 Adjoint-based SQP Method with Block-wise quasi-Newton Jacobian Updates for Nonlinear Optimal Control

Optimization based control and estimation techniques have attracted an increasing attention over the past decades. They allow a model-based design framework, in which the system dynamics, performance metrics and constraints can directly be taken into account. Receding horizon techniques such as model predictive control (MPC) and moving horizon estimation (MHE) have been studied extensively because of their desirable properties [163] and these optimization-based techniques have already been applied in a wide range of applications [94]. One of the main practical challenges in implementing such an optimization-based predictive control or estimation scheme, lies in the ability to solve the corresponding nonlinear and generally non-convex optimal control problem (OCP) under strict timing constraints and typically on embedded hardware with limited computational capabilities and available memory.

Let us consider the following continuous-time formulation of the optimal control problem that needs to be solved at each sampling instant

$$\min_{x(\cdot),u(\cdot)} \quad \int_0^T \ell(x(t),u(t))\,\mathrm{d}t + m(x(T)) \tag{5.1a}$$

$$\text{s.t.} \quad x_0 - \hat{x}_0 \;=\; 0, \tag{5.1b}$$

$$0 \;=\; f(\dot{x}(t), x(t), u(t)), \qquad \forall t \in [0,T], \tag{5.1c}$$

$$\pi(x(t),u(t)) \leq 0, \qquad \forall t \in [0,T], \tag{5.1d}$$

where T denotes the control horizon length, $x(t) \in \mathbb{R}^{n_x}$ denotes the differential states and $u(t) \in \mathbb{R}^{n_u}$ are the control inputs. The function $\ell(\cdot)$ defines the stage cost, $m(\cdot)$ denotes the terminal cost and the nonlinear dynamics are formulated as an implicit system of ordinary differential equations (ODE) in (5.1c), which could be extended with implicit algebraic equations. A common assumption is that the resulting system of differential-algebraic equations (DAE) is of index 1 [51]. The dynamic optimization problem is parametric, since it depends on the state estimate $\hat{x}_0$ at the current sampling instant, through the initial value condition in (5.1b). The path constraints are defined by the function $\pi(\cdot)$ in Eq. (5.1d) and, for simplicity of notation, they are further assumed to be affine. Note that a similar problem

as in (5.1) needs to be solved for optimization-based state and parameter estimation, without the given initial state value.

In direct optimal control methods, one forms a discrete-time approximation of the continuous time OCP in (5.1) based on an appropriate parameterization of the state and control trajectories over the time horizon $t \in [0, T]$, resulting in a tractable nonlinear program (NLP) that needs to be solved. Popular examples of this approach include the direct multiple shooting method [50] and direct collocation [41, 46]. Note that these techniques often need to rely on implicit integration methods in order to deal with stiff and/or implicit systems of differential or differential-algebraic equations [206]. The resulting constrained optimization problem can be handled by standard Newton-type algorithms such as interior point methods [248] and sequential quadratic programming (SQP) [53] techniques for nonlinear optimization [186].

Quasi-Newton optimization methods are generally popular for solving such a constrained NLP. They result in computationally efficient Newton-type methods that solve the first order necessary conditions of optimality, i.e., the Karush-Kuhn-Tucker (KKT) conditions, without evaluating the complete Hessian of the Lagrangian and/or even without evaluating the Jacobian of the constraints [186]. Instead, quasi-Newton methods are based on low-rank update formulas for the Hessian and Jacobian matrix approximations [74]. Popular examples of this approach include the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [57] and the symmetric rank-one (SR1) update formula [71] for approximating the Hessian of the Lagrangian. Similarly, quasi-Newton methods can be used for approximating Jacobian matrices, e.g., of the constraint functions, such as the good and bad Broyden methods [58] as well as the more recently proposed two-sided rank-one (TR1) update formula [113].

For the purpose of real-time predictive control and estimation, continuation-based online algorithms have been proposed that aim at further reducing the computational effort by exploiting the fact that a sequence of closely related parametric optimization problems is solved [51, 78]. One popular technique consists of the real-time iteration (RTI) algorithm that performs a single SQP iteration per time step, in combination with a sufficiently high sampling rate and a prediction-based warm starting in order to allow for closed-loop stability of the system [77]. The RTI algorithm can be implemented efficiently based on (fixed-step) integration schemes with tailored sensitivity propagation for discretization and linearization of the system dynamics [206] in combination with structure-exploiting quadratic programming solvers [94]. In addition, a lifted algorithm implementation has been proposed in [207] to directly embed the iterative procedure of implicit integration schemes, e.g., collocation methods, within a Newton-type optimization framework for optimal control.

Unlike standard inequality constrained optimization, nonlinear optimal control problems typically result in a particular sparsity structure in the Hessian of the Lagrangian and in the Jacobian matrix for the equality constraints. In direct optimal control methods, the objective function is typically separable resulting in a block-diagonal Hessian matrix. This property has been exploited in partitioned quasi-Newton methods that approximate and update each of the Hessian block matrices separately, as proposed and studied in [111, 110, 135]. On the other hand, the Jacobian matrix corresponding to the discretized system dynamics has a block bidiagonal sparsity structure, because of the stage wise coupling of the optimization

variables at subsequent time steps of the control horizon. For this purpose, the present article analyzes a novel tailored quasi-Newton method for optimal control using a partitioned or block-structured TR1-based Jacobian update formula. This adjoint-based SQP method for nonlinear optimal control, based on a Gauss-Newton Hessian approximation in combination with inexact Jacobian matrices, was proposed recently in [122].

We provide a complete presentation of the block-TR1 based SQP method for nonlinear optimal control, including a detailed discussion of the lifted collocation type implementation, extending earlier work of the same authors in [122]. Unlike the latter publication, a convergence analysis of this novel quasi-Newton type optimization algorithm is provided. More specifically, we prove convergence of the block-structured quasi-Newton Jacobian approximations to the exact Jacobian matrix within the null space of the active inequality constraints. Based on this result, under mild conditions, convergence of the overall inexact SQP method can be guaranteed. Locally linear or superlinear convergence rates can be shown, respectively, when using a Gauss-Newton or quasi-Newton based Hessian approximation scheme. In addition, it is shown how this convergence analysis extends to our lifted collocation implementation that avoids any matrix factorization or matrix-matrix operations. These convergence analysis results as well as the computational performance of the optimization algorithms are illustrated numerically for two simulation case studies of NMPC.

Block-wise TR1 based Sequential Quadratic Programming

A popular approach for direct optimal control is based on direct multiple shooting [50] that performs a time discretization, based on a numerical integration scheme [118] to solve the following initial value problem

$$0 = f(\dot{x}(\tau), x(\tau), u(\tau)), \quad \tau \in [t_i, t_{i+1}], \quad x(t_i) = x_i, \tag{5.2}$$

on each of N shooting intervals that are defined by a grid of consecutive time points t_i for $i = 0, \ldots, N$. For the sake of simplicity, we consider here an equidistant grid over the control horizon, i.e., $t_{i+1} - t_i = \frac{T}{N}$, and a piecewise constant control parametrization $u(\tau) = u_i$ for $\tau \in [t_i, t_{i+1})$ in (5.2). An explicit fixed-step integration scheme defines the discrete-time system dynamics $x_{i+1} = F_i(x_i, u_i)$ for the shooting interval $[t_i, t_{i+1}]$. For example, this can correspond to the popular Runge-Kutta method of order 4 (RK4) as defined in [118]. Note that explicit methods are only suitable in case the Jacobian $\frac{\partial f}{\partial \dot{x}}(\cdot)$ is non-singular. Otherwise, implicit schemes need to be used for implicit differential or differential-algebraic equations. Based on the explicit discretization scheme, the resulting block-structured optimal control

problem reads as

$$\min_{X,U} \quad \sum_{i=0}^{N-1} l_i(w_i) + l_N(w_N) \tag{5.3a}$$

$$\text{s.t.} \quad \hat{x}_0 \;=\; x_0, \tag{5.3b}$$

$$F_i(w_i) \;=\; x_{i+1}, \qquad i = 0, \ldots, N-1, \tag{5.3c}$$

$$P_i\, w_i \leq p_i, \qquad i = 0, \ldots, N, \tag{5.3d}$$

where the affine path constraints (5.3d) have been imposed on each of the shooting nodes and the compact notation $w_i := (x_i, u_i)$ for $i = 0, \ldots, N-1$ and $w_N := x_N$ is defined. Alternative discretization techniques exist with guaranteed constraint satisfaction [97, 200], which remains outside the scope of the present paper. Note that the optimization variables in (5.3) are directly the state $X = [x_0^\top, \ldots, x_N^\top]^\top$ and control trajectory $U = [u_0^\top, \ldots, u_{N-1}^\top]^\top$. Lastly, we define the joint state-input trajectory $w = [w_0^\top, \ldots, w_N^\top]^\top$.

SQP algorithm with inexact Jacobians

For a local minimum w^* of the NLP in (5.3), for which the linear independence constraint qualification (LICQ) holds, there must exist a unique set of multiplier values λ^*, μ^* such that the following Karush-Kuhn-Tucker (KKT) conditions are satisfied

$$\nabla_w \mathcal{L}(w^*, \lambda^*) + P^T \mu^* = 0 \tag{5.4a}$$

$$F(w^*) = X^*_{1:N} \tag{5.4b}$$

$$P\, w^* \leq p \tag{5.4c}$$

$$\mu^* \geq 0 \tag{5.4d}$$

$$\mu^*_j (P w^* - p)_j = 0, \quad j = 1, \ldots, n_\mathrm{p}, \tag{5.4e}$$

where $F(\cdot)$ and P are appropriate block-wise concatenations of the equality and inequality constraints, respectively, in (5.3c) and (5.3d), and n_p denotes the total number of inequality constraints. Here, we define $X_{1:N} = [x_1^\top, \ldots, x_N^\top]^\top$ and we include the initial condition constraint as part of the matrix P since we can represent a linear equality as two linear inequality constraints. Lastly, $\mathcal{L}(w, \lambda)$ denotes the 'truncated Lagrangian', omitting inequality constraints, and is therefore given by

$$\mathcal{L}(w, \lambda) = \sum_{i=0}^{N-1} \big(l_i(x_i, u_i) + \lambda_i^\top (F_i(w_i) - x_{i+1}) \big) + l_N(x_N). \tag{5.5}$$

Given the set of indices $\mathcal{A}$ for the inequality constraints that are active at the local minimum, the KKT system reduces to a nonlinear system of equations that can be solved directly by a Newton-type method. In particular, we are interested in a quasi-Newton algorithm where

we will approximate $\nabla^2_{ww}\mathcal{L}(w^k, \lambda^k)$ by a matrix H^k and $\frac{\partial F}{\partial w}(w^k)$ by a matrix A^k. Namely, we solve the following linearized system

$$\begin{bmatrix} H^k & A^{k^\top} - E^\top & P_{\mathcal{A}}^\top \\ A^k - E & & \\ P_{\mathcal{A}} & & \end{bmatrix} \begin{bmatrix} \Delta w^k \\ \Delta \lambda^k \\ \Delta \mu_A^k \end{bmatrix} = - \begin{bmatrix} g(w^k, \lambda^k) \\ F(w^k) - X^k_{1:N} \\ P_{\mathcal{A}}\, w^k - p_A \end{bmatrix}, \tag{5.6}$$

where $g(w^k, \lambda^k) = \nabla_w \mathcal{L}(w^k, \lambda^k) + \mu_A^{k^\top}(P_A\, w^k - p_A)$ at each Newton-type iteration k. Note that the matrix $P_{\mathcal{A}}$ is defined as the part of P that corresponds to the inequality constraints (5.4c) in the active set $\mathcal{A}$, and E denotes the constant matrix corresponding to the right-hand side of the equality constraints in (5.4b).

In order to efficiently solve the inequality constrained OCP in (5.3), let us consider the adjoint-based SQP algorithm with Gauss-Newton type Hessian approximation and inexact Jacobian information as introduced originally in [51, 253] for fast NMPC. Each SQP iteration solves a convex QP subproblem

$$\min_{\Delta W} \quad \sum_{i=0}^{N} \frac{1}{2} \Delta w_i^\top H_i^k\, \Delta w_i \,+\, h_i^{k^\top} \Delta w_i \tag{5.7a}$$

$$\text{s.t.} \quad \Delta x_0 \;=\; \hat{x}_0 - x_0^k, \tag{5.7b}$$

$$a_i^k + A_i^k\, \Delta w_i \;=\; \Delta x_{i+1}, \qquad i = 0, \ldots, N-1, \tag{5.7c}$$

$$P_i\, \Delta w_i \leq p_i^k, \qquad i = 0, \ldots, N, \tag{5.7d}$$

where the notation $\Delta w = [\Delta w_0^\top, \ldots, \Delta w_N^\top]^\top$ is used to denote the increments $\Delta w_i := w_i - w_i^k$, given the current solution guess X^k, U^k for the state and control trajectories at iteration k of the adjoint-based SQP method. The function $\pi(\cdot)$ that defines the path constraint (5.1d) was assumed to be affine and $p_i^k := p_i - P_i\, w_i^k$. Note that tracking formulations for NMPC typically include a stage cost that is defined by a (nonlinear) least squares term $l_i(x_i, u_i) = \frac{1}{2}\|R(x_i, u_i)\|_2^2$ for $i = 0, \ldots, N$. The generalized Gauss-Newton (GGN) method from [49] uses the block-structured Hessian approximation $H_i^k := \nabla R(w_i^k) \nabla R(w_i^k)^\top \approx \nabla^2_{w_i w_i} \mathcal{L}(\cdot)$.

The matrix $A_i^k \approx \frac{\partial F_i}{\partial w_i}(w_i^k)$ denotes the Jacobian approximation and $a_i^k := F_i(w_i^k) - x_{i+1}^k$ for the discrete-time system dynamics in Eq. (5.7c). For real-time NMPC, such a Jacobian approximation can be obtained by reusing information from a previous NLP solution [51, 253]. The gradient term in the objective (5.7a) reads as

$$h_i^k := \nabla_{w_i} l_i(w_i^k) + \left(\frac{\partial F_i}{\partial w_i}(w_i^k) - A_i^k \right)^\top \lambda_i^k, \tag{5.8}$$

for $i = 0, \ldots, N-1$, in which λ_i^k denotes the current value of the Lagrange multipliers for the nonlinear continuity constraints in (5.3c). Note that the linearized KKT conditions in (5.6) correspond to the KKT optimality conditions for the QP in (5.7), for a fixed active set $\mathcal{A}$. In addition, each QP subproblem is convex because $H^k \succeq 0$, e.g., for the Gauss-Newton

Hessian approximation. A full-step inexact SQP method will sequentially solve each QP subproblem (5.7) and perform the following updates:

$$w^{k+1} = w^k + \Delta w^k \quad \text{and} \quad \lambda^{k+1} = \lambda^k + \Delta\lambda^k = \lambda_{QP}^{k+1}, \tag{5.9}$$

where λ_{QP}^{k+1} denote the Lagrange multiplier values for Eq. (5.7c) at the QP solution. We do not need to perform explicit updates for the Lagrange multipliers associated with the inequality constraints, because they are assumed to be affine, hence not impacting any computation on the QP formulation in (5.7).

Dynamic block-wise TR1 Jacobian updates

At each SQP iteration, we perform the block-wise two-sided rank-one (TR1) Jacobian update, as proposed recently in [122]. Following the work in [113], given current Jacobian approximations A_i^k for $i = 0, \ldots, N-1$, we would like that each updated approximation matrix A_i^{k+1} satisfies the following two secant conditions

$$\begin{aligned} &\text{Adjoint Condition (AC):} \quad {\sigma_i^k}^\top A_i^{k+1} = {\gamma_i^k}^\top \\ &\text{Forward Condition (FC):} \quad A_i^{k+1} s_i^k = y_i^k, \end{aligned} \tag{5.10}$$

where we define the adjoint vector $\gamma_i^k = \frac{\partial F_i}{\partial w_i}(w_i^{k+1})^\top \sigma_i^k$, given ${\sigma_i^k}^\top = (\lambda_i^{k+1} - \lambda_i^k)^\top$, and the difference in function evaluations $y_i^k = F(w_i^{k+1}) - F(w_i^k)$. Note that λ_i^{k+1} and λ_i^k, respectively, denote the new and old Lagrange multipliers for the linearized equality constraints in Eq. (5.7c). Similarly, $w_i^k := (x_i^k, u_i^k)$ and $w_i^{k+1} := w_i^k + \Delta w_i^k$ denote, respectively, the old and new primal variables, such that $s_i^k := w_i^{k+1} - w_i^k$. Note that the gradient $\gamma_i^k = \frac{\partial F_i}{\partial w_i}(w_i^{k+1})^\top \sigma_i^k$ can be computed efficiently using the backward or adjoint mode of algorithmic differentiation (AD), e.g., see [109].

The proposed block-wise TR1 update formula then reads as follows

$$A_i^{k+1} = A_i^k + \alpha_i^k \left(y_i^k - A_i^k s_i^k\right) \left({\gamma_i^k}^\top - {\sigma_i^k}^\top A_i^k\right), \tag{5.11}$$

for $i = 0, \ldots, N-1$ and where α_i^k is a scalar that will be defined further. Aside from the case where the function $F(\cdot)$ is affine, the two conditions in Eq. (5.10) are not consistent with each other and they can therefore generally not both be satisfied by the updated matrix A_i^{k+1} at each iteration. Thus, similar to the standard TR1 update in [113], the block-wise update will only be able to satisfy one or the other. In the adjoint variant of the update, the scaling value is defined as

$$\alpha_{\mathrm{A},i}^k = \frac{1}{{\sigma_i^k}^\top (y_i^k - A_i^k s_i^k)}, \tag{5.12}$$

such that the adjoint condition in (5.10) is satisfied exactly and the forward condition holds up to some accuracy. Similarly, this value reads as follows for the forward variant

$$\alpha_{\mathrm{F},i}^k = \frac{1}{({\gamma_i^k}^\top - {\sigma_i^k}^\top A_i^k)\, s_i^k}, \tag{5.13}$$

where the forward condition is satisfied exactly. As discussed in [114], an additional damping of the Jacobian updates can be introduced in order to avoid singularity. It is interesting to note that, since we apply the block-wise TR1 update from (5.11) for each shooting interval $i = 0, \ldots, N-1$, the resulting update for the complete constraint Jacobian matrix of the QP in (5.7) corresponds to a rank-N update.

As in [113], we impose a skipping condition in order to avoid a potential blow-up of the block-wise TR1 update when the denominator of the scaling factor becomes small or even zero. For our purposes, the skipping condition itself depends on the type of formula that is used. We update the block matrix A_i^k only if the following holds

$$\left|(\gamma_i^{k^\top} - \sigma_i^{k^\top} A_i^k) s_i^k\right| \geq c_1 \left\|\sigma_i^k\right\| \left\|y_i^k - A_i^k s_i^k\right\|, \tag{5.14}$$

with $c_1 \in (0,1)$ if $\alpha_i^k = \alpha_{\mathrm{F},i}^k$ in the forward TR1 update, and

$$\left|\sigma_i^{k^\top} (y_i^k - A_i^k s_i^k)\right| \geq c_1 \left\|s_i^k\right\| \left\|\gamma_i^k - A_i^{k^\top} \sigma_i^k\right\|, \tag{5.15}$$

with $c_1 \in (0,1)$ if $\alpha_i^k = \alpha_{\mathrm{A},i}^k$ in the adjoint TR1 update. In addition to consistently choosing either the forward or adjoint Jacobian update formula, we propose a more dynamic variant of the algorithm that picks either $\alpha_{\mathrm{F},i}^k$ or $\alpha_{\mathrm{A},i}^k$ for each block matrix at any given iteration. It may not be clear what is the best approach to select which type of update is to be executed for each block matrix at a given iteration. However, in the next section, we prove the local convergence properties of the algorithm under any arbitrary sequence of updates that satisfy the skipping conditions in (5.14) and (5.15) for each block i at every iteration k.

Algorithm 4 One iteration of SQP method with block-wise TR1 Jacobian updates.

Input: $w_i^k = (x_i^k, u_i^k)$, λ_i^k and A_i^k for $i = 0, \ldots, N-1$.

`Problem linearization and QP preparation`

1: Formulate the QP in (5.7) with Jacobian matrices A_i^k, Gauss-Newton Hessian approximations H_i^k and vectors a_i^k, p_i^k and h_i^k in (5.8) for $i = 0, \ldots, N-1$.

`Computation of Newton-type step direction`

2: Solve the QP subproblem in Eq. (5.7) to update optimization variables: $w_i^{k+1} \leftarrow w_i^k + \Delta w_i^k$ and $\lambda_i^{k+1} \leftarrow \lambda_i^k + \Delta\lambda_i^k$. ▷ full step

`Block-wise TR1 Jacobian updates`

3: **for** $i = 0, \ldots, N-1$ **do in parallel**

4: Choose $\alpha_i^k = \alpha_{\mathrm{F},i}^k$ or $\alpha_i^k = \alpha_{\mathrm{A},i}^k$ via some decision rule.

5: $A_i^{k+1} \leftarrow A_i^k + \alpha_i^k \left(y_i^k - A_i^k s_i^k\right) \left(\gamma_i^{k^\top} - \sigma_i^{k^\top} A_i^k\right)$.

6: **end for**

Output: $w_i^{k+1} = (x_i^{k+1}, u_i^{k+1})$, λ_i^{k+1} and A_i^{k+1} for $i = 0, \ldots, N-1$.

The complete adjoint-based SQP method that uses parallelizable block-wise TR1 Jacobian updates is summarized in Algorithm 4. Note that, for simplicity, the SQP algorithm is

presented as a full-step method without any globalization strategies to ensure convergence to a local minimum [186]. This is also further motivated by the use of online algorithms for real-time NMPC as discussed in [78].

5.2 Convergence Results for Block-wise TR1-based SQP Method

For the convergence analysis of sequential quadratic programming, it is standard to rely on a result that the active set, i.e., the set of active inequality constraints in the QP subproblems is stable in a neighbourhood around a local minimizer of the nonlinear program [186]. This allows us to study the local convergence properties of the block-TR1 based SQP method under the assumption that the active set has already been fixed, resulting, locally, in an equality constrained problem.

Stability of the active set and local convergence

Let us start by briefly repeating the result from [81] on the stability of the active set in the QP subproblems near the NLP solution and the corresponding conditions on local convergence properties for an adjoint-based SQP method with inexact Jacobians.

Theorem 5.2.1. *(Stability of active set and local convergence) Let the NLP solution vectors w^*, λ^* be given and assume that:*

(i) at w^* LICQ holds, and there exist Lagrange multiplier values μ^* such that (w^*, λ^*, μ^*) satisfies the KKT conditions in (5.4)*.*

(ii) at w^* strict complementarity holds, i.e., the multipliers $\mu^*_{\mathcal{A}}$ of the active inequalities $P_{\mathcal{A}} w^* = p_{\mathcal{A}}$ satisfy $\mu^*_{\mathcal{A}} > 0$, where $P_{\mathcal{A}}$ is a matrix consisting of all rows of P that correspond to the active inequalities at the NLP solution.

(iii) there are two sequences of uniformly bounded matrices (A^k, H^k), each H^k positive semidefinite on the null space of A^k, such that the sequence of matrices

$$J^k := \begin{bmatrix} N^\top H^k & N^\top A^{k\top} \\ A^k & \\ P_{\mathcal{A}} & \end{bmatrix} \approx \frac{\partial \mathcal{F}}{\partial y}(y^k), \quad \text{where} \quad \mathcal{F}(y) := \begin{bmatrix} N^\top \nabla_w \mathcal{L}(w, \lambda) \\ F(w) - X_{1:N} \\ P_{\mathcal{A}}\, w - p_{\mathcal{A}} \end{bmatrix},$$

is uniformly bounded and invertible with a uniformly bounded inverse. Here, N is a null space matrix with appropriate dimensions with orthonormal column vectors such that $N^\top N = \mathbb{1}$ and $P_{\mathcal{A}}\, N = 0$.

(iv) there is a sequence of iterates $y^k := (w^k, \lambda^k)$ generated according to

$$w^{k+1} = w^k + \Delta w^k \quad \text{and} \quad \lambda^{k+1} = \lambda^k + \Delta \lambda^k = \lambda^{k+1}_{QP},$$

where Δw^k is the primal solution of the QP subproblem in (5.7) and λ_{QP}^{k+1} denote the Lagrange multipliers corresponding to the equality constraints (5.7c). Each iteration can be written in compact form as $y^{k+1} = y^k - J^{k^{-1}}\mathcal{F}(y^k)$.

(v) there exists $\kappa < 1$ such that, for all $k \in \mathbb{N}$, it can be guaranteed that

$$\left\| J^{k+1^{-1}} \left(J^k - \frac{\partial \mathcal{F}}{\partial y}(y^k + t\Delta y^k) \right) \Delta y^k \right\| \leq \kappa \|\Delta y^k\|, \quad \forall t \in [0,1]. \tag{5.16}$$

Then, there exists a neighbourhood $\bar{\mathcal{N}}$ of (w^*, λ^*) such that for all initial guesses $(w^0, \lambda^0) \in \bar{\mathcal{N}}$ the sequence (w^k, λ^k) converges q-linearly towards (w^*, λ^*) with rate κ, and the solution of each QP (5.7) has the same active set as w^*.

In addition to the latter result that guarantees a q-linear local convergence rate in a neighbourhood of the NLP solution, the following theorem states a condition under which q-superlinear local convergence can be obtained instead.

Theorem 5.2.2. *(Superlinear convergence) If the equality*

$$\lim_{k\to\infty} \begin{bmatrix} N^\top H^k N & N^\top A^{k^\top} \\ A^k N & \mathbb{0} \end{bmatrix} = \begin{bmatrix} N^\top \nabla^2_{ww} \mathcal{L}(w^\star, \lambda^\star) N & N^\top \frac{\partial F}{\partial w}(w^\star)^\top \\ \frac{\partial F}{\partial w}(w^\star) N & \mathbb{0} \end{bmatrix}, \tag{5.17}$$

holds in addition to the assumptions of Theorem 5.2.1, then the local convergence rate is q-superlinear instead.

The proofs for both Theorem 5.2.1 and 5.2.2 can be found in [81] for an adjoint-based SQP method with inexact Jacobians that matches our problem formulation.

Convergence of the block-wise TR1 Jacobian updates

Theorem 5.2.1 holds for a general class of constraint Jacobian and Hessian approximation matrices (A^k, H^k). Therefore, we have to show that our block-wise TR1 updates produce a sequence of block-structured matrices that converge to the exact Jacobian, which is itself block-structured, projected onto the null space of the active inequality constraint matrix P_A. Namely, defining a null space matrix N as in Theorem 5.2.1, we need to prove that the following holds

$$\lim_{k\to\infty} \left\| \left(A_i^k - \frac{\partial F_i}{\partial w}(w_i^*) \right) N_i \right\| = 0, \quad \forall i = 0, \ldots, N-1, \tag{5.18}$$

where N_i is the projection of the null space matrix N in the variable space corresponding to block i. The only non-zero entries that are inexact in the Jacobian approximation matrix A^k are those corresponding to the block-TR1 matrices A_i^k, $i = 0, \ldots, N-1$.

Assumption 5.2.3. *Let us make the following assumptions:*

(AS1) *The Lagrangian function is twice continuously differentiable.*

(AS2) *The function $\nabla_w F(w)$ is Lipschitz continuous, i.e., there exists a constant c_3 such that $\|\nabla_w F(w_1) - \nabla_w F(w_2)\| \leq c_3 \|w_1 - w_2\|$, for any w_1, w_2.*

(AS3) *Let $\{(w^k, \lambda^k)\}$ be a sequence of iterates generated by our block-TR1 based SQP method in Algorithm 4, with a corresponding sequence of update parameters $\{\alpha_i^k\}$, while satisfying the skipping criteria in eqs.* (5.14)-(5.15).

(AS4) *The SQP iterates $\{(w^k, \lambda^k)\}$ converge to a limit point (w^*, λ^*).*

(AS5) *There is k_0 such that the active set is stable for all iterates $k \geq k_0$.*

(AS6) *For each block i, the sequence of projections of $\{s^k\}$ on the subspace associated with block i, namely $\{s_i^k\}$ is uniformly linearly independent in the projected null space N_i. There exist $c_4 > 0$ and l such that $l \geq q_i, i = 0, \ldots, N-1$, and for each $k_i \geq k_0$, there exist q_i distinct indices k_i^j with $k_i \leq k_i^1 < \ldots < k_i^{q_i} \leq k_i + l$, $s_{N,i}^{k_i^j} \in \mathbb{R}^{q_i}$, $s_i^{k_i^j} = N_i s_{N,i}^{k_i^j}$, $j = 1, \ldots, q_i$ and the minimum singular value $\sigma_{min}(S_{N_i}^{k_i})$ of the matrix*

$$S_{N_i}^{k_i} = \begin{bmatrix} \frac{s_{N,i}^{k_i^1}}{\|s_{N,i}^{k_i^1}\|} & \cdots & \frac{s_{N,i}^{k_i^{q_i}}}{\|s_{N,i}^{k_i^{q_i}}\|} \end{bmatrix} \tag{5.19}$$

is bounded below by c_4, i.e., $\sigma_{min}(S_{N_i}^{k_i}) \geq c_4$.

Note that the assumptions $(AS1)$-$(AS5)$ are relatively mild and quite standard in the Newton-type convergence analysis of SQP methods [81]. Especially, condition $(AS5)$ holds due to the local stability result in Theorem 5.2.1 for the active set near the NLP solution. Even though $(AS6)$ seems relatively strong, a very similar assumption is made in existing convergence results for quasi-Newton type matrix update schemes [71, 81]. Here, we only require uniform linear independence inside each block i.

We proceed now to prove the convergence of the quasi-Newton block-structured constraint Jacobian approximation matrices, using ideas from [71] and [81]. We start by first showing an intermediate result in the following lemma.

Lemma 5.2.4. *Given $(AS1)$-$(AS3)$ in Assumption 5.2.3, then the following holds for each Jacobian block matrix approximation*

$$\left\| y_i^k - A_i^l s_i^k \right\| \leq \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1 \right)^{l-k} \eta_i^{l,k} \|s_i^k\|, \quad \forall l \geq k+1, \tag{5.20a}$$

$$\left\| \gamma_i^k - A_i^{l^\top} \sigma_i^k \right\| \leq \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1 \right)^{l-k} \eta_i^{l,k} \|\sigma_i^k\|, \quad \forall l \geq k+1, \tag{5.20b}$$

where $i = 0, \ldots, N-1$ and $\eta_i^{l,k} = \max\{\|w_i^r - w_i^s\| \mid k \leq s \leq r \leq l\}$ is defined.

Proof. Our proof follows closely the proof of Lemma 4.1 in [81] but extended to our block-structured method and generalized to include both the forward and adjoint TR1 Jacobian update formulas.

Step 1: Eq. (5.20a) *based on forward Jacobian update*
We start by showing the result of Eq. (5.20a) when $\alpha_i = \alpha_{\mathrm{F},i}$. The proof is by induction on l for each block $i = 0, \ldots, N-1$. For $l = k+1$, we know that $y_i^k - A_i^l s_i^k = 0$ based on the forward update. Assume that the result in (5.20a) holds for all $\{k+1, \ldots, l\}$. Then, we have the following

$$\|y_i^k - A_i^{l+1} s_i^k\| = \left\| y_i^k - A_i^l s_i^k - \alpha_{\mathrm{F},i}^l \rho_i^l \tau_i^{l^\top} s_i^k \right\| \leq \|y_i^k - A_i^l s_i^k\| + \left| \frac{(\tau_i^l, s_i^k)}{(\tau_i^l, s_i^l)} \right| \|\rho_i^l\| \tag{5.21}$$

where we use the notation $(\cdot, \cdot)$ to denote an inner product. In addition, $\tau_i^l = \gamma_i^l - A_i^{l^\top} \sigma_i^l$ and $\rho_i^l = y_i^l - A_i^l s_i^l$ such that $\alpha_{\mathrm{F},i}^l = \frac{1}{(\tau_i^l, s_i^l)}$. Then, using the result in Eq. (5.20a), we can write

$$\begin{aligned} \left|(\tau_i^l, s_i^k)\right| = \left|(\gamma_i^l - A_i^{l^\top} \sigma_i^l, s_i^k)\right| &\leq \left|(\gamma_i^l, s_i^k) - (\sigma_i^l, y_i^k)\right| + \left|(\sigma_i^l, y_i^k) - (\sigma_i^l, A_i^l s_i^k)\right| \\ &\leq \left|(\gamma_i^l, s_i^k) - (\sigma_i^l, y_i^k)\right| + \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1\right)^{l-k} \eta_i^{l,k} \left\|\sigma_i^l\right\| \left\|s_i^k\right\|. \end{aligned} \tag{5.22}$$

Using the mean-value theorem, it follows that

$$\begin{aligned} \left|(\gamma_i^l, s_i^k) - (\sigma_i^l, y_i^k)\right| &= \left| \sigma_i^{l^\top} \left(\frac{\partial F_i}{\partial w}(w_i^l + s_i^l) - \int_0^1 \frac{\partial F_i}{\partial w}(w_i^k + t\, s_i^k) dt \right) s_i^k \right| \\ &\leq c_3\, \eta_i^{l+1,k} \left\|\sigma_i^l\right\| \left\|s_i^k\right\|, \end{aligned} \tag{5.23}$$

based on the Lipschitz continuity in $(AS2)$. From the skipping condition in (5.14), we know that $\left|(\tau_i^l, s_i^l)\right| \geq c_1 \left\|\sigma_i^l\right\| \left\|\rho_i^l\right\|$. In addition, given that $\eta_i^{l,k} \leq \eta_i^{l+1,k}$, we obtain

$$\begin{aligned} &\left\|y_i^k - A_i^{l+1} s_i^k\right\| \leq \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1\right)^{l-k} \eta_i^{l,k} \left\|s_i^k\right\| + \\ &\left(c_3 \eta_i^{l+1,k} + \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1\right)^{l-k} \eta_i^{l,k} \right) \frac{\left\|\sigma_i^l\right\| \left\|s_i^k\right\|}{\left|(\tau_i^l, s_i^l)\right|} \left\|\rho_i^l\right\| \leq \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1\right)^{l+1-k} \eta_i^{l+1,k} \left\|s_i^k\right\|. \end{aligned} \tag{5.24}$$

Step 2: Eq. (5.20a) *based on adjoint Jacobian update*
Let us continue this proof by induction on l for Eq. (5.20a), based on the adjoint Jacobian update formula. First, we derive the following error bound for the adjoint Jacobian update

in case $l = k+1$ in Eq. (5.20a)

$$\begin{aligned}\left\|y_i^k - A_i^{k+1} s_i^k\right\| &= \left\| y_i^k - A_i^k s_i^k - \frac{1}{(\sigma_i^k, \rho_i^k)} \rho_i^k {\tau_i^k}^\top s_i^k \right\| = \left\| \rho_i^k - \frac{1}{(\sigma_i^k, \rho_i^k)} \rho_i^k {\tau_i^k}^\top s_i^k \right\| \\ &= \left| 1 - \frac{(\tau_i^k, s_i^k)}{(\sigma_i^k, \rho_i^k)} \right| \left\|\rho_i^k\right\| = \left| \frac{(\sigma_i^k, y_i^k - A_i^k s_i^k) - (\gamma_i^k - {A_i^k}^\top \sigma_i^k, s_i^k)}{(\sigma_i^k, \rho_i^k)} \right| \left\|\rho_i^k\right\| \\ &= \frac{\left|(\sigma_i^k, y_i^k) - (\gamma_i^k, s_i^k)\right|}{\left|(\sigma_i^k, \rho_i^k)\right|} \left\|\rho_i^k\right\|.\end{aligned} \tag{5.25}$$

From the skipping conditions in (5.14)-(5.15), we know that $\left|(\sigma_i^k, \rho_i^k)\right| \geq c_1 \left\|s_i^k\right\| \left\|\tau_i^k\right\|$ and $\left\|\rho_i^k\right\| \leq \frac{\left|(\tau_i^k, s_i^k)\right|}{c_1 \left\|\sigma_i^k\right\|} \leq \frac{\left\|\tau_i^k\right\| \left\|s_i^k\right\|}{c_1 \left\|\sigma_i^k\right\|}$ holds. We can use these lower and upper bounds to rewrite the latter expression as

$$\begin{aligned}\left\|y_i^k - A_i^{k+1} s_i^k\right\| &= \frac{\left|(\sigma_i^k, y_i^k) - (\gamma_i^k, s_i^k)\right|}{\left|(\sigma_i^k, \rho_i^k)\right|} \left\|\rho_i^k\right\| \leq \frac{\left|(\sigma_i^k, y_i^k) - (\gamma_i^k, s_i^k)\right|}{c_1 \left\|s_i^k\right\| \left\|\tau_i^k\right\|} \left\|\rho_i^k\right\| \\ &\leq \frac{\left|(\sigma_i^k, y_i^k) - (\gamma_i^k, s_i^k)\right|}{c_1^2 \left\|\sigma_i^k\right\|} \\ &\leq \frac{c_3}{c_1^2} \left\|s_i^k\right\|^2,\end{aligned} \tag{5.26}$$

where we additionally used the result

$$\begin{aligned}\left|(\gamma_i^k, s_i^k) - (\sigma_i^k, y_i^k)\right| &= \left| {\sigma_i^k}^\top \left(\frac{\partial F_i}{\partial w}(w_i^{k+1}) - \int_0^1 \frac{\partial F_i}{\partial w}(w_i^k + t\, s_i^k) dt \right) s_i^k \right| \\ &\leq c_3 \left\|\sigma_i^k\right\| \left\|s_i^k\right\|^2.\end{aligned} \tag{5.27}$$

Note that $\eta_i^{k+1,k} = \left\|s_i^k\right\|$ such that Eq. (5.20a) holds in case $l = k+1$. Assume that the result in (5.20a) holds for all $\{k+1, \ldots, l\}$. Then, we have the following

$$\begin{aligned}\|y_i^k - A_i^{l+1} s_i^k\| &= \left\| y_i^k - A_i^l s_i^k - \alpha_{\mathrm{A},i}^l \rho_i^l {\tau_i^l}^\top s_i^k \right\| \\ &\leq \|y_i^k - A_i^l s_i^k\| + \left| \frac{(\tau_i^l, s_i^k)}{(\sigma_i^l, \rho_i^l)} \right| \|\rho_i^l\|,\end{aligned} \tag{5.28}$$

for the adjoint Jacobian update formula in which $\alpha_{\mathrm{A},i} = \frac{1}{(\sigma_i^l, \rho_i^l)}$. From the skipping conditions in (5.14)-(5.15), we know that $\left|(\sigma_i^l, \rho_i^l)\right| \geq c_1 \left\|s_i^l\right\| \left\|\tau_i^l\right\|$ and $\left\|\rho_i^l\right\| \leq \frac{\left|(\tau_i^l, s_i^l)\right|}{c_1 \left\|\sigma_i^l\right\|} \leq \frac{\left\|\tau_i^l\right\| \left\|s_i^l\right\|}{c_1 \left\|\sigma_i^l\right\|}$ holds such that $\frac{\left\|\sigma_i^l\right\| \left\|\rho_i^l\right\|}{\left|(\sigma_i^l, \rho_i^l)\right|} \leq \frac{1}{c_1} \frac{\left\|\sigma_i^l\right\| \left\|\rho_i^l\right\|}{\left\|s_i^l\right\| \left\|\tau_i^l\right\|} \leq \frac{1}{c_1^2}$. In addition, given eqs. (5.22) and (5.23) and given

that $\eta_i^{l,k} \leq \eta_i^{l+1,k}$, we obtain

$$\begin{aligned}
&\left\|y_i^k - A_i^{l+1} s_i^k\right\| \leq \\
&\frac{c_3}{c_1}\left(\frac{2}{c_1^2}+1\right)^{l-k} \eta_i^{l,k}\left\|s_i^k\right\| + \left(c_3\eta_i^{l+1,k} + \frac{c_3}{c_1}\left(\frac{2}{c_1^2}+1\right)^{l-k}\eta_i^{l,k}\right)\frac{\left\|\sigma_i^l\right\|\left\|s_i^k\right\|}{\left|(\sigma_i^l,\rho_i^l)\right|}\left\|\rho_i^l\right\| \\
&\leq \frac{c_3}{c_1}\left(\frac{1}{c_1} + \left(\frac{1}{c_1^2}+1\right)\left(\frac{2}{c_1^2}+1\right)^{l-k}\right)\eta_i^{l+1,k}\left\|s_i^k\right\| \\
&\leq \frac{c_3}{c_1}\left(\frac{2}{c_1^2}+1\right)^{l+1-k}\eta_i^{l+1,k}\left\|s_i^k\right\|.
\end{aligned} \tag{5.29}$$

Note that the induction proof of step 1 and 2 implies that Eq. (5.20a) additionally holds when switching between the forward and adjoint Jacobian update formulas. A similar induction-based proof can be used to show the result of Eq. (5.20b) for the dynamic block-TR1 Jacobian updates.

□

Now, we present the resulting theorem on the convergence of the Jacobian approximation for the block-wise TR1 scheme under any sequence of decision rules that select the adjoint or forward updates at every iteration k and for each block $i = 0, \ldots, N-1$.

Theorem 5.2.5. *Given (AS1)-(AS6) in Assumption 5.2.3, then the following holds for each Jacobian block matrix $i = 0, \ldots, N-1$*

$$\lim_{k\to\infty}\left\|\left(A_i^k - \frac{\partial F_i}{\partial w_i}(w_i^*)\right)N_i\right\| = 0, \tag{5.30}$$

such that the following holds for the complete Jacobian approximation

$$\lim_{k\to\infty}\left\|\left(A^k - \frac{\partial F}{\partial w}(w^*)\right)N\right\| = 0. \tag{5.31}$$

Proof. Based on the inequality $\|w_i^r - w_i^s\| \leq \|w_i^r - w_i^*\| + \|w_i^s - w_i^*\|$ and using the definition $\eta_i^{l,k} = \max\{\|w_i^r - w_i^s\| \mid k \leq s \leq r \leq l\}$, one obtains

$$\eta_i^{k+l+1,k} \leq 2\,\nu_i^k \quad \text{for} \quad \nu_i^k = \max\{\|w_i^s - w_i^*\| \mid k \leq s \leq k+l+1\}, \tag{5.32}$$

for $l \geq q_i$ and q_i is defined as in Assumption 5.2.3. In addition, the following holds

$$\left\|y_i^j - \frac{\partial F_i}{\partial w_i}(w_i^*)s_i^j\right\| = \left\|\left(\int_0^1 \frac{\partial F_i}{\partial w_i}(w_i^j + ts_i^j)dt\right)s_i^j - \frac{\partial F_i}{\partial w_i}(w_i^*)s_i^j\right\| \tag{5.33a}$$

$$= \left\|\left(\int_0^1 \frac{\partial F_i}{\partial w_i}(w_i^j + ts_i^j)dt - \frac{\partial F_i}{\partial w_i}(w_i^*)\right)s_i^j\right\| \tag{5.33b}$$

$$\leq c_3\nu_i^k\left\|s_i^j\right\|, \tag{5.33c}$$

at an iteration j, where $k \leq j \leq k+l$, regardless of whether the forward or adjoint Jacobian update formula has been used. Moreover, from Lemma 5.2.4, we have that

$$\begin{aligned} \left\| y_i^j - A_i^{k+l+1} s_i^j \right\| &\leq \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1 \right)^{k+l+1-j} \eta_i^{k+l+1,j} \|s_i^j\|, \quad k \leq j \leq k+l, \\ &\leq 2 \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1 \right)^{l+1} \nu_i^k \|s_i^j\|. \end{aligned} \tag{5.34}$$

We use the triangle inequality to obtain

$$\left\| \left(A_i^{k+l+1} - \frac{\partial F_i}{\partial w_i}(w_i^*) \right) \frac{s_i^j}{\|s_i^j\|} \right\| \leq \frac{1}{\|s_i^j\|} \left(\left\| y_i^j - A_i^{k+l+1} s_i^j \right\| + \left\| y_i^j - \frac{\partial F_i}{\partial w_i}(w_i^*) s_i^j \right\| \right) \tag{5.35a}$$

$$\leq \left(2 \frac{c_3}{c_1} \left(\frac{2}{c_1^2} + 1 \right)^{l+1} + c_3 \right) \nu_i^k, \tag{5.35b}$$

which holds for a sequence of indices $j = k_i^1, \ldots, k_i^{q_i}$. Then, we use the linear independence condition $(AS6)$ in Assumption 5.2.3 that guarantees both existence of the inverse $(S_{N_i}^{k_i})^{-1}$ and the upper bound $\|(S_{N_i}^{k_i})^{-1}\| \leq 1/c_4$, such that

$$\left\| \left(A_i^{k+l+1} - \frac{\partial F_i}{\partial w_i}(w_i^*) \right) N_i \right\| \leq \frac{1}{c_4} \left\| \left(A_i^{k+l+1} - \frac{\partial F_i}{\partial w_i}(w_i^*) \right) N_i S_{N_i}^{k_i} \right\| \tag{5.36a}$$

$$\leq c_5 \, \nu_i^k, \tag{5.36b}$$

where $c_5 = \frac{c_3}{c_4} \left(\frac{2}{c_1} \left(\frac{2}{c_1^2} + 1 \right)^{l+1} + 1 \right) \sqrt{q_i}$ has been defined. Lastly, the result in Eq. (5.30) follows from the fact that assumption $(AS4)$ implies that ν_i^k tends to zero. Note that this asymptotic result holds regardless of which Jacobian update (adjoint or forward TR1 formula) is performed for each block $i = 0, \ldots, N-1$. The same convergence result then holds for the complete Jacobian matrix in (5.31), based on separability of the active inequality constraints and of the nonlinear constraint functions. □

Local rate of linear convergence for Gauss-Newton based SQP

One iteration of the adjoint-based Gauss-Newton SQP method solves the linear system in Eq. (5.6), which can be written in the following compact form

$$\tilde{J}_{\text{IN}}(z^k) \Delta z = -\mathcal{F}(z^k), \tag{5.37}$$

where $\mathcal{F}(\cdot)$ denotes the KKT optimality conditions in the right-hand side of Eq. (5.6). Let us define regularity for a local minimizer $z^\star := (w^\star, \lambda^\star, \mu^\star)$ of the NLP, given a particular set of active inequality constraints. For this purpose, we rely on the linear independence constraint

qualification (LICQ) and the second order sufficient conditions (SOSC) for optimality, of which the latter requires that the Hessian of the Lagrangian is strictly positive definite in the directions of the critical cone [186].

Definition 5.2.6. A minimizer of an equality constrained NLP is called a regular KKT point, if both LICQ and SOSC are satisfied at this KKT point.

The convergence of this Newton-type optimization method then follows the classical and well-known local contraction theorem from [52, 75, 81, 188, 199]. We use a particular version of this theorem from [80, 203], providing sufficient and necessary conditions for the existence of a neighbourhood of the solution where the Newton-type iteration converges locally. Let $\rho(P)$ denote the spectral radius, i.e., the maximum absolute value of the eigenvalues for the square matrix P.

Theorem 5.2.7 (Local Newton-type contraction [188])**.** *We consider the twice continuously differentiable function $\mathcal{F}(z)$ from Eq.* (5.6) *and the regular KKT point $\mathcal{F}(z^\star) = 0$ from Definition 5.2.6. We then apply the Newton-type iteration in Eq.* (5.37)*, where $\tilde{J}_{IN}(z) \approx J(z)$ is additionally assumed to be continuously differentiable and invertible in a neighbourhood of the solution. If all eigenvalues of the iteration matrix have a modulus smaller than one, i.e., if the spectral radius satisfies*

$$\kappa^\star := \rho\left(\tilde{J}_{IN}(z^\star)^{-1} J(z^\star) - \mathbb{1}\right) < 1, \tag{5.38}$$

then this fixed point $z^\star$ is asymptotically stable. Additionally, the iterates z^k converge linearly to the KKT point $z^\star$ with the asymptotic contraction rate $\kappa^\star$ when initialized sufficiently close. On the other hand, the fixed point $z^\star$ is unstable if $\kappa^\star > 1$.

A proof for Theorem 5.2.7 can be found in [80, 206], based on nonlinear systems theory. Using this result, let us define the linear contraction rate for a Gauss-Newton method with exact Jacobian information

$$\kappa^\star_{\mathrm{GN}} := \rho\left(\begin{bmatrix} H & (\frac{\partial F}{\partial w} - E)^\top & P_{\mathcal{A}}^\top \\ \frac{\partial F}{\partial w} - E & & \\ P_{\mathcal{A}} & & \end{bmatrix}^{-1} \begin{bmatrix} \nabla_w^2 \mathcal{L} & (\frac{\partial F}{\partial w} - E)^\top & P_{\mathcal{A}}^\top \\ \frac{\partial F}{\partial w} - E & & \\ P_{\mathcal{A}} & & \end{bmatrix} - \mathbb{1}\right) < 1, \tag{5.39}$$

at the local solution point $z^\star := (w^\star, \lambda^\star, \mu^\star)$ of the KKT conditions. In what follows, we show that the local contraction rate for the block-TR1 Gauss-Newton SQP method

$$\kappa^\star_{\mathrm{BTR1}} := \rho\left(\begin{bmatrix} H & (A - E)^\top & P_{\mathcal{A}}^\top \\ A - E & & \\ P_{\mathcal{A}} & & \end{bmatrix}^{-1} \begin{bmatrix} \nabla_w^2 \mathcal{L} & (\frac{\partial F}{\partial w} - E)^\top & P_{\mathcal{A}}^\top \\ \frac{\partial F}{\partial w} - E & & \\ P_{\mathcal{A}} & & \end{bmatrix} - \mathbb{1}\right) < 1, \tag{5.40}$$

coincides with the exact Jacobian based linear convergence rate in (5.39). The following result states that the spectrum of the iteration matrix $\tilde{J}_{\mathrm{IN}}(z^\star)^{-1} J(z^\star) - \mathbb{1}$ at the solution point $z^\star := (w^\star, \lambda^\star, \mu^\star)$ coincides with the spectrum of the iteration matrix $\tilde{J}_{\mathrm{GN}}(z^\star)^{-1} J(z^\star) - \mathbb{1}$, using the notation $\sigma(P)$ to denote the spectrum, i.e., the set of eigenvalues for a matrix P.

Lemma 5.2.8. *Given (AS1)-(AS6) in Assumption 5.2.3, for a regular KKT point $z^\star := (w^\star, \lambda^\star, \mu^\star)$, eigenvalues of the block-TR1 iteration matrix $\tilde{J}_{IN}(z^\star)^{-1}J(z^\star) - \mathbb{1}$ satisfy*

$$\sigma\left(\tilde{J}_{IN}(z^\star)^{-1}J(z^\star) - \mathbb{1}\right) = \sigma\left(\tilde{J}_{GN}(z^\star)^{-1}J(z^\star) - \mathbb{1}\right). \tag{5.41}$$

Proof. Let us define the eigenvalues θ of the iteration matrix $\tilde{J}_{\mathrm{IN}}(z^\star)^{-1}J(z^\star) - \mathbb{1}$ as the zeros of

$$\det\left(\tilde{J}_{\mathrm{IN}}(z^\star)^{-1}J(z^\star) - (\theta+1)\mathbb{1}\right) = 0, \tag{5.42}$$

which, given that the Jacobian approximation $\tilde{J}_{\mathrm{IN}}$ is invertible, this is equivalent to

$$\det\left(J(z^\star) - (\theta+1)\tilde{J}_{\mathrm{IN}}(z^\star)\right) = 0. \tag{5.43}$$

This block matrix then reads as

$$J(z^\star) - (\theta+1)\tilde{J}_{\mathrm{IN}}(z^\star) = \begin{bmatrix} \nabla_w^2\mathcal{L} - (\theta+1)H & \left(\frac{\partial F}{\partial w} - (\theta+1)A\right)^\top + \theta E^\top & -\theta P_{\mathcal{A}}^\top \\ \left(\frac{\partial F}{\partial w} - (\theta+1)A\right) + \theta E & & \\ -\theta P_{\mathcal{A}} & & \end{bmatrix}. \tag{5.44}$$

The result follows from Theorem 5.2.5 that claims the following asymptotic result for the block-TR1 based Jacobian approximation

$$\lim_{k\to\infty}\left(A^k - \frac{\partial F}{\partial w}(w^*)\right)N = \left(A - \frac{\partial F}{\partial w}(w^*)\right)N = 0, \tag{5.45}$$

where N is a null space matrix with appropriate dimensions and orthonormal column vectors such that $N^\top N = \mathbb{1}$ and $P_{\mathcal{A}}\,N = 0$.

We rewrite Eq. (5.43) as follows

$$\begin{aligned} &\det\left(J(z^\star) - (\theta+1)\tilde{J}_{\mathrm{IN}}(z^\star)\right) = \\ &(-\theta)^{2\,n_A}\det\left(\begin{bmatrix} \nabla_w^2\mathcal{L} - (\theta+1)H & \left(\frac{\partial F}{\partial w} - (\theta+1)A\right)^\top + \theta E^\top & P_{\mathcal{A}}^\top \\ \left(\frac{\partial F}{\partial w} - (\theta+1)A\right) + \theta E & & \\ P_{\mathcal{A}} & & \end{bmatrix}\right). \end{aligned} \tag{5.46}$$

It can be verified that $\det\left(J(z^\star) - (\theta+1)\tilde{J}_{\mathrm{IN}}(z^\star)\right) = 0$ holds for $\theta = 0$ with an algebraic multiplicity of $2\,n_A$ as well as for the values of θ that satisfy

$$\begin{aligned} &\det\left(\begin{bmatrix} N^\top & \mathbb{0} \\ \mathbb{0} & \mathbb{1} \end{bmatrix}\begin{bmatrix} \nabla_w^2\mathcal{L} - (\theta+1)H & \left(\frac{\partial F}{\partial w} - (\theta+1)A\right)^\top + \theta E^\top \\ \left(\frac{\partial F}{\partial w} - (\theta+1)A\right) + \theta E & \mathbb{0} \end{bmatrix}\begin{bmatrix} N & \mathbb{0} \\ \mathbb{0} & \mathbb{1} \end{bmatrix}\right) \\ &= \det\left(\begin{bmatrix} N^\top\Delta H\,N & N^\top\left(\frac{\partial F}{\partial w} - (\theta+1)A\right)^\top + \theta N^\top E^\top \\ \left(\frac{\partial F}{\partial w} - (\theta+1)A\right)N + \theta E\,N & \mathbb{0} \end{bmatrix}\right) \\ &= (-\theta)^{2\,n_F}\det\left(\begin{bmatrix} N^\top\Delta H\,N & N^\top\left(\frac{\partial F}{\partial w} - E\right)^\top \\ \left(\frac{\partial F}{\partial w} - E\right)N & \mathbb{0} \end{bmatrix}\right) = 0, \end{aligned} \tag{5.47}$$

in the limit for $k \to \infty$, where the compact notation $\Delta H := (\nabla^2_w \mathcal{L} - (\theta+1)H)$ has been used for the Gauss-Newton Hessian approximation. Therefore, the eigenvalues of the iteration matrix $\tilde{J}_{\mathrm{IN}}(z^\star)^{-1}J(z^\star) - \mathbb{1}$ for the proposed block-TR1 approach, evaluated at a regular KKT point, are equal to the eigenvalues of the iteration matrix $\tilde{J}_{\mathrm{GN}}(z^\star)^{-1}J(z^\star) - \mathbb{1}$ for the exact Jacobian based Gauss-Newton method. The latter can be verified by performing the same sequence of transformations for the equation $\det\Big(J(z^\star) - (\theta+1)\tilde{J}_{\mathrm{GN}}(z^\star)\Big) = 0$.

□

Corollary 5.2.9. *Based on Lemma 5.2.8, the linear contraction rate for the block-TR1 based optimization algorithm coincides with the linear contraction rate of the exact Jacobian based Gauss-Newton method $\kappa^\star_{\mathrm{BTR1}} = \kappa^\star_{\mathrm{GN}}$, when the iterates are sufficiently close to the regular KKT point $z^\star := (w^\star, \lambda^\star, \mu^\star)$.*

Superlinear convergence for SQP with quasi-Newton Hessian updates

Even though the majority of this article is focused on the generalized Gauss-Newton method for nonlinear least squares type optimization problems that occur frequently in predictive control applications, note that superlinear convergence results can be recovered when a block-structure preserving quasi-Newton method is additionally used to approximate the Hessian of the Lagrangian. For example, let us consider the following lemma that represents a block-structured or *partitioned* version [111, 110] of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [14] or the symmetric rank-one (SR1) formula [71, 135] to approximate the block-diagonal Hessian matrix.

Theorem 5.2.10. *Given (AS1)-(AS6) in Assumption 5.2.3, then the following holds for each Hessian block matrix approximation*

$$\lim_{k\to\infty} \left\| \left(H_i^k - \nabla^2_{w_i w_i}\mathcal{L}(w_i^\star, \lambda_i^\star)\right) N_i \right\| = 0, \tag{5.48}$$

$i = 0, \ldots, N-1$, such that the following holds for the complete Hessian approximation

$$\lim_{k\to\infty} \left\| \left(H^k - \nabla^2_{ww}\mathcal{L}(w^\star, \lambda^\star)\right) N \right\| = 0. \tag{5.49}$$

Theorem 5.2.10 on the convergence of a separable quasi-Newton type Hessian approximation method in combination with our main result in Theorem 5.2.5 on the block-structured quasi-Newton type Jacobian update formula can be used directly to prove the following result on convergence of the reduced KKT matrix.

Theorem 5.2.11. *Given (AS1)-(AS6) in Assumption 5.2.3, the following holds*

$$\lim_{k\to\infty} \left\| \begin{bmatrix} N^\top H^k N & N^\top A^{k\top} \\ A^k N & \mathbb{0} \end{bmatrix} - \begin{bmatrix} N^\top \nabla^2_{ww}\mathcal{L}(w^\star, \lambda^\star) N & N^\top \frac{\partial F}{\partial w}(w^\star)^\top \\ \frac{\partial F}{\partial w}(w^\star) N & \mathbb{0} \end{bmatrix} \right\| = 0. \tag{5.50}$$

Based on Theorem 5.2.2, the above result ensures q-superlinear convergence of the SQP iterates when using a quasi-Newton method to update both the block-structured Hessian and Jacobian matrices. The proof for Theorem 5.2.11, based on the intermediate convergence results in Theorem 5.2.5 and 5.2.10 can be found in [81].

5.3 Lifted Collocation Algorithm with Block-TR1 Jacobian Updates

As mentioned earlier, implicit integration schemes are often used in direct optimal control because of their relatively high order of accuracy and their improved numerical stability properties [118]. More specifically, problem formulations based on a system of stiff and/or implicit differential or differential-algebraic equations require the use of an implicit integration scheme. Collocation methods are a popular family of implicit Runge-Kutta methods. This section presents a novel lifted collocation algorithm based on tailored block-TR1 Jacobian updates. The standard lifted collocation method with exact Jacobian information was proposed in [207] as a structure-exploiting implementation of direct collocation, even though it shows similarities to multiple shooting.

Direct collocation for nonlinear optimal control

In direct transcription methods, such as direct collocation [41, 46], the integration scheme and its intermediate variables are directly made part of the nonlinear optimization problem. In this context, where the simulation routine is defined implicitly as part of the equality constraints in the dynamic optimization problem, one typically relies on implicit integration schemes for their relatively high order of accuracy and improved numerical stability properties. The discrete-time optimal control problem can generally be written as

$$\begin{aligned}
\min_{X,U,K} \quad & \sum_{i=0}^{N-1} l_i(x_i, u_i) + l_N(x_N) && && \text{(5.51a)}\\
\text{s.t.} \quad & \hat{x}_0 \;=\; x_0, && && \text{(5.51b)}\\
& x_i + B_i\, K_i \;=\; x_{i+1}, && i = 0, \ldots, N-1, && \text{(5.51c)}\\
& G_i(x_i, u_i, K_i) \;=\; 0, && i = 0, \ldots, N-1, && \text{(5.51d)}\\
& P_i\, w_i \leq p_i, && i = 0, \ldots, N, && \text{(5.51e)}
\end{aligned}$$

where the additional trajectory $K = [K_0^\top, \ldots, K_{N-1}^\top]^\top$ denotes the intermediate variables of the numerical integration method. These variables are defined implicitly by the equations in (5.51d), such that the continuity condition reads as in Eq. (5.51c). More specifically, the Jacobian $\frac{\partial G_i}{\partial K_i}(\cdot)$ will generally be invertible for an integration scheme applied to a well-defined set of differential equations in (5.1c). A popular approach of this type is better known

as direct collocation [44]. It relies on a collocation method, a subclass of implicit Runge-Kutta (IRK) methods [118], to accurately discretize the continuous time dynamics. In this case, the equations in (5.51d) define the collocation polynomial on each control interval $i = 0, \ldots, N-1$.

In a similar fashion as described in the previous section, the adjoint-based SQP method can be applied directly to the direct collocation problem in (5.51) by solving the following convex QP subproblem at each iteration

$$\min_{\Delta W,\, \Delta K} \sum_{i=0}^{N} \frac{1}{2} \Delta w_i^\top H_i^k \, \Delta w_i \; + \; h_i^{c^\top} \begin{bmatrix} \Delta w_i \\ \Delta K_i \end{bmatrix} \tag{5.52a}$$

$$\text{s.t.} \quad \Delta x_0 \; = \; \hat{x}_0 - x_0^k, \tag{5.52b}$$

$$e_i^k + \Delta x_i + B_i \, \Delta K_i \; = \; \Delta x_{i+1}, \qquad i = 0, \ldots, N-1, \tag{5.52c}$$

$$c_i^k + D_i^k \Delta w_i + C_i^k \Delta K_i \; = \; 0, \qquad i = 0, \ldots, N-1, \tag{5.52d}$$

$$P_i \, \Delta w_i \leq p_i^k, \qquad i = 0, \ldots, N, \tag{5.52e}$$

based on $c_i^k := G_i(w_i^k, K_i^k)$, $e_i^k := x_i^k + BK_i^k - x_{i+1}^k$, and the Jacobian approximations $D_i^k \approx \frac{\partial G_i}{\partial w_i}(w_i^k, K_i^k)$ and $C_i^k \approx \frac{\partial G_i}{\partial K_i}(w_i^k, K_i^k)$. The corresponding gradient correction reads as

$$h_i^c := \begin{bmatrix} \nabla_{w_i} l(w_i^k) + \left(\frac{\partial G_i}{\partial w_i}(w_i^k, K_i^k) - D_i^k \right)^\top \omega_i^k \\ \left(\frac{\partial G_i}{\partial K_i}(w_i^k, K_i^k) - C_i^k \right)^\top \omega_i^k \end{bmatrix}, \tag{5.53}$$

where ω_i^k denotes the current value of the multipliers for the nonlinear constraints in (5.51d) and λ_i^k again denotes the multipliers for the continuity constraints in (5.51c).

Tailored structure exploitation for direct collocation

As mentioned earlier, the Jacobian matrix $\frac{\partial G_i}{\partial K_i}$ for the collocation equations needs to be invertible. Therefore, given an invertible approximation $C_i^k \approx \frac{\partial G_i}{\partial K_i}(w_i^k, K_i^k)$, we can rewrite the linearized expression in Eq (5.52d) as follows

$$\Delta K_i = -C_i^{k^{-1}} \left(c_i^k + D_i^k \Delta w_i \right). \tag{5.54}$$

By substituting the above expression for ΔK_i back into the direct collocation structured QP in (5.52), one obtains the condensed but equivalent formulation

$$\min_{\Delta W} \quad \sum_{i=0}^{N} \frac{1}{2} \Delta w_i^\top H_i^k \, \Delta w_i \; + \; \tilde{h}_i^{c^\top} \Delta w_i \tag{5.55a}$$

$$\text{s.t.} \quad \Delta x_0 \; = \; \hat{x}_0 - x_0^k, \tag{5.55b}$$

$$d_i^k + \Delta x_i - B_i \, C_i^{k^{-1}} D_i^k \Delta w_i \; = \; \Delta x_{i+1}, \quad i = 0, \ldots, N-1, \tag{5.55c}$$

$$P_i \, \Delta w_i \leq p_i^k, \qquad i = 0, \ldots, N, \tag{5.55d}$$

where $d_i^k = e_i^k - B_i\, C_i^{k^{-1}} c_i^k$ is defined and the condensed gradient reads as

$$\tilde{h}_i^c = \nabla_{w_i} l(w_i^k) + \left(\frac{\partial G_i}{\partial w_i} - \frac{\partial G_i}{\partial K_i}\, C_i^{k^{-1}} D_i^k \right)^\top \omega_i^k, \tag{5.56}$$

given the original gradient correction in (5.53).

Note that the resulting QP formulation in Eq. (5.55) is of the same problem dimensions and exhibits the same sparsity as the multiple shooting structured QP subproblem in Eq. (5.7). Therefore, state of the art block-structured QP solvers can be used, for which an overview can be found in [94]. After solving the condensed QP in (5.55), the collocation variables can be obtained from the expansion step in Eq. (5.54). Based on the optimality conditions of the original direct collocation structured QP in (5.52), the corresponding Lagrange multipliers can be updated as follows

$$\omega_i^{k+1} = \omega_i^k - C_i^{k^{-\top}} \left(\frac{\partial G_i}{\partial K_i}^\top \omega_i^k + B_i^\top \lambda_i^{k+1} \right), \tag{5.57}$$

where λ_i^{k+1} denote the new values of the Lagrange multipliers for the continuity conditions in (5.55c) or in (5.52c).

Block-TR1 Jacobian update for lifted collocation

The block-TR1 update formula from Eq. (5.11) can be readily applied to the direct collocation equations, resulting in

$$[D_i^{k+1}\ C_i^{k+1}] = [D_i^k\ C_i^k] + \alpha_i^k \left(y_i^k - [D_i^k\ C_i^k]\, s_i^k \right) \left(\gamma_i^{k^\top} - \sigma_i^{k^\top} [D_i^k\ C_i^k] \right), \tag{5.58}$$

where the quantities $\gamma_i^{k^\top} = \sigma_i^{k^\top} \frac{\partial G_i}{\partial (w_i, K_i)}(w_i^{k+1}, K_i^{k+1})$ and $\sigma_i^k = \omega_i^{k+1} - \omega_i^k$ are defined. In addition, $s_i^k := \begin{bmatrix} w_i^{k+1} - w_i^k \\ K_i^{k+1} - K_i^k \end{bmatrix}$ and $y_i^k = G_i(w_i^{k+1}, K_i^{k+1}) - G_i(w_i^k, K_i^k)$ is defined. In order to use this block-TR1 update formula in combination with the lifted collocation method, one needs to be able to efficiently form the condensed QP in Eq. (5.55). For this purpose, we need to avoid the costly computations of the inverse matrix $C_i^{k^{-1}}$ as well as the matrix-matrix multiplication $C_i^{k^{-1}} D_i^k$. In what follows, we present a procedure to directly obtain a rank-one update formula for the inverse matrix $C_i^{k+1^{-1}}$ and for the corresponding product $E_i^{k+1} := C_i^{k+1^{-1}} D_i^{k+1}$.

Avoiding expensive matrix-matrix operations

Based on the Sherman-Morrison formula, one can directly update the matrix inverse given the previous invertible approximation $C_i^{k^{-1}} \approx \frac{\partial G_i}{\partial K_i}^{-1}$. Let us first rewrite the block-TR1 update from Eq. (5.58) as follows

$$D_i^{k+1} = D_i^k + \alpha_i^k\, \rho_i^k\, \tau_{\mathrm{D},i}^{k^\top} \quad \text{and} \quad C_i^{k+1} = C_i^k + \alpha_i^k\, \rho_i^k\, \tau_{\mathrm{C},i}^{k^\top}, \tag{5.59}$$

where $\rho_i^k = y_i^k - [D_i^k \; C_i^k] s_i^k$ and $[\tau_{\mathrm{D},i}^{k^\top} \; \tau_{\mathrm{C},i}^{k^\top}] = \gamma_i^{k^\top} - \sigma_i^{k^\top} [D_i^k \; C_i^k]$. The Sherman-Morrison formula then reads as

$$C_i^{k+1^{-1}} = C_i^{k^{-1}} - \alpha_i^k \beta_i^k \, C_i^{k^{-1}} \rho_i^k \tau_{\mathrm{C},i}^{k^\top} C_i^{k^{-1}}, \tag{5.60}$$

where $\beta_i^k = \frac{1}{1 + \alpha_i^k \tau_{\mathrm{C},i}^{k^\top} C_i^{k^{-1}} \rho_i^k}$. Let us define $\tilde{\rho}_i^k = C_i^{k^{-1}} \rho_i^k$ such that we obtain the following update for the condensed Jacobian

$$\begin{aligned} E_i^{k+1} &= C_i^{k+1^{-1}} D_i^{k+1} = C_i^{k^{-1}} \left(D_i^k + \alpha_i^k \rho_i^k \tau_{\mathrm{D},i}^{k^\top} \right) - \alpha_i^k \beta_i^k C_i^{k^{-1}} \rho_i^k \tau_{\mathrm{C},i}^{k^\top} C_i^{k^{-1}} \left(D_i^k + \alpha_i^k \rho_i^k \tau_{\mathrm{D},i}^{k^\top} \right) \\ &= E_i^k + \alpha_i^k \tilde{\rho}_i^k \tau_{\mathrm{D},i}^{k^\top} - \alpha_i^k \beta_i^k \tilde{\rho}_i^k \tau_{\mathrm{C},i}^{k^\top} (E_i^k + \alpha_i^k \tilde{\rho}_i^k \tau_{\mathrm{D},i}^{k^\top}) \\ &= E_i^k + \alpha_i^k \tilde{\rho}_i^k \tilde{\tau}_i^{k^\top}, \end{aligned} \tag{5.61}$$

where $\tilde{\tau}_i^{k^\top} = \tau_{\mathrm{D},i}^{k^\top} - \beta_i^k \tau_{\mathrm{C},i}^{k^\top} (E_i^k + \alpha_i^k \tilde{\rho}_i^k \tau_{\mathrm{D},i}^{k^\top})$ has been defined. It is readily seen that the update for E_i^k in Eq. (5.61) is a rank-one update for the condensed Jacobian matrix. An additional damping, pivoting or splitting of the Jacobian updates [114, 160] can be introduced in order to avoid singularity and/or blow-up of the matrix. As proposed in [123], corresponding low-rank update formulas for the condensed Hessian can be obtained for a pseudospectral method based on a global collocation polynomial.

Lifted collocation SQP method with block-TR1 Jacobian updates

It is important to stress that the novel block-TR1 update formula for the condensed Jacobian matrix $E_i^{k+1} = C_i^{k+1^{-1}} D_i^{k+1}$ in Eq. (5.61) provides an efficient manner to directly compute the rank-one update to the matrices in the condensed QP formulation of Eq. (5.55), without the need for a matrix factorization, inversion and without any matrix-matrix multiplications. Instead, the proposed implementation merely requires matrix-vector multiplications and outer products, resulting in a quadratic instead of cubic computational complexity with respect to the number of optimization variables within each control interval. However, this comes at the cost of a slightly increased memory footprint, since additionally the matrices C_i^{-1} and E_i need to be stored from one iteration to the next. The implementation of the lifted block-TR1 based SQP method for direct collocation is presented in Algorithm 5.

We observe that the TR1 Jacobian updates of the lifted collocation implementation are equivalent to the updates of the direct collocation method. More specifically, the Jacobian approximation matrices are the same at each SQP iteration, regardless of whether we perform the condensing and expansion procedure for the collocation variables in the proposed lifted implementation of Algorithm 5. Therefore, the convergence properties shown in the previous section also hold for both the standard and lifted collocation based block-TR1 SQP method.

Corollary 5.3.1. *If the assumptions of Theorem 5.2.1 and Assumption 5.2.3 hold, then the lifted collocation SQP method with block-wise TR1 Jacobian updates in Algorithm 5, with a Gauss-Newton Hessian approximation, produces iterates $\{w^k, \lambda^k, \mu^k\}$ that converge q-linearly within a neighbourhood around the KKT point (w^*, λ^*, μ^*) of the NLP.*

Algorithm 5 One lifted collocation SQP iteration with block-wise TR1 updates.

Input: $w_i^k = (x_i^k, u_i^k)$, K_i^k, λ_i^k, ω_i^k, C_i^k, D_i^k, $C_i^{k^{-1}}$ and E_i^k.

`Problem linearization and QP preparation`

1: Formulate the QP in (5.55) with Jacobian matrices E_i^k, Gauss-Newton Hessian approximations H_i^k and vectors d_i^k, p_i^k and $\tilde{h}_i^c$ in (5.56) for $i = 0, \ldots, N-1$.

`Computation of Newton-type step direction`

2: Solve the QP subproblem in Eq. (5.55) to update optimization variables: $w_i^{k+1} \leftarrow w_i^k + \Delta w_i^k$ and $\lambda_i^{k+1} \leftarrow \lambda_i^k + \Delta\lambda_i^k$. ▷ full step

`Block-wise TR1 Jacobian updates`

3: **for** $i = 0, \ldots, N-1$ **do in parallel**

4: Choose $\alpha_i^k = \alpha_{\mathrm{F},i}^k$ or $\alpha_i^k = \alpha_{\mathrm{A},i}^k$ via some decision rule.

5: $K_i^{k+1} \leftarrow K_i^k - C_i^{k^{-1}} c_i^k - E_i^k \Delta w_i^k$,

6: $\omega_i^{k+1} \leftarrow \omega_i^k - C_i^{k^{-\top}} \left(\frac{\partial G_i}{\partial K_i}^\top \omega_i^k + B_i^\top \lambda_i^{k+1} \right)$,

7: $D_i^{k+1} \leftarrow D_i^k + \alpha_i^k \, \rho_i^k \, \tau_{\mathrm{D},i}^{k^\top}$ and $C_i^{k+1} \leftarrow C_i^k + \alpha_i^k \, \rho_i^k \, \tau_{\mathrm{C},i}^{k^\top}$,

8: $C_i^{k+1^{-1}} \leftarrow C_i^{k^{-1}} - \alpha_i^k \beta_i^k \, \tilde{\rho}_i^k \tau_{\mathrm{C},i}^{k^\top} C_i^{k^{-1}}$,

9: $E_i^{k+1} \leftarrow E_i^k + \alpha_i^k \tilde{\rho}_i^k \tilde{\tau}_i^{k^\top}$.

10: **end for**

Output: w_i^{k+1}, K_i^{k+1}, λ_i^{k+1}, ω_i^{k+1}, C_i^{k+1}, D_i^{k+1}, $C_i^{k+1^{-1}}$ and E_i^{k+1}.

Proof. It follows from the equivalence of the SQP iterations between the direct and lifted collocation formulation based on the numerical condensing and expansion of the collocation variables in Eq. (5.54). In particular, the direct collocation QP subproblem (5.52) is a special case of the QP formulation in (5.7), with additional intermediate variables and corresponding equations. The block-TR1 Jacobian matrix convergence results of Theorem 5.2.5 therefore hold for direct collocation as well as for the proposed lifted implementation in Algorithm 5. □

5.4 Numerical Case Studies of Nonlinear Model Predictive Control

In this section, we illustrate numerically how the proposed block-TR1 SQP method can be used in the context of NMPC using an algorithm implementation based on the real-time iterations (RTI), as proposed originally in [79] with exact Jacobian information. The approach is based on one block-TR1 SQP iteration per control time step, and using a continuation-based warm starting of the state and control trajectories from one time step to the next [122]. Each iteration consists of two steps:

1. *Preparation phase*: discretize and linearize the system dynamics, linearize the remain-

ing constraint functions, and evaluate the quadratic objective approximation to build the optimal control structured QP subproblem.

2. *Feedback phase*: solve the QP to update the current values for all optimization variables and obtain the next control input to apply feedback to the system.

The proposed block-wise TR1 based Jacobian updates in Algorithm 4 and 5 become part of the preparation step, in order to construct the linearized continuity equations. Therefore, the feedback step remains unchanged and the Jacobian updates do not affect the computational delay between obtaining the new state estimate and applying the next control input value to the system.

We validate the closed-loop performance of these novel block-TR1 based RTI algorithms by presenting numerical simulation results for two NMPC case studies. Motivated by real embedded control applications, we present the computation times for the proposed NMPC algorithms using the ARM Cortex-A53 processor in the Raspberry Pi 3. The block-sparse QP solution in the feedback phase will be carried out by the primal active-set method, called `PRESAS`, that was recently presented in [204].

NMPC for a chain of spring-connected masses

In our first case study, the control task is to return a chain of n_m masses connected with springs to its steady state, starting from a perturbed initial configuration, without hitting a wall that is placed close to the equilibrium state configuration. The mass at one end is fixed, while the control input $u(t) \in \mathbb{R}^3$ to the system is the direct force applied to the mass at the other end of the chain. The state of each free mass $x^j := [p^{j\top}, v^{j\top}]^\top \in \mathbb{R}^6$ consists in its position $p^j := [p_x^j, p_y^j, p_z^j]^\top \in \mathbb{R}^3$ and velocity $v^j \in \mathbb{R}^3$ for $j = 1, \ldots, n_m - 1$, such that the dynamic system can be described by the concatenated state vector $x(t) \in \mathbb{R}^{6(n_m-1)}$. Similar to the work in [207], the nonlinear chain of masses can be used to validate the computational performance and scaling of an optimal control algorithm for a range of numbers of masses n_m, resulting in a range of different problem dimensions. The nonlinear system dynamics and the resulting optimal control problem formulation can be found in [253].

Local convergence: Gauss-Newton SQP with block-TR1 Jacobian updates

We illustrate the impact of the proposed block-wise TR1 Jacobian updates on the local convergence rate of the resulting inexact adjoint-based SQP algorithm. Figure 5.1 shows a comparison of the convergence between different SQP variants for the solution of the nonlinear chain of masses OCP. In particular, the comparison includes the exact Jacobian-based SQP method, the standard dense TR1 update [113], and the good and bad Broyden update formulas [58]. For the proposed block-TR1 based SQP implementation, the figure illustrates both the adjoint and forward variant by using, respectively, the scaling factor in (5.12) and (5.13). The performance of the block-TR1 method is additionally illustrated for an implementation where α_i is chosen dynamically, depending on which of the two variants

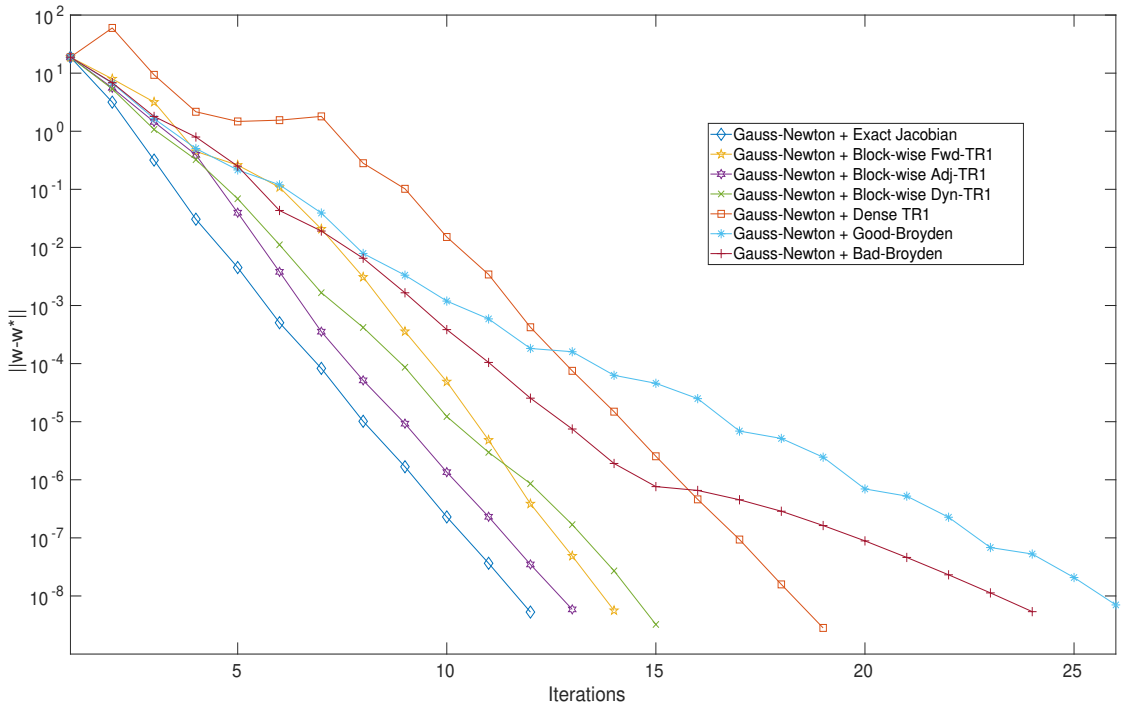

Figure 5.1: Local convergence analysis: comparison between different variants of the inexact adjoint-based SQP method as described in Algorithm 5 based on lifted collocation, using either the exact Jacobian or different quasi-Newton type Jacobian update formulas for the nonlinear chain of 6 masses.

results in the largest denominator in order to avoid the need to skip a block-wise Jacobian update.

It is known that an exact Jacobian-based SQP method with Gauss-Newton type Hessian approximation results in locally linear convergence, for which the asymptotic contraction rate depends on the optimal residual value in the least squares type objective [186]. It can be observed in Figure 5.1 that all three variants of the proposed block-wise TR1 update formula result in the same asymptotic rate of convergence as for the exact Jacobian based algorithm, i.e., the rate of convergence appears to be the same close to the local solution of the NLP. Note that this confirms numerically the result of Corollary 5.2.9. In addition, the block-wise TR1 Jacobian updates result in a smaller total number of SQP iterations, compared to the standard dense Jacobian update formulas for the particular example in Figure 5.1. In the latter case, the direct application of a standard rank-one update formula destroys the block sparsity in the QP subproblems and is therefore computationally unattractive.

Computational timing results for block-TR1 based lifted collocation

Figure 5.2 illustrates the computation times of both the preparation and feedback steps of an NMPC implementation for a chain of $n_m = 2, \ldots, 8$ masses, using the lifted collo-

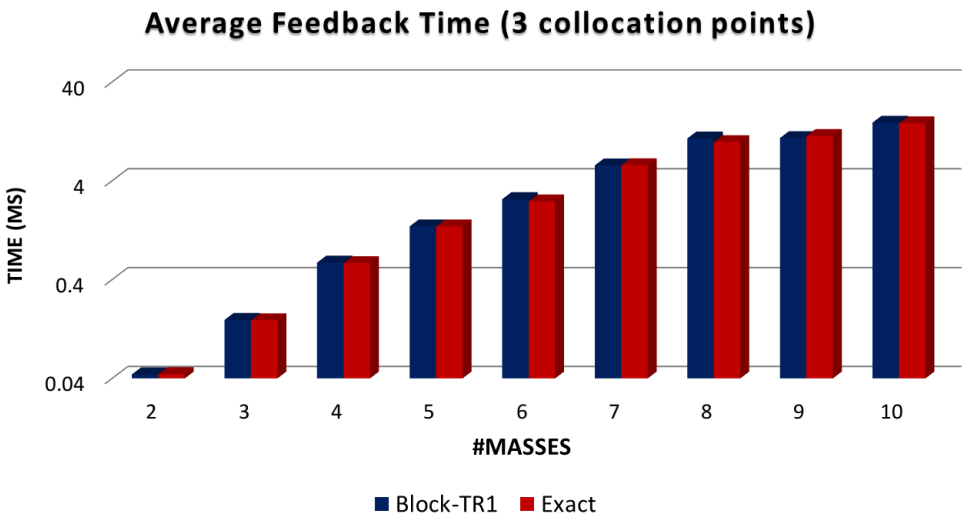

Figure 5.2: Comparison of the average preparation and feedback computation times (in ms, logarithmic scale): block-TR1 versus exact Jacobian based lifted collocation SQP method. [1]

Table 5.1: Average computation times (in ms) for NMPC on a chain of $n_m = 6$ masses, i.e., 30 differential states (4 Gauss collocation nodes versus 10 steps of RK4).

	Explicit (RK4 in Alg. 4)			Implicit (GL4 in Alg. 5)		
	exact	block-TR1		exact	block-TR1	
Linearization	32.36	5.33	**16%**	291.37	35.99	**12%**
QP solution	23.22	37.82		26.33	27.86	
Total RTI step	56.39	43.99	78%	318.58	64.69	20%

cation based SQP method in Algorithm 5. It can be observed that the preparation time scales quadratically with the number of states for the block-TR1 implementation, instead of the cubic computational complexity when using the exact Jacobian. More specifically, the Jacobian evaluation, the factorization and matrix-matrix multiplications are replaced by adjoint differentiation sweeps and matrix-vector operations in Algorithm 5. On the other hand, the feedback time remains essentially the same because, after the linearization and QP preparation, both approaches lead to the solution of a similarly structured QP in Eq. (5.7) or (5.55).

Table 5.1 provides a more detailed comparison between the exact Jacobian and the proposed block-TR1 variant of the real-time iterations for NMPC, using a sequential algorithm implementation on an ARM Cortex-A53 processor. The table shows these results for both the explicit Runge-Kutta method of order 4 (RK4) in combination with Algorithm 4 and

[1] The computation times in Figure 5.2 have been obtained using a sequential algorithm implementation on an Intel i7-7700k processor @ 4.20 GHz on Windows 10 with 64 GB of RAM.

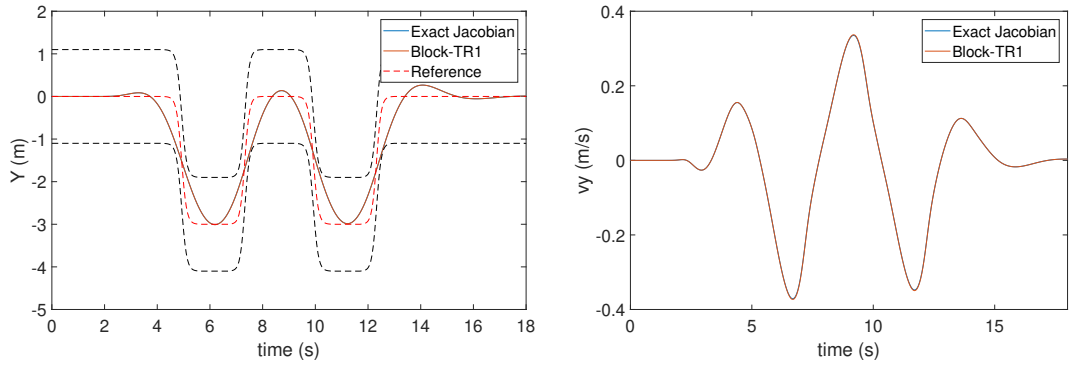

Figure 5.3: Closed-loop NMPC performance of two double lane changes at a vehicle speed of 10 m/s on snow-covered road conditions, using model parameters from [38] in the nonlinear OCP formulation [201]. These time trajectories of Y and v_y, respectively, denote the position and velocity along the Y-axis.

using the implicit 4-stage Gauss-Legendre (GL4) method within Algorithm 5. The proposed block-TR1 algorithm results in a computational speedup of about factor $6-8$ for the problem linearization step. In order to obtain a relatively fair comparison, the number of integration steps for RK4 has been chosen such that the numerical accuracy is close to that of the 4-stage GL method. However, since the system dynamics for the chain of masses are non-stiff, an explicit integration scheme should instead typically perform better in terms of computational efficiency.

NMPC for vehicle control on a snow-covered road

Our second case study considers NMPC for real-time vehicle control as motivated by automotive applications in autonomous driving. The nonlinear optimal control problem formulation is based on single-track vehicle dynamics with a Pacejka-type tire model [201]. The experimentally validated model parameters can be found in [38]. As often the case in practice, these vehicle dynamics are rather stiff such that an implicit integration scheme should preferably be used. Therefore, it forms an ideal case study for the proposed lifted collocation based RTI method of Algorithm 5. Let us perform the closed-loop NMPC simulations as presented in [201], but using the proposed block-TR1 based RTI implementation. We carried out numerical simulations for two successive double lane changes on snow-covered road conditions. The resulting closed-loop trajectories for both the exact Jacobian and the block-TR1 method are indistinguishable from each other, as illustrated in Figure 5.3.

The corresponding computation times for a sequential algorithm implementation on the ARM Cortex-A53 processor are illustrated in Table 5.2. Because of the relatively stiff system dynamics, the proposed block-TR1 lifted collocation method from Algorithm 5 becomes attractive and additionally provides a computational speedup of about factor 3 over the

Table 5.2: Average computation times (in ms) for vehicle control based on a single-track vehicle model within NMPC (4 Gauss collocation nodes versus 30 steps of RK4).

	Explicit (RK4 in Alg. 4)			Implicit (GL4 in Alg. 5)		
	exact	block-TR1		exact	block-TR1	
Linearization	106.73	75.78	**71%**	52.22	18.27	**35%**
QP solution	4.46	4.51		4.59	4.72	
Total RTI step	111.79	80.94	72%	57.43	23.64	41%

standard exact Jacobian based implementation. Note that, even though the Raspberry Pi 3 is not an embedded processor by itself, it uses an ARM core of the same type as those that are used by multiple high-end automotive microprocessors. Therefore, the proposed algorithm implementation as well as the corresponding numerical results form a motivation for real-time embedded control applications that involve a relatively large, implicit and/or stiff system of differential equations.

Chapter 6

Advanced applications of Real-time MPC

In this Chapter we will develop two extensions of the real-time MPC algorithm we presented on the preceding chapter. First we will focus on pseudospectral methods which are special type of direct method to solving continuous-time optimal control problems, in comparison to collocation methods discussed so far. Second, we will extend our non-linear MPC to the mixed-Integer environment, that is to optimal control problems where some states/controls are discrete decision variables (that is take integer values). This environment is typical Hybrid Systems formulations and we will leverage our structure exploiting algorithm in order to design an efficient mixed-integer optimal control solver.

Pseudospectral and collocation methods form a popular direct approach to handling and solving continuous-time optimal control problems. Lifted Newton-type algorithms have been proposed as a computationally efficient way to implement online pseudospectral methods for nonlinear model predictive control (NMPC). The present paper extends this work based on a rank-one Jacobian update formula for the nonlinear system dynamics. In addition, we describe an algorithm implementation where this rank-one Jacobian update can be used directly to compute a low-rank update to the condensed Hessian, resulting in an overall quadratic computational complexity for each iteration. A preliminary C code implementation is shown to allow considerable numerical speedups for the optimal control case study of the nonlinear chain of masses.

Mixed-integer model predictive control (MI-MPC) requires the solution of a mixed-integer quadratic program (MIQP) at each sampling instant under strict timing constraints, where part of the state and control variables can only assume a discrete set of values. Several applications in automotive, aerospace and hybrid systems are practical examples of how such discrete-valued variables arise. We utilize the sequential nature and the problem structure of MI-MPC in order to provide a branch-and-bound algorithm that can exploit not only the block-sparse optimal control structure of the problem but that can also be warm started by propagating information from branch-and-bound trees and solution paths at previous time steps. We illustrate the computational performance of the proposed algorithm and compare

against current state-of-the-art solvers for multiple MPC case studies, based on a preliminary implementation in and C code.

The chapter is organized as follows: On section 6.1 we summarize collocation schemes and their use in direct optimal control methods, from the previous chapter. Then we extend the lifted Newton implementation of a pseudospectral method with quasi-Newton Jacobian updates and low-rank Hessian updates. The proposed algorithms are illustrated based on numerical results of NMPC for the chain of masses in Section 6.2. Then, on section 6.3 we present the basic idea of branch-and-bound methods for mixed-integer programming and presolve techniques in the context of mixed-integer optimal control. The resulting MI-MPC algorithm and its tailored warm-starting strategies are discussed in Section 6.4 and its performance is illustrated based on multiple numerical case studies in Section 6.5.

6.1 Quasi-Newton Jacobian and Hessian Updates for Pseudospectral based NMPC

There has been an increasing interest in using dynamic optimization for real-time applications, i.e., in the context of model predictive control (MPC) and moving horizon estimation (MHE) [211]. For this purpose, an optimal control problem (OCP) needs to be solved at each time instant, under strict timing constraints. Tailored continuation based online optimization algorithms have been developed for real-time optimal control as discussed in [78]. A popular example is the real-time iteration (RTI) algorithm [79], an online variant of sequential quadratic programming (SQP) for nonlinear MPC (NMPC) applications.

Again our aim is at solving the following OCP formulation in continuous time

$$\begin{aligned}
\min_{x(\cdot),\,u(\cdot)} \quad & \int_0^T \|F(x(t),u(t))\|_2^2\,dt && && \text{(6.1a)}\\
\text{s.t.} \quad & 0 = x(0) - \hat{x}_0, && && \text{(6.1b)}\\
& 0 = f(\dot{x}(t), x(t), u(t)), && \forall t \in [0,T], && \text{(6.1c)}\\
& 0 \geq h(x(t), u(t)), && \forall t \in [0,T], && \text{(6.1d)}\\
& 0 \geq r(x(T)), && && \text{(6.1e)}
\end{aligned}$$

where $x(t) \in \mathbb{R}^{n_x}$ denote the differential states and $u(t) \in \mathbb{R}^{n_u}$ are the control inputs at time t. The objective in Eq. (6.1a) consists of a nonlinear least squares type Lagrange term. The problem depends on the parameter value $\hat{x}_0$ through the initial condition of Eq. (6.1b). The nonlinear dynamics in Eq. (6.1c) are described by an implicit system of ordinary differential equations (ODE), even though this can generally be extended to differential-algebraic equations (DAE) of index 1. Respectively, Eqs. (6.1d) and (6.1e) denote the path and terminal inequality constraints.

A popular direct technique for solving the optimal control problem in (6.1) is based on orthogonal collocation, where a distinction is made between the use of *local* and *global*

collocation polynomials [209]. In local collocation, also referred to as direct collocation [41, 46], one uses piecewise polynomials which are typically of a fixed degree. Pseudospectral methods form an extreme case of such an approach, by mainly increasing or decreasing the degree of a global collocation polynomial. Given a smooth and well-behaved optimal control solution, this approximation is known to converge at an exponential rate [209]. Another reason for their popularity is that any collocation method can readily be applied to problems involving stiff or implicit systems of differential equations. Orthogonal collocation methods are typically used, based on the roots of Chebyshev or Legendre polynomials. We focus on Legendre collocation methods, which employ a quadrature rule based on either Gauss, Radau or Lobatto points [118].

It has been shown how collocation schemes can be used within a lifted Newton-type implementation, which bridges the gap between direct collocation and direct multiple shooting [48] as discussed in [208]. Recently, a lifted Newton-type optimization algorithm for pseudospectral based NMPC has been proposed in [202], based on a tailored Jacobian approximation technique and the inexact Newton method with iterated sensitivities (INIS) from [203]. In addition, adjoint based quasi-Newton Jacobian update schemes for constrained optimization [82, 112] have effectively been applied to the lifted collocation algorithm in [122].

Our work extends both the work in [202] and in [122] by proposing a tailored quasi-Newton type Jacobian and Hessian update scheme with numerical condensing and expansion of the collocation variables, resulting in a pseudospectral based NMPC algorithm with an overall quadratic computational complexity. Because of our focus on real-time NMPC applications, this work does not aim to compete directly with general-purpose state of the art nonlinear optimization solvers, such as `Ipopt` [248] or `SNOPT` [104].

Direct Optimal Control Methods

Direct optimal control [48] tackles the continuous time OCP (6.1) by forming a discrete approximation and solving the resulting NLP. We adopt the differential formulation of a collocation method [45], to be consistent with the notation in [202] on lifted Newton-type collocation and with the literature on Runge-Kutta methods.

Collocation based Numerical Simulation

In order to arrive at a compact notation, we consider a collocation polynomial of degree N for the parametrization of both the state and control profile. Let us define the time transformation $\tau := \frac{t}{T}$, such that $\tau \in [0, 1]$ for $t \in [0, T]$. The polynomial approximation for the differential state can then be obtained as follows

$$p_x(c) = x_0 + T \sum_{i=1}^{N} k_i \int_0^c \ell_i(\tau)\, \mathrm{d}\tau, \tag{6.2}$$

where $\ell_i(\tau)$ denote the Lagrange interpolating polynomials, given a set of collocation nodes $0 \leq c_i \leq 1$ for $i = 1, \ldots, N$ and the corresponding stage values k_i and u_i, respectively, for the

state derivatives and the control inputs. Note that the parametrized control profile reads as $p_u(c) = \sum_{i=1}^{N} \ell_i(c) u_i$ such that $p_u(c_i) = u_i$, for $i = 1, \ldots, N$. The collocation variables k_i are defined by imposing the system dynamics in Eq. (6.1c):

$$G(x_0, U, K) = \begin{bmatrix} f(k_1, x_0 + T \sum_{j=1}^{N} a_{1j} k_j, u_1) \\ \vdots \\ f(k_N, x_0 + T \sum_{j=1}^{N} a_{Nj} k_j, u_N) \end{bmatrix} = 0, \tag{6.3}$$

which denotes the nonlinear system of collocation equations and where $a_{ij} = \int_0^{c_i} \ell_j(\tau) \, \mathrm{d}\tau$ is defined. The numerical simulation result at the end of the interval reads as

$$x(T) \approx x_T(K) = x_0 + T \sum_{i=1}^{N} b_i k_i = p_x(1), \tag{6.4}$$

where $b_i = \int_0^1 \ell_i(\tau) \, \mathrm{d}\tau$. All collocation schemes belong to the family of implicit Runge-Kutta (IRK) methods, which are often defined based on their Butcher tableau.

Pseudospectral Optimal Control

Based on the same Gaussian quadrature rule as used for the collocation scheme in (6.4), let us define a discretization for the least squares type objective in (6.1a):

$$\int_0^T \|F(x(t), u(t))\|_2^2 \approx T \sum_{i=1}^{N} b_i \|F(x_i, u_i)\|_2^2, \tag{6.5}$$

where $x_i(K) = x_0 + T \sum_{j=1}^{N} a_{ij} k_j$. Direct transcription, of which pseudospectral methods form a special subclass, is then based on including the additional variables and equations (6.3) directly into the discrete time OCP formulation. Based on the discretized cost and by imposing the path constraints in (6.1d) at the collocation nodes, the resulting dense nonlinear program (NLP) reads as

$$\begin{aligned}
\min_{x_0, U, K} \quad & T \sum_{i=1}^{N} b_i \|F(x_i(K), u_i)\|_2^2 && && \text{(6.6a)} \\
\text{s.t.} \quad & 0 = x_0 - \hat{x}_0, && && \text{(6.6b)} \\
& 0 = f(k_i, x_i(K), u_i), && i = 1, \ldots, N, && \text{(6.6c)} \\
& 0 \geq h(x_i(K), u_i), && i = 1, \ldots, N, && \text{(6.6d)} \\
& 0 \geq r(x_T(K)), && && \text{(6.6e)}
\end{aligned}$$

where the stage values $U = [u_1^\top, \ldots, u_N^\top]^\top \in \mathbb{R}^{N n_u}$ and $K = [k_1^\top, \ldots, k_N^\top]^\top \in \mathbb{R}^{N n_x}$ are defined.

Gauss-Newton based SQP Method

An adjoint based Gauss-Newton SQP algorithm for the NLP in (6.6) relies on the solution of a quadratic program (QP) approximation in each iteration:

$$\min_{\Delta U, \Delta K} \quad T \sum_{i=1}^{N} b_i \| F_i + J_i^x \Delta x_i + J_i^u \Delta u_i \|_2^2 \tag{6.7a}$$

$$+ \bar{\omega}^\top \left[\left(\tfrac{\partial G}{\partial U} - D \right) \quad \left(\tfrac{\partial G}{\partial K} - C \right) \right] \begin{bmatrix} \Delta U \\ \Delta K \end{bmatrix} \tag{6.7b}$$

$$\text{s.t.} \qquad 0 = g + D\Delta U + C\Delta K, \tag{6.7c}$$

$$0 \geq a + A_u \Delta U + A_k \Delta K, \tag{6.7d}$$

where $F_i := F(\hat{x}_0 + T\sum_{j=1}^{N} a_{ij}\bar{k}_j, \bar{u}_i)$, $\Delta x_i = T\sum_{j=1}^{N} a_{ij}\Delta k_j$ and $g := G(\hat{x}_0, \bar{U}, \bar{K})$ are defined. Note that the initial state variable $x_0 = \hat{x}_0$ has been eliminated to arrive at a more compact notation. The values $\bar{U}, \bar{K}$ denote the current linearization point and $\bar{\omega}$ denotes the current values for the Lagrange multipliers $\omega \in \mathbb{R}^{Nn_x}$ corresponding to the collocation equations in (6.6c). The Jacobian matrices read $J_i^x = \frac{\partial F_i}{\partial x_i}$ and $J_i^u = \frac{\partial F_i}{\partial u_i}$ for the objective. The constraint Jacobian approximations $D \approx \frac{\partial G}{\partial U}(\cdot)$ and $C \approx \frac{\partial G}{\partial K}(\cdot)$ will be discussed in the following. Given these constraint Jacobian approximations, the gradient correction for the adjoint based SQP method [253] is defined as in Eq. (6.7b). The inequality constraints in (6.7d) denote an exact linearization of the path and terminal constraints in (6.6d) and (6.6e).

In embedded NMPC applications, one needs to solve the nonlinear OCP of Eq. (6.6) at each sampling instant under strict timing constraints. For this purpose, we instead use the real-time iteration (RTI) scheme [78, 79] for nonlinear MPC, which is a continuation based variant of a fixed-step SQP method. More specifically, by warm-starting the algorithm based on the (approximate) solution to the OCP at a previous time instant, only one QP subproblem of the form in (6.7) needs to be solved at each time step. The general idea is that one prefers to obtain new measurement information from the system, rather than iterating until convergence for an optimization problem that is becoming outdated.

Lifted Newton-Type Optimization wih Rank-one Jacobian Updates

Let us describe the proposed lifted Newton-type optimization algorithm for pseudospectral based NMPC, using a quasi-Newton type rank-one Jacobian update formula. One efficient way to solve the QP subproblem in (6.7) is based on the combination of condensing and expansion. This corresponds to a numerical elimination of the collocation variables, by defining the following quantities

$$\Delta\tilde{K} = -C^{-1}g \quad \text{and} \quad E = -C^{-1}D, \tag{6.8}$$

such that $\Delta K = \Delta\tilde{K} + E\Delta U$. Based on the inexact Newton step in (6.8), the subproblem can be reformulated as the following dense QP

$$\min_{\Delta U} \quad \frac{1}{2}\Delta U^\top H_c\, \Delta U + h_c^\top \Delta U \tag{6.9a}$$

$$\text{s.t.} \quad 0 \geq a_c + A_c \Delta U, \tag{6.9b}$$

where the vectors $a_c = a + A_k\Delta\tilde{K}$ and $h_c = \begin{bmatrix}\mathbb{1} & E^\top\end{bmatrix} h$ are defined and the condensed matrices read as $A_c = A_u + A_k E$ and $H_c = E^\top H E$. The condensed Gauss-Newton based objective is defined, using the Hessian matrix H and gradient vector h for the objective function in Eq. (6.7a) including the gradient correction in Eq. (6.7b).

Based on the solution of the condensed QP subproblem in (6.9), the inexact Newton (IN) method requires the additional computation of the Lagrange multipliers corresponding to the collocation equations in (6.6c). For this purpose, we use λ to denote the Lagrange multipliers for the inequality constraints in (6.9b), which are equal to those for the inequality constraints in (6.7d). Based on the optimality conditions for the QP in Eq. (6.7), using the Jacobian approximation $C \approx \frac{\partial G}{\partial K}(\cdot)$, this results in the following Newton-type update for these multipliers:

$$\Delta\omega = -C^{-\top}\left(h_k + \frac{\partial G}{\partial K}^\top \bar{\omega} + A_k^\top \bar{\lambda}^+\right), \tag{6.10}$$

where $h_k \in \mathbb{R}^{Nn_x}$ denotes the gradient of the QP objective term in (6.7a) with respect to the collocation variables. The updated multiplier values read as $\bar{\omega}^+ = \bar{\omega}^o + \Delta\omega$ and the collocation variables are updated as follows $\bar{K}^+ = \bar{K}^o + \Delta\tilde{K} + E\Delta U$, given the multiplier values $\bar{\lambda}^+$ and solution vector $\Delta U^\star$ from solving the dense QP (6.9).

Quasi-Newton Jacobian Update Formula

Unlike standard Broyden type methods [57], a two-sided rank-one (TR1) update formula has been proposed in [82, 112] as a generalization of the symmetric rank-one (SR1) update scheme in [70] for constrained optimization. The TR1 formula enjoys several benefits over classical methods, such as heredity and linear transformation invariance [112].

Let us apply the TR1 update formula to the Jacobian approximation $\begin{bmatrix}D & C\end{bmatrix} \approx \frac{\partial G(\cdot)}{\partial(U,K)}$. The key ingredient of the TR1 method is that it aims to simultaneously satisfy the *direct* secant condition

$$\begin{bmatrix}D^+ & C^+\end{bmatrix} s = y, \tag{6.11}$$

and the *adjoint* or *transposed* secant condition

$$\sigma^\top \begin{bmatrix}D^+ & C^+\end{bmatrix} = \mu^\top, \tag{6.12}$$

where we define the adjoint $\mu^\top = \sigma^\top \frac{\partial G}{\partial(U,K)}(\hat{x}_0, U^+, K^+)$, given $\sigma = \omega^+ - \omega^o$, the difference in function evaluations $y = G(\hat{x}_0, U^+, K^+) - G(\hat{x}_0, U^o, K^o)$ and $s := \begin{bmatrix}U^+ - U^o \\ K^+ - K^o\end{bmatrix}$. Note that

the gradient $\sigma^\top \frac{\partial G}{\partial (U,K)}(\cdot)$ can be computed efficiently using the backward mode of algorithmic differentiation (AD) [109]. The TR1 based Jacobian update formula then reads as

$$[D^+ \; C^+] = [D^o \; C^o] + \alpha \left(y - [D^o \; C^o]s\right) \left(\mu^\top - \sigma^\top [D^o \; C^o]\right), \tag{6.13}$$

where the scalar α can be defined differently for different variants of the update scheme. Aside from the case where the function $G(\cdot)$ is affine, the two secant conditions in Eq. (6.11) and (6.12) are not consistent with each other and they can therefore not both be satisfied by the updated matrix $\begin{bmatrix} D^+ & C^+ \end{bmatrix}$. In the adjoint variant of the TR1 update, the value $\alpha_{\mathrm{A}} = 1/(\sigma^\top y - \sigma^\top [D^o \; C^o]s)$ is defined such that the adjoint secant condition (6.12) is satisfied exactly and the forward condition holds up to some accuracy. Similarly, the forward variant is based on $\alpha_{\mathrm{F}} = 1/(\mu^\top s - \sigma^\top [D^o \; C^o]s)$ and instead satisfies the direct secant condition (6.11) exactly.

Real-Time Iteration Scheme for NMPC

In order to use the TR1 Jacobian update formula (6.13) within a lifted Newton-type optimization algorithm, one needs to be able to efficiently form the condensed QP in (6.9). For this purpose, the work in [122] described how to directly update the condensed matrix $E = -C^{-1}D$ in Eq. (6.8) from one iteration to the next. Let us write the rank-one update formula from Eq. (6.13) as follows

$$D^+ = D^o + \alpha \, u v_{\mathrm{D}}^\top \quad \text{and} \quad C^+ = C^o + \alpha \, u v_{\mathrm{C}}^\top, \tag{6.14}$$

where $u = y - [D^o \; C^o]s$ and $[v_{\mathrm{D}}^\top \; v_{\mathrm{C}}^\top] = \mu^\top - \sigma^\top [D^o \; C^o]$. The Sherman-Morrison formula can be used to update the matrix inverse approximation $C^{o^{-1}} \approx \frac{\partial G}{\partial K}^{-1}$ as

$$C^{+^{-1}} = C^{o^{-1}} - \alpha\beta \, u_1 v_{\mathrm{C}}^\top C^{o^{-1}}, \tag{6.15}$$

where $u_1 = C^{o^{-1}} u$ and $\beta = \frac{1}{1 + \alpha v_{\mathrm{C}}^\top u_1}$. It can be shown that the rank-one update formula then reads as

$$E^+ = -C^{+^{-1}} D^+ = E^o + u_1 v_1^\top, \tag{6.16}$$

where $v_1^\top = \alpha\beta v_{\mathrm{C}}^\top (E^o + \alpha u_1 v_{\mathrm{D}}^\top) - \alpha v_{\mathrm{D}}^\top$.

This rank-one update for the matrix $E^+ = -C^{+^{-1}} D^+$ in Eq. (6.16) provides an efficient manner to directly compute the matrices in the condensed QP (6.9), without the need for a matrix factorization, matrix inversion and without any matrix-matrix multiplications. Instead, the proposed algorithm merely requires matrix-vector multiplications and outer products, resulting in an overall quadratic $\mathcal{O}\left(N^2 m^2\right)$ instead of cubic $\mathcal{O}\left(N^3 m^3\right)$ computational complexity, where $m = (n_x + n_u)$ denotes the number of state and control variables. The resulting implementation of the real-time iteration (RTI) scheme with TR1 based Jacobian updates for pseudospectral based nonlinear MPC is presented in Algorithm 6.

Algorithm 6 Pseudospectral Method with TR1 Jacobian Updates within a Real-Time Iteration Scheme for NMPC.

Input: U^o, K^o, λ^o, ω^o, C^o, D^o, $C^{o^{-1}}$ and E^o.

`Problem linearization`

1: Formulate the dense QP in (6.9) with $A_c = A_u + A_k E^o$ and condensed Hessian $H_c = E^{o^\top} H E^o$.

`Computation of step direction`

2: Obtain current state estimate $\hat{x}_0$. ▷ from system

3: Evaluate the vectors a_c, h_c and solve the QP (6.9): $U^+ \leftarrow U^o + \Delta U^\star$ and $\lambda^+ \leftarrow \lambda^\star$.

4: Apply new control input value. ▷ to system

`TR1 Jacobian update`

5: $K^+ \leftarrow K^o + \Delta\tilde{K} + E^o \Delta U^\star$,

6: $\omega^+ \leftarrow \omega^o - C^{-\top}\left(h_k + \frac{\partial G}{\partial K}^\top \bar{\omega} + A_k^\top \bar{\lambda}^+\right)$,

7: $D^+ \leftarrow D^o + \alpha u v_{\mathrm{D}}^\top$ and $C^+ \leftarrow C^o + \alpha u v_{\mathrm{C}}^\top$,

8: $C^{+^{-1}} \leftarrow C^{o^{-1}} - \alpha\beta\, u_1 v_{\mathrm{C}}^\top C^{o^{-1}}$ and $E^+ \leftarrow E^o + u_1 v_1^\top$.

Output: U^+, K^+, λ^+, ω^+, C^+, D^+, $C^{+^{-1}}$ and E^+.

Remark 6.1.1. The new input value can be applied to the controlled system in step 4 of Alg. 6. A pseudospectral method provides a continuous time control profile that is represented by the polynomial $p_u(c) = \sum_{i=1}^{N} \ell_i(c) u_i$. This continuous time trajectory can more or less accurately be applied to the system, depending on the particular actuation in the control application and its sampling frequency. For simplicity, let us further assume a piecewise constant actuation, where we use the value $p_u(\frac{T_s}{T})$ of the collocation polynomial in which T_s denotes the MPC sampling time and T the control horizon length.

Quasi-Newton-type Update Scheme for the condensed Hessian

The TR1 Jacobian update scheme has quadratic computational complexity of $\mathcal{O}(N^2 m^2)$. Constructing the condensed Hessian $H_c = E^\top H E$ and computing a matrix factorization or inverse for the condensed Hessian however requires a cubic computational complexity of $\mathcal{O}(N^3 m^3)$ in general. Given the condensed Hessian and its inverse or matrix decomposition, the runtime computational cost for solving the dense QP (6.9) can be made of quadratic complexity $\mathcal{O}(N^2 m^2)$ instead, e.g., using a dense variant of the active-set method from [205]. Let us focus on how to avoid the operations with cubic complexity in case of a constant Hessian approximation or when using a quasi-Newton type update scheme.

Constant Hessian Approximation: Gauss-Newton

Note that the Gauss-Newton type Hessian approximation in Eq. (6.7a) corresponds to a constant matrix H, in case of a quadratic objective (6.6a) in the original NLP formulation. This is rather common in practical applications of MPC when tracking a reference for a linear output function of the state and control variables. Let us look at the condensed Hessian H_c, given the constant matrix H and a rank-one Jacobian update as in (6.16).

Lemma 6.1.2 (SR2)**.** *Given a rank-one update to the condensed Jacobian $E^+ = E^o + u_1 v_1^\top$, the condensed Hessian matrix $H_c = E^\top H E$ can be computed using the following symmetric rank-two update:*

$$H_c^+ = H_c^o + \tilde{u}_1 v_1^\top + v_1 \left(\tilde{u}_1^\top + \beta_1 \, v_1^\top \right). \tag{6.17}$$

Proof. This follows directly from the expression for the updated condensed Hessian matrix

$$\begin{aligned} H_c^+ &= {E^+}^\top H E^+ = \left(E^o + u_1 v_1^\top \right)^\top H \left(E^o + u_1 v_1^\top \right) \\ &= H_c^o + {E^o}^\top H u_1 v_1^\top + v_1 u_1^\top H E^o + v_1 u_1^\top H u_1 v_1^\top \\ &= H_c^o + \tilde{u}_1 v_1^\top + v_1 \left(\tilde{u}_1^\top + \beta_1 \, v_1^\top \right), \end{aligned} \tag{6.18}$$

where $\beta_1 := u_1^\top H u_1$ and $\tilde{u}_1 := {E^o}^\top H u_1$, such that the symmetric update is readily identified to be of rank 2. □ □

Note that the symmetric rank-two (SR2) update (6.17) can alternatively be represented as follows[101]:

$$H_c^+ = H_c^o + \left(\frac{1}{\tilde{\beta}_1} \tilde{u}_1 + \tilde{\beta}_1 v_1 \right) \left(\frac{1}{\tilde{\beta}_1} \tilde{u}_1 + \tilde{\beta}_1 v_1 \right)^\top - \frac{1}{\beta_1} \tilde{u}_1 \tilde{u}_1^\top, \tag{6.19}$$

where $\tilde{\beta}_1 := \sqrt{\beta_1}$ given that $\beta_1 = u_1^\top H u_1 > 0$. This means that the condensed Hessian matrix can be updated, from one iteration to the next, using the SR2 update or using two consecutive symmetric rank-one updates as in Eq. (6.19). Similarly, the Cholesky factorization, or the matrix inverse using the Sherman-Morrison-Woodbury formula, can be updated directly for the condensed Hessian. The resulting algorithm implementation, with overall quadratic computational complexity based on the TR1 and SR2 update formulas, for pseudospectral based nonlinear MPC is presented in Algorithm 7.

Quasi-Newton Type Hessian Approximation

We can construct a similar update formula for the condensed Hessian in case that a quasi-Newton type method is used instead of a constant Hessian approximation. For simplicity, let us consider the symmetric rank-one (SR1) update formula [70] to approximate the Hessian of the Lagrangian. This results in the STR1 update procedure as described in [82].

Lemma 6.1.3 (SR3). *Given a rank-one update to the condensed Jacobian $E^+ = E^o + u_1 v_1^\top$ and a symmetric rank-one Hessian update $H^+ = H^o + \alpha_2\, u_2 u_2^\top$, the condensed Hessian matrix $H_c = E^\top H E$ can be computed using the symmetric rank-three update:*

$$\begin{aligned} H_c^+ = H_c^o + \alpha_2\, \tilde{u}_2 \tilde{u}_2^\top + v_1 \left(\tilde{u}_1^\top + \beta_3\, \tilde{u}_2^\top + \beta_4\, v_1^\top \right) \\ + \left(\tilde{u}_1 + \beta_3\, \tilde{u}_2 + \beta_4\, v_1 \right) v_1^\top . \end{aligned} \tag{6.20}$$

Proof. It follows from the expression for the updated condensed Hessian matrix

$$\begin{aligned} H_c^+ &= {E^+}^\top H^+ E^+ \\ &= \left(E^o + u_1 v_1^\top \right)^\top \left(H^o + \alpha_2\, u_2 u_2^\top \right) \left(E^o + u_1 v_1^\top \right) \\ &= H_c^o + \tilde{u}_1 v_1^\top + v_1 \tilde{u}_1^\top + \beta_1\, v_1 v_1^\top + \alpha_2\, \tilde{u}_2 \tilde{u}_2^\top \\ &\quad + \alpha_2\, \beta_2\, \tilde{u}_2 v_1^\top + \alpha_2\, \beta_2\, v_1 \tilde{u}_2^\top + \alpha_2\, \beta_2^2\, v_1 v_1^\top , \end{aligned} \tag{6.21}$$

where $\beta_1 := u_1^\top H u_1$, $\beta_2 := u_2^\top u_1$, $\tilde{u}_1 := {E^o}^\top H u_1$ and $\tilde{u}_2 := {E^o}^\top u_2$. This can be further simplified to

$$\begin{aligned} H_c^+ = H_c^o + \alpha_2\, \tilde{u}_2 \tilde{u}_2^\top + v_1 \left(\tilde{u}_1^\top + \beta_3\, \tilde{u}_2^\top + \beta_4\, v_1^\top \right) \\ + \left(\tilde{u}_1 + \beta_3\, \tilde{u}_2 + \beta_4\, v_1 \right) v_1^\top , \end{aligned} \tag{6.22}$$

where $\beta_3 := \alpha_2\, \beta_2$ and $\beta_4 := \frac{\beta_1 + \alpha_2\, \beta_2^2}{2}$, such that the symmetric update is readily identified to be of rank 3. □ □

In a similar manner as for the symmetric rank-two update from Lemma 6.1.2, an alternative representation of the symmetric rank-three (SR3) formula (6.20) can be constructed as a sequence of three consecutive symmetric rank-one updates. In addition, the Cholesky factorization, or the matrix inverse using the Sherman-Morrison-Woodbury formula, can be computed directly for the condensed Hessian based on this update scheme. Algorithm 7 describes an implementation of pseudospectral based NMPC, using the TR1 and SR3 update formulas, respectively, for the condensed Jacobian and Hessian approximations.

6.2 NMPC Case Study: Chain of Masses

We consider the chain mass problem as a benchmark example for nonlinear MPC, which allows one to intuitively change the number of masses and therefore the state dimension in the problem. For reasons of brevity, we do not repeat the complete optimal control problem formulation here but we instead refer the reader to [208, 253]. The control task is to return a chain of n_m masses connected with springs to its steady state, starting from a perturbed initial configuration. The mass at one end is fixed, while the control input $u \in \mathbb{R}^3$ to the system is the direct force applied to the mass at the other end of the chain. This dynamic

Algorithm 7 Pseudospectral Method with TR1 Jacobian and SR2/SR3 condensed Hessian Updates for NMPC.

Input: U^o, K^o, λ^o, ω^o, C^o, D^o, $C^{o^{-1}}$, E^o, H^o and H_c^o.

`Problem linearization`

1: Formulate the dense QP in (6.9) with A_c and H_c^o.

2: `Computation of step direction`: line 2-4 in Alg. 6

3: `TR1 Jacobian update`: line 5-9 in Alg. 6

Option 1: `SR2 condensed Hessian update`

4: $H^+ \leftarrow H^o$.

5: $H_c^+ \leftarrow H_c^o + \tilde{u}_1 v_1^\top + v_1 \left(\tilde{u}_1^\top + \beta_1 \, v_1^\top\right)$.

Option 2: `SR3 condensed Hessian update`

6: $H^+ \leftarrow H^o + \alpha_2 \, u_2 u_2^\top$

7: $H_c^+ \leftarrow H_c^o + \alpha_2 \, \tilde{u}_2 \tilde{u}_2^\top + v_1 \left(\tilde{u}_1^\top + \beta_3 \, \tilde{u}_2^\top + \beta_4 \, v_1^\top\right)$
$+ \left(\tilde{u}_1 + \beta_3 \, \tilde{u}_2 + \beta_4 \, v_1\right) v_1^\top$.

Output: U^+, K^+, λ^+, ω^+, C^+, D^+, $C^{+^{-1}}$, E^+, H^+, H_c^+.

system can be described by a state vector $x \in \mathbb{R}^{6(n_m-1)}$, which is governed by the set of nonlinear differential equations in [208, 253].

Our aim is to validate the computational performance for Algorithm 6 and 7, using a lifted Newton-type optimization method with TR1 based Jacobian and corresponding condensed Hessian updates, in comparison with the standard RTI scheme based on exact Jacobian evaluations, similar to the pseudospectral based optimal control setup in [202]. The preliminary software implementation of the presented algorithms consists of C code for the TR1 and SR2 update formulas, in combination with a dense variant of the `PRESAS` active-set QP solver [205] and code generated evaluations of the system dynamics and the adjoint derivatives using `CasADi` [13]. In addition, we include a comparison with the direct collocation based RTI scheme with block-TR1 Jacobian updates as presented in [122].

Pseudospectral versus Direct Collocation Methods

The condensing procedure in a classical lifted Newton optimization algorithm for pseudospectral based optimal control requires a factorization of the exact Jacobian matrix at each iteration, resulting in a computational complexity of $\mathcal{O}\left(N^3 m^3\right)$. The proposed TR1 Jacobian update scheme with numerical condensing in Algorithm 6 avoids all cubic operations for the Jacobian approximation and Algorithm 7 additionally avoids such costly operations for the condensed Hessian, resulting in an overall computational complexity of $\mathcal{O}\left(N^2 m^2\right)$. The block-TR1 Jacobian update formula itself has a computational complexity of $\mathcal{O}\left(N_s m^2\right)$ for direct collocation [122], where N_s denotes the number of collocation intervals. However, the cost of solving the block-structured QP subproblems is typically $\mathcal{O}\left(N_s m^3\right)$. For exam-

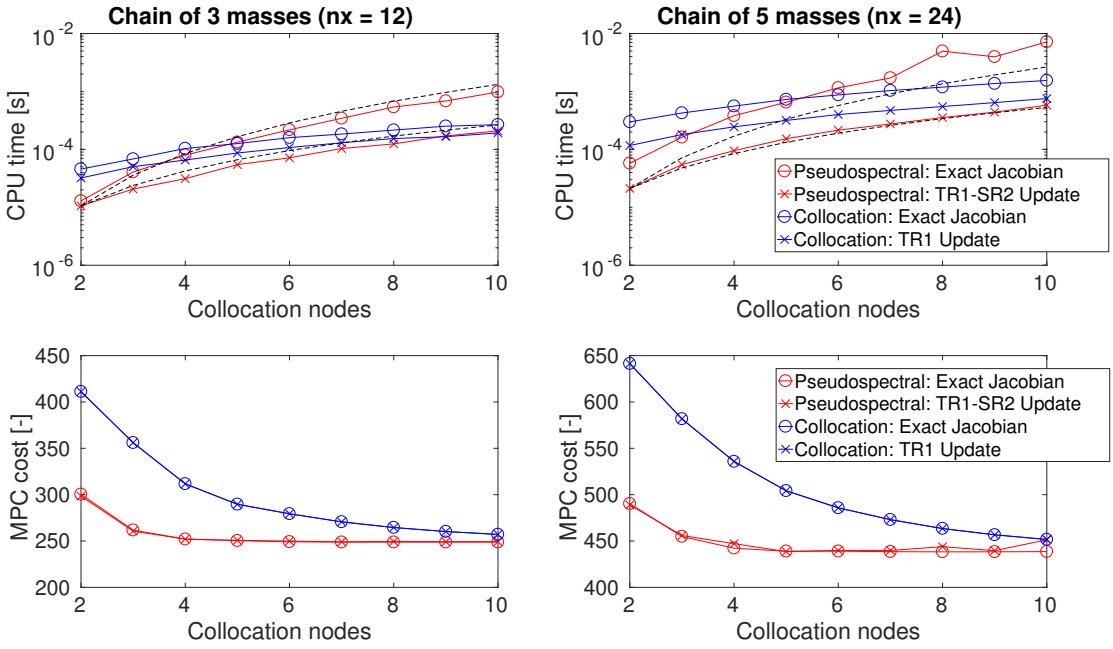

Figure 6.1: Average computation time per RTI step and overall closed-loop cost of NMPC based on direct collocation (with $N = 2$) versus a pseudospectral method, respectively, for a varying number N_s of shooting intervals or varying number N of collocation nodes.

ple, the sparsity exploiting `PRESAS` active-set QP solver [205] enjoys a setup computational complexity of $\mathcal{O}(N_s m^3)$ and a per iteration complexity of $\mathcal{O}(N_s m^2)$.

On the other hand, a pseudospectral method converges exponentially to a smooth continuous time optimal control solution [209] for an increasing degree N of the collocation polynomial. Alternatively, a piecewise constant control parametrization is typically used in combination with direct multiple shooting [48] or direct collocation [46]. Figure 6.1 shows the resulting trade-off between closed-loop control performance and computational cost, using an increasing number of collocation intervals N_s (direct collocation) or an increasing polynomial degree N (pseudospectral), for NMPC on the chain with $n_m = 3$ or 5 masses. The results for direct collocation are based on a Gauss-Legendre (GL) method with $N = 2$ nodes for each interval. Figure 6.1 shows the performance for both the exact Jacobian and the TR1 based Newton-type optimization algorithms.

Note that an alternative NMPC implementation could be based on a N-degree polynomial, for both the state and control parametrization, over each of the N_s collocation intervals in order to combine the advantages from both approaches for optimal control, such as in a spectral patching [85] or in a pseudospectral knotting method [216].

Table 6.1 shows the average computation times of the closed-loop NMPC simulation results using $N = 8$ Gauss collocation nodes for the chain of $n_m = 3$, 5 and 7 masses. The table shows the detailed timing results for pseudospectral based NMPC, using either the

Table 6.1: Average timing results (in ms) of pseudospectral based NMPC for the chain of masses using $N = 8$ Gauss collocation nodes.

		Gauss-Newton with TR1	
$n_m = 3, n_x = 12$	Exact	Alg. 6 (TR1)	Alg. 7 (TR1-SR2)
Linearization	0.474	0.093	0.084
Dense QP solution	0.020	0.021	0.016
Total RTI step	**0.539**	**0.161**	**0.124**

		Gauss-Newton with TR1	
$n_m = 5, n_x = 24$	Exact	Alg. 6 (TR1)	Alg. 7 (TR1-SR2)
Linearization	4.856	0.295	0.296
Dense QP solution	0.023	0.042	0.019
Total RTI step	**4.961**	**0.419**	**0.355**

		Gauss-Newton with TR1	
$n_m = 7, n_x = 36$	Exact	Alg. 6 (TR1)	Alg. 7 (TR1-SR2)
Linearization	15.403	0.628	0.609
Dense QP solution	0.024	0.024	0.018
Total RTI step	**15.560**	**0.782**	**0.682**

exact Jacobian or the TR1 based Jacobian update scheme. It can be observed that the computation time for the problem linearization and condensing procedure can be reduced significantly based on the TR1 method, resulting in a speedup of about factor 4, 10 and 20, respectively, for the chain of $n_m = 3$, 5 and 7 masses. On the other hand, the closed-loop NMPC performance is indistinguishable for the exact Jacobian and the TR1 based RTI scheme as shown earlier in Figure 6.1. Note that the additional speedup of using Algorithm 7 instead of 6 is small for this particular case study, given the small number of control inputs $n_u = 3$ and therefore the relatively small dimension of the dense QP in (6.9), compared to the amount of state variables $n_x = 6(n_m - 1)$.

6.3 A Structure Exploiting Branch-and-Bound Algorithm for Mixed-Integer Model Predictive Control

Optimization based control and estimation techniques, such as model predictive control (MPC) and moving horizon estimation (MHE), allow a model-based design framework in which the system dynamics and constraints can directly be taken into account [163]. This framework can be further extended to hybrid systems [33], providing a powerful technique to model a large range of problems, e.g., including dynamical systems with mode switchings or quantized control, problems with logic rules or no-go zone constraints. However, the resulting optimization problems are highly non-convex because they contain variables that only take integer values. When using a quadratic objective in combination with linear system dynamics and linear inequality constraints, the resulting optimal control problem (OCP) can be formulated as a mixed-integer quadratic program (MIQP).

We aim to solve MIQP problems of the following form:

$$\begin{aligned}
\min_{X,U} \quad & \frac{1}{2}\sum_{i=0}^{N-1} x_i^\top Q_i x_i + u_i^\top R_i u_i + x_N^\top P x_N && && \text{(6.23a)}\\
\text{s.t.} \quad & x_0 - \hat{x}_0 \;=\; 0, && && \text{(6.23b)}\\
& A_i x_i + B_i u_i + a_i \;=\; x_{i+1}, && i \in \{0,\dots,N-1\}, && \text{(6.23c)}\\
& l_i^c \le C_i x_i + D_i u_i \le u_i^c, && i \in \{0,\dots,N-1\}, && \text{(6.23d)}\\
& F_i u_i \in \{0,1\}, && i \in \{0,\dots,N-1\}, && \text{(6.23e)}\\
& l_N^c \le C_N x_N \le u_N^c, && && \text{(6.23f)}
\end{aligned}$$

where the optimization variables are the state $X = [x_0^\top, \dots, x_N^\top]^\top$ and control trajectory $U = [u_0^\top, \dots, u_{N-1}^\top]^\top$. The set of constraints (6.23e) are binary equality constraints, since the left-hand side needs to be equal to either 0 or 1. For simplicity of notation, we further consider only binary control variables instead of more general integer constraints for an affine function of both state and control variables. MPC for several classes of hybrid systems can be straightforwardly formulated as in (6.23). Notable examples are mixed logical systems [33], where auxiliary continuous and discrete variables can be added to the input vector. Moreover, in combination with the binary constraints (6.23e), the affine inequalities (6.23d) can model various complicated but practical restrictions on the feasible region, such as no-go zones and disjoint polyhedral constraints for states and inputs.

A hybrid MPC controller aims to solve the MIQP (6.23) at every sampling time instant. This is a difficult task, given that mixed-integer programming is $\mathcal{NP}$-hard in general, and several methods for solving such a sequence of MIQPs have been explored in the literature. These approaches can be divided into heuristic techniques, which seek to efficiently find sub-optimal solutions to the problem, and optimization algorithms which attempt to solve the MIQPs to optimality. Examples of the former include rounding and pumping

schemes [4, 5], approximate optimization algorithms [76, 178], and approximate dynamic programming [235]. The downside of fast heuristic approaches is often the lack of guarantees for finding an optimal or even an integer-feasible solution. Heuristic rounding-based approaches to mixed-integer nonlinear OCPs can be found, e.g., in [142, 220].

As for solving these problems to optimality, most of the optimization algorithms for MIQPs are based on the classical branch-and-bound (B&B) technique [95]. For the purpose of mixed-integer MPC, the standard B&B strategy has been combined with various methods for solving the relaxed convex QPs. For example, a B&B algorithm for mixed-integer MPC (MI-MPC) has been proposed in combination with a dual active-set solver in [24], with an interior point algorithm in [96], dual projected gradient methods in [23, 178], a nonnegative least squares solver in [34], and the alternating direction method of multipliers (ADMM) in [236]. Branch-and-bound methods for solving mixed-integer nonlinear OCPs have also been studied, e.g., in [102].

Another important research topic focuses on general pre-processing and modeling techniques to reduce the size and strengthen the mixed-integer problem formulations [181]. These *presolve* techniques are vital to the good performance of current state-of-the-art mixed-integer solvers [7], such that these methods can often solve seemingly intractable problems in practice. Lastly, the branch-and-bound method itself has been extensively studied with several improvements in branching and variable selection techniques [6, 151], including recent developments in applying machine learning techniques in order to learn "better" branching rules [29]. Finally, the branch-and-bound strategy has been generalized further, e.g., using cutting planes to tighten the convex problem relaxations, resulting in *branch-and-cut* or *branch-and-price* variants of the algorithm [95, 181]. Unlike state-of-the-art mixed-integer solvers, e.g., `GUROBI` [116] and `MOSEK` [177], our aim is to propose a tailored algorithm and its solver implementation for fast embedded MI-MPC applications, i.e., running on microprocessors with considerably less computational resources and available memory. The optimization algorithm should be relatively simple to code with a moderate use of resources, while the software implementation is preferably compact and library independent.

In this section, we propose a branch-and-bound based MPC algorithm, which exploits the features of the structure-exploiting primal active-set solver called `PRESAS` [204]. The latter algorithm is tailored to efficiently solve QPs with a block-sparse optimal control structure. Our second contribution is to bring various mixed-integer programming techniques, such as bound strengthening, domain propagation, and advanced branching rules, to the context of MI-MPC. In particular, we present an algorithm that exploits the sequential nature of MPC, in order to warm-start the branch-and-bound search tree and to re-use information gathered at previous time steps. A similar type of approach was proposed recently by [34], but in this work we provide not only a warm-start procedure for the integer variables but we also show how to improve the branching strategy by warm starting and how to efficiently combine this with presolving techniques for MI-MPC. Finally, the computational performance of the proposed algorithm, for a preliminary implementation in MATLAB and C code, is illustrated and compared against current state-of-the-art solvers for multiple MPC case studies.

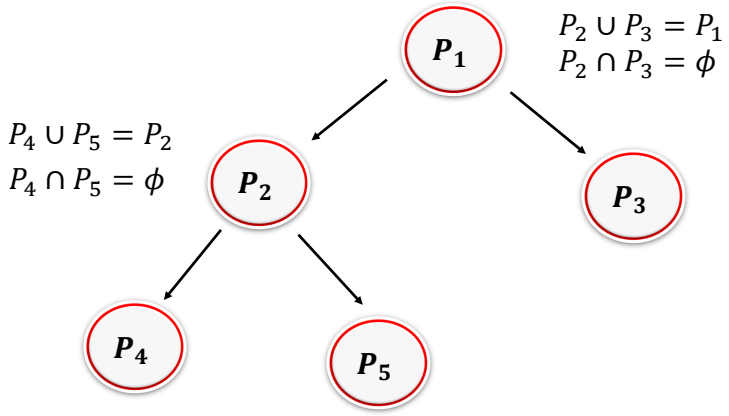

Figure 6.2: Illustration of the branch-and-bound method as a binary search tree. A selected node can be either *branched*, resulting in two partitions for each bound value in (6.23e), or *pruned* based on feasibility or the current upper bound.

Mixed-Integer Quadratic Programming

We first introduce some of the basic concepts in mixed-integer programming based on branch-and-bound methods, such as convex relaxations and branching strategies. A standard approach to solve the MIQP (6.23) is to create convex relaxations of this problem (by either dropping some constraints or by re-formulating the problem and providing an approximation scheme) and then solve the relaxations in order to approach the solution to the original MIQP. A straightforward idea is to obtain convex QP relaxations by dropping the binary equality constraints (6.23e) and instead enforcing the affine inequality constraints $0 \leq F_i u_i \leq 1$. Other convex relaxations for MIQPs have been studied in the literature such as moment or SDP relaxations that are often tighter than QP relaxations [157, 25], but they can be relatively expensive to solve for larger problems.

For the purpose of this paper, we will focus our attention on QP relaxations where we allow the binary variables to take on real values. The main reason for choosing this relaxation is that we utilize a tailored structure exploiting active-set solver, called `PRESAS` [204], proposed recently for efficiently solving the convex QP relaxations. The latter solver has been shown to be competitive with state-of-the-art QP solvers for embedded MPC, and it benefits strongly from warm-starting, which can be exploited when solving the sequence of QPs within the branch-and-bound strategy. Note that the relaxations need to be convex, i.e., the weight matrices Q_i, R_i and P need to be positive (semi-) definite in (6.23a) such that each solution to a QP relaxation is globally optimal.

Branch-and-Bound Algorithm

The main idea of the branch-and-bound (B&B) algorithm is to sequentially create partitions of the original problem and then attempt to solve those partitions. While solving each partition may still be challenging, it is fairly efficient to obtain local lower bounds on the optimal objective value, by solving relaxations of the mixed-integer program or by using duality. If we happen to obtain an integer-feasible solution while solving a relaxation, we can then use it to obtain a global upper bound for the solution to the original problem. This may help to avoid solving or branching certain partitions that were already created, i.e., these partitions or nodes can be *pruned*. The general algorithmic idea of partitioning is better illustrated as a binary search tree, see Figure 6.2.

A key step in this approach is how to create the partitions, i.e., which node to choose and which binary variable to select for branching. Since we solve a QP relaxation at every node of the tree, it is natural to branch on one of the binary variables with fractional values in the optimal solution of the QP relaxation. Therefore, if a variable, e.g., $u_{i,k} \in \{0, 1\}$ has a fractional value in a given QP relaxation, then we create two partitions where we respectively add the equality constraint $u_{i,k} = 0$ and $u_{i,k} = 1$. Another key step is how to choose the order in which the created subproblems are solved. These two steps have been extensively explored in the literature and various heuristics are implemented in state-of-the-art tools [6]. We provide next a brief description of strategies that we implemented in our B&B solver.

Tree Search: Node Selection Strategies

A common implementation of the branch-and-bound method is based on a *depth-first* node selection strategy, which can be readily implemented using a last-in-first-out (LIFO) buffer. The next node to be solved is selected as one of the children of the current node and this process is repeated until a node is pruned, i.e., the node is either infeasible, optimal or dominated by the upper bound, which is followed by a backtracking procedure. Instead, a *best-first* strategy selects the node with the lowest local lower bound so far. In what follows, we will employ a combination of the depth-first and best-first node selection approach. This idea is motivated by aiming to find an integer-feasible solution quickly at the start of the branch-and-bound procedure (depth-first) to allow for early pruning, followed by a more greedy search for better feasible solutions (best-first).

Reliability Branching for Variable Selection

The idea of *reliability branching* is to combine two powerful concepts for variable selection: strong branching and pseudo-costs [6]. Strong branching relies on temporarily branching, both up (to higher integer) and down (to lower integer), for every binary variable that has a fractional value in the solution of a QP relaxation in a given node, before committing to the variable that provides the highest value for a particular score function. The increase in objective values $\Delta^+_{i,k}$, $\Delta^-_{i,k}$ are computed when branching the binary variable $u_{i,k}$, respectively, up and down. Given these quantities, a simple scoring function score$(\cdot, \cdot)$ is computed for

each binary variable. For instance, based on the product [151]:

$$S_{i,k} = \text{score}(\Delta^-_{i,k}, \Delta^+_{i,k}) = \max(\Delta^+_{i,k}, \epsilon) \cdot \max(\Delta^-_{i,k}, \epsilon), \tag{6.24}$$

given a small positive value $\epsilon > 0$. This branching rule has been empirically shown to provide smaller search trees in practice [6]. The downside is that this procedure is relatively expensive since several QP relaxations are solved in order to select one variable to branch on.

The idea of pseudo-costs aims at approximating the increase of the objective function to decide which variable to branch on, without having to solve additional QP relaxations. This can be done by keeping statistic information for each binary variable, i.e., the *pseudo-costs* that represent the average increase in the objective value per unit change in that particular binary variable when branching. Every time that a given variable is chosen to be branched on, and the resulting relaxation is feasible, then we update each corresponding pseudo-cost with the observed increase in the objective, divided by the distance of the real to the binary value, in the form of a cumulative average. Therefore, each variable has two pseudo-costs, $\phi^-_{i,k}$ when the variable was branched "down" and $\phi^+_{i,k}$ when it was branched "up". Given the solution to a QP relaxation, one can then use the pseudo-costs to select the binary variable with the highest score value to be branched on next:

$$S_{i,k} = \text{score}(\bar{u}_{i,k}\, \phi^-_{i,k},\ (1 - \bar{u}_{i,k})\, \phi^+_{i,k}), \tag{6.25}$$

given a fractional value $\bar{u}_{i,k}$ in the QP relaxation.

This way, we select variables based on their past behavior throughout the branch-and-bound tree. However, at the beginning of the algorithm, the pseudo-costs are not yet initialized, which is when branching decisions typically impact the tree size the most. *Reliability branching* uses strong branching to initialize the pseudo-costs until a certain condition of reliability is satisfied, e.g., one switches to using pseudo-costs only once that particular variable has been branched on a specified number η_{rel} of times [6]. The resulting branching rule is summarized in Algorithm 8. Note that reliability branching coincides with pseudo-cost branching if $\eta_{rel} = 0$, with strong branching if $\eta_{rel} = \infty$, but typically a value $1 \leq \eta_{rel} \leq 4$ is chosen.

This rule can be further augmented by implementing a look ahead limit in the number of candidates, as well as a limit in the number of iterations for each QP relaxation in the strong branching step. Note that many other branching rules exist such as, e.g., "most infeasible" branching which selects the binary variable with fractional part that is closest to 0.5. Even though the latter rule is used quite often, e.g., in [34], it generally does not perform very well in practice [6]. Extensive empirical experiments with different branching strategies are beyond our scope. We next detail how can presolve techniques be used for general Mixed-Integer Optimal Control problems.

Algorithm 8 Reliability Branching Strategy

Input: η_{rel}, set C of candidate variables for branching.

for candidate variables $u_{i,k}$ in C **do**
  if #`branch`$(u_{i,k}) \leq \eta_{\text{rel}}$ **then**
    Strong branching on $u_{i,k}$ to compute score $S_{i,k}$.
    Update pseudo-costs $\phi^-_{i,k}$ and $\phi^+_{i,k}$.
  else
    $S_{i,k} = \text{score}(\bar{u}_{i,k}\,\phi^-_{i,k},\ (1-\bar{u}_{i,k})\,\phi^+_{i,k})$.
  end if
end for

Output: Select variable with highest score $S^* = \max_{i,k} S_{i,k}$.

Presolve Techniques for Mixed-Integer Optimal Control

As mentioned earlier, presolve techniques are often crucial in making convex relaxations tighter such that typically fewer nodes need to be explored, sometimes to such an extent that seemingly intractable problems become tractable. Next, we briefly describe some of these concepts with a focus on domain propagation for bound strengthening and its implementation for mixed-integer optimal control.

Domain Propagation for Condensed QP Subproblem

Several strengthening techniques are implemented as part of "presolve" routines in commercial solvers [7]. One particular technique that is suitable to mixed-integer optimal control is based on *domain propagation*, in which the goal is to strengthen bound values based on the inequality constraints (6.23d)-(6.23f) in the problem. However, the results of such a strategy are rather weak when directly applied to the block-sparse QP in (6.23), because the stage-wise coupling of the state variables (6.23c) needs to be taken into account. Therefore, we use instead the equivalent dense QP formulation in which the state variables are numerically eliminated, such that stronger bounds can be obtained for the control variables. Hence, we can use the block-structured sparsity to efficiently solve the QP relaxations, while we use the equivalent but dense format to effectively perform domain propagation.

Let us concatenate all state variables in a vector X and all control variables in the vector U, such that Eqs. (6.23b)-(6.23c) can be written more compactly as

$$\bar{A}X = \bar{B}U + b + E_0\hat{x}_0, \tag{6.26}$$

where we define the block-sparse matrices

$$\bar{A} = \begin{bmatrix} I & & & \\ -A_1 & I & & \\ & \ddots & \ddots & \\ & & -A_{N-1} & I \end{bmatrix}, \tag{6.27}$$

$$\bar{B} = \text{blkdiag}(B_0, \ldots, B_{N-1}) \ , \ E_0 = [A_0^\top, 0, \ldots, 0]^\top. \tag{6.28}$$

The matrix $\bar{A}$ is invertible such that we can write:

$$X = \bar{A}^{-1}\bar{B}U + \bar{A}^{-1}(b + E_0\hat{x}_0). \tag{6.29}$$

Now, we can substitute the latter expression for the state vector in OCP (6.23) to obtain the condensed form

$$\min_U \quad \frac{1}{2}U^\top H_c U + h_c^\top U \tag{6.30}$$

$$\text{s.t.} \quad \bar{l}^c \leq D_c U \leq \bar{u}^c \tag{6.31}$$

$$F_i u_i \in \{0, 1\}, \quad i \in \{0, \ldots, N-1\}, \tag{6.32}$$

where the condensed matrices and vectors read as

$$\begin{aligned} H_c &= (\bar{A}^{-1}\bar{B})^\top Q \bar{A}^{-1}\bar{B} + R, \ D_c = C\bar{A}^{-1}\bar{B} + D, \\ h_c &= (\bar{A}^{-1}\bar{b})^\top Q \bar{A}^{-1}\bar{B}, \\ \bar{l}^c &= l^c - C\bar{A}^{-1}\bar{b}, \quad \bar{u}^c = u^c - C\bar{A}^{-1}\bar{b}, \end{aligned} \tag{6.33}$$

where $\bar{b} := b + E_0\hat{x}_0$ is defined and given

$$Q = \text{blkdiag}(Q_1, \ldots, Q_{N-1}, P), \quad R = \text{blkdiag}(R_0, \ldots, R_{N-1}), \tag{6.34}$$

and $l^c = [l_1^{c\top}, \ldots, l_N^{c\top}]^\top$ and $u^c = [u_1^{c\top}, \ldots, u_N^{c\top}]^\top$.

Given the condensed problem formulation, which can be computed offline and which is parametric in the current state value $\hat{x}_0$, we can then apply the following bound strengthening procedure, which is explained next for a single affine constraint $l_b \leq \sum_i d_i u_i \leq u_b$ in (6.31). This constraint can be used to try and tighten bound values for all control variables u_i for which $d_i \neq 0$, where u_i denotes a single control variable in the vector U. Let $\bar{u}_i, \underline{u}_i$ be the current upper/lower bounds for u_i such that

$$d_i u_i \leq u_b - \sum_{j \neq i} d_j u_j \leq \underbrace{u_b - \sum_{j \neq i, d_j > 0} d_j \underline{u}_j - \sum_{j \neq i, d_j < 0} d_j \bar{u}_j}_{=: \bar{u}_{b,i}}, \tag{6.35}$$

in which we divide by d_i in order to obtain

$$u_i \leq \frac{\bar{u}_{b,i}}{d_i}, \text{ if } d_i > 0 \quad \text{or} \quad u_i \geq \frac{\bar{u}_{b,i}}{d_i}, \text{ if } d_i < 0. \tag{6.36}$$

Algorithm 9 Domain Propagation for Bound Strengthening

Input: Inequality constraints (6.31), variable bounds $\bar{u}_i, \underline{u}_i$.

```
while stopping criterion == False do
    for every row of D_c do
        for every u_i ∈ U, d_i ≠ 0 do
            Obtain bound values ū_{b,i}, l̄_{b,i} using Eq. (6.36).
            Update variable bounds using (6.37) or (6.38).
        end for
    end for
end while
```

Output: Updated bounds $\bar{u}_i, \underline{u}_i$ for all control variables.

This results, respectively, in the updated bound values

$$\bar{u}_i = \min(\bar{u}_i, \frac{\bar{u}_{b,i}}{d_i}), \quad \text{or} \quad \underline{u}_i = \max(\underline{u}_i, \frac{\bar{u}_{b,i}}{d_i}), \tag{6.37}$$

or, in case u_i is an integer or binary variable,

$$\bar{u}_i = \min(\bar{u}_i, \left\lfloor \frac{\bar{u}_{b,i}}{d_i} \right\rfloor), \quad \text{or} \quad \underline{u}_i = \max(\underline{u}_i, \left\lceil \frac{\bar{u}_{b,i}}{d_i} \right\rceil). \tag{6.38}$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are the floor and ceiling operations, respectively. Thus, this can result in strengthening of bound values for both continuous and integer/binary control variables. The procedure can be executed for each control variable and each inequality constraint in an iterative manner, see Algorithm 9, since bound strengthening for one variable can lead to strengthening for other variables [7]. The process is typically stopped when the bound values do not sufficiently change or a certain limit on the computation time is met.

Domain propagation can lead to considerable reductions in the amount of explored nodes, e.g., because variables are fixed, when $\bar{u}_i = \underline{u}_i$, or because of infeasibility detection, when $\bar{u}_i < \underline{u}_i$, without the need to solve any QP relaxations. In addition, the updated bound values for all control variables can be used to strengthen QP relaxations in the future. Lastly, we can use domain propagation in order to improve and generalize Hessian-based fixing strategies, such as the one proposed in [22]. Hessian-based fixing typically can only be applied to unconstrained problems, since it fixes the variables solely based on the objective. Here, we propose to use domain propagation to compute the feasibility impact of certain variable fixings. More specifically, a particular variable can be fixed based on optimality, if and only if this fixing does not induce feasibility-based fixings.

Probing Strategies and Cutting Planes

Probing [224] is a classical technique that can be incorporated in any branch-and-bound method to derive stronger inequalities or better bounds. It consists of tentatively trying

to fix some variables and to derive potential logical implications on other variables. We do not further describe probing strategies in detail, but we refer to [7] for an overview. The computational cost and performance of probing can be greatly improved by relying on some of the other techniques that were discussed earlier. For example, the pseudo-costs can be used in order to choose the bound value for each binary variable that is likely to result in a low objective value. In turn, the QP relaxations that are solved in the probing procedure can be used to update the pseudo-cost values. In addition, domain propagation and other variable fixing strategies can be used to reduce the amount of QP relaxations that need to be solved.

Other presolving techniques such as cut generation can be applied using the condensed problem, and can be fully transferred to the original OCP formulation. In the present paper, we refrain from using cut generation techniques as they produce inequalities that potentially couple variables across stages. Such coupling between stages is not desirable as we rely on a block-sparsity exploiting QP solver.

Resulting MIQP Algorithm for Optimal Control

Algorithm 10 describes the most important steps in our proposed B&B method for solving the MIQP in (6.23). It solves a block-structured QP relaxation using `PRESAS` [204] at every node and utilizes reliability branching (Algorithm 8) to decide the branching variables. As discussed earlier, the node selection strategy is based on a depth-first search followed by a best-first search as soon as an integer-feasible solution has been found. Note that the upper bound value UB provided to Alg. 10 can be based on an integer-feasible solution guess or it can be set to $+\infty$. Because of space limitations, the present paper will not further discuss all parameter choices in the algorithm such as, e.g., the reliability branching parameters, presolving frequency, memory usage, etc.

6.4 Mixed-Integer MPC Algorithm

In embedded applications of mixed-integer MPC, one needs to solve an MIQP (6.23) at each sampling instant under strict timing constraints. We can leverage the fact that we solve a sequence of similar problems (parametrized by the initial condition $\hat{x}_0$), in order to warm-start the B&B-algorithm. We refer to our proposed warm-starting procedure as *tree propagation*, because the main goal is to "propagate" the B&B tree forward by one time step. We describe this process in detail below. Then, we present the resulting mixed-integer MPC algorithm.

Warm Starting based on Tree Propagation

The warm-starting procedure aims to use knowledge of one MIQP, i.e., the search tree after solving the problem, in order to improve the B&B search for the next MIQP. Our idea

Algorithm 10 B&B Method for the MIQP-OCP in (6.23)

Input: Upper bound UB, tolerance ϵ.

LB=$-\infty$ and initialize $L = \{P_0\}$ with root node.
Select current node $P_c \leftarrow P_0$.
while $UB - LB > \epsilon$ **do**
    Apply domain propagation to P_c using Alg. 9.
    Solve resulting QP relaxation with `PRESAS`.
    if QP is feasible and $J(\bar{X}, \bar{U}) \leq$ UB **then**
        if QP solution is not integer-feasible **then**
            $LB \leftarrow \min_{P \in L} J(P)$.
            Select branching variable v using Alg. 8.
            Create subproblems P_u "up" and P_l "down".
            Append $\{P_l, P_u\}$ to L if $(1 - \bar{v})\phi_v^+ < \bar{v}\phi_v^-$
            or append $\{P_u, P_l\}$ to L, otherwise.
        else
            $UB \leftarrow J(\bar{X}, \bar{U})$ and $(X^*, U^*) \leftarrow (\bar{X}, \bar{U})$.
        end if
    end if
    Remove current node P_c from to-do list in L.
    Select next node based on depth-first (last node in list L) or based on best lower bound.
end while

Output: MIQP solution vector (X^*, U^*).

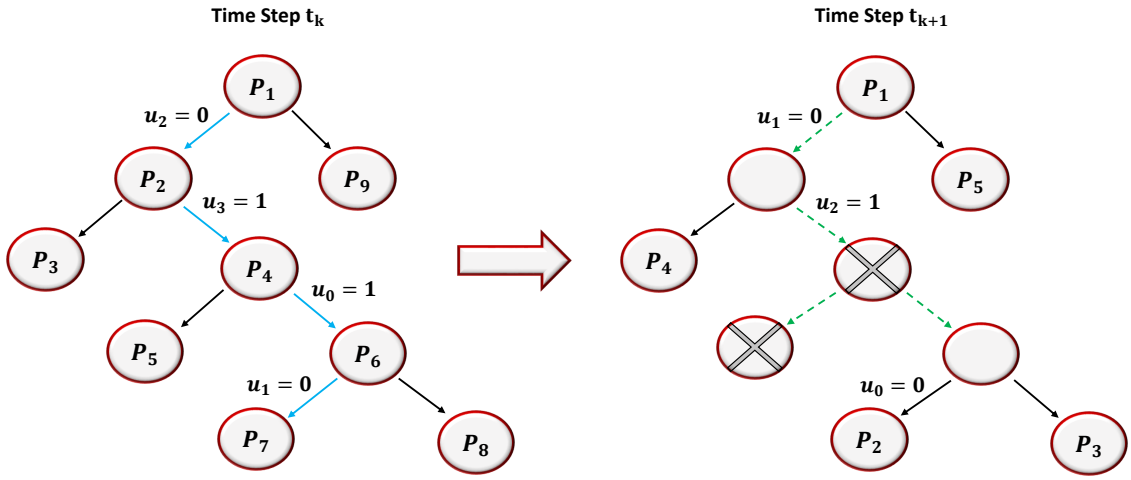

Figure 6.3: Illustration of the tree propagation technique from one time point to the next in the MI-MPC algorithm: index i denotes the order in which each node P_i is solved.

is to store the path from the root to the leaf node where the optimal solution to the MIQP was found, as well as the branching order of the variables. We can then perform a shifting of this path in order to obtain a "warm-started tree" to start our search to solve the MIQP at the next time step. We illustrate this procedure in Figure 6.3, where the optimal path at the current time step is denoted by the sequence of nodes $P_1 \rightarrow P_2 \rightarrow P_4 \rightarrow P_6 \rightarrow P_7$. Let us consider a corresponding sequence of variables $u_2 \rightarrow u_3 \rightarrow u_0 \rightarrow u_1$ that we branched on in order to create such optimal path. After shifting by one time step, all branched variables in the first control interval can be ignored, e.g., resulting in a shifted and shorter path of variables $u_1 \rightarrow u_2 \rightarrow u_0$.

At the subsequent time step, after obtaining the new state estimate, we execute all presolving techniques and we solve the QP relaxation corresponding to the root node. After removing from the warm-started tree the nodes that correspond to branched variables which are already integer feasible in the relaxed solution at the root node, we proceed by solving all the leaf nodes on the warm-started path. As we solve both children of a node on this path, we do not have to solve the parent node itself and therefore reduce computations by solving less QP relaxations. Hence, we go over the tree in the order depicted by the index of each node in Fig. 6.3. After the warm-started branch has been explored, we resume normal procedure of the B&B method. Algorithm 11 summarizes the proposed tree propagation technique.

Algorithm 11 Tree Propagation for Warm-Started B&B

Input: Optimal path P from root to leaf node.

1: Shift index of branched variables by 1 stage along path.
2: Solve root node of shifted path P, including presolve.
3: **for** (branched variables on stage -1 after shifting)
   || (variables are integer feasible in root node)
   || (variables without pseudo-costs) **do**
4: remove associated node from the path P.
5: **end for**
6: Shift the QP relaxation solution on every node of the path and store it as a warm start for the QP solver.
7: Re-order sequence of branched variables by scoring based on warm-started pseudo-cost information.
8: Initialize the B&B tree along the shifted path P, creating nodes along the path and their respective children.
9: Create the warm-started list L, excluding parent nodes.

Output: Warm-started tree for next MIQP, given by list L.

The sequential nature of the problem also allows to shift and re-use the pseudo-cost information from one MPC time step to the next. This idea has the potential of producing smaller search trees as the MPC progresses, without the need to perform strong branching at every MPC step. The propagation of pseudo-costs can be coupled with an update of

the reliability parameters to improve the overall performance. For example, the reliability number should be reduced for each variable from one time step to the next, in order to force strong branching for variables that have not been branched on in a sufficiently long time. In addition, nodes can be removed from the warm-started path in case they correspond to branched variables for which there is no pseudo-cost information or it is not sufficiently reliable, in an attempt to avoid bad branching decisions. Finally, these warm-started pseudo-costs can also be used to re-order the warm-started tree, in order to result in smaller search tree sizes.

The proposed tree propagation technique, with the additional re-use of pseudo-cost information, has been summarized in Algorithm 11. This procedure can improve the overall performance of the B&B method in multiple ways. First of all, the optimal path and pseudo-cost information is re-used to make better branching decisions for the mixed-integer program at the next time step, because the search trees are often similar for two subsequent problems. Also, the computational cost can be reduced by solving less QP relaxations to explore the warm-started tree. In addition, the shifted optimal path can be used in an attempt to efficiently obtain an integer-feasible solution, and therefore an important upper bound in the B&B algorithm, for the MPC problem at the next time step. Lastly, one can store the relaxed QP solutions on the optimal path, shift them by one time step and use them to warm-start the QP solver for nodes on the shifted optimal path.

MI-MPC Algorithm Implementation

Algorithm 12 summarizes the proposed MI-MPC algorithm. It solves a sequence of MIQPs where the branch-and-bound tree is warm-started at every time step, as well as the pseudo-cost and QP condensing information. As mentioned earlier, the B&B strategy and the additional presolve, warm-start and heuristic branching techniques have been implemented in MATLAB, based on a C code implementation of the `PRESAS` algorithm [204] to solve each QP relaxation. In Section 5.4, we illustrate the computational performance of the presented MI-MPC algorithm, including these presolving and warm-starting techniques, for two numerical case studies of mixed-integer MPC. A self-contained C code implementation is part of ongoing work, in order to illustrate the computational efficiency of the proposed algorithmic techniques.

Note that, in practice, the proposed warm-starting strategies often allow one to obtain an integer-feasible solution in a computationally efficient manner. However, even if the tree propagation immediately provides the globally optimal solution to the MIQP (6.23), a branch-and-bound algorithm still needs to perform relatively many iterations to prove optimality by pruning remaining nodes in the search tree. This motivates the use of a maximum number of B&B iterations in order to meet strict timing requirements of the embedded control application. Even if the algorithm does not terminate within this specified number of iterations, a feasible or even optimal solution may be available. This and other heuristic strategies for real-time implementation of MI-MPC are straightforward but outside the scope of this paper.

Algorithm 12 Warm-Started B&B Algorithm for MI-MPC

Input: Current state $\hat{x}_0$, list of nodes L and pseudo-costs.

`Solve MIQP`

1: Update condensing information, given current state $\hat{x}_0$.
2: Formulate MIQP (6.23) and solve it using Algorithm 10.
3: Apply new control input $u_0^\star$ to the system.

`Propagation Step`

4: Warm-start and shift pseudo-cost information.
5: Perform tree propagation to warm-start node list L for the next MI-MPC time step (see Algorithm 11).

6.5 Case Studies: Mixed-Inter MPC

We report two numerical case studies to illustrate the computational performance of our MIQP-based MPC algorithm: a hybrid MPC test example and a satellite orbit re-centering application with a no-go zone in the orbital path. Our branch-and-bound algorithm has been implemented in MATLAB in conjunction with the `PRESAS` active-set solver in C. To evaluate the performance, we compare our algorithm with the state-of-the-art `GUROBI` [116] and `MOSEK` [177] solvers for mixed-integer programming. It is important to emphasize that all advanced presolve and heuristic options have been activated for both software tools, resulting in fair computational comparisons.

Hybrid MPC: Benchmark Example

The first case study is a hybrid MPC problem from [33], with the default settings as in `bm99sim.m`, which is a part of the Hybrid Toolbox for MATLAB. This demo example has been used also more recently for numerical comparisons in [34]. The system is modeled using the HYSDEL toolbox [240] to obtain the mixed logical dynamical (MLD) system formulation. Figure 6.4 illustrates the average and worst-case CPU times taken by our algorithm, `GUROBI` and `MOSEK` for a range of control horizon lengths N.

Table 6.2 presents a detailed comparison for this test example, including additional timing results for the `MI-NNLS` solver that are taken directly from [34]. The latter computational results can serve only as a reference since they have been obtained on a different computer, with respect to the one used here with a 2.80 GHz Intel Xeon E3-1505M v5 processor and 32 GB of RAM. An important feature of our method is that its worst-case computation time is often rather close to the average performance in closed-loop MI-MPC simulations. This highlights the effectiveness of our tree propagation warm-starting procedure, such that consecutive branch-and-bound trees have approximately the same size. In addition, it can be observed from Table 6.2 that our proposed `BB-PRESAS` solver is either competitive with, or is a factor 2 or 3 times faster than `GUROBI`. The computational speedup is much larger

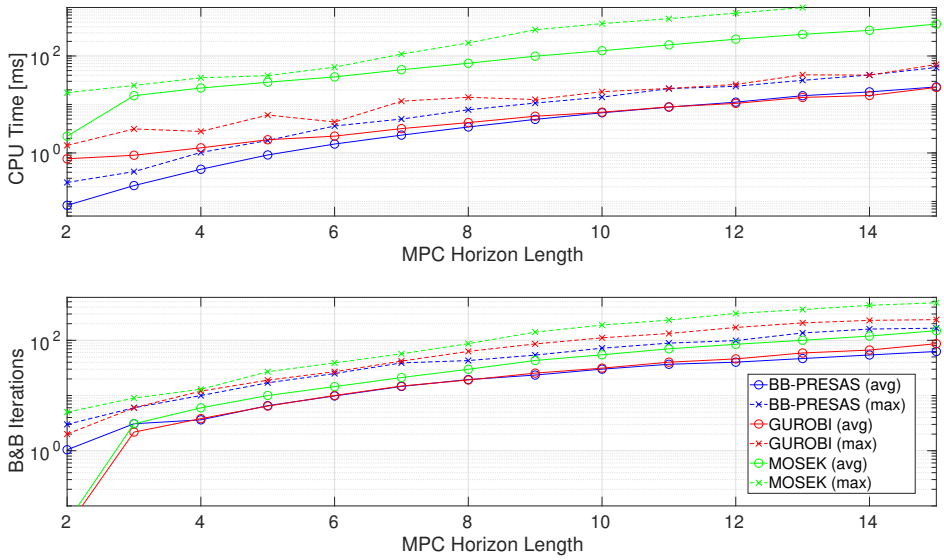

Figure 6.4: Computational results for closed-loop mixed-integer MPC of the `bm99` example: `BB-PRESAS` versus `GUROBI` and `MOSEK` solvers for varying control horizon length N.

when compared with other state-of-the-art tools such as `MOSEK`, our solver can be more than 10 times faster in this particular MI-MPC test example. It shall be noted that `GUROBI` is a heavily optimized and fairly large software, which is unlikely to be amenable for embedded microprocessors, due to its code size, memory requirements, and software library dependencies.

Satellite Station Keeping with No-Go Zones

The second case study is motivated by a real-world application, namely, orbit control of a satellite in a circular low earth orbit, 400km from earth surface. The satellite propulsion system is composed of two on/off thrusters, one on each of the in-track faces of the satellite, with gimbals rotating along the vertical axis and subject to angle constraints [250]. Thus, the propulsion system is controlled by two binary and two continuous control signals. The satellite dynamics are formulated by relative motion equations (HCW) with respect to the target position along the orbit, and the cone constraints of the thrust forces are approximated as simplexes [250]. Here, we consider a re-centering maneuver in which the satellite, previously drifting, is re-centered close to the target position along the orbit. Furthermore, the error coordinates from the target position are constrained in a station keeping window ($-300 \leq X \leq 300, -150 \leq Y \leq 150$).

Thus, our problem is simplified from [250], by considering only the orbital dynamics in the orbital plane, i.e., ignoring the out-of-orbital-plane and attitude dynamics, and as a consequence using a simpler propulsion system with only two thrusters.

Table 6.2: Timing results (ms) per sampling step of hybrid MPC test problem for different horizon lengths N. Computation times for `MI-NNLS` solver are taken directly from [34].

N	`BB-PRESAS` (mean/max)	`GUROBI` (mean/max)	`MOSEK` (mean/max)	`MI-NNLS` (mean/max)
2	0.1/0.2	0.7/1.4	2.1/4.0	2.0/2.6
3	0.2/0.3	1.0/2.3	15.1/24.7	2.5/4.8
4	0.4/0.9	1.7/4.6	21.7/35.5	3.1/6.9
5	0.9/1.7	2.5/4.9	28.7/39.3	3.9/13.0
6	1.5/3.5	3.2/7.5	36.8/58.8	5.1/18.3
7	2.3/4.9	4.0/6.9	51.8/109.3	6.4/30.2
8	3.5/7.6	5.1/10.0	70.4/185.8	8.1/43.4
9	5.1/10.3	6.6/12.5	98.7/347.1	11.1/69.8
10	6.8/14.3	8.4/16.1	126.7/465.3	14.4/103.2
11	8.8/22.1	9.8/17.2	168.2/587.8	20.6/179.1
12	11.3/23.7	11.6/20.5	219.2/765.0	26.9/263.4
13	15.0/31.6	14.3/29.5	276.3/996.0	35.5/384.9
14	17.8/35.1	16.4/44.6	334.1/1241.9	46.3/562.4
15	21.0/41.6	21.9/71.6	450.8/1606.8	61.7/766.9

To better highlight the potential of the MI-MPC method, we add an exclusion zone in the station keeping window, i.e., an area that must be avoided, which makes the allowed region of positions to be non-convex. This additional constraint is modeled using standard integer programming techniques (see, e.g., [181]), resulting in three additional binary variables for each prediction step of the mixed-integer OCP to implement the logical exclusion zone constraints.

In Figure 6.5, we show the trajectory of the satellite in relative coordinates, where the origin is the desired satellite position along the orbit, for the simulation of the satellite controlled by the mixed-integer MPC. The depicted area in the figure corresponds to the station keeping window, in which the satellite should be kept, and the shaded area is the exclusion zone that must be avoided, at least pointwise in time. The computational timing results for this particular closed-loop MPC simulation can be found in Figure 6.6. One can observe that our proposed algorithm has a very competitive runtime at every MPC time step, when compared to the commercial `GUROBI` solver. Most importantly, the `BB-PRESAS` algorithm appears to perform at least as good for this particular case study in terms of worst-case computation times.

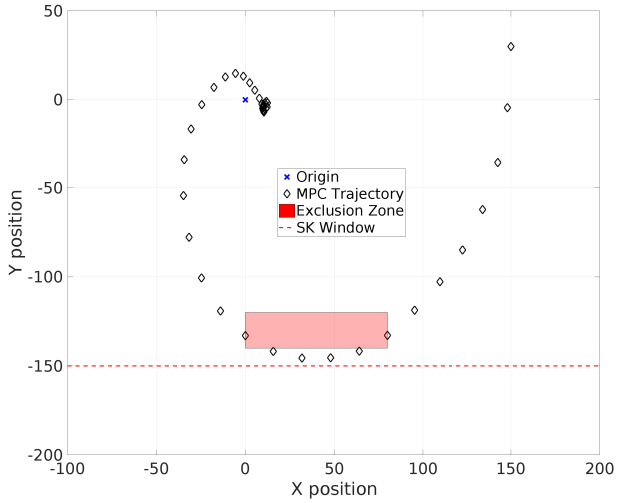

Figure 6.5: MPC state evolution for satellite station keeping around the origin: rectangular no-go zone is depicted in red.

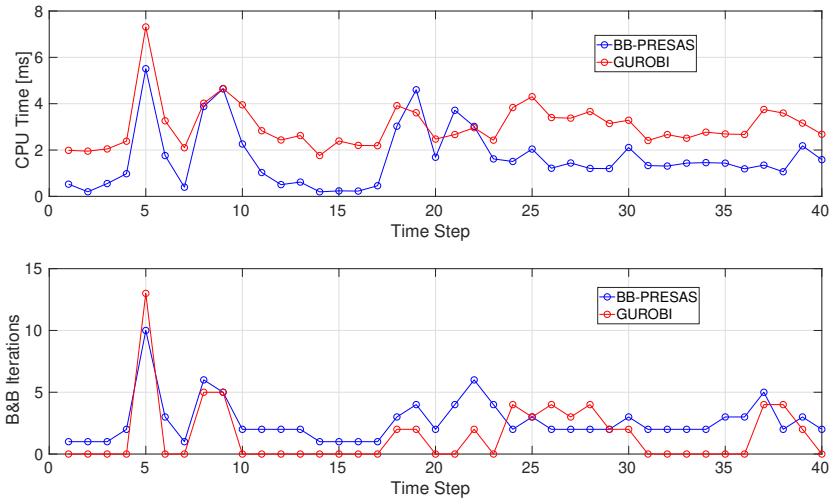

Figure 6.6: Closed-loop results of mixed-integer MPC for satellite station keeping: comparison between `BB-PRESAS` versus `GUROBI` solver.

Chapter 7

Conclusion and Outlook

This thesis, in Chapters 2 and 3 we constructed a dynamic watermarking approach for detecting malicious sensor attacks for general LTI systems, and the two main contributions were: to extend dynamic watermarking to general LTI systems under a specific attack model that is more general than replay attacks, and to show that modeling is important for designing watermarking techniques by demonstrating how persistent disturbances can negatively affect the accuracy of dynamic watermarking. Our approach to resolve this issue was to incorporate a model of the persistent disturbance via the internal model principle. We extended our methodology to detect sensor and communication attacks on networked LTI systems. Unlike the non-networked case, watermarking in the networked case requires a K so that $A+BK$ is controllable with respect to each B_i. This ensures the private watermarking signal of each subcontroller is seen in the output of all subcontrollers. We provided two algorithms to compute such a controller. The efficacy of our watermarking approach was demonstrated by a simulation of a three car platoon. One possible direction for future work is to explore the performance of our watermarking test under specific types of attacks, such as replay attacks or network disturbances. Future work includes generalizing the attack models that can be detected by our approach, for example network disturbances that change the communication structure itself, rather than it;s measurements. Another line of related work is the work done by [197, 196], which extends our ideas of Dynamic Watermarking to linear time-varying (LTV) systems, while maintaining all the statistical properties and guarantees. An additional direction for future work is to study the problem of robust controller design in the regime of when an attack is detected. Lastly, we proved statistical consistency of the set-membership estimator for identification of switched linear systems, and we demonstrated the consistency properties through two examples: one consisting of a comparison to OLS (which is inconsistent) and the other the construction of an algorithm that identifies the stable mode under additional assumptions.

In Chapter 4 we proposed a hypothesis testing framework in order to decide whether agents are behaving competitively or not. In our setting, a regulator formulates an inverse variational problem in order to estimate the unknown private information vectors as well as estimate the residuals of the approximate equilibrium that arises from the agents' com-

petition. Our setting is flexible as the regulator only require access to prices and shock values. The assumption of common knowledge on the shock can be relaxed, leading to a new set of challenges in the residual estimation. A future direction of work is to derive precise theory about consistency of our estimates in the context of inverse optimization. We demonstrated our method in a simple two-player game with a polyhedral feasible action space. We stress that our setting is more general and allows for any number of players with arbitrarily conic-representable sets, as long as they satisfy some regularity condition. Another exciting direction of future research is to employ our estimation method and hypothesis testing framework in the context of problem studied in [62, 153], where groups of agents employ machine learning-based methods, and those algorithms "learn" to collude instead of competing. This problem is more challenging but can be explored in the light of inverse variational problems and our estimation formulation. we also studied a dynamical system with several non-cooperative strategic agents. We proposed a mechanism where the agents interact via a platform and characterized the equilibrium strategies. We provided an HVAC control test case to highlight the need of designing mechanisms that have low-communication requirements in an MPC setting. Our goal for future research is to rigorously find a learning process that allow agents to converge their messages to the equilibrium behavior and to explore conditions such that we can strengthen Theorem 1 toward strong implementability and uniqueness of the resulting equilibrium.

In Chapter 5 and 6, we proposed a block-wise sparsity preserving two-sided rank-one Jacobian update (TR1 update) for an adjoint-based inexact SQP method to efficiently solve the nonlinear optimal control problems arising in NMPC. We proved local convergence for the block-structured quasi-Newton type Jacobian matrix updates. In case of a Gauss-Newton based SQP implementation, we additionally showed that the asymptotic rate of contraction remains the same. We also presented how this approach can be implemented efficiently in a tailored lifted collocation framework, in order to avoid matrix factorizations and matrix-matrix multiplications. Finally, we illustrated the local convergence properties as well as the computational complexity results numerically for two NMPC case studies. The effect of the presented contraction properties on the convergence and closed-loop stability of the block-TR1 based real-time iterations is an important topic that is part of ongoing research. In addition, we proposed a lifted Newton-type optimization method for pseudospectral based nonlinear model predictive control, using a rank-one Jacobian update formula in combination with numerical condensing and expansion of the collocation variables. We showed how the condensed Hessian can be updated directly, using either a symmetric rank-two or a rank-three update, in case that a quasi-Newton type method is used to additionally approximate the Hessian of the Lagrangian. The proposed pseudospectral optimization algorithm has a quadratic computational complexity of $\mathcal{O}\left(N^2m^2\right)$, compared to the typical complexity of $\mathcal{O}\left(Nm^3\right)$ for sparsity exploiting optimal control algorithms based on direct collocation. A preliminary C code implementation has shown to allow considerable numerical speedups for the NMPC case study of the nonlinear chain of masses. Lastly, we proposed a branch-and-bound algorithm for mixed-integer MPC that exploits the optimal control problem structure to strengthen variable bounds, re-use pseudo-costs and warm-start the search tree at every

MPC time step. More specifically, tailored domain propagation and tree propagation strategies have been presented. We showed preliminary results that illustrate the computational performance of our algorithm for two different MI-MPC case studies. A compact, efficient, but self-contained C code implementation of the proposed algorithm is under development to enable real-time embedded applications of hybrid MPC.

The future is promising for research in Cyber-Physical Systems. The new technologies and specialized hardware allow engineers and researchers to design efficient and reliable algorithms and methods to provide secure and resilient systems that operate in high performance. Our thesis is but a step in this exciting direction. Several challenges still lay ahead, in particular the human-robot interactions (for example of human drivers and self-driving vehicles) where the algorithms will need to make inference on the behavior of not only other CPS bu also of humans. Statistical tests, such as Dynamic Watermarking, provide as a quantitative method that contains finite-time guarantees in detection and in inference. Our analysis and methodology based on Concentration Inequalities and Stein's Method can be potentially extended to inference problems where instead of detecting attacks on the CPS we try to detect some specific behavior of other (possibly human) agents. On the performance side, our Mixed-Integer MPC is able to handle both continuous and discrete state and control variables and efficiently exploits the optimal control structure. However there are still several scenarios that require further infestation, such as what happens when there two different operational modes, each with it's on sets of constraints and performance evaluation criterion. For example, how the self-driving vehicle should behave when comparing trajectories that come from different modes, where one mode for example is overtaking and another is lane-following. These two modes are vastly different and it is still not clear how a CPS should react when faced with uncertainty and external agent's behavior. But it is clear that real-time algorithms such as the Adjoint-Based TR1 algorithm presented here will play key role in developing extremely fast decision-making software for specialized applications. All in all, the future is brimming with potential for Cyber-Physical Systems.

Bibliography

[1] Yasin Abbasi-Yadkori, Nevena Lazic, and Csaba Szepesvari. "Regret Bounds for Model-Free Linear Quadratic Control". In: *arXiv:1804.06021* (2018).

[2] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. "Online least squares estimation with self-normalized processes: An application to bandit problems". In: *arXiv preprint arXiv:1102.2670* (2011).

[3] Marshall Abrams and Joe Weiss. "Malicious control system cyber security attack case study–Maroochy Water Services, Australia". In: *MITRE* (2008).

[4] Tobias Achterberg and Timo Berthold. "Improving the feasibility pump". In: *Discrete Optimization* 4.1 (2007), pp. 77–86.

[5] Tobias Achterberg, Timo Berthold, and Gregor Hendel. "Rounding and Propagation Heuristics for Mixed Integer Programming". In: *Operations Research Proceedings 2011*. Springer Berlin Heidelberg, 2012, pp. 71–76.

[6] Tobias Achterberg, Thorsten Koch, and Alexander Martin. "Branching rules revisited". In: *Operations Research Letters* 33.1 (2005), pp. 42–54.

[7] Tobias Achterberg et al. "Presolve reductions in mixed integer programming". In: *ZIB Report* (2016), pp. 16–44.

[8] Amir Ali Ahmadi and Pablo A Parrilo. "Non-monotonic Lyapunov functions for stability of discrete time nonlinear and switched systems". In: *IEEE Conference on Decision and Control (CDC)*. 2008, pp. 614–621.

[9] Ravindra K Ahuja and James B Orlin. "Inverse optimization". In: *Operations Research* 49.5 (2001), pp. 771–783.

[10] Alessandro Alessio, Davide Barcelli, and Alberto Bemporad. "Decentralized model predictive control of dynamically coupled linear systems". In: *Journal of Process Control* 21.5 (2011), pp. 705–714.

[11] Gad Allon, Awi Federgruen, and Margaret Pierson. "How much is a reduction of your customers' wait worth? An empirical study of the fast-food drive-thru industry based on structural estimation methods". In: *Manufacturing & Service Operations Management* 13.4 (2011), pp. 489–507.

[12] Saurabh Amin, Alvaro A Cárdenas, and S Shankar Sastry. "Safe and secure networked control systems under denial-of-service attacks". In: *International Workshop on Hybrid Systems: Computation and Control*. Springer. 2009, pp. 31–45.

[13] J. Andersson. "A General-Purpose Software Framework for Dynamic Optimization". PhD thesis. KU Leuven, Oct. 2013.

[14] Jonas Asprion, Oscar Chinellato, and Lino Guzzella. "Partitioned Quasi-Newton Approximation for Direct Collocation Methods and Its Application to the Fuel-Optimal Control of a Diesel Engine". In: *Journal of Applied Mathematics* 2014 ().

[15] A. Aswani and C. Tomlin. "Game-theoretic routing of GPS-assisted vehicles for energy efficiency". In: *ACC*. 2011, pp. 3375–3380.

[16] A. Aswani et al. "Provably safe and robust learning–based model predictive control". In: *Automatica* 49.5 (2013), pp. 1216–1226.

[17] Anil Aswani. "Statistics with set-valued functions: applications to inverse approximate optimization". In: *Mathematical Programming* 174.1-2 (2019), pp. 225–251.

[18] Anil Aswani. "Statistics with set-valued functions: applications to inverse approximate optimization". In: *Mathematical Programming* 174.1-2 (2019), pp. 225–251.

[19] Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. "Inverse optimization with noisy data". In: *Operations Research* 66.3 (2018), pp. 870–892.

[20] Anil Aswani et al. "Energy-efficient building HVAC control using hybrid system LBMPC". In: *IFAC Proceedings* 45.17 (2012), pp. 496–501.

[21] Anil Aswani et al. "Identifying models of HVAC systems using semiparametric regression". In: *ACC*. 2012, pp. 3675–3680.

[22] D. Axehill and A. Hansson. "A preprocessing algorithm for MIQP solvers with applications to MPC". In: *CDC*. 2004, pp. 2497–2502.

[23] Daniel Axehill and Anders Hansson. "A dual gradient projection quadratic programming algorithm tailored for mixed integer predictive control".

[24] Daniel Axehill and Anders Hansson. "A mixed integer dual quadratic programming algorithm tailored for MPC". In: *Decision and Control, 2006 45th IEEE Conference on*. IEEE. 2006, pp. 5693–5698.

[25] Daniel Axehill, Lieven Vandenberghe, and Anders Hansson. "Convex relaxations for mixed integer predictive control". In: *Automatica* 46.9 (2010), pp. 1540 –1545. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2010.06.015. URL: http://www.sciencedirect.com/science/article/pii/S0005109810002657.

[26] Cheng-Zong Bai, Fabio Pasqualetti, and Vijay Gupta. "Security in stochastic control systems: Fundamental limitations and performance bounds". In: *American Control Conference (ACC), 2015*. IEEE. 2015, pp. 195–200.

[27] Patrick Bajari, C Lanier Benkard, and Jonathan Levin. "Estimating dynamic models of imperfect competition". In: *Econometrica* 75.5 (2007), pp. 1331–1370.

[28] Patrick Bajari and Lixin Ye. "Deciding between competition and collusion". In: *Review of Economics and statistics* 85.4 (2003), pp. 971–989.

[29] Maria-Florina Balcan et al. "Learning to Branch". In: *arXiv preprint arXiv:1803.10150* (2018).

[30] Laura H Baldwin, Robert C Marshall, and Jean-Francois Richard. "Bidder collusion at forest service timber sales". In: *Journal of Political Economy* 105.4 (1997), pp. 657–699.

[31] Sumanta Basu and George Michailidis. "Regularized estimation in sparse high-dimensional time series models". In: *Annals of Statistics* 43.4 (2015), pp. 1535–1567.

[32] Amir Beck and Yonina C Eldar. "Regularization in regression with bounded noise: A Chebyshev center approach". In: *SIAM Journal on Matrix Analysis and Applications* 29.2 (2007), pp. 606–625.

[33] Alberto Bemporad and Manfred Morari. "Control of systems integrating logic, dynamics, and constraints". In: *Automatica* 35 (1999), pp. 407–427.

[34] Alberto Bemporad and Vihangkumar V Naik. "A Numerically Robust Mixed-Integer Quadratic Programming Solver for Embedded Hybrid Model Predictive Control". In: *Proc. 6th IFAC NMPC Conf.* Madison, USA, 2018.

[35] Sorin C Bengea and Raymond A DeCarlo. "Optimal control of switching systems". In: *automatica* 41.1 (2005), pp. 11–27.

[36] Jakub Bernat and Slawomir Stepien. "Multi-modelling as new estimation schema for high-gain observers". In: *International Journal of Control* 88.6 (2015), pp. 1209–1222.

[37] B Douglas Bernheim, Bezalel Peleg, and Michael D Whinston. "Coalition-proof nash equilibria i. concepts". In: *Journal of Economic Theory* 42.1 (1987), pp. 1–12.

[38] Karl Berntorp et al. "Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres". In: *Vehicle System Dynamics* 52.10 (2014), pp. 1304–1332.

[39] D Bertsekas and I Rhodes. "Recursive state estimation for a set-membership description of uncertainty". In: *IEEE TAC* 16.2 (1971), pp. 117–128.

[40] Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. "Data-driven estimation in equilibrium using inverse optimization". In: *Mathematical Programming* 153.2 (2015), pp. 595–633.

[41] J.T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd. SIAM, 2010.

[42] Rajendra Bhatia. *Matrix analysis*. Vol. 169. Springer Science & Business Media, 2013.

[43] Peter J Bickel and Kjell A Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics, Volumes I-II Package*. Chapman and Hall/CRC, 2015.

[44] Lorenz T. Biegler. *Nonlinear Programming*. MOS-SIAM Series on Optimization. SIAM, 2010.

[45] Lorenz T. Biegler. *Nonlinear Programming*. MOS-SIAM Series on Optimization. SIAM, 2010.

[46] L.T. Biegler. "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation". In: *Computers and Chemical Engineering* 8.3–4 (1984), pp. 243–248.

[47] Joaquim Blesa, Vicencc Puig, and Jordi Saludes. "Robust fault detection using polytope-based set-membership consistency test". In: *IET Control Theory & Applications* 6.12 (2012), pp. 1767–1777.

[48] H. G. Bock and K. J. Plitt. "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems". In: *Proc. IFAC World Congr.* Pergamon Press, 1984, pp. 242–247.

[49] Hans Georg Bock. "Recent advances in parameter identification techniques for ODE". In: *Numerical treatment of inverse problems in differential and integral equations*. Springer, 1983, pp. 95–121.

[50] Hans Georg Bock and Karl-Josef Plitt. "A multiple shooting algorithm for direct solution of optimal control problems". In: *IFAC Proceedings Volumes* 17.2 (1984), pp. 1603–1608.

[51] Hans Georg Bock et al. "Numerical Methods for Efficient and Fast Nonlinear Model Predictive Control". In: *Proc. "Int. Workshop on assessment and future directions of NMPC"*. Springer, 2005.

[52] H.G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Vol. 183. Bonner Mathematische Schriften. Bonn: Universität Bonn, 1987.

[53] P. T. Boggs and J. W. Tolle. "Sequential Quadratic Programming". In: *Acta Numerica* (1995), pp. 1–51.

[54] C. Bonebrake and L. Ross O'Neil. "Attacks on GPS Time Reliability". In: *IEEE Security Privacy* 12.3 (2014), pp. 82–84. ISSN: 1540-7993. DOI: 10.1109/MSP.2014.40.

[55] Severin Borenstein. "Rapid price communication and coordination: The airline tariff publishing case (1994)". In: *The Antitrust Revolution: Economics, Competition, and Policy* 4 (2004).

[56] Stephen Boyd et al. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[57] C. G. Broyden. "Quasi-Newton methods and their application to function minimization". In: *Maths. Comp.* 21 (1967), pp. 368–381.

[58] CG Broyden. "On the discovery of the "good Broyden" method". In: *Mathematical programming* 87.2 (2000), pp. 209–213.

[59] Boris Buchmann and Ngai Hang Chan. "Asymptotic theory of least squares estimators for nearly unstable processes under strong dependence". In: *Annals of Statistics* 35.5 (2007), pp. 2001–2017.

[60] Boris Buchmann and Ngai Hang Chan. "Unified asymptotic theory for nearly unstable AR (p) processes". In: *Stochastic Process. Appl.* 123.3 (2013), pp. 952–985.

[61] Emilio Calvano et al. "Algorithmic Pricing What Implications for Competition Policy?" In: *Review of Industrial Organization* 55.1 (2019), pp. 155–171.

[62] Emilio Calvano et al. "Artificial intelligence, algorithmic pricing and collusion". In: (2018).

[63] Alvaro A Cárdenas, Saurabh Amin, and Shankar Sastry. "Research Challenges for the Security of Control Systems." In: *HotSec*. 2008.

[64] Alvaro A Cardenas, Saurabh Amin, and Shankar Sastry. "Secure control: Towards survivable cyber-physical systems". In: *ICDCS*. 2008, pp. 495–500.

[65] Le Chen, Alan Mislove, and Christo Wilson. "An empirical analysis of algorithmic pricing on Amazon marketplace". In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2016, pp. 1339–1349.

[66] Tianshi Chen et al. "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques". In: *IEEE TAC* 59.11 (2014), pp. 2933–2945.

[67] Edward H Clarke. "Multipart pricing of public goods". In: *Public Choice* 11.1 (1971), pp. 17–33.

[68] Joel E Cohen. "Cooperation and self-interest: Pareto-inefficiency of Nash equilibria in finite random games". In: *Proceedings of the National Academy of Sciences* 95.17 (1998), pp. 9724–9731.

[69] Giacomo Como, Enrico Lovisari, and Ketan Savla. "Convexity and Robustness of Dynamic Network Traffic Assignment for Control of Freeway Networks". In: *IFAC-PapersOnLine* 49.3 (2016), pp. 335–340.

[70] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. "Convergence of quasi-Newton matrices generated by the symmetric rank one update". In: *Mathematical Programming* 50.1 (1991), pp. 177–195. ISSN: 1436-4646. DOI: 10.1007/BF01594934. URL: http://dx.doi.org/10.1007/BF01594934.

[71] Andrew R Conn, Nicholas IM Gould, and Ph L Toint. "Convergence of quasi-Newton matrices generated by the symmetric rank one update". In: *Mathematical Programming* 50.1-3 (1991), pp. 177–195.

[72] Samuel Coogan et al. "Energy management via pricing in LQ dynamic games". In: *American Control Conference*. 2013, pp. 443–448.

[73] JR Deller Jr, M Nayeri, and MS Liu. "Unifying the landmark developments in optimal bounding ellipsoid identification". In: *Int J Adapt Control* 8.1 (1994), pp. 43–60.

[74] J. E. Dennis and J. J. Moré. "Quasi-Newton Methods, Motivation and Theory". In: *SIAM Review* 19.1 (1977), pp. 46–89.

[75] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*. Vol. 35. Springer, 2011.

[76] Steven Diamond, Reza Takapoui, and Stephen Boyd. "A general system for heuristic solution of convex problems over nonconvex sets". In: *arXiv preprint arXiv:1601.07277* (2016).

[77] M. Diehl, H. G. Bock, and J. P. Schlöder. "A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control". In: *SIAM Journal on Control and Optimization* 43.5 (2005), pp. 1714–1736. DOI: 10.1137/S0363012902400713. URL: http://epubs.siam.org/sicon/resource/1/sjcodc/v43/i5/p1714_s1.

[78] M. Diehl, H. J. Ferreau, and N. Haverbeke. "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation". In: *Nonlinear model predictive control*. Vol. 384. Lecture Notes in Control and Information Sciences. Springer, 2009, pp. 391–417.

[79] M. Diehl et al. "Nominal Stability of the Real-Time Iteration Scheme for Nonlinear Model Predictive Control". In: *IEE Proc.-Control Theory Appl.* 152.3 (2005), pp. 296–308. DOI: 10.1049/ip-cta:20040008.

[80] Moritz Diehl. *Lecture Notes on Numerical Optimization*. 2016.

[81] Moritz Diehl et al. "An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization". In: *Optimization Methods & Software* 25.4 (2010), pp. 531–552.

[82] Moritz Diehl et al. "An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization". In: *Opt. Meth. Softw.* 25.4 (2010), pp. 531–552. DOI: 10.1080/10556780903027500. URL: http://www.tandfonline.com/doi/abs/10.1080/10556780903027500.

[83] Francis Y Edgeworth. "The pure theory of monopoly". In: *Papers relating to political economy* 1 (1925), pp. 111–142.

[84] Ariel Ezrachi and Maurice E Stucke. "Artificial intelligence & collusion: When computers inhibit competition". In: *U. Ill. L. Rev.* (2017), p. 1775.

[85] Fariba Fahroo and I. Michael Ross. "A spectral patching method for direct trajectory optimization". In: *Journal of the Astronautical Sciences* 48.106 (2000), pp. 269–286.

[86] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. "Finite time identification in unstable linear systems". In: *Automatica* 96 (2018), pp. 342–353.

[87] Farzaneh Farhadi, S Jamaloddin Golestani, and Demosthenis Teneketzis. "A surrogate optimization-based mechanism for resource allocation and routing in networks with strategic agents". In: *IEEE Transactions on Automatic Control* 64.2 (2018), pp. 464–479.

[88] Marcello Farina and Riccardo Scattolini. "Distributed non-cooperative MPC with neighbor-to-neighbor communication". In: *IFAC Proceedings* 44.1 (2011), pp. 404–409.

[89] Joseph Farrell. "Communication, coordination and Nash equilibrium". In: *Economics Letters* 27.3 (1988), pp. 209–214.

[90] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. "Secure estimation and control for cyber-physical systems under adversarial attacks". In: *IEEE Transactions on Automatic Control* 59.6 (2014), pp. 1454–1467.

[91] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. "Secure state-estimation for dynamical systems under active adversaries". In: *Allerton Conference*. IEEE. 2011, pp. 337–344.

[92] Riccardo M Ferrari and André M Teixeira. "Detection and isolation of replay attacks through sensor watermarking". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 7363–7368.

[93] Giancarlo Ferrari-Trecate et al. "Model predictive control schemes for consensus in multi-agent systems with single-and double-integrator dynamics". In: *IEEE Transactions on Automatic Control* 54.11 (2009), pp. 2560–2572.

[94] H. J. Ferreau et al. "Embedded Optimization Methods for Industrial Automatic Control". In: 2017.

[95] Christodoulos A Floudas. *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press, 1995.

[96] Damian Frick, Alexander Domahidi, and Manfred Morari. "Embedded optimization for mixed logical dynamical systems". In: *Computers & Chemical Engineering* 72 (2015), pp. 21–33.

[97] Jun Fu et al. "Local optimization of dynamic programs with guaranteed satisfaction of path constraints". In: *Automatica* 62 (2015), pp. 184 –192. ISSN: 0005-1098.

[98] Alexander J Gallo et al. "Distributed watermarking for secure control of microgrids under replay attacks". In: *IFAC-PapersOnLine* 51.23 (2018), pp. 182–187.

[99] Andrea Garulli, Simone Paoletti, and Antonio Vicino. "A survey on switched and piecewise affine system identification". In: *IFAC Proceedings Volumes* 45.16 (2012), pp. 344–355.

[100] Andrea Garulli, Antonio Vicino, and Giovanni Zappa. "Conditional central algorithms for worst case set-membership identification and filtering". In: *IEEE TAC* 45.1 (2000), pp. 14–23.

[101] David M. Gay. *Representing Symmetric Rank Two Updates*. Working Paper 124. National Bureau of Economic Research, 1976. DOI: 10.3386/w0124. URL: http://www.nber.org/papers/w0124.

[102] Matthias Gerdts. "Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift". In: *Optimal Control Appl. and Methods* 26.1 (), pp. 1–18. DOI: 10.1002/oca.751. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.751. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.751.

[103] Franco Giannessi, Antonino Maugeri, and Panos M Pardalos. *Equilibrium problems: nonsmooth optimization and variational inequality models*. Vol. 58. Springer Science & Business Media, 2006.

[104] P. Gill, W. Murray, and M. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Review* 47.1 (2005), pp. 99–131. DOI: 10.1137/S0036144504446096. eprint: http://epubs.siam.org/doi/pdf/10.1137/S0036144504446096. URL: http://epubs.siam.org/doi/abs/10.1137/S0036144504446096.

[105] John C Gittins. "Bandit processes and dynamic allocation indices". In: *JRSS, B.* (1979), pp. 148–177.

[106] Antonio Gomez-Exposito and Ali Abur. *Power system state estimation: theory and implementation*. CRC press, 2004.

[107] H Gonzalez and E Polak. "On the perpetual collision-free RHC of fleets of vehicles". In: *Journal of optimization theory and applications* 145.1 (2010), pp. 76–92.

[108] Joseph Greenberg. "Coalition structures". In: *Handbook of game theory with economic applications* 2 (1994), pp. 1305–1337.

[109] A. Griewank. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. Frontiers in Appl. Math. 19. Philadelphia: SIAM, 2000.

[110] A. Griewank and Ph. L. Toint. "Local convergence analysis for partitioned quasi-Newton updates". In: *Numerische Mathematik* 39.3 (1982), pp. 429–448. ISSN: 0945-3245. DOI: 10.1007/BF01407874. URL: http://dx.doi.org/10.1007/BF01407874.

[111] A. Griewank and Ph.L. Toint. "Partitioned variable metric updates for large structured optimization problems". In: *Numerische Mathematik* 39 (1982), pp. 119–137. DOI: 10.1007/BF01399316. URL: http://www.springerlink.com/content/l7x1j4112mt73616/.

[112] A. Griewank and A. Walther. "On Constrained Optimization by Adjoint based quasi-Newton Methods". In: *Optimization Methods and Software* 17 (2002), pp. 869–889.

[113] Andreas Griewank and Andrea Walther. "On constrained optimization by adjoint based quasi-Newton methods". In: *Optimization Methods and Software* 17.5 (2002), pp. 869–889.

[114] Andreas Griewank, Andrea Walther, and Maciek Korzec. "Maintaining factorized KKT systems subject to rank-one updates of Hessians and Jacobians". In: *Optimization Methods and Software* 22.2 (2007), pp. 279–295. DOI: 10.1080/10556780500487867. eprint: http://dx.doi.org/10.1080/10556780500487867. URL: http://dx.doi.org/10.1080/10556780500487867.

[115] Theodore Groves. "Incentives in teams". In: *Econometrica* (1973), pp. 617–631.

[116] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2018.

[117] W.M. Haddad and V.S. Chellaboina. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, 2011.

[118] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems*. 2nd. Berlin Heidelberg: Springer, 1991.

[119] Patrick T Harker and Jong-Shi Pang. "Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications". In: *Mathematical programming* 48.1-3 (1990), pp. 161–220.

[120] Malo LJ Hautus. "A simple proof of Heymann's lemma". In: *IEEE Transactions on Automatic Control* 22.5 (1977), pp. 885–886.

[121] P. Hespanhol and R. Quirynen. "A Real-Time Iteration Scheme with Quasi-Newton Jacobian Updates for Nonlinear Model Predictive Control". In: *2018 European Control Conference (ECC)*. 2018, pp. 1517–1522.

[122] P. Hespanhol and R. Quirynen. "A Real-Time Iteration Scheme with Quasi-Newton Jacobian Updates for Nonlinear Model Predictive Control". In: *Proc. European Control Conf. (ECC)*. 2018.

[123] P. Hespanhol and R. Quirynen. "Quasi-Newton Jacobian and Hessian Updates for Pseudospectral based NMPC". In: *IFAC Conference on Nonlinear Model Predictive Control*. Vol. 51. 20. 2018, pp. 22 –27.

[124] Pedro Hespanhol and Anil Aswani. "Hypothesis Testing Approach to Detecting Collusion in Competitive Environments". In: *arXiv preprint arXiv:2003.09967* (2020).

[125] Pedro Hespanhol and Anil Aswani. "Statistically-Consistent Identification of Switched Linear Systems". In: *arXiv preprint arXiv:1903.07552* (2019).

[126] Pedro Hespanhol and Anil Aswani. "Surrogate Optimal Control for Strategic Multi-Agent Systems". In: *arXiv preprint arXiv:1903.07559* (2019).

[127] Pedro Hespanhol and Rien Quirynen. "Adjoint-based SQP method with block-wise quasi-Newton Jacobian updates for nonlinear optimal control". In: *Optimization Methods and Software* (2019), pp. 1–29.

[128] Pedro Hespanhol and Rien Quirynen. "Quasi-Newton Jacobian and Hessian updates for pseudospectral based NMPC". In: *IFAC-PapersOnLine* 51.20 (2018), pp. 22–27.

[129] Pedro Hespanhol, Rien Quirynen, and Stefano Di Cairano. "A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 2763–2768.

[130] Pedro Hespanhol et al. "Dynamic watermarking for general LTI systems". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 1834–1839.

[131] Pedro Hespanhol et al. "Sensor Switching Control Under Attacks Detectable by Finite Sample Dynamic Watermarking Tests". In: *arXiv preprint arXiv:1909.00014* (2019).

[132] Pedro Hespanhol et al. "Statistical Watermarking for Networked Control Systems". In: *American Control Conference (ACC)*. 2018, pp. 5467–5472.

[133] Pedro Hespanhol et al. "Statistical watermarking for networked control systems". In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 5467–5472.

[134] Michael Heymann. "Comments on "On pole assignment in multi-input controllable linear systems"". In: *IEEE Transactions on Automatic Control* 13.6 (1968), pp. 748–749.

[135] Dennis Janka et al. "An SR1/BFGS SQP algorithm for nonconvex nonlinear programs with block-diagonal Hessian matrix". In: *Mathematical Programming Computation* 8.4 (2016), pp. 435–459.

[136] Ramesh Johari and John Tsitsiklis. "Efficiency loss in a network resource allocation game". In: *Math. Oper. Res.* 29.3 (2004), pp. 407–435.

[137] Ramesh Johari and John N Tsitsiklis. "Efficiency of scalar-parameterized mechanisms". In: *Operations Research* 57.4 (2009), pp. 823–839.

[138] Tosio Kato. *Perturbation theory for linear operators*. Vol. 132. Springer Science & Business Media, 2013.

[139] Frank Kelly. "Charging and rate control for elastic traffic". In: *European transactions on Telecommunications* 8.1 (1997), pp. 33–37.

[140] Frank Kelly, Aman Maulloo, and David Tan. "Rate control for communication networks: shadow prices, proportional fairness and stability". In: *J. Oper. Res. Soc.* 49.3 (1998), pp. 237–252.

[141] Kyoung-Dae Kim and Panganamala R Kumar. "Cyber–physical systems: A perspective at the centennial". In: *Proc. of IEEE* 100 (2012), pp. 1287–1308.

[142] Christian Kirches. "Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control". PhD thesis. Uni. Heidelberg, 2010.

[143] Woo-Hyun Ko, Bharadwaj Satchidanandan, and PR Kumar. "Theory and implementation of dynamic watermarking for cybersecurity of advanced transportation systems". In: *Proc. of IEEE CNS*. 2016, pp. 416–420.

[144] PR Kumar. "Convergence of adaptive control schemes using least-squares parameter estimates". In: *IEEE TAC* 35.4 (1990), pp. 416–424.

[145] Vipin Kumar, Jaideep Srivastava, and Aleksandar Lazarevic. *Managing cyber threats: issues, approaches, and challenges*. Vol. 5. Springer, 2006.

[146] TL Lai and CZ Wei. "Asymptotic properties of general autoregressive models and strong consistency of least-squares estimates of their parameters". In: *J. Multivar. Anal.* 13.1 (1983), pp. 1–23.

[147] TL Lai and CZ Wei. "Asymptotic properties of multivariate weighted sums with applications to stochastic regression in linear dynamic systems". In: *Multivariate Analysis VI* (1985), pp. 375–393.

[148] Tze Leung Lai and Herbert Robbins. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[149] Ralph Langner. "Stuxnet: Dissecting a cyberwarfare weapon". In: *IEEE Security & Privacy* 9.3 (2011), pp. 49–51.

[150] Fabien Lauer and Gérard Bloch. "Hybrid system identification". In: *Hybrid System Identification*. Springer, 2019, pp. 77–101.

[151] Pierre Le Bodic and George Nemhauser. "An abstract model for branching and its application to mixed integer programming". In: *Mathematical Programming* 166.1-2 (2017), pp. 369–405.

[152] Na Li and Jason R Marden. "Designing games to handle coupled constraints". In: *Conference on Decision and Control*. 2010, pp. 250–255.

[153] Xin Li and Leen-Kiat Soh. "Investigating reinforcement learning in multiagent coalition formation". In: *Amer. Assoc. Artif. Intell. Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems Tech. Rep. WS-04-06*. 2004, pp. 22–28.

[154] Hubert W Lilliefors. "On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown". In: *Journal of the American Statistical Association* 64.325 (1969), pp. 387–389.

[155] TM Lin, M Nayeri, and JR Deller Jr. "A consistently convergent OBE algorithm with automatic estimation of error bounds". In: *Int J Adapt Control* 12.4 (1998), pp. 305–324.

[156] Lennart Ljung. *System Identification*. Prentice-hall, 1987.

[157] Zhi-Quan Luo et al. "Semidefinite relaxation of quadratic optimization problems". In: *IEEE Signal Processing Magazine* 27.3 (2010), pp. 20–34.

[158] Yudong Ma, Garrett Anderson, and Francesco Borrelli. "A distributed predictive control approach to building temperature regulation". In: *American Control Conference*. 2011, pp. 2089–2094.

[159] Lester Mackey et al. "Matrix concentration inequalities via the method of exchangeable pairs". In: *The Annals of Probability* 42.3 (2014), pp. 906–945.

[160] P. Maponi. "The solution of linear systems by using the Sherman–Morrison formula". In: *Linear Algebra and its Applications* 420.2 (2007), pp. 276 –294. ISSN: 0024-3795. DOI: https://doi.org/10.1016/j.laa.2006.07.007. URL: http://www.sciencedirect.com/science/article/pii/S0024379506003351.

[161] Jason Marden and Adam Wierman. "Overcoming limitations of game-theoretic distributed control". In: *CDC*. 2009, pp. 6466–6471.

[162] Frank J Massey Jr. "The Kolmogorov-Smirnov test for goodness of fit". In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.

[163] D. Mayne and J. Rawlings. *Model Predictive Control*. Nob Hill, 2013.

[164] Adam J Mersereau, Paat Rusmevichientong, and John N Tsitsiklis. "A structured multiarmed bandit problem and the greedy policy". In: *IEEE TAC* 54.12 (2009), pp. 2787–2802.

[165] Mario Milanese and Antonio Vicino. "Optimal estimation theory for dynamic systems with set membership uncertainty: an overview". In: *Automatica* 27.6 (1991), pp. 997–1009.

[166] Y. Mintz et al. "Behavioral Analytics for Myopic Agents". In: *arXiv:1702.05496* (2017).

[167] Yonatan Mintz et al. "Control synthesis for bilevel linear model predictive control". In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 2338–2343.

[168] Yonatan Mintz et al. "Non-Stationary Bandits with Habituation and Recovery Dynamics". In: *Operations Research* (2019). Accepted.

[169] Aritra Mitra and Shreyas Sundaram. "Distributed observers for lti systems". In: *IEEE Transactions on Automatic Control* 63.11 (2018), pp. 3689–3704.

[170] Aritra Mitra and Shreyas Sundaram. "Secure distributed observers for a class of linear time invariant systems in the presence of byzantine adversaries". In: *IEEE Conference on Decision and Control (CDC)*. 2016, pp. 2709–2714.

[171] Yilin Mo, Rohan Chabukswar, and Bruno Sinopoli. "Detecting integrity attacks on SCADA systems". In: *IEEE CST* 22.4 (2014), pp. 1396–1407.

[172] Yilin Mo and Bruno Sinopoli. "Secure control against replay attacks". In: *Allerton Conference*. IEEE. 2009, pp. 911–918.

[173] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs". In: *IEEE Control Systems* 35.1 (2015), pp. 93–109.

[174] Yilin Mo et al. "False data injection attacks against state estimation in wireless sensor networks". In: *Proc. of IEEE CDC*. 2010, pp. 5967–5972.

[175] Shankar Mohan and Ram Vasudevan. "Convex computation of the reachable set for hybrid systems with parametric uncertainty". In: *Proc. of ACC*. 2016, pp. 5141–5147.

[176] Mosek, ApS. *The MOSEK optimization toolbox for MATLAB manual*. 2015.

[177] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual.* 2017.

[178] Vihangkumar V Naik and Alberto Bemporad. "Embedded mixed-integer quadratic optimization using accelerated dual gradient projection". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 10723–10728.

[179] Reinhard Neck and Engelbert Dockner. "Conflict and cooperation in a model of stabilization policies: A differential game approach". In: *Journal of Economic Dynamics and Control* 11.2 (1987), pp. 153–158.

[180] Rudy R Negenborn et al. "Distributed model predictive control of irrigation canals." In: *NHM* 4.2 (2009), pp. 359–380.

[181] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience, 1988. ISBN: 0-471-82819-X.

[182] Bent Nielsen. "Order determination in general vector autoregressions". In: *Time series and related topics*. IMS, 2006, pp. 93–112.

[183] Bent Nielsen. "Singular vector autoregressions with deterministic terms: Strong consistency and lag order determination." In: (2008).

[184] Bent Nielsen. "Strong consistency results for least squares estimators in general vector autoregressions with deterministic terms". In: *Econometric Theory* 21.3 (2005), pp. 534–561.

[185] Tyler Nighswander et al. "GPS Software Attacks". In: *ACM CCS*. 2012, pp. 450–461.

[186] Jorge Nocedal and S Wright. "Numerical optimization: Springer science & business media". In: *New York* (2006).

[187] Maurício C de Oliveira, Jacques Bernussou, and José C Geromel. "A new discrete-time robust stability condition". In: *Systems & control letters* 37.4 (1999), pp. 261–265.

[188] A.M. Ostrowski. *Solutions of Equations and Systems of Equations*. New York: Academic Press, 1966.

[189] Yi Ouyang, Mukul Gagrani, and Rahul Jain. "Learning-based Control of Unknown Linear Systems with Thompson Sampling". In: *arXiv preprint arXiv:1709.04047* (2017).

[190] Samet Oymak and Necmiye Ozay. "Non-asymptotic identification of lti systems from a single trajectory". In: *ACC* (2019).

[191] Necmiye Ozay et al. "A sparsification approach to set membership identification of switched affine systems". In: *IEEE TAC* 57.3 (2011), pp. 634–648.

[192] Bryan Parno et al. "Secure sensor network routing: A clean-slate approach". In: *ACM CoNEXT*. 2006.

[193] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. "Attack detection and identification in cyber-physical systems". In: *IEEE Transactions on Automatic Control* 58.11 (2013), pp. 2715–2729.

[194] Martin Pesendorfer. "A study of collusion in first-price auctions". In: *The Review of Economic Studies* 67.3 (2000), pp. 381–411.

[195] Peter CB Phillips and Tassos Magdalinos. "Inconsistent VAR regression with common explosive roots". In: *Econometric Theory* 29.4 (2013), pp. 808–837.

[196] Matthew Porter et al. "Detecting Deception Attacks on Autonomous Vehicles via Linear Time-Varying Dynamic Watermarking". In: *arXiv preprint arXiv:2001.09859* (2020).

[197] Matthew Porter et al. "Detecting Generalized Replay Attacks via Time-Varying Dynamic Watermarking". In: *arXiv preprint arXiv:1909.08111* (2019).

[198] Robert H Porter and J Douglas Zona. "Detection of bid rigging in procurement auctions". In: *Journal of political economy* 101.3 (1993), pp. 518–538.

[199] Andreas Potschka. "A direct method for the numerical solution of optimization problems with time-periodic PDE constraints". PhD thesis. University of Heidelberg, 2011. URL: `http://www.iwr.uni-heidelberg.de/~Andreas.Potschka/publications.html`.

[200] Andreas Potschka, Hans Georg Bock, and Johannes P. Schlöder. "A minima tracking variant of semi-infinite programming for the treatment of path constraints within direct solution of optimal control problems". In: *Optimization Methods and Software* 24.2 (2009), pp. 237–252. DOI: `10.1080/10556780902753098`. eprint: `http://dx.doi.org/10.1080/10556780902753098`. URL: `http://dx.doi.org/10.1080/10556780902753098`.

[201] R. Quirynen, K. Berntorp, and S. Di Cairano. "Embedded Optimization Algorithms for Steering in Autonomous Vehicles based on Nonlinear Model Predictive Control". In: *Proc. American Control Conference (ACC)*. 2018.

[202] R. Quirynen and M. Diehl. "Lifted Newton-Type Optimization for Pseudospectral Methods in Nonlinear Model Predictive Control". In: *Proc. American Control Conf. (ACC)*. 2018.

[203] R. Quirynen, S. Gros, and M. Diehl. "Inexact Newton-Type Optimization with Iterated Sensitivities". In: *SIAM Journal on Optimization* 28.1 (2018), pp. 74–95.

[204] R. Quirynen, A. Knyazev, and S. Di Cairano. "Block Structured Preconditioning within an Active-Set Method for Real-Time Optimal Control". In: *Proc. European Control Conference (ECC)*. 2018.

[205] R. Quirynen, A. Knyazev, and S. Di Cairano. "Block Structured Preconditioning within an Active-Set Method for Real-Time Optimal Control". In: *Proc. European Control Conf. (ECC)*. 2018.

[206] Rien Quirynen. "Numerical Simulation Methods for Embedded Optimization". PhD thesis. KU Leuven and University of Freiburg, 2017.

[207] Rien Quirynen et al. "Lifted collocation integrators for direct optimal control in ACADO Toolkit". In: *Math. Prog. Computation* 9.4 (2017), pp. 527–571. ISSN: 1867-2957. DOI: 10.1007/s12532-017-0119-0. URL: http://dx.doi.org/10.1007/s12532-017-0119-0.

[208] Rien Quirynen et al. "Lifted collocation integrators for direct optimal control in ACADO Toolkit". In: *Math. Progr. Comp.* 9.4 (2017), pp. 527–571. ISSN: 1867-2957. DOI: 10.1007/s12532-017-0119-0. URL: http://dx.doi.org/10.1007/s12532-017-0119-0.

[209] Anil V. Rao. "A Survey of Numerical Methods for Optimal Control". In: *Adv. Astron. Sciences* 135.1 (2010).

[210] Lillian Ratliff et al. "Pricing in linear-quadratic dynamic games". In: *Allerton*. 2012, pp. 1798–1805.

[211] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2nd Edition. Nob Hill, 2017.

[212] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.

[213] Stefan Reichelstein and Stanley Reiter. "Game forms with minimal message spaces". In: *Econometrica* (1988), pp. 661–692.

[214] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*. Vol. 317. Springer Science & Business Media, 2009.

[215] R. Romagnoli, S. Weerakkody, and B. Sinopoli. "A Model Inversion Based Watermark for Replay Attack Detection with Output Tracking". In: *American Control Conference (ACC)*. 2019, pp. 384–390.

[216] I. Michael Ross and Fariba Fahroo. "Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems". In: *J. Guidance, Control, Dynamics* 27.3 (2004), pp. 397–405.

[217] John Rust. "Structural estimation of Markov decision processes". In: *Handbook of econometrics* 4 (1994), pp. 3081–3143.

[218] Walid Saad, Zhu Han, and H Vincent Poor. "Coalitional game theory for cooperative micro-grid distribution networks". In: *2011 IEEE international conference on communications workshops (ICC)*. IEEE. 2011, pp. 1–5.

[219] Walid Saad et al. "Coalitional game theory for communication networks: A tutorial". In: *arXiv preprint arXiv:0905.4057* (2009).

[220] S. Sager, H.G. Bock, and M. Diehl. "Solving Mixed-integer Control Problems by Sum Up Rounding With Guaranteed Integer Gap". In: *SIAM Journal on Control and Optimization* (2008).

[221] Bruno Salcedo. "Pricing algorithms and tacit collusion". In: *Manuscript, Pennsylvania State University* (2015).

[222] Bharadwaj Satchidanandan and PR Kumar. "Dynamic Watermarking: Active Defense of Networked Cyber-Physical Systems". In: *Proc. of IEEE* (2016).

[223] Bharadwaj Satchidanandan and PR Kumar. "On minimal tests of sensor veracity for dynamic watermarking-based defense of cyber-physical systems". In: *9th International Conference on Communication Systems and Networks (COMSNETS)*. 2017, pp. 23–30.

[224] Martin WP Savelsbergh. "Preprocessing and probing techniques for mixed integer programming problems". In: *ORSA Journal on Computing* 6.4 (1994), pp. 445–454.

[225] Thomas A Severini. *Likelihood methods in statistics*. Oxford University Press, 2000.

[226] Qaisar Shafi. "Cyber Physical Systems Security: A Brief Survey". In: *2012 12th International Conference on Computational Science and Its Applications*. 2012, pp. 146–150.

[227] Jeff Shamma. *Cooperative control of distributed multi-agent systems*. John Wiley & Sons, 2008.

[228] Daniel P. Shepard, Todd E. Humphreys, and Aaron A. Fansler. "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks". In: *International Journal of Critical Infrastructure Protection* 5.3 (2012), pp. 146–153.

[229] Max Simchowitz, Ross Boczar, and Benjamin Recht. "Learning Linear Dynamical Systems with Semi-Parametric Least Squares". In: *arXiv preprint arXiv:1902.00768* (2019).

[230] Max Simchowitz et al. "Learning Without Mixing: Towards A Sharp Analysis of Linear System Identification". In: *arXiv preprint arXiv:1802.08334* (2018).

[231] Torsten Söderström. *Discrete-Time Stochastic Systems: Estimation and Control*. Springer Science & Business Media, 2012.

[232] Torsten Söderström and Petre Stoica. "System identification". In: (1989).

[233] John M Staatz. "The cooperative as a coalition: a game-theoretic approach". In: *American Journal of Agricultural Economics* 65.5 (1983), pp. 1084–1089.

[234] Charles Stein et al. "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables". In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. The Regents of the University of California. 1972.

[235] B. Stellato, T. Geyer, and P. J. Goulart. "High-Speed Finite Control Set Model Predictive Control for Power Electronics". In: *IEEE Transactions on Power Electronics* 32.5 (2017), pp. 4007–4020. ISSN: 0885-8993. DOI: 10.1109/TPEL.2016.2584678.

[236] B Stellato et al. "Embedded mixed-integer quadratic optimization using the OSQP solver". In: *European Control Conference*. 2018.

[237] Brett T Stewart et al. "Cooperative distributed model predictive control". In: *Systems & Control Letters* 59.8 (2010), pp. 460–469.

[238] A. M. Teixeira et al. "Revealing stealthy attacks in control systems". In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2012, pp. 1806–1813. DOI: 10.1109/Allerton.2012.6483441.

[239] F Tjarnstrom and Andrea Garulli. "A mixed probabilistic/bounded-error approach to parameter estimation in the presence of amplitude bounded white noise". In: *IEEE CDC*. Vol. 3. 2002, pp. 3422–3427.

[240] Fabio Danilo Torrisi and Alberto Bemporad. "HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems". In: *IEEE trans. control sys. techn.* 12.2 (2004), pp. 235–249.

[241] Valerio Turri et al. "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads". In: *Proc. of IEEE ITSC*. 2013, pp. 378–383.

[242] A Galip Ulsoy, Huei Peng, and Melih cCakmakci. *Automotive control systems*. Cambridge University Press, 2012.

[243] Ram Vasudevan et al. "Safe semi-autonomous control with enhanced driver modeling". In: *ACC*. 2012, pp. 2896–2903.

[244] Aswin Venkat, James Rawlings, and Stephen Wright. "Stability and optimality of distributed model predictive control". In: *Conference on Decision and Control*. 2005, pp. 6680–6685.

[245] Aswin Venkat et al. "Distributed MPC strategies with application to power system automatic generation control". In: *IEEE T-CST* 16.6 (2008), pp. 1192–1206.

[246] William Vickrey. "Counterspeculation, auctions, and competitive sealed tenders". In: *The Journal of Finance* 16.1 (1961), pp. 8–37.

[247] René Vidal. "Recursive identification of switched ARX systems". In: *Automatica* 44.9 (2008), pp. 2274–2287.

[248] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (2006), pp. 25–57.

[249] Abraham Wald. "Note on the consistency of the maximum likelihood estimate". In: *The Annals of Mathematical Statistics* 20.4 (1949), pp. 595–601.

[250] A. Walsh, S. Di Cairano, and A. Weiss. "MPC for Coupled Station Keeping, Attitude Control, and Momentum Management of Low-Thrust Geostationary Satellites". In: *ACC*. July 2016, pp. 7408–7413. DOI: 10.1109/ACC.2016.7526842. URL: http://www.merl.com/publications/TR2016-047.

[251] Wenye Wang and Zhuo Lu. "Cyber security in the Smart Grid: Survey and challenges". In: *Computer Networks* 57.5 (2013), pp. 1344–1371.

[252] Sean Weerakkody, Yilin Mo, and Bruno Sinopoli. "Detecting integrity attacks on control systems using robust physical watermarking". In: *Proc. of IEEE CDC*. 2014, pp. 3757–3764.

[253] L. Wirsching, H. G. Bock, and M. Diehl. "Fast NMPC of a chain of masses connected by springs". In: *Proc. IEEE International Conference on Control Applications, Munich*. 2006, pp. 591–596. DOI: 10.1109/CACSD-CCA-ISIC.2006.4776712.

[254] Sichao Yang and Bruce Hajek. "VCG-Kelly mechanisms for allocation of divisible goods: Adapting VCG mechanisms to one-dimensional signals". In: *IEEE Journal on Selected Areas in Communications* 25.6 (2007), pp. 1237–1243.

[255] Wei Zhang et al. "A hierarchical flight planning framework for air traffic management". In: *Proceedings of the IEEE* 100.1 (2012), pp. 179–194.

[256] Mattia Zorzi and Alessandro Chiuso. "Sparse plus low rank network identification: A nonparametric approach". In: *Automatica* 76 (2017), pp. 355–366.