Control and State-Estimation of Jump Stochastic Systems by Learning Recurrent Spatiotemporal Patterns

Thesis by
SooJean Han

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended January 12th, 2023

SooJean Han
ORCID: 0000-0003-1195-6465

I'd like to thank my academic advisors Professor John C. Doyle and Professor Soon-Jo Chung for their unwavering support, mentorship, and guidance throughout my graduate student career. They have broadened my perspective with stimulating discussions, and helped me gain the necessary skills to tackle research in a rapidly-progressing field like control engineering. I will always be grateful for Professor Chung's availability to hold consistent weekly meetings with me as well as his thorough reading of my papers and insightful feedback, which have kept me productive and open-minded in my research.

I thank Professor Richard M. Murray, Professor Michelle Effros, and Professor Adam Wierman for being valuable coauthors. It was through their time and counsel where I learned the most about collaboration and project management. Professor Murray and Professor Effros have dedicated their time and energy to meet with me almost biweekly for four years, and they have given me invaluable feedback and advice that I will remember forever.

I will eternally cherish the fun memories and deep conversations I've shared with my friends and labmates at Caltech: Carmen Amo Alonso, James Anderson, Natalie Bernat, Jiexin Chen, Lauren Conger, Dimitar (Dim) Ho, Mandy Huo, Lisa Li, Nikolai (Nik) Matni, Noah Olsman, Anish Sarma, Shih-Hao Tseng, Fangzhou (Fang) Xiao, and Jing Yu in the Doyle lab; Matt Anderson, Karena Cai, Michael (Mike) O' Connell, Lu Gan, John Lathrop, Connor Lee, Sorina Lupu, Kai Matsuka, Yashwanth Kumar Nakka, Jimmy Ragan, Benjamin (Ben) Riviere, Guanya Shi, Ellande Tang, Hiroyasu (Hiro) Tsukamoto, and Fengze Xie in the ARCL Lab; Sara Beery, Elijah (Eli) Cole, Gautam Goel, Riley Murray, and Jennifer Sun as notable members of the CMS department.

Finally, I would like to express deep gratitude to my parents for their encouragement throughout this long journey.

This thesis establishes control and estimation architectures that combine both model-based and model-free methods by theoretically characterizing several types of *jump stochastic systems (JSSs)*, i.e., systems with random and repetitive jump phenomena. By expanding the capabilities of model-based stochastic control and estimation, there is potential for artificial intelligence to be implemented as a supplement to theory-influenced design instead of being used end-to-end. We begin by deriving sufficient conditions for stochastic incremental stability for nonlinear systems perturbed by two types of non-Gaussian noise: 1) shot noise processes represented as compound Poisson processes, and 2) finite-measure Lévy processes constructed as affine combinations of Gaussian white and Poisson shot noise processes. We then present a controller architecture based on a concept we call *pattern-learning for prediction (PLP)* for discrete-time/discrete-event systems, in which we can take advantage of the fact that the underlying jump process is a sequence of random variables that occurs as repeated patterns of interest. Finally, we demonstrate control and estimation for JSSs in three real-world applications. First, we consider the control of networks with dynamic topology (e.g., power grid with fault-tolerance to downed lines), for which PLP is integrated with variations of the novel system-level synthesis framework for disturbance-rejection. Second, we perform congestion control of vehicle traffic flow over metropolitan intersection networks, for which PLP is extended to *pattern-learning with memory and prediction (PLMP)* via the inclusion of episodic control, designed to reduce memory consumption by exploiting structural symmetries and temporal repetition in the network. Third, we perform estimation and forecasting (the dual problem to control) for epidemic spread throughout a population network under jump phenomena such as superspreader effects and the emergence of variant viruses. Our results indicate that learning patterns in the jump process makes controller/observer design efficient in data-consumption and computation time, which suggests that it can potentially be used for other JSSs in the real world.

[1] SooJean Han, Soon-Jo Chung, and John C. Doyle. "Predictive Control of Linear Discrete-Time Markovian Jump Systems by Learning Recurrent Patterns". In: *Automatica* 156 (Oct. 2023), p. 111197.
SJ Han participated in the conception of the project, development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text. [Link].

[2] SooJean Han and Soon-Jo Chung. "Incremental Nonlinear Stability Analysis for Stochastic Systems Perturbed by Lévy Noise". In: *International Journal of Robust and Nonlinear Control* 32.12 (June 2022), pp. 7174–7201.
SJ Han participated in the conception of the project, development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text. [Link].

[3] SooJean Han, Soon-Jo Chung, and Johanna Gustafson. "Congestion Control of Vehicle Traffic Networks by Learning Structural and Temporal Patterns". In: vol. 211. Proceedings of Machine Learning Research. PMLR, June 2023, pp. 903–914.
SJ Han participated in the conception of the project, development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text. [Link].

[4] SooJean Han. "Localized Learning of Robust Controllers for Networked Systems with Dynamic Topology". In: *Proceedings of the 2nd Conference on Learning for Dynamics and Control (L4DC)*. Vol. 120. Proceedings of Machine Learning Research. PMLR, June 2020, pp. 687–696.
SJ Han participated in the conception of the project, development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text. [Link].

[5] SooJean Han, Soon-Jo Chung, Shuyue Yu, and John C. Doyle. "Modeling and Estimation of Epidemic Spread in Population Networks using Compartmental ODEs and Coupled Hidden Markov Models". In: *SIAM Journal on Control and Optimization (SICON)* (2023).
To be submitted. **SJ Han participated in the conception of the project, development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text.** [Link].

[6] SooJean Han, Soon-Jo Chung, Shuyue Yu, and John C. Doyle. "A New Insight Into the Modeling and Estimation of Epidemic Spread in Population Networks Using Com-

partmental ODEs and Hidden Markov Models". In: (Dec. 2020).
SJ Han participated in the conception of the project, development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text. [Link].

[7] SooJean Han, Michelle Effros, and Richard M. Murray. "OUTformation: Distributed Data Gathering with Feedback under Unknown Environment and Communication Delay Constraints". In: *ArXiv* (Aug. 2022).
SJ Han participated in the development of the problem formulation, the theoretical analyses, design of the simulation experiments, generation of the figures, and the writing of the text. [Link].

[8] Shih-Hao Tseng, SooJean Han, and Adam Wierman. "Trading Throughput for Freshness: Freshness-Aware Traffic Engineering and In-Network Freshness Control". In: *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (ToMPECS)* 8.4 (Mar. 2023), pp. 1–26.
SJ Han participated in the theoretical analyses and the writing of the text. [Link].

[9] Mo Chen, Sylvia L. Herbert, Haimin Hu, Ye Pu, Jaime Fernández Fisac, Somil Bansal, SooJean Han, and Claire J. Tomlin. "FaSTrack:A Modular Framework for Real-Time Motion Planning and Guaranteed Safe Tracking". In: *IEEE Transactions on Automatic Control* 66.12 (2021), pp. 5861–5876.
SJ Han participated in the design of the simulation experiments and the generation of the figures. [Link].

[10] Shih-Hao Tseng, Carmen Amo Alonso, and SooJean Han. "System Level Synthesis via Dynamic Programming". In: *IEEE 59th Annual Conference on Decision and Control*. Dec. 2020, pp. 1718–1725.
SJ Han participated in the theoretical analyses, generation of the figures, and the writing of the text. [Link].

[11] Sylvia L. Herbert, Mo Chen, SooJean Han, Somil Bansal, Jaime F. Fisac, and Claire J. Tomlin. "FaSTrack: A Modular Framework for Fast and Guaranteed Safe Motion Planning". In: *IEEE 56th Annual Conference on Decision and Control*. Dec. 2017, pp. 1517–1522.
SJ Han participated in the design of the simulation experiments and the generation of the figures. [Link].

INTRODUCTION

1.1 Motivation

Current model-based methods of performing control or estimation for stochastic systems are able to handle a rather limited scope of distributions. One common assumption, in particular, is Gaussian white noise and many model-based controllers or observers in the literature have typically been designed for robustness to Gaussian white noise perturbations. The choice to use Gaussian white noise models is justifiable in practice; vision-based localization and mapping [161], spacecraft navigation [28], and motion-planning [75] are notable examples of applications where controllers and observers designed under the Gaussian white noise assumption perform well. Moreover, Gaussian white noise distributions are appealing to study because they demonstrate properties (e.g., independent increments, Central Limit Theorem, etc.) which make them convenient to analyze. As a result, there has been a wealth of literature studying Gaussian white noise systems, particularly in stability analysis, controller, and observer design. For example, a model-based controller can be developed via the well-known linear quadratic Gaussian (LQG) approach [17, 51], while a model-based observer can be designed via Kalman filtering [76] and its extensions [128, 151]. More recent methods of model-based control methods for Gaussian white noise systems include the path integral approach [145], convex optimization-based approaches [127, 109], as well as a number of reinforcement learning-based approaches [107, 45].

Control and estimation of stochastic systems that fall outside of the scope of model-based design are often left for model-free techniques in artificial intelligence (AI). For example, one especially prevalent distribution of non-Gaussian noise is impulsive shot noise, which arises in real-world applications almost as frequently as Gaussian white noise. Some examples are wind turbulence in agile quadrotor flight [112], meteorite collisions in spacecraft control [162], power outages in the power grid [144], spikes of brain activity in neuroscience [53], and large fluctuations of stock prices in financial analysis [115]. Fortunately, modern-day technological advancements and ever-increasing capabilities for sensing, computing, actuating, and communicating have made it possible to rely almost entirely on AI for controlling and forecasting complex stochastic systems. In fact, machine learning is already being used for a variety of

Figure 1.1: Examples of jump stochastic systems (JSSs) in the real-world: nonhomogeneous arrival processes in vehicle traffic networks, meteorite collisions in spacecraft control [162], and wind turbulence in agile quadrotor flight [112].

real-world systems with highly promising results. In the field of robotics, for example, meta-learning algorithms are applied to the problem of quadrotor flying and landing in random environments [137, 111] to perform adaptive control while learning characteristics of the random environment simultaneously. Other examples are the neural network architectures that are being used for control and prediction tasks, e.g., vehicle traffic forecasting using variations of the graph convolutional neural network [93, 40].

However, training AI methods can often be exhaustive in computation time and energy, especially if they redundantly learn information that could be more easily obtained from structured models. More efficient controllers/observers can be developed for stochastic systems by taking advantage of the theoretical properties of their sources of randomness. This concept is best demonstrated with Gaussian white noise systems, and can be extended to shot noise systems too. Just as how the basis for modeling various forms of Gaussian white noise is the standard Brownian motion process [114], shot noise is modeled using the Poisson processes [10, 81], especially the compound Poisson process. Both Brownian motion and Poisson processes are special cases of the more general Lévy processes [154, 8], which have useful properties like stationary and independent increments, and can be used to model noise with both small and large variations. Moreover, for many discrete-time/discrete-event sys-

tems, we can take advantage of the fact that the underlying stochastic process is a sequence of random variables which occurs as repeated "patterns of interest." It is thus clear that control/estimation of many stochastic systems can be made more efficient in data-consumption and time if a theoretical characterization of the underlying stochastic process can be made.

1.2 Related Work

The general need to balance between model-based and model-free methods for efficient design is not a new idea in the engineering community. Recent trends in research have seen an increasing number of proposed algorithms incorporating both model-based and model-free methods of control.

Controlling Uncertain Systems A related branch of work emerges from the field of robust and adaptive control, where the stochasticity in the dynamics (e.g., model parameters, external perturbations) is uncertain. They discuss a need to integrate system identification procedures (via data-driven design) into standard control techniques. [44] demonstrates this by addressing the linear quadratic regulator (LQR) problem with the goal of maintaining safety. More recently, there have also been works that consider robust online control such that the system does not need to be fully identified before it can be controlled; for example, [70] explores this problem for nonlinear systems with large model uncertainties. Controlling uncertain systems, especially with safety constraints, is an important problem to address in a wide array of applications such as spacecraft guidance and motion-planning [110], decision and control in connected automated vehicles [113], and agile quadrotor flight in the face of wind turbulence [112]. Many of these algorithms involve a multi-step procedure where the original uncertain dynamics and constraints are mapped down to an approximate model, which is then used for planning and control. For instance, [110] develops a surrogate optimization problem with chance constraints by leveraging polynomial chaos expansion, then applies sequential convex programming to generate approximate solution trajectories. This reinforces the natural control design approach to construct (approximate) models from the original dynamics before learning about its uncertainties by gathering data.

Meta-Learning In the field of AI, *meta-learning algorithms* refer to algorithms which not only focus on learning the subject matter (e.g., classification tasks), but also focus on learning the learning procedure itself. Recently, meta-learning approaches have demonstrated great potential to be used for learning-based control in robotics applications. For example, the method of [112] has an offline learning phase which learns the wind characteristics and

encodes them into a set of functions, i.e., by building a model; the adaptive control strategies are then designed online. This offline modeling of the wind characteristics enables the algorithm to be efficient in data-consumption, requiring "only a total of 12 min of flight data in just six different wind conditions to train." A branch of meta-learning that has started to become popular in machine learning literature is called *imitation learning*, which seeks to emulate behaviors while being data-efficient. One algorithm based on imitation learning is known as *programmatic reinforcement learning (PRL)* [149, 148], which was developed with the idea of committing learned policies to memory for future use instead of re-computing them for environments that have already been observed before. Both works [149] and [148] address efficient representation of neural policies into a set of PID controls that can be directly applied to the system in the future, effectively constructing a model-based controller.

Storing Past Information while Learning *Bayesian updating* is perhaps the most common and fundamental approach for learning, control, and state-estimation in engineering applications; there has been a wide variety of literature and standard texts in the last several decades on algorithms which use it (e.g., hidden Markov models, Bayesian neural networks [74], Kalman and particle filtering). In contrast to probabilistic schemes such as Bayesian updating, which encodes prior knowledge as a probability distribution, there has also been a class of methods that "hard-codes" data-driven policies into model-based format for update and reuse. Imitation learning, described previously, is not the only approach that does this; in problems that can be solved using deep reinforcement learning methods, the *experience replay* [56] mechanism manages to improve sample and data efficiency by storing the last few experiences into memory and "replaying" them. A related approach is called *episodic control* [91, 20, 124], which incorporates *episodic memory* [21] into traditional learning techniques with the goal of speeding up training by recalling specific instances of highly rewarding experiences. Although most experience replay or episodic memory methods deterministically maintain an explicit table of mappings from policy to reward, they are similar to Bayesian methods in that they both encode past information to avoid redundant computation in handling recurring data.

Predictions with Structure On the opposite end, rather than storing past trajectories into memory to reduce redundant data, algorithms can also invoke predictions to reduce redundant computation in the future. *Model predictive control (MPC)* is one of the most popular control algorithms that demonstrates this, and many variations of MPC for various different applications have been proposed in the literature [60, 41]. Both short-term and

long-term predictions for online control have been proven to be beneficial even in the face of either purely stochastic or adversarial disturbances [31]. In [164], this is demonstrated explicitly by applying greedy conventional MPC to the linear quadratic tracking problem, and proving near-optimality in the dynamic regret performance metric. The benefit of predictions is especially notable when there is spatial or temporal structure to the problem. *Graph neural networks (GNNs)* [15] are an example of a learning-based approach which encodes the topology of the graph for tasks such as graph classification and representation learning [82]. Recently, extensions of GNNs are also being used for congestion control problems in computer networks [132] and vehicle traffic forecasting [93, 40]. Many of these GNN extensions tend to include either a convolutional neural network or a recurrent neural network to capture spatial and temporal repetitions, respectively.

1.3 Jump Stochastic Systems

The class of *jump stochastic systems (JSSs)*, i.e., systems with random and repetitive jump phenomena, serve as an excellent example in demonstrating the model-based/model-free balance motivated in Section 1.1 because of its prevalence in both theory and application.

- Because Gaussian white noise is usually small in magnitude and continuous in the sense that changes occur over a measurable duration of time, it cannot be used to model jump phenomena. However, there exist tools and theories in mathematics specifically for various jump processes, including Poisson processes, renewal processes, and Markov jump processes.

- JSSs are interesting to study because they are highly prevalent in real-world applications from diverse fields. This includes large-scale networks (e.g., electricity outages in the power grid, vehicle arrival processes in traffic networks), biology (e.g., super-spreaders and the emergence of variant viruses in epidemic spread, neuronal spikes in brain signal imaging), and robotics (e.g., sudden wind gusts in agile quadrotor flight, meteorite collisions in spacecraft).

Suppose the stochastic process underlying the JSS is represented as a discrete-time stochastic process $\{Y_n\}_{n\in\mathbb{N}}$. Further let T_n be the time of the nth arrival, ξ_n be the magnitude of the nth arrival, and $A_n \triangleq T_n - T_{n-1}$ be the nth interarrival time. Then we can express $Y_n \triangleq \sum_{i=1}^{n} \xi_i$.

We can alternatively define a continuous-time stochastic process $\{Y(t), t \geq 0\}$ with discrete arrivals. The counting process, defined below, is the most basic process underpinning the

stochastic processes that drive the types of JSSs we are interested in throughout this thesis.

Definition 1 (Counting Process)**.** The process $Y(t)$ also has associated with it a *counting process* $\{N(t), t \geq 0\}$ which counts the number of arrivals by time t; formally, it is defined as $N(t) \triangleq \sup\{n \in \mathbb{N} : T_n \leq t\}$.

Similarly, we can represent $Y(t) \triangleq \sum_{i=1}^{N(t)} \xi_i$, and relate the continuous and discrete-time versions with $Y_n \equiv Y_{N(t)} = Y(t)$ when $N(t) = n$. Various types of jump processes emerge depending on how we characterize the distribution $\mathbb{P}(A_n \leq a, \xi_n = x_n | (\xi_{n-1} = x_{n-1}, T_{n-1}), \cdots, (\xi_0, T_0))$ for each $n \in \mathbb{N}$. We are interested in the following ones.

Definition 2 (Renewal Processes)**.** Y_n is a *renewal process* if A_n are i.i.d. and $\xi_n = 1$ for all $n \in \mathbb{N}$, and A_n and ξ_n are independent of each other. Y_n is a *renewal reward process* if A_n are i.i.d. and ξ_n are i.i.d., and A_n and ξ_n are independent of each other.

$$\mathbb{P}(A_n \leq a, \xi_n = x_n | (\xi_{n-1} = x_{n-1}, T_{n-1}), \cdots, (\xi_0, T_0)) = \mathbb{P}(A_n \leq a)\mathbb{P}(\xi_n = x_n).$$

Definition 3 (Poisson Processes)**.** Y_n is a *(standard) Poisson process* if $A_n \sim \text{Exp}(\lambda)$ and $\xi_n = 1$ for all $n \in \mathbb{N}$, and A_n and ξ_n are independent of each other. Y_n is a *compound Poisson process* if $A_n \sim \text{Exp}(\lambda)$ and ξ_n are i.i.d., and A_n and ξ_n are independent of each other.

Definition 4 (Markov (Jump) Process)**.** Y_n is a *Markov process* if $A_n \sim \text{Exp}(\lambda)$, ξ_n depends on ξ_{n-1}, and A_n and ξ_n are independent of each other. The distribution becomes

$$\mathbb{P}(A_n \leq a, \xi_n = x_n | (\xi_{n-1} = x_{n-1}, T_{n-1}), \cdots, (\xi_0, T_0)) = (1 - e^{-\lambda a})\mathbb{P}(\xi_n = x_n | \xi_{n-1} = x_{n-1}).$$

Y_n is a *Markov jump process* if ξ_n depends on ξ_{n-1} and A_n only depends on current state ξ_n.

$$\begin{aligned}\mathbb{P}(A_n \leq a, \xi_n = x_n | (\xi_{n-1} = x_{n-1}, T_{n-1}), \cdots, (\xi_0, T_0)) \\ = \mathbb{P}(A_n \leq a | \xi_n = x_n)\mathbb{P}(\xi_n = x_n | \xi_{n-1} = x_{n-1}).\end{aligned}$$

1.4 Contributions

The contributions of this thesis are two-fold. First, we theoretically characterize various jump processes to learn recurrent patterns in JSSs. Second, we use our characterizations to perform control and estimation with less redundant computation and data-consumption in various applications. Because of the properties motivated in Section 1.3, this thesis focuses exclusively on JSSs driven by the jump processes defined in Section 1.3. Despite this, the hope is that the contributions of this thesis can be used as a foundation for broadly designing efficient learning-based controllers for non-Gaussian stochastic systems, not necessarily within the class of JSSs.

1.4.1 Theory and Architecture

Two main examples of JSSs that are theoretically studied here are: 1) nonlinear systems externally perturbed by impulsive noise phenomena (e.g., Poisson shot noise), and 2) systems that are driven by random repeated behaviors of interest (e.g., switching phases in Markovian jump systems (MJSs)).

Stability Analysis for Non-Gaussian Stochastic Systems Our first theoretical contribution uses theory from Poisson random measures and Lévy processes to develop incremental stability criteria for nonlinear systems perturbed by two types of non-Gaussian noise: 1) shot noise processes represented as compound Poisson processes, and 2) finite-measure Lévy processes constructed as affine combinations of Gaussian white and compound Poisson shot noise processes (see Chapter 2). For each noise type, we compare trajectories of the perturbed system with distinct noise sample paths against trajectories of the nominal, unperturbed system. We show that for a finite number of jumps arising from the noise process, the mean-squared error between the trajectories exponentially converges toward a bounded error ball across a finite interval of time under practical boundedness assumptions. The impact of our work is that understanding the relationship between system stability and the characteristics of the jump noise allows us to infer the types of model-based controllers we need to design. We demonstrate our results using several simple and intuitive case studies: a 1D linear reference-tracking system, 2D linear time-varying systems, a 2D nonlinear system, and a 2D nonlinear observer design problem.

The Pattern-Occurrence Problem Our second theoretical contribution is the characterization of recurrent patterns in the jump process. We leverage results from *renewal theory* and *martingales* (some reference texts are [131, 130, 38, 39]) to develop the theoretical background necessary to address the following two problems pertaining to the occurrence and recurrence of patterns: 1) the *expected minimum occurrence time* of any pattern from some collection of patterns, and 2) the *first-occurrence probability* of a pattern being the first to occur among the collection at the expected occurrence time. For the purposes of real-world application, we also ensure these formulas operate on two key extensions of prior pattern-occurrence literature: 1) the statistics of the jump process are unknown (e.g., the transition matrix of the Markov jump process), and 2) the realization of the jump process over time is not observable.

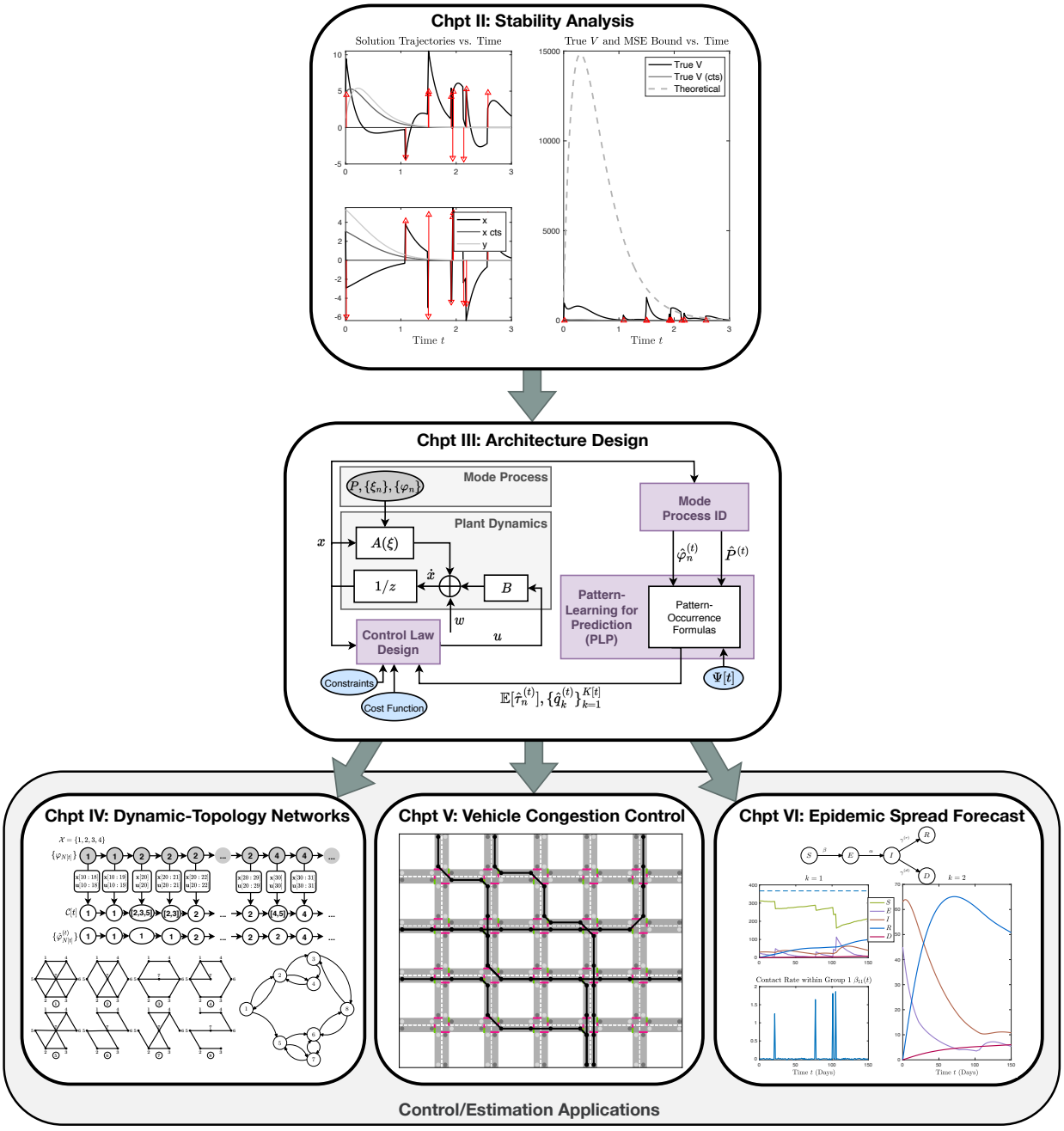

Figure 1.2: A flow diagram representing the overall structure of the thesis. Chapter 2 discusses stability analysis–an important prerequisite to control and estimation. A concrete controller architecture based on learning patterns is developed in Chapter 3. The following three chapters then present specific applications of control and estimation for JSSs: dynamic-topology network control in Chapter 4, vehicle traffic congestion control in Chapter 5, and epidemic spread forecasting in Chapter 6.

1.4.2 Applications in Control and Estimation

One specific architecture we design for controlling discrete-time/discrete-event JSSs implements *pattern-learning for prediction (PLP)*. PLP identifies recurrent patterns in the JSS's jump process, then leverages two key functions:

- memorization of past patterns to prevent redundant computation of control policies;
- prediction of future patterns to schedule control policies in advance.

The following describes the three main applications we consider throughout this thesis.

Dynamic-Topology Network Control In our first application contribution, we demonstrate a concrete implementation of PLP for linear discrete-time Markovian jump systems (MJSs) that switch among a finite number of modes. PLP is then applied to the control of networks with dynamic topology; the specific dynamics we use are those of a power grid with fault-tolerance to downed lines, modeled as a MJS whose modes represent the different topologies. The combinations of line failures are chosen to represent the "patterns" in the power grid system. The pattern learning component can be thought of as a mechanism that recognizes previously-occurred network topologies, and uses the relevant policy to control it. The control policies are designed using variations of the novel *system-level synthesis (SLS)* [153, 104, 5] framework, including a version which is robust to dynamic topologies (Chapter 4) and another which is data-driven [160, 3]. By comparing our architecture against a baseline controller and a topology-robust extension of the baseline (both without PLP), we showed that a controller with PLP is able to achieve three things: 1) match the control effort cost of the baseline, 2) stabilize the network just as well as the topology-robust extension, and 3) achieve runtime faster than both.

Vehicle Traffic Network Congestion Control Our second application contribution is the design of a controller architecture for vehicle traffic flow based on *pattern-learning with memory and prediction (PLMP)*. Here, PLMP is an extension of PLP with an explicit implementation of a memory component based on our version of episodic control [91, 20, 124], which builds equivalence classes to group together patterns that are controlled using the same light signals. The significance of PLMP is that it exploits the natural spatial and temporal symmetries prevalent in traffic networks to perform congestion control by scheduling light signals in advance without redundant computation. We apply our model to

two synthetic datasets, one synthesized from scratch and one synthesized from real-world data, and compare two periodic baseline light signals to various implementations of our PLMP controller, including a version without prediction called *pattern-learning with memory (PLM)*. We evaluate the performance of each implementation according to three different congestion metrics and find that, on average, PLM outperforms the periodic baselines while PLMP outperforms PLM with mild variation among the different implementations.

Epidemic Spread Forecasting in Population Networks For our third application contribution, we consider estimation and forecasting for the epidemic spread across population networks by using a multiscale combined compartmental ODE/ hidden Markov model approach. One significance of our model is that it takes advantage of two recurrent spatiotemporal patterns and combines extensions of two models that are traditionally used separately in the study of epidemics. Another significance of our work is the explicit modeling of jump phenomena that are unique to epidemic spreads, which are *superspreader effects* and the *emergence of variant viruses*. Virus superspreaders are considered through a stochastic setting, which incorporates Gaussian white and Poisson jump noise in the dynamics. The chronic emergence of new virus mutations can be represented as a jump to a set of dynamics with different parameter values, i.e., event-driven switching. Various different experiments are performed to demonstrate our model on two datasets, constructed based on real contact-tracing COVID-19 data from different countries.

STOCHASTIC INCREMENTAL STABILITY OF NONLINEAR LÉVY NOISE SYSTEMS

Many model-based designs for controllers or observers typically aim for robustness against Gaussian white noise. However, we cannot use Gaussian white noise to model sudden impulsive perturbations because it is small in magnitude and continuous in the sense that changes occur gradually over a measurable duration of time. Instead, one class of non-Gaussian noise which is more suitable for modeling impulsive perturbations is *shot noise* [10].

Shot noise is a type of jump phenomenon that is interesting to study in both theory and application. On one hand, shot noise phenomena in the real world are omnipresent: in the field of robotics, there is wind turbulence in agile quadrotor flight [112], meteorite collisions in spacecraft control [162], and in cyberphysical systems like the power grid, there are power outages [144]. Other examples of shot noise arise as spikes of brain activity in neuroscience [118] and large fluctuations of stock prices in financial analysis [115]. On the other hand, there is an abundance of theoretical literature (e.g., Applebaum 2009 [8], Øksendal 2007 [115], Mao 1991 [103], and Ikeda and Watanabe 1989 [71]) which provides useful tools to model the shot noise phenomenon such as *Poisson processes* (Definition 3). Both Brownian motion and Poisson processes are special cases of the more general *Lévy processes* [154, 8], and a particularly useful result is the *Lévy-Khintchine Theorem*, which describes Lévy processes as affine combinations of Brownian motion processes and compound Poisson processes [8].

Despite both the theoretical and practical appeal of shot noise phenomena, there are few model-based control or estimation methods designed for robustness to shot noise. In this chapter, we address an important prerequisite question to the problem of controller and observer design for JSSs: stability analysis and characterization. We specifically develop sufficient conditions for *incremental (globally exponential) stability* of nonlinear stochastic systems perturbed by *Poisson shot noise* and *Lévy noise*. We borrow, from the literature of applied mathematics, the theory of Poisson random measures and Lévy processes to lay out the foundations for studying these systems. We are specifically interested in addressing a standard stochastic stability question posed by [87]: can trajectories of the system, arising

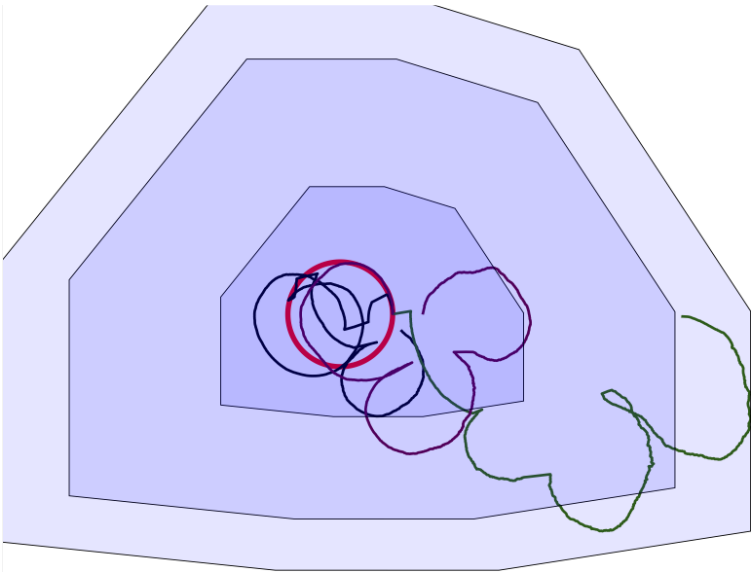

Figure 2.1: A simplified Dubins' car nonlinear system perturbed by different sample paths of Lévy noise for three different initial conditions starting randomly on the red circle. The car tries to track the reference path outlined by the red circle. The objective is to determine stability criteria about the system to find a bound on the deviation between the red circle and the system trajectories over some elapsed time.

from 1) different sample paths of noise and 2) different initial conditions within a bounded set, be bounded in some region after a sufficiently elapsed time? In Figure 2.1, several trajectories are depicted for the simplified Dubins' car (dynmaics in [119]) perturbed by Lévy noise for three different initial conditions starting randomly on the red circle.

Chapter Organization

We begin in Section 2.1 with a brief review of Poisson and Lévy processes and their theoretical properties. In Section 2.2, we set up the stochastic differential equations (SDEs) representing the Poisson shot and Lévy noise systems we consider throughout the chapter; this includes a discussion of Itô's formula for jump-diffusions and the existence and uniqueness of solutions. We provide a survey on both deterministic and stochastic Lyapunov-sense stability as well as the stronger notion of incremental stability in Section 2.3. Our main results are summarized in Section 2.4, where we review the stochastic contraction theorem for Gaussian white noise systems, and establish analogous results for Poisson shot and Lévy noise systems, with specialization to linear time-varying Poisson shot noise systems. Extensive numerical simulations of the bounds we derive in Section 2.4 are presented in Section 2.5.

Terminology and Mathematical Notation

For the sake of simplifying terminology, we will henceforth refer to additive Gaussian white, additive compound Poisson shot, and additive bounded-measure Lévy noise systems as simply white, shot, and Lévy noise systems, respectively. The understanding is that the shortened terminology throughout this chapter does not refer to the general cases (e.g., non-Gaussian white noise, or Lévy noise whose measures have infinite total mass). We use the ℓ_p norm $\|\cdot\| \triangleq \|\cdot\|_p$ for vectors and the corresponding induced norm $\|A\| \triangleq \sup_{\mathbf{x}\neq 0}(\|A\mathbf{x}\| \,/\, \|\mathbf{x}\|)$ for matrices. We abuse the notation $\|\cdot\|$ to apply to both matrices and vectors where the context is relevant. For any g which is a function of time, we denote the left-limit of $g(t)$ at time t as $g(t-) \triangleq \lim_{s\to t^-} g(s)$ for any $t > 0$. For any $g : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$, a function of both a scalar parameter $a \in \mathbb{R}$ and a vector $\mathbf{x} \in \mathbb{R}^n$, we write the partial derivatives in the following shorthand notation. First, $\partial_a g \equiv \partial_a g(a, \mathbf{x}) \triangleq \partial g(a, \mathbf{x})/\partial a$, and $\nabla_{\mathbf{x}} g \equiv \nabla_{\mathbf{x}} g(a, \mathbf{x}) \in \mathbb{R}^{m\times n}$ denotes the gradient with respect to $\mathbf{x}$. Moreover, we denote $\partial^2_{x_i x_j} \triangleq \partial^2 g(a, \mathbf{x})/(\partial x_i \partial x_j)$ as the double partial derivative of g with respect to two distinct components x_i and x_j of $\mathbf{x}$, $i \neq j$, and likewise $\partial^2_{x_i} \triangleq \partial^2 g(a, \mathbf{x})/(\partial^2 x_i)$ for the derivative with respect to the same component x_i. When describing a dynamical system of the form $\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t))$, where $\mathbf{x} : \mathbb{R}^+ \to \mathbb{R}^n$ is a function of time and $f : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$, we often simplify the notation by writing without the argument t, as in $\dot{\mathbf{x}} = f(t, \mathbf{x})$ with the understanding that f is not taking the function $\mathbf{x}$ as input, but rather the vector $\mathbf{x}(t)$. Furthermore, we define the dot notation as the time-derivative, meaning $\dot{\mathbf{x}}(t) \triangleq (d/dt)\mathbf{x}(t)$. For any function $H(t, \mathbf{x}(t))$ which takes in as argument time $t \in \mathbb{R}^+$ and state $\mathbf{x}(t) \in \mathbb{R}^n$ which evolves over time according to some dynamics of the form $\dot{\mathbf{x}} = f(t, \mathbf{x})$, has the time-derivative $\dot{H} \triangleq \partial_t H + \nabla_{\mathbf{x}} H \cdot \dot{\mathbf{x}}(t) = \partial_t H + \nabla_{\mathbf{x}} H \cdot f(t, \mathbf{x})$.

2.1 Poisson and Lévy Processes

The general *Poisson random measure*, typically denoted $N(dt, dy)$ over the space $[0, t] \times E$ with "jump space" E, is characterized by the intensity measure Leb$\times\nu$, where Leb denotes the standard Lebesgue measure in time and $\nu(dy)$ is the probability measure on E describing the distribution of jumps.

Definition 5 (Poisson Random Measure)**.** Let $E \subseteq \mathbb{N}^\ell$. We define random measure $N([0, t] \times E)$ on *jump space* E until some time $t > 0$ with *intensity measure* Leb$\times\nu$, where Leb denotes the standard Lebesgue measure (the measure in time) and ν is the probability measure on the jump space E (describing the distribution of the jumps). We denote the *intensity (parameter)* for the Poisson process corresponding to the intensity measure ν as λ and denote the time of the ith arrival with random variable T_i. One can think of λ as the average number of arrivals

over time. N is called a *Poisson random measure* if the following are satisfied:

1. if $E_1, \cdots, E_n$ are pairwise disjoint subsets of E, then $N([0,t] \times E_1), \cdots, N([0,t] \times E_n)$ are independent;
2. for each $E_i \subseteq E$, the random measure $N([0,t] \times E_i)$ has a corresponding Poisson process with intensity parameter $\lambda_i := (\nu(E_i)/\nu(E))\lambda$.

While most results in the theory of Poisson processes are written with respect to Poisson random measures, our scope in this chapter is specifically on the compound Poisson process. Below, we provide a more in-depth defintion of Poisson processes compared to Definition 3 by relating them to the Poisson random measure. Moreover, [81] does an excellent job of describing all the different properties of the Poisson process; we will organize just the few that are relevant to this thesis.

Definition 6 (Poisson Processes: Extended)**.** Let $E \subseteq \mathbb{N}^\ell$ and $t > 0$. The *standard Poisson process* $N(t)$ counts the number of jumps in E that have occurred in the time interval $[0,t]$ for $t \leq T$. It is characterized by an *intensity parameter* $\lambda > 0$, which essentially describes the rate at which jumps occur in the process. The *compound Poisson process* is a simple generalization of the standard Poisson process to weighted jumps, and can be expressed in terms of the sum $\sum_{i=1}^{N(t)} Y(T_i)$, where $T_i < t$ is the arrival time of the ith jump and $Y : [0,T] \to E$ is a function describing the jump distribution over the space E. The intensity $\lambda > 0$ of a compound Poisson process is the same as that of its corresponding standard Poisson process.

Definition 7 (Poisson Integral)**.** Let $\xi : \mathbb{R}^+ \times \mathbb{R}^n \times E \to \mathbb{R}$ be a predictable, bounded Borel-measurable function and N be a Poisson random measure on $[0,T] \times E$ with intensity measure Leb $\times\, \nu$. We define the *Poisson integral* of ξ as follows:

$$I_\xi := \int_{[0,T]\times E} \xi(s, \mathbf{x}(s), y) N(ds, dy), \tag{2.1}$$

where $\mathbf{x}(t)$ is the solution to (2.8).

Campbell's formula is a useful notion which allows us to determine the distribution of the sum $\sum_{y \in N} \xi(y)$ for some function ξ of points y in the Poisson point process N. The formula for time-invariant functions $\xi(y)$ is often presented in the literature, e.g., Section 3.2 of [81]

and Proposition 2.7 of [90]. The lemma below is a simple extension of the formula to functions which are both time-varying and dependent on the state $\mathbf{x}(t)$ of some stochastic differential equation, i.e., $\xi \equiv \xi(t, \mathbf{x}(t), y)$.

Lemma 1 (Campbell's Formula)**.** Let $\xi : \mathbb{R}^+ \times \mathbb{R}^n \times E \to \mathbb{R}$ be a predictable, locally-bounded Borel-measurable function and $N([0,t] \times E)$ denote the Poisson random measure with intensity λ over the jump space E. If ξ also satisfies the integrability condition

$$\int_{[0,t]\times E} |\xi(s, \mathbf{x}(s), y)| ds\nu(dy) < \infty$$

a.s., where we write "a.s." for "almost-surely," then

$$\mathbb{E}[I_\xi] = \int_{[0,t]\times E} \mathbb{E}[\xi(s, \mathbf{x}(s), y)] ds\nu(dy). \tag{2.2}$$

Another useful result similar to Campbell's formula is the mean squared integral case.

Lemma 2 (Square-Integral Formula)**.** Consider the setup of Definition 7 above. If the following properties hold:

$$\int_{[0,T]\times E} |\xi(t, \mathbf{x}(t), y)| dt\nu(dy) < \infty, \quad \int_{[0,T]\times E} \xi^2(t, \mathbf{x}(t), y) dt\nu(dy) < \infty, \tag{2.3}$$

then

$$\mathbb{E}[I_\xi^2] = \int_{[0,T]\times E} \mathbb{E}[\xi(t, \mathbf{x}(t), y)^2] dt\nu(dy) + \left(\int_{[0,T]\times E} \mathbb{E}[\xi(t, \mathbf{x}(t), y)] dt\nu(dy) \right)^2. \tag{2.4}$$

Remark 1. Instead of the general Poisson random measure, our focus in this chapter is on standard and compound Poisson processes. Thus, we will be using the simpler function $\xi : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}$ to describe the jumps of the process, which only takes time and the state $\mathbf{x}(t)$ of the SDE as input. Following the construction of (2.1):

$$\int_0^t \xi(s, \mathbf{x}(s)) dN(s) = \sum_{0<t\leq T} \xi(t, \Delta\mathbf{x}(t)) \mathbb{1}\{\Delta N(t) \neq 0\} = \sum_{i=1}^{N(t)} \xi(T_i, \Delta\mathbf{x}(T_i)), \tag{2.5}$$

where $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}(t-)$, $\Delta N(t) = N(t) - N(t-)$, and the indicator in the second line of the equation determines whether or not a jump occurred at time t. We denote this special case by $N(t)$ without the argument y, since we have essentially isolated the jump as the multiplicative factor $\xi(T_i, \mathbf{x}(T_i))$ for each i; all we need to represent is the time of

each arrival and the cumulative number of arrivals until time t. This representation has an intuitive interpretation: if we think of the noise process as a sequence of impulses where the ith impulse arrives at time T_i, then integrating a function ξ with respect to it over an interval of time $[0, t]$ would only pick out the values of ξ at $T_i \in [0, t]$.

For further properties of the standard and compound Poisson processes, we refer to the same above references on Poisson processes, as well as Kingman 1993 [81] or Baccelli and Blaszczyszyn 2009 [10].

We can generalize beyond the class of Poisson shot noise by comparing the definitions for a Brownian motion process and a Poisson process.

1. A real-valued process $\{W(t) : t \geq 0\}$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is said to be a Brownian motion if the following hold:
   - the paths of W are $\mathbb{P}$ almost-surely continuous;
   - $\mathbb{P}(W(0) = 0) = 1$;
   - for $0 \leq s \leq t$, $W(t) - W(s)$ is equal in distribution to $W(t - s)$;
   - for $0 \leq s \leq t$, $W(t) - W(s)$ is independent of $W(r)$ for $r \leq s$;
   - for $t > 0$, $W(t)$ is equal in distribution to a normal random variable with variance t.
2. A real-valued process $\{N(t) : t \geq 0\}$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is said to be a Poisson process with intensity $\lambda > 0$ if the following hold:
   - the paths of N are $\mathbb{P}$ right-continuous with left-limits;
   - $\mathbb{P}(N(0) = 0) = 1$;
   - for $0 \leq s \leq t$, $N(t) - N(s)$ is equal in distribution to $N(t - s)$;
   - for $0 \leq s \leq t$, $N(t) - N(s)$ is independent of $N(r)$ for $r \leq s$;
   - for $t > 0$, $N(t)$ is equal in distribution to a Poisson random variable with parameter λt.

The similarity between the two definitions motivates the definition of a general Lévy process.

Definition 8 (Lévy Processes [154, 73])**.** A process $\{L(t) : t \geq 0\}$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is said to be a *Lévy process* if the following hold:

- Cádlág Paths: the paths of L are almost-surely $\mathbb{P}$ right-continuous with left-limits.
- Zero Initial Condition: $\mathbb{P}(L(0) = 0) = 1$
- Stationary Increments: for $0 \leq s \leq t$, $L(t) - L(s)$ is equal in distribution to $L(t - s)$.
- Independent Increments: for $0 \leq s \leq t$, $L(t) - L(s)$ is independent of $L(r)$ for $r \leq s$.

Under this definition, we can observe that both Gaussian white noise processes and compound Poisson shot noise processes are both Lévy processes. This implies that the affine combination of the two is also a Lévy process. In fact, a well-known result called the *Lévy-Khintchine Theorem*, stated formally in Theorem 1.6 of Watson 2016 [154], Theorem 2.7 of Bass 2009 [14], or Theorem 1.2.14 of Applebaum 2009 [9], says that Lévy processes can be represented as affine combinations of Brownian motion processes and Poisson processes. This even includes Lévy processes with intensity measures that have unbounded jumps. One example of this is a Gamma process, which has intensity measure on $\mathbb{R}^+$ given by $\nu(dy) = ay^{-1}e^{-by}dy$ such that on any finite interval of time, the number of jumps lying in the interval $(0, 1)$ is infinite. However, we emphasize that we only consider *bounded-measure Lévy processes* in this chapter, excluding Lévy processes like the Gamma process, since they rarely occur in the practical control and engineering applications of our target scope.

A particularly useful theoretical result known as the Lévy-Khintchine Decomposition Theorem, stated formally below, leads us to a direct extension of Poisson shot noise by combining it with Gaussian white noise. We closely follow the version of the formula stated formally in Theorem 1.6 of [154] or Theorem 2.7 of [14].

Theorem 1 (Lévy-Khintchine Formula)**.** Let L be a Lévy process with characteristic exponent Ψ. Then there exist (unique) $a \in \mathbb{R}, \sigma \geq 0$ and a measure ν satisfying $\int_{\mathbb{R}} 1 \wedge x^2 \nu(dx) < \infty$ such that

$$\Psi(\theta) = ia\theta - \frac{1}{2}\sigma^2\theta^2 + \int_{\mathbb{R}} \left(e^{i\theta x} - 1\right) \nu(dx) - \int_{\mathbb{R}} i\theta x \mathbb{1}_{[-1,1]}(x)\nu(dx). \tag{2.6}$$

Conversely, given any triplet (a, σ, ν), there exists a Lévy process L with characteristic exponent given by (2.6).

a is called the *center* of L and captures the deterministic drift component, σ is the *Gaussian coefficient* and captures the variance of the Brownian motion component, and the Lévy measure ν captures the size and intensity of the "large" jumps of L.

2.2 Shot and Lévy Noise Systems

We begin this chapter by setting up the specific form of stochastic differential equation (SDE) we consider for analysis throughout this chapter. Namely, we consider systems that can be expressed as SDEs of the following form:

$$d\mathbf{x}(t) = f(t,\mathbf{x})dt + \sigma(t,\mathbf{x})dW(t) + \xi(t,\mathbf{x})dN(t), \tag{2.7}$$

where

- $f : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$ is a deterministic function in $\mathcal{C}^{(1,2)}$, i.e., f is once-differentiable in time and twice-differentiable in state;
- $\sigma(t,\mathbf{x})dW(t)$ is the additive Gaussian white noise of the system, where $\sigma : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^{n\times d}$, $\sigma \in \mathcal{C}^{(1,2)}$ is the variation of the white noise, and $W : \mathbb{R}^+ \to \mathbb{R}^d$ is a d-dimensional standard Brownian motion process;
- $\xi(t,\mathbf{x})dN(t)$ is a compound Poisson process which enters into the system as an additive disturbance, where $\xi : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$, $\xi \in \mathcal{C}^{(1,2)}$ describes the jumps that occur, and $N(t)$ is the scalar standard Poisson process with intensity λ. The "derivative" of the standard Poisson process, written as $dN(t)$, is understood as a function which takes value 1 if a jump has occurred at time t, and value 0 otherwise.

Assumption 1 (Bounded Noise Magnitudes)**.** For the system (2.7), there exist constants $\gamma, \eta > 0$ such that

$$\sup_{t,\mathbf{x}} \|\sigma(t,\mathbf{x})\| \leq \gamma, \quad \sup_{t,\mathbf{x}} \|\xi(t,\mathbf{x})\| \leq \eta,$$

where the norms used are described in the terminology and mathematical notation defined at the beginning of this chapter.

Note that if $\sigma(t,\mathbf{x}) \equiv 0$, we have the following *shot-noise SDE*

$$d\mathbf{x}(t) = f(t,\mathbf{x})dt + \xi(t,\mathbf{x})dN(t), \tag{2.8}$$

and if $\xi(t, \mathbf{x}) \equiv 0$, we recover the *white-noise SDE*:

$$d\mathbf{x}(t) = f(t, \mathbf{x})dt + \sigma(t, \mathbf{x})dW(t). \tag{2.9}$$

While there may exist forms of white and shot noise processes that are more general than what is used in (2.7), we focus our scope to the specific case of the Gaussian white noise process and the compound Poisson process. This is because their applicability to most practical problems in stochastic control and robotic engineering is more widespread than any other type of white and shot noise.

Since much of the conventional stochastic system dynamics have been concentrated around white noise injection, the form of (2.9) may be familiar to the reader (see, e.g., [114]). However, there has not been as much attention given to the shot noise case (2.8). Thus, the remainder of this chapter is devoted to the review of relevant background material necessary to understand the notation associated with the shot noise term, and the nature of solutions to the combined SDE (2.7). As shown in Section 2.4, these are especially important for the stability analysis of systems governed by dynamics of the form (2.8) or (2.7).

2.2.1 Itô's Formula

The version of the *Itô's Formula* for functions of stochastic processes which are driven by more general Poisson random measures are described in standard stochastic references such as Chapter 4, Section 4.4 of Applebaum 2009 [8] and Chapter 6 of Ikeda and Watanabe 1989 [71]. Itô's formula for functions of scalar-valued stochastic processes is standard; see, for example, Theorem 32 in Chapter 2 of Protter 1992 [125] or Theorem 3.7 of Bass 2009 [14] for the formula. An extension to functions which depend on time is discussed in Chapter 1 of Jeanblanc 2007 [73], and an extension to multi-dimensional stochastic processes is presented in Chapter II, Section 5 of Ikeda and Watanabe 1989 [71]. For the purposes of this chapter, we present a version of the formula for systems (2.7) in the lemma below.

Lemma 3 (Itô's Formula)**.** For functions $G \in \mathcal{C}^{(1,2)}$,

$$\begin{aligned} G(t, \mathbf{x}(t)) = G(0, \mathbf{x}_0) &+ \int_{0+}^{t} \partial_s G(s, \mathbf{x}(s-))ds + \sum_{i=1}^{n} \int_{0+}^{t} \partial_{x_i} G(s, \mathbf{x}(s-))dx_i^c(s) \\ &+ \frac{1}{2} \sum_{i,j=1}^{n} \int_{0+}^{t} \partial_{x_i}\partial_{x_j} G(s, \mathbf{x}(s-))d[x_i, x_j]^c(s) + \sum_{i=1}^{N(t)} \left(G(T_i, \mathbf{x}(T_i)) - G(T_i, \mathbf{x}(T_i-))\right), \end{aligned} \tag{2.10}$$

where integrals from $0+$ to t indicate an integral over the interval $(0, t]$, $\mathbf{x} \in \mathbb{R}^n$, $0 < T_i \leq t$ is the time of the ith arrival of $N(t)$, and we use the left-limit notation introduced in

the beginning of this chapter. Furthermore, dx_i^c represents the continuous part of the SDE for component x_i, and $d[x_i, x_j]^c$ represents the continuous part of the quadratic variation between the two SDEs corresponding to components x_i and x_j of $\mathbf{x}$. For example, for (2.7), $dx_i^c(t) = f_i(t, \mathbf{x})dt + \sigma_i(t, \mathbf{x})dW(t)$, and $d[x_i, x_j]^c(t) = (\sigma\sigma^T)_{ij}(t)$.

Remark 2. The sum over s, for $0 < s \leq t$, in the last term of (2.10) can be thought of as the contribution of $G(t, \mathbf{x}(t))$ coming from the jump-discontinuities of the stochastic process $\mathbf{x}(t)$ within the interval $(0, t]$. Note that the terms of the sum are nonzero only when $\lim_{r \to s^-} G(r, \mathbf{x}(r)) \neq G(s, \mathbf{x}(s))$, and since $G \in \mathcal{C}^{(1,2)}$, this only occurs when there is a jump of $\mathbf{x}$ at time s. By the form of the SDE (2.7), the jumps of $\mathbf{x}$ arise from the jumps of its shot noise part, specifically the standard Poisson process $N(t)$. Hence, the last term of (2.10) can alternatively be expressed as follows:

$$\sum_{0<s\leq t} \left(G(s, \mathbf{x}(s)) - G(s-, \mathbf{x}(s-))\right) = \sum_{i=1}^{N(t)} \left(G(T_i, \mathbf{x}(T_i)) - G(T_i-, \mathbf{x}(T_i-))\right),$$

where $0 < T_i \leq t$ is the time of the ith arrival of $N(t)$. We refer to Chapter 1, Section 4 of Protter 1992 [125], Section 1.3.4 of Jeanblanc 2007 [73] for further details on this notation.

A comprehensive treatment of the scalar version of (2.10) can be found in many references (see e.g., Theorem 32 of [125] or Theorem 3.7 of [14]). The specific version of the formula stated is also used in [86, 9]. The term $d[\mathbf{x}_i, \mathbf{x}_j]^c(s)$ in (2.10) is the continuous part of the quadratic variation between two stochastic processes $\mathbf{x}_i$ and $\mathbf{x}_j$, which is defined in the following remark.

Remark 3 (Quadratic Variation)**.** Consider two generic scalar SDEs of the form (2.7):

$$dx_i = f_i(t, \mathbf{x})dt + \sigma_i(t, \mathbf{x})dW(t) + \xi_i(t, \mathbf{x})dN(t), \tag{2.11a}$$

for $i = 1, 2$, and where f_i, σ_i, ξ_i denote the ith row of each respective function. Then the *quadratic variation* term $d[x_1, x_2](t)$ is computed to be $\sigma_1(t, \mathbf{x})\sigma_2(t, \mathbf{x})dt + \xi_1(t, \mathbf{x})\xi_2(t, \mathbf{x})dN(t)$ since $dW(t) \cdot dW(t) = dt$ and $dN(t) \cdot dN(t) = dN(t)$ while the dot products between every other pair of terms vanishes. For the case of (2.7), the quadratic variation is comprised of two parts: the continuous part $d[x_1, x_2]^c(t) = dt$ and the purely discontinuous part $d[x_1, x_2]^d(t) = dN(t)$. For further information about this notation, one may refer to [114, 86] and references therein.

Example 1. One can obtain intuition about (2.10) by considering the following specific example. Let $\mathbf{x} \equiv x \in \mathbb{R}$ be the state of the scalar process $dx(t) = \mu dt + dN(t)$, and let $G(t,x) = tx^2$. Consider using (2.10) to evaluate $G(t,x)$ until the time of the first jump $T_1 = t_1 > 0$ of $N(t)$. Then note that

$$x(t) = \begin{cases} \mu t & \text{if } t < t_1 \\ \mu t + 1 & \text{if } t = t_1 \end{cases}. \tag{2.12}$$

On the left side of (2.10), we have

$$G(t_1, x(t_1)) = t_1 x^2(t_1) = t_1(\mu t_1 + 1)^2, \tag{2.13}$$

and on the right side of (2.10), we get:

$$\begin{aligned} 0 + \int_0^{t_1} x^2(s-)ds + \int_0^{t_1} 2sx(s-) \cdot \mu ds + 0 + \sum_{s \le t_1} \left(sx^2(s) - sx^2(s-)\right) \\ = \int_0^{t_1} x(s-)\left(x(s-) + 2\mu(s)\right) ds + t_1(2\mu t_1 + 1) \\ = \mu^2 t_1^3 + t_1(2\mu t_1 + 1), \end{aligned} \tag{2.14}$$

which is indeed equivalent to the left side (2.13).

Definition 9 (Infinitesimal Generator)**.** For $G \in \mathcal{C}^{(1,2)}$, the *infinitesimal generator* is defined to be

$$\mathcal{L}G = \lim_{t \to 0} \frac{\mathbb{E}_{\mathbf{x}_0}\left[G(t, \mathbf{x}(t))\right] - G(0, \mathbf{x}_0)}{t},$$

where $\mathbf{x}(t)$ is the trajectory of a given SDE which starts with initial condition $\mathbf{x}(0) \triangleq \mathbf{x}_0$.

This definition extends the standard definition of the infinitesimal generator (see Definition 1.21 in Øksendal 2009 [115] or (4-12) in Kushner 1967 [87] for instance) to functions G which are also dependent on time t.

Remark 4. From the formula of the infinitesimal generator, it is easy to see its close relationship with Itô's formula (Lemma 3) because it can be used to compute the $\mathbb{E}_{\mathbf{x}_0}\left[G(t, \mathbf{x}(t))\right]$ term in Definition 9. For the case where the SDE mentioned in Definition 9 is a scalar version of the white noise system (2.9) (i.e., $\mathbf{x}(t) \triangleq x(t) \in \mathbb{R}$), the generator is given by $\mathcal{L}G = \partial_t G(t, x(t)) + \partial_x G(t, x(t)) f(t, x(t)) + (1/2)\partial_x^2 G(t, x(t))\sigma^2(t, x(t))$. The generator for a version of the Lévy noise system (2.7) which models the shot noise component using more general Poisson random measures, is shown in Theorem 1.22 of Øksendal 2009 [115].

2.2.2 Existence and Uniqueness of Solutions

The conditions for existence and uniqueness of solutions for white noise systems (2.9) have been studied in a widely in literature (e.g., [114], Chapter 4 of [83]), and they are standard.

1. f and σ must be Lipschitz with respect to the time argument: there exists $C > 0$ such that $\|f(t, \mathbf{x}) - f(t, \mathbf{y})\| + \|\sigma(t, \mathbf{x}) - \sigma(t, \mathbf{y})\|_F \leq C \|\mathbf{x} - \mathbf{y}\|$ for all $t \geq 0$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$;
2. f and σ must have bounded growth with respect to the state argument: there exists $K > 0$ such that $\|f(t, \mathbf{x})\|^2 + \|\sigma(t, \mathbf{x})\|_F^2 \leq K(1 + \|\mathbf{x}\|^2)$ for all $t \geq 0$, $\mathbf{x} \in \mathbb{R}^n$.

Similar conditions can be derived in the case of shot noise systems (2.8) and the combined Lévy noise system (2.7). The Lévy noise case was presented in [115], and also in [8] for various cases of $\mathbb{E}[\mathbf{x}_0]$. Because we are specializing to the case of the standard Poisson process instead of considering the Poisson integral with respect to the general Poisson random measure, the proof from [8] can be simplified, and we do so below.

First, we present the well-known Gronwall inequality, a standard result of which can be found in any classical control-theoretic textbook (e.g., [80, 138]).

Lemma 4 (Gronwall inequality)**.** Let $I = [t_1, t_2] \subset \mathbb{R}$ and $\phi, \psi, \rho : I \to \mathbb{R}^+$ be continuous, nonnegative functions.

If the following inequality holds true:

$$\phi(t) \leq \psi(t) + \int_{t_1}^{t} \rho(s)\phi(s)ds \;\; \forall\, t \in [t_1, t_2], \tag{2.15}$$

then it follows that

$$\phi(t) \leq \psi(t) + \int_{t_1}^{t} \psi(s)\rho(s)e^{\int_s^t \rho(\tau)d\tau}ds. \tag{2.16}$$

For (2.7), we impose some additional Lipschitz and bounded-growth conditions so that we have the existence and uniqueness of solutions. We adapt a combination of conditions from Theorem 6.2.3 of Applebaum 2009 [8] and Section IV.3 of Ikeda and Watanabe 1989 [71] specifically to our case.

Theorem 2 (Existence and Uniqueness for SDE (2.8))**.** For fixed $T > 0$, let $f : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ and $\xi : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ be measurable functions satisfying the following conditions:

1. Lipschitz: $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, t \in [0,T]$, $\exists K > 0$

$$\|f(t,\mathbf{x}) - f(t,\mathbf{y})\| + \|\sigma(t,\mathbf{x}) - \sigma(t,\mathbf{y})\|_{\mathrm{F}} + \|\xi(t,\mathbf{x}) - \xi(t,\mathbf{y})\| \leq K \|\mathbf{x}(t) - \mathbf{y}(t)\|, \tag{2.17}$$

2. Bounded-growth: $\forall \mathbf{x} \in \mathbb{R}^n, t \in [0,T]$, $\exists C > 0$

$$\|f(t,\mathbf{x})\|^2 + \|\sigma(t,\mathbf{x})\|_{\mathrm{F}}^2 + \|\xi(t,\mathbf{x})\|^2 \leq C(1 + \|\mathbf{x}(t)\|^2), \tag{2.18}$$

for positive constants C and K where the norm on ξ is the Frobenius norm and the norms on the vector-valued functions are any vector norm. Further, let $\mathbf{x}_0 \in \mathbb{R}^n$ have $\mathbb{E}[\|\mathbf{x}_0\|] < \infty$ and be independent of the noise processes. Then the SDE (2.8) with initial condition $\mathbf{x}(0) = \mathbf{x}_0$ has a unique solution $\mathbf{x}(t)$ adapted to the filtration $\mathcal{F}_t$ generated by $\mathbf{x}_0$ and $N(s)$, where $s \leq t$ and

$$\mathbb{E}\left[\int_0^T \|\mathbf{x}(t)\|^2 \, dt\right] < \infty$$

where the definition of "uniqueness" that we use is as follows: for $\mathbf{x}, \mathbf{y}$ two solutions of (2.8) with the same initial conditions, then

$$\mathbb{P}(\mathbf{x}(t) = \mathbf{y}(t), \ \forall t > 0) = 1.$$

Proof. First we construct an approximate sequence using Picard iterations, recursively defined as

$$\mathbf{z}^{(n)}(t) = \mathbf{z}_0^{(n)} + \int_0^t f(s, \mathbf{z}^{(n-1)}(s))ds + \int_0^t \xi(s, \mathbf{z}^{(n-1)}(s))dN(s), \tag{2.19}$$

where $n \in \mathbb{N}$.

Taking the difference between two trajectories $\mathbf{z}^{(n)}(t)$ and $\mathbf{z}^{(m)}(t)$ results in

$$\mathbf{z}^{(n,m)}(t) = \mathbf{z}_0^{(n,m)} + \int_0^t f^{(n-1,m-1)}(s)ds + \int_0^t \xi^{(n-1,m-1)}(s)dN(s), \tag{2.20}$$

with $n, m \in \mathbb{N}$ and the notation

$$\mathbf{z}^{(n,m)}(t) := \mathbf{z}^{(n)}(t) - \mathbf{z}^{(m)}(t), \ \mathbf{z}^{(n,m)}(0) := \mathbf{z}^{(n,m)}(0)$$
$$f^{(n,m)}(t) := f(t, \mathbf{z}^{(n)}(t)) - f(t, \mathbf{z}^{(m)}(t))$$

$$\xi^{(n,m)}(t) := \xi(t, \mathbf{z}^{(n)}(t)) - \xi(t, \mathbf{z}^{(m)}(t)).$$

Taking the mean-squared difference, and applying the triangle and Cauchy-Schwarz inequalities leads to

$$\mathbb{E}\left[\left\|\mathbf{z}^{(n,m)}(t)\right\|^2\right] \leq \mathbb{E}\left[\left(\left\|\mathbf{z}_0^{(n,m)}\right\| + \int_0^t \left\|f^{(n-1,m-1)}(s)\right\| ds + \int_0^t \left\|\xi^{(n-1,m-1)}(s)\right\| dN(s)\right)^2\right] \leq$$
$$3\mathbb{E}\left[\left\|\mathbf{z}_0^{(n,m)}\right\|^2\right] + 3\mathbb{E}\left[\int_0^t ds \int_0^t \left\|f^{(n-1,m-1)}(s)\right\|^2 ds\right] + 3\mathbb{E}\left[\left(\int_0^t \left\|\xi^{(n-1,m-1)}(s)\right\| dN(s)\right)^2\right]. \tag{2.21}$$

Note that the Lipschitz bound (2.17) can be squared on both sides:

$$\begin{aligned}&\|f(t,\mathbf{x}) - f(t,\mathbf{y})\|^2 + \|\xi(t,\mathbf{x}) - \xi(t,\mathbf{y})\|^2 \\ &\quad + 2\|f(t,\mathbf{x}) - f(t,\mathbf{y})\| \|\xi(t,\mathbf{x}) - \xi(t,\mathbf{y})\| \leq K^2 \|\mathbf{x} - \mathbf{y}\|^2.\end{aligned} \tag{2.22}$$

Because norms are nonnegative and the integral of nonnegative functions (whether it is standard ds or Poisson $dN(s)$) is also nonnegative, the bound also holds for each individual term in the left-hand side sum. Using (2.22) on the second expectation term yields:

$$\mathbb{E}\left[\int_0^t ds \int_0^t \left\|f^{(n-1,m-1)}(s)\right\|^2 ds\right] \leq K^2 t \int_0^t \mathbb{E}\left[\left\|\mathbf{z}^{(n-1,m-1)}(s)\right\|^2\right] ds, \tag{2.23}$$

and for the final term, we can apply Remark 1 and extend Lemma 2 to vector-valued functions:

$$\begin{aligned}&\mathbb{E}\left[\left(\int_0^t \left\|\xi^{(n-1,m-1)}(s)\right\| dN(s)\right)^2\right] \\ &\quad = \lambda \int_0^t \mathbb{E}\left[\left\|\xi^{(n-1,m-1)}(s)\right\|^2\right] ds + \mathbb{E}\left[\left(\lambda \int_0^t \left\|\xi^{(n-1,m-1)}(s)\right\| ds\right)^2\right],\end{aligned} \tag{2.24}$$

using the fact that $\lambda := \int_{\{1\}^\ell} \nu(dy)$, as in Remark 1. By Cauchy-Schwarz inequality and the squared Lipschitz bound (2.22):

$$\begin{aligned}(2.24) &\leq K^2\lambda \int_0^t \mathbb{E}\left[\left\|\mathbf{z}^{(n-1,m-1)}(s)\right\|^2\right] ds + \lambda^2 \mathbb{E}\left[\int_0^t ds \int_0^t \left\|\xi^{(n-1,m-1)}(s)\right\|^2 ds\right] \\ &\leq K^2\lambda \int_0^t \mathbb{E}\left[\left\|\mathbf{z}^{(n-1,m-1)}(s)\right\|^2\right] ds + K^2\lambda^2 t \int_0^t \mathbb{E}\left[\left\|\mathbf{z}^{(n-1,m-1)}(s)\right\|^2\right] ds.\end{aligned} \tag{2.25}$$

Finally, note that $\mathbf{z}_0^{(n,m)} = 0$ because both trajectories $\mathbf{z}^{(n)}$ and $\mathbf{z}^{(m)}$ begin with the same initial conditions. In combination, we get:

$$\mathbb{E}\left[\left\|\mathbf{z}^{(n)}(t) - \mathbf{z}^{(m)}(t)\right\|^2\right] \leq 3K^2(t + \lambda + \lambda^2 t)\int_0^t \mathbb{E}\left[\left\|\mathbf{z}^{(n-1)}(s) - \mathbf{z}^{(m-1)}(s)\right\|^2\right] ds. \tag{2.26}$$

Choose $n = k+1, m = k$ for $k > 0$. By induction, we get:

$$\mathbb{E}\left[\left\|\mathbf{z}^{(k+1)}(t) - \mathbf{z}^{(k)}(t)\right\|^2\right] \leq \frac{c^k t^{k+1}}{(k+1)!} \quad \forall k \geq 0, t \in [0, T], \tag{2.27}$$

where $c := 3K^2(T + \lambda + \lambda^2 T)$. From there, it is straightforward to show that $\left\{\mathbf{z}^{(k)}(t)\right\}$ is a Cauchy sequence which converges to a limit since $\mathbf{z} \in \mathbb{R}^n$.

To show that the solution is unique, consider two solution trajectories $\mathbf{x}(t, \omega)$ and $\mathbf{y}(t, \omega)$ of (2.8) with respective initial conditions $\mathbf{x}_0$ and $\mathbf{y}_0$ where ω is a specific sample path of the noise process N. We can apply the same calculations as before on the mean-squared error difference between $\mathbf{x}$ and $\mathbf{y}$ to get

$$\mathbb{E}\left[\|\mathbf{x}(t) - \mathbf{y}(t)\|^2\right] \leq 3\mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{y}_0\|^2\right] + c\mathbb{E}\left[\int_0^t \|\mathbf{x}(s) - \mathbf{y}(s)\|^2 ds\right]. \tag{2.28}$$

By Gronwall's inequality (Lemma 4), (2.28) becomes

$$\mathbb{E}\left[\|\mathbf{x}(t) - \mathbf{y}(t)\|^2\right] \leq 3\mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{y}_0\|^2\right] e^{ct}. \tag{2.29}$$

Now we set the two initial conditions $\mathbf{x}_0$ and $\mathbf{y}_0$ equal to each other. This implies that $c_1 = 0$ and so $h(t) = 0$ for all $t \geq 0$. Thus,

$$\mathbb{P}\left(\|\mathbf{x} - \mathbf{y}\| = 0\right) = 1 \quad \text{for all } t \geq 0.$$

This holds for all sample paths of N. Thus, the solution is indeed unique for all $t \in [0, T]$. The proof is complete. ■

Remark 5 (Alternative Conditions)**.** There have also been previous works done on describing existence and uniqueness conditions for solutions to SDEs of the form (2.8) while imposing different, non-Lipschitz conditions on f and ξ. For instance, [96] relaxes the Lipschitz conditions by instead assuming that f and ξ are bounded above by any concave function of the normed difference in trajectories $\|\mathbf{x} - \mathbf{y}\|$. Alternatively, [77] presents a result for conditions

where f is upper-bounded in norm by a constant and the bound on ξ depends on the size of the jump (which is not easily applicable to our case because we are only considering standard Poisson process noise, i.e., the jump size is always one). We choose to work with simple Lipschitz conditions because it is easier to relate to the well-known white noise version.

2.3 Common Notions of Stability and Additional Background

In this section, we provide a brief review of previous incremental stability results for deterministic and white noise systems (2.9). Our main results to be presented in the following Section 2.4, the *Shot Noise Stochastic Contraction Theorem* and the *Lévy Noise Stochastic Contraction Theorem*, derive incremental stability conditions for the shot noise system (2.8) and Lévy noise system (2.7), respectively. For simplicity, we henceforth refer to each theorem as the *Shot Contraction Theorem* and the *Lévy Contraction Theorem*, respectively.

2.3.1 Deterministic Stability

Characterizations of deterministic stability are traditionally defined in the *Lyapunov sense*. For both autonomous nonlinear systems $d\mathbf{x} = f(\mathbf{x})dt$ and non-autonomous nonlinear systems $d\mathbf{x} = f(t, \mathbf{x})dt$, stability in the sense of Lyapunov is either characterized with respect to an equilibrium point $\mathbf{x}^*$ (see Section 1.2.2 of Sastry 1999 [135]) or a limit cycle $\mathcal{O} := \{\mathbf{x}(t) | \mathbf{x}(t) = \mathbf{x}(t+T)\}$, where $T > 0$ (see Definition 2.13 of Sastry 1999 [135]). The most common types of deterministic stability are defined as follows with respect to an equilibrium point. First, *asymptotic stability* occurs when there exists $r > 0$ such that for all $\|\mathbf{x}(t) - \mathbf{x}^*\| < r$, we have $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$. The stronger *exponential stability* occurs when there exists $\kappa, \alpha > 0$ such that $\|\mathbf{x}(t) - \mathbf{x}^*\| \leq \kappa \|\mathbf{x}_0 - \mathbf{x}^*\|^{\alpha(t-t_0)}$ for all $t \geq t_0$. More formal details regarding these standard types of stability can be found in well-known control theory references such as Khalil 2002 [80].

For many systems, determining the type of stability can be difficult to do using the conditions prescribed by their literal definitions. Instead, it is common to invoke either the *Indirect Lyapunov method* (Theorem 4.7 of Khalil 2002 [80]) or the *Direct Lyapunov methods* (Section 4.4 of Khalil 2002 [80]). Another approach to determine stability is via the use of *Krasovskii's Theorem* (Theorem 3.7 in Slotine and Li 1991 [138]), and we will see in Section 2.4.1 that contraction theory is a generalization of this theorem for determining incremental stability of a system. Many of these approaches require the construction of a differentiable, real-valued, nonnegative *Lyapunov function* $V(\mathbf{x})$, which essentially captures the potential energy of the system, and determines stability type depending on how quickly its value descreases along the

solution trajectories of the system. We refer to seminal works such as Krasovskii 1870 [85], Sontag 1983 [140], or LaSalle and Lefschetz 1973 [89] for further reading on deterministic stability.

2.3.2 Deterministic Incremental Stability

Incremental stability generalizes Lyapunov-sense stability by considering the convergence of solution trajectories toward a desired time-varying trajectory rather than an equilibrium point or a limit cycle. Furthermore, systems that satisfy the incremental stability property have guaranteed global exponential convergence towards the desired trajectory. There has been an extensive amount of work characterizing incremental stability for deterministic nonlinear dynamics [98, 97]. Applications of incremental stability arise in numerous settings such as cooperative control over multi-agent swarm systems [33] and phase synchronization in directed networks [139, 32]. A recent tutorial paper on incremental stability and connections to machine learning is Tsukamoto et. al. 2021 [147].

Definition 10 (Incremental Exponential Stability)**.** For any nonlinear function $f \in \mathcal{C}^{(1,2)}$, the deterministic system $d\mathbf{x} = f(t, \mathbf{x})dt$ is said to be *incrementally (globally exponentially) stable* if there exist constants $\kappa, \alpha > 0$ such that

$$\|\mathbf{x}_2(t) - \mathbf{x}_1(t)\| \leq \kappa \|\mathbf{x}_{2,0} - \mathbf{x}_{1,0}\| e^{-\alpha t} \tag{2.30}$$

for any vector norm $\|\cdot\|$, and all $t \geq 0$. The trajectories $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ are solutions of the system $d\mathbf{x} = f(t, \mathbf{x})dt$ with respective initial conditions $\mathbf{x}_{1,0}$ and $\mathbf{x}_{2,0} \neq \mathbf{x}_{1,0}$.

Following the notation from Definition 10, we denote $\delta\mathbf{x} \in \mathbb{R}^n$ to be the *infinitesimal displacement length* between $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ over a fixed infinitesimal interval of time. Formally, the infinitesimal displacement length is represented as a path integral:

$$\|\mathbf{x}_2(t) - \mathbf{x}_1(t)\| \leq \int_{\mathbf{x}_1}^{\mathbf{x}_2} \|\delta\mathbf{x}(t)\| \,. \tag{2.31}$$

The evolution of the infinitesimal displacement over time can be approximated by the dynamics

$$d\delta\mathbf{x} = F\delta\mathbf{x}dt, \tag{2.32}$$

where $F \triangleq \nabla_{\mathbf{x}} f(t, \mathbf{x})$ is the *Jacobian* of the system. These dynamics, associated with the state $\delta\mathbf{x}$, are commonly referred to as the *virtual dynamics*.

Similar to the indirect and direct Lyapunov methods of testing Lyapunov stability, there is a test to determine incremental stability of a system without needing to directly apply Definition 10. Oftentimes, performing a differential coordinate transform from $\delta\mathbf{x}$ to $\delta\mathbf{z} \triangleq \Theta(t,\mathbf{x})\delta\mathbf{x}$, where $\Theta(t,\mathbf{x}) \in \mathbb{R}^{n\times n}$ is a smooth invertible square matrix, makes it easier to verify the conditions of this test. The new virtual dynamics under this coordinate transform become

$$d\delta\mathbf{z} = F_g\delta\mathbf{z}dt, \tag{2.33}$$

where

$$F_g \triangleq (\dot{\Theta}(t,\mathbf{x}) + \Theta(t,\mathbf{x})F)\Theta^{-1}(t,\mathbf{x}) \tag{2.34}$$

is the *generalized Jacobian* of the system, and the dot notation is defined in the beginning of this chapter.

An equivalent way to say that a system $d\mathbf{x}(t) = f(t,\mathbf{x})dt$ is incrementally stable in the sense of Definition 10 is to say that it is *contracting* with rate $\alpha > 0$. Similar to (2.33), we can extend the notion of contraction to more general metrics. We say that the system is contracting with respect to a uniformly positive definite metric $M(t,\mathbf{x}) \triangleq \Theta(t,\mathbf{x})^\top\Theta(t,\mathbf{x})$ and convergence rate α. For most practical applications, we are able to make the following assumption on $M(t,\mathbf{x})$.

Assumption 2 (Bounded Metric)**.** The metric $M(t,\mathbf{x})$ described in the setup above is bounded in both arguments $\mathbf{x}$ and t from above and below, and its first and second derivatives with respect to the $\mathbf{x}$ argument are also bounded from above. We thus define the following constants

$$\underline{m} = \inf_{t,\mathbf{x}} \lambda_{\min}(M(t,\mathbf{x})), \qquad \overline{m} = \sup_{t,\mathbf{x}} \lambda_{\max}(M(t,\mathbf{x})) \tag{2.35}$$
$$m' = \sup_{t,\mathbf{x},i,j} \left\|(\partial_x M(t,\mathbf{x}))_{i,j}\right\|, \qquad m'' = \sup_{t,\mathbf{x},i,j} \left\|(\partial_x^2 M(t,\mathbf{x}))_{i,j}\right\|.$$

The inequality (2.30) is obtained for the special case where $M(t,\mathbf{x}) = I_n$, the n-dimensional identity matrix. For general deterministic system dynamics $d\mathbf{x}(t) = f(t,\mathbf{x})dt$, the criterion for testing incremental stability is stated in the theorem below.

Theorem 3 (Basic Contraction)**.** Consider the deterministic dynamics $d\mathbf{x} = f(t,\mathbf{x})dt$. If there exists a uniformly positive definite metric $M(t,\mathbf{x})$ and $\alpha > 0$ such that the following

condition is satisfied:

$$F^\top M(t,\mathbf{x}) + M(t,\mathbf{x})F + \dot{M}(t,\mathbf{x}) \leq -2\alpha M(t,\mathbf{x}), \tag{2.36}$$

then the system is contracting. In relation to (2.34), we have $\lambda_{\max}((F_g + F_g^\top)/2) \leq -\alpha$.

2.3.3 Stochastic Stability

Lyapunov-sense stability characterizations have also been derived for stochastic nonlinear systems. The seminal work of Kushner 1967 [87] laid out the foundations of Lyapunov-based stochastic stability theory for systems perturbed by noise processes that are right-continuous, strong Markov processes. The Lévy noise processes of our consideration belong in this class. This section informally presents some of these foundations in order to set the context for stochastic incremental stability to be discussed in the following section.

For a given stochastic system of the form (2.7), suppose we design a corresponding Lyapunov-like function $V(t,\mathbf{x})$ such that

$$\mathcal{L}V(t,\mathbf{x}(t)) \leq -\beta V(t,\mathbf{x}(t)), \tag{2.37}$$

for some $\beta > 0$, initial condition $\mathbf{x}(0) = \mathbf{x}_0$, and $\mathcal{L}$ as the infinitesimal generator defined in Definition 9. Rearranging this inequality can give us

$$\mathbb{E}[V(t,\mathbf{x}(t)] \leq V(0,\mathbf{x}_0)e^{-\beta t}, \tag{2.38}$$

which implies that, under the assumption that $\mathbb{E}[V(t,\mathbf{x}(t)] < \infty$ for all $t > 0$, $V(t,\mathbf{x}(t))$ is a *supermartingale* since $e^{-\beta t} \leq 1$ for all $t \geq 0$. Using Doob's supermartingale inequality (stated originally in Chapter VII.3 of Doob 1953 [49], also in Chapter 1.7 of Kushner 1967 [87]), (2.38) implies:

$$0 \leq \mathbb{P}\left(\sup_{s\leq t<\infty} V(t,\mathbf{x}(t)) \geq c\right) \leq \frac{1}{c}\mathbb{E}[V(s,\mathbf{x}(s))] \leq \frac{1}{c}V(0,\mathbf{x}_0)e^{-\beta s}$$

for $s < t$ and any constant $c > 0$. This further implies *almost-sure stability* of the system since the right side of the inequality tends to 0 as $s \to \infty$, and hence the probability that the "energy" of the system will ever exceed any constant $c > 0$ tends to zero.

However, unlike deterministic systems, it is often the case for stochastic systems that convergence to an equilibrium does not occur exactly with zero error; at best, the solution trajectories are guaranteed to remain within some bounded-error ball of the equilibrium

point. This is especially true for the large jump disturbances that arise for the shot and Lévy noise processes considered in this chapter. There have been approaches to characterizing stability for stochastic systems perturbed by non-Gaussian noise: asymptotic stability of systems driven by Lévy noise is developed in Applebaum 2009 [9] while exponential stability is studied in Mao 1990 [102]. The alternative approach we take in this work is through the use of contraction theory, which allows us to achieve *incremental stability* [6], which generalizes the mentioned Lyapunov-sense stability to account for convergence among multiple different solution trajectories of the system, rather than toward a single equilibrium point or a fixed limit cycle. We show in Section 2.4 that the specific form of the error bound enters into the right side of (2.38) additively, and derive their specific expressions by considering the mean-squared error between solution trajectories with distinct initial conditions and noise sample paths.

2.3.4 Stochastic Incremental Stability

For deterministic systems, incremental stability has been established as a concept of convergence between different solution trajectories with different initial conditions [99, 4]. However, in the stochastic setting, the difference between trajectories also arises from using different noise processes. To this end, we require a change in notation from deterministic incremental stability analysis. The infinitesimal displacement length now considers the difference between a solution trajectory $\mathbf{x}(t)$ of a stochastic system with one noise sample path and a solution trajectory $\mathbf{y}(t)$ of a stochastic system with another noise sample path. This can be viewed as a comparison between solution trajectories coming from distinct systems, which is different from the comparison of two solution trajectories from the same system done for deterministic incremental stability. To make this distinction clear, we use the notation $\delta\mathbf{q}$ in place of $\delta\mathbf{x}$, and the path integral (2.31) is now written instead as

$$\|\mathbf{y}(t) - \mathbf{x}(t)\| \leq \int_{\mathbf{x}(t)}^{\mathbf{y}(t)} \|\delta\mathbf{q}\| = \int_0^1 \|\partial_\mu \mathbf{q}(t)\|\, d\mu. \tag{2.39}$$

In particular, for the Lévy noise system (2.7) we consider a parametrization of a new state $\mathbf{q}(\mu, t) \in \mathbb{R}^n$, with $\mu \in [0, 1]$, such that:

$$\begin{aligned}
&\mathbf{q}(\mu = 0, t) = \mathbf{x}(t),\ \mathbf{q}(\mu = 1, t) = \mathbf{y}(t) && (2.40)\\
&\sigma_{\mu=0}(t) = \sigma_1(t, \mathbf{x}),\ \sigma_{\mu=1}(t) = \sigma_2(t, \mathbf{y}), \quad \xi_{\mu=0}(t) = \xi_1(t, \mathbf{x}),\ \xi_{\mu=1}(t) = \xi_2(t, \mathbf{y})\\
&W_{\mu=0}(t) = W_1(t),\ W_{\mu=1}(t) = W_2(t), \qquad N_{\mu=0}(t) = N_1(t),\ N_{\mu=1}(t) = N_2(t),
\end{aligned}$$

where $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are solution trajectories of, respectively:

$$\begin{aligned} d\mathbf{x}(t) &= f(t,\mathbf{x})dt + \sigma_1(t,\mathbf{x})dW_1(t) + \xi_1(t,\mathbf{x})dN_1(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \\ d\mathbf{y}(t) &= f(t,\mathbf{y})dt + \sigma_2(t,\mathbf{y})dW_2(t) + \xi_2(t,\mathbf{y})dN_2(t), \quad \mathbf{y}(0) = \mathbf{y}_0, \end{aligned}$$

e.g., $\mathbf{q}(\mu,t) \triangleq \mu\mathbf{x}(t) + (1-\mu)\mathbf{y}(t)$. This parametrization allows us to construct a *virtual system* with state $\mathbf{q}(\mu,t)$, written as

$$d\mathbf{q}(\mu,t) = f(t,\mathbf{q}(\mu,t))dt + \sigma_\mu(t,\mathbf{q}(\mu,t))dW_\mu(t) + \xi_\mu(t,\mathbf{q}(\mu,t))dN_\mu(t). \tag{2.41}$$

The virtual dynamics become

$$d\delta\mathbf{q}(t) = F\delta\mathbf{q}(t)dt + \delta\sigma_\mu dW_\mu(t) + \delta\xi_\mu dN_\mu(t), \tag{2.42}$$

where F is the Jacobian defined in (2.32) and

$$\begin{aligned} \delta\sigma_\mu(t,\mathbf{q}) &\triangleq [\nabla_{\mathbf{q}}\sigma_{\mu,1}(t,\mathbf{q})\delta\mathbf{q}(t), \cdots, \nabla_{\mathbf{q}}\sigma_{\mu,d}(t,\mathbf{q})\delta\mathbf{q}(t)] \in \mathbb{R}^{n\times d} \\ \delta\xi_\mu(t,\mathbf{q}) &\triangleq \nabla_{\mathbf{q}}\xi_\mu(t,\mathbf{q})\delta\mathbf{q}(t) \in \mathbb{R}^n, \end{aligned} \tag{2.43}$$

where $\sigma_{\mu,i}$ is the ith column of σ_μ.

Assumption 3 (Initial Conditions)**.** The initial conditions adhere to some probability distribution $p(\mathbf{q}_0) = p(\mathbf{x}_0,\mathbf{y}_0)$, where p is either compactly-supported on the set of initial conditions $\mathcal{X}_0 \subset \mathbb{R}^n$, or is a distribution with finite second moment on $\mathcal{X}_0$ such that

$$\|\mathbf{x}_0 - \mathbf{y}_0\| \leq c_0, \quad \forall \mathbf{x}_0, \mathbf{y}_0 \in \mathcal{X}_0. \tag{2.44}$$

The work of Pham 2009 [121] considered stochastic incremental stability for the specific case of additive Gaussian white noise perturbations, and Dani 2015 [42] extended this theory to more general state-dependent metrics. Stochastic contraction is defined in Definition 2 of Pham 2009 [121], but is only applicable to white noise systems (2.9). For the purposes of including shot noise, we create a more general definition of stochastic contraction.

Definition 11 (Stochastically Contracting)**.** The system (2.7) is said to be *stochastically contracting* if:

1. the nominal, unperturbed system $d\mathbf{x} = f(t,\mathbf{x})dt$ is contracting with some differential coordinate transform $\Theta(t,\mathbf{x})$ and convergence rate α, i.e., (2.36) is satisfied;

2. there exists a Lyapunov-like function $V(t, \mathbf{x})$ such that

$$\mathbb{E}[V(t, \mathbf{q}(t))] \leq V(0, \mathbf{q}_0)e^{-\beta t} + \kappa(t), \tag{2.45}$$

for some contraction rate $\beta > 0$ and bounded function $\kappa : \mathbb{R}^+ \to \mathbb{R}^+$, and initial condition $\mathbf{q}_0 \in \mathbb{R}^n$.

Separately, white noise SDE (2.9) is stochastically contracting under the same two conditions above without η and ξ in condition 2, while shot noise SDE (2.8) removes mention of γ and σ.

Remark 6. The equation (2.45) is a version of (2.38) with a nonzero steady-state error bound $\kappa(t)$ added. This is because it is often the case for stochastic systems that convergence to an equilibrium does not occur with zero error due to considering trajectories arising from different noise sample paths. Moreover, for the impulsive shot noise in (2.7) and (2.8), almost-sure convergence is nearly impossible to demonstrate. We show this in Section 2.4, where the bound is derived probabilistically, with a condition on having a finite number of jumps within the interval of time $[0, t]$.

Demonstrating stochastic incremental stability for stochastic systems perturbed by some class of noise processes involves rewriting (2.45) and deriving specific forms of β and $\kappa(t)$ based on the parameters of the stochastic system. One common choice of Lyapunov function is the metric-weighted norm-squared difference between solution trajectories with distinct initial conditions and noise sample paths:

$$V(t, \mathbf{q}(\mu, t), \delta\mathbf{q}(t)) = \int_0^1 \partial_\mu \mathbf{q}^\top(\mu, t) M(t, \mathbf{q}(\mu, t)) \partial_\mu \mathbf{q}(\mu, t) d\mu, \tag{2.46}$$

where $M(t, \mathbf{q}) \triangleq \Theta^\top \Theta(t, \mathbf{q})$ is the metric defined as before, and the parametrization over μ is such that $V(t, \mathbf{q}_0, \delta\mathbf{q}_0) = \|\mathbf{y}_0 - \mathbf{x}_0\|$.

Similar to the direct method of Lyapunov, we analyze the behavior of the system by analyzing the derivative of the Lyapunov-like function $V(t, \mathbf{q}, \delta\mathbf{q})$ along trajectories of the virtual system (2.41). This requires us to use the infinitesimal generator from Definition 9, which can be thought of as the stochastic analogue to the differentiation operator used in the deterministic case. For the shot and Lévy noise systems, we invoke Itô's formula (Lemma 3) directly instead of using the infinitesimal generator by virtue of the relationship described in Remark 4.

2.3.5 Other Relevant Inequalities and Results

The *Comparison Lemma* (Khalil 2002[80], p.102-103) can be roughly stated as follows: Suppose we have an initial-value problem of the form $\{\dot{u} = g(u,t), u(0) = u_0\}$ and corresponding solution $u(t)$. Then if we were to consider an analogous problem $\{D^+v \leq g(v,t), v(0) \leq u_0\}$, the solution $v(t)$ satisfies $v(t) \leq u(t)$ for all $t \geq 0$, where D^+ denotes the right side limit defined as

$$D^+v(t) \triangleq \limsup_{\Delta t \to 0^+} \frac{v(t+\Delta t) - v(t)}{\Delta t}. \tag{2.47}$$

Note that this definition implies that if there exists a function $g(\Delta t, t)$ such that

$$\frac{|v(t+\Delta t) - v(t)|}{\Delta t} \leq g(\Delta t, t), \quad \forall \Delta t \in (0, a] \tag{2.48}$$

for some $a > 0$, and $g_0(t) \triangleq \lim_{\Delta t \to 0+} g(\Delta t, t)$ exists, then $D^+v(t) \leq g_0(t)$.

We use the Comparison Lemma to derive the following result, which is used in the proofs of the stochastic contraction results of Section 2.4.

Lemma 5. Let $y : \mathbb{R}^+ \to \mathbb{R}^{\geq 0}$ be a piecewise-continuous, nonnegative function which is right-continuous with left limits. Let $\mu > 0$ be a positive constant, and $\theta : \mathbb{R}^+ \to \mathbb{R}$ be a nonconstant, continuously-differentiable function with $\theta(0) = \zeta > 0$. Fix a value of time $T > 0$, and suppose that there are a finite number $k \in \mathbb{N}$ jump-discontinuities of y in the interval $[0, T)$ which occur at times t_i given by $0 < t_1 \leq t_2 \leq \cdots \leq t_k < T$. Further suppose that there exists a continuous, nonnegative function $h(t)$ which bounds the jumps, i.e., $y(t_i) - y(t_i-) \leq h(t_i)$. Given this setup, if the following inequality holds:

$$y(t) - y(0) \leq \theta(t) - \mu \int_{0+}^{t} y(s-)ds, \quad \forall t \in [0, T). \tag{2.49}$$

Then

$$y(t) \leq \int_{0+}^{t} \frac{d\theta(s)}{ds} e^{-\mu(t-s)} ds + \zeta e^{-\mu t} + y(0)e^{-\mu t}, \quad \forall t \in [0, T), \tag{2.50}$$

where the integral from $0+$ to t denotes integration over the interval $(0, t]$.

Proof. Suppose that $u : \mathbb{R}^+ \to \mathbb{R}$ is the continuously-differentiable state of the dynamics (2.49) with equality instead of inequality, and set $u(0) = y(0)$. Then we can directly solve for $u(t)$ by using an integrating factor of $e^{\mu t}$:

$$u(t) - y(0) = \theta(t) - \mu \int_0^t u(s)ds$$

$$\implies \frac{d}{dt}\left(u(t)e^{\mu t}\right) = \frac{d\theta(t)}{dt}e^{\mu t}$$
$$\implies u(t) = \int_{0+}^{t} \frac{d\theta(s)}{ds}e^{-\mu(t-s)}ds + (y(0)+\zeta)e^{-\mu t}. \tag{2.51}$$

Note that by the bound on the magnitude of the jumps at each time t_i, $D^+y(t) \leq h(t)$ for all $t \in [0,T)$, which means that (2.48) is satisfied with $g(\Delta t, t) \equiv g_0(t) \triangleq h(t)$. Hence, we can apply the Comparison Lemma to conclude that $y(t) \leq u(t)$ for all $t \in [0,T)$, yielding the desired result (2.50). ■

Lemma 6 (Maximum of Nonnegative Functions)**.** Suppose $\{\alpha_i\}_{i=1}^k$ and $\{\beta_i\}_{i=1}^k$ are finite, nonnegative, real-valued sequences. Then

$$\max_{i=1,\cdots,k} \alpha_i\beta_i \leq \max_{i=1,\cdots,k} \alpha_i \max_{i=1,\cdots,k} \beta_i. \tag{2.52}$$

Proof. Suppose the maximum value of the sequences $\{\alpha_i\}_{i=1}^k$ and $\{\beta_i\}_{i=1}^k$ occur at $i_a, i_b \in \{1,\cdots,k\}$, respectively. Further suppose that the maximum value of the product sequence $\{\alpha_i\beta_i\}_{i=1}^k$ occurs at index $j \in \{1,\cdots,k\}$. Then

$$\max_{i,\cdots,k} \alpha_i\beta_i = \alpha_j\beta_j \leq \alpha_{i_a}\beta_{i_b} = \max_{i=1,\cdots,k} \alpha_i \max_{i=1,\cdots,k} \beta_i$$

since $\alpha_j \leq \alpha_{i_a}$ and $\beta_j \leq \beta_{i_b}$. This concludes the proof. ■

2.4 Stochastic Contraction Theorems for Incremental Stability

2.4.1 For Gaussian White Noise Systems

Both Pham 2009 [121] and Dani 2015 [42] specified (2.45) for white noise systems (2.9); Pham 2009 investigated systems for which a state-independent metric $M(t,\mathbf{x}) \equiv M(t)$ can be used, while Dani 2015 used the original metric $M(t,\mathbf{x})$ with motivation toward nonlinear observer design. Both works compared between two noise-perturbed trajectories: $\mathbf{x}(t)$, solution to (2.9) with noise term $\sigma_1(t,\mathbf{x})dW_1(t)$, and $\mathbf{y}(t)$, solution to (2.9) with $\sigma_2(t,\mathbf{x})dW_2(t)$. However, we emphasize that we compare one noise-perturbed trajectory $\mathbf{x}(t)$ against a trajectory $\mathbf{y}(t)$ of the deterministic system $d\mathbf{y} = f(t,\mathbf{y})dt$ in order to easily combine the white noise result with the shot noise result (to be presented in Section 2.4.2) for the purposes of the Lévy noise result in Section 2.4.4. That is, we define the measure μ such that:

$$\mathbf{q}(\mu=0,t) = \mathbf{x}(t),\ \mathbf{q}(\mu=1,t) = \mathbf{y}(t), \qquad \sigma_{\mu=0}(t,\mathbf{q}) = \sigma(t,\mathbf{x}),\ \sigma_{\mu=1}(t,\mathbf{q}) = 0, \tag{2.53}$$

where $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are solution trajectories of, respectively, $d\mathbf{x}(t) = f(t,\mathbf{x})dt + \sigma(t,\mathbf{x})dW(t)$ with initial condition $\mathbf{x}(0) = \mathbf{x}_0$, and $d\mathbf{y}(t) = f(t,\mathbf{y})dt$ with initial condition $\mathbf{y}(0) = \mathbf{y}_0$. The virtual system corresponding to (2.53) becomes:

$$d\mathbf{q}(\mu,t) = f(t,\mathbf{q}(\mu,t))dt + \sigma_\mu(t,\mathbf{q}(\mu,t))dW(t),$$

while the virtual dynamics become

$$d\delta\mathbf{q}(t) = F\delta\mathbf{q}(t)dt + \delta\sigma_\mu dW(t).$$

Remark 7. Another significant distinction between our stochastic incremental stability results and previous versions is that we derive an error bound over a fixed interval of time $[s,t]$ for any $s < t$ instead of fixing $s = 0$ and including the initial state. This allows us to interpret our stability theorems as a measure of how far the perturbed trajectory will deviate away from the nominal within a local horizon of time, which has practical applicability in online, adaptive design of controllers and observers.

Theorem 4 (White Noise Stochastic Contraction Theorem)**.** Suppose that (2.9) is stochastically contracting in the sense of Definition 11 under a differential coordinate transform $\Theta(t,\mathbf{x})$ which satisfies Assumption 2. We consider a parametrization $\mu \in [0,1]$ such that the virtual system and virtual dynamics are established as in (2.41) and (2.42) without the shot noise terms $\xi_\mu(t,\mathbf{q})$, $\delta\xi\mu$, and $dN(t)$. Then, for a fixed interval of time $[s,t]$ for $s < t$, (2.45) can be rewritten as:

$$\mathbb{E}[\|\mathbf{y}-\mathbf{x}\|^2] \leq \frac{1}{\underline{m}}\mathbb{E}\left[\|\mathbf{y}(s)-\mathbf{x}(s)\|^2\right] e^{-\beta_w(t-s)} + \frac{\kappa_w(\beta_w,s,t)}{\underline{m}\beta_w}, \tag{2.54}$$

where

$$\beta_w = 2\alpha - \frac{\gamma^2}{\underline{m}}\left(m' + \frac{m''}{2}\right), \tag{2.55a}$$

$$\kappa_w(\beta_w,s,t) = \gamma^2(m'+\overline{m})\left(1 - e^{-\beta_w(t-s)}\right). \tag{2.55b}$$

Here, γ is the bound defined in Assumption 1, $\underline{m}, \overline{m}, m', m''$ are the constants defined in Assumption 2, and α is the deterministic contraction rate from Theorem 3.

Dani 2015 [42] demonstrates an application of white noise incremental stability to the problem of model-based nonlinear observer design. Thus, extending Theorem 4 to account for non-Gaussian noise gives us a potential way to design model-based observers and controllers

for systems perturbed by non-Gaussian noise. With this motive, the next section presents our main results, the Shot Contraction Theorem and the Lévy Contraction Theorem, which are incremental stability theorems for the shot noise system (2.8) and Lévy noise system (2.7), respectively.

2.4.2 For Compound Poisson Shot Noise Systems

We begin with the Shot Contraction Theorem for shot noise systems (2.8). Similar to the parametrization from (2.53), we compare a trajectory $\mathbf{x}(t)$ of the shot noise system (2.8) against a trajectory $\mathbf{y}(t)$ of the nominal system $d\mathbf{y}(t) = f(t, \mathbf{y})dt, \mathbf{y}(0) = \mathbf{y}_0$. We define the parameter $\mu \in [0, 1]$ such that

$$\mathbf{q}(\mu = 0, t) = \mathbf{x}(t),\ \mathbf{q}(\mu = 1, t) = \mathbf{y}(t), \qquad \xi_{\mu=0}(t, \mathbf{q}) = \xi(t, \mathbf{x}),\ \xi_{\mu=1}(t, \mathbf{q}) = 0. \tag{2.56}$$

The virtual system for (2.56) is given by

$$d\mathbf{q}(\mu, t) = f(t, \mathbf{q}(\mu, t))dt + \xi_\mu(t, \mathbf{q}(\mu, t))dN(t), \tag{2.57}$$

and the virtual dynamics are

$$d\delta\mathbf{q}(t) = F\delta\mathbf{q}(t)dt + \delta\xi_\mu dN(t), \tag{2.58}$$

with the Jacobian F from (2.32).

Assumption 4. Consider the shot noise system (2.8), virtual system (2.57), virtual dynamics (2.58), and corresponding Lyapunov-like function (2.46). For any fixed $t > 0$, there exists a deterministic, locally-bounded, continuously-differentiable function $h : \mathbb{R}^+ \to \mathbb{R}^+$ such that

$$\mathbb{E}\left[\sup_{r\in[0,t]} V(r, \mathbf{q}, \delta\mathbf{q}) - V(r-, \mathbf{q}, \delta\mathbf{q})\right] \leq h(t). \tag{2.59}$$

For the remainder of this section, the expectation operator $\mathbb{E}[\cdot]$ is understood to be taken over all sources of randomness in the argument. For $\mathbb{E}\left[V(t, \mathbf{q}, \delta\mathbf{q})\right]$ from Assumption 4, $\mathbb{E}[\cdot]$ is taken with respect to the initial conditions $p(\mathbf{q}_0)$, the random function ξ describing the jump distribution, and the standard Poisson process $N(t)$.

Remark 8. The reason for introducing an abstract function $h(t)$ in Assumption 4 is because the Lyapunov-like function $V(t, \mathbf{q}, \delta\mathbf{q})$ takes in arguments $\mathbf{q}$ and $\delta\mathbf{q}$, which depend on the shot noise process of (2.8). Thus, more information is needed about $f(t, \mathbf{x})$ and $\xi(t, \mathbf{x})$ in order

to conclude a bound for nonlinear systems of the form (2.8), and determine a more specific function $h(t)$. In Section 2.4.3, we specialize (2.8) to linear time-varying (LTV) systems where $f(t, \mathbf{x}) \triangleq A(t)\mathbf{x}$ and jumps $\xi(t, \mathbf{x}) \equiv \xi(t)$ which are independent of the state $\mathbf{x}$. We will see that the expression of $h(t)$ in terms of system parameters relies on knowing the exact form of the solution trajectory $\mathbf{x}(t)$ of (2.8), which is easy to obtain for LTV systems.

Remark 9. The existence of a $h(t)$ in (2.59) is roughly justified using the following argument. Note that we can simplify the difference in (2.59) as follows:

$$\begin{aligned} &V(t, \mathbf{q}, \delta\mathbf{q}) - V(t-, \mathbf{q}, \delta\mathbf{q}) \\ &\qquad = \int_0^1 \left[\partial_\mu \mathbf{q}(\mu, t)^\top M(t, \mathbf{q}(\mu, t))\partial_\mu \mathbf{q}(\mu, t) - \partial_\mu \mathbf{q}(\mu, t-)^\top M(t, \mathbf{q}(\mu, t))\partial_\mu \mathbf{q}(\mu, t-)\right] d\mu \\ &\qquad \leq \int_0^1 \left(\overline{m}\, \|\partial_\mu \mathbf{q}(\mu, t)\|^2 - \underline{m}\, \|\partial_\mu \mathbf{q}(\mu, t-)\|^2\right) d\mu, \end{aligned} \tag{2.60}$$

where the last inequality follows from Assumption 2. By continuity of the nominal system, $\mathbf{y}(t) = \mathbf{y}(t-)$. Hence, the difference (2.60) is nonzero only when there exists a jump at time t. But we know that the jumps of the SDE (2.8) are assumed to be bounded by a constant η by Assumption 1. Therefore, we can assume that (2.60) is also bounded at each fixed time t.

Definition 12 (Condition on the Number of Jumps)**.** Let $N(t)$ be the standard Poisson process driving the shot noise process behind systems of the form (2.8) or (2.7). For fixed values of time $0 \leq s \leq t$, define $\mathbb{E}_k[\cdot] \triangleq \mathbb{E}[\cdot | N(t) - N(s) = k]$ to be the expectation operator $\mathbb{E}[\cdot]$ conditioned on the occurrence of k jumps within the fixed interval of time $[s, t]$. By the stationarity property of Poisson processes, the event is equivalent to the event that $N(t - s) = k$, which occurs with probability

$$p_k(t - s) \triangleq \mathbb{P}(N(t) - N(s) = k) = e^{-\lambda(t-s)} \frac{(\lambda(t - s))^k}{k!}, \tag{2.61}$$

and we have the relationship $\mathbb{E}[\cdot] = \sum_{k=0}^{\infty} p_k \mathbb{E}_k[\cdot]$.

Theorem 5 (Shot Noise Stochastic Contraction Theorem)**.** Consider the shot noise system (2.8) which has trajectories that begin from different initial conditions and are perturbed by noise processes which satisfy Assumption 1. Let $N(t)$ be the standard Poisson process in (2.8). Take as convention $T_0 \triangleq s$, where $s \geq 0$ is fixed. Define T_i, $i \geq 1$ to be the arrival time of the ith jump of N after time s. Further suppose that (2.8) is stochastically contracting in the sense of Definition 11 under a differential coordinate transform $\Theta(t, \mathbf{x})$

which satisfies (2.35). Suppose the Lyapunov-like function (2.46) satisfies Assumption 4. For any $t > s$, if k jumps occur between s and t with probability $p_k(t-s)$ given by (2.61), we achieve the following bound:

$$\mathbb{E}_k[\|\mathbf{y}(t) - \mathbf{x}(t)\|^2] \leq \frac{1}{\underline{m}}\mathbb{E}_k[\|\mathbf{y}(s) - \mathbf{x}(s)\|^2]e^{-\beta_s(t-s)} + \frac{\kappa_s(\beta_s, s, t)}{\underline{m}} \tag{2.62}$$

for any $k \in \mathbb{N}$, where

$$\beta_s \triangleq 2\alpha, \tag{2.63a}$$

$$\kappa_s(\beta_s, s, t) \triangleq k\int_{s+}^{t} \frac{dh(\tau)}{d\tau} e^{-\beta_s(t-\tau)} d\tau + kh(s)e^{-\beta_s(t-s)}. \tag{2.63b}$$

Here, $\alpha > 0$ is the deterministic contraction rate from (2.36), $\underline{m}$ is defined in (2.35), and the function h is defined in Assumption 4.

Proof of Theorem 5. Using Lemma 3, (2.46) becomes

$$V(t, \mathbf{q}, \delta\mathbf{q}) = V(s, \mathbf{q}, \delta\mathbf{q}) + \int_{s+}^{t} \partial_\tau V(\tau-, \mathbf{q}, \delta\mathbf{q}) d\tau \tag{2.64a}$$

$$+ \int_{s+}^{t} \sum_{i=1}^{n} \left[\partial_{q_i} V(\tau-, \mathbf{q}, \delta\mathbf{q}) f_i(\tau, \mathbf{q}) + \partial_{\delta q_i} V(\tau-, \mathbf{q}, \delta\mathbf{q}) \left(F\delta\mathbf{q}\right)_i\right] d\tau \tag{2.64b}$$

$$+ \sum_{i=N(s)+1}^{N(t)} \left(V(T_i, \mathbf{q}, \delta\mathbf{q}) - V(T_i-, \mathbf{q}, \delta\mathbf{q})\right), \tag{2.64c}$$

where F is the Jacobian from (2.32), T_i is the time of the ith arrival in the Poisson process $N(t)$ driving the shot noise system (2.8), and we use the left-limit notation from the beginning of this chapter. Note that in (2.64c), we abuse the notation for the subscript i in T_i for both sums which range over $i = N(s)+1$ to $N(t)$ and sums which range over $i = 1$ to k. Because we will later condition on $N(t) - N(s) = k$, we abuse the notation in this way for the sake of simplicity. Similar to the argument of Remark 9, note that each term of the sum (2.64c) is nonzero only if there is a jump at time s, where $s \leq t$. Furthermore, the terms of (2.10) which correspond to the continuous part of the quadratic variation are zero for dynamics (2.8).

We condition on the number of jumps being $N(t) = k$ by applying $\mathbb{E}_k$ across the entire equation (2.64).

$$\mathbb{E}_k\left[V(t, \mathbf{q}, \delta\mathbf{q})\right] - \mathbb{E}_k\left[V(s, \mathbf{q}, \delta\mathbf{q})\right] = \mathbb{E}_k\left[\int_{s+}^{t} \partial_\tau V(\tau, \mathbf{q}, \delta\mathbf{q}) d\tau\right] \tag{2.65a}$$

$$+\sum_{i=1}^{n}\mathbb{E}_k\left[\int_{s+}^{t}\left(\partial_{q_i}V(\tau,\mathbf{q},\delta\mathbf{q})f_i(\tau,\mathbf{q})+\partial_{\delta q_i}V(\tau,\mathbf{q},\delta\mathbf{q})\left(F\delta\mathbf{q}\right)_i\right)d\tau\right] \tag{2.65b}$$

$$+\mathbb{E}_k\left[\sum_{i=N(s)+1}^{N(t)}\left(V(T_i,\mathbf{q},\delta\mathbf{q})-V(T_i-,\mathbf{q},\delta\mathbf{q})\right)\right]. \tag{2.65c}$$

Recall that the Lyapunov-like function (2.46) is twice continuously-differentiable with respect to its arguments $\mathbf{q}$ and $\delta\mathbf{q}$. This means there is a jump-discontinuity in V only if there is a jump discontinuity in $\mathbf{q}$ or $\delta\mathbf{q}$. But by the relationship between (2.41) and (2.42), $\mathbf{q}$ and $\delta\mathbf{q}$ experience jumps at the same times. Hence, the number of jumps experienced by V in a fixed interval of time $[s,t]$ is equal to the number of jumps experienced by the trajectory $\mathbf{q}$ in $[s,t]$.

A bound on (2.65a) and (2.65b) is derived from Theorem 3. We bound (2.65c) in the following way:

$$\mathbb{E}_k\left[\sum_{i=N(s)+1}^{N(t)}\left(V(T_i,\mathbf{q},\delta\mathbf{q})-V(T_i-,\mathbf{q},\delta\mathbf{q})\right)\right]=\mathbb{E}_k\left[\sum_{i=1}^{k}\left(V(T_i,\mathbf{q},\delta\mathbf{q})-V(T_i-,\mathbf{q},\delta\mathbf{q})\right)\right] \tag{2.66a}$$

$$\leq k\mathbb{E}_k\left[\max_{i\in\{1,\cdots,k\}}\left(V(T_i,\mathbf{q},\delta\mathbf{q})-V(T_i-,\mathbf{q},\delta\mathbf{q})\right)\right] \tag{2.66b}$$

$$\leq kh(t), \tag{2.66c}$$

where $h(t)$ is defined in Assumption 4. The inequality (2.66c) comes from (2.59) and the fact that $T_i\in[s,t]$ for all $i=1,\cdots,k$.

In combination, we get:

$$\mathbb{E}_k\left[V(t,\mathbf{q},\delta\mathbf{q})\right]-\mathbb{E}_k\left[V(s,\mathbf{q},\delta\mathbf{q})\right]\leq-2\alpha\int_{s+}^{t}\mathbb{E}_k[V(\tau,\mathbf{q},\delta\mathbf{q})]d\tau+kh(t), \tag{2.67}$$

where α is the contraction rate of the nominal system. Applying Lemma 5 with $y(t)\triangleq\mathbb{E}_k[V(t,\mathbf{q},\delta\mathbf{q})]$, $\zeta=kh(s)$, $\mu\triangleq2\alpha$, and $\theta(t)\triangleq kh(t)$ turns (2.67) into

$$\mathbb{E}_k[V(t,\mathbf{q},\delta\mathbf{q})]\leq\mathbb{E}_k[V(s,\mathbf{q},\delta\mathbf{q})]e^{-\beta_s(t-s)}+\kappa_s(\beta_s,s,t), \tag{2.68}$$

where β_s and $\kappa_s(\beta_s, s, t)$ are defined in (2.63a) and (2.63b), respectively. Note that by Assumption 2, (2.39), and Cauchy-Schwarz:

$$\underline{m}\mathbb{E}_k\left[\|\mathbf{y}(t)-\mathbf{x}(t)\|^2\right] \leq \underline{m}\mathbb{E}_k\left[\int_0^1 \|\partial_\mu \mathbf{q}(\mu,t)\|^2\, d\mu\right] \leq \mathbb{E}_k[V(t,\mathbf{q},\delta\mathbf{q})]. \tag{2.69}$$

We use (2.69) to write (2.68) as an inequality on the norm mean-squared error between the two trajectories $\mathbf{x}$ and $\mathbf{y}$. Because the condition that $N(t)-N(s)=k$ occurs with probability $p_k(t-s)$ given by (2.61), we obtain our desired bound (2.62). ■

Both the shot noise system and the nominal system have the same contraction rate. This is because the shot noise system behaves exactly as the nominal system in between consecutive jumps. The displacements of the shot noise trajectory incurred by the jumps of the shot noise process can each be thought of as a reset to a different initial condition from which the system evolves nominally.

Remark 10. In conjunction with Remark 7, we note two important differences between the white noise incremental stability result Theorem 4 and incremental stability results for systems which include impulsive shot noise, such as (2.8) and (2.7). First, taking $t \to \infty$ in the inequality of Theorem 4 yields a bound which can be interpreted as the steady-state error ball that solution trajectories will ultimately converge toward. However, stability for shot or Lévy noise systems is more comparable to finite-time stability theory, described in Chapter III of Kushner 1967 [87]. Due to the impulsive, large-norm jumps of the noise process, convergence between solution trajectories is considered only within a finite interval of time $[s,t]$ for each $0 \leq s < t$. The second difference is that the mean-squared error bounds for the cases of shot and Lévy noise systems are provided with a specific probability of satisfaction, which is weaker than the traditional mean-squared sense of convergence. This probability is dependent upon the number of jumps incurred by the noise process. Using the practical interpretation of Remark 7, this probabilistic guarantee allows for a smarter design for controllers and observers by including a predictive component. For instance, the probability of (2.61) can be used as a measure of expectation that k jumps will arise in a fixed, future horizon of time; given this event, the bound of Theorem 5 provides a guarantee on the mean-squared deviation of the perturbed trajectory away from the nominal.

Remark 11. The error ball (2.63b) depends on intensity λ and the jump norm bound η (defined in Assumption 1) of the shot noise process. For example, in Section 2.4.3, we consider a specific linear time-varying form of (2.8) to derive a concrete expression of $h(t)$ (from (2.59))

which is directly proportional to η. This implies that larger jump norms result in larger error bounds. While the relationship between $\kappa_s(\beta_s, s, t)$ and λ is not as straightforward to discern from the derivation of Section 2.4.3, we expect to see the following trend. An increasing λ is indicative of a more rapid accumulation of jumps, resulting in a larger overall deviation of the perturbed trajectory away from the nominal trajectory, and thus a larger error ball. In Section 2.5, we use numerical simulations to verify these relationships between η, λ, and the size of the error bound.

2.4.3 Specialization to the Linear Time-Varying Case

To demonstrate a concrete example of the function $h(t)$ from (2.59), we consider a specialization of Theorem 5 to the class of linear time-varying (LTV) systems where $f(t, \mathbf{x}) = A(t)\mathbf{x}(t)$ is a linear function and $\xi(t, \mathbf{x}) \equiv \xi(t)$ is a function is only dependent on time:

$$d\mathbf{x}(t) = A(t)\mathbf{x}(t)dt + \xi(t)dN(t). \tag{2.70}$$

Here, $A : \mathbb{R}^+ \to \mathbb{R}^{n\times n}$ is continuous for all $t \geq 0$, and $\xi : \mathbb{R}^+ \to \mathbb{R}^n$ is a random function which maps time to a random vector in $\mathbb{R}^n$ such that the bound in Assumption 1 is still satisfied. By virtue of Remark 8, we can leverage the additional knowledge that the shot noise system is LTV in order to further simplify the bound (2.60).

Note that a solution trajectory of (2.70) with value $\mathbf{x}(s) \in \mathbb{R}^n$ at time s is given by:

$$\mathbf{x}(t) = \Phi(t, s)\mathbf{x}(s) + \int_{s+}^{t} \Phi(t, \tau)\xi(\tau)dN(\tau) = \Phi(t, s)\mathbf{x}(s) + \sum_{i=N(s)+1}^{N(t)} \Phi(t, T_i)\xi(T_i), \tag{2.71}$$

where the second equality follows from the definition of the Poisson integral from Section 2.3.2 of Applebaum 2009 [8].

Instead of using a parameter $\mu \in [0, 1]$, we consider the construction of the virtual system by stacking the SDEs (2.70) and the nominal system $d\mathbf{y}(t) = A(t)\mathbf{y}(t)dt$:

$$d\mathbf{q}(t) = \begin{bmatrix} d\mathbf{x}(t) \\ d\mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} A(t)\mathbf{x}(t) \\ A(t)\mathbf{y}(t) \end{bmatrix} dt + \begin{bmatrix} \xi(t) \\ 0 \end{bmatrix} dN(t), \tag{2.72}$$

where $N(t)$ is the standard Poisson process with intensity $\lambda > 0$.

Assumption 5. There exists a continuously-differentiable, uniformly positive-definite, symmetric matrix $P(t)$ such that

1. we can define the bounds

$$\alpha_1 \triangleq \inf_t \lambda_{\min} P(t), \quad \alpha_2 \triangleq \sup_t \lambda_{\max} P(t) \tag{2.73}$$

for finite constants $\alpha_1, \alpha_2 > 0$;

2. for all $t \geq 0$ and a fixed $\alpha > 0$,

$$\partial_t P(t) + P(t)A(t) + A^\top(t)P(t) \leq -2\alpha P(t). \tag{2.74}$$

The decomposition of $f(t, \mathbf{x})$ into $A(t)\mathbf{x}$, which can be seen as a product of a function of time and a function of state, allows us to consider metrics $M(t, \mathbf{q}) \equiv M(t)$ which are independent of state $\mathbf{q} \in \mathbb{R}^{2n}$ of the virtual system (2.72). Moreover, we choose the metric to be $M(t) \triangleq P(t)$, where $P(t)$ satisfies Assumption 5. The condition (2.74) can be viewed as a simplification of (2.36) for LTV systems. Because the metric is independent of state, we can construct a Lyapunov-like function which is simplified compared to (2.46):

$$V(t, \mathbf{q}(t)) = (\mathbf{y}(t) - \mathbf{x}(t))^\top P(t)(\mathbf{y}(t) - \mathbf{x}(t)). \tag{2.75}$$

We further assume that the nominal system admits a solution with the state transition matrix $\Phi(t, s) \triangleq e^{\int_s^t A(r)dr}$ satisfying the following assumption, which is derived from Theorem 4.11 of Khalil 2002 [80].

Assumption 6 (Bounded State-Transition Matrix)**.** The state-transition matrix $\Phi(t, \tau) \triangleq e^{\int_\tau^t A(r)dr}$ satisfies the following condition

$$\|\Phi(t, \tau)\| \leq \kappa e^{-\beta(t-\tau)}, \quad \forall\ 0 \leq \tau \leq t \tag{2.76}$$

for some $\kappa, \beta > 0$ and any trajectory $\mathbf{q}(t)$ from a system with nominal dynamics $d\mathbf{q}(t) = A(t)\mathbf{q}(t)dt$.

Theorem 6 (Shot Noise Stochastic Contraction Theorem: LTV Systems)**.** Suppose the LTV shot noise system (2.70) is perturbed by noise processes which satisfy Assumption 1, and is stochastically contracting in the sense of Definition 11 under the metric $P(t)$ from Assumption 5. Further suppose that the nominal LTV system is such that Assumption 6 and Assumption 5 holds. If for a fixed interval of time $[s, t]$ for $0 \leq s < t$, $k \in \mathbb{N}$ jumps occur with probability $p_k(t - s)$ given by (2.61), then (2.45) can be written explicitly as:

$$\mathbb{E}_k[\|\mathbf{y}(t) - \mathbf{x}(t)\|^2] \leq \frac{1}{\alpha_1}\mathbb{E}_k[\|\mathbf{y}(s) - \mathbf{x}(s)\|^2]e^{-\beta_s(t-s)} + \frac{\kappa_s(\beta_s, s, t)}{\alpha_1}, \tag{2.77}$$

where

$$\beta_s \triangleq 2\alpha, \tag{2.78a}$$

$$\kappa_s(\beta_s, s, t) \triangleq \int_{s+}^{t} \frac{d\psi_k(s,\tau)}{d\tau} e^{-\beta_s(t-\tau)} d\tau + k\alpha_2\eta^2 e^{-\beta_s(t-s)}, \tag{2.78b}$$

and

$$\psi_k(s,t) \triangleq \mathbb{E}_k \left[\sum_{i=1}^{k} \left\{ 2\alpha_2\eta\kappa \left\| \mathbf{y}(s) - \mathbf{x}(s) \right\| e^{-\beta(T_i - s)} + 2\alpha_2\kappa\eta^2 \left(\sum_{j=1}^{i-1} e^{-\beta(T_i - T_j)} \right) \right\} \right]. \tag{2.79}$$

Here, $N(t)$ is the standard Poisson process in (2.8), and $T_i \geq s$, $i \geq 1$ is the arrival time of the ith jump after time s in the Poisson process N driving (2.70), with convention $T_0 \triangleq s$. The variable λ is the intensity of $N(t)$, α_1 and α_2 are defined in Assumption 5, and η is the bound on the norm of the jumps $\xi(t)$ described by Assumption 1.

Proof of Theorem 6. Applying Lemma 3 to the Lyapunov-like function (2.75) yields

$$V(t,\mathbf{q}) - V(s,\mathbf{q}) =$$

$$\int_{s+}^{t} \partial_\tau V(\tau-,\mathbf{q}) d\tau + \int_{s+}^{t} \left(\nabla_{\mathbf{x}} V(\tau-,\mathbf{q}) \cdot A(\tau)\mathbf{x}(\tau) + \nabla_{\mathbf{y}} V(\tau-,\mathbf{q}) \cdot A(\tau)\mathbf{y}(\tau) \right) d\tau \tag{2.80a}$$

$$+ \sum_{i=N(s)+1}^{N(t)} \left(V(T_i,\mathbf{q}) - V(T_i-,\mathbf{q}) \right). \tag{2.80b}$$

As in the proof to Theorem 5, we again abuse the notation of the subscript i in T_i for both sums which range over $i = N(s)+1$ to $N(t)$ and sums which range over $i = 1$ to k for the sake of simplicity. Using left-limit notation, we write

$$\nabla_{\mathbf{x}} V(s,\mathbf{q}(\tau)) = -2(\mathbf{y}(\tau) - \mathbf{x}(\tau))^\top P(t), \quad \nabla_{\mathbf{y}} V(s,\mathbf{q}(\tau)) = 2(\mathbf{y}(\tau) - \mathbf{x}(\tau))^\top P(t). \tag{2.81}$$

Let $\mathbf{x}(t)$ be the solution trajectory described in (2.71) with value $\mathbf{x}(s) \in \mathbb{R}^n$ at time t_0. Further denote $\mathbf{y}(t) = \Phi(t,s)\mathbf{y}(s)$ to be the solution trajectory of the nominal system with value $\mathbf{y}(s) \in \mathbb{R}^n$ at time t_0. We can simplify each term in the sum (2.80b) as follows:

$$V(T_i,\mathbf{q}) - V(T_i-,\mathbf{q}) = \mathbf{y}(T_i)^\top P(T_i)(\mathbf{x}(T_i-) - \mathbf{x}(T_i)) + (\mathbf{x}(T_i-) - \mathbf{x}(T_i))^\top P(T_i)\mathbf{y}(T_i)$$
$$+ (\mathbf{x}^\top P\mathbf{x}(T_i) - \mathbf{x}^\top P\mathbf{x}(T_i-)) = -2\mathbf{y}^\top P\xi(T_i) + (\mathbf{x}^\top P\mathbf{x}(T_i) - \mathbf{x}^\top P\mathbf{x}(T_i-)), \tag{2.82}$$

where $\mathbf{y}^\top P\xi(\tau)$ is shorthand notation for $\mathbf{y}^\top(\tau)P(\tau)\xi(\tau)$ for any τ, and likewise for other similar notation. The first equality comes from the fact that $\mathbf{y}(T_i) = \mathbf{y}(T_i-)$ for all $T_i \leq t$

due to its continuity. The second equality is obtained by by virtue of $N(T_i-) = N(T_i) - 1$ and $N(T_i) = i$.

The second term of (2.82) can be simplified as follows:

$$\mathbf{x}^\top P\mathbf{x}(T_i) - \mathbf{x}^\top P\mathbf{x}(T_i-) = 2\xi^\top(T_i)P(T_i)\Phi(T_i,s)\mathbf{x}(s) + 2\left(\sum_{j=N(s)+1}^{i-1}\xi^\top(T_i)\Phi^\top(T_i,T_j)\right)P(T_i)\xi(T_i) + \xi^\top P\xi(T_i), \quad (2.83)$$

where the second equality is obtained by using the fact that $N(T_i-) = N(T_i) - 1$ and that $N(T_i) = i$. Substituting (2.82) into (2.80) yields:

$$\begin{aligned}
V(t,\mathbf{q}) - V(s,\mathbf{q}) &= \int_{s+}^{t}(\mathbf{y}(\tau)-\mathbf{x}(\tau-))^\top(\partial_\tau P(\tau) + 2P(\tau)A(\tau))(\mathbf{y}(\tau)-\mathbf{x}(\tau-))d\tau \\
&\quad + \sum_{i=N(s)+1}^{N(t)}\Bigg[-2\xi^\top(T_i)P(T_i)\Phi(T_i,s)(\mathbf{y}(s)-\mathbf{x}(s)) \\
&\quad + 2\left(\sum_{j=N(s)+1}^{i-1}\xi^\top(T_i)\Phi^\top(T_i,T_j)\right)P(T_i)\xi(T_i) + \xi^\top P\xi(T_i)\Bigg] \qquad (2.84a) \\
&\le -2\alpha\int_{s+}^{t}V(\tau-,\mathbf{q})d\tau + \sum_{i=N(s)+1}^{N(t)}\Bigg[2\alpha_2\left\|\xi(T_i)\right\|\left\|\Phi(T_i,s)\right\|\left\|\mathbf{y}(s)-\mathbf{x}(s)\right\| \\
&\quad + 2\alpha_2\left(\sum_{j=N(s)+1}^{i-1}\xi^\top(T_j)\Phi^\top(T_i,T_j)\right)\xi(T_i) + \alpha_2\xi^\top\xi(T_i)\Bigg] \qquad (2.84b) \\
&\le -2\alpha\int_{s+}^{t}V(\tau-,\mathbf{q})d\tau + \sum_{i=N(s)+1}^{N(t)}\Bigg[2\alpha_2\left\|\xi(T_i)\right\|\kappa\left\|\mathbf{y}(s)-\mathbf{x}(s)\right\|e^{-\beta(T_i-s)} \\
&\quad + 2\alpha_2\kappa\left\|\xi(T_i)\right\|\sum_{j=N(s)+1}^{i-1}\left\|\xi(T_j)\right\|e^{-\beta(T_i-T_j)} + \alpha_2\left\|\xi(T_i)\right\|^2\Bigg] \qquad (2.84c) \\
&\le -2\alpha\int_{s+}^{t}V(\tau-,\mathbf{q})d\tau + \sum_{i=N(s)+1}^{N(t)}\Bigg[2\alpha_2\eta\kappa\left\|\mathbf{y}(s)-\mathbf{x}(s)\right\|e^{-\beta(T_i-s)} \\
&\quad + 2\alpha_2\kappa\eta^2\left(\sum_{j=N(s)+1}^{i-1}e^{-\beta(T_i-T_j)}\right) + \alpha_2\eta^2\Bigg]. \qquad (2.84d)
\end{aligned}$$

Here, (2.84b) from using Assumption 5, the bound on $P(t)$, and submultiplicativity. The third inequality (2.84c) comes from submultiplicativity, triangle inequality, and (2.76). Fi-

nally, (2.84d) is obtained by using Assumption 1. Taking the conditional expectation $\mathbb{E}_k[\cdot]$ over (2.84) yields:

$$\mathbb{E}_k[V(t,\mathbf{q})] - \mathbb{E}_k[V(s,\mathbf{q})] \leq -2\alpha \int_{s+}^{t} \mathbb{E}_k\left[V(\tau-,\mathbf{q})\right] d\tau + \psi_k(s,t) + k\alpha_2\eta^2, \tag{2.85}$$

where $\psi_k(s,t)$ is as in (2.79). Apply Lemma 5 to (2.85) with $y(t) \triangleq \mathbb{E}_k[V(t,\mathbf{q})]$, $\zeta \triangleq k\alpha_2\eta^2$, $\mu \triangleq 2\alpha$, and $\theta(t) \triangleq \psi_k(s,t)$. Use (2.69) and the fact that $(1/\alpha_2)I_n \leq P^{-1}(t) \leq (1/\alpha_1)I_n$ to write the inequality in terms of the norm mean-squared error between $\mathbf{x}$ and $\mathbf{y}$. This gives us the desired bound (2.77), with β_s as in (2.78a) and $\kappa_s(\beta_s,t)$ as in (2.78b). ■

Note that there are several ways to simplify $\psi_k(s,t)$. One can explicitly write out the integral form of the expectation with the knowledge that T_i are Gamma-distributed with parameter i and λ for all $i = 1,\cdots,k$. While this direct computation of (2.79) yields the tightest bound, this method requires computing k integrals and is thus increasingly difficult to compute with increasing k. We provide two ways to derive a looser bound.

The first term of (2.79) simplifies as

$$\begin{aligned} 2\alpha_2\eta\kappa\mathbb{E}_k\left[\sum_{i=1}^{k} \|\mathbf{y}(s)-\mathbf{x}(s)\|\, e^{-\beta(T_i-s)}\right] &\leq 2\alpha_2\eta\kappa e^{\beta s}\mathbb{E}_k[\|\mathbf{y}(s)-\mathbf{x}(s)\|]k\mathbb{E}_k\left[\max_{i=1,\cdots,k} e^{-\beta T_i}\right] \\ &= 2\alpha_2\eta\kappa e^{\beta s}\mathbb{E}_k[\|\mathbf{y}(s)-\mathbf{x}(s)\|]k\mathbb{E}\left[e^{-\beta T_1}\right] \end{aligned} \tag{2.86}$$

because the random variables $T_i, i = 1,\cdots,k$ are such that $T_1 \leq T_2 \leq \cdots T_k$, and so $e^{-\beta T_i}$ takes the largest value with the smallest index i.

For the second term of (2.79), we use Lemma 6:

$$\begin{aligned} 2\alpha_2\kappa\eta^2\mathbb{E}_k\left[\sum_{i=1}^{k}\sum_{j=1}^{i-1} e^{-\beta(T_i-T_j)}\right] &\leq 2\alpha_2\kappa\eta^2 k(k-1)\mathbb{E}_k\left[\max_{i=1,\cdots,k} e^{-\beta T_i}\max_{j=1,\cdots,k} e^{\beta T_j}\right] \\ &= 2\alpha_2\kappa\eta^2 k(k-1)\mathbb{E}_k\left[e^{-\beta T_1}e^{\beta T_k}\right]. \end{aligned} \tag{2.87}$$

Here, (2.87) holds because $T_i, i = 1,\cdots,k$ are such that $T_1 \leq T_2 \leq \cdots T_k$. This means the value of $i \in \{1,\cdots,k\}$ which maximizes $e^{-\beta T_i}$ is $i = 1$, and the value of $j \in \{1,\cdots,k\}$ which maximizes $e^{\beta T_j}$ is $j = k$. Use the fact that $T_1 \sim \text{Exp}(\lambda)$ and $T_k \sim \text{Gamma}(k,\lambda)$ to further simplify the resulting inequality. The derivative of the function $\psi_k(s,t)$ from (2.79) simplifies to

$$\frac{d\psi_k(s,t)}{dt} \triangleq 2\alpha_2\eta\kappa\,\|\mathbf{y}(s)-\mathbf{x}(s)\|\, ke^{\beta s}\lambda e^{-(\lambda+\beta)t} + \frac{2\alpha_2\kappa\eta^2 k(k-1)\lambda^{k+1}}{(\lambda+\beta)\Gamma(k)} t^{k-1}\left(e^{(\beta-\lambda)t} - e^{-2\lambda t}\right). \tag{2.88}$$

Alternatively, we can bound the second term of (2.79) in the following way:

$$2\alpha_2\kappa\eta^2\mathbb{E}_k\left[\sum_{i=1}^{k}\sum_{j=1}^{i-1}e^{-\beta(T_i-T_j)}\right] \leq 2\alpha_2\kappa\eta^2\mathbb{E}_k\left[\sum_{i=1}^{k}(i-1)\max_{j\in\{1,\cdots,i-1\}}e^{-\beta(T_i-T_j)}\right]$$
$$= \alpha_2\kappa\eta^2k(k-1)\mathbb{E}_k\left[e^{-\beta S_i}\right], \tag{2.89}$$

where $S_i \triangleq T_i - T_{i-1}$ is exponentially-distributed with parameter λ. The maximum value of j is achieved at $j = i-1$ because the difference $T_i - T_{i-1}$ is the smallest value in the range, and thus $e^{-\beta(T_i-T_{i-1})}$ is the largest. Note that S_i represent the interarrival times of the Poisson jumps and are thus i.i.d. for all $i = 1, \cdots, k$. The derivative of the function $\psi_k(s,t)$ from (2.79) simplifies to

$$\frac{d\psi_k(s,t)}{dt} \triangleq \left(2\alpha_2\eta\kappa\left\|\mathbf{y}(s)-\mathbf{x}(s)\right\|ke^{\beta s} + \alpha_2\kappa\eta^2k(k-1)\right)\lambda e^{-(\lambda+\beta)t}. \tag{2.90}$$

As mentioned in Remark 11, $\psi_k(s,t)$ of (2.79), and thus $\kappa_s(\beta_s,s,t)$ of (2.78b), is directly proportional to the jump norm bound η.

Remark 12. A comparison of the results between Theorem 6 and Theorem 5 show that the form of the stability bounds is the same. First, note that $\underline{m} = \alpha_1$ and β_s from (2.78a) in Theorem 6 is as in (2.63a) from Theorem 5. Second, and more importantly, having more knowledge about the system dynamics allows us to derive a more concrete bound compared to the bounds of Section 2.4.2, which are dependent upon some abstract function $h(t)$. In particular, for this LTV case, the difference (2.60) can be computed exactly using the precise solution form (2.71), and the metric $M(t,\mathbf{x}) \triangleq P(t)$ does not depend on the state $\mathbf{x}$. From (2.85), we have $h(t) = \psi_k(s,t)$.

Remark 13. Similar to Remark 11, the strength of the stability bound in Theorem 6 is contingent on both inherent parameters and design parameters, but the explicit form of (2.79) allows us to derive additional insights. First, an additional inherent parameter we can consider is the the maximum norm bound η: (2.79) is directly proportional to η, which implies that larger jump norms result in larger error bounds. Second, among the design parameters, the contraction metric $M(t,\mathbf{x}) \equiv P(t)$ can be chosen such that α_1 is small and α_2 is large. This enables a looser bound on (2.77), and so (2.77) is tighter and more meaningful when we

choose a metric whose condition number is close to 1. For exponentially-stable unperturbed LTV systems, (2.79) also demonstrates that a larger deterministic contraction rate allows for faster convergence to a smaller error ball. Correspondingly, this effect can also be achieved for controllable open-loop unstable unperturbed LTV systems by designing a control law such that, in Assumption 6, κ is small and β is large. In Section 2.5, we use numerical simulations to investigate how the stability bounds vary by varying the different parameters discussed in here and in Remark 11.

2.4.4 For Bounded-Measure Lévy Noise Systems

In this section, the stochastic contraction theorem for Lévy noise system (2.7) is presented and proven to be a combination of the white and the shot noise cases. Consider two trajectories of a system: $\mathbf{x}(t)$ a solution of (2.7) with initial condition $\mathbf{x}_0$, and $\mathbf{y}(t)$ a solution of the nominal system $d\mathbf{y}(t) = f(t, \mathbf{y})dt$ with initial condition $\mathbf{y}_0$. We define the parameter $\mu \in [0, 1]$ which yields the parametrization (2.40); in particular, we again use the specific parametrization (2.63). Analogous to the white noise parameters β_w, κ_w from (2.55) and the shot noise parameters β_s, κ_s from (2.63), denote β_ℓ, κ_ℓ to be the contraction rate and error bound, respectively, for the Lévy noise system. The derivation of the error bound makes use of the following lemma.

Lemma 7. Consider the function (2.46) with respect to a metric M which satisfies Assumption 2. Then the following identities hold true for each time $t > 0$:

$$\sum_{i,j=1}^{n} \partial^2_{\delta q_i \delta q_j} V(t, \mathbf{q}, \delta\mathbf{q}) d[\delta q_i, \delta q_j]^c(t) \leq 2\overline{m}\gamma^2 dt, \tag{2.91a}$$

$$\sum_{i,j=1}^{n} \partial^2_{q_i q_j} V(t, \mathbf{q}, \delta\mathbf{q}) d[q_i, \delta q_j]^c(t) \leq m'\gamma^2 \left(\int_0^1 \|\partial_\mu \mathbf{q}(\mu, t)\|^2 \, d\mu + 1 \right) dt, \tag{2.91b}$$

$$\sum_{i,j=1}^{n} \partial^2_{q_i q_j} V(t, \mathbf{q}, \delta\mathbf{q}) d[q_i, q_j]^c(t) \leq \left(m''\gamma^2 \int_0^1 \|\partial_\mu \mathbf{q}(\mu, t)\|^2 \, d\mu \right) dt, \tag{2.91c}$$

where the constants are as in Assumption 2 and Assumption 1.

Proof of Lemma 7. First, we compute the quadratic variation terms.

$$d[\delta q_i, \delta q_j]^c(t) = \sum_{k=1}^{d} \delta\sigma_{\mu,ik}(t, \mathbf{q}) \delta\sigma_{\mu,jk}(t, \mathbf{q}) dt,$$

$$d[q_i, \delta q_j]^c(t) = \sum_{k=1}^{d} \sigma_{\mu,ik}(t, \mathbf{q}) \delta\sigma_{\mu,jk}(t, \mathbf{q}) dt,$$

$$d[q_i, q_j]^c(t) = \sum_{k=1}^{d} \sigma_{\mu,ik}(t, \mathbf{q})\sigma_{\mu,jk}(t, \mathbf{q})dt,$$

where $\delta\sigma_\mu$ is defined in (2.43) and σ_μ comes from the notation of the virtual system.

a. *Proof of (2.91a)*: From matrix multiplication, and the fact that $M(t, \mathbf{q})$ is independent of $\delta\mathbf{q}$:

$$\delta\mathbf{q}^\top(t)M(t, \mathbf{q})\delta\mathbf{q}(t) = \sum_{k=1}^{n}\sum_{l=1}^{n} \delta q_k \delta q_l M_{kl}(t, \mathbf{q})$$
$$\partial^2_{\delta q_i \delta q_j} V(t, \mathbf{q}(t), \delta\mathbf{q}(t)) = 2\int_0^1 M_{ij}(t, \mathbf{q}(\mu, t))d\mu.$$

Substituting into the left side of (2.91a), we get:

$$\begin{aligned}
&\sum_{i,j=1}^{n} \partial^2_{\delta q_i \delta q_j} V(t, \mathbf{q}(t), \delta\mathbf{q}(t))d[\delta q_i, \delta q_j]^c(t) \\
&= 2\int_0^1 \sum_{i,j=1}^{n} M_{ij}(t, \mathbf{q}) \sum_{k=1}^{d} \left(\partial_\mu \sigma_\mu(t, \mathbf{q})\right)_{ik} \left(\partial_\mu \sigma_\mu(t, \mathbf{q})\right)_{jk} d\mu dt \\
&\leq 2\overline{m} \sum_{i,j=1}^{n}\sum_{k=1}^{d} \int_0^1 \left(\partial_\mu \sigma_\mu(t, \mathbf{q})\right)_{ik} \left(\partial_\mu \sigma_\mu(t, \mathbf{q})\right)_{jk} d\mu dt. \qquad (2.92)
\end{aligned}$$

We have the following identity for any square matrix A and any pair $i, j = 1, \cdots, n$ such that $i \neq j$:

$$\frac{1}{2}\mathrm{tr}(A^\top A) = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{d} a_{ik}^2 \geq \sum_{i\neq j}^{n}\sum_{k=1}^{d} a_{ik}a_{jk},$$

which is easily seen by completing the squares. This allows us to bound (2.92) by splitting the sum up into terms with $i = j$ and terms with $i \neq j$ and bounding both parts by a trace:

$$2\overline{m} \sum_{i,j=1}^{n}\sum_{k=1}^{d} \int_0^1 \left(\partial_\mu \sigma_\mu(t, \mathbf{q})\right)_{ik} \left(\partial_\mu \sigma_\mu(t, \mathbf{q})\right)_{jk} d\mu dt \leq 2\overline{m}\mathrm{tr}\left(\sigma_\mu^\top \sigma_\mu(t, \mathbf{q})\right) \leq 2\overline{m}\gamma^2 dt.$$

b. *Proof of (2.91b)*: First, we can compute the matrix derivative as follows.

$$\partial^2_{q_i \delta q_j} V(t, \mathbf{q}(t), \delta\mathbf{q}(t)) = 2\int_0^1 \partial_{q_i} M_{j,i}(t, \mathbf{q}) \left(\partial_\mu \mathbf{q}(\mu, t)\right)_i d\mu \quad \forall 1 \leq i, j \leq n.$$

This gives:

$$
\begin{aligned}
&2\int_0^1 \sum_{i,j=1}^{n} \partial_{q_i} M_{j,i}(t,\mathbf{q}) \left(\partial_\mu \mathbf{q}(\mu,t)\right)_i \sum_{k=1}^{d} \sigma_{\mu,ik} \left(\partial_\mu \sigma_\mu(t,\mathbf{q})\right)_{jk} d\mu dt \\
&\leq 2m' \int_0^1 \sum_{i=1}^{n} \left(\partial_\mu \mathbf{q}(\mu,t)\right)_i \sum_{k=1}^{d} \sigma_{\mu,ik}(t,\mathbf{q}) \delta\sigma_{\mu,jk}(t,\mathbf{q}) d\mu dt \\
&\leq 2m'\gamma^2 \int_0^1 \left\|\partial_\mu \mathbf{q}(\mu,t)\right\| d\mu dt, \qquad (2.93)
\end{aligned}
$$

where the second-to-last inequality follows from the same trace bound used in (2.91a). Applying the fact that $2a \leq (a^2+1)$ for all $a > 0$ yields our final bound:

$$
(2.93) \leq m'\gamma^2 \left(\int_0^1 \left\|\partial_\mu \mathbf{q}(\mu,t)\right\|^2 d\mu + 1 \right) dt.
$$

c. *Proof of (2.91c)*: Again, start by computing the matrix derivative

$$
\partial^2_{q_i q_j} V(t,\mathbf{q}(t),\delta\mathbf{q}(t)) = \int_0^1 \sum_{k,l=1}^{n} \partial^2_{q_i q_j} M_{kl}(t,\mathbf{q}) \left(\partial_\mu \mathbf{q}(\mu,t)\right)_k \left(\partial_\mu \mathbf{q}(\mu,t)\right)_l d\mu.
$$

Now we can use the same technique as in (2.91b).

$$
\begin{aligned}
&\sum_{i,j=1}^{n} \partial^2_{q_i q_j} V(t,\mathbf{q}(t),\delta\mathbf{q}(t)) d[q_i,q_j]^c(t) \\
&= \sum_{i,j=1}^{n} \sum_{k,l=1}^{n} \int_0^1 \partial^2_{q_i q_j} M_{kl}(t,\mathbf{q}) \left(\partial_\mu \mathbf{q}(\mu,t)\right)_k \cdot \sum_{r=1}^{d} \sigma_{\mu,ir}(t,\mathbf{q}) \sigma_{\mu,jr}(t,\mathbf{q}) \cdot \left(\partial_\mu \mathbf{q}(\mu,t)\right)_l d\mu dt \\
&\leq \left(m''\gamma^2 \int_0^1 \left\|\partial_\mu \mathbf{q}(\mu,t)\right\|^2 d\mu \right) dt.
\end{aligned}
$$

■

Theorem 7 (Lévy Noise Stochastic Contraction Theorem)**.** Consider the Lévy noise system (2.7) which has trajectories that begin from different initial conditions and are perturbed by noise processes which satisfy Assumption 1. Let $N(t)$ be the standard Poisson process in (2.7), and define T_i, $i \geq 1$ to be the arrival time of the ith jump of $N(t)$. Further suppose that (2.7) is stochastically contracting in the sense of Definition 11 under a differential coordinate transform $\Theta(t,\mathbf{x})$ which satisfies (2.35). If the Lyapunov-like function (2.46) satisfies

and Assumption 4, then with probability $p_k(t-s)$ given by (2.61), we achieve the following bound:

$$\mathbb{E}_k[\|\mathbf{y}(t)-\mathbf{x}(t)\|^2] \leq \frac{1}{\underline{m}}\mathbb{E}_k\left[\|\mathbf{y}(s)-\mathbf{x}(s)\|^2\right] e^{-\beta_\ell(t-s)} + \frac{\kappa_\ell(\beta_\ell,s,t)}{\underline{m}}, \tag{2.94}$$

where

$$\beta_\ell \triangleq 2\alpha - \frac{\gamma^2}{\underline{m}}\left(m' + \frac{m''}{2}\right) = \beta_w \tag{2.95a}$$

$$\begin{aligned}\kappa_\ell(\beta_\ell,s,t) &\triangleq k\int_{s+}^{t}\frac{dh(\tau)}{d\tau}e^{-\beta_\ell(t-\tau)}d\tau + kh(s)e^{-\beta_\ell(t-s)} + \frac{\gamma^2}{\beta_\ell}\left(m'+\overline{m}\right)\left(1-e^{-\beta_\ell(t-s)}\right)\\ &= \kappa_s(\beta_\ell,s,t) + \frac{1}{\beta_\ell}\kappa_w(\beta_\ell,s,t) = \kappa_s(\beta_w,s,t) + \frac{1}{\beta_w}\kappa_w(\beta_w,s,t).\end{aligned} \tag{2.95b}$$

Here, α is the deterministic contraction rate from (2.36), γ is the norm bound on the variation of the white noise process from Assumption 1, and $\underline{m}, m', m''$ are defined in (2.35). The function $h(\tau)$ is defined in Assumption 4. The function h is defined in Assumption 4.

Proof of Theorem 7. Applying Lemma 3 to (2.46):

$$V(t,\mathbf{q},\delta\mathbf{q}) = V(s,\mathbf{q},\delta\mathbf{q}) + \int_{s+}^{t}\partial_\tau V(\tau-,\mathbf{q},\delta\mathbf{q})d\tau \tag{2.96a}$$

$$+\int_{s+}^{t}\sum_{i=1}^{n}\left[\partial_{q_i}V(\tau-,\mathbf{q},\delta\mathbf{q})f_i(\tau,\mathbf{q}) + \partial_{\delta q_i}V(\tau-,\mathbf{q},\delta\mathbf{q})\left(F\delta\mathbf{q}\right)_i\right]d\tau \tag{2.96b}$$

$$+\int_{s+}^{t}\sum_{i=1}^{n}\left[\partial_{q_i}V(\tau-,\mathbf{q},\delta\mathbf{q})\sigma_{\mu,i}(\mathbf{q}) + \partial_{\delta q_i}V(\tau-,\mathbf{q},\delta\mathbf{q})\delta\sigma_{\mu,i}(\mathbf{q})\right]dW(\tau) \tag{2.96c}$$

$$+\frac{1}{2}\left[\int_{s+}^{t}\sum_{i,j=1}^{n}\partial^2_{\delta q_i\delta q_j}V(\tau-,\mathbf{q},\delta\mathbf{q})d\left[\delta q_i,\delta q_j\right]^c(\tau)\right. \tag{2.96d}$$

$$+2\int_{s+}^{t}\sum_{i,j=1}^{n}\partial^2_{q_i\delta q_j}V(\tau-,\mathbf{q},\delta\mathbf{q})d\left[q_i,\delta q_j\right]^c(\tau) \tag{2.96e}$$

$$\left.+\int_{s+}^{t}\sum_{i,j=1}^{n}\partial^2_{q_iq_j}V(\tau-,\mathbf{q},\delta\mathbf{q})d\left[q_i,q_j\right]^c(\tau)\right] \tag{2.96f}$$

$$+\sum_{i=N(s)+1}^{N(t)}\left(V(T_i,\mathbf{q},\delta\mathbf{q}) - V(T_i-,\mathbf{q},\delta\mathbf{q})\right), \tag{2.96g}$$

where F is the Jacobian from (2.32), and T_i denotes the time of the ith jump. As in the proof of Theorem 5, a bound on (2.96a) and (2.96b) are derived from Theorem 3. Simplifying the

quadratic variation terms (2.96d) to (2.96f) requires computing the partial derivatives of V. From Lemma 7, we obtain the following inequalities:

$$\sum_{i,j=1}^{n} \partial^2_{\delta q_i \delta q_j} V(\tau-, \mathbf{q}, \delta\mathbf{q}) d[\delta q_i, \delta q_j]^c(\tau) \leq 2\overline{m}\gamma^2 d\tau, \tag{2.97a}$$

$$\sum_{i,j=1}^{n} \partial^2_{q_i \delta q_j} V(\tau-, \mathbf{q}, \delta\mathbf{q}) d[q_i, \delta q_j]^c(\tau) \leq m'\gamma^2 \left(\int_0^1 \left\| \partial_\mu \mathbf{q}(\mu, \tau) \right\|^2 d\mu + 1 \right) d\tau, \tag{2.97b}$$

$$\sum_{i,j=1}^{n} \partial^2_{q_i q_j} V(\tau-, \mathbf{q}, \delta\mathbf{q}) d[q_i, q_j]^c(\tau) \leq \left(m''\gamma^2 \int_0^1 \left\| \partial_\mu \mathbf{q}(\mu, \tau) \right\|^2 d\mu \right) d\tau, \tag{2.97c}$$

where γ is the white-noise bound from Definition 11, and $\overline{m}, m', m''$ are the metric bounds defined in Assumption 2. Note that we can use (2.69) to further simplify equations (2.96d) to (2.96f). Moreover, applying Assumption 4 and following logic similar to that of the proof to Theorem 5 gives us a bound on (2.96g).

Taking $\mathbb{E}_k[\cdot]$ across the entire inequality, note that the white noise term (2.96c) disappears due to being a martingale with zero mean. Combining the bounds of each remaining term from (2.96) yields the following:

$$\begin{aligned} \mathbb{E}_k\left[V(t, \mathbf{q}, \delta\mathbf{q})\right] - \mathbb{E}_k[V(s, \mathbf{q}, \delta\mathbf{q})] \leq -\left(2\alpha - \frac{m''\gamma^2}{2\underline{m}} - \frac{m'\gamma^2}{\underline{m}} \right) \int_{s+}^{t} \mathbb{E}_k[V(\tau-, \mathbf{q}, \delta\mathbf{q})] d\tau \\ + (m'\gamma^2 + \overline{m}\gamma^2)t + kh(t). \end{aligned} \tag{2.98}$$

We obtain a bound on the solution $\mathbb{E}_k[V(t, \mathbf{q}, \delta\mathbf{q})]$ using Lemma 5 with $y(t) \triangleq \mathbb{E}_k[V(t, \mathbf{q}, \delta\mathbf{q})]$, $\zeta = kh(s)$, $\mu \triangleq 2\alpha$, and $\theta(t) \triangleq (m'\gamma^2 + \overline{m}\gamma^2)t + kh(t)$. Then we use (2.69) to write the resulting inequality in terms of the norm mean-squared error between $\mathbf{x}$ and $\mathbf{y}$. This gives us the desired bound (2.94). ■

Remark 14 (Lévy Parameters in Terms of White and Shot Parameters)**.** The parameters (2.95) for the Lévy noise SDE (2.7) can be expressed as a combination of (2.55) and (2.63) in the following way. β_ℓ can be interpreted as being the direct sum of β_w and $\beta_s (= 2\alpha)$ with the extra 2α term removed to prevent double-counting of the convergence rate from the nominal system:

$$\beta_\ell = \beta_s + \beta_w - 2\alpha = \beta_w.$$

Furthermore, $\kappa_\ell(\beta_\ell, s, t)$ is a sum of the white noise error ball and the shot noise error ball with contraction rate β_ℓ used in place of β_w or β_s; this is written in the last two equations

of (2.95b). We emphasize the importance of this remark because of its likeness to the Lévy-Khintchine theorem, which represents Lévy processes as an additive combination of Brownian motion processes and compound Poisson processes.

Remark 15. Our contraction theorems have shown that the unperturbed and noise-perturbed trajectories converge toward each other in expectation. It is weaker than the almost-sure sense of convergence, meaning we do not guarantee trajectory convergence for every noise process sample paths ω. For a more comprehensive treatment of this topic, see [72], [1], and Chapter 2 of [65].

2.5 Numerical Simulations

2.5.1 1D Linear Reference-Tracking

We use the Shot Noise Stochastic Contraction Theorem to derive a stability bound for a simple linear system perturbed by shot noise, as a further specialization to the LTV system from Section 2.4.3. Suppose we have the following scalar system, which can be viewed as the *Ornstein-Uhlenbeck process* [114] augmented with shot noise instead of the usual white noise:

$$dx(t) = ax(t)dt + u(t)dt + \xi(t)dN(t), \tag{2.99}$$

where $a > 0$ so that it is unstable in open-loop, $N(t)$ is a standard Poisson process with rate $\lambda > 0$, and jump height distribution $\xi(t)$ is a Bernoulli random variable which takes value $\eta > 0$ with probability p, and $-\eta$ with probability $q \triangleq 1 - p$.

We are interested in the problem of tracking some given reference trajectory $x_r(t)$. To achieve this, we design the following control law:

$$u_r(t) = \dot{x}_r(t) - ax_r(t), \qquad u(t) = u_r(t) - \gamma(x(t) - x_r(t)), \tag{2.100}$$

with control gain $\gamma > a$. As described in the beginning of this chapter, the dot notation of $\dot{x}_r(t)$ refers to the time-derivative of $x_r(t)$. The system (2.99) can be solved directly:

$$x(t) = x_0 e^{(a-\gamma)t} + \int_0^t \left(u_r(s) + \gamma x_r(s)\right) e^{(a-\gamma)(t-s)} ds + \sum_{i=1}^{N(t)} \xi(T_i) e^{(a-\gamma)(t-T_i)}, \tag{2.101}$$

where $x_0 \in \mathbb{R}$ is the initial condition. The nominal closed-loop system has the dynamics $dy(t) = ay(t) + u(t)$, with the same $u(t)$ as in (2.100). A trajectory of this nominal system with initial condition $y_0 \in \mathbb{R}$ is thus given by

$$y(t) = y_0 e^{(a-\gamma)t} + \int_0^t \left(u_r(s) + \gamma x_r(s)\right) e^{(a-\gamma)(t-s)} ds. \tag{2.102}$$

Like (2.72), we design a virtual system by stacking the nominal closed-loop system on top of the noise-perturbed closed-loop system (2.99), and define $\mathbf{q} = (x, y)^\top \in \mathbb{R}^2$ to be the virtual system state. The contraction metric $P(t)$ from Assumption 5 is chosen to be the identity I_2, meaning $\alpha_1 = \alpha_2 = 1$. The Lyapunov-like function is chosen to be $V(\mathbf{q}) = (y - x)^2$.

Following an argument similar to the proof of Theorem 6 with $t_0 = 0$, we get:

$$\mathbb{E}_k[V(\mathbf{q}(t))] - \mathbb{E}_k[V(\mathbf{q}_0)] \leq$$
$$2(a - \gamma) \int_{0+}^{t} \mathbb{E}_k[V(\mathbf{q}(s))]ds + \mathbb{E}_k \left[\sum_{i=1}^{k} (y - x)^2(T_i) - (y - x)^2(T_i-) \right], \tag{2.103}$$

conditioned on the number of jumps being $N(t) = k$ by time t. Here, the last term can be simplified by using (2.101) and (2.102):

$$(y - x)^2(T_i) - (y - x)^2(T_i-) = -2(y_0 - x_0)e^{(a-\gamma)T_i}\xi(T_i) + 2\xi(T_i) \sum_{j=1}^{i-1} \xi(T_j)e^{(a-\gamma)(T_i - T_j)} + \xi^2(T_i). \tag{2.104}$$

Substituting (2.104) into (2.103) and using Theorem 6 yields the following bound with probability $p_k(t)$ given by (2.61)

$$\mathbb{E}_k \left[|y(t) - x(t)|^2 \right] \leq \mathbb{E}_k \left[|y_0 - x_0|^2 \right] e^{-\beta_s t} + \kappa_s(\beta_s, 0, t), \tag{2.105}$$

where the contraction rate is $\beta_s \triangleq 2(\gamma - a) > 0$ and the error bound $\kappa_s(\beta_s, 0, t)$ comes from (2.63b).

We simulate (2.99) with the lower-level tracking controller (2.100) implemented to track $x_r(t) \triangleq \sin(t)$. We use the theoretical bound created for the each of the two different versions of ψ_k, (2.88) and (2.90). The results are organized in Figure 2.2. Both figures share the following experiment setup. The number of jumps is fixed to be $k = 5$, and we consider the evolution of trajectories starting from $t_0 = 0$. The intensity of the shot noise process varies across $\lambda \in \{1, 2, 4\}$, corresponding to each row of subfigures. We simulate the system until just before the $(k + 1)$th jump occurs at time T_{k+1}. In the left column of subfigures, we simulate a sample nominal closed-loop reference trajectory $y(t)$ (gray line) with a sample closed-loop noise-perturbed reference trajectory $x(t)$ (black line). The initial conditions x_0 and y_0 are sampled uniformly in the range $[1, 6]$. In the middle column of subfigures, the empirical squared-difference trajectory $|y(t) - x(t)|^2$ for the specific $x(t)$ and $y(t)$ from

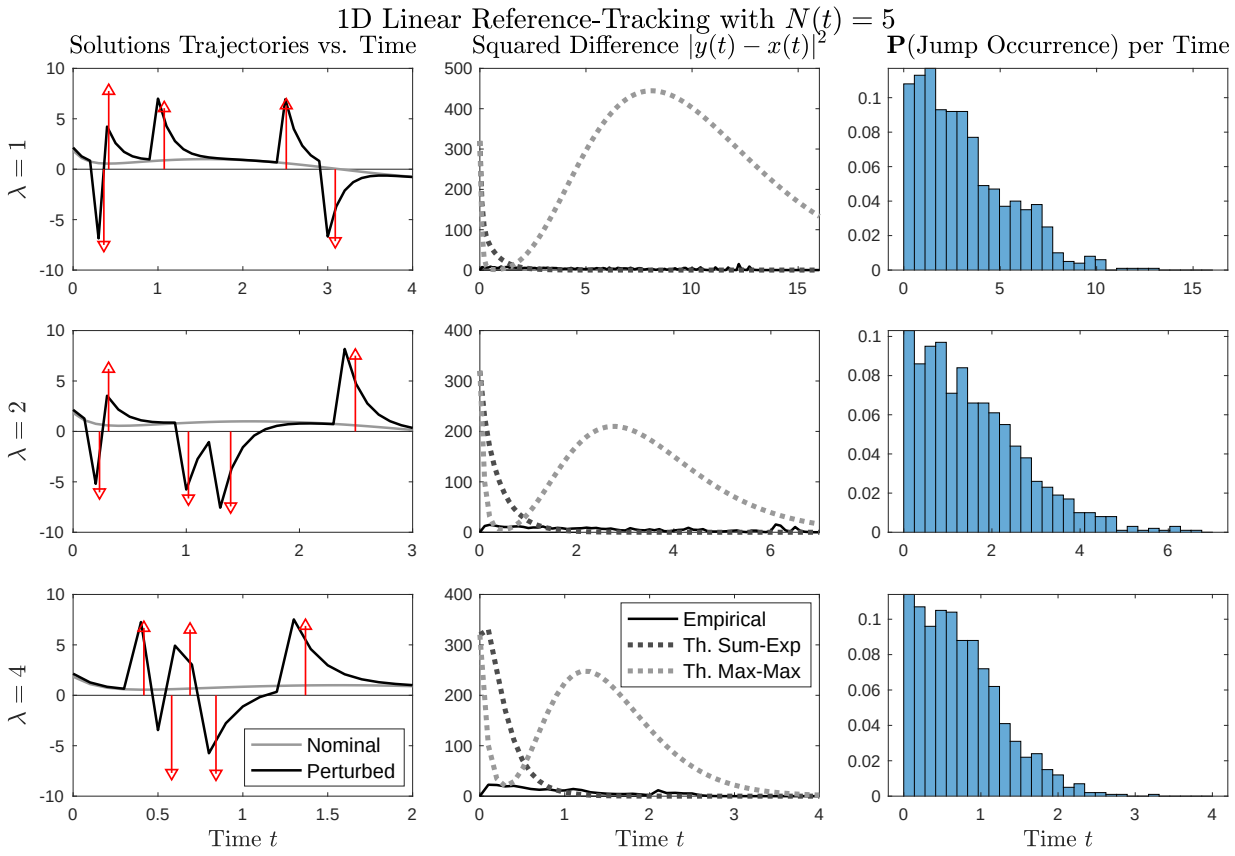

Figure 2.2: Simulation of the 1D linear reference-tracking system over values of $\lambda \in \{1, 2, 4\}$, constrained to $k = 5$ jumps. [Left column] A sample trajectory of the nominal system (light grey line) and a trajectory of the shot noise perturbed system (black line). Jump occurrences and heights are marked with red stems. [Middle column] The empirical mean squared-difference is obtained by taking the timewise average over all Monte-Carlo trials of trajectory squared-differences. Two theoretical bounds are computed, one using $d\psi_k/dt$ from (2.90) (dark grey dashed line), one using $d\psi_k/dt$ from (2.88) (light grey dashed line). [Right column] The empirical probability that a jump occurs at a certain time is plotted over time. The histogram is constructed by discretizing the span of time into subintervals and computing the proportion of jumps over all Monte-Carlo trials which fall into each subinterval.

the left column is plotted (black-dashed line), along with the theoretical mean-squared error bound (2.105) (gray-dashed line). The empirical squared-difference shown in the figures comes from timewise-averaging over 200 Monte-Carlo trials. In the right column of subfigures, the empirical probability that a jump occurs at a certain time is plotted as a histogram over time. The histogram is constructed by discretizing the maximum length of time into 30 subintervals and computing the proportion of jumps (over all Monte-Carlo trials) which fall into each subinterval. We note that it is possible for the empirical squared-difference to exceed the theoretical bound. This is because the bound is on the expected behavior, and should not be treated as an almost-sure guarantee for all sample paths. Moreover, we observe that in both figures, the trajectories converge toward each other in between consecutive jumps, which aligns with the incrementally stable nature of the nominal system.

We can make a few insightful observations based on the two figures. The theoretical bound derived in Section 2.4.3 yields an expression for $\psi_k(0,t)$ which is proportional to both λ and $e^{-\lambda t}$. This effect is demonstrated in both experiments. First, note that an increasing value of λ corresponds to a larger accumulation of jumps. This corresponds to an increasing constant initial value for the theoretical bound, i.e., the gray-dashed line. Second, an increasing value of λ also corresponds to a faster accumulation of jumps, i.e., all $k = 5$ jumps of the system occur earlier in time for larger λ. This corresponds to a faster speed of decay in the first bumps of both grey dashed lines in the middle column of Figure 2.2. Moreover, for the light grey dashed line, the effect of λ is illustrated through the proximity between the line $t = 0$ and the second bump; as λ grows larger, more jumps occur earlier in time, and the second bump occurs closer to $t = 0$.

Another observation is that the second bump which occurs when plotting the theoretical error bound as the light grey dashed line Figure 2.2 arises from the distribution of T_k. The second bump essentially accounts for the possibility of seeing jumps which occur closer to t in the interval $[0, t]$. In contrast, the theoretical error bound with $\psi_k(t)$ as in (2.90) only has the initial bump. All the weight is assigned to the initial value, from which it exponentially decays over time.

2.5.2 2D LTV Systems

In this section, we extend the experiment of Section 2.5.1 by considering more complex 2D LTV shot noise systems of the form:

$$d\mathbf{x}(t) = d\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) = \underbrace{\begin{bmatrix} a_{11}(t) & a_{12}(t) \\ a_{21}(t) & a_{22}(t) \end{bmatrix}}_{=:A(t)} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t)dt + \underbrace{\begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix}}_{=:\xi(t)} dN(t) \tag{2.106}$$

with $\sup_{t>0} \|\xi(t)\| \leq \eta$ for some $\eta > 0$. Note that constructing virtual system (2.72) for 2D dynamics yields virtual system state vector $\mathbf{q}(t) \in \mathbb{R}^4$. For a more practical setup, we can follow the design of the previous 1D example from Section 2.5.1 and consider an open-loop unstable system $A(t)$ with a control law $u(t) \triangleq K(t)\mathbf{x}(t)$ that makes the system exponentially stable. However, for the simplicity of the example, we do not consider the control law design problem, and demonstrate the contraction theorems on systems are already open-loop exponentially stable.

We apply Section 2.4.3 to derive the theoretical mean-squared error bounds for 2D LTV systems with one of two types of matrices $A(t)$: diagonal and (upper) triangular. We note that the conceptual observations for both systems are similar to what is observed in Section 2.5.1: the relationship with increasing λ, and the two different approaches ((2.90) and (2.88)) to simplifying the error bound. Hence, in contrast to the experiment of Section 2.5.1, we will illustrate the results for only one choice of λ and the expression of $\psi_k(t_0, t)$ from (2.90). The purpose of these simulations is to demonstrate the computation of the theoretical bound for more complex LTV systems, especially choosing the metric $P(t)$ and all the appropriate parameters values to satisfy the assumptions of Theorem 6.

Diagonal $A(t)$ Matrix First, consider the case where $A(t)$ is a diagonal matrix, i.e., $a_{12}(t) = a_{21}(t) \equiv 0$ for all $t \geq 0$. Specifically:

$$A(t) \triangleq \begin{bmatrix} -3t^2 - 1 & 0 \\ 0 & -2t - 1 \end{bmatrix}. \tag{2.107}$$

Because $A(t)$ is diagonal, the corresponding state-transition matrix is easily computed:

$$\Phi(t, s) \triangleq \exp\left(\begin{bmatrix} \int_s^t a_{11}(r)dr & 0 \\ 0 & \int_s^t a_{22}(r)dr \end{bmatrix}\right) = \begin{bmatrix} e^{-(t^3+t)+(s^3+s)} & 0 \\ 0 & e^{-(t^2+t)+(s^2+s)} \end{bmatrix}. \tag{2.108}$$

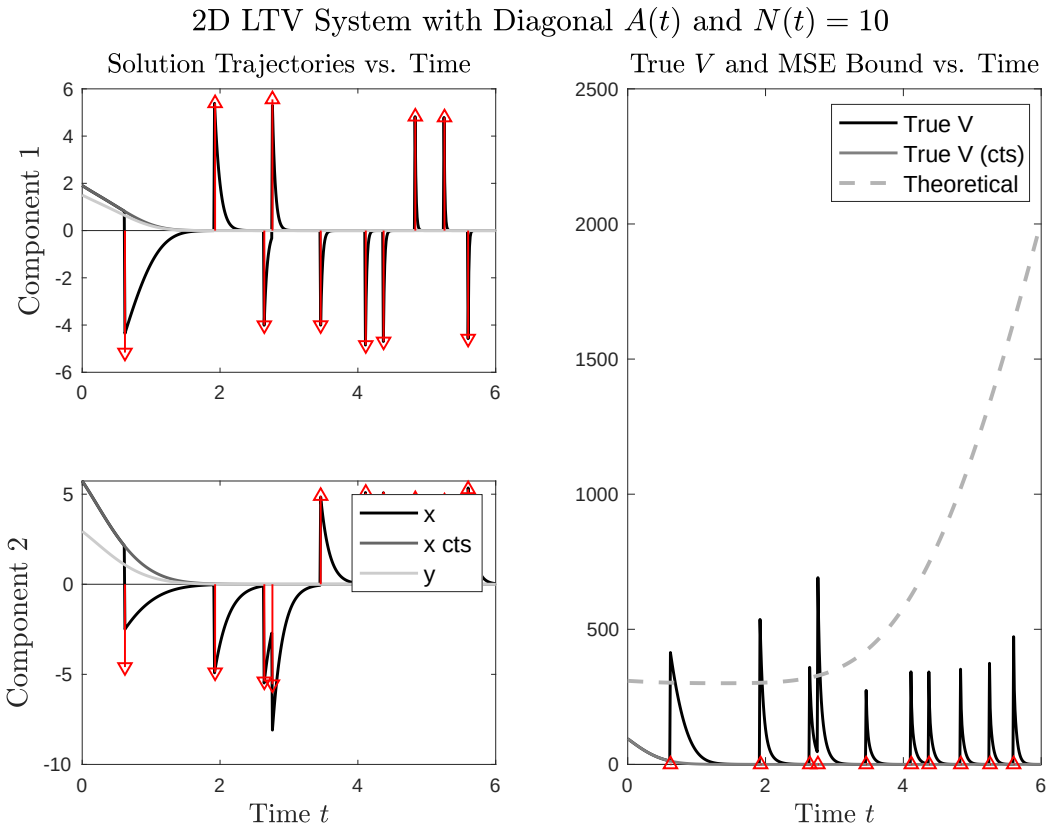

Figure 2.3: Behavior of the exponentially stable 2D diagonal LTV system with bounded $P(t)$ metric. The number of jumps is fixed at $N(t) = 10$. [Left column] Evolution of solution trajectories $\mathbf{x}(t)$, $\mathbf{x}^c(t)$, and $\mathbf{y}$ over time. [Right] Evolution of Lyapunov-like functions V, V^c, and the theoretical bound over time.

One choice of parameters such that Assumption 6 is satisfied with the induced 2-norm is when $\kappa = 1$ and $\beta = 1$.

Solution trajectories for the perturbed and nominal systems can now be written explicitly as follows. From (2.71), we have the perturbed system trajectory $\mathbf{x}(t)$ and the nominal trajectory $\mathbf{y}(t)$ given by:

$$\mathbf{x}(t) = \begin{bmatrix} e^{-(t^3+t)}x_{0,1} \\ e^{-(t^2+t)}x_{0,2} \end{bmatrix} + \sum_{i=1}^{N(t)} \begin{bmatrix} e^{-(t^3+t)+(T_i^3-T_i)}\xi_1(T_i) \\ e^{-(t^2+t)+(T_i^2-T_i)}\xi_2(T_i) \end{bmatrix}, \quad \mathbf{y}(t) = \begin{bmatrix} e^{-(t^3+t)}y_{0,1} \\ e^{-(t^2+t)}y_{0,2} \end{bmatrix}, \tag{2.109}$$

with initial conditions $\mathbf{x}_0, \mathbf{y}_0 \in \mathbb{R}^2$, respectively.

Note that one such choice of metric P which satisfies (2.74) is:

$$P(t) \triangleq \begin{bmatrix} \sin(t)+3 & 0 \\ 0 & \cos(t)+3 \end{bmatrix}, \tag{2.110}$$

with $\alpha = 2$. Clearly, $P(t)$ is positive definite for each $t > 0$, and satisfies the boundedness inequality in Assumption 5 with $\alpha_1 = 2$ and $\alpha_2 = 4$. Hence, the Lyapunov-like function (2.75) yields:

$$\begin{aligned}
V(t,\mathbf{q}(t)) &= (\sin(t)+3)(y_1(t)-x_1(t))^2 + (\cos(t)+3)(y_2(t)-x_2(t))^2 \\
&= e^{-2(t^3+t)}(\sin(t)+3)\left(y_{0,1}-x_{0,1}-\sum_{i=1}^{N(t)} e^{T_i^3+T_i}\xi_1(T_i)\right)^2 \\
&\quad + e^{-2(t^2+t)}(\cos(t)+3)\left(y_{0,2}-x_{0,2}-\sum_{i=1}^{N(t)} e^{T_i^2+T_i}\xi_2(T_i)\right)^2 \\
&= V^c(t,\mathbf{q}(t)) + e^{-2(t^3+t)}(\sin(t)+3)\left\{2(y_{0,1}-x_{0,1})\sum_{i=1}^{N(t)} e^{T_i^3+T_i}\xi_1(T_i) + \left(\sum_{i=1}^{N(t)} e^{T_i^3+T_i}\xi_1(T_i)\right)^2\right\} \\
&\quad + e^{-2(t^2+t)}(\cos(t)+3)\left\{2(y_{0,2}-x_{0,2})\sum_{i=1}^{N(t)} e^{T_i^2+T_i}\xi_2(T_i) + \left(\sum_{i=1}^{N(t)} e^{T_i^2+T_i}\xi_2(T_i)\right)^2\right\},
\end{aligned} \tag{2.111}$$

where the continuous part of the Lyapunov function $V^c(t,\mathbf{q}(t))$ is defined as

$$\begin{aligned}
V^c(t,\mathbf{q}(t)) &\triangleq (\sin(t)+3)(y_1(t)-x_1^c(t))^2 + (\cos(t)+3)(y_2(t)-x_2^c(t))^2 \\
&= e^{-2(t^3+t)}(\sin(t)+3)(y_{0,1}-x_{0,1})^2 + e^{-2(t^2+t)}(\cos(t)+3)(y_{0,2}-x_{0,2})^2,
\end{aligned} \tag{2.112}$$

and $\mathbf{x}^c(t) \triangleq \Phi(t,0)\mathbf{x}_0$ denotes the continuous part of the perturbed trajectory $\mathbf{x}(t)$. We can again compute the error bound on $V(t,\mathbf{q}(t))$ using the argument of Theorem 6. Conditioning on the number of jumps by time t being $N(t) = k$, we obtain:

$$\mathbb{E}_k[\|\mathbf{y}(t)-\mathbf{x}(t)\|^2] \leq \frac{1}{2}\mathbb{E}_k[\|\mathbf{y}_0-\mathbf{x}_0\|^2]e^{-\beta_s t} + \frac{\kappa_s(\beta_s,0,t)}{2}, \tag{2.113}$$

where $\beta_s \triangleq 2\alpha = 4$ and κ_s is as in (2.78b) with $\psi_k(0,t)$ defined as (2.79). We use the simplification of $\psi_k(0,t)$ using the sum of exponentials bound (2.90), and substitute $k = 10, \alpha_2 = 4, \beta = 1, \kappa = 2$. The computation follows similarly to that of Section 2.5.1 and will not be repeated here for conservation of space. The 1) evolution of V, V^c, the theoretical upper bound on $\mathbb{E}_{10}[\|\mathbf{y}(t)-\mathbf{x}(t)\|]$, and 2) the evolution of the trajectories $\mathbf{x}$, $\mathbf{x}^c$, and $\mathbf{y}$ over time are visualized in Figure 2.3.

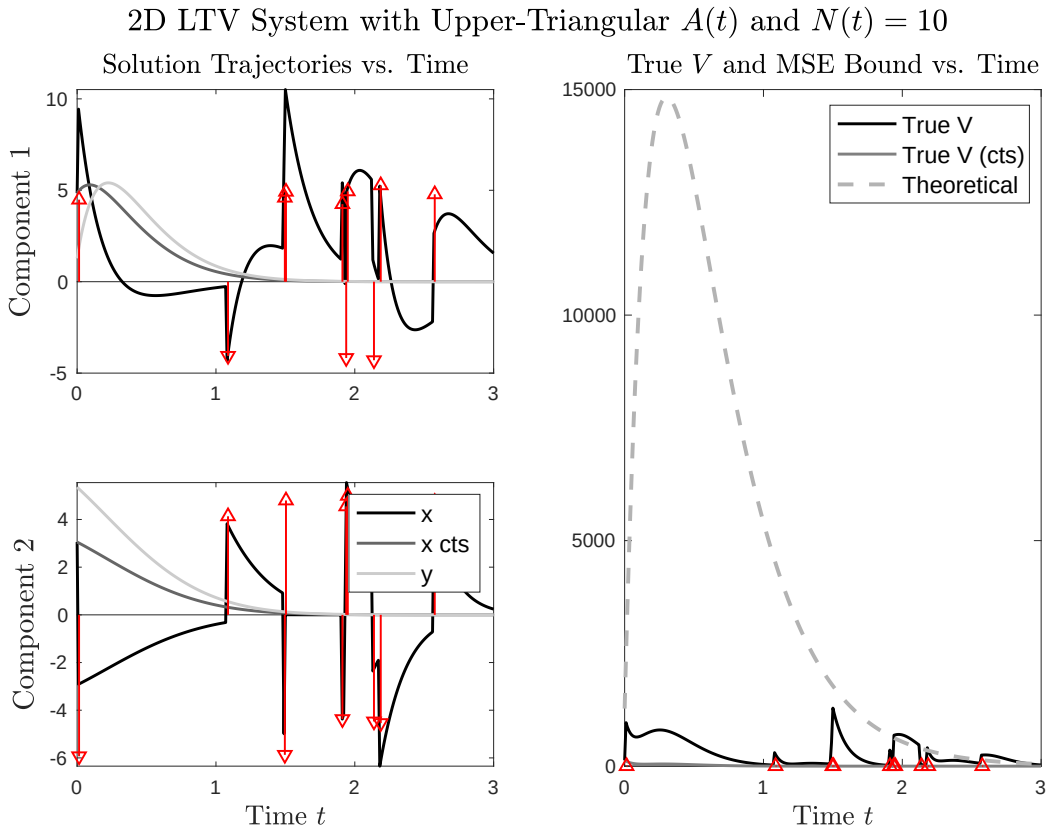

Figure 2.4: Figure 2.3 for the exponentially stable 2D upper-triangular LTV system with bounded $P(t)$ metric. The number of jumps is fixed at $N(t) = 10$.

Triangular $A(t)$ Matrix Now consider instead the case where $A(t)$ is an upper-triangular matrix, i.e., $a_{21}(t) \equiv 0$. Specifically, let

$$A(t) = \begin{bmatrix} -\cos(t) - 5 & 10 \\ 0 & -2t - 1 \end{bmatrix}, \tag{2.114}$$

which has state-transition matrix

$$\begin{aligned} \Phi(t,s) &\triangleq \begin{bmatrix} e^{(\sin(t)-5t)-(\sin(s)-5s)} & 0 \\ 0 & e^{-(t^2+t)+(s^2+s)} \end{bmatrix} \cdot \exp\left(\begin{bmatrix} 0 & 10(t-s) \\ 0 & 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} e^{(\sin(t)-5t)-(\sin(s)-5s)} & 0 \\ 0 & e^{-(t^2+t)+(s^2+s)} \end{bmatrix} \left(I_2 + \begin{bmatrix} 0 & 10(t-s) \\ 0 & 0 \end{bmatrix} \right) \quad \text{by Taylor expansion} \\ &= \begin{bmatrix} e^{(\sin(t)-5t)-(\sin(s)-5s)} & 10(t-s)e^{(\sin(t)-5t)-(\sin(s)-5s)} \\ 0 & e^{-(t^2+t)+(s^2+s)} \end{bmatrix}. \end{aligned} \tag{2.115}$$

This allows us to write the solution trajectories (2.71) for the perturbed ($\mathbf{x}(t)$) and nominal ($\mathbf{y}(t)$) systems explicitly as follows:

$$\begin{aligned}\mathbf{x}(t) &= \begin{bmatrix} e^{b_1(t)}(x_{0,1} + 10tx_{0,2}) \\ e^{b_2(t)}x_{0,2} \end{bmatrix} + \sum_{i=1}^{N(t)} \begin{bmatrix} e^{b_1(t)-b_1(T_i)}(\xi_1(T_i) + 10(t-T_i)\xi_2(T_i)) \\ e^{b_2(t)-b_2(T_i)}\xi_2(T_i) \end{bmatrix}, \\ \mathbf{y}(t) &= \begin{bmatrix} e^{b_1(t)}(y_{0,1} + 10ty_{0,2}) \\ e^{b_2(t)}y_{0,2} \end{bmatrix}, \end{aligned} \tag{2.116}$$

where $b_1(t) \triangleq \sin(t) - 5t$ and $b_2(t) \triangleq -(t^2 + t)$.

In contrast to the previous diagonal $A(t)$ case, we now consider a symmetric $P(t)$ with nonzero off-diagonal elements. Namely:

$$P(t) \triangleq \begin{bmatrix} \sin(t) + 3 & 1 \\ 1 & \cos(t) + 3 \end{bmatrix}, \tag{2.117}$$

which satisfies (2.74) with $\alpha = 2$. This choice of $P(t)$ is also uniformly positive definite for each $t > 0$, and $P(t)$ is bounded as in Assumption 5 with $\alpha_2 = 4.7071$ and $\alpha_1 = 1.2929$. We can now construct the Lyapunov-like function (2.75) for this particular system. Compared to the previous diagonal matrix case, the inclusion of cross-terms make computation a little trickier.

$$\begin{aligned} V(t, \mathbf{q}(t)) &\triangleq (\mathbf{y}(t) - \mathbf{x}(t))^T P(t)(\mathbf{y}(t) - \mathbf{x}(t)) \\ &= V^c(t, \mathbf{q}) - 2(\sin(t) + 3)z_1(t)d_1(t) - 2(\cos(t) + 3)z_2(t)d_2(t) \\ &\quad - 2\left[z_2(t)d_1(t) + z_1(t)d_2(t) - d_1(t)d_2(t)\right] + (\sin(t) + 3)d_1^2(t) + (\cos(t) + 3)d_2^2(t), \end{aligned} \tag{2.118}$$

where

$$\begin{aligned} V^c(t, \mathbf{q}(t)) &\triangleq (\mathbf{y}(t) - \mathbf{x}^c(t))^T P(t) (\mathbf{y}(t) - \mathbf{x}^c(t)) \\ &= (\sin(t) + 3)z_1^2(t) + 2z_1(t)z_2(t) + (\cos(t) + 3)z_2^2(t), \end{aligned} \tag{2.119}$$

with $\mathbf{x}^c(t) \triangleq \Phi(t, 0)\mathbf{x}_0$ being the continuous part of the perturbed solution $\mathbf{x}(t)$, and

$$\begin{aligned} z_1(t) &\triangleq e^{b_1(t)} \left(y_{0,1} + 10ty_{0,2} - x_{0,1} - 10tx_{0,2}\right), \quad z_2(t) \triangleq e^{b_2(t)} \left(y_{0,2} - x_{0,2}\right) \\ d_1(t) &\triangleq \sum_{i=1}^{N(t)} e^{b_1(t)-b_1(T_i)} \left(\xi_1(T_i) + 10(t - T_i)\xi_2(T_i)\right), \quad d_2(t) \triangleq \sum_{i=1}^{N(t)} e^{b_2(t)-b_2(T_i)}\xi_2(T_i). \end{aligned}$$

Instead of dealing with this explicit version of the Lyapunov-like function, we can use the bound from Theorem 6. When the induced 2-norm is used, state-transition matrix $\Phi(t,s)$ is bounded for a choice of $\kappa = 5$ and $\beta = 0.5$. Conditioning on the number of jumps by time t being $N(t) = k$, we obtain:

$$\mathbb{E}_k[\|\mathbf{y}(t) - \mathbf{x}(t)\|^2] \leq \frac{1}{1.2929} \left[\mathbb{E}_k[\|\mathbf{y}_0 - \mathbf{x}_0\|^2]\right]^+ e^{-\beta_s t} + \frac{\kappa_s(\beta_s, 0, t)}{1.2929}, \tag{2.120}$$

where $\beta_s \triangleq 2\alpha = 4$ and κ_s is as in (2.78b) with $\psi_k(0,t)$ defined as (2.79) with the simplified version as in (2.90). Substitute $k = 10, \alpha_2 = 4.7071, \beta = 0.5, \kappa = 5$ to obtain the bound. The 1) evolution of V, V^c, the theoretical upper bound on $\mathbb{E}_{10}[\|\mathbf{y}(t) - \mathbf{x}(t)\|]$, and 2) the evolution of the trajectories $\mathbf{x}$, $\mathbf{x}^c$, and $\mathbf{y}$ over time are visualized in Figure 2.4.

2.5.3 2D Nonlinear System

We demonstrate the general bound from the Shot Noise Stochastic Contraction Theorem Theorem 5 by considering a 2D nonlinear shot noise system:

$$d\mathbf{x}(t) = d\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) = \underbrace{\begin{bmatrix} f_1(t,\mathbf{x}) \\ f_2(t,\mathbf{x}) \end{bmatrix}}_{f(t,\mathbf{x})} dt + \underbrace{\begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix}}_{=:\xi(t)} dN(t), \tag{2.121}$$

with $\sup_{t>0} \|\xi(t)\| \leq \eta$, for some $\eta > 0$.

We consider a system which is already open-loop exponentially stable for reasons of simplicity. We also consider a nonlinear function $f(t,\mathbf{x})$ such that its Jacobian matrix F turns into a diagonal matrix, and such that (2.36) is satisfied with a state-dependent $M(t,\mathbf{x})$. One such dynamics is given by

$$f(t,\mathbf{x}) \triangleq \begin{bmatrix} -(2t+1)(\sin(x_1)+3)x_1 \\ -(2t+1)(\cos(x_2)+5)x_2 \end{bmatrix}, \tag{2.122}$$

which has diagonal matrix Jacobian

$$\begin{aligned} F(t,\mathbf{x}) &\triangleq \nabla_{\mathbf{x}} f(t,\mathbf{x}) \\ &= \begin{bmatrix} -(2t+1)\left[x_1\cos(x_1) + \sin(x_1) + 3\right] & 0 \\ 0 & -(2t+1)\left[-x_2\sin(x_2) + \cos x_2 + 5\right]. \end{bmatrix} \end{aligned} \tag{2.123}$$

Choose metric

$$M(t,\mathbf{x}) \triangleq \begin{bmatrix} (\sin(t)+3)(e^{-x_1^2}+1) & 0 \\ 0 & (\cos(t)+3)(e^{-x_2^2}+1) \end{bmatrix}, \tag{2.124a}$$

$$\partial_t M(t,\mathbf{x}) = \begin{bmatrix} \cos(t)(e^{-x_1^2}+1) & 0 \\ 0 & -\sin(t)(e^{-x_2^2}+1) \end{bmatrix}, \tag{2.124b}$$

$$\nabla_{\mathbf{x}} M(t,\mathbf{x}) = \begin{bmatrix} -2x_1(\sin(t)+3)(e^{-x_1^2}+1) & 0 \\ 0 & -2x_2(\cos(t)+3)(e^{-x_2^2}+1) \end{bmatrix}, \tag{2.124c}$$

with $\dot{M}(t,\mathbf{x}) \triangleq \partial_t M(t,\mathbf{x}) + \nabla_{\mathbf{x}} M(t,\mathbf{x}) \cdot \dot{\mathbf{x}}(t)$.

We empirically generate trajectories with initial condition $\mathbf{x}_0 \sim U[1,6]$ up until the maximum time t where the number of jumps is $N(t) = 3$ such that (2.36) is satisfied and that $M(t,\mathbf{x})$ is uniformly positive definite with $\overline{m} = 4, \underline{m} = 2$. To construct the virtual system, we choose the specific affine parametrization $\mathbf{q}(\mu,t) = (1-\mu)\mathbf{x}(t) - \mu\mathbf{y}(t), \partial_\mu \mathbf{q}(t) = \mathbf{y}(t) - \mathbf{x}(t)$. Then the Lyapunov-like function (2.46) is written explicitly as:

$$\begin{aligned} V(t,\mathbf{q},\delta\mathbf{q}) &= \int_0^1 \partial_\mu \mathbf{q}^\top M(t,\mathbf{q}(\mu,t))\partial_\mu \mathbf{q} d\mu \\ &= \int_0^1 (\mathbf{y}(t)-\mathbf{x}(t))^\top M\left(t,(1-\mu)\mathbf{x}(t)-\mu\mathbf{y}(t)\right)(\mathbf{y}(t)-\mathbf{x}(t))d\mu \\ &= \int_0^1 \Big[(y_1(t)-x_1(t))^2(\sin(t)+3)\left(e^{-((1-\mu)x_1(t)-\mu y_1(t))^2}+1\right) \\ &\quad + (y_2(t)-x_2(t))^2(\cos(t)+3)\left(e^{-((1-\mu)x_2(t)-\mu y_2(t))^2}+1\right)\Big] d\mu. \end{aligned} \tag{2.125}$$

With the Lyapunov-like function (2.125), each term in the sum of (2.65c) yields:

$$\begin{aligned} &V(T_i,\mathbf{q}(T_i),\delta\mathbf{q}(T_i)) - V(T_i-,\mathbf{q}(T_i-),\delta\mathbf{q}(T_i-)) \\ &= \int_0^1 (\sin(T_i)+3)\Big[(y_1(T_i)-x_1(T_i))^2\left(e^{-((1-\mu)x_1(T_i)-\mu y_1(T_i))^2}+1\right) \\ &\quad - (y_1(T_i)-x_1(T_i-))^2\left(e^{-((1-\mu)x_1(T_i-)-\mu y_1(T_i))^2}+1\right)\Big] d\mu \\ &\quad + \int_0^1 (\cos(T_i)+3)\Big[(y_2(T_i)-x_2(T_i))^2\left(e^{-((1-\mu)x_2(T_i)-\mu y_2(T_i))^2}+1\right) \\ &\quad - (y_2(T_i)-x_2(T_i-))^2\left(e^{-((1-\mu)x_2(T_i-)-\mu y_2(T_i))^2}+1\right)\Big] d\mu \end{aligned} \tag{2.126}$$

for each arrival time $T_i > 0$. Because we cannot compute the explicit form of $\mathbf{x}(t)$ and $\mathbf{y}(t)$ from the dynamics, the bound $h(t)$ from (2.59) is difficult to compute analytically. We plot the evolution of (2.125) over time for multiple sample trajectories of $\mathbf{x}(t)$ and $\mathbf{y}(t)$, and demonstrate a bound empirically. The evolution of some sample solution trajectories

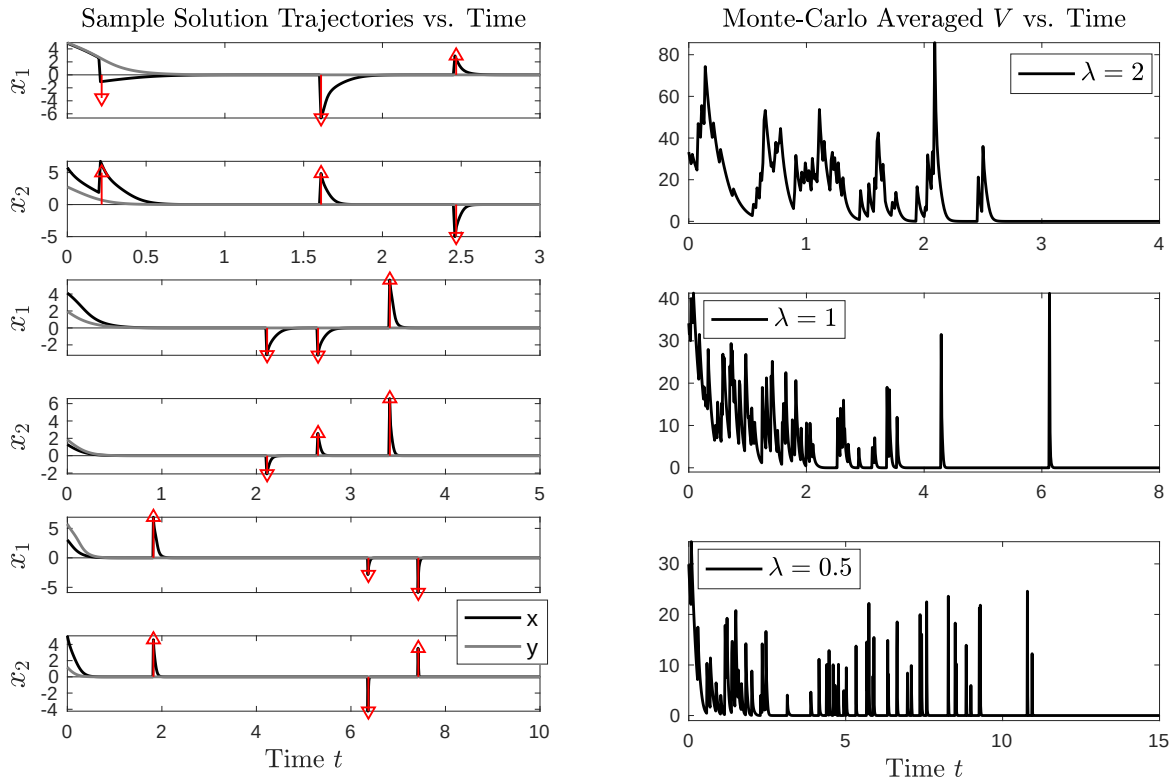

Figure 2.5: Figure 2.3 for varying values of $\lambda \in [0.5, 1, 2]$ for the exponentially stable 2D nonlinear system. [Left] Evolution of a sample solution trajectories $\mathbf{x}$ and $\mathbf{y}$ with the corresponding value of λ over time. [Right] Lyapunov-like function V averaged over 20 Monte-Carlo trials, then plotted over time. The number of jumps is fixed at $N(t) = 3$.

and an empirical average over multiple Monte-Carlo trials for three different values of λ are portrayed in Figure 2.5. We note that as λ increases, the support of V increases over time and the average height of the spikes decreases.

2.5.4 2D Observer Design

In this case study, we look at a stochastic observer design problem for a simple 2D nonlinear system, and derive stability bounds using the Lévy Noise Stochastic Contraction Theorem. Consider the following nominal system

$$d\mathbf{x} \triangleq d\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \underbrace{\begin{bmatrix} ax_1 + bx_2 - x_1(x_1^2 + x_2^2)^{-\frac{1}{2}}(9x_1^2 + 4x_2^2 - R^2) \\ cx_1 + dx_2 - x_2(x_1^2 + x_2^2)^{-\frac{1}{2}}(9x_1^2 + 4x_2^2 - R^2)\end{bmatrix}}_{\triangleq f(t,\mathbf{x})} dt, \tag{2.127}$$

where the constants $a, b, c, d, R \in \mathbb{R}$ are chosen to manipulate the behavior of the system in ways we detail later.

Suppose that we wanted to design an observer for this system based on measurements that are perturbed by Lévy noise. That is, we assume a measurement equation of the following linear form:

$$y = h(t, \mathbf{x}) \triangleq C(t)\mathbf{x} + \sigma(t)W(t) + \xi(t)N(t), \tag{2.128}$$

where $W(t)$ is the standard Brownian motion process, $N(t)$ is the standard Poisson process, and $\sigma(t), \xi(t)$ are their respective variances. In the case of additive white noise, the observer can be designed and related to the white noise contraction theorem according to Section 4.1 of Pham 2009 [121]. We apply a similar methodology for our Lévy noise case, and relate it to Theorem 7. Choose an output injection matrix $K(t)$ such that the observer dynamics

$$d\hat{\mathbf{x}} = (f(t, \hat{\mathbf{x}}) + K(t)(y(t) - \hat{y}(t)))\, dt, \qquad \hat{y}(t) \triangleq h(t, \hat{\mathbf{x}}) = C(t)\hat{\mathbf{x}} + \sigma(t)W(t) + \xi(t)N(t)$$

is stochastically contracting. The full observer dynamics can be written as follows:

$$d\hat{\mathbf{x}}(t) = [f(t, \hat{\mathbf{x}}) + K(t)C(t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t))]\, dt + K\sigma(t)dW(t) + K\xi(t)dN(t). \tag{2.129}$$

The objective of any state-estimation problem is to ensure the estimate $\hat{\mathbf{x}}$ quickly tracks the true state $\mathbf{x}$. To this end, we are interested in achieving global exponential convergence of $\hat{\mathbf{x}}$ towards $\mathbf{x}$. We can apply Theorem 7 to obtain a probabilistic bound on $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$.

Note that a trajectory $\mathbf{y}(t)$ of the original dynamics (2.127) satisfies the noiseless version of (2.129), given by $d\mathbf{y} = [f(t, \mathbf{y}) + K(t)C(t)(\mathbf{x} - \mathbf{y})]\, dt$. Hence, we can use the form of (2.129) to construct a virtual system:

$$d\mathbf{q}(\mu, t) = [f(t, \mathbf{q}(\mu, t)) + K(t)C(t)(\mathbf{x}(t) - \mathbf{q}(\mu, t))]\, dt + \sigma_\mu(t)dW(t) + \xi_\mu(t)dN(t), \tag{2.130}$$

with state $\mathbf{q}$ parametrized by $\mu \in [0, 1]$ such that $\mathbf{q}(\mu = 0, t) = \hat{\mathbf{x}}(t)$ and $\mathbf{q}(\mu = 1, t) = \mathbf{x}(t)$. The rest of the parametrization follows in a way similar to (2.40).

Thus, by combining Theorem 7 and Section 4.1 of Pham 2009 [121] using the Lyapunov-like function $V(t, \mathbf{q}) \triangleq (\mathbf{x} - \hat{\mathbf{x}})^\top K^\top(t)K(t)(\mathbf{x} - \hat{\mathbf{x}})$, we get that the bound (2.94) is satisfied with probability $p_k(t)$ given by (2.61), where

$$\beta_\ell = \inf_{t, \mathbf{q}} |\lambda_{\max}(\nabla_{\mathbf{q}} f(t, \mathbf{q}) - K(t)C(t))|, \tag{2.131}$$

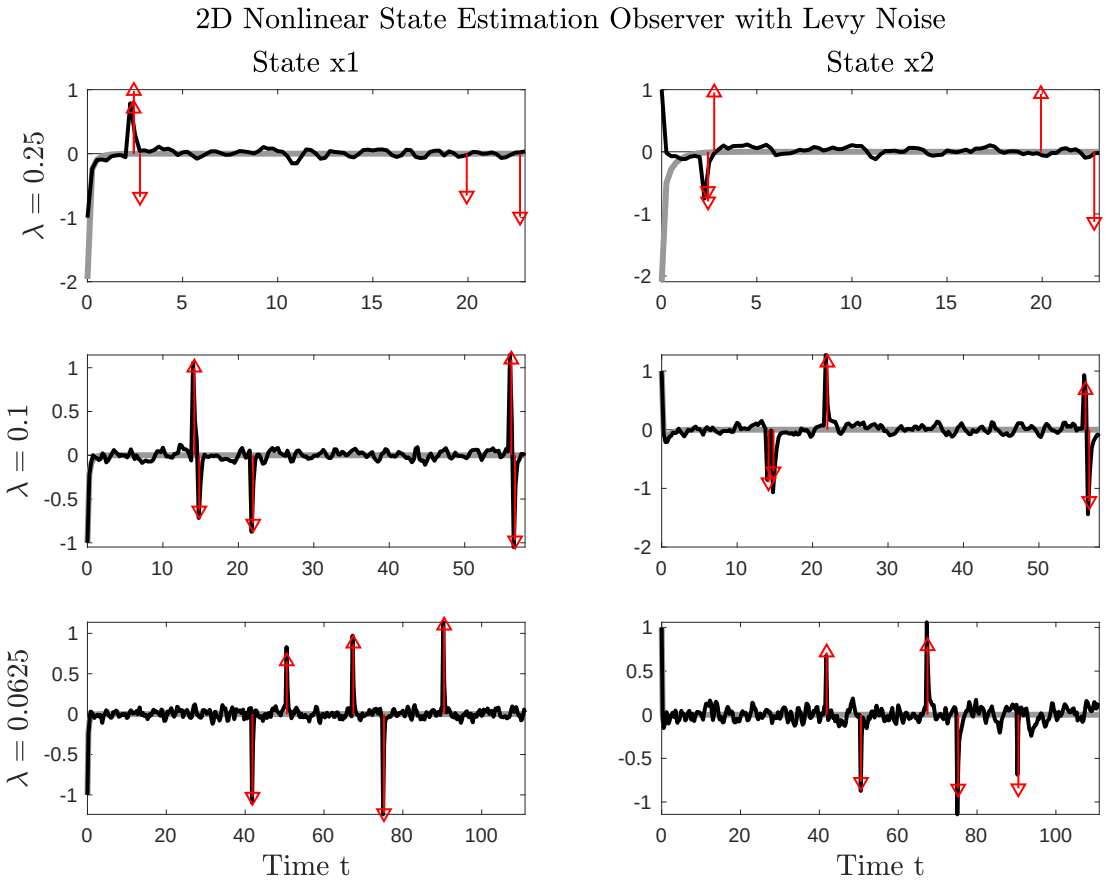

Figure 2.6: Three different noisy observers with three sample noise processes across varying values of λ, estimating the state of a reference trajectory (gray) of the 2D system. The observers' estimates (black) are overlaid on top of the reference state, and the jumps of the Poisson shot noise are indicated with red stems.

$$\kappa_\ell(\beta_\ell, 0, t) \triangleq k \int_{0+}^{t} \frac{dh(\tau)}{d\tau} e^{-\beta_\ell(t-\tau)} d\tau + kh(\tau)e^{-\beta_\ell t} + \frac{1}{\beta_\ell} \sup_{t\geq 0} tr(\sigma^\top(t)K^\top(t)K(t)\sigma(t)) \left(1 - e^{-\beta_\ell t}\right). \tag{2.132}$$

As seen in Section 2.5.3, it is easier to derive forms of the $h(t)$ from Assumption 4 using empirical methods. We simulate (2.127) for various parameter values, and we choose constant matrices $C \triangleq [1, 0]$ (i.e., we only observe noise-perturbed measurements of the first component x_1) and $K(t) \triangleq (1, 1)^\top$ for simplicity. First, consider $a = d = -2, b = 0, c = 4$, and $R = 0$. Using the direct method of Lyapunov with $V(x) = x_1^2 + x_2^2$ demonstrates that the origin is globally exponentially stable for the nominal system. In the visualizations pro-

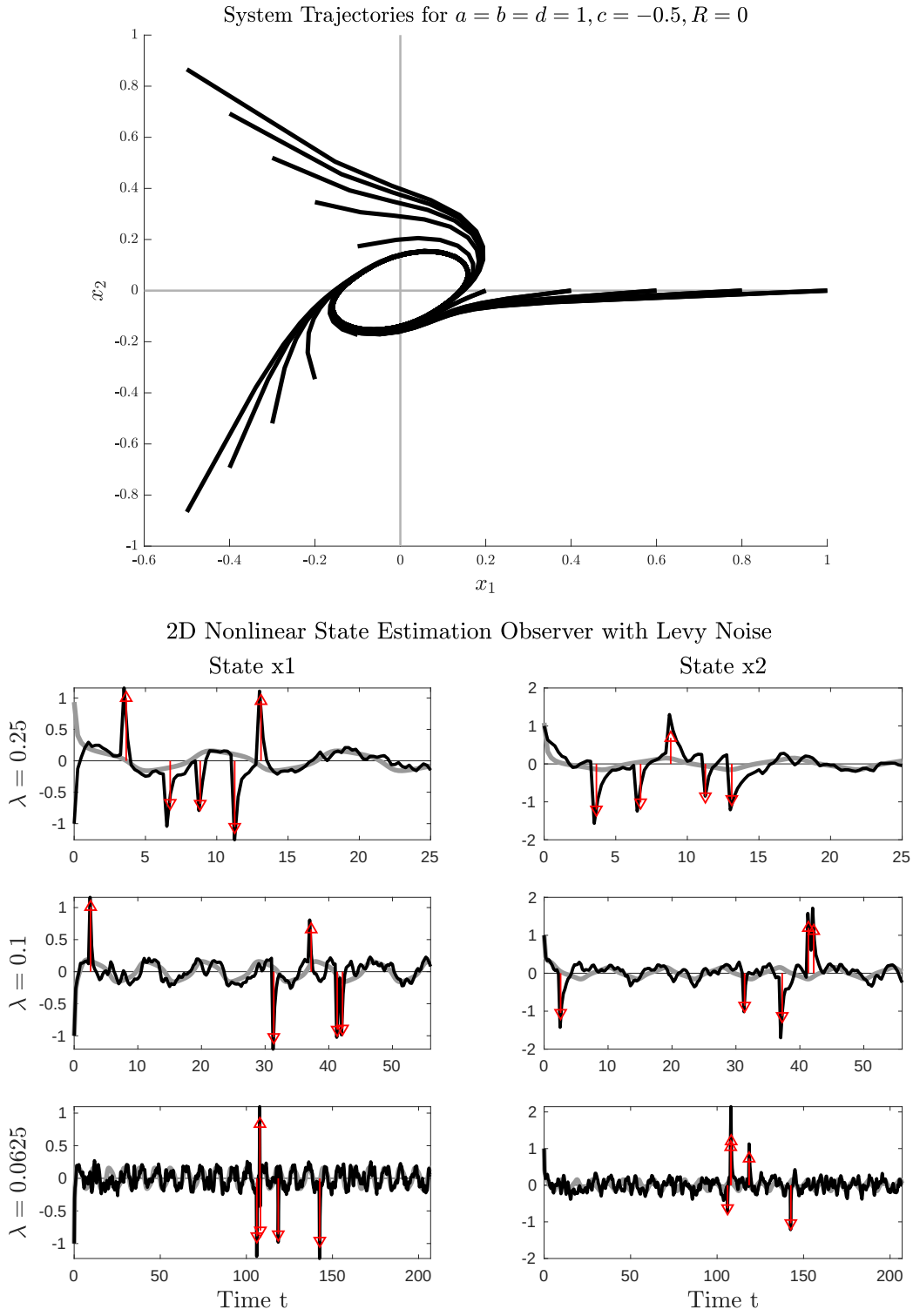

Figure 2.7: Similar graph as in Figure 2.6, but for the case where system parameters are chosen so that nominal trajectories converge globally exponentially to a limit cycle instead.

vided in Figure 2.6, we simulate a sample trajectory of the nominal system (2.127) (gray), then overlay it with a trajectory representing the observer state values perturbed by Lévy noise processes with the shot noise component varied across λ values $1/4, 1/10$, and $1/16$. Moreover, we condition specifically on the number of jumps being $N(t) = 5$. We observe the same trends as seen in the 1D linear system of Section 2.5.1: larger values of λ provide less time for convergence in between consecutive jumps.

Alternatively, when $a = b = d = 1, c = -0.5, R = 0$, the observer can be designed so that all trajectories converge exponentially to a limit cycle. We create a similar visualization for this set of parameters in Figure 2.7, constrained to $N(t) = 2$ jumps.

2.6 Concluding Remarks

In this chapter, we designed incremental stability criteria for nonlinear stochastic systems perturbed by two types of non-Gaussian noise, particularly those characterized by impulsive jumps. The Shot Contraction Theorem (Theorem 5) was designed for compound Poisson shot noise systems of the form (2.8) while the Lévy Contraction Theorem (Theorem 7) was designed for finite-measure Lévy noise systems of the form (2.7). In Theorem 6, a specialization of the Shot Contraction Theorem was presented for linear time-varying nominal dynamics of the form (2.70). All three theorems showed that, under the assumption that a finite number of jumps arise from the noise process over a finite interval of time, solution trajectories corresponding to different initial conditions and different realizations of the noise process converged exponentially to within a bounded error ball of each other in the mean-squared sense under certain practical boundedness conditions on the parameters of the noise process and contraction metric. We've shown that the convergence rate for (2.8) is equal to that of the nominal system $\dot{\mathbf{x}} = f(t, \mathbf{x})$ because the shot noise system behaves exactly as the deterministic system in between consecutive jumps. Remark 11 discussed properties of the error bound $\kappa_s(\beta_s, t)$ defined in (2.63b), and made the claim that 1) larger jump norm bounds η, and 2) shorter interarrival times between jumps correspond to a larger error ball. Furthermore, the convergence rate (2.95a) and error bound (2.95b) of (2.7) were shown to be nearly direct sums of the parameters for the white noise case (2.55) and the shot noise case (2.63), which is similar to the implications of the Lévy-Khintchine theorem. The numerical simulations of Section 2.5 demonstrated our results. The 1D simple linear reference-tracking shot noise system in Section 2.5.1 illustrated the tradeoff of Remark 11 by considering three different intensities of shot noise processes. We also demonstrated how to derive expressions for the theoretical $\kappa_s(\beta_s, t)$ in Sections 2.5.1 and 2.5.2. The computations performed in the proof

of Theorem 5 were also shown for the specific 2D nonlinear system of Section 2.5.3, and the error bound $\kappa_s(\beta_s, t)$ was derived empirically. Section 2.5.4 illustrated the decomposition of the Lévy noise parameters by looking at a Lévy noise stochastic observer design problem for a simple 2D nonlinear system.

We emphasize that the benefits of our work are two-fold. First, the phenomenon of impulsive jumps in noise processes is understudied for nonlinear stochastic systems in the controls community compared to Gaussian white noise despite being prevalent and important for many applications. Second, by addressing the prerequisite problem of stability characterization for shot and Lévy noise systems, we established the foundations to enable model-based design of stochastic controllers and observers that are robust to shot and Lévy noise. By considering a class of noise models broader than the Gaussian assumption, we can expand the capabilities of model-based controller and observer design which consume less training time and data.

LEARNING RECURRENT PATTERNS IN A JUMP PROCESS

Our previous analysis in Chapter 2 explored the possibility of making controller and observer design more efficient in data-consumption and computation time for stochastic systems perturbed by Poisson shot noise and finite-measure Lévy noise. The primary benefit of answering the stability question posed there is that methods of stochastic controller/observer design, previously suitable for only Gaussian white noise systems, can now be expanded to account for JSSs. For many discrete-time/discrete-event JSSs, we can take advantage of the fact that the underlying jump process is a sequence of random variables that occurs as repeated *patterns* of interest over time, space, or both. In fault-tolerance control or manufacturing process applications, for example, a "pattern of interest" may be a specific sequence of modes which represents a critical fault. Another example can be found in queuing-based systems such as vehicle intersection networks, where repetition arises naturally while counting the number of entities in the queue over time. Mathematically, it is common to model these phenomena using *renewal processes* (see Definition 2), which generalize Poisson processes by allowing the interarrival times to be any i.i.d. sequence of random variables not necessarily Exponentially-distributed. A related class of jump processes are *Markov (jump) processes* (see Definition 4), which enforces some dependency between the interarrival time and the magnitude of the jump.

This chapter develops a procedure for learning recurrent patterns in the jump process by addressing *pattern-occurrence problems*. The pattern-occurrence problems learn the properties of the jump process to determine which specific sequences of values should be considered patterns, then solves for two quantities pertaining to their occurrence and recurrence: 1) the expected minimum time until a pattern from the pattern collection appears in the sequence, and 2) the probability that a pattern will be the first among the collection to be observed in the sequence. This way, control policies can be explicitly designed for each pattern, memorized to prevent redundant computation, then scheduled to be applied at the pattern's expected future occurrence time. We refer to this principle as *Pattern-Learning for Prediction (PLP)*. For the purposes of this chapter, we design a PLP-based controller architecture specifically for a class of uncertain linear discrete-time Markovian jump systems (MJSs),

which are then used in some of the applications discussed in the subsequent chapters.

Chapter Organization

In Section 3.1, we set up the pattern-occurrence problem and derive closed-form expressions to the *pattern-occurrence quantities* for two concrete stochastic process distributions: 1) when patterns are generated from an i.i.d. sequence, and 2) when patterns are generated from a Markov chain. In Section 3.2, we set up the specific form of linear, discrete-time Markovian jump system (MJS) used for our demonstration of a concrete implementation of PLP which learns patterns in the underlying mode-switching process. The implementation of the PLP controller architecture for MJSs is described in the subsequent Section 3.3.

3.1 Learning Patterns in Sequences

We begin with the formal definition of patterns and set up the pattern-occurrence quantities as generically as possible. Let $\{\xi_n\}$ be the underlying stochastic process where each random variable $\xi_n : \Omega \to \mathcal{X}$ is defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$ defined by $\mathcal{F}_n \triangleq \sigma(\xi_0, \xi_1, \cdots, \xi_n)$. Here, $\mathcal{X} \triangleq \{\zeta_1, \cdots, \zeta_M\}$ is a discrete, finite set of values from where the stochastic process is generated according to a probability distribution $\xi_n \sim \mathbb{P}(\cdot | \xi_{n-1} = \varphi_{n-1}, \cdots, \xi_1 = \varphi_1, \varphi_0)$. We denote all sequences of the form $\{\cdot\}_{n=1}^{\infty}$ using the shorthand $\{\cdot\}$, e.g., $\{\xi_n\} \equiv \{\xi_n\}_{n=1}^{\infty}$ and $\{\mathcal{F}_n\} \equiv \{\mathcal{F}_n\}_{n=1}^{\infty}$. The letter ξ is specifically reserved to denote random variables in the stochastic process and φ is reserved for the deterministic value it takes. Furthermore, sequences denoted using other Greek letters are deterministic unless explicitly stated otherwise. For any two $n, m \in \mathbb{N}$ such that $n_1 < n_2$, we use MATLAB notation to represent sequences $\xi_{n_1:n_2} \triangleq (\xi_{n_1}, \xi_{n_1+1}, \cdots, \xi_{n_2})$ and likewise $\varphi_{n_1:n_2}$. Also, we denote the concatenation of $\boldsymbol{\alpha} \triangleq (\alpha_1, \cdots, \alpha_a)$ and $\boldsymbol{\beta} \triangleq (\beta_1, \cdots, \beta_b)$ as $\boldsymbol{\alpha} \circ \boldsymbol{\beta} \triangleq (\alpha_1, \cdots, \alpha_a, \beta_1, \cdots, \beta_b)$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are placeholders for either deterministic or random mode sequences.

Definition 13 (Patterns)**.** Define the set $\Psi \triangleq \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_K\}$ to be a *collection of patterns*, where $K \in \mathbb{N}$, each $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,d_k})$ is a sequence with length $d_k \in \mathbb{N}$, and elements $\psi_{k,j} \in \mathcal{X}$. Each $\boldsymbol{\psi}_k$ is referred to as a *pattern* if we are interested in observing its occurrence in the stochastic process $\{\xi_n\}$ over time (e.g., because it models a system fault).

Definition 14 (Pattern-Occurrence Times)**.** For each of the patterns in the collection Ψ (from Definition 13), define the following stopping times for each $k \in \{1, \cdots, K\}$:

$$\tau_{k|n} \triangleq \min\{i \in \mathbb{N} \,|\, \xi_n = \varphi_n, \xi_{n+i-d_k+1:n+i} = \boldsymbol{\psi}_k\}. \tag{3.1}$$

Definition 15 (Time and Probability of First Occurrence)**.** Under the setup of Definition 14 and given the collection Ψ, the 1) *minimum time of occurrence* of any pattern in Ψ and 2) the *first-occurrence probabilities* (i.e., the probability that each pattern $\boldsymbol{\psi}_k$ will be the first observed) are given by:

$$\tau_n \triangleq \min_{k\in\{1,\cdots,K\}} \tau_{k|n}, \qquad q_k \triangleq \mathbb{P}(\tau_n = \tau_{k|n}). \tag{3.2}$$

This means $\xi_{n+\tau_n-d_k+1:n+\tau_n} = \boldsymbol{\psi}_k$ if pattern $\boldsymbol{\psi}_k$ will be the first observed among the entire collection Ψ.

Problem 1 (Pattern-Occurrence)**.** We are interested in characterizing the following *pattern-occurrence quantities* described in Definition 15:

- the estimate $\mathbb{E}[\tau_n]$ of the *mean minimum occurrence time*, which counts the number of timesteps to observe the occurrence of any pattern from Ψ, given the estimated current value at time $n \in \mathbb{N}$ is φ_n;
- the estimated *first-occurrence probabilities* $\{q_k\}$, where $q_k \in [0,1]$ is the probability that pattern $\boldsymbol{\psi}_k \in \Psi$ is the first to be observed among all of Ψ.

Finally, we restate two fundamental results that are commonly used in the analyses of martingales, and will be used in the derivation of our pattern-occurrence quantities. For further results and properties, we refer the reader to standard probability texts including [49], [57], and [18].

Lemma 8 (Martingale Convergence)**.** Let $\{M_n\}_{n=1}^{\infty}$ be a martingale on the filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_n\}, \mathbb{P})$, and let τ be a stopping time. Denote $\Omega_n \triangleq \{\omega \in \Omega : |\tau(\omega)| > n\}$. If $\mathbb{E}[M_\tau] < \infty$ and $\mathbb{E}[M_{n\wedge\tau}] < \infty$, then M_n is uniformly-integrable, i.e.,

$$\liminf_{n\to\infty} \int_{\Omega_n} |M_n(\omega)| d\mathbb{P}(\omega) = 0.$$

Lemma 9 (Optional Stopping Theorem)**.** Consider a continuous, real-valued, cádlág (right-continuous, left-limit) random process $M(t)$ adapted to the filtration $\{\mathcal{F}(s), s \le t\}$. Let $\tau \ge 0$ be a stopping time for M. Suppose one of the following three conditions holds:

- τ is almost-surely (a.s.) bounded, i.e. there exists a fixed time $T \in \mathbb{R}^{\ge 0}$ such that $\tau \le T$ a.s.;

- $\mathbb{E}[\tau] < \infty$ and there exists a constant $C \geq 0$ such that $\mathbb{E}[|M(t) - M(s)| \mid \mathcal{F}(s)] \leq C$ a.s. on the event that $\{\tau > s\}$;
- there exists a constant c such that $|M(t \wedge \tau)| \leq c$ a.s. for all $t \geq 0$.

Then

- $\mathbb{E}[M(\tau)] = \mathbb{E}[M(0)]$ if $M(t)$ is a martingale;
- $\mathbb{E}[M(\tau)] \geq \mathbb{E}[M(0)]$ if $M(t)$ is a submartingale;
- $\mathbb{E}[M(\tau)] \leq \mathbb{E}[M(0)]$ if $M(t)$ is a supermartingale.

3.1.1 Independent, Identically-Distributed Sequences

The first concrete stochastic process we consider in solving Problem 1 is an *i.i.d. sequence*. We fix $\{\xi_n\}$ to be generated from probability distribution $p(\varphi) \triangleq \mathbb{P}(\xi_n = \varphi)$ for all $n \in \mathbb{N}$ and $\varphi \in \mathcal{X}$. Because of the independence among the random variables, i.e., $\mathbb{P}(\xi_n = \varphi | \xi_{n-1} = \varphi_{n-1}, \cdots, \xi_1 = \varphi_1, \varphi_0) \equiv \mathbb{P}(\xi_n = \varphi)$, we consider the following additional stopping times other than the ones from Definition 15.

Definition 16 (Pattern Occurrence Times for i.i.d. Sequence)**.** For each of the patterns in pattern collection Ψ, define the following stopping times. $\tau_{k|0}$ is the time until the first occurrence of pattern $\boldsymbol{\psi}_k$ from the start of the stochastic process, τ_k is the time between the first and second occurrences of $\boldsymbol{\psi}_k$, and $\tau_{k_2|k_1}$ is the time until the first occurrence of pattern $\boldsymbol{\psi}_{k_2}$ given pattern $\boldsymbol{\psi}_{k_1}$ was already observed, with $\tau_{k|k} = 0$:

$$\begin{aligned} \tau_{k|0} &\triangleq \min\{n \in \mathbb{N} \mid (\xi_n, \cdots, \xi_{n+d_k-1}) = \boldsymbol{\psi}_k\}, \\ \tau_k &\triangleq \min\{n \in \mathbb{N} \mid (\xi_{n+\tau_{k|0}}, \cdots, \xi_{n+\tau_{k|0}+d_k-1}) = \boldsymbol{\psi}_k\}, \\ \tau_{k_2|k_1} &\triangleq \min\{n \in \mathbb{N} \mid (\xi_{n+\tau_{k_1|0}}, \cdots, \xi_{n+\tau_{k_1|0}+d_{k_2}-1}) = \boldsymbol{\psi}_{k_2}\}. \end{aligned} \tag{3.3}$$

Related to the Geometric Distribution

Because $\{\xi_n\}$ are generated i.i.d., the occurrence of any specific pattern $\boldsymbol{\psi}_k$ in $\{\xi_n\}$ can be related to a Geometric distribution using the following case-by-case argument, inspired by [130]. For any $k \in \{1, \cdots, K\}$, we derive $\mathbb{E}[\tau_{k|0}]$ and $\mathbb{E}[\tau_k]$. There are two cases that need to be considered, depending on the amount of self-overlap there is in $\boldsymbol{\psi}_k$, where overlaps are defined as follows.

Definition 17 (Pattern Overlaps)**.** We say that a pattern $\boldsymbol{\psi}_k$ has a *self-overlap* of size $v_k < d_k$ if

$$v_k \triangleq \max\{r < d_k \mid (\psi_{k,1}, \cdots, \psi_{k,r}) = (\psi_{k,d_k-r+1}, \cdots, \psi_{k,d_k})\},$$

and we say that a pattern $\boldsymbol{\psi}_{k_2}$ *has an overlap with* $\boldsymbol{\psi}_{k_1}$ of size $v_{k_2|k_1} < \min(d_{k_1}, d_{k_2})$ if

$$v_{k_2|k_1} \triangleq \max\{r < \min(d_{k_1}, d_{k_2}) \mid (\psi_{k_2,1}, \cdots, \psi_{k_2,r}) = (\psi_{k_1,d_{k_1}-r+1}, \cdots, \psi_{k_1,d_{k_1}})\}.$$

Lemma 10 (Occurrence Time without Self-Overlaps)**.** When there are no self-overlaps in $\boldsymbol{\psi}_k$,

$$\mathbb{E}[\tau_k] = \mathbb{E}[\tau_{k|0}] = \left(\prod_{i=1}^{d_k} p(\psi_{k,i})\right)^{-1}. \tag{3.4}$$

Proof. Since $\prod_{i=1}^{d_k} p(\psi_{k,i})$ is the probability of observing exactly the sequence $\boldsymbol{\psi}_k$, the formula (3.4) arises naturally when treated like a Geometric random variable. The formula for $\mathbb{E}[\tau_k]$ is equivalent to $\mathbb{E}[\tau_{k|0}]$ because there are no self-overlaps; thus, we focus our derivation on $\mathbb{E}[\tau_{k|0}]$. Note that $\tau_{k|0} > n + d_k$ iff the pattern does not occur for the first n values and the next d_k values are $\boldsymbol{\psi}_k$:

$$\{\tau_{k|0} > n + d_k\} \iff \{\tau_{k|0} > n \text{ and } (\xi_{n+1}, \cdots, \xi_{n+d_k}) = \boldsymbol{\psi}_k\}.$$

In terms of probabilities:

$$\mathbb{P}(\tau_{k|0} > n + d_k) = \mathbb{P}(\tau_{k|0} > n)\mathbb{P}((\xi_{n+1}, \cdots, \xi_{n+d_k}) = \boldsymbol{\psi}_k) = \mathbb{P}(\tau_{k|0} > n) \prod_{i=1}^{d_k} p(\psi_{k,i}).$$

By the definition of expected value:

$$1 = \sum_{n=0}^{\infty} \mathbb{P}(\tau_{k|0} > n + m) = \prod_{i=1}^{d_k} p(\psi_{k,i}) \sum_{n=0}^{\infty} \mathbb{P}(\tau_{k|0} > n) = \prod_{i=1}^{d_k} p(\psi_{k,i}) \mathbb{E}[\tau_{k|0}].$$

Dividing through by $\prod_{i=1}^{d_k} p(\psi_{k,i})$ yields (3.4). ∎

Lemma 11 (Occurrence Time with Self-Overlaps)**.** When there is a self-overlap of size $v_k < d_k$ in $\boldsymbol{\psi}_k$, and $\psi_{k,1:v_k}$ does not have a self-overlap itself, $\mathbb{E}[\tau_k]$ is as in (3.4) and

$$\mathbb{E}[\tau_{k|0}] = \left(\prod_{i=1}^{d_k} p(\psi_{k,i})\right)^{-1} + \left(\prod_{i=1}^{v_k} p(\psi_{k,i})\right)^{-1}. \tag{3.5}$$

Proof. Define the augmented pattern $\boldsymbol{\gamma}_k \triangleq \boldsymbol{\psi}_k \circ (\zeta)$ for any $\zeta \in \mathcal{X}$, and define

$$\tau_k^* \triangleq \min\{n \in \mathbb{N} \mid (\xi_{n+\tau_{k|0}^*}, \cdots, \xi_{n+\tau_{k|0}^*+d_k}) = \boldsymbol{\gamma}_k\}$$
$$\text{where } \tau_{k|0}^* \triangleq \min\{n \in \mathbb{N} \mid (\xi_n, \cdots, \xi_{n+d_k}) = \boldsymbol{\gamma}_k\}$$

to be the time it takes to observe the augmented pattern $\boldsymbol{\gamma}_k$ again after its first occurrence. Further define $\Delta\tau_k$ to be the time after the next occurrence of the original pattern $\boldsymbol{\psi}_k$ it takes to observe the first occurrence of $\boldsymbol{\gamma}_k$, i.e., $\tau_k^* = \tau_k + \Delta\tau_k$. Clearly $\Delta\tau_k > 0$. Because the addition of ζ to the original sequence removes the self-overlap, we can use Lemma 10 to compute $\mathbb{E}[\tau_k^*]$:

$$\mathbb{E}[\tau_k^*] = \left(p(\zeta)\prod_{i=1}^{d_k} p(\psi_{k,i})\right)^{-1}.$$

Next, we can compute $\mathbb{E}[\Delta\tau_k]$ by conditioning on the value of ζ, assuming the pattern $\boldsymbol{\psi}_k$ has already been observed.

$$\mathbb{E}[\Delta\tau_k \,|\, \xi = \alpha] = \begin{cases} 1 + \mathbb{E}[\tau_k^* \,|\, \psi_{k,1}, \cdots, \psi_{k,v_k+1}] & \text{if } \alpha = \psi_{k,v_k+1} \\ 1 + \mathbb{E}[\tau_k^* \,|\, \psi_{k,1}] & \text{if } \alpha = \psi_{k,1} \\ 1 & \text{if } \alpha = \zeta \\ 1 + \mathbb{E}[\tau_k^*] & \text{if } \alpha \notin \{\psi_{k,1}, \psi_{k,v_k+1}, y\} \end{cases}$$

from which we can construct the equation

$$\mathbb{E}[\Delta\tau_k] = 1 + p(\psi_{k,v_k+1})\mathbb{E}[\tau_k^* | \psi_{k,1}, \cdots, \psi_{k,v_k+1}] \tag{3.6}$$
$$+ p(\psi_{k,1})\mathbb{E}[\tau_k^* | \psi_{k,1}] + (1 - p(\psi_{k,v_k+1}) - p(\psi_{k,1}) - p(\zeta))\mathbb{E}[\tau_k^*]. \tag{3.7}$$

Note that by definition of τ_k^*:

$$\mathbb{E}[\tau_k^*] = \mathbb{E}[\tau(\psi_{k,1:v_k+1})] + \mathbb{E}[\tau_k^* | \psi_{k,1:v_k+1}] \text{ and } \mathbb{E}[\tau_k^*] = \mathbb{E}[\tau(\psi_{k,1})] + \mathbb{E}[\tau_k^* | \psi_{k,1}], \tag{3.8}$$

where $\tau(\psi_{k,1:j})$ (for any $j \in \{1, \cdots, d_k\}$) represents the time it takes to observe the next occurrence of the subpattern $\psi_{k,1:j}$ after the first occurrence of $\boldsymbol{\psi}_k$. Assume ψ_{v_k+1} is such that the sub-pattern $\psi_{k,1:v_k+1}$ has no self-overlaps. Then since neither of the sequences $\psi_{k,1:v_k+1}$ nor $(\psi_{k,1})$ have self-overlaps, we can use Lemma 10 to compute:

$$\mathbb{E}[\tau(\psi_{k,1:v_k+1})] = \left(\prod_{i=1}^{v_k+1} p(\psi_{k,i})\right)^{-1} \text{ and } \mathbb{E}[\tau(\psi_{k,1})] = p(\psi_{k,1})^{-1}.$$

Hence, rearranging (3.8) and substituting back into (3.6):

$$\begin{aligned}\mathbb{E}[\tau_k^*] = \mathbb{E}[\tau_k] + 1 + p(\psi_{k,v_k+1})\left(\mathbb{E}[\tau_k^*] - \mathbb{E}[\tau(\psi_{k,1}, \cdots, \psi_{k,v_k+1})]\right) \\ + p(\psi_{k,1})\left(\mathbb{E}[\tau_k^*] - \mathbb{E}[\tau(\psi_{k,1})]\right) + (1 - p(\psi_{k,v_k+1}) - p(\psi_{k,1}) - p(\zeta))\mathbb{E}[\tau_k^*],\end{aligned}$$

which yields the desired result (3.5) after rearranging the terms. ■

Remark 16. If the subsequence $\psi_{k,1:v_k}$ itself has a self-overlap of size $v_k' < v_k$, then we can repeat Case 2 by induction:

$$\mathbb{E}[\tau_{k|0}] = \left(\prod_{i=1}^{d_k} p(\psi_{k,i})\right)^{-1} + \left(\prod_{i=1}^{v_k} p(\psi_{k,i})\right)^{-1} + \left(\prod_{i=1}^{v_k'} p(\psi_{k,i})\right)^{-1}. \tag{3.9}$$

Interpretation with Martingales

In contrast to relating them to the Geometric distribution, the pattern-occurrence problem can be solved from an alternative perspective using *martingales*, which we adapt as a combination of [92] and [123]. As typical of martingale analyses, it becomes convenient to view Problem 1 from a game perspective.

Definition 18 (Agents and Rewards: i.i.d. Case)**.** We introduce the notion of an *agent*, which observes the process $\{\xi_n\}$ and accumulates *rewards* at each time with the goal of observing a pattern from Ψ. We refer to a *type-k agent* to be an agent who accumulates rewards by specifically observing the occurrence of $\boldsymbol{\psi}_k \in \Psi$ in $\{\xi_n\}$. At each time $n \in \mathbb{N}$, K new agents, one for each type k, $k \in \{1, \cdots, K\}$, are introduced to system; we refer to a type-k agent which is introduced at time n as *type-k agent n*. In order for the game to be fair-odds, each type-k agent n observes the stochastic process $\{\xi_{n+1}, \xi_{n+2}, \cdots,\}$ and accumulates rewards at a rate that is inversely-proportional to the action it took, starting from some arbitrary *initial reward* $c_k \in \mathbb{R}$. When an agent first loses everything he has, he leaves the game and never returns.

We next make the following notations. For two patterns $\boldsymbol{\psi}_{k_1}$ and $\boldsymbol{\psi}_{k_2}$ defined in Definition 13, let

$$D_{i,j}(\boldsymbol{\psi}_{k_1}, \boldsymbol{\psi}_{k_2}) \triangleq \begin{cases} p(\psi_{k_2,j})^{-1} & \text{if } \psi_{k_1,i} = \psi_{k_2,j} \\ 0 & \text{else} \end{cases} \tag{3.10}$$

for all $i \in \{1, \cdots, d_{k_1}\}$, $j \in \{1, \cdots, d_{k_2}\}$. Intuitively, $D_{i,j}(\boldsymbol{\psi}_{k_1}, \boldsymbol{\psi}_{k_2})$ can be thought of as a fair-odds weighting factor that is earned when the ith element of pattern $\boldsymbol{\psi}_{k_1}$ and the jth element of pattern $\boldsymbol{\psi}_{k_2}$ coincide. We further define

$$\boldsymbol{\psi}_{k_1} \Diamond \boldsymbol{\psi}_{k_2} = \tag{3.11}$$
$$\prod_{i=1}^{\min(d_{k_1}, d_{k_2})} D_{i,i}(\boldsymbol{\psi}_{k_1}, \boldsymbol{\psi}_{k_2}) + \prod_{i=1}^{\min(d_{k_1}-1, d_{k_2})} D_{i+1,i}(\boldsymbol{\psi}_{k_1}, \boldsymbol{\psi}_{k_2}) + \cdots + D_{d_{k_1},1}(\boldsymbol{\psi}_{k_1}, \boldsymbol{\psi}_{k_2}).$$

Essentially, (3.11) conveys the total winnings obtained by an agent who is betting to observe (partial) occurrences of $\boldsymbol{\psi}_{k_2}$ in the pattern $\boldsymbol{\psi}_{k_1}$.

Now suppose a type-k agent n arrives and bets on the event $\{\xi_{n:n+d_k-1} = \boldsymbol{\psi}_k\}$ using *betting strategy* $\{B_{n,j}^{(k)}\}_{j=0}^{d_k}$, starting from $B_{n,0}^{(k)} = c_k \triangleq 1$. Because the game is fair-odds, we have:

$$B_{n,j+1}^{(k)} \triangleq \begin{cases} p(\psi_{k,j+1})^{-1} \cdot \sum_{jj=0}^{j} B_{n,jj}^{(k)} - \sum_{jj=0}^{j} B_{n,jj}^{(k)} & \text{if } \xi_{n+d_k-j} = \psi_{k,j+1} \\ 0 & \text{if } \sum_{jj=0}^{j} B_{n,jj}^{(k)} = 0 \\ -\sum_{jj=0}^{j} B_{n,jj}^{(k)} & \text{else} \end{cases} \tag{3.12}$$

for each $j \in \{0, \cdots, d_k - 1\}$.

Definition 19 (Cumulative Net Rewards: i.i.d. Case)**.** Given the setup above, the *type-k agent n cumulative net reward* by time $\overline{n} > n$ is given by $R_{n,\overline{n}}^{(k)} = \sum_{j=0}^{d_k-1} B_{n,j}^{(k)} \mathbb{1}\{n + j \leq \overline{n}\}$, which can be expanded as

$$R_{n,\overline{n}}^{(k)} = \tag{3.13}$$
$$\begin{cases} \left(\prod_{\substack{j \in \{0,\cdots,d_k-1\} \\ n+j \leq \overline{n}}} p(\psi_{k,j}) \right)^{-1} c_k \text{ if } \forall j \in \{0, \cdots, d_k - 1\} \text{ s.t. } n + j \leq \overline{n},\ D_{n+j,j+1}(\xi_{1:n}, \boldsymbol{\psi}_k) > 0 \\ 0 \qquad \text{if } \exists j \in \{0, \cdots, d_k - 1\} \text{ s.t. } n + j \leq \overline{n},\ D_{n+j,j+1}(\xi_{1:n}, \boldsymbol{\psi}_k) = 0 \end{cases}.$$

Note that $R_{n,n}^{(k)} = c_k$. Moreover, the *type-k cumulative net reward* over all type-k agents that arrive until time $\overline{n} > n$ is defined by $R_{\overline{n}}^{(k)} \triangleq \sum_{n=1}^{\overline{n}} R_{n,\overline{n}}^{(k)}$. Finally, with $\tau \triangleq \min_k \tau_{k|0}$ the minimum occurrence time, let R_τ be the net reward accumulated over all agents over all types by τ. Then

$$R_\tau \triangleq \begin{cases} (\boldsymbol{\psi}_1 \Diamond \boldsymbol{\psi}_1)c_1 + \cdots + (\boldsymbol{\psi}_1 \Diamond \boldsymbol{\psi}_K)c_K & \text{if } \tau = \tau_{1|0} \\ \qquad\qquad\qquad\qquad\qquad\quad \vdots & \\ (\boldsymbol{\psi}_K \Diamond \boldsymbol{\psi}_1)c_1 + \cdots + (\boldsymbol{\psi}_K \Diamond \boldsymbol{\psi}_K)c_K & \text{if } \tau = \tau_{K|0} \end{cases}. \tag{3.14}$$

In the i.i.d. case, we set up the betting strategy of each agent in the form of (3.12) so that $\{R_n^{(k)}\}$ is a martingale. To demonstrate this formally, we require Lemma 8 and Lemma 9. To build intuition, we first begin by deriving the formula for two patterns, inspired by Li 1980 [92], before moving on to the general $K \geq 2$ patterns.

Lemma 12 (Expected Occurrence Times: i.i.d. Case, 2 Patterns)**.** Let $\Psi \triangleq \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2\}$ be a collection of two patterns. Then the expected first occurrence time for pattern $\boldsymbol{\psi}_{k_2}$ is given by

$$\mathbb{E}[\tau_{k_2|0}] \triangleq \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2}, \tag{3.15}$$

where the $\Diamond$ notation is defined in (3.11). The expected waiting time for pattern $\boldsymbol{\psi}_{k_2}$ to occur for the first time after the first occurrence of another pattern $\boldsymbol{\psi}_{k_1}$ is given by

$$\mathbb{E}[\tau_{k_2|k_1}] \triangleq \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2} - \boldsymbol{\psi}_{k_1} \Diamond \boldsymbol{\psi}_{k_2}. \tag{3.16}$$

Proof of Lemma 12. First, by the argument shown in the previous Section 3.1.1, both $\tau_{k_2|k_1}$ and $\tau_{k_2|0}$ are dominated by Geometric random variables. Hence, $\mathbb{E}[\tau_{k_2|k_1}], \mathbb{E}[\tau_{k_2|0}] < \infty$. For indexing simplicity, we re-index the sequence $\{\xi_n\}$ such that $\boldsymbol{\psi}_{k_1}$ has occurred before time zero, i.e., $\xi_{-d_{k_1}+1} = \psi_{k_1,1}, \cdots, \xi_0 = \psi_{k_1,d_{k_1}}, \xi_1 \leftarrow \xi_{\tau_{k_1|0}+1}, \cdots, \xi_h \leftarrow \xi_{\tau_{k_1|0}+h}$ for some $h \in \mathbb{N}$. A type-k_2 agent $n \in \{-d_{k_1}+1, \cdots, h \wedge \tau_{k_2|k_1}\}$ arrives and bets on observing $\{\xi_{n:d_{k_2}-1} = \boldsymbol{\psi}_{k_2}\}$. Note that the reward process of type-k_2 agent n, $\{R^{(k_2)}_{n,h\wedge\tau_{k_2|k_1}}\}_{h\in\mathbb{N}}$, is a martingale by the fair-odds betting strategy construction from Definitions 18, 19, etc. The net reward over all type-k_2 agents $\{R^{(k_2)}_{h\wedge\tau_{k_2|k_1}}\}_{h\in\mathbb{N}}$ can be expressed as $R^{(k_2)}_h = \sum_{n=-d_{k_2}+1}^{h} R^{(k_2)}_{n,h}$, and since the weighted sum of martingales is also a martingale, $\{R^{(k_2)}_{h\wedge\tau_{k_2|k_1}}\}_{h\in\mathbb{N}}$ is also a martingale.

By construction, this simplifies to:

$$R^{(k_2)}_{h\wedge\tau_{k_2|k_1}} = \left(\boldsymbol{\psi}_{k_1} \circ \xi_{1:h\wedge\tau_{k_2|k_1}}\right) \Diamond \boldsymbol{\psi}_{k_2} - (h \wedge \tau_{k_2|k_1} + d_{k_1}), \tag{3.17}$$

and consequently:

$$R^{(k_2)}_{\tau_{k_2|k_1}} = \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2} - (\tau_{k_2|k_1} + d_{k_1}), \tag{3.18}$$

which implies

$$\mathbb{E}[R^{(k_2)}_{\tau_{k_2|k_1}}] = \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2} - (\mathbb{E}[\tau_{k_2|k_1}] + d_{k_1}) < \infty \tag{3.19}$$

since we've established $\mathbb{E}[\tau_{k_2|k_1}] < \infty$.

Denote $\Omega_h \triangleq \{\omega \in \Omega | h < \tau_{k_2|k_1}(\omega)\}$. By triangle inequality, $|R^{(k_2)}_{\tau_{k_2|k_1}}| \leq |\boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2}| + |\tau_{k_2|k_1}| + |d_{k_1}| = \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2} + \tau_{k_2|k_1} + d_{k_1}$, which is finite. We apply Lemma 8 to get

$$\liminf_{h\to\infty} \int_{\Omega_h} |\sum_{n=1}^{h} R^{(k_2)}_{n,h}(\omega)| d\mathbb{P}(\omega) \leq \liminf_{h\to\infty} \int_{\Omega_h} (\boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2} + \tau_{k_2|k_1} + d_{k_1}) d\mathbb{P}(\omega) = 0. \tag{3.20}$$

Hence, with the fact that $\{R^{(k_2)}_{h\wedge\tau_{k_2|k_1}}\}_{h\in\mathbb{N}}$ is a martingale and the conditions (3.19) and (3.20) are satisfied with stopping time $\tau_{k_2|k_1}$, we can apply Optional Stopping (Lemma 9). Note that $\mathbb{E}[R^{(k_2)}_0] = c_{k_2}(\boldsymbol{\psi}_{k_1} \Diamond \boldsymbol{\psi}_{k_2} - d_{k_2} + 1)$ since $\xi_{1:0}$ is the empty sequence, so we are looking for occurrences of $\boldsymbol{\psi}_{k_2}$ in $\boldsymbol{\psi}_{k_1}$. Thus

$$\mathbb{E}[R^{(k_2)}_{\tau_{k_2|k_1}}] = \mathbb{E}[R^{(k_2)}_0] \implies \mathbb{E}[\tau_{k_2|k_1}] = \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2} - \boldsymbol{\psi}_{k_1} \Diamond \boldsymbol{\psi}_{k_2}. \tag{3.21}$$

A similar argument can be used to show $\mathbb{E}[\tau_{k_2|0}] = \boldsymbol{\psi}_{k_2} \Diamond \boldsymbol{\psi}_{k_2}$. ■

Theorem 8 (Expected Occurrence Times: i.i.d. Case)**.** Define Ψ to be the pattern collection from Definition 13 with $K > 2$ patterns. From the notation of Definition 15, define $\tau \equiv \min_k \tau_{k|0}$ to be the i.i.d. version of the minimum occurrence time from Problem 1, defined in Definition 19. Then

$$\mathbb{E}[\tau] = \left(\sum_{k=1}^{K} c^*_k\right)^{-1}, \tag{3.22}$$

where $\mathbf{c}^* \triangleq (c^*_1, \cdots, c^*_K)^\top$ such that $W\mathbf{c}^* = \mathbf{1}^K$, where $\mathbf{1}^K$ is the K-dimensional vector of all ones and

$$W \triangleq \begin{bmatrix} \boldsymbol{\psi}_1 \Diamond \boldsymbol{\psi}_1 & \boldsymbol{\psi}_1 \Diamond \boldsymbol{\psi}_2 & \cdots & \boldsymbol{\psi}_1 \Diamond \boldsymbol{\psi}_K \\ \boldsymbol{\psi}_2 \Diamond \boldsymbol{\psi}_1 & & & \\ \vdots & & \ddots & \vdots \\ \boldsymbol{\psi}_K \Diamond \boldsymbol{\psi}_1 & & & \boldsymbol{\psi}_K \Diamond \boldsymbol{\psi}_K \end{bmatrix} \tag{3.23}$$

is called the *gain matrix*.

Proof of Theorem 8. By extending Lemma 12 to more than two patterns, we observe that the process $\{R_n\}$ (Definition 19) is a zero-mean martingale with stopping time τ, and $\mathbb{E}[\tau] < \infty$ with appropriate assumptions. Thus

$$\left(\sum_{k=1}^{K} c_k\right) \mathbb{E}[\tau] = \mathbb{E}[R_0] = \mathbb{E}[R_\tau] = \begin{bmatrix} q_1 & q_2 & \cdots & q_K \end{bmatrix} W \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{bmatrix}, \tag{3.24}$$

where q_k are the first-occurrence probabilities from the second question of Problem 1. Choosing $\mathbf{c}^*$ as above turns the first term of (3.24) into 1, since $\sum_k q_k = 1$. Rearranging the terms yields the desired equation (3.22). ■

Although the first-occurrence probabilities $(q_1, q_2, \cdots, q_K)^\top$ were unknown, we were able to compute $\mathbb{E}[\tau]$ in Theorem 8 without knowing what they were by choosing the initial bets $\mathbf{c}^*$ appropriately. However, any other choice of $\mathbf{c} \neq \mathbf{c}^*$ satisfies (3.24); we compute the first-occurrence probabilities using this observation.

Theorem 9 (First-Occurrence Probabilities: i.i.d. Case)**.** In the i.i.d. case, the vector of first-occurrence probabilities from Problem 1, denoted $\mathbf{q} \triangleq (q_1, \cdots, q_K)^\top$, are given by

$$\mathbf{q} = W^{-\top} \mathbb{1}^K \mathbb{E}[\tau], \tag{3.25}$$

where again, W is the gain matrix (3.23) and $\mathbb{1}^K$ is the K-dimensional all-ones vector.

Proof of Theorem 9. First, rearrange the terms of (3.24):

$$\mathbb{E}[\tau_{k_1|0}] = \mathbb{E}[\tau] + \sum_{k_2=1}^{K} q_{k_2} \mathbb{E}[\tau_{k_1|k_2}], \quad \mathbb{E}[\tau_{k_1}] = \boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_{k_1}.$$

Using the $\lozenge$ notation:

$$\boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_{k_1} = \mathbb{E}[\tau] + \sum_{k_2=1}^{K} q_{k_2} (\boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_{k_1} - \boldsymbol{\psi}_{k_2} \lozenge \boldsymbol{\psi}_{k_1}) \quad \Longrightarrow \quad 0 = \mathbb{E}[\tau] - \sum_{k_2=1}^{K} q_{k_2} \boldsymbol{\psi}_{k_2} \lozenge \boldsymbol{\psi}_{k_1}.$$

Consequently,

$$0 = \sum_{k_2=1}^{K} c_{k_2} \mathbb{E}[\tau] - \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} q_{k_1} (\boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_{k_2}) c_{k_2},$$

and we chose an appropriate $\mathbf{c}^*$ so that we could solve for $\mathbb{E}[\tau]$ without needing to solve for the $\{q_k\}$. Once $\mathbb{E}[\tau]$ is obtained, however, we can choose specific $\mathbf{c} = \{[1, 0, \cdots, 0], \cdots, [0, 0, \cdots, 1]\}$ to obtain the linear system of equations which allows us to solve for $\{q_k\}$:

$$\mathbf{c} = [1, 0, \cdots, 0]^\top \Longrightarrow \; 0 = \mathbb{E}[\tau] - \sum_{k_1=1}^{K} q_{k_1} (\boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_1),$$

$$\mathbf{c} = [0, 1, \cdots, 0]^\top \implies 0 = \mathbb{E}[\tau] - \sum_{k_1=1}^{K} q_{k_1}(\boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_2),$$

$$\vdots$$

$$\mathbf{c} = [0, 0, \cdots, 1]^\top \implies 0 = \mathbb{E}[\tau] - \sum_{k_1=1}^{K} q_{k_1}(\boldsymbol{\psi}_{k_1} \lozenge \boldsymbol{\psi}_K).$$

In terms of the gain matrix, $W^\top \mathbf{q} = \mathbb{E}[\tau]\mathbb{1}^K$, and solving for $\mathbf{q}$ yields the desired result. ■

3.1.2 Markov Chain Sequences

Now, we consider the pattern-occurrence problems for when $\{\xi_n\}$ is generated from an irreducible Markov chain over the state-space $\mathcal{X}$, i.e., $\mathbb{P}(\xi_n = \varphi | \xi_{n-1} = \varphi_{n-1}, \cdots, \xi_1 = \varphi_1, \varphi_0) \equiv \mathbb{P}(\xi_n = \varphi | \xi_{n-1} = \varphi_{n-1})$. The transition probability matrix (TPM) is denoted by $P \triangleq [P(\zeta, \zeta')] \in \mathbb{R}^{M \times M}$, and we take initial probability distribution to be $\{p_0(\zeta)\}_{\zeta \in \mathcal{X}}$ and stationary distribution to be $\{\pi(\zeta)\}_{\zeta \in \mathcal{X}}$.

Definition 20 (Feasibility)**.** A pattern $\boldsymbol{\psi}_k$ or an arbitrary sequence of modes $(\alpha_1, \cdots, \alpha_a)$ with length $a \in \mathbb{N}$ is *feasible with respect to* P if it can be generated by the Markov chain with TPM P, i.e., $P[\alpha_i, \alpha_{i+1}] > 0$ for all $i \in \{1, \cdots, a-1\}$.

In contrast to i.i.d. sequences, there are constraints on the degrees of freedom on possible Markov chain sample path trajectories due to the memorylessness property. Thus, we take inspiration from [122] and consider the occurrence of feasible augmented patterns up to two extra values. We choose two additional variables because when $n = 0$, the probability of observing $\xi_0 = \zeta$ is distributed differently from $\xi_n = \zeta, n \geq 2$ (i.e., the initial distribution $p_0(\zeta)$ instead of the transition probability $P(\xi_{n-1}, \zeta)$).

Definition 21 (Augmented Pattern)**.** Suppose we are given a collection of patterns Ψ (from Definition 13). An *augmented pattern* $\boldsymbol{\gamma}$ corresponding to a pattern $\boldsymbol{\psi}_k \in \Psi$ is defined by prefixing two values $m_1, m_2 \in \mathcal{X}$ such that the resulting sequence is feasible in the sense of Definition 20. We define the *augmented collection*

$$\Gamma \triangleq \{\text{feasible } (m_1, m_2) \circ \boldsymbol{\psi}_k \mid m_1, m_2 \in \mathcal{X}; \boldsymbol{\psi}_k \in \Psi\} \tag{3.26}$$

to be the collection of augmented patterns, and we define $K_L \in \mathbb{N}$ to be its cardinality. We henceforth enumerate each augmented pattern $\boldsymbol{\gamma}_\ell$ in the augmented collection Γ using subscript $\ell \in \{1, \cdots, K_L\}$.

It is easier to solve for Problem 1 by conditioning on the type of "ending string" observed, where an "ending string" is formally defined below.

Definition 22 (Ending Strings)**.** Given the collection of patterns Ψ and current time $n \in \mathbb{N}$, suppose we let the sequence $\{\xi_n, \xi_{n+1}, \cdots\}$ run until one of the patterns from Ψ has been observed. Then an *ending string* associated with pattern $\boldsymbol{\psi}_k \in \Psi$ terminates the process at time $\tau_n > n$ if $\xi_{\tau_n - d_k + 1:\tau_n} = \boldsymbol{\psi}_k$. We characterize two primary types of ending strings:

- An *initial-ending string* $\boldsymbol{\beta}$ occurs when part of an augmented pattern is observed immediately after the current value. We classify initial-ending strings into two further subcases:
  - A *Case* 0 *initial-ending string* $\boldsymbol{\beta} \triangleq \boldsymbol{\psi}_k$ occurs when $\xi_{n:n+d_k-1} = \boldsymbol{\psi}_k$. Define $\mathcal{S}_I^{(0)}$ to be the set of Case 0 initial-ending strings, with cardinality $K_I^{(0)} \in \mathbb{N}$.
  - A *Case* 1 *initial-ending string* $\boldsymbol{\beta} \triangleq (m_1) \circ \boldsymbol{\psi}_k$ occurs when $\xi_{n:n+d_k} = (m_1) \circ \boldsymbol{\psi}_k$. Here, $m_1 \in \mathcal{X}$ is such that the above ending string is feasible. Define $\mathcal{S}_I^{(1)}$ to be the set of Case 1 initial-ending strings, with cardinality $K_I^{(1)} \in \mathbb{N}$.
- A *later-ending string* $(*, m_1, m_2) \circ \boldsymbol{\psi}_k$ occurs when an augmented pattern is observed long after the current value, i.e., when $\tau_n > n + d_k + 1$ and $\xi_{\tau_n - d_k - 1:\tau_n} = (m_1, m_2) \circ \boldsymbol{\psi}_k$. Here, $m_1, m_2 \in \mathcal{X}$ are such that the above ending string is feasible, and $*$ is a placeholder for any feasible sequence of values (see Definition 20) including the empty string. Define $\mathcal{S}_L \triangleq \{(*) \circ \boldsymbol{\gamma}_\ell \,|\, \boldsymbol{\gamma}_\ell \in \Gamma\}$ to be the *set of later-ending strings*, with the same cardinality K_L as Γ.

Define $\mathcal{S}_I \triangleq \mathcal{S}_I^{(0)} \cup \mathcal{S}_I^{(1)}$ to be the *set of initial-ending strings*, with cardinality $K_I = K_I^{(0)} + K_I^{(1)}$. Together, define the *set of ending strings* to be $\mathcal{S} = \mathcal{S}_I \cup \mathcal{S}_L$. We henceforth enumerate each ending string $\boldsymbol{\beta}_s$ in $\mathcal{S}$ using the subscript $s \in \{1, \cdots, K_I + K_L\}$.

Example 2 (Ending Strings Construction)**.** We provide intuition behind the notation described by Definition 22. Let $M = 4$, i.e., $\mathcal{X} = \{1, 2, 3, 4\}$, and let the TPM P be such that $P[m_1, m_2] > 0$ for all $m_1, m_2 \in \mathcal{X}$ except when $m_1 = m_2$ and when $(m_1, m_2) \in \{(3, 2), (2, 3), (3, 4), (4, 3)\}$. The pattern collection of interest consists of $K = 3$ patterns $\Psi = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3\}$ with $\boldsymbol{\psi}_1 = (213)$, $\boldsymbol{\psi}_2 = (412)$, and $\boldsymbol{\psi}_3 = (314)$. The augmented pattern collection is defined as $\Gamma \triangleq \cup_{i=1}^3 \Gamma_i$ with $\Gamma_1 = \{\boldsymbol{\alpha} \circ \boldsymbol{\psi}_1 | \boldsymbol{\alpha} \in \{(14), (21), (24), (31), (41)\}\}$, $\Gamma_2 = \{\boldsymbol{\alpha} \circ \boldsymbol{\psi}_2 | \boldsymbol{\alpha} \in \{(12), (21), (31), (41), (42)\}\}$, $\Gamma_3 = \{\boldsymbol{\alpha} \circ \boldsymbol{\psi}_3 | \boldsymbol{\alpha} \in \{(21), (31), (41)\}\}$. Hence, the number of later-ending

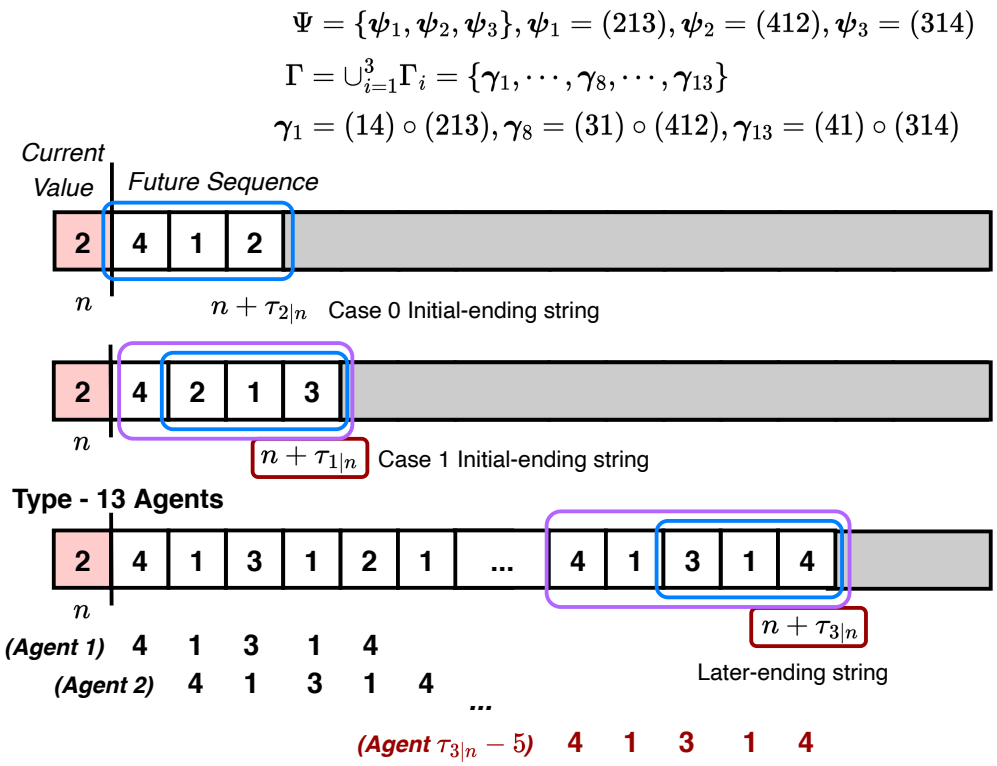

Figure 3.1: A visualization of the ending strings and agent-reward construction using the setup of Example 2. The red box marks the current time $n \in \mathbb{N}$, and each of the three sequences demonstrate the three different types of ending strings which terminate the stochastic process in the sense of Definition 22. The grey rectangles hide future values which have not occurred because of sequence termination. For the last case where γ_{13} terminates the process as a later-ending string, type-13 agents at indices $1, 2, \cdots, \tau_{3|n} - 5$ are shown. By the reward construction of Definition 23, type-13 agent $\tau_{3|n} - 5$ is the only agent who receives a nonzero reward.

strings is $K_L = 13$. Suppose the current state is $\varphi_n = 2$. Then the set of feasible augmented Case 0 initial-ending strings is $\mathcal{S}_I^{(0)} = \{\psi_2\}$ since $P[2,4] > 0$, and for Case 1 initial-ending strings, $\mathcal{S}_I^{(1)} = \{(1) \circ \psi_1, (4) \circ \psi_1, (1) \circ \psi_2, (1) \circ \psi_3\}$. Thus, $K_I^{(0)} = 1$ and $K_I^{(1)} = 4$.

Definition 23 (Agents and Rewards: MC Case)**.** Let Γ be the augmented pattern collection associated with original collection Ψ (see Definition 21). Similar to the i.i.d. case (Definition 18), we introduce the notion of an *agent*, except with respect to the augmented collection Γ instead of the original collection Ψ. We refer to a *type-ℓ agent* to be an agent which accumulates rewards by specifically observing the occurrence of $\gamma_\ell \in \Gamma$ in $\{\xi_n\}$. At each time $n \in \mathbb{N}$, K_L new agents, one for each type ℓ, $\ell \in \{1, \cdots, K_L\}$, are introduced to the process; we refer to a type-ℓ agent that is introduced at time n as *type-ℓ agent n*. A type-ℓ agent n

observes realizations in the future sequence $\{\xi_{n+1}, \xi_{n+2}, \cdots,\}$ and accumulates rewards at a rate which is inversely-proportional to the action it took, starting with some arbitrary *initial reward* $c_\ell \in \mathbb{R}$.

Each type-ℓ agent n aiming to observe augmented pattern $\boldsymbol{\gamma}_\ell \triangleq (m_1, m_2) \circ \boldsymbol{\psi}_k$ arrives and chooses one betting strategy between $\{B_{n,j}^{(1,\ell)}\}_{j=1}^{d_k+1}$ or $\{B_{n,j}^{(2,\ell)}\}_{j=1}^{d_k}$ depending on the observed outcome φ_n. Both betting strategies are initialized with $B_{n,0}^{(1,\ell)} = B_{n,0}^{(2,\ell)} = c_\ell$. If $\varphi_n = m_1$, type-ℓ agent n aims to observe the event $\{\xi_{n+1:n+d_k+1} = (m_2) \circ \boldsymbol{\psi}_k\}$:

$$B_{n,j}^{(1,\ell)} = \begin{cases} \left(P(\gamma_{\ell,j}, \gamma_{\ell,j+1})\right)^{-1} \sum_{i=0}^{j-1} B_{n,i}^{(1,\ell)} - \sum_{i=0}^{j-1} B_{n,i}^{(1,\ell)} & \text{if } \xi_{n+j-1} = \gamma_{\ell,j}, \xi_{n+j} = \gamma_{\ell,j+1} \\ 0 & \text{if } \sum_{jj=0}^{j-1} B_{n,jj}^{(1,\ell)} = 0 \\ -\sum_{i=0}^{j-1} B_{n,i}^{(1,\ell)} & \text{else} \end{cases}. \quad (3.27)$$

Otherwise, if $\varphi_n \neq m_1$, type-ℓ agent n aims to observe the event $\{\xi_{n+1:n+d_k} = \boldsymbol{\psi}_k\}$:

$$B_{n,j}^{(2,\ell)} = \begin{cases} \left(P(\xi_n, \psi_{k,1})\right)^{-1} c_\ell - c_\ell & \text{if } \xi_{n+1} = \psi_{k,1} \\ -c_\ell & \text{else} \end{cases};$$

for all $j = 2, \cdots, d_k$,

$$B_{n,j}^{(2,\ell)} = \begin{cases} \left(P(\psi_{k,j-1}, \psi_{k,j})\right)^{-1} \sum_{i=0}^{j-1} B_{n,i}^{(2,\ell)} - \sum_{i=0}^{j-1} B_{n,i}^{(2,\ell)} & \text{if } \xi_{n+j-1} = \psi_{k,j-1}, \xi_{n+j} = \psi_{k,j} \\ 0 & \text{if } \sum_{jj=0}^{j-1} B_{n,jj}^{(2,\ell)} = 0 \\ -\sum_{i=0}^{j-1} B_{n,i}^{(2,\ell)} & \text{else} \end{cases}. \quad (3.28)$$

In both strategies $\{B_{n,j}^{(1,\ell)}\}_{j=1}^{d_k+1}$ and $\{B_{n,j}^{(2,\ell)}\}_{j=1}^{d_k}$, the middle case of 0 arises because the type-ℓ agent n leaves the game on the first time he loses the entire reward he accumulated so far.

Given the setup above, the *type-ℓ agent n cumulative net reward* by time $\overline{n} \in \mathbb{N}$ is given by:

$$R_{n,\overline{n}}^{(\ell)} = \mathbb{1}\{\xi_n = m_1\} \sum_{j=0}^{d_k+1} B_{n,j}^{(1,\ell)} \mathbb{1}\{n + j \leq \overline{n}\} + \mathbb{1}\{\xi_n \neq m_1\} \sum_{j=0}^{d_k} B_{n,j}^{(2,\ell)} \mathbb{1}\{n + j \leq \overline{n}\}. \quad (3.29)$$

Remark 17. It becomes necessary to distinguish the occurrence time of a pattern $\boldsymbol{\psi}_k$ from that of an augmented pattern $\boldsymbol{\gamma}_\ell \triangleq (m_1, m_2) \circ \boldsymbol{\psi}_k$. We define $\tau_{\ell|n}^a$ and τ_n^a to be the versions of (3.1) and (3.2) for $\gamma_\ell \in \Gamma$.

Remark 18. Due to the stationarity of $\{\xi_n\}$, the distributions of $\tau_{k|n_1} - n_1$ and $\tau_{k|n_2} - n_2$ are equivalent for each $k \in \{1, \cdots, K\}$, and any times $n_1, n_2 \in \mathbb{N}$, such that $\varphi_{n_1} = \varphi_{n_2}$. Likewise, the distributions of $\tau_{n_1} - n_1$ and $\tau_{n_2} - n_2$ are equivalent. For notation simplicity in the following presentation, we remove the subscript $n \in \mathbb{N}$ in all variables, and use the above stationarity property to shift times to $n = 0$ in variables such that the current value is given by φ_0 instead of φ_n. Furthermore, we apply the shorthand notation to Definitions 14 and 15 such that $\tau_k \equiv \tau_{k|0}$ and $\tau \equiv \tau_0$; the notation for the augmented patterns (Remark 17) follow similarly as $\tau_\ell^a \equiv \tau_{\ell|0}^a$ and $\tau^a \equiv \tau_0^a$.

Definition 24 (Ending String Probabilities)**.** Define $\mathbb{P}(\boldsymbol{\beta}_s)$ to be the probability that ending string $\boldsymbol{\beta}_s \in \mathcal{S}$ terminates the stochastic process $\{\xi_n\}$ in the sense of Definition 22. For initial-ending strings $\boldsymbol{\beta}_s \in \mathcal{S}_I$ which is explicitly denoted as $(\beta_1, \cdots, \beta_{b_s})$ with length $b_s \in \mathbb{N}$, we get $\mathbb{P}(\boldsymbol{\beta}_s) = p(\varphi_0) P[\varphi_0, \beta_1] \prod_{j=1}^{b_s - 1} P[\beta_j, \beta_{j+1}]$. We later demonstrate how to compute $\mathbb{P}(\boldsymbol{\beta}_s)$ for later-ending strings $\boldsymbol{\beta}_s \in \mathcal{S}_L$, as part of solving Problem 1.

Definition 25 (Gain Matrix)**.** We construct the Markov chain version of the gaim matrix (3.23). Let $\boldsymbol{\beta}_s \in \mathcal{S}$ be an ending string which is explicitly denoted as $\boldsymbol{\beta}_s \triangleq (\beta_1, \cdots, \beta_{b_s}) \in \mathcal{S}$ with length $b_s \in \mathbb{N}$. Further let augmented pattern $\boldsymbol{\gamma}_\ell \in \Gamma$ be associated with original pattern $\boldsymbol{\psi}_k \in \Psi$, i.e., $\boldsymbol{\gamma}_\ell \triangleq (m_1, m_2) \circ \boldsymbol{\psi}_k$ for some $m_1, m_2 \in \mathcal{X}$. Then the *total gain* $W_{s\ell}$ accumulated over all type-ℓ agents from observing (partial) occurrences of $\boldsymbol{\gamma}_\ell$ in $\boldsymbol{\beta}_s$, is given by

$$W_{s\ell} \triangleq \sum_{j=1}^{\min(b_s - 1, d_k + 1)} D_j^{(1)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) + \sum_{j=1}^{\min(b_s - 1, d_k)} D_j^{(2)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell),$$

with $D_i^{(1)}$ and $D_i^{(2)}$ defined based on the reward strategy from Definition 23. First,

$$D_i^{(1)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) \triangleq \left(P[m_1, m_2] P[m_2, \psi_{k,1}] \prod_{j=1}^{i-2} P[\psi_{k,j}, \psi_{k,j+1}] \right)^{-1}$$

if $\beta_{b_s - i} = m_1$ and $\beta_{b_s - i + 1} = m_2, \beta_{b_s - i + j} = \psi_{k,j-1}$ for all $j \in \{2, \cdots, i\}$; else, $D_i^{(1)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) = 0$. Second,

$$D_i^{(2)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) \triangleq \left(P[\beta_{b_s - i}, \psi_{k,1}] \prod_{j=2}^{i} P[\psi_{k,j-1}, \psi_{k,j}] \right)^{-1}$$

if $\beta_{b_s - i} \neq m_1$ and $\beta_{b_s - i + j} = \psi_{k,j}$ for all $j \in \{1, \cdots, i\}$; else, $D_i^{(2)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) = 0$. A *gain matrix* $W \in \mathbb{R}^{(K_I + K_L) \times K_L}$ is constructed with entries $W_{s\ell}$ for each pair of $\boldsymbol{\beta}_s \in \mathcal{S}$ and $\boldsymbol{\gamma}_\ell \in \Gamma$.

Definition 26 (Cumulative Net Rewards: MC Case)**.** The expected *type-ℓ cumulative net reward* over all type-ℓ agents by time τ is defined $\mathbb{E}[R_\tau^{(\ell)}] \triangleq c_\ell[\mathbb{P}(\boldsymbol{\beta}_1), \cdots, \mathbb{P}(\boldsymbol{\beta}_{K_I+K_L})]W_{\cdot,\ell}$, where the $\mathbb{P}(\boldsymbol{\beta}_s)$ are the probabilities from Definition 24 and $W_{\cdot,\ell}$ denotes the ℓth column of the gain matrix (see Definition 25). Correspondingly, the expected *cumulative net reward* over all agents by time $\overline{n}$ is defined as $R_{\overline{n}} \triangleq \sum_{\ell=1}^{K_L} R_{\overline{n}}^{(\ell)}$, and

$$\mathbb{E}[R_\tau] = [\mathbb{P}(\boldsymbol{\beta}_1) \cdots \mathbb{P}(\boldsymbol{\beta}_{K_I+K_L})]W\mathbf{c}, \tag{3.30}$$

where $\mathbf{c} \triangleq [c_1, \cdots, c_{K_L}]^\top$ is the vector of initial rewards (see Definition 23). Note that this expression is similar to (3.24) from the i.i.d. case.

We are now ready to use our construction to present our main results, which address the questions in Problem 1.

Theorem 10 (Expected Time of Occurrence: MC Case)**.** Denote τ as in Remark 18 with current mode φ_0 for the collection Ψ from Definition 13 and corresponding augmented collection Γ. Then

$$\mathbb{E}[\tau] = \frac{1}{\sum_{\ell=1}^{K_L} c_\ell^*}\left[\left(1 - \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s)\right) + \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^*\right], \tag{3.31}$$

where $\boldsymbol{\gamma}_{\boldsymbol{\ell}} \in \Gamma$, $\boldsymbol{\beta}_s \in \mathcal{S}$, $\mathbb{P}(\boldsymbol{\beta}_s)$ is from Definition 24, W is from Definition 25, and $\mathbf{c}^* \in \mathbb{R}^{K_L}$ is the vector of initial rewards (see Definition 25) such that $\sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^* = 1$ for all $s \in \{K_I + 1, \cdots, K_I + K_L\}$.

Proof. Because the Markov chain is irreducible and finite-state, $\mathbb{E}[\tau_\ell^a] < \infty$, for each τ_ℓ^a defined in Remark 17. Note that $\tau_k = \min_{\boldsymbol{\gamma}_\ell \in \Gamma_k} \tau_\ell^a$, where Γ_k is the subset of Γ containing augmented patterns $\boldsymbol{\gamma} \triangleq (m_1, m_2) \circ \boldsymbol{\psi}_k$ corresponding to original pattern $\boldsymbol{\psi}_k \in \Psi$. We have that $\tau \triangleq \min_k \tau_k$, and by Definition 15, we also have $\mathbb{E}[\tau] < \infty$. By the construction of the gain matrix W and the fact that linear combinations of martingales are martingales, both $\{R^{(\ell)}_{n\wedge\tau_\ell^a}\}_{n\in\mathbb{N}}$ and $\{R_{n\wedge\tau}\}_{n\in\mathbb{N}}$ are martingales. This implies that $\mathbb{E}[R^{(\ell)}_{\tau_\ell^a}] < \infty$ since $\mathbb{E}[\tau_\ell^a] < \infty$. Furthermore, $\mathbb{E}[R_\tau] < \infty$ because $\tau \leq \tau_\ell^a$ for all ℓ. Define the set $\Omega_n^{(\ell)} \triangleq \{\omega \in \Omega | n < \tau_\ell^a\}$. By Lemma 8 and the triangle inequality, $\lim_{n\to\infty} \int_{\Omega_n^{(\ell)}} |R_n^{(\ell)}(\omega)| d\mathbb{P}(\omega) = 0$, which implies $\{R^{(\ell)}_{n\wedge\tau_\ell^a}\}$ is uniformly-integrable over $\Omega_n^{(\ell)}$. Thus, $\{R_{n\wedge\tau}\}$ is uniformly-integrable over $\Omega_n \triangleq \{\omega \in \Omega \,|\, n < \tau\} \subseteq \cap_{\ell=1}^{K_L} \Omega_n^{(\ell)}$. With the above conditions satisfied, we apply Optional Stopping (Lemma 9) to the stopped process $\{R_{n\wedge\tau}\}$, which implies $\mathbb{E}[R_\tau] = \mathbb{E}[R_0]$. Note $\mathbb{E}[R_0] = 0$ by the construction

of Definition 26. After choosing the initial rewards $\mathbf{c}^*$ as in the theorem statement, and substituting into (3.30):

$$\sum_{\ell=1}^{K_L} c_\ell^* \mathbb{E}[\tau] = \mathbb{E}[R_0] = \mathbb{E}[R_\tau] = \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^* + \left(1 - \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s)\right).$$

Rearranging the terms to isolate $\mathbb{E}[\tau]$ yields (3.31). ■

This addresses the first question in Problem 1. To address the second question, we use the following theorem, which also addresses the computation of $\mathbb{P}(\boldsymbol{\beta}_s)$ for later-ending strings $\boldsymbol{\beta}_s \in \mathcal{S}_L$ (see Definition 24).

Theorem 11 (First-Occurrence Probabilities: MC Case)**.** In addition to the setup of Theorem 10, explicitly denote ending string $\boldsymbol{\beta}_s \triangleq (\beta_1, \cdots, \beta_{b_s}) \in \mathcal{S}$ to have length $b_s \in \mathbb{N}$. Then the first-occurrence probabilities $\{q_k\}$ (see Definition 15) are given by

$$q_k = \sum_{\boldsymbol{\beta}_s \in \mathcal{S}} \mathbb{P}(\boldsymbol{\beta}_s) \mathbb{1}\{\beta_{b_s - d_k + 1:b_s} = \boldsymbol{\psi}_k\}. \tag{3.32}$$

Proof. Rearranging the terms of (3.30):

$$\sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell = -\sum_{s=K_I+1}^{K_I+K_L} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell + \sum_{\ell=1}^{K_L} c_\ell \mathbb{E}[\tau]. \tag{3.33}$$

We are given $\mathbb{E}[\tau]$ from Theorem 10, and $\mathbb{P}(\boldsymbol{\beta}_s)$ can be computed via Definition 24 when $\boldsymbol{\beta}_s \in \mathcal{S}_I$. For $s \in \{K_I + 1, \cdots, K_I + K_L\}$, choose one of K_L vectors $\mathbf{c} \in \{\mathbf{e}_1, \cdots, \mathbf{e}_{K_L}\}$ (where $\mathbf{e}_i$ is the ith standard basis vector of $\mathbb{R}^{K_L}$) to substitue into (3.33) and construct K_L different equations. Solve the resulting linear system for the K_L unknowns $\{\mathbb{P}(\boldsymbol{\beta}_{K_I+1}), \cdots, \mathbb{P}(\boldsymbol{\beta}_{K_I+K_L})\}$. By Definition 15, $q_k \triangleq \mathbb{P}(\tau = \tau_k) = \sum_{\boldsymbol{\beta}_s \in \mathcal{S}} \mathbb{P}(\boldsymbol{\beta}_s) \mathbb{P}(\tau = \tau_k \,|\, \boldsymbol{\beta}_s)$, where we denote shorthand $\mathbb{P}(\tau = \tau_k \,|\, \boldsymbol{\beta}_s)$ to be the probability of $\boldsymbol{\psi}_k$ being the first pattern observed at mode-index τ given $\boldsymbol{\beta}_s$ is the ending string which terminated the mode process in the sense of Definition 22. Clearly, $\mathbb{P}(\boldsymbol{\psi}_k \,|\, \boldsymbol{\beta}_s) = 1$ if $\beta_{b_s - d_k + 1:b_s} = \boldsymbol{\psi}_k$ holds, otherwise it is 0. We thus obtain the desired equation. ■

3.2 Uncertain Markovian Jump Systems (MJSs)

The Poisson shot and Lévy noise systems we presented in Chapter 2 are comparable to hybrid systems or Markovian jump systems (MJSs). In shot or Lévy noise systems, large deviations away from nominal behaviors arise solely from the jump-discontinuous noise process, which

is independent of the open-loop dynamics. In contrast, hybrid systems have switches (i.e., jumps) which arise as an inherent property of the open-loop dynamics. Despite this important distinction, the two settings can still be closely related to one another in two ways. First, stability analysis techniques are primarily focused on handling the jump-discontinuities more than any other property of the system. For hybrid systems, literature toward this direction of research include Lyapunov-sense conditions for asymptotic stability [94, 67] and characterizations of incremental stability [120]. Second, *dwell time* can be related to the interarrival time by viewing it as a form of stability criteria which ensures that the system has sufficient time to converge toward a desired state in between consecutive switching phases. Likewise, the stability results of Section 2.4 can be alternatively interpreted as conditions imposed on the shot or Lévy noise system such that the mean time between consecutive jumps (which depends on the intensity parameter λ) is long enough for the system to be reasonably close to the nominal trajectory. One notable example which utilizes dwell-time criteria for nonlinear systems is in Hespanha 1999 [68], where it is shown that input-to-state induced norms should be bounded uniformly between switches. In terms of applications, dwell-time criteria for attaining exponential stability has been shown to be effective for robotic systems, in particular walking locomotion and flapping flight [50] as well as autonomous vehicle steering [105].

In this section, we consider a specific class of MJSs with the purpose of demonstrating the efficiency of controller architectures designed around applying the pattern-occurrence problem to the underlying Markov chain mode process.

3.2.1 Setup and Preliminaries

We consider linear Markovian jump systems (MJSs) of the following form:

$$\mathbf{x}[t+1] = A(\xi_{N[t]})\mathbf{x}[t] + B\mathbf{u}[t] + \mathbf{w}[t]. \tag{3.34}$$

Here, $\mathbf{x}[t] \in \mathbb{R}^{n_x}$ is the state, $A(\xi_{N[t]}) \in \mathbb{R}^{n_x \times n_x}$ is the dynamics matrix which changes according to the phase variable $\xi_{N[t]}$, $\mathbf{u}[t] \in \mathbb{R}^{n_u}$ is the control input, and $\mathbf{w}[t] \in \mathbb{R}^{n_x}$ is an unobservable external noise process whose distribution is assumed known or partially known. For each $t \in \mathbb{N}$, $N[t]$ is the number of *modes* (i.e., number of phases switches, or jumps arising from the underlying Markov chain) that have been observed by time t. We henceforth say that the current *mode-index* at time $t \in \mathbb{N}$ is $n \in \mathbb{N}$ if $N[t] = n$, and the transition from mode ξ_{n-1} to ξ_n occurs at time $T_n \triangleq \min\{s \in \mathbb{N} \,|\, N[s] = n\}$. The discrete mode process $\{\xi_n\}_{n=1}^{\infty}$ takes values from the set $\mathcal{X} \triangleq \{1, \cdots, M\}$, where $M \in \mathbb{N}$, and is defined such that $\xi_n : \Omega \to \mathcal{X}$ on probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$, $\mathcal{F}_n \triangleq \sigma(\xi_0, \xi_1, \cdots, \xi_n)$. We assume

B is a known constant matrix. Again, we distinguish $\{\xi_n\}$ from the sequence of deterministic values $\{\varphi_n\}$ which it takes, i.e., $\xi_n = \varphi_n$ for all past mode-indices $n \in \mathbb{N}$. With similar MATLAB notation as before, we denote $\mathbf{x}[s:t] \triangleq \{\mathbf{x}[s], \cdots, \mathbf{x}[t]\}$ for any $s < t$, likewise for $\mathbf{u}[s:t], \mathbf{w}[s:t]$.

Assumption 7. The mode process $\{\xi_n\}$ operates on a timescale which is $\Delta T \in \mathbb{N}$ times longer than the timescale of the system (3.34), i.e., if $N[t] = n$, then $N[t + a\Delta T] = n + a$ for any $a \in \mathbb{N}$. This means $T_n - T_{n-1} = \Delta T$ for all $n \in \mathbb{N}$. In certain applications, ΔT can be interpreted as the minimum time needed between switching modes, and for simplicity we assume that its value is known. Consequently, we assume that $N[t]$ and the sequence of transition times $\{T_n\}$ are also known.

The mode process $\{\xi_n\}$ is generated from an irreducible Markov chain over the state-space $\mathcal{X}$ with transition probability matrix (TPM) denoted by $P \in \mathbb{R}^{M \times M}$ and initial probability vector $\mathbf{p}_0 \triangleq [p_0(1), \cdots, p_0(M)]^\top \in \{0,1\}^M$. As before, in Section 3.1.2, we represent the entries of the TPM using brackets, so that $P[m_1, m_2]$ denotes the probability of the mode switching from m_1 to m_2, for any $m_1, m_2 \in \mathcal{X}$. Suppose the probability distribution of ξ_n is given by $\mathbf{p}_n \in \{0,1\}^M$ at mode-index $n \in \mathbb{N}$, then the mode process dynamics are updated in the usual Markov chain way $\mathbf{p}_{n+1}^\top = \mathbf{p}_n^\top P$. This implies that given $\xi_n = \varphi_n \in \mathcal{X}$, we have $\xi_{n+1} = m$ with probability $P[\varphi_n, m]$ for any $m \in \mathcal{X}$.

Assumption 8. In addition to knowing the values of ΔT, $N[t]$, and $\{T_n\}$ (see Assumption 7), we consider the following settings. The true realizations $\{\varphi_n\}$ of the mode process $\{\xi_n\}$ are unknown over time, but the set $\mathcal{X}$ of values that it takes and the initial mode $\xi_0 = \varphi_0$ are known. The sparsity structure of the TPM P is known, but the values of the nonzero entries are unknown. As a simple example of this setting, one can think of an industrial machine which has three known modes (e.g., working, stalled, and failed), but the process in which the machine varies among the three modes is largely unknown.

3.3 Controller Architecture for MJSs Based on Pattern-Learning

The controller architecture we propose is visualized in Figure 3.2. It consists of three main parts: 1) Mode Process Identification (ID), 2) Pattern-Learning for Prediction (PLP) on the mode process, and 3) Control Law Design for the system dynamics. In this section, we begin with a brief description of each part–including an introduction of the main notations used–to provide a coherent view of the architecture (Figure 3.2) as a whole. The details of each individual part and the choice of algorithms used to implement them are discussed in the next

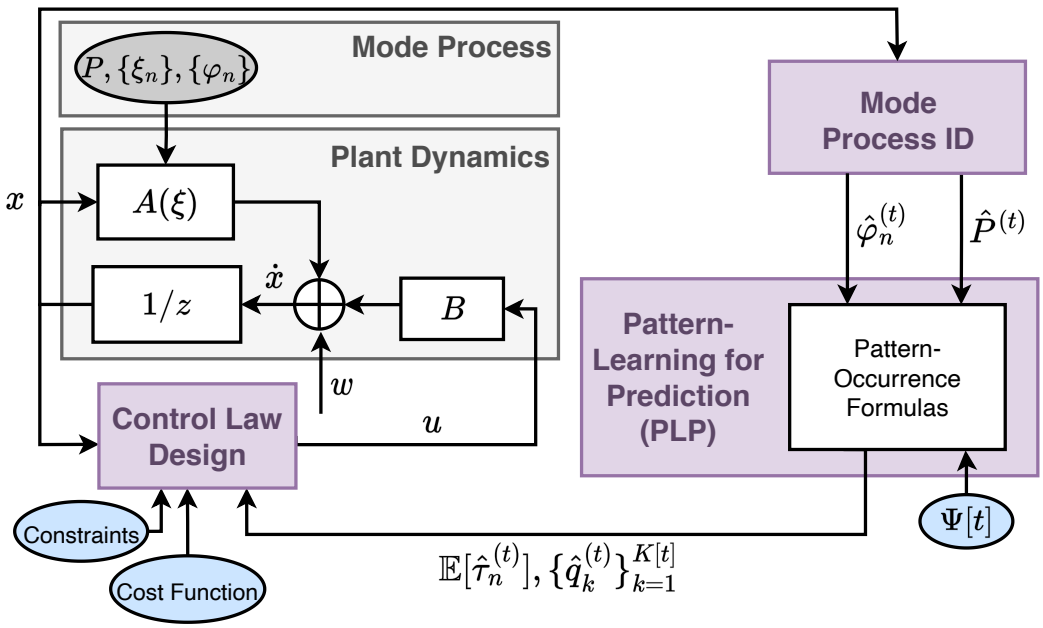

Figure 3.2: A flow diagram representation of the proposed controller architecture specifically for linear MJS dynamics of the form (3.34). Circles represent inputs to the algorithm; user-defined inputs are colored blue and unknown/unobservable parameters are colored gray. The architecture consists of three main parts (violet boxes): 1) Mode Process ID (Section 3.3.1), 2) Pattern-Learning for Prediction (Section 3.3.2), 3) and Control Law Design (Section 3.3.3).

sections: Mode Process ID in Section 3.3.1, PLP in Section 3.3.2, and Control Law Design in Section 3.3.3. We emphasize that our choice of algorithm to implement each component is unique to the uncertain linear discrete-time MJS setup described in Section 3.2 and that alternative implementations can be made for other dynamics; this will be demonstrated in the applications described in the subsequent chapters.

3.3.1 Mode Process Identification

For each time $t \in \mathbb{N}$ and corresponding mode-index $n \triangleq N[t]$, the system maintains the following estimated statistics about the mode process $\{\xi_n\}$ and system dynamics (3.34): an estimate $\hat{P}^{(t)}$ of the true TPM P, and an estimate $\hat{\varphi}_n^{(t)}$ of the current mode φ_n. The first part of our architecture, *Mode Process Identification (ID)* is responsible for learning these unknown statistics of the mode process. We use hats and superscripts (t) to emphasize that these quantities are estimates which change over time.

Definition 27 (Mode Process Estimates)**.** For each time $t \in \mathbb{N}$ and corresponding mode-index $n \triangleq N[t]$, the system maintains the following estimated statistics about the mode

process $\{\xi_n\}$ and system dynamics (3.34): an estimate $\hat{P}^{(t)}$ of the true TPM P, and an estimate $\hat{\varphi}_n^{(t)}$ of the current mode φ_n. We use hats and superscripts (t) to emphasize that these quantities are estimates which change over time.

3.3.2 Pattern-Learning in the Mode Process

Once $\hat{P}^{(t)}$ and $\hat{\varphi}_n^{(t)}$ are obtained from Mode Process ID (Section 3.3.1) for each $t \in \mathbb{N}$ and $n \triangleq N[t]$, *Pattern-Learning for Prediction (PLP)* in Figure 3.2 computes additional statistics about the mode process that facilitate the creation of *predictions*, which will be used in the Control Law Design component. These additional statistics are precisely the pattern-occurrence quantities described in Section 3.1, which can be solved using the method of Section 3.1.2 for Markov chains. In the context of the uncertain linear MJS (3.34), however, we require some additional adjustments, detailed below.

Definition 28 (Prediction Horizon)**.** Define the constant $L \in \mathbb{N}$ to be the *prediction horizon* on the mode process, i.e., the length of the sequences of modes.

In the setup of the Markovian jump system from Section 3.2, "patterns" refer to finite-length sequences of modes in the mode process underlying the system (3.34), formalized in Definition 13. Because we consider a fixed future horizon of length L, however, the length of each pattern $\boldsymbol{\psi}_k$ is $d_k \equiv L$ for each $k \in \{1, \cdots, K\}$.

Definition 29 (Pattern-Occurrence Times for Uncertain MJS)**.** Denote $n \triangleq N[t] \in \mathbb{N}$ to be the current mode-index at current time $t \in \mathbb{N}$, and suppose the estimated current mode is $\xi_n = \hat{\varphi}_n^{(t)}$. Then for each of the patterns in the collection Ψ from Definition 13, define the following stopping times for each $k \in \{1, \cdots, K\}$:

$$\hat{\tau}_{k|n}^{(t)} \triangleq \min\{i \in \mathbb{N} \,|\, \xi_n = \hat{\varphi}_n^{(t)}, \xi_{n+i-L+1:n+i} = \boldsymbol{\psi}_k\}. \tag{3.35}$$

Definition 30 (Time and Probability of First Occurrence for Uncertain MJS)**.** Under the setup of Definition 29 and given the collection Ψ, the 1) *minimum time of occurrence* of any pattern in Ψ and 2) the *first-occurrence probabilities* (i.e., the probability that each pattern $\boldsymbol{\psi}_k$ will be the first observed) are given by:

$$\hat{\tau}_n^{(t)} \triangleq \min_{k \in \{1,\cdots,K\}} \hat{\tau}_{k|n}^{(t)}, \qquad \hat{q}_k^{(t)} \triangleq \mathbb{P}(\hat{\tau}_n^{(t)} = \hat{\tau}_{k|n}^{(t)}). \tag{3.36}$$

This means $\xi_{n+\hat{\tau}_n^{(t)}-L+1:n+\hat{\tau}_n^{(t)}} = \boldsymbol{\psi}_k$ if pattern $\boldsymbol{\psi}_k$ will be the first observed.

To generate predictions from the mode process, we are interested in characterizing the pattern-occurrence quantities defined by Problem 1. In the context of the MJS (3.34), these are the following. First, we want the estimate $\mathbb{E}[\hat{\tau}_n^{(t)}]$ of the *mean minimum occurrence time*, which counts the number of mode-indices to observe the occurrence of any pattern from Ψ, given the estimated current mode $\hat{\varphi}_n^{(t)}$. Second, we want the estimated *first-occurrence probabilities* $\{\hat{q}_k^{(t)}\}$, where $\hat{q}_k^{(t)} \in [0,1]$ is the probability that pattern $\boldsymbol{\psi}_k \in \Psi$ is the first to be observed among all of Ψ. Because the statistics of the mode process are estimates instead of true values, all the quantities from Section 3.1 are expressed with hats and superscript (t)s. Moreover, it becomes necessary to consider a pattern collection Ψ (from Definition 13) which varies with time.

Definition 31 (Time-Varying Collection)**.** Let $L \in \mathbb{N}$ be the prediction horizon from Definition 28. We construct the collection of patterns $\Psi[t]$, with time-varying cardinality $K[t] \in \mathbb{N}$, to be a subset of feasible length-L future sequences of modes given the estimated current mode $\hat{\varphi}_n^{(t)}$:

$$\Psi[t] \triangleq \{\boldsymbol{\psi}_1^{(t)}, \cdots, \boldsymbol{\psi}_{K[t]}^{(t)}\} \subseteq \{\text{feasible } (\alpha_1, \cdots, \alpha_L) | \hat{P}^{(t)}[\hat{\varphi}_n^{(t)}, \alpha_1] > 0, \alpha_i \in \mathcal{X}\}. \tag{3.37}$$

That is, each pattern $\boldsymbol{\psi}_k \in \Psi[t]$ is feasible with respect to $\hat{P}^{(t)}$ in the sense of Definition 20.

A key difference between here and Problem 1 is that we keep the hat and superscript (t) in the τ and q_k quantities because we emphasize they are dependent on $\Psi[t]$ and $\hat{P}^{(t)}$, $\hat{\varphi}_n^{(t)}$ from Section 3.3.1, which change over time. Thus, we can obtain these formulas by extending the results derived from the Markov chain case of the previous Section 3.1 to uncertain statistics $\hat{P}^{(t)}$ and $\hat{\varphi}_n^{(t)}$.

3.3.3 Control Law Design from Memory and Prediction

Let $g : \mathbb{R}^+ \times \mathcal{X} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ be a generic function representing the mode-dependent state-feedback control law designed by the Control Law Design component in Figure 3.2. The Control Law Design component uses the expected occurrence time $\mathbb{E}[\hat{\tau}_n^{(t)}]$ and probabilities $\{\hat{q}_k\}_{k=1}^{K[t]}$ computed from PLP (Section 3.3.2) to store the control policies of previously-occurred patterns and to schedule control policies in advance. This procedure is described more carefully in the following two propositions.

Proposition 1 (Scheduling Future Control Inputs)**.** Suppose we are given the estimated pattern-occurrence quantities $\mathbb{E}[\hat{\tau}_n^{(t)}]$ and $\{\hat{q}_k^{(t)}\}_k$ from PLP. Let $\tau \equiv \mathbb{E}[\hat{\tau}_n^{(t)}]$ be the shorthand

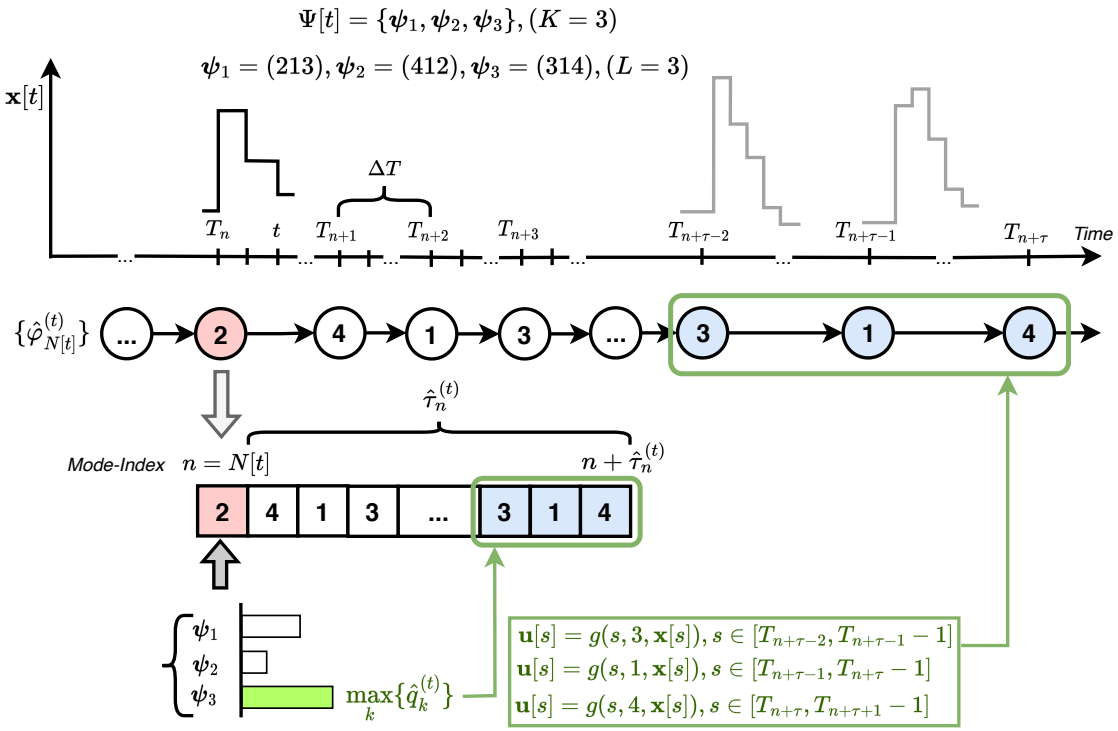

Figure 3.3: A visualization of PLP with the pattern collection of three patterns described in Example 2. Note $L = 3$. The middle row of circles shows the evolution of $\hat{\varphi}^{(t)}_n$ over time, while the bottom row of boxes shows the process on the mode timescale (see Assumption 7). The red circle and box indicate the estimated mode at current time $t \in \mathbb{N}$. The blue circles indicate the expected pattern which is first to occur; in this example, $(3, 1, 4)$ has the highest first-occurrence probability among any pattern in the collection $\Psi[t]$. Control input sequences are then scheduled according to Proposition 1, shown in the green box.

notation (with a temporary abuse of notation) for the estimated expected minimum occurrence time for the specific pattern collection $\Psi[t]$ given estimated current mode $\hat{\varphi}^{(t)}_n$. To schedule a control law in advance, we simply choose the pattern $\boldsymbol{\psi}^{(t)}_k \in \Psi[t]$ corresponding to the largest occurrence probability $\hat{q}^{(t)}_k$. Then, until mode-index τ, the future sequence of control inputs $\mathbf{u}[t : T_{n+\lfloor\tau\rfloor+1} - 1]$ is

$$\begin{aligned} \mathbf{u}[s] &= g(s, \psi^{(t)}_{k,1}, \mathbf{x}[s]), \ s \in [t : T_{n+1} - 1] \\ &\vdots \\ \mathbf{u}[s] &= g(s, \psi^{(t)}_{k,L}, \mathbf{x}[s]), \ s \in [T_{n+\lfloor\tau\rfloor} : T_{n+\lfloor\tau\rfloor+1} - 1]. \end{aligned} \tag{3.38}$$

Aside from operating on a longer timescale (mode process instead of system dynamics),

Proposition 1 is similar in principle to standard model predictive control (MPC): only the first control law in the sequence (3.38), corresponding to the first mode $\psi_{k,1}^{(t)}$, is applied at the next mode-index $n + \lfloor \tau \rfloor$.

Proposition 2 (Storing Past Control Inputs in Memory)**.** Define $\mathcal{U}$ to be a table which maps mode patterns $\boldsymbol{\psi}_k^{(t)}$ to control policies $\{g(t, \psi_{k,1}^{(t)}, \cdot), \cdots, g(t, \psi_{k,L}^{(t)}, \cdot)\}$ and the accumulated state and control trajectories over each occurrence time. When $\boldsymbol{\psi}_k^{(t)} \in \Psi[t]$ is first observed, a new entry $\mathcal{U}[\boldsymbol{\psi}_k^{(t)}](t)$, defined by (3.38) for the specific $\boldsymbol{\psi}_k^{(t)}$, is created. For anticipated future occurrences of $\boldsymbol{\psi}_k^{(t)}$, the system predicts future control inputs using control gain $\mathcal{U}[\boldsymbol{\psi}_k^{(t)}](t)$ in the form of (3.38). The table entry for $\boldsymbol{\psi}_k^{(t)}$ is then updated at every occurrence time after its first.

3.4 Concluding Remarks

In this chapter, we introduced the notion of *pattern-learning* in stochastic systems with repeated behaviors of interest, and discussed how the pattern-occurrence problems (Problem 1) could be used in the design framework of *Pattern-Learning for Prediction (PLP)*. We derived formulas for the pattern-occurrence problem in both the i.i.d. sequence case (Section 3.1.1) and the Markov chain case (Section 3.1.2) to determine the minimum expected occurrence times of patterns and the first-occurrence probabilities. We provided an example of a PLP-based controller architecture (Figure 3.2) for a specific form of linear discrete-time MJS with uncertain statistics. There, the PLP component used the pattern-occurrence formulas for the Markov chain case to take advantage of patterns that occurred in the underlying mode process. Essentially, the controller architecture proposed in this chapter extends traditional uncertain system controllers (which often leverage well-researched techniques in system identification, experience replay, and predictive control) to learn patterns in the system behavior.

In the following chapters, we provide in-depth discussions around the implementation of each component for a variety of applications. We will show that incorporating PLP into the control law design can save time, data consumption, and computational energy because there is no need to devote resources toward re-computing a control action for a pattern that has been observed before.

CONTROL OF DYNAMIC-TOPOLOGY NETWORKS

This chapter describes the first of three applications that demonstrate the effectiveness of learning patterns to efficiently control JSSs: the control of networks that undergo changes in topology. The control of networks which undergo parametric and/or topological changes has become an important and widely-studied problem with recent trends in large-scale systems and smart technology. In literature, an adaptive, consensus-based control scheme for complex networks with time-varying, switching network topology was discussed in [32]. Distributed target-detection and tracking using a dynamic sensor network was studied in [12], while [133] described fault-tolerance against actuator failures in a multiagent system connected by a switching topology network.

In this chapter, the specific type of dynamic-topology network application we consider is that of a power grid which goes through topological changes due to faults like downed power lines. The original power grid dynamics are linearized and modeled as a MJS whose modes correspond to the different topologies. Sequences of line failures over time are chosen to represent the "patterns," and we implement the modular controller architecture from Chapter 3 to perform disturbance-rejection in a way that is robust to these failures. We employ PLP to recognize previously-occurred network topologies, then use the relevant closed-loop response (which was stored in memory) to control it.

Chapter Organization

In Section 4.1, we specialize the MJS dynamics from the previous Section 3.2 to the power grid. Sections 4.2 and 4.3 describe the concrete algorithms chosen to implement the Mode Process ID (see Section 3.3.1) and Control Law Design (see Section 3.3.3) components for this application. In particular, the Control Law Design is implemented with the novel *system level synthesis (SLS)* framework for distributed, localized disturbance-rejection of linear large-scale networks. Our experimental results in Section 4.4 compare the performance of our PLP-based controller architecture, with Mode Process ID and Control Law Design implemented as described previously, against two controllers without PLP. All three controllers are designed around variations of the SLS framework, including a version that is robust to changing

network topology and a version which is data-driven. We discuss several important tradeoffs, such as the size of the pattern collection and the system scale versus the accuracy of the mode predictions, which show how different PLP implementations affect stabilization and runtime performance.

4.1 System Setup

In this chapter, we demonstrate the proposed controller architecture to the following extension of (3.34), which switches among a finite number of different topologies $\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m)), m \in \{1, \cdots, M\}, M \in \mathbb{N}$:

$$\mathbf{x}_i[t+1] = A_{ii}(\xi_{N[t]})\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i(\xi_{N[t]})} A_{ij}(\xi_{N[t]})\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t]. \tag{4.1}$$

Here, $n_s \triangleq |\mathcal{V}|$, $i \in \{1, \cdots, n_s\}$, the neighboring nodes of subsystem i are $\mathcal{N}_i(m) \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}(m)\}$, and $A(m) \triangleq [A_{ij}(m)] \in \mathbb{R}^{n_x \times n_x}$ for each topology $m \in \{1, \cdots, M\}$. Since the topology-switching mechanism abides by a Markov chain, the assumptions from Section 3.2 still hold. The mode process $\{\xi_n\}$ is the index of the current topology at time $t \in \mathbb{N}$ with $N[t]$ being the number of topology changes made by time t.

4.2 Mode Process Identification

Recall from Section 3.3.1, the Mode Process ID component estimates the current mode $\hat{\varphi}^{(t)}_{N[t]}$ and the TPM $\hat{P}^{(t)}$ based on state and control trajectories $\mathbf{x}[0:t]$, $\mathbf{u}[0:t]$. In the context of this dynamic-topology network, $\hat{\varphi}^{(t)}_{N[t]}$ is estimated using *consistent set narrowing*, a variation of *nested convex body chasing* [70] extended specifically to dynamic-topology network systems. Second, $\hat{P}^{(t)}$ is estimated using empirical counts based on $\hat{\varphi}^{(t)}_{N[t]}$ and on estimates of the previous modes $\{\hat{\varphi}^{(s)}_{N[s]}\}_{s=0}^{t-1}$.

4.2.1 Consistent Set Narrowing

Because the distribution of the external noise process $\mathbf{w}[t]$ is unknown other than its norm bound, we employ a consistent set narrowing approach, which checks the set of modes that are "consistent" with the state/control observations. This method is similar to the nested convex body chasing approach described in [70], which was used for model approximation and selection for designing robust controls.

Denote the current mode-index as $n \triangleq N[t] \in \mathbb{N}$. By Assumption 7, there are at most $\Delta T - 1$ state and control observations, $\mathbf{x}[T_n : t]$ and $\mathbf{u}[T_n : t]$, associated with a single mode φ_n. We thus construct a consistent set as follows.

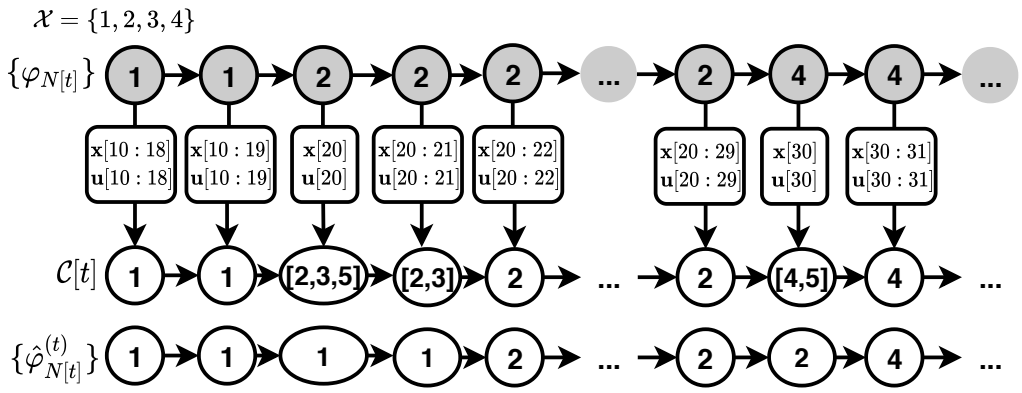

Figure 4.1: A visual diagram depicting Mode Process ID with consistent set narrowing. Here, $\Delta T = 10$ and $M = 5$. With $n \triangleq N[t]$, the upper row of gray circles denotes a realization $\{\varphi_n\}$ of the original unobservable mode process $\{\xi_n\}$. The middle row denotes the evolution of the consistent set, updated via (4.2) using the state and control observations $\mathbf{x}[T_n : t]$, $\mathbf{u}[T_n : t]$. The estimate $\{\hat{\varphi}^{(t)}_{N[t]}\}$ of the mode process is at the bottom row of white circles.

Definition 32 (Consistent Sets)**.** Over time, we construct a sequence of *consistent sets* $\{\mathcal{C}[t]\}_{t\in\mathbb{N}}$ in the following way. For each $n \in \mathbb{N}$, we initially set $\mathcal{C}[T_n] \triangleq \mathcal{X}$ because no observations about the current mode φ_n have been made yet. Then for each $t \in (T_n, T_{n+1})$, if $\mathcal{C}[t-1] \neq \varnothing$, a new consistent set is formed by retaining all modes $m \in \mathcal{C}[t-1]$ from the previous iteration where each one-step value of state and control $(\mathbf{x}[t], \mathbf{x}[t+1], \mathbf{u}[t])$ satisfies the norm-boundedness condition of the noise $\mathbf{w}[t]$:

$$\mathcal{C}[t] = \left\{ m \in \mathcal{C}[t-1] \,|\, \bigwedge_{r=T_n}^{t-1} \mathbb{1}\{\|\mathbf{x}[r+1] - A(m)\mathbf{x}[r] - B\mathbf{u}[r]\|_\infty \leq \overline{w}\} \right\}. \tag{4.2}$$

As (4.2) is being performed for each $t \in \mathbb{N}$, we update the estimate $\hat{\varphi}^{(t)}_{N[t]}$ of the current mode. We use $\hat{\varphi}^{(t)}_{N[t]} \in \operatorname{argmax}_{\zeta\in\mathcal{X}} \hat{P}^{(t-1)}[\hat{\varphi}^{(t-1)}_{N[t-1]}, \zeta]$ if $|\mathcal{C}[t]| > 1$; otherwise, we update $\hat{\varphi}^{(t)}_{N[t]} \in \mathcal{C}[t]$.

Remark 19. One property of consistent set narrowing, also observed in nested convex body chasing approaches [70], is that at each time $t \in \mathbb{Z}^{\geq 0}$, the consistent set $\mathcal{C}[t]$ always contains the true mode $\varphi_{N[t]}$. This is by definition of the consistent set, and the deterministic nature of the condition (4.2) which defines the narrowing process (equivalent to verifying a simple linear inequality). In the MJS literature, there have been notions of consistency similar to (4.2) according to which unknown modes of MJS are estimated. For example, [136] verifies consistency under the assumption that imperfect measurements $\mathbf{y}[t] \neq \mathbf{x}[t]$ of the state $\mathbf{x}[t]$ are collected. Thus, instead of using the state history $\mathbf{x}[0:t]$ directly, the consistency condition

is designed around the collected measurements $\mathbf{y}[0:t]$ and propagated estimates of $\mathbf{x}[0:t]$ based on the initial condition $\mathbf{x}_0$ and the measurement equation.

Mode detectability is also a concept that has been studied; for instance, in [37], the mode variable (analogous to $\xi_t \in \mathcal{X}$ in our notation) emits its own signal (analogous to $\hat{\varphi}_t^{(t)}$ in our notation) independently of the system dynamics and the previous modes. We note the subscripts of t instead of n because the mode process in [37] is assumed to operate on the same timescale as the system dynamics. In the consistent set narrowing approach, we obtain $\hat{\varphi}_n^{(t)}$ from the state and control trajectories $(\mathbf{x}[t], \mathbf{x}[t+1], \mathbf{u}[t])$; this estimate also changes with time as we collect more data about the trajectories.

4.2.2 Empirical Estimation of the TPM

For any $n \in \mathbb{N}$, the estimate of φ_n is most accurate when the maximum possible amount of data from the system has been obtained to create the estimate, i.e., among all $t \in [T_n, T_{n+1})$, the value of $\hat{\varphi}_n^{(t)}$ is most accurate at time $t = T_{n+1} - 1$. For general $T_{N[t]} < t < T_{N[t]+1}$, $\hat{P}^{(t)}$ is estimated based on $\hat{\varphi}_{N[t]}^{(t)}$ and only the most accurate estimates of the previous modes $\{\hat{\varphi}_{N[s]}^{(T_{N[s]}-1)}\}_{s=0}^{t-1}$. Thus, in the TPM estimation procedure, there is only one estimate associated each true mode φ_n. For simplicity of notation in this section only, we fix $n \triangleq N[t]$ and denote shorthand $\hat{\varphi}_{n'} \equiv \hat{\varphi}_{n'}^{(T_{n'}-1)}$ for $n' < n$ and $\hat{\varphi}_n \equiv \hat{\varphi}_n^{(t)}$.

If $t = T_n$ for some $n \in \mathbb{N}$, estimating $\hat{P}^{(t)}$ given $\{\hat{\varphi}_{n'}\}_{n'=1}^{n}$ is straightforward. By Assumption 8, it is known which entries of the TPM are nonzero. Thus, we initialize $\hat{P}^{(t)}$ to be an $M \times M$ matrix with a 1 in the nonzero entries; when normalized, this corresponds to a stochastic matrix which has uniform distribution over the feasible transitions (e.g., 1/3 probability each for a row with three nonzero entries) but for estimation purposes, we keep the estimate of the TPM unnormalized until the end of the simulation duration. For each consecutive pair of transitions $(\hat{\varphi}_{n'}, \hat{\varphi}_{n'+1})$ for $n' \in \{0, \cdots, n-1\}$, we take $\hat{P}^{(t)}[\hat{\varphi}_{n'}, \hat{\varphi}_{n'+1}] = \hat{P}^{(t)}[\hat{\varphi}_{n'}, \hat{\varphi}_{n'+1}] + 1$.

If $T_n < t < T_{n+1}$ for some $n \in \mathbb{N}$, we have two separate subcases. If $\hat{\varphi}_n^{(t-1)} = \hat{\varphi}_n^{(t)}$, then we simply follow the approach above and compute $\hat{P}^{(t)}$ using the sequence $\{\hat{\varphi}_{n'}\}_{n'=1}^{n}$. Otherwise, if $\hat{\varphi}_n^{(t-1)} \neq \hat{\varphi}_{N[t]}^{(t)}$, then we again follow the approach above and compute $\hat{P}^{(t)}$, but using the sequence $\{\hat{\varphi}_{n'}\}_{n'=1}^{n-1}$ instead. To incorporate the mode estimate at current mode-index n, we first need to reset the TPM estimate of the last transition via $\hat{P}^{(t)}[\hat{\varphi}_{n-1}, \hat{\varphi}_n^{(t-1)}] = \hat{P}^{(t)}[\hat{\varphi}_{n-1}, \hat{\varphi}_n^{(t-1)}] - 1$; then we update as usual $\hat{P}^{(t)}[\hat{\varphi}_{n-1}, \hat{\varphi}_n^{(t)}] = \hat{P}^{(t)}[\hat{\varphi}_{n-1}, \hat{\varphi}_n^{(t)}] + 1$. Once the mode sequence estimates have been processed until current time t, we update $\hat{P}^{(t)}$ such that each row is

normalized to sum to 1.

Remark 20. The need for including Mode Process ID in the controller architecture Figure 3.2 is closely related to the notion of *mode observability*, which has been studied extensively in the literature [150, 2, 11, 136]. One common setup is that the measurements come from a (linear) noisy measurement equation such that $\mathbf{y}[t] \neq \mathbf{x}[t]$, and derive mode observability conditions from the imperfect observations $\mathbf{y}[t]$ of the state $\mathbf{x}[t]$. Also, the mode process is assumed to operate on the same timescale as the system dynamics. Compared to these methods, the algorithms we chose for implementing Mode Process ID hinge upon assumptions that simplify the mode observability problem. For example, in Assumption 8, the state $\mathbf{x}[t]$ is observable and in Assumption 7, we fix the mode switching times to be constant and deterministic rather than stochastic.

We again emphasize that this is because the focus of our paper is on the impact of PLP on control design rather than mode observability, and we aimed to set up a simple scenario to show that our approach can be used when the system has uncertainties. Thus, not all of our assumptions are limiting; for example, compared to our approach, [150] explicitly imposes that the external noise processes $\{\mathbf{w}[t]\}_t, \{\mathbf{v}[t]\}_t$ are Gaussian white and neither [150] nor [2] consider the impact of control.

Remark 21. We qualitatively discuss some conditions for mode observability in our specific implementation of Mode Process ID. First, the modes $\{A(1), \cdots, A(M)\}$ cannot be too "similar" to each other with respect to a certain metric d, (e.g., if $d(A(m_1), A(m_2)) < \epsilon$ for some threshold $\epsilon > 0$ and two distinct modes $m_1 \neq m_2$ and $m_1, m_2 \in \mathcal{X}$). Second, when ΔT is too short, the consistent set may not converge to a single mode even if $d(A(m_1), A(m_2)) \geq \epsilon$ for all pairs $(m_1, m_2) \in \mathcal{X}$ such that $m_1 \neq m_2$. Rigorous derivation of these conditions for our specific case are deferred to future work. This includes designing d and ϵ for the consistent set narrowing approach, and deriving conditions on ΔT and the set $\{A(1), \cdots, A(M)\}$ for guaranteed convergence towards a singleton consistent set. Although these conditions are contingent upon our simplifying assumptions, they are expected to be similar to those derived in the aforementioned literature.

4.3 Control Law Design

In this section, we describe our choice of implementation for Memory and Prediction, then tie it into our choice of implementation for the the Control Law Design component. The design of Memory is inspired by experience replay techniques that are commonly employed

for reinforcement learning, in particular, episodic memory for control [91, 20]. One well-known control method that explicitly incorporates the predictions is model predictive control (MPC), and so we address Prediction and Control Law Design simultaneously. For the purposes of our topology-switching network case study here, we also implement non-predictive Control Law Design using the novel system level synthesis approach [152, 5], including a topology-robust version (see Section 4.3.3) and a data-driven version [160, 3].

4.3.1 Incorporating Predictions

Our implementation of the memory component in this controller architecture is achieved using a table $\mathcal{U}$ which maps patterns of interest to the optimal control sequences we designed for them in our experiment so far (see Proposition 2); this also includes explicit state and control trajectories. This implementation was inspired by *episodic memory* [91] which can be added to learning-based control methods (e.g., reinforcement learning) to recall specific experiences and their rewards [20]. Our table $\mathcal{U}$ is implemented according to Proposition 2 and its entries are updated in two ways: 1) the control law is updated in an entry for an existing pattern, or 2) a new entry is created for a newly-observed pattern $\boldsymbol{\psi}$ at time t, where $\boldsymbol{\psi} \in \Psi[t+1]$ but $\boldsymbol{\psi} \notin \Psi[t]$. We describe the control law synthesis and update procedures in the following Section 4.3.2.

For the prediction component, we specifically recall model predictive control (MPC). Standard MPC for discrete-time linear dynamics seeks to predict a future sequence of controls $\{\mathbf{u}[t], \mathbf{u}[t+1], \cdots, \mathbf{u}[t+H]\}$ which minimizes some cost functional at each timestep $t \in \mathbb{N}$, for some prediction horizon $H \in \mathbb{N}$. Once the first control input $\mathbf{u}[t]$ is applied to the system, the procedure is repeated at the next time $t+1$. Although intuitive, incorporating both short-term and long-term predictions for online control have been proven to be beneficial, even when the system to be controlled is perturbed by either random and adversarial disturbances [31]; in [164], this is demonstrated explicitly with the linear quadratic regulator. For concreteness, we are inspired by the methods of [117] and [101], which discuss MPC for MJSs, and we extend their approaches to our setting (Section 4.1).

We remark that H, like prediction horizon L for the mode process, is a user-chosen hyperparameter; one reasonable choice could be to make it time-varying and set it equal to $\Delta T - (t - T_{N[t]})$ at each t. Given the estimated current mode $m \triangleq \hat{\varphi}^{(t)}_{N[t]}$, the cost function we

seek to optimize is the following mode-dependent quadratic cost function:

$$J(t,m) \triangleq \sum_{s=t}^{H} (\mathbf{x}[s]^\top Q(m)\mathbf{x}[s] + \mathbf{u}[s]^\top R(m)\mathbf{u}[s]) + \mathbf{x}[H]^\top Q_f(m)\mathbf{x}[H]. \tag{4.3}$$

The main distinction is that the prediction part of MPC is done on the estimated mode process instead of the system dynamics. Let $t \in \mathbb{N}$ and $n \triangleq N[t]$, and suppose the consistent set narrowing approach of Section 3.3.1 estimates the current mode to be $\hat{\varphi}_n^{(t)}$. Again, by Assumption 7, there are at most $\Delta T - 1$ state and control observations $\mathbf{x}[T_n : t]$ and $\mathbf{u}[T_n : t]$ associated with each mode φ_n. Thus, for the control input $\mathbf{u}[t] = K(t, \hat{\varphi}_n^{(t)})\mathbf{x}[t]$ at time t, the gain $K(t, \hat{\varphi}_n^{(t)}) \in \mathbb{R}^{n_x \times n_u}$ associated specifically with mode $\hat{\varphi}_n^{(t)}$ can be designed using standard linear optimal control tools such as LQR minimization.

4.3.2 System Level Synthesis

For the purposes of this application, we employ the novel *system level synthesis* (SLS) [152, 5] approach for distributed disturbance-rejection in linear discrete-time network systems with static topologies $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$, expressed as

$$\mathbf{x}[t+1] = A\mathbf{x}[t] + B\mathbf{u}[t] + \mathbf{w}[t]. \tag{4.4}$$

The standard state-feedback control law for systems of this form is given by $\mathbf{u}[t] = K\mathbf{x}[t]$ and in z-transform expression, the resulting closed-loop system is given by $\mathbf{x} = (zI - A - BK)^{-1}\mathbf{w}$. However, for large-scale systems (i.e., large-dimensional matrices A and B), optimizing over the transfer function $(zI - A - BK)^{-1}$ by solving for K is difficult. Thus, a key feature of SLS is that it reparametrizes the control problem: instead of designing just the open-loop feedback gain K, SLS designs for the entire closed-loop system via response maps $\mathbf{\Phi} \triangleq \{\mathbf{\Phi}_x, \mathbf{\Phi}_u\}$ such that $\mathbf{x}[0:t] = \mathbf{\Phi}_x \mathbf{w}[0:t]$ and $\mathbf{u}[0:t] = \mathbf{\Phi}_u \mathbf{w}[0:t]$, where $\mathbf{w}[t]$ is an additive external disturbance. The main result can be summarized as follows.

Lemma 13. For the linear, discrete-time static dynamics (4.4), the following are true. First, the affine subspace described by

$$\begin{bmatrix} I - Z\hat{A} & -Z\hat{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I \tag{4.5}$$

parametrizes all possible system responses $\mathbf{\Phi}$, where $\hat{A} \triangleq \text{blkdiag}(A, \cdots, A, \mathbf{0}) \in \mathbb{R}^{Hn_x \times Hn_x}$, $\hat{B}$ is defined similarly, Z is the block-downshift operator, $n_x \in \mathbb{N}$ is the state dimension,

and $H \in \mathbb{N}$ is a chosen finite horizon over which control is performed. Second, for any $\mathbf{\Phi}$ which satisfies the condition in (4.5), the feedback gain $K \triangleq \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$ achieves the desired internally-stabilizing system response.

Then the state-feedback controller is implemented as follows:

$$\hat{\mathbf{x}}[t] = \sum_{k=2}^{T} \Phi_x[k]\hat{\mathbf{w}}[t+1-k], \quad \hat{\mathbf{w}}[t] = \mathbf{x}[t] - \hat{\mathbf{x}}[t], \quad \mathbf{u}[t] = \sum_{k=1}^{T} \Phi_u[k]\hat{\mathbf{w}}[t+1-k], \tag{4.6}$$

with the controller's internal state $\hat{\mathbf{w}}$ and *system responses* $\{\Phi_x, \Phi_u\}$, which are closed-loop transfer function maps defined as $\mathbf{x} = \Phi_x \mathbf{w}$ and $\mathbf{u} = \Phi_u \mathbf{w}$. These transfer function maps are constrained to finite time horizon H, for which we will denote $\{\Phi_x, \Phi_u\} \in \mathcal{F}_H$. It was shown in [104] that even when this relationship is approximately satisfied, the implementation (4.6) produces a stable closed-loop response.

Definition 33. We associate a local h-hop set $\mathcal{L}_{i,h}$ with each system $i \in \mathcal{V}$ to be the set of systems j for which the (i,j)th entry of G^h is nonzero, where $G \in \{1,0\}^{M \times M}$ is the adjacency matrix of A in (4.4). The system response is said to be h-localizable iff for every $i \in \mathcal{V}, j \notin \mathcal{L}_{i,h}$, we have $\Phi_{x,ij} = 0$, and analogously for Φ_u. We denote this as $\{\Phi_x, \Phi_u\} \in \mathcal{L}_h$.

The $\mathbf{\Phi}$ implementation also makes SLS more suitable for distributed and localized control law design in large-scale linear systems, and so $\mathbf{\Phi}$ is often implemented as $\mathbf{\Phi}^{(i)} \triangleq \{\Phi_x^{(i)}[s], \Phi_u^{(i)}[s]\}$ for each node $i \in \mathcal{V}$ and its local subsystem $\mathcal{L}_{i,h}$. Here, $s \in \{1, \cdots, H\}$ is the index of the *spectral component*. Both time horizon H and number of neighboring hops h are parameters chosen by design based on properties such as the scale and topology of $\mathcal{G}$.

The desired behavior can then be achieved by constraining $\{\Phi_x, \Phi_u\}$ to lie in an appropriate convex set $\mathcal{S}$, and solving an optimization problem of the form:

$$\min_{\{\Phi_x, \Phi_u\}} f(\Phi_x, \Phi_u, Q, R) \text{ s.t. } \{\Phi_x, \Phi_u\} \in \mathcal{S}, \tag{4.7}$$

where $Q \in \mathbb{R}^{N_x \times N_x}, R \in \mathbb{R}^{N_u \times N_u}$ are cost matrices which assign weight to Φ_x, Φ_u respectively. The set $\mathcal{S}$ typically includes system-to-system communication delay constraints as well as the necessary robustness constraints to keep the closed-loop response stable during the process of learning the uncertainties.

4.3.3 Robust Adaptation to Dynamic Topologies

We can also extend SLS to account for dynamic topologies $\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m))$ for $m \in \mathbb{N}$ representing the index of the topology. Let $\mathbf{\Phi}_m^{(i,t)} \triangleq \{\Phi_{x,m}^{(i,t)}[s], \Phi_{u,m}^{(i,t)}[s]\}$ define the ith local response map $\mathbf{\Phi}^{(i)}$ which is created specifically for topology $m \in \{1, \cdots, M\}$. As we demonstrate for this case study, the mode in our original dynamics (3.34) corresponds to the index of the current topology the system is in. *Topology-robust SLS* (see Section 4.3.3) essentially attempts to design a single $\{\Phi_x, \Phi_u\}$ response that can simultaneously stabilize multiple topologies (i.e., distinct A matrices). Conditions for *simultaneous stabilization* for a collection of discrete-time LTI systems have been studied extensively in past literature: some results (e.g., [19]) express the condition by ensuring that the closed-loop transfer function between every possible plant-controller pair does not have any pole-zero cancellations, while others (e.g., [26]) derive conditions based on the algebraic Riccati equation. To keep our discussion focused, we do not state these conditions here.

Centralized Implementation

We begin with a nominal topological structure A^* of the network. We are aware that at least one link has been disconnected, and although we do not know which one(s), we are given a finite collection of K candidate link failure matrices D, one of which gives us the true topology $A = A^* + D$. This setup is consistent with real-world scenarios where we are oftentimes able to vaguely identify the local region in which a potential link failure has occurred. We assume that none of the candidate matrices causes the graph to become disconnected.

With this premise, the system dynamics are given by $\mathbf{x}[t+1] = (A^* + D)\mathbf{x}[t] + B\mathbf{u}[t] + \mathbf{w}[t]$. The matrix A^* denotes the known nominal system and link failures D enter in the form of perturbations to A^*.

To characterize the set of D, we introduce basis matrices $\mathcal{A}_l$ to encode all possible single-link modifications so that linear combinations can be used to model a general number of failures corresponding to each candidate D. We will denote this set as $\mathcal{P}_0$, and formally refer to it as the initial *consistent set*:

$$\mathcal{P}_0 := \left\{ \sum_{l=1}^{M} \xi_l \mathcal{A}_l \ : \ \xi_l \in \{-1, 0, 1\} \right\}, \tag{4.8}$$

where coefficient $\xi_l = 1$ is for when a link is added, $\xi_l = -1$ for when a link is deleted, $\xi_l = 0$ for when a link is unchanged. Because it is a discrete combinatorial set, we will impose $K << 2^M$ to make the problem tractable.

At each timestep, the consistent set is updated using new observations of $(\mathbf{x}[t+1], \mathbf{x}[t], \mathbf{u}[t])$:

$$\mathcal{P}_{t+1} := \left\{ D \in \mathcal{P}_t \ : \ \left\| \mathbf{x}[t+1] - \left(A^* + \sum_{l=1}^{M} \xi_l \mathcal{A}_l \right) \mathbf{x}[t] - B\mathbf{u}[t] \right\|_\infty \leq \eta \right\}. \tag{4.9}$$

We will now use SLS to design the controller $\left\{ \Phi_x^{(t)}, \Phi_u^{(t)} \right\}$, where the superscript (t) is included to show that the control laws may change over time as more of the topology is learned. In the context of our topology adaptation problem, the following inequalities should be satisfied:

$$\sum_{k=1}^{T} \left\| \Delta_k(A', B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)}) \right\| \leq \lambda_t \ \ \forall A' = A^* + D, \ D \in \mathcal{P}_t, \tag{4.10a}$$

$$\left\| \sum_{k=1}^{T} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)} \right) [k+1] \hat{\mathbf{w}}[t-k] \right\| \leq \gamma. \tag{4.10b}$$

λ_t is referred to as the *robustness margin* and for each timestep t it determines whether the controller is stabilizable with the tth polytope of uncertainties. γ is the *adaptation margin* and ensures that the system response Φ_x doesn't fluctuate wildly with largely-varying $\mathbf{w}$.

The full optimization problem for centralized robust control which adapts to topological changes is hence presented:

$$\min_{\left\{ \Phi_x^{(t)}[k], \Phi_u^{(t)}[k] \right\}_{k=1}^{T}, \lambda_t} f\left(\Phi_x^{(t)}, \Phi_u^{(t)}, Q, R \right) = \begin{cases} \lambda_t & \text{if } \lambda_t \leq \lambda^* \\ \sum_{k=1}^{T} \left\| Q\Phi_x^{(t)}[k] + R\Phi_u^{(t)}[k] \right\|_1 & \text{else} \end{cases}. \tag{4.11}$$

s.t. $\{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{F}_T$ and (4.10)

The two separate steps expressed in the objective function above are taken because optimizing for a performance objective is only reasonable if robust stability is feasible with uncertainty $\mathcal{P}_t$.

Remark 22. In implementation, the inclusion of (4.10b) to $\mathcal{S}$ is made optional. This is because the incorrect system response may be learned and closely adhered to for the rest of time if γ is chosen too small, resulting in an unstable controller. This is problematic in the case of topological uncertainties, where the sparsity patterns of all the candidate topologies D may be different.

Localized Implementation

A localized version of the algorithm essentially decomposes (4.11) into multiple independent subproblems. For system i, the submatrix A_i of consideration only includes the rows of A corresponding to the systems in $\mathcal{L}_d(i)$ (Definition 33). This means each system only keeps track of link modifications within its own local subset. Further, let d_c be the communication delay matrix between systems of the network, defined as $d_c(i,j) = |j-i|$ if $j \in \mathcal{L}_d(i)$, and ∞ otherwise.

Each system i begins with a local initial consistent set $\mathcal{P}_0^{(i)}$, defined the same way as in (4.8) but instead with M_i basis matrices $\mathcal{A}_l^{(i)}$ which have dimensions equal to A_i. Each consistent set is locally updated from $\mathcal{P}_t^{(i)}$ to $\mathcal{P}_{t+1}^{(i)}$ in a fashion similar to (4.9).

To design local controllers, we solve a local optimization problem of the form (4.7) for the ith columns of the system response matrices $\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}$. The constraints follow analogously to (4.10). Define the submatrix

$$\Delta_k^j\left(A, B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) := \Phi_{x,ji}^{(t)}[k+1] - \sum_{l\in\mathcal{N}(j)} A_{jl}\Phi_{x,li}^{(t)}[k] - B_j\Phi_{u,ji}^{(t)}[k], \tag{4.12}$$

where $i, j \in \mathcal{V}$, $k = 1, \cdots, T$, and t iterates over the simulation time. This allows us to define our robustness margin constraints

$$\left\| \sum_{j\in\mathcal{L}_d(i)} \Delta_k^j\left(A', B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) \right\| \le c_i\rho^{k-1} \quad \forall A' = A^* + D, D \in \mathcal{P}_t \quad \forall k \le T-1, \tag{4.13a}$$

$$c_i \sum_{k=1}^{T} \rho^{k-1} \le \lambda_t^{(i)} + \epsilon, \ \epsilon \ge 0. \tag{4.13b}$$

Unlike the centralized formulation, instead of a constant, we introduce $\rho > 0$ and $c_i > 0$ to ensure faster exponential convergence to zero, which is motivated by the possibility of local disturbances propagating throughout the network in a cascading manner if it is not killed quickly enough within the local region.

The full optimization problem for localized robust, topologically-adaptive control is hence presented:

$$\min_{\left\{\Phi_x^{(t)}[k], \Phi_u^{(t)}[k]\right\}_{k=1}^{T}, \lambda_t, \epsilon} f\left(\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}, Q_i, R_i\right) = \sum_{k=1}^{T} \left\| Q\Phi_{x,i}^{(t)}[k] + R\Phi_{u,i}^{(t)}[k] \right\|_1 + r_c\epsilon. \tag{4.14}$$

s.t. $\{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{L}_d \cap \mathcal{F}_T$ and (4.13)

Remark 23. The objective function described here is equivalent to the two-part objective function in (4.11). The slack variable ϵ (scaled by fixed $r_c > 0$) helps reduce the two-part process into a single-step optimization problem; when ϵ is very large, the optimization problem effectively focuses on shrinking the polytope until ϵ is sufficiently reduced.

Iterative Implementation for Dynamic Topologies

The iterative localized robust extension of SLS for the original topology-switching network dynamics (4.1) is a simple variation of the localized implementation from the previous discussion, namely with time-varying local sets $\mathcal{L}_{i,h}^{(t)}$. According to [116, 157], consensus among distributed systems is achievable with time-varying topologies, under conditions such as joint-connectedness (the union of the graphs in the entire collection of topologies is connected) among topologies that are visited infinitely many times. The initial true system topology $A(\xi_0)$ (with adjacency matrix $G(\xi_0)$) is known.

Each subsystem keeps a nominal topology estimate $A^{(i)}(\alpha^*(t))$ and updates it whenever it detects that a switch has been made. The transition probability matrix P of the chain is unknown to the system, and each subsystem maintains an estimate $\hat{P}^{(i)}$, which it updates both locally and via simple averaging with the values of its other neighbors [157]. Since the methodology is the same across all subsystems $i \in \mathcal{V}$, the subscript i is henceforth removed for notational simplicity.

Similar to (4.8), the initial consistent sets are formed from M_k basis matrices $\mathcal{A}_\ell^{(i,k)}$ where $k = 1, \cdots, K$, $i \in \mathcal{V}$, and the collective modification is expressed as a linear combination of these bases. At each timestep t, an observation $\mathbf{x}[t]$ is made from the system (4.1). We identify which coefficients remain consistent with the system dynamics $(\mathbf{x}[t], \mathbf{x}[t-1], \mathbf{u}[t-1])$ by updating the consistent set in a fashion similar to (4.9) for each i. Because identification for each system i was only done using information local to i, additional consensus may be performed to further narrow down the consistent set in order to estimate the state of the Markov chain more precisely.

As before, it is most important to maintain system stability while this identification and consensus process is being done. To construct a topologically-robust controller $\{\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}\}$ for each system i, we simply solve the optimization problem (4.14) with the same communication delay matrix d_c defined as before and time horizon H.

4.3.4 A Data-Driven Formulation

To be able to use PLP with the SLS approach, we require a formulation of SLS which is driven by data. Toward that end, we leverage *data-driven SLS* [160, 3], which extends traditional SLS using a characterization based on Willems' fundamental lemma [156], which parametrizes state and input trajectories based on past trajectories under the conditions of persistence of excitation. Define the Hankel matrix

$$\hat{H}_r(\mathbf{x}[0:H]) \triangleq \begin{bmatrix} \mathbf{x}[0] & \mathbf{x}[1] & \cdots & \mathbf{x}[H-r] \\ \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \mathbf{x}[H-r+1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}[r-1] & \mathbf{x}[r] & \cdots & \mathbf{x}[H-1] \end{bmatrix}$$

for finite time horizon H and some $r \in \mathbb{N}$. We say the finite-horizon state trajectory $\mathbf{x}[0:H]$ is *persistently-exciting* of order r if $\hat{H}_r(\mathbf{x}[0:H])$ is full rank. In the data-driven formulation of SLS, the achievable subspace described by (4.5) can be equivalently written as the set

$$\left\{ \begin{bmatrix} \hat{H}_H(\mathbf{x}[0:H]) \\ \hat{H}_H(\mathbf{u}[0:H]) \end{bmatrix} G \;\middle|\; G \text{ s.t. } \hat{H}_1(\mathbf{x}[0:H])G = I \right\}. \tag{4.15}$$

Now, let $n \in \mathbb{N}$ and $n' \in \mathbb{N}$, $n' > n$, be such that at times T_n and $T_{n'}$, the system (3.34) have switched to the same mode $m \in \mathcal{X}$. For our PLP approach, the state/control trajectories $\{\mathbf{x}[T_{n-1}:T_n-1], \mathbf{u}[T_{n-1}:T_n-1]\}$ and $\{\mathbf{x}[T_{n'-1}:T_{n'}-1], \mathbf{u}[T_{n'-1}:T_{n'}-1]\}$ can be collectively used to design the optimal control law for mode m, i.e., we use horizon $T_{n-1}:T_n-1$ in place of $[0:H]$ in (4.15). To implement memory, we store (in $\mathcal{U}$) previous trajectories of the system corresponding to the same mode, and continue to append to it as the simulation progresses. To apply Proposition 1, SLS is run more than once to compute a new $\mathbf{\Phi}$ for every new estimated mode $m \triangleq \hat{\varphi}_n^{(t)}$, hence the dependence of $\mathbf{\Phi}_m^{(i,t)}$ on time $t \in \mathbb{N}$. By Proposition 2, the $\mathbf{\Phi}_m^{(i,t)}$ are stored and updated over time in the table $\mathcal{U}$.

4.4 Experiment Results

The overall control objective is to minimize the mode-dependent quadratic cost function (4.3) subject to constraints imposed by various implementations of SLS from Section 4.3.2. Namely, we consider three versions of the controller architecture Figure 3.2; a visual distinction among the three is shown in Figure 4.2:

- **Baseline** [*First row of Figure 4.2*]: here, Figure 3.2 is implemented only using Mode Process ID; both PLP and MPC are not used. Control Law Design uses the basic SLS

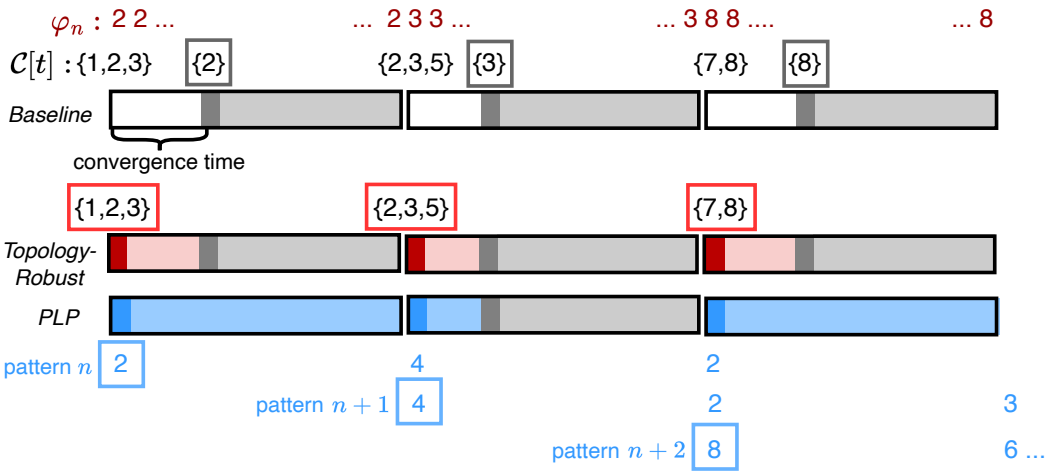

Figure 4.2: The time-varying control law for each of the three versions of the controller architecture, designed based on the estimated mode $\hat{\varphi}_n^{(t)}$ and the consistent set $\mathcal{C}[t]$. Each horizontal bar represents a time duration of length ΔT. The baseline uses the previous law until the consistent set converges to a singleton set (white sub-bars). Topology-Robust is able to control multiple modes simultaneously, so it uses a robust law (red sub-bars) until the convergence. PLP (future horizon $L=3$) uses the law corresponding to the predicted next mode (blue sub-bars) until convergence; note that when the mode in the converged consistent set is equivalent to the predicted next mode, the control policy need not be changed.

approach from Section 4.3.2. We minimize the cost (4.3) subject to the achievability constraint described by (4.5) and the locality constraint described with the sets $\{\mathcal{L}_{i,h}\}_{i\in\mathcal{V}}$. Because the topology changes over time and basic SLS is not designed for time-varying topologies, this requires the optimization to be solved multiple times.

- **Topology-Robust** [*Second row of Figure 4.2*]: we have the same architecture as above, but SLS is replaced with the method of Section 4.3.3, an extension of SLS to network dynamics under time-varying topological changes. A single common control law $\mathbf{\Phi}^{(i,t)}$ is designed for all consistent modes in $\mathcal{C}[t]$, and this common law is used until time $t^* > t$ when $|\mathcal{C}[t^*]| = 1$, after which standard SLS is used.

- **PLP** [*Third row of Figure 4.2*]: we combine the original architecture proposed by Figure 3.2 with the extended SLS approach described in Section 4.3.2. We minimize the cost (4.3) subject to the data-driven achievability constraint described by (4.15) and the locality constraint described with the sets $\{\mathcal{L}_{i,h}\}_{i\in\mathcal{V}}$. Given pattern collection $\Psi[t]$ at time $t\in\mathbb{N}$ and mode-index $n \triangleq N[t]$, if $\boldsymbol{\psi} \triangleq (\psi_1, \cdots, \psi_L) \in \Psi[t]$ is expected to occur at mode-index $n + \mathbb{E}[\hat{\tau}_n^{(t)}] \in \mathbb{N}$, the control law for node $i \in \mathcal{V}$ is scheduled to be $\mathbf{\Phi}_m^{(i,s)}$,

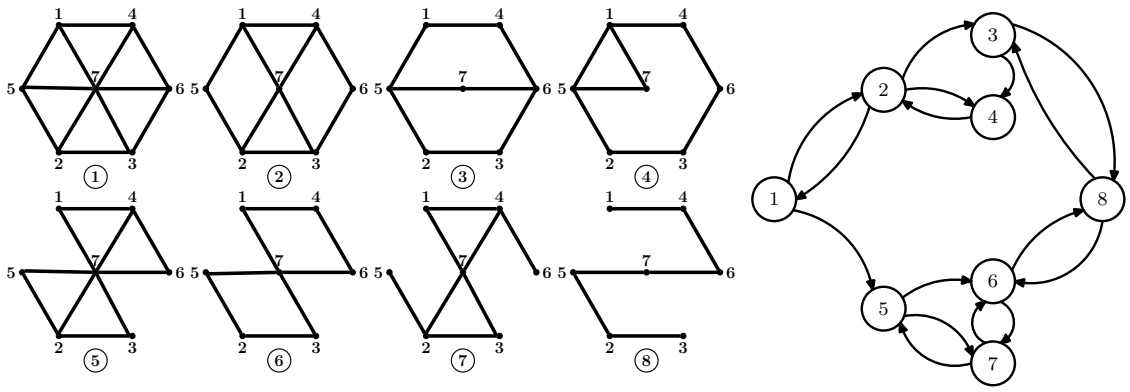

Figure 4.3: [Left] The different possible topologies of the Hexagon System. [Right] The underlying Markov chain for topology transitions.

where $m = \psi_1$ and $s \in [T_{n+\mathbb{E}[\hat{\tau}_n^{(t)}]}, t^*)$, where T_n is defined in Assumption 7 and t^* is the time after $T_{n+\mathbb{E}[\hat{\tau}_n^{(t)}]}$ when $|\mathcal{C}[t^*]| = 1$. For times $s \in [t, T_{n+\mathbb{E}[\hat{\tau}_n^{(t)}]})$ where a prediction is not available, we revert to the baseline controller.

The three architectures are each tested on two specific network systems of the form given in (4.1).

- **(Small-Scale) Hexagon System**: the network system (4.1) consists of a hexagonal arrangement of $n_s = 7$ nodes and $M = 8$ topologies (see Figure 4.3). When PLP is included, the collection of patterns $\Psi[t]$ is constructed with equality in (3.37); hence, Problem 1 become easy to solve–every ending string in $\mathcal{S}$ is an initial-ending string, $\mathbb{E}[\hat{\tau}_n^{(t)}] = L$ for each $t \in \mathbb{N}$, $n \triangleq N[t]$, and determining $\text{argmax}_k\{\hat{q}_k^{(t)}\}$ reduces to a maximum likelihood problem.
- **(Large-Scale) Rectangular Grid System**: the network system (4.1) consists of a 10×10 rectangular grid arrangement of $n_s = 100$ nodes and $M = 20$ topologies (see Figure 4.4). The true TPM is a $M \times M$ stochastic matrix with no self-transitions. When PLP is included, the collection $\Psi[t]$ is constructed with strict subset in (3.37), which means the pattern-occurrence formulas from Theorems 10 and 11 must be used to solve Problem 1.

For both hexagon and grid systems, the specific A and B matrices in (4.1) are the linearized discrete-time power grid dynamics given detailed below. Following the model used in [152],

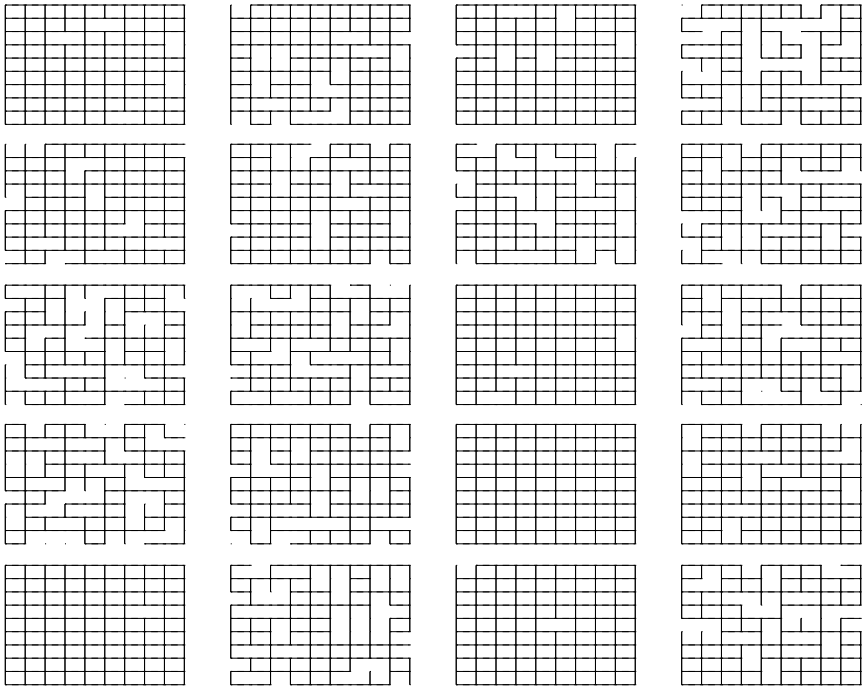

Figure 4.4: The different possible topologies of the 10×10 Rectangular Grid System.

power networks are composed of several synchronized oscillators coupled together to operate under the following nonlinear second-order ODE swing dynamics

$$c_i\ddot{\theta}_i + b_i\dot{\theta}_i = -\sum_{j\in\mathcal{N}_i} a_{ij}(\theta_i - \theta_j) + w_i + u_i,$$

where for the ith system (typically called a bus) of the network, c_i is its inertia, b_i is a damping factor, w_i is the external disturbance, and u_i is the control action. The state of each system is described by the relative phase angle θ_i between the bus's rotor axis and resultant magnetic field, as well as its derivative, the frequency $\dot{\theta}_i$. Use $\mathbf{x}^{(i)} = \begin{bmatrix}\theta_i & \dot{\theta}_i\end{bmatrix}^T$ and rewrite the dynamics in matrix form as:

$$\begin{bmatrix}\dot{\theta}_i \\ \ddot{\theta}_i\end{bmatrix} = \begin{bmatrix}0 & 1 \\ -\frac{a_i}{c_i} & -\frac{b_i}{c_i}\end{bmatrix}\begin{bmatrix}\theta_i \\ \dot{\theta}_i\end{bmatrix} + \sum_{j\in\mathcal{N}(i)}\begin{bmatrix}0 & 0 \\ \frac{a_{ij}}{c_i} & 0\end{bmatrix}\begin{bmatrix}\theta_j \\ \dot{\theta}_j\end{bmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix} w_i + \begin{bmatrix}0 \\ 1\end{bmatrix} u_i, \quad a_i := \sum_{j\in\mathcal{N}(i)} a_{ij}. \quad (4.16)$$

For our experiments, we use the same power grid parameters as in [152]. Discretize the system with sampling time Δt (distinct from ΔT) using the mean-value theorem to obtain

the following approximated model:

$$\begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}[t+1] = \underbrace{\begin{bmatrix} 1 & \Delta t \\ -\frac{a_i}{c_i}\Delta t & 1-\frac{b_i}{c_i}\Delta t \end{bmatrix}}_{\triangleq A_{ii}} \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}[t] + \sum_{j\in\mathcal{N}(i)} \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{a_{ij}}{c_i}\Delta t & 0 \end{bmatrix}}_{\triangleq A_{ij}} \begin{bmatrix} x_1^{(j)} \\ x_2^{(j)} \end{bmatrix}[t] + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} w_i[t] + \underbrace{\begin{bmatrix} 0 \\ \Delta t \end{bmatrix}}_{\triangleq B_i} u_i[t]. \tag{4.17}$$

Even though the Control Law Design component of all three architectures is localized and distributed by the nature of SLS, we initially assume Mode Process ID and PLP are centralized. This is reasonable under Assumption 7, which imposes that communications among subsystems are much faster compared to the switching of the topologies. This is often the case in fault-tolerance for large-scale network applications such as the power grid and the internet, where faults are expected to occur rarely. Furthermore, in Section 4.4.2, we introduce the implementation of localized, distributed Mode Process ID and PLP. For simplification of terminology in this section only, we overload the terminology "PLP" to refer to both the controller with PLP (third row of Figure 4.2) and a component of the controller architecture in Figure 3.2 that leverages other algorithms, with the understanding that PLP truly refers to the latter.

4.4.1 Tradeoff Comparison Results

Each simulation is run by applying one of the three controller architectures to one of the two network systems. We run a total of 20 Monte-Carlo experiment trials and each trial is run for $T_{\text{sim}} = 400$ timesteps with $\Delta T = 10$. The PLP architecture also uses a future horizon of $L = 3$. A sample trajectory of the states and control versus time for all three architectures is shown in Figure 4.5 for the hexagon system; we reduce the time horizon to 80 timesteps for this figure only so that there is better clarity in distinguishing the lines. Because the objective is the reject external disturbances, the state values waver around the zero line. Moreover, under Topology-Robust, the state has the smallest oscillations around zero (green), followed by PLP (red), and finally the baseline (blue). A sample evolution of the consistent set narrowing approach applied for Mode Process ID is also shown in Figure 4.6 for the baseline and PLP architectures; again, we plot for a shorter horizon of time (120 timesteps) for easier visibility. The PLP architecture manages to successfully narrow the consistent set down to a singleton within the ΔT time interval more often than the baseline, and consequently also

manages to track the true mode more precisely.

The comparisons among the different scenarios are performed by evaluating one of the following four performance metrics. First, to measure the control effort, an LQR-like cost (4.18a) is averaged over the entire simulation duration T_{sim}. Second, to measure the disturbance-rejection performance, we consider the the time-average error norm (4.18b). Third, for the baseline controller architectures, we measure the proportion (4.18c) of the simulation duration in which the matching control law is used to control the current topology. Here, if the true mode is given by φ_n at time t, we say that the *matching control law* $\{\mathbf{\Phi}_m^{(i,t)} : i \in \mathcal{V}\}$ is used if $m \triangleq \hat{\varphi}_n^{(t)} = \varphi_n$. Fourth, the total runtime is recorded.

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbf{x}[t]^\top I_{n_x} \mathbf{x}[t] + \mathbf{u}[t]^\top I_{n_u} \mathbf{u}[t], \tag{4.18a}$$

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \|\mathbf{x}[t]\|_2 , \tag{4.18b}$$

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbb{1}\{\hat{\varphi}_n^{(t)} = \varphi_n\}, \tag{4.18c}$$

where I_{n_x}, I_{n_u} are identity matrices of the appropriate dimensions.

The metrics (4.18) are further averaged over 20 Monte-Carlo simulations with varying initial condition $\mathbf{x}_0$, noise process $\mathbf{w}[t]$, and true realization $\{\varphi_n\}$ of the mode process $\{\xi_n\}$. The results are tabulated in Table 4.1 with the three architecture names abbreviated: "Base" as the baseline, and "TR" as Topology-Robust. The proportion of time the matching control law is irrelevant for Topology-Robust because it computes a single law to be used for multiple topologies, hence the "–" entries. We also plot a sample evolution of $\left\| P - \hat{P}^{(t)} \right\|$ for one Monte-Carlo trial in Figure 4.7, where the norm taken is the Frobenius norm. Because $\hat{P}^{(0)}$ begins with uniform probabilities in the nonzero positions, there are some variations in the norm difference, but overall, the curve decreases with time, indicating convergence to within a small error ball of the true TPM. This also allows for the pattern-occurrence quantities to be solved more accurately, which improves the prediction performance of PLP. As Table 4.1 shows, this also enables better controller performance (LQR Cost and Error Norm) of PLP over the other two architectures.

The values in both sub-rows of the "LQR Cost" row in Table 4.1 suggest that the time-average LQR costs of all three controller architectures increase as the scale of the system gets larger.

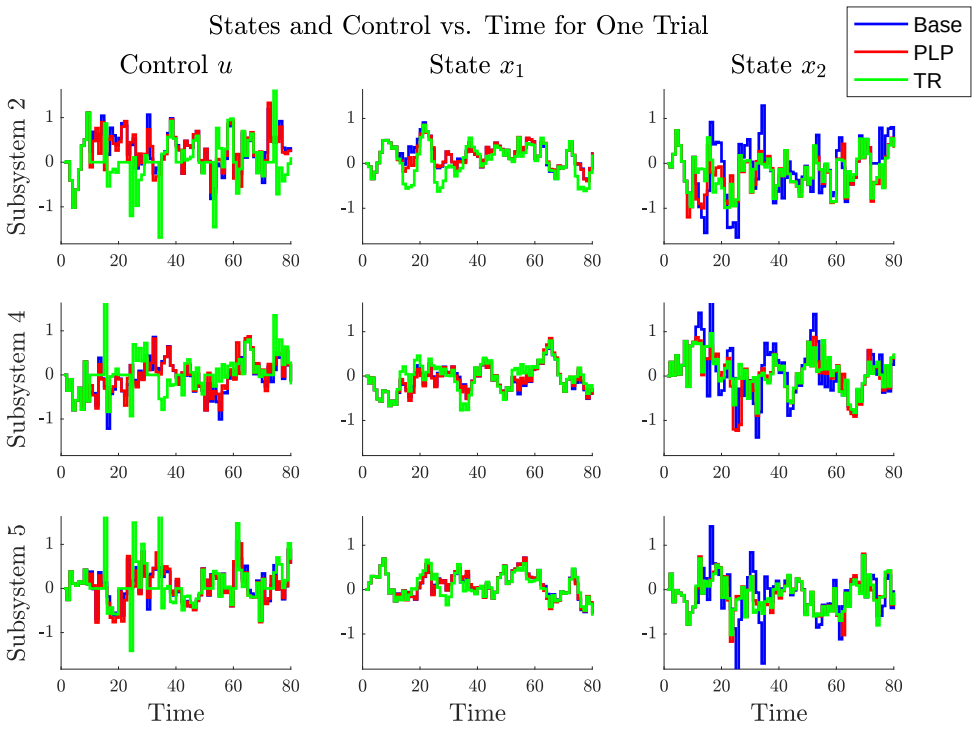

Figure 4.5: States and control versus time for one Monte-Carlo trial in the hexagon system. We abbreviate the baseline controller as "Base", and Topology-Robust as "TR."

This is expected because the same values of horizon H and number of hops h (defined in Section 4.3.2) were chosen for the SLS implementation of both systems. In practice, H and h must be adjusted as the scale of the system changes, but for fairer comparison we use the same values for both the hexagon and grid systems. Furthermore, assuming a small margin of error, Topology-Robust should theoretically stabilize the system better than the baseline at the expense of increased control effort because Topology-Robust uses a single common law is for multiple different modes. This can be validated empirically by the entries in the "LQR Cost" and "Error Norm" rows, and is also supported by Figure 4.5, where the state's oscillations around the zero line are the largest in magnitude with the baseline and the least with Topology-Robust.

More interestingly, the PLP architecture manages to balance the performance metrics better compared to the the other architectures: LQR cost similar to the baseline architecture, error norm similar to the Topology-Robust architecture, and runtime faster than either the baseline or the topology-robust extension. The improved runtime comes from the PLP component's

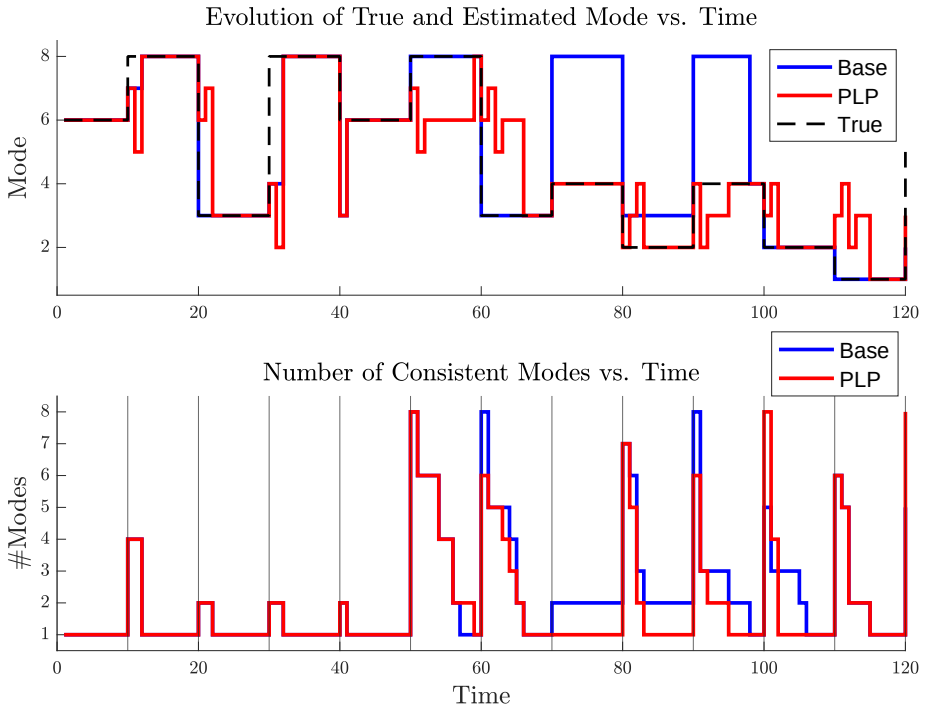

Figure 4.6: Modes versus time for one Monte-Carlo trial of the hexagon system. In the bottom subfigure, thin black vertical lines indicate intervals of length ΔT.

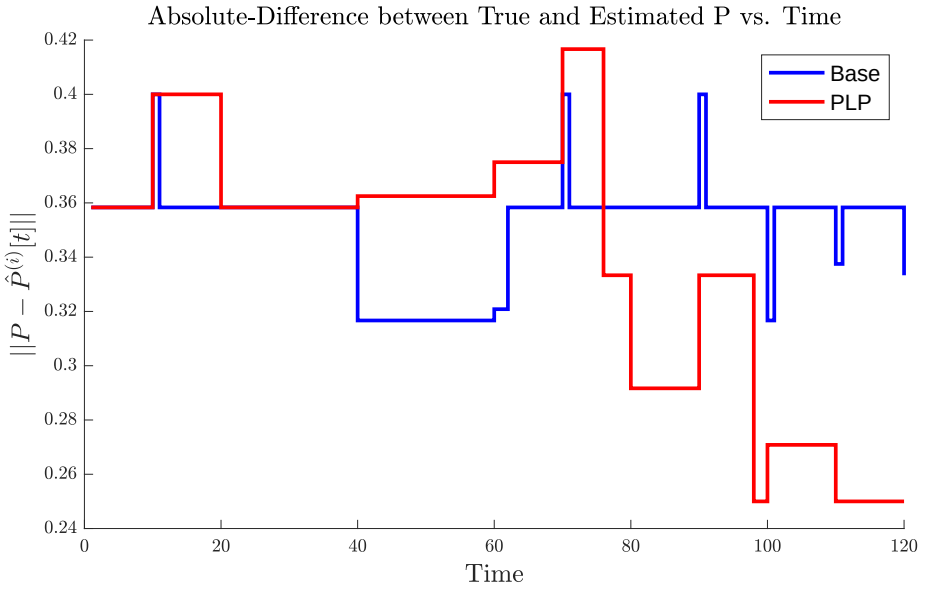

Figure 4.7: Frobenius norm of the difference between the true TPM P and estimated TPM $\hat{P}^{(t)}$ versus time for one Monte-Carlo trial of the hexagon system.

Metric / Controller	Base	TR	PLP
LQR Cost	36.4537	42.5596	34.7242
	445.8137	472.1195	442.1264
Error Norm	2.2146	1.5546	1.5236
	6.1294	5.9453	5.8244
Prop. Match	0.4304	–	0.615
	0.1533	–	0.16
Runtime	11.8314	67.2254	2.2689
	101.3741	X	38.5824

Table 4.1: The average performance metrics [row] over 20 Monte-Carlo simulations of $T_{\text{sim}} = 400$ timesteps, for each pair of controller architecture [column]. In each cell, the top value is recorded for the hexagon system and the bottom is for the grid system. For the sake of space, we abbreviate "Base" as the baseline controller, and "TR" as Topology-Robust.

ability to refrain from recomputing parts of the original SLS optimization by preserving the control inputs of previously-observed topologies and state/control trajectories (see Proposition 2). Moreover, the ability of PLP to predict the expected occurrence times of future mode patterns allows for the scheduling of SLS controllers in advance (see Proposition 1); as seen in Figure 4.2, this improves the error norm when Pattern-Learning manages to predict the future mode correctly. The "Prop Match" row of Table 4.1 shows that this is indeed the case: the PLP architecture consistently uses the matching control law more often than the baseline regardless of network system. This is expected since PLP can be viewed as an additional mode estimation algorithm, and so the estimate $\hat{\varphi}_n^{(t)}$ is on average better with PLP than without. In general, this suggests that appending PLP to a baseline controller that is neither predictive nor designed to be robust to time-varying topologies could be used as an alternative to Topology-Robust, especially in complex systems where simultaneous stabilization isn't possible or takes too much computation time, memory, and control effort.

We remark that the difference in the construction of the pattern collection $\Psi[t]$ in the hexagon system versus the grid system also has a role in the relationship among the performance metrics, especially in how often the matching control law is used and the error-norm performance. Recall that for the hexagon system, $\Psi[t]$ is created by accumulating every feasible mode sequence of length L, which implies $\mathbb{E}[\hat{\tau}_n^{(t)}] = L$, whereas in the grid system, a random subset of feasible mode sequences is chosen per time t. In the PLP column of the "Prop. Match" row, we see the matching control law is used less often in the grid system than the hexagon

system, which is expected since $\mathbb{E}[\hat{\tau}_n^{(t)}] \geq L$ for the grid system and predictions for a longer horizon of mode-indices become less accurate. Thus, increasing the number of patterns in the pattern collection decreases the expected minimum occurrence time, which yields more accurate estimates of future modes. The Base and PLP columns in the "Error Norm" row suggest that better predictions enable better disturbance-rejection; this implies that PLP will more closely resemble the error norm of the baseline when less patterns are included in $\Psi[t]$.

4.4.2 Localized Pattern-Learning and Prediction

Table 4.1 demonstrates that performance deteriorates with larger scale, and this can be attributed to the fact that both Pattern-Learning and Mode Process ID are implemented in a centralized fashion, which conflicts with the localized, distributed nature of SLS. We now briefly discuss an extension of PLP to a localized, distributed implementation of PLP. Since the previous section already compared the performance of PLP to those of the controllers without PLP, we focus our discussion here on how the localized implementation of PLP compares to the centralized version.

Let current time be $t \in \mathbb{N}$ and $n \triangleq N[t]$. Based on information from its own local subsystem (4.1), each node $i \in \mathcal{V}$ stores and updates three objects: 1) its own estimates of the current mode $\hat{\varphi}_n^{(i,t)}$ and TPM $\hat{P}^{(i,t)}$ (computed via Section 3.3.1), 2) its own estimates of the pattern-occurrence quantities $\mathbb{E}[\hat{\tau}^{(i,t)}]$, $\{\hat{q}_k^{(i,t)}\}_{k=1}^K$ (computed from Section 3.1.2), and 3) its own pattern collection $\Psi^{(i)}[t]$ and pattern-to-control law table $\mathcal{U}^{(i)}$ (see Section 3.3.3). Each node $i \in \mathcal{V}$ employs the consistent set narrowing approach of (4.2) to update its own set $\mathcal{C}^{(i)}[t]$ of consistent topologies over time t. Each subsystem $i \in \mathcal{V}$ then extracts $\hat{\varphi}_n^{(i,t)}$, $\mathcal{C}^{(i)}[t]$, and estimates $\hat{P}^{(i,t)}$ by empirically counting the proportion of transitions across the entire estimated past history $\hat{\varphi}_{0:n}^{(i,t)}$. For the TPMs, we also implement consensus averaging of the estimates to neighboring subsystems that are one link away. Essentially, the key distinction is that we add an additional enumeration $i \in \mathcal{V}$ to the usual sets, tables, and estimated quantities from Section 3.3.1 and Problem 1 in order to emphasize that each subsystem maintains local estimates of everything.

In Figure 4.8, we plot the estimated pattern-occurrence quantities for this localized extension of PLP applied to the hexagon system. To demonstrate the evolution of the pattern-occurrence quantities over time, each subsystem i's pattern collection $\Psi^{(i)}[t]$ is chosen to contain more than half of the full combinatorial set of feasible length-L mode sequences

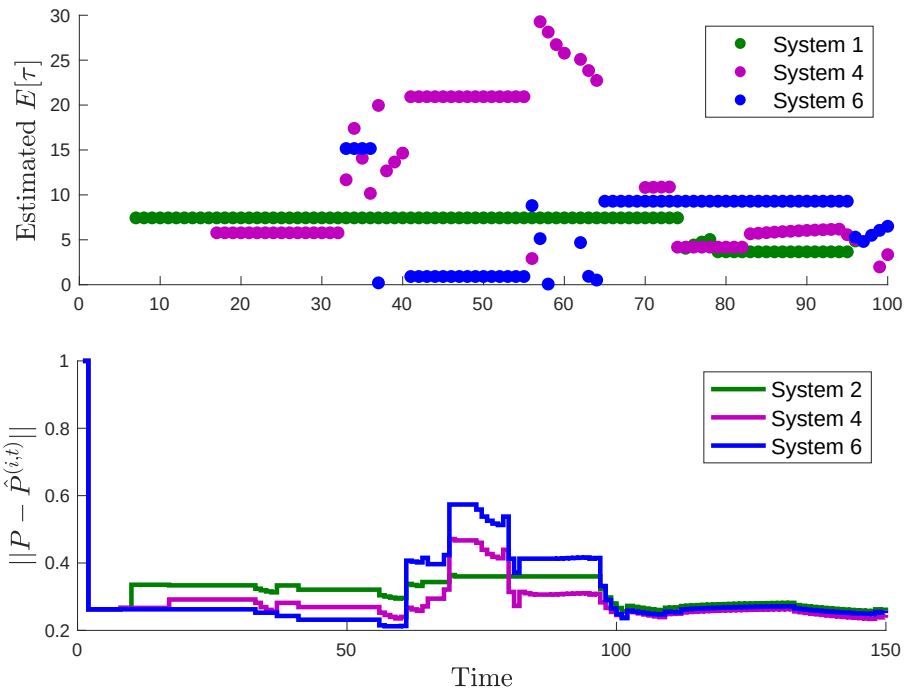

Figure 4.8: [Top] The evolution of $\mathbb{E}[\hat{\tau}^{(i,t)}]$ over time for subsystems $i \in \{1, 4, 6\}$. [Bottom] The evolution over time of the Frobenuis norm of the difference between P and $\hat{P}^{(i,t)}$ for subsystems $i \in \{2, 4, 6\}$.

initially considered in Section 4.4.1, such that the true value of $\mathbb{E}[\hat{\tau}^{(i,t)}]$ is 5.83328 via Theorem 10. The evolution of the estimated minimum occurrence time $\mathbb{E}[\hat{\tau}^{(i,t)}]$ over t is shown at the top, while the Frobenius norm difference $\|P - \hat{P}^{(i,t)}\|$ of the TPM estimate is shown at the bottom. We use varying groups of subsystems for these figures in order to demonstrate the locality property.

Note in Figure 4.8, that as time increases, the estimates $\mathbb{E}[\hat{\tau}^{(i,t)}]$ of tend to converge towards the true value 5.83328 as more of the TPM gets learned. The piecewise nature of the $\mathbb{E}[\hat{\tau}^{(i,t)}]$ over t in Figure 4.8 arises because the pattern collection $\Psi^{(i)}[t]$ may change over time, which in turn changes each subsystem's estimate of the expected minimum occurrence time. At the bottom of Figure 4.8, the matrix norm difference between the true and estimated TPMs for each of the three subsystems decreases over time, which is expected as each subsystem gathers more data to learn the true transition probabilities of P. Compared to the centralized TPM estimate evolution over time (Figure 4.7) there is more rapid variation in each subsystem's estimate in the bottom figure of Figure 4.8; this could be attributed to the consensus averaging among the subsystems. Viewing topologies at a local level can make the modes

look similar to one another, and so a localized implementation of consistent set narrowing may perform worse than the centralized implementation. This is a well-known tradeoff between centralized and distributed control: for more efficient computation, we are trading performance optimality. This is a well-known tradeoff between centralized and distributed/decentralized controller architectures: for more efficient computation, we are trading off optimality of performance. Still, even a localized implementation of PLP further supports the insights obtained from centralized PLP in Section 4.4.1: the time delays in identifying the current topology via Mode Process ID and time delays in the convergence of the estimated TPM to its true value are key factors in determining the extent of performance improvement that PLP brings.

4.5 Concluding Remarks

In this chapter, we implemented our PLP controller architecture from the previous Chapter 3 on the control of a network with dynamic topology by integrating the pattern-occurrence quantities into MPC (Section 4.3.1) and using variations of SLS (Section 4.3.2) for the Control Law Design component. We provided an empirical comparison study of its performance against a baseline controller and a topology-robust extension of the baseline. Because PLP can be viewed as an additional mode estimation algorithm, it enabled the estimated mode to match the true mode more often, although this was mainly possible for an optimal choice of pattern collection. Compared to the baseline, PLP was able to achieve better disturbance-rejection at significantly reduced computation time, redundancy, and control cost, which suggested its potential to be used in place of a robust controller for more complex applications where designing for robustness is expensive. Overall, the merit of our work in this application contributes to the concept of learning patterns for efficient control. Namely, less redundant control design for stochastic systems with uncertain dynamics can be performed by learning patterns in the system's behavior, which eliminates computation time and redundancy by storing past patterns into memory and predicting the future occurrence of patterns.

VEHICLE TRAFFIC CONGESTION CONTROL IN SIGNALIZED INTERSECTION NETWORKS

In this chapter, we describe the second of three applications that demonstrate the effectiveness of learning patterns for efficient control: the congestion control of vehicles in urban traffic networks of signalized intersections. One of the most common methods of modeling vehicle traffic flow is via queueing theory [106, 95, 108]. While queuing-based results are useful for benchmarking performance, they often rely on assumptions that are not reflective of real-world traffic (e.g., Poisson arrivals). Another class of models encompass discrete-time ODE dynamics [36], which allow for the explicit formulaic construction of control laws, but sometimes rely on the knowledge of parameters (e.g., turning proportion) whose values may be difficult to obtain in practice. Recently, neural networks are gaining traction as suitable methods for vehicle congestion control due to their ability to accommodate realistic traffic characteristics and complex network topologies. For example, [165] developed a framework for automated incident detection based on Bayesian networks, with an emphasis on being able to flexibly incorporate domain-specific knowledge into an otherwise all-data-driven approach. More recently, works such as [163] and [93] have considered variations on graph neural network (GNN) architectures to predict the spatiotemporal behavior of traffic spread across complex networks.

In the context of traffic networks, patterns can be defined according to the temporal repetition and structural symmetry that arises naturally in a variety of ways. For example, many road networks are typically arranged like a rectangular grid, and in America, T-junctions and X-junctions (4-way intersections) are highly common. These kinds of repeated topological structures fundamentally impact the travel behavior of vehicles and consequently, congestion level [159]. Repetition can be observed in the traffic density over time not only due to the grid structure of the network, but also due to human routine: rush hours during the weekdays are a notable example of this. Even for special events that do not occur regularly (e.g., traffic jams near the venue of a music concert), a certain level of congestion can be predicted if this special event was planned beforehand [88]. Many of the neural network architectures described above are designed to account for general topologies, and may be

less efficient when considering environments where structural symmetry could be leveraged. To account for environment repetition, methods based on explicit rule-based construction have also emerged [100, 47]. For vehicle routing, reinforcement learning methods are especially suitable [129], and for repetitive environments like urban grid intersection networks, experience replay (e.g., [56]) approaches can be used. A related approach is called episodic control [91, 20, 124], which incorporates episodic memory [21] into traditional learning techniques with the goal of speeding up training by recalling specific instances of highly rewarding experiences.

These natural spatial and temporal structures suggest that a congestion control mechanism designed around some suitable choice of "pattern" can improve the time and computation efficiency at which light signal sequences are designed. For this purpose, this chapter constructs a controller architecture similar to the one proposed in Chapter 3, with an extension of PLP called *pattern-learning with memory and prediction (PLMP)*. Compared to PLP, PLMP explicitly implements a memory component in the form of a table that maps patterns and light signal sequences to rewards; our architecture employs an approach which extends the state-of-the-art episodic control methods (e.g., [20]) by building *equivalence classes* to group patterns that can be controlled using the same sequence of light signals. The "patterns" at each intersection are the intersection's snapshots, e.g., traffic camera photos which display the distribution of vehicles present in each lane and direction. Moreover, in Control Law Design, prediction is implemented with a one-timestep lookahead that augments, to the original pattern, the distribution of vehicles in the adjacent links of the intersection and schedules future light signal sequences in advance. We apply our model to two synthetic datasets, one synthesized from scratch and one synthesized from real-world data, and compare two periodic baseline light signals to variations of our PLMP controller, including a version without prediction called *pattern-learning with memory (PLM)*. We evaluate the performance of each controller on a variety of traffic scenarios according to three different congestion metrics: 1) average waiting time per vehicle, 2) average time deviation away from the optimal travel duration, and 3) the number of vehicles that have not yet reached the end of their routes. We find that, on average, PLM outperforms the periodic baselines while PLMP outperforms PLM with mild variation among the different implementations.

Chapter Organization

First, Section 5.1 introduces the relevant notation and assumptions, then sets up the scenario of vehicle traffic flow as a Markov decision process (MDP). Next, Section 5.2 describes the

concrete implementation of the PLMP component. Lastly, numerical simulations comparing variations of our architecture to several baseline light signals are discussed in Section 5.3.

5.1 System Setup and MDP Formulation

5.1.1 The Grid Network of Signalized Intersections

A $H \times L$ rectangular grid network of four-way intersections is represented by a graph $\mathcal{G} = (\mathcal{I}, \mathcal{N}, \mathcal{E})$, with set of intersections $\mathcal{I}$, nodes $\mathcal{N}$, and directed edges $\mathcal{E}$ that connect between two nodes. Each intersection is denoted with a tuple $I := (h, i) \in \mathcal{I}$ marking its location in the grid, $h \in \{0, \cdots, H-1\}$ and $i \in \{0, \cdots, L-1\}$; there are a total of HL intersections in the network and they are assumed to be spaced equally apart. Each node is represented as a tuple $(I, \mathtt{D}, \chi, f) \in \mathcal{N}$, where I is the intersection ID, $\mathtt{D}$ is one of the four directions $\{\mathtt{E}, \mathtt{N}, \mathtt{W}, \mathtt{S}\}$, $\chi \in \{1, 0\}$ indicates whether vehicles are incoming (1) or outgoing (0) at the node. The variable $f \in \{0, 1\}$ indicates whether the node is located at the fringes of the network or not; we partition the set of nodes $\mathcal{N}$ into the set $\mathcal{N}_F$ of *fringe nodes* and the set $\mathcal{N}_I := \mathcal{N}/\mathcal{N}_F$ of *intermediate nodes*. Each intersection I is controlled by a traffic light signal; let $m \in \mathcal{M}$ be the mode of the traffic light and $\mathcal{M}$ be the set of possible modes. We assume there are $|\mathcal{M}| = 8$ possible modes each signal can take: 1) `E-W` forward green, 2) `E-W` left-turn green, 3) `N-S` forward green, 4) `N-S` left-turn green, 5) `E` forward and left green, 6) `N` forward and left green, 7) `W` forward and left green, 8) `S` forward and left green. Right-turns are permitted whenever.

5.1.2 Vehicle Arrival Processes

Let $\mathcal{V}_A[t]$ represent the time-varying set of vehicle arrivals from the fringes of the network, i.e., $\mathcal{V}_A[t] = \varnothing$ if no vehicles entered the grid at time t and $\mathcal{V}_A[t] = \{v_1, \cdots, v_K\}$ if some number $K \in \mathbb{N}$ vehicles v_1 to v_K have entered at time t. Note that $\mathcal{V}[t] := \cup_{s=0}^{t} \mathcal{V}_A[t]$ is the total number of vehicles that are in the grid network by time t. Note that $\mathcal{V}_D[t]$ is an increasing set, and the hope is that $\mathcal{V}_D[T_{\text{sim}}] = \cup_{s=0}^{T_{\text{sim}}} \mathcal{V}_A[t]$ by the end of the simulation duration. Let $\mathcal{V}_D[t] \subseteq \mathcal{V}[t]$ be the set of departed vehicles (i.e., vehicles that have reached their destination) and let $\mathcal{V}_C[t] := \mathcal{V}[t] \backslash \mathcal{V}_D[t]$ be the set of circulating vehicles. For all sets of the form $\mathcal{V}_*[t]$, we use $V_*[t] = |\mathcal{V}_*[t]|$ to represent its cardinality. The evolution of $\mathcal{V}[t]$ depends on both the vehicle arrival process and the vehicle departure process; the vehicle arrival process is extracted from sensing data and the vehicle departure process depends on the light signals.

All vehicles are identical with some length and travel at a constant speed, which means they travel to and across each intersection at a constant amount of time. Let Δt_L be the time it

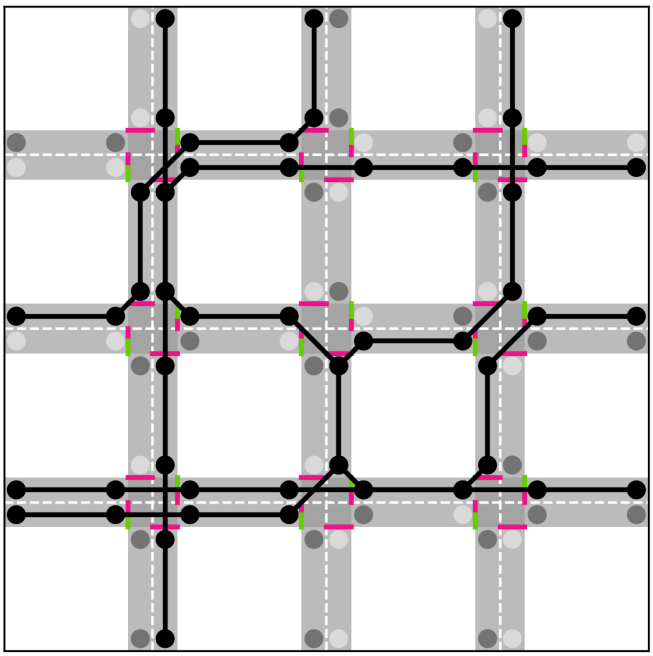

Figure 5.1: Grid network visualization of intersection graph $\mathcal{G}$ for $H = L = 3$. Nodes are distinguished by incoming ($\chi = 1$, light gray dots) or outgoing ($\chi = 0$, dark gray dots). Sample routes for five vehicles are also shown as an alternating sequence of nodes (black dots) and links (black lines). In this snapshot (e.g., traffic camera photos which display the distribution of each intersection in the grid), forward-going `E`/`W` traffic are allowed to pass through each intersection (red and green line segments); over time, and as the vehicles trace their respective routes, these traffic light colors would change.

takes a single vehicle to travel an uncongested link between intersections, and let Δt_I be the time it takes to cross an intersection. Define $\mathcal{R}(H, L)$ to be the entire combinatorial set of all routes from fringe to fringe of a grid network with dimensions $H \times L$. We assume each $v \in \mathcal{V}[t]$ is traversing the grid network according to a pre-determined route $\mathbf{r}_v \in \mathcal{R}(H, L)$ that starts at a node of entry $e_v \in \mathcal{N}_F$ and ends at a node of departure $d_v \in \mathcal{N}_F$. We represent $\mathbf{r}_v$ as an alternating sequence of nodes and links $\mathbf{r}_v = [e_v, \ell_v^{(e)}, n_1, \ell_1, \cdots, n_{k-1}, \ell_{k-1}, n_k, \ell_v^{(d)}, d_v]$, where $k \in \mathbb{N}$ is the route length, $n_i \in \mathcal{N}_I$ for $i \in \{1, \cdots, k\}$, and $\ell_v^{(e)}, \ell_v^{(d)}, \ell_i \in \mathcal{E}$ for $i \in \{1, \cdots, k-1\}$. We distinguish $\ell_v^{(e)}$ and $\ell_v^{(d)}$ from the other links ℓ_i as the fringe links of the route, i.e., links that connect to or from a fringe node $(I, \mathtt{D}, \chi, 1)$. A sample visualization of $\mathcal{G}$ with vehicle routes and light signals is in Figure 5.1.

Definition 34 (Vehicle Quantities)**.** Let $T_{\text{sim}} \in \mathbb{N}$ be the time duration of the experiment. Each vehicle $v \in \mathcal{V}[T_{\text{sim}}]$ locally keeps track of two congestion quantities. First, $W_v[t] \in \mathbb{N}$ is

its cumulative waiting time by time $t \in [0, T_{\text{sim}}]$, which increments by 1 for each timestep it waits at an intersection on a red light. Second, if $v \in \mathcal{V}_D[T_{\text{sim}}]$, $D_v \in \mathbb{N}$ is the total time it took to travel its entire route.

5.1.3 The Vehicle MDP (VMDP) Formulation

States: The state space $\mathcal{S} := \mathcal{S}_N \times \mathcal{S}_L$ is composed of two distinct parts. First, $\mathcal{S}_N \subseteq (\mathbb{Z}^{\geq 0})^{12HL}$ denotes the number of vehicles at each incoming intermediate node $\{(I, \texttt{D}, 1, 0) : \texttt{D} \in \{\texttt{E}, \texttt{N}, \texttt{W}, \texttt{S}\}\}$, partitioned by direction and turn (right, left, or forward); the elements of each $\mathbf{s}_{t,N} \in \mathcal{S}_N$ are ordered $[\texttt{E}\text{-rt}, \texttt{E}\text{-lft}, \texttt{E}\text{-fwd}, \texttt{N}\text{-rt}, \cdots, \texttt{W}\text{-rt}, \cdots, \texttt{S}\text{-rt}, \texttt{S}\text{-lft}, \texttt{S}\text{-fwd}]$ where rt, lft, and fwd are shorthand for right, left, and forward, respectively. Second, $\mathcal{S}_L \subseteq (\mathbb{Z}^{\geq 0})^{3(4HL-2H-2L)}$ represents the number of vehicles that are present in each link, partitioned again by turn, and each $\mathbf{s}_{t,L} \in \mathcal{S}_L$ is ordered in the same way as $\mathbf{s}_{t,N}$. The full state vector is concatenated as $\mathbf{s}_t = [{\mathbf{s}_{t,N}}^\top, {\mathbf{s}_{t,L}}^\top]^\top \in \mathcal{S}$.

Actions: The action space $\mathcal{A} := \mathcal{M}^{HL}$ describes the mode of each light signal at each intersection.

Transition Function: The transition function $\mathscr{T}(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$ for two states $\mathbf{s}_t, \mathbf{s}_{t+1} \in \mathcal{S}$ and action $\mathbf{a}_t \in \mathcal{A}$ is defined by the constraints of vehicle movement along the grid (i.e., to get from intersection $(0,0) \to (1,1)$, take either $(0,0) \to (0,1) \to (1,1)$ or $(0,0) \to (1,0) \to (1,1)$). We assume the time spent in each link is directly proportional to the level of congestion: if a vehicle enters a link with $X \in (\mathbb{Z}^{\geq 0})$ vehicles inside, it takes $(X+1)\Delta t_L$ timesteps to travel it if the link is between two intersections and $(X+1)\Delta t_I$ timesteps if the link is across an intersection. We define $v^* \in \mathbb{N}$ to be the maximum number of vehicles per turn that can cross an intersection in one timestep.

Rewards: The reward function $\mathscr{R}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) := \mathbf{1}^\top(\mathbf{s}_{t+1,N} - \mathbf{s}_{t,N})$ is the *rate of intersection clearance*, which computes the total number of vehicles that are removed from each intersection through an action $\mathbf{a}_t$ that drives $\mathbf{s}_t$ to $\mathbf{s}_{t+1}$. Here, $\mathbf{1} \in (\mathbb{Z}^{\geq 0})^{12HL}$ is the vector of all ones.

Definition 35 (Congestion Metrics)**.** Let $T_{\text{sim}} \in \mathbb{N}$ be the time duration of the experiment, and define $D_v^* \leq D_v$ to be the optimal travel time of each vehicle $v \in \mathcal{V}_D[T_{\text{sim}}]$ (i.e., the time taken to reach its destination assuming an empty network and all-green light signals). With the vehicle quantities described in Definition 34, we use the following metrics to evalu-

ate the performance of our controller. First, define the average cumulative waiting time to be $W := (1/V_D[T_{\text{sim}}]) \sum_{v \in \mathcal{V}_D[T_{\text{sim}}]} W_v[T_{\text{sim}}]$. Second, define the average travel deviation to be $D := (1/V_D[T_{\text{sim}}]) \sum_{v \in \mathcal{V}_D[T_{\text{sim}}]} (D_v - D_v^*)$. Third, we keep track of $V_C[t]$, the number of vehicles that did not reach their destinations by $t \in [0, T_{\text{sim}}]$.

5.2 Pattern-Learning with Memory and Prediction

We now describe the controller architecture based on *pattern-learning with memory and prediction (PLMP)*, an extension of PLP (see Chapter 3) for the VMDP set up in Section 5.1.3. With $I := (h, i)$, let the set $\Psi_I[t] = \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_{K[t]}\}$ be the collection of patterns for intersection I at time t, where $K[t] \in \mathbb{N}$ is the number of patterns currently recorded and each $\boldsymbol{\psi}_k$ represents a pattern. Note that for any $0 < s < t$, $\Psi_I[s] \subseteq \Psi_I[t]$. In our VMDP, the "patterns" of intersection I correspond to the distribution of vehicles in its local snapshot; for concreteness, we choose $\boldsymbol{\psi}_k \in (\mathbb{Z}^{\geq 0})^8$ to be a projection of a state $\mathbf{s}_{t,N} \in \mathcal{S}_N$ down to left and forward turns per direction; since vehicles can turn right whenever, they are not considered in the pattern.

5.2.1 Learning from Spatial Patterns

The VMDP implements the memory part of the PLMP controller architecture by storing any patterns that have frequently occurred in the past. This is motivated by the concept that snapshots often repeat over time, e.g., a snapshot containing X number of vehicles in the North-South lanes and no vehicles in the East-West lanes is likely to occur again later in time. In this implementation, we are inspired by *episodic memory*, which can be added to reinforcement learning methods to recall specific experiences and their rewards; some common implementations of episodic memory for control are [91], [20], and [124]. In our VMDP, episodic memory is implemented for each intersection I with a *memory table* $\mathcal{Q}_I : \mathbb{Z}^{\geq 0} \times \texttt{Eq}(\Psi_I[t]) \times \mathcal{M} \to \mathbb{R}$, which maps patterns and light signal modes to best rewards. Compared to previous episodic memory approaches, each memory table in our VMDP also uses *equivalence classes* to limit its size: the original pattern collection $\Psi_I[t]$ is divided into multiple classes such that all patterns in a class are assigned the same optimal traffic light. We define $\texttt{Eq}(\Psi_I[t]) \subseteq \Psi_I[t]$ to be the *unique keys* of the equivalence classes for $\Psi_I[t]$. Each entry $\mathcal{Q}_I(t, \boldsymbol{\psi}, m) = r$ means that as of time t, the best reward of r can be obtained by applying mode m to intersection I if the given pattern is $\boldsymbol{\psi}$.

For each intersection I, equivalence classes are constructed in the following way. For the first pattern $\boldsymbol{\psi}_1 \in \Psi_I[0]$, $\boldsymbol{\psi}_1$ is placed inside $\texttt{Eq}(\Psi_I[t])$ and its associated equivalence class is

constructed as $\texttt{Eq}(\boldsymbol{\psi}_1) = \texttt{Eq}_0(\boldsymbol{\psi}_1)$, where

$$\texttt{Eq}_0(\boldsymbol{\psi}_j) := \{v \cdot \boldsymbol{\psi}_j, v \in \{2, \cdots, v^*\}\} \cup \{[v_1 + \psi_{j,1}, \cdots, v_8 + \psi_{j,8}], [v_1, \cdots, v_8] \in \{0, \cdots, v^*\}^8 \backslash \mathbf{0}\}, \quad (5.1)$$

where $\cdot$ denotes multiplication by a scalar, v^* is from Section 5.1.3, and $\mathbf{0} \in \mathbb{R}^8$ is the all-zeros vector. This means $\texttt{Eq}_0(\boldsymbol{\psi}_j)$ contains the following two types of elements: 1) every elementwise multiple of $\boldsymbol{\psi}_j$ up to a factor of v^*, 2) every nonzero additive variation of the entries of $\boldsymbol{\psi}_j$ up to v^*.

For each time $t+1$ when a new pattern $\boldsymbol{\psi}_k \notin \texttt{Eq}(\Psi_I[t])$ is observed at intersection I, its equivalence class is constructed iteratively as:

$$\texttt{Eq}(\boldsymbol{\psi}_k) := \begin{cases} \varnothing & \text{if } \exists\, \boldsymbol{\psi}_j \in \texttt{Eq}(\Psi_I[t]) \text{ s.t. } \boldsymbol{\psi}_k \in \texttt{Eq}(\boldsymbol{\psi}_j) \\ f(\texttt{Eq}_0(\boldsymbol{\psi}_k), \{\texttt{Eq}(\boldsymbol{\psi}_j)\}_{j=1}^{K[t]}, \texttt{Eq}(\Psi_I[t])) & \text{else} \end{cases}, \quad (5.2)$$

where $\texttt{Eq}_0$ is defined in (5.1). The function f is designed to check if every $\boldsymbol{\psi} \in \texttt{Eq}_0(\boldsymbol{\psi}_k)$ is already in the pattern collection, whether as a unique key or an equivalence class member:

$$f(\texttt{Eq}_0(\boldsymbol{\psi}_k), \{\texttt{Eq}(\boldsymbol{\psi}_j)\}_{j=1}^{K[t]}, \texttt{Eq}(\Psi_I[t])) := \left\{\boldsymbol{\psi} \in \texttt{Eq}_0(\boldsymbol{\psi}_k) : \tilde{f}(\boldsymbol{\psi}, \{\texttt{Eq}(\boldsymbol{\psi}_j)\}_{j=1}^{K[t]}, \texttt{Eq}(\Psi_I[t])) = 1\right\},$$

$$\tilde{f}(\boldsymbol{\psi}, \{\texttt{Eq}(\boldsymbol{\psi}_j)\}_{j=1}^{K[t]}, \texttt{Eq}(\Psi_I[t])) = \begin{cases} 0 & \text{if } \exists\, \boldsymbol{\psi}_j \in \texttt{Eq}(\Psi_I[t]) \text{ s.t. } (\boldsymbol{\psi} = \boldsymbol{\psi}_j \vee \boldsymbol{\psi} \in \texttt{Eq}(\boldsymbol{\psi}_j)) \\ 1 & \text{else} \end{cases}. \quad (5.3)$$

This construction allows all elements of $\Psi_I[t]$ to be partitioned into its unique keys and disjoint equivalence classes for all time t, i.e., $\Psi_I[t] = \texttt{Eq}(\Psi_I[t]) \cup \texttt{Eq}(\boldsymbol{\psi}_1) \cup \cdots \cup \texttt{Eq}(\boldsymbol{\psi}_{K[t]})$. Looking up Q-values then amounts to looking through only $\texttt{Eq}(\Psi_I[t])$ instead of the entire collection $\Psi_I[t]$, which reduces memory compared to other episodic control approaches.

The update method of each intersection's memory table follows similarly to episodic control. At specific intersection I, suppose $\boldsymbol{\psi}$ is the current pattern snapshot observed at time t. If $\boldsymbol{\psi} \notin \Psi_I[t]$, the Q-value is approximated with $\hat{Q}_I$, which averages the Q-values of the k-nearest-neighbor (kNN) patterns in $\texttt{Eq}(\Psi_I[t])$:

$$\hat{Q}_I(t, \boldsymbol{\psi}, m) := \begin{cases} \frac{1}{k} \sum_{j=1}^{k} \mathcal{Q}_I(t, \hat{\boldsymbol{\psi}}_j, m) & \text{if } \boldsymbol{\psi} \notin \Psi_I[t] \\ \mathcal{Q}_I(t, \boldsymbol{\psi}, m) & \text{else} \end{cases}, \quad (5.4)$$

where $\{\hat{\boldsymbol{\psi}}_j\}_{j=1}^{k} \subseteq \texttt{Eq}(\Psi_I[t])$ are the k unique keys with the nearest distance to $\boldsymbol{\psi}$ at time t. Here, "nearest" is measured with ℓ_1-norm difference, modulus the structure of the equivalence

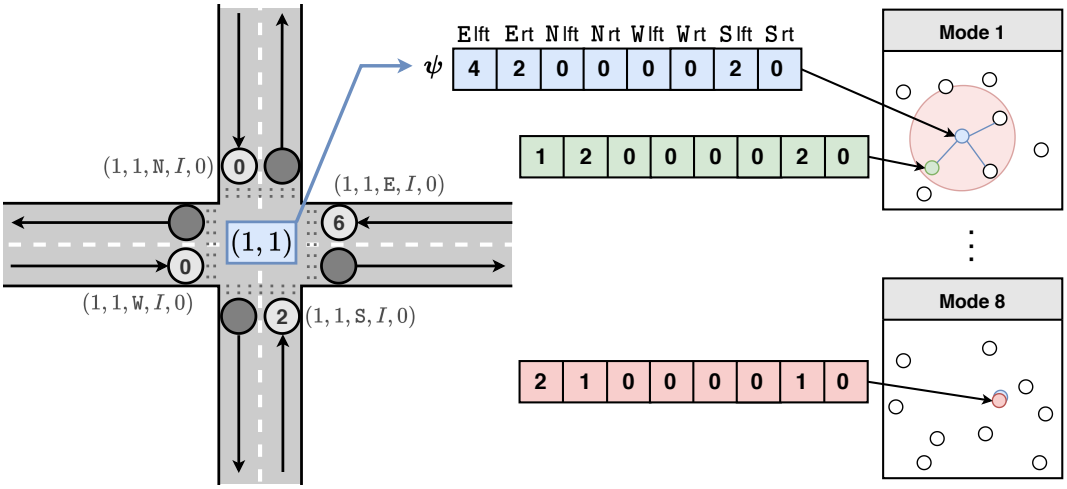

Figure 5.2: Example memory table $\mathcal{Q}_I$ for intersection $I := (1,1)$ with current pattern $\boldsymbol{\psi} = [4,2,0,0,0,0,2,0]$ (blue), $v^* = 2$, and $k = 3$ nearest neighbors. Entries in $\texttt{Eq}(\Psi_I[t])$ are marked with white circles. For mode 1, $\boldsymbol{\psi}$ does not exist in $\mathcal{Q}_I$, so the 3 nearest patterns (large red ball) are used during lookup; one example of a "near" pattern is in green, where the left-turn lane in the East direction has three less vehicles. For mode 8, an entry for $\boldsymbol{\psi}$ already exists because it is equivalent to the red pattern, which is $\boldsymbol{\psi}/2$.

classes: $d(\boldsymbol{\psi}_k, \boldsymbol{\psi}_j) := \|(\{\boldsymbol{\psi}_k\} \cup \texttt{Eq}(\boldsymbol{\psi}_k)) - (\{\boldsymbol{\psi}_j\} \cup \texttt{Eq}(\boldsymbol{\psi}_j))\|_1$ where we briefly abuse notation to denote $\|\mathcal{B}_1 - \mathcal{B}_2\|_1 := \min\{\|b_1 - b_2\|_1, b_1 \in \mathcal{B}_1, b_2 \in \mathcal{B}_2\}$. During training, the Q-values of the memory table are updated by comparing the existing value with the Bellman update. Denote $\overline{\boldsymbol{\psi}} \in \mathcal{S}_N$ to be the expansion of $\boldsymbol{\psi}$ where zeros are placed in the positions of right-turning vehicles. Suppose the pair $(\boldsymbol{\psi}, m)$ at time t transitions to the pattern $\boldsymbol{\psi}^*$ via transition function $\mathscr{T}_I(\overline{\boldsymbol{\psi}}^* | \overline{\boldsymbol{\psi}}, m)$ and yields reward $\mathscr{R}_I(\overline{\boldsymbol{\psi}}, m, \overline{\boldsymbol{\psi}}^*)$, where $\mathscr{T}_I$ and $\mathscr{R}_I$ are dimension-reduced versions of $\mathscr{T}$ and $\mathscr{R}$ (from Section 5.1.3) for individual intersections. Then define:

$$r^* := (1-\alpha)\hat{Q}_I(t, \boldsymbol{\psi}, m) + \alpha(\mathscr{R}_I(\overline{\boldsymbol{\psi}}, m, \overline{\boldsymbol{\psi}}^*) + \gamma \hat{Q}_I(t, \boldsymbol{\psi}^*, m^*)). \tag{5.5}$$

Here, $\hat{Q}_I$ is the estimated Q-value computed through (5.4), $\alpha \in [0,1]$ is the learning rate, and $\gamma \in [0,1]$ is the reward discount rate. Mode m^* is the optimal light signal mode from pattern $\boldsymbol{\psi}^*$ (and varies by algorithm, e.g., Q-learning, SARSA). The update for entry $(\boldsymbol{\psi}, m)$ is then performed as follows:

$$\mathcal{Q}_I(t+1, \boldsymbol{\psi}, m) \leftarrow \begin{cases} \max\{\mathcal{Q}_I(t, \boldsymbol{\psi}, m), r^*\} & \text{if } (t, \boldsymbol{\psi}, m) \in \mathcal{Q}_I \\ r^* & \text{else} \end{cases}. \tag{5.6}$$

The action $\mathbf{a}_t \in \mathcal{A}$ is then constructed by putting together all the optimal modes m^* of each intersection into a single vector. The PLMP algorithm with only memory implemented (with-

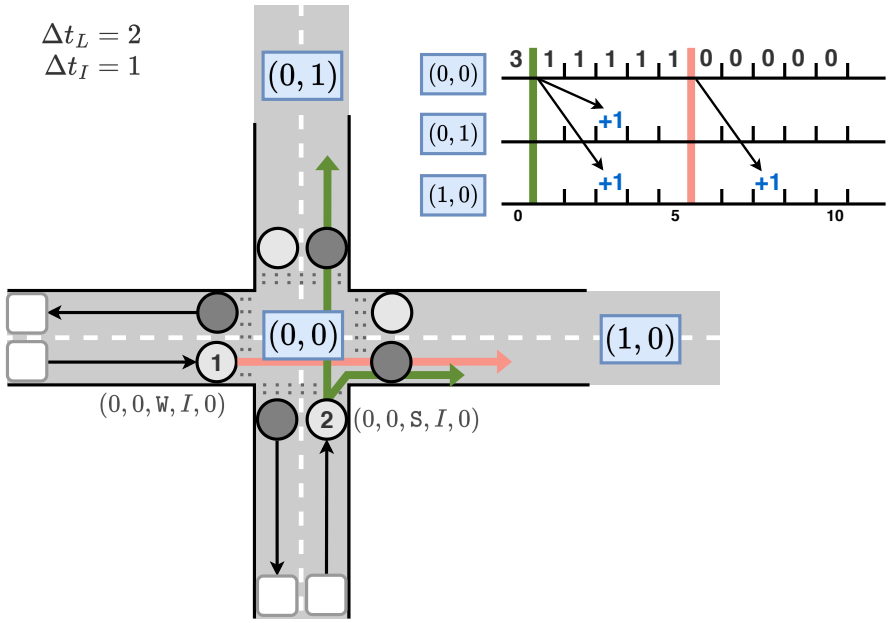

Figure 5.3: Sample prediction procedure for intersection $(0,0)$ and its neighbors $(0,1)$ and $(1,0)$. Here, $\Delta t_L = 2$ and $\Delta t_I = 1$. There are a total of three vehicles at $(0,0)$ at time 0: two vehicles (one right-turning, one forward-going) at direction `S` are given the green light to pass at time 0 while one vehicle (forward-going) at direction `W` is given the green light to pass at time 6. Here, there are no other vehicles in the system, so each vehicle takes $\Delta t_I + \Delta t_L = 3$ timesteps to reach their next intersection.

out prediction) will henceforth be called *pattern-learning with memory (PLM)*; note that it differs from episodic control by implementation of the equivalence classes. For concreteness and variety, we consider two different ways of choosing the optimal mode m given pattern $\boldsymbol{\psi}$. First, *greedy exploitation* uses transition function $\mathscr{T}_I$ to approximate the next $\boldsymbol{\psi}^*$ and chooses the mode m that maximizes the immediate reward $\mathscr{R}_I(\overline{\boldsymbol{\psi}}, m, \overline{\boldsymbol{\psi}}^*)$. Second, *episodic control (EC) exploitation* chooses the action m^* which maximizes (5.4). We also enable *exploration* with some probability $\epsilon \in [0,1)$, i.e., randomly choose $m \in \mathcal{M}$.

5.2.2 Learning from Temporal Patterns

The VMDP implements the prediction part of the PLMP controller architecture by approximating future occurrences of patterns so that future light signal sequences can be scheduled in advance. Because the objective is to demonstrate the advantage of enabling prediction, we use a simple one-timestep lookahead assuming that all predictions are accurate due to sensors being abundantly placed throughout the grid; we defer the treatment of noisy predictions to future work.

We employ an *augmented pattern* representation $\boldsymbol{\phi}_k = [\boldsymbol{\psi}_k^\top, \boldsymbol{\zeta}_k^\top]^\top \in (\mathbb{Z}^{\geq 0})^{16}$ associated with

each original pattern $\boldsymbol{\psi}_k \in \Psi_I[t]$. The eight additional entries $\boldsymbol{\zeta}_k \in (\mathbb{Z}^{\geq 0})^8$ contain the counts of incoming vehicles in its adjacent links, and can be viewed as a projection of state $\mathbf{s}_{t,L} \in \mathcal{S}_L$ down to left and forward turns per direction. Define $\mathcal{P} : (\mathbb{Z}^{\geq 0})^{16} \to (\mathbb{Z}^{\geq 0})^8$ to be a *projection mapping* such that $\mathcal{P}(\boldsymbol{\phi}_k)$ is equal to the pattern which will occur in the next timestep. Because a vehicle's transition time from a link to an incoming node depends on the number of other vehicles that are currently present on the link, we do not write the explicit form of $\mathcal{P}$; essentially, we achieve accurate predictions by enabling one-timestep lookahead using the augmented pattern. For example, when $\Delta t_L = 1$ and there are no other vehicles in the left-turn lane of the link to the East of intersection I, we get $\mathcal{P}([\mathbf{0}^\top, \mathbf{e}_1^\top]^\top) = \mathbf{e}_1^\top$, where $\mathbf{e}_1$ is the first standard basis vector of $(\mathbb{Z}^{\geq 0})^8$.

We conclude this section with a side-by-side comparison of the algorithm pseudocode for vehicle traffic congestion control with PLM and PLMP.

Algorithm 1 Congestion Control via PLM

Initialize VMDP.
Initialize pattern tables $\{\Psi_I[0]\}$.
Create next pattern $\boldsymbol{\psi}$.
Create next traffic light from $\boldsymbol{\psi}$.
for $t = 1 : T_{\text{sim}}$ **do**
    Propagate 1 step.
    Add any new vehicle arrivals.
    Update VMDP state.
    Update pattern tables $\{\Psi_I[t]\}$.
    Create next pattern $\boldsymbol{\psi}$.
    Create next traffic light from $\boldsymbol{\psi}$.
end for

Algorithm 2 Congestion Control via PLMP

Initialize VMDP.
Initialize pattern tables $\{\Psi_I[0]\}$.
Predict next pattern $\boldsymbol{\psi}^* = \mathcal{P}(\boldsymbol{\phi})$.
Create next traffic light from $\boldsymbol{\psi}^*$.
for $t = 1 : T_{\text{sim}}$ **do**
    Propagate 1 step.
    Add any new vehicle arrivals.
    Update VMDP state.
    Update pattern tables $\{\Psi_I[t]\}$.
    Predict next pattern $\boldsymbol{\psi}^* = \mathcal{P}(\boldsymbol{\phi})$.
    Create next traffic light from $\boldsymbol{\psi}^*$.
end for

5.3 Numerical Simulations

We demonstrate the performance of various implementations of Algorithms 1 and 2. We compare the two ways from Section 5.2.1 in which actions are chosen: exploration with probability ϵ together with greedy or EC exploitation. We distinguish the way in which each intersection updates its memory table by varying the learning rate α: an *episodic control (EC) update* computes the new potential Q-value as in (5.5) with $0 < \alpha < 1$ (specifically chosen $\alpha = 0.9$), while a *greedy update* uses $\alpha = 0$. We also consider a *periodic baseline* controller, where the light at each intersection cycles through the modes repeatedly with some cycle duration $C \in \mathbb{N}$.

Strategy	Description
Periodic16	Periodic with $C = 16$.
Periodic8	Periodic with $C = 8$.
EC(0)	PLM with EC exploitation, $\epsilon = 0$ exploration, EC update
PLM(0,1,1)	PLM with greedy exploitation $\epsilon = 0$ exploration, greedy update
PLM(5e-3,1,0)	PLM with greedy exploitation $\epsilon =$ 5e-3 exploration, EC update
PLMP(0,0,0)	PLMP with EC exploitation $\epsilon = 0$ exploration, EC update
PLMP(0,1,1)	PLMP with greedy exploitation $\epsilon = 0$ exploration, greedy update
PLMP(5e-3,1,0)	PLMP with greedy exploitation $\epsilon =$ 5e-3 exploration, EC update

Table 5.1: The eight different controller implementations compared in the experiment: two periodic baselines and six different versions of Algorithms 1 and 2. PLM$(0, 0, 0)$ is equivalent to episodic control (EC), but with equivalence classes implemented.

5.3.1 Dataset Preprocessing

We apply each variation of our proposed architecture to the following two datasets.

Pure Synthetic The number $V_A[t]$ of vehicles arriving into the network from the fringes as a function of time t is described as follows. Vehicles enter into the fringe intersections as platoons. Let T_n be the time of arrival for the nth platoon. Interarrival times $T_n - T_{n-1}$ are generated independently from a Geometric distribution with a time-varying parameter $p[t] \in (0, 1)$. At each arrival time T_n, the size of each platoon is a random nonzero integer generated between some minimum and maximum size. We choose experiment duration $T_{\text{sim}} = 200$, $p[t] \in [0.1, 0.25]$, and each platoon generated has some size between 30 to 50 vehicles.

Real-World Intersections We use real-world data of traffic flowing through several single intersections provided in [146]. For our grid network setting, we assign the behavior of one intersection to each of the four fringes of the grid: intersections on the East fringe of the grid behave according to the `kn-hz` intersection, the North fringe is according to the `qc-yn`

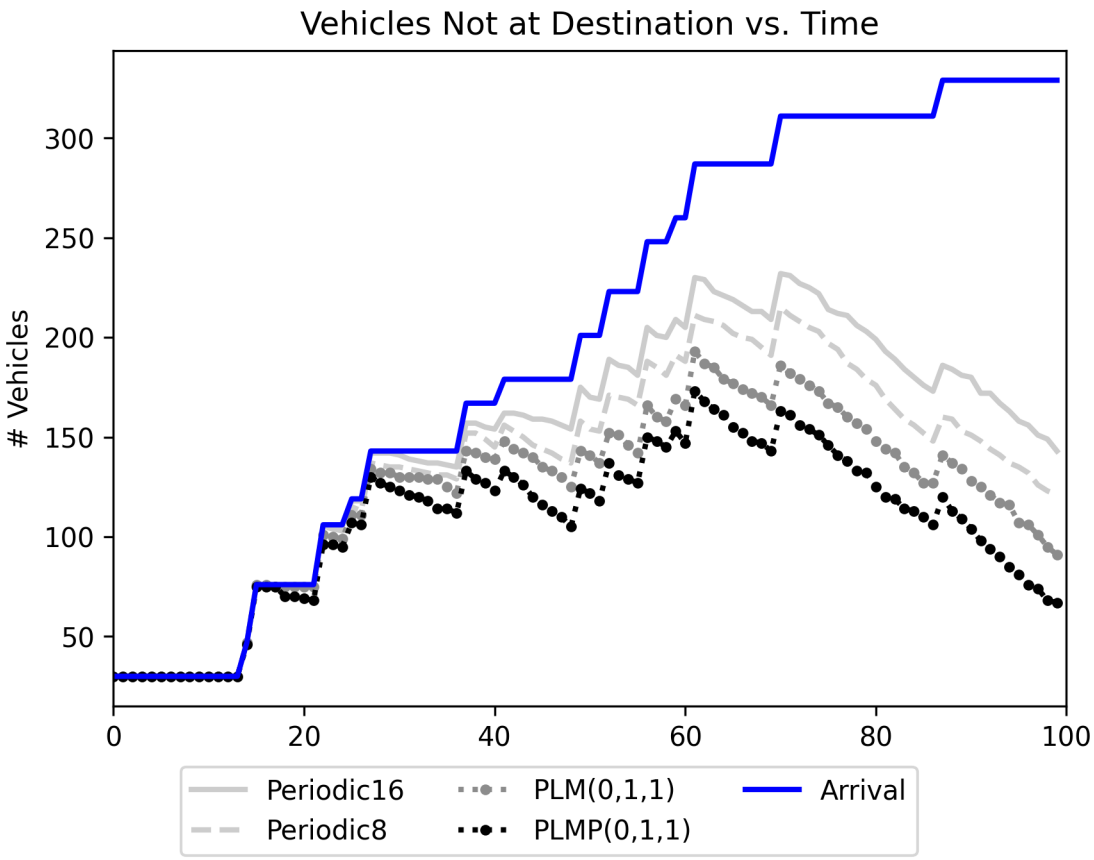

Figure 5.4: Congestion metric $V_C[t]$ from Definition 35 plotted over time until $t = 100$ for four strategies from Table 5.1, applied to the pure synthetic dataset. The cumulative number of vehicles in the system over time, $V[t]$, is shown in solid blue.

intersection, West to `sb-sx`, and South to `tms-xy`. Each intersection in the original dataset consists of two one-hour arrivals of vehicles; we add the two arrival processes together and discretize arrivals into 5-second bins, i.e., $T_{\text{sim}} = 3600/5 = 720$. The original vehicle routes are also modified to respect the constraints of being a fringe intersection, e.g., traffic emerging from the East side of `sb-sx` are rerouted to emerge from the West side (because `sb-sx` corresponds to fringe intersections of the form $(0, i)$ for $i \in \{0, \cdots, L-1\}$).

5.3.2 Results

We evaluate eight different controller implementations, using the congestion metrics from Definition 35, on a grid network with dimensions $H = L = 5$. Each implementation is described in Table 5.1. Our results for $v^* = 2$, $\Delta t_L = 2$, $\Delta t_I = 1$, $k = 3$ nearest neighbors, and $\gamma = 0.1$ averaged over 20 Monte-Carlo trials, are demonstrated in Table 5.2 for the synthetic dataset

Strategy	W	D
Periodic16	12.21145	44.37445
Periodic8	20.25	58.79762
EC(0)	1.2622	24.27622
PLM(0,1,1)	1.10247	23.75265
PLM(5e-3,1,0)	4.62031	30.0
PLMP(0,0,0)	0.41584	22.19802
PLMP(0,1,1)	0.38158	21.92434
PLMP(5e-3,1,0)	2.42907	28.37716

Strategy	W	D
Periodic16	12.13208	38.20283
Periodic8	6.87660	26.78723
EC(0)	0.89105	14.80156
PLM(0,1,1)	0.89412	14.93334
PLM(5e-3,1,0)	0.90551	14.79528
PLMP(0,0,0)	0.22509	9.98524
PLMP(0,1,1)	0.23443	9.17216
PLMP(5e-3,1,0)	0.25368	9.49632

Table 5.2: The average cumulative waiting time W and the average travel deviation D (from Definition 35) for each of the two datasets. [Left] Pure Synthetic. [Right] Real-World.

(left subtable) and the real-world dataset (right subtable). A sample plot of one trial of $V_C[t]$ (the number of circulating vehicles metric from Definition 35) for the synthetic dataset until $t = 100$ timesteps is shown in Figure 5.4; the figure for the real-world dataset yielded a similar trend, but with smaller values because the arrival process is much thinner than the synthetic dataset despite being over longer time interval.

The basic trend for both datasets is that PLMP does better on average than PLM and PLM does better on average than the periodic baseline. In Figure 5.4, all four controllers experience at least a 15-timestep delay after time 0 from which vehicles begin to reach their respective destinations; thus, each line follows the cumulative number of vehicles in the system (blue line) precisely until timestep 15. Afterwards, however, PLMP consistently begins to drop first, followed by PLM, then finally the periodic baseline, indicating that PLMP enables vehicles to reach their destination the fastest on average and periodic enables the slowest. This is also consistent with the magnitude of the measurements in Table 5.2. For the periodic baselines applied to the synthetic dataset (heavier traffic), the left subtable in Table 5.2 shows that a smaller period (i.e., faster light signal switching) can cause more congestion than relief. For PLM and PLMP, the average waiting time per vehicle (W) and average travel deviation (D) are mostly consistent to how they increase or decrease in value together. Adding exploration causes both PLM and PLMP to perform worse, which is expected because traffic in a structured, predictable setting like a rectangular grid leaves very little chance that choosing a random mode will perform better than pure exploitation.

5.4 Concluding Remarks

This chapter presented new controller architectures based on pattern-learning with memory and prediction (PLMP) and pattern-learning with memory (PLM) for vehicle traffic conges-

tion control over a metropolitan grid of signalized intersections. The architectures exploited the natural spatial symmetries and temporal repetition in the traffic network to perform control without redundant computation of light signal sequences. In particular, the memory component used an extension of episodic memory which builds equivalence classes to group together patterns that are controlled using the same light signals. In addition, accurate predictions are incorporated with a one-timestep lookahead that augments vehicle counts in the adjacent links of the intersection and schedules light signals in advance. We demonstrated the performance advantages of multiple implementations of PLM and PLMP with respect to three congestion metrics over two different traffic scenarios, and found that on average, PLM outperforms the periodic baselines while PLMP outperforms PLM with mild variation among the different implementations. Similar to the dynamic-topology network application of Chapter 4, this suggests that a control architecture built around learning patterns enables better performance; in this application, the specific performance metrics considered are the congestion metrics from Definition 35.

EPIDEMIC SPREAD MITIGATION IN POPULATION NETWORKS

Similar to the urban vehicle traffic networks from Chapter 5, population networks are another example of stochastic dynamics that involve spatial symmetries and temporal repetition that can be exploited for efficient control or estimation. Population networks tend to be partitioned according to distinct social ties and communities based on various factors such as geographical location, mutual interests, etc., which enable patterns to be identified in the relationships among the individuals [155]. Furthermore, the travel behavior of individuals repeat due to daily human routine [62], which can be used in making predictions about the spread of viruses.

Our third and final application used to demonstrate the impact of learning patterns comes from forecasting epidemic spread in population networks, especially the recent COVID-19 disease which was arguably ongoing as of December 2022. Mathematical models (e.g., [52, 54]) have been shown to be quite effective in informing the design of mitigation policies for various past outbreaks such as the influenza pandemics [35, 34] and the SARS epidemic of 2003 [64]. The seminal work [79] in 1927 introduced the original *compartmental model* for emulating epidemic/pandemic dynamics, which moves the population through the different phases of illness (called "compartments") undergone upon exposure to the virus. Since then, a variety of extensions have been made to this traditional model, including stochastic factors [141], nonlinear interactions with the infected [27], and temporary immunity [69]. An alternative compartmental model method uses *coupled hidden Markov models (CHMMs)*, e.g., [134, 142, 48], which allows for heterogeneity in the population network (e.g., different immune responses) by modeling at the individual level. An added benefit to the CHMM architecture is that there are standard algorithms which enable the inference of unknown parameters (e.g., death rate, recovery rate) from observational data such as contact-tracing information; multiple separate observation processes may be incorporated [158] and the parameters to be estimate may be time-varying [7].

Motivated by the above, this chapter proposes a novel multiscale model for the propagation of the SARS-CoV-2 virus throughout a population by exploiting community structures in the

interaction network and observing the effects of heterogeneous population characteristics. One primary novelty of our work is the explicit modeling of jump phenomena that are unique to epidemic spreads, which are *superspreader effects* [61] and the *emergence of variant viruses* [25], and posing the multiscale model as JSSs for each type of phenomena. Namely, gene sequencing analysis suggests that the possibility of infection from a variant of the virus is highly probable even if an individual has recovered from infection of a different strain [55]. Thus, infection to a variant virus can be represented as a jump to a set of dynamics with different parameter values, because different mutations have different parameter values [30]. This is similar to the switching among different topologies from Chapter 4, and requires more compartments to be included compared to traditional compartmental models. Moreover, superspreader effects can be captured by the shot noise phenomenon analyzed in Chapter 2.

Chapter Organization

In Section 6.1, we describe how to transform a given population into a graph network of communities and individuals for the purposes of our model, and we introduce the SEIRD phases of the COVID-19 disease. In Section 6.2, we outline the SEIRD compartmental ODE module, the larger scale of our multiscale model, and mathematically define the transition rates and the ODE dynamics. Section 6.2.2 presents an extension of the ODE compartmental model that explicitly considers superspreader effects; the model takes the form of a Poisson shot noise SDE similar to the one described in Chapter 2. Section 6.2.3 presents an extension of the ODE compartmental model that accounts for the emergence of variant viruses; the model switches among different parameter values in a way that is similar to the MJS which was used to model the dynamic topology network in Chapter 4. Next, in Section 6.3, we outline the coupled HMM module, the smaller scale of our multiscale model, and define the transition probabilities per individual. In particular, for estimating the parameters of the HMM module, we emphasize an extension of the Baum-Welch (expectation-maximization) algorithm in two key ways: incorporating multiple observation processes simultaneously, and accounting for time-varying parameters. In Section 6.4, we discuss how the two levels of available data, infected/death counts and individual-level contact-tracing, are used together to estimate the parameters of both parts of the multiscale model. In Section 6.5, we demonstrate the application of our multiscale model to multiple experiments, using two datasets that are constructed based on real-world data available online.

6.1 Preliminaries

The overall multiscale model proposed in this chapter is summarized in Figure 6.1. Before we discuss each part of the model in detail, we set up and introduce two concepts: first, the representation of the population as a network of communities and second, the SEIRD phases of the COVID-19 disease. These model choices are used in the design of both the large-scale compartmental model and the small-scale CHMMs throughout the chapter.

6.1.1 The Community Network Graph

Let $\mathcal{V}$ denote the set of individuals in the population, with a constant cardinality $N := |\mathcal{V}|$. We are primarily interested in the propagation of disease throughout a large-scale network, meaning $|\mathcal{V}|$ is very large. The interaction network of this population is represented as a network of *communities* $\mathcal{G}^c := (\mathscr{C}, \mathcal{E})$, where $\mathscr{C} := \{\mathcal{C}_k\}_{k=1}^K$, where $K \leq |\mathcal{V}|$ is the total number of communities and $\mathcal{E}$ is the set of edges. An edge $e(i,j) \in \mathcal{E}$ is created between two communities C_i and C_j if they share at least one member; the edge is assigned a weight $w(i,j) \in \mathbb{R}^{\geq 0}$ proportional to the number of members they share. Let $N_k := |\mathcal{C}_k|$ be the cardinality of community k, assumed constant, and denote $\mathscr{N}_k := \{\mathcal{C}_j \in \mathscr{C} | e(k,j) \in \mathcal{E}\}$. Every community forms a fully-connected subgraph where the nodes are the individuals. The interpretation of the communities is geographic, in that each community is assigned a certain *type* and each individual may belong to more than one community. For simplicity of presentation, we consider only *static types* in which the membership of individuals is mostly constant over time (e.g., routine locations such as schools, workplaces, private residences), as opposed to dynamic types where the membership of individuals may vary (stochastically) over time (e.g., special-occasion locations such as restaurants, amusement parks, musical concerts). Let $M \in \mathbb{N}$ be the total number of community types with $\mathscr{C} := \cup_{j=1}^M \mathscr{C}^{(j)}$, where $\mathscr{C}^{(j)}$ is the set of all communities of type j. For each individual $n \in \mathcal{V}$, let $\mathscr{V}_n := \{\mathcal{C} \in \mathscr{C} | n \in \mathcal{C}\}$ be the set of communities that n belongs to. A visual representation of the two graphs with $M = 3$ types of communities is shown in Figure 6.2.

Assumption 9. For all $k \in \{1, \cdots, K\}$, the members of community k remain as members of community k, with no new members joining and no present members departing. Thus, all susceptible individuals of k transition to becoming infectious members of k, from which they either die from the disease, or become a recovered member of group k. This assumption may be relaxed for time-varying networks, but for the main focus of this application, we assume static topologies.

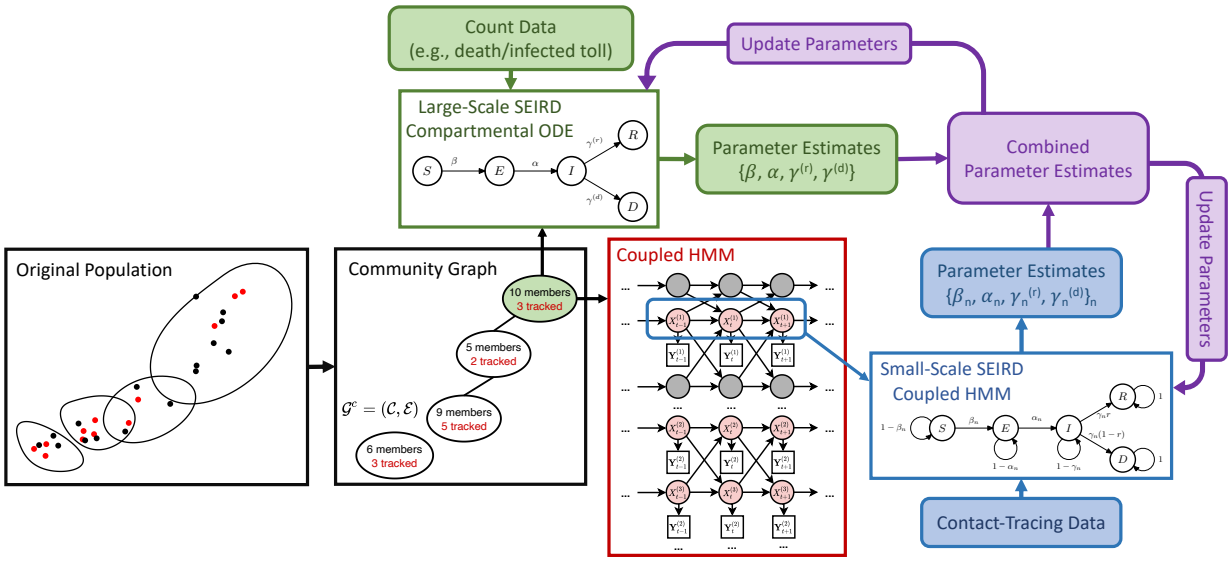

Figure 6.1: A flow diagram representation of the multiscale model for parameter estimation of COVID-19. Two entities are identified from the *original population* of individuals (dots): the different communities the individuals belong in (black circles), and the *tracked individuals*, i.e., those whose *contact-tracing data* is available (red dots). The resulting conversion is referred to as the *community graph* (see Section 6.1.1). Each community in the community graph is modeled according to the large-scale *SEIRD compartmental ODE model* (in green), and the parameters (i.e., transition rates) of the model is estimated using the available *count data* for the community (see Section 6.2). The interaction network of all individuals in each community is extracted as a *coupled hidden Markov model* (CHMM, in red), where the hidden state is defined as the individual's health status. Specifically, the SEIRD compartments for the different phases of COVID are embedded into each CHMM, and so each tracked individual is modeled according to a small-scale *SEIRD CHMM* (in blue). Gray nodes represent un-tracked individuals, whose health status remains unknown, and red nodes represent the health status of a tracked individual, which can be identified through observed symptoms (see Section 6.3). Contact tracing data is used in each tracked individual's CHMM to estimate his/her unique parameters (i.e., transition probabilities, true health status sequence). The estimates of the model parameters for the overall population network are then combined (in purple) via averaging; both the compartmental ODE and individual CHMM parameters are updated (see Section 6.4), and the models are then used for predicting quantities such as the future death toll.

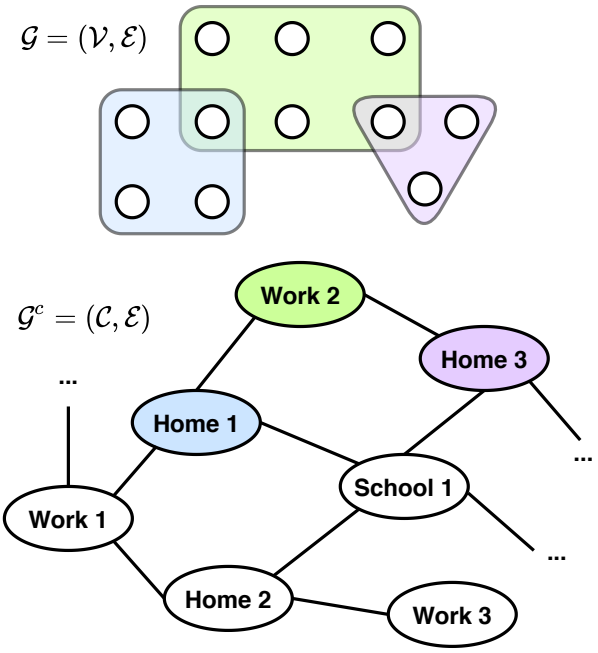

Figure 6.2: The full graph network of the individuals and the corresponding communities. Here, there are $M = 3$ types of communities, "work," "school," and "home'.'

6.1.2 SEIRD: The Phases of COVID-19

The evolution of COVID-19 spread within each community in $\mathcal{G}^c$ is modeled according to five main phases $\mathcal{X} := \{\mathtt{S}, \mathtt{E}, \mathtt{I}, \mathtt{R}, \mathtt{D}\}$. A majority of the population begins as *susceptible* (S) individuals, meaning they do not test positive for traces of virus on their bodies. In the next phase, individuals are *exposed* (E) to the virus (i.e., asymptomatic infected): they are carriers of the virus without any noticeable symptoms of the disease, but they are still capable of infecting susceptible individuals. After a certain period of time, individuals would either begin to display symptoms, upon which they transition to the *ill* (I) phase. Finally, any individual in (I) would either 1) succumb to death by the illness, transition to the *death* (D) phase, and remain there for the rest of time, or 2) successfully fight the disease and transition to the *recovered (R) phase*, where they are now considered invincible to the virus.

The SEIRD model generalizes the simpler but more standard SIR/SIRS models [79] by 1) distinguishing between recovered and deceased individuals, and 2) distinguishing two types of infectious individuals separately as exposed (i.e., asymptomatic infected) and ill (i.e., symptomatic infected) individuals. Note that both exposed and ill individuals are capable of spreading the virus to susceptible individuals; in the case of Singapore, as of March 17,

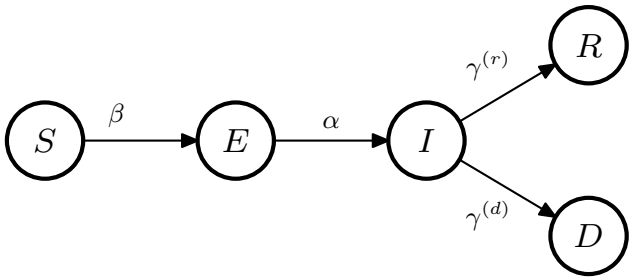

Figure 6.3: SEIRD for the ODE module.

2020, it is reported that roughly 48% of individuals who are ill with COVID-19 had been infected from exposed people alone [46]. However, in the case of COVID-19, it is important to maintain (E) and (I) as separate phases because of the time delay taken for symptoms to show; COVID-19 symptoms typically present within 2 to 14 days upon exposure [143]. For this reason, SEIRD captures a more accurate description of the COVID-19 disease compared to other types of compartment models [59].

Definition 36 (Infectious)**.** An *infectious* individual is one in which any other susceptible individual is liable to becoming exposed via social contact with him/her. These individuals test positive for the presence of virus on their bodies, but may not always display symptoms of the disease. In this specific setting, both exposed and ill individuals are considered infectious compartments.

6.2 ODE Module: Large-Scale Disease Propagation

6.2.1 Basic Framework

We begin by designing the ODE module, which is used to emulate virus spread throughout the community network $\mathcal{G}^c$. Traditional compartmental models assume random uniform interactions among individuals in a population, and do not consider groups, communities, or other heterogeneous interaction dynamics. This may lead to coarse approximations of quantities such as the predicted death toll, especially when a population is very large. In contrast to the traditional approach, we partition the population into a network of strongly-connected communities; this allows us to introduce basic heterogeneity into the model, distinguishing inter-community rate of spread from the intra-community rate of spread by changing the edge weights of $\mathcal{G}^c$.

For the purposes of modeling the COVID-19 pandemic, we choose to employ the five-compartment *SEIRD compartmental model*, visualized in Figure 6.3, where $\mathcal{X} := \{S, E, I, R, D\}$

are the phases described in Section 6.1.2. Mathematically, the ODE dynamics for a SEIRD model is governed by the following set of differential equations:

$$\frac{dS(t)}{dt} = -\beta(I(t) + E(t))S(t), \tag{6.1a}$$
$$\frac{dE(t)}{dt} = \beta(I(t) + E(t))S(t) - \alpha E(t), \tag{6.1b}$$
$$\frac{dI(t)}{dt} = \alpha E(t) - (\gamma^{(r)} + \gamma^{(d)})I(t), \tag{6.1c}$$
$$\frac{dR(t)}{dt} = \gamma^{(r)} I(t), \tag{6.1d}$$
$$\frac{dD(t)}{dt} = \gamma^{(d)} I(t). \tag{6.1e}$$

Here, the parameters are the *transition rates*, given by $\boldsymbol{\theta} := [\beta, \alpha, \gamma^{(r)}, \gamma^{(d)}]^\top$. We define β to be the contact rate between exposed and ill individuals with susceptible individuals in the population, α is the rate at which exposed individuals fall ill, and $\gamma^{(z)}$ is the rate that ill individuals leave their ill status, becoming recovered if $z = r$ or deceased if $z = d$.

Assumption 10. We do not take into account the natural births and natural deaths, as we are primarily focused on the effects of the disease itself on the existing population. We also consider the case where the network is closed, in that no interaction with any individuals outside of the population captured by the community network is allowed.

The SEIRD model for multiple communities (and multiple community types) is a simple extension of the original compartmental model (6.1). The compartmental model for the entire community network $\mathcal{G}^c$ can be visualized as K identical copies of Figure 6.3, connected to each other according to the topology imposed by $\mathcal{G}^c$. Toward this end, we define the parameter vector for community k as $\boldsymbol{\theta}_k := [\{\beta_{kj}\}_{j\in\{1,\cdots,K\}}, \alpha_k, \gamma_k^{(r)}, \gamma_k^{(d)}]^\top$ with subscript k on each parameter. Here, $[\beta_{kj}] \in \mathbb{R}^{K\times K}$ (with $\{\beta_{kj}\}_{j\in\{1,\cdots,K\}}$ its vectorized form for each $k \in \{1, \cdots, K\}$) defines the contact rate between each community of the network. Namely, β_{kk} denotes the rate of contact among susceptible and infectious members which belong in the same community k, while β_{kj} denotes the average rate of contact among susceptible individuals in community k and infectious individuals in community j for all other communities $j \neq k$. The intuition for this definition is that a susceptible individual in some community k has at most K different ways of becoming infected (i.e., by coming into contact with a individual of any other community), depending on the topology of the community network.

Mathematically, for each k, the dynamics are governed by the following set of differential equations:

$$\frac{dS_k(t)}{dt} = -\sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{N_j} S_k(t), \tag{6.2a}$$

$$\frac{dE_k(t)}{dt} = \sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{N_j} S_k(t) - \alpha_k E_k(t), \tag{6.2b}$$

$$\frac{dI_k(t)}{dt} = \alpha_k E_k(t) - (\gamma_k^{(r)} + \gamma_k^{(d)}) I_k(t), \tag{6.2c}$$

$$\frac{dR_k(t)}{dt} = \gamma_k^{(r)} I_k(t), \tag{6.2d}$$

$$\frac{dD_k(t)}{dt} = \gamma_k^{(d)} I_k(t), \tag{6.2e}$$

where the parameters are defined previously.

Assumption 11. Each community $\mathcal{C}_k$ has at least one *tracked individual*, i.e., an individual whose contact-tracing data over time is available. We define $\overline{\mathcal{C}}_k$ to be the set of all tracked individuals in community k.

Remark 24. The contact parameter β_{kj} between communities k and j is proportional to the edge weight $w(i, j)$, which describes the strength of the connection between the two communities. The weight $w(i, j)$ increases when the two communities interact more frequently with each other.

Now, we describe a method for estimating the transition rate parameters in $\boldsymbol{\theta}_k$ for each community k based on count data. First, the estimates $\{\tilde{\gamma}_k^{(r)}, \tilde{\gamma}_k^{(d)}\}$ of community k's true rates of recovery and death $\{\gamma_k^{(r)}, \gamma_k^{(d)}\}$ can be determined by observing time-series data of its recovered and death tolls over time, which are some of the most well-reported statistics pertaining to COVID-19. Second, estimating α_k is difficult from count data alone since, unlike the recovery and death tolls, $E_k(t)$ is difficult to obtain due to the lack of visible symptoms in exposed individuals. We thus rely entirely on contact-tracing data from individuals in $\overline{\mathcal{C}}_k$ to estimate α_k; this estimation procedure is detailed further in Section 6.4. Finally, estimating the rates $[\beta_{kj}]_{k,j\in\{1,\cdots,K\}}$ in which susceptible individuals of each community transition to becoming exposed (asymptomatic infected) is tricky due to their dependence on other communities, and we also rely on individual contact-tracing data. Thus, we also defer discussion of their estimation to Section 6.4.

6.2.2 A Stochastic Extension

The effect of *superspreaders* typically arises when a small handful of infectious people come in contact with an abnormally large number of susceptible individuals. In COVID-19, a notable example of this effect is the individual known as Patient 31, who is believed to have been a primary cause for the sudden spike in cases of COVID-19 in South Korea, which eventually lead to hundreds of deaths between late February and early March of 2020 [78]. Modeling phenomena like these requires some modification to the SEIRD compartmental model described previously, especially since the deterministic ODEs (6.1) and (6.2) cannot capture the randomness of the superspreader effect.

The superspreader effect can be emulated by varying the rates $[\beta_{kj}]$ of contact between susceptible and infectious individuals in communities. For compartmental ODEs, white-noise perturbations to the contact rates have been accounted for in prior work [63, 24, 23] to mimic slight variations in the daily number of contacts. However, these variations are small and occur steadily as a function of time; white noise is therefore insufficient to represent sudden drastic increases in the contact rate, which give rise to the phenomenon of superspreaders. Rather, impulse disturbances are better modeled as *Poisson shot noise*, which has as much real-world presence and mathematical development in literature as Gaussian white noise does (see Chapter 2).

Using shot noise, the evolution of $\beta_{kj}(t)$, now a time-varying quantity, for each $k, j \in \{1, \cdots, K\}$ can be characterized by the stochastic differential equation (SDE):

$$\beta_{kj}dt = \beta_{kj,0}dt + \sigma_{kj}dW(t) + \xi_{kj}dN_{kj}(t), \tag{6.3}$$

where $\beta_{kj,0}$ is the baseline constant value of the parameter, W is standard white noise, N_{kj} is standard Poisson noise with intensity parameter λ_{kj}, and σ_{kj} and ξ_{kj} are their corresponding variances. An SDE of the form (6.3) replaces each entry in the original constant matrix $[\beta_{kj}]$ considered before in (6.2). The parameters λ_{kj}, σ_{kj}, and ξ_{kj} are chosen based on community type, especially those where it is suspected that a higher-than-average frequency of contact might occur (e.g., homes, schools). For example, when superspreaders emerge only from interactions within group k, β_{kk} abides by (6.3) while the other rates $\beta_{kj}, j \neq k$ remain constant, and the deterministic ODE (6.2) is modified as follows.

$$\begin{aligned} dS_k(t) = &\left(-Z(t) - \frac{\beta_{kk,0}(I_k(t) + E_k(t))}{N_k} S_k(t)\right) dt \\ &- \frac{(I_k(t) + E_k(t))S_k(t)}{N_k}(\sigma_{kk}dW(t) + \xi_{kk}dN(t)), \end{aligned} \tag{6.4a}$$

$$dE_k(t) = \left(Z(t) + \frac{\beta_{kk,0}(I_k(t) + E_k(t))}{N_k} S_k(t) - \alpha_k E_k(t) \right) dt \tag{6.4b}$$
$$+ \frac{(I_k(t) + E_k(t))S_k(t)}{N_k} (\sigma_{kk} dW(t) + \xi_{kk} dN(t)),$$

where the variable Z refers to the sum:

$$Z(t) := \sum_{j \neq k} \frac{\beta_{kj}(I_j(t) + E_j(t))}{N_j} S_k(t),$$

and the other equations in (6.2) are kept the same. For further generality, $\lambda_{kj}(t), \sigma_{kj}(t)$, and $\xi_{kj}(t)$ can also be time-varying random variables.

6.2.3 Including Multiple Variants

With the recent rise of mutations of SARS-CoV-2 such as the Delta variant and the Omicron variant, it now becomes imperative to consider more than one strain of virus. Gene sequencing analysis suggests that the possibility of infection from a variant virus is highly probable even if an individual has recovered from infection of the original strain. It becomes necessary to modify the previously-described SEIRD compartment model in the following way: once an individual has recovered from a variant virus, (s)he transitions to being susceptible to the variants (s)he has not yet been exposed to.

Let $A \in \mathbb{N}$ be the total number of known strains of SARS-CoV-2 which are being propagated throughout the network. In order to accommodate multiple strains of the virus, we make the following modifications to the states of the SEIRD compartmental model described previously:

- (S)usceptible individuals may now be susceptible to a variety of combinations of strains. We hence revise the notation of compartment (S) as $(S[x_1, \cdots, x_A])$, where $x_a = 1$ for individuals susceptible to strain $a \in \{1, \cdots, A\}$ and $x_a = 0$ for individuals protected from strain a. The vector $[x_1, \cdots, x_A]$, which we refer to as the *(virus) susceptibility profile*, takes values in $\mathscr{A}^{(S)} \subset \{0, 1\}^A$; in fact, there are a total of $2^A - 1$ possible combinations of strains any individual may be susceptible to because $[0, \cdots, 0]$ corresponds to a fully-recovered individual, which corresponds to the (R) compartment.

- The (E)xposed compartment is revised similarly to the (S)usceptible compartment, with notation $(E[y_1, \cdots, y_A])$ such that $y_a = 1$ for individuals exposed to strain $a \in \{1, \cdots, A\}$, $y_a = 0$ if protected from strain a, and $y_a = -1$ if susceptible but not

exposed to strain a. We assume multiple strains of the virus may not simultaneously infect an individual, so any vector $[y_1, \cdots, y_A]$ can only contain exactly one zero; we henceforth define $\mathscr{A}^{(I)} \subset \{-1, 0, 1\}^A$ to be the set of possible values of $[y_1, \cdots, y_A]$. Likewise, the (I)ll compartment is revised with notation $(I[y_1, \cdots, y_A])$ such that $y_a = 1$ for individuals ill from strain $a \in \{1, \cdots, A\}$, $y_a = 0$ if protected from strain a, and $y_a = -1$ if susceptible but not ill from strain a. Note the possible set of values $[y_1, \cdots, y_A]$ for (I) compartment is the same as those of the (E) compartment; we henceforth refer to the set $\mathscr{A}^{(I)}$ as the set of possible *(virus) infectiousness profiles*.

- There is still only a single (R)ecovered compartment and a single (D)eath compartment.

Correspondingly, we redefine the phase space as

$$\mathcal{X} := \{R, D\} \cup \left(\bigcup_{[x_1, \cdots, x_A] \in \mathscr{A}^{(S)}} S[x_1, \cdots, x_A] \right) \cup \left(\bigcup_{[y_1, \cdots, y_A] \in \mathscr{A}^{(I)}} E[y_1, \cdots, y_A] \cup I[y_1, \cdots, y_A] \right), \tag{6.5}$$

and to account for the uniqueness of each virus strain, the vector $\boldsymbol{\theta}_k, k \in \{1, \cdots, K\}$ of transition rate parameters from the compartmental model of Section 6.2 is updated as follows:

$$\boldsymbol{\theta}_k := \Bigg[\{\beta_{i,kj}\}_{\substack{i \in \{1, \cdots, |\mathscr{A}^{(I)}|\} \\ j \in \{1, \cdots, K}}}, \{\alpha_{i,k}\}_{i \in \{1, \cdots, |\mathscr{A}^{(I)}|}}, \{\gamma^{(r)}_{i,k}\}_{i \in \{1, \cdots, A\}}, \{\gamma^{(d)}_{i,k}\}_{i \in \{1, \cdots, |\mathscr{A}^{(I)}|\}}, \{\nu_{i,k}\}_{i \in \{1, \cdots, |\mathscr{A}^{(I)}| - A\}} \Bigg], \tag{6.6}$$

where $\nu_{i,k}$ is the rate of recovery of community k from virus strain i of individuals who are susceptible to multiple strains (including i).

Remark 25. In this paper, we only consider the case where the total number of variants A is known beforehand. However, when a new $(A+1)$th strain is identified to be propagating throughout the population, a new group is created with the sole infectious individual being its only member for the time being.

The equations governing the compartment model for multiple strains is a simple extension of (6.1). With Assumption 10 in place, we have that for each time t:

$$\sum_{[x_1, \cdots, x_A] \in \mathscr{A}^{(S)}} S[x_1, \cdots, x_A](t) + R(t) + D(t)$$

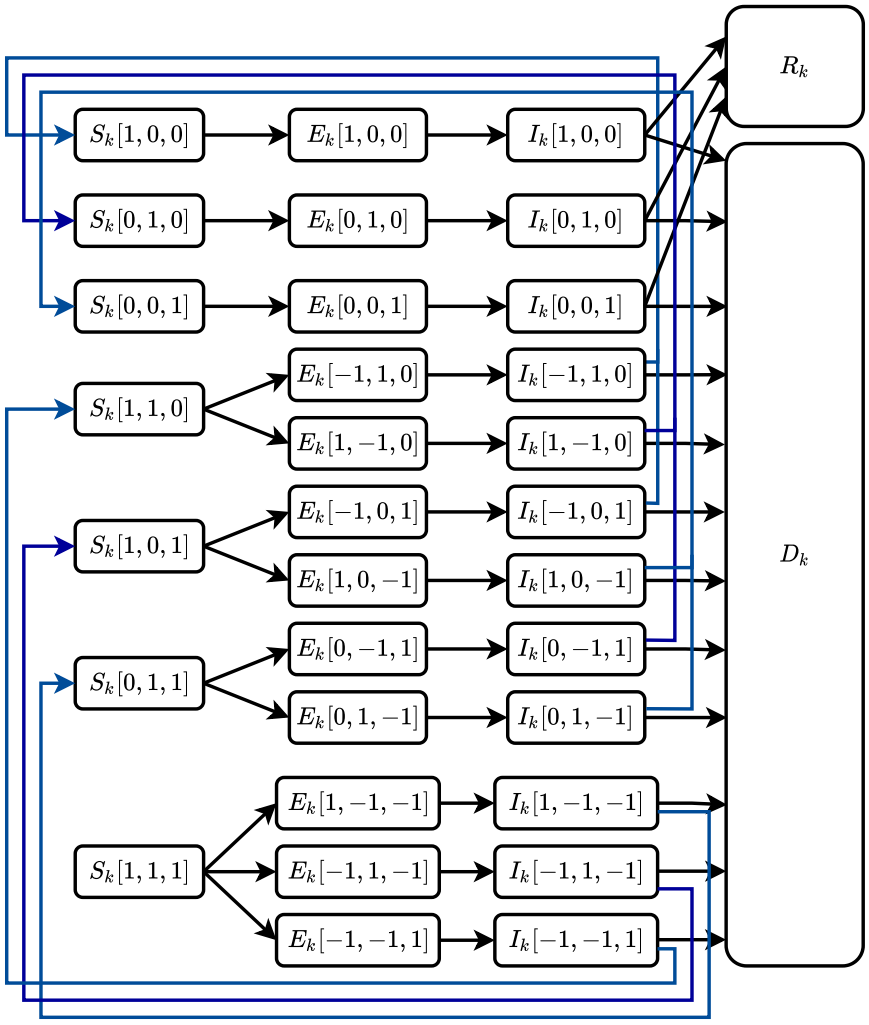

Figure 6.4: SEIRD for the ODE module for a single individual $v \in \mathcal{V}$ with $A = 3$ possible virus strains, using transition rates as parameters. Here, a (S) compartment with value 0 in strain $a \in \{1, \cdots, A\}$ means the individual is protected against strain a, and a 1 means that (s)he is susceptible. A (E) compartment with value 0 in strain a means protection, 1 for exposure (asymptomatic infection), and -1 for susceptibility but not exposure; similarly for the (I) compartment with 1 representing illness instead.

$$+ \sum_{[y_1,\cdots,y_A]\in\mathscr{A}^{(I)}} (E[y_1,\cdots,y_A](t) + I[y_1,\cdots,y_A](t)) = N_k.$$

For concreteness and simplicity of description, we henceforth limit our discussion to the case of $A = 3$ different strains; the SEIRD compartmental model for a single community different strains is thus shown in Figure 6.4. For simplicity of notation, we enumerate the vectors

$$\begin{aligned}
&\mathbf{c}_1 := [1,0,0], \quad \mathbf{c}_2 := [0,1,0], \quad \mathbf{c}_3 := [0,0,1], \quad \mathbf{c}_4 := [1,1,0]\\
&\mathbf{c}_5 := [1,0,1], \quad \mathbf{c}_6 := [0,1,1], \quad \mathbf{c}_7 := [1,1,1], \quad \mathbf{d}_1 := [-1,1,0]\\
&\mathbf{d}_2 := [1,-1,0], \quad \mathbf{d}_3 := [-1,0,1], \quad \mathbf{d}_4 := [1,0,-1], \quad \mathbf{d}_5 := [0,-1,1]\\
&\mathbf{d}_6 := [0,1,-1], \quad \mathbf{d}_7 := [1,-1,-1], \quad \mathbf{d}_8 := [-1,1,-1], \quad \mathbf{d}_9 := [-1,-1,1]
\end{aligned}$$

$\forall\, i = 1,2,3:$

$$\begin{aligned}
\frac{dS_k[\mathbf{c}_i](t)}{dt} = &- \sum_{j\in\mathcal{N}_k} \frac{\beta_{i,kj}(I_j[\mathbf{c}_i](t) + E_j[\mathbf{c}_i](t))}{N_j} S_k[\mathbf{c}_i](t)\\
&+ \begin{cases} \nu_1 I_k[\mathbf{d}_1](t) + \nu_3 I_k[\mathbf{d}_3](t) & \text{if } i = 1\\ \nu_2 I_k[\mathbf{d}_2](t) + \nu_5 I_k[\mathbf{d}_5](t) & \text{if } i = 2\\ \nu_4 I_k[\mathbf{d}_4](t) + \nu_6 I_k[\mathbf{d}_6](t) & \text{if } i = 3 \end{cases}
\end{aligned} \tag{6.7a}$$

$\forall\, i = 4,5,6:$

$$\frac{dS_k[\mathbf{c}_i](t)}{dt} = - \sum_{j\in\mathcal{N}_k} \frac{\beta_{i,kj} \sum_{\ell=2i-7}^{2i-6} (I_j[\mathbf{d}_\ell](t) + E_j[\mathbf{d}_\ell](t))}{N_j} S_k[\mathbf{c}_i](t) + \nu_{i+3,k} I_k[\mathbf{d}_{i+3}](t)$$

$$\frac{dS_k[\mathbf{c}_7](t)}{dt} = - \sum_{j\in\mathcal{N}_k} \frac{\sum_{\ell=7}^{9} \beta_{\ell+3,kj} (I_j[\mathbf{d}_\ell](t) + E_j[\mathbf{d}_\ell](t))}{N_j} S_k[\mathbf{c}_7](t)$$

$\forall\, i = 1,2,3:$

$$\frac{dE_k[\mathbf{c}_i](t)}{dt} = \sum_{j=1}^{K} \frac{\beta_{i,kj}(I_j[\mathbf{c}_i](t) + E_j[\mathbf{c}_i](t))}{N_j} S_k[\mathbf{c}_i](t) - \alpha_{i,k} E_k[\mathbf{c}_i](t) \tag{6.7b}$$

$\forall\, i = 1,\cdots 6:$

$$\frac{dE_k[\mathbf{d}_i](t)}{dt} = \sum_{j=1}^{K} \frac{\beta_{i+3,kj}(I_j[\mathbf{d}_i](t) + E_j[\mathbf{d}_i](t))}{N_j} S_k[\mathbf{c}_{\lceil i/2\rceil+3}](t) - \alpha_{i,k} E_k[\mathbf{d}_i](t)$$

$$\forall \mathbf{e} \in \{\mathbf{c}_1, \cdots, \mathbf{c}_7, \mathbf{d}_1, \cdots, \mathbf{d}_9\}:$$

$$\frac{dI_k[\mathbf{e}](t)}{dt} = \alpha_{\cdot,k} E_k[\mathbf{e}](t) - \gamma^{(d)}_{\cdot,k} I_k[\mathbf{e}](t) \tag{6.7c}$$

$$\frac{dR_k(t)}{dt} = \sum_{i=1}^{3} \gamma^{(r)}_{i,k} I_k(t) \tag{6.7d}$$

$$\frac{dD_k(t)}{dt} = \sum_{\mathbf{e} \in \{\mathbf{c}_1, \cdots, \mathbf{d}_9\}} \gamma^{(d)}_{\cdot,k} I_k(t) \tag{6.7e}$$

where the parameters are defined in (6.6) and the subscript of $\cdot$ in (6.7c) and (6.7e) is replaced with the appropriate index.

6.3 CHMM Module: Individual-Level Propagation

While the SEIRD compartmental model is used to propagate the health status of each community, *coupled hidden Markov models (CHMMs)* [22] are used to represent the health status of certain tracked individuals. Detailed contact-tracing data from a single individual offers further insight into the parameters of the dynamics of the disease spread within smaller subsets of each community.

6.3.1 Parameter Estimation from Contact-Tracing

Let $X_n(t) \in \mathcal{X}$ be the hidden state of the HMM for individual $n \in \mathcal{V}$ for all time $t \in \mathbb{N}$, with $\mathcal{X}$ the set of possible phases defined in Section 6.1.2; in the context of epidemic modeling, $X_n(t)$ is the health status of individual $n \in \mathcal{V}$ at each time $t \in \mathbb{N}$. A *chain* $n \in \mathcal{V}$ of the CHMM is a sequence of hidden states $\{X_n(t), t \geq 0\}$ over time for the single individual v, and they are connected to the chains corresponding to the other individuals v interacts with. We denote random vector $\mathbf{X}_n(t_1 : t_2) := (X_n(t_1), X_n(t_1 + 1), \cdots, X_n(t_2))$ for $0 < t_1 < t_2$ to be the sequence of health statuses for individual $n \in \mathcal{V}$ between times t_1 and t_2, and the vector of deterministic values it takes as lowercase $\mathbf{x}_n(t_1 : t_2) := (x_n(t_1), x_n(t_1 + 1), \cdots, x_n(t_2))$.

Although the true health status of any individual may be unknown (especially when (s)he is in the (E) phase), each individual releases a set of observable symptoms (e.g., fever, sore throat, etc.) which can be used to determine whether or not (s)he is infectious (exposed or ill). Suppose we are interested in keeping track of $B \in \mathbb{N}$ specific symptoms of COVID-19; for simplicity, we assume that they occur independently of each other. Let $Y_n(t) := [Y_{n,1}(t), \cdots, Y_{n,B}(t)]^\top \in \{0, 1\}^B$ represent the observed symptoms vector of individual $n \in \mathcal{V}$ at time t, where $Y_{n,j} = 0$ if the individual does not exhibit symptom $j \in \{1, \cdots, B\}$, and 1 otherwise. Define the *observation probability matrix (OPM)* $O_n(Y_{n,\cdot}(t)|\cdot) \in \mathbb{R}^{B \times |\mathcal{X}|}$ for

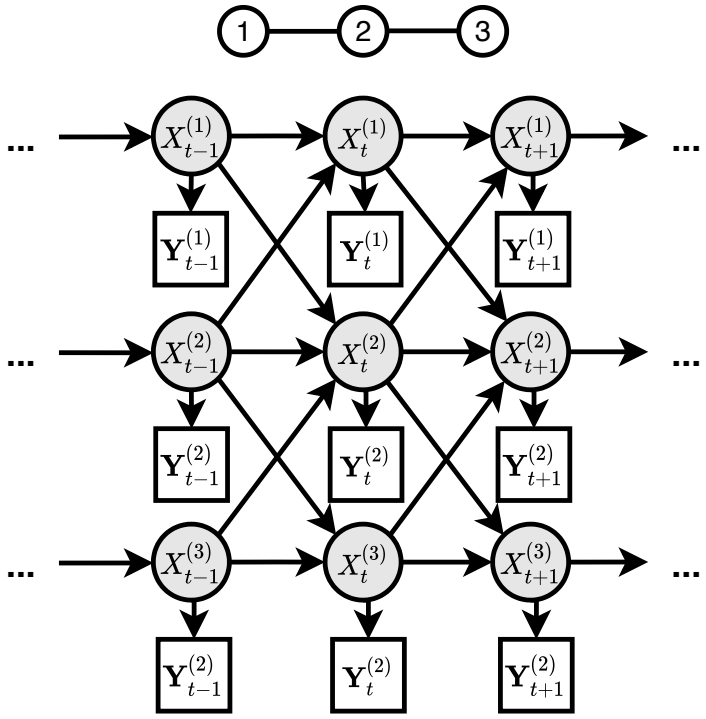

Figure 6.5: A sample coupled hidden Markov model relating the health statuses of three individuals in a community, along with the observed symptoms.

individual $n \in \mathcal{V}$, with entries $O_n(Y_{n,j}(t)|X_n(t))$ denoting the probability of observing the positive presence symptom j in individual v at time t, i.e., $Y_{n,j}(t) = 1$. Note that each row of this matrix does not necessarily need to sum to 1 since the entries correspond only to the case $y_{n,j} = 1$. The boldfaced notation for $\mathbf{X}$ and $\mathbf{x}$ also extends to $\mathbf{Y}_n, \mathbf{y}_n$ in the same way, as $\{\mathbf{Y}_n(0:t) = \mathbf{y}_n(0:t)\} \equiv \{\mathbf{Y}_{n,1}(0:t) = \mathbf{y}_{n,1}(0:t), \cdots, \mathbf{Y}_{n,B}(0:t) = \mathbf{y}_{n,B}(0:t)\}$.

Assumption 12. For each $n \in \mathcal{V}$, the OPM O_n is given and known.

A sample CHMM of a specific community with three individuals, not strongly-connected, is visualized in Figure 6.5. Because contact-tracing data only provides us information about the evolution of the observed symptoms over time for a subset of tracked individuals, the transition probabilities among the different phases in $\mathcal{X}$ are unknown. Each individual is assigned a vector of unknown parameters similar to $\boldsymbol{\theta}_k$ for the compartmental model of each community $k \in \{1, \cdots, K\}$. For individual $n \in \mathcal{V}$, the full vector of transition probability parameters is given by $\boldsymbol{\eta}_n(t) := [\beta_n(t), \alpha_n, \gamma_n^{(r)}, \gamma_n^{(d)}]$, and the sparsity pattern of the *transition*

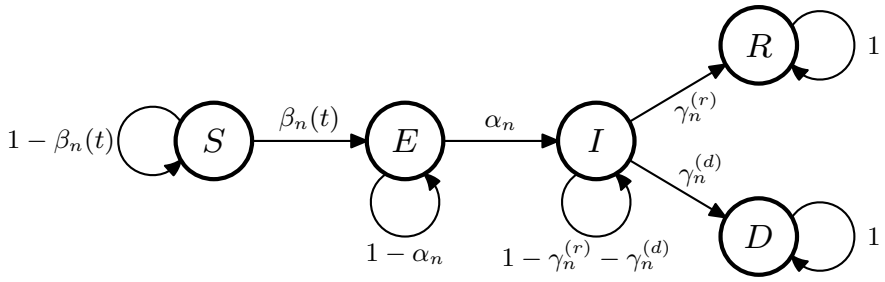

Figure 6.6: The underlying Markov chain for a single chain of the CHMM module, using transition probabilities as parameters.

probability matrix (TPM) corresponding to the chain of v is given by

$$P_n(t) := \begin{bmatrix} 1-\beta_n(t) & \beta_n(t) & 0 & 0 & 0 \\ 0 & 1-\alpha_n & \alpha_n & 0 & 0 \\ 0 & 0 & 1-\gamma_n^{(r)}-\gamma_n^{(d)} & \gamma_n^{(r)} & \gamma_n^{(d)} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.8}$$

Note that the probability for transitioning from `S` to `E` is time-varying because it is dependent on the time-varying health statuses of his/her immediate neighbors.

Given a complete sequence $\{\mathbf{y}_n(0:T_{\text{sim}})\}$ of observed symptoms over some time duration $T_{\text{sim}} > 0$, we address two questions for each individual $n \in \mathcal{V}$. Question 1: how can we estimate the values of the TPM $P_n(t)$ in the CHMM? Question 2: given the TPM estimates $\hat{P}_n(t)$ for all $t \in [0, T_{\text{sim}}]$, how can we estimate the true health status $x_n(0:t)$? Both question can be addressed by extending standard HMM techniques (see, e.g., [65]). For Question 1, the *forward-backward algorithm* (e.g., [126]) and *Baum-Welch (expectation-maximization)* (e.g., [16]) are standard procedures in the HMM literature which can estimate the transition and observation probabilities in P_n and O_n. For the purposes of this application, we make two simultaneous extensions: 1) multiple different time series of observations can be incorporated at once, and 2) the unknown parameters are assumed to be time-varying.

Define $f_{n,j}(t,x) := \mathbb{P}(X_n(t) = x, \mathbf{Y}_{n,j}^{(t)} = \mathbf{y}_{n,j}^{(t)})$ to be the probability that the individual is in state $x \in \mathcal{X}$ at time t and the past observed symptom sequence is given by $\mathbf{Y}_{n,j}^{(t)} = \mathbf{y}_{n,j}^{(t)}$. Define $b_{n,j}(t,x) := \mathbb{P}(\mathbf{Y}_{n,j}^{(t+1:T)} = \mathbf{y}_{n,j}^{(t+1:T)} | X_n(t) = x)$ to be the probability of observing a future sequence of symptoms $\mathbf{Y}_{n,j}^{(t+1:T)} = \mathbf{y}_{n,j}^{(t+1:T)}$ given we know the individual is in state x. The

recursive equations for $f_{n,j}$ and $b_{n,j}$ are then given by:

$$f_{n,j}(t,x) = \sum_{z\in\mathcal{X}} f_{n,j}(t-1,z)O_n(y_{n,j}(t)|x)\hat{P}_{n,j}^{(t-1)}(z,x), \;\; f_{n,j}(0,x) := q_n(x)O_n(y_{n,j}(0)|x), \tag{6.9a}$$

$$b_{n,j}(t,x) = \sum_{z\in\mathcal{X}} b_{n,j}(t+1,z)\hat{P}_{n,j}^{(t)}(x,z)O_n(y_{n,j}(t+1)|z), \quad b_{n,j}(T,x) = 1 \;\forall x \in \mathcal{X}. \tag{6.9b}$$

Given observation sequence $\mathbf{Y}_{n,j}(0:T_{\text{sim}}) = \mathbf{y}_{n,j}(0:T_{\text{sim}})$, define $g_{n,j}(t,x)$ to be the probability that the state of individual n at time t is x given observation sequence j, and $h_{n,j}(t,x,z)$ to be the probability that the state of individual n makes a transition from x to z at time t:

$$g_{n,j}(t,x) := \mathbb{P}(X_n(t) = x|\mathbf{Y}_{n,j}(0:T_{\text{sim}}) = \mathbf{y}_{n,j}(0:T_{\text{sim}})), \tag{6.10a}$$

$$h_{n,j}(t,x,z) := \mathbb{P}(X_n(t) = x, X_n(t+1) = z|\mathbf{Y}_{n,j}(0:T_{\text{sim}}) = \mathbf{y}_{n,j}(0:T_{\text{sim}})). \tag{6.10b}$$

The variables defined in (6.9) allow us to simplify (6.10) beyond their definitions:

$$g_{n,j}(t,x) = \frac{f_{n,j}(t,x)b_{n,j}(t,x)}{\sum_{z\in\mathcal{X}} f_{n,j}(t,z)b_{n,j}(t,z)}, \tag{6.11a}$$

$$h_{n,j}(t,x,z) = \frac{f_{n,j}(t,x)\hat{P}_{n,j}^{(t)}(x,z)O_n(\mathbf{y}_{n,j}(t+1)|z)b_{n,j}(t+1,z)}{\sum_{u,w\in\mathcal{X}} f_{n,j}(t,u)\hat{P}_{n,j}^{(t)}(u,w)O_n(\mathbf{y}_{n,j}(t+1)|w)b_{n,j}(t+1,w)}. \tag{6.11b}$$

Note that the expressions for (6.11) are dependent on previous estimates of the TPM $\hat{P}_{n,j}^{(0:t-1)}$ for each time t; essentially, we are recursively building new estimates of $\hat{P}_{n,j}^{(t)}$ based on the previous time's estimates. For a single individual $n \in \mathcal{V}$, estimating the TPM $P_n(t)$ based on a single observation sequence $j \in \{1,\cdots,B\}$ can be solved according to the standard Baum-Welch algorithm [16]. Define $\hat{\eta}_{n,j}^{(t)}$ to be the estimate of the true parameter vector $\boldsymbol{\eta}_n$ at time t based on observation sequence j, and define a corresponding auxiliary function as:

$$Q_{n,j}(t) := \mathbb{E}\left[\log\left(p_{n,j}^{(c)}(\mathbf{X}_n(0{:}t), \mathbf{Y}_{n,j}(0{:}t)|\boldsymbol{\eta}_n(0{:}t))\right)\middle|\mathbf{y}_{n,j}(0{:}T), \hat{\eta}_{n,j}^{(0:t)}\right], \tag{6.12}$$

where $p_{n,j}^{(c)}$ denotes the joint probability distribution of observing a complete set of data $\{\mathbf{x}_n(0{:}t), \mathbf{y}_{n,j}(0{:}t)\}$ for individual n:

$$p_{n,j}^{(c)}(\mathbf{x}_n(0{:}t), \mathbf{y}_{n,j}(0{:}t)|\boldsymbol{\eta}_n(0{:}t)) = q_n(x_n(0))\prod_{s=0}^{t-1} P_n(s,x_n(s),x_n(s+1))\prod_{s=0}^{t} O_n(y_{n,j}(s)|x_n(s)).$$

We maximize the (6.12) to determine the optimal initial probability distribution $\hat{q}_{n,j}^{(t)}$ and the optimal TPM $\hat{P}_{n,j}^{(t)}$. Note that the maximization must be done subject to the regularity conditions $\sum_{u\in\mathcal{X}} \hat{P}_{n,j}^{(t)}(x,u) = 1$ and $\sum_{x\in\mathcal{X}} \hat{q}_{n,j}^{(t)}(x) = 1$ for all $x \in \mathcal{X}$. The optimal point has the following closed-form expression:

$$\hat{q}_{n,j}^{(t)}(x) = g_{n,j}(0,x), \tag{6.13a}$$

$$\hat{P}_{n,j}^{(t)}(x,z) = \left(\sum_{s=0}^{t-1} h_{n,j}(s,x,z)\right)\left(\sum_{s=0}^{t} g_{n,j}(s,x)\right)^{-1}, \tag{6.13b}$$

where the $g_{v,j}(t,x)$ and $h_{v,j}(t,x,z)$ are defined in (6.10). The procedure is repeated for each $t \in [0, T_{\text{sim}}]$ so that we obtain an estimate of $\hat{q}_{n,j}^{(t)}$ and $\hat{P}_{n,j}^{(t)}$ which evolves over time.

In order to account for time-varying parameters, we apply a discounting factor $a \in (0,1]$ which weights the values of past estimates less the further back in the past they were observed. To aggregate multiple observations into a single definitive estimate, define $\mathbf{w} \in \mathbb{R}^B$ to be weights such that $\sum w_j = 1$:

$$\hat{P}_n(t,x,z) = \left(\sum_{j=1}^{B} w_j \sum_{s=0}^{t-1} a^{t-s} h_{n,j}(s,x,z)\right)\left(\sum_{j=1}^{B} w_j \sum_{s=0}^{t} a^{t-s} g_{n,j}(s,x)\right)^{-1}. \tag{6.14}$$

The assignment of weights is chosen via two metrics: 1) the observation sequences are statistically correlated with each other, or 2) one observation sequence yields more information about a state than another, e.g., observing a fever on an individual may be more reflective of his/her ill state than a runny nose. For simplicity, we assume that these weights are known beforehand and that our observation processes are independent of each other, meaning that the weights are only chosen according to how well they represent the true state.

Question 2 can be addressed by applying the standard Viterbi algorithm to each separate observation sequence, then aggregating them. Specifically, recall $X_n(t) \in \mathcal{X}$ refers to the hidden state of individual $n \in \mathcal{V}$, and suppose we are given a time series of observations $Y_{n,j}(0:t)$ for symptom $j \in \{1, \cdots, B\}$. The estimated time-varying TPM underlying the HMM is given by $\hat{P}_n^{(t)}$ at time t, and the known OPM is given by O_n. The initial state is known and given by $X_n(0) = x_n(0)$. Then the standard *Viterbi algorithm* (e.g., [58]) can be applied to the observation sequence $j \in \{1, \cdots, B\}$ to estimate the sequence $\hat{x}_{n,j}(1:t)$ of likely hidden states over time based on symptom j. The probability of observing some specific sequence of health statuses $x_n(1:t)$ for some $t \leq T_{\text{sim}}$ is given by:

$$\mathbb{P}(\{X_n(t) = x_n(t), n \in \mathcal{V}, t \in [0,T]\})$$

$$= \prod_{n\in\mathcal{V}} q_n(x_n(0)) \prod_{\substack{t\in[0,T-1]\\ n\in\mathcal{V}}} \mathbb{P}(X_n(t+1)|X_n(t), \{X_m(t), m\in\mathcal{N}(n)\}),$$

where $q_n(x)$ denotes the initial probability that individual n starts off at state x. Based on the observations of an individual's symptoms, we recursively compute:

$$\begin{aligned}
\delta_{n,j}(0,x) &= q_n(x)O_n(y_{n,j}(0)|x),\\
\delta_{n,j}(t,x) &= \max_{z\in\mathcal{X}} \delta_{n,j}(t-1,z)\hat{P}_n^{(t-1)}(z,x)O_n(y_{n,j}(t)|x),\ t\geq 1.
\end{aligned}$$

Then for the specific observation sequence j, the optimal sequence of states is given by $\hat{x}_{n,j}(t) := \text{argmax}_{z\in\mathcal{X}}\delta_{n,j}(t,z)$. Thus, $\hat{x}_{n,j}(t)\in\mathcal{X}$ is the most likely health status of individual $n\in\mathcal{V}$ at time $t\in[0,T_{\text{sim}}]$ given observation process $j\in\{1,\cdots,B\}$. Then the health status $\hat{x}_n(t)$ determined by considering all observation processes simultaneously is then given by whichever phase in $\mathcal{X}$ occurs most often in the aggregate set $\{\hat{x}_{n,1}(t),\cdots,\hat{x}_{n,B}(t)\}$. Ties are broken according to the state which is more "harmful" to the network, e.g., if the most likely state is tied between susceptible (`S`) or exposed (`E`), then we take the individual to be exposed because (s)he is liable to infecting more people in the network.

6.3.2 Including Multiple Variants

We can extend the CHMM module to account for variant viruses and mutations in a way similar to what was done for the compartmental ODE module (Section 6.2.3). The unknown probabilities for the CHMM module, expanded to consider multiple strains, is given by

$$\begin{aligned}
\boldsymbol{\eta}_v(t) := \Big[&\{\beta_{i,v}(t)\}_{\substack{i\in\{1,\cdots,|\mathscr{A}^{(S)}|\}\\ j\in\{1,\cdots,K\}}}, \{\alpha_{i,v}(t)\}_{i\in\{1,\cdots,|\mathscr{A}^{(S)}|+|\mathscr{A}^{(I)}|\}},\\
&\{\gamma_{i,v}^{(r)}\}_{i\in\{1,\cdots,A\}}, \{\gamma_{i,v}^{(d)}(t)\}_{i\in\{1,\cdots,|\mathscr{A}^{(S)}|+|\mathscr{A}^{(I)}|\}}, \{\nu_{i,v}(t)\}_{i\in\{1,\cdots,|\mathscr{A}^{(I)}|\}}\Big].
\end{aligned} \tag{6.15}$$

Furthermore, the TPM $P_v^{(t)}$ for each $v\in\mathcal{V}$ and time $t\in\mathbb{N}$ is updated similarly to (6.7):

$$P_v^{(t)} = \begin{bmatrix}
P_{v,SS}^{(t)} & P_{v,SE}^{(t)} & 0 & 0 & 0\\
0 & P_{v,EE}^{(t)} & P_{v,EI}^{(t)} & 0 & 0\\
P_{v,IS}^{(t)} & 0 & P_{v,II}^{(t)} & P_{v,IR}^{(t)} & P_{v,ID}^{(t)}\\
0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1
\end{bmatrix},$$

where each of the submatrices $P^{(t)}_{v,\cdot}$ are defined as follows, using the parameters defined in (6.15):

$$P^{(t)}_{v,SE} = \text{diag}\Big\{ \text{diag}\{\beta_{1,v}(t), \cdots, \beta_{A,v}(t)\}, \tag{6.16}$$
$$\text{diag}\{\beta_{A+1,v}(t) + \beta_{A+2,v}(t), \cdots, \beta_{2A-1,v}(t) + \beta_{2A,v}(t)\}, \sum_{i=2A+1}^{|\mathscr{A}^{(S)}|} \beta_{i,v}(t) \Big\}$$
$$P^{(t)}_{v,SS} = I - P^{(t)}_{v,SE}$$
$$P^{(t)}_{v,EI} = \text{diag}\{\alpha_{1,v}(t), \cdots, \alpha_{|\mathscr{A}^{(I)}|,v}(t)\}, \quad P^{(t)}_{v,EE} = I - P^{(t)}_{v,EI} \tag{6.17}$$
$$P^{(t)}_{v,IS} = \begin{bmatrix} \overline{P}^{(t)}_{v,IS} \\ \hline \mathbf{0}_{4\times 12} \end{bmatrix}$$
$$P^{(t)}_{v,IR} = \sum_{i=1}^{A} \gamma^{(r)}_{i,v} \tag{6.18}$$
$$P^{(t)}_{v,ID} = \text{diag}\{\gamma^{(d)}_{1,v}(t), \cdots, \gamma^{(d)}_{|\mathscr{A}^{(I)}|,v}(t)\}$$
$$P^{(t)}_{v,II} = I - \sum_{\chi \in \{S,R,D\}} P^{(t)}_{v,I\chi}$$

where

$$\overline{P}^{(t)}_{v,IS} = \begin{bmatrix} & \nu_{1,v}(t) & 0 & \nu_{3,v}(t) & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & \nu_{2,v}(t) & 0 & 0 & \nu_{5,v}(t) & 0 & 0 & 0 & 0 \\ \mathbf{0}_{6\times 3} & 0 & 0 & 0 & \nu_{4,v}(t) & 0 & \nu_{6,v}(t) & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & \nu_{7,v}(t) & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \nu_{8,v}(t) & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \nu_{9,v}(t) \end{bmatrix},$$

and $I - M$ for a rectangular matrix $M \in \mathbb{R}^{n\times m}$ is intended to mean a $n \times n$ matrix where the diagonal elements are 1 minus the row sum of M. The procedure for estimating parameters, which was detailed in Sections 6.2 (and will be detailed further in 6.4), is then used with the multi-strain versions of the ODE and CHMM dynamics.

6.4 Parameter Estimation with the Two Modules

The relationship between the small-scale CHMM model (Section 6.3) and the large-scale compartmental model (Section 6.2) can be illustrated as follows. An individual in community k who has health status `S` at time t is counted in the number $S_k(t)$. When (s)he transitions to health status `E` at time $t+1$, we decrement $S_k(t+1) = S_k(t) - 1$ and increment

$E_k(t+1) = E_k(t) + 1$. This relationship suggests that both count data (e.g., the number of total infections/deaths) and the individual contact-tracing data can be used to estimate the transition rates of the compartmental model.

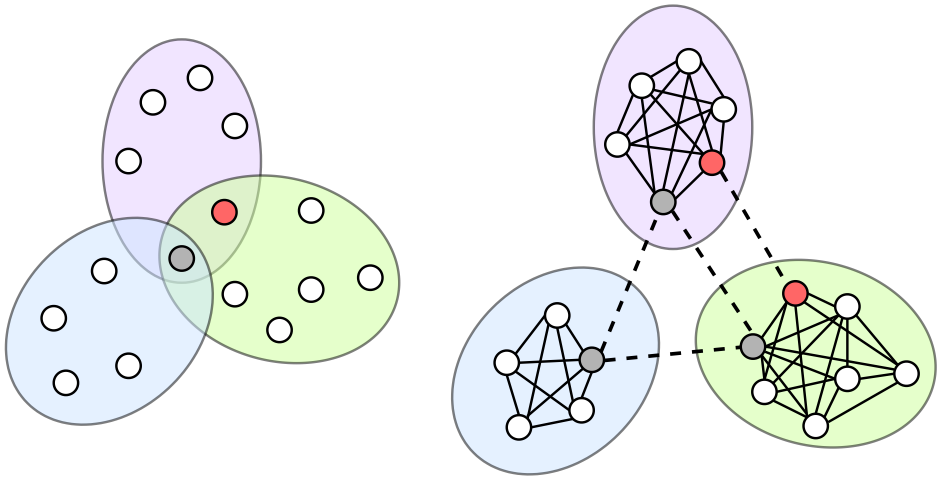

Figure 6.7: [Left] Original local community graph for two individuals (gray and red) who are being tracked. [Right] The corresponding graph representation of the local community graph. Communities are strongly-connected, and nodes which are shared among communities are replicated. Edges between replicated nodes (dashed black lines) between them have weights that depend on the interaction parameters and/or mitigation strategy employed.

Define $\tilde{\theta}_k(t) := [\{\tilde{\beta}_{kj}(t)\}_{j\in\{1,\cdots,K\}}, \tilde{\alpha}_k(t), \tilde{\gamma}_k^{(r)}(t), \tilde{\gamma}_k^{(d)}(t)]^\top$ to be the estimate of parameter vector $\boldsymbol{\theta}_k$ for the specific community $\mathcal{C}_k$, $k \in \{1, \cdots, K\}$, at time $t \in (0, T_{\text{sim}}]$. Similarly, define $\hat{\boldsymbol{\eta}}_n(t) := [\hat{\beta}_n(t), \hat{\alpha}_n(t), \hat{\gamma}_n^{(r)}(t), \hat{\gamma}_n^{(d)}(t)]^\top$ to be the estimate of the parameter vector $\boldsymbol{\eta}_n(t)$ for the tracked subset $\overline{\mathcal{V}} \subseteq \mathcal{V}$ of the total population. Note that even for original parameters which are constant (e.g., α_k, α_n, etc.), we consider a time-varying estimate due to the time-varying nature of the estimated TPM $\hat{P}_n(t)$. Moreover, the transition probability estimates in the small-scale CHMM model and the transition rate estimates in the large-scale compartmental model need to be reconciled. The conversion from any probability p to any rate r for is done through the standard formula $p := 1 - e^{-r}$.

First, by the structure of the estimated TPM $\hat{P}_n(t)$, we can estimate some parameters by matching the corresponding entries: $\hat{\beta}_n(t) = \hat{P}_n(t, \texttt{S}, \texttt{E})$, $\hat{\alpha}_n(t) = \hat{P}_n(t, \texttt{E}, \texttt{I})$, $\hat{\gamma}_n^{(r)}(t) = \hat{P}_n(t, \texttt{I}, \texttt{R})$, $\hat{\gamma}_n^{(d)}(t) = \hat{P}_n(t, \texttt{I}, \texttt{D})$. To estimate the $\{\tilde{\beta}_{kj}(t)\}_j$ parameters, first define $\kappa_n(t)$ to be the number of infectious tracked neighbors of $n \in \overline{\mathcal{C}}_k$ in community $\mathcal{C}_k$. To get an esti-

mate $\hat{\kappa}_n(t)$ of $\kappa_n(t)$, individual n exchanges Viterbi algorithm results with his/her tracked neighbors $\{\hat{x}_{m,j}(t), m \in \mathcal{N}(n)\}$, then simply counts the number of people inferred to be infectious.

Note the probability $\beta_n(t)$ ($\hat{\beta}_n(t)$) for individual n to transfer from `S` to `E` is dependent on $\kappa_n(t)$ ($\hat{\kappa}_n(t)$). A sensible choice for $\beta_n(t)$ should meet two criteria: 1) it should be impossible for n to become exposed if $\kappa_n(t) = 0$, and 2) it should be a function which increases with $\kappa_n(t)$. One such choice, from [48] is a modified Beta distribution; here, we use the straightforward probabilistic relationship:

$$\hat{P}(t, \mathtt{S}, \mathtt{E}) = \hat{\beta}_n(t) = 1 - \left(1 - \hat{\beta}_n^{(1)}(t)\right)^{\hat{\kappa}_n(t)}. \tag{6.19}$$

Here $\beta_n^{(1)} \in [0, 1]$ is the constant probability of a susceptible person becoming exposed upon contact with an infectious individual, and $\hat{\beta}_n^{(1)}(t)$ is its time-varying estimate. We use a superscript 1 to emphasize that the probability corresponds to a *single* susceptible individual and a *single* infectious individual. Because $\kappa_n(t) = 0$ implies $\beta_n(t) = 0$, we specifically consider the case where $\kappa_n(t) > 0$, from which $\hat{\beta}_n^{(1)}(t)$ is determined by algebraic rearrangement of (6.19). Once $\hat{\beta}_n^{(1)}(t)$ is determined for all tracked n, we estimate the $\{\tilde{\beta}_{kj}(t)\}_j$ for each k via

$$\tilde{\beta}_{kk}(t) = -\sum_{n \in \overline{\mathcal{C}}_k} \ln(1 - \hat{\beta}_n^{(1)}(t)), \quad \tilde{\beta}_{kj}(t) = -\sum_{n \in \overline{\mathcal{C}}_k \cap \overline{\mathcal{C}}_j} \ln(1 - \hat{\beta}_n^{(1)}(t)), \tag{6.20}$$

where $k, j \in \{1, \cdots, K\}$ and $\overline{\mathcal{C}}_k$ is the set of tracked individuals in community $\mathcal{C}_k$. Similar to the $\hat{\beta}$'s, we rely entirely on contact-tracing data from individuals in $\overline{\mathcal{C}}_k$ to estimate $\tilde{\alpha}_k(t)$:

$$\tilde{\alpha}_k(t) := -\sum_{n \in \overline{\mathcal{C}}_k} \ln(1 - \hat{\alpha}_n(t)) = -\sum_{n \in \overline{\mathcal{C}}_k} \ln(1 - \hat{P}_n(t, \mathtt{E}, \mathtt{I})). \tag{6.21}$$

Note that both the $\{\tilde{\beta}_{kj}(t)\}_j$ and $\tilde{\alpha}_k(t)$ were estimated using only individual contact-tracing data because the lack of visible symptoms in exposed individuals makes it difficult to obtain the counts $E_k(t)$ for each k. In contrast, the ill, recovered, and death tolls have been some of the most frequently reported statistics throughout the worst of the COVID-19 pandemic, and so we can obtain additional estimates from the large-scale count data. The time cycle period for each rate in the compartmental model is one day:

$$\tilde{\gamma}_k^{(r)}(t) := \frac{1}{2}\left(\frac{1}{I_k(t)}(R_k(t+1) - R_k(t)) - \sum_{n \in \overline{\mathcal{C}}_k} \ln(1 - \hat{\gamma}_n^{(r)}(t))\right),$$

$$\tilde{\gamma}_k^{(d)}(t) := \frac{1}{2}\left(\frac{1}{I_k(t)}(D_k(t+1) - D_k(t)) - \sum_{n\in\bar{\mathcal{C}}_k} \ln(1-\hat{\gamma}_n^{(d)}(t))\right),$$

for time $t \in [0, T_{\text{sim}})$.

Remark 26. The multiscale model can be reduced to either a full compartmental ODE or a full coupled HMM, models which are prevalent throughout much of the existing literature on epidemic modeling (see the beginning of the chapter for a review). In particular, the multiscale model can be reduced to a full CHMM model by using the Markov chain dynamics: $X_n(t+1) = x$ with probability $P_n^{(t)}[X_n(t), x]$ for some $x \in \mathcal{X}$; the number of individuals per compartment at each time t can then be obtained by counting the number of individuals which belong in each phase, e.g., $S_k(t) = \sum_{n\in\mathcal{C}_k} \mathbb{1}\{X_n(t) = \texttt{S}\}$. If the CHMM module replaces the ODE module, a means of propagating individuals across each phase of COVID-19, the multiscale model becomes fully stochastic, and further details about intra-community interactions may effectively remove the need to introduce the community structure. However, the key point is that when handling large-scale populations, individual-level modeling is not ideal due to computation time incurred from the large number of parameters that need to be estimated. Using coarser community partitions via the ODE module allows us to abstract away an appropriate number of details.

6.5 Simulation

We apply our multiscale model to specific case studies pertaining to the ongoing pandemic of COVID-19 and demonstrate its performance via numerical simulations. The common assumptions used for each experiment are described below.

Performance Metrics: We consider the following metrics to evaluate the performance of our multiscale model on the datasets described above. Specific to the CHMM module, we look at two metrics. First, the *binary correlation between the predicted and true sequence of states*, i.e., the ratio of correctly-inferred states determined by the multi-observation Viterbi algorithm (see Section 6.3.1) in comparison to the true health states of the individual:

$$\sum_{t=1}^{T_{\text{sim}}} \mathbb{1}\{\hat{x}_{n,j}(t) = x_{n,j}(t)\}. \tag{6.22}$$

Second, we investigate the average entrywise absolute-value difference between true and estimated TPMs for each time $t \in [0, T_{\text{sim}}]$:

$$\frac{1}{|\mathcal{X}|^2}\sum_{x,z} |P_n(t,x,z) - \hat{P}_n^{(t)}(x,z)|. \tag{6.23}$$

We choose the metric (6.23) in this way, as opposed to the standard $\mathcal{L}_1$ or $\mathcal{L}_2$ norms, so that any deviations away from the true matrix caused by a massive error in a single entry do not greatly impact the overall error.

For parameter estimation of both communities and individuals, we consider mean absolute-value differences:

$$\frac{1}{K}\sum_{k=1}^{K}|\hat{\chi}^{(t)}-\chi|,\ \chi\in\{\alpha_k,\gamma_k^{(r)},\gamma_k^{(d)}\},\quad \frac{1}{\overline{N}}\sum_{n=1}^{\overline{N}}|\hat{\chi}^{(t)}-\chi|,\ \chi\in\{\beta_n(t),\alpha_n,\gamma_n^{(r)},\gamma_n^{(d)}\},\quad (6.24)$$

where χ is a placeholder variable for the original parameters in $\boldsymbol{\theta}_k$ or $\boldsymbol{\eta}_n$, and $\overline{N}\leq N$ is the number of tracked individuals in the total population.

Symptoms Observation Matrix: We consider the symptoms of 1) fever/headache/migraine, 2) difficulty breathing/blockage in lungs, 3) sore throat/scratchy throat/coughing, 4) stomach pain/indigestion/diarrhea,5) sneezing/runny nose/itchy nose, and 6) dead based on the real symptoms observed of a person infected with COVID-19. We assign the following concrete probability values, which were chosen based on the real-data statistics about the symptoms given by the CDC [29].

$$O_n(y_{n,j}=1|x)=\begin{bmatrix}0.1 & 0.1 & 0.9 & 0 & 0\\ 0.05 & 0.05 & 0.65 & 0.01 & 0\\ 0.07 & 0.07 & 0.73 & 0.01 & 0\\ 0 & 0 & 0.03 & 0.01 & 0\\ 0 & 0.9 & 0.95 & 0 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}\in\mathbb{R}^{B\times|\mathcal{X}|}.$$

That is, the (j,x)th entry of the matrix above defines the probability of observing symptom $j\in\{1,\cdots,B\}$ from individual n (i.e., $y_{n,j}=1$) given his/her current health status is $x\in\mathcal{X}$.

Prediction and Forecasting: We substitute the estimated parameters

$$\tilde{\boldsymbol{\theta}}_k(T_{\text{sim}}):=[\{\tilde{\beta}_{kj}(T_{\text{sim}})\}_{j\in\{1,\cdots,K\}},\tilde{\alpha}_k(T_{\text{sim}}),\tilde{\gamma}_k^{(r)}(T_{\text{sim}}),\tilde{\gamma}_k^{(d)}(T_{\text{sim}})]$$

obtained from Section 6.4 into the compartmental module (6.2) in order to forecast the evolution of counts up to some future time $T_{\text{sim}}^{+}>T_{\text{sim}}$. For the purposes of prediction, we keep the rates constant at the final value estimate at time T_{sim}. To determine the accuracy of the prediction, we consider the absolute-value difference between the true and predicted

counts $X(t)$ over time $t \in (T_{\text{sim}}, T_{\text{sim}}^{+}]$, where X is a placeholder for one of the original compartments $X \in \{I, R, D\}$. Note that the susceptible and exposed individuals are not considered since they are difficult to obtain the true values of in the real world.

6.5.1 Performance of HMM Model: Parameter Estimation

We evaluate the performance of the parameter estimation scheme from Sections 6.3.1 and 6.4 using two datasets, described below.

Dataset 1: We construct a dataset based off of [43], which is a collection of real-world contact-tracing time series data about the spread of COVID-19 in South Korea during 2020. The communities corresponding to the SEIRD compartmental module from Section 6.2 are constructed based off of Korea's provinces, so there is only $M = 1$ type of community for this dataset. Individuals within each province are assumed to interact according to a strongly-connected graph, while the edge weights between each pair of provinces are designed based off passenger traffic data collected in 2018 from KOSIS [84]. The total number of individuals considered in this dataset is approximately 5000, and each individual $v \in \mathcal{V}$ is modeled as a CHMM with nominal parameter values $\boldsymbol{\eta}_v(t)$ calculated based on their "age", "symptom-onset-date", "confirmed-date", and "released-date" fields from the original contact-tracing dataset.

Dataset 2: We construct a more artificial dataset in the following way. We choose a specific number of communities K, each of some size N_k, $k \in \{1, \cdots, K\}$ which is randomly chosen from some range. From each community $\mathcal{C}_k$, a subset of members belong to more than one community, and another subset $\overline{\mathcal{C}}_k \subseteq \mathcal{C}_k$ of members individual is chosen to be tracked. Some initial distribution of health statuses across the total population is chosen, and the "true" behavior of disease spread throughout the network is emulated by propagating the CHMM of every individual (tracked or not). This generates a sample path of compartment counts over time per community, as well as health status sequences and observed symptoms over time for each tracked individual in the population; this sample path of values is precisely our dataset. In one of our experiments, we consider different parameters (e.g., community structures, proportion of population which is tracked) in order to demonstrate their effects on the forecasting performance of our multiscale model.

We use a network with $K = 10$ communities and $N = 116$ individuals in the population. The communities are roughly equally-partitioned: $[15, 21, 19, 26, 18, 18, 20, 24, 14, 20]$ and there

Code Hyperparam	Definition
`K`	Num communities
`community_size_range`	[min, max] number of members per community
`multi_communities_prop`	Prop. of total population in multiple communities
`range_simult_communities`	[min, max] num communities one person can be in
`tracked_prop`	Prop. of people in each community tracked

Table 6.1: Possible simulation parameters to adjust in the code (Dataset 2).

	Home	Work	School
α	$0.8 \times 4/14$	$0.8 \times 4/14$	$0.8 \times 4/14$
$\gamma^{(r)}$	$0.979/14$	$0.979/7$	$0.979/7$
$\gamma^{(d)}$	$0.021/7$	$0.021/14$	$0.021/14$

Table 6.2: True parameter values for each type of community in Dataset 2.

are a total of 55 tracked individuals. The results can be observed in Figure 6.8 and Figure 6.9.

6.5.2 Impact of Network Topology on Virus Propagation

In this simulation, we apply the full multiscale model described in Sections 6.2 to 6.4 to Dataset 2. We especially determine the effect of the population network structure on the speed and breadth of the virus spread. The number of communities is chosen at random and interconnected according to various topologies:

1. *Dense Graph*: the number of edges in each community is between 60% and 80% among all possible edges in the community. The number of edges between any two communities is between 1% and 2% among all possible edges. More specifically, the three communities have 564, 1044, and 1274 many edges respectively;

2. *Sparse Graph*: the number of edges in each community is between 1% and 2% among all possible edges in the community. The number of edges between any two communities is between 1% and 2% among all possible edges;

3. *Tree with Multiple Branches*: each community is a tree, and the number of branches for each node is between 3 and 6. This corresponds to a structure which is somewhere

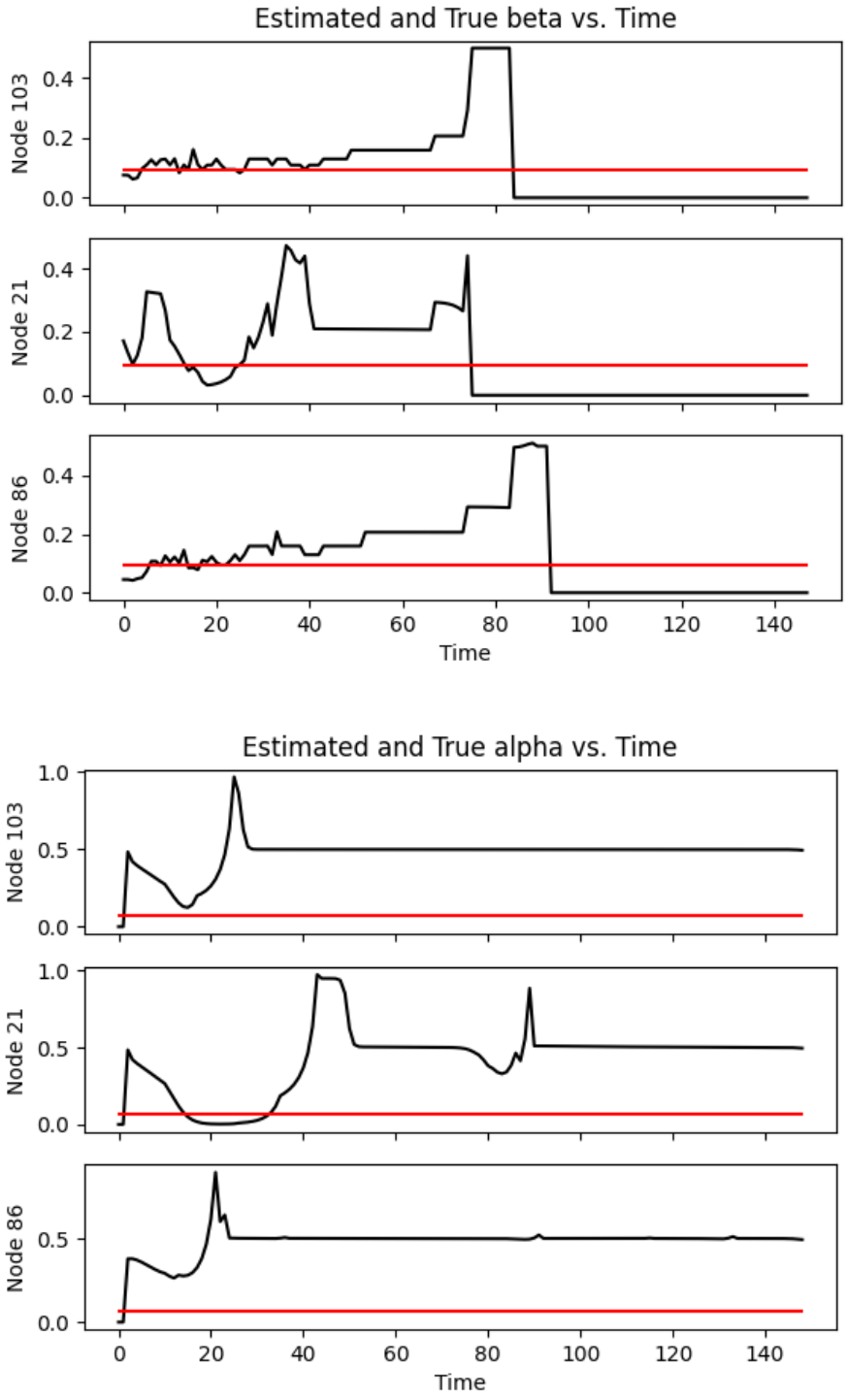

Figure 6.8: The estimated parameters $\hat{\beta}_n^{(1)}(t)$ and $\hat{\alpha}_n(t)$ over time, with constant red line denoting the true value.

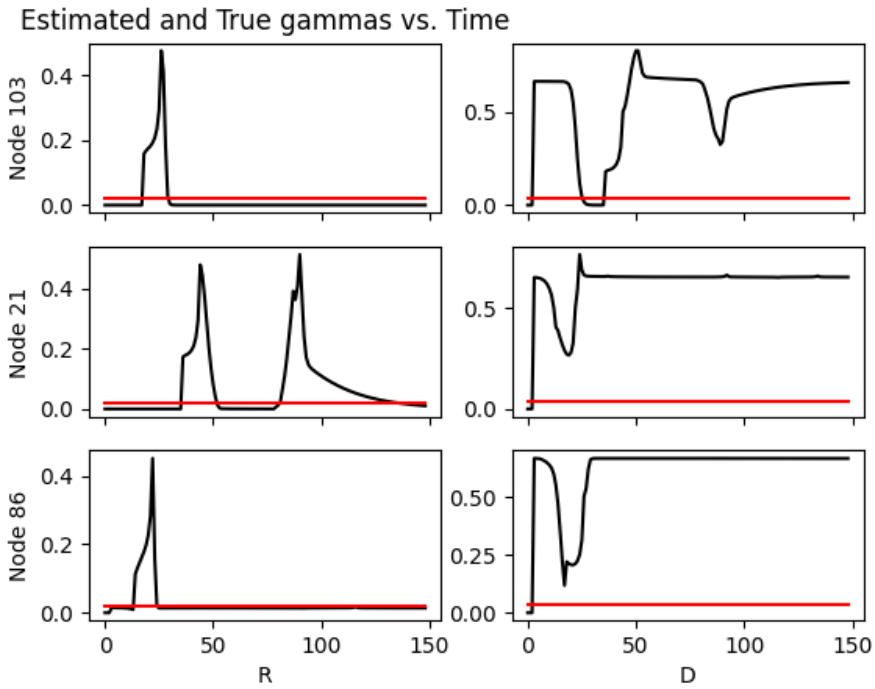

Figure 6.9: The estimated parameters $\hat{\gamma}_n^{(r)}(t)$ and $\hat{\gamma}_n^{(d)}(t)$ over time, with constant red line denoting the true value.

in between the densities of the Dense graph and the Sparse graph.

Using the parameters estimated via the HMM model in Section 6.5.2, we now consider the propagation of disease at the large scale. After a similar experiment is performed for Dataset 2 and converting the transition probabilities to transition rates, we use the values given in Table 6.2.

We visualize the trajectories of each compartment in Figure 6.10. Even for this smaller network of 200 nodes, the Sparse Graph took approximately an hour to run on a 2.2-GHz Intel Core i7 Macbook Air. This result provides further experimental motivation for why the CHMM module should be used strictly for small-scale parameter estimation; the ODE module is better suited for community-wide propagation.

We verify the hypothesis that reducing the frequency of contacts between people via interventions methods such as quarantine is helpful in preventing the spread of the virus. For a fixed simulation time of $T_{\text{sim}} = 2000$, the pandemic in the Dense Graph continues to spread

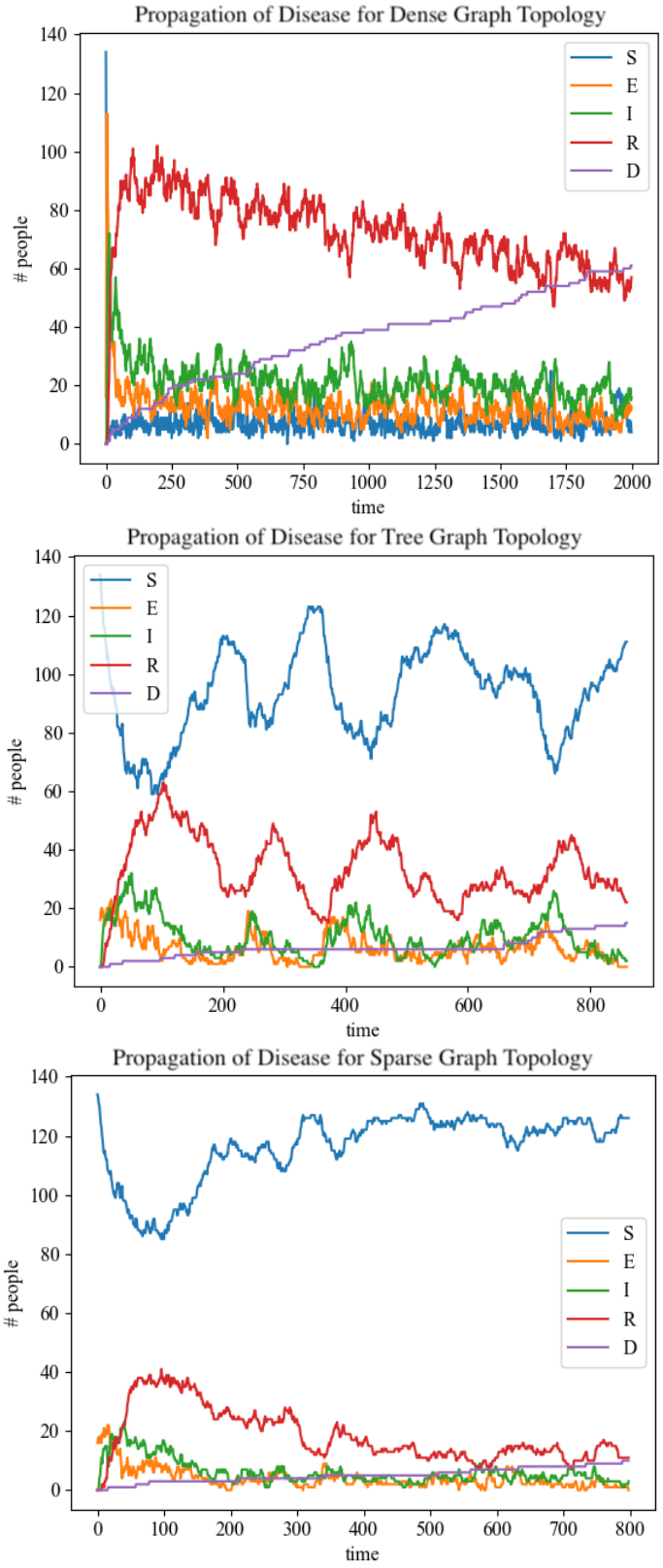

Figure 6.10: Emulation via HMM module of the dense, tree, and sparse graphs.

until all the people have been infected at least once. For the Tree with Multiple Branches, the pandemic ends at time 860 with 146 people infected at least once, and for the Sparse Graph, it ends at time 799 with only 116 people infected at least once.

6.5.3 The Effects of Superspreaders

In this experiment, we emulate the effect of superspreaders by accounting for high variability in the contact rate of individuals by applying the stochastic model (6.4) to Dataset 1. For the sake of graph simplicity, we reduce the original 23 provinces in the dataset down to two groups: Group 1, corresponding to highly popular locations, contains about 400 members total, and Group 2 contains the remaining 915 individuals. The $[\beta_{kj}]$ matrix, substituting (6.3) for the interactions of people, are given by:

$$[\beta_{kj}] := \begin{bmatrix} \beta_0 + \sigma W(t) + \xi N(t) & 0.0021 \\ 0.01 & 0.005 \end{bmatrix}_{(k,j)},$$

where $\beta_0 = 0.03, \sigma = 0.01, \lambda = 0.0143$, and $\xi_i \sim \text{Unif}[0.5, 2]$. The values of α_k, γ_k are determined by what was estimated previously from the HMM experiment of Section 6.5.1.

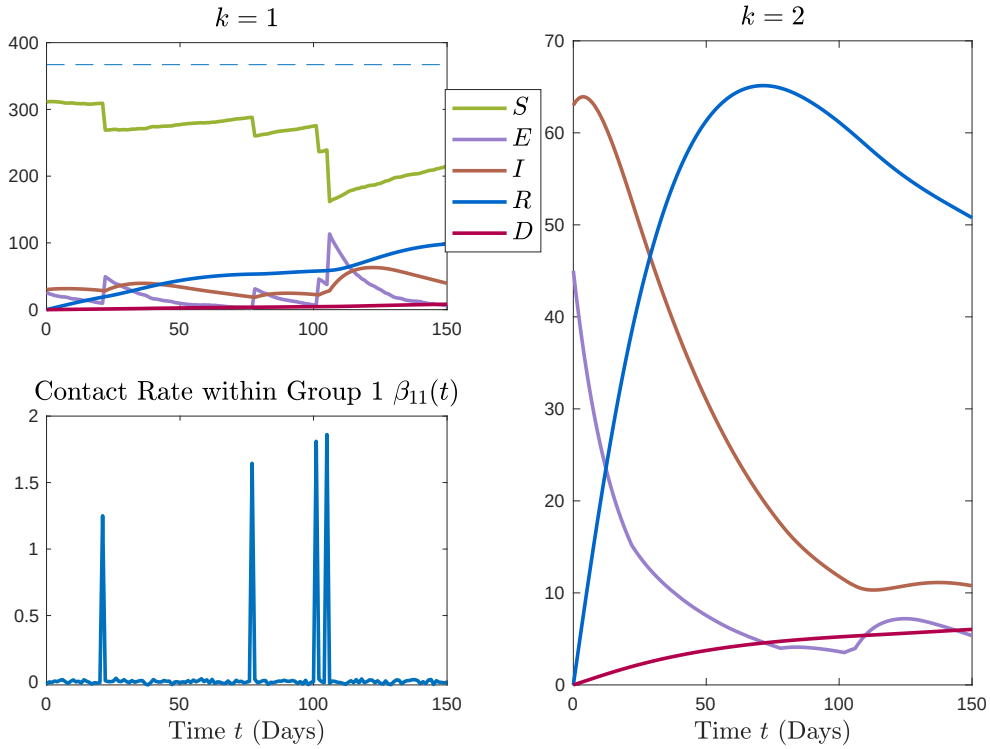

Figure 6.11: The spread of virus visualized over the five compartments for two disjoint groups. The rate of interactions in the Group 1 is subjected to stochastic noise.

The trajectories of both groups over time are shown in Figure 6.11. The initial number of people in each compartment are chosen to be $I_1(0) = 26, E_1(0) = 30$ for Group 1, and $I_2(0) = 45, E_2(0) = 63$ for Group 2. For both groups, the remaining individuals are all susceptible. We observe that Group 2 initially enjoys a reduction in the number of infectious individuals for the two months, but increases as a result of coming into contact with members of Group 1, despite the fact that interactions outside of Group 1 are kept at far less than the interactions within Group 1. This experiment verifies an intuitive hypothesis: as a result of the fragile stability, the pandemic ends more slowly than the time it would have taken without the emergence of superspreaders.

6.6 Conclusion

In this chapter, we addressed the problem of modeling epidemic spread for control by proposing a novel multiscale model consisting of two main parts: 1) a compartmental model which separates among population communities and emulates large-scale disease using ODE dynamics, and 2) a coupled HMM model which is embedded within each individual to emulate the evolution of an his/her health status over time. The novelty in our proposed technique is the combination and extensions of two models which are traditionally used separately in the study of epidemics. Furthermore, the combination of two scales of epidemic modeling allows for the incorporation of multiple scales of data, such as large-scale infected/death counts and individual-level contact-tracing. Model parameters, i.e., transition rates, transition probabilities, and edge weights, are estimated using extensions of standard methods of parameter estimation (e.g., Viterbi's algorithm, Baum-Welch), an the trained model is then used to forecast predictions on the spread of SARS-CoV-2. In particular, we have made additional stochastic extensions to our model by accounting for jump phenomena in two ways: 1) the effect of possible superspreaders via Poisson shot noise and 2) the emergence of variant viruses via jump-switching to different parameters. The implementation of the model is demonstrated on two datasets which were constructed based on real data from at least three different countries.

CONCLUSION AND FUTURE WORK

7.1 Conclusion and Thesis Contributions

In this thesis, we used the broad class of *jump stochastic systems (JSSs)*, i.e., systems with random and repetitive jump phenomena, to demonstrate that tools and theory from mathematics can be leveraged to better balance the interplay between model-based and model-free learning for efficient stochastic control and estimation. Although it is becoming increasingly popular to rely entirely on unsupervised AI methods for controlling/estimating complex stochastic systems, training them can often be exhaustive in computation time and energy if they redundantly learn information that could be more easily obtained from structured models. A key aim of this thesis was to demonstrate that learning recurrent patterns in the underlying jump process can make controller and observer design of JSSs more efficient in data and time by reducing redundant computation.

Our contributions in this thesis were two-fold: in theory and in application. On the theory side, two main examples of JSSs were studied: systems perturbed by Poisson shot noise and Markovian jump systems (MJSs). We began by characterizing sufficient conditions for stochastic incremental stability for both Poisson shot and Lévy noise systems. This work explicitly leveraged theory from Poisson random measures and Lévy processes to develop these conditions, which showed that understanding the relationship between system stability and the jump noise characteristics enabled the potential to more efficiently design controllers and observers for JSSs. We then presented a controller architecture based on the principle of *pattern-learning for prediction (PLP)* for discrete-time/discrete-event systems, where recurrent *patterns* are learned and memorized to minimize redundant computation of control policies. The key aspect of PLP in our controller architecture was the usage of martingales to derive closed-form expressions for two pattern-occurrence quantities: 1) the *expected minimum occurrence time* of any pattern from a collection of patterns, and 2) the *first-occurrence probability* of a pattern being the first to occur among the collection. The pattern-occurrence quantities allowed us to schedule control policies for future patterns, which further reduced redundancy.

On the application side, we highlighted three real-world network control problems from a diversity of fields to show that control and observer design can exploit the natural patterns that emerge from spatial structures and temporal repetitions present in the network. Our first application was a concrete implementation of the PLP controller architecture for dynamic-topology networks, specifically, a power grid whose topology changed due to downed lines over time. There, PLP was integrated with model predictive control (MPC) and the novel system level synthesis (SLS) framework for disturbance-rejection, then compared against variations such as SLS which was robust to changing topologies. We showed that the controller with PLP was able to achieve three things: 1) match the control effort cost of the SLS baseline, 2) stabilize the network just as well as the topology-robust extension, and 3) achieve runtime faster than either. Our second application was the congestion control of vehicle traffic flow over metropolitan intersection networks via an architecture based on *pattern-learning with memory and prediction (PLMP)*. Here, PLMP was an extension of PLP with an explicit implementation of a memory component based on a version of episodic control that built equivalence classes to group together patterns that can be controlled using the same light signals. We demonstrated the performance advantage of PLMP with respect to several congestion metrics and traffic scenarios, and found that on average, it outperformed the baseline controllers with mild variation among the different implementations. Our third application was the estimation and forecasting for the epidemic spread across population networks by using a multiscale combined compartmental ODE/ hidden Markov model. Jump phenomena were regularly present in the spread of epidemics through the chronic emergence of new virus mutations, which can be represented as a jump to a set of dynamics with different parameter values, and superspreader effects, which can be modeled with impulsive shot noise.

Overall, our research demonstrated the performance advantages of learning patterns for less redundant computation by memorizing past patterns to schedule future policies in advance, which made the overall procedure more efficient in data-consumption and computation time.

7.2 Future Work

Continuous-State Dynamics and Inexact Patterns

The impact of Chapter 2 is that understanding the relationship between system stability and the characteristics of the jump noise allows us to infer the types of model-based controllers we need to design. However, if the system is complex enough, model-based techniques may also be intensive in computation time. For example, the *Hamilton-Jacobi-Bellman (HJB) equation* is a well-known method for optimal control in deterministic settings with determin-

istic disturbances [66, 13]. *Impulse control* is a HJB-based method developed for stochastic systems similar to the shot and Lévy noise dynamics described in Chapter 2; extensive discussion on impulse control has been developed in [115] with applications to finance. Yet a well-known limitation of HJB approaches in many scenarios is the large amount of computation time and the inability to analytically solve for the value function from the partial differential equation (PDE). Our implementation and applications of the PLP architecture throughout this thesis have considered discrete values and depend on exact matching between patterns that are observed and patterns that are stored in memory. This does not fully cover cases where the dynamics behave as in Chapter 2, where the possible values the jump could take is a continuous spectrum instead of a discrete set. One way to bypass the scalability to larger pattern spaces is to project multiple patterns into the same equivalence class, similar to the PLMP architecture for the vehicle traffic problem (see Chapter 5). A future point of work would be to design a procedure which constructs a similarity metric in the pattern space that can be used to reduce the dimensionality of the problem prior to control. A relevant branch of literature that could be consulted for this direction is *feature extraction* in machine learning.

Uncertainty Quantification

The current PLP architecture described in Chapter 3 is just one specific type of implementation among a wide variety of controller frameworks that build upon the concept of learning patterns to reduce time, data consumption, and redundant computation. Typical schemes using *Bayesian updates* (e.g., Kalman filtering) encode prior knowledge as a probability distribution. While methodically different to the idea of the pattern-occurrence problem from Chapter 3, which adopts a more frequentist approach by using martingales to construct closed-form formulas, both ideas are similar in that they avoid redundant computation in handling recurring data. Thus, a natural topic for the future is developing alternative implementations of the pattern-learning component, especially Bayesian approaches to solving the pattern-occurrence problems. One extension is to predict the occurrence of future patterns with more state-of-the-art approaches in *uncertainty quantification*, especially when some statistics of the dynamics are unknown (e.g., the transition probability matrix of the MJS in Chapter 4). In addition, a combination of the two approaches can be considered: instead of relying entirely on Bayesian means, explicitly learning specific patterns and keeping track of a table (such as the episodic control approach used in Chapter 5) may potentially provide further benefits depending on the application setting.

Other Real-World Applications

The applications considered throughout this thesis have involved extensive software simulations with a number of simplifying assumptions. A broad subject of future work will involve incorporating domain-specific knowledge into the algorithms designed here, as well as a focus on actual deployment to real-world systems. We also aim to design more learning-based stochastic controller and estimation architectures around these systematic model-based/model-free tradeoffs for other applications, including wireless communication, the internet and datacenter resource allocation, biological networks and neuroscience, and in air traffic management, which can be viewed as a 3D extension to the vehicle traffic problem from Chapter 5. A poignant part of future research would be devoted to providing more explicit theoretical guarantees on performance based on these tradeoffs.

[1] David J. Aldous and Geoffrey K. Eagleson. "On Mixing and Stability of Limit Theorems". In: *The Annals of Probability* 6.2 (1978), pp. 325–331.

[2] Angelo Alessandri, Marco Baglietto, and Giorgio Battistelli. "Receding-Horizon Estimation for Switching Discrete-Time Linear Systems". In: *IEEE Transactions on Automatic Control* 50 (Dec. 2005), pp. 1736–1748.

[3] Carmen Amo Alonso, Fengjun Yang, and Nikolai Matni. "Data-Driven Distributed and Localized Model Predictive Control". In: *IEEE Open Journal of Control Systems* 1 (2022), pp. 29–40.

[4] Zahra Aminzare and Eduardo. D. Sontag. "Contraction Methods for Nonlinear Systems: A Brief Introduction and Some Open Problems". In: *53rd IEEE Conference on Decision and Control*. 2014, pp. 3835–3847.

[5] James Anderson, John C. Doyle, Steven H. Low, and Nikolai Matni. "System Level Synthesis". In: *Annual Reviews in Control* 47 (2019), pp. 364–393. ISSN: 1367-5788.

[6] David Angeli. "A Lyapunov Approach to Incremental Stability Properties". In: *IEEE Transactions on Automatic Control* 47.3 (Mar. 2002), pp. 410–421.

[7] Carles Anton-Haro, José A. R. Fonollosa, Claudi Fauli, and Javier R. Fonollosa. "On the Inclusion of Channel's Time Dependence in a Hidden Markov Model for Blind Channel Estimation". In: *IEEE Transactions on Vehicular Technology* 50.3 (2001), pp. 867–873.

[8] David Applebaum. *Lévy Processes and Stochastic Calculus*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2009.

[9] David Applebaum and Michailina Siakalli. "Asymptotic Stability of Stochastic Differential Equations Driven by Lévy Noise". In: *Journal of Applied Probability* 46 (Dec. 2009), pp. 1116–1129.

[10] Francois Baccelli and Bartlomiej Blaszczyszyn. *Stochastic Geometry and Wireless Networks, Part I: Theory*. Hanover, MA, USA: Now Publishers Inc., 2009.

[11] Marco Baglietto, Giorgio Battistelli, and Luca Scardovi. "Active Mode Observability of Switching Linear Systems". In: *Automatica* 43.8 (2007), pp. 1442–1449.

[12] Saptarshi Bandyopadhyay and Soon-Jo Chung. "Distributed Bayesian Filtering using Logarithmic Opinion Pool for Dynamic Sensor Networks". In: *Automatica* 97 (Nov. 2018), pp. 7–17.

[13] Somil Bansal, Mo Chen, Jaime Fisac, and Claire Tomlin. "Safe Sequential Path Planning of Multi-Vehicle Systems Under Presence of Disturbances and Imperfect Information". In: *American Control Conference*. May 2017.

[14] Richard F. Bass. "Jump Processes". In: (2014).

[15] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinıcius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. "Relational Inductive Biases, Deep Learning, and Graph Networks". In: *ArXiv preprint, arXiv:1806.01261* (Apr. 2018).

[16] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". In: *Ann. Math. Statist.* 41.1 (Feb. 1970), pp. 164–171.

[17] Dennis S. Bernstein and Wassim M. Haddad. "LQG control with an H_∞ performance bound: a Riccati equation approach". In: *1988 American Control Conference*. June 1988, pp. 796–802.

[18] Patrick Billingsley. *Convergence of Probability Measures*. Second. Wiley Series in Probability and Statistics: Probability and Statistics. New York: John Wiley & Sons Inc., 1999.

[19] Vincent Blondel, G. Campion, and Michel Gevers. "A Sufficient Condition for Simultaneous Stabilization". In: *IEEE Transactions on Automatic Control* 38.8 (1993), pp. 1264–1266.

[20] Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. "Model-Free Episodic Control". In: *ArXiv preprint, arXiv:1606.04460* (June 2016).

[21] Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. "Reinforcement Learning, Fast and Slow". In: *Trends in Cognitive Sciences* 23.5 (2019), pp. 408–422.

[22] Matthew Brand, Nuria Oliver, and Alex Pentland. "Coupled Hidden Markov Models for Complex Action Recognition". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (Jan. 1997).

[23] Yongli Cai, Jianjun Jiao, Zhanji Gui, Yuting Liu, and Weiming Wang. "Environmental Variability in a Stochastic Epidemic Model". In: *Appl. Math. Comput.* 329.C (July 2018), pp. 210–226.

[24] Yongli Cai, Yun Kang, Malay Banerjee, and Weiming Wang. "A Stochastic SIRS Epidemic Model With Infectious Force Under Intervention Strategies". In: *Journal of Differential Equations* 259 (Dec. 2015), pp. 7463–7502.

[25] Ewen Callaway. *Are COVID Surges Becoming More Predictable? New Omicron Variants Offer a Hint.* URL: https://www.nature.com/articles/d41586-022-01240-x.

[26] Yong-Yan Cao, You-Xian Sun, and J. Lam. "Simultaneous Stabilization via Static Output Feedback and State Feedback". In: *IEEE Transactions on Automatic Control* 44.6 (1999), pp. 1277–1282.

[27] Vincenzo Capasso and Gabriella Serio. "A Generalization of the Kermack-McKendrick Deterministic Epidemic Model". In: *Mathematical Biosciences* 42.1 (Nov. 1978), pp. 43–61.

[28] Vincenzo Capuano, Alexei Harvard, Yvette Lin, and Soon-Jo Chung. "DGNSS-Vision Integration for Robust and Accurate Relative Spacecraft Navigation". In: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. Sept. 2019, pp. 2923–2939.

[29] Centers for Disease Control and Prevention, COVID-19 Response. *COVID-19 Case Surveillance Public Data Access, Summary, and Limitations.*

[30] Chiranjib Chakraborty, Manojit Bhattacharya, and Ashish Ranjan Sharma. "Present Variants of Concern and Variants of Interest of Severe Acute Respiratory Syndrome Coronavirus 2: Their Significant Mutations in S-Glycoprotein, Infectivity, Re-infectivity, Immune Escape and Vaccines Activity". In: *Reviews in Medical Virology* 32.2 (2022), e2270.

[31] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan L.H. Andrew. "Online Convex Optimization Using Predictions". In: *SIGMETRICS Perform. Eval. Rev.* 43.1 (June 2015), pp. 191–204.

[32] Soon-Jo Chung, Saptarshi Bandyopadhyay, Insu Chang, and Fred Y. Hadaegh. "Phase Synchronization Control of Complex Networks of Lagrangian Systems on Adaptive Digraphs". In: *Automatica* 49.5 (May 2013), pp. 1148–1161.

[33] Soon-Jo Chung and Jean-Jacques E. Slotine. "Cooperative Robot Control and Concurrent Synchronization of Lagrangian Systems". In: *IEEE Transactions on Robotics* 25.3 (June 2009), pp. 686–700.

[34] Brian J. Coburn, Bradley G. Wagner, and Sally M. Blower. "Modeling Influenza Epidemics and Pandemics: Insights into the Future of Swine Flu (H1N1)". In: *BMC Med.* 7.30 (June 2009).

[35] Vittoria Colizza, Alain Barrat, Marc Barthelemy, Alain-Jacques Valleron, and Alessandro Vespignani. "Modeling the Worldwide Spread of Pandemic Influenza: Baseline Case and Containment Interventions". In: *PLoS Med.* 4.1 (Jan. 2007).

[36] Samuel Coogan, Ebru Aydin Gol, Murat Arcak, and Calin Belta. "Controlling a Network of Signalized Intersections from Temporal Logical Specifications". In: *2015 American Control Conference (ACC)*. 2015, pp. 3919–3924. DOI: 10.1109/ACC.2015.7171941.

[37] Oswaldo Luiz do Valle Costa, Marcelo D. Fragoso, and Marcos Garcia Todorov. "A Detector-Based Approach for the H_2 Control of Markov Jump Linear Systems With Partial Information". In: *IEEE Transactions on Automatic Control* 60.5 (2015), pp. 1219–1234.

[38] David R. Cox. *Renewal Theory*. London: John Wiley & Sons Inc., 1962.

[39] David R. Cox and Hilton D. Miller. *The Theory of Stochastic Processes*. Taylor & Francis Group, 1965.

[40] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting". In: *IEEE Transactions on Intelligent Transportation Systems* 21.11 (2020), pp. 4883–4894.

[41] Francesco A. Cuzzola, Jose C. Geromel, and Manfred Morari. "An Improved Approach for Constrained Robust Model Predictive Control". In: *Automatica* 38.7 (2002), pp. 1183–1189.

[42] Ashwin P. Dani, Soon-Jo Chung, and Seth Hutchinson. "Observer Design for Stochastic Nonlinear Systems via Contraction-Based Incremental Stability". In: *IEEE Transactions on Automatic Control* 60.3 (Mar. 2015), pp. 700–714.

[43] *Data Science for COVID-19 in South Korea*. URL: https://www.kaggle.com/kimjihoo/coronavirusdataset?select=PatientInfo.csv.

[44] Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. "Safely Learning to Control the Constrained Linear Quadratic Regulator". In: *2019 American Control Conference (ACC)*. 2019, pp. 5582–5588.

[45] Marc P. Deisenroth, Dieter Fox, and Carl E. Rasmussen. "Gaussian Processes for Data-Efficient Learning in Robotics and Control". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.2 (Feb. 2015), pp. 408–423.

[46] Frank Diamond. *Asymptomatic Carriers of COVID-19 Make It Tough to Target*. https://www.infectioncontroltoday.com/covid-19/asymptomatic-carriers-covid-19-make-it-tough-target.

[47] François Dion and Bruce Hellinga. "A Rule-Based Real-Time Traffic Responsive Signal Control System with Transit Priority: Application to an Isolated Intersection". In: *Transportation Research Part B: Methodological* 36.4 (2002), pp. 325–343.

[48] Wen Dong, Alex Sandy Pentland, and Katherine A. Heller. "Graph-Coupled HMMs for Modeling the Spread of Infection". In: *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. UAI'12. AUAI Press, 2012, pp. 227–236.

[49] Joseph L. Doob. *Stochastic Processes*. John Wiley & Sons Inc., 1953.

[50] Michael Dorothy and Soon-Jo Chung. "Switched Systems with Multiple Invariant Sets". In: *Systems & Control Letters* 96 (Oct. 2016), pp. 103–109.

[51] John C. Doyle. "Guaranteed margins for LQG regulators". In: *IEEE Transactions on Automatic Control* 23.4 (1978), pp. 756–757.

[52] P van den Dreessche and James Watmough. "Reproduction Numbers and Sub-threshold Endemic Equilibria for Compartmental Models of Disease Transmission". In: *Mathematical Biosciences* 180 (Nov. 2002), pp. 29–48.

[53] Ulrich Egert, D. Heck, and A. Aertsen. "Two-Dimensional Monitoring of Spiking Networks in Acute Brain Slices. " In: *Experimental Brain Research* 142 (2002), pp. 268–274.

[54] Stephen Eubank, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltán Toroczkai, and Nan Wang. "Modelling Disease Outbreaks in Realistic Urban Social Networks". In: *Nature* 429 (May 2004), pp. 180–184.

[55] David W. Eyre, Donald Taylor, Mark Purver, David Chapman, Tom Fowler, Koen B. Pouwels, A. Sarah Walker, and Tim E.A. Peto. "Effect of Covid-19 Vaccination on

Transmission of Alpha and Delta Variants". In: *New England Journal of Medicine* 386.8 (2022), pp. 744–756.

[56] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. "Revisiting Fundamentals of Experience Replay". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 3061–3071.

[57] William Feller. *An Introduction to Probability Theory and Its Applications*. Vol. 1. Wiley, Jan. 1968.

[58] G. D. Forney. "The Viterbi algorithm". In: *Proceedings of the IEEE* 61.3 (1973), pp. 268–278.

[59] James Gallagher. *Covid Reinfection: Man Gets Covid Twice and Second Hit 'More Severe'*. https://www.bbc.com/news/health-54512034.

[60] Carlos E. García, David M. Prett, and Manfred Morari. "Model predictive control: Theory and practice—A survey". In: *Automatica* 25.3 (1989), pp. 335–348.

[61] Alberto Gómez-Carballa, Jacobo Pardo-Seco, Xabier Bello, Federico Martinón-Torres, and Antonio Salas. "Superspreading in the Emergence of COVID-19 Variants". In: *Trends in Genetics* 37.12 (2021), pp. 1069–1080.

[62] Marta C. González, Hidalgo César A., and Albert-László Barabási. "Understanding Individual Human Mobility Patterns". In: *Nature* 453 (2008), pp. 779–782.

[63] Alison Gray, David Greenhalgh, L. Hu, Xuerong Mao, and Jiafeng Pan. "A Stochastic Differential Equation SIS Epidemic Model". In: *SIAM Journal on Applied Mathematics* 71.3 (2011), pp. 876–902.

[64] Abba Gumel, Shigui Ruan, Troy Day, James Watmough, Fred Brauer, P Driessche, Dave Gabrielson, Chris Bowman, Murray Alexander, Sten Ardal, Jianhong Wu, and Beni Sahai. "Modeling Strategies for Controlling SARS Outbreak". In: *Proceedings. Biological sciences / The Royal Society* 271 (Nov. 2004), pp. 2223–2232.

[65] Bruce Hajek. *Notes for ECE534: An Exploration of Random Processes for Engineers*. 2006. URL: http://www.ifp.uiuc.edu/~hajek/Papers/randomprocJuly06.pdf.

[66] Sylvia L. Herbert, Mo Chen, SooJean Han, Somil Bansal, Jaime F. Fisac, and Claire J. Tomlin. "FaSTrack: A Modular Framework for Fast and Guaranteed Safe Motion Planning". In: *IEEE 56th Annual Conference on Decision and Control*. Dec. 2017, pp. 1517–1522.

[67] João P. Hespanha. "Uniform Stability of Switched Linear Systems: Extensions of LaSalle's Invariance Principle". In: *IEEE Transactions on Automatic Control* 49.4 (Apr. 2004), pp. 470–482.

[68] João P. Hespanha and A. Stephen Morse. "Stability of Switched Systems with Average Dwell-Time". In: *Proceedings of the 38th IEEE Conference on Decision and Control*. Vol. 3. Dec. 1999, pp. 2655–2660.

[69] Herbert W. Hethcote and Simon A. Levin. "Periodicity in Epidemiological Models". In: *Applied Mathematical Ecology* 18.1 (1989), pp. 193–211.

[70] Dimitar Ho, Hoang Le, John Doyle, and Yisong Yue. "Online Robust Control of Nonlinear Systems with Large Uncertainty". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. Proceedings of Machine Learning Research. PMLR, Apr. 2021, pp. 3475–3483.

[71] Nobuyuki Ikeda and Shinzo Watanabe. *Stochastic Differential Equations and Diffusion Processes*. North-Holland mathematical library. Amsterdam: North-Holland, 1989.

[72] Jean Jacod and Albert Shiryaev. *Limit Theorems for Stochastic Processes*. Berlin, Germany: Springer Berlin, 2003.

[73] Monique Jeanblanc. "Jump Processes". In: (Apr. 2007).

[74] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". In: *IEEE Computational Intelligence Magazine* 17.2 (2022), pp. 29–48.

[75] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. "STOMP: Stochastic Trajectory Optimization for Motion Planning". In: *2011 IEEE International Conference on Robotics and Automation*. Aug. 2011, pp. 4569–4574.

[76] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45.

[77] Yuji Kasahara and Keigo Yamada. "Stability Theorem for Stochastic Differential Equations with Jumps". In: *Stochastic Processes and their Applications* 38.1 (1991), pp. 13–32.

[78] Kelly Kasulis. *'Patient 31' and South Korea's Sudden Spike in Coronavirus Cases*. URL: https://www.aljazeera.com/news/2020/03/31-south-korea-sudden-spike-coronavirus-cases-200303065953841.html.

[79] William O. Kermack and Anderson G. McKendrick. "Contributions to the Mathematical Theory of Epidemics—I". In: *Bulletin of Mathematical Biology* 53.1 (1991), pp. 33–55.

[80] Hassan K Khalil. *Nonlinear Systems; 3rd ed.* Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[81] John F. C. Kingman. *Poisson processes*. Vol. 3. NY, USA: Oxford University Press, 1993.

[82] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *Proceedings of the 5th International Conference on Learning Representations* (ICLR). ICLR '17. 2017.

[83] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2011.

[84] *KOrean Statistical Information Service (KOSIS)*. URL: https://kosis.kr/eng/statisticsList/statisticsListIndex.do?menuId=M_01_01&vwcd=MT_ETITLE&parmTabId=M_01_01#content-group.

[85] Nikolay N. Krasovskii and Solomon Lefschetz. *Stability of Motion: Applications of Lyapunov's Second Method to differential systems and equations with delay*. Stanford University Press, 1870.

[86] Hiroshi Kunita. "Itô's Stochastic Calculus: Its Surprising Power for Applications". In: *Stochastic Processes and their Applications* 120 (May 2010), pp. 622–652.

[87] Harold J. Kushner. *Stochastic Stability and Control*. Academic Press, 1967.

[88] Simon Kwoczek, Sergio Di Martino, and Wolfgang Nejdl. "Predicting and Visualizing Traffic Congestion in the Presence of Planned Special Events". In: *Journal of Visual Languages & Computing* 25.6 (2014), pp. 973–980.

[89] Joseph P. LaSalle and Solomon Lefschetz. *Stability by Liapunov's Direct Method with Applications*. Vol. 4. Academic Press New York, 1973.

[90] Günther Last and Mathew Penrose. *Lectures on the Poisson Process*. NY, USA: Cambridge University Press, 2017.

[91] Máté Lengyel and Peter Dayan. "Hippocampal Contributions to Control: The Third Way". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2007, pp. 1–8.

[92] Shuo-Yen Robert Li. "A Martingale Approach to the Study of Occurrence of Sequence Patterns in Repeated Experiments". In: *The Annals of Probability* 8.6 (1980), pp. 1171–1176.

[93] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: *International Conference on Learning Representations*. 2018.

[94] Daniel Liberzon, João P. Hespanha, and A.Stephen Morse. "Stability of Switched Systems: a Lie-Algebraic Condition". In: *Systems & Control Letters* 37.3 (July 1999), pp. 117–122.

[95] Jennie Lioris, Ramtin Pedarsani, Fatma Yildiz Tascikaraoglu, and Pravin Varaiya. "Platoons of connected vehicles can double throughput in urban roads". In: *Transportation Research Part C: Emerging Technologies* 77 (2017), pp. 292–305. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2017.01.023.

[96] Jicheng Liu. "On the Existence and Uniqueness of Solutions to Stochastic Differential Equations of Mixed Brownian and Poissonian Sheet Type". In: *Stochastic Processes and their Applications* 94.2 (Aug. 2001), pp. 339–354.

[97] Winfried Lohmiller and Jean-Jacques E. Slotine. "Contraction Analysis of Nonlinear Distributed Systems". In: *International Journal of Control* 9 (2005), pp. 678–688.

[98] Winfried Lohmiller and Jean-Jacques E. Slotine. "Nonlinear Process Control using Contraction Theory". In: *American Institute of Chemical Engineers Journal* 3 (Mar. 2000), pp. 588–596.

[99] Winfried Lohmiller and Jean-Jacques E. Slotine. "On Contraction Analysis for Nonlinear Systems". In: *Automatica* 34.6 (June 1998), pp. 683–696.

[100] Guangquan Lu, Lumiao Li, Yunpeng Wang, Ran Zhang, Zewen Bao, and Haichong Chen. "A Rule Based Control Algorithm of Connected Vehicles in Uncontrolled Intersection". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2014, pp. 115–120.

[101] Jianbo Lu, Dewei Li, and Yugeng Xi. "Constrained Model Predictive Control Synthesis for Uncertain Discrete-Time Markovian Jump Linear Systems". In: *IET Control Theory & Applications* 7 (5 Mar. 2013), pp. 707–719.

[102] Xuerong Mao. "Exponential Stability for Stochastic Differential Equations with Respect to Semimartingales". In: *Stochastic Processes and their Applications* 35.2 (Aug. 1990), pp. 267–277.

[103] Xuerong Mao. *Stability of stochastic differential equations with respect to semimartingales*. Pitman research notes in mathematics series. Longman Scientific & Technical, 1991.

[104] Nikolai Matni, Yuh-Shyang Wang, and James Anderson. "Scalable System Level Synthesis for Virtually Localizable Systems". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 3473–3480.

[105] Alexandre R. Mesquita and João P. Hespanha. "Jump Control of Probability Densities With Applications to Autonomous Vehicle Motion". In: *IEEE Transactions on Automatic Control* 57.10 (Oct. 2012), pp. 2588–2598.

[106] Alan J. Miller. "A Queueing Model for Road Traffic Flow". In: *Journal of the Royal Statistical Society: Series B* 23.1 (1961), pp. 64–90.

[107] Remi Munos and Paul Bourgine. "Reinforcement Learning for Continuous Stochastic Control Problems". In: *Advances in Neural Information Processing Systems 10*. MIT Press, 1998, pp. 1029–1035.

[108] Ajith Muralidharan, Ramtin Pedarsani, and Pravin Varaiya. "Analysis of Fixed-time Control". In: *ArXiv preprint, arXiv:1408.4229* (2015).

[109] Yashwanth K. Nakka and Soon-Jo Chung. "Trajectory Optimization for Chance-Constrained Nonlinear Stochastic Systems". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. Dec. 2019, pp. 3811–3818.

[110] Yashwanth K. Nakka, Anqi Liu, Guanya Shi, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. "Chance-Constrained Trajectory Optimization for Safe Exploration and Learning of Nonlinear Systems". In: *IEEE Robotics and Automation Letters* 2 (Apr. 2021), pp. 389–396.

[111] Michael O'Connell, Guanya Shi, Xichen Shi, and Soon-Jo Chung. "Meta-Learning-Based Robust Adaptive Flight Control under Uncertain Wind Conditions". In: *ArXiv preprint, arxiv:2103.01932* (May 2022).

[112] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. "Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds". In: *Science Robotics* 7.66 (2022), pp. 1–15.

[113] Sanghoon Oh, Linjun Zhang, Eric Tseng, Wayne Williams, Helen Kourous, and Gabor Orosz. "Safe Decision and Control of Connected Automated Vehicles for an Unprotected Left Turn". In: *Dynamic Systems and Control Conference*. Oct. 2020.

[114] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Berlin, Germany: Springer Berlin, 2003. URL: https://books.google.com/books?id=EQZEAAAAQBAJ.

[115] Bernt Øksendal and Agnès Sulem-Bialobroda. *Applied Stochastic Control of Jump Diffusions*. Universitext. Berlin, Germany: Springer-Verlag, 2007.

[116] Reza Olfati-Saber and Richard M. Murray. "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays". In: *IEEE Transactions on Automatic Control* 49.9 (Sept. 2004), pp. 1520–1533.

[117] Byung-Gun Park and Wook Hyun Kwon. "Robust One-Step Receding Horizon Control of Discrete-Time Markovian Jump Uncertain Systems". In: *Automatica* 38.7 (2002), pp. 1229–1235.

[118] Ashok Patel and Bart Kosko. "Stochastic Resonance in Continuous and Spiking Neuron Models With Lévy Noise". In: *IEEE Transactions on Neural Networks* 19 (Jan. 2009), pp. 1993–2008.

[119] Romain Pepy, Alain Lambert, and Hugues Mounier. "Path Planning using a Dynamic Vehicle Model". In: *2006 2nd International Conference on Information & Communication Technologies*. Vol. 1. 2006, pp. 781–786.

[120] Quang Cuong Pham. "Analysis of Discrete and Hybrid Stochastic Systems by Nonlinear Contraction Theory". In: *International Conference on Control, Automation, Robotics and Vision*. Jan. 2009, pp. 1054–1059.

[121] Quang-Cuong Pham, Nicolas Tabareau, and Jean-Jacques E. Slotine. "A Contraction Theory Approach to Stochastic Incremental Stability". In: *IEEE Transactions on Automatic Control* (2009), pp. 816–820.

[122] Vladimir Pozdnyakov. "On Occurrence of Patterns in Markov Chains: Method of Gambling Teams". In: *Statistics & Probability Letters* 78.16 (2008), pp. 2762–2767.

[123] Vladimir Pozdnyakov and Martin Kulldorff. "Waiting Times for Patterns and a Method of Gambling Teams". In: *The American Mathematical Monthly* 113.2 (2006), pp. 134–143.

[124] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. "Neural Episodic Control". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 2827–2836.

[125] Philip Protter. *Stochastic Integration and Differential Equation*. Second. Berlin, Heidelberg: Springer-Verlag, 1992.

[126] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[127] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. "On Stochastic Optimal Control and Reinforcement Learning by Approximate Inference". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. Aug. 2013, pp. 3052–3056.

[128] Konrad Reif, Stefan Gunther, Engin Yaz, and Rolf Unbehauen. "Stochastic Stability of the Discrete-Time Extended Kalman Filter". In: *IEEE Transactions on Automatic Control* 44.4 (Apr. 1999), pp. 714–728.

[129] Benjamin Rivière and Soon-Jo Chung. "H-TD2: Hybrid Temporal Difference Learning for Adaptive Urban Taxi Dispatch". In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2022), pp. 10935–10944.

[130] Sheldon M. Ross. *A First Course in Probability*. seventh. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2006.

[131] Sheldon M. Ross. *Stochastic Processes*. second. New York, USA: John Wiley & Sons Inc., 1996.

[132] Krzysztof Rusek, José Suárez-Varela, Paul Almasan, Pere Barlet-Ros, and Albert Cabellos-Aparicio. "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN". In: *IEEE Journal on Selected Areas in Communications* 38.10 (2020), pp. 2260–2270.

[133] Iman Saboori and Khashayar Khorasani. "Actuator Fault Accommodation Strategy for a Team of Multi-Agent Systems Subject to Switching Topology". In: *Automatica* 62 (2015), pp. 200–207.

[134] Marcel Salathe, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W. Feldman, and James H. Jones. "A High-Resolution Human Contact Network for Infectious Disease Transmission". In: *Proceedings of the National Academy of Sciences (PNAS)*. Dec. 2010.

[135] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999.

[136] Mathijs Schuurmans and Panagiotis Patrinos. "Data-driven Distributionally Robust Control of Partially Observable Jump Linear Systems". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. 2021, pp. 4332–4337.

[137] Guanya Shi, Wolfgang Hönig, Yisong Yue, and Soon-Jo Chung. "Neural-Swarm: Decentralized Close-Proximity Multirotor Control Using Learned Interactions". In: *IEEE Int. Conf. Robot. Autom.* 2020.

[138] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Eaglewood Cliffs, NJ, USA: Prentice Hall, 1991.

[139] Jean-Jacques E. Slotine, Wei Wang, and Khalid Rifai. "Contraction Analysis of Synchronization in Networks of Nonlinearly Coupled Oscillators". In: *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*. Jan. 2004, pp. 5–8.

[140] Eduardo Sontag. "A Lyapunov-Like Characterization of Asymptotic Controllability". In: *Siam Journal on Control and Optimization* 21.3 (May 1983), pp. 462–471.

[141] Herbert E. Soper. "The Interpretation of Periodicity in Disease Prevalence". In: *Journal of the Royal Statistical Society* 92.1 (1929), pp. 34–73.

[142] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Vittoria Colizza, Lorenzo Isella, Corinne Regis, Jean-François Pinton, Nagham Khanafer, Wouter Van den Broeck, and Philippe Vanhems. "Simulation of an SEIR Infectious Disease Model on the Dynamic Contact Network of Conference Attendees". In: *BMC Medicine* 9.87 (July 2011).

[143] "Symptoms of Coronavirus". In: *Center for Disease Control and Prevention (CDC)* (). URL: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html.

[144] *The 2003 Northeast Blackout : Five Years Later*. https://www.scientificamerican.com/article/2003-blackout-five-years-later/.

[145] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. "A Generalized Path Integral Control Approach to Reinforcement Learning". In: *Journal of Machine Learning Research* 11 (Mar. 2010), pp. 3137–3181.

[146] TSCC. *Reinforcement Learning for Traffic Signal Control: Benchmark Dataset for the Traffic Signal Control Competition (TSCC)*. 2019.

[147] Hiroyasu Tsukamoto, Soon-Jo Chung, and Jean-Jaques E. Slotine. "Contraction Theory for Nonlinear Stability Analysis and Learning-Based Control: A Tutorial Overview". In: *Annual Reviews in Control* 52 (2021), pp. 135–169.

[148] Abhinav Verma, Hoang Le, Yisong Yue, and Swarat Chaudhuri. "Imitation-Projected Programmatic Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[149] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. "Programmatically Interpretable Reinforcement Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 5045–5054.

[150] René Vidal, Alessandro Chiuso, and Stefano Soatto. "Observability and Identifiability of Jump Linear Systems". In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* Vol. 4. 2002, pp. 3614–3619.

[151] Eric A. Wan and Rudolph Van Der Merwe. "The Unscented Kalman Filter for Nonlinear Estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*. Oct. 2000, pp. 153–158.

[152] Yuh-Shyang Wang, Nikolai Matni, and John Doyle. "Separable and Localized System Level Synthesis for Large-Scale Systems". In: *IEEE Transactions on Automatic Control* 63.12 (Dec. 2018), pp. 4234–4249.

[153] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. "A System Level Approach to Controller Synthesis". In: *ArXiv preprint, arXiv:1610.04815* (2016).

[154] Alex Watson. "Introduction to Lévy Processes". In: (Sept. 2016).

[155] Harrison C. White, Scott A. Boorman, and Ronald L. Breiger. "Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions". In: *American Journal of Sociology* 81.4 (1976), pp. 730–780.

[156] Jan C. Willems and Jan Willem Polderman. *Introduction to Mathematical Systems Theory : A Behavorial Approach*. Texts in Applied Mathematics. New York: Springer, 1997.

[157] Lin Xiao, Stephen Boyd, and Sanjay Lall. "A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus". In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* Apr. 2005, pp. 63–70.

[158] Xiaolin Li, M. Parizeau, and R. Plamondon. “Training Hidden Markov Models with Multiple Observations–A Combinatorial Method”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.4 (2000), pp. 371–377.

[159] Feng Xie and David Levinson. “Measuring the Structure of Road Networks”. In: *Geographical Analysis* 39.3 (2007), pp. 336–356.

[160] Anton Xue and Nikolai Matni. “Data-Driven System Level Synthesis”. In: *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. Ed. by Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger. Vol. 144. Proceedings of Machine Learning Research. PMLR, June 2021, pp. 189–200.

[161] Junho Yang, Ashwin Dani, Soon-Jo Chung, and Seth Hutchinson. “Vision-Based Localization and Robot-Centric Mapping in Riverine Environments”. In: *J. Field Robot* 34.3 (2017), pp. 429–450.

[162] Shen Yin, Bing Xiao, Steven X. Ding, and Donghua Zhou. “A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System”. In: *IEEE Transactions on Industrial Electronics* 63.5 (2016), pp. 3311–3320.

[163] Bing Yu, Haoteng Yin, and Zhanxing Zhu. “Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting”. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI’18. Stockholm, Sweden: AAAI Press, 2018, pp. 3634–3640.

[164] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. “The Power of Predictions in Online Control”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NeurIPS’20. Red Hook, NY, USA: Curran Associates Inc., 2020, pp. 1–11.

[165] Kun Zhang and Michael A.P. Taylor. “Effective Arterial Road Incident Detection: A Bayesian Network Based Algorithm”. In: *Transportation Research Part C: Emerging Technologies* 14.6 (2006), pp. 403–417.