Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns

SooJean Han[*b,1], Soon-Jo Chung[b], John C. Doyle[b]

[a]*Division of Engineering and Applied Science, California Institute of Technology, Pasadena, 91125, CA, USA*

Highlights

Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns

SooJean Han[*], Soon-Jo Chung, John C. Doyle

- By considering linear discrete-time Markovian jump systems (MJSs) with unknown mode-switching dynamics, we make concrete the broad notion that controller synthesis can be made more efficient by reducing computation time and redundancy via pattern-learning and prediction (PLP), i.e., learning patterns in the behavior of uncertain stochastic systems, preserving past patterns into memory, and predicting the future pattern occurrences.
- PLP is developed and included in an otherwise straightforward controller framework which leverages well-researched techniques in system identification and control. The construction of this component leverages martingale methods from prior literature, but with two important extensions which make it more suitable for real-world MJS applications: 1) the distribution of the mode process is unknown, and 2) the realization of the mode process over time is not observable.
- We apply our proposed framework to the fault-tolerant control of a network with dynamic topology, and perform an extensive numerical study which compares the performance of a controller with PLP against non-predictive and topology-robust baselines. Our study provides insights into important tradeoffs which emphasize the impact of PLP, e.g., the size of the pattern collection and the system scale versus the accuracy of the mode predictions.
- A controller with PLP is able to match the control effort of the non-predictive baseline, maintain a disturbance-rejection error similar to the topology-robust baseline, and achieve runtime faster than either. This suggests its potential to be used in place of a robust controller for more complex applications where designing for robustness is expensive.

[*]Corresponding author (e-mail: soojean@caltech.edu)

[1]This work is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1745301 and the Aerospace Corporation.

Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns

SooJean Han*[b], Soon-Jo Chung[b], John C. Doyle[b]

[b]*Division of Engineering and Applied Science, California Institute of Technology, Pasadena, 91125, CA, USA*

Abstract

Incorporating pattern-learning and prediction (PLP) in many discrete-time or discrete-event systems allows for less redundant controller synthesis in two ways: preserving past patterns into memory, and predicting future occurrences of patterns. In this paper, we demonstrate the effect of PLP by designing a three-part controller framework for a class of linear Markovian jump systems (MJS) where the aforementioned "patterns" correspond to finite-length sequences of modes. In our analysis of recurring patterns, we use martingale theory to derive closed-form solutions to quantities pertaining to the occurrence of patterns: 1) the expected minimum occurrence time of any pattern from some predefined collection, 2) the probability of a pattern being the first to occur among the collection. Our method is applicable to real-world dynamics by two extensions of common assumptions in prior pattern-occurrence literature: 1) the distribution of the mode process is unknown, and 2) the true realization of the mode process is not observable. As demonstration, we consider the fault-tolerant control of a dynamic topology network, and empirically compare our framework to two baselines without PLP: non-predictive and topologically-robust. A non-robust PLP controller is able to reject disturbances as effectively as a robust controller at reduced computation time and control effort. We further 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 performance.

Keywords: Analytic design, Pattern learning, Statistical approaches, Control for switching systems, Fault tolerant

1. Introduction

Model-based controller synthesis methods can be developed for stochastic systems if a theoretical characterization of their stochastic process distribution exists. In the present literature, this concept is most notable for Gaussian white noise systems Doyle (1978); Reif et al. (1999); Theodorou et al. (2010) or Markovian jump systems (MJSs) Xiong et al. (2005); Shi and Li (2015), and our analysis in Han and Chung (To appear 2022) suggested the possibility of expanding such methods to Poisson shot noise perturbations. For many discrete time or discrete-event systems, 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. For example, in fault-tolerance control or manufacturing process applications, a pattern of interest may be a specific sequence of modes which corresponds to a critical system fault Cho and Lim (1998); Hanmer (2013). Another example can be found in queuing-based systems such as vehicle intersection networks Boon and van Leeuwaarden (2016); van Leeuwaarden (2006), where repetition arises naturally when counting the number of entities in the queue over time.

Considering repeating patterns in the underlying stochastic process of many discrete stochastic systems allows for at least two ways of more efficient controller synthesis. First, we may preserve past sample paths of the stochastic process into memory so that if a certain pattern occurs multiple times, we do not need to recompute the corresponding control policy at every occurrence. Second, we may predict the expected occurrence times of patterns in the future and schedule to apply the corresponding control policies at the predicted times. This idea is present in many applications. For example, a collision-avoidance trajectory over a future horizon of time can be computed for a moving vehicle based on repeated experiences of obstacle behavior, instead of relying only on instantaneous measurements of each obstacle's position Richards and How (2006); Mesquita and Hespanha (2012); Shim et al. (2012). For the class of discrete-event systems, labeled transition representations are invoked to solve fault diagnosis and prediction problems because they enable easier identification of repeated patterns over time Jéron et al. (2008, 2006).

Related Work

Reducing Repetitive Computation: Making controller design efficient by taking advantage of any repetition in the system behavior is a fairly common concept in the engineering community. For example, Chen and Liu (2017) proposes repetitive learning control for a class of nonlinear systems tracking reference signals which are periodic; Zheng et al. (2021) discusses a method to approximate linear Gaussian systems using a hidden Markov model (HMM), then trains it by exploiting its periodic structural properties. Some notable machine learning

**Corresponding author (e-mail: soojean@caltech.edu)

[2]This work is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1745301 and the Aerospace Corporation.

approaches for control, i.e. long short-term memory networks (LSTMs) and imitation learning Verma et al. (2019), are also designed to reduce redundant learning.

Computing Pattern-Occurrence Quantities: Repetition in stochastic processes is often addressed in theory by "pattern-occurrence problems", which characterizes one or more specific sequences of values as "patterns" then solves for quantities such as the expected time until their next observation in the stochastic process. Scan statistics Pozdnyakov and Steele (2014) is a popular tool founded on martingale theory, and is often used to characterize the distribution of pattern occurrences in applications dealing such as fault-tolerance and anomaly-detection. For example, Guerriero et al. (2009) uses a scan statistics approach for distributed target-sensing using stationary sensors and a moving agent under simplified assumptions on the distribution of the sensors' positions. Formulas for predicting the occurrence of patterns are discussed in Gerber and Li (1981); Li (1980); Pozdnyakov and Kulldorff (2006) for scalar i.i.d. sequences, and in Glaz et al. (2006); Pozdnyakov (2008) for scalar Markov chain sequences.

Controlling Uncertain Systems: One notable drawback to current pattern-occurrence methods, such as Pozdnyakov and Kulldorff (2006); Glaz et al. (2006); Pozdnyakov (2008), are their reliance on the assumptions that we are able to precisely observe the occurrence of patterns and that the distribution of the underlying stochastic process is known. These assumptions often do not hold in real world applications, which motivate research such as Dean et al. (2018); Ho et al. (2021), which discuss the need to integrate system identification procedures into standard control techniques. Moreover, model predictive control (MPC) is a popular tool for incorporating prediction Yu et al. (2020) and additional state/input constraints Lu et al. (2013) into these standard methods of control. MPC can also be developed for specific classes of systems; in particular, Park and Kwon (2002) considers MPC for discrete-time MJSs when the dynamics are linear and uncertain.

Contributions

This paper aims to demonstrate the inclusion of pattern-learning and prediction (PLP) by considering the predictive control of a class of linear MJS whose underlying mode-switching dynamics are unknown. Here, "patterns" are recurrent finite-length sequences of modes which arise in the MJS, and PLP is an entity that uses these patterns to make controller synthesis efficient in two ways: 1) utilizing memory to prevent the recomputation of the control laws associated with previously-occurred patterns, and 2) utilizing prediction to enable scheduling of control laws associated with patterns which may occur in the future. The controller framework consists of three parts. First, *mode process identification* uses state and control sequences to learn the unknown statistics of the mode process; in the context of MJSs, these are the transition probability matrix (TPM) and the mode at the current time. Second, *pattern-learning* uses the estimated TPM and current mode to compute quantities pertaining to the future occurrence of patterns. Third, *mode process model predictive control (MPC)* performs the appropriate optimization to compute the control law associated with each pattern when it first occurs.

Our first main contribution is the development and inclusion of the pattern-learning component in an otherwise straightforward framework which leverages well-researched techniques in system identification and predictive control. In our analysis of recurring patterns, we use martingale theory to derive mathematical expressions for two quantities pertaining to the prediction of patterns: the expected minimum occurrence time of any pattern from some (user-defined) collection of patterns, and the probability of a pattern being the first to occur among the collection at the expected tiem. Our method operates on two key extensions of prior pattern-occurrence literature which makes it applicable to real-world dynamics: 1) the distribution of the mode process is unknown, and 2) the realization of the mode process over time is not observable. To our understanding, our proposed framework is the first to apply a martingale method to the learning-based control of a stochastic system.

Our second main contribution is an extensive comparison study which demonstrates the effects of PLP on a version of the proposed three-part framework applied to the fault-tolerant control of a network with dynamic topology, where the modes correspond to the different possible topology variations. The comparison is performed against two baseline controllers without PLP: a non-predictive framework and an extension of the non-predictive framework which is topologically-robust. Our results provide insight into several important tradeoffs among four performance metrics which determine how PLP affects a controller's performance in stabilizing the system.

Before we proceed, a summary of some of the most important notations used throughout the paper is provided in Table 1.

2. Setup and Preliminaries

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

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

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 $T_n \triangleq \min\{s \in \mathbb{N} \mid N[s] = n\}$. 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 that for all $m \in \mathcal{X}$, the pair $(A(m), B)$ is controllable, and that B is fixed constant and known.

Throughout this paper, the letter ξ is specifically reserved to denote random variable modes. We distinguish $\{\xi_n\}$ from the sequence of deterministic values $\{\varphi_n\}$ which it takes, i.e.,

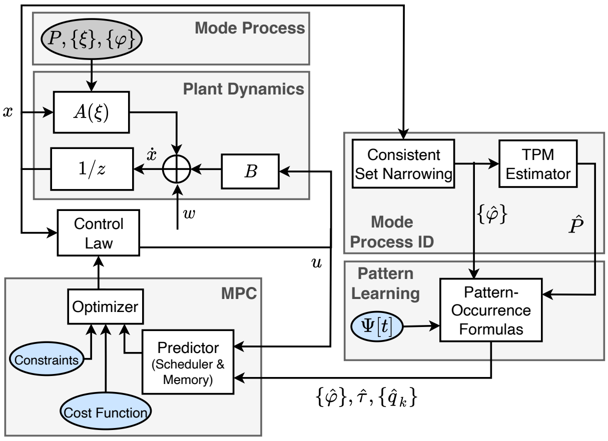


Figure 1: A flow diagram representation of the proposed controller framework specifically for linear MJS dynamics of the form (1). Circles represent inputs to the algorithm; user-defined inputs are colored blue, whereas unknown/unobservable parameters are colored gray. The framework is of three main parts: 1) mode process ID (Sec. 3.1), 2) pattern-learning (Sec. 3.2 and 4), and 3) mode process MPC (Sec. 3.3).

$\xi_n = \varphi_n$ for all past mode-indices $n \in \mathbb{N}$. Mode sequences denoted using other Greek letters are deterministic unless explicitly stated otherwise. We henceforth denote all sequences of the form $\{\cdot\}_{n=1}^{\infty}$ using the shorthand notation $\{\cdot\}$, e.g., $\{\xi_n\}_{n=1}^{\infty} \equiv \{\xi_n\}$ and $\{\mathcal{F}_n\}_{n=1}^{\infty} \equiv \{\mathcal{F}_n\}$, and denote $\mathbf{x}[s:t] = \{\mathbf{x}[s], \cdots, \mathbf{x}[t]\}$ for any $s < t$, likewise for $\mathbf{u}[s:t], \mathbf{w}[s:t]$. For any two $n, m \in \mathbb{N}$ such that $n_1 < n_2$, we denote random vectors of mode sequences $\xi_{n_1:n_2} \triangleq (\xi_{n_1}, \xi_{n_1+1}, \cdots, \xi_{n_2})$, and likewise $\varphi_{n_1:n_2}$. 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.

Assumption 1. The mode process $\{\xi_n\}$ operates on a timescale which is $\Delta T \in \mathbb{N}$ times longer than the timescale of the system (1), 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 $\mathbf{p}_0 \triangleq [p_0(1), \cdots, p_0(M)]^\top \in \{0,1\}^M$. We henceforth 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 2. In addition to knowing the values of ΔT, $N[t]$, and $\{T_n\}$ (see Assumption 1), 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.

Sym.	Definition
Δt	Timescale of mode w.r.t. system (Assum. 1)
$\hat{\varphi}_n^{(t)}$	Est. current mode at time t, $n \triangleq N[t]$ (Def. 1)
$\hat{P}^{(t)}[m_1, m_2]$	Est. TPM entry for $m_1, m_2 \in \mathcal{X}$ (Def. 1)
$C[t]$	Set of consistent modes at time t (2)
$\Psi[t]$	Time-varying pattern collection (Defs.2,6)
$\boldsymbol{\psi}_k$	A pattern from Ψ, enum. $k \in \{1, \cdots, K\}$ (Def.2)
L	Future horizon, pattern length (Def. 6)
$\mathcal{U}$	Control law table in memory (Prop. 2)
$\hat{\tau}^{(t)}$	Min. occurrence time of $\Psi[t]$ (Def.5, Rmk.3)
$\hat{q}_k^{(t)}$	First occurrence prob. of $\boldsymbol{\psi}_k \in \Psi[t]$ (Def.5, Rmk.3)
Γ	Augmented pattern collection (7)
$\boldsymbol{\gamma}_\ell$	Augmented pattern, enum. $\ell \leq \lvert\mathcal{X}\rvert^2 \lvert\Psi\rvert$ (Def. 8)
$\mathcal{S}_I^{(0)}$	Set of Case 0 initial-ending strings (Def.9)
$\mathcal{S}_I^{(1)}$	Set of Case 1 initial-ending strings (Def.9)
$\mathcal{S}_L$	Set of later-ending strings (Def.9)
$\mathcal{S}$	$= \mathcal{S}_L \cup \mathcal{S}_I = \mathcal{S}_L \cup (\cup_{i \in \{0,1\}} \mathcal{S}_I^{(i)})$ (Def.9)
$K_I^{(\chi)}$	$= \lvert\mathcal{S}_I^{(\chi)}\rvert, \chi \in \{0,1\}$ cardinality (Def.9)
K_L	$= \lvert\mathcal{S}_L\rvert$ cardinality (Def.9)
$\boldsymbol{\beta}_s$	Ending string in $\mathcal{S}$, enum. $s \in \{1, \cdots, K_I + K_L\}$
$\mathbb{P}(\boldsymbol{\beta}_s)$	Prob. that $\boldsymbol{\beta}_s$ terminates $\{\xi_n\}$ (Def. 11)
c_ℓ	Initial reward of each type-ℓ agent (Def.10)
$R_\tau^{(\ell)}$	Type-ℓ cumu. net reward (Def.13)
R_τ	Cumu. net reward (Def.13)
$W_{s\ell}$	Gain matrix entry (s, ℓ): total gain earned by type-ℓ agent via ending string $\boldsymbol{\beta}_s$ (Def.12)

Table 1: Summary of some of the most important notations used in the controller framework, listed in pairs of symbols ('Sym.') and definitions.

3. The Controller Framework

The controller framework we propose to use is visualized in Fig. 1. It consists of three main parts: 1) mode process identification (ID), 2) pattern-learning in the mode process, and 3) mode process model predictive control (MPC). Here, we discuss each individual part in detail.

3.1. Mode Process Identification

Because the mode process $\{\xi_n\}$ abides by a Markov chain, we can employ HMM procedures to estimate the quantities in Definition 1. At each time t, the estimate of the current mode can

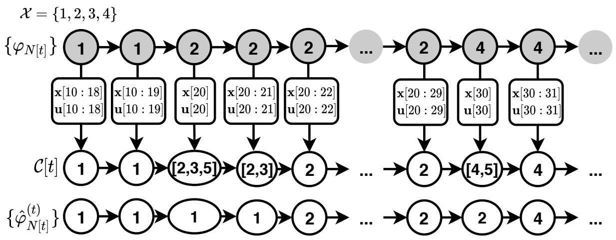


Figure 2: A visual diagram depicting mode process identification. Here, $\Delta T = 10$ and $M = 5$. With $n \triangleq N[t]$, the upper row of gray circles denotes a realization $\{\varphi_n\}$ of the original unobservable mode process $\{\xi_n\}$. The middle row denotes the evolution of the consistent set, updated via (2) using the state and control observations $\mathbf{x}[T_n : t]$, $\mathbf{u}[T_n : t]$. The estimate $\{\hat{\varphi}^{(t)}_n\}$ of the mode process is at the bottom row of white circles.

be determined by applying *Viterbi's algorithm* (see, e.g., Forney (1973)). By Assumption 1, $n \triangleq N[t]$ is known and 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. Hence, it becomes necessary to modify the standard Viterbi algorithm to account for multiple observations per mode.

When $\mathbf{w}[t]$ is norm-bounded but has unknown distribution, as in (1), Viterbi's algorithm simplifies into the following *consistent set narrowing approach*, which checks the set of modes which are 'consistent' with the state/control observations (similar to Ho et al. (2021) or (4) of Han (2020)). Denote the current mode-index as $n \triangleq N[t] \in \mathbb{N}$. If $t = T_n$, no observations about the current mode φ_n have been made yet; we thus set an initial *consistent set* to be $C[T_n] \triangleq \mathcal{X}$. For each $t \in (T_n, T_{n+1})$, if $C[t-1] \neq \varnothing$, a new consistent set is formed by retaining all modes $m \in C[t-1]$ from the previous consistent set $C[t-1]$ if each one-step pair of state and control observations $(\mathbf{x}[r], \mathbf{x}[r+1], \mathbf{u}[t])$ satisfies the norm-boundedness condition of the noise $\mathbf{w}[t]$.

$$C[t] = \left\{ m \in C[t-1] \,\middle|\, \bigwedge_{r=T_n}^{t-1} \mathbb{1}\{\|\mathbf{x}[r+1] - A(m)\mathbf{x}[r] - B\mathbf{u}[r]\|_\infty \leq \overline{w}\} \right\} \tag{2}$$

As (2) is being performed for each $t \in \mathbb{N}$, we update the estimate $\hat{\varphi}^{(t)}_{N[t]}$ of the current mode. We keep $\hat{\varphi}^{(t)}_{N[t]} = \hat{\varphi}^{(t-1)}_{N[t-1]}$ if $|C[t]| > 1$; otherwise, we update $\hat{\varphi}^{(t)}_{N[t]} \in C[t]$.

Definition 1 (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 (1): an estimate $\hat{P}^{(t)}$ of the true TPM P, and an estimate $\hat{\varphi}^{(t)}_n$ of the current mode φ_n. We use hats and superscripts (t) to emphasize that these quantities are estimates which change over time.

3.2. Pattern-Learning in the Mode Process

Once $\hat{P}^{(t)}$ and $\hat{\varphi}^{(t)}_n$ are obtained from Sec. 3.1 for each $t \in \mathbb{N}$ and $n \triangleq N[t]$, *pattern-learning* in Fig. 1 predicts the occurrence of patterns after mode-index n. In this paper, "patterns" refer to the patterns of modes in the Markov chain mode process underlying the system (1), formalized in the following definition.

Definition 2 (Patterns)**.** Define the set $\Psi \triangleq \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_K\}$, where each $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,L})$ is a mode sequence with constant length $L \in \mathbb{N}$ and elements $\psi_{k,j} \in \mathcal{X}$. Each $\boldsymbol{\psi}_k$ is referred to as a *(mode) pattern* if we are interested in observing its occurrence in the mode process $\{\xi_n\}$ over time (e.g., because it models a system fault). We refer to Ψ as a *collection of patterns*. We use the notation of Definition 2 throughout the rest of the paper.

Definition 3. A pattern or an arbitrary sequence of modes $(\alpha_1, \cdots, \alpha_a)$ with length $a \in \mathbb{N}$ is *feasible* if it can be generated by the Markov chain with TPM P, i.e., for all $i \in \{1, \cdots, a-1\}$, $P[\alpha_i, \alpha_{i+1}] > 0$.

Definition 4 (Pattern-Occurrence Times)**.** 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 $\xi_n = \hat{\varphi}^{(t)}_n$. Then for each of the patterns in the collection Ψ from Definition 2, define the following stopping times for each $k \in \{1, \cdots, K\}$:

$$\hat{\tau}^{(t)}_{k|n} \triangleq \min\{i \in \mathbb{N} \,|\, \xi_n = \hat{\varphi}^{(t)}_n, \xi_{n+i-L+1:n+i} = \boldsymbol{\psi}_k\} \tag{3}$$

Definition 5 (Time and Probability of First Occurrence)**.** Under the setup of Definition 4 suppose $\xi_{n+\hat{\tau}^{(t)}_n-L+1:n+\hat{\tau}^{(t)}_n} = \boldsymbol{\psi}_k$. Then define the following for the collection Ψ:

$$\hat{\tau}^{(t)}_n \triangleq \min_{k \in \{1,\cdots,K\}} \hat{\tau}^{(t)}_{k|n}, \qquad \hat{q}^{(t)}_k \triangleq \mathbb{P}(\hat{\tau}^{(t)}_n = \hat{\tau}^{(t)}_{k|n}) \tag{4}$$

In Definitions 4 and 5, we keep the hat and superscript (t) in the τ and q_k quantities because we emphasize that they are dependent on the estimates $\hat{P}^{(t)}$ and $\hat{\varphi}^{(t)}_n$, which may change over time.

Problem 1 (Pattern-Occurrence)**.** We are interested in characterizing the following *pattern-occurrence quantities* described in Definition 5.

- the estimate $\mathbb{E}[\hat{\tau}^{(t)}_n]$ 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 $\hat{\varphi}^{(t)}_n$.
- the estimated *first-occurrence probabilities* $\{\hat{q}^{(t)}_k\}$, where $\hat{q}^{(t)}_k \in [0, 1]$ is the probability that pattern $\boldsymbol{\psi}_k \in \Psi$ is the first to be observed among all of Ψ.

As the pattern-learning component is one of our main contributions, we defer the details of solving Problem 1 to Sec. 4.

3.3. Mode Process Model Predictive Control

Standard MPC for discrete-time linear dynamics seeks to predict a future sequence of controls $\{\mathbf{u}[t], \mathbf{u}[t+1], \cdots, \mathbf{u}[t+L]\}$ which minimizes some cost functional at each timestep $t \in \mathbb{N}$, for some finite horizon $L \in \mathbb{N}$. Once the first control input $\mathbf{u}[t]$ is applied to the system, the procedure is repeated at the next time $t + 1$. For the concreteness of this paper, we are inspired by the method of Park and Kwon (2002); Lu et al. (2013), which discusses MPC for MJSs, and we extend their approaches to our setting from Sec. 2.

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 Sec. 3.1 estimates the current mode to be $\hat{\varphi}_n^{(t)}$. Again, by Assumption 1, 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.

Example 1 (System Level Synthesis)**.** Instead of designing just the open-loop feedback gain K (in the usual law $\mathbf{u} = K\mathbf{x}$), the novel *system-level synthesis* (SLS) approach Wang et al. (2018); Anderson et al. (2019) 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 $\mathbf{u}[0:t] = \mathbf{\Phi}_u \mathbf{w}[0:t]$ for additive external disturbance $\mathbf{w}[t]$. This makes SLS more suitable for distributed and localized control law design in large-scale linear systems over network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and so $\mathbf{\Phi}$ is often implemented locally as $\mathbf{\Phi}^{(i)} \triangleq \{\Phi_x^{(i)}[s], \Phi_u^{(i)}[s]\}$ for each node $i \in \mathcal{V}$. Here, $s \in \{1, \cdots, H\}$ is index of the *spectral component*, $H \in \mathbb{N}$ is some finite horizon, and $\mathcal{L}_{i,h}$ is the set of all $j \in \mathcal{V}$ which is within h edges away from i, and $h \in \mathbb{N}$ is some number of *hops*; both H and h are parameters chosen by design based on properties such as the scale and topology of $\mathcal{G}$. □

The prediction part of MPC interplays with the pattern-learning component (Sec. 3.2) by predicting the future sequence of modes and scheduling the appropriate control laws in advance. Because the statistics of the mode process and the pattern-occurrence quantities are all estimates (see Definitions 1 and 5), it becomes necessary to consider a pattern collection Ψ (see Definition 2) which varies with time.

Definition 6 (Time-Varying Pattern-Collection)**.** To solve the Problem 1, we construct the collection of patterns $\Psi[t]$ to be a set of feasible length-L future sequences of modes for some chosen *future horizon* $L \in \mathbb{N}$, given the estimated current mode $\hat{\varphi}_n^{(t)}$:

$$\Psi[t] \subseteq \{\text{feasible } (\alpha_1, \cdots, \alpha_L) | \hat{P}^{(t)}[\hat{\varphi}_n^{(t)}, \alpha_1] > 0, \alpha_i \in \mathcal{X}\} \quad (5)$$

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 Sections 3.2 and 4. Let $\tau \equiv \mathbb{E}[\hat{\tau}_n^{(t)}]$ be the shorthand 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 $\hat{\varphi}_n^{(t)}$. To schedule a feedback control law in advance, we simply choose the pattern $\boldsymbol{\psi}_k \in \Psi[t]$ corresponding to the largest occurrence probability $\hat{q}_k^{(t)}$. Then, until mode-index τ, the future sequence of control inputs $\mathbf{u}[t : T_{n+\tau+1} - 1]$ is defined by

$$\mathbf{u}[s] = K(s, \psi_{k,1})\mathbf{x}[s], \ \ s \in [t : T_{n+1} - 1] \quad (6)$$

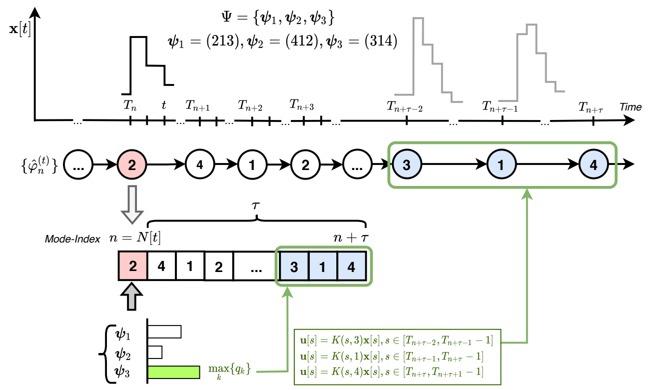


Figure 3: A visualization of PLP (see Definition 7) with a pattern collection of three patterns with $L = 3$. The middle row of circles shows the evolution of $\hat{\varphi}_n^{(t)}$ over time, while the bottom row of boxes shows the process on the mode timescale (see Assumption 1). The red circle and 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 Ψ. Control input sequences are scheduled accordingly in the dark green box.

$$\vdots$$

$$\mathbf{u}[s] = K(s, \psi_{k,L})\mathbf{x}[s], \ \ s \in [T_{n+\tau} : T_{n+\tau+1} - 1]$$

Similar to standard MPC, only the first control law in the sequence (6), corresponding to the first mode $\psi_{k,1}$, is applied at the next mode-index $n + \tau$.

Treating mode process repetitions as patterns also allows us to preserve these patterns into memory so that if the pattern occurs again, we do not need to recompute the corresponding control policy from scratch.

Proposition 2 (Storing Past Control Inputs in Memory)**.** Define $\mathcal{U}$ to be a table which maps mode patterns $\boldsymbol{\psi}_k$ to control laws $\{K(t, \psi_{k,1}), \cdots, K(t, \psi_{k,L})\}$ and the corresponding accumulated state and control trajectories over each occurrence time. When $\boldsymbol{\psi}_k \in \Psi[t]$ is first observed, a new entry $\mathcal{U}[\boldsymbol{\psi}_k](\cdot)$ defined by (6) for the specific $\boldsymbol{\psi}_k$ is created. For anticipated future occurrences of $\boldsymbol{\psi}_k$, the system predicts future control inputs using control gain $\mathcal{U}[\boldsymbol{\psi}_k]$ in the form of (6). The table entry for $\boldsymbol{\psi}_k$ is then updated at every occurrence time after its first.

Example 2 (SLS with Scheduling and Memory)**.** We can extend SLS from Example 1 to mode-switching linear dynamics (1). 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 for each mode $m \in \{1, \cdots, M\}$. 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 $\mathcal{U}$. □

Definition 7 (Pattern-Learning and Prediction)**.** The proposed three-part controller framework extends traditional uncertain system controllers (which often leverage well-researched system identification techniques) via the incorporation of *pattern-learning and prediction* (PLP), which is encompassed by pattern-learning (Sec. 3.2 and 4) and Propositions 1 and 2. Removing PLP from Fig. 1 would be represented by the removal of the pattern-learning component, and replacing MPC with some non-predictive controller (e.g., stochastic LQR, standard SLS from Example 1).

4. Solving the Pattern-Occurrence Problem

Taking a martingale approach to compute the pattern-occurrence quantities from Problem 1 makes the resulting formulas more interpretable and elegant in mathematical expression. Furthermore, when applied to scan statistics, a martingale approach also allows for a more accurate test of experiment results than simple hypothesis testing Guerriero et al. (2009).

4.1. Construction Based on Game Interpretation

As typical of martingale analyses, it becomes convenient to view Problem 1 from a game perspective.

Remark 1. We simplify the notation and remove the hats and the superscripts of (t) in the estimated quantities throughout Sec. 4 only. That is, for each n and t satisfying $N[t] = n$, we denote $\varphi_n \equiv \hat{\varphi}_n^{(t)}$, $P \equiv \hat{P}^{(t)}$, $\tau_{k|n} \equiv \hat{\tau}_{k|n}^{(t)}$, $\tau_n \equiv \hat{\tau}_n^{(t)}$, and $q_k \equiv \hat{q}_k^{(t)}$. Furthermore, we also remove the bracket $[t]$ in the pattern collection $\Psi[t]$ (see Definition 6), and use the notation Ψ instead. However, we emphasize the understanding that the computation done at time t uses the original estimates and the time-varying pattern collection.

Note that there are constraints on the degrees of freedom on possible Markov chain sample path trajectories. Thus, we take inspiration from Pozdnyakov (2008) and consider the occurrence of feasible augmented patterns up to two extra modes.

Definition 8 (Augmented Pattern)**.** Suppose we are given a collection of patterns Ψ (from Definition 6). An *augmented pattern* $\boldsymbol{\gamma}$ corresponding to a pattern $\boldsymbol{\psi}_k \in \Psi$ is defined by prefixing two modes $m_1, m_2 \in \mathcal{X}$ such that the resulting sequence is feasible in the sense of Definition 3. We define the *augmented collection*

$$\Gamma \triangleq \{\text{feasible } (m_1, m_2) \circ \boldsymbol{\psi}_k \mid m_1, m_2 \in \mathcal{X}; \boldsymbol{\psi}_k \in \Psi\} \tag{7}$$

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 $\ell \in \{1, \cdots, K_L\}$. Here, $\circ$ denotes the concatenation operation (see above Assumption 1), and note that each augmented pattern has length $L + 2$.

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 9 (Ending Strings)**.** Given the collection of patterns Ψ and current mode-index $n \in \mathbb{N}$, suppose we let the mode 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 mode process* at mode-index $\tau_n > n$ if $\xi_{\tau_n - L+1:\tau_n} = \boldsymbol{\psi}_k$. We characterize two primary types of ending strings:

- An *initial-ending string* $\boldsymbol{\beta}$ occurs when either [Case 0] $\xi_{n+1:n+L} = \boldsymbol{\psi}_k$, in which $\boldsymbol{\beta} \triangleq \boldsymbol{\psi}_k$, or [Case 1] $\xi_{n+1:n+L+1} = (m_1) \circ \boldsymbol{\psi}_k$, in which case $\boldsymbol{\beta} \triangleq (m_1) \circ \boldsymbol{\psi}_k$. Define $\mathcal{S}_I^{(0)}$ to be the set of Case 0 initial-ending strings and $\mathcal{S}_I^{(1)}$ to be the set of Case 1 initial-ending strings with respective cardinalities $K_I^{(0)}$ and $K_I^{(1)} \in \mathbb{N}$.
- A *later-ending string* $(*, m_1, m_2) \circ \boldsymbol{\psi}_k$ occurs when $\tau_n > n + L + 1$ and $\xi_{\tau_n - L+1:\tau_n} = (m_1, m_2) \circ \boldsymbol{\psi}_k$, where $*$ is a placeholder for any feasible sequence of modes (see Definition 3), 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.

Here, $m_1, m_2 \in \mathcal{X}$ are such that the above ending strings are feasible. Define $\mathcal{S}_I \triangleq \mathcal{S}_I^{(0)} \cup \mathcal{S}_I^{(1)}$ with cardinality $K_I = K_I^{(0)} + K_I^{(1)}$, and let the *set of ending strings* 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 3 (Ending Strings Construction)**.** We provide intuition behind the notation described by Definition 9. Let $M = 4$, i.e., $\mathcal{X} = \{1, 2, 3, 4\}$, and let the (estimated) 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 $\Psi = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3\}$ of length $L = 3$, 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 $\Gamma_1 = \{\boldsymbol{\alpha} \circ \boldsymbol{\gamma}_1 | \boldsymbol{\alpha} \in \{(14), (21), (24), (31), (41)\}\}$, $\Gamma_2 = \{\boldsymbol{\alpha} \circ \boldsymbol{\gamma}_2 | \boldsymbol{\alpha} \in \{(12), (21), (31), (41), (42)\}\}$, $\Gamma_3 = \{\boldsymbol{\alpha} \circ \boldsymbol{\gamma}_3 | \boldsymbol{\alpha} \in \{(21), (31), (41)\}\}$. Hence, the number of later-ending strings is $K_L = 13$. Suppose the (estimated) current state is $\varphi_n = 2$. Then the set of feasible augmented Case 0 initial-ending strings is $\mathcal{S}_I^{(0)} = \{\boldsymbol{\psi}_2\}$ since $P[2, 4] > 0$, and for Case 1 initial-ending strings, $\mathcal{S}_I^{(1)} = \{(1) \circ \boldsymbol{\psi}_1, (1) \circ \boldsymbol{\psi}_2, (1) \circ \boldsymbol{\psi}_3, (2) \circ \boldsymbol{\psi}_2\}$. Thus, $K_I^{(0)} = 1$ and $K_I^{(1)} = 4$. □

Definition 10 (Agents and Rewards)**.** Let Γ be the augmented pattern collection associated with original collection Ψ (see Definition 8). We introduce the notion of an *agent*, which observes the mode process $\{\xi_n\}$ and accumulates *rewards* at each mode-index with the goal of observing a pattern from Γ (vicariously observing a pattern from Ψ). We refer to a *type-ℓ agent* to be an agent which accumulates rewards by specifically observing the occurrence of $\boldsymbol{\gamma}_\ell \in \Gamma$ in $\{\xi_n\}$. At each mode-index $n \in \mathbb{N}$, K_L new agents, one for each type-ℓ for $\ell \in \{1, \cdots, K_L\}$ agent, are introduced to the mode process; we refer to a type-ℓ agent which is introduced at mode-index n as *type-ℓ agent n*. A type-ℓ agent n observes (estimated) mode 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* $c_\ell \in \mathbb{R}$. If $\varphi_n = m_1$, type-ℓ agent n aims to observe the event $\{\xi_{n+1:n+L+1} = (m_2) \circ \boldsymbol{\psi}_k\}$. Otherwise, if $\varphi_n \neq m_1$, type-ℓ agent n aims to observe the event $\{\xi_{n+1:n+L} = \boldsymbol{\psi}_k\}$.

Remark 2. 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) and (4) for $\gamma_\ell \in \Gamma$.

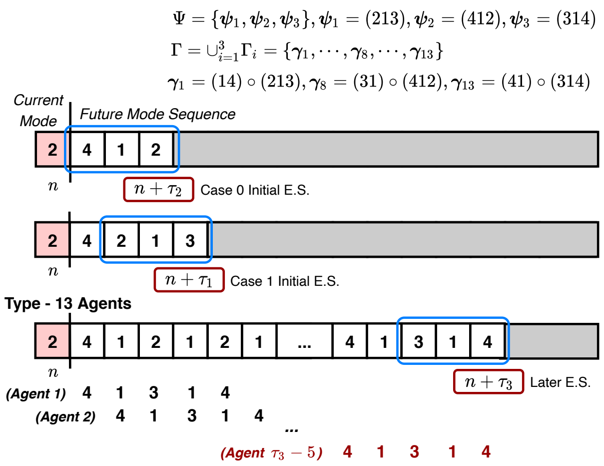


Figure 4: A visualization of the ending strings and agent-reward construction using the setup of Example 3. The red box marks the current mode-index $n \in \mathbb{N}$, and each of the three sequences demonstrate the three different types of ending strings (abbreviated "E.S." in the figure) which terminate the mode process in the sense of Definition 9. For the last case where γ_{13} terminates the mode process as a later-ending string, type-13 agents at mode indices $1, 2, \cdots, \tau_3 - 5$ are shown. By the reward construction of Definition 10, type-13 agent $\tau_3 - 5$ is the only agent who receives a nonzero reward.

Remark 3. 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 mode-indices $n_1, n_2 \in \mathbb{N}$, 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 mode-indices to $n = 0$ in variables such that the current mode is given by φ_0 instead of φ_n. Furthermore, we apply the shorthand notation to Definitions 4 and 5 such that $\tau_k \equiv \tau_{k|0}$ and $\tau \equiv \tau_0$; the notation for the augmented patterns (Remark 2) follow similarly as $\tau_\ell^a \equiv \tau_{\ell|0}^a$ and $\tau^a \equiv \tau_0^a$.

Definition 11 (Ending String Probabilities)**.** Define $\mathbb{P}(\boldsymbol{\beta}_s)$ to be the probability that ending string $\boldsymbol{\beta}_s \in \mathcal{S}$ terminates the mode process $\{\xi_n\}$ in the sense of Definition 9. 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 $\mathbb{P}(\boldsymbol{\beta}_s) = P[\varphi_0, \beta_1] \prod_{j=1}^{b_s-1} P[\beta_j, \beta_{j+1}]$. We demonstrate how to compute $\mathbb{P}(\boldsymbol{\beta}_s)$ for later-ending strings $\boldsymbol{\beta}_s \in \mathcal{S}_L$ in the following Sec. 4.2, as part of solving Problem 1.

Definition 12 (Gain Matrix)**.** 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 $W_{s\ell} \triangleq \sum_{j=1}^{\min(b_s-1, L+1)} D_j^{(1)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) + \sum_{j=1}^{\min(b_s-1, L)} D_j^{(2)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell)$ with $D_i^{(1)}$ and $D_i^{(2)}$ defined based on the reward strategy from Definition 10:
(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 $j \in \{2, \cdots, i\}$, then $D_i^{(1)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) \triangleq (P[m_1, m_2] P[m_2, \psi_{k,1}] \prod_{j=2}^{i-1} P[\psi_{k,j-1}, \psi_{k,j}])^{-1}$; else, $D_i^{(1)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) = 0$.
(2) If $\beta_{b_s-i} \neq m_2$ and $\beta_{b_s-i+j} = \psi_j$ for all $j \in \{1, \cdots, i\}$, then $D_i^{(2)}(\boldsymbol{\beta}_s, \boldsymbol{\gamma}_\ell) \triangleq (P[\beta_{b_s-i}, \psi_1] \prod_{j=2}^{i} P[\psi_{k,j-1}, \psi_j])^{-1}$; 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 13 (Cumulative Net Reward)**.** The expected *type-ℓ cumulative net reward* over all type-ℓ agents by mode-index τ is defined $\mathbb{E}[R_\tau^{(\ell)}] \triangleq c_\ell([\mathbb{P}(\boldsymbol{\beta}_1), \cdots, \mathbb{P}(\boldsymbol{\beta}_{K_I+K_L})] W_{\cdot,\ell} - \mathbb{E}[\tau])$, where the $\mathbb{P}(\boldsymbol{\beta}_s)$ are the probabilities from Definition 11 and $W_{\cdot,\ell}$ denotes the ℓth column of the gain matrix (see Definition 12). Correspondingly, the expected *cumulative net reward* over all agents by mode-index $\overline{n}$ is defined as $R_{\overline{n}} \triangleq \sum_{\ell=1}^{K_L} R_{\overline{n}}^{(\ell)}$, and

$$\mathbb{E}[R_\tau] = [\mathbb{P}(\boldsymbol{\beta}_1) \cdots \mathbb{P}(\boldsymbol{\beta}_{K_I+K_L})] W \mathbf{c} - \left(\sum_{\ell=1}^{K_L} c_\ell \right) \mathbb{E}[\tau] \tag{8}$$

where $\mathbf{c} \triangleq [c_1, \cdots, c_{K_L}]^\top$ is the vector of initial rewards (see Definition 10).

4.2. The Pattern-Occurrence Quantities

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

Theorem 1 (Expected Time of Occurrence)**.** Denote τ as in Remark 3 with (estimated) current mode φ_0 for the collection Ψ from Definition 2 and corresponding augmented collection Γ. Then

$$\mathbb{E}[\tau] = \frac{1}{\sum_{\ell=1}^{K_L} c_\ell^*} \left[\left(1 - \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \right) + \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^* \right] \tag{9}$$

where $\boldsymbol{\gamma}_\ell \in \Gamma$, $\boldsymbol{\beta}_s \in \mathcal{S}$, $\mathbb{P}(\boldsymbol{\beta}_s)$ is from Definition 11, W is from Definition 12, and $\mathbf{c}^* \in \mathbb{R}^{K_L}$ is the vector of initial rewards (see Definition 12) 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 2. Note that $\tau_k = \min_{\gamma_\ell \in \Gamma_k} \tau_\ell^a$, 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 5, 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 $\{R_{n \wedge \tau_\ell^a}^{(\ell)}\}_{n \in \mathbb{N}}$ and $\{R_{n \wedge \tau}\}_{n \in \mathbb{N}}$ are martingales. This implies that $\mathbb{E}[R_{\tau_\ell^a}^{(\ell)}] < \infty$ 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 Doob's martingale convergence theorem and the triangle inequality, $\lim_{n \to \infty} \int_{\Omega_n^{(\ell)}} |R_n^{(\ell)}(\omega)| d\mathbb{P}(\omega) = 0$, which implies $\{R_{n \wedge \tau_\ell^a}^{(\ell)}\}$ 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 the optional stopping theorem 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 13. After choosing the initial rewards $\mathbf{c}^*$ as in the theorem statement, and substituting into (8):

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

Rearranging the terms to isolate $\mathbb{E}[\tau]$ yields (9). □

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 11).

Theorem 2 (First-Occurrence Probabilities)**.** In addition to the setup of Theorem 1, 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 (estimated) first-occurrence probabilities $\{q_k\}$ (see Definition 5 and Remark 1) are given by $q_k = \sum_{\boldsymbol{\beta}_s \in \mathcal{S}} \mathbb{P}(\boldsymbol{\beta}_s) \mathbb{1}\{\beta_{b_s - L+1:b_s} = \boldsymbol{\psi}_k\}$.

Proof. Rearranging the terms of (8):

$$\sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell = - \sum_{s=K_I+1}^{K_I+K_L} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell + \sum_{\ell=1}^{K_L} c_\ell \mathbb{E}[\tau] \tag{10}$$

We are given $\mathbb{E}[\tau]$ from Theorem 1, and $\mathbb{P}(\boldsymbol{\beta}_s)$ can be computed via Definition 11 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 (10) 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 5, $q_k \triangleq \mathbb{P}(\tau = \tau_k) = \sum_{\boldsymbol{\beta}_s \in \mathcal{S}} \mathbb{P}(\boldsymbol{\beta}_s) \mathbb{P}(\tau = \tau_k \,|\, \boldsymbol{\beta}_s)$, 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 9. Clearly, $\mathbb{P}(\boldsymbol{\psi}_k \,|\, \boldsymbol{\beta}_s) = 1$ if $\beta_{b_s - L+1:b_s} = \boldsymbol{\psi}_k$ holds, otherwise it is 0. We thus obtain the desired equation. □

Remark 4. In order to fit the closed-form expressions of Theorems 1 and 2 into the original framework described throughout Sec. 3, we unsimplify the notation from Remark 1 and Remark 3 for general time $t \in \mathbb{N}$ and corresponding mode-index $n \triangleq N[t]$. This yields the original time-dependent pattern-occurrence quantities desired in Problem 1. Namely, with estimated current mode $\hat{\varphi}_n^{(t)}$ and TPM $\hat{P}^{(t)}$, the estimated expected minimum occurrence time $\mathbb{E}[\hat{\tau}_n^{(t)}]$ is the $\mathbb{E}[\tau]$ computed from Theorem 1, while the estimated first occurrence probabilities $\{\hat{q}_k^{(t)}\}$ are the $\{q_k\}$ computed from Theorem 2.

Remark 5. The pattern-learning component (Sec. 3.2) and the formulas developed in Sec. 4 apply to collections of patterns with arbitrary pattern length, e.g., $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,d_k})$ for some $d_k \in \mathbb{N}$ which varies with $k \in \{1, \cdots, K\}$. However, for the purposes of mode process MPC where the future horizon is fixed at some $L \in \mathbb{N}$, we consider only constant-length patterns for each $k \in \{1, \cdots, K\}$, hence the construction of the pattern collections in Definitions 2 and 6.

5. Case Study: Topology-Switching Network

The control of networks which undergo faults (i.e., parametric and topological changes) has become an important and widely-studied problem with recent trends in large-scale systems and smart technology. Among recent literature, an adaptive, consensus-based control scheme for complex networks with time-varying, switching network topology was discussed in Chung et al. (2013). Distributed target-detection and tracking using a dynamic sensor network was studied in Bandyopadhyay and Chung (2018), while Saboori and Khorasani (2015) described fault-tolerance against actuator failures in a multiagent system connected by a switching topology network. For the purposes of this paper, we demonstrate the proposed controller framework to the following extension of (1), which switches among a finite number of different topologies $\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m)), m \in \{1, \cdots, M\}, M \in \mathbb{N}$.

$$\mathbf{x}_i[t+1] = A_{ii}(\xi_{N[t]})\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i(\xi_{N[t]})} A_{ij}(\xi_{N[t]})\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t] \tag{11}$$

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 $m \in \{1, \cdots, M\}$. The assumptions from Sec. 2 still hold, and 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.

5.1. Experiment Setup

We consider three versions of the controller framework Fig. 1; a visual distinction among the three is shown in Fig. 5. Here, PLP refers to pattern-learning and prediction defined in Definition 7.

Non-Robust without PLP [*First row of Fig. 5*]: here, Fig. 1 is implemented only using mode process ID; the pattern-learning block does not exist, and model predictive control is replaced with the SLS approach from Example 1.

Topologically-Robust without PLP [*Second row of Fig. 5*]: we have the same framework as above, but SLS is replaced with the method of Han (2020), 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 $C[t]$, and this common law is used until time $t^* > t$ when $|C[t^*]| = 1$, after which standard SLS is used.

Non-Robust with PLP [*Third row of Fig. 5*]: we combine the original framework proposed by Fig. 1 with the extended SLS approach described in Example 2. 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 $\mathbf{\Phi}_m^{(i,s)}$, where $m = \psi_1$ and $s \in [T_{n+\mathbb{E}[\hat{\tau}_n^{(t)}]}, t^*)$, where T_n is defined in Assumption 1 and t^* is the time after $T_{n+\mathbb{E}[\hat{\tau}_n^{(t)}]}$ when $|C[t^*]| = 1$. For times $s \in [t, T_{n+\mathbb{E}[\hat{\tau}_n^{(t)}]})$ where a prediction is not available, we revert to Non-Robust without PLP.

We emphasize that even though SLS is localized and distributed, we assume mode process ID and pattern-learning are

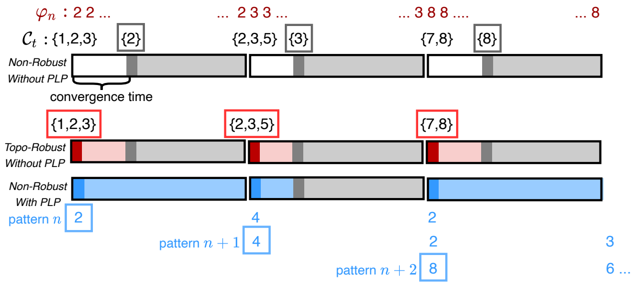


Figure 5: The time-varying control law for each of the three versions of the controller framework, designed based on the estimated mode $\hat{\varphi}_n^{(t)}$ and the consistent set $C[t]$. Each horizontal bar represents a time duration of length ΔT (see Assumption 1). Non-Robust without PLP uses the previous law until the consistent set converges to a singleton set before being able to design a control law (shown in white). Topologically-Robust is able to control multiple modes simultaneously, so it uses a robust law (shown in red) until the convergence. Non-Robust with PLP (future horizon $L = 3$) uses the law corresponding to the predicted next mode (shown in blue) until convergence; note that when the mode in the converged consistent set is equivalent to the predicted next mode, the control gain need not be changed.

centralized for the purposes of demonstrating the effects of PLP. In Sec. 5.3, we discuss the effect of localized, distributed implementation for mode process ID and pattern-learning.

The three frameworks are each tested on two specific network systems of the form given in (11). For both systems, the specific A and B matrices in (11) are the linearized discrete-time power grid dynamics given in Section 5 of Han (2020), which we do not repeat here for the sake of brevity.

(Small-Scale) Hexagon System: the network system (11) consists of a hexagonal arrangement of $n_s = 7$ nodes and $M = 8$ topologies (see Fig. 6). When PLP is included, the collection of patterns $\Psi[t]$ is constructed with equality in (5); hence, Problem 1 become easy to solve – every ending string in $\mathcal{S}$ is an initial-ending string, $\mathbb{E}[\hat{\tau}^{(t)}] = L$ for each $t \in \mathbb{N}$, and determining $\operatorname{argmax}_k\{\hat{q}_k^{(t)}\}$ reduces to a maximum likelihood problem.

(Large-Scale) Rectangular Grid System: the network system (11) consists of a 10×10 rectangular grid arrangement of $n_s = 100$ nodes and $M = 20$ topologies (see Fig. 7). 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 (5), which means the pattern-occurrence formulas from Theorems 1 and 2 must be used to solve Problem 1.

5.2. Tradeoff Comparison Results

Each simulation is run by applying one of the three controller frameworks to one of the two network systems. The comparisons are performed by evaluating one of the following four performance metrics. First, to measure the control effort, an LQR-like cost (12a) is averaged over the entire simulation duration T_{sim}. Second, to measure the disturbance-rejection performance, we consider the the time-average error norm (12b). Third, for the non-robust controller frameworks, we measure the proportion (12c) of the simulation duration in which the

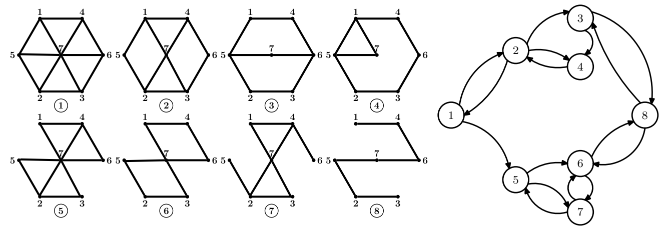


Figure 6: [Left] The different possible topologies of the Hexagon System. [Right] The underlying Markov chain for topology transitions.

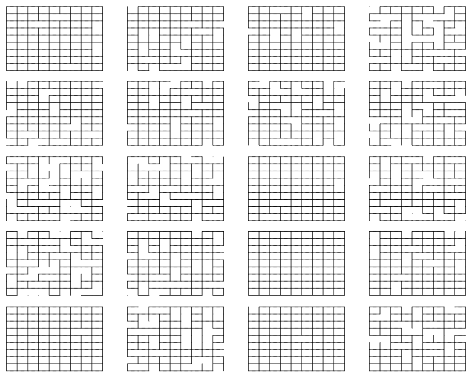


Figure 7: The different possible topologies of the 10×10 Rectangular Grid System.

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.

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbf{x}[t]^\top I_{n_x} \mathbf{x}[t] + \mathbf{u}[t]^\top I_{n_u} \mathbf{u}[t] \tag{12a}$$

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \|\mathbf{x}[t]\|_2 \tag{12b}$$

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbb{1}\{\hat{\varphi}_n^{(t)} = \varphi_n\} \tag{12c}$$

where I_{n_x}, I_{n_u} are identity matrices of the appropriate dimensions. The metrics (12) 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 duration of time for each simulation is $T_{\text{sim}} = 300$ timesteps, and Non-Robust with PLP included uses a future horizon of $L = 3$. The results are tabulated in Table 2, with the three framework names abbreviated; ‘NR’ as Non-Robust without PLP, ‘TR’ as Topo-Robust without PLP, and ‘NRPLP’ as Non-Robust with PLP.

The values in both rows of the ‘LQR Cost’ table in Table 2 suggest that the time-average LQR cost of all three controller frameworks increase as the scale of the system gets larger. This is expected because the same values of horizon H and number of hops h (defined in Example 1) 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, TR should theoretically stabilize the system better than NR at the expense of increased control effort because TR 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' tables.

More interestingly, the NRPLP framework manages to achieve the best of all three worlds: LQR cost similar to the NR framework, error norm similar to the TR framework, and runtime faster than either TR or NR. The improved runtime results from the pattern-learning component's 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 NRPLP 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 Fig. 5, this improves the error norm when pattern-learning manages to predict the future mode correctly. The 'Prop Match' table of Table 2 shows that this is indeed the case: the NRPLP framework consistently uses the matching control law more often than NR 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 for a good choice of pattern collections $\Psi[t]$, appending PLP to a non-robust controller framework could be used as an alternative to a robust controller, especially in complex systems where computing a robust controller takes too much computation time, memory, and control effort.

The difference between the construction of $\Psi[t]$ in the hexagon system versus the grid system also provides insight into an interesting tradeoff among three quantities: the size of the pattern collection, how often the matching control law is used, and the disturbance-rejection 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 NRPLP column of the 'Prop Match' table, 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 NR and NRPLP columns in the 'Error Norm' table suggest that better predictions enable better disturbance-rejection; this implies that NRPLP will more closely resemble the error norm of NR when less patterns are included in $\Psi[t]$.

5.3. Localized Pattern-Learning and Prediction

Table 2 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 NRPLP to a localized, distributed implementation of PLP.

LQR Cost	NR	TR	NRPLP
Hexagon	32.0972	43.7638	31.1688
Grid	467.8021	473.0120	467.0066

Error Norm	NR	TR	NRPLP
Hexagon	2.9032	1.5984	1.6029
Grid	7.6266	5.5633	5.5349

Prop. Match	NR	NRPLP
Hexagon	0.4304	0.615
Grid	0.1533	0.16

Runtime	NR	TR	NRPLP
Hexagon	11.8314	67.2254	2.2689
Grid	101.3741	X	38.5824

Table 2: The average performance metrics over 20 Monte-Carlo simulations of $T_{\text{sim}} = 300$ timesteps, for each pair of controller framework [column] and network system [row]. For the sake of space, we abbreviate 'NR' as Non-Robust without PLP, 'TR' as Topo-Robust without PLP, and 'NRPLP' as Non-Robust with PLP. [Top] Time-average LQR cost. [Middle] Proportion of time during which the matching control law is used on the corresponding true mode. [Bottom] Runtime duration.

Let current time be $t \in \mathbb{N}$ and $n \triangleq N[t]$. Based on information from its own local subsystem (11), 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 Sec. 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 via Sec. 3.2 and 4.2), and 3) its own pattern collection $\Psi^{(i)}[t]$ and pattern-to-control law table $\mathcal{U}^{(i)}$ (see Sec. 3.3). Each node $i \in \mathcal{V}$ employs the consistent set narrowing approach of (2) to update its own set $C^{(i)}[t]$ of consistent topologies over time t. Essentially, the key distinction is that we add an additional enumeration $i \in \mathcal{V}$ to the usual sets, tables, and estimated quantities from Definition 1 and Problem 1 in order to emphasize that each subsystem maintains its own values.

In Fig. 8, we plot the estimated pattern-occurrence quantities for this localized extension of NRPLP applied to the hexagon system. To demonstrate the evolution of the pattern-occurrence quantities over time, each subsystem's pattern collection $\Psi^{(i)}[t]$ is chosen to be a strict subset of the full combinatorial set of feasible length-L mode sequences initially considered in Sec. 5.2, and we apply Theorems 1 and 2 to each local estimate. The evolution of the estimated minimum occurrence time $\mathbb{E}[\hat{\tau}^{(i,t)}]$ over t is shown at the top, while the MSE 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.

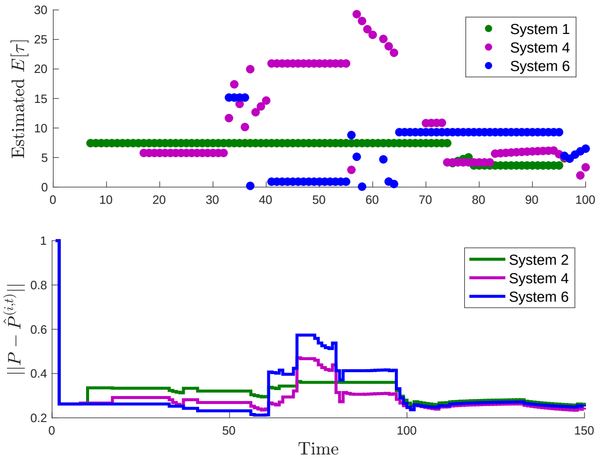


Figure 8: [Top] The evolution of $\mathbb{E}[\hat{\tau}^{(i,t)}]$ over time for subsystems $i \in \{1, 4, 6\}$. [Bottom] The evolution over time of the error matrix norm between P and $\hat{P}^{(i,t)}$ for subsystems $i \in \{2, 4, 6\}$.

The piecewise nature of the $\mathbb{E}[\hat{\tau}^{(i,t)}]$ over t in Fig. 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 Fig. 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. Thus, even a localized implementation of PLP further supports the insights obtained from centralized PLP in Sec. 5.2: 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 the performance improvement that PLP brings.

6. Conclusion

Pattern-learning and prediction (PLP) is an entity which leverages patterns in the behavior of stochastic uncertain systems to make controller synthesis efficient in two ways: 1) by utilizing memory to prevent the recomputation of the control laws associated with previously-occurred patterns (see Proposition 2), and 2) utilizing prediction to enable scheduling of control laws associated with patterns which may occur in the future (see Proposition 1). In this paper, we aimed to demonstrate the effects of including PLP in otherwise straightforward controller frameworks for stochastic uncertain systems by specifically considering the predictive control of a class of linear MJS whose underlying mode-switching dynamics are unknown; here, the aforementioned patterns are recurrent finite-length sequences of modes which arise in the MJS. Our controller framework consists of three parts. First, mode process ID (Sec. 3.1) identified the unknown statistics of the mode process given by Definition 1 by combining HMM estimation techniques with a consistent set narrowing approach (2). Second, pattern-learning (Sec. 4) predicts the occurrence of patterns by deriving estimates of two key pattern-occurrence quantities: the expected minimum occurrence time of any pattern from a user-defined pattern collection, and the probability of a pattern being the first to occur among the collection; closed-form expressions of the estimated quantities are derived by Theorems 1 and 2. Third, mode process MPC (Sec. 3.3) computes the optimal control action corresponding to the pattern most likely to occur next, and schedules to apply it at its expected time of occurrence, while maintaining a table mapping patterns to scheduled control inputs.

We implement our controller framework (NRPLP) on the fault-tolerant control of a network with dynamic topology, and provide an empirical comparison study of its performance against non-predictive (NR) and topology-robust (TR) baselines to demonstrate the effect of PLP. Because pattern-learning can be viewed as an additional mode estimation algorithm, it enables the estimated mode to match the true mode more often, although this is mainly possible for an optimal choice of pattern collection. Compared to NR, NRPLP is able to achieve better disturbance-rejection at significantly reduced computation time, redundancy, and control cost, which suggests 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 can be summarized as follows: more efficient controller synthesis for stochastic systems with uncertain dynamics can be performed by learning patterns in the system's behavior, which eliminates computation time and redundancy by preserving past patterns into memory and predicting the future occurrence of patterns.

ACKNOWLEDGMENT

The authors would like to thank John Brader and Benjamen Bycroft of the Aerospace Corporation for their technical inputs.

References

Anderson, J., Doyle, J.C., Low, S., Matni, N., 2019. System level synthesis. ArXiv preprint, arXiv:1904.01634 .

Bandyopadhyay, S., Chung, S.J., 2018. Distributed bayesian filtering using logarithmic opinion pool for dynamic sensor networks. Automatica 97, 7–17.

Boon, M., van Leeuwaarden, J., 2016. Networks of fixed-cycle intersections. ArXiv preprint, arXiv:1611.02947 .

Chen, P., Liu, X., 2017. Repetitive learning control for a class of partially linearizable uncertain nonlinear systems. Automatica 85, 397–404.

Cho, K.H., Lim, J.T., 1998. Synthesis of fault-tolerant supervisor for automated manufacturing systems: a case study on photolithographic process. IEEE Transactions on Robotics and Automation 14, 348–351.

Chung, S.J., Bandyopadhyay, S., Chang, I., Hadaegh, F.Y., 2013. Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs. Automatica 49, 1148–1161.

Dean, S., Tu, S., Matni, N., Recht, B., 2018. Safely learning to control the constrained linear quadratic regulator. ArXiv preprint, arXiv:1809.10121 .

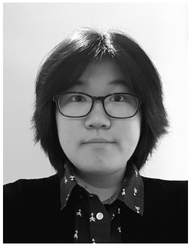
Doyle, J., 1978. Guaranteed margins for LQG regulators. IEEE Transactions on Automatic Control 23, 756–757.

Forney, G., 1973. The Viterbi algorithm. Proceedings of the IEEE 61, 268–278.

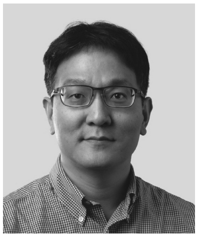
Gerber, H.U., Li, S.Y.R., 1981. The occurrence of sequence patterns in repeated experiments and hitting times in a Markov chain. Stochastic Proc. their App. 11, 101–108.

Glaz, J., Kulldorff, M., Pozdnyakov, V., Steele, J.M., 2006. Gambling teams and waiting times for patterns in two-state Markov chains. J. Appli. Probability 43, 127–140.

Guerriero, M., Willett, P., Glaz, J., 2009. Distributed target detection in sensor networks using scan statistics. IEEE Transactions on Signal Processing 57, 2629–2639.
Han, S., 2020. Localized learning of robust controllers for networked systems with dynamic topology, in: Pro. 2nd Conf. Learning Dynamics Control, PMLR. pp. 687–696.
Han, S., Chung, S.J., To appear 2022. Incremental nonlinear stability analysis for stochastic systems perturbed by Lévy noise. Int. J. Robust Nonlinear Control .
Hanmer, R.S., 2013. Patterns for fault tolerant software. John Wiley & Sons, USA.
Ho, D., Le, H., Doyle, J., Yue, Y., 2021. Online robust control of nonlinear systems with large uncertainty, in: Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, PMLR. pp. 3475–3483.
Jéron, T., Marchand, H., Genc, S., Lafortune, S., 2008. Predictability of sequence patterns in discrete event systems. IFAC Proceedings Volumes 41, 537–543.
Jéron, T., Marchand, H., Pinchinat, S., Cordier, M.O., 2006. Supervision patterns in discrete event systems diagnosis, in: 8th Int. Workshop Discrete Event Sys., pp. 262–268.
van Leeuwaarden, J.S.H., 2006. Delay analysis for the fixed-cycle traffic-light queue. Transportation Science 40, 189–199.
Li, S.Y.R., 1980. A Martingale Approach to the Study of Occurrence of Sequence Patterns in Repeated Experiments. Annals Probability 8, 1171 – 1176.
Lu, J., Li, D., Xi, Y., 2013. Constrained model predictive control synthesis for uncertain discrete-time Markovian jump linear systems. IET Control Theory & Applications 7, 707–719.
Mesquita, A.R., Hespanha, J.P., 2012. Jump control of probability densities with applications to autonomous vehicle motion. IEEE Transactions on Automatic Control 57, 2588–2598.
Park, B.G., Kwon, W.H., 2002. Robust one-step receding horizon control of discrete-time Markovian jump uncertain systems. Automatica 38, 1229–1235.
Pozdnyakov, V., 2008. On occurrence of patterns in Markov chains: Method of gambling teams. Statistics Probability Lett. 78, 2762–2767.
Pozdnyakov, V., Kulldorff, M., 2006. Waiting times for patterns and a method of gambling teams. American Math. Monthly 113, 134–143.
Pozdnyakov, V., Steele, J.M., 2014. Scan Statistics: Pattern Relations and Martingale Methods. https://faculty.wharton.upenn.edu/wp-content/uploads/2014/06/Pozdnyakov_Steele-Patterns_survey.pdf.
Reif, K., Gunther, S., Yaz, E., Unbehauen, R., 1999. Stochastic stability of the discrete-time extended Kalman filter. IEEE Transactions on Automatic Control 44, 714–728.
Richards, A., How, J., 2006. Robust variable horizon model predictive control for vehicle maneuvering. International Journal of Robust and Nonlinear Control 16, 333 – 351.
Saboori, I., Khorasani, K., 2015. Actuator fault accommodation strategy for a team of multi-agent systems subject to switching topology. Automatica 62, 200–207.
Shi, P., Li, F., 2015. A survey on Markovian jump systems: Modeling and design. International Journal of Control, Automation and Systems 13, 1–16.
Shim, T., Adireddy, G., Yuan, H., 2012. Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 226, 767–778.
Theodorou, E., Buchli, J., Schaal, S., 2010. A generalized path integral control approach to reinforcement learning. Journal of Machine Learning Research 11, 3137–3181.
Verma, A., Murali, V., Singh, R., Kohli, P., Chaudhuri, S., 2019. Programmatically interpretable reinforcement learning. ArXiv preprint, arxiv:1804.02477 .
Wang, Y.S., Matni, N., Doyle, J., 2018. Separable and localized system level synthesis for large-scale systems. IEEE Transactions on Automatic Control 63, 4234–4249.
Xiong, J., Lam, J., Gao, H., Ho, D.W., 2005. On robust stabilization of Markovian jump systems with uncertain switching probabilities. Automatica 41, 897–903.
Yu, C., Shi, G., Chung, S.J., Yue, Y., Wierman, A., 2020. The power of predictions in online control, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 1994–2004.
Zheng, K., Shi, D., Shi, L., 2021. Learning hidden markov models for linear gaussian systems with applications to event-based state estimation. Automatica 128, 109560.



SooJean Han received her B.S. degree in Electrical Engineering and Computer Science, and Applied Mathematics at the University of California, Berkeley (UC Berkeley) in 2016. In 2017, she was a research assistant in the Hybrid Systems Lab at UC Berkeley, and a research assistant in the Center for Autonomous Systems and Technology (CAST) Lab at California Institute of Technology (Caltech). She is currently a Ph.D. candidate in Control and Dynamical Systems at Caltech. Her research interests include stochastic nonlinear systems, decision-making networked systems, robotics, and machine learning. She received the Caltech Special EAS Fellowship and the National Science Foundation Graduate Research Fellowship (NSF GRFP).



Soon-Jo Chung is the Bren Professor of Aerospace and Control and Dynamical Systems in the Department of Aerospace/ Graduate Aerospace Laboratories (GALCIT) and Professor in the Department of Computing + Mathematical Sciences (CMS) in the California Institute of Technology, USA, as well as a Jet Propulsion Laboratory Senior Research Scientist. He was with the faculty of the University of Illinois at Urbana-Champaign during 2009–2016. He received the B.S. degree (*summa cum laude*) in aerospace engineering from the Korea Advanced Institute of Science and Technology, South Korea, in 1998, and the S.M. degree in aeronautics and astronautics and the Sc.D. degree in estimation and control from Massachusetts Institute of Technology, USA, in 2002 and 2007, respectively. His research interests include spacecraft and aerial swarms and autonomous aerospace systems, and in particular, on the theory and application of complex nonlinear dynamics, control, estimation, guidance, and navigation of autonomous space and air vehicles. Dr. Chung was the recipient of the UIUC Engineering Deans Award for Excellence in Research, the Beckman Faculty Fellowship of the UIUC Center for Advanced Study, the U.S. Air Force Office of Scientific Research Young Investigator Award, the National Science Foundation Faculty Early Career Development Award, a 2020 Honorable Mention for the IEEE Robotics and Automation Letters Best Paper Award, and three Best Conference Paper Awards from the IEEE and AIAA. He is an Associate Editor of IEEE Transactions on Automatic Control and AIAA Journal of Guidance, Control, and Dynamics. He was an Associate Editor of IEEE Transactions on Robotic (T-RO), and the Guest Editor of a Special Section on Aerial Swarm Robotics published in the IEEE T-RO. He is an Associate Fellow of AIAA.

John C. Doyle is the Jean-Lou Chameau Professor of Control and Dynamical Systems, Electrical Engineer, and BioEngineering at Caltech, and received the BS&MS in EE, MIT (1977), and PhD in Math, UC Berkeley (1984)). He was a consultant at Honeywell Systems and Research Center from 1976 to 1990. His early work was on robustness of feedback control systems with applications to aerospace and process control. Current re-

search includes the mathematical foundations for complex networks with applications in biology, technology, medicine, ecology, neuroscience, and multiscale physics that integrates theory from control, computation, communication, optimization, and statistics, with an emphasis on universal laws and architectures, robustness/efficiency and speed/accuracy tradeoffs, adaptability, and evolvability and large scale systems, in particular, control with sparse, local, delayed, saturating, quantized, uncertain sensing, communications, computing, and actuation using System Level Synthesis. His students and research group developed software packages like the Matlab Robust Control Toolbox and the Systems Biology Markup Language (SBML). He received the 1990 IEEE Baker Prize (for all IEEE publications), also listed in the world top 10 "most important" papers in mathematics 1981-1993, IEEE Automatic Control Transactions Award (x3 1998, 1999,2021), 1994 AACC American Control Conference Schuck Award, 2004 ACM Sigcomm Paper Prize and 2016 "test of time" award, and inclusion in Best Writing on Mathematics 2010. Individual awards include 1977 IEEE Power Hickernell, 1983 AACC Eckman, 1984 UC Berkeley Friedman, 1984 IEEE Centennial Outstanding Young Engineer (a one-time award for IEEE 100th anniversary), and 2004 IEEE Control Systems Field Award. He is best known for the shortest abstract in IEEE history, with fabulous friends, partner, colleagues, and students, and has held world and national records and championships in various sports (but is otherwise quite fragile).