Control of Jump Stochastic Systems via Characterization of Their Recurrent Patterns

---

SooJean Han (한수진)

October 5th, 2022

California Institute of Technology

Caltech

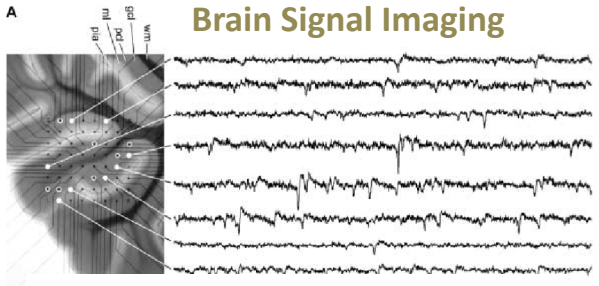
“Jump Stochastic Systems”

Systems with random and repetitive jumps.

Agile Quadrotor Flight

*O’Connell et al, 2022. *Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds*

Brain Signal Imaging

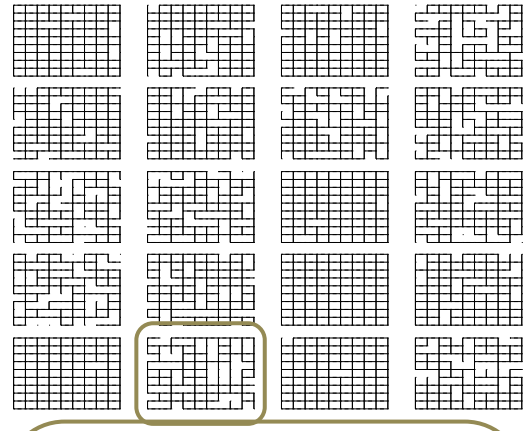


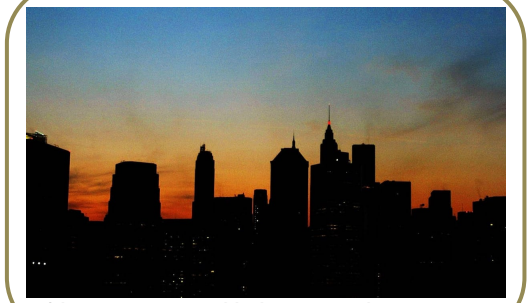
* Egert et al, 2002. *Two-dimensional monitoring of spiking networks in acute brain slices*

Spacecraft Control

* Yin et al, 2016. *A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System*

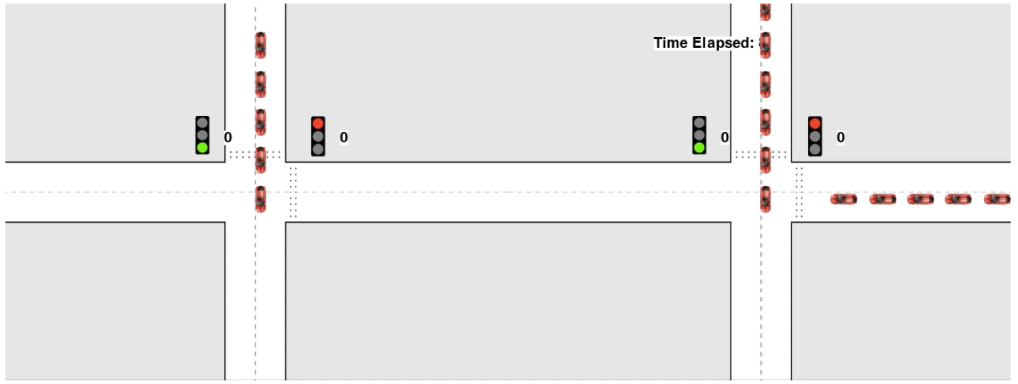
Power Grid



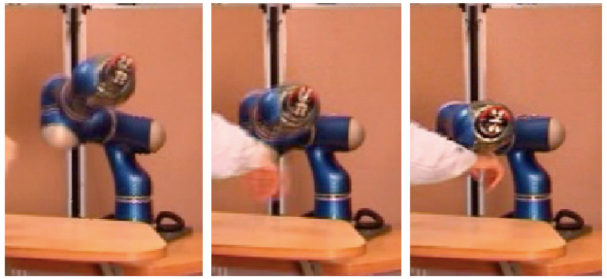


* history.com, 2003 blackout in Northeast USA

Vehicle Traffic Congestion

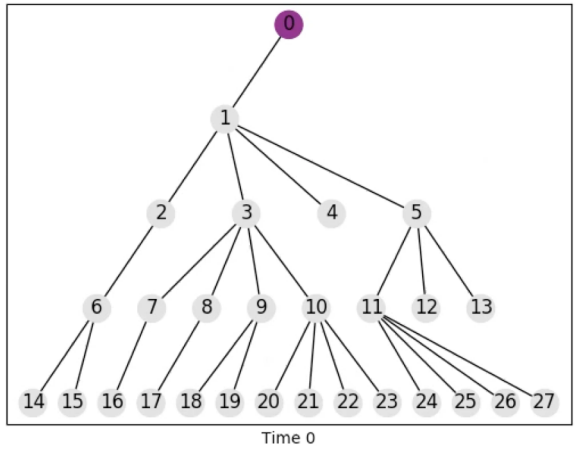


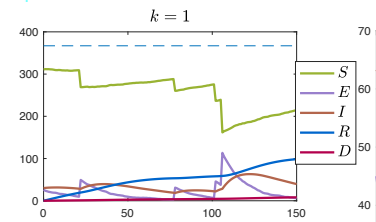
Robot Manipulator

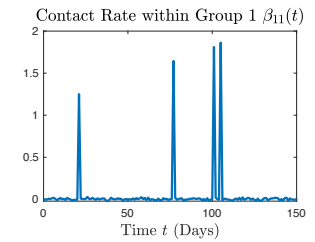


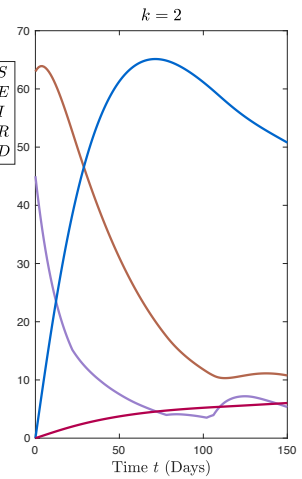
* De Luca et al, 2006. *Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm*

Epidemic Spread









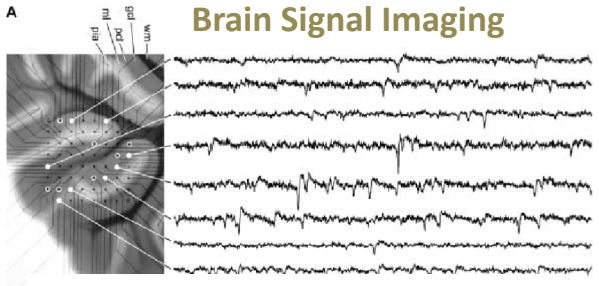
"Jump Stochastic Systems"

Systems with random and repetitive jumps.

Agile Quadrotor Flight

*O'Connell et al, 2022. *Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds*

Brain Signal Imaging



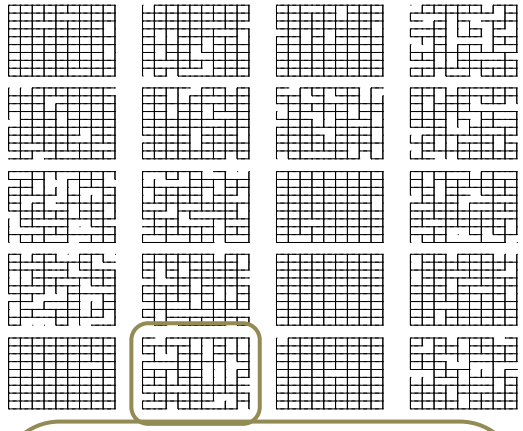
* Egert et al, 2002. *Two-dimensional monitoring of spiking networks in acute brain slices*

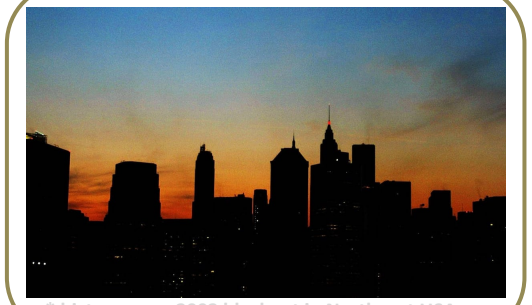
Spacecraft Control

* Yin et al, 2016. *A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System*

Fault-Tolerant Systems
(faults = external disturbances)

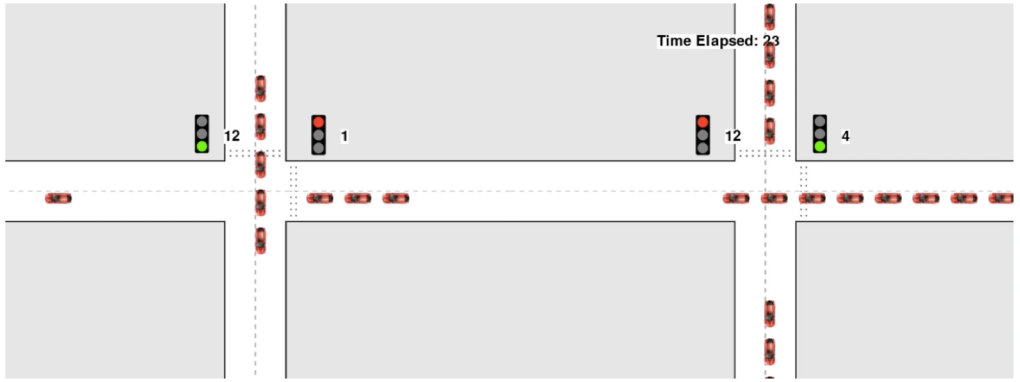
Power Grid



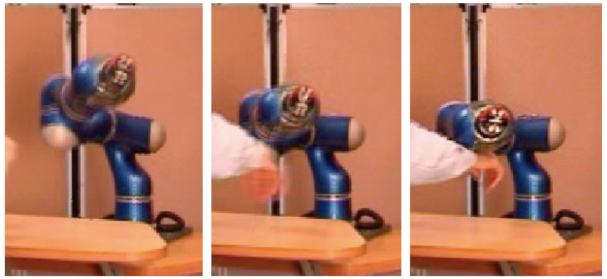


* history.com, 2003 blackout in Northeast USA

Vehicle Traffic Congestion

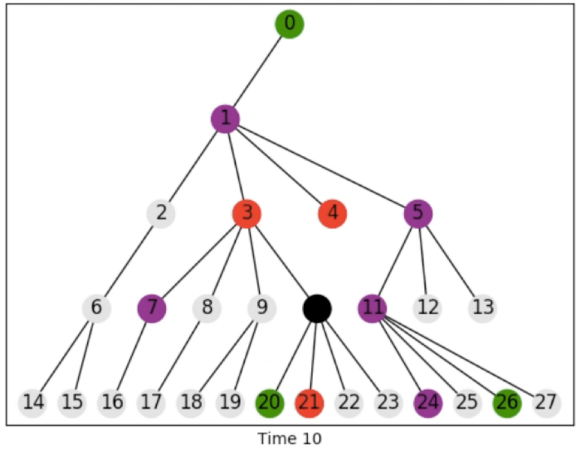


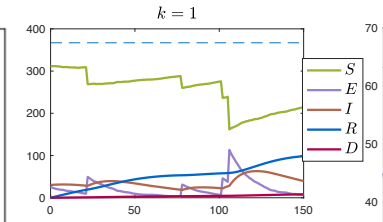
Robot Manipulator

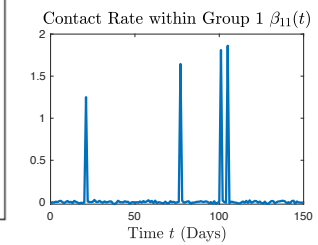


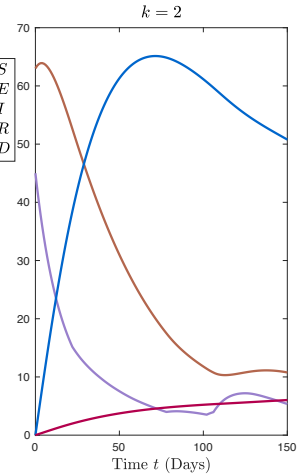
* De Luca et al, 2006. *Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm*

Epidemic Spread









“Jump Stochastic Systems”

Systems with random and repetitive jumps.

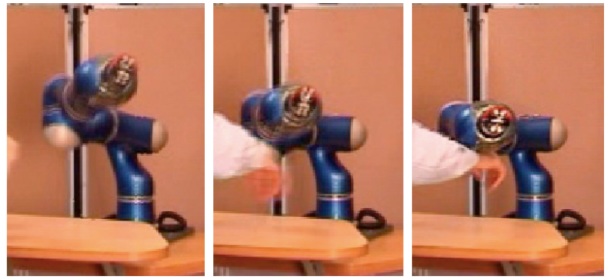
Agile Quadrotor Flight

*O’Connell et al, 2022. *Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds*

Spacecraft Control

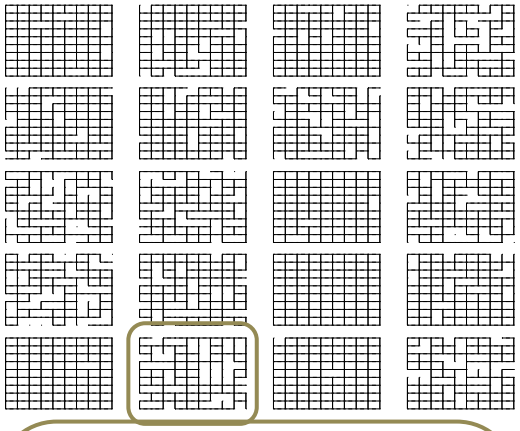
* Yin et al, 2016. *A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System*

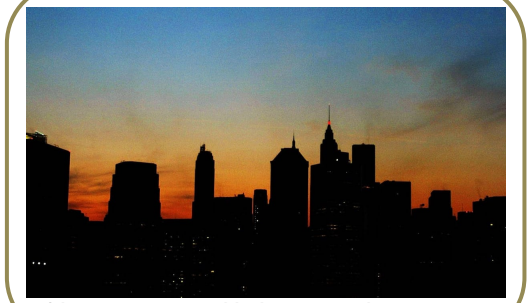
Robot Manipulator



* De Luca et al, 2006. *Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm*

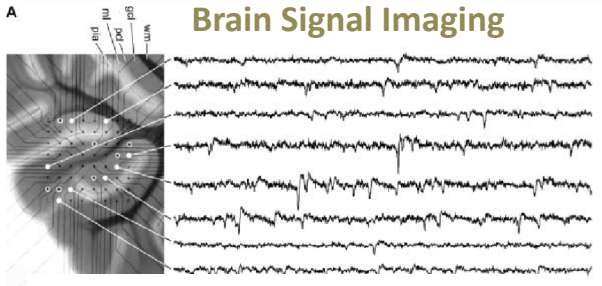
Power Grid





* history.com, 2003 blackout in Northeast USA

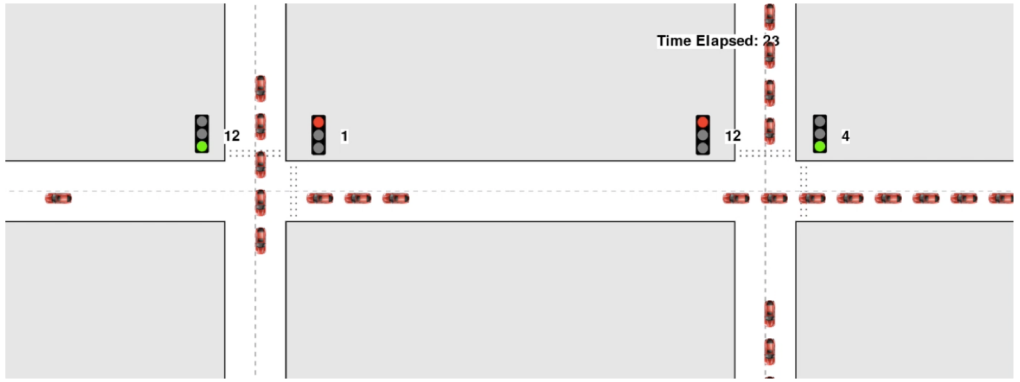
Brain Signal Imaging



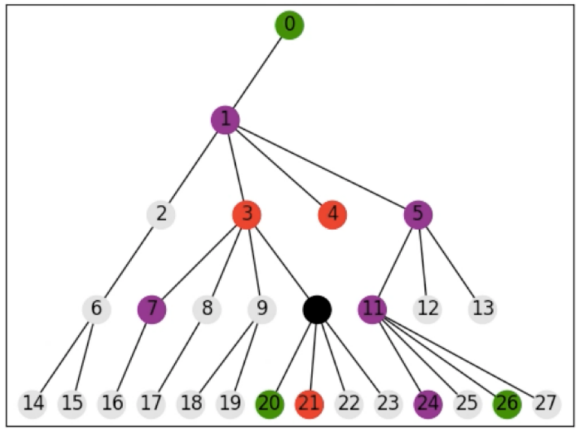
* Egert et al, 2002. *Two-dimensional monitoring of spiking networks in acute brain slices*

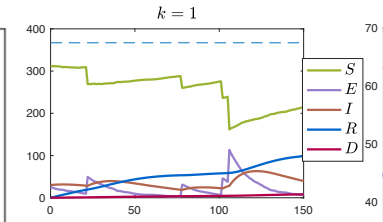
Event-Driven Systems
(jumps can be inherent)

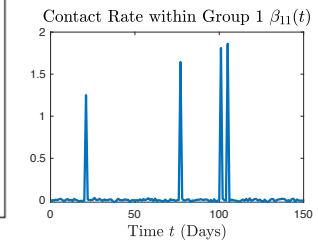
Vehicle Traffic Congestion

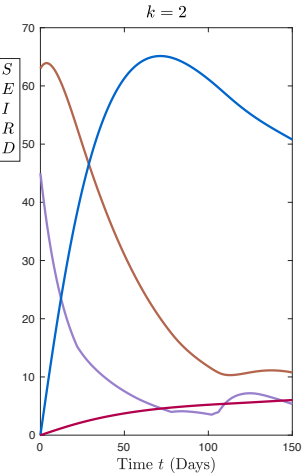


Epidemic Spread









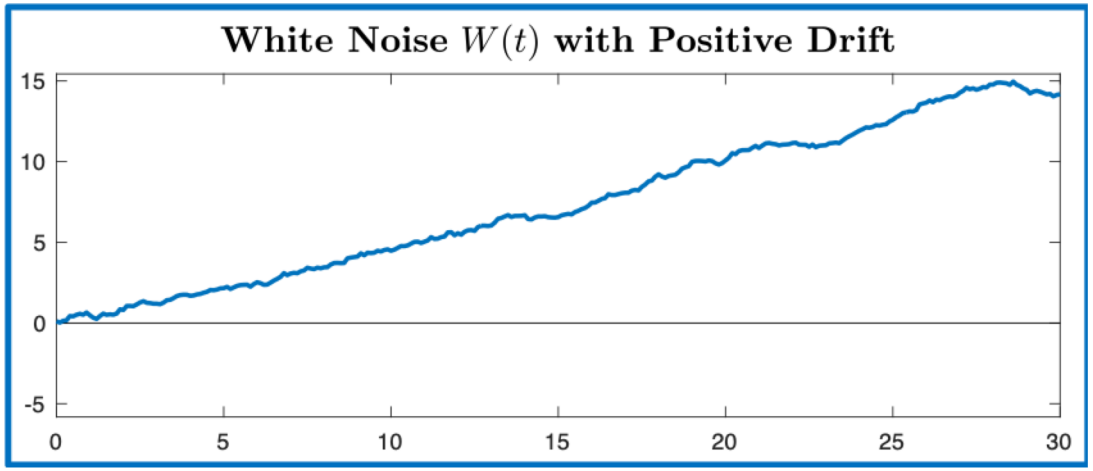
“Jump Stochastic Systems”

Poisson Shot Noise System:

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

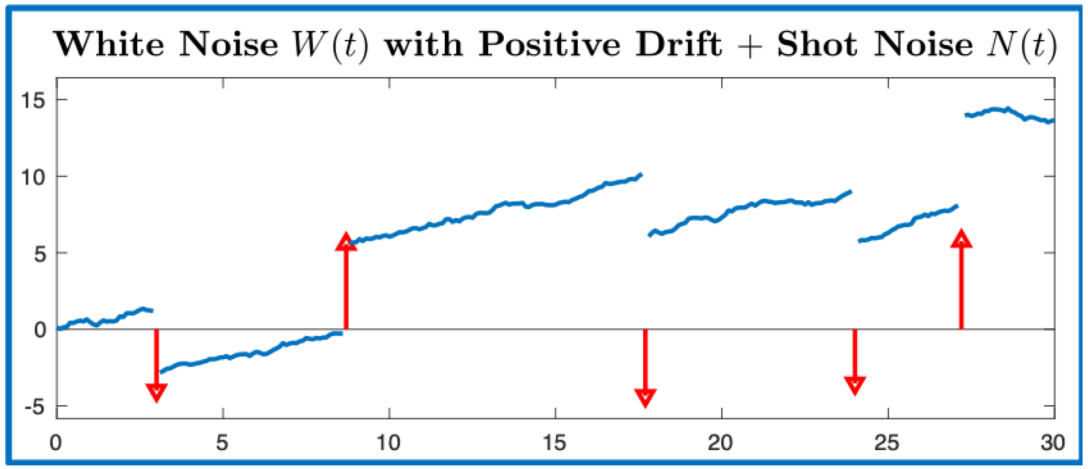
Gaussian White Noise System: Brownian motion

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



Lévy Noise System:

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



* SooJean Han and Soon-Jo Chung. *Incremental Nonlinear Stability Analysis of Stochastic Systems Perturbed by Lévy Noise*, International Journal of Robust and Nonlinear Control, 2022.

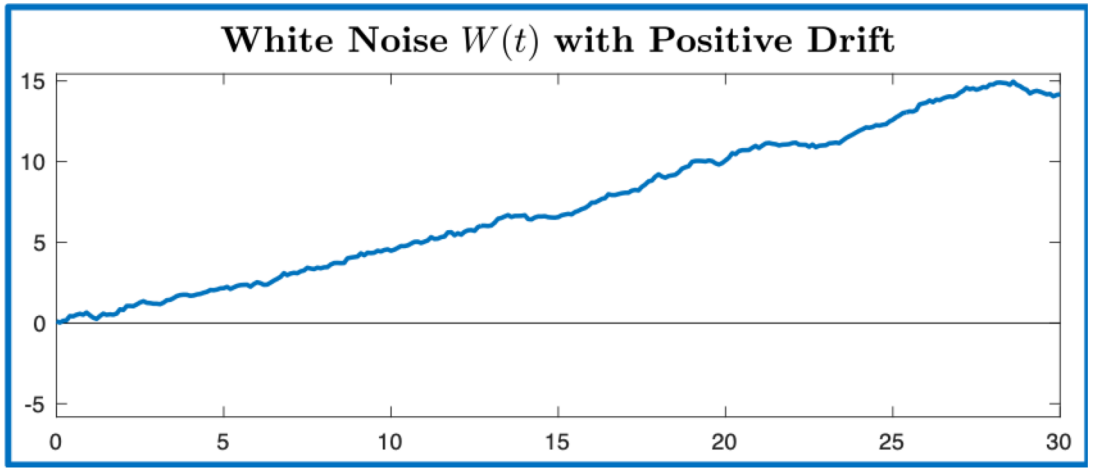
“Jump Stochastic Systems”

Poisson Shot Noise System: compound Poisson

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

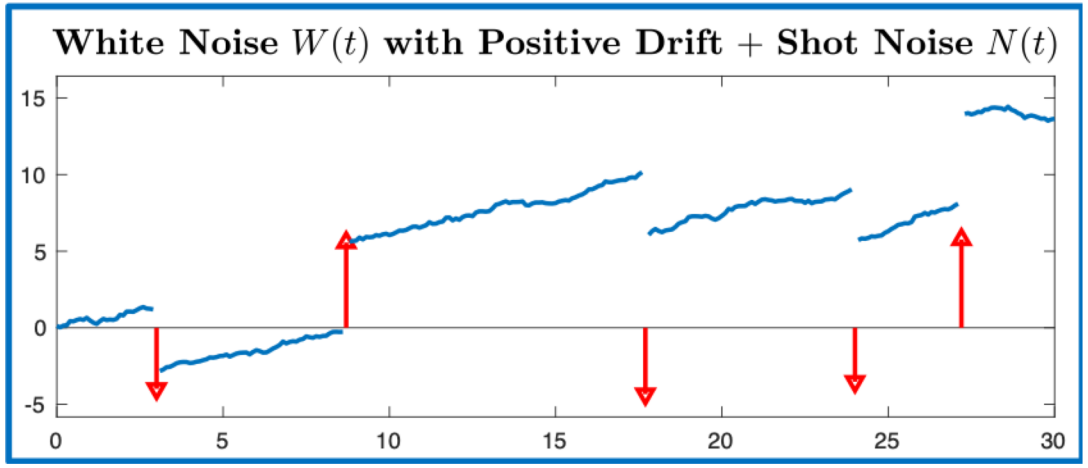
Gaussian White Noise System: Brownian motion

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



Lévy Noise System: Brownian motion + compound Poisson

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

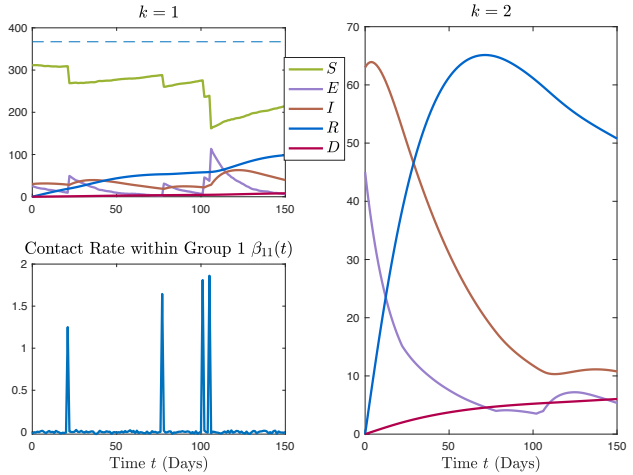


* SooJean Han and Soon-Jo Chung. *Incremental Nonlinear Stability Analysis of Stochastic Systems Perturbed by Lévy Noise*, International Journal of Robust and Nonlinear Control, 2022.

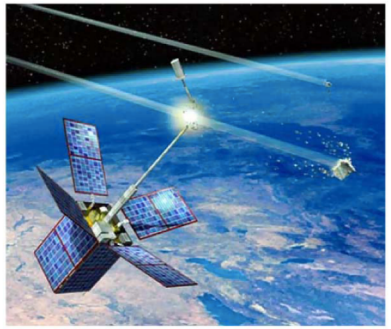
Stochastic Systems”

Agile Quadrotor Flight

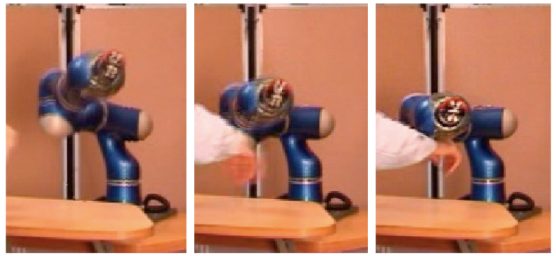
Epidemic Spread



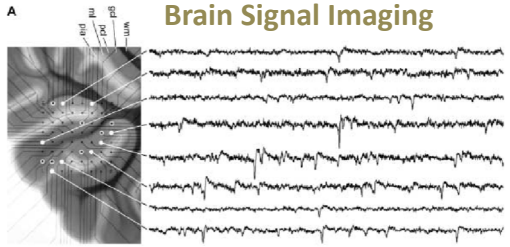
Spacecraft Control



Robot Manipulator



Brain Signal Imaging



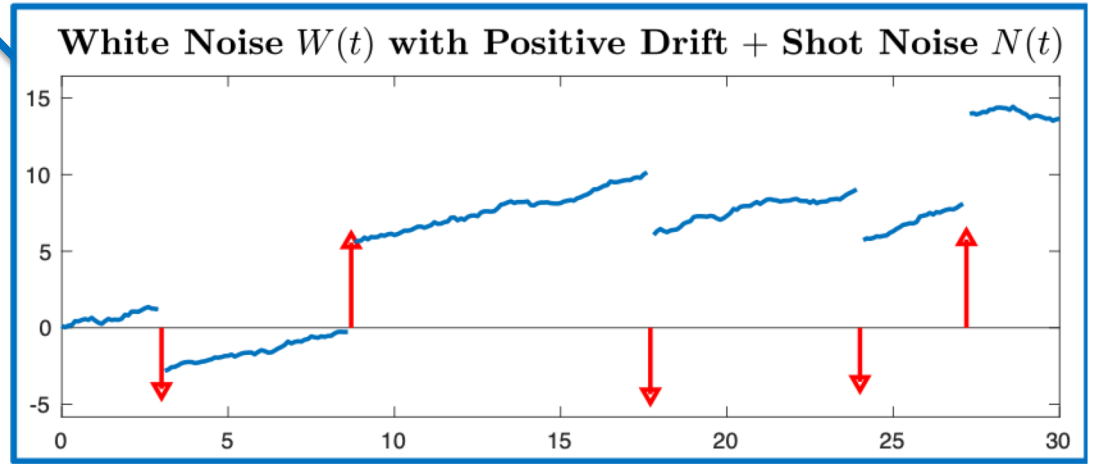
Many jump behaviors can be modeled with shot or Lévy noise

Poisson Shot Noise System: compound Poisson

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

Lévy Noise System: Brownian motion + compound Poisson

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



* SooJean Han and Soon-Jo Chung. *Incremental Nonlinear Stability Analysis of Stochastic Systems Perturbed by Lévy Noise*, International Journal of Robust and Nonlinear Control, 2022.

Infinitesimal generator: $G \in \mathcal{C}^{(1,2)}$

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

Stochastic Stability

(e.g., Kushner 1967, Mao 1990, Applebaum 2009)

Preliminary Question: what conditions are required for shot/Lévy noise to keep the system **stable**?

Let $\mathbf{x}(t), \mathbf{x}(0) = \mathbf{x}_0$ be a solution trajectory of a shot/Lévy noise stochastic system.

Design Lyapunov function $V(t, \mathbf{x}(t))$ such that $\mathcal{L}V \leq -\beta V(t, \mathbf{x}(t))$ for some $\beta > 0$.

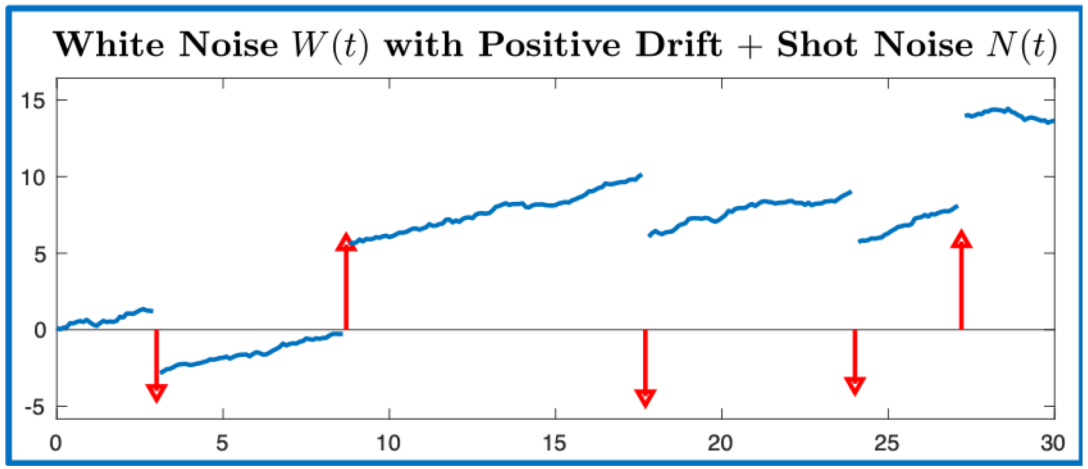
By Dynkin's formula*: $\mathbb{E}_{\mathbf{x}_0}[V(t, \mathbf{x}(t)] \leq V(0, \mathbf{x}_0)e^{-\beta t}$

Poisson Shot Noise System: compound Poisson

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

Lévy Noise System: Brownian motion + compound Poisson

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



* SooJean Han and Soon-Jo Chung. *Incremental Nonlinear Stability Analysis of Stochastic Systems Perturbed by Lévy Noise*, International Journal of Robust and Nonlinear Control, 2022.

* Eqn. (2-9) in Kushner 1967

Infinitesimal generator: $G \in \mathcal{C}^{(1,2)}$

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

Stochastic Stability

(e.g., Kushner 1967, Mao 1990, Applebaum 2009)

Preliminary Question: what conditions are required for shot/Lévy noise to keep the system **stable**?

Let $\mathbf{x}(t), \mathbf{x}(0) = \mathbf{x}_0$ be a solution trajectory of a shot/Lévy noise stochastic system.

Design Lyapunov function $V(t, \mathbf{x}(t))$ such that $\mathcal{L}V \leq -\beta V(t, \mathbf{x}(t))$ for some $\beta > 0$.

By Dynkin's formula*: $\mathbb{E}_{\mathbf{x}_0}[V(t, \mathbf{x}(t)] \leq V(0, \mathbf{x}_0)e^{-\beta t}$

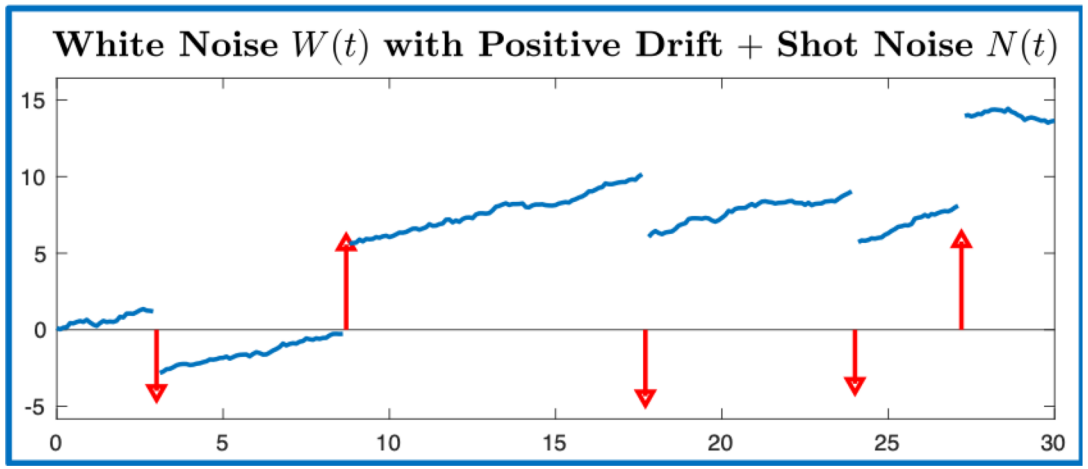
***Stochastic incremental stability* generalizes this inequality.**

Poisson Shot Noise System: compound Poisson

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

Lévy Noise System: Brownian motion + compound Poisson

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



* SooJean Han and Soon-Jo Chung. *Incremental Nonlinear Stability Analysis of Stochastic Systems Perturbed by Lévy Noise*, International Journal of Robust and Nonlinear Control, 2022.
* Dani, Chung, Hutchinson. *Observer design for stochastic nonlinear systems via contraction-based incremental stability*, IEEE Transactions on Automatic Control, 2015.
* Pham, Tabareau, Slotine. *A Contraction Theory Approach to Stochastic Incremental Stability*. IEEE Transactions on Automatic Control, 2007.

* Eqn. (2-9) in Kushner 1967

Control of Systems with Repeating Jump Patterns

Control for *jump stochastic systems* can be achieved by learning **patterns** in the system behavior.

At least 2 ways of improving performance:

- **memorize past patterns**
  → no re-computation of repeating control policies
- **predict future patterns**
  → schedule control policies in advance

Control of Systems with Repeating Jump Patterns

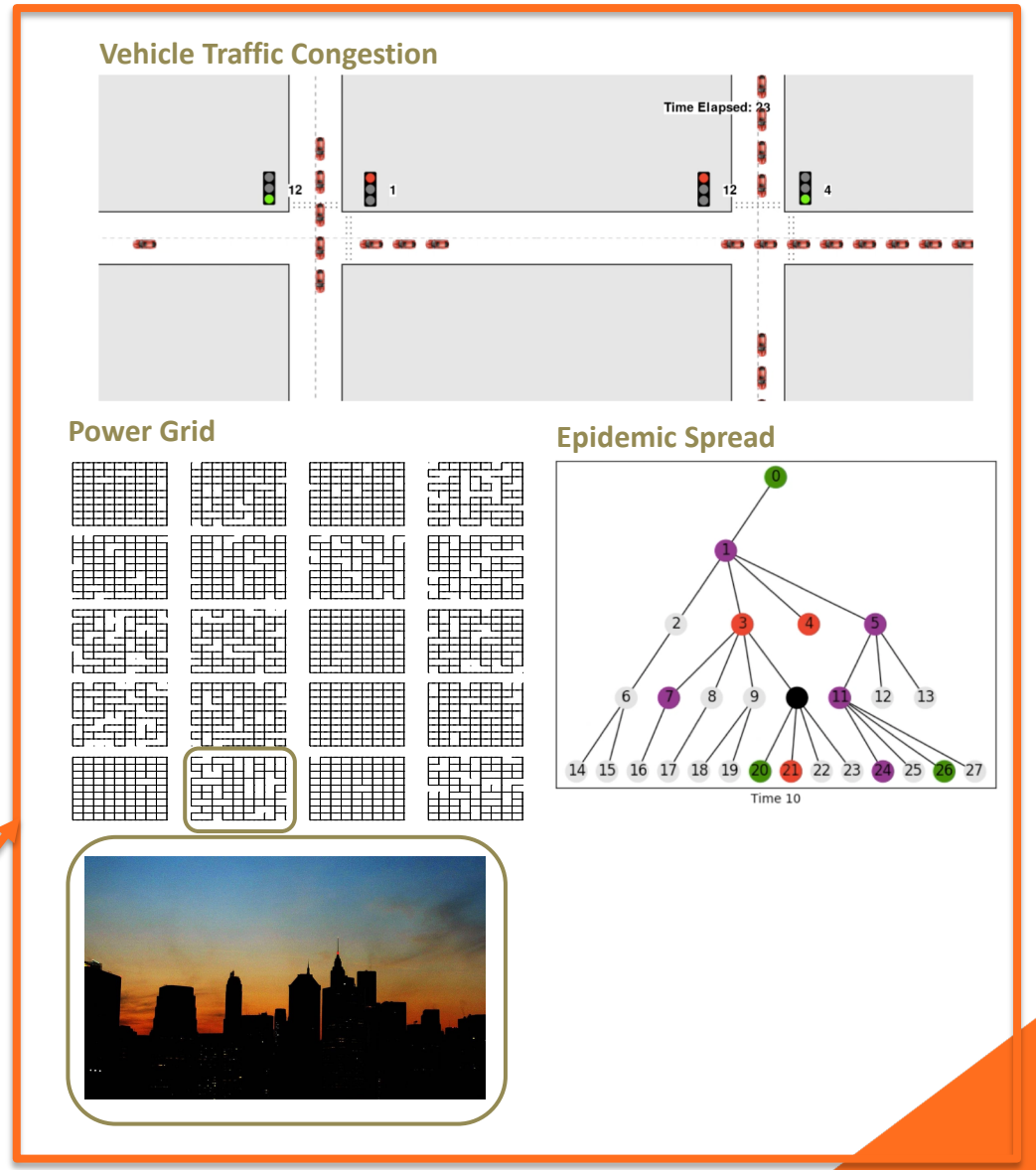
Control for *jump stochastic systems* can be achieved by learning **patterns** in the system behavior.

At least 2 ways of improving performance:

- **memorize past patterns**
  → no re-computation of repeating control policies
- **predict future patterns**
  → schedule control policies in advance

Systems that have jump patterns:

- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- Vehicle traffic congestion
  - **patterns** = intersection snapshots
- Epidemic spread
  - **patterns** = contact-tracing data, travel routines



Discrete-Time Linear Markovian Jump Systems

$$T_n \triangleq \min\{s \in \mathbb{N} \,|\, N[s] = n\}$$

$$\|\mathbf{w}[t]\|_\infty \leq \overline{w}$$

$$\{\xi_n\}_{n=1}^{\infty}, \quad \xi_n : \Omega \to \mathcal{X} \triangleq \{1, \cdots, M\}$$

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

$$\mathbf{p}_{n+1}^\top = \mathbf{p}_n^\top P \qquad \text{Given } \xi_n = \varphi_n, \quad \xi_{n+1} = m \text{ w.p. } P[\varphi_n, m], m \in \mathcal{X}$$

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Discrete-Time Linear Markovian Jump Systems

$$T_n \triangleq \min\{s \in \mathbb{N} \,|\, N[s] = n\}$$

$$\|\mathbf{w}[t]\|_\infty \leq \overline{w}$$

"mode-index"

$$\{\xi_n\}_{n=1}^{\infty}, \quad \xi_n : \Omega \to \mathcal{X} \triangleq \{1, \cdots, M\}$$

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

$$\mathbf{p}_{n+1}^\top = \mathbf{p}_n^\top P \qquad \text{Given } \xi_n = \varphi_n, \;\; \xi_{n+1} = m \text{ w.p. } P[\varphi_n, m], m \in \mathcal{X}$$

Assumption: Mode process $\{\xi_n\}$ operates on a timescale which is $\Delta T \in \mathbb{N}$ times longer than the timescale of the system, i.e. if $N[t] = n$, then $N[t + a\Delta T] = n + a$ for any $a \in \mathbb{N}$.

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Discrete-Time Linear Markovian Jump Systems

$$T_n \triangleq \min\{s \in \mathbb{N} \,|\, N[s] = n\}$$

$$\|\mathbf{w}[t]\|_\infty \leq \overline{w}$$

"mode-index"

$$\{\xi_n\}_{n=1}^{\infty}, \quad \xi_n : \Omega \to \mathcal{X} \triangleq \{1, \cdots, M\}$$

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

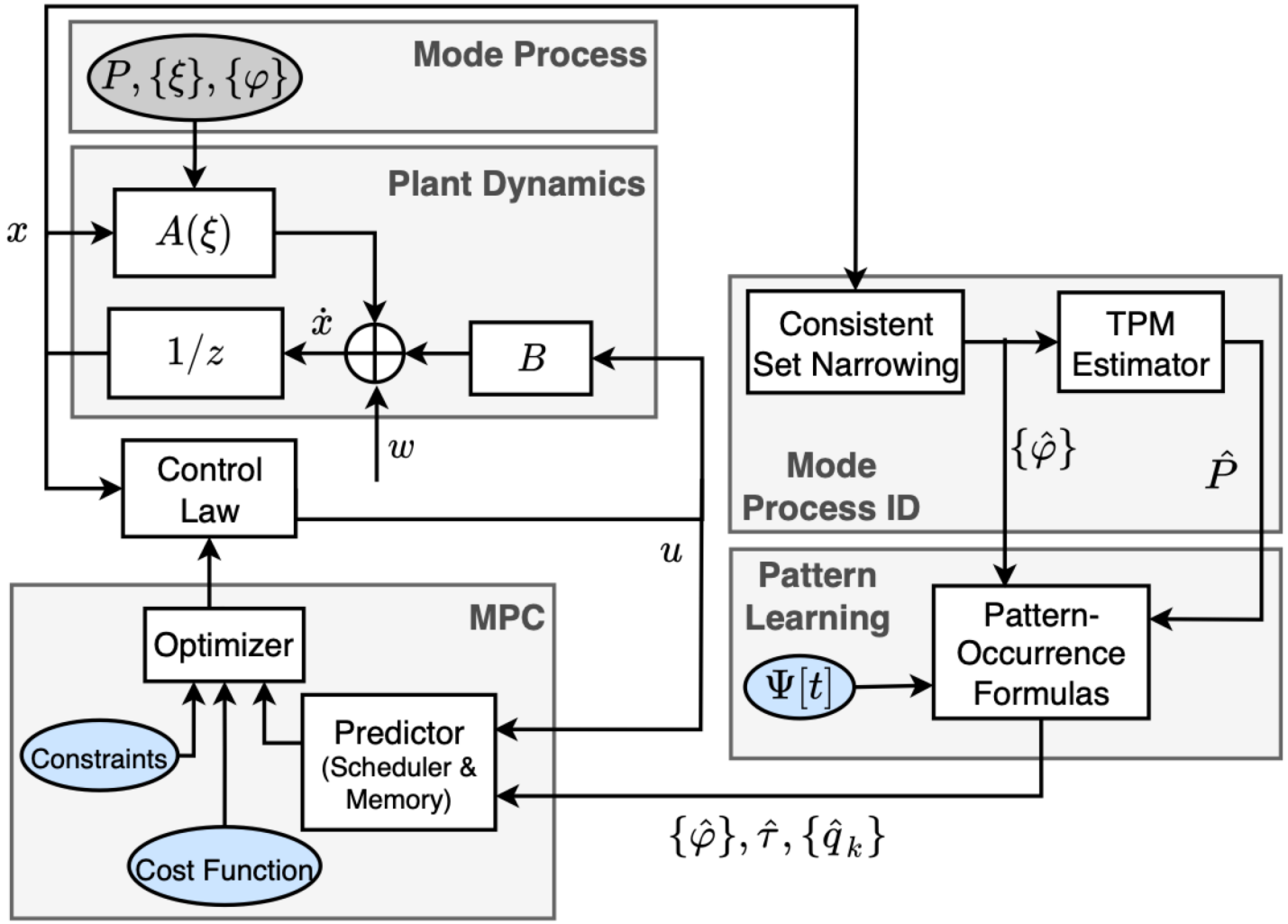
$$\mathbf{p}_{n+1}^\top = \mathbf{p}_n^\top P \qquad \text{Given } \xi_n = \varphi_n, \;\; \xi_{n+1} = m \text{ w.p. } P[\varphi_n, m], m \in \mathcal{X}$$

Assumption: Mode process $\{\xi_n\}$ operates on a timescale which is $\Delta T \in \mathbb{N}$ times longer than the timescale of the system, i.e. if $N[t] = n$, then $N[t + a\Delta T] = n + a$ for any $a \in \mathbb{N}$.

- mode process $\{\xi_n\}$ unobservable.
- $\mathcal{X}$, matrices $\{A(1), \cdots, A(M)\}$ and B, initial mode ξ_0 are known.
- sparsity pattern of TPM P known, but transition probabilities unknown.

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Proposed Controller Framework

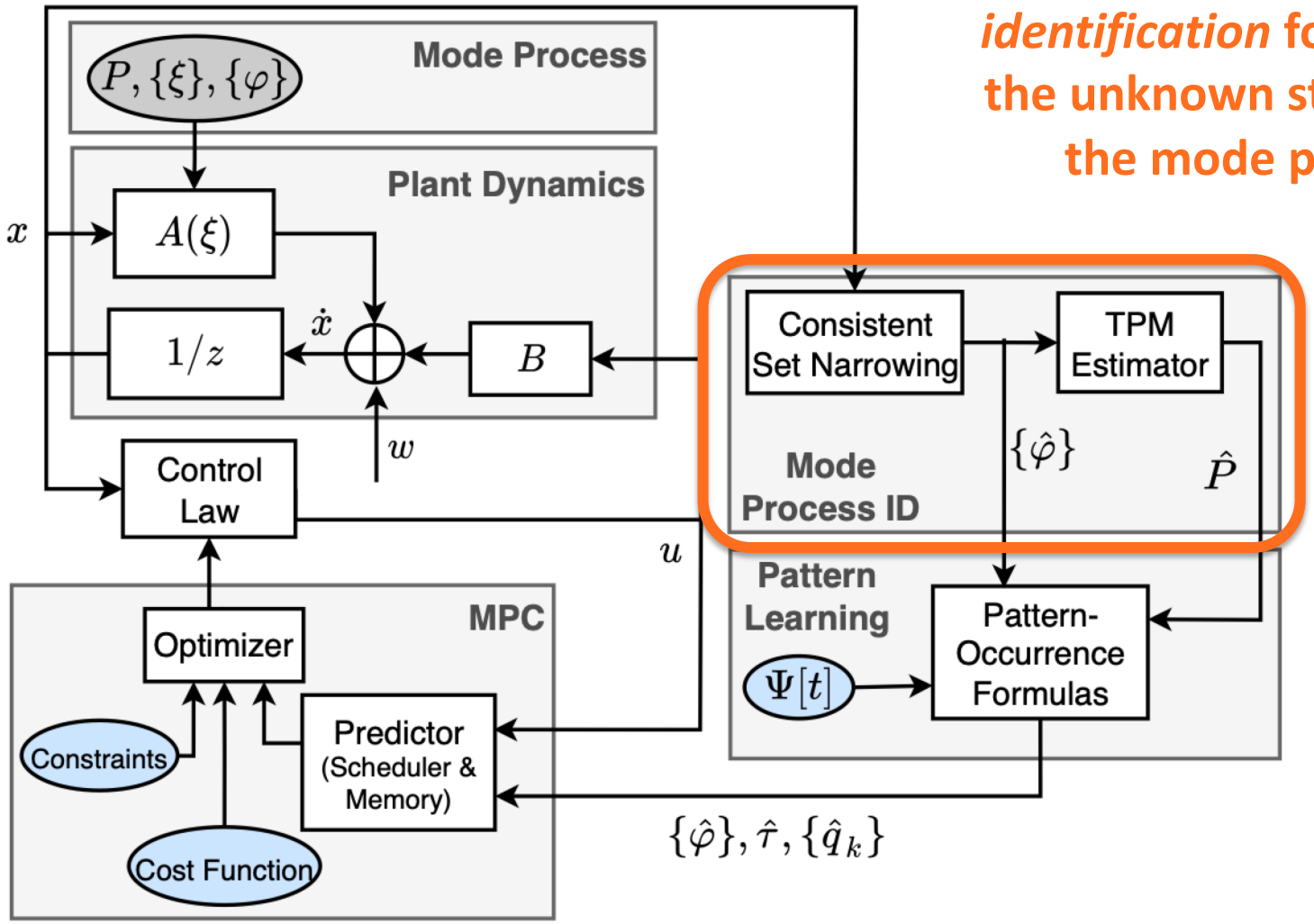


* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Caltech

Mode Process Identification (ID)

Mode process identification **for learning the unknown statistics of the mode process**



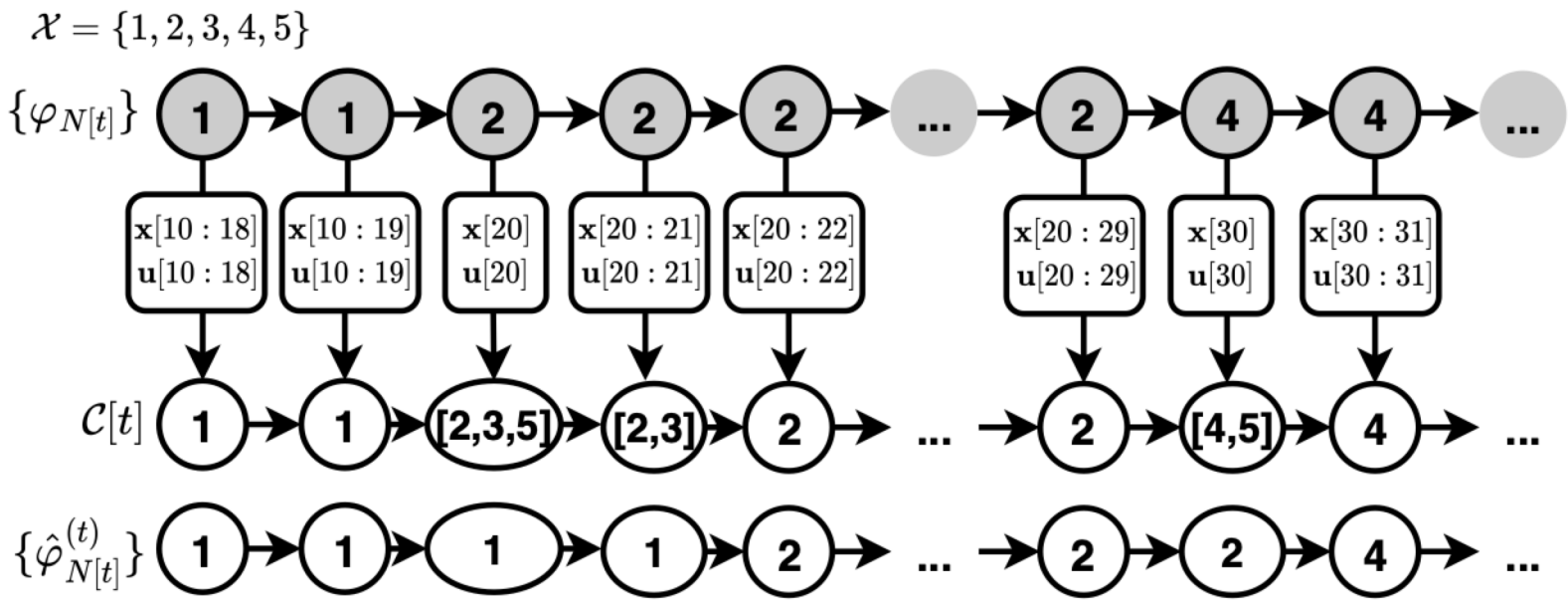
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Caltech

Mode Process Identification (ID)

Assumption: Mode process $\{\xi_n\}$ operates on a timescale which is $\Delta T \in \mathbb{N}$ times longer than the timescale of the system, i.e. if $N[t] = n$, then $N[t + a\Delta T] = n + a$ for any $a \in \mathbb{N}$.

$\Delta T = 10$
$\mathrm{M} = 5$



Goal: Use observations $\mathbf{x}[T_{N[t]} : t], \mathbf{u}[T_{N[t]} : t]$ to estimate:

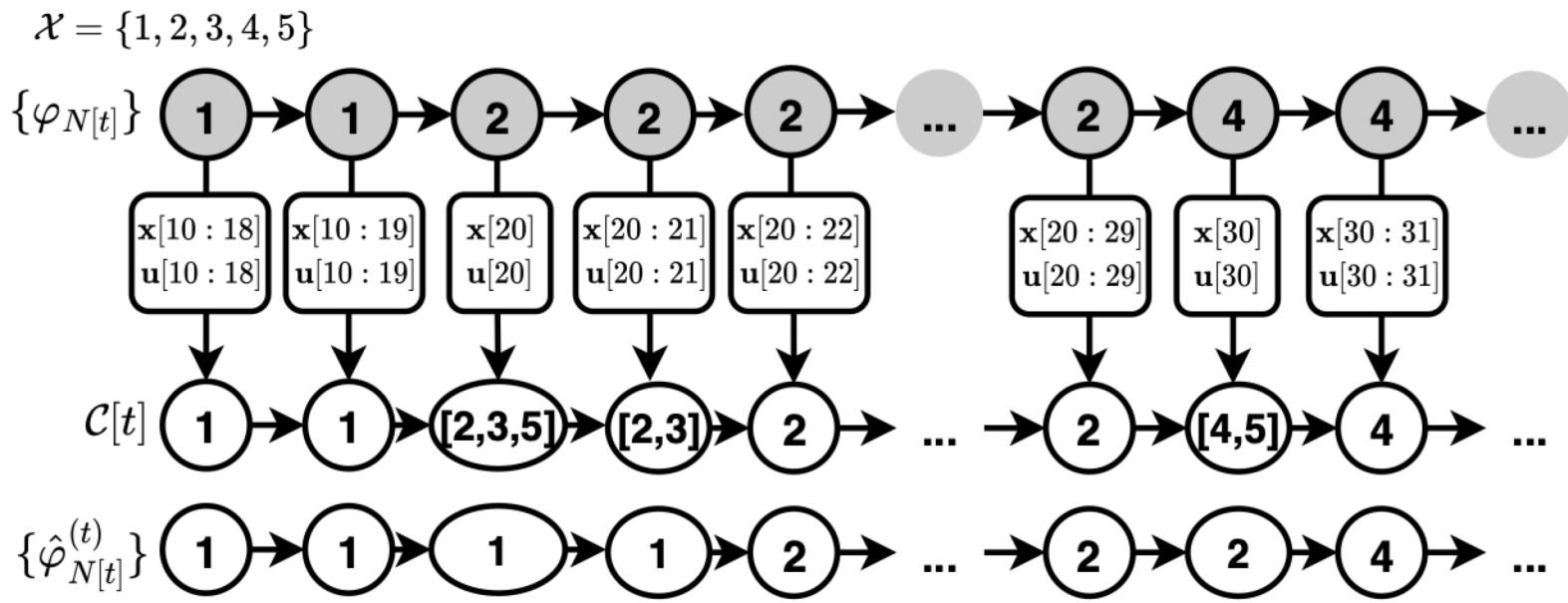
$\hat{P}^{(t)}$ of the true TPM P. (e.g., Metropolis-Hastings, Gibbs' sampling, or Baum-Welch)

$\hat{\varphi}^{(t)}_n$ of the true current mode φ_n. (e.g., Viterbi's algorithm)

Mode Process Identification (ID)

Assumption: Mode process $\{\xi_n\}$ operates on a timescale which is $\Delta T \in \mathbb{N}$ times longer than the timescale of the system, i.e. if $N[t] = n$, then $N[t + a\Delta T] = n + a$ for any $a \in \mathbb{N}$.

$\Delta T = 10$
$\mathrm{M} = 5$

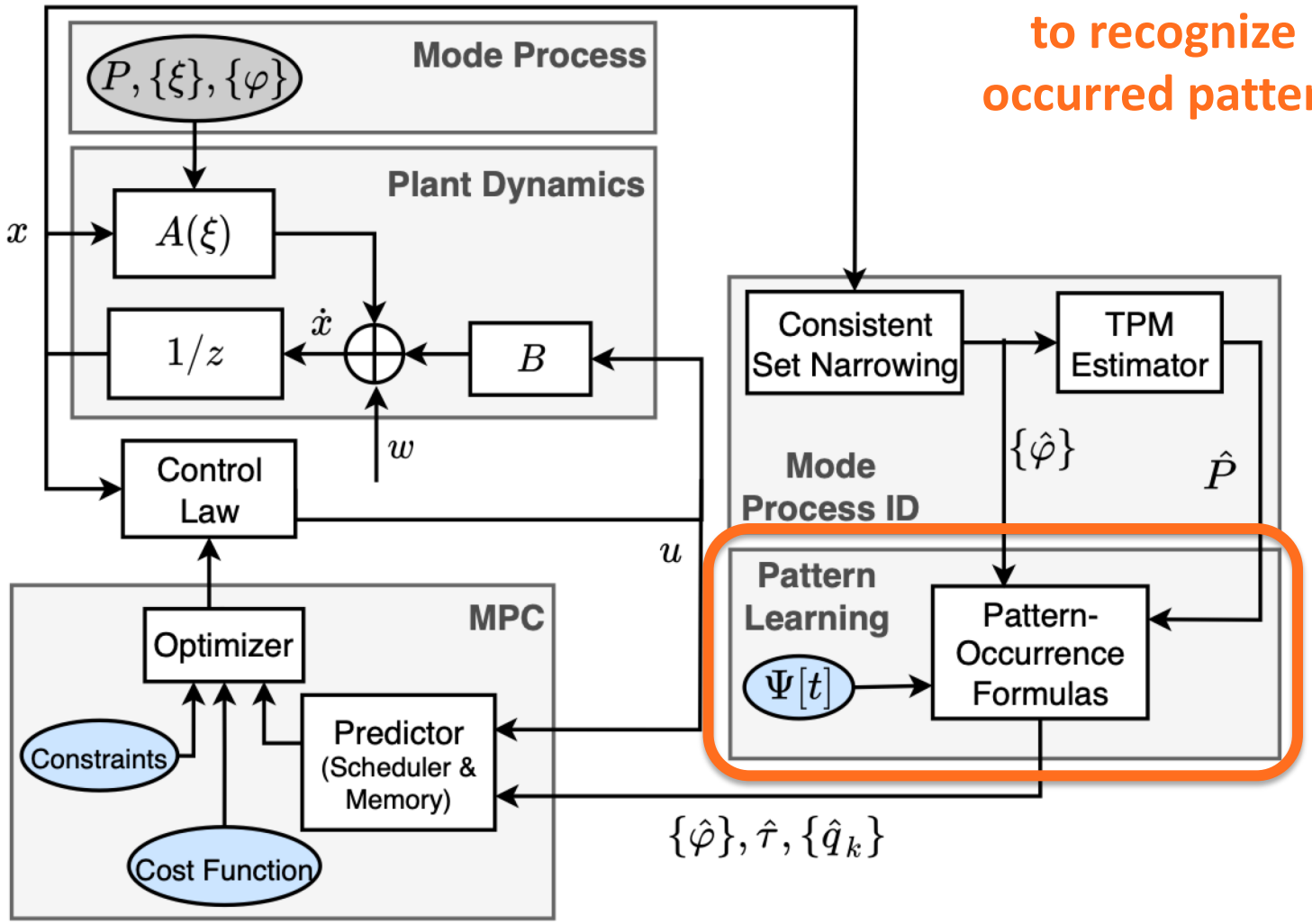


Consistent Set Narrowing:

$$\mathcal{C}[0] \triangleq \mathcal{X},\ \mathcal{C}[t] = \left\{ m \in \mathcal{C}[t-1] \mid \bigwedge_{r=T_n}^{t-1} \mathbb{1}\{\|\mathbf{x}[r+1] - A(m)\mathbf{x}[r] - B\mathbf{u}[r]\|_\infty \leq \overline{w}\} \right\}$$

$$\mathbf{x}[t+1] = A(\xi_{N[t]})\mathbf{x}[t] + B\mathbf{u}[t] + \mathbf{w}[t] \quad \|\mathbf{w}[t]\|_\infty \leq \overline{w}$$

Pattern-Learning in the Mode Process



Pattern-learning component **to recognize previously-occurred patterns of events.**

Pattern: a single mode or a sequence of modes over time which is "interesting to the user", e.g., faults in fault-tolerance control

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Caltech

Pattern-Learning in the Mode Process

Definition 1 (Patterns). *Collection of patterns* $\Psi \triangleq \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_K\}$, where each $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,L})$ is a *(mode) pattern* with constant length $L \in \mathbb{N}$ and elements $\psi_{k,j} \in \mathcal{X}$.

Pattern-Learning in the Mode Process

Definition 1 (Patterns). *Collection of patterns* $\Psi \triangleq \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_K\}$, where each $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,L})$ is a *(mode) pattern* with constant length $L \in \mathbb{N}$ and elements $\psi_{k,j} \in \mathcal{X}$.

Definition 2 (Feasible Sequence). $(\alpha_1, \cdots, \alpha_a)$ is *feasible* if for all $i \in \{1, \cdots, a-1\}$, $P[\alpha_i, \alpha_{i+1}] > 0$.

Pattern-Learning in the Mode Process

Definition 1 (Patterns). *Collection of patterns* $\Psi \triangleq \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_K\}$, where each $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,L})$ is a *(mode) pattern* with constant length $L \in \mathbb{N}$ and elements $\psi_{k,j} \in \mathcal{X}$.

Definition 2 (Feasible Sequence). $(\alpha_1, \cdots, \alpha_a)$ is *feasible* if for all $i \in \{1, \cdots, a-1\}$, $P[\alpha_i, \alpha_{i+1}] > 0$.

Definition 3 (Pattern Occurrence Times). Let $n \triangleq N[t] \in \mathbb{N}$ be the current mode-index at current time $t \in \mathbb{N}$, and suppose $\xi_n = \hat{\varphi}_n^{(t)}$. *Pattern occurrence times* for each $k \in \{1, \cdots, K\}$:

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

Pattern-Learning in the Mode Process

Definition 1 (Patterns). *Collection of patterns* $\Psi \triangleq \{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_K\}$, where each $\boldsymbol{\psi}_k \triangleq (\psi_{k,1}, \cdots, \psi_{k,L})$ is a *(mode) pattern* with constant length $L \in \mathbb{N}$ and elements $\psi_{k,j} \in \mathcal{X}$.

Definition 2 (Feasible Sequence). $(\alpha_1, \cdots, \alpha_a)$ is *feasible* if for all $i \in \{1, \cdots, a-1\}$, $P[\alpha_i, \alpha_{i+1}] > 0$.

Definition 3 (Pattern Occurrence Times). Let $n \triangleq N[t] \in \mathbb{N}$ be the current mode-index at current time $t \in \mathbb{N}$, and suppose $\xi_n = \hat{\varphi}_n^{(t)}$. *Pattern occurrence times* for each $k \in \{1, \cdots, K\}$:

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

Problem (Pattern-Learning). Given Ψ, for each time $t \in \mathbb{N}$ and $n \triangleq N[t]$, derive estimates:

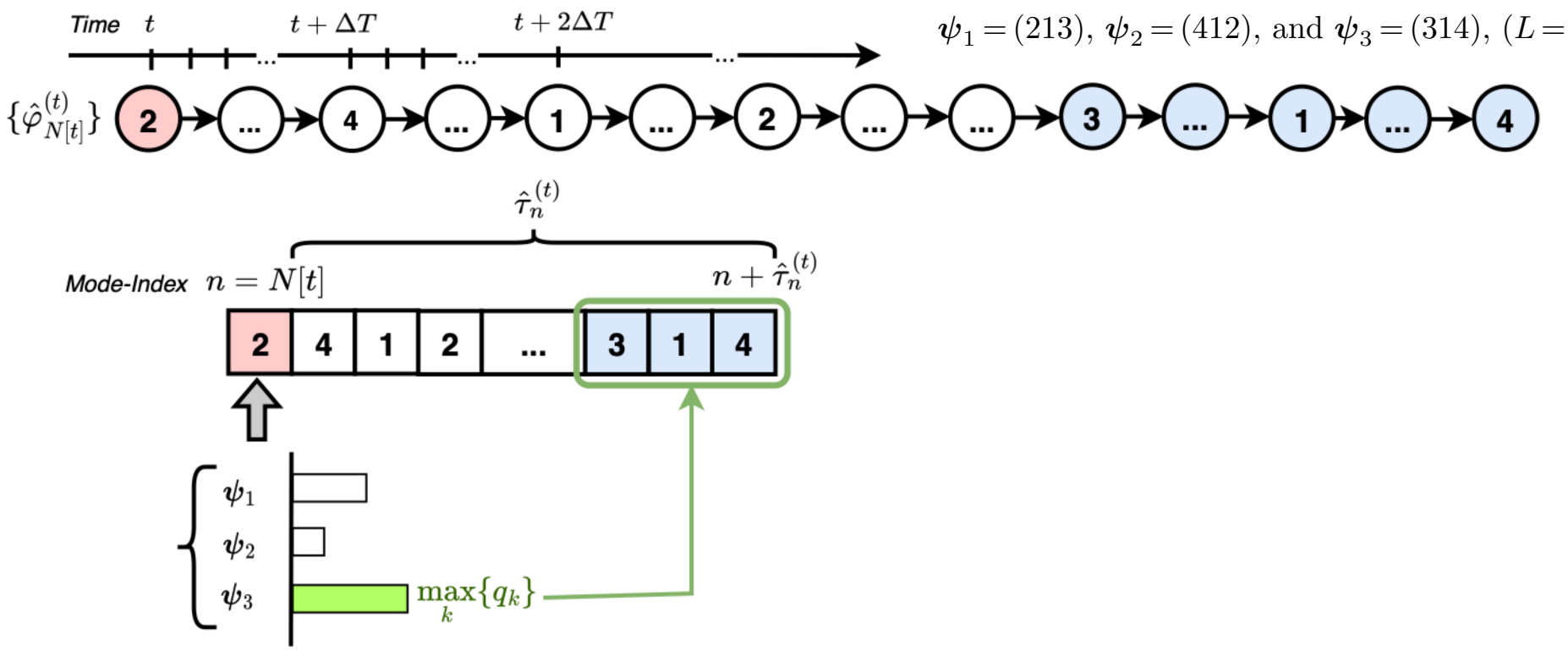
- $\mathbb{E}[\hat{\tau}_n^{(t)}]$ of the *mean minimum occurrence time*, where $\hat{\tau}_n^{(t)} \triangleq \min_{k \in \{1, \cdots, K\}} \hat{\tau}_{k|n}^{(t)}$.
- $\{\hat{q}_k^{(t)}\}_{k=1}^K$ of the *first-occurrence probabilities*, where $\hat{q}_k^{(t)} \triangleq \mathbb{P}(\hat{\tau}_n^{(t)} = \hat{\tau}_{k|n}^{(t)})$.

Pattern-Learning in the Mode Process

Problem (Pattern-Learning). Given Ψ, for each time $t \in \mathbb{N}$ and $n \triangleq N[t]$, derive estimates:

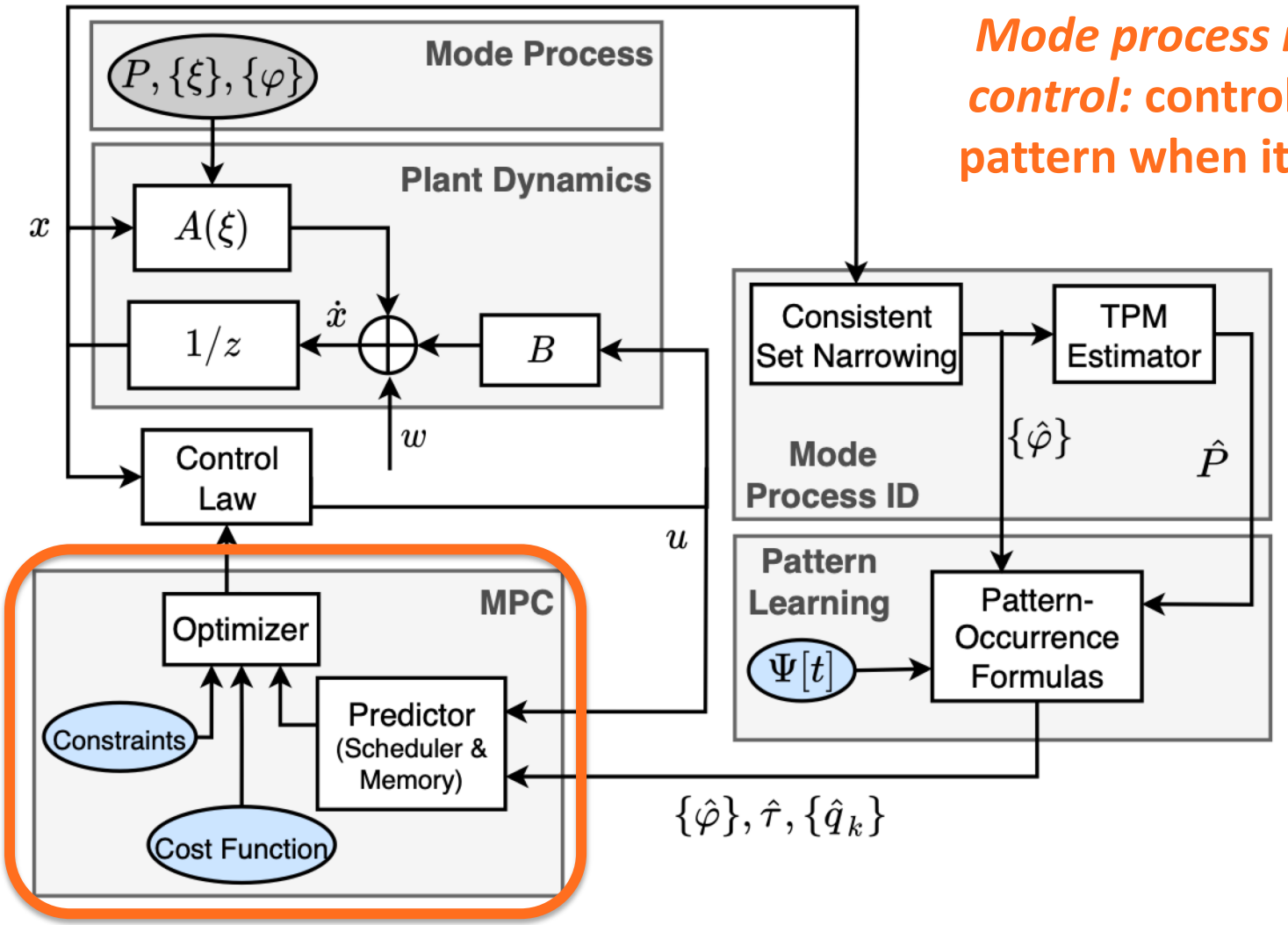
- $\mathbb{E}[\hat{\tau}_n^{(t)}]$ of the *mean minimum occurrence time*, where $\hat{\tau}_n^{(t)} \triangleq \min_{k\in\{1,\cdots,K\}} \hat{\tau}_{k|n}^{(t)}$.
- $\{\hat{q}_k^{(t)}\}_{k=1}^K$ of the *first-occurrence probabilities*, where $\hat{q}_k^{(t)} \triangleq \mathbb{P}(\hat{\tau}_n^{(t)} = \hat{\tau}_{k|n}^{(t)})$.

Collection $\Psi = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3\}$, $(K = 3)$.
$\boldsymbol{\psi}_1 = (213)$, $\boldsymbol{\psi}_2 = (412)$, and $\boldsymbol{\psi}_3 = (314)$, $(L = 3)$.



Caltech

Mode Process Model Predictive Control (MPC)



Mode process model predictive control: **control law design for a pattern when it is first observed.**

(Implementation varies by application.)

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Caltech

Mode Process Model Predictive Control (MPC)

Estimated current mode: $\hat{\varphi}_{N[t]}^{(t)} = m \in \mathcal{X}$. Constant-gain state-feedback control law $\mathbf{u}[t] = K(t,m)\mathbf{x}[t]$

E.g., obtained via minimizing standard LQR cost

$$J(t,m) = \sum_{s=t}^{T} \left(\mathbf{x}[s]^\top Q(m)\mathbf{x}[s] + \mathbf{u}[s]^\top R(m)\mathbf{u}[s]\right) + \mathbf{x}[T]^\top Q_f(m)\mathbf{x}[T],\ T \triangleq T_{N[t]+1} - 1$$

Mode Process Model Predictive Control (MPC)

Estimated current mode: $\hat{\varphi}^{(t)}_{N[t]} = m \in \mathcal{X}$. Constant-gain state-feedback control law $\mathbf{u}[t] = K(t,m)\mathbf{x}[t]$

E.g., obtained via minimizing standard LQR cost

$$J(t,m) = \sum_{s=t}^{T} \left(\mathbf{x}[s]^\top Q(m)\mathbf{x}[s] + \mathbf{u}[s]^\top R(m)\mathbf{u}[s]\right) + \mathbf{x}[T]^\top Q_f(m)\mathbf{x}[T],\ T \triangleq T_{N[t]+1} - 1$$

Time-varying collection of patterns: choose set of feasible length-L future sequences of modes for some chosen *future horizon* $L \in \mathbb{N}$.

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

Mode Process Model Predictive Control (MPC)

Estimated current mode: $\hat{\varphi}^{(t)}_{N[t]} = m \in \mathcal{X}$. Constant-gain state-feedback control law $\mathbf{u}[t] = K(t,m)\mathbf{x}[t]$

E.g., obtained via minimizing standard LQR cost

$$J(t,m) = \sum_{s=t}^{T} \left(\mathbf{x}[s]^\top Q(m)\mathbf{x}[s] + \mathbf{u}[s]^\top R(m)\mathbf{u}[s]\right) + \mathbf{x}[T]^\top Q_f(m)\mathbf{x}[T],\ T \triangleq T_{N[t]+1} - 1$$

Time-varying collection of patterns: choose set of feasible length-L future sequences of modes for some chosen *future horizon* $L \in \mathbb{N}$.

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

Let $k \triangleq \text{argmax}_{k'} \hat{q}^{(t)}_{k'}$ and mode-index $\tau \equiv \mathbb{E}[\hat{\tau}^{(t)}_n]$.

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

$$\vdots$$

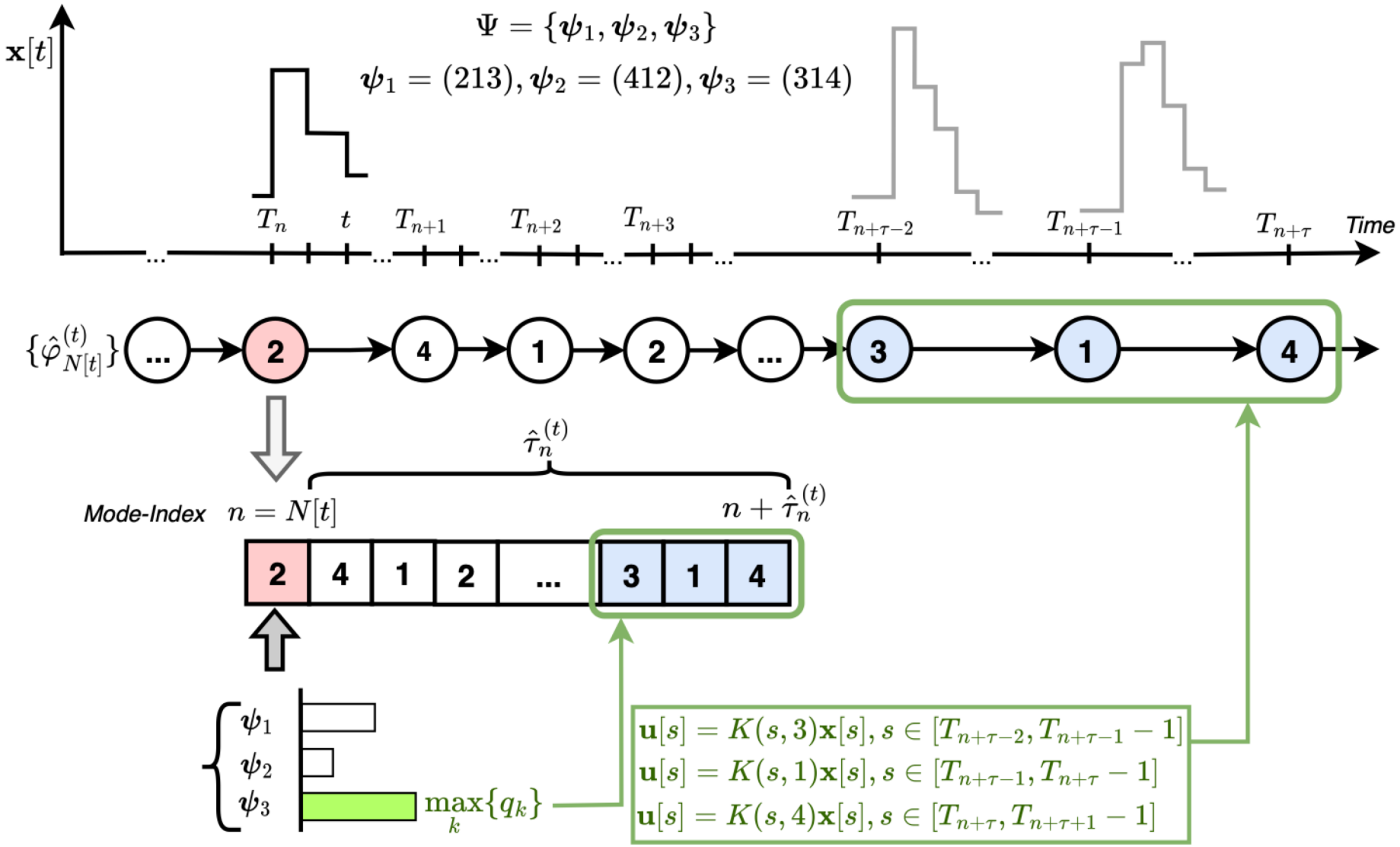
$$\mathbf{u}[s] = K(s, \psi_{k,L})\mathbf{x}[s], \quad s \in [T_{n+\tau} : T_{n+\tau+1} - 1]$$

Only apply control law for first predicted mode.

Then repeat at time T_{n+1}.

Mode Process Model Predictive Control (MPC)

$\Delta T = 10$
$\mathrm{M} = 5$



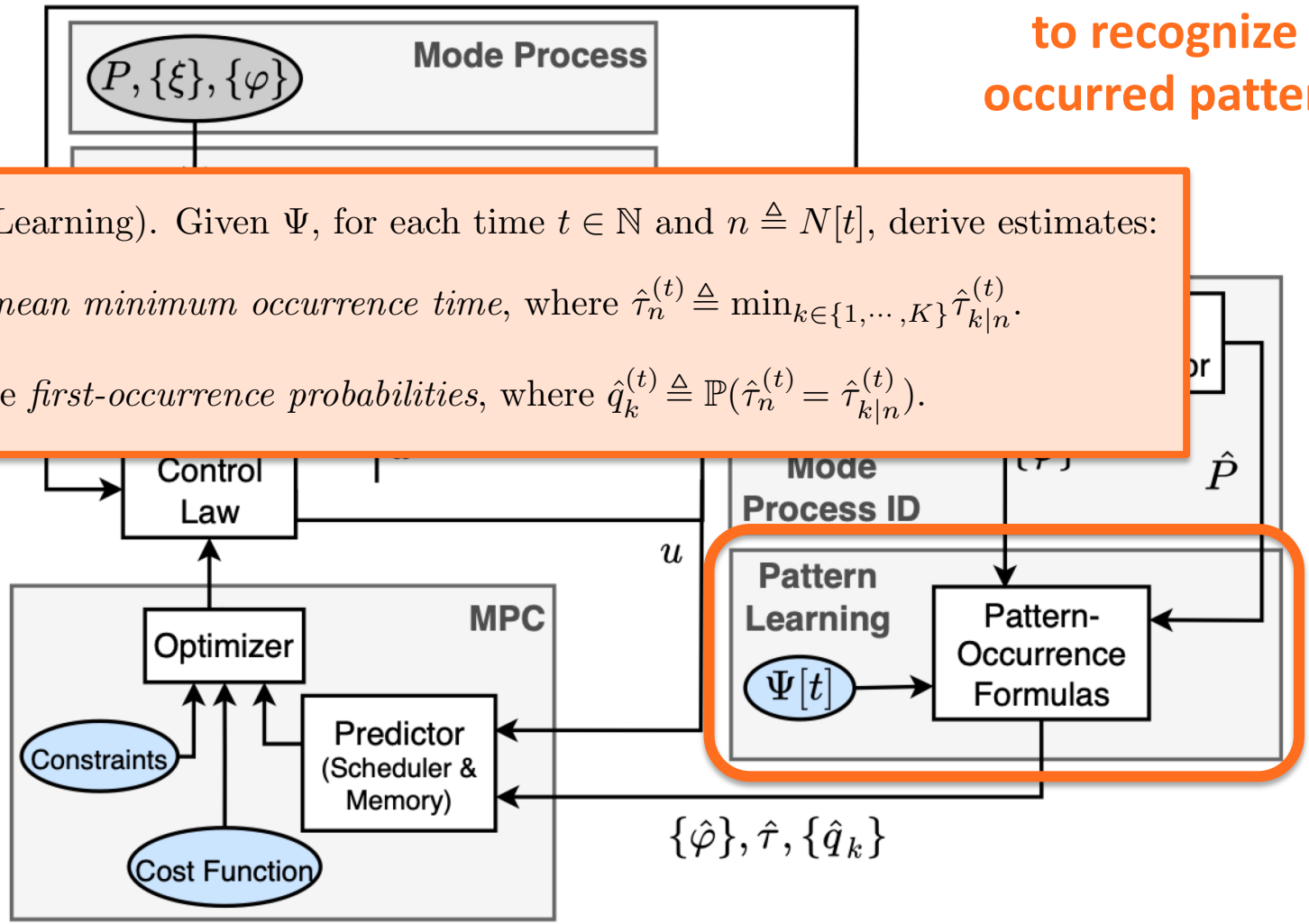
Caltech

Solving the Pattern-Occurrence Problem

Pattern-learning component **to recognize previously-occurred patterns of events.**

Problem (Pattern-Learning). Given Ψ, for each time $t \in \mathbb{N}$ and $n \triangleq N[t]$, derive estimates:

- $\mathbb{E}[\hat{\tau}_n^{(t)}]$ of the *mean minimum occurrence time*, where $\hat{\tau}_n^{(t)} \triangleq \min_{k\in\{1,\cdots,K\}} \hat{\tau}_{k|n}^{(t)}$.
- $\{\hat{q}_k^{(t)}\}_{k=1}^K$ of the *first-occurrence probabilities*, where $\hat{q}_k^{(t)} \triangleq \mathbb{P}(\hat{\tau}_n^{(t)} = \hat{\tau}_{k|n}^{(t)})$.



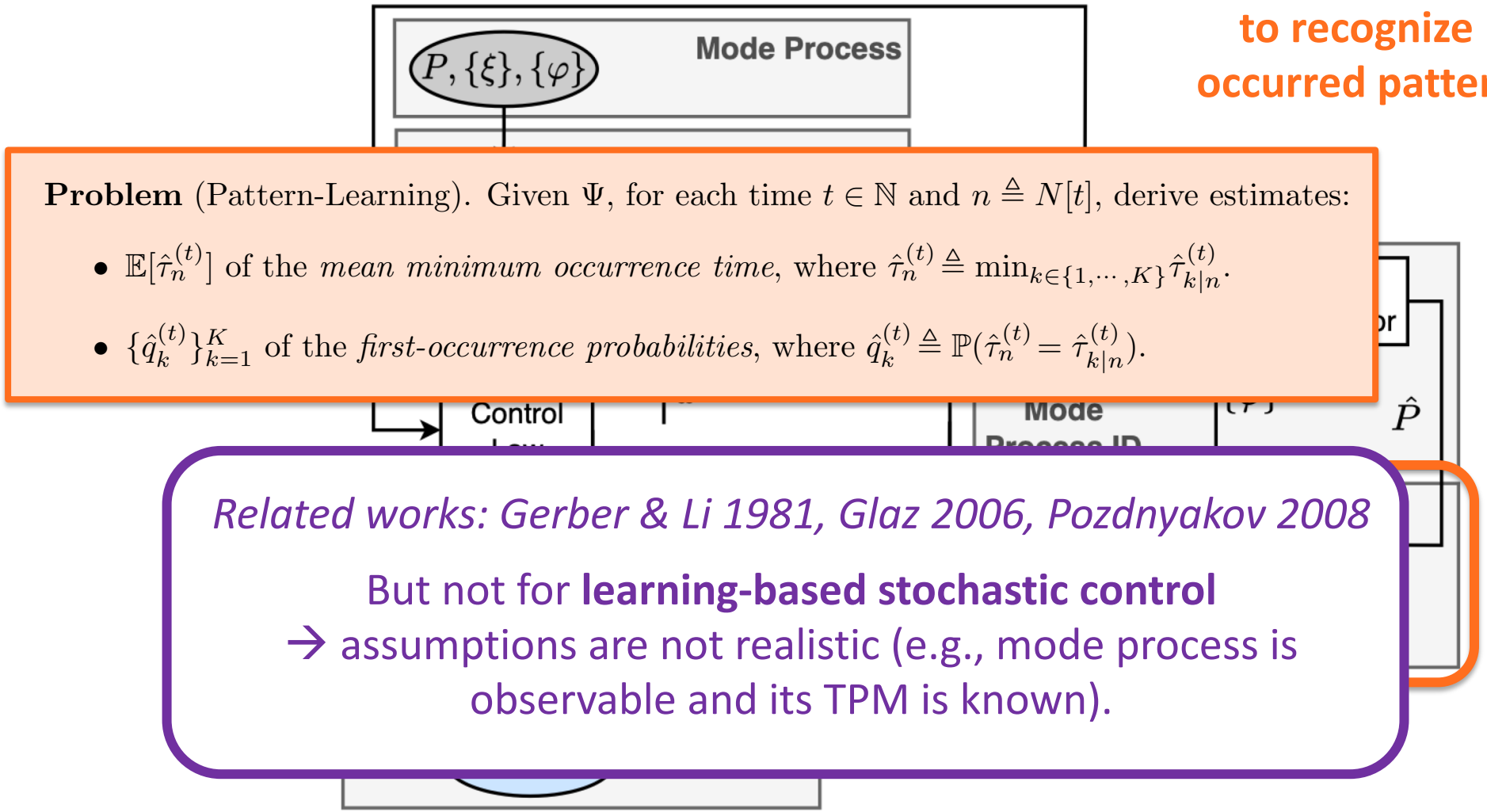
Pattern: a single mode or a sequence of modes over time which is "interesting to the user", e.g., faults in fault-tolerance control

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Caltech

Solving the Pattern-Occurrence Problem

***Pattern-learning component* to recognize previously-occurred patterns of events.**



Problem (Pattern-Learning). Given Ψ, for each time $t \in \mathbb{N}$ and $n \triangleq N[t]$, derive estimates:

- $\mathbb{E}[\hat{\tau}_n^{(t)}]$ of the *mean minimum occurrence time*, where $\hat{\tau}_n^{(t)} \triangleq \min_{k\in\{1,\cdots,K\}} \hat{\tau}_{k|n}^{(t)}$.
- $\{\hat{q}_k^{(t)}\}_{k=1}^K$ of the *first-occurrence probabilities*, where $\hat{q}_k^{(t)} \triangleq \mathbb{P}(\hat{\tau}_n^{(t)} = \hat{\tau}_{k|n}^{(t)})$.

Pattern: a single mode or a sequence of modes over time which is "interesting to the user", e.g., faults in fault-tolerance control

→ Martingale Theory

Related works: Gerber & Li 1981, Glaz 2006, Pozdnyakov 2008

But not for **learning-based stochastic control**
→ assumptions are not realistic (e.g., mode process is observable and its TPM is known).

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Caltech

Construction: Game Interpretation

Notation simplification: given $N[t]=n$, 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 $q_k \equiv \hat{q}_k^{(t)}$.

Definition 5 (Augmented Pattern)**.** In collection of patterns Ψ (see Definition 1), *augmented pattern* $\boldsymbol{\gamma}$ corresponding to 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.

$$\text{Augmented collection } \Gamma \triangleq \{\text{feasible } (m_1, m_2) \circ \boldsymbol{\psi}_k \mid m_1, m_2 \in \mathcal{X}; \boldsymbol{\psi}_k \in \Psi\}, \qquad K_L \triangleq |\Gamma| \in \mathbb{N}$$

Example: Let $M = 4$ and $\mathcal{X} \triangleq \{1,2,3,4\}$.

TPM P is s.t. $P[m_1, m_2] = 0$ if $m_1 = m_2$ or $(m_1, m_2) \in \{(3,2),(2,3),(3,4),(4,3)\}$. $P = \begin{bmatrix} 0 & * & * & * \\ * & 0 & 0 & * \\ * & 0 & 0 & 0 \\ * & * & 0 & 0 \end{bmatrix}$

Collection $\Psi = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3\}$, $(K=3)$.

$\boldsymbol{\psi}_1 = (213)$, $\boldsymbol{\psi}_2 = (412)$, and $\boldsymbol{\psi}_3 = (314)$, $(L=3)$.

Augmented collection $\Gamma \triangleq \cup_{i=1}^{3} \Gamma_i$. $\Gamma_1 \triangleq \{\boldsymbol{\alpha} \circ \boldsymbol{\gamma}_1 | \boldsymbol{\alpha} \in \{(14),(21),(24),(31),(41)\}\}$

$\Gamma_2 \triangleq \{\boldsymbol{\alpha} \circ \boldsymbol{\gamma}_2 | \boldsymbol{\alpha} \in \{(12),(21),(31),(41),(42)\}\}$

$\Gamma_3 \triangleq \{\boldsymbol{\alpha} \circ \boldsymbol{\gamma}_3 | \boldsymbol{\alpha} \in \{(21),(31),(41)\}\}$ $\qquad (K_L = 13)$.

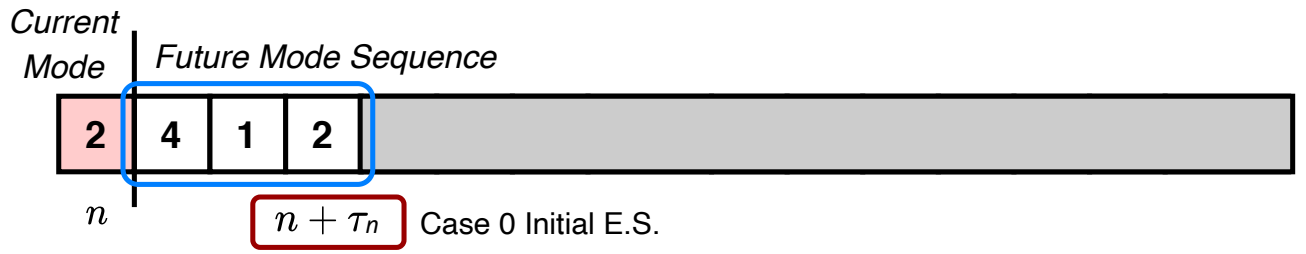
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Ending Strings

Collection $\Psi = \{\psi_1, \psi_2, \psi_3\}$, $(K = 3)$.
$\psi_1 = (213)$, $\psi_2 = (412)$, and $\psi_3 = (314)$, $(L = 3)$.
Augmented collection:
$\Gamma = \cup_{i=1}^{3} \Gamma_i = \{\gamma_1, \cdots, \gamma_8, \cdots, \gamma_{13}\}$
$\gamma_1 = (14) \circ \psi_1$, $\gamma_8 = (31) \circ \psi_2$, $\gamma_{13} = (41) \circ \psi_3$

Definition 6 (Ending String). Let the mode sequence $\{\xi_n, \xi_{n+1}, \cdots\}$ run until one pattern from Ψ is observed. An *ending string* associated with pattern $\psi_k \in \Psi$ terminates the mode process at mode-index $\tau_n > n$ if $\xi_{\tau_n - L + 1:\tau_n} = \psi_k$.

Case-0 initial-ending string $\boldsymbol{\beta}$: $\xi_{n+1:n+L} = \psi_k \Longrightarrow \boldsymbol{\beta} \triangleq \psi_k$. $(\mathcal{S}_I^{(0)} = \{\psi_2\}, K_I^{(0)} = 1, \tau_n = L)$.



Ending Strings

Collection $\Psi = \{\psi_1, \psi_2, \psi_3\}$, $(K = 3)$.
$\psi_1 = (213)$, $\psi_2 = (412)$, and $\psi_3 = (314)$, $(L = 3)$.
Augmented collection:
$\Gamma = \cup_{i=1}^{3} \Gamma_i = \{\gamma_1, \cdots, \gamma_8, \cdots, \gamma_{13}\}$
$\gamma_1 = (14) \circ \psi_1$, $\gamma_8 = (31) \circ \psi_2$, $\gamma_{13} = (41) \circ \psi_3$

Definition 6 (Ending String). Let the mode sequence $\{\xi_n, \xi_{n+1}, \cdots\}$ run until one pattern from Ψ is observed. An *ending string* associated with pattern $\psi_k \in \Psi$ terminates the mode process at mode-index $\tau_n > n$ if $\xi_{\tau_n - L + 1:\tau_n} = \psi_k$.

Case-0 *initial-ending string* $\boldsymbol{\beta}$: $\xi_{n+1:n+L} = \psi_k \Longrightarrow \boldsymbol{\beta} \triangleq \psi_k$. $(\mathcal{S}_I^{(0)} = \{\psi_2\}, K_I^{(0)} = 1, \tau_n = L)$.

Case-1 *initial-ending string* $\boldsymbol{\beta}$:

$\xi_{n+1:n+L+1} = (m_1) \circ \psi_k \Longrightarrow \boldsymbol{\beta} \triangleq (m_1) \circ \psi_k$. $(\mathcal{S}_I^{(1)} = \{(1) \circ \psi_1, (1) \circ \psi_2, (1) \circ \psi_3, (2) \circ \psi_2\}, K_I^{(1)} = 4, \tau_n = L + 1)$.

Case 1 Initial E.S.

Ending Strings

Collection $\Psi = \{\psi_1, \psi_2, \psi_3\}$, $(K = 3)$.
$\psi_1 = (213)$, $\psi_2 = (412)$, and $\psi_3 = (314)$, $(L = 3)$.
Augmented collection:
$\Gamma = \cup_{i=1}^{3} \Gamma_i = \{\gamma_1, \cdots, \gamma_8, \cdots, \gamma_{13}\}$
$\gamma_1 = (14) \circ \psi_1$, $\gamma_8 = (31) \circ \psi_2$, $\gamma_{13} = (41) \circ \psi_3$

Definition 6 (Ending String). Let the mode sequence $\{\xi_n, \xi_{n+1}, \cdots\}$ run until one pattern from Ψ is observed. An *ending string* associated with pattern $\psi_k \in \Psi$ terminates the mode process at mode-index $\tau_n > n$ if $\xi_{\tau_n - L + 1:\tau_n} = \psi_k$.

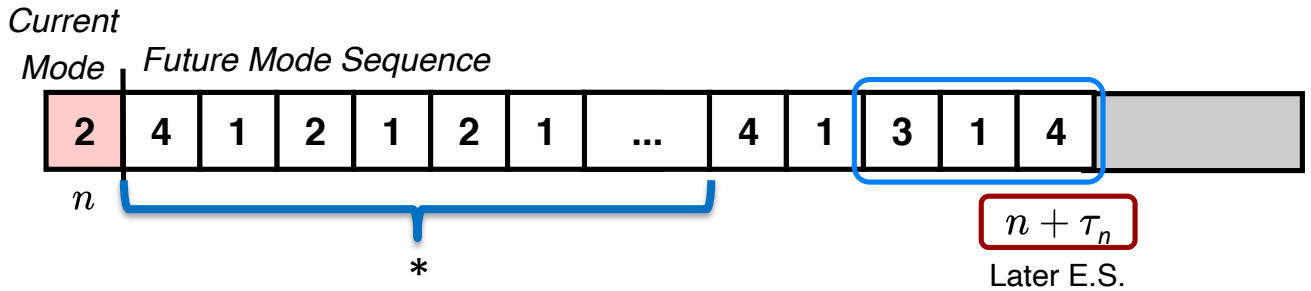
Case-0 initial-ending string $\boldsymbol{\beta}$: $\xi_{n+1:n+L} = \psi_k \Longrightarrow \boldsymbol{\beta} \triangleq \psi_k$. $(\mathcal{S}_I^{(0)} = \{\psi_2\}, K_I^{(0)} = 1, \tau_n = L)$.

Case-1 initial-ending string $\boldsymbol{\beta}$:

$\xi_{n+1:n+L+1} = (m_1) \circ \psi_k \Longrightarrow \boldsymbol{\beta} \triangleq (m_1) \circ \psi_k$. $(\mathcal{S}_I^{(1)} = \{(1) \circ \psi_1, (1) \circ \psi_2, (1) \circ \psi_3, (2) \circ \psi_2\}, K_I^{(1)} = 4, \tau_n = L + 1)$.

Later-ending string $\boldsymbol{\beta}$:

$\xi_{\tau_n - L + 1:\tau_n} = (*, m_1, m_2) \circ \psi_k \Longrightarrow \boldsymbol{\beta} \triangleq (m_1, m_2) \circ \psi_k$. $(\mathcal{S}_L = \{(*) \circ \gamma_\ell : \gamma_\ell \in \Gamma\}, K_L, \tau_n > n + L + 1)$.



$*$ is a placeholder for any feasible sequence of modes.

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Construction: Game Interpretation

$$\boldsymbol{\gamma}_\ell \triangleq (m_1, m_2) \circ \boldsymbol{\psi}_k$$

A *type-ℓ agent* accumulates rewards over mode process $\{\xi_n\}$ with the goal of observing augmented pattern $\boldsymbol{\gamma}_\ell \in \Gamma$.

1. If $\varphi_n = m_1$, type-ℓ agent n aims to observe the event $\{\xi_{n+1:n+L+1} = (m_2) \circ \boldsymbol{\psi}_k\}$.
2. Otherwise, if $\varphi_n \neq m_1$, type-ℓ agent n aims to observe the event $\{\xi_{n+1:n+L} = \boldsymbol{\psi}_k\}$.

For each $\ell \in \{1, \cdots, K_L\}$, a new *type-$\ell$ agent* n enters the game at mode-index $n \in \mathbb{N}$ with *initial reward* $c_\ell \in \mathbb{R}$.

Type-ℓ agent n total reward by mode-index $\overline{n} > n$:

$$r_{n,\overline{n}}^{(\ell)} = \mathbb{1}\{\varphi_n = m_2\} \sum_{j=1}^{L+2} B_{n,j}^{(1,\ell)} \mathbb{1}\{n + j \leq \overline{n}\} + \mathbb{1}\{\varphi_n \neq m_2\} \sum_{j=1}^{L+1} B_{n,j}^{(2,\ell)} \mathbb{1}\{n + j \leq \overline{n}\}$$

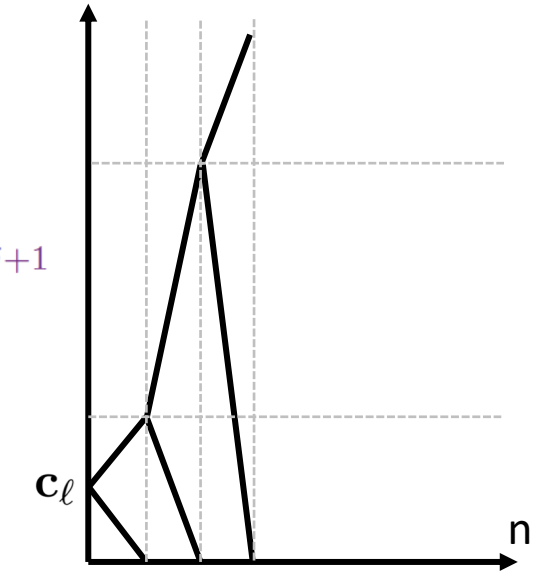
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Betting strategies are defined such that the game is fair-odds.

$$B_{n,0}^{(1,\ell)} = B_{n,0}^{(2,\ell)} = c_\ell$$

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

$B_{n,j}^{(2,\ell)}$ similarly.



$$r_{n,\overline{n}}^{(\ell)} = \mathbb{1}\{\varphi_n = m_2\} \sum_{j=1}^{L+2} B_{n,j}^{(1,\ell)} \mathbb{1}\{n + j \le \overline{n}\} + \mathbb{1}\{\varphi_n \neq m_2\} \sum_{j=1}^{L+1} B_{n,j}^{(2,\ell)} \mathbb{1}\{n + j \le \overline{n}\}$$

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

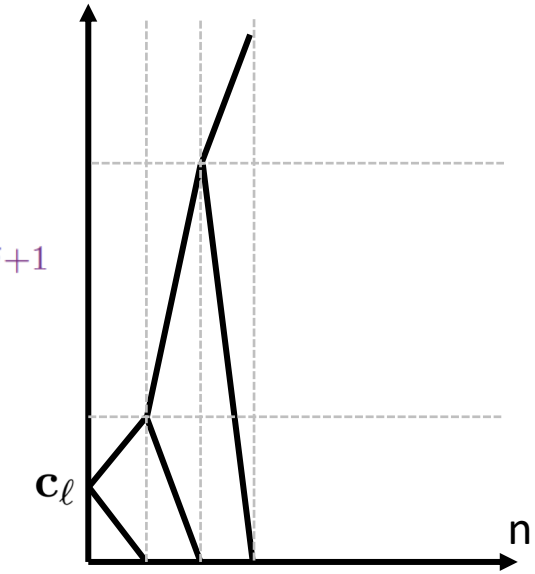
Caltech

Betting strategies are defined such that the game is fair-odds.

$$B^{(1,\ell)}_{n,0} = B^{(2,\ell)}_{n,0} = c_\ell$$

$$B^{(1,\ell)}_{n,j} = \begin{cases} (P(\gamma_{\ell,j}, \gamma_{\ell,j+1}))^{-1} B^{(1,\ell)}_{n,j-1} - B^{(1,\ell)}_{n,j-1} & \text{if } \xi_{n+j-1} = \gamma_{\ell,j}, \xi_{n+j} = \gamma_{\ell,j+1} \\ 0 & \text{if } \sum_{jj=0}^{j-1} B^{(1,\ell)}_{n,jj} = 0 \\ -B^{(1,\ell)}_{n,j-1} & \text{else} \end{cases}$$

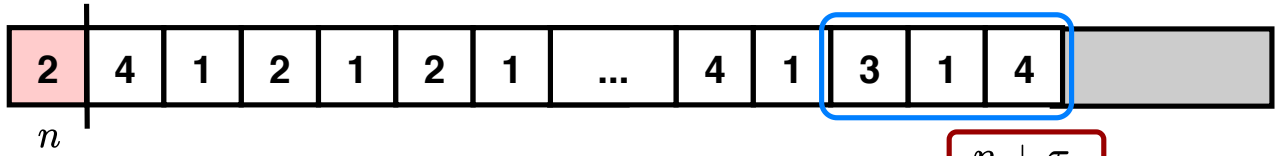
$B^{(2,\ell)}_{n,j}$ similarly.



$$r^{(\ell)}_{n,\overline{n}} = \mathbb{1}\{\varphi_n = m_2\} \sum_{j=1}^{L+2} B^{(1,\ell)}_{n,j} \mathbb{1}\{n + j \le \overline{n}\} + \mathbb{1}\{\varphi_n \neq m_2\} \sum_{j=1}^{L+1} B^{(2,\ell)}_{n,j} \mathbb{1}\{n + j \le \overline{n}\}$$

$$\psi_3 = (314),\ \gamma_{13} = (41) \circ \psi_3$$

Type - 13 Agents



(Agent 1) 4 1 3 1 4

(Agent 2) 4 1 3 1 4

...

(Agent $\tau_n - 5$) 4 1 3 1 4

$$\frac{1}{P[4,1]P[1,3]P[3,1]P[1,4]} c_\ell - c_\ell$$

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

First Quantity: Expected Minimum Time

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

$$\mathbb{P}(\boldsymbol{\beta}_s) = P[\varphi_0, \beta_1] \prod_{j=1}^{b_s - 1} P[\beta_j, \beta_{j+1}]$$

Stationarity of $\{\xi_n\} \Longrightarrow \tau \equiv \tau_n \ \forall\, n \in \mathbb{N}$.

$\mathbf{c}^* \in \mathbb{R}^{K_L}$ initial rewards chosen s.t.
$\sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^* = 1 \ \ \forall\, s \in \{K_I + 1, \cdots, K_I + K_L\}$.

$W_{s\ell}$: *total gain* accumulated over all type-ℓ agents from observing (partial) occurrences of $\boldsymbol{\gamma}_\ell$ in ending string $\boldsymbol{\beta}_s$.

$\mathbb{E}[R_\tau^{(\ell)}]$: *expected type-$\ell$ cumulative net reward* over all type-ℓ agents by mode-index τ. $R_\tau^{(\ell)} \triangleq \sum_{n=1}^{\tau} r_{n,\tau}^{(\ell)}$

$\mathbb{E}[R_{\overline{n}}]$: *expected cumulative net reward* over all agents by mode-index $\overline{n}$. $R_{\overline{n}} \triangleq \sum_{\ell=1}^{K_L} R_{\overline{n}}^{(\ell)}$

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

First Quantity: Expected Minimum Time

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

$$\mathbb{P}(\boldsymbol{\beta}_s) = P[\varphi_0, \beta_1]\prod_{j=1}^{b_s-1} P[\beta_j, \beta_{j+1}]$$

Stationarity of $\{\xi_n\} \Longrightarrow \tau \equiv \tau_n \ \forall\, n \in \mathbb{N}$.

$\mathbf{c}^* \in \mathbb{R}^{K_L}$ initial rewards chosen s.t.
$\sum_{\ell=1}^{K_L} W_{s\ell}c_\ell^* = 1 \ \ \forall\, s \in \{K_I+1, \cdots, K_I+K_L\}$.

$W_{s\ell}$: *total gain* accumulated over all type-ℓ agents from observing (partial) occurrences of $\boldsymbol{\gamma}_\ell$ in ending string $\boldsymbol{\beta}_s$.

$\mathbb{E}[R_\tau^{(\ell)}]$: *expected type-$\ell$ cumulative net reward* over all type-ℓ agents by mode-index τ. $R_\tau^{(\ell)} \triangleq \sum_{n=1}^{\tau} r_{n,\tau}^{(\ell)}$

$\mathbb{E}[R_{\overline{n}}]$: *expected cumulative net reward* over all agents by mode-index $\overline{n}$. $R_{\overline{n}} \triangleq \sum_{\ell=1}^{K_L} R_{\overline{n}}^{(\ell)}$

Proof: Apply Optional Stopping Theorem to martingale construction $\{R_{n\wedge\tau}^{(\ell)}\}_{n\in\mathbb{N}}$ and $\{R_{n\wedge\tau}\}_{n\in\mathbb{N}}$

Finite stopping time: $\mathbb{E}[\tau] < \infty$

Finite-valued martingale: $\mathbb{E}[|R_\tau^{(\ell)}|] < \infty$ and $\mathbb{E}[|R_\tau|] < \infty$

Uniform integrability: $\lim_{n\to\infty}\int_{\Omega_n} |R_n^{(\ell)}(\omega)|d\mathbb{P}(\omega) = 0 \quad \Omega_n \triangleq \{\omega \in \Omega | n < \tau\}$

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

First Quantity: Expected Minimum Time

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

$$\mathbb{P}(\boldsymbol{\beta}_s) = P[\varphi_0, \beta_1] \prod_{j=1}^{b_s - 1} P[\beta_j, \beta_{j+1}]$$

Stationarity of $\{\xi_n\} \Longrightarrow \tau \equiv \tau_n \ \forall\, n \in \mathbb{N}$.

$\mathbf{c}^* \in \mathbb{R}^{K_L}$ initial rewards chosen s.t.
$\sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^* = 1 \ \ \forall\, s \in \{K_I + 1, \cdots, K_I + K_L\}$.

$W_{s\ell}$: *total gain* accumulated over all type-ℓ agents from observing (partial) occurrences of $\boldsymbol{\gamma}_\ell$ in ending string $\boldsymbol{\beta}_s$.

$\mathbb{E}[R_\tau^{(\ell)}]$: *expected type-$\ell$ cumulative net reward* over all type-ℓ agents by mode-index τ. $R_\tau^{(\ell)} \triangleq \sum_{n=1}^{\tau} r_{n,\tau}^{(\ell)}$

$\mathbb{E}[R_{\overline{n}}]$: *expected cumulative net reward* over all agents by mode-index $\overline{n}$. $R_{\overline{n}} \triangleq \sum_{\ell=1}^{K_L} R_{\overline{n}}^{(\ell)}$

Proof: Apply Optional Stopping Theorem to martingale construction $\{R_{n\wedge\tau}^{(\ell)}\}_{n\in\mathbb{N}}$ and $\{R_{n\wedge\tau}\}_{n\in\mathbb{N}}$

Finite stopping time: $\mathbb{E}[\tau] < \infty$

Finite-valued martingale: $\mathbb{E}[|R_\tau^{(\ell)}|] < \infty$ and $\mathbb{E}[|R_\tau|] < \infty$

Uniform integrability: $\lim_{n\to\infty} \int_{\Omega_n} |R_n^{(\ell)}(\omega)| d\mathbb{P}(\omega) = 0 \quad \Omega_n \triangleq \{\omega \in \Omega | n < \tau\}$

$$0 = \mathbb{E}[R_0] = \mathbb{E}[R_\tau] = \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s) \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell^* + \left(1 - \sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s)\right) - \sum_{\ell=1}^{K_L} c_\ell^* \mathbb{E}[\tau]$$

Second Quantity: First-Occurrence Probabilities

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

Proof: Use previous expression for $\mathbb{E}[\tau]$.

Rearrange expression for expected time.

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

Second Quantity: First-Occurrence Probabilities

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

Proof: Use previous expression for $\mathbb{E}[\tau]$.

Rearrange expression for expected time.

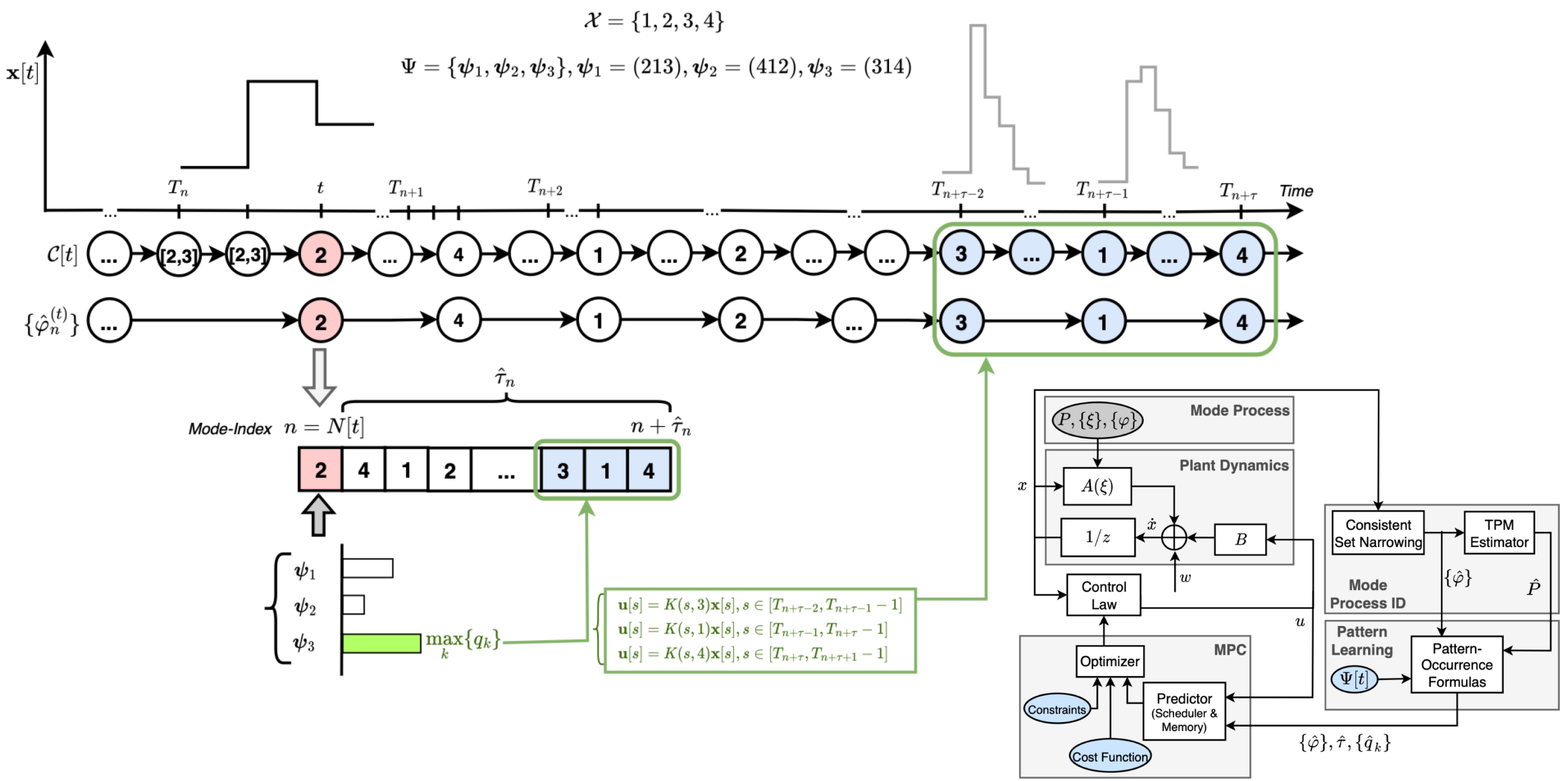
$$\underbrace{\sum_{s=1}^{K_I} \mathbb{P}(\boldsymbol{\beta}_s)}_{\text{known}} \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell = \underbrace{-\sum_{s=K_I+1}^{K_I+K_L} \mathbb{P}(\boldsymbol{\beta}_s)}_{\text{unknown}} \sum_{\ell=1}^{K_L} W_{s\ell} c_\ell + \sum_{\ell=1}^{K_L} c_\ell \mathbb{E}[\tau]$$

Solve for K_L unknowns by creating equations by choosing $\mathbf{c} \in \{[1, 0, \cdots, 0], \cdots, [0, 0, \cdots, 1]\}$

Use conditional probabilities: $q_k \triangleq \mathbb{P}(\tau = \tau_k) = \sum_{\boldsymbol{\beta}_s \in \mathcal{S}} \mathbb{P}(\boldsymbol{\beta}_s) \mathbb{P}(\tau = \tau_k \mid \boldsymbol{\beta}_s)$

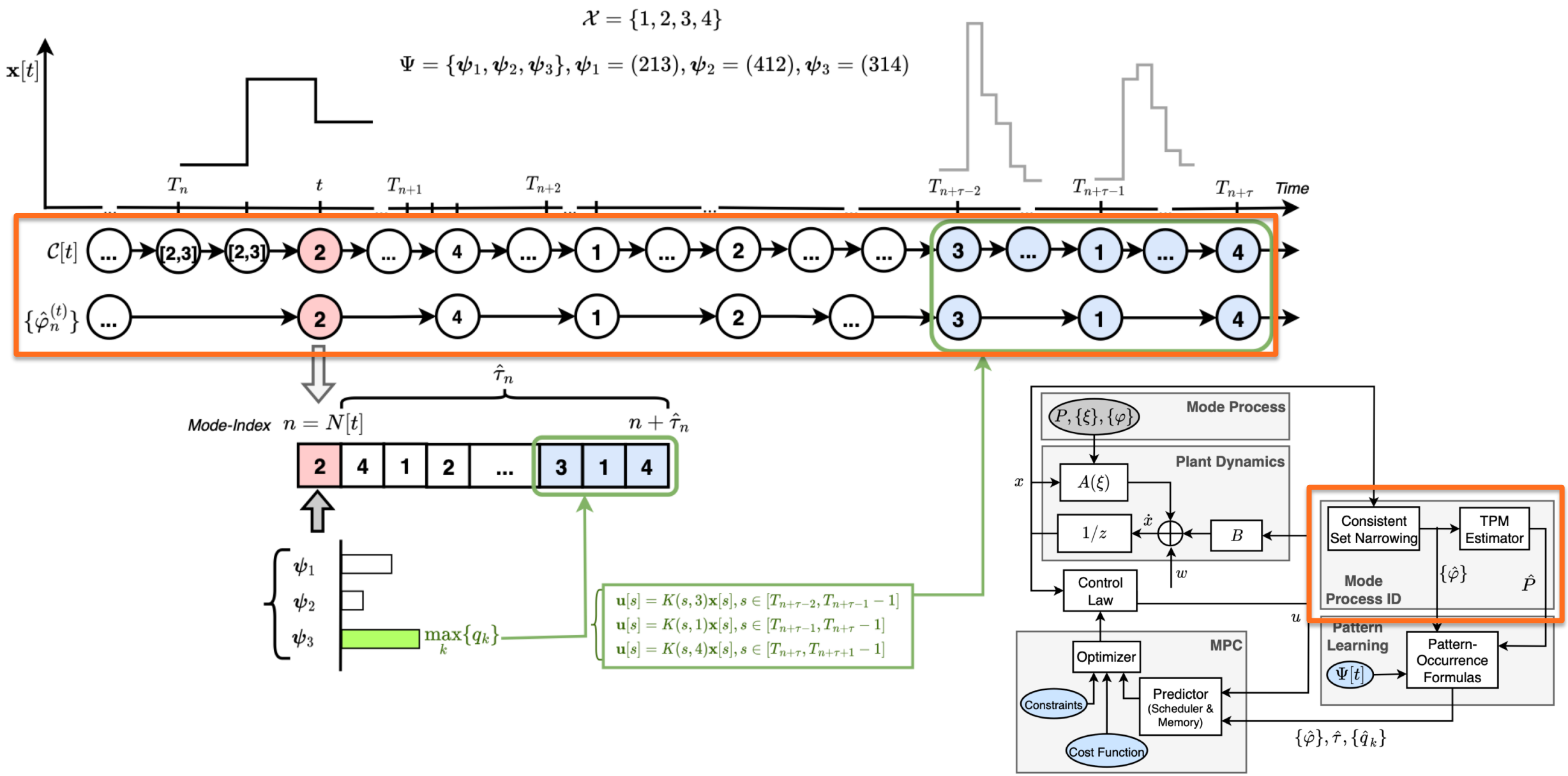
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Proposed Controller Framework



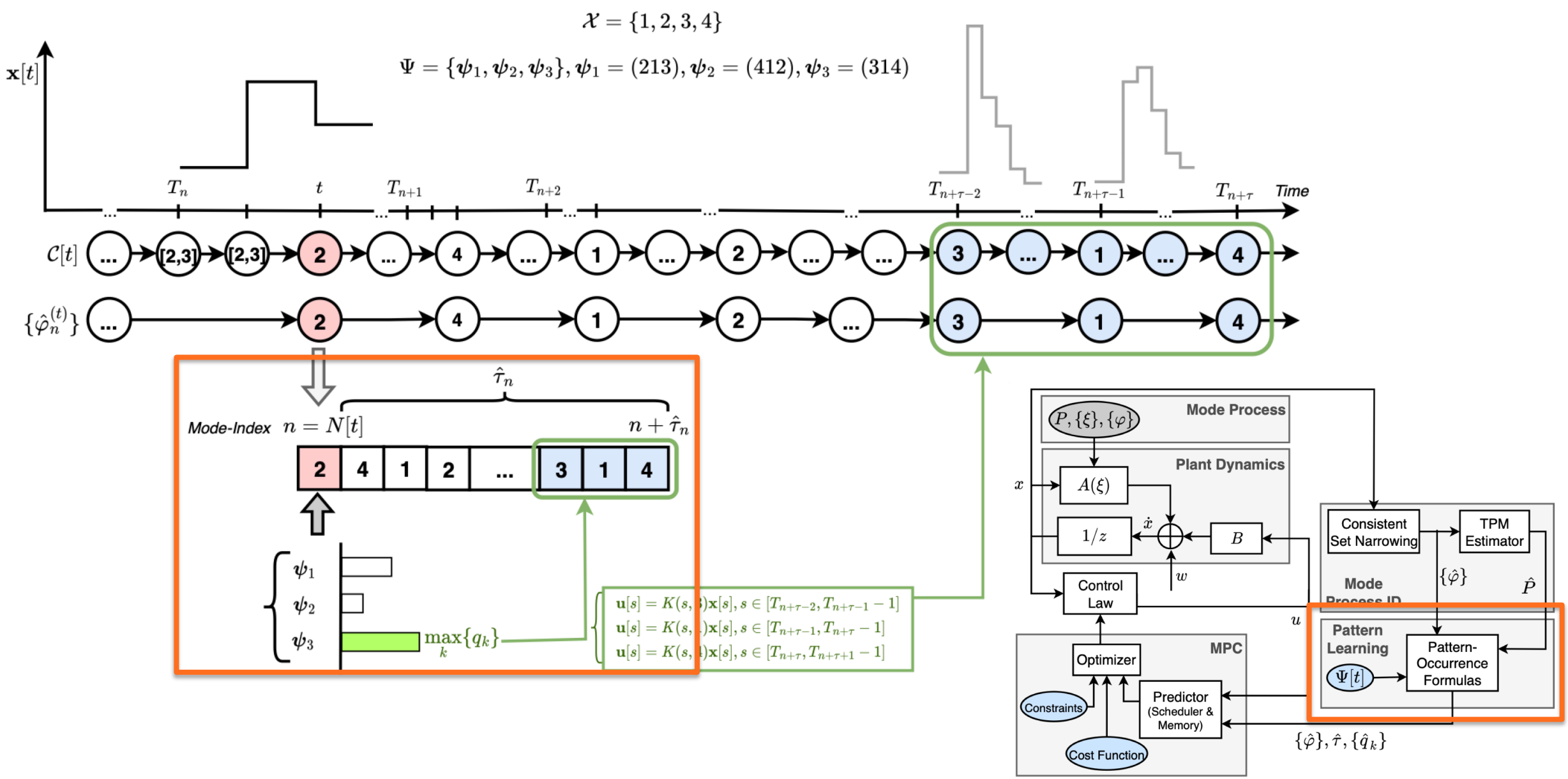
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Proposed Controller Framework



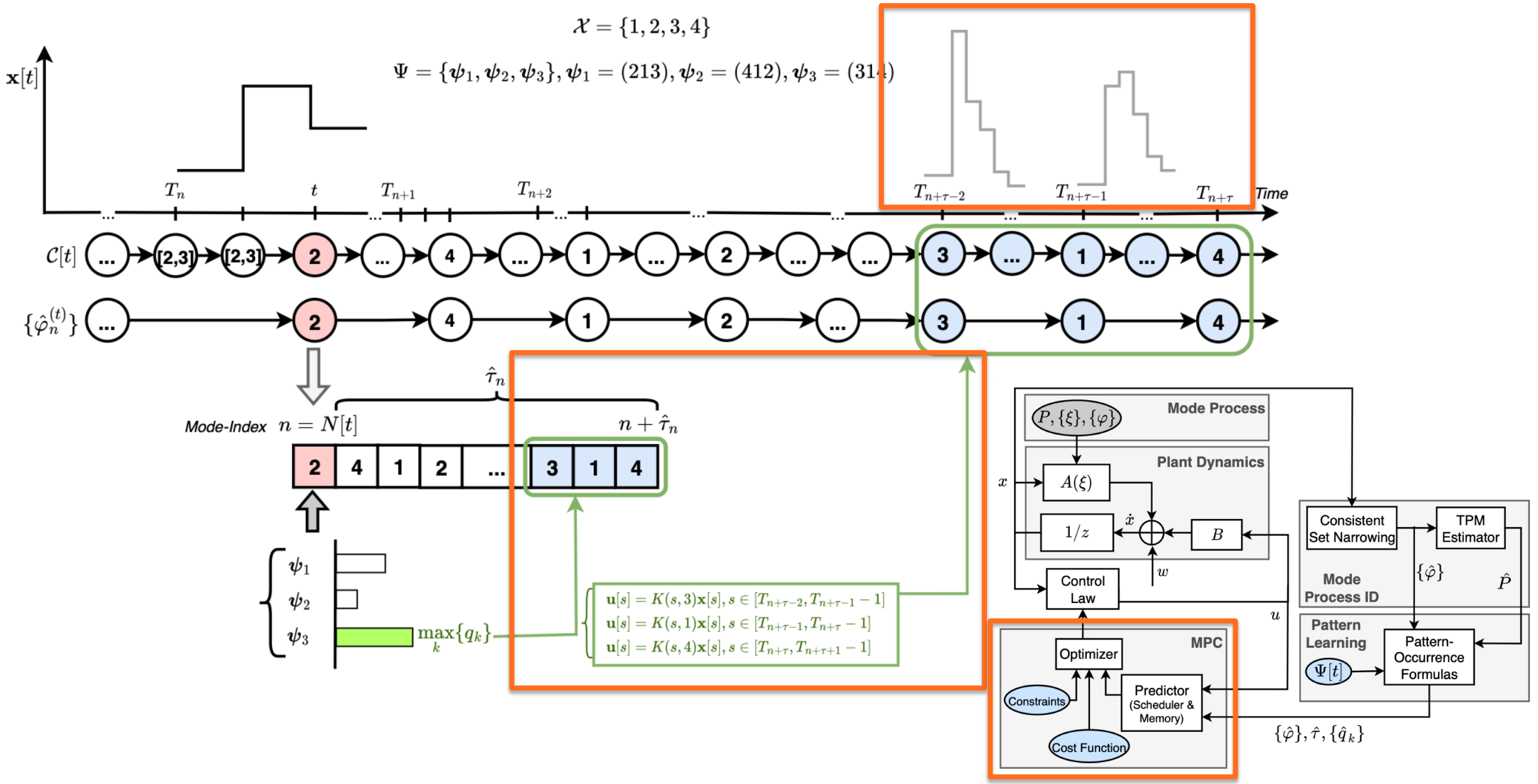
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Proposed Controller Framework



* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Proposed Controller Framework



* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Systems with Repeating Jump Patterns

Control for *jump stochastic systems* can be achieved by learning **patterns** in the system behavior.

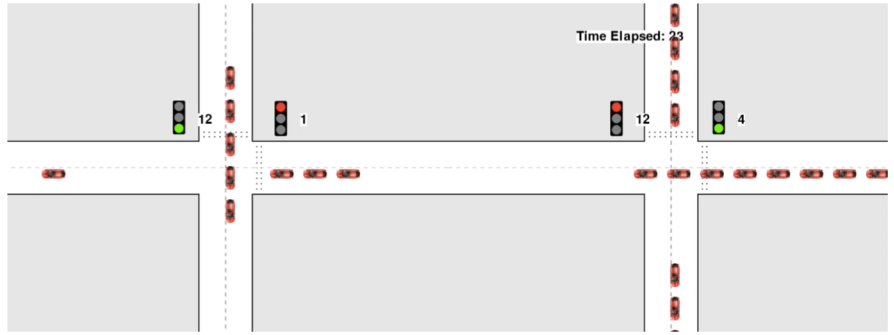
At least 2 ways of improving performance:

- **memorize past patterns**
  → **no re-computation of repeating control policies**
- **predict future patterns**
  → **schedule control policies in advance**

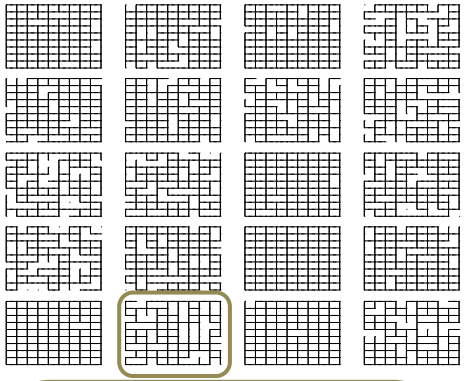
Systems that have jump patterns:

- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- Vehicle traffic congestion
  - **patterns** = intersection snapshots
- Epidemic spread
  - **patterns** = contact-tracing data, travel routines

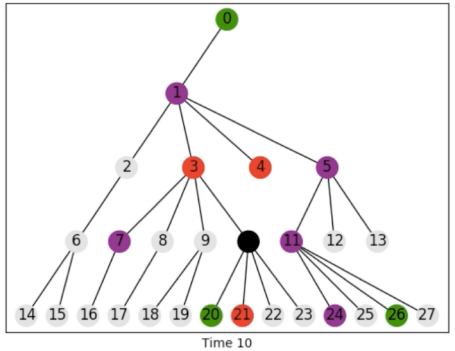
Vehicle Traffic Congestion

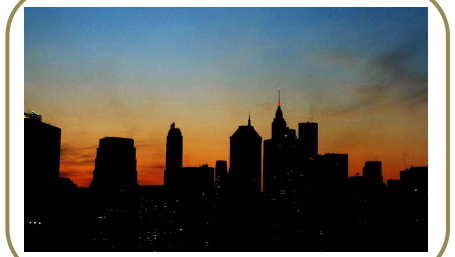


Power Grid



Epidemic Spread





Caltech

Systems with Repeating Jump Patterns

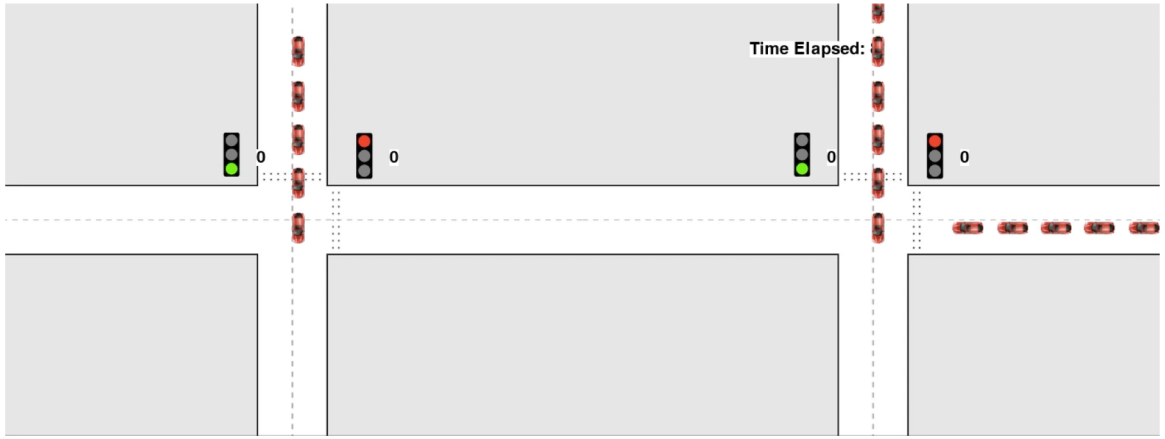
Systems that have jump patterns:

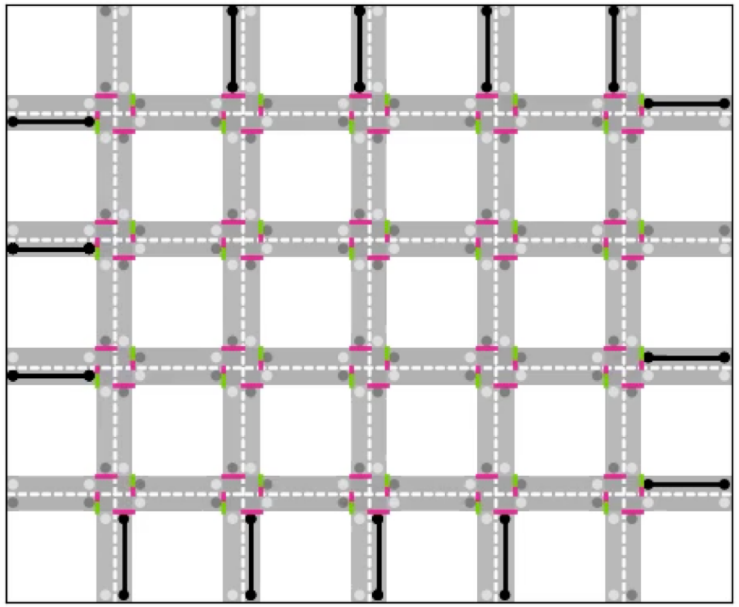
- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- Vehicle traffic congestion
  - **patterns** = intersection snapshots
- Epidemic spread
  - **patterns** = contact-tracing data, travel routines

Systems with Repeating Jump Patterns

Vehicle traffic networks:

- *Pattern-Learning*: identify recurrent snapshots of the intersection to recycle light signal control actions.
- *Mode Process Controller*: rule-based table / optimization to minimize level of congestion as quickly as possible.





Systems that have jump patterns:

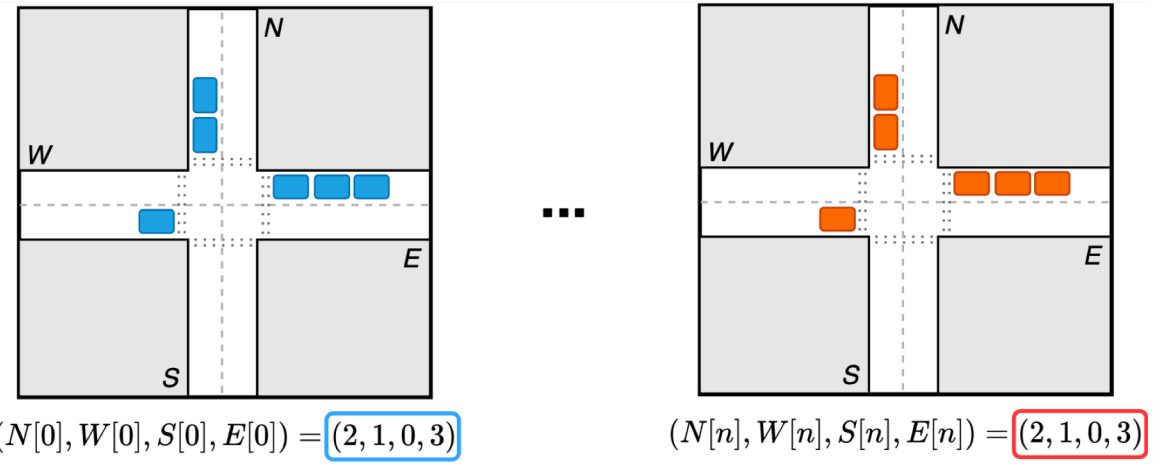
- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- **Vehicle traffic congestion**
  - **patterns = intersection snapshots**
- Epidemic spread
  - **patterns** = contact-tracing data, travel routines

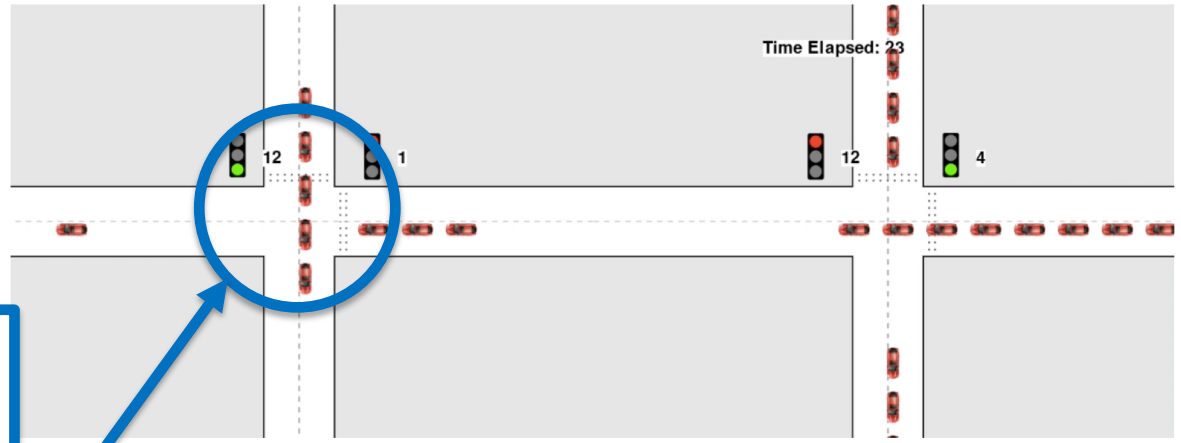
*** SooJean Han, Johanna Gustafson, and Soon-Jo Chung. *Rule-Based Adaptive Control of an Isolated Intersection*, L4DC 2023, to be submitted.**
*** SooJean Han and Soon-Jo Chung. *Quantum Annealing for Iterative Traffic Flow Optimization over Vehicle Routes and Light Signals*, L4DC 2023, to be submitted.**

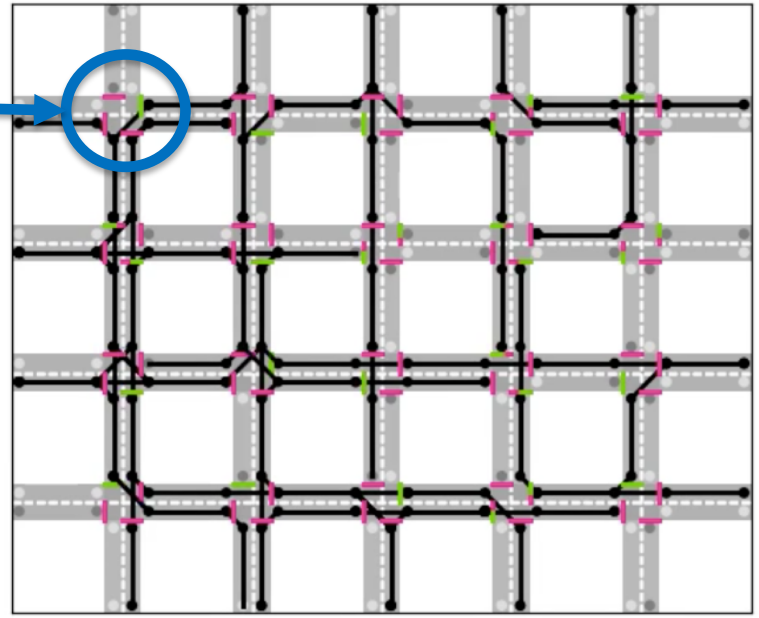
Systems with Repeating Jump Patterns

Vehicle traffic networks:

- ***Pattern-Learning*: identify recurrent snapshots of the intersection to recycle light signal control actions.**







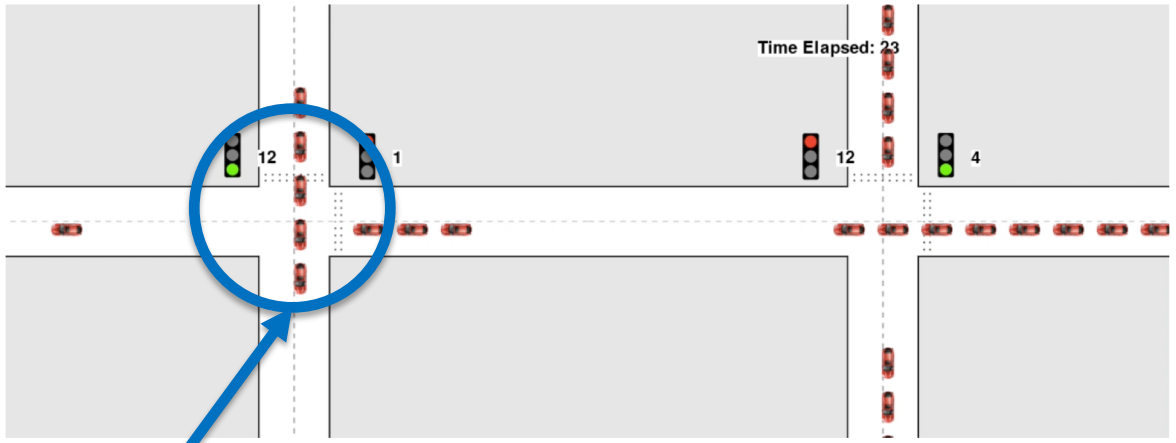
- Networks with time-varying topology (e.g., power grid)
  - patterns = power outages, line failures
- **Vehicle traffic congestion**
  - **patterns = intersection snapshots**
- Epidemic spread
  - patterns = contact-tracing data, travel routines

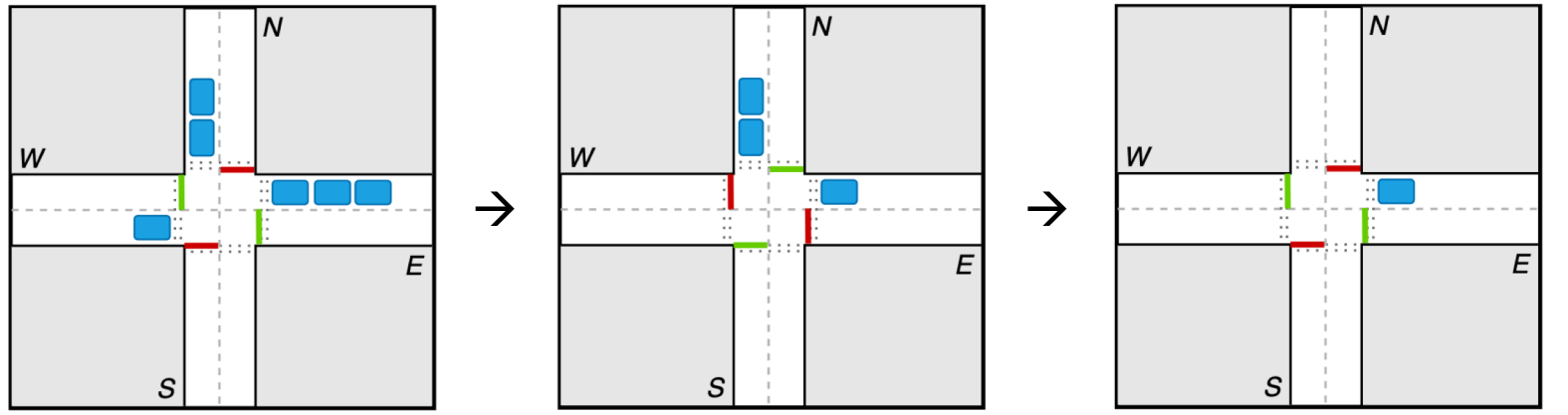
* SooJean Han, Johanna Gustafson, and Soon-Jo Chung. *Rule-Based Adaptive Control of an Isolated Intersection*, L4DC 2023, to be submitted.
* SooJean Han and Soon-Jo Chung. *Quantum Annealing for Iterative Traffic Flow Optimization over Vehicle Routes and Light Signals*, L4DC 2023, to be submitted.

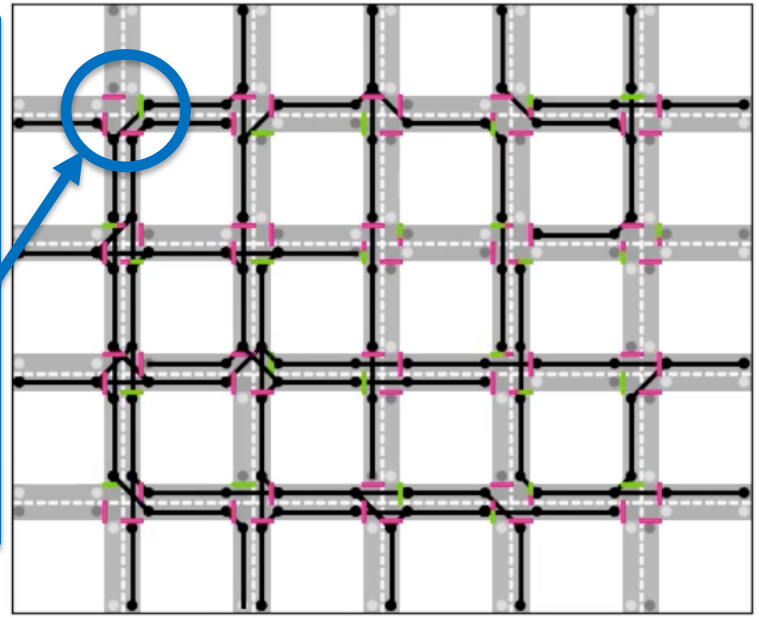
Systems with Repeating Jump Patterns

Vehicle traffic networks:

- *Pattern-Learning*: identify recurrent snapshots of the intersection to recycle light signal control actions.
- ***Mode Process Controller*: rule-based table / optimization to minimize level of congestion as quickly as possible.**







- Epidemic spread
  - patterns = contact-tracing data, travel routines

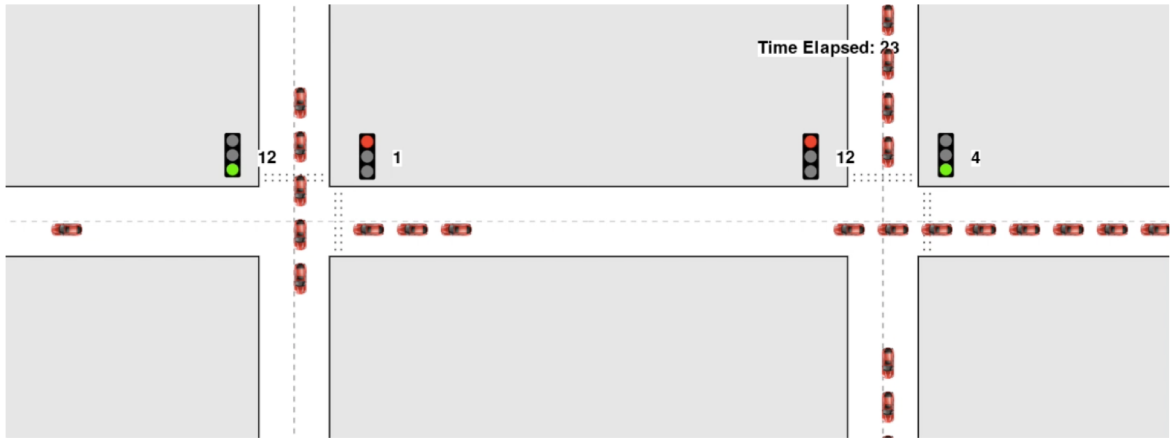
* SooJean Han, Johanna Gustafson, and Soon-Jo Chung. *Rule-Based Adaptive Control of an Isolated Intersection*, L4DC 2023, to be submitted.
* SooJean Han and Soon-Jo Chung. *Quantum Annealing for Iterative Traffic Flow Optimization over Vehicle Routes and Light Signals*, L4DC 2023, to be submitted.

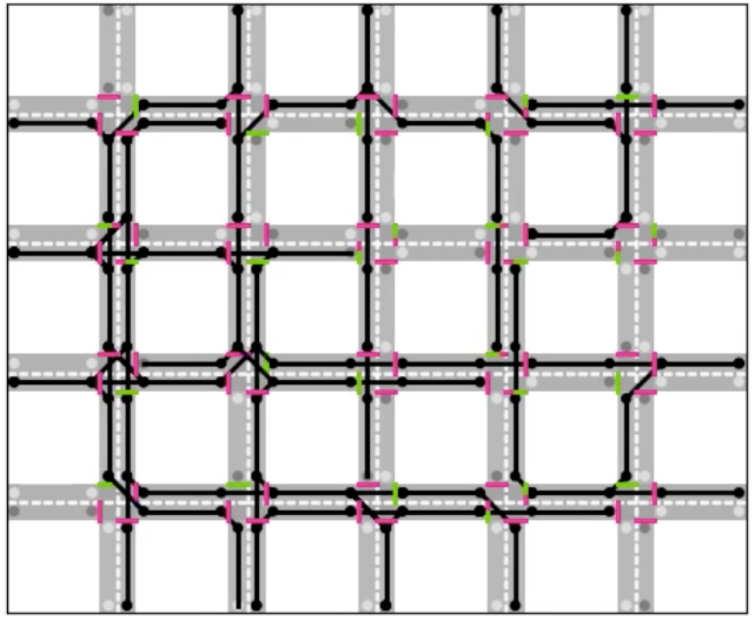
Systems with Repeating Jump Patterns

Vehicle traffic networks:

In urban intersection networks, structural patterns can also be exploited

- *Space* → rectangular grid, 4-way directions, constant distance between adjacent intersections.
- *Time* → traffic density evolution repeats due to human routine (e.g., on weekdays, traffic is heavy due to morning and evening commutes).





Systems that have jump patterns:

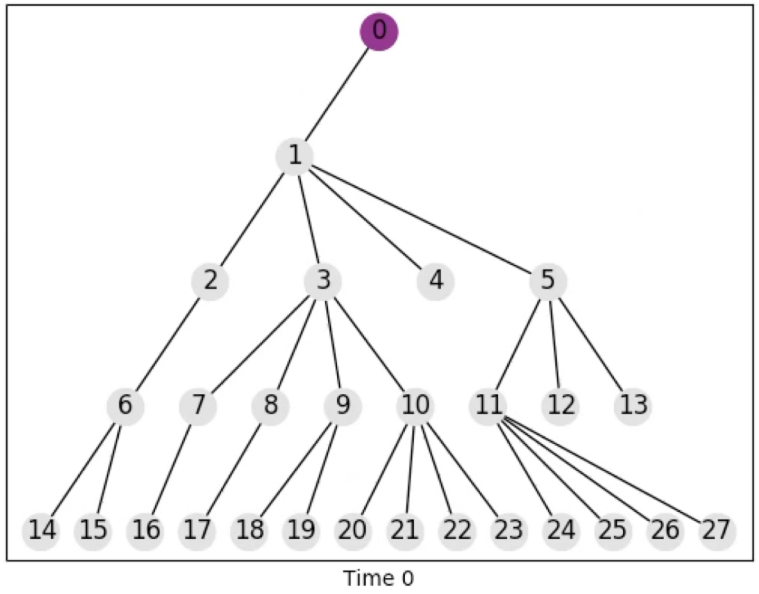
- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- **Vehicle traffic congestion**
  - **patterns = intersection snapshots**
- Epidemic spread
  - **patterns** = contact-tracing data, travel routines

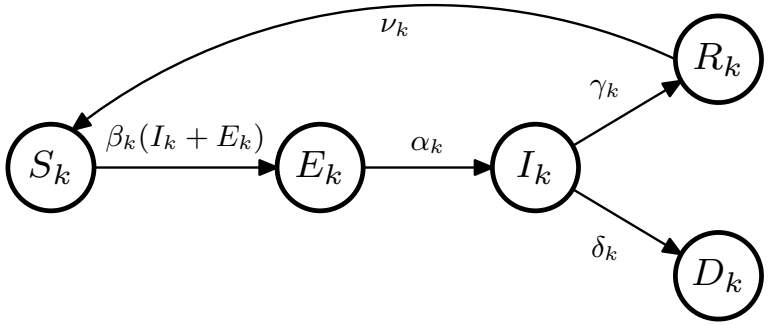
*** SooJean Han, Johanna Gustafson, and Soon-Jo Chung. *Rule-Based Adaptive Control of an Isolated Intersection*, L4DC 2023, to be submitted.**

*** SooJean Han and Soon-Jo Chung. *Quantum Annealing for Iterative Traffic Flow Optimization over Vehicle Routes and Light Signals*, L4DC 2023, to be submitted.**

Systems with Repeating Jump Patterns

Epidemic spread:





$$\frac{dS_k(t)}{dt} = -\sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{M_j(t)} S_k(t) + \nu_k R_k(t)$$

$$\frac{dE_k(t)}{dt} = \sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{M_j(t)} S_k(t) - \alpha_k E_k(t)$$

$$\frac{dI_k(t)}{dt} = \alpha_k E_k(t) - (\gamma_k + \delta_k) I_k(t)$$

$$\frac{dR_k(t)}{dt} = \gamma_k I_k(t) - \nu_k R_k(t)$$

$$\frac{dD(t)}{dt} = \sum_{k=1}^{K} \delta_k I_k(t) \qquad M = \sum_{k=1}^{K} M_k(t),\ \forall t$$

Systems that have jump patterns:

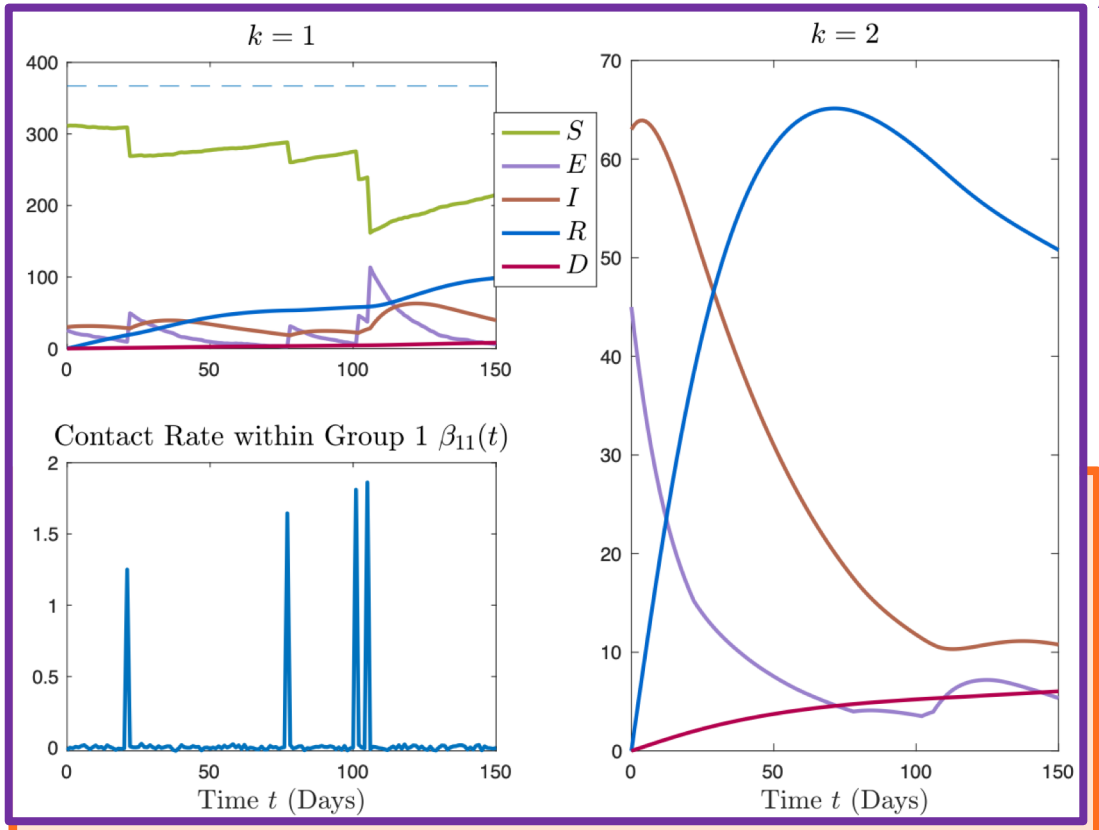
- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- Vehicle traffic congestion
  - **patterns** = intersection snapshots
- **Epidemic spread**
  - **patterns = contact-tracing data, travel routines**

*** SooJean Han, Shuyue Yu, Soon-Jo Chung, and John C. Doyle. *Modeling and Estimation of Epidemic Spread in Population Networks Using Compartmental ODEs and Coupled Hidden Markov Models*, SIAM 2022, to be submitted.**

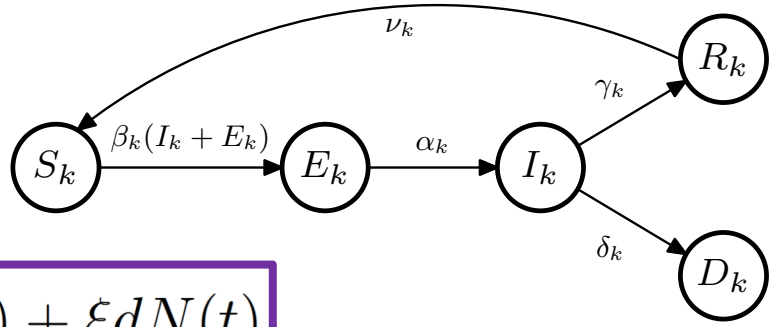
Systems with Repeating Jump Patterns

Epidemic spread:

Super-spreader phenomena: average rate of contact per day is stochastic with Poisson shot noise.



- **Epidemic spread**
  - **patterns = contact-tracing data, travel routines**



$$\beta dt = \beta_0 dt + \sigma dW(t) + \xi dN(t)$$

$$\frac{dS_k(t)}{dt} = -\sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{M_j(t)} S_k(t) + \nu_k R_k(t)$$

$$\frac{dE_k(t)}{dt} = \sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{M_j(t)} S_k(t) - \alpha_k E_k(t)$$

$$\frac{dI_k(t)}{dt} = \alpha_k E_k(t) - (\gamma_k + \delta_k) I_k(t)$$

$$\frac{dR_k(t)}{dt} = \gamma_k I_k(t) - \nu_k R_k(t)$$

$$\frac{dD(t)}{dt} = \sum_{k=1}^{K} \delta_k I_k(t) \qquad M = \sum_{k=1}^{K} M_k(t),\ \forall t$$

* SooJean Han, Shuyue Yu, Soon-Jo Chung, and John C. Doyle. *Modeling and Estimation of Epidemic Spread in Population Networks Using Compartmental ODEs and Coupled Hidden Markov Models*, SIAM 2022, to be submitted.

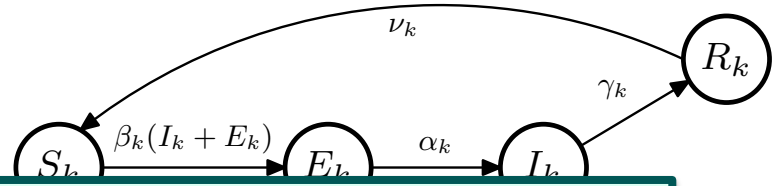
Systems with Repeating Jump Patterns

Epidemic spread:

Super-spreader phenomena: average rate of contact per day is stochastic with Poisson shot noise.

Emergence of variant strains / related viruses:
(e.g., Omicron / Monkeypox)

- All parameters change value (event-driven switching).
- The interarrival times between consecutive switches to a different virus → Poisson or renewal process.



$$\{\beta_{i,kj},\ \alpha_{i,k},\ \gamma_{i,k},\ \delta_{i,k},\ \nu_{i,k}\}_{i=1}^{A}$$

A total variant viruses (i.e., distinct modes).

$$\frac{dS_k(t)}{dt} = -\sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{M_j(t)} S_k(t) + \nu_k R_k(t)$$

$$\frac{dE_k(t)}{dt} = \sum_{j=1}^{K} \frac{\beta_{kj}(I_j(t) + E_j(t))}{M_j(t)} S_k(t) - \alpha_k E_k(t)$$

$$\frac{dI_k(t)}{dt} = \alpha_k E_k(t) - \gamma_k + \delta_k I_k(t)$$

$$\frac{dR_k(t)}{dt} = \gamma_k I_k(t) - \nu_k R_k(t)$$

$$\frac{dD(t)}{dt} = \sum_{k=1}^{K} \delta_k I_k(t) \qquad M = \sum_{k=1}^{K} M_k(t),\ \forall t$$

Systems that have jump patterns:

- Networks with time-varying topology (e.g., power grid)
  - patterns = power outages, line failures
- Vehicle traffic congestion
  - patterns = intersection snapshots
- **Epidemic spread**
  - **patterns = contact-tracing data, travel routines**

* SooJean Han, Shuyue Yu, Soon-Jo Chung, and John C. Doyle. *Modeling and Estimation of Epidemic Spread in Population Networks Using Compartmental ODEs and Coupled Hidden Markov Models*, SIAM 2022, to be submitted.

Systems with Repeating Jump Patterns

$$\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m))$$
$$m \in \{1, \cdots, M\}$$

Pattern-Learning: identify recurrent topologies of the network to recycle closed-loop responses.

Mode Process Controller: perform disturbance-rejection using an approach based on *system-level synthesis (SLS)*[*].

Systems that have jump patterns:

- **Networks with time-varying topology (e.g., power grid)**
  - **patterns = power outages, line failures**
- Vehicle traffic congestion
  - patterns = intersection snapshots
- Epidemic spread
  - patterns = contact-tracing data, travel routines

Application: Networks with Time-Varying Topology

$\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m))$
$m \in \{1, \cdots, M\}$

Pattern-Learning: identify recurrent topologies of the network to recycle closed-loop responses.

Mode Process Controller: perform disturbance-rejection using an approach based on *system-level synthesis (SLS)**.

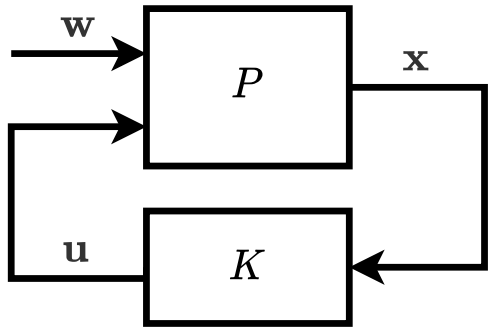
$$\mathbf{x}_i[t+1] = A_{ii}(\xi_{N[t]})\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i(\xi_{N[t]})} A_{ij}(\xi_{N[t]})\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t], \quad i \in \mathcal{V}$$

$$\mathcal{N}_i(m) \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}(m)\}$$

Application: Networks with Time-Varying Topology

Pattern-Learning: identify recurrent topologies of the network to recycle closed-loop responses.

Mode Process Controller: perform disturbance-rejection using an approach based on *system-level synthesis (SLS)*[*].



$$\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$$

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

State-feedback control: $\mathbf{u} = K\mathbf{x}$.

Closed-loop system z-transform:

$$\mathbf{x} = (zI - A - BK)^{-1}\mathbf{w}$$

Optimizing this transfer function by solving for K is difficult when (A, B) is large-scale.

$$\mathbf{x}_i[t+1] = A_{ii}\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i} A_{ij}\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t], \quad i \in \mathcal{V}$$

$$\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$$

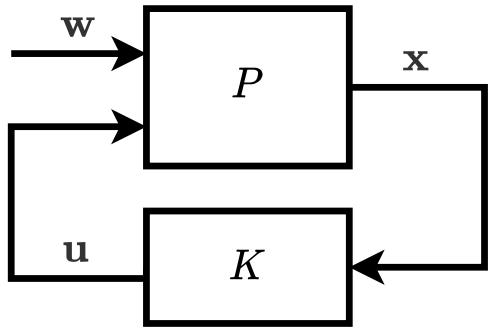
* Yuh-Shyang Wang, Nikolai Matni, John C. Doyle. *Separable and localized system level synthesis for large-scale systems.* IEEE Transactions on Automatic Control, 2018.

* James Anderson, John C. Doyle, Steven Low, Nikolai Matni. *System Level Synthesis*. Annual Reviews in Control, 2019.

Application: Networks with Time-Varying Topology

Pattern-Learning: identify recurrent topologies of the network to recycle closed-loop responses.

Mode Process Controller: perform disturbance-rejection using an approach based on *system-level synthesis (SLS)**.



$$\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$$

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

State-feedback control: $\mathbf{u} = K\mathbf{x}$.

Closed-loop system z-transform:

$$\mathbf{x} = (zI - A - BK)^{-1}\mathbf{w}$$

SLS: Design for the entire closed-loop response.

$$\mathbf{\Phi} \triangleq \{\mathbf{\Phi}_x, \mathbf{\Phi}_u\} \text{ s.t. } \begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix}\mathbf{w}$$

$$\mathbf{\Phi}_x \leftrightarrow (zI - A - B_2K)^{-1}$$

$$\mathbf{\Phi}_u \leftrightarrow K(zI - A - B_2K)^{-1}$$

Optimizing over $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\}$ instead of K.

$$\mathbf{x}_i[t+1] = A_{ii}\mathbf{x}_i[t] + \sum_{j\in\mathcal{N}_i} A_{ij}\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t], \quad i \in \mathcal{V}$$

$$\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$$

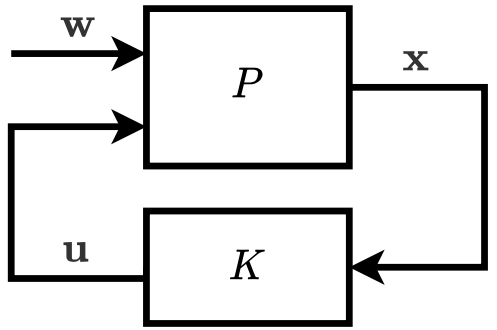
* Yuh-Shyang Wang, Nikolai Matni, John C. Doyle. *Separable and localized system level synthesis for large-scale systems.* IEEE Transactions on Automatic Control, 2018.

* James Anderson, John C. Doyle, Steven Low, Nikolai Matni. *System Level Synthesis*. Annual Reviews in Control, 2019.

Application: Networks with Time-Varying Topology

Pattern-Learning: identify recurrent topologies of the network to recycle closed-loop responses.

Mode Process Controller: perform disturbance-rejection using an approach based on *system-level synthesis (SLS)*[*].



$$\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$$

$$\mathbf{\Phi} \triangleq \{\mathbf{\Phi}_x, \mathbf{\Phi}_u\} \text{ s.t. } \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \mathbf{w}$$

Implementation in terms of original signals: finite-horizon case

$$\hat{\mathbf{x}}[t] = \sum_{k=2}^{T} \Phi_x[k]\hat{\mathbf{w}}[t+1-k]$$

$$\hat{\mathbf{w}}[t] = \mathbf{x}[t] - \hat{\mathbf{x}}[t]$$

$$\mathbf{u}[t] = \sum_{k=1}^{T} \Phi_u[k]\hat{\mathbf{w}}[t+1-k]$$

$$\mathbf{x}_i[t+1] = A_{ii}\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i} A_{ij}\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t], \quad i \in \mathcal{V}$$

$$\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$$

z-transform

$$\mathbf{\Phi}_x \equiv \mathbf{\Phi}_x(z) \triangleq \sum_{i=1}^{\infty} \Phi_x[i]\frac{1}{z^i}$$

$$\mathbf{\Phi}_u \equiv \mathbf{\Phi}_u(z) \triangleq \sum_{i=1}^{\infty} \Phi_u[i]\frac{1}{z^i}$$

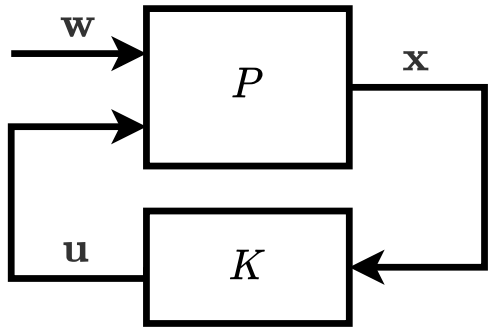
Local implementation over $\mathcal{G}$: $\mathbf{\Phi}^{(i)} \triangleq \{\Phi_x^{(i)}[s], \Phi_u^{(i)}[s]\}_{s=1}^{T} \quad \forall\, i \in \mathcal{V}.$

* Yuh-Shyang Wang, Nikolai Matni, John C. Doyle. *Separable and localized system level synthesis for large-scale systems.* IEEE Transactions on Automatic Control, 2018.
* James Anderson, John C. Doyle, Steven Low, Nikolai Matni. *System Level Synthesis*. Annual Reviews in Control, 2019.

Application: Networks with Time-Varying Topology

Pattern-Learning: identify recurrent topologies of the network to recycle closed-loop responses.

Mode Process Controller: perform disturbance-rejection using an approach based on *system-level synthesis (SLS)**.



$$\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m))$$
$$m \in \{1, \cdots, M\}$$

$$\mathbf{\Phi} \triangleq \{\mathbf{\Phi}_x, \mathbf{\Phi}_u\} \text{ s.t. } \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \mathbf{w}$$

z-transform

$$\mathbf{\Phi}_x \equiv \mathbf{\Phi}_x(z) \triangleq \sum_{i=1}^{\infty} \Phi_x[i] \frac{1}{z^i}$$

$$\mathbf{\Phi}_u \equiv \mathbf{\Phi}_u(z) \triangleq \sum_{i=1}^{\infty} \Phi_u[i] \frac{1}{z^i}$$

Time-varying local implementation over dynamic topology $\mathcal{G}(m)$:

$$\mathbf{\Phi}_m^{(i,t)} \triangleq \{\Phi_{x,m}^{(i,t)}[s], \Phi_{u,m}^{(i,t)}[s]\} \quad \forall\, m \in \{1, \cdots, M\}, i \in \mathcal{V}.$$

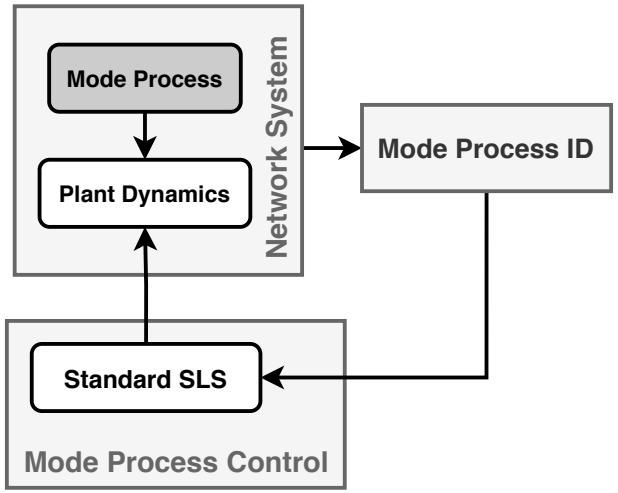
$$\mathbf{x}_i[t+1] = A_{ii}(\xi_{N[t]})\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i(\xi_{N[t]})} A_{ij}(\xi_{N[t]})\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t], \quad i \in \mathcal{V}$$

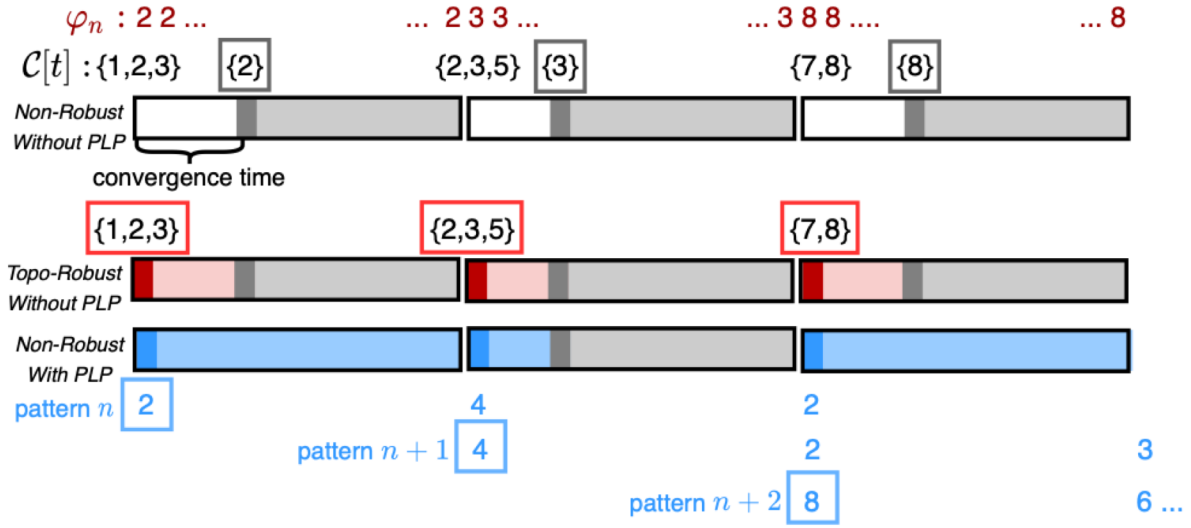
$$\mathcal{N}_i(m) \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}(m)\}$$

* SooJean Han. *Localized Learning of Robust Controllers for Networked Systems with Dynamic Topology.* L4DC, 2020.

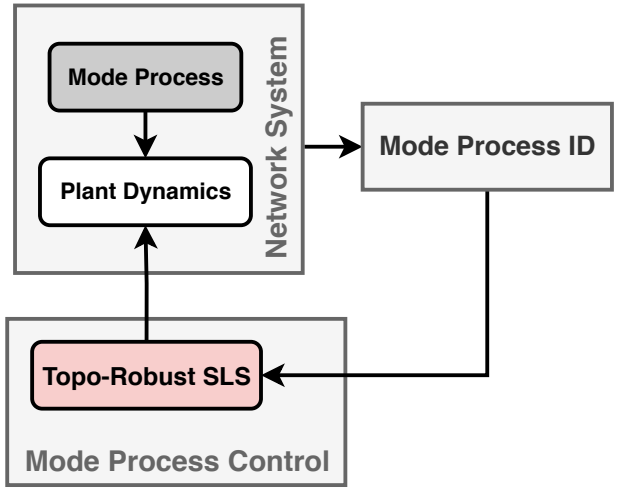
Three Controller Frameworks

Non-Robust w/o PLP:

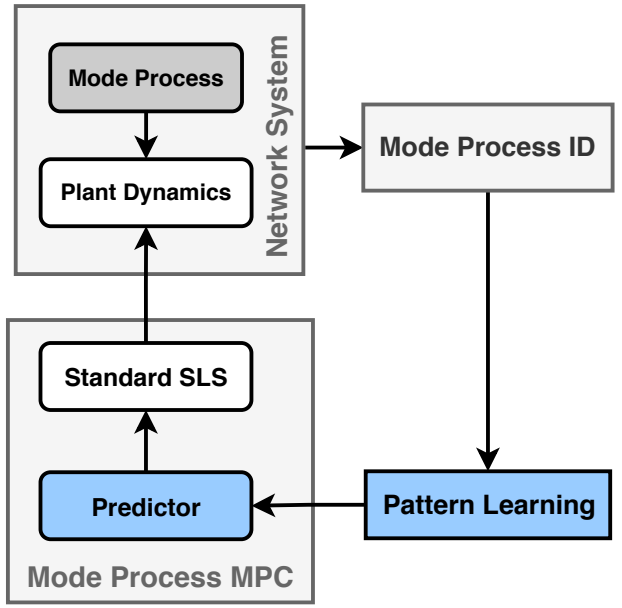




Topologically-Robust w/o PLP:



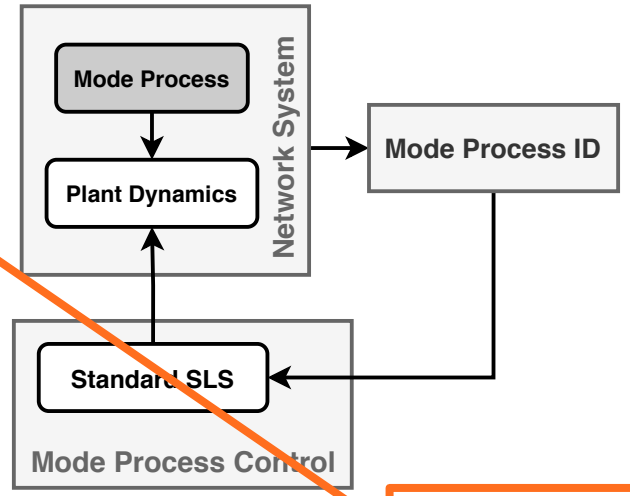
Non-Robust w/ PLP:

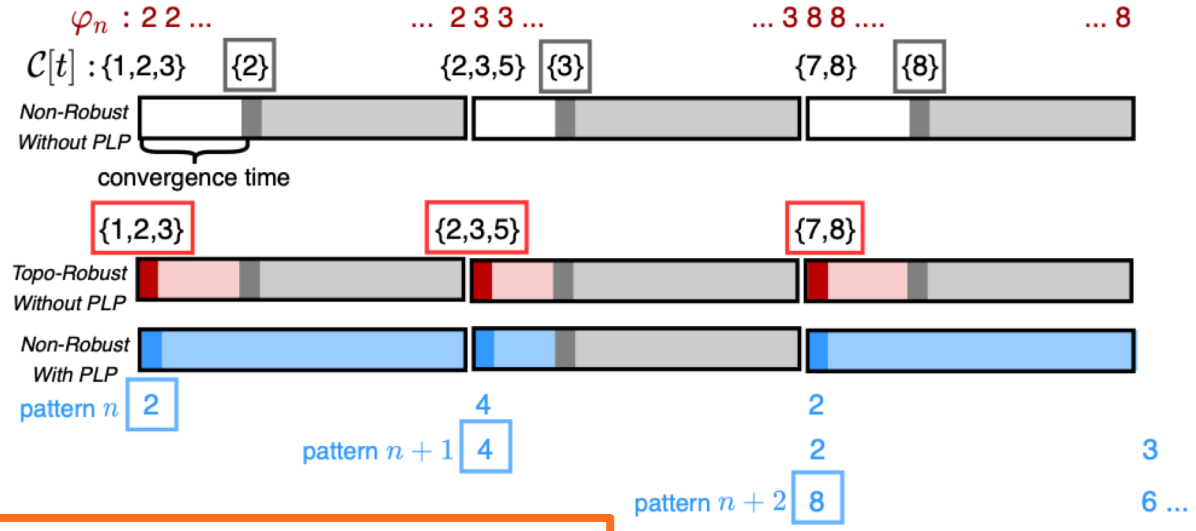


* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Three Controller Frameworks

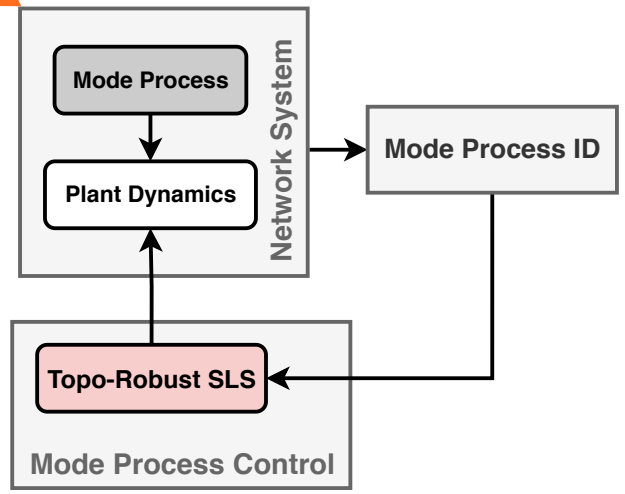
Non-Robust w/o PLP:



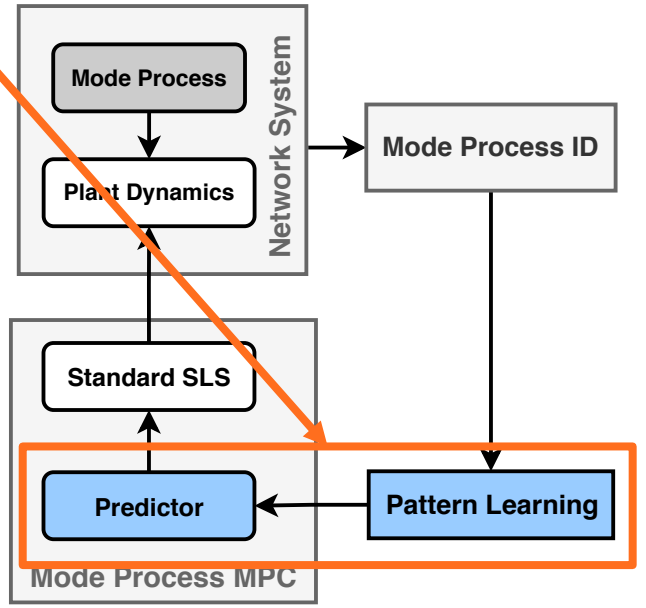


("PLP" = pattern learning and prediction)

Topologically-Robust w/o PLP:



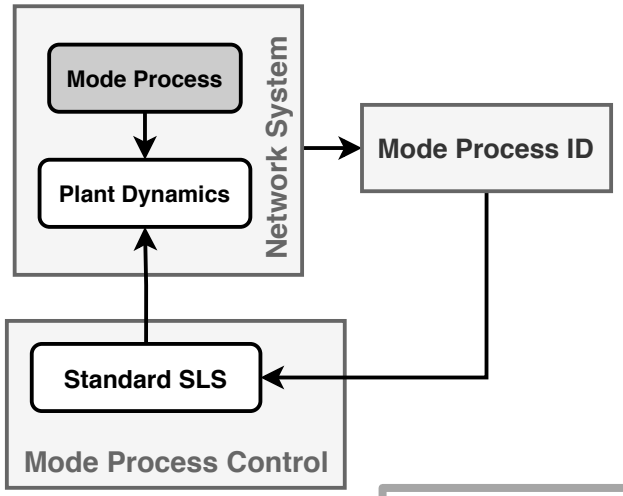
Non-Robust w/ PLP:

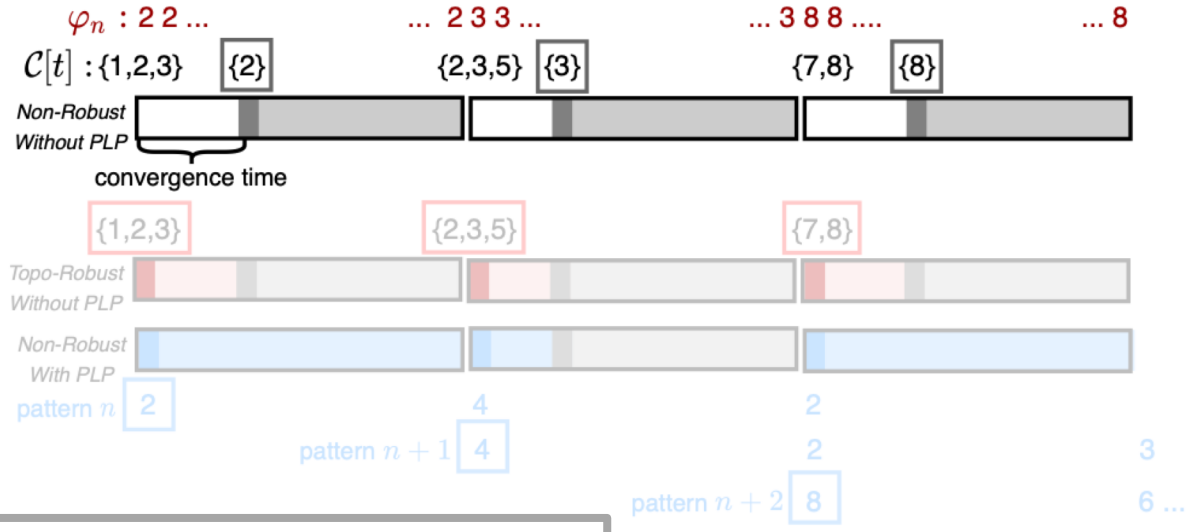


* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Three Controller Frameworks

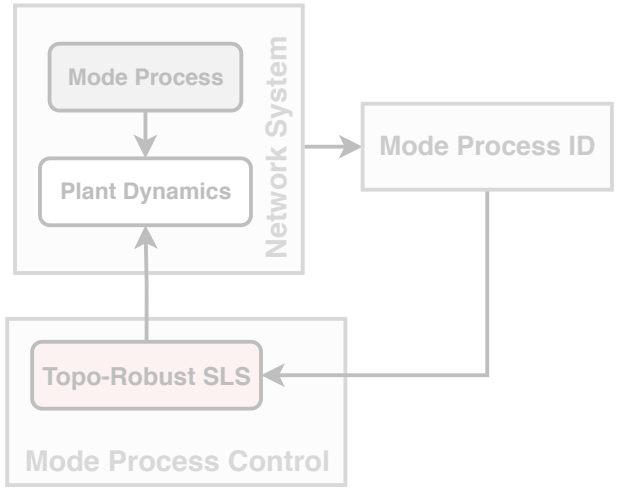
Non-Robust w/o PLP (NR):



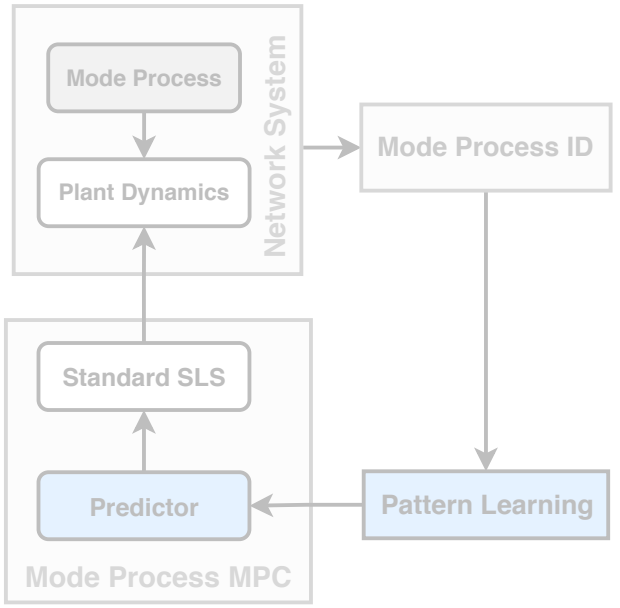


("PLP" = pattern learning and prediction)

Topologically-Robust w/o PLP:



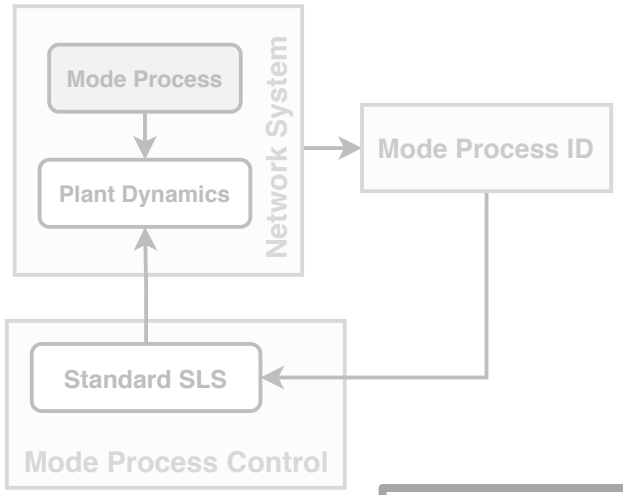
Non-Robust w/ PLP:

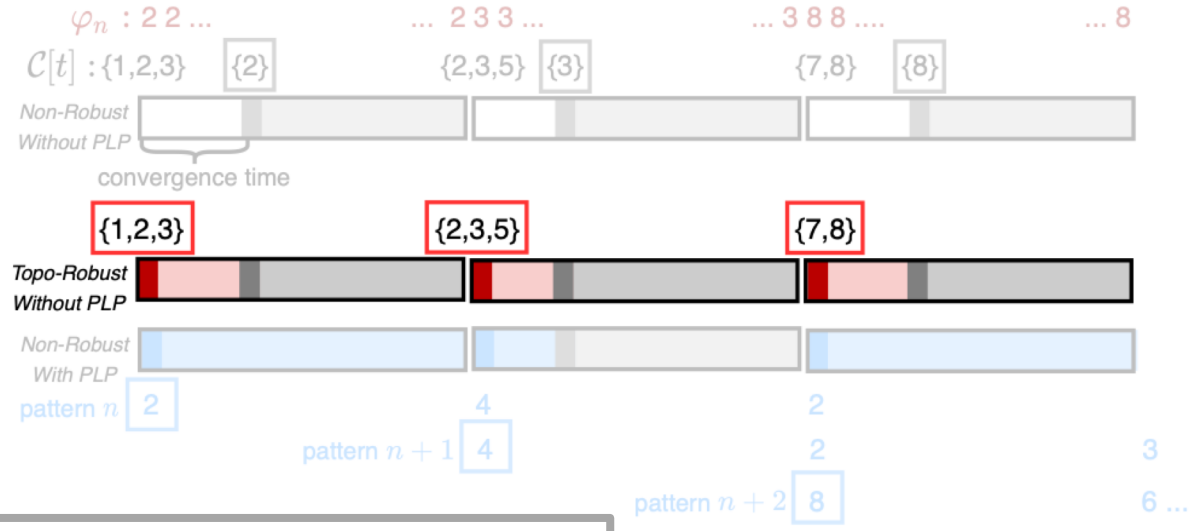


* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Three Controller Frameworks

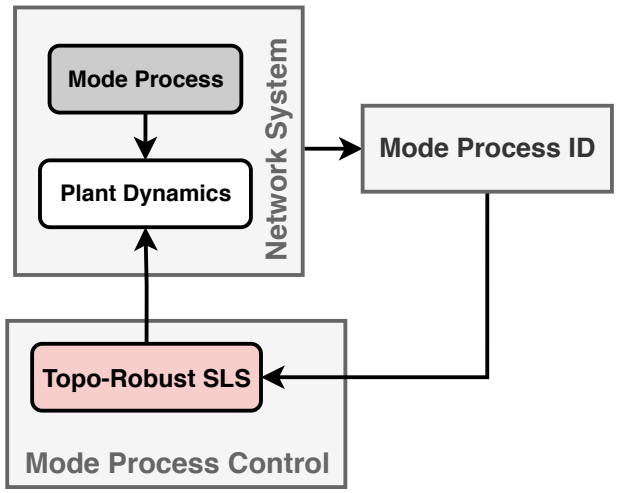
Non-Robust w/o PLP (NR):



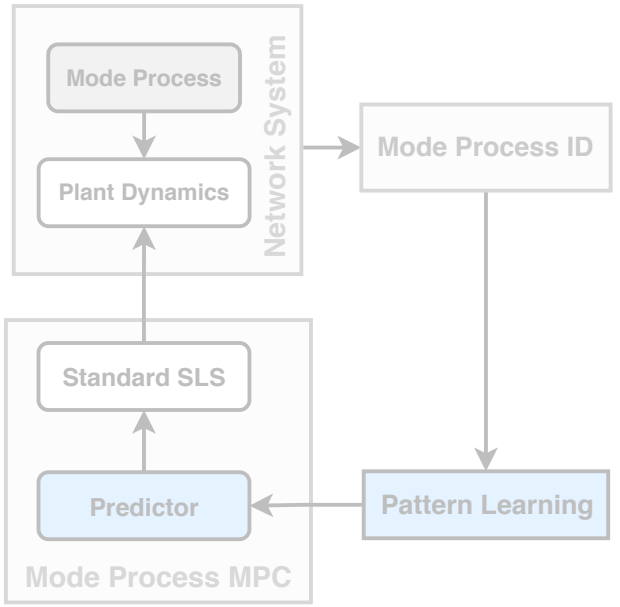


("PLP" = pattern learning and prediction)

Topologically-Robust w/o PLP (TR):



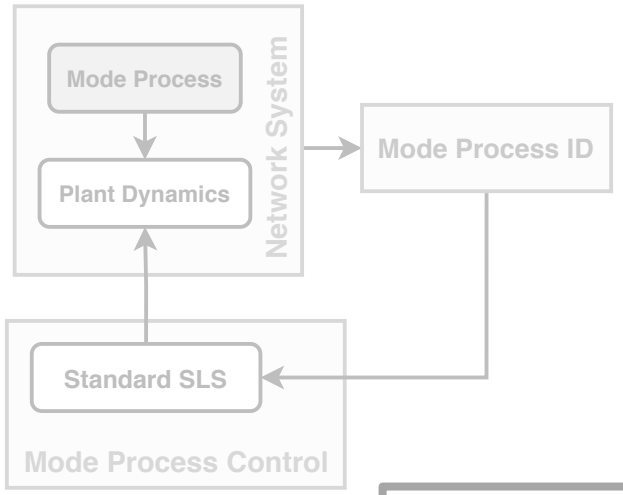
Non-Robust w/ PLP:

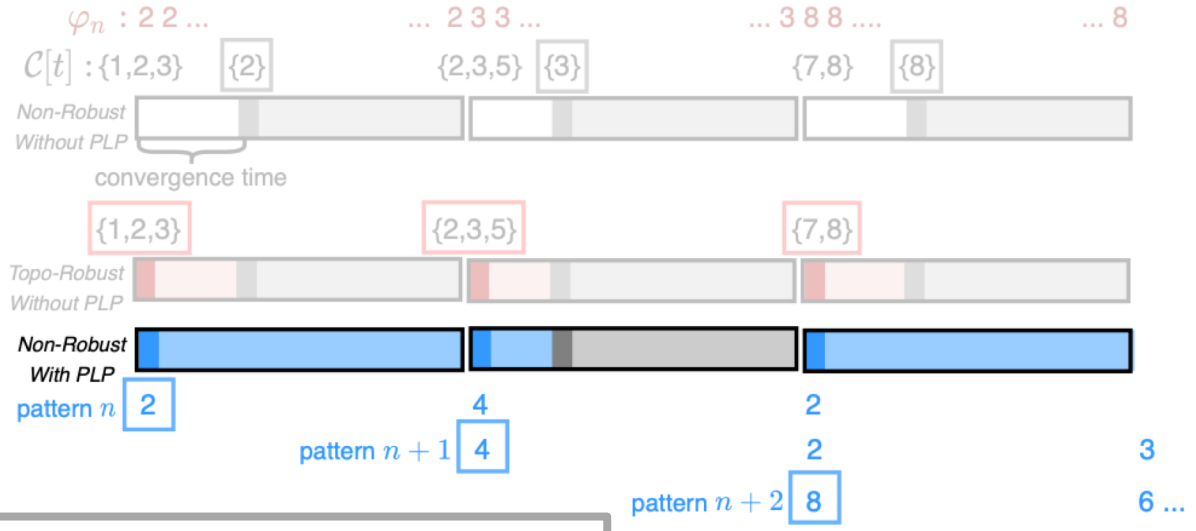


* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Three Controller Frameworks

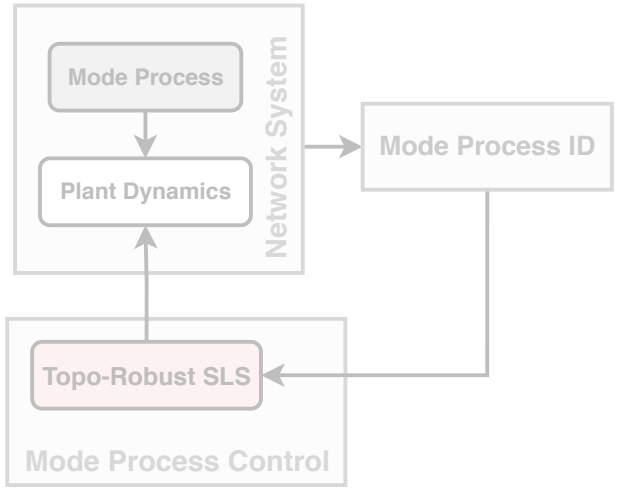
Non-Robust w/o PLP (NR):



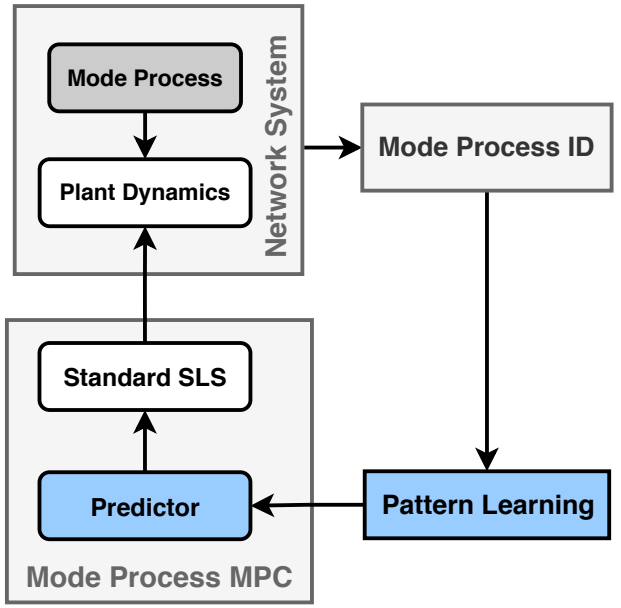


("PLP" = pattern learning and prediction)

Topologically-Robust w/o PLP (TR):



Non-Robust w/ PLP (NRPLP):



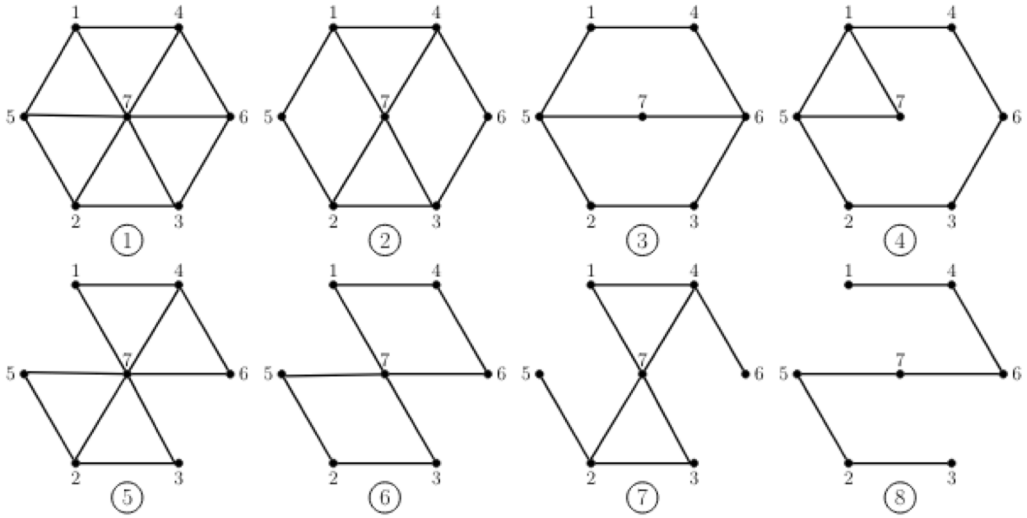
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

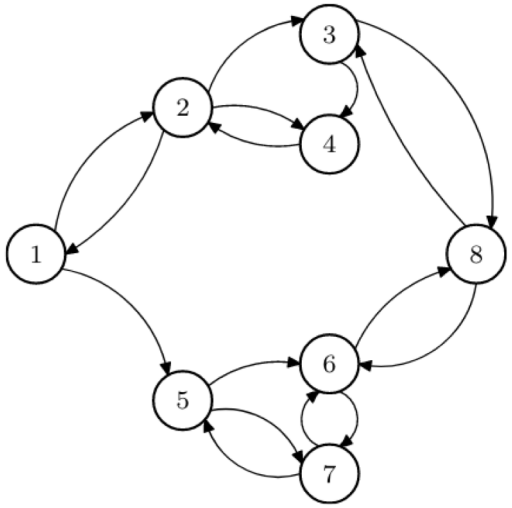
Two Network Systems

$$\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m))$$
$$m \in \{1, \cdots, M\}$$

Hexagon System:

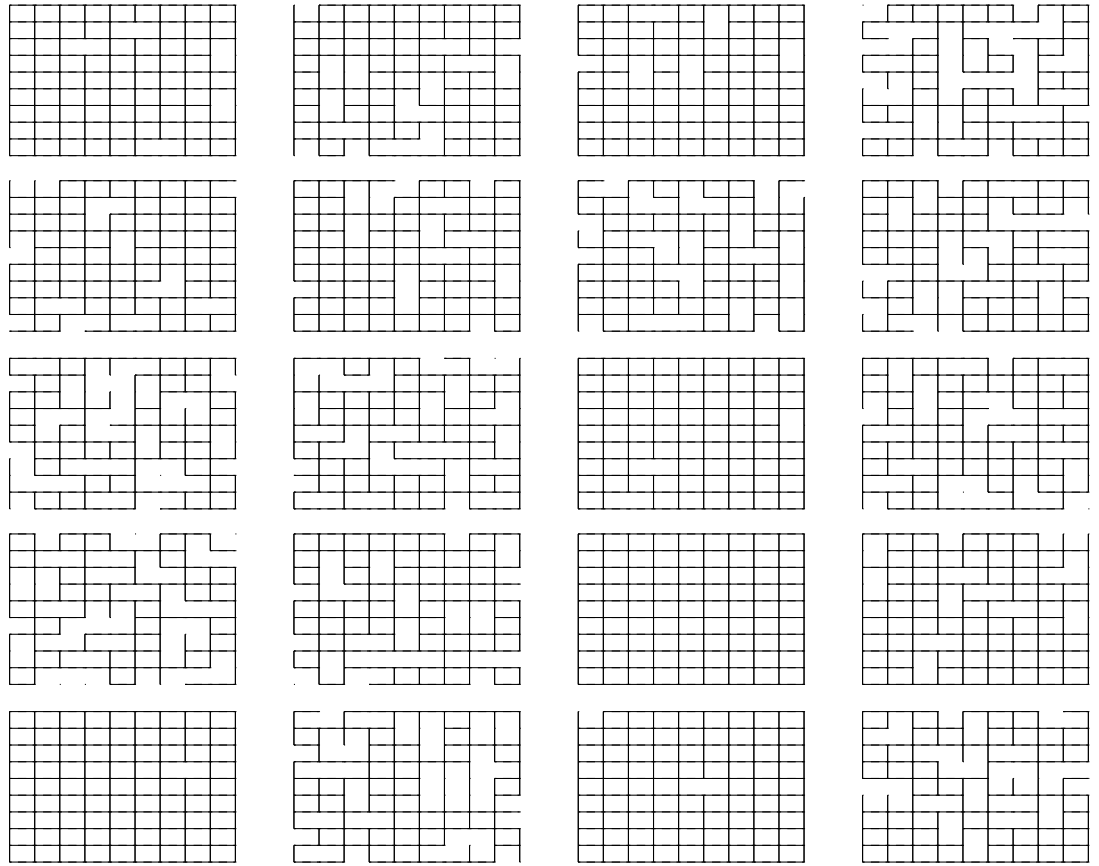
7 nodes, M = 8 modes





Rectangular Grid System:

10x10 = 100 nodes, M = 20 modes

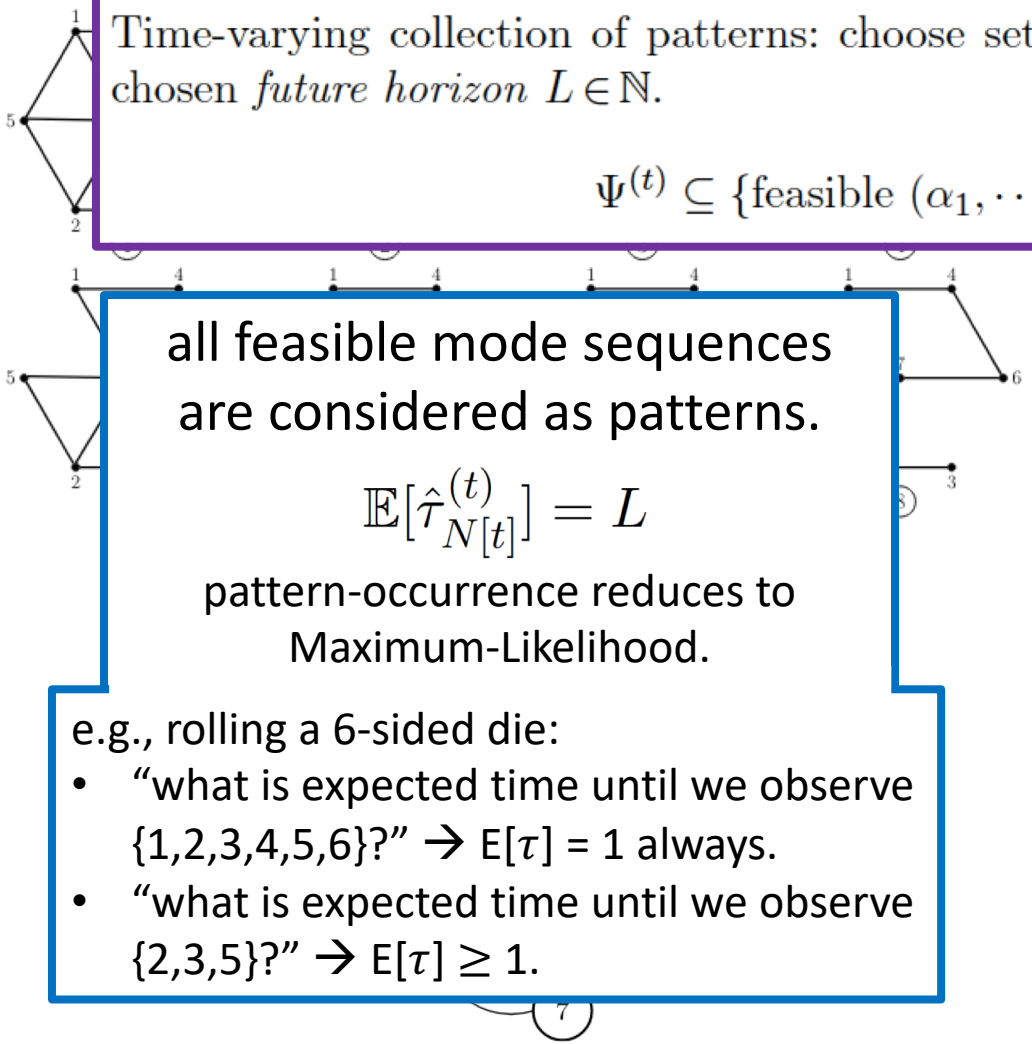


Two Network Systems

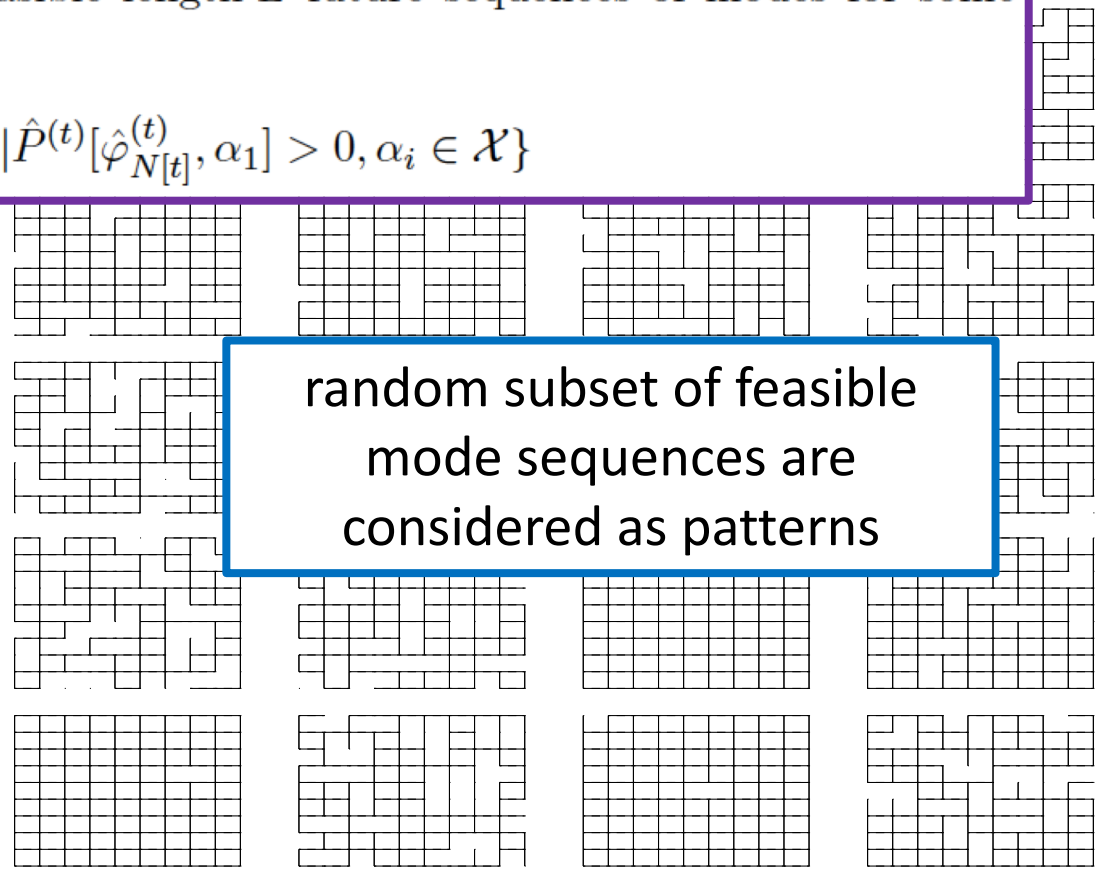
$$\mathcal{G}(m) \triangleq (\mathcal{V}, \mathcal{E}(m))$$
$$m \in \{1, \cdots, M\}$$

Hexagon System:

7 nodes, M = 8



Rectangular Grid System:



Recall for mode process MPC...

Time-varying collection of patterns: choose set of feasible length-L future sequences of modes for some chosen *future horizon* $L \in \mathbb{N}$.

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

all feasible mode sequences are considered as patterns.

$$\mathbb{E}[\hat{\tau}^{(t)}_{N[t]}] = L$$

pattern-occurrence reduces to Maximum-Likelihood.

e.g., rolling a 6-sided die:

- "what is expected time until we observe {1,2,3,4,5,6}?" → E[τ] = 1 always.
- "what is expected time until we observe {2,3,5}?" → E[τ] ≥ 1.

random subset of feasible mode sequences are considered as patterns

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Application: Networks with Time-Varying Topology

LQR Cost: $$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbf{x}[t]^\top Q \mathbf{x}[t] + \mathbf{u}[t]^\top R \mathbf{u}[t]$$

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

Error Norm: (disturbance-rejection)

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

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

Prop(ortion of Time**) Match(**ing Control Law Used**) :**

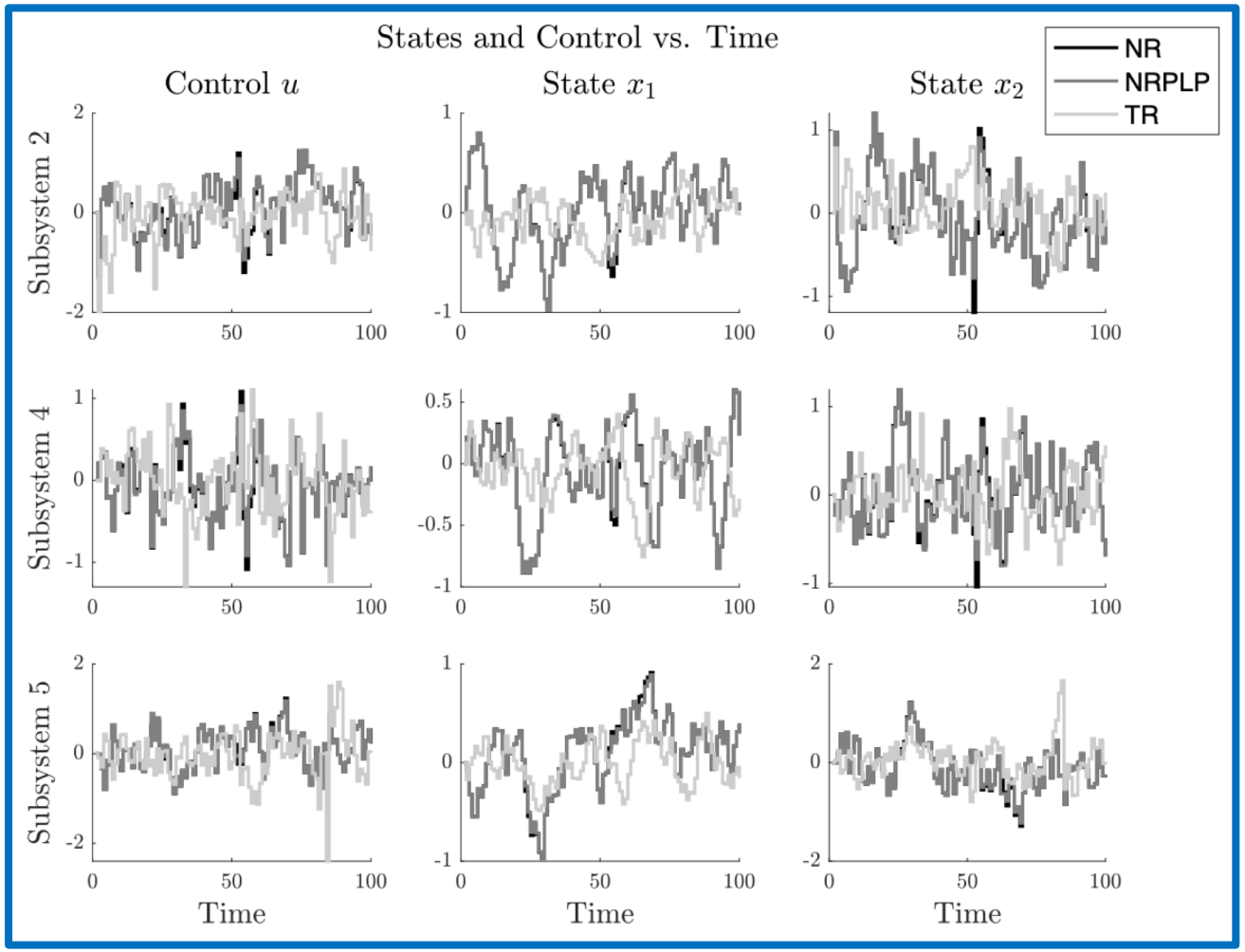
$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbb{1}\{\hat{\varphi}_{N[t]}^{(t)} = \varphi_{N[t]}\}$$

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

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Application: Networks with Time-Varying Topology



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

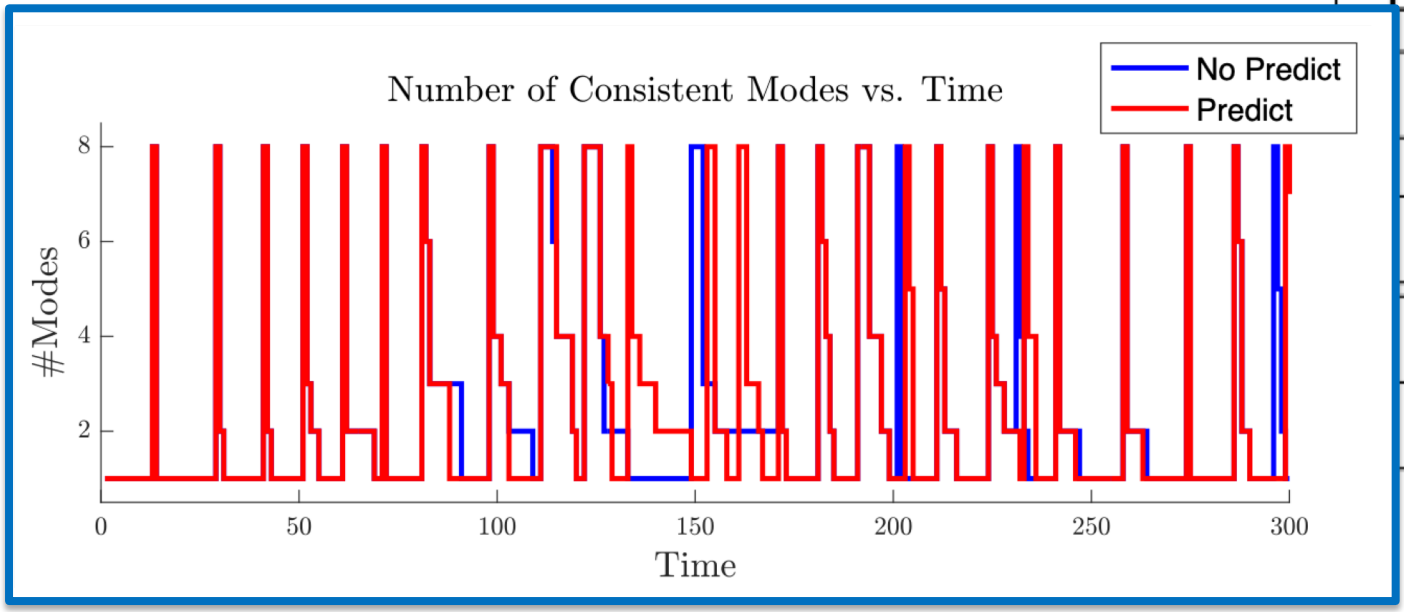
Error Norm	NR	TR	NRPLP
Hexagon	2.9032	1.5984	1.6029
Grid	7.6266	5.5633	5.5349

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

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Application: Networks with Time-Varying Topology



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
	1533	0.16

	NR	TR	NRPLP
Hexagon	11.8314	67.2254	2.2689
Grid	101.3741	X	38.5824

$$\mathcal{C}[0] \triangleq \mathcal{X},\ \mathcal{C}[t] = \left\{ m \in \mathcal{C}[t-1] \mid \bigwedge_{r=T_n}^{t-1} \mathbb{1}\{\|\mathbf{x}[r+1] - A(m)\mathbf{x}[r] - B\mathbf{u}[r]\|_\infty \leq \overline{w}\} \right\}$$

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Application: Networks with Time-Varying Topology

At least 2 ways of improving performance:

- **memorize past patterns**
  → no re-computation of repeating control policies
- **predict future patterns**
  → schedule control policies in advance

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

Implementation Details + Runtime:

- SLS is localized. Mode process ID and pattern learning are centralized.
- Unknown topologies → SLS computes a new law for every new mode observed.
- NR and TR (without PLP) does not store the SLS control law into memory when it is learned. → "*memorize past patterns → no re-computation of control law needed for those patterns*".
- Topology-robust SLS solves for a single control law to stabilize multiple topologies simultaneously → longer runtime than standard SLS.

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

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Application: Networks with Time-Varying Topology

At least 2 ways of improving performance:

- **memorize past patterns**
  → no re-computation of repeating control policies
- **predict future patterns**
  → schedule control policies in advance

Prop(ortion of Time**) Match(**ing Control Law Used**) :**

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbb{1}\{\hat{\varphi}_{N[t]}^{(t)} = \varphi_{N[t]}\}$$

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

Proportion of Time Spent Using Matching Control Law:

- NRPLP consistently uses the matching control law more often than NR → pattern-learning can be an additional mode estimation algorithm. → "*predict future patterns → schedule control policies in advance*".
- Increasing the number of patterns in the pattern collection → more accurate consistent mode estimates.
- Decreasing the number of patterns in the pattern collection → increases expected minimum occurrence time → PLP more closely resembles the performance without PLP.

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

* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Application: Networks with Time-Varying Topology

At least 2 ways of improving performance:

- **memorize past patterns**
  → no re-computation of repeating control policies
- **predict future patterns**
  → schedule control policies in advance

LQR Cost:
$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbf{x}[t]^\top Q \mathbf{x}[t] + \mathbf{u}[t]^\top R \mathbf{u}[t]$$

Error Norm: (disturbance-rejection):
$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \|\mathbf{x}[t]\|_2$$

Controller Performance + Effort:

- LQR cost for TR is largest although error norm is one of the smallest → there is more control effort that goes into it.
- Stabilization performance of NRPLP is just as good as TR.
- *Implication*: For a good choice of pattern collections, appending PLP to a non-robust controller could be used as an efficient alternative to a robust controller.

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

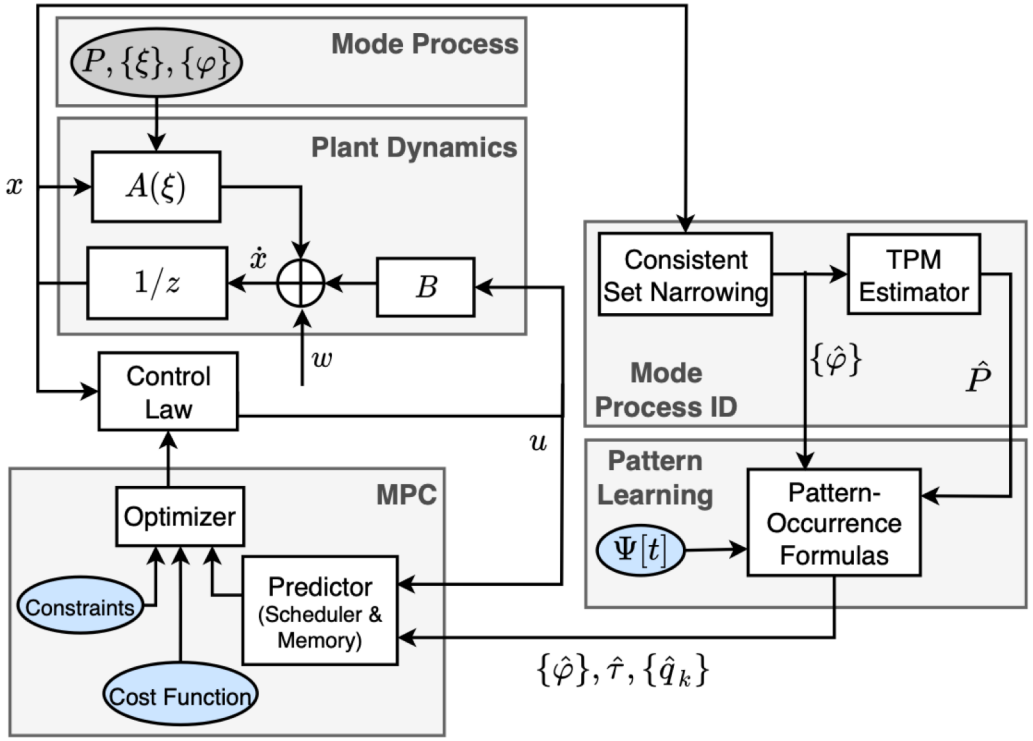
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Summary: Jump Stochastic Systems

Control/estimation for *jump stochastic systems* can be achieved by learning **patterns** in the system behavior.

Learning patterns improves performance in at least 2 ways:

- **memorize past patterns**
  → **no re-computation of repeating control policies**
- **predict future patterns**
  → **schedule control policies in advance**



Systems that have jump patterns:

- Networks with time-varying topology (e.g., power grid)
  - **patterns** = power outages, line failures
- Vehicle traffic congestion
  - **patterns** = intersection snapshots
- Epidemic spread
  - **patterns** = contact-tracing data, travel routines

* SooJean Han and Soon-Jo Chung. *Incremental Nonlinear Stability Analysis of Stochastic Systems Perturbed by Lévy Noise*, International Journal of Robust and Nonlinear Control, 2022.
* SooJean Han. *Localized Learning of Robust Controllers for Networked Systems with Dynamic Topology*. L4DC, 2020.
* SooJean Han, Soon-Jo Chung, and John C. Doyle. *Predictive Control of Linear Discrete-Time Markovian Jump Systems via the Analysis of Recurrent Patterns*, Automatica, submitted 2022.

Thank you!

감사합니다!

Caltech