An Optimal Architecture for Decentralized Control over Posets

Parikshit Shah  Pablo A. Parrilo

Laboratory for Information and Decision Systems
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Motivation

Many decision-making problems are large-scale and complex.

Complexity, cost, physical constraints $\Rightarrow$ Decentralization.

Fully distributed control is notoriously hard.

A common underlying theme: flow of information.

What are the right language and tools to think about flow of information?

Contributions

A framework to reason about information flow in terms of partially ordered sets (posets).

An architecture for decentralized control, based on Möbius inversion, with provable optimality properties.
Motivation

- Many decision-making problems are large-scale and complex.
- Complexity, cost, physical constraints $\Rightarrow$ Decentralization.
- Fully distributed control is notoriously hard.
- A common underlying theme: flow of information.
- What are the right language and tools to think about flow of information?

Contributions

A framework to reason about information flow in terms of partially ordered sets (posets).

An architecture for decentralized control, based on Möbius inversion, with provable optimality properties.
Motivation

- Many interesting examples can be unified in this framework.
- Example: Nested Systems [Voulgaris00].

Emphasis: *Flow of information*. Can abstract away this flow of information to picture on right.

Natural for problems of causal or hierarchical nature.
Basic Machinery: Posets and Incidence Algebras.
Decentralized control problems and posets.
$\mathcal{H}_2$ case: state-space solution
Zeta function, Möbius inversion
Controller architecture
Partially ordered sets (posets)

Definition

A poset $\mathcal{P} = (P, \preceq)$ is a set $P$ along with a binary relation $\preceq$ which satisfies for all $a, b, c \in P$:

1. $a \preceq a$ (reflexivity)
2. $a \preceq b$ and $b \preceq a$ implies $a = b$ (antisymmetry)
3. $a \preceq b$ and $b \preceq c$ implies $a \preceq c$ (transitivity).

Will deal initially with finite posets (i.e. $|P|$ is finite).

Will relate posets to decentralized control.
Incidence Algebras

Definition

The set of functions $f : P \times P \rightarrow \mathbb{Q}$ with the property that $f(x, y) = 0$ whenever $y \preceq x$ is called the incidence algebra $I$.

- Concept developed and studied in [Rota64] as a unifying concept in combinatorics.
- For finite posets, elements of the incidence algebra can be thought of as matrices with a particular sparsity pattern.
Example

\[ \begin{bmatrix}
  a & b & c \\
  a & * & 0 & 0 \\
  b & * & * & 0 \\
  c & * & 0 & * \\
\end{bmatrix} \]
Example

- Closure under addition and scalar multiplication.
- What happens when you multiply two such matrices?

\[
\begin{bmatrix}
* & 0 & 0 \\
* & * & 0 \\
* & 0 & * \\
\end{bmatrix}
\begin{bmatrix}
* & 0 & 0 \\
* & * & 0 \\
* & 0 & * \\
\end{bmatrix}
= 
\begin{bmatrix}
* & 0 & 0 \\
* & * & 0 \\
* & 0 & * \\
\end{bmatrix}
\]

- Not a coincidence!
Example

- Closure under addition and scalar multiplication.
- What happens when you multiply two such matrices?

\[
\begin{bmatrix}
* & 0 & 0 \\
* & * & 0 \\
* & 0 & *
\end{bmatrix}
\begin{bmatrix}
* & 0 & 0 \\
* & * & 0 \\
* & 0 & *
\end{bmatrix} =
\begin{bmatrix}
* & 0 & 0 \\
* & * & 0 \\
* & 0 & *
\end{bmatrix}
\]

- Not a coincidence!
Incidence Algebras

- Closure properties are true in general for all posets.

**Lemma**

Let $\mathcal{P}$ be a poset and $\mathcal{I}$ be its incidence algebra. Let $A, B \in \mathcal{I}$ then:

1. $c \cdot A \in \mathcal{I}$
2. $A + B \in \mathcal{I}$
3. $AB \in \mathcal{I}$.

Thus the incidence algebra is an associative algebra.

- A simple corollary: Since $I$ is in every incidence algebra, if $A \in \mathcal{I}$ and invertible, $A^{-1} \in \mathcal{I}$.
- Properties useful in Youla domain.
Incidence Algebras

- Closure properties are true in general for all posets.

Lemma

Let $P$ be a poset and $\mathcal{I}$ be its incidence algebra. Let $A, B \in \mathcal{I}$ then:

1. $c \cdot A \in \mathcal{I}$
2. $A + B \in \mathcal{I}$
3. $AB \in \mathcal{I}$.

Thus the incidence algebra is an associative algebra.

- A simple corollary: Since $1$ is in every incidence algebra, if $A \in \mathcal{I}$ and invertible, $A^{-1} \in \mathcal{I}$.
- Properties useful in Youla domain.
Control problem

A given matrix $P$.
Design $K$.
Interconnect $P$ and $K$.
A given matrix $P$.

Design $K$.

Interconnect $P$ and $K$.

\[
f(P, K) = P_{11} + P_{12}K(I - P_{22}K) - P_{21}.
\]
Control problem

- A given matrix $P$.
- Design $K$.
- Interconnect $P$ and $K$

$$f(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$
A given matrix $P$.

Design $K$.

Interconnect $P$ and $K$

$$f(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$ 

Find “best” $K$. 
Control problem

A given matrix $P$.
Design $K$.
Interconnect $P$ and $K$

$$f(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$ 

Find “best” $K$. 
Modeling decentralized control problems using posets

- All the action happens at $P_{22} = G$. Focus here.
- $G$ (called the plant) interacts with the controller.
- Plant divided into subsystems:

$$
\begin{bmatrix}
G_{11} & 0 & 0 \\
G_{21} & G_{22} & 0 \\
G_{31} & 0 & G_{33}
\end{bmatrix}
$$

Subsystem Outputs
Let $G$ be the transfer function matrix of the plant. We divide up the plant into subsystems:
Let $G$ be the transfer function matrix of the plant. We divide up the plant into subsystems:
Modeling decentralized control problems using posets

Denote this by $1 \preceq 2$ and $1 \preceq 3$.

Subsystems 2 and 3 are in cone of influence of 1.

This relationship is a causality relation between subsystems.

We call systems with $G \in \mathcal{I}$ poset-causal systems.
Controller Structure

- Given a poset causal plant $G \in \mathcal{I}$.
- Decentralization constraint: mirror the information structure of the plant.
- In other words we want poset-causal $K \in \mathcal{I}$.
- Similar causality interpretation.
- Intuitively, $i \preceq j$ means subsystem $j$ is more information rich.
- The poset arranges the subsystems according to the amount of information richness.
Given a poset causal plant $G \in \mathcal{I}$.

Decentralization constraint: mirror the information structure of the plant.

In other words we want poset-causal $K \in \mathcal{I}$.

Similar causality interpretation.

Intuitively, $i \leq j$ means subsystem $j$ is more information rich.

The poset arranges the subsystems according to the amount of information richness.
Examples of poset systems

- Independent subsystems
- Nested systems
- Closures of directed acyclic graphs
Goal is to capture what is essential about causal decision-making.

- Elements of the poset do not necessarily have to represent only subsystems.
  - “Standard” case discussed earlier corresponds to the product of the spatial interaction poset (space) and a linear chain (time)
  - Posets model branching time, nondeterminism, etc.
  - Posets in space-time (e.g., distributed systems)

- Controller structure does not necessarily have to mirror plant.
  - Generalizations via Galois connections
Classical work: Witsenhausen, Radner, Ho-Chu.

Mullans-Elliot (1973), linear systems on partially ordered time sets

Voulgaris (2000), showed that a wide class of distributed control problems became convex through a Youla-type reparametrization.

Rotkowitz-Lall (2002) introduced *quadratic invariance* (QI) an important unifying concept for convexity in decentralized control.


Swigart-Lall (2010) gave a state-space solution for the two-controller case, via a spectral factorization approach.

Shah-P. (2010), provided a full solution for all posets, with controller degree bounds. Separability a key idea, which is missing in past work. Introduced simple Möbius-based architecture (in slightly different form).
Given a system $P$ with plant $G$, find a stabilizing controller $K \in \mathcal{I}$.

\[
\begin{align*}
\text{minimize}_K & \quad \|f(P, K)\| \\
\text{subject to} & \quad K \text{ stabilizes } P \\
& \quad K \in \mathcal{I}.
\end{align*}
\]

Here $f(P, K) = P_{11} + P_{12}K(I - GK)^{-1}P_{21}$ is the closed loop transfer function.

Problem is nonconvex.

Standard approach: reparametrize the problem by getting rid of the nonconvex part of the objective.
Convex reparametrization

- "Youla domain" technique: define $R = K(I - GK)^{-1}$.

\[
\text{minimize } \| \hat{P}_{11} + \hat{P}_{12} R \hat{P}_{21} \|
\]
\[
\text{subject to } R \text{ stable}
\]
\[
R \in \mathcal{I}.
\]

Algebraic structure of $\mathcal{I}$ allows to rewrite a convex constraint in $K$ into a convex constraint in $R$.

- Main difficulty: Infinite dimensional problem.
- Can be approximated by various techniques, but there are drawbacks.
- Desire state-space techniques. Advantages:
  1. Computationally efficient
  2. Degree bounds
  3. Provide insight into structure of optimal controller.
State-Space Setup

- Have state feedback system:

\[
\begin{align*}
    x[t + 1] &= Ax[t] + Bu[t] + w[t] \\
    y[t] &= x[t] \\
    z[t] &= Cx[t] + Du[t]
\end{align*}
\]

- Wish to find controller \( u = Kx \) which is stabilizing and optimal.

\[
\min_K \| P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \|^2
\]

\( K \in \mathcal{I} \)

\( K \) stabilizing.

- Key property we exploit: separability of the \( \mathcal{H}_2 \) norm.
Decentralized Control Problem

- System poset causal: $A, B \in \mathcal{I}(\mathcal{P})$.
- Solve:

\[ \min_K \|P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\|^2 \]

$K \in \mathcal{I}$

$K$ stabilizing.

- Due to state-feedback: $P_{21} = (zI - A)^{-1}$.
- Define $Q := K(I - GK)^{-1}P_{21}$.
- Problem reduces to:

\[ \min_Q \|P_{11} + P_{12}Q\|^2 \]

$Q \in \mathcal{I}$

$Q$ stabilizing.
\( \mathcal{H}_2 \) Decomposition Property

- Let \( G = [G_1, \ldots, G_k] \).

\[
\| G \|^2 = \sum_{i=1}^{k} \| G_i \|^2.
\]

- This separability property is the key feature we exploit.

Example

\[
\begin{align*}
\min & \quad \left\| P_{11} + P_{12} \begin{bmatrix} Q_{11} & 0 & 0 \\ Q_{21} & Q_{22} & 0 \\ Q_{31} & 0 & Q_{33} \end{bmatrix} \right\|^2 \\
\text{s.t.} & \quad Q \text{ stabilizing.}
\end{align*}
\]

\[
\begin{align*}
\min. & \quad \left\| P_{11}(1) + P_{12} \begin{bmatrix} Q_{11} \\ Q_{21} \\ Q_{31} \end{bmatrix} \right\|^2 + \| P_{11}(2) + P_{12}(2)Q_{22} \|^2 \\
\text{s.t.} & \quad Q \text{ stabilizing.}
\end{align*}
\]
This decomposition idea extends to all posets.

**Theorem (Shah-P., CDC2010/11)**

*Problem can be reduced to decoupled problems:*

\[
\begin{align*}
\text{minimize} & \quad \| P_{11}(j) + P_{12}(\uparrow j) Q^{\uparrow j} \|_2 \\
\text{subject to} & \quad Q^{\uparrow j} \text{ stabilizing} \\
& \quad \text{for all } j \in P.
\end{align*}
\]

- Optimal $Q$ can be obtained by solving a set of decoupled centralized sub-problems.
- Each sub-problem requires solution of a Riccati equation.
\(H_2\) State Space Solution

- Can recover \(K\) from optimal \(Q\).
- \(Q\) and \(K\) are in bijection, \(K = QP_{21}^{-1}(I + P_{22}QP_{21}^{-1})^{-1}\).
- Further analysis gives:
  1. Explicit state-space formulae.
  2. Controller degree bounds.
  3. Insight into structure of optimal controller.
Great. We solved the problem for all posets! But, what is the structure here?

- Swigart-Lall (2010) had a nice interpretation for the two-controller case, in terms of the first controller estimating the state of the second subsystem.

- No “obvious” generalizations:
  - In general, do not have enough information to predict upstream states. Also, there may be incomparable states.
  - More importantly, too many predictions from downstream! How to combine them?
General Controller Architecture

What is the “right” architecture?

Ingredients:
1. Lower sets and upper sets
2. Local variables (partial state predictions)

Simple separation principle

Optimality of architecture for $\mathcal{H}_2$. 
General Controller Architecture

What is the “right” architecture?

Ingredients:
1. Lower sets and upper sets
2. Local variables (partial state predictions)

Simple separation principle

Optimality of architecture for $\mathcal{H}_2$. 
Subsystems have “local future” and “local past”
Subsystems make “local partial predictions” of their actions for everybody on their “local future”
Each subsystem computes “error” between (many!) partial predictions and reality
Choose “local” control action
Combine all local actions from past, and implement action
Each “node” in $P$ is a subsystem with state $x_i$ and input $u_i$.

- Upper set: $\uparrow p = \{ q \mid p \preceq q \}$.
- Corresponds to unknown information.

- Lower set: $\downarrow p = \{ q \mid q \preceq p \}$.
- Corresponds to known information.

$u_i$ has access to $x_j$ for $j \in \downarrow i$. 

![Diagram showing upper and lower sets with nodes and arrows indicating access.](attachment:image.png)
Upper sets and lower sets

- Each “node” in $\mathcal{P}$ is a subsystem with state $x_i$ and input $u_i$.
- Upper set: $\uparrow p = \{ q \mid p \preceq q \}$.
- Corresponds to unknown information.

- Lower set: $\downarrow p = \{ q \mid q \preceq p \}$.
- Corresponds to known information.
- $u_i$ has access to $x_j$ for $j \in \downarrow i$. 
Local Variables

- Overall state $x$ and input $u$ are **global** variables.
- Subsystems carry **local** copies.

- Local variable $X_i : \uparrow i \rightarrow \mathbb{R}$.
  - Can think of it as a vector in $\mathbb{R}^{|P|}$.

- Two local variables of interest:
  1. $X$: $X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U$: $U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$. 
Local Variables

- Overall state $x$ and input $u$ are **global** variables.
- Subsystems carry **local** copies.

- Local variable $X_i: \uparrow i \rightarrow \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$

- Two local variables of interest:
  1. $X$: $X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U$: $U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$. 

```
Local Variables

- Overall state $x$ and input $u$ are **global** variables.
- Subsystems carry **local** copies.

- Local variable $X_i: \uparrow i \rightarrow \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$

- Two local variables of interest:
  1. $X$: $X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U$: $U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$. 
```
Local Variables

- Overall state $x$ and input $u$ are global variables.
- Subsystems carry local copies.

- Local variable $X_i : \uparrow i \to \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$.

- Two local variables of interest:
  1. $X: X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U: U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$. 
Local Products

- Local gain: $G(i) : \uparrow i \times \uparrow i \rightarrow \mathbb{R}$. Think of it as zero-padded matrix:

$$G(2) = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & G_{22} & 0 & G_{24} \\
0 & 0 & 0 & 0 \\
0 & G_{42} & 0 & G_{44}
\end{bmatrix}$$

- Define $G = \{ G(1), \ldots, G(s) \}$.
- Local Product: $G \circ X$ defined columnwise via:

$$(G \circ X)_i = G(i)X_i.$$ 

- If $Y = G \circ X$, then local variables $(X_i, Y_i)$ decoupled.
Zeta and Möbius

For any poset $\mathcal{P}$, two distinguished elements of its incidence algebra:

- The Zeta matrix is

$$\zeta_{\mathcal{P}}(x, y) = \begin{cases} 
1, & \text{if } y \preceq x \\
0, & \text{otherwise}
\end{cases}$$

- Its inverse is the Möbius matrix of the poset:

$$\mu_{\mathcal{P}} = \zeta_{\mathcal{P}}^{-1}.$$ 

E.g., for the poset below, we have:

$$\zeta_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 0 & 1 \end{bmatrix}, \quad \mu_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\
-1 & 1 & 0 \\
-1 & 0 & 1 \end{bmatrix}$$
Zeta and Möbius

For any poset $\mathcal{P}$, two distinguished elements of its incidence algebra:

- The Zeta matrix is
  \[
  \zeta_\mathcal{P}(x, y) = \begin{cases} 
  1, & \text{if } y \preceq x \\
  0, & \text{otherwise}
  \end{cases}
  \]

- Its inverse is the Möbius matrix of the poset:
  \[
  \mu_\mathcal{P} = \zeta_\mathcal{P}^{-1}.
  \]

E.g., for the poset below, we have:

\[
\zeta_\mathcal{P} = \begin{bmatrix} 
1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 0 & 1 
\end{bmatrix}, \quad \mu_\mathcal{P} = \begin{bmatrix} 
1 & 0 & 0 \\
-1 & 1 & 0 \\
-1 & 0 & 1 
\end{bmatrix}
\]
Möbius inversion

Given \( f : P \to \mathbb{Q} \), we can define

\[
(\zeta f)(x) = \sum_y \zeta(x, y)f(y), \quad (\mu f)(x) = \sum_y \mu(x, y)f(y).
\]

These operations are obviously inverses of each other.

For our example:

\[
\zeta(a_1, a_2, a_3) = (a_1, a_1 + a_2, a_1 + a_3), \quad \mu(b_1, b_2, b_3) = (b_1, b_2 - b_1, b_3 - b_1).
\]

Möbius inversion formula

\[
g(y) = \sum_{x \preceq y} h(x) \iff h(y) = \sum_{x \preceq y} \mu(x, y)g(x)
\]
Möbius inversion

Given $f : P \to \mathbb{Q}$, we can define

$$(\zeta f)(x) = \sum_y \zeta(x, y)f(y), \quad (\mu f)(x) = \sum_y \mu(x, y)f(y).$$

These operations are obviously inverses of each other.

For our example:

$$\zeta(a_1, a_2, a_3) = (a_1, a_1 + a_2, a_1 + a_3), \quad \mu(b_1, b_2, b_3) = (b_1, b_2 - b_1, b_3 - b_1).$$

Möbius inversion formula

$$g(y) = \sum_{x \leq y} h(x) \iff h(y) = \sum_{x \leq y} \mu(x, y)g(x).$$
Möbius inversion

Given $f : P \rightarrow \mathbb{Q}$, we can define

$$(\zeta f)(x) = \sum_y \zeta(x, y)f(y), \quad (\mu f)(x) = \sum_y \mu(x, y)f(y).$$

These operations are obviously inverses of each other.

For our example:

$\zeta(a_1, a_2, a_3) = (a_1, a_1 + a_2, a_1 + a_3), \quad \mu(b_1, b_2, b_3) = (b_1, b_2 - b_1, b_3 - b_1).$

Möbius inversion formula

$$g(y) = \sum_{x \leq y} h(x) \quad \Leftrightarrow \quad h(y) = \sum_{x \leq y} \mu(x, y)g(x)$$
Möbius inversion: examples

- If $P$ is a chain: then $\zeta$ is “integration”, $\mu := \zeta^{-1}$ is “differentiation”.
- If $P$ is the subset lattice, then $\mu$ is inclusion-exclusion
- If $P$ is the divisibility integer lattice, then $\mu$ is the number-theoretic Möbius function.
- Many others: vector spaces, faces of polytopes, graphs/circuits, ...
 Möbius inversion is local

- Key insight: Möbius inversion respects the poset structure.
- No additional communication requirements to compute them.
- Thus, can view as operators on local variables: $\zeta(X), \mu(X)$.

$$\zeta(X) = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_2 + x_2(1) & x_3 + x_3(1) & x_4 \\ x_3 & x_3(1) & x_4 + x_4(1) & x_4 + x_4(2) + x_4(3) \\ x_4 & x_4(1) & x_4(2) + x_4(1) & x_4(3) + x_4(1) \end{bmatrix}$$

$$\mu(X) = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2(1) & x_2 - x_2(1) & x_3 - x_3(1) & x_4 \\ x_3(1) & x_3(2) & x_3(3) - x_3(1) & x_4 + x_4(1) - x_4(2) \\ x_4 & x_4(1) & x_4(2) - x_4(1) & x_4(3) - x_4(1) \end{bmatrix}$$
Let the system dynamics be $x[t + 1] = Ax[t] + Bu[t]$, where $A, B \in \mathcal{I}(\mathcal{P})$.

Define controller state variables $X_{ij}$ for $j \leq i$, where $X_{ii} = x_i$.

Propose a control law:

$$U = \zeta(G \circ \mu(X)).$$

where $G = \{G(1), \ldots, G(s)\}$. 
Controller Architecture

\[ G \] 

\[ \bar{\mu} \] 

\[ \bar{\zeta} \] 

\[ F(1) \] 

\[ F(2) \] 

\[ F(s) \] 

\[ U^1 \] 

\[ U^s \] 

\[ X^1 \] 

\[ X^s \] 

\[ \mu(U)^1 \] 

\[ \mu(U)^s \] 

\[ \mu(X)^1 \] 

\[ \mu(X)^s \] 

Simulator
Controller Architecture

- Let the system dynamics be $x[t + 1] = Ax[t] + Bu[t]$, where $A, B \in \mathcal{I}(\mathcal{P})$.
- Define controller state variables $X_{ij}$ for $j \leq i$, where $X_{ii} = x_i$.
- Propose a control law:
  $$U = \zeta(G \circ \mu(X)).$$
  where $G = \{G(1), \ldots, G(s)\}$.
- Can compactly write closed-loop dynamics as matrix equations:
  $$X[t + 1] = AX[t] + B\zeta(G \circ \mu(X[t])).$$

  - Each column corresponds to a different subsystem
  - Equations have structure of $\mathcal{I}$, only need entries with $j \leq i$
  - Diagonal is the plant, off-diagonal is the controller
  - Since $\zeta$ and $\mu$ are local, so is the closed-loop
Controller Architecture: \( U = \zeta(G \circ \mu(X)) \)

- “Local errors” computed by \( \mu(X) \) (differentiation)
- Compute “local corrections”
- Aggregate them via \( \zeta(\cdot) \) (integration)
Separation Principle

- Closed-loop equations:

\[ X[t + 1] = AX[t] + B\zeta(G \circ \mu(X[t])). \]

- Apply \( \mu \), and use the fact that \( \mu \) and \( \zeta \) are inverses:

\[
\mu(X)[t + 1] = A\mu(X)[t] + B(G \circ \mu(X)[t]) \\
= (A + BG) \circ \mu(X).
\]

where \( (A + BG)_i = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i). \)

- “Innovation” dynamics at subsystems decoupled!

- Stabilization easy: simply pick \( G(i) \) to stabilize \( A(\uparrow i, \uparrow i), B(\uparrow i, \uparrow i). \)
Separation Principle

- Closed-loop equations:

\[ X[t + 1] = AX[t] + B\zeta(G \circ \mu(X[t])). \]

- Apply \( \mu \), and use the fact that \( \mu \) and \( \zeta \) are inverses:

\[ \mu(X)[t + 1] = A\mu(X)[t] + B(G \circ \mu(X)[t]) \]
\[ = (A + BG) \circ \mu(X). \]

where \( (A + BG)_i = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i). \)

- “Innovation” dynamics at subsystems decoupled!

- Stabilization easy: simply pick \( G(i) \) to stabilize \( A(\uparrow i, \uparrow i), B(\uparrow i, \uparrow i). \)
Separation Principle

- Closed-loop equations:
  \[ X[t + 1] = AX[t] + B\zeta(G \circ \mu(X[t])). \]

- Apply \( \mu \), and use the fact that \( \mu \) and \( \zeta \) are inverses:
  \[
  \mu(X)[t + 1] = A\mu(X)[t] + B(G \circ \mu(X)[t]) \\
  = (A + BG) \circ \mu(X). 
  \]

  where \( (A + BG)_i = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i). \)

- “Innovation” dynamics at subsystems decoupled!

- Stabilization easy: simply pick \( G(i) \) to stabilize \( A(\uparrow i, \uparrow i), B(\uparrow i, \uparrow i). \)
Optimality

Theorem (Shah-P., CDC2010/11)

$\mathcal{H}_2$-optimal controllers have the described architecture.

- Gains $G(i)$ obtained by solving decoupled Riccati equations.
- States in the controller are precisely predictions $X_{ij}$ for $j \prec i$.
- Controller order is number of intervals in the poset.
Controller architecture

Möbius-inversion controller

\[ U = \zeta(G \circ \mu(X)). \]

Simple and natural structure, for any locally finite poset.

- Can exploit further restrictions (e.g., distributive lattices)
- For product posets, well-understood composition rules for \( \mu \)
- Generalizes many concepts (Youla parameterization, “purified outputs”, etc)
- Extensions to output feedback, different plant/controller posets (Galois connections), …
Conclusions

- Posets provide useful framework to reason about decentralized decision-making on causal or hierarchical structures.
  - Conceptually nice, computationally tractable.
  - Simple controller structure, based on Möbius inversion.
  - $\mathcal{H}_2$-optimal controllers have this structure.
- Want to know more? → www.mit.edu/~pari
  - “A partial order approach to decentralized control”, CDC2010/11.
  - “$\mathcal{H}_2$-optimal decentralized control over posets: a state-space solution for state-feedback,” arXiv:1111.1498.