A Partial Order Approach to Decentralized Control

Parikshit Shah

Joint work with Pablo A. Parrilo

LIDS, EE/CS, MIT

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?

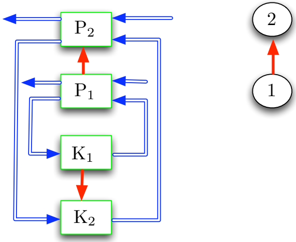
Contributions

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.

Outline

- 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 with finite posets (i.e. $|P|$ is finite).
- Will relate posets to decentralized control.

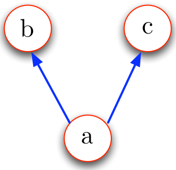
Incidence Algebras

Definition

The set of functions $f : P \times P \to \mathbb{Q}$ with the property that $f(x, y) = 0$ whenever $y \not\leq x$ is called the incidence algebra $\mathcal{I}$.

- 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



$$\begin{array}{c} \\ a \\ b \\ c \end{array} \quad \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix} \end{array}$$

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

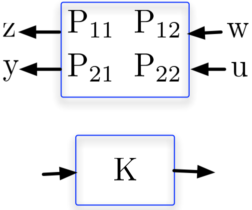
Let $\mathcal{P}$ be a poset and $\mathcal{I}$ be its incidence algebra. Let $A, B \in \mathcal{I}$ then:

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.

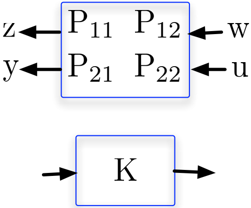
- A simple corollary: Since I is in every incidence algebra, if $A \in \mathcal{I}$ and invertible, $A^{-1} \in \mathcal{I}$.
- Properties useful in Youla domain.

Control problem



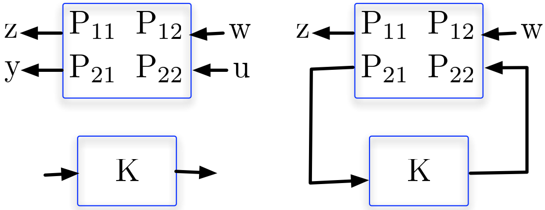
- A given matrix P.
- Design K.
- Interconnect P and K

Control problem



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

Control problem

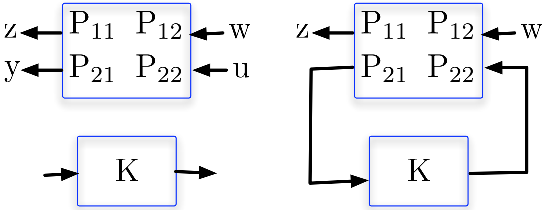


- A given matrix P.
- Design K.
- Interconnect P and K

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

- Find "best" K.

Control problem

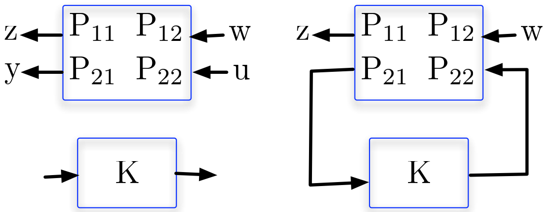


- A given matrix P.
- Design K.
- Interconnect P and K

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

- Find "best" K.

Control problem



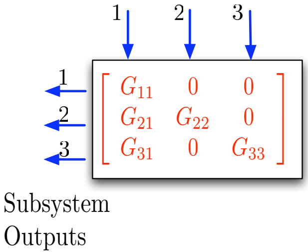
- A given matrix P.
- Design K.
- Interconnect P and K

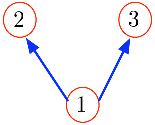
$$f(P,K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$

- Find "best" K.

Modeling decentralized control problems using posets

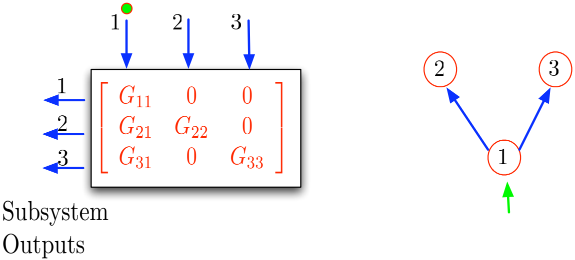
- All the action happens at $P_{22} = G$. Focus here.
- G (called the plant) interacts with the controller.
- Plant divided into subsystems:





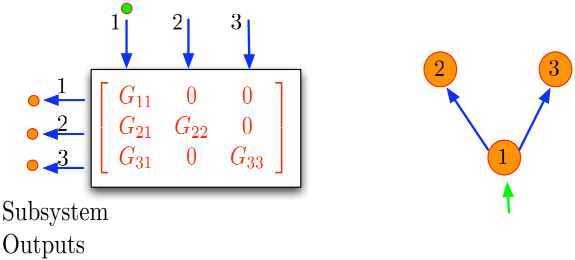
Modeling decentralized control problems using posets

- Let G be the transfer function matrix of the plant. We divide up the plant into subsystems:

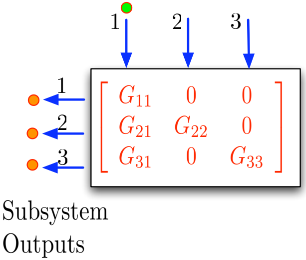


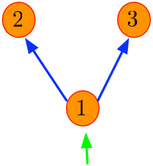
Modeling decentralized control problems using posets

- Let G be the transfer function matrix of the plant. We divide up the plant into subsystems:



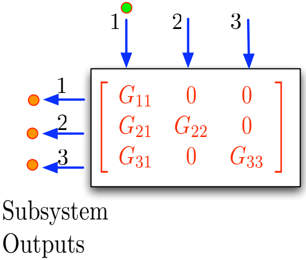
Modeling decentralized control problems using posets

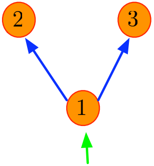




- Denote this by $1 \preceq 2$ and $1 \preceq 3$.
- Subsystems 2 and 3 are in cone of influence of 1
- This relationship is a causality relation between subsystems.
- We call systems with $G \in \mathcal{I}$ poset-causal systems.

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.

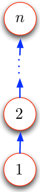
Controller Structure

- 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.

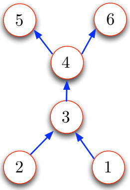
Examples of poset systems

- Independent subsystems

- Nested systems



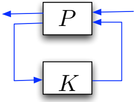
- Closures of directed acyclic graphs



Optimal Control Problem

Given a system P with plant G, find a stabilizing controller $K \in \mathcal{I}$.

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



- 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 $R = K(I - GK)^{-1}$.

$$\begin{array}{ll} \text{minimize} & \|\hat{P}_{11} + \hat{P}_{12} R \hat{P}_{21}\| \\ \text{subject to} & R \text{ stable} \\ & R \in \mathcal{I}. \end{array}$$

- Algebraic structure of $\mathcal{I}$ allows reparametrization.
- Recover via $K = (I + GR)^{-1} R$.
- Extensions:
  1. Can extend to different constraints: Galois connections.
  2. Time-delayed systems.
  3. Spatio-temporal systems.

Posets and Quadratic Invariance

- Quadratic invariance: $K \in S \Rightarrow KGK \in S$.
- Algebraic property guarantees quadratic invariance.
- Question: Does Quadratic Invariance imply existence of poset structure?
- In certain settings, yes.
- Key: Quadratic invariance can be interpreted as a transitivity property.
- Posets have lot more structure. Can we extract more out of it?

Drawbacks

- Control problems convex in Youla parameter.
- Main difficulty: Infinite dimensional problem.
- Approximation techniques, but 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:

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

- Poset causal: $A, B \in \mathcal{I}$.
- Find K^* which is stabilizing, optimal.

$$\begin{aligned} &\text{Min}_K \|P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\|^2 \\ &K \in \mathcal{I} \\ &K \text{ stabilizing.} \end{aligned}$$

- Key property we exploit: separability of the $\mathcal{H}_2$ norm.

$\mathcal{H}_2$ Optimal Control

- Recall Frobenius norm:

$$\|H\|_F^2 = \text{Trace}(H^T H).$$

- $\mathcal{H}_2$ norm is its extension to operators.
- Solution to optimal centralized problem standard.
- Based on algebraic Riccati equations:

$$X = C^T C + A^T XA - A^T XB(D^T D + B^T XB)^{-1} B^T XA$$
$$K = (D^T D + B^T XB)^{-1} B^T XA.$$

Decentralized Control Problem

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

$$\begin{aligned} \text{minimize}_K &\|P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\|^2 \\ &K \in \mathcal{I} \\ &K \text{ stabilizing.} \end{aligned}$$

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

$$\begin{aligned} \text{minimize}_Q &\|P_{11} + P_{12}Q\|^2 \\ &Q \in \mathcal{I} \\ &Q \text{ stabilizing.} \end{aligned}$$

$\mathcal{H}_2$ Decomposition Property

- Let $G = [G_1, \dots G_k]$.

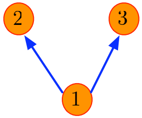
$$\|G\|^2 = \sum_{i=1}^{k} \|G_i\|^2.$$

- This separability property is the key feature we exploit.

Example

$$\begin{array}{ll} \min & \left\| P_{11} + P_{12} \begin{bmatrix} Q_{11} & 0 & 0 \\ Q_{21} & Q_{22} & 0 \\ Q_{31} & 0 & Q_{33} \end{bmatrix} \right\|^2 \\ \text{s.t.} & Q \text{ stabilizing.} \end{array}$$

$$\begin{array}{ll} \text{min.} & \left\| P_{11}(1) + P_{12} \begin{bmatrix} Q_{11} \\ Q_{21} \\ Q_{31} \end{bmatrix} \right\|^2 + \|P_{11}(2) + P_{12}(2) Q_{22}\|^2 \\ & + \|P_{11}(3) + P_{12}(3) Q_{33}\|^2 \\ \text{s.t.} & Q \text{ stabilizing.} \end{array}$$



$\mathcal{H}_2$ State Space Solution

This decomposition idea extends to all posets.

Theorem (S.-Parrilo)

Problem can be reduced to decoupled problems:

$$\begin{aligned} \text{minimize} \quad & \|P_{11}(j) + P_{12}(\uparrow j)Q^{\uparrow j}\|^2 \\ \text{subject to} \quad & Q^{\uparrow j} \text{ stabilizing} \\ & \text{for all } j \in P. \end{aligned}$$

- Optimal Q can be obtained by solving a set of decoupled centralized sub-problems.
- Each sub-problem requires solution of a Riccati equation.

$\mathcal{H}_2$ 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.

General Controller Architecture

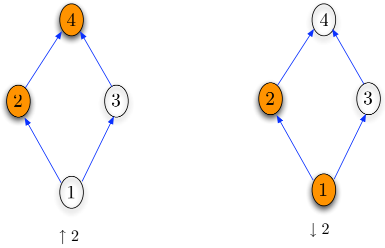
- What is the “right” architecture?
- Ingredients:
  1. Lower sets and upper sets
  2. Local variables (partial state predictions)
  3. Zeta function and Möbius function.
- Simple separation principle
- Optimality of architecture for $\mathcal{H}_2$.

General Controller Architecture

- What is the “right” architecture?
- Ingredients:
  1. Lower sets and upper sets
  2. Local variables (partial state predictions)
  3. Zeta function and Möbius function.
- Simple separation principle
- Optimality of architecture for $\mathcal{H}_2$.

Lower sets and upper sets

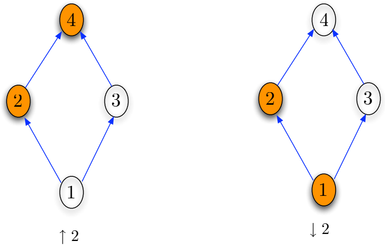
- Each "node" in $\mathcal{P}$ is a subsystem with state x_i and input u_i.
- Lower set: $\downarrow p = \{q \mid q \preceq p\}$.
- Corresponds to "downstream" known information.



- Upper set: $\uparrow p = \{q \mid p \preceq q\}$.
- Corresponds to "upstream" unknown information.
- u_i has access to x_j for $j \in \downarrow i$ (downstream).

Lower sets and upper sets

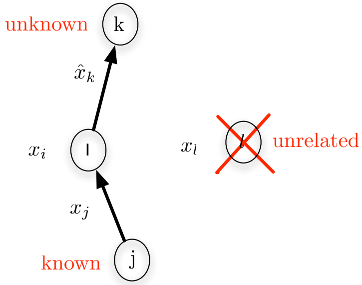
- Each "node" in $\mathcal{P}$ is a subsystem with state x_i and input u_i.
- Lower set: $\downarrow p = \{q \mid q \preceq p\}$.
- Corresponds to "downstream" known information.



- Upper set: $\uparrow p = \{q \mid p \preceq q\}$.
- Corresponds to "upstream" unknown information.
- u_i has access to x_j for $j \in \downarrow i$ (downstream).

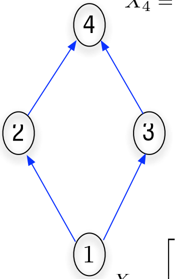
Local Variables

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



Local Variables

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

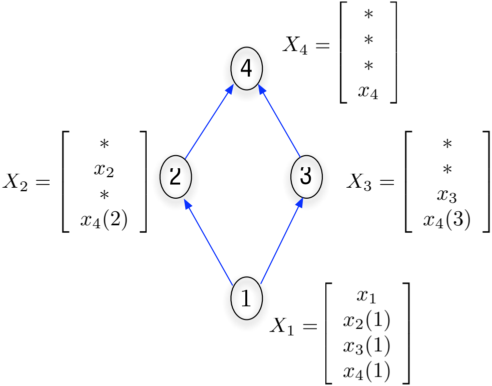


$$X_4 = \begin{bmatrix} * \\ * \\ * \\ x_4 \end{bmatrix} \quad X_2 = \begin{bmatrix} * \\ x_2 \\ * \\ x_4(2) \end{bmatrix} \quad X_3 = \begin{bmatrix} * \\ * \\ x_3 \\ x_4(3) \end{bmatrix} \quad X_1 = \begin{bmatrix} x_1 \\ x_2(1) \\ x_3(1) \\ x_4(1) \end{bmatrix}$$

- 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

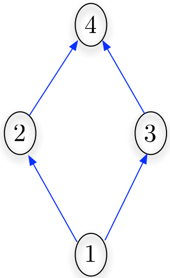
- Local variable $X_i : \uparrow i \to \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$



- 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: $G(i) : \uparrow i \times \uparrow i \to \mathbb{R}$. 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 $\mathbf{G} = \{G(1), \ldots, G(s)\}$.
- Local Product: $\mathbf{G} \circ X$ defined columnwise via:

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

- If $Y = \mathbf{G} \circ X$, then local variables (X_i, Y_i) decoupled.

Zeta and Möbius

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

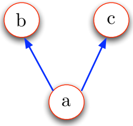
- The *Zeta* matrix is

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

- Its inverse is the *Möbius* matrix of the poset:

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

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



$$\zeta_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \qquad \mu_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Zeta and Möbius

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

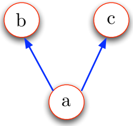
- The *Zeta* matrix is

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

- Its inverse is the *Möbius* matrix of the poset:

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

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



$$\zeta_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \qquad \mu_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Möbius inversion

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

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

These operations are obviously inverses of each other.

For our example:

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

Möbius inversion formula

$$g(y) = \sum_{x \preceq y} h(x) \qquad \Leftrightarrow \qquad h(y) = \sum_{x \preceq y} \mu(x, y) g(x)$$

Möbius inversion

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

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

These operations are obviously inverses of each other.

For our example:

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

Möbius inversion formula

$$g(y) = \sum_{x \preceq y} h(x) \qquad \Leftrightarrow \qquad h(y) = \sum_{x \preceq y} \mu(x,y) g(x)$$

Möbius inversion

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

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

These operations are obviously inverses of each other.

For our example:

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

Möbius inversion formula

$$g(y) = \sum_{x \preceq y} h(x) \qquad \Leftrightarrow \qquad h(y) = \sum_{x \preceq y} \mu(x,y) g(x)$$

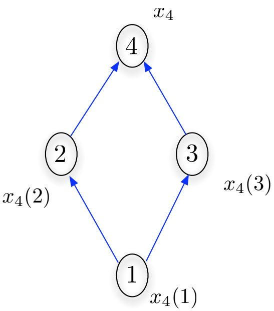
Möbius inversion: examples

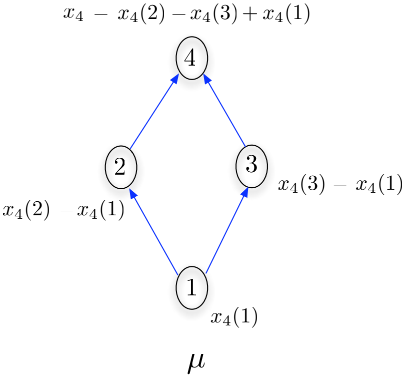
- If $\mathcal{P}$ is a chain: then ζ is "integration", $\mu := \zeta^{-1}$ is "differentiation".
- If $\mathcal{P}$ is the subset lattice, then μ is inclusion-exclusion
- If $\mathcal{P}$ is the divisibility integer lattice, then μ is the number-theoretic Möbius function.
- Many others: vector spaces,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

Möbius inversion for control

- μ and ζ operators on X.
- $\mu(X)$ innovations.
- ζ combines downstream information.
- Key insight: Möbius inversion respects the poset structure.
- No additional communication required to compute it.

Möbius operator





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

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 $j \preceq i$, where $X_{ii} = x_i$.
- Propose a control law:

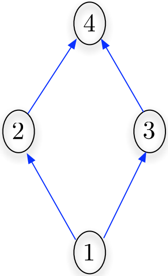
$$U = \zeta(\mathbf{G} \circ \mu(X)).$$

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

Controller Architecture: $U = \zeta(\mathbf{G} \circ \mu(X))$

- "Local innovations" computed by $\mu(X)$ (differentiation)
- Compute "local corrections"
- Aggregate them via $\zeta(\cdot)$ (integration)

$$\begin{bmatrix} * \\ u_2 \\ * \\ u_4(2) \end{bmatrix} = G(1) \begin{bmatrix} x_1 \\ x_2(1) \\ x_3(1) \\ x_4(1) \end{bmatrix} + G(2) \begin{bmatrix} * \\ x_2 - x_2(1) \\ * \\ x_4(2) - x_4(1) \end{bmatrix}$$



$$\begin{bmatrix} u_1 \\ u_2(1) \\ u_3(1) \\ u_4(1) \end{bmatrix} = G(1) \begin{bmatrix} x_1 \\ x_2(1) \\ x_3(1) \\ x_4(1) \end{bmatrix}$$

Closed-loop

Can compactly write closed-loop dynamics as matrix equations:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])) + Z_d[t].$$

- Each column corresponds to a different subsystem
- Equations have structure of $\mathcal{I}$, only need entries with $j \preceq i$
- Diagonal is the plant, off-diagonal is the controller
- Z_d downstream influence
- Since ζ and μ are local, so is the closed-loop

Separation Principle

- Closed-loop equations:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])) + Z_d[t].$$

- Apply μ, and use the fact that μ and ζ are inverses:

$$\begin{aligned} \mu(X)[t+1] &= A\mu(X)[t] + B(\mathbf{G} \circ \mu(X)[t]) \\ &= (\mathbf{A} + \mathbf{BG}) \circ \mu(X). \end{aligned}$$

where $(\mathbf{A} + \mathbf{BG})(i) = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i)$.

- "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:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])) + Z_d[t].$$

- Apply μ, and use the fact that μ and ζ are inverses:

$$\begin{aligned} \mu(X)[t+1] &= A\mu(X)[t] + B(\mathbf{G} \circ \mu(X)[t]) \\ &= (\mathbf{A} + \mathbf{BG}) \circ \mu(X). \end{aligned}$$

where $(\mathbf{A} + \mathbf{BG})(i) = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i)$.

- "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:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])) + Z_d[t].$$

- Apply μ, and use the fact that μ and ζ are inverses:

$$\begin{aligned} \mu(X)[t+1] &= A\mu(X)[t] + B(\mathbf{G} \circ \mu(X)[t]) \\ &= (\mathbf{A} + \mathbf{BG}) \circ \mu(X). \end{aligned}$$

  where $(\mathbf{A} + \mathbf{BG})(i) = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i)$.

- "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 (S.-Parrilo)

$\mathcal{H}_2$-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.

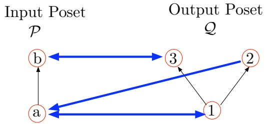
Interesting Directions

Möbius-inversion controller

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

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 μ
- Generalization of related concepts (Youla parameterization, "purified outputs", etc)?
- Extensions to output feedback, different plant/controller posets (Galois connections), . . .



Related Work

- 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.
- Poset framework introduced in S.-Parrilo (2008). Special case of QI, with richer and better understood algebraic structure.
- Swigart-Lall (2010) gave a state-space solution for the two-controller case, via a spectral factorization approach.
- S.-Parrilo (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).

Conclusions

- Posets/Incidence algebras: interesting objects in their own right!
- Provide general/useful framework for flow of information.
  1. Conceptually nice.
  2. Computationally tractable.
- Presented $\mathcal{H}_2$-optimal state-space solutions.
- Simple controller structure based on Möbius inversion.

Conclusions

- Posets/Incidence algebras: interesting objects in their own right!
- Provide general/useful framework for flow of information.
  1. Conceptually nice.
  2. Computationally tractable.
- Presented $\mathcal{H}_2$-optimal state-space solutions.
- Simple controller structure based on Möbius inversion.

Conclusions

- Posets/Incidence algebras: interesting objects in their own right!
- Provide general/useful framework for flow of information.
  1. Conceptually nice.
  2. Computationally tractable.
- Presented $\mathcal{H}_2$-optimal state-space solutions.
- Simple controller structure based on Möbius inversion.