Grand séminaire MACS 2025
Paris
October 15th

Localized Data-Driven Design of Positive Stabilizing Feedback Controllers for a Network System

Takumi Iwata (Hiroshima Univ.)

* Joint work with Shun-ichi Azuma and Masaaki Nagahara

Introduction of positive linear system

- Consider the system

$$x(t+1) = Ax(t) \qquad x(t) \in \mathbf{R}^n : \text{state} \qquad A \in \mathbf{R}^{n\times n} : \text{constant matrix}$$

> [Definition]
>
> The system $x(t+1) = Ax(t)$ is said to be *positive*
>
> if $x(t) \in \mathbf{R}^n_{\geq 0}$ $(t = 1, 2, \ldots)$ for any nonnegative initial state $x(0) \in \mathbf{R}^n_{\geq 0}$

$\mathbf{R}_{\geq 0}$: the set of nonnegative real numbers

- Examples of positive systems are
  - Reservoir networks
  - Transportation systems
  - Biological networks etc.

- On positive linear systems, the following lemma is well-known

> [Lemma] [Farina--Rinaldi, 2000]
>
> $x(t+1) = Ax(t)$ is positive $\iff$ A is a nonnegative matrix

Positive stabilization (model-based case)

- Consider the system

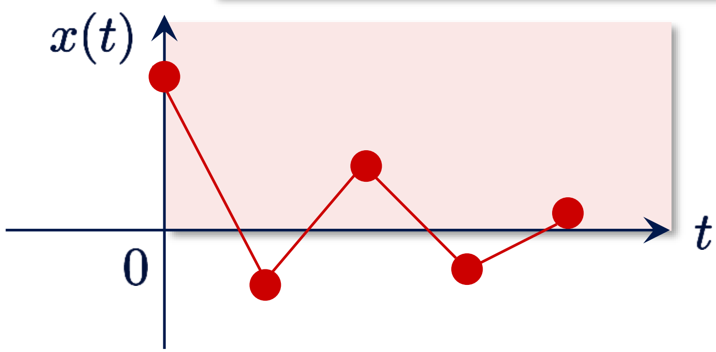
$$x(t+1) = Ax(t) + Bu(t) \qquad x(t) \in \mathbf{R}^n \text{ : state} \qquad u(t) \in \mathbf{R}^m \text{ : input}$$

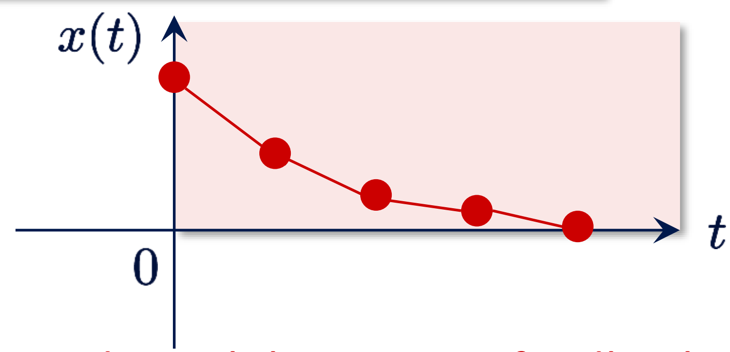
$$A \in \mathbf{R}^{n\times n},\quad B \in \mathbf{R}^{n\times m} \quad \text{: constant matrices}$$

  - $u(t) \in \mathbf{R}^m$ is determined by $u(t) = Fx(t)$
  - $F \in \mathbf{R}^{m\times n}$ is the state feedback gain to be designed

- Positive stabilization problem can be formulated as follows:

> [Problem]
>
> Find $\quad F \in \mathbf{R}^{m\times n}$
>
> s.t. $\quad x(t+1) = (A+BF)x(t)$ is positive and stable





Find a stabilizing state feedback gain preserving positivity

Positive stabilization (data-driven case)

- Consider the system and situation

$$x(t+1) = \hat{A}x(t) + \hat{B}u(t) \qquad x(t) \in \mathbf{R}^n \text{ : state} \qquad u(t) \in \mathbf{R}^m \text{ : input}$$

$$\hat{A} \in \mathbf{R}^{n\times n},\ \hat{B} \in \mathbf{R}^{n\times m} \qquad \text{: constant matrices}$$

  - Suppose that $\hat{A} \in \mathbf{R}^{n\times n}$ and $\hat{B} \in \mathbf{R}^{n\times m}$ are unknown
  - Instead, the following data matrices are known

$$U_- = \begin{bmatrix} u(0) & u(1) & \cdots & u(T-1) \end{bmatrix} \text{ and } X = \begin{bmatrix} x(0) & x(1) & \cdots & x(T) \end{bmatrix}$$

  - Note here that
    - U_- and X are pre-collected offline data obtained from the system
    - U_- is the input sequence applied to the system
    - X is the state trajectory corresponding to the input sequence U_-

[Problem]

Given Dataset, i.e., pair of data matrices $\mathcal{D} = (U_-, X)$

Find $F \in \mathbf{R}^{m\times n}$

s.t. $x(t+1) = (\hat{A} + \hat{B}F)x(t)$ is positive and stable

Related works

- Data-driven stabilization
  - Linear matrix inequality (LMI)-based approach [De Persis--Tesi, 2020]

    [van Waarde *et al.*, 2020]

- Data-driven positive stabilization
  - Linear programming (LP)-based approach [Miller *et al.*, 2023]

- In this talk
  - Positive stabilization is considered for a network system
  - Proposed approach is building on the projected consensus algorithm

    [Nedic *et al.*, 2010]

    [Takahashi--Kawashima, 2018]

    etc.

Problem formulation

System description

$\mathcal{N}_i$: Index set of neighboring of agent i

- Consider the network system with N agents

  - The dynamics of agent i is denoted by

$$x_i(t+1) = \widehat{A}_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} \widehat{A}_{ij}x_j(t) + \widehat{B}_{ii}u_i(t)$$

  - The local controller of agent i is given by

$$u_i(t) = F_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} F_{ij}x_j(t)$$

$F_{ij} \in \mathbf{R}^{m_i \times n_j}$ is the state feedback gain to be designed

  - The dynamics of entire network system is described by

$$x(t+1) = \widehat{A}x(t) + \widehat{B}u(t)$$

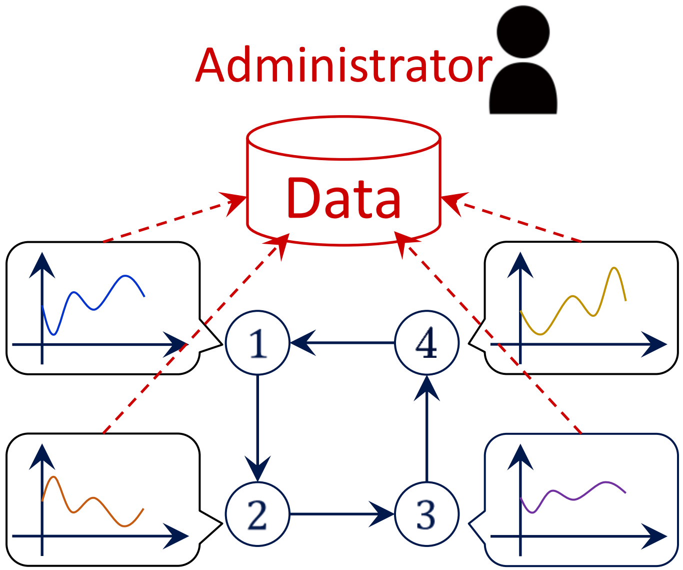
where

$$\widehat{A} = \begin{bmatrix} \widehat{A}_{11} & \widehat{A}_{12} & \cdots & \widehat{A}_{1N} \\ \widehat{A}_{21} & \widehat{A}_{22} & \cdots & \widehat{A}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{A}_{N1} & \widehat{A}_{N2} & \cdots & \widehat{A}_{NN} \end{bmatrix} \qquad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} \qquad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{bmatrix}$$

$$\widehat{B} = \mathrm{diag}(\widehat{B}_{11}, \widehat{B}_{22}, \dots, \widehat{B}_{NN})$$

Approaches to data-driven positive stabilization

Existing approach

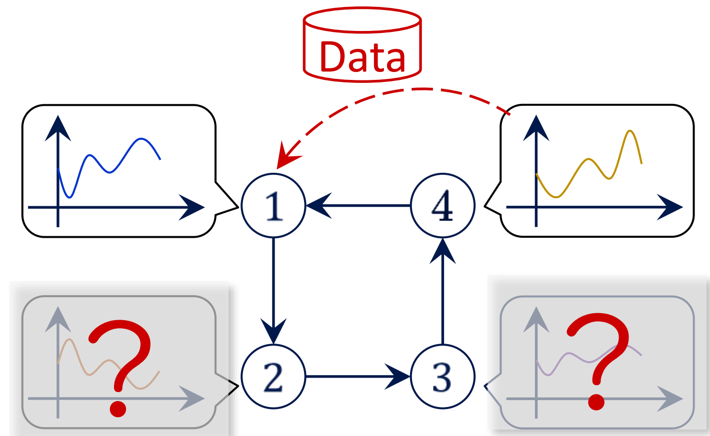


- ■ Administrator
  - collects data of all agents
  - designs all controllers

✓ All agents' data are required

✓ Not scalable

Proposed approach



- ■ Each agent
  - accesses data of neighboring agents
  - designs its own local controller

✓ Neighboring agents' data are required

✓ Scalable

Problem formulation

- Locally available dataset of agent i is denoted by $\mathcal{D}_i$
  - For the entire network system $x(t+1) = \widehat{A}x(t) + \widehat{B}u(t)$,
    $U_- = \begin{bmatrix} u(0) & u(1) & \cdots & u(T-1) \end{bmatrix}$ and $X = \begin{bmatrix} x(0) & x(1) & \cdots & x(T) \end{bmatrix}$
    are the offline time-series data matrices
  - Among of the above data, the agent i can use the following dataset:

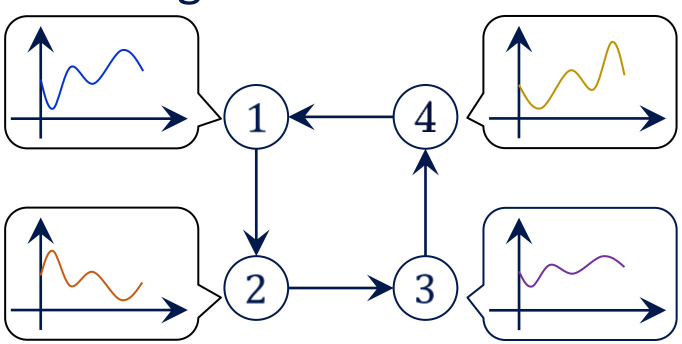
$$\mathcal{D}_i = (\underbrace{U_{i-}, X_i}_{\text{Data of agent } i}, \underbrace{X_{j_1}, X_{j_2}, \ldots, X_{j_c}}_{\text{Data of Neighboring agents for agent } i})$$

$$U_{i-} = \begin{bmatrix} u_i(0) & u_i(1) & \cdots & u_i(T-1) \end{bmatrix}$$

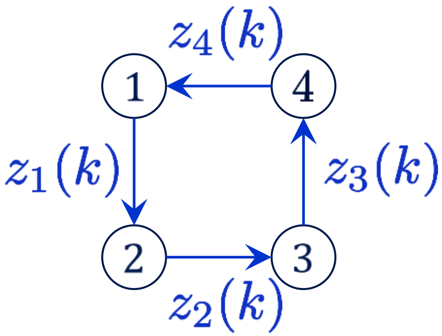
$$X_i = \begin{bmatrix} x_i(0) & x_i(1) & \cdots & x_i(T) \end{bmatrix}$$

$$\{j_1, j_2, \ldots, j_c\} = \mathcal{N}_i \text{ and } c = |\mathcal{N}_i|$$

ex) Agent 1 has access data of agents 1 and 4.



- Also, suppose that each agent can communicate over the network



- Note that
  - Each agent cannot send other agents' dataset
  - This is considered in case preserving privacy or security

- Our goal is to find state feedback gains in each local controller such that the closed-loop system $x(t+1) = (\widehat{A} + \widehat{B}F)x(t)$ is positive and stable
  - To this end, consider the update rule for the local controller of the form:

$$\begin{cases} z_i(k+1) = f_i(z_{\mathcal{N}_i}(k), \mathcal{D}_i, \mathcal{N}_i), \\ F_{\mathcal{N}_i}(k) = g_i(z_i(k)) \end{cases}$$

$$u_i(t) = \underline{F_{ii}} x_i(t) + \sum_{j \in \mathcal{N}_i} \underline{F_{ij}} x_j(t)$$

updated

k : number of updates

$F_{\mathcal{N}_i}(k)$: $(F_{ii}(k), F_{ij_1}(k), F_{ij_2}(k), \ldots, F_{ij_c}(k))$

$z_i(k)$: vector whose value is determined by $(z_{\mathcal{N}_i}(k), \mathcal{D}_i, \mathcal{N}_i)$

exchanged information to update $F_{\mathcal{N}_i}(k)$

$z_{\mathcal{N}_i}(k)$: $(z_i(k), z_{j_1}(k), z_{j_2}(k), \ldots, z_{j_c}(k))$ locally available information

- We now formulate the following problem

[Problem]

Given $\mathcal{D}_i$ and $\mathcal{N}_i$ for the agent $i \in \{1, 2, \ldots, N\}$

locally available dataset and index set of neighboring agents

Find $f_i(\cdot)$ and $g_i(\cdot)$ in the update rule

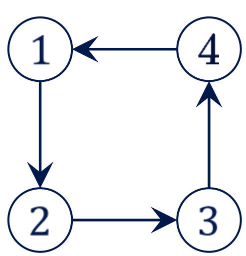
s.t. $x(t+1) = (\widehat{A} + \widehat{B}\bar{F})x(t)$ is positive and stable

- The objective of each agent is to find state feedback gains that make the entire system positive and stable.

- $\bar{F} \in \mathbf{R}^{m \times n}$ is a matrix such that its (i, j) block is

$\bar{F}_{ij} = \lim_{k \to \infty} F_{ij}(k)$ if $j \in \mathcal{N}_i \cup \{i\}$ and $\bar{F}_{ij} = 0$ otherwise

ex)



$$\bar{F} = \begin{bmatrix} \bar{F}_{11} & 0 & 0 & \bar{F}_{14} \\ \bar{F}_{21} & \bar{F}_{22} & 0 & 0 \\ 0 & \bar{F}_{32} & \bar{F}_{33} & 0 \\ 0 & 0 & \bar{F}_{43} & \bar{F}_{44} \end{bmatrix}$$

updated by agent 1

Key ideas for solution

Key idea (i): model-based positive stabilization

- We now introduce the lemma on positive stabilization in the model-based case
  - Consider the closed-loop system $x(t+1) = (A+BF)x(t)$
  - $F \in \mathbf{R}^{m\times n}$ is the state feedback gain to be designed
  - The following lemma can be used for positive stabilization

[Lemma] [Rami--Tadeo--Benzaouia, 2007, Theorem 3.1]

There exists $F \in \mathbf{R}^{m\times n}$ such that $x(t+1) = (A+BF)x(t)$ is positive and stable

if and only if there exist $Q \in \mathbf{PD}_n, Y \in \mathbf{R}^{m\times n}$, and $\mu \in \mathbf{R}_{>0}$ such that

$$(AQ+BY)\mathbf{1}_n \leq (Q-\mu I_n)\mathbf{1}_n \text{ and } AQ+BY \geq 0$$

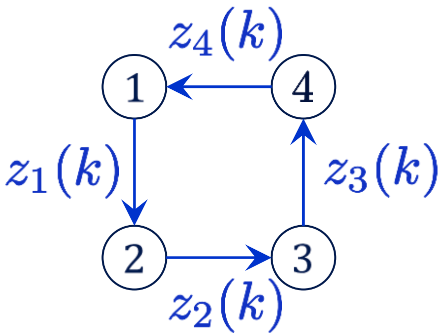
Moreover, $F = YQ^{-1}$ is a state feedback gain such that the closed-loop system $x(t+1) = (A+BF)x(t)$ is positive and stable

$\mathbf{R}_{>0}$: the set of positive real numbers

$\mathbf{PD}_n$: the set of $n \times n$ diagonal and positive definite matrices

✓ Positive stabilization can be reduced an LP Problem

- In our problem, each agent can use
  - $\mathcal{D}_i$ and $\mathcal{N}_i$ Locally available data and index set of neighboring agents
  - Information obtained from agent $j \in \mathcal{N}_i$



- Recall that
  - Each agent cannot send other agents' dataset
  - This is considered in a case preserving privacy or security

- If there exists an update rule makes agents (over the network)

find	$Q \in \mathbf{PD}_n, Y \in \mathbf{R}^{m\times n},$ and $\mu \in \mathbf{R}_{>0}$
s.t.	$(\widehat{A}Q + \widehat{B}Y)\mathbf{1}_n \leq (Q - \mu I_n)\mathbf{1}_n$ and $\widehat{A}Q + \widehat{B}Y \geq 0$

using the limited data and info, then the update rule is a solution to our problem

- To develop this rule, we consider the use of the projected consensus algorithm

[Nedic *et al.*, 2010]   [Takahashi--Kawashima, 2018]

Key idea (ii): projected consensus algorithm

- We here introduce the projected consensus algorithm
  - This is a distributed algorithm to solve the following problem

[Problem]

Given $\mathcal{X}_i \subseteq \mathbf{R}^d$ and $z_{i0} \in \mathcal{X}_i$ for the agent $i \in \{1, 2, \ldots, N\}$

closed and convex subset and its element

Find $z \in \mathbf{R}^d$

s.t. $z \in \bigcap_{i=1}^{N} \mathcal{X}_i$

- Suppose that each agent is connected over the network
  - Index set of neighboring of agent i is denoted by $\mathcal{N}_i$
- Here, the objective is to make agents find an element belonging to the intersection via information exchange over the network

- To solve this distributed problem, consider the following update rule

$$z_i(k+1) = \mathrm{proj}_{\mathcal{X}_i}\left(\sum_{j \in \mathcal{N}_i} p_{ij} z_j(k)\right), \qquad z_i(0) = z_{i0} \in \mathcal{X}_i$$

z_i : state of agent i

$\mathrm{proj}_{\mathcal{X}_i}(\cdot)$: projection onto $\mathcal{X}_i$

p_{ij} : nonnegative weight (determined by the network structure)

- If there exists $z \in \mathbf{R}^d$ such that $\underbrace{\lim_{k \to \infty} \|z_i(k) - z\| = 0 \quad \forall i \in \{1, 2, \ldots, N\}}_{\text{achieve consensus}}$

  then $z \in \bigcap_{i=1}^{N} \mathcal{X}_i$

  - Using this update rule, we can solve the problem under some condition
  - Indeed, a sufficient condition to be solvable has been investigated in:

[Takahashi--Kawashima, 2018]

- To solve our problem, the projected consensus algorithm can be utilized

Solution

- We now give a solution to our problem. To this end,
  - Let μ be a sufficiently small positive scalar
  - Given $\mathcal{D}_i$, $\mathcal{N}_i$, and μ, we define the following set:

$$\mathcal{X}_i(\mathcal{D}_i, \mathcal{N}_i, \mu) = \Bigg\{ z \in \mathbf{R}^{n+mn} \,\Bigg|\, \exists (M, Q, Y) \in \mathbf{R}^{q' \times p'}_{\geq 0} \times \mathbf{PD}_n \times \mathbf{R}^{m \times n} \text{ such that } Y_{ij} = 0 \ (\forall j \notin \mathcal{N}_i \cup \{i\}), \ MG_i = H_i, \ M\gamma_i \leq \eta_i, \ Q\mathbf{1}_n \geq \mu \mathbf{1}_n, \text{ and } z = \begin{bmatrix} Q\mathbf{1}_n \\ \mathrm{vec}(Y) \end{bmatrix} \Bigg\}$$

  - This set can be provided by referring to

[Rami--Tadeo--Benzaouia, 2007, Theorem 3.1] As introduced in key idea (i)

[Henrion--Tarbouriech--Kucera, 1999 Lemma 2] Extended Farkas Lemma (or Haar's Lemma)

  - For $G_i, \gamma_i, H_i, \text{and}, \eta_i$, see the reference that I will mention later
  - If there exists an element of $\bigcap_{i=1}^{N} \mathcal{X}_i(\mathcal{D}_i, \mathcal{N}_i, \mu)$, then we can

find	$Q \in \mathbf{PD}_n, Y \in \mathbf{R}^{m \times n}$, and $\mu \in \mathbf{R}_{>0}$
s.t.	$(\widehat{A}Q + \widehat{B}Y)\mathbf{1}_n \leq (Q - \mu I_n)\mathbf{1}_n$ and $\widehat{A}Q + \widehat{B}Y \geq 0$

Main result

- We give the following theorem

[Theorem]
Consider the update rule

$$\begin{cases} z_i(k+1) = \mathrm{proj}_{\mathcal{X}_i(\mathcal{D}_i,\mathcal{N}_i,\mu)}\left(\sum_{j\in\mathcal{N}_i\cup\{i\}} p_{ij} z_j(k)\right), \; z_i(0) = z_{i0} \in \mathcal{X}_i(\mathcal{D}_i,\mathcal{N}_i,\mu), \\ F_{\mathcal{N}_i}(k) = \Big(Y_{ii}^{(i)}(k)(Q_i^{(i)}(k))^{-1}, Y_{ij_1}^{(i)}(k)(Q_{j_1}^{(i)}(k))^{-1}, \\ \qquad Y_{ij_2}^{(i)}(k)(Q_{j_2}^{(i)}(k))^{-1}, \ldots, Y_{ij_c}^{(i)}(k)(Q_{j_c}^{(i)}(k))^{-1}\Big) \end{cases}$$

- update $z_i(k)$ based on $z_j(k)$ $(j \in \mathcal{N}_i \cup \{i\})$
- search for a common (Q, Y)

- gains of agent i
- determined by $z_i(k)$

If there exists $\alpha \in \mathbf{R}^{n+mn}$ such that

$$\lim_{k\to\infty} \|z_i(k) - \alpha\| = 0 \quad (\forall i \in \{1,2,\ldots,N\}),$$

then the above update rule is a solution to the problem

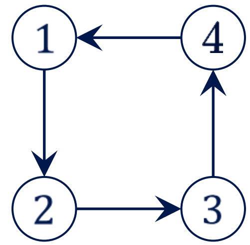
- For $Q \in \mathbf{PD}_n$ and $Y \in \mathbf{R}^{m\times n}$ satisfying $\alpha = \begin{bmatrix} Q\mathbf{1}_n \\ \mathrm{vec}(Y) \end{bmatrix}$,
  $\bar{F} = YQ^{-1}$ is a positive stabilizing gain for $x(t+1) = (\widehat{A} + \widehat{B}\bar{F})x(t)$

- Stop criteria for the update rule has not been given (this is our future work)

Numerical example

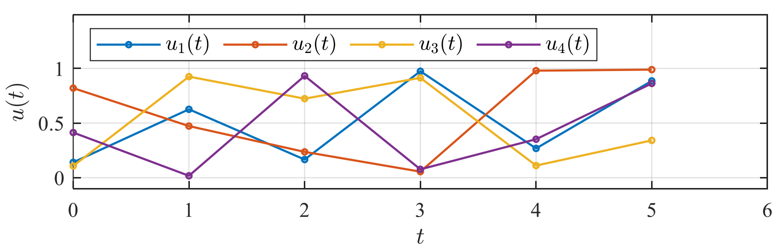
Numerical example

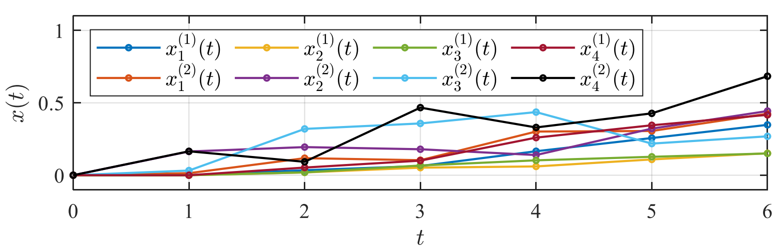
- System



$$\begin{bmatrix} x_1(t+1) \\ \hline x_2(t+1) \\ \hline x_3(t+1) \\ \hline x_4(t+1) \end{bmatrix} = \left[\begin{array}{cc|cc|cc|cc} 0.5 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.1 \\ 0.2 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.2 \\ \hline 0.1 & 0.2 & 0.3 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.2 & 0.6 & 0.0 & 0.0 & 0.0 & 0.0 \\ \hline 0.0 & 0.0 & 0.3 & 0.1 & 0.5 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.2 & 0.2 & 0.4 & 0.0 & 0.0 \\ \hline 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.1 & 0.7 & 0.3 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.3 & 0.5 \end{array}\right] \begin{bmatrix} x_1(t) \\ \hline x_2(t) \\ \hline x_3(t) \\ \hline x_4(t) \end{bmatrix} + \left[\begin{array}{c|c|c|c} 0.0 & & & \\ 0.1 & & & \\ \hline & 0.0 & & \\ & 0.2 & & \\ \hline & & 0.0 & \\ & & 0.3 & \\ \hline & & & 0.0 \\ & & & 0.4 \end{array}\right] \begin{bmatrix} u_1(t) \\ \hline u_2(t) \\ \hline u_3(t) \\ \hline u_4(t) \end{bmatrix}$$

- Dataset (of the entire system)





Each agent

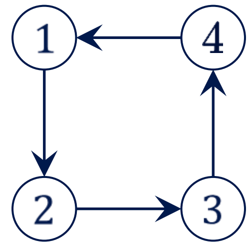
- can access the data of its neighboring agent(s)
- updates its local controller

$$u_i(t) = \underline{F_{ii}} x_i(t) + \sum_{j \in \mathcal{N}_i} \underline{F_{ij}} x_j(t)$$

updated

Numerical example

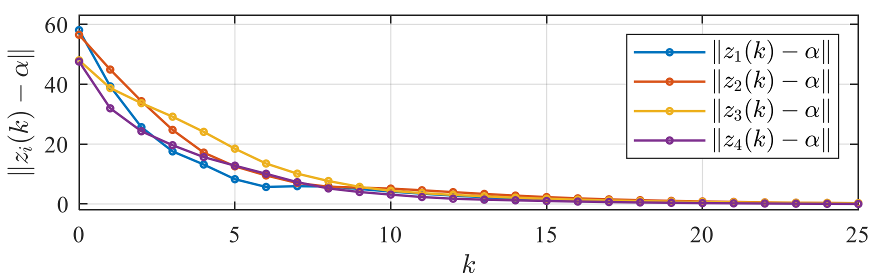
- System



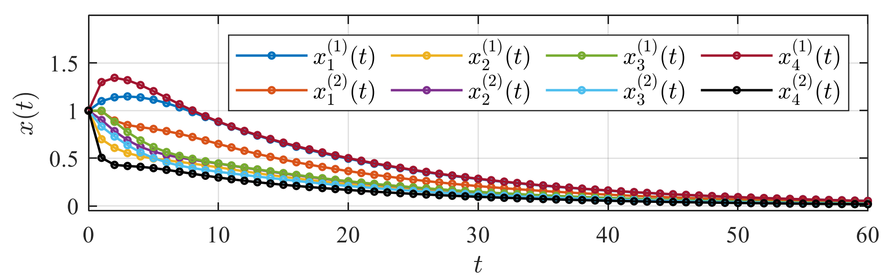
$$
\begin{bmatrix} x_1(t+1) \\ \hline x_2(t+1) \\ \hline x_3(t+1) \\ \hline x_4(t+1) \end{bmatrix}
=
\left[\begin{array}{cc|cc|cc|cc}
0.5 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.1 \\
0.2 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.2 \\ \hline
0.1 & 0.2 & 0.3 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.1 & 0.2 & 0.6 & 0.0 & 0.0 & 0.0 & 0.0 \\ \hline
0.0 & 0.0 & 0.3 & 0.1 & 0.5 & 0.1 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.1 & 0.2 & 0.2 & 0.4 & 0.0 & 0.0 \\ \hline
0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.1 & 0.7 & 0.3 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.3 & 0.5
\end{array}\right]
\begin{bmatrix} x_1(t) \\ \hline x_2(t) \\ \hline x_3(t) \\ \hline x_4(t) \end{bmatrix}
+
\left[\begin{array}{c|c|c|c}
0.0 & & & \\
0.1 & & & \\ \hline
 & 0.0 & & \\
 & 0.2 & & \\ \hline
 & & 0.0 & \\
 & & 0.3 & \\ \hline
 & & & 0.0 \\
 & & & 0.4
\end{array}\right]
\begin{bmatrix} u_1(t) \\ \hline u_2(t) \\ \hline u_3(t) \\ \hline u_4(t) \end{bmatrix}
$$

- Proposed update rule
- State trajectory

consensus



positivity and stability



$$
\bar{F} = \left[\begin{array}{cc|cc|cc|cc}
0.0973 & -0.1179 & 0 & 0 & 0 & 0 & -0.0072 & -0.0197 \\ \hline
0.0791 & 0.0717 & 0.0185 & -0.1624 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0.0492 & 0.0864 & 0.0106 & -0.0343 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0.0000 & -0.2500 & -0.1983 & -0.5392
\end{array}\right]
$$

Conclusion

- In this talk
  - Data-driven positive stabilization was considered for a network system
  - This was provided building on the projected consensus algorithm
- As an extension
  - Our approach can be used for noisy data case
- Future work
  - Development of stop criteria for the update rule

Acknowledgements

- This is a joint work with

Prof. Shun-ichi Azuma
(Kyoto University)

Prof. Masaaki Nagahara
(Hiroshima University)

- Thank you very much for hosting my stay in Toulouse

Prof. Dimitri Peaucelle
(LAAS-CNRS)

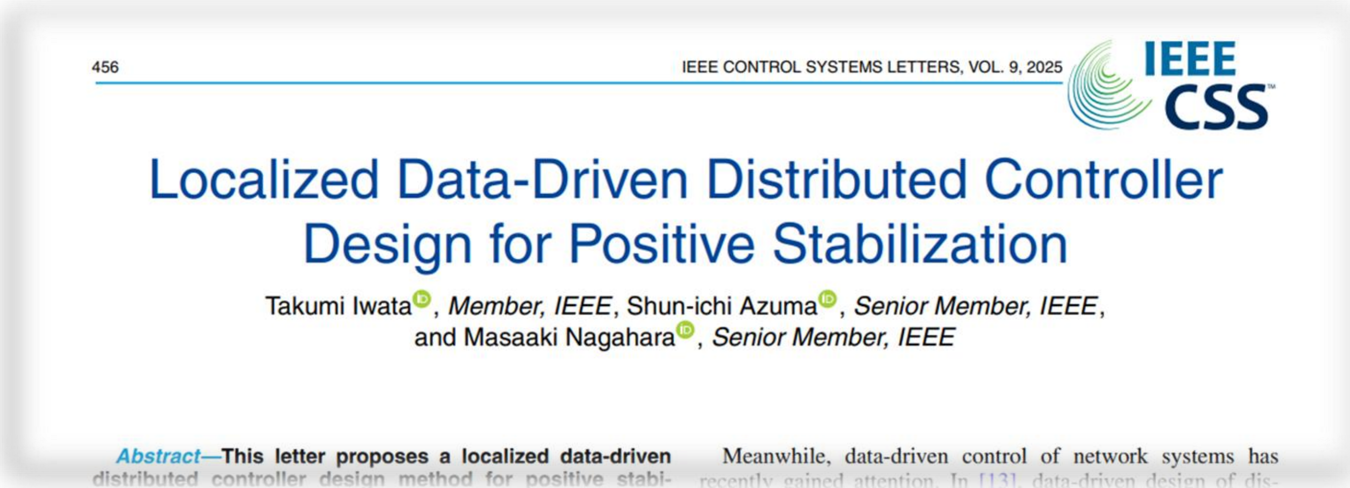
References

- This talk's results are based on

TI, S. Azuma, and M. Nagahara, "Localized Data-Driven Distributed Controller Design for Positive Stabilization," in *Proceedings of the 64th IEEE Conference on Decision and Control*, Rio de Janeiro, Brazil, 2025, to appear.

or

IEEE Control Systems Letters, vol. 9, pp. 456--461, 2025.


456 IEEE CONTROL SYSTEMS LETTERS, VOL. 9, 2025

Localized Data-Driven Distributed Controller Design for Positive Stabilization

Takumi Iwata, *Member, IEEE*, Shun-ichi Azuma, *Senior Member, IEEE*, and Masaaki Nagahara, *Senior Member, IEEE*

Abstract—**This letter proposes a localized data-driven distributed controller design method for positive stabi-**

Meanwhile, data-driven control of network systems has recently gained attention. In [13], data-driven design of dis-

[This is an open access article.]

Thank you very much.

Merci beaucoup.

Appendix

- Definitions

$$X_{i-} := \begin{bmatrix} x_i(0) & x_i(1) & \cdots & x_i(T-1) \end{bmatrix}$$

$$X_{i+} := \begin{bmatrix} x_i(1) & x_i(2) & \cdots & x_i(T) \end{bmatrix}$$

$$\mathcal{N}_i^{\mathrm{a}} := \mathcal{N}_i \cup \{i\}$$

$$\varPhi_i := \mathrm{diag}(\phi_1, \phi_2, \ldots, \phi_N) \in \mathbf{R}^{n \times n} \text{ with } \phi_j := \begin{cases} I_{n_j} & \text{if } j = i, \\ 0_{n_j \times n_j} & \text{otherwise.} \end{cases}$$

$$\varPsi_i := \mathrm{diag}(\psi_1, \psi_2, \ldots, \psi_N) \in \mathbf{R}^{m \times m} \text{ with } \psi_j := \begin{cases} I_{m_j} & \text{if } j = i, \\ 0_{m_j \times m_j} & \text{otherwise.} \end{cases}$$

$$G_i := \begin{bmatrix} \begin{bmatrix} X_-^{\mathrm{T}} \sum_{j \in \mathcal{N}_i^{\mathrm{a}}} \varPhi_j & U_-^{\mathrm{T}} \varPsi_i \end{bmatrix} \otimes \varPhi_i \\ -\begin{bmatrix} X_-^{\mathrm{T}} \sum_{j \in \mathcal{N}_i^{\mathrm{a}}} \varPhi_j & U_-^{\mathrm{T}} \varPsi_i \end{bmatrix} \otimes \varPhi_i \\ \begin{bmatrix} \sum_{j \notin \mathcal{N}_i^{\mathrm{a}}} \varPhi_j & 0 \end{bmatrix} \otimes \varPhi_i \\ -\begin{bmatrix} \sum_{j \notin \mathcal{N}_i^{\mathrm{a}}} \varPhi_j & 0 \end{bmatrix} \otimes \varPhi_i \\ \begin{bmatrix} 0 & \sum_{j \neq i} \varPsi_j \end{bmatrix} \otimes \varPhi_i \\ -\begin{bmatrix} 0 & \sum_{j \neq i} \varPsi_j \end{bmatrix} \otimes \varPhi_i \end{bmatrix}, \gamma_i := \begin{bmatrix} \mathrm{vec}(\varPhi_i X_+) \\ -\mathrm{vec}(\varPhi_i X_+) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H_i := \begin{bmatrix} \left(\mathbf{1}_n^{\mathrm{T}} \begin{bmatrix} Q & Y^{\mathrm{T}} \end{bmatrix}\right) \otimes \varPhi_i \\ -\begin{bmatrix} Q & Y^{\mathrm{T}} \end{bmatrix} \otimes \varPhi_i \end{bmatrix}, \eta_i := \begin{bmatrix} \varPhi_i (Q - \mu I_n) \mathbf{1}_n \\ 0 \end{bmatrix}.$$