An Emulation Approach to Output-Feedback Sampled-Data Synchronization

Gal Barkai   Leonid Mirkin   Daniel Zelazo

Technion - Israel Institute of Technology

22nd European Control Conference

Outline

Problem formulation

Agents:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad i \in [1, \ldots, \nu]$$
$$y_i(t) = Cx_i(t)$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{m\times n}$.

Goal: Given $A_0 \in \mathbb{R}^{n\times n}$ with $\mathrm{spec}(A_0) \cap \mathbb{C}_0 = \varnothing$ and semi-simple imaginary eigenvalues, design control signals $u_i(t)$ to synchronize the states $x_i(t)$ with some trajectory generated by $\dot{r}(t) = A_0 r(t)$.

Assumptions:

- $\mathcal{A}_1$: (C, A, B) is stabilizable and detectable,
- $\mathcal{A}_2$: there is $\bar{F}$ such that $A_0 = A + B\bar{F}$,

Problem formulation

Goal: Given $A_0 \in \mathbb{R}^{n\times n}$ with $\mathrm{spec}(A_0) \cap \mathbb{C}_0 = \varnothing$ and semi-simple imaginary eigenvalues, design control signals $u_i(t)$ to synchronize the states $x_i(t)$ with some trajectory generated by $\dot{r}(t) = A_0 r(t)$.

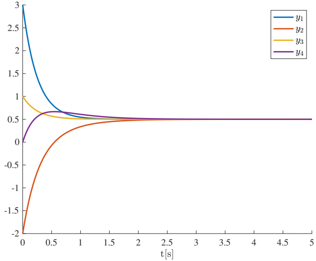
This encompasses two classic problems.

Problem formulation

Goal: Given $A_0 \in \mathbb{R}^{n\times n}$ with $\mathrm{spec}(A_0) \cap \mathbb{C}_0 = \emptyset$ and semi-simple imaginary eigenvalues, design control signals $u_i(t)$ to synchronize the states $x_i(t)$ with some trajectory generated by $\dot{r}(t) = A_0 r(t)$.

This encompasses two classic problems.

- When $A_0 = A = 0$ this is the consensus problem.

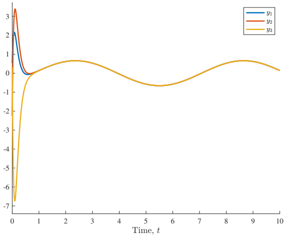


Problem formulation

Goal: Given $A_0 \in \mathbb{R}^{n \times n}$ with $\operatorname{spec}(A_0) \cap \mathbb{C}_0 = \varnothing$ and semi-simple imaginary eigenvalues, design control signals $u_i(t)$ to synchronize the states $x_i(t)$ with some trajectory generated by $\dot{r}(t) = A_0 r(t)$.

This encompasses two classic problems.

- When $A_0 = A = 0$ this is the consensus problem.
- When $A_0 = A$ this is the synchronization problem.



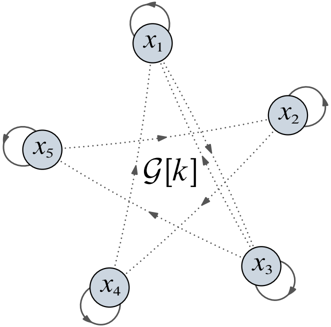
Information constraints

Local information: $y_i(t)$ and $u_i(t)$ are continuously available

Spatial constraints: agents communicate within their neighborhoods, $\mathcal{N}_i(t)$.

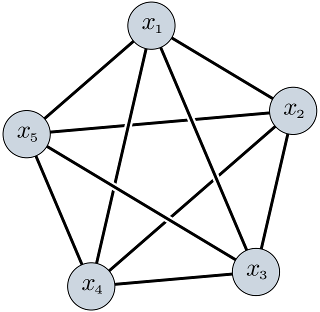
Temporal constraints: communicate only at discrete sampling instances, $t \in \{s_k\}$.

- Aperiodic $s_{k+1} - s_k \neq s_{k+l} - s_{k+l+1}$
- Asynchronous $\mathcal{N}_i(s_k) \neq \emptyset \not\Longrightarrow \mathcal{N}_j(s_k) \neq \emptyset$.



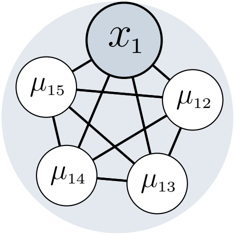
Emulation approach

1. Find a "good" controller, an "ideal" agreeing world.



Emulation approach

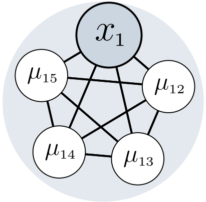
1. Find a "good" controller, an "ideal" agreeing world.
2. Emulate the "ideal" world, locally between samples.



$$\mu_i(t) = \left[\ \mu'_{i,1} \ \ \cdots \ \ \mu'_{i\nu}(t) \ \right]'$$
$$s_k < t \leq s_{k+1}$$

Emulation approach

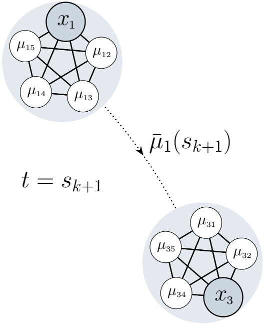
1. Find a "good" controller, an "ideal" agreeing world.
2. Emulate the "ideal" world, locally between samples.
3. Generate the control, using local local information.



$$\mu_i(t) = \left[\ \mu'_{i,1} \ \ \cdots \ \ \mu'_{i\nu}(t)\ \right]'$$
$$u_i(t) = \kappa\big(y_i(t), \mu_i(t)\big)$$
$$s_k < t \leq s_{k+1}$$

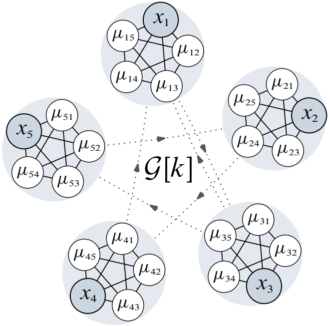
Emulation approach

1. Find a "good" controller, an "ideal" agreeing world.
2. Emulate the "ideal" world, locally between samples.
3. Generate the control, using local local information.
4. Update the emulation at sampling instances.



Emulation approach

1. Find a "good" controller, an "ideal" agreeing world.
2. Emulate the "ideal" world, locally between samples.
3. Generate the control, using local local information.
4. Update the emulation at sampling instances.



Each agent operates in its own "perfect world", which interact intermittently.

Outline

Problem formulation - full state feedback

Agents:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad i \in [1, \ldots, \nu]$$
$$y_i(t) = x_i(t)$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{m \times n}$.

Assumptions:

- $\mathcal{A}_1$: (C, A, B) is stabilizable and detectable,
- $\mathcal{A}_2$: there is $\bar{F}$ such that $A_0 = A + B\bar{F}$,

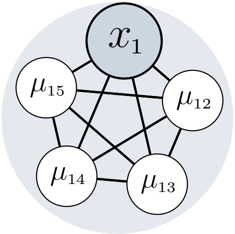
Goal: Given $A_0 \in \mathbb{R}^{n \times n}$ with $\operatorname{spec}(A_0) \cap \mathbb{C}_0 = \varnothing$ and semi-simple imaginary eigenvalues, design control signals $u_i(t)$ to synchronize the states $x_i(t)$ with some trajectory generated by $\dot{r}(t) = A_0 r(t)$.

Emulator with full-state feedback (Barkai, Mirkin, and Zelazo, 2023)

- Each agent locally emulates the *entire* ensemble under the ideal control law,

$$
\begin{cases}
\dot{\mu}_i(t) = f(\mu_i(t), x_i(t)) \\
u_i(t) = F_d \mu_{ii}(t) + (\bar{F} - F_\mathrm{d})\bar{\mu}_i(t) \\
\mu_{ii}(t) \equiv x_i(t), \quad \bar{\mu}_i(t) = (1/\nu)\sum_{j=1}^{\nu} \mu_{ij}(t)
\end{cases}
$$

where $\mu_i(t)$ is the ith agent's emulation of the entire group.



$$\mu_i(t) = \begin{bmatrix} \mu'_{i,1} & \cdots & \mu'_{i\nu}(t) \end{bmatrix}'$$

$$s_k < t \le s_{k+1}$$

Flow dynamics

Emulator with full-state feedback (Barkai, Mirkin, and Zelazo, 2023)

- Each agent locally emulates the *entire* ensemble under the ideal control law,

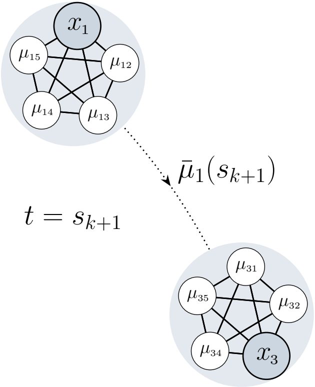
$$\begin{cases} \dot{\mu}_i(t) = f(\mu_i(t), x_i(t)) \\ u_i(t) = F_d \mu_{ii}(t) + (\bar{F} - F_{\rm d})\bar{\mu}_i(t) \\ \mu_{ii}(t) \equiv x_i(t), \quad \bar{\mu}_i(t) = (1/\nu)\sum_{j=1}^{\nu} \mu_{ij}(t) \end{cases}$$

where $\mu_i(t)$ is the ith agent's emulation of the entire group.

- The agents exchange their emulated centroids, and update their emulators according to

$$\mu_i(s_k^+) = \mu_i(s_k) - (\alpha_i \otimes I) \sum_{j \in \mathcal{N}_i[k]} \big(\bar{\mu}_i(s_k) - \bar{\mu}_j(s_k)\big),$$

for any $\alpha_i \in \mathbb{R}^\nu$ such that $\mathbb{1}'\alpha_i = 1$ and $e_i'\alpha_i = 0$.



Jump dynamics

State-feedback controller structure (Barkai, Mirkin, and Zelazo, 2023)

Define

$$e(t) = x(t) - \bar{\mu}(t), \quad \mu_{\delta,i}(t) = \mu_i(t) - (\mathbb{1} \otimes I_n)\bar{\mu}_i(t)$$

the aggregate closed loop is given by

$$\begin{cases} \begin{bmatrix} \dot{e}(t) \\ \dot{\mu}_\delta(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \begin{bmatrix} I \otimes A_{\mathsf{d}} & 0 & 0 \\ 0 & I_{\nu^2} \otimes A_{\mathsf{d}} & 0 \\ 0 & 0 & I_\nu \otimes A_0 \end{bmatrix} \begin{bmatrix} e(t) \\ \mu_\delta(t) \\ \bar{\mu}(t) \end{bmatrix} \\ \begin{bmatrix} e(s_k^+) \\ \mu_\delta(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I \\ 0 & I & -B_{\mathsf{jmp}}\mathcal{L}[k] \otimes I \\ 0 & 0 & (I - (1/\nu)\mathcal{L}[k]) \otimes I \end{bmatrix} \begin{bmatrix} e(s_k) \\ \mu_\delta(s_k) \\ \bar{\mu}(s_k) \end{bmatrix} \end{cases}, \quad A_{\mathsf{d}} = A + BF_{\mathsf{d}}$$

where $\mathcal{L}[k]$ is the Laplacian matrix associated with the network connectivity graph $\mathcal{G}[k]$ at s_k.

State-feedback controller structure (Barkai, Mirkin, and Zelazo, 2023)

Define

$$e(t) = x(t) - \bar{\mu}(t), \quad \mu_{\delta,i}(t) = \mu_i(t) - (\mathbb{1} \otimes I_n)\bar{\mu}_i(t)$$

the aggregate closed loop is given by

$$\begin{cases} \begin{bmatrix} \dot{e}(t) \\ \dot{\mu}_\delta(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \begin{bmatrix} I \otimes A_\mathsf{d} & 0 & 0 \\ 0 & I_{\nu^2} \otimes A_\mathsf{d} & 0 \\ 0 & 0 & I_\nu \otimes A_0 \end{bmatrix} \begin{bmatrix} e(t) \\ \mu_\delta(t) \\ \bar{\mu}(t) \end{bmatrix} \\ \begin{bmatrix} e(s_k^+) \\ \mu_\delta(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I \\ 0 & I & -B_\mathsf{jmp}\mathcal{L}[k] \otimes I \\ 0 & 0 & (I - (1/\nu)\mathcal{L}[k]) \otimes I \end{bmatrix} \begin{bmatrix} e(s_k) \\ \mu_\delta(s_k) \\ \bar{\mu}(s_k) \end{bmatrix} \end{cases}, \quad A_\mathsf{d} = A + BF_\mathsf{d}$$

where $\mathcal{L}[k]$ is the Laplacian matrix associated with the network connectivity graph $\mathcal{G}[k]$ at s_k.

- The flows are decoupled and A_d is Hurwitz.

State-feedback controller structure (Barkai, Mirkin, and Zelazo, 2023)

Define

$$e(t) = x(t) - \bar{\mu}(t), \quad \mu_{\delta,i}(t) = \mu_i(t) - (\mathbb{1} \otimes I_n)\bar{\mu}_i(t)$$

the aggregate closed loop is given by

$$\begin{cases} \begin{bmatrix} \dot{e}(t) \\ \dot{\mu}_\delta(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \begin{bmatrix} I \otimes A_\mathsf{d} & 0 & 0 \\ 0 & I_{\nu^2} \otimes A_\mathsf{d} & 0 \\ 0 & 0 & I_\nu \otimes A_0 \end{bmatrix} \begin{bmatrix} e(t) \\ \mu_\delta(t) \\ \bar{\mu}(t) \end{bmatrix} \\ \begin{bmatrix} e(s_k^+) \\ \mu_\delta(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I \\ 0 & I & -B_\mathsf{jmp}\mathcal{L}[k] \otimes I \\ 0 & 0 & (I - (1/\nu)\mathcal{L}[k]) \otimes I \end{bmatrix} \begin{bmatrix} e(s_k) \\ \mu_\delta(s_k) \\ \bar{\mu}(s_k) \end{bmatrix} \end{cases}, \quad A_\mathsf{d} = A + BF_\mathsf{d}$$

where $\mathcal{L}[k]$ is the Laplacian matrix associated with the network connectivity graph $\mathcal{G}[k]$ at s_k.

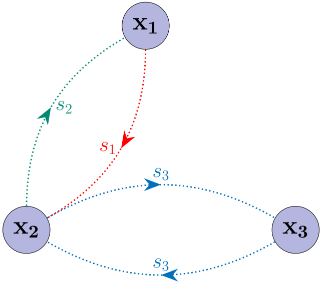
- The flows are decoupled and A_d is Hurwitz.
- Since $\mathcal{L}[k]\mathbb{1} = 0$, if $\bar{\mu}_i$ synchronize then so would x_i and $\mu_\delta \to 0$.

Assumption: persistency of eventual connectivity

- $\mathcal{A}_3$: there is a strictly increasing sub-sequence of sampling indices $\{k_p\}$ such that for all $p \in \mathbb{Z}_+$
  1. the intervals $s_{k_{p+1}} - s_{k_p}$ are uniformly bounded;
  2. $\bigcup_{k=k_p+1}^{k_{p+1}} \mathcal{G}[k]$ contains a directed rooted tree.

$$\bigcup_{k=k_0+1}^{k_1} \mathcal{G}[k] = \mathcal{G}[s_1] \cup \mathcal{G}[s_2] \cup \mathcal{G}[s_3]$$

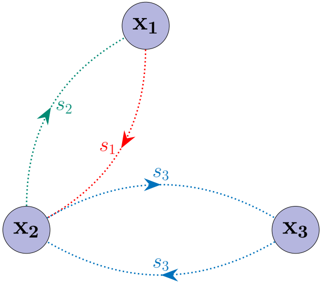
- A common assumption in coordination under switching graphs, e.g., (Ren and Beard, 2008).



Assumption: persistency of eventual connectivity

- $\mathcal{A}_3$: there is a strictly increasing sub-sequence of sampling indices $\{k_p\}$ such that for all $p \in \mathbb{Z}_+$
  1. the intervals $s_{k_{p+1}} - s_{k_p}$ are uniformly bounded;
  2. $\bigcup_{k=k_p+1}^{k_{p+1}} \mathcal{G}[k]$ contains a directed rooted tree.

$$\bigcup_{k=k_0+1}^{k_1} \mathcal{G}[k] = \mathcal{G}[s_1] \cup \mathcal{G}[s_2] \cup \mathcal{G}[s_3]$$



- A common assumption in coordination under switching graphs, e.g., (Ren and Beard, 2008).

If $\mathcal{A}_3$ holds then $\bar{\mu}(t) \to \mathbb{1} \otimes \mathrm{e}^{A_0 t} r_0$ for some r_0.

Outline

What changes for output feedback?

- $x_i(t)$ is no longer available $\implies$ $\mu_{ii}(t) \not\equiv x_i(t)$.
- Using an observer, we have that $\mu_{ii}(t) \equiv \hat{x}_i(t)$ for some

$$\dot{\hat{x}}_i(t) = A\hat{x}_i(t) + Bu_i(t) - L(y_i(t) - C\hat{x}_i(t)).$$

What changes for output feedback?

- $x_i(t)$ is no longer available $\implies$ $\mu_{ii}(t) \not\equiv x_i(t)$.
- Using an observer, we have that $\mu_{ii}(t) \equiv \hat{x}_i(t)$ for some

$$\dot{\hat{x}}_i(t) = A\hat{x}_i(t) + Bu_i(t) - L(y_i(t) - C\hat{x}_i(t)).$$

- Repeating the procedure yields

$$\begin{cases} \begin{bmatrix} \dot{e}(t) \\ \dot{\epsilon}(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \left[\begin{array}{cc|c} I \otimes (A_\mathrm{d} + B\bar{F}) & -I \otimes M & 0 \\ 0 & I \otimes (A + LC) & 0 \\ \hline 0 & -I \otimes (\frac{1}{\nu}LC) & I \otimes A_0 \end{array}\right] \begin{bmatrix} e(t) \\ \epsilon(t) \\ \bar{\mu}(t) \end{bmatrix} \\ \begin{bmatrix} e(s_k^+) \\ \epsilon(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I_n \\ 0 & I & 0 \\ 0 & 0 & (I_\nu - (1/\nu)\mathcal{L}[k]) \otimes I_n \end{bmatrix} \begin{bmatrix} e(s_k) \\ \epsilon(s_k) \\ \bar{\mu}(s_k) \end{bmatrix} \end{cases}$$

where

$$e(t) := x(t) - \bar{\mu}(t), \quad \epsilon := x - \hat{x}, \quad \text{and} \quad M = BF_\mathrm{d} + (1/\nu)LC.$$

What changes for output feedback?

- $x_i(t)$ is no longer available $\implies$ $\mu_{ii}(t) \not\equiv x_i(t)$.
- Using an observer, we have that $\mu_{ii}(t) \equiv \hat{x}_i(t)$ for some

$$\dot{\hat{x}}_i(t) = A\hat{x}_i(t) + Bu_i(t) - L(y_i(t) - C\hat{x}_i(t)).$$

- Repeating the procedure yields

$$\begin{cases} \begin{bmatrix} \dot{e}(t) \\ \dot{\epsilon}(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \left[\begin{array}{cc|c} I \otimes (A_\mathrm{d} + B\bar{F}) & -I \otimes M & 0 \\ 0 & I \otimes (A + LC) & 0 \\ \hline 0 & -I \otimes (\frac{1}{\nu} LC) & I \otimes A_0 \end{array}\right] \begin{bmatrix} e(t) \\ \epsilon(t) \\ \bar{\mu}(t) \end{bmatrix} \\ \begin{bmatrix} e(s_k^+) \\ \epsilon(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I_n \\ 0 & I & 0 \\ 0 & 0 & (I_\nu - (1/\nu)\mathcal{L}[k]) \otimes I_n \end{bmatrix} \begin{bmatrix} e(s_k) \\ \epsilon(s_k) \\ \bar{\mu}(s_k) \end{bmatrix} \end{cases}$$

where

$$e(t) := x(t) - \bar{\mu}(t), \quad \epsilon := x - \hat{x}, \quad \text{and} \quad M = BF_\mathrm{d} + (1/\nu)LC.$$

A world without disagreement

- Emulate an even more optimistic scenario: assume our emulators instantly synchronize.

A world without disagreement

- Emulate an even more optimistic scenario: assume our emulators instantly synchronize.
- Hence, $\mu_\delta \equiv 0 \implies$ We can directly emulate $\bar{\mu}_i$.

A world without disagreement

- Emulate an even more optimistic scenario: assume our emulators instantly synchronize.
- Hence, $\mu_\delta \equiv 0 \implies$ We can directly emulate $\bar{\mu}_i$.
- We retain the same control-law structure

$$u_i(t) = F_d \hat{x}_i(t) + (\bar{F} - F_{\mathsf{d}}) \bar{\mu}_i(t),$$

but now

$$\bar{\mu}_i \neq (1/\nu) \left(\sum_{j=1, j \neq i}^{\nu} \mu_{ij}(t) + \hat{x}_i(t) \right)$$

but rather an independent variable.

Outline

The main result

Theorem

If F_d and L are such that $A + BF_d$ and $A + LC$ are Hurwitz and $\bar{F}$ is such that $A_0 = A + B\bar{F}$, then the local sampled-data controllers

$$\begin{cases} \dot{\hat{x}}_i(t) = A\hat{x}_i(t) + Bu_i(t) - L(y_i(t) - C\hat{x}_i(t)) \\ \dot{\bar{\mu}}_i(t) = A_0\bar{\mu}_i(t) \\ \bar{\mu}_i(s_k^+) = \bar{\mu}_i(s_k) - \frac{1}{\nu}\sum_{j\in\mathcal{N}_i[k]}\big(\bar{\mu}_i(s_k) - \bar{\mu}_j(s_k)\big) \\ u_i(t) = F_d\hat{x}_i(t) + (\bar{F} - F_d)\bar{\mu}_i(t) \end{cases}$$

will asymptotically synchronize the agents for all initial conditions and all sampling sequences $\{s_k\}$ satisfying $\mathcal{A}_3$.

Proof outline - error dynamics

Once more defining

$$e := x - \bar{\mu}, \quad \text{and} \quad \epsilon := x - \hat{x}$$

yields the now decoupled *stable* flow dynamics

$$\begin{bmatrix} \dot{e}(t) \\ \dot{\epsilon}(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \left[\begin{array}{cc|c} I \otimes A_\mathsf{d} & -I \otimes (BF_\mathsf{d}) & 0 \\ 0 & I \otimes (A + LC) & 0 \\ \hline 0 & 0 & I \otimes A_0 \end{array}\right] \begin{bmatrix} e(t) \\ \epsilon(t) \\ \bar{\mu}(t) \end{bmatrix},$$

while at each sampling instance, $\{s_k\}$, the system obeys the discrete equation

$$\begin{bmatrix} e(s_k^+) \\ \epsilon(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I_n \\ 0 & I & 0 \\ 0 & 0 & (I - (1/\nu)\mathcal{L}[k]) \otimes I_n \end{bmatrix} \begin{bmatrix} e(s_k) \\ \epsilon(s_k) \\ \bar{\mu}(s_k) \end{bmatrix}$$

where $\mathcal{L}[k]$ is the Laplacian matrix associated with the network connectivity graph $\mathcal{G}[k]$ at s_k.

Proof outline - error dynamics

Once more defining

$$e := x - \bar{\mu}, \quad \text{and} \quad \epsilon := x - \hat{x}$$

yields the now decoupled *stable* flow dynamics

$$\begin{bmatrix} \dot{e}(t) \\ \dot{\epsilon}(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \left[\begin{array}{cc|c} I \otimes A_\mathrm{d} & -I \otimes (BF_\mathrm{d}) & 0 \\ 0 & I \otimes (A + LC) & 0 \\ \hline 0 & 0 & I \otimes A_0 \end{array}\right] \begin{bmatrix} e(t) \\ \epsilon(t) \\ \bar{\mu}(t) \end{bmatrix},$$

while at each sampling instance, $\{s_k\}$, the system obeys the discrete equation

$$\begin{bmatrix} e(s_k^+) \\ \epsilon(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I_n \\ 0 & I & 0 \\ 0 & 0 & (I - (1/\nu)\mathcal{L}[k]) \otimes I_n \end{bmatrix} \begin{bmatrix} e(s_k) \\ \epsilon(s_k) \\ \bar{\mu}(s_k) \end{bmatrix}$$

where $\mathcal{L}[k]$ is the Laplacian matrix associated with the network connectivity graph $\mathcal{G}[k]$ at s_k.

Proof outline - error dynamics

Once more defining

$$e := x - \bar{\mu}, \quad \text{and} \quad \epsilon := x - \hat{x}$$

yields the now decoupled *stable* flow dynamics

$$\begin{bmatrix} \dot{e}(t) \\ \dot{\epsilon}(t) \\ \dot{\bar{\mu}}(t) \end{bmatrix} = \left[\begin{array}{cc|c} I \otimes A_{\mathsf{d}} & -I \otimes (BF_{\mathsf{d}}) & 0 \\ 0 & I \otimes (A + LC) & 0 \\ \hline 0 & 0 & I \otimes A_0 \end{array}\right] \begin{bmatrix} e(t) \\ \epsilon(t) \\ \bar{\mu}(t) \end{bmatrix},$$

while at each sampling instance, $\{s_k\}$, the system obeys the discrete equation

$$\begin{bmatrix} e(s_k^+) \\ \epsilon(s_k^+) \\ \bar{\mu}(s_k^+) \end{bmatrix} = \begin{bmatrix} I & 0 & (1/\nu)\mathcal{L}[k] \otimes I_n \\ 0 & I & 0 \\ 0 & 0 & (I - (1/\nu)\mathcal{L}[k]) \otimes I_n \end{bmatrix} \begin{bmatrix} e(s_k) \\ \epsilon(s_k) \\ \bar{\mu}(s_k) \end{bmatrix}$$

where $\mathcal{L}[k]$ is the Laplacian matrix associated with the network connectivity graph $\mathcal{G}[k]$ at s_k.

Numerical example (I)

Consider $\nu = 3$ agents with

$$\left[\begin{array}{c|c} A & B \\ \hline C & \end{array}\right] = \left[\begin{array}{cc|c} 4 & 9 & 2 \\ 1 & 4 & 1 \\ \hline 1 & 0 & \end{array}\right], \quad A_0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

and gains

$$\bar{F} = -\begin{bmatrix} 2 & 4 \end{bmatrix}, \quad F_{\mathsf{d}} = -\begin{bmatrix} 7 & 1 \end{bmatrix}, \quad \text{and} \quad L = -\begin{bmatrix} 19 \\ 11 \end{bmatrix}.$$

Numerical example (I)

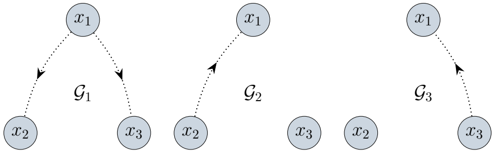
Consider $\nu = 3$ agents with

$$\left[\begin{array}{c|c} A & B \\ \hline C & \end{array}\right] = \left[\begin{array}{cc|c} 4 & 9 & 2 \\ 1 & 4 & 1 \\ \hline 1 & 0 & \end{array}\right], \quad A_0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

and gains

$$\bar{F} = -\begin{bmatrix} 2 & 4 \end{bmatrix}, \quad F_{\mathsf{d}} = -\begin{bmatrix} 7 & 1 \end{bmatrix}, \quad \text{and} \quad L = -\begin{bmatrix} 19 \\ 11 \end{bmatrix}.$$

- The sampling sequence is randomized between 3 graphs and satisfy $\mathcal{A}_{\mathbf{3}}$, with $s_{k+1} - s_k \in [0.45, 2.25]$ [sec]



Numerical example (II)

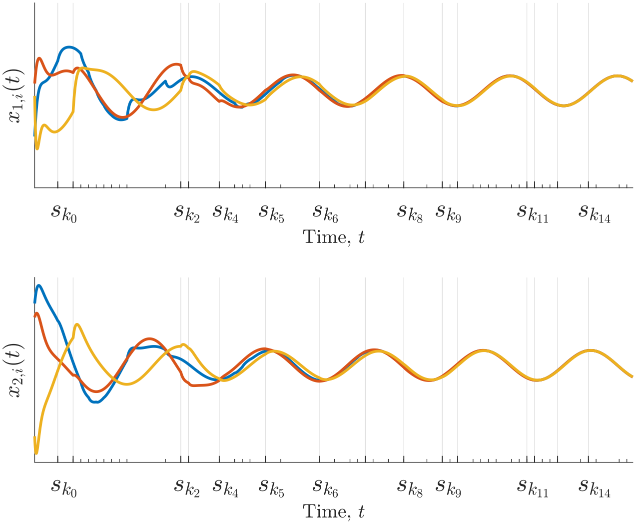


Figure: States' trajectories

Numerical example (II)

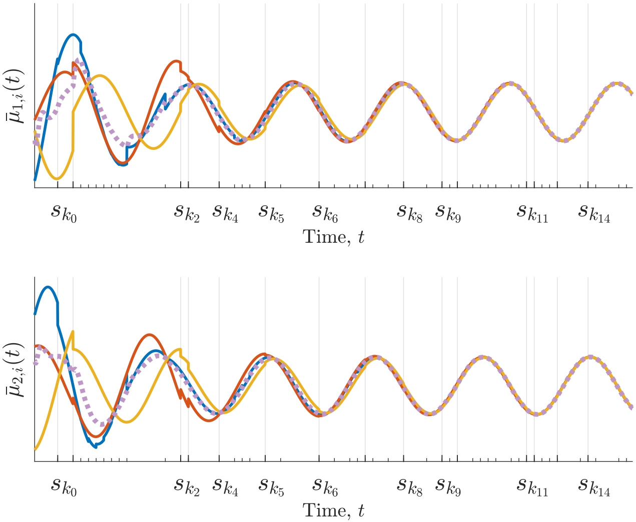


Figure: Centroid's trajectories

Numerical example (II)

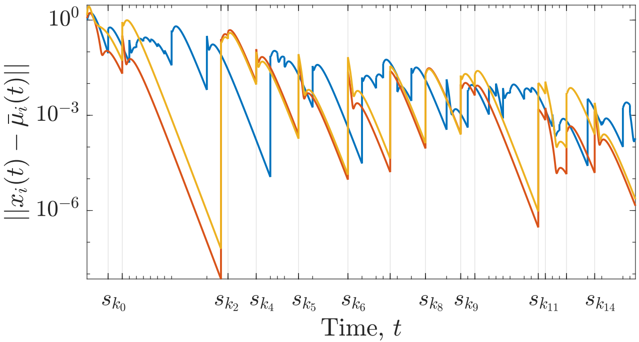


Figure: $\|e_i(t)\|$ (log scale)

Numerical example (II)

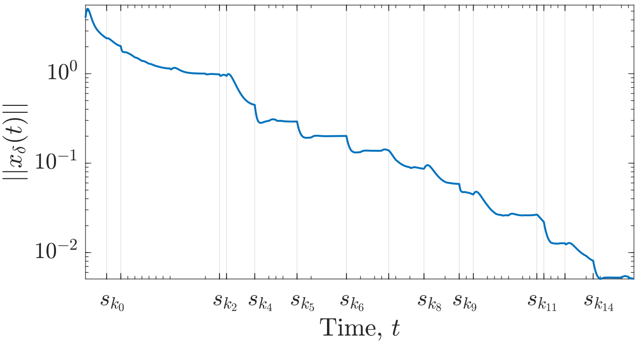


Figure: $\|x_\delta(t)\|$ (log scale)

Outline

Concluding remarks

- State synchronization is guaranteed under weak assumptions despite (almost) arbitrary sampling and output measurements.
- The gains design is intuitive, non-restrictive, and independent of the sampling sequence.
- If the union graph is *strongly* connected the convergence is exponential (in the paper).
- Low-order controller with scaleable design.
- Work in progress: output synchronization, heterogeneous agents, transmission delays, performance guarantees.