Practical Synchronization of Heterogeneous Multi-agent System Using Adaptive Law for Coupling Gains

Seungjoon Lee, Hyeonjun Yun and Hyungbo Shim

Control & Dynamic Systems Lab
Seoul National University

June 27th, 2018
Milwaukee, ACC 2018

Synchronization of Multi-agent System




- Dynamic network topology and heterogeneous agents

Dynamic Network Topology

- Synchronization has been studied with time-varying / switched network with fixed number of agent.
- However, there are cases where new agents may join and leave network during the operation.

In power network:

- Local renewable resources join and leave power network.

We study:

- Total number of agent is not necessarily fixed.

Dynamic Network Topology

- Synchronization has been studied with time-varying / switched network with fixed number of agent.
- However, there are cases where new agents may join and leave network during the operation.

In power network:

- Local renewable resources join and leave power network.

We study:

- Total number of agent is not necessarily fixed.

Practical Synchronization Problem

Synchronization problem of N agents can be formulated as

$$\dot{x}_i = F_i(x_i, t) + u_i$$

where $x_i \in \mathbb{R}^n$ is state, $F_i(x_i, t) : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ are heterogeneous vector fields, and u_i is distributed input (to be designed).

In particular, we consider input given by

$$u_i = h(e_i, \theta_i).$$

- e_i: stack of relative difference between x_i and its neighbors
- θ_i: design parameter
- $h(e_i, \theta_i)$: static or dynamic mapping

Practical Synchronization Problem

Synchronization problem of N agents can be formulated as

$$\dot{x}_i = F_i(x_i, t) + u_i$$

where $x_i \in \mathbb{R}^n$ is state, $F_i(x_i, t) : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ are heterogeneous vector fields, and u_i is distributed input (to be designed).

In particular, we consider input given by

$$u_i = h(e_i, \theta_i).$$

- e_i: stack of relative difference between x_i and its neighbors
- θ_i: design parameter
- $h(e_i, \theta_i)$: static or dynamic mapping

Practical Synchronization Problem

Practical Synchronization

Agents achieve practical synchronization if, for given $\epsilon > 0$, there exists θ_i such that

$$\limsup_{t\to\infty} |x_i(t) - x_j(t)| \leq \epsilon$$

holds for any i, j.

An example of valid coupling law is static diffusive coupling which is given by

$$u_i = k \sum_{j\in\mathcal{N}_i} (x_j - x_i)$$

where $k > 0$ is a common coupling gain .

Practical Synchronization Problem

Practical Synchronization

Agents achieve practical synchronization if, for given $\epsilon > 0$, there exists θ_i such that

$$\limsup_{t\to\infty} |x_i(t) - x_j(t)| \leq \epsilon$$

holds for any i, j.

An example of valid coupling law is static diffusive coupling which is given by

$$u_i = k \sum_{j\in\mathcal{N}_i} (x_j - x_i)$$

where $k > 0$ is a common coupling gain .

Review of High Gain Coupling

Recall with static coupling, we have

$$\dot{x}_i = F_i(x_i, t) + k \sum_{j \in \mathcal{N}_i} (x_j - x_i).$$

Consider *blended dynamics* which is defined as

$$\dot{s} = \frac{1}{N} \sum_{i=1}^{N} F_i(s, t),$$

with $s(0) = \frac{1}{N} \sum_{i=1}^{N} x_i(0)$.

Review of High Gain Coupling

Blended dynamics: $\dot{s} = \frac{1}{N}\sum_{i=1}^{N} F_i(s,t)$.

Theorem [JK16,JL18]

Suppose blended dynamics is contractive*. Then, for any $\epsilon > 0$, there exists k^* such that for all $k \geq k^*$,

$$\limsup_{t\to\infty} |x_i(t) - s(t)| \leq \epsilon, \quad \forall i \in \mathcal{N}.$$

- Practical synchronization is achieved.
- Trajectories of heterogeneous agents are described by the blended dynamics.

---

* $\dot{x} = f(x,t)$ is contractive if there exists positive definite matrix H and constant $p > 0$ such that $H(\partial f/\partial x)(x,t) + (\partial f/\partial x)^T(x,t)H \leq -pH$ for all $x \in \mathbb{R}^n$ and $t \geq 0$.

[JK16] Kim, Yang, Shim, Kim, Seo, (TAC, 2016)

[JL18] Lee, Shim (Arxiv, 2018)

Contribution of this work

Challenge

Coupling gain cannot be designed in a completely decentralized manner.

Previous works:

- Completely decentralized design was proposed for homogeneous case [ZL13,HK17].

In this paper:

- Achieve practical synchronization of heterogeneous multi-agent system using a completely decentralized design.
- Propose algorithm to maintain synchronization performance under dynamic network topology.

[ZL13] Li, Ren, Liu, Fu (TAC, 2013)
[HK17] Kim, Claudio (IJRNC, 2017)

Contribution of this work

Challenge

Coupling gain cannot be designed in a completely decentralized manner.

Previous works:

- Completely decentralized design was proposed for homogeneous case [ZL13,HK17].

In this paper:

- Achieve practical synchronization of heterogeneous multi-agent system using a completely decentralized design.
- Propose algorithm to maintain synchronization performance under dynamic network topology.

[ZL13] Li, Ren, Liu, Fu (TAC, 2013)
[HK17] Kim, Claudio (IJRNC, 2017)

Main Result: Adaptive Design

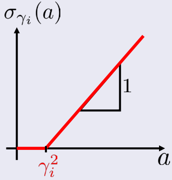
Adaptive Design

We propose the input to be

$$u_i = k_i(t) \sum_{j \in \mathcal{N}_i} (x_j - x_i)$$

$$\dot{k}_i = \sum_{j \in \mathcal{N}_i} \sigma_{\gamma_i}(e_{ji}^T e_{ji}) + \sum_{j \in \mathcal{N}_i} (k_j - k_i), \quad k_i(0) > 0$$

where $e_{ji} := x_j - x_i$ and $\sigma_{\gamma_i} : [0, +\infty) \to [0, +\infty)$ is the deadzone function with threshold $\gamma_i^2 > 0$.



Main Result: Adaptive Design

Consider the dynamics given by

$$\dot{x}_i = F_i(x_i, t) + k_i(t) \sum_{j \in \mathcal{N}_i} (x_j - x_i)$$

$$\dot{k}_i = \sum_{j \in \mathcal{N}_i} \sigma_{\gamma_i}(e_{ji}^T e_{ji}) + \sum_{j \in \mathcal{N}_i} (k_j - k_i), \quad k_i(0) > 0.$$

Theorem 1 (Node-wise Performance)

Suppose that $|F_i(x,t)| \leq M$, $\forall x \in \mathbb{R}^n$, $t \geq 0$, and the graph is connected. Then, the solution of the multi-agent system satisfies

$$\limsup_{t \to \infty} |x_i(t) - x_j(t)| \leq \gamma_i, \quad \forall j \in \mathcal{N}_i,$$

for all $i = 1, \ldots, N$. Moreover, there exists a constant $k^* > 0$ such that $\lim_{t \to \infty} k_i(t) = k^*$ for all $i = 1, \ldots, N$.

Main Result: Adaptive Design

Theorem 1 (Node-wise Performance)

Suppose that $|F_i(x,t)| \leq M$, $\forall x \in \mathbb{R}^n$, $t \geq 0$, and the graph is connected. Then, the solution of the multi-agent system satisfies

$$\limsup_{t\to\infty} |x_i(t) - x_j(t)| \leq \gamma_i, \quad \forall j \in \mathcal{N}_i,$$

for all $i = 1, \ldots, N$. Moreover, there exists a constant $k^* > 0$ such that $\lim_{t\to\infty} k_i(t) = k^*$ for all $i = 1, \ldots, N$.

- Only guarantees "node-wise performance"
- Due to symmetry, if i and j are neighbors,

$$\limsup_{t\to\infty} |x_i(t) - x_j(t)| \leq \min(\gamma_i, \gamma_j).$$

Main Result: Adaptive Design

Corollory 1 (Worst Case Performance)

Suppose Theorem 1 holds. Then, the multi-agent system achieves practical synchronization. In particular,

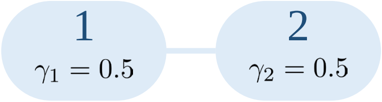
$$\limsup_{t\to\infty} |x_i(t) - x_j(t)| \leq (N-1)\cdot\bar{\gamma}$$

holds for all i, j where $\bar{\gamma} := \max_{i\in\mathcal{N}} \gamma_i$.

- Ensures “global performance”
- Worst case performance degrades as N grows.

Main Result: Algorithm

- Consider following system where $\gamma_1 = \gamma_2 = 0.5$.

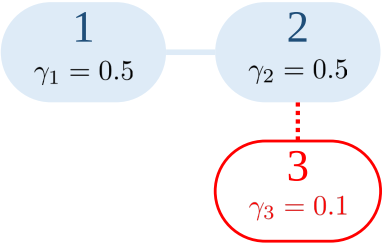


- Thus, the initial worst case performance can be obtained as

$$\begin{aligned}\limsup_{t\to\infty} |x_1(t) - x_2(t)| &\leq \gamma_1 \\ &= 0.5\end{aligned}$$

Main Result: Algorithm

- Suppose a new node is added to the system with $\gamma_3 = 0.1$.

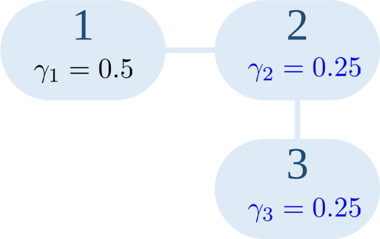


- Then, the worst case performance becomes

$$\begin{aligned}\limsup_{t\to\infty} |x_1 - x_3| &\leq \limsup_{t\to\infty} |x_1 - x_2| + \limsup_{t\to\infty} |x_2 - x_3| \\ &\leq \gamma_1 + \gamma_3 \\ &= 0.6\end{aligned}$$

Main Result: Algorithm

- Let γ_i's are updated such that $\gamma_2 = \gamma_3 = 0.25$ while $\gamma_1 = 0.5$ stays same.



- Then we can recover the worst case performance as

$$\begin{aligned}\limsup_{t\to\infty} |x_1 - x_3| &\leq \limsup_{t\to\infty} |x_1 - x_2| + \limsup_{t\to\infty} |x_2 - x_3| \\ &\leq \gamma_2 + \gamma_2 \\ &= 0.5\end{aligned}$$

Main Result: Algorithm

Threshold Update Protocol (TUP)

1. Agent N joins the network. Let its neighbors $\mathcal{N}_N$.
2. For all $i \in \mathcal{N}_N$, let $\hat{\gamma}_i^{[N]} := \min_{j \in \mathcal{N}_i \cup \{i\}, j \neq N}(\gamma_j^{[N-1]})$.
3. Agent N receives the value of $\hat{\gamma}_i^{[N]}$ for all $i \in \mathcal{N}_N$.
4. Agent N computes $\gamma^* := \min_{i \in \mathcal{N}_N} \hat{\gamma}_i^{[N]}$ and set $\gamma_N^{[N]} = \gamma^*/2$.
5. Agent N sends γ^* to its neighbors $\mathcal{N}_N$.
6. For all $i \in \mathcal{N}_N$, let $\gamma_i^{[N]} = (\hat{\gamma}_i^{[N]} - \gamma^*/2)$.

Finally, let $\gamma_i^{[N]} = \gamma_i^{[N-1]}$ for all $i \in \{1, \ldots, N-1\} \backslash \mathcal{N}_N$.

Theorem 2 (Summarized)

Suppose system with worst case performance of $\epsilon > 0$. Then the overall system with TUP maintains the performance.

Application of Adaptive Design to Distributed Optimization

Distributed Solution of Economic Dispatch Problem

Economic dispatch problem (EDP) is:

- Network of N nodes, where each node has power generation (p_i) and demand (p_i^d).
- Find optimal generation for each node to minimize the overall generation cost.

EDP can be written as

$$\min_{p_i} \sum a_i p_i^2 + b_i p_i + c_i \tag{1a}$$

$$\text{subject to} \quad p_{i,\min} \leq p_i \leq p_{i,\max} \tag{1b}$$

$$\sum p_i^d = \sum p_i \tag{1c}$$

- $p_{i,\min}$, $p_{i,\max}$: min/max generation capacity

Distributed Solution of Economic Dispatch Problem

Economic dispatch problem (EDP) is:

- Network of N nodes, where each node has power generation (p_i) and demand (p_i^d).
- Find optimal generation for each node to minimize the overall generation cost.

EDP can be written as

$$\min_{p_i} \sum a_i p_i^2 + b_i p_i + c_i \tag{1a}$$

$$\text{subject to} \quad p_{i,\min} \leq p_i \leq p_{i,\max} \tag{1b}$$

$$\sum p_i^d = \sum p_i \tag{1c}$$

- $p_{i,\min}$, $p_{i,\max}$: min/max generation capacity

Distributed Solution of Economic Dispatch Problem

Using Lagrangian dual functions, it is equivalent to solve [HY18]

$$\max_{\lambda \in \mathbb{R}} g(\lambda),$$

where λ is dual variable, $g(\lambda) = \sum g_i(\lambda)$ and $g_i(\lambda)$ can be computed locally by an agent.

Maximization problem can be solved by the gradient ascent method given by

$$\dot{\lambda} = \alpha \nabla g(\lambda) = \alpha \sum_{i=1}^{N} \frac{dg_i(\lambda)}{d\lambda}.$$

- $\alpha > 0$ is some constant.
- Stable if the optimization problem is feasible.
- Centralized method

[HY18] Yun, Shim, Ahn (Arxiv, 2018)

Distributed Solution of Economic Dispatch Problem

Using Lagrangian dual functions, it is equivalent to solve [HY18]

$$\max_{\lambda \in \mathbb{R}} g(\lambda),$$

where λ is dual variable, $g(\lambda) = \sum g_i(\lambda)$ and $g_i(\lambda)$ can be computed locally by an agent.

Maximization problem can be solved by the gradient ascent method given by

$$\dot{\lambda} = \alpha \nabla g(\lambda) = \alpha \sum_{i=1}^{N} \frac{dg_i(\lambda)}{d\lambda}.$$

- $\alpha > 0$ is some constant.
- Stable if the optimization problem is feasible.
- Centralized method

[HY18] Yun, Shim, Ahn (Arxiv, 2018)

Distributed Solution of Economic Dispatch Problem

We propose the distributed solution given by

$$\dot{\lambda}_i = \frac{dg_i}{d\lambda}(\lambda_i) + k_i(t) \sum_{j \in \mathcal{N}_i} (\lambda_j - \lambda_i)$$

$$\dot{k}_i = \sum_{j \in \mathcal{N}_i} \sigma_{\gamma_i} \left(e_{ji}^2 \right) + \sum_{j \in \mathcal{N}_i} (k_j - k_i)$$

where λ_i is the estimate of λ by agent i and $\frac{dg_i}{d\lambda}$ is uniformly bounded.

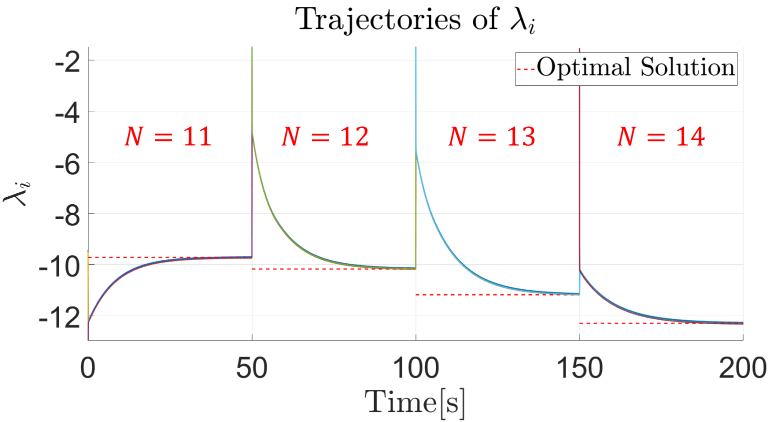
Recalling the high gain coupling, $\lambda_i(t)$ will converge to solution of blended dynamics given by

$$\dot{s} = \frac{1}{N} \sum_{i=1}^{N} \frac{dg_i(s)}{d\lambda} = \frac{1}{N} \nabla g(s)$$

which is exactly gradient ascent method.

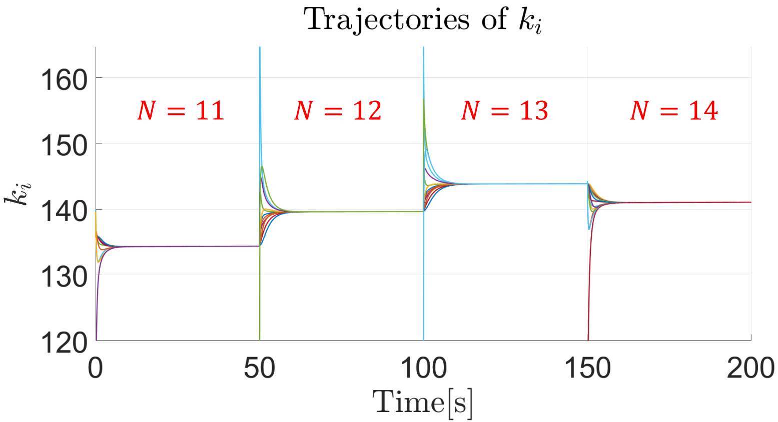
Simulation Results

Dual Variable λ_i



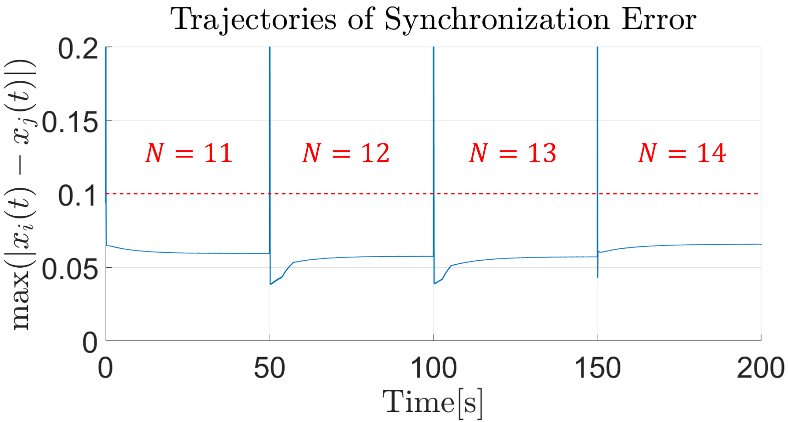
Simulation Results

Coupling Gains k_i



Simulation Results

Synchronization Error



Conclusion

Decentralized Design

- High gain coupling and practical synchronization of heterogeneous agents
- Adaptive design to achieve decentralized design
- Usage of deadzone function due to heterogeneity
- Synchronization of coupling gains to recover static gain and blended dynamics

Threshold Update Protocol

- Distributed algorithm to maintain worst case performance under dynamic graph topology

Thank You!