Zeno-free, distributed event-triggered coordination for multi-agent average consensus

Cameron Nowzari[1] Jorge Cortés[2]

[1] Electrical and Systems Engineering
University of Pennsylvania
cnowzari@seas.upenn.edu

[2] Mechanical and Aerospace Engineering
University of California, San Diego

American Control Conference
Portland, Oregon
June 5, 2014

-Multi-agent average consensus-

Consider

- N agents with state $x = (x_1, \dots, x_N) \in \mathbb{R}^N$
- each agent i can **communicate with neighbors** $j \in \mathcal{N}_i$ in undirected communication graph $\mathcal{G}$

-Multi-agent average consensus-

Consider

- N agents with state $x = (x_1, \dots, x_N) \in \mathbb{R}^N$
- each agent i can **communicate with neighbors** $j \in \mathcal{N}_i$ in undirected communication graph $\mathcal{G}$

$$\dot{x}_i(t) = u_i(t)$$

-Multi-agent average consensus-

Consider

- N agents with state $x = (x_1, \ldots, x_N) \in \mathbb{R}^N$
- each agent i can **communicate with neighbors** $j \in \mathcal{N}_i$ in undirected communication graph $\mathcal{G}$

$$\dot{x}_i(t) = u_i(t)$$

Well known distributed solution

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

-Multi-agent average consensus-

Consider

- N agents with state $x = (x_1, \ldots, x_N) \in \mathbb{R}^N$
- each agent i can **communicate with neighbors** $j \in \mathcal{N}_i$ in undirected communication graph $\mathcal{G}$

$$\dot{x}_i(t) = u_i(t)$$

Well known distributed solution

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

- Continuous local state information
- Continuous communication
- Continuous actuation

Digital controllers

Consider a single plant being controlled by a **microprocessor** through a feedback control loop

-25

$$\dot{x} = f(x, u)$$

Digital controllers

Consider a single plant being controlled by a **microprocessor** through a feedback control loop

-25measure x

$$\dot{x} = f(x, u)$$

Digital controllers

Consider a single plant being controlled by a **microprocessor** through a feedback control loop

-25 measure x compute $k(x)$

$$\dot{x} = f(x, u)$$

Digital controllers

Consider a single plant being controlled by a **microprocessor** through a feedback control loop

-25feedsure $\boldsymbol{k}(\boldsymbol{x})$

$$\dot{x} = f(x, u)$$

Digital controllers

Consider a single plant being controlled by a **microprocessor** through a feedback control loop

-25 feed measure x compute $k(x)$

$$\dot{x} = f(x, u)$$

Notice that this is different from the **idealistic** system

$$\dot{x} = f(x, k(x))$$

Digital controllers

-25feedsure compute $k(x)$

$$\dot{x} = f(x, u) \qquad \neq \qquad \dot{x} = f(x, k(x))$$

Most existing control theory was developed ignoring the implementation details

Digital controllers

-25feedsure compute $k(x)$

$$\dot{x} = f(x, u) \qquad \neq \qquad \dot{x} = f(x, k(x))$$

Most existing control theory was developed ignoring the implementation details

As long as $k(x)$ is updated **sufficiently fast**, everything will be okay

Digital controllers
-Time-triggered- control

Controller is updated **periodically** at an a priori chosen period T

Digital controllers

-Time-triggered- control

Controller is updated **periodically** at an a priori chosen period T

Benefits:

- **simple** and easy to implement
- does not require extra computations

Digital controllers

-Time-triggered- control

Controller is updated **periodically** at an a priori chosen period T

Benefits:

- **simple** and easy to implement
- does not require extra computations

Drawbacks:

- controller is often designed assuming perfect information
- state is **sampled** and controllers are **updated** periodically
- robustness analysis done a posteriori

Digital controllers

-Event-triggered- control

Consider a linear system

$$\dot{x} = Ax + Bu,$$

with ideal control law $u^* = Kx$ rendering the closed loop system asymptotically stable

Digital controllers

-Event-triggered- control

Consider a linear system

$$\dot{x} = Ax + Bu,$$

with ideal control law $u^* = Kx$ rendering the closed loop system asymptotically stable

Since we cannot apply u^* continuously, we will update it at a sequence of times $\{t_\ell\}$ instead. Between updates, the applied control is

$$u(t) = Kx(t_\ell) \qquad t \in [t_\ell, t_{\ell+1})$$

Digital controllers

-Event-triggered- control

Consider a linear system

$$\dot{x} = Ax + Bu,$$

with ideal control law $u^* = Kx$ rendering the closed loop system asymptotically stable

Since we cannot apply u^* continuously, we will update it at a sequence of times $\{t_\ell\}$ instead. Between updates, the applied control is

$$u(t) = Kx(t_\ell) \qquad t \in [t_\ell, t_{\ell+1})$$

Defining the error in the system as $e(t) = x(t) - x(t_\ell)$, the closed loop dynamics is

$$\begin{aligned} \dot{x} &= Ax(t) + BKx(t_\ell) \\ &= (A + BK)x(t) + BKe(t) \end{aligned}$$

Defining the error in the system as $e(t) = x(t) - x(t_\ell)$, the closed loop dynamics is

$$\begin{aligned}\dot{x} &= Ax(t) + BKx(t_\ell) \\ &= (A + BK)x(t) + BKe(t)\end{aligned}$$

Since $(A + BK)$ is stable, there exists a Lyapunov function V such that

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\|$$

Digital controllers
-Event-triggered- control

Defining the error in the system as $e(t) = x(t) - x(t_\ell)$, the closed loop dynamics is

$$\begin{aligned}\dot{x} &= Ax(t) + BKx(t_\ell) \\ &= (A + BK)x(t) + BKe(t)\end{aligned}$$

Since $(A + BK)$ is stable, there exists a Lyapunov function V such that

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\|$$

If we can now **enforce** that

$$\|e\| \leq \sigma \frac{a}{b}\|x\|$$

for some $\sigma \in (0, 1)$, then

$$\dot{V} \leq -(1 - \sigma)a\|x\|^2 < 0$$

Digital controllers

-Event-triggered- control

Event-trigger is given by

$$\|e\| = \sigma \frac{a}{b} \|x\|$$

Digital controllers

-Event-triggered- control

Event-trigger is given by

$$\|e\| = \sigma \frac{a}{b} \|x\|$$

Solves the problem of continuous actuation requirement

Event-trigger is given by

$$\|e\| = \sigma \frac{a}{b} \|x\|$$

Solves the problem of continuous actuation requirement

Still requires continuous communication in a network

Outline

Problem statement

The distributed, continuous control law

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

is **well known** to have each agent state asymptotically converge to the initial average of all agent states.

Problem statement

The distributed, continuous control law

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

is **well known** to have each agent state asymptotically converge to the initial average of all agent states.

Instead, we will use the control law

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t)),$$

where $\hat{x}_i(t)$ is the **last broadcast state** of agent i.

Problem statement

The distributed, continuous control law

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

is **well known** to have each agent state asymptotically converge to the initial average of all agent states.

Instead, we will use the control law

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t)),$$

where $\hat{x}_i(t)$ is the **last broadcast state** of agent i.

Problem (Multi-agent average consensus)

How should agents decide to broadcast their state to ensure their state converges to the initial average of all agent states?

Lyapunov design

Ideal controller

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

Implementable controller

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$$

Lyapunov design

Ideal controller

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

$$u^* = -Lx$$

Implementable controller

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$$

$$u = -L\hat{x}$$

Lyapunov design

Ideal controller

$$u_i^*(t) = -\sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t))$$
$$u^* = -Lx$$

Implementable controller

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$$
$$u = -L\hat{x}$$

Lyapunov function

$$V = x^T L x$$

Lyapunov design

Ideal controller

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

$$u^* = -Lx$$

Implementable controller

$$u_i(t) = -\sum_{j\in\mathcal{N}_i}(\hat{x}_i(t) - \hat{x}_j(t))$$

$$u = -L\hat{x}$$

Lyapunov function

$$V = x^T L x$$

$$\mathcal{L}_{u^*}V(x) = x^T L\dot{x}$$

$$\mathcal{L}_{u}V(x) = x^T L\dot{x}$$

Lyapunov design

Ideal controller

$$u_i^*(t) = -\sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t))$$
$$u^* = -Lx$$

Implementable controller

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$$
$$u = -L\hat{x}$$

Lyapunov function

$$V = x^T L x$$

$$\begin{aligned} \mathcal{L}_{u^*} V(x) &= x^T L \dot{x} \\ &= -x^T L L x \leq 0 \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{u} V(x) &= x^T L \dot{x} \\ &= -x^T L L \hat{x} \leq ??? \end{aligned}$$

Lyapunov design

Problem

How does $\hat{x}$ need to be updated such that

$$\mathcal{L}_u V(x) = \dot{V} = -x^T L L \hat{x} \leq 0$$

at all times?

Lyapunov design

Problem

How does $\hat{x}$ need to be updated such that

$$\mathcal{L}_u V(x) = \dot{V} = -x^T L L \hat{x} \leq 0$$

at all times?

The ideal way would be to update $\hat{x}$ each time $-x^T L L \hat{x} = 0$. But the problem with this is that it is centralized and requires perfect information.

Lyapunov design

Problem

How does $\hat{x}$ need to be updated such that

$$\mathcal{L}_u V(x) = \dot{V} = -x^T L L \hat{x} \leq 0$$

at all times?

The ideal way would be to update $\hat{x}$ each time $-x^T L L \hat{x} = 0$. But the problem with this is that it is centralized and requires perfect information.

So how can we do this in a **distributed** way?

Lyapunov design

Problem

How does $\hat{x}$ need to be updated such that

$$\mathcal{L}_u V(x) = \dot{V} = -x^T L L \hat{x} \leq 0$$

at all times?

The ideal way would be to update $\hat{x}$ each time $-x^T L L \hat{x} = 0$. But the problem with this is that it is centralized and requires perfect information.

So how can we do this in a **distributed** way?

Let $e = \hat{x} - x$, then

$$\begin{aligned} \dot{V} &= -(\hat{x} - e)^T L L \hat{x} \\ &= -\|L\hat{x}\|^2 + (L\hat{x})^T L e \end{aligned}$$

Lyapunov design

Let $\hat{z} = L\hat{x}$, then

$$\dot{V} = -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \hat{z}_i (e_i - e_j)$$

Lyapunov design

Let $\hat{z} = L\hat{x}$, then

$$\begin{aligned}\dot{V} &= -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \hat{z}_i (e_i - e_j) \\ &= -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} |\mathcal{N}_i| \hat{z}_i e_i - \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \hat{z}_i e_j\end{aligned}$$

Lyapunov design

Let $\hat{z} = L\hat{x}$, then

$$\dot{V} = -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \hat{z}_i(e_i - e_j)$$
$$= -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} |\mathcal{N}_i| \hat{z}_i e_i - \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \hat{z}_i e_j$$

By Young's inequality,

$$\sum_{i=1}^{N} |\mathcal{N}_i| \hat{z}_i e_i \leq \sum_{i=1}^{N} \left(\frac{1}{2} |\mathcal{N}_i| \hat{z}_i a + \frac{1}{2a} |\mathcal{N}_i| e_i^2 \right)$$

and

$$-\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \hat{z}_i e_j \leq \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left(\frac{1}{2} \hat{z}_i^2 a + \frac{1}{2a} e_j^2 \right)$$

Lyapunov design

So we can bound

$$\dot{V} \leq -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} \left(\frac{1}{2} |\mathcal{N}_i| \hat{z}_i a + \frac{1}{2a} |\mathcal{N}_i| e_i^2 \right) + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left(\frac{1}{2} \hat{z}_i^2 a + \frac{1}{2a} e_j^2 \right)$$

Lyapunov design

So we can bound

$$\dot{V} \le -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} \left(\frac{1}{2} |\mathcal{N}_i| \hat{z}_i a + \frac{1}{2a} |\mathcal{N}_i| e_i^2 \right) + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left(\frac{1}{2} \hat{z}_i^2 a + \frac{1}{2a} e_j^2 \right)$$

Since the graph is undirected, we know

$$\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{1}{2a} e_j^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{1}{2a} e_i^2 = \sum_{i=1}^{N} \frac{1}{2a} |\mathcal{N}_i| e_i^2$$

Lyapunov design

So we can bound

$$\dot{V} \leq -\sum_{i=1}^{N} \hat{z}_i^2 + \sum_{i=1}^{N} \left(\frac{1}{2} |\mathcal{N}_i| \hat{z}_i a + \frac{1}{2a} |\mathcal{N}_i| e_i^2 \right) + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left(\frac{1}{2} \hat{z}_i^2 a + \frac{1}{2a} e_j^2 \right)$$

Since the graph is undirected, we know

$$\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{1}{2a} e_j^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{1}{2a} e_i^2 = \sum_{i=1}^{N} \frac{1}{2a} |\mathcal{N}_i| e_i^2$$

Then,

$$\dot{V} \leq \sum_{i=1}^{N} (a|\mathcal{N}_i| - 1) \hat{z}_i^2 + \frac{1}{a} |\mathcal{N}_i| e_i^2$$

Lyapunov design

We can now present the **distributed** **event-triggering condition** as

$$e_i^2 = \frac{a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\sigma_i \hat{z}_i^2$$

Lyapunov design

We can now present the **distributed** **event-triggering condition** as

$$e_i^2 = \frac{a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\sigma_i \hat{z}_i^2$$

This means that

$$e_i^2 \leq \frac{a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\sigma_i \hat{z}_i^2$$

at all times,

Lyapunov design

We can now present the **distributed** **event-triggering condition** as

$$e_i^2 = \frac{a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\sigma_i \hat{z}_i^2$$

This means that

$$e_i^2 \leq \frac{a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\sigma_i \hat{z}_i^2$$

at all times, then

$$\dot{V} \leq -\sum_{i=1}^{N}(1-\sigma_i)(1-a|\mathcal{N}_i|)\hat{z}_i^2 \leq 0 \quad \text{for} \quad a < \min_{i\in\{1,\ldots,N\}} \frac{1}{|\mathcal{N}_i|}$$

Lyapunov design

Theorem (E. Garcia et al. 2013)

The system with the described control law and event-triggered broadcasting algorithm converges to the average consensus state.

Lyapunov design

Theorem (E. Garcia et al. 2013)

The system with the described control law and event-triggered broadcasting algorithm converges to the average consensus state.

Drawbacks:

- Parameter a needs to be found and agreed on a priori

Lyapunov design

Theorem (E. Garcia et al. 2013)

The system with the described control law and event-triggered broadcasting algorithm converges to the average consensus state.

Drawbacks:

- Parameter a needs to be found and agreed on a priori
- Possibility of Zeno executions

Lyapunov design

Theorem (E. Garcia et al. 2013)

The system with the described control law and event-triggered broadcasting algorithm converges to the average consensus state.

Drawbacks:

- Parameter a needs to be found and agreed on a priori
- Possibility of Zeno executions

Theorem (E. Garcia et al. 2013)

The inter-event times for each agent $i = \{1, \ldots, N\}$ *are strictly positive.*

Modified algorithm

Main trigger

$$e_i^2 \geq \sigma_i \frac{a_i}{|\mathcal{N}_i|} \left(1 - \frac{1}{2} a_i |\mathcal{N}_i| - \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_j \right) \hat{z}_i^2$$

Agents only need to know **local parameters** a_i and a_j for $j \in \mathcal{N}_i$

Modified algorithm

Main trigger

$$e_i^2 \geq \sigma_i \frac{a_i}{|\mathcal{N}_i|} \left(1 - \frac{1}{2} a_i |\mathcal{N}_i| - \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_j \right) \hat{z}_i^2$$

Agents only need to know **local parameters** a_i and a_j for $j \in \mathcal{N}_i$

Additional trigger added to ensure **no Zeno behavior** can occur

Theorem

The system with the described control law and modified event-triggered broadcasting algorithm exponentially converges to the average consensus state and is guaranteed to avoid Zeno executions.

Simulations

- Simulation with $N = 5$ agents
- $a_1 = a_3 = a_5 = 0.3$
- $a_2 = a_4 = 0.2$
- $\sigma_i = 0.999$ for all agents

Evolution of state trajectories

Number of events triggered

Evolution of Lyapunov function V

Conclusions

Distributed event-triggered broadcasting and control algorithm

- does not require any global a priori knowledge
- **no Zeno** executions
- **exponential** convergence rate
- extension to **time-varying topologies**

Conclusions

Distributed event-triggered broadcasting and control algorithm

- does not require any global a priori knowledge
- **no Zeno** executions
- **exponential** convergence rate
- extension to **time-varying topologies**

Future work:

- sampled-data implementations
- directed graphs
- more general algorithms

Thank You!