Finite-time Average Consensus in a Byzantine Environment Using Set-Valued Observers

D. Silvestre, P. Rosa, J. Hespanha and C. Silvestre

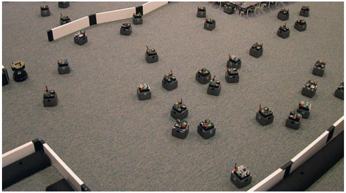
2014 American Control Conference
Portland, Oregon, USA.

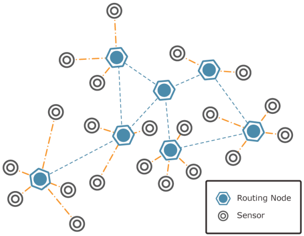
5th June 2014

Outline

Motivation

- Average Consensus Problem - m nodes wish to agree on the average of their initial values.
- Robot Coordination - Fleet of robots wishes to agree on direction/speed or rendezvous point.
- Sensor Network - Compute the mean of a noise corrupted set of sensor measurements.



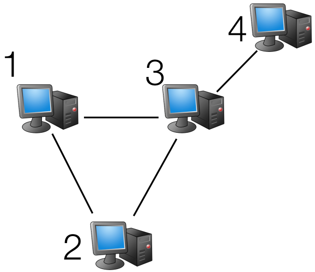


Gossip Consensus

- Pairs of nodes exchange their states at random times.
- Nodes transmit only to adjacent nodes in the connectivity graph.
- Both nodes average their states.
- See S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," IEEE Transactions on Information Theory, vol.52, no.6, pp.2508 - 2530, June 2006.

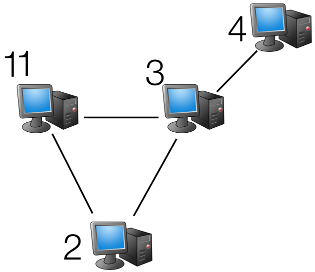
Motivating Example

- Take the network given by the following graph and state at time instant k, $x(k) = [1, 2, 3, 4]^\mathsf{T}$
- The consensus algorithm will asymptotically converge to all the nodes having $x_{\text{av}} = 2.5$
- If an attacker controlling one node in one time instant resets the state to $\tilde{x}_1(k) = x_1(k) + m(c - x_{\text{av}})$
- In this example $c = 5$, but the attacker can force the network to converge to any constant c!



Motivating Example

- Take the network given by the following graph and state at time instant k, $x(k) = [1, 2, 3, 4]^\intercal$
- The consensus algorithm will asymptotically converge to all the nodes having $x_{\text{av}} = 2.5$
- If an attacker controlling one node in one time instant resets the state to $\tilde{x}_1(k) = x_1(k) + m(c - x_{\text{av}})$
- In this example $c = 5$, but the attacker can force the network to converge to any constant c!



Problem Outline

- Take m agents running a consensus system of the form

$$\begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) \\ y(k) = C(k)x(k) \end{cases}$$

- $u(k)$ is the attacker signal, $B(k)$ the matrix selecting which nodes are corrupted and $A(k)$ a matrix randomly selected.

Byzantine Consensus Problem

Either detect non-zero signals $u(k)$ using $y(k)$ without the knowledge of $B(k)$ or compute the final consensus value.

Problem Model

- Each agent i has a system of the form

$$S^i : \begin{cases} x(k+1) = \left(A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell(k) A_\ell\right) x(k) + B(k)u(k) \\ y^i(k) = C^i(k)x(k) \end{cases}$$

- Each S^i is a Linear Parameter-Varying (LPV) system
- n_Δ number of uncertainties
- $\Delta_\ell(k)$ are scalar uncertainties with $|\Delta_\ell(k)| = 1$
- A_ℓ are constant matrices

Proposed Solution

Without Sharing Estimates

- Use Set-Valued Observers (SVOs) to generate a set $\tilde{X}(k)$ where the state can take values;
- Each node only uses state measurements;
- Conservative results but it communicates less information.

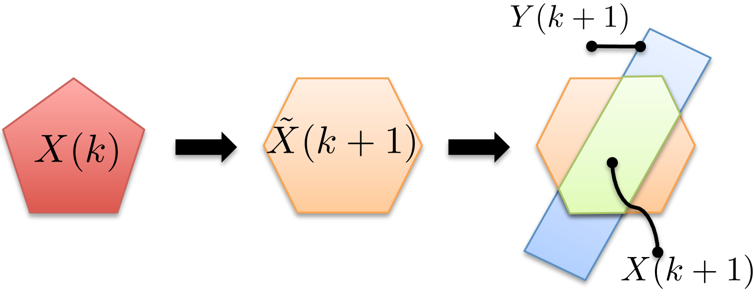
Sharing State Estimates

- The state of each node is a vector defining a hyper-parallelepiped overbound for the state estimates;
- Requires exchanging set-valued estimates between communicating neighbors;
- Achieves finite-time consensus for any horizon value.

SVOs

Given the previous set $X(k)$:

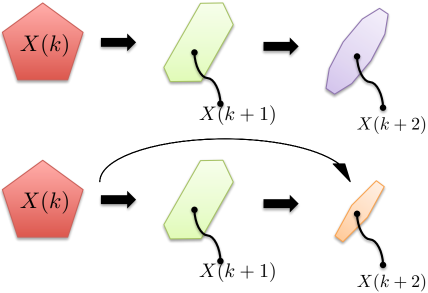
- Using SVOs, the algorithm predicts $\tilde{X}(k+1)$ using the dynamics;
- Then, the set is intersected with the measurement set $Y(k+1)$.



SVOs with horizon N

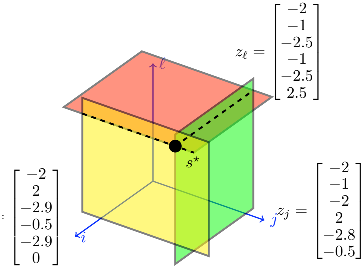
Given the previous set $X(k)$:

- A possible way to reduce the conservatism is by considering larger horizon values;
- The previous set is propagated once and intersected with the double propagation of the set from two time instants ago.



Algorithm

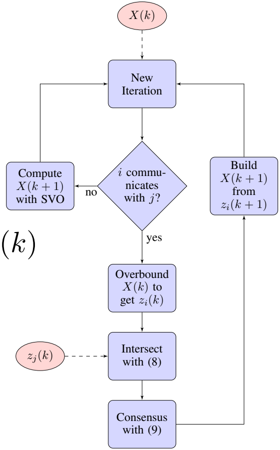
- Each node runs an independent SVO that uses only local information;
- Each node computes a hyper-parallelepiped overbound;
- When communicating, nodes exchange and intersect the vector defining its overbound;
- If the intersection is void a fault is declared, otherwise nodes perform an average consensus rule.



Algorithm steps

$$(8) z_i(k) = z_j(k) = \max(z_i(k), z_j(k))$$

$$(9) z_i(k+1) = \left[\left(\frac{1}{2}(e_i - e_j)(e_j - e_i)^{\intercal} + I_{n_x}\right) \otimes I_2\right] z_i(k)$$



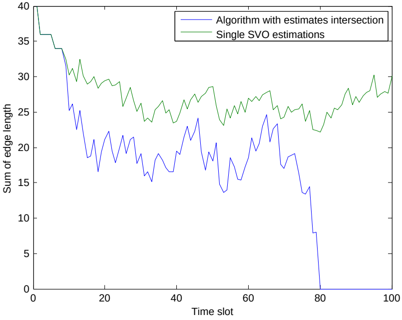
Properties

- In the absence of faults, the algorithm converges almost surely, in expectation and in mean square to the average consensus;
- Finite-time average consensus is achieved for any horizon value for some communication patterns;
- Using a token-passing scheme implements the type of communication patterns;
- It enables finite-time consensus with low computational burden when compared to the setup where nodes do not exchange estimates.

Simulation Results (1/2)

Setup: 5 node network and initial states $x_i(0) = i - 1$ and a nominal bound $|x_i| \leq 5$ and a horizon $N = 1$.

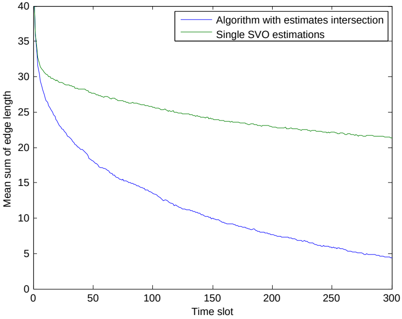
- In a typical run, the proposed algorithm achieves finite-time consensus.
- Without sharing state estimates, the size of the estimate set decreases slowly.
- Mean volume of the estimate set across all nodes.
- In addition to achieving finite-time consensus, the results are less conservative.



Simulation Results (1/2)

Setup: 5 node network and initial states $x_i(0) = i - 1$ and a nominal bound $|x_i| \leq 5$ and a horizon $N = 1$.

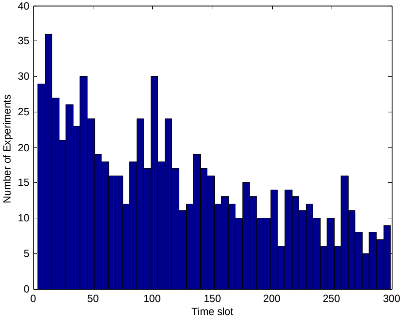
- In a typical run, the proposed algorithm achieves finite-time consensus.
- Without sharing state estimates, the size of the estimate set decreases slowly.
- Mean volume of the estimate set across all nodes.
- In addition to achieving finite-time consensus, the results are less conservative.



Simulation Results (2/2)

Running a 1000 Monte-Carlo runs, we depict the histogram of the stopping time.

- For a horizon $N = 1$, 21.9% of the runs did not achieve finite-time consensus.
- Using the same communication pattern, if $N = 5$ only 13.4% did not stop within the 300 time instants of the simulation.
- To get 100%, either increase the horizon N or the runtime.



Concluding Remarks

Contributions:

- Finite-time consensus is shown to be a property of the SVOs at the expenses of large horizon values;
- The introduction of an algorithm that shares set estimates to obtain less conservative results:
  - Possible conservativeness due to the overbound;
  - Finite-time consensus for any horizon value.
- In Simulation is shown:
  - The algorithm halts faster than when no estimates are exchanged;
  - Set volume is smaller leading to faster fault detection.

The end

- Thank you for your time.

Finite-time Average Consensus in a Byzantine Environment Using Set-Valued Observers

D. Silvestre, P. Rosa, J. Hespanha and C. Silvestre

2014 American Control Conference
Portland, Oregon, USA.

5th June 2014