Adaptation and Synchronization over a Network: stabilization without a reference model

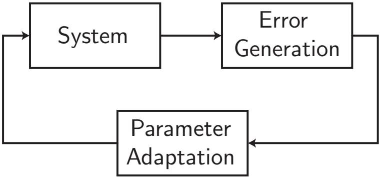
Travis E. Gibson (tgibson@mit.edu)

Harvard Medical School
Department of Pathology, Brigham and Women's Hospital

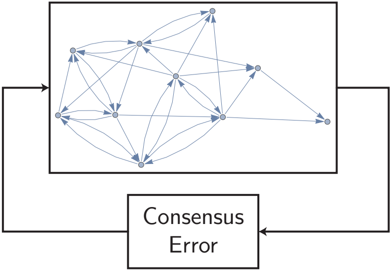
55$^{\text{th}}$ Conference on Decision and Control
December 12-14, 2016

Problem Statement

Adaptive Systems



Network Consensus



- How do we achieve consensus and learning **without** a reference model?
- Can synchronous inputs enhance adaptation?

Introduction and Outline

- Synchronization can hurt learning
  - Example of two unstable systems (builds on Narendra's recent work)
- Synchronization and Learning in Networks
  - Results Using Graph Theory
- Concrete connections to classic adaptive control (if time allows)

Synchronization vs. Learning: Tradeoffs

Two systems stabilizing each other

Consider two unstable systems [Narendra and Harshangi (2014)]

$$\Sigma_1 : \quad \dot{x}_1(t) = a_1(t)x_1(t) + u_1(t)$$
$$\Sigma_2 : \quad \dot{x}_2(t) = a_2(t)x_2(t) + u_2(t)$$

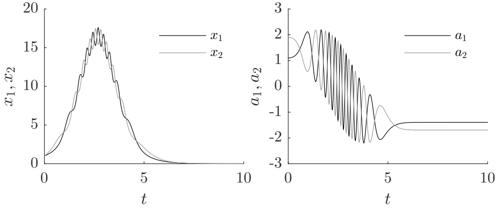
Update laws

$$\dot{a}_1(t) = -x_1(t)e(t) \quad a_1(0) > 0$$
$$\dot{a}_2(t) = x_2(t)e(t) \quad a_2(0) > 0$$

with $e = x_1 - x_2$.

No Input

$$u_1 = 0$$
$$u_2 = 0$$

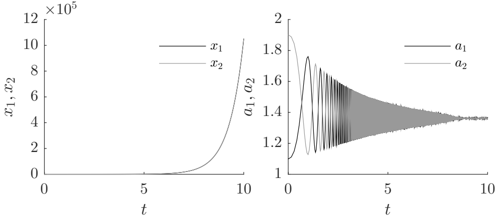


Synchronization **Hurts** Learning

Synchronous Input

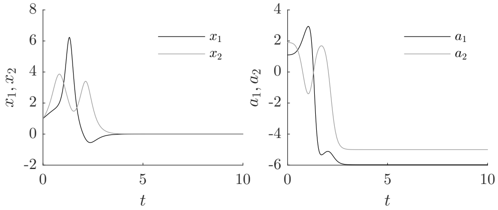
$$u_1 = -e$$
$$u_2 = +e$$

$$e = x_1 - x_2$$



Desynchronous Input

$$u_1 = +e$$
$$u_2 = -e$$



Stability Results for Synchronous and Desynchronous Inputs

$$\Sigma_1 : \quad \dot{x}_1(t) = a_1(t)x_1(t) + u_1(t)$$
$$\Sigma_2 : \quad \dot{x}_2(t) = a_2(t)x_2(t) + u_2(t)$$
$$\dot{a}_1(t) = -x_1(t)e(t)$$
$$\dot{a}_2(t) = x_2(t)e(t)$$

Theorem: Synchronous Inputs

The dynamics above with synchronous inputs have a set of initial conditions with non-zero measure for which x_1 and x_2 tend to infinity while $e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ and $e \to 0$ as $t \to \infty$. Furthermore, this set of initial conditions that are unstable is also unbounded.

Theorem: Desynchronous Inputs

The dynamics above with desynchronous inputs are stable for all $a_1(0) \neq a_2(0)$

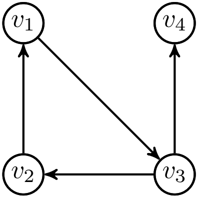
Synchronization and learning in networks

Graph Notation and Consensus

Graph : $\mathcal{G}(\mathcal{V}, \mathcal{E})$

Vertex Set : $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$

Edge Set : $(v_i, v_j) \in \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$



Adjacency Matrix : $[\mathcal{A}]_{ij} = \begin{cases} 1 & \text{if } (v_j, v_i) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$

In-degree Laplacian : $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$

In-degree of Node i : $[\mathcal{D}]_{ii}$

Consensus Problem

$$\Sigma_i : \quad \dot{x}_i = -\sum_{j \in \mathcal{N}_{\text{in}}(i)} (x_i - x_j)$$

Using Graph Notation

$$\Sigma : \quad \dot{x} = -\mathcal{L}x, \qquad x = [x_1,\ x_2, \ldots,\ x_n]^{\mathsf{T}}$$

Review: Sufficient Condition for Consensus

$$\Sigma : \quad \dot{x} = -\mathcal{L}x$$

Theorem: (Olfati-Saber and Murray, 2004)

For the dynamics above with $\mathcal{G}$ **strongly connected** it follows that $\lim_{t\to\infty} x(t) = \zeta\mathbf{1}$, for some finite $\zeta \in \mathbb{R}$. If $\mathcal{G}$ is also **balanced** then $\zeta = \frac{1}{n}\sum_{i=1}^{n} x_i(0)$, i.e. average consensus is reached.

strongly connected there is a walk between any two vertices in the network.

balanced if the in-degree of each node is equal to its out-degree.

- Any strongly connected digraph can be balanced (Marshall and Olkin, 1968).
- Distributed algorithms exist to balance a digraph (Dominguez-Garcia and Hadjicostis, 2013).

Return to Adaptive Stabilization

Consider a set of n possibly unstable systems

$$\Sigma_i \quad \dot{x}_i(t) = a_i x_i + \theta_i(t) x$$

Update Law

$$\dot{\theta}_i = -x_i \sum_{j \in \mathcal{N}_{\text{in}}(i)} (x_i - x_j)$$

Compact form

$$\begin{aligned} \Sigma: \qquad & \dot{x} = Ax + \operatorname{diag}(\theta) x \qquad & [A]_{ii} = a_i \\ & \dot{\theta} = -x \circ \mathcal{L} x \qquad & \theta = [\theta_1,\ \theta_2, \ldots,\ \theta_n]^{\mathsf{T}} \end{aligned}$$

Stabilization over Strongly Connected Graphs

$$\dot{x} = Ax + \operatorname{diag}(\theta)x$$
$$\dot{\theta} = -x \circ \mathcal{L}x$$

Theorem

For the dynamics above with $\mathcal{G}$ a strongly connected digraph, and all the $a_i + \theta_i(0)$ not identical it follows that $\lim_{t\to\infty} x(t) = \mathbf{0}$.

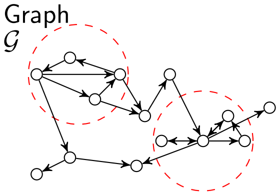
- $\mathcal{G}$ is strongly connected $\implies$ $\lambda_i(\mathcal{L}) \in$ closed right-half plane of $\mathbb{C}$.
- $-\mathcal{L}$ is **Metzler** $\implies$ $\exists$ Diagonal $D > 0$ s.t. $-\mathcal{L}^{\mathsf{T}}D - D\mathcal{L} \leq 0$.
- Non-increasing function

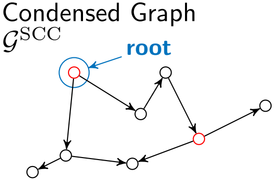
$$\sum_{i=1}^{n}[D]_{ii}\theta_i(t) = -\int_0^t x^{\mathsf{T}} D\mathcal{L}x \,\mathrm{d}t + \sum_{i=1}^{n}[D]_{ii}\theta_i(0)$$
$$= -\frac{1}{2}\int_0^t x^{\mathsf{T}}(D\mathcal{L} + \mathcal{L}^{\mathsf{T}}D)x \,\mathrm{d}t + \sum_{i=1}^{n}[D]_{ii}\theta_i(0).$$

- $\mathcal{L}\mathbf{1} = \mathbf{0} \implies \mathbf{1}^{\mathsf{T}}(D\mathcal{L} + \mathcal{L}^{\mathsf{T}}D)\mathbf{1} = 0.$
- $\exists\ \kappa \triangleq \lambda_2(D\mathcal{L} + \mathcal{L}^{\mathsf{T}}D) > 0 \implies \sum_i [D]_{ii}\theta_i(t) \leq -\kappa \int x^{\mathsf{T}}x \,\mathrm{d}t + \sum_i \theta_i(0)$ when $x \notin \operatorname{span}(\mathbf{1})$. □

Stabilization over Connected Graphs

- Any connected digraph can be partitioned into disjoint subsets called **Strongly Connected Components** (SCCs) where each subsets is a maximal strongly connected subgraph





condensed nodes in **red**

- For any connected $\mathcal{G}$ the corresponding $\mathcal{G}^{\mathrm{SCC}}$ is a **Directed Acyclic Graph** (**DAG**)

- Every connected DAG contains a **root** node (not unique).

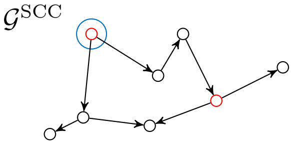
Stabilization over Connected Graphs Cont.

$$\dot{x} = Ax + \mathrm{diag}(\theta)x$$
$$\dot{\theta} = -x \circ \mathcal{L}x$$

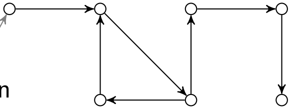
Theorem

For the dynamics above with the adaptation occurring over a connected graph $\mathcal{G}$ such that a root can be chosen in $\mathcal{G}^{\mathrm{SCC}}$ that is a condensed node, then $\lim_{t\to\infty} x(t) = \mathbf{0}$

- The root is a strongly connected subgraph (thus stabilizes itself)
- All information flowing over $\mathcal{G}$ decimates from a stable SCC.
- Stability of each SCC then follows from the hierarchical structure of the DAG.



Stabilization over Connected Graphs: Example of Necessity

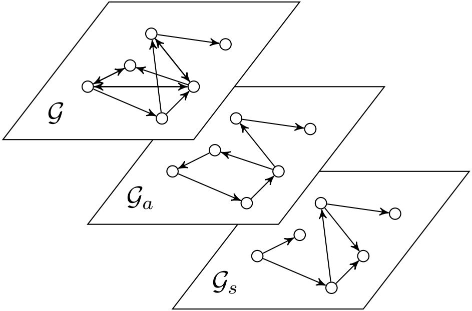


This node can
never stabilize if
initially unstable

Consensus and Leaning

Bring everything together as a layered architecture

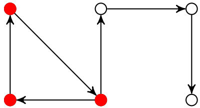
- The communication graph is $\mathcal{G}$
- $\mathcal{G}_a$ is the adaptation graph and is constrained by the communication in $\mathcal{G}$
- $\mathcal{G}_s$ is the synchronization graph and is similarly constrained

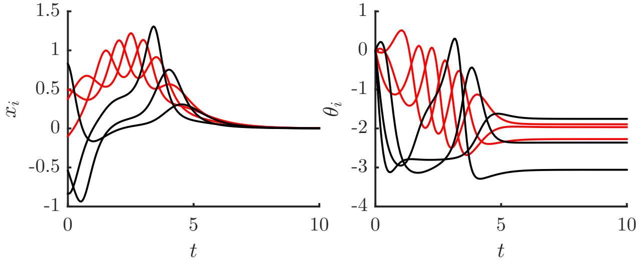


(Doyle and Csete, 2011), (Alderson and Doyle, 2010)

Adaptive Stabilization over a Network

$$\Sigma: \quad \dot{x} = Ax + \mathrm{diag}(\theta)x$$
$$\dot{\theta} = -x \circ \mathcal{L}_a x$$

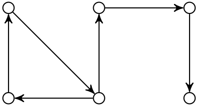
$\mathcal{G}_a =$ 

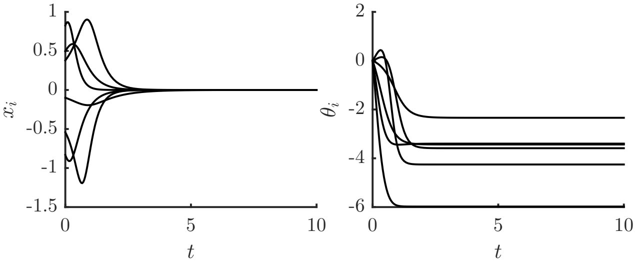


Adaptive Stabilization and Desynchronous Input

$$\Sigma: \quad \dot{x} = Ax + \mathcal{L}_s x + \mathrm{diag}(\theta)x$$
$$\dot{\theta} = -\Gamma x \circ \mathcal{L}_a x$$

$\mathcal{G}_a = \mathcal{G}_s =$





Summary

Borrowing from **Narendra**, **Murray**, and **My Thesis**, we have

- Found that synchronization can hurt learning.
- As always context is important
- What about other learning paradigms, i.e. Jadbabaie's work or the broader Machine Learning literature

Bibliography

Alderson, D. L and J. C Doyle. 2010. *Contrasting views of complexity and their implications for network-centric infrastructures*, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **40**, no. 4, 839–852.

Dominguez-Garcia, A. D. and C. N. Hadjicostis. 2013. *Distributed matrix scaling and application to average consensus in directed graphs*, Automatic Control, IEEE Transactions on **58**, no. 3, 667–681.

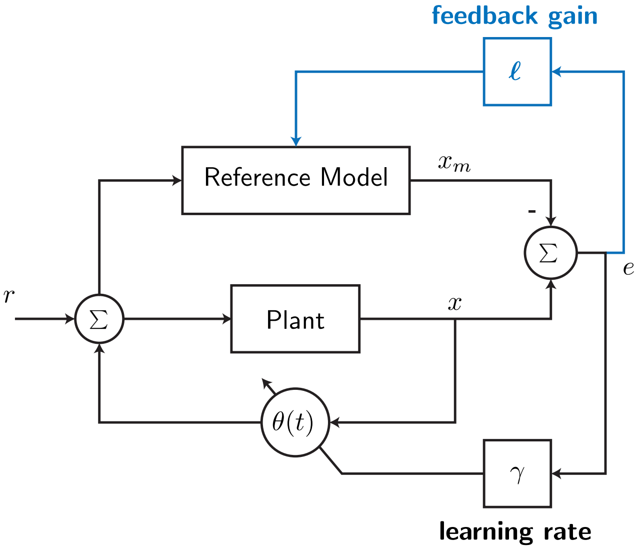
Doyle, J. C. and M. Csete. 2011. *Architecture, constraints, and behavior*, Proceedings of the National Academy of Sciences **108**, no. Supplement 3, 15624–15630.

Marshall, A. W and I. Olkin. 1968. *Scaling of matrices to achieve specified row and column sums*, Numerische Mathematik **12**, no. 1, 83–90.

Narendra, K. S. and P. Harshangi. 2014. *Unstable systems stabilizing each other through adaptation*, American Control Conference, pp. 7–12.

Olfati-Saber, R. and R. M Murray. 2004. *Consensus problems in networks of agents with switching topology and time-delays*, Automatic Control, IEEE Transactions on **49**, no. 9, 1520–1533.

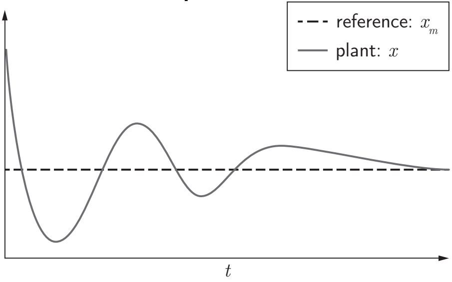
Closed-loop **R**eference **M**odel (CRM)



How does CRM Help?

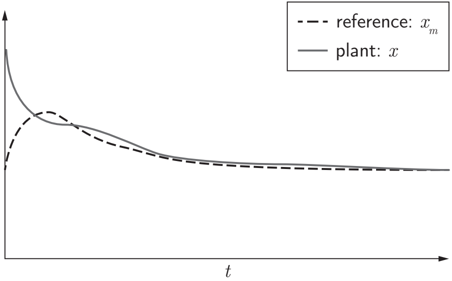
Classic **Open-loop Reference Model** (ORM) Adaptive ($\ell = 0$)

- The reference model does not adjust to any outside factors



Closed-loop Reference Model (CRM) Adaptive

- The reference model adjusts to rapidly reduce the model following error $e = x - x_m$

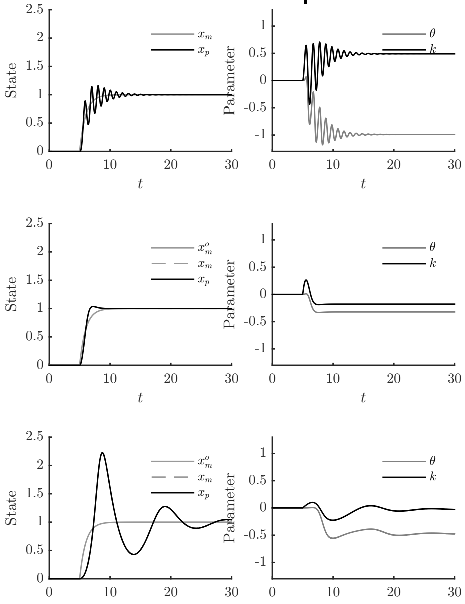


CRM Simulation Examples

$\gamma = 10$
$\ell = 0$

$\gamma = 100$
$\ell = -100$

$\gamma = 100$
$\ell = -1000$



How do you choose γ and ℓ?