Submitted to CDC 2016

Adaptation and Synchronization over a Network: Stabilization Without a Reference Model

Travis E. Gibson

***Abstract*—Fundamental properties such as learning and consensus have been studied both in the adaptive control and network control literature. The use of an error feedback is essential for the realization of both properties. In adaptive control, error feedback is used to update adaptive parameters in an effort to accomplish learning and tracking. In network control, error feedback is used to achieve consensus. The two types of error feedback are seldom studied in concert without a pinning trajectory. This paper explores the implications of concomitantly achieving consensus and learning in adaptive and networked systems. Conditions under which synchronous inputs can enhance adaptation and learning are analyzed. The tradeoff between synchronization and learning is explored both in the context of two interacting dynamical systems and a network of dynamical systems interacting over a graph.**

I. Introduction

Error feedback between dynamical systems, often referred to as agents, is the central dogma studied in synchronization, flocking, and consensus [9, 12, 22, 27, 32, 38–40, 43, 46–48]. Error feedback is also the central theme in adaptive control when it comes to updating the time-varying parameters [2,20,34,44]. In this work we wish to illustrate how synchronization impacts learning in adaptive control. Towards that end we will not introduce a reference model or a pinning node (trajectory). This allows us to directly illustrate the impact of synchronization on learning. Let us now give some motivation for why this is an important trade-off to understand.

Adaptive systems inherently have the undesired trade-off that higher learning rates, while resulting in better reference model tracking, generate high frequency oscillations in the adaptive parameter, which then propagate through to the control input. So as to address this a class of *Closed-loop Reference Model* (CRM) adaptive controllers have been recently studied where by the model following error is fed back into the reference model, see [28] and [15, and references within]. In CRM adaptive controllers the error fed back in the reference model acts to guide the reference model toward the plant, alleviating some of the burden of adaptation, which in turn leads to smoother transients. Under this paradigm the reference model meets the plant "half-way" and the reference model is more gentle in its demands. It was found that while this method can be used to improve the transient performance their is still a trade off to be made. If the gain used to feedback the error into the reference model[1] was too large in comparison to the learning gain[2] then undesirable phenomenon such as peaking can occur [14]. The fundamental trade off can be expressed as follows, when the error which is fed back into the reference model (consensus protocol) is dominating, then it is not possible for learning to occur. That is, the adaptive controller needs the errors to be present in the system to learn. Introducing a consensus protocol, while improving the transients, can retard the learning [24]. We note that similar phenomenon can occur when consensus and adaptation are studied in a network context. We now give a brief literature review of related works in the area of networks, adaptation, and consensus.

Decentralized Adaptive Control (DAC) was first studied in the 80's [13, 21, 23, 45]. In the DAC setting each node has its own reference model and adaptation is carried out in the presence of unknown neighbors. No notion of synchronization is addressed in this body of work. Some of the first work to analyze *Distributed Adaptive Control with Synchronization* (DACS) can be found in [4, 54]. In the DACS paradigm adaptation is incorporated so as to overcome uncertainty in the local dynamics while a linear non-adaptive synchronization input is given to each agent. In the specific DACS strategies just referenced it is worth noting that a pinning trajectory is used as a reference.

Moving the adaptation from the local controller to the synchronization input *Distributed Adaptive Synchronization* (DAS) algorithms have been proposed [6, 29, 55–58]. In the DAS structure a weighted laplacian is used in the synchronization protocol where the weights in the laplacian are updated adaptively. In [5] DAS and DACS are combined, resulting in *Distributed Adaptive Control with Adaptive Synchronization* (DACAS). Note that in DACAS a pinning strategy is deployed as well. For a recent review of adaptive pinning control see [49]. We note that a related area of research where parameter estimates amongst agents are shared so as to learn has also recently been explored [17, 37, 41]. Our work is most closely related to DACS, however we **do not** have a pinning trajectory in the construction of the error signals. The trade off between synchronization and learning is amplified when a pinning trajectory is not present.

The contribution of this work is thus an illustration of the trade off between learning through a technical exposition of

T. E. Gibson is with Harvard Medical School and the Channing Division of Network Medicine, Department of Medicine, Brigham and Women's Hospital, Boston MA, 02115, email: (travis.gibson@channing.harvard.edu).
Some of the results here were also discussed at the 17th Yale Workshop on Adaptive and Learning Systems [16]

[1]The gain that multiplies the error before it is fed back into the reference model is like an observer gain and in the context of CRM is denoted as ℓ or L [15].

[2]Usually in the context of adaptive control this term is γ or Γ and controls the rate of adaptation [15].

distributed adaptive control under different scenarios: with a consensus protocol, without a consensus protocol, and with a de-synchronous protocol, all without pinning. The contribution is more than merely a superficial discussion as the stability analysis is different from those given in the DAS, DACS, and DACAS literature. Instead of analyzing stability with a Lyapunov function we instead analyze the adaptive gains directly and prove stability with a monotonicity argument, motivated by [36].

This paper is organized as follows. In section II two systems stabilizing each other through adaptation are studied, as in [36], but with the added component of synchronous error feedback. Section III discusses synchronization and learning over a network. Section IV closes with final comments and a disscusion regarding this work in the context of some recent work by Jadbabaie and co-authors in the area of social learning [33, 42]. Social learning was indeed the motivation put forward by Narendra and Harshangi in [36].

II. Synchronization Hurts Learning

In this section we wish to illustrate how the use of synchronous error feedback hurts learning in adaptive systems. We begin with a variation on a recent example from [36] where two unstable dynamical systems stabilize each other through adaptation. The example begins with the two dynamical states $x_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $x_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}$

$$\begin{aligned} \Sigma_1 : \quad & \dot{x}_1(t) = a_1(t)x_1(t) + u_1(t), \\ \Sigma_2 : \quad & \dot{x}_2(t) = a_2(t)x_2(t) + u_2(t), \end{aligned} \tag{1}$$

where the parameters $a_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $a_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}$ are adjusted adaptively according to

$$\begin{aligned} \dot{a}_1(t) &= -x_1(t)e(t), \quad a_1(0) > 0, \\ \dot{a}_2(t) &= x_2(t)e(t), \quad a_2(0) > 0, \end{aligned} \tag{2}$$

with $e = x_1 - x_2$. Before discussing the stability properties of the above proposed system consider the following simulation scenarios:

1. No input, $u_1 = u_2 = 0$ [36, §II.B].
2. Synchronizing input $u_1 = -e$, $u_2 = e$.
3. Desynchronizing input $u_1 = e$, $u_2 = -e$.

Simulation results for the above three scenarios are shown in Figures 1-3 respectively. For each of the three scenarios $a_1(0) \neq a_2(0)$. Figure 1 illustrates the two unstable systems stabilizing each other through adaptation without any other external inputs into the dynamics, just as in [36, §II.B]. When synchronous inputs are used the systems Σ_1 and Σ_2 are unstable with x_1 and x_2 growing without bound while a_1 and a_2 converge to the same positive constant, see Figure 2. Thus we begin to see how synchronization can degrade learning in an adaptive system. When a desynchronizing input is used as in Figure 3 the transients in both the state and adaptive parameters is smoother as compared to either Scenario 1 or 2.

This result seems counterintuitive because synchronization is often thought to improve transients. As eluded to earlier, in CRM adaptive control, with an appropriately tuned learning rate adaptive systems can be shown to have smooth transients. In contrast to Σ_1 and Σ_2, in CRM adaptive control one of the systems is a-priori stable, *the reference model*. When both dynamic systems are initially unstable, as in this section, it is imperative that the adaptation occurs quickly, and therefore synchronization, by slowing down the adaptation, has a destabilizing effect on Σ_1 and Σ_2. More details on this slowing of convergence because of synchronization can be found in [24–26]. A short discussion on the stability analysis of these scenarios now follows.

A. Discussion on Stability

For Scenario 1, so long as $a_1(0) \neq a_2(0)$ it follows that e asymptotically converges to zero and $\lim_{t\to\infty} a_1(t) < 0$ and $\lim_{t\to\infty} a_2(t) < 0$ [36, §II.B]. Note that a_1 and a_2 need not have the same limit. It was also shown in [36] that as the difference between the initial conditions $a_1(0)$ and $a_2(0)$ is reduced the frequency of the oscillations in the trajectories of a_1 and a_2 increases.

We now give a brief analysis of the stability properties of Scenario 2. Consider the following Lyapunov candidate $V(e, a_1, a_2) = e^2 + \tilde{a}_1^2 + \tilde{a}_2^2$ where $\tilde{a}_2 = a_1 - a^*$ and $\tilde{a}_2 = a_2 - a^*$ for a constant $a^* \in \mathbb{R}$. The error between the plant dynamics in (1) with the input as in Scenario 2 can be written as $\dot{e} = -2e + a^*e + \tilde{a}_1 x_1 - \tilde{a}_2 x_2$. Differentiating V along the error dynamics in the previous sentence and the

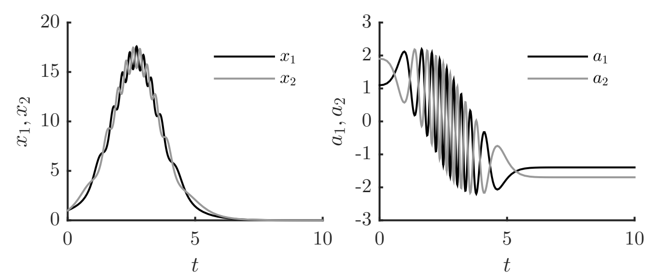


Fig. 1. Self stabilizing systems Scenario 1: $u_1 = u_2 = 0$.

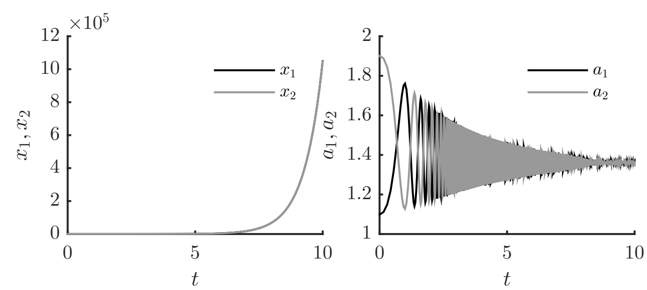


Fig. 2. Self stabilizing systems Scenario 2: $u1 = -e$, $u_2 = e$.

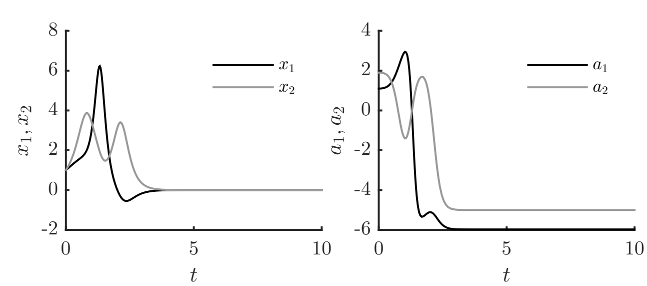


Fig. 3. Self stabilizing systems Scenario 3: $u1 = e$, $u_2 = -e$.

update law in (2) we have that $\dot{V} = (-4 + 2a^*)e^2$. Thus for Scenario 2 it is possible that $e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ while $a^* > 0$. Furthermore, if $\lim_{t\to\infty} a_1(t) = \lim_{t\to\infty} a_2(t)$ then it follows that $\lim_{t\to\infty} \dot{e}(t)$ is bounded and thus with Barbalăt Lemma [3] it follows that $e(t)$ can converge to zero while the dynamics Σ_1 and Σ_2 can be unstable.

A formal treatment is now given for the stability analysis of Scenario 3.

Theorem 1. *With the initial conditions of the adaptive parameters such that $a_1(0) \neq a_2(0)$, and with the inputs u_1, u_2 chosen as in Scenario 3, the dynamics in* (1) *and* (2) *are uniformly stable and $e(t)$ asymptotically converges to zero.*

Proof: First note that

$$a_1(t) + a_2(t) = -\int_0^t e^2(\tau)\,\mathrm{d}\tau + a_1(0) + a_2(0) \quad (3)$$

is a non-increasing function. Therefore, $a_1(t) + a_2(t)$ decreases until $e(t)$ remains zero. Thus it is only necessary to study the above dynamics under the assumption $e = 0$. First consider the case that $a_1 = a_2 = \alpha$. Under these conditions it follows that $\dot{e} = (2 + \alpha)e$, which is unstable for $\alpha > -2$, and therefore $e(t)$ will remain non-zero until $\alpha = 2$. By monotonicity in the expression in (3) it follows that if $\lim_{t\to\infty} a_1 = \lim_{t\to\infty} a_2$ it must be such that $\lim_{t\to\infty} a_1 = \lim_{t\to\infty} a_2 \leq -2$.

If $a_1 \neq a_2$ and $e = 0$, then it follows that

$$\begin{aligned} \dot{x}_1 &= a_1 x_1 \\ \dot{x}_2 &= a_2 x_2. \end{aligned} \quad (4)$$

If a_1 and a_2 are positive, then e can not remain zero and thus by monotonicity in (3) the expression $a_1 + a_2$ will decrease until either a_1 or a_2 is negative. Without loss of generality assume that a_1 is negative and a_2 is positive. Thus for the dynamics in (4) x_1 is stable while x_2 is unstable and thus e can not remain zero. Therefore it follows that $a_1 + a_2$ decreases until both a_1 and a_2 are negative. The proof is complete, but it is worth mentioning that with the Lyapunov function $V(e, a_1, a_2) = e^2 + \tilde{a}_1^2 + \tilde{a}_2^2$ whose time derivative is $\dot{V} = (4 + 2a^*)e^2$, it follows that this method, when combined with a persistence of excitation condition, will give the same upperbound on $\lim_{t\to\infty} a_1, \lim_{t\to\infty} a_2$ as just previously mentioned, $\lim_{t\to\infty} a_1 = \lim_{t\to\infty} a_2 \leq -2$. Thus $e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, and $x_1, x_2, \tilde{a}_1, \tilde{a}_2 \in \mathcal{L}_\infty$ and thus $\dot{e} \in \mathcal{L}_\infty$. Thus $e(t)$ converges to zero. ■

Remark 1. As previously mentioned in [36, §IV.D] the analysis of multidimensional adaptively stabilizing dynamics in companion form is rather tricky. It is not clear how to choose the SPR direction for updating the adaptive parameters. The use of synchronous or desynchronous inputs may be useful in overcoming this hurdle.

III. Synchronization and Learning Over Networks

The previous two sections have illustrated that the trade offs between synchronization and adaptation are non-trivial and can even lead to initially counter-intuitive results. The purpose of this section is to study these two concepts in a multi-agent context. Before that can be done, we first review necessary tools and nomenclature from synchronization of linear systems [40]. Adaptive stabilization over networks is then addressed. Finally the two ideas are combined and discussed as a hierarchical architecture.

A. Synchronization Over Networks: a Review

We begin with n scalar pure integrator dynamical agents,

$$\Sigma_i: \quad \dot{x}_i(t) = u_i(t). \quad (5)$$

The u_i are to be defined shortly. First we must define the graph over which the agents can communicate. Let $\mathcal{G}$ denote a directed graph consisting of n vertices with edges going between the vertices if and only if communication can travel between the agents at the vertices. Formally the digraph is defined by the double $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the vertex set and the directed edges are defined by the ordered pairs $(v_i, v_j) \in \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An element $(v_i, v_j) \in \mathcal{E}$ if and only if there is a directed edge from vertex v_i to vertex v_j. A useful algebraic component when discussing graphs is the adjacency matrix $\mathcal{A}(\mathcal{G})$, whose components are defined as follows $[\mathcal{A}]_{ij} = 1$ if $(v_j, v_i) \in \mathcal{E}$ and $[\mathcal{A}]_{ij} = 0$, otherwise. In this work *all* graphs are assumed to be directed graphs and thus the term graph can be used without ambiguity and the word digraph is only used for emphasis.

A common synchronizing input studied in the literature is the following

$$u_i = -\sum_{j\in\mathcal{N}(i)} (x_i - x_j) \quad (6)$$

where $\mathcal{N}(i)$ denotes the neighbors of agent i with in-links pointing toward i. Letting $x = [x_1,\ x_2,\ \ldots,\ x_n]^\mathsf{T}$ denote the state vector the dynamics in (5) with the controller in (6) can be compactly represented by the equation

$$\Sigma: \quad \dot{x} = -\mathcal{L}x \quad (7)$$

where

$$\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G}) \quad (8)$$

is the in-degree *laplacian* of $\mathcal{G}$, and $\mathcal{D}(\mathcal{G})$ a diagonal matrix with each $[\mathcal{D}]_{ii}$ equal to the in-degree of node i.

Before continuing a brief discussion regarding the spectral properties of graph laplacians is in order. A quick notational comment, throughout this section $\mathbf{1} \triangleq [1,\ 1,\ \ldots,\ 1]^\mathsf{T}$ and similarly $\mathbf{0} \triangleq [0,\ 0,\ \ldots,\ 0]^\mathsf{T}$. In the case that $\mathcal{G}$ is a *connected* (i.e. no isolated nodes) undirected graph the corresponding laplacian $\mathcal{L}$ is symmetric and the smallest eigenvalue is zero with algebraic multiplicity one. This implies that $\lambda_1(\mathcal{L}) = 0 < \lambda_2(\mathcal{L}) \leq \lambda_3(\mathcal{L}) \leq \cdots \leq \lambda_n(\mathcal{L})$. A similar result holds for directed graphs that are *strongly connected* (there is a walk between any two nodes following the edges in the directed graph). If $\mathcal{G}$ is strongly connected then $\mathcal{L}$ is rank $n-1$. By construction $\mathcal{L}\mathbf{1} = \mathbf{0}$. Given that the zero eigenvalue has algebraic multiplicity one, there can be only one right eigenvector associated with that eigenvalue.

Thus, $\mathbf{1}$ is the right eigenvector for the zero eigenvalue. Furthermore, all the other $n-1$ eigenvalues of $\mathcal{L}$ for a directed strongly connected graph have positive real parts [40, Theorem 1 and 2]. When each node of a digraph has the same in-degree and out-degree, then the digraph is denoted as *balanced*. Balanced digraphs satisfy the following property.

Lemma 1 ([40, Theorem 7])**.** *If $\mathcal{G}$ is a balanced strongly connected digraph then the symmetric component of the laplacian is possitive semi-definite, i.e. $\mathcal{L}+\mathcal{L}^{\mathsf{T}} \succeq 0$ with the smallest eigenvalue $\lambda_1(\mathcal{L}+\mathcal{L}^{\mathsf{T}}) = 0$ of algebraic multiplicty one.*

We are now ready to present a classic result regarding consensus of linear pure integrator dynamics.

Theorem 2 ([40, Corollary 1 and Theorem 4])**.** *For the dynamics in* (7) *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.*

B. Adaptive Stabilization over Networks

For this problem consider the following dynamics

$$\Sigma_i: \quad \dot{x}_i(t) = a_i x_i(t) + \theta_i(t) x_i(t) \tag{9}$$

with the update law

$$\dot{\theta}_i = -x_i \sum_{j\in\mathcal{N}(i)} (x_i - x_j). \tag{10}$$

Letting $\theta = [\theta_1,\ \theta_2,\ \ldots,\ \theta_n]^{\mathsf{T}}$ denote the parameter vector and $A \triangleq \mathrm{diag}([a_1,\ a_2,\ \ldots,\ a_n]^{\mathsf{T}})$, the dynamics in (9) and update law in (10) can be compactly represented by the two equations

$$\Sigma: \quad \dot{x} = Ax + \mathsf{diag}(\theta)x \tag{11}$$

$$\dot{\theta} = -x \circ \mathcal{L}x \tag{12}$$

Lemma 2. *For the dynamics in* (11) *and* (12) *with $\mathcal{G}$ a balanced strongly connected graph, and all the $a_i + \theta_i(0)$ not identical, it follows that $\lim_{t\to\infty} x(t) = \mathbf{0}$.*

Proof: From (11) it follows that

$$\begin{aligned}\sum_{i=1}^{n} \theta_i(t) &= -\int_0^t x^{\mathsf{T}}\mathcal{L}x \,\mathrm{d}t + \sum_{i=1}^{n}\theta_i(0) \\ &= -\frac{1}{2}\int_0^t x^{\mathsf{T}}(\mathcal{L}+\mathcal{L}^{\mathsf{T}})x \,\mathrm{d}t + \sum_{i=1}^{n}\theta_i(0).\end{aligned}$$

From the fact that $\mathcal{G}$ is balanced and strongly connected it follows that $\kappa \triangleq \lambda_2(\mathcal{L}+\mathcal{L}^{\mathsf{T}})/2 > 0$ (Lemma 1). Thus $\sum_i \theta_i(t) \le -\kappa \int x^{\mathsf{T}}x\,\mathrm{d}t + \sum_i \theta_i(0)$ when $x_i \ne x_j$ for any $i, j \in \{1,\ 2,\ \ldots,\ n\}$. Thus $\sum_i \theta_i(t)$ is a non-increasing monotonic function which will continue to decrease untill all states are equal and furthermore all $a_i + \theta_i(t) < 0$. ■

The above theorem relied heavily on the notion that the graph was balanced. The fundamental reason that balanced graphs are so important in the larger body of literature surrounding consensus is the fact that the symmetric component of the laplacian is negative semidefinite. This makes them readily suitable for the analysis of consensus over switching topologies. If for example the dynamics in (11) and (12) were switching between a finite set of k graphs so that $\mathcal{G}(t) \in \{\mathcal{G}_1,\ \mathcal{G}_2,\ \ldots,\mathcal{G}_k\}$ and that each $\mathcal{G}_i$ is strongly connected and balanced, then there exists a $\kappa > 0$ such that

$$\begin{aligned}x^{\mathsf{T}}(\mathcal{L}(\mathcal{G}(t)))x &= \frac{1}{2}x^{\mathsf{T}}(\mathcal{L}(t)+\mathcal{L}^{\mathsf{T}}(t))x \\ &\ge \kappa x^{\mathsf{T}}x\end{aligned}$$

when $x \notin \mathrm{span}(\mathbf{1})$. The above discussion allows one to prove the following theorem regarding switching topologies.

Lemma 3. *For the dynamics in* (11) *and* (12) *with $\mathcal{G}$ a time varying balanced strongly connected graph switching between a finite collection of possible graphs, and all the $a_i + \theta_i(0)$ not identical, it follows that $\lim_{t\to\infty} x(t) = \mathbf{0}$.*

The assumption that the underlying graphs are balanced is rather restrictive. It just so happens that any strongly connected graph can be balanced through a weighting of the links however [7, 8, 19, 30, 31, 51–53]. While the existence of a balanced realization for every strongly connected graph could be used to give a more general solution to the average consensus problem, one is still left with the burden of finding that weighted digraph realization. In the following theorem we will show that for the adaptive stabilization problem the graph need not be balanced. The result will depend on the following property for a class of matrices called Metzler matrices.

Definition 1. An $n \times n$ real square matrix is *Metzler* if all the off diagonal elements are non-negative.

Theorem 3 ([18, Proposition 3.1])**.** *If $A \in \mathbb{R}^{n\times n}$ is: irreducible,* semi-stable *(all eigenvalues in the closed left-half plane), Metzler, and $[A]_{ii} \le 0$, then there exists a diagonal positive matrix D such that $A^{\mathsf{T}}D + DA \preceq 0$.*

Theorem 4. *For the dynamics in* (11) *and* (12)*, with $\mathcal{G}$ a strongly connected digraph, and all the $a_i + \theta_i(0)$ not identical it follows that $\lim_{t\to\infty} x(t) = \mathbf{0}$.*

Proof: The graph is strongly connected and thus $\mathcal{L}$ has all its eigenvalues in the closed right-half plane and is irredducible. Also, $-\mathcal{L}$ is Metzler with negative diagonal elements. Therefore, by applying Theorem 3 there exists a positive diagonal matrix D such that $-\mathcal{L}^{\mathsf{T}}D - D\mathcal{L} \preceq 0$. The weighted sum of the adaptive parameters satisfies the following

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

Given that $\mathcal{L}$ is strongly connected it follows that $\mathcal{L}$ is rank $n-1$ and by definition $\mathcal{L}\mathbf{1} = \mathbf{0}$. From the fact that D is full rank it follows that the product $D\mathcal{L}$ is rank $n-1$. Therefore,

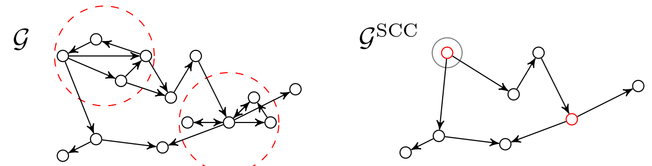


Fig. 4. Condensation of $\mathcal{G}$ to $\mathcal{G}^{\mathrm{SCC}}$ with the condensed SCCs shown in red and the root circled in gray.

$\mathbf{1}^\mathsf{T}(D\mathcal{L} + \mathcal{L}^\mathsf{T} D)\mathbf{1} = 0$. Therefore there exists $\kappa \triangleq \lambda_2(D\mathcal{L} + \mathcal{L}^\mathsf{T} D)/2 > 0$ and thus

$$\sum_i [D]_{ii}\theta_i(t) \leq -\kappa \int x^\mathsf{T} x \,\mathrm{d}t + \sum_i [D]_{ii}\theta_i(0)$$

when $x \notin \mathrm{span}(\mathbf{1})$. ∎

Remark 2. The D in the above theorem can actually be chosen to be an *output balancing of the graph*, that is it can be chosen such that $\mathbf{1}^\mathsf{T} D\mathcal{L} = \mathbf{0}^\mathsf{T}$ in addition to simply being such that $-\mathcal{L}^\mathsf{T} D - D\mathcal{L} \preceq 0$ [51–53].

The same technique used in the analysis of strongly connected but not necessarily balanced digraphs can be used to analyze tuning laws of the form

$$\dot{\theta} = -\Gamma x \circ \mathcal{L} x$$

where Γ is a diagonal positive matrix of free design parameters that control the adaptation rate. Stability of the underlying adaptive system then follows by showing that $\sum_i [\Gamma^{-1}]_{ii}[D]_{ii}\theta_i(t)$ is a non-increasing function. So far we have only discussed the stabilizability of strongly connected graphs. We have shown that if a graph is strongly connected then after a finite time t_1 all the state jacobians become stable, i.e. $a_i + \theta_i(t) < 0$ for all $t \geq t_1$. Next we will exploit this fact to prove stabilizability of a broader class of networks. Before moving on we need a few more definitions.

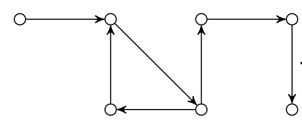
A subgraph $\mathcal{G}_1 \subseteq \mathcal{G}$ is *maximal* with respect to some property $P(\mathcal{G}_1)$ so long as there does not exist a proper supergraph $\mathcal{G}_2$ of $\mathcal{G}_1$ such that $\mathcal{G}_1 \subset \mathcal{G}_2 \subseteq \mathcal{G}$ and $P(\mathcal{G}_2)$ holds as well. Any connected digraph can be partitioned into disjoint subsets called *Strongly Connected Components* (SCCs) where each subset is a maximal strongly connected subgraph. A graph is called *acyclic* if for any directed path from node v_i to node v_j there is no directed path from v_j to v_i. The *condensation* of an unweighted digraph $\mathcal{G}$ is an unweighted digraph $\mathcal{G}^{\mathrm{SCC}}$ where each node in $\mathcal{G}^{\mathrm{SCC}}$ corresponds to an SCC in $\mathcal{G}$. A node in $\mathcal{G}^{\mathrm{SCC}}$ that corresponds to an SCC in $\mathcal{G}$ with more than one node is referred to as a *condensed* node in $\mathcal{G}^{\mathrm{SCC}}$. If there is a directed edge between two SCCs in $\mathcal{G}$ then there is also a directed edge between the corresponding nodes in $\mathcal{G}^{\mathrm{SCC}}$. For any connected $\mathcal{G}$ the corresponding $\mathcal{G}^{\mathrm{SCC}}$ is a *Directed Acyclic Graph* (DAG). If $\mathcal{G}$ is a connected DAG then there exists a *root* node $r \in \mathcal{V}$, not necessarily unique, such that there is a directed path from r to any $v \in \mathcal{V}$. This condensation and labeling of the corresponding root in the resulting DAG is shown in Figure 4.

Assumption 1. The graph $\mathcal{G}$ is connected and a root can be chosen in $\mathcal{G}^{\mathrm{SCC}}$ that is a condensed node. Furthermore, for the nodes in $\mathcal{G}$ associated with the condensed SCC that is the root, all of the $a_i + \theta_i(0)$ are not equal.

Theorem 5. *For the dynamics in* (11) *and* (12) *with the adaptation occurring over a graph satisfying Assumption 1 it follows that* $\lim_{t\to\infty} x(t) = \mathbf{0}$

Proof: Let the SCC corresponding to the root in $\mathcal{G}^{\mathrm{SCC}}$ be denoted as $\mathcal{G}'$. Given that $\mathcal{G}'$ has no incoming links it follows that the stability of nodes in $\mathcal{G}'$ can be analyzed independently from the rest of the graph. By definition $\mathcal{G}'$ is strongly connected. Therefore by applying Theorem 4 it follows that all the state trajectories associated with the nodes in $\mathcal{G}'$ are stable. Thus all information flowing over $\mathcal{G}$ decimates from a stable SCC. Thus stability of each SCC then follows from the hierarchical structure of the DAG determined by $\mathcal{G}^{\mathrm{SCC}}$. A more formal treatment of the stability analysis will be given elsewhere. ∎

The graph in Figure 4 satisfies the conditions of the above theorem. This is an example of a graph that is not stabilizable



While one of the SCCs does contain three nodes, when condensed in the condensation graph it can not be the root. Thus far we have studied synchronization and adaptation over networks separately. In the next section we study them jointly.

C. Layered Architectures: Adaptation and Synchronization

In this section we will take the point of view that complex networks should be studied as layered architectures [1, 11]. Thus we will consider the problem of synchronization and learning in a layered fashion as shown in Figure 5. The top layer is the graph $\mathcal{G}$ which shows all possible communication directions over the network. The second layer shows how information will flow in updating the adaptive parameters. The next layer denotes how state information will flow in performing error feedback into the state dynamics. We will refer to the second layer as the adaptation graph and the last layer as the synchronization graph.

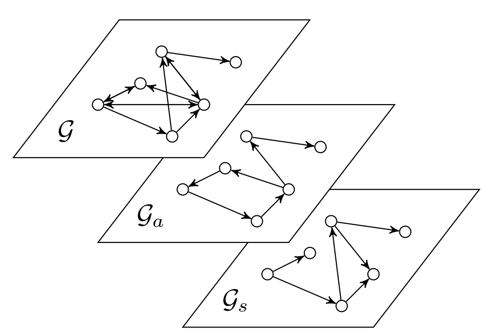


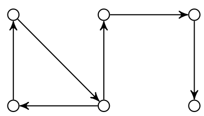
Fig. 5. Synchronization and learning as a layered architecture.

With this problem posed as a layered architecture the possible research directions far outnumber our current understanding of what would be practical to study. Based on the insight gained in §II we will give simulation results for the following adaptive *desynchronous* stabilizing dynamics

$$\Sigma: \quad \dot{x} = Ax + \mathcal{L}_s x + \mathsf{diag}(\theta)x \tag{13}$$
$$\dot{\theta} = -\Gamma x \circ \mathcal{L}_a x \tag{14}$$

where A is a diagonal matrix as defined just after (10), Γ is a diagonal and positive free design parameter, and $\mathcal{L}_s$ and $\mathcal{L}_a$ are the laplacians of $\mathcal{G}_s$ and $\mathcal{G}_a$ respectively. The plus sign in front of the term $\mathcal{L}_s x$ in (13) lets us know that the error feedback is non-synchronizing. The dynamics were then simulated for the case that $\mathcal{G}_s = \mathcal{G}_a = \mathcal{G}$ where

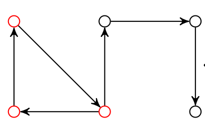
$$\mathcal{G}_a = \mathcal{G}_s = \tag{15}$$



and for the case when $\mathcal{G}_s = \emptyset$. The results for these two scenarios are shown in Figures 6 and 7. For both scenarios

$$A = \text{diag}([1.11,\ 1.1,\ 1.2,\ 1.1,\ 1.1,\ 1.1]^\mathsf{T}),$$

$\Gamma = 10I_{6\times 6}$, the initial conditions for the states $x_i(0)$ were drawn from a normal distribution with mean zero and variance one, and the adaptive parameters were initialized to $\theta(0) = \mathbf{0}$. The trends observed in Figures 6 and 7 are similar to those observed in §II. Desynchronous inputs improve learning and remove the oscillations that can occur when adaptive stabilization is performed. The oscillations that occur in Figure 7 arise from the nodes contained in the SCC highlighted in red



$$. \tag{16}$$

In order to illustrate this point more clearly the trajectories of the states and adaptive parameters for the red nodes in (16) are highlighted in red in Figure 8 as well.

The dynamics in (13) and (14) are stable with $\lim_{t\to\infty} x(t) = \mathbf{0}$ if $\mathcal{G}_a$ satisfies the conditions of Theorem 5 and the linear error feedback in (13) is desynchronous. This type of layered architecture for synchronization and learning really is just the tip of the iceberg. Several natural extensions should and we assume will be studied. For instance we did not address even indirectly the sharing of parameter estimates, which is common in multiple model adaptive control [35] and second level learning adaptive control [17]. What is needed now are some real world examples for this direction of research.

IV. Conclusions

One would assume that the addition of a consensus protocol to an adaptive system would only improve the performance of the adaptive system. This however is not the case. It can be shown that an otherwise stable adaptive protocol can be destabilized by the very act of synchronous inputs.

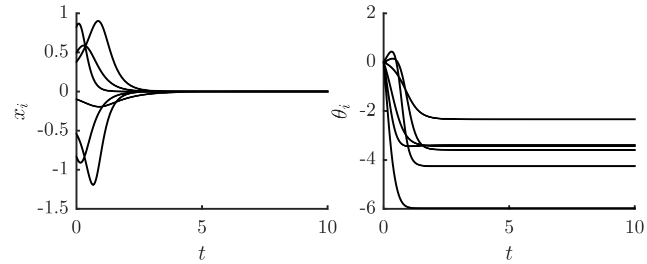


Fig. 6. Trajectories of self stabilizing systems with $\mathcal{G}_s = \mathcal{G}_a = \mathcal{G}$ (desynchronous) defined in (15).

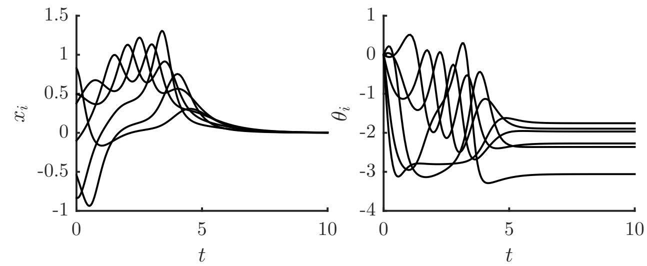


Fig. 7. Trajectories of self stabilizing systems with $\mathcal{G}_a = \mathcal{G}$ as defined in (15) and $\mathcal{G}_s = \emptyset$.

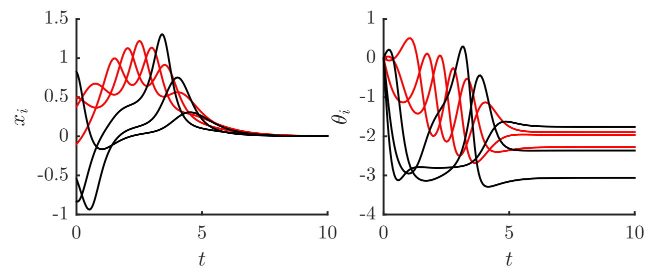


Fig. 8. Identical to Figure 7 with the states and adaptive parameters highlighted that pertain to the SCC highlighted in Equation (16).

The results of this paper can be viewed as a few fundamental properties of synchronization and learning systems, but are reminded by [10] and [50] that context is always important.

It would be interesting to combine ideas from adaptive control with the more statistically driven approaches, see for instance the following paper [42] studying bayesian learning without recall and the following paper on non-bayesian social learning [33]. Does synchronization also retard learning in other settings?

V. Acknowledgements

I would like to thank Anuradha Annaswamy for many fruitful conversations as well as Prashanth Harshangi and Kumpati Narendra. Prashanth pointed out some mistakes in a previous version of this paper.

References

[1] David L Alderson and John C Doyle, *Contrasting views of complexity and their implications for network-centric infrastructures*, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **40** (2010), no. 4, 839–852.
[2] Karl J Åström and Björn Wittenmark, *Adaptive control*, Dover, 2008.
[3] I Barbalăt, *Systemes d'équations différentielles d'oscillations non linéaires*, Rev. Math. Pures Appl **4** (1959), no. 2, 267–270.
[4] Abhijit Das and Frank L Lewis, *Distributed adaptive control for synchronization of unknown nonlinear networked systems*, Automatica **46** (2010), no. 12, 2014–2021.

[5] Pietro DeLellis, Mario di Bernardo, and Franco Garofalo, *Adaptive pinning control of networks of circuits and systems in lur'e form*, Circuits and Systems I: Regular Papers, IEEE Transactions on **60** (2013), no. 11, 3033–3042.

[6] Pietro DeLellis, Francesco Garofalo, et al., *Novel decentralized adaptive strategies for the synchronization of complex networks*, Automatica **45** (2009), no. 5, 1312–1318.

[7] DŽ Djoković, *Note on nonnegative matrices*, Proceedings of the American Mathematical Society **25** (1970), no. 1, 80–82.

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

[9] Florian Dörfler and Francesco Bullo, *Synchronization in complex networks of phase oscillators: A survey*, Automatica **50** (2014), no. 6, 1539–1564.

[10] John C. Doyle, David L. Alderson, Lun Li, Steven Low, Matthew Roughan, Stanislav Shalunov, Reiko Tanaka, and Walter Willinger, *The "robust yet fragile" nature of the internet*, Proceedings of the National Academy of Sciences **102** (2005), no. 41, 14497–14502.

[11] John C. Doyle and Marie Csete, *Architecture, constraints, and behavior*, Proceedings of the National Academy of Sciences **108** (2011), no. Supplement 3, 15624–15630.

[12] J.A. Fax and R.M. Murray, *Information flow and cooperative control of vehicle formations*, Automatic Control, IEEE Transactions on **49** (2004Sept), no. 9, 1465–1476.

[13] Donald T Gavel and Dragoslav D Šiljak, *Decentralized adaptive control: structural conditions for stability*, Automatic Control, IEEE Transactions on **34** (1989), no. 4, 413–426.

[14] T. E. Gibson, A. M. Annaswamy, and E. Lavretsky, *On Adaptive Control with Closed-loop Reference Models: Transients, Oscillations, and Peaking*, IEEE Access **1** (2013), 703–717, available at http://arxiv.org/abs/1304.7278.

[15] Travis E. Gibson, *Closed-loop reference model adaptive control: with application to very flexible aircraft*, Ph.D. Thesis, 2014.

[16] Travis E. Gibson and Anuradha M. Annaswamy, *Learning and consensus in adaptive and networked systems*, 17th yale workshop on adaptive and learning systems, 2015.

[17] Z. Han and K. S. Narendra, *New concepts in adaptive control using multiple models*, IEEE Trans. Automat. Contr. (2012).

[18] Daniel Hershkowitz and Hans Schneider, *Lyapunov diagonal semistability of real h-matrices*, Linear algebra and its applications **71** (1985), 119–149.

[19] Loh Hooi-Tong, *On a class of directed graphs—with an application to traffic-flow problems*, Operations Research **18** (1970), no. 1, 87–94.

[20] P.A. Ioannou and J. Sun, *Robust adaptive control*, Dover, 2013.

[21] Petros A Ioannou, *Decentralized adaptive control of interconnected systems*, Automatic Control, IEEE Transactions on **31** (1986), no. 4, 291–298.

[22] Ali Jadbabaie, Jie Lin, et al., *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, Automatic Control, IEEE Transactions on **48** (2003), no. 6, 988–1001.

[23] Sandeep Jain and Farshad Khorrami, *Decentralized adaptive control of a class of large-scale interconnected nonlinear systems*, Automatic Control, IEEE Transactions on **42** (1997), no. 2, 136–154.

[24] Benjamin M. Jenkins, Eugene Lavretsky, Anuradha M. Annaswamy, and Travis E. Gibson, *Convergence properties of adaptive systems and the definition of exponential stability*, available at http://arxiv.org/abs/1511.03222.

[25] B.M Jenkins, T.E. Gibson, A.M. Annaswamy, and E. Lavretsky, *Convergence Properties of Adaptive Systems with Open-and Closed-loop Reference Models*, AIAA guidance navigation and control conference, 2013.

[26] B.M Jenkins, T.E. Gibson, A.M. Annaswamy, and E. Lavretsky, *Uniform Asymptotic Stability and Slow Convergence in Adaptive Systems*, IFAC International Workshop on Adaptation and Learning in Control and Signal Processing, 2013, pp. 446–451.

[27] Yoshiki Kuramoto, *Self-entrainment of a population of coupled nonlinear oscillators*, International symposium on mathematical problems in theoretical physics, 1975, pp. 420–422.

[28] Tea-Gyoo Lee and Uk-Youl Huh, *An error feedback model based adaptive controller for nonlinear systems*, Proceedings of the ieee international symposium on industrial electronics, 1997.

[29] Zhongkui Li, Guanghui Wen, Zhisheng Duan, and Wei Ren, *Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs*, Automatic Control, IEEE Transactions on **60** (2015), no. 4, 1152–1157.

[30] Ali Makhdoumi and Asuman Ozdaglar, *Graph balancing for distributed subgradient methods over directed graphs*, Decision and control (cdc), 2015 ieee 54th annual conference on, 2015, pp. 1364–1371.

[31] Albert W Marshall and Ingram Olkin, *Scaling of matrices to achieve specified row and column sums*, Numerische Mathematik **12** (1968), no. 1, 83–90.

[32] Renato E Mirollo and Steven H Strogatz, *Synchronization of pulse-coupled biological oscillators*, SIAM Journal on Applied Mathematics **50** (1990), no. 6, 1645–1662.

[33] Pooya Molavi, Alireza Tahbaz-Salehi, and Ali Jadbabaie, *Foundations of non-bayesian social learning*, Columbia Business School Research Paper (2015).

[34] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*, Dover, 2005.

[35] K. S. Narendra and J. Balakrishnan, *Adaptive control using multiple models*, IEEE Trans. Automat. Contr. (1997).

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

[37] Kumpati S Narendra, Yu Wang, and Wei Chen, *Stability, robustness, and performance issues in second level adaptation*, American control conference, 2014, pp. 2377–2382.

[38] Reza Olfati-Saber, *Flocking for multi-agent dynamic systems: Algorithms and theory*, Automatic Control, IEEE Transactions on **51** (2006), no. 3, 401–420.

[39] Reza Olfati-Saber, J Alex Fax, and Richard M Murray, *Consensus and cooperation in networked multi-agent systems*, Proceedings of the IEEE **95** (2007), no. 1, 215–233.

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

[41] Ivan Papusha, Eugene Lavretsky, and Richard M Murray, *Collaborative system identification via parameter consensus*, American control conference, 2014, pp. 13–19.

[42] M Amin Rahimian and Ali Jadbabaie, *Bayesian learning without recall*, arXiv preprint arXiv:1601.06103 (2016).

[43] Craig W Reynolds, *Flocks, herds and schools: A distributed behavioral model*, ACM Siggraph Computer Graphics **21** (1987), no. 4, 25–34.

[44] Shankar Sastry and Marc Bodson, *Adaptive control: stability, convergence and robustness*, Dover, 2011.

[45] Jeffrey T Spooner and Kevin M Passino, *Decentralized adaptive control of nonlinear systems using radial basis neural networks*, Automatic Control, IEEE Transactions on **44** (1999), no. 11, 2050–2057.

[46] Steven H Strogatz, *From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators*, Physica D: Nonlinear Phenomena **143** (2000), no. 1, 1–20.

[47] John N Tsitsiklis, *Problems in decentralized decision making and computation*, Ph.D. Thesis, 1984.

[48] John N Tsitsiklis, Dimitri P Bertsekas, Michael Athans, et al., *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE transactions on automatic control **31** (1986), no. 9, 803–812.

[49] LFR Turci, P De Lellis, EEN Macau, M Di Bernardo, and MMR Simões, *Adaptive pinning control: A review of the fully decentralized strategy and its extensions*, The European Physical Journal Special Topics **223** (2014), no. 13, 2649–2664.

[50] Walter Willinger, David Alderson, and John C Doyle, *Mathematics and the internet: A source of enormous confusion and great potential*, Notices of the AMS **56** (2009), no. 5.

[51] Chai Wah Wu, *Algebraic connectivity of directed graphs*, Linear and Multilinear Algebra **53** (2005), no. 3, 203–223.

[52] Chai Wah Wu, *On rayleigh–ritz ratios of a generalized laplacian matrix of directed graphs*, Linear algebra and its applications **402** (2005), 207–227.

[53] Chai Wah Wu, *Synchronization in networks of nonlinear dynamical systems coupled via a directed graph*, Nonlinearity **18** (2005), no. 3, 1057.

[54] Hui Yu and Xiaohua Xia, *Adaptive consensus of multi-agents in networks with jointly connected topologies*, Automatica **48** (2012), no. 8, 1783–1790.

[55] W. Yu, J. Lü, X. Yu, and G. Chen, *Distributed adaptive control for synchronization in directed complex networks*, SIAM Journal on Control and Optimization **53** (2015/01/01), no. 5, 2980–3005.

[56] Wenwu Yu, Pietro DeLellis, Guanrong Chen, Mario Di Bernardo,

and Jürgen Kurths, *Distributed adaptive control of synchronization in complex networks*, Automatic Control, IEEE Transactions on **57** (2012), no. 8, 2153–2158.

[57] Jin Zhou, Jun-An Lu, and Jinhu Lü, *Adaptive synchronization of an uncertain complex dynamical network*, Automatic Control, IEEE Transactions on **51** (2006), no. 4, 652–656.

[58] Jin Zhou, Jun-an Lu, and Jinhu Lü, *Pinning adaptive synchronization of a general complex dynamical network*, Automatica **44** (2008), no. 4, 996–1003.