To appear, 2005 IFAC World Congress
http://www.cds.caltech.edu/~murray/papers/2005f_som05-ifac.html

DYNAMIC CONSENSUS FOR MOBILE NETWORKS

Demetri P. Spanos * **Reza Olfati-Saber** ** **Richard M. Murray** *

* *California Institute of Technology*
** *University of California, Los Angeles*

Abstract: This work examines several dynamical aspects of average consensus in mobile networks. The results herein allow consensus on general time-varying signals, and allow tracking analysis using standard frequency-domain techniques. Further, the frequency-domain analysis naturally inspires a robust small-gain version of the algorithm, which tolerates arbitrary non-uniform time delays. Finally, we show how to exploit a dynamical conservation property in order to ensure consensus tracking despite splitting and merging of the underlying mobile network.

Keywords: Agents, Decentralized control, Decentralized systems, Graph theoretic models, Mobile Robots

1. INTRODUCTION

Consensus problems have attracted much attention among researchers studying distributed and decentralized automated systems. Broadly speaking, a consensus problem is one in which several spatially distributed agents or processors must reach a common output value, but without recourse to a central coordinator or global communication. This work focuses on dynamical aspects of linear consensus, i.e. consensus in which all agents must converge to a linear combination of their individual input values.

Nearly all consensus problems studied thus far have been static; a classical example is that of the Byzantine generals, in which a single collective decision must be made to attack an enemy. However, the nature of decentralized control requires tight coordination among agents *in a dynamic environment*. Consensus on static inputs is thus insufficient, and this work shows how to achieve and analyze tracking of linear consensus on time-varying inputs.

Communication networks are intrinsically dynamic, and careful analysis must be done to ensure consensus in spite of network reconfiguration. Most of this analysis to date has focused on a single stationary network which is only dynamic in the sense that its links may occasionally fail. However, mobile networks demonstrate much richer dynamics, including splitting and merging of subnetworks, as well as strongly time-dependent delay properties. This paper exhibits a completely distributed algorithm which ensures proper consensus tracking despite splitting and merging, as well as a robust consensus algorithm that converges despite arbitrarily large non-uniform delays. A companion paper, (Spanos *et al.* 2005), addresses dynamical aspects of general multivariable consensus problems and their application in distributed Kalman filtering.

The following sections summarizes previous work on the Laplacian consensus dynamics, and introduce the notation we will utilize in the remainder of the paper. Sections 4, 5, and 6 discuss our main results for dynamic consensus, robust consensus, and consensus with splitting and merging of mobile networks. We close with a discussion of some potential applications of these results.

2. BACKGROUND

Consensus problems have been considered by many authors. A general introduction from a computer science perspective is available in the textbook by Lynch (Lynch 1997). The Control community has considered the problem of *average consensus*, in which all agents must converge to the average of their initial states. In fact, simple modifications of this problem can yield arbitrary linear functions of the inputs, but we restrict our discussion to average consensus for clarity.

Average consensus has been considered in the context of vehicle formations by (Fax and Murray 2002), and in the general framework of distributed consensus by (Olfati-Saber and Murray 2004). Both of these studies focused on a special matrix known as the *Laplacian* of a graph, which forms the basis for distributed consensus dynamics.

The Laplacian matrix of a graph G is defined in terms of the adjacency matrix A. Recall that the adjacency matrix of a graph with N nodes is an $N \times N$ matrix where the ij entry is one if the edge (i, j) is included in the graph, and zero otherwise. The Laplacian is constructed from the adjacency matrix, and the diagonal degree matrix D as follows:

$$D_{ii} = \sum_j A_{ij}$$
$$L = D - A.$$

The Laplacian consensus dynamics is given by the differential equation

$$\dot{\mathbf{x}} = -L\mathbf{x}.$$

Examining the dynamics of an individual agent, we have

$$\dot{x}_i = \sum_{j \in N_i} (x_j - x_i).$$

This dynamics is completely distributed, in that each agent need only obtain the values of its neighbors (the set N_j) in order to implement its own update.

Three things must be said regarding this dynamics. First, for an undirected graph G, the Laplacian is a symmetric positive-semidefinite matrix. This implies that the dynamics is stable, and must converge to a steady-state. Second, note that this same assumption guarantees $\sum_j L_{ij} = 0$. This implies that for any vector $\mathbf{x}$ with identical components $x_i = x_j$ for all i and j, we have $Lx = 0$. Thus, any consensus is an equilibrium. It can be shown that all equilibria correspond to a consensus, provided the graph G is connected. Finally, $\sum_j L_{ij} = 0$ also implies the following dynamical conservation property:

$$\frac{d}{dt}\left(\sum_i x_i\right) = 0.$$

The Laplacian dynamics thus drives any initial condition to a consensus, and conserves the sum of the initial states. This implies that all components of the vector x must approach the value

$$\frac{1}{N}\mathbf{1}^T\mathbf{x}(0) = \frac{1}{N}\sum_i x_i(0),$$

where $\mathbf{1}$ denotes the N-element vector of ones. Each component of this vector is equal to the average of the initial values $x_i(0)$.

Additional work utilizing the Laplacian dynamics includes that of (Xiao and Boyd 2003), which examines weighting strategies for optimizing convergence rates, as well as (Hatano and Mesbahi 2004) which studies the case of stochastic link failures. A study of consensus dynamics on completely asynchronous peer-to-peer networks is provided in (Mehyar *et al.* 2005).

There is much related work in the Control community pertaining to coordination of multi-agent systems. We refer to (Moreau 2003) and (Jadbabaie *et al.* 2002) as representative of the field, and direct the reader to references therein for additional work in this area.

3. NOTATION AND SETUP

Consider a set V of N agents, labeled by an index $i = 1, 2, \ldots, N$. Their communication is modeled by a graph $G = (V, E)$ where an edge (i, j) is in E if and only if agent i can communicate with agent j. We will only consider undirected communication structures, i.e. graphs in which $(i, j) \in E \;\Leftrightarrow\; (j, i) \in E$. We will also assume that G is connected, i.e. there is a path from any node i to any other node j. As per the previous discussion, these assumptions imply that the Laplacian matrix is a symmetric, positive semi-definite matrix, and that the nullspace of the Laplacian corresponds to consensus, i.e. span$\{\mathbf{1}\}$.

Each agent has an associated signal $z_i(t)$, which represents some quantity for which the network must reach a dynamic consensus. We will occasionally refer to the vector $\mathbf{z}(t)$, which contains the individual z_i terms as its components. Dynamic consensus is simply a situation in which all agents asymptotically track the evolution of some aggregate network quantity; an example considered in (Fax and Murray 2002) is that of finding the time-varying center of mass of a mobile vehicle network.

For the sake of simplicity, we will focus on *average consensus*, i.e. the problem of tracking the time-varying average of the z_i terms. That is, we wish each agents to track the quantity

$$\bar{z}(t) = \frac{1}{N}\sum_i z_i(t). \tag{1}$$

The results we will present can easily be extended to track general linear functions.

In addition to the input signals z_i, each agent maintains a local variable x_i, which is a time-varying estimate of the instantaneous average value $\bar{z}$. We will present dynamics for $\mathbf{x}$ which ensures tracking of $\bar{z}$. Further, we will show how to obtain this result robustly, despite arbitrary time-delays and network reconfiguration.

4. DYNAMIC CONSENSUS

Recall the Laplacian consensus dynamics with initial value $\mathbf{z}_0$:

$$\begin{aligned}\dot{\mathbf{x}} &= -L\mathbf{x}\\ \mathbf{x}(0) &= \mathbf{z}_0.\end{aligned}$$

We wish to view the vector of initial conditions as an input, in order to obtain a frequency-domain view of the consensus dynamics. Since this represents a static consensus problem, it is natural to consider a step function input, i.e. $\mathbf{Z}(s) = \frac{\mathbf{z_0}}{s}$. Note that the Laplace transform of $\mathbf{x}$ is just

$$\begin{aligned}\mathbf{X}(s) &= (sI + L)^{-1}\mathbf{z}_0\\ &= s\,(sI+L)^{-1}\left(\frac{\mathbf{z}_0}{s}\right)\\ &= s\,(sI+L)^{-1}\mathbf{Z}(s).\end{aligned}$$

Now, this equation has been derived by assuming a specific $\mathbf{Z}(s)$. However, the transfer function it suggests,

$$H_{xz} = s\,(sI+L)^{-1}$$

is a very natural mechanism for a simple dynamic consensus. To see this, we will consider the associated LTI system with input $\mathbf{z}(t)$:

$$\begin{aligned}\dot{\mathbf{x}} &= -L\mathbf{x} + \dot{\mathbf{z}}\\ \mathbf{x}(0) &= \mathbf{z}(0).\end{aligned}$$

Before proceeding with the analysis, some comments are in order regarding the structure of this dynamics. First, this is a completely distributed modification, in that the input terms added to the Laplacian dynamics are purely local. Second, the conservation property of the Laplacian dynamics *without input* implies that the above dynamics has the following conservation property:

$$\frac{d}{dt}\left(\sum_i x_i\right) = \frac{d}{dt}\left(\sum_i z_i\right).$$

Thus, the instantaneous sum of the estimate variables x_i is equal to the instantaneous sum of the input variables z_i. Intuitively, this is precisely the property we would expect in order to track a time-varying average consensus. This intuition is formalized in the following propositions.

Proposition 1. Consider the LTI system described by the MIMO transfer function

$$H_{\mathbf{xz}} = s\,(sI+L)^{-1},$$

where L is the Laplacian of a connected undirected graph. Suppose the input signal $\mathbf{Z}(s)$ has all its poles in the left half-plane, and has at most one pole at $s = 0$. Then, for all i,

$$\lim_{t\to\infty}\left(x_i(t) - \frac{1}{N}\sum_j z_j(t)\right) = 0.$$

That is, each agent tracks the dynamic consensus with zero steady-state error.

PROOF. Consider the error signal $\mathbf{e}(t)$, and its Laplace transform

$$\mathbf{E}(s) = \mathbf{X}(s) - \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{Z}(s).$$

This is the vector of deviations between the x_i estimates, and the instantaneous average of the $z_i(t)$ terms. We then have the following MIMO transfer function from $\mathbf{Z}(s)$ to $\mathbf{E}(s)$:

$$H_{\mathbf{ez}} = s\,(sI+L)^{-1} - \frac{1}{N}\mathbf{1}\mathbf{1}^T. \tag{2}$$

The Laplacian is a symmetric matrix, and so admits a spectral decomposition,

$$L = \sum_i \lambda_i P_i,$$

where the λ_i terms are real eigenvalues, and the P_i terms are orthogonal projections onto mutually orthogonal eigenspaces. It is a fact from graph theory that connectedness of G implies the following:

(1) $\lambda_1 = 0$,
(2) $P_1 = \frac{1}{N}\mathbf{1}\mathbf{1}^T$,

(3) $\lambda_i > 0$ for all $i > 1$.

Rewriting 2 with the spectral decomposition, we have

$$H_{\mathbf{ez}} = \left(\frac{1}{N}\mathbf{1}\mathbf{1}^T + \sum_{i>1} \frac{s}{s+\lambda_i} P_i \right) - \frac{1}{N}\mathbf{1}\mathbf{1}^T \quad (3)$$

$$= \sum_{i>1} \frac{s}{s+\lambda_i} P_i. \quad (4)$$

This transfer function has a single zero at $s = 0$, and all the terms in the summation are stable. Thus, for an arbitrary stable input signal $\mathbf{Z}(s)$ with at most one pole at $s = 0$, the final value theorem implies that $\mathbf{e}(t) \to 0$ as $t \to \infty$. This is the desired result. $\square$

Corollary 2. Suppose $C(s)$ is a stable transfer function with stable inverse, and k zeros at $s = 0$. Let the input $\mathbf{Z}(s)$ be a signal with no right-half-plane poles, and at most k poles at $s = 0$. Then the consensus dynamics given by the transfer function

$$H_{\mathbf{xz}} = C(s)\left(C(s)I + L\right)^{-1},$$

tracks average consensus on the input $\mathbf{Z}(s)$ with zero steady-state error.

Proposition 1 shows that the addition of a differentiator provides dynamic consensus on any signal that has a steady-state value. Corollary 2 is the obvious generalization, which shows that the addition of an appropriate "predictor" filter $C(s)$ can accomplish dynamic consensus on more general time-varying signals with polynomially bounded growth. For example, vehicles equipped with accelerometers can achieve dynamic consensus on their positions with zero steady-state error for ramp inputs.

5. ROBUST CONSENSUS

All communication media introduce time-delays in signal propagation between source and receiver. The Laplacian dynamics, like any linear system, is sensitive to time-delays; this sensitivity was explored in detail in (Olfati-Saber and Murray 2004). In particular, the integrator dynamics results in a finite phase-margin. We will now show a small-gain version of this dynamics that will converge despite arbitrarily large time-delays, provided a certain *a priori* eigenvalue bound is satisfied. Specifically, we assume that the Laplacian satisfies the bound

$$\lambda_i(L)(t) \leq \beta$$

for all i and for all t. This can be guaranteed by ensuring that the maximal degree within the network is at most $\frac{\beta}{2}$. We do not discuss how such networks may arise; for a distributed topology-control algorithm guaranteeing such a bound, see (Spanos and Murray 2004).

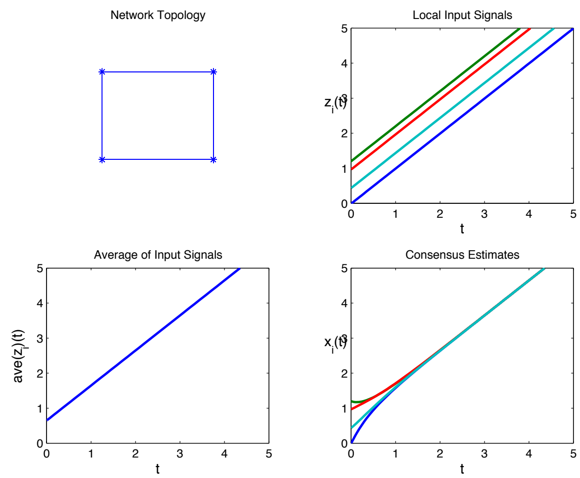

Fig. 1. Dynamic consensus for linearly growing input signals, using the single differentiator dynamics from Proposition 1. There is a small steady-state error due to the ramp inputs, but it is not visible on the graphs.

We will only show the robust version of the single-differentiator dynamics from Proposition 1; the general case follows with a similar analysis.

Observe that the Laplacian dynamics can be realized as shown in Figure 2. The feedback loop has unbounded gain at low frequency, but we can correct this by adding some dynamics within the loop. Specifically, we will use a system of the form $x_i = z_i + v_i$, where the dynamics of v_i is as follows (illustrated in Figure 3):

$$\dot{v_i}(t) = -\gamma v_i(t) - \sum_{j \in N_i} \left(x_j(t-\tau) - x_i(t-\tau) \right).$$

Proposition 3. Consider the feedback interconnection in Figure 3, where L is the Laplacian of a connected undirected graph and τ is an arbitrary time-delay. Suppose that the bound

$$\lambda_i(L) \leq \beta < \gamma$$

holds for all i. Again suppose that the input signal $\mathbf{Z}(s)$ has all its poles in the left half-plane, and has at most one pole at $s = 0$. Then, for each i

$$\lim_{t\to\infty} \left(x_i(t) - \frac{1}{N} \sum_j z_j(t) \right) = 0.$$

That is, each agent tracks the dynamic consensus with zero steady-state error, for an arbitrary value of the time-delay τ.

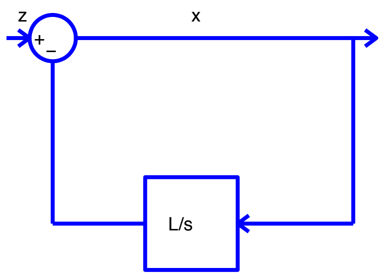

Fig. 2. Integral feedback realization of dynamic consensus.

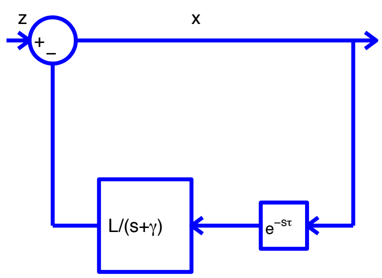

Fig. 3. A robust consensus loop (See Proposition 3 for explanation).

PROOF. It is straightforward to show that the time-delay does not affect the conservation property; see (Olfati-Saber and Murray 2004) for details. Further, the closed-loop transfer function in the absence of the time-delay is just

$$H_{xz} = \left(\frac{1}{N} \mathbf{1}\mathbf{1}^T + \sum_{i>1} \frac{s+\gamma}{s+\gamma+\lambda_i} P_i \right),$$

which achieves dynamic consensus. We must show that the loop is stable despite the addition of the time-delay. Consider the loop transfer function, *without* the time-delay:

$$\frac{1}{s+\gamma} L = \left(\sum_{i>1} \frac{\lambda_i}{s+\gamma} P_i \right).$$

All the terms in the summation are stable, and the maximum L_2 gain of the summation is bounded by $\frac{\lambda_n(L)}{\gamma}$, which is strictly bounded by unity. Since the time-delay is a non-expansive operator, we have from the small-gain theorem that the loop is stable, and must go to a steady-state. From previous arguments, this implies that the agents each track the dynamic consensus with zero steady-state error. □

Corollary 4. The above result holds with the global time-delay τ replaced by non-uniform link delays τ_{ij}, provided that the link delay is symmetric (i.e. $\tau_{ij} = \tau_{ji}$). That is, the following dynamics for v_i achieves dynamic consensus for arbitrary values of τ_{ij}:

$$\dot{v_i}(t) = -\gamma v_i(t) - \sum_{j \in N_i} \left(x_j(t-\tau_{ij}) - x_i(t-\tau_{ij}) \right).$$

6. CONSENSUS UNDER NETWORK SPLITTING AND MERGING

The Laplacian dynamics utilizes a dynamical conservation property to achieve the correct consensus value. This conservation property ensures that the sum of the x_i variables across any connected component of a graph G is constant in time. However, mobile networks do not necessarily consist of a fixed number of connected components; large networks split into small ones, and small networks merge into large ones. As a consequence, the standard Laplacian dynamics does not provide consensus tracking for these situations.

To see the source of the problem, consider a trivial example: a two-node network which splits into two one-node networks. If the input values are constants, $z_1(t) = 0$ and $z_2(t) = 1$, we would like the estimate variable $x_i(t)$ to approach $\frac{z_1+z_2}{2}$ while the network is connected, and then return to $x_i = z_i$ when the network splits. However, the Laplacian dynamics does not accomplish this. After the network splits, say at time T, the estimate variables will be fixed at $x_i(T)$, never returning to z_i. We will present a modification that corrects this behavior.

Consider the following agent dynamics:

$$x_i(t) = z_i(t) + \sum_{j \in N_i} \delta_{ij} \tag{5}$$

$$\dot{\delta}_{ij} = (x_i - x_j) \quad j \in N_i \tag{6}$$

$$\delta_{ij} = 0 \quad j \notin N_i. \tag{7}$$

A few comments are in order at this point. First, note that the overall dynamics of the x variables is just the Laplacian dynamics, i.e.

$$\dot{\mathbf{x}} = -L(t)\mathbf{x} + \dot{\mathbf{z}}.$$

Thus, for a fixed network, this dynamics achieves dynamic consensus as discussed in section 4. Second, note that the algorithm requires no more communication than the standard Laplacian dynamics, it merely requires additional memory to implement the updates on each of the δ_{ij} terms. Thus, each agent tracks as many δ_{ij} terms as it has links. We will now show that this dynamics ensures correct consensus tracking despite splitting and merging.

We present the following two lemmas without proof, due to length limitations.

Lemma 5. Consider the modified Laplacian dynamics in 5-7. Suppose that at some time T,

$A_{ij}(T) = 0$ but $A_{ij}(T^-) = 1$, i.e. the link (i,j) is lost. Suppose also that the loss of the link (i,j) does not separate the connected component of G containing i and j, say H. Then,

$$\sum_{i \in H} x_i(T^-) = \sum_{i \in H} x_i(T^+).$$

Lemma 6. Consider the dynamics 5-7, and let H_1 and H_2 be two arbitrary disjoint subsets of V whose union is V. Let E_b be the set of *boundary links* between H_1 and H_2, i.e. the links from agents in H_1 to agents in H_2. Then, we have the following transport property:

$$\sum_{k \in H_2} x_k(t) = \sum_{k \in H_2} z_k(t) + \sum_{(k,j) \in E_b} \delta_{kj}$$

Proposition 7. Consider again the modified dynamics 5-7. Let $H(t)$ be an arbitrary connected component of the graph $G(t)$. Then,

$$\sum_{i \in H} x_i(t) = \sum_{i \in H} z_i(t)$$

PROOF. [Sketch] This property holds at $t = 0$ by construction of the dynamics. Lemma 5 shows that this property holds until a time T when the connected component H splits into two connected components H_1 and H_2. Lemma 6 shows that prior to the splitting, the sum over H_2 differs from the desired value by exactly the sum of the δ_{ij} terms corresponding to the lost links. □

Thus, in steady-state each connected component of the network converges to its own average consensus despite reconfiguration (see Figure 4).

7. CONCLUSIONS

We have shown how to extend the Laplacian consensus dynamics in three directions, all relevant to applications in mobile networks. First, we have shown how to achieve consensus on time-varying signals, and how to analyze tracking in the frequency domain. Second, we have shown how to design a small-gain consensus loop, thus making the consensus dynamics robust to arbitrarily large uncertain time-delays. Finally, we have shown how to modify the agent dynamics using a completely local algorithm that compensates for splitting and merging of subnetworks.

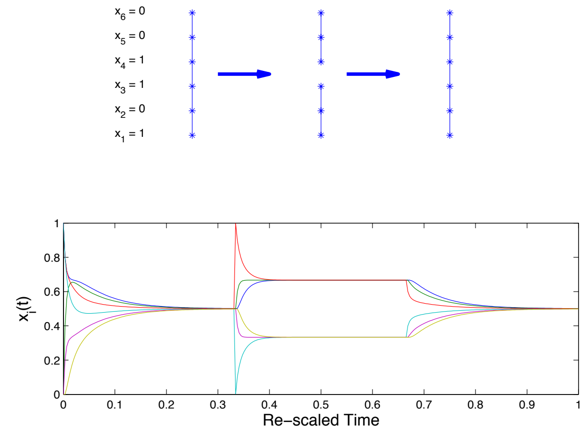

Fig. 4. The network splits then merges. The consensus variables initially approach 1/2, then after the topology change, approach 1/3 and 2/3. When the network merges, they return to 1/2.

REFERENCES

Fax, J.A. and R. M. Murray (2002). Graph laplacians and vehicle formation stabilization. In: *IFAC World Congress*.

Hatano, Y. and M. Mesbahi (2004). Consensus over random networks. In: *Proceedings of the Conference on Decision and Control*.

Jadbabaie, A., J. Lin and A. S. Morse (2002). Coordination of groups of mobile autonomous agents using nearest neighbor rules.. In: *Proceedings of the Conference on Decision and Control*.

Lynch, N. (1997). *Distributed Algorithms*. Morgan Kauffman Publishers.

Mehyar, M., D. Spanos and S. Low (2005). Distributed averaging on a peer-to-peer network. In: *Preprint (Submitted to IEEE Infocom)*.

Moreau, L. (2003). Leaderless coordination via bidirectional and unidirectional time-dependent communication. In: *Proceedings of the Conference on Decision and Control*.

Olfati-Saber, R. and R. M. Murray (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*.

Spanos, D. and R. M. Murray (2004). Robust connectivity of networked vehicles. In: *Proceedings of the Conference on Decision and Control, 2004*.

Spanos, D., R. Olfati-Saber and R. M. Murray (2005). Distributed sensor fusion using dynamic consensus. In: *Preprint (Submitted to IFAC)*.

Xiao, L. and S. Boyd (2003). Fast linear iterations for distributed averaging. In: *Proceedings of the Conference on Decision and Control*.