2005 American Control Conference
June 8-10, 2005. Portland, OR, USA



# A Control-Theoretic Perspective on the Design of Distributed Agreement Protocols

Sandip Roy Ali Saberi Kristin Herlugson

*Abstract*— **We provide a control-theoretic perspective on the design of distributed agreement protocols. First, we explore agreement-protocol analysis and design for a network of agents with single-integrator dynamics and arbitrary linear observations. One key contribution of our work is the analysis of protocols for networks with quite general observation topologies, including with multiple observations made by each agent. Another contribution is the development of techniques for agreement law design—i.e., for assignment of the dependence of the agreed-upon value on the initial states of the agents. Second, we explore agreement in a quasi-linear model with a stochastic protocol, which we call the controlled voter model. We motivate our study of this model, develop tests for whether agreement is achieved, and consider design of the agreement law. Finally, we provide some further thoughts regarding our control-theoretic perspective on agreement, including ideas for fault-tolerant protocol design using our approach.**

## I. Introduction

We explore the problem of agreement from a control-theoretic perspective, in the context of two linear models—one deterministic, one stochastic. While agreement and agreement protocols are well-studied (see [1] for a thorough development), the control-theoretic approach is relatively new ([2] and [3] are two signi£cant contributions) and, we believe, capable of providing fresh insight into agreement protocol design. Our control-theoretic approach allows us to make the following contributions:

- We show how agreement laws—functions that relate the agents' initial states to their £nal, agreed-upon value—can be designed: in particular, we show how to construct static agreement protocols (memoryless controllers) that achieve pre-speci£ed linear agreement laws, for our models. In this respect, our work builds on the analysis of [2], which discusses how to check whether a static linear agreement protocol is successful (in the same model as our deterministic one), but does not consider agreement law design. Agreement protocol design using control-theoretic methods has also been considered in [3], though the focus there is on optimizing the convergence rate rather than designing the agreement law.
- We develop agreement protocols for networks in which each agent makes multiple observations. We £nd that the multiple observations can provide greater ¤exibility in agreement law design. We also brie¤y discuss other issues related to agreement of linear networks, including fault-tolerant design and use of protocols with memory.
- We show how the control-theoretic perspective can be used to achieve agreement when the agents have discrete-valued opinions (states).

The authors are with the School of Electrical Engineering and Computer Science at the Washington State University. Correspondence should be directed to the £rst author, at sroy@eecs.wsu.edu. Support for this work was provided to the second and third authors by the Of£ce of Naval Research under grant number N000140310848.

## II. Distributed Deterministic Protocols for Continuous-Valued Agreement

In this section, we seek to identify whether or not agreement can be achieved among a group of communicating or sensing agents with single-integrator dynamics, using a static linear agreement protocol. Agreement protocols for single-integrator networks have also been considered in [2]. Our work differs from [2] in that we design protocols for achieving a desired agreement law, instead of simply checking whether a given protocol can achieve an agreement law such as average consensus. Our design-based philosophy is more similar in spirit to the approach of [3], but we focus on agreement law design rather than optimization of the protocol for a given agreement law.

### A. Model Formulation

We consider a network of $n$ **agents**, where each agent $i$ has a scalar **state** $x_i$. We assume that each agent's state is the integral of a **control input** $u_i$: $\dot{x}_i = u_i$, where the control input $u_i$ is determined by a protocol (described precisely below) used by agent $i$. This single-integrator model for individual agents is representative of various physical systems (e.g., the velocity of a vehicle in a platoon is governed by an acceleration input according to a single-integrator model), and so constitutes a useful context for studying agreement. Also, our studies (as well as those of, e.g., [3] and [2]) highlight that agreement in networks of single-integrators is quite feasible, and hence that a single-integrator model may be a reasonable choice when an agent's dynamic update can be designed along with (or as part of) its protocol (e.g., in a computer network). For convenience, we also de£ne a **state vector** $\mathbf{x}^T = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$ and an **input vector** $\mathbf{u}^T = \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix}$.

0-7803-9098-9/05/$25.00 

Each agent $i$ has available $m_i$ observations, which are used by its protocol to determine its control input. The observations made by agent $i$ may in general be arbitrary linear combinations of the state variables. That is, the vector $\mathbf{y}_i$ of observations made by agent $i$ has the form $\mathbf{y}_i = G_i\mathbf{x}$ where each $G_i$ is an $m_i \times n$ matrix. Because the matrix $G_i$ speci£es how agent $i$'s observations are in¤uenced by the other agents in the network, we call $G_i$ the **graph matrix** for agent $i$. We £nd it convenient to de£ne the **full graph matrix**

$$G = \begin{bmatrix} G_1 \\ \vdots \\ G_n \end{bmatrix}.$$

We also £nd it useful to stack the observation vectors in a single vector: $\mathbf{y}^T = \begin{bmatrix}\mathbf{y}_1^T & \ldots & \mathbf{y}_n^T\end{bmatrix}$. In this notation, $\mathbf{y} = G\mathbf{x}$.

Thus, we have speci£ed the state update and observation processes for our network of agents. We refer to the complete model as a **single-integrator network**.

### *B. Protocols, Agreement Protocols, and Agreement Laws*

We are interested in developing protocols, or mappings between observations and inputs, that achieve agreement in a single-integrator network. In this article, we shall consider static linear protocols, as de£ned below:

*De£nition 1:* A single-integrator network is said to be governed by (or to have) the static linear protocol $(K, \mathbf{z})$, where $K$ is the block-diagonal matrix $K = diag(\mathbf{k}_i')$ and $\mathbf{z}$ is an $n$-component vector, if each agent $i$'s input $u_i$ is given by $u_i = \mathbf{k}_i'\mathbf{y}_i + z_i$.

Notice that $\mathbf{k}_i$ is a column vector with $m_i$ elements, so that the protocol $K$ is a matrix of dimension $n \times \sum_{i=1}^{n} m_i$. In words, a single-integrator network is governed by (has) a static linear protocol, if each agent's input at each time is an af£ne combination of its observations at that time.

We are concerned with understanding whether a single-integrator network with full graph matrix $G$ and protocol $(K, \mathbf{z})$ achieves agreement. We de£ne agreement as follows:

*De£nition 2:* A single-integrator network with graph matrix $G$ and protocol $(K, \mathbf{z})$ is in agreement (or reaches agreement), if the states of all agents in the network converge to the same (but in general initial condition-dependent) value for all initial conditions. We refer to the value $\alpha$ reached by the agents, which is in general a function of the initial conditions, as the agreement value. A protocol $(K, \mathbf{z})$ that achieves agreement given a full graph matrix $G$ is said to be an agreement protocol or valid agreement protocol.

At the most basic level, we are interested in identifying whether or not a protocol is an agreement protocol, for a single-integrator network with given graph matrix $G$. Once we know that a protocol is an agreement protocol, we aim to characterize the agreement value $\alpha$ for the protocol—in particular, to identify the dependence of $\alpha$ on the initial states of the agents. By doing so, we can design protocols that achieve desired dependencies on the initial conditions. To this end, we de£ne the notion of an agreement law:

*De£nition 3:* Consider a single-integrator network with graph matrix $G$ and agreement protocol $(K, \mathbf{z})$. Because the single-integrator network has linear dynamics, the agreement value for this network is an af£ne function of the agents' initial states:

$$\alpha = \mathbf{p}'\mathbf{x}(0) + q \tag{1}$$

We refer to the pair $(\mathbf{p}, q)$ as the agreement law for the network.

The notion of an agreement law captures, in a general manner, the dependence of the agreed-upon value on the initial conditions of the single-integrator network.

Because the idea of an agreement law is central to our development, and because it is novel, we £nd it worthwhile to brie¤y discuss some examples. In [2] and [3], protocols that achieve *average consensus* are studied. These are agreement protocols for which the agreement value is the arithmetic average of the agents' initial conditions. In our terminology, a network that reaches average consensus is one that has the agreement law ($\mathbf{p} = \begin{bmatrix}\frac{1}{n}, & \ldots, & \frac{1}{n}\end{bmatrix}'$, $q = 0$). While average consensus is indeed a reasonable design goal for some networks, other design goals may be desired for some networks. For instance, a network may require that all agents converge to the initial value of one of the agents, say Agent 1. In our terminology, this design goal can be stated as the goal of achieving the agreement law ($\mathbf{p} = \begin{bmatrix}1 & 0 & \ldots 0\end{bmatrix}'$, $q = 0$). Yet other design goals may be agreement on an initial-condition independent value (agreement law $(\mathbf{0}, q)$ for some $q$), or agreement on a particular weighted average of agents' initial states (agreement law $(\mathbf{p}, 0)$ for some $\mathbf{p}$).

### *C. Test for Agreement and Identi£cation of Agreement Laws*

In this section, we specify tests for determining whether a particular static linear protocol is in fact an agreement protocol, and calculate the agreement laws when these agreement protocols are used. To specify these test, we £rst note that the state vector of a single-integrator network with graph matrix $G$ and protocol $(K, \mathbf{z})$ satis£es the following differential equation:

$$\dot{\mathbf{x}} = KG\mathbf{x} + \mathbf{z}. \tag{2}$$

From this closed-loop system equation, we can straightforwardly identify whether or not the single-integrator network reaches agreement, and determine the agreement law if it does. We £nd it useful to differentiate between two cases, in specifying conditions for agreement. In particular, we specify conditions for agreement to a value that is independent of the initial conditions, and then separately specify conditions for agreement to an initial-condition dependent value. These conditions are described in the following theorem:

*Theorem 1: A single-integrator network with graph matrix $G$ and protocol $(K, \mathbf{z})$ reaches agreement if and only if one of the following two conditions hold:*

- *all the eigenvalues of $KG$ lie in the open left half plane (i.e., have strictly negative real parts), and $-(KG)^{-1}\mathbf{z} = \alpha\mathbf{1}$ for some $\alpha$. In this case, the agreement law for the network is $(\mathbf{0}, \alpha)$. We refer to this scenario as Type 1 Agreement.*
- *$KG$ has one eigenvalue of 0 with corresponding right eigenvector $\mathbf{1}$, the remaining eigenvalues of $KG$ are in the open left half plane, and $\mathbf{z} = 0$. In this case, the agreement law for the network is $(\mathbf{w}, 0)$ where $\mathbf{w}'$ is the left eigenvector of $KG$ corresponding to the 0 eigenvalue. (In particular, we are referring to the left eigenvector $\mathbf{w}'$ such that $\mathbf{w}'KG = 0$ and $\mathbf{w}'\mathbf{1} = 1$. Henceforth, we refer to such a left eigenvector as a* standard *left eigenvector*[1]*.) We refer to this scenario as Type 2 Agreement.*

### *D. Existence and Design of Agreement Laws*

Now that we have developed tests for checking whether agreement is achieved by a given protocol, we are in a position to study the existence and design of protocols that achieve desired agreement laws. In this section, we present results that facilitate design of protocols to achieve such agreement laws. The results fall into one of two categories:

- Given a single-integrator network with full graph matrix $G$ and a set of allowed agreement laws, we specify conditions on $G$ for the existence of a protocol that achieves some agreement law within the set.
- Given a single-integrator network with full graph matrix $G$ and a set of desired agreement laws, we specify conditions on $G$ such that we can design protocols to achieve all agreement laws within the set. That is, we specify conditions on $G$ for *arbitrary assignability* of the agreement law within the set. As far as we know, our study of agreement law assignment is novel not only among control-theoretic studies but more generally in the computer science community.

Several of the conditions that we specify are of the following form: if $G$ belongs to a certain class of matrices, then existence/assignability of the agreement law is guaranteed.

Let us £rst consider Type 1 Agreement Laws—i.e., the set of agreement laws $(0, \alpha)$, where $\alpha \in \mathcal{R}$. The following theorem presents a condition for both existence of protocol for achieving an agreement law in this set, and assignability of any arbitrary agreement law in the set.

*Theorem 2: Consider a single-integrator network with graph matrix $G$. A protocol exists such that an agreement law of the form $(0, \alpha)$, where $\alpha \in \mathcal{R}$, is achieved, if and only if there is a block-diagonal matrix $K$ (of the proper dimensions) such that all eigenvalues of $KG$ are in the OLHP. Furthermore, in this case, protocols can be designed to achieve any agreement law of the form $(0, \alpha)$.*

A condition for the existence of an agreement protocol that is phrased explicitly in terms of the graph matrix $G$ can also be developed:

*Theorem 3: Consider a single-integrator network with square graph matrix $G$. A protocol exists such that an agreement law of the form $(0, \alpha)$, where $\alpha \in \mathcal{R}$, is achieved, if there is a permutation of $G$ such that all leading principal minors have full rank. Furthermore, in this case, protocols can be designed to achieve any agreement law of the form $(0, \alpha)$.*

Next, let us consider design of protocols for Type 2 Agreement (i.e., that achieve agreement laws of the form $(\mathbf{p}, 0)$. We £rst study existence of a protocol that achieves some Type 2 Agreement Law.

*Theorem 4: Assume there exists an appropriately-dimensioned block diagonal matrix $K$ such that $KG$ has a single zero eigenvalue with right eigenvector $\mathbf{1}$, and the remaining eigenvalues of $KG$ are negative. Then we can design a protocol such that some agreement law of the form $(\mathbf{p}, 0)$ can be achieved. In particular, the protocol $(K, \mathbf{0})$ achieves the agreement law $(\mathbf{w}, 0)$, where $\mathbf{w}'$ is the left eigenvector of $KG$ corresponding to the zero eigenvalue.*

A condition that is explicit in $G$ can also be developed:

*Theorem 5: Assume that we can £nd vectors $\mathbf{v}_1 \in Ra(G_1^T), \ldots, \mathbf{v}_n \in Ra(G_n^T)$, such that $V^T = \begin{bmatrix} \mathbf{v}_1 & \ldots & \mathbf{v}_n \end{bmatrix}$ has one zero eigenvalue with corresponding right eigenvector $\mathbf{1}$, and that there is a permutation of $V$ whose leading principal minors have full rank. Then we can design a protocol such that some agreement law of the form $(\mathbf{p}, 0)$ can be achieved.*

In the remainder of this section, we describe conditions on $G$ that guarantee existence of an agreement law within a particular **quadrant**, as well as conditions on $G$ that guarantee that every agreement law within a quadrant can be achieved using some protocol. (We use the term **quadrant** of a Type 2 agreement law $(\mathbf{p}, 0)$ to refer to the sign pattern of the entries in $\mathbf{p}$. For instance, if all entries in $\mathbf{p}$ are positive, we refer to agreement law as lying in the £rst or positive quadrant.) For convenience, we assume that each agents makes only a single observation (i.e., that $G$ is square) in these studies. The generalization to non-square $G$ can be achieved in much the same way as for Theorem 5.

We are especially interested in identifying $G$ for which all agreement laws within a quadrant can be achieved, because these are graph matrices for which essentially arbitrary agreement law design is possible[2]. That is, for such graph matrices we can decide on a desired dependence of the agreement value on the initial states of the agents (at least

[1]Notice that the sum of the components of the agreement law is always 1.

[2]When we study whether "every" agreement law within a quadrant can be achieved, we implicitly consider only agreement laws $(\mathbf{p}, 0)$ such that $\mathbf{p}'\mathbf{1} = 1$. In other words, since the sum of the entries in $\mathbf{p}$ is 1 for any achievable agreement law, we implicitly assume that this constraint is met for any desired agreement law.

within a quadrant), and £nd a protocol that achieves this agreement law. Thus, we begin with this case.

We £nd it most enlightening to relate arbitrary assignment of the agreement law within a quadrant to the notion of **$D$-semistability** (e.g., [6]), so we begin with a de£nition of $D$-semistability.

*De£nition 4:* The matrix $A$ is said to be $D$-semistable if the eigenvalues of the matrix $DA$ are in the closed left half plane and the eigenvalues of $DA$ on the $j\omega$-axis are simple, for all positive diagonal $D$. [3]

We shall show that arbitrary assignment of the agreement law is possible when the graph matrix (or another matrix that is closely related with the graph matrix) is $D$-semistable. The advantage of characterizing arbitrary assignment using $D$-semistability is that many common classes of matrices are known to be $D$-semistable, so that we are immediately able to identify classes of matrices for which arbitrary assignment is possible. The relationship between $D$-semistability and arbitrary assignment is described in the following theorem:

*Theorem 6: Consider a single-integrator network in which each agent makes a single observation. We can develop a protocol to achieve any agreement law[4] of Type II (i.e., of form* $(\mathbf{p}, 0)$*) within some quadrant if and only if the following three conditions hold:*

- *The matrix* $ZG$ *is D-semistable, where* $Z = diag(-sign(g_{ii}))$.
- *The right eigenvector of G corresponding to the single eigenvalue at the origin is the vector* $\mathbf{1}$*. Also, the corresponding left eigenvector of G has strictly non-zero entries.*
- $ZG$ *has no eigenvalues on the* $j\omega$ *axis other than the single eigenvalue at the origin.*

*In this case, the quadrant in which any agreement law can be achieved is the one with sign pattern given by* $\mathbf{w}'Z$*, where* $\mathbf{w}'$ *is the left eigenvector of G corresponding to the zero eigenvalue.*

Although the conditions required for arbitrary assignment of the agreement law in a quadrant seem unwieldy, they can straightforwardly be checked because they are phrased directly in terms of the graph matrix $G$. In particular, the following steps can be followed to identify whether the conditions for arbitrary agreement are met:

1) $D$-semistability of $ZG$ can be veri£ed by determining that $ZG$ belongs to one of several well-known classes of matrices. These classes of matrices are discussed in some detail below.
2) The remaining conditions can be checked through eigenanalysis of $ZG$.

[3] Our notion of $D$-semistability differs from the linear algebra notion, in that we constrain eigenvalues on the $j\omega$ axis to be simple. We believe that this de£nition for $D$-semistability is germane in our context because internal stability of linear systems requires that imaginary axis eigenvalues are simple. We shall clarify classes of matrices that are $D$-semistable by our de£nition later in the article.

[4] As always, we implicitly consider only agreement laws whose components sum to one, since only these are possible.

The procedure above highlights that $D$-semistability of the graph matrix must be determined to check whether arbitrary assignment is possible. Unfortunately, there is no systematic procedure for checking $D$-semistability of a matrix. Luckily, however, there are several broad classes of matrices whose members are known to be $D$-semistable and also can be easily identi£ed. We list several such classes of matrices, brie¤y describing techniques for determining whether a matrix is a member of each class as needed. We also illustrate the relationships between these classes of matrices in Figure II-D. We omit the justi£cations that matrices in these classes are $D$-semistable; the reader is referred to [6] for these details.

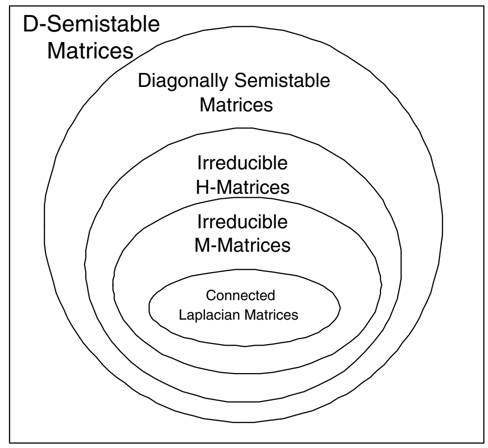


Fig. 1. A Venn diagram of some classes of $D$-stable matrices is shown.

*(1)* Matrices that are **diagonally semistable** are also $D$-semistable. The matrix $A$ is said to be diagonally semistable if there exists a positive diagonal matrix $D$ such that $A^T D + DA$ is positive semi-de£nite. Optimization machinery has been used to develop a test for whether a matrix is diagonally semistable (see [5]).

*(2)* If $-A$ is an irreducible $H$-matrix with nonnegative diagonal entries, then $A$ is diagonally semistable, and hence $D$-semistable. $H$-matrices are a fairly straightforward generalization of the class of $M$ matrices (see below).

*(3)* If $-A$ is an irreducible $M$-matrix with nonnegative diagonal entries, then $-A$ is an irreducible $H$ matrix with nonnegative diagonal entries, and so is diagonally semistable and hence $D$-semistable. Recall that an $M$-matrix is one with non-negative principal minors and non-positive off-diagonal entries.

*(4)* If $-A$ is an irreducible Laplacian matrix, then $-A$ is an irreducible $M$ matrix, and hence $D$-semistable. When $-A$ is an irreducible Laplacian matrix, all the other conditions of Theorem 6 also hold. Hence, arbitrary assignment of the agreement law to a quadrant (in particular, the all-positive quadrant) is possible.

We conclude this section with a test for whether some agreement law within a speci£ed quadrant can be achieved using a valid agreement protocol:

*Theorem 7: Consider a single integrator network with square graph matrix G, and say that we wish to see whether some agreement law with the same sign pattern as a particular n-component vector* $\mathbf{v}$ *can be achieved. Assume that G has a zero eigenvalue, with corresponding right eigenvector* $\mathbf{1}$ *and left eigenvector* $\mathbf{w}'$*. (We assume that the entries of* $\mathbf{w}'$ *are non-zero; agreement within a quadrant is impossible if they are not.) Also, de£ne a diagonal matrix* $Z = diag(\frac{sign(w_1)}{sign(v_1)})$*. Then we can £nd some agreement protocol such that the agreement law has the same sign pattern as* $\mathbf{v}$ *if there is a permutation of ZG for which all leading principal minors of order less than n are positive.*

We note that Theorem 7 and Theorem 5 are closely related. In particular, if the premise for Theorem 5—that there is a permutation of $G$ such that the £rst $n-1$ leading principal minors have full rank—is satis£ed then we can necessarily £nd a diagonal matrix $Z$ with entries of $\pm 1$ on the diagonal, such that the £rst $n-1$ leading principal minors of $ZG$ are positive. Thus, we verify that there is a quadrant in which we can place the agreement law, as we would expect.

## III. Controlled Voter Model for Discrete-Valued Agreement

In this second part, we discuss the design of agreement protocols in the context of a quasi-linear discrete-time, discrete-state stochastic model, which we call the *controlled voter model*. Our motivations for studying agreement in a discrete-state and stochastic model are threefold: £rst, agreement among agents with discrete-valued states is required in several contexts, such as among jurors deciding on a defendant's guilt or several parallel process comparing the binary output of a computation. Second, protocols that are based on copying or choosing among several decisions—such as the one to be developed for our model—are easy to implement in some applications, since they often require minimal computation. Third, when agreement among agents with discrete-valued states is required, probabilistic decision-making often is necessary to reach agreement in an equitable manner, and hence stochastic models for protocols are relevant.

The controlled voter model provides a realistic context for studying agreement, yet is suf£ciently structured to permit signi£cant analysis of state dynamics and design of agreement laws. Essentially, the model is tractable because expected value of the state of the closed-loop system (the system when the protocol is applied) satis£es a linear recursion. Thus, we can re-phrase the problem of agreement as a linear control problem.

### A. Model Formulation and Connection to Literature

A controlled voter model comprises a network of $n$ agents, each with a scalar state variable $x_i \in \{0, 1\}$ that is updated in discrete time. The state update of each agent is governed by a stochastic protocol: in particular, the state of agent $i$ at time-$k+1$ is given by

$$x_i[k+1] = 1, \text{ w.p. } u_i[k]$$
$$x_i[k+1] = 0, \text{ w.p. } 1 - u_i[k],$$

where the **mean input** $u_i[k] \in [0, 1]$ is computed by the protocol from agent $i$'s concurrent observations[5]. Notice that we de£ne the protocol to include both the computation of $u_i[k]$ from the observations, and the stochastic determination of the next-state $x_i[k+1]$ based on $u_i[k]$. That is, the protocol uses the observations to set the probability that the next state will be 1, and then realizes the next state based on this probability. (This is in contrast to the single-integrator network, in which we view the relationship between the input and state as part of the intrinsic dynamics of the model rather than the protocol.) For convenience, we de£ne a **state vector** $\mathbf{x}^T = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$ and a **mean input vector** $\mathbf{u}^T = \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix}$.

Each agent $i$ makes $m_i$ observations. Each observation is a weighted average of the concurrent state variables. That is, the observations made by agent $i$ are given by $\mathbf{y}_i = G_i \mathbf{x}$, where the $m_i \times n$ graph matrix $G_i$ is a **row-stochastic** matrix—i.e., one in which the elements in each row are non-negative and sum to 1. Notice that observations in our formulation can include states variables of single agents and weighted averages of multiple agents' states (e.g., of neighboring agents in a graph). Because we have enforced that $G_i$ is row-stochastic, the entries in each vector $\mathbf{y}_i$ are necessarily in the interval $[0, 1]$ for any state vector. For convenience, we again de£ne a **full observation vector** $\mathbf{y}^T = \begin{bmatrix} \mathbf{y}_1^T & \dots & \mathbf{y}_n^T \end{bmatrix}$ and a **full graph matrix** $G = \begin{bmatrix} G_1 \\ \vdots \\ G_n \end{bmatrix}$.

Our protocol calculates each agent's mean input $u_i$ from its observation vector $\mathbf{y}_i$. We assume that this mapping is static and linear: the mean input $u_i$ is determined as

$$u_i = \mathbf{k}_i' \mathbf{y}_i. \quad (3)$$

We further enforce that the entries in $\mathbf{k}_i'$ are non-negative and sum to 1, so that $u_i$ is a weighted average of the entries in $\mathbf{y}_i$. This further constraint ensures that each input $u_i$ is in the interval $[0, 1]$ at each time-step, as required. We note, further, that all $u_i$ equal 0 at a given time if all agents' states at that time are zero, and that all $u_i$ equal one if all agents' concurrent states are unity. For convenience, we codify the protocol using the block-diagonal matrix $K = diag(\mathbf{k}_i')$. Notice that $\mathbf{k}_i$ is a column vector with $m_i$ elements, so that the **protocol matrix** $K$ is a matrix of dimension $n \times \sum_{i=1}^{n} m_i$. The protocol matrix relates the observation vector to the mean input vector, as $\mathbf{u} = K\mathbf{y}$. A controlled voter model is speci£ed completely by its graph matrix $G$ and protocol matrix $K$, and hence we identify a particular model with its graph $G$ and protocol $K$.

[5]We use the term *mean input* since the expected value of the "actual input" (which is the next-state) is $u_i[k]$.

Our controlled voter model is a natural extension (in discrete time) of the *voter model* (equivalently, *invasion process*), originally introduced in [7] and [8] and studied in further detail in [9] and [10]. The stochastic realization of the next-state from the mean input in these models is identical to ours; they are different in that the mean inputs are prescpeci£ed linear combinations of state variables, rather than being speci£ed as an observation operation followed by a decentralized control operation. Thus, although the closed-loop dynamics of the voter model and controlled voter model are identical, the voter model is viewed as representing a £xed process (that may or may not reach agreement) while the controlled voter model is a tool for designing protocols for agreement. It is our belief that the protocol-design perspective on the voter model is a valuable one. While the voter model may be too simplistic to represent many pre-existing systems, we are free to design protocols as we see £t, and the voter model turns out to be a good choice because of its tractability and performance.

### B. De£nition of Agreement

The notion of agreement in the controlled voter model is essentially the same as for the single-integrator network: a model is in agreement if the states of all the agents asymptotically reach the same value. In the case of a controlled voter model that reaches agreement, this asymptotic value is either zero or one. The asymptotic value reached by the agents is in general stochastic—convergence to either all zeros and all ones is possible on any given trial. Thus, in contrast to the determinstic model, the agents' initial states do not exactly specify the value that is agreed upon by the agents; instead, these initial conditions specify the probability that agreement to the zero state or the unity state is achieved. With this difference in mind, we de£ne the notion of an agreement value and agreement law in terms of the probabilities of reaching either asymptotic state. It is also worth mentioning that the notion of convergence to the same value is now a probabilistic notion (e.g., *convergence in probability* or *convergence with probability* 1, see [11] for instance). We formalize these notions in the following de£nitions.

*De£nition 5:* A controlled voter model with graph $G$ and protocol $K$ is in agreement, if the states of all agents in the model converge to the same value (i.e., become identical) with probability 1, for all initial conditions. We refer to the probability $\alpha$ that the agents reach the unity state (which, as our subsequent analysis shows, is a function of the initial conditions) as the agreement probability. A protocol $K$ that achieves agreement for a given graph $G$ is said to be an agreement protocol.

*De£nition 6:* Consider a single-integrator network with graph matrix $G$ and agreement protocol $K$. As shown in a following section, the agreement probability for this network turns out to be a linear function of the agents' initial states:

$$\alpha = \mathbf{p}'\mathbf{x}[0]. \tag{4}$$

We refer to $\mathbf{p}$ as the agreement law for the network.

Sections III-D and III-E describe analysis of agreement protocols and design of protocols to achieve speci£c agreement laws, respectively. As with the deterministic model, the ageement law achieved by a given protocol, and the possibility for designing protocols, are strongly dependent on the structure of the graph matrix $G$.

### C. Summary of Graph-Theoretic Concepts

Our study of protocol analysis and design for the controlled voter model turns out to be deeply related to some graph-theoretic concepts for matrices with non-negative entries. In particular, let us consider a square matrix $n \times n$ matrix $A$ with non-negative entries. The **pictorial graph** of $A$, denoted $\Gamma(A)$, comprises $n$ nodes or vertices. A directed edge (arrow) is drawn from vertex $i$ to vertex $j$, if and only if $A_{ij}$ is non-zero. The vertices and edges together constitute the graph. (Notice that we use the term *pictorial graph* rather than *graph* to distinguish from the graph matrix.)

Several concepts regarding the structure of the pictorial graph are of importance. In the interest of space, we omit de£nitions of these concepts and refer the reader to [12] for details.

### D. Analysis of Protocols

In this section, we consider controlled voter models with given graph $G$ and protocol $K$, and 1) determine whether the model reaches agreement and, if so, 2) characterize the agreement law for the model. Thus, as with the single-integrator network, we develop conditions that can be used to check whether or not agreement is achieved, and to determine the agreement law achieved by a particular protocol. Because the graph matrix and protocol matrix for the controlled voter model are rather strictly constrained (each matrix is row-stochastic), it turns out that agreement is achieved for almost all $G$ and $K$: unlike the deterministic model, there is no possibility for strictly unstable closed-loop dynamics, and achievement of agreement is instead solely contigent on whether a single status value can dominate the dynamics. Like the single-integrator network, we £nd that the agreement law for the controlled voter model can be determined through eigenanalysis of $KG$. Our analysis of agreement in the controlled voter model is formalized in two theorems below:

*Theorem 8: A controlled voter model with graph G and protocol K reaches agreement, if and only if the pictorial graph $\Gamma((KG)')$ has a single autonomous class and that autonomous class is ergodic.*

The next theorem characterizes the agreement law for a controlled voter model with graph $G$ and agreement protocol $K$. For convenience, we only consider the case that $KG$ comprises a single ergodic class, although the theorem readily generalizes to the necessary and suf£cient case considered in Theorem 8.

*Theorem 9: Consider a controlled voter model with graph $G$ and protocol $K$, for which $\Gamma((KG)')$ comprises a single ergodic class. Notice that, from Theorem 8, $K$ is an agreement protocol. The agreement law for this controlled voter model is $\mathbf{p}$, where $\mathbf{p}'$ is the left eigenvector of $KG$ corresponding to its unity eigenvalue.*

### E. Design of Agreement Laws

Just as in our deterministic model, design of the agreement law is feasible in the controlled voter model. That is, we can characterize the set of agreement laws that can be achieved by some protocol, for a given graph matrix. The ease with which agreement laws can be designed is a compelling feature of our formulation, since it allows us to design agreement protocols that weight each agent's initial state differently.

The design of agreement laws for the controlled voter model is simpler than for the single-integrator network: because stability is guaranteed for any protocol, the set of allowed agreement laws can be characterized solely by determining how the protocol $K$ impacts the left eigenvector of $KG$ corresponding to the unity eigenvalue[6]. Our characterization of the set of achievable agreement laws is phrased in terms of left eigenvectors of certain submatrices of $G$. We begin our development by de£ning appropriate notations for these submatrices and eigenvectors:

*De£nition 7:* We shall consider $n \times n$ submatrices of $G$ comprising single rows from each agent's graph matrix $G(i)$. In particular, let us consider a matrix whose $j$th row is row $i_j \in 1, \dots, m_j$ of $G(j)$. We refer to this matrix as the reduced full graph matrix with observation list $\mathbf{i} = \{i_1, \dots, i_n\}$, and use the notation $\widehat{G}_{\mathbf{i}}$ for the matrix. We also de£ne the protocol $K_{\mathbf{i}}$ for observation list $\mathbf{i}$ to be the protocol for which each block-diagonal matrix (vector) $\mathbf{k}_j$ is an indicator vector with unity entry at entry $i_j$. We note that $\widehat{G}_{\mathbf{i}} = K_{\mathbf{i}}G$. If $\widehat{G}_{\mathbf{i}}$ is ergodic, it has a single unity eigenvalue. In such cases, we use the notation $\widehat{\mathbf{p}}'_{\mathbf{i}}$ for the corresponding left eigenvector. Finally, we note that there are $m = \prod_{i=1}^{n} m_i$ reduced full graph matrices (and corresponding unity eigenvectors).

We are now ready to present our main theorem concerning agreement law design:

*Theorem 10: Consider a controlled voter model with graph matrix $G$, and assume that all reduced full graph matrices for this controlled voter model are ergodic. Then an agreement law $\mathbf{p}$ can be achieved using some protocol $K$, if and only if $\mathbf{p}$ can be written in the form*

$$\sum_{\mathbf{i}} \alpha_{\mathbf{i}} \widehat{\mathbf{p}}_{\mathbf{i}}, \tag{5}$$

*where the $m$ coef£cients $\alpha_{\mathbf{i}}$ are positive and sum to one. That is, an agreement law can be achieved if and only if it is a convex combination of the left eigenvectors of the full graph matrices corresponding to the unity eigenvalue.*

A few further notes about application of the stochastic protocol design are worthwhile:

- To determine whether a given agreement law $\mathbf{p}$ can be achieved, one must check whether $\mathbf{p}$ lies in the convex hull de£ned by the vectors $\mathbf{p}_{\mathbf{i}}$. Methods for checking are well-known (see, e.g., [13]). These methods also serve to identify the coef£cients $\alpha_{\mathbf{i}}$ that relate $\mathbf{p}$ to the $\mathbf{p}_i$.
- We have identi£ed the set of achievable agreement laws, but have not yet shown how to choose a protocol to achieve a particular agreement law in this set. In fact, the procedure for choosing the protocol is implicitly contained in our suf£ciency proof above. In particular, given a desired agreement law $\mathbf{p}$, we need to £rst compute $\mathbf{q}' = (\sum_{\mathbf{i}} \alpha_{\mathbf{i}} \widehat{\mathbf{p}}'_{\mathbf{i}} K_{\mathbf{i}})$, where the $\alpha_{\mathbf{i}}$ are found as described in the £rst item in this list. Next, we need to choose $\overline{K}$ so that $\mathbf{q}' = (\sum_{\mathbf{i}} \alpha_{\mathbf{i}} \mathbf{p}'_{\mathbf{i}})\overline{K}$. This can be done easily, by normalizing[7] the components of $\mathbf{q}$ corresponding to £nd $\mathbf{k}_i$.

*Example 1:* We consider a controlled voter model with three agents. The £rst and third agents each make two observations, while the second agent only makes a single observations. The full graph matrix for this example is the following:

$$G = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.6 & 0.3 & 0.1 \\ \hdashline 0.2 & 0.5 & 0.3 \\ \hdashline 0 & 0.2 & 0.8 \\ 0.4 & 0.3 & 0.3 \end{bmatrix} \tag{6}$$

We apply Theorem 10 to identify the agreement laws that we can achieve using some protocol. The agreement laws that can be achieved are illustrated in Figure III-E. We only show the £rst two components $p_1$ and $p_2$ of the agreement law on the plot, since the third component is determined explicitly from the £rst two. Using the procedure described above, we can £nd the protocol matrix required to achieve any desired agreement law in this set if we wish.

## IV. Further Directions

We have developed agreement protocols for two applicable dynamic models, focusing in particular on exposing the role of the communication network structure on protocol development. We believe our studies provide a compelling framework for understanding agreement in several dynamic systems. However, some further analyses can signi£cantly expand the applicability of our framework. Here, we brie¤y list several directions of analysis that we are currently pursuing, along with some initial results.

[6] To be precise, we further need to check that $KG$ is ergodic, but we envision that ergodicity will be achieved naturally in many applications.

[7] to a unity sum

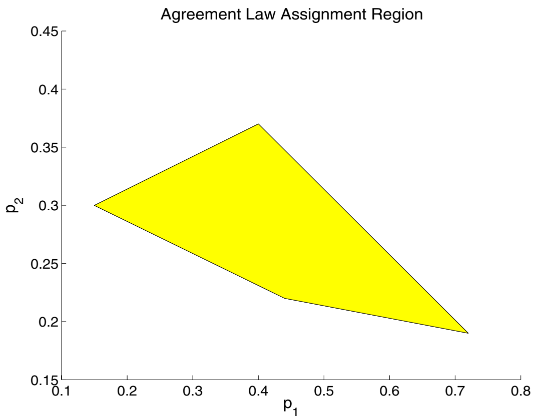


Fig. 2. The agreement laws that can be achieved using some protocol are illustrated for an example controlled voter model with three agents. We only show the £rst two components $p_1$ and $p_2$ of the agreement law on the plot, since the third component is determined explicitly from the £rst two.

*a) Fault Tolerance:* Fault tolerance is often a requirement for an agreement protocol, especially in distributed computing applications (see, e.g., [1]). In these applications, protocols must be tolerant of random loss of communication, as well as purposeful miscommunications. Loss-of-communication faults may also be prevalent in autonomous vehicle applications (e.g., [14]) and other applications that involve transmission through a noisy environment.

We believe that our framework permits study of fault tolerance, because we allow for agents that make multiple observations, and because we explicitly consider design of agreement protocols. Thus, we can hope to design agreement protocols that are robust to common faults. For instance, for the single-integrator network, we can aim to develop a protocol that is robust to a single fault among a set of observations. Another common fault in a single-integrator network may be complete failure of an agent (i.e., loss of control of the agent as well as exclusion of observation of it by the other agents). In such a situation, our protocol should seek to achieve agreement among the remaining agents despite the lost observations. We expect that the ability to design such a fault-tolerant protocol is deeply connected to the notion of $D$-semistability, since $D$-semistable systems are robust to many changes in the graph matrix.

In a similar manner, we can construct a stochastic agreement protocol in order to minimize dependence on communications that are faulty or on potential agent failures. Further, by assuming a stochastic model for faults, we can characterize the expected dynamics of our network once a protocol is applied.

*b) More Complicated Network Dynamics:* Another direction that we are currently pursuing is the development of agreement protocols for networks with more complex intrinsic dynamics. In particular, agents with double-integrator dynamics (rather than single-integrator dynamics) are common in mechanical systems, and so are of interest to us. For instance, if we are interested in achieving agreement among the positions (rather than velocities) of autonomous vehicles in a network, double-integrator dynamics must be considered since typically the accelerations of the vehicles are controlled. The techniques used to design agreement protocols in this article can readily be adapted for double-integrator networks, by meshing them with the analysis techniques discussed in [4]. We expect to discuss agreement in double-integrator networks in a subsequent article.

We have just begun to study agreement for networks in which agents' dynamics are intrinsically interconnected. Power networks and air traf£c networks are examples of systems in which agents' dynamics are dependent on each other. We believe that agreement in such networks can be analyzed by applying the decentralized control analysis of [15], as we have done for integrator networks in [4]. We are also interested in considering interdependent dynamics in the controlled voter model, by meshing uncontrollable interactions (i.e., standard voter model updates) with controllable dynamics.

## REFERENCES


[1] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann: San Mateo, CA, 1996.

[2] R. Olfati Saber and R. M. Murray, 'Agreement Problems in Networks with Directed Graphs and Switching Topology', *Proceedings of the 2003 Conference on Decision and Control*, Maui, Hawaii, Dec. 9-12, 2003.

[3] L. Xiao and S. Boyd, 'Fast Linear Iterations for Distributed Averaging', accepted for publication in *Systems and Control Letters*, 2004.

[4] S. Roy, A. Saberi, and K. Herlugson, 'Formation and Alignment of Distributed Sensing Agents with Double-Integrator Dynamics', accepted for publication in the IEEE Press Monograph *Sensor Network Operations*, 2004.

[5] H. K. Khalil, 'On the existence of positive diagonal $P$ such that $PA + AP < 0$, *IEEE Transactions on Automatic Control*, Vol. AC-27, pp. 181-184, 1982.

[6] D. Hershkowitz, 'Recent Directions in Matrix Stability', *Linear Algebra and its Applications*, Vol. 171, pp. 161-186, Jul., 1992.

[7] R. A. Holley and T. M. Liggett, 'Ergodic theorems for weakly interacting in£nite systems and the voter models', *Annals of Probability*, Vol. 3, pp. 643-663, 1975.

[8] P. Clifford and A. Sudbury, 'A model for spatial con¤ict', *Biometrika*, Vol. 60, No. 3, pp. 581-588, 1973.

[9] C. Asavathiratham and S. Roy and B. C. Lesieutre and G. C. Verghese, 'The In¤uence Model', *IEEE Control Systems Magazine*, Vo. 21, No. 6, pp. 52-64, Dec. 2001.

[10] C. Asavathiratham, *The In¤uence Model: A Tractable Representation for the Dynamics of Networked Markov Chains*, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Oct. 2000.

[11] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes,* $3^{rd}$ *ed.*, Oxford University Press, Aug. 2001.

[12] R. G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers: Boston, 1996.

[13] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison Wesley: Menlo Park, CA, 1973.

[14] P. Seiler and R. Sengupta, "Analysis of Communication Losses in Vehicle Control Problems", *Proceedings of the American Control Conference*, Arlington, Virginia, June 25-27, 2001.

[15] S. Wang and E. J. Davison, "On the stabilization of decentralized control systems", *IEEE Transactions on Automatic Control*, vol. AC-18, pp. 473-478, Oct. 1973.