Structural Controllability and Observability of Linear Systems Over Finite Fields with Applications to Multi-Agent Systems

Shreyas Sundaram, *Member, IEEE*,
and Christoforos N. Hadjicostis, *Senior Member, IEEE*

Abstract

We develop a graph-theoretic characterization of controllability and observability of linear systems over finite fields. Specifically, we show that a linear system will be structurally controllable and observable over a finite field if the graph of the system satisfies certain properties, and the size of the field is sufficiently large. We also provide graph-theoretic upper bounds on the controllability and observability indices for structured linear systems (over arbitrary fields). We then use our analysis to design nearest-neighbor rules for multi-agent systems where the state of each agent is constrained to lie in a finite set. We view the discrete states of each agent as elements of a finite field, and employ a linear iterative strategy whereby at each time-step, each agent updates its state to be a linear combination (over the finite field) of its own state and the states of its neighbors. Using our results on structural controllability and observability, we show how a set of leader agents can use this strategy to place all

This material is based upon work supported in part by the National Science Foundation under NSF CNS Award 0834409. The research leading to these results has also received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreements INFSO-ICT-223844 and PIRG02-GA-2007-224877. Part of this research has also received support from the Natural Sciences and Engineering Research Council of Canada (NSERC). Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF, EC or NSERC. Parts of this paper were presented in preliminary form at the 2010 American Control Conference and 2009 Conference on Decision and Control.

S. Sundaram is with the Department of Electrical and Computer Engineering, University of Waterloo, 200 University Ave. W., Waterloo, ON, Canada, N2L 3G1. E-mail: ssundara@uwaterloo.ca. C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, 75 Kallipoleos Avenue, P.O. Box 20537, 1678 Nicosia, Cyprus, and also with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. E-mail: chadjic@ucy.ac.cy.

agents into any desired state (within the finite set), and how a set of sink agents can recover the set of initial values held by all of the agents.

Index Terms

Structured system theory, linear system theory, finite fields, structural controllability, structural observability, quantized control, multi-agent systems, distributed consensus, distributed function calculation

I. INTRODUCTION

The emergence of sensor and robotic networks has prompted a tremendous amount of research into the problems of distributed control and information dissemination in multi-agent systems [1], [2], [3], [4], [5]. The topics of *distributed consensus* (where all agents converge to a common decision after interactions with their neighbors) and multi-agent control (where the agents are driven to some desired state by some leader agents) are examples of canonical problems in this setting [2], [3], [6], [7], [8], [9], [10], [11]. Researchers have also started to consider what happens when the interactions and dynamics in the system are constrained in various ways. One of the main thrusts along these lines has been to investigate the problem of *quantization*, where the agents can only occupy a fixed number of states, or can only exchange a finite number of bits with their neighbors (e.g., due to bandwidth restrictions in the communication channels). In the context of distributed consensus, various works (including [12], [13], [14], [15], [16], [17]) have revealed that nearest-neighbor rules can be adapted in different ways in order to obtain agreement despite the quantized nature of the interactions. The proposed solutions range from using *gossip*-type algorithms (where an agent randomly contacts a neighbor and then the two bring their values as close together as possible, under the quantization constraint) [12], [16], [17], to incorporating quantization steps into (otherwise) linear update strategies for each agent [13], [14], [15], [18], [19]. Along similar lines, the topic of logical consensus (where agents are expected to reach agreement on a Boolean function of various Boolean inputs) has been studied in [20].

Related approaches for transmitting information (as opposed to reaching consensus) in networks have also been extensively studied by the communications community under the moniker of *network coding* [4], [21]. Much of the work in this area focuses on the topic of sending

streams of information from a set of source nodes to a set of sink nodes in the network, and uses *finite (algebraic) fields* in order to deal with bandwidth constraints in the communication channels. More specifically, information is transmitted in packets consisting of a finite number of bits, and each group of bits is viewed as an element of a finite field, allowing ease of analysis [4], [22]. Linear network codes (where each node repeatedly sends a linear combination of incoming packets to its neighbors) can be viewed as linear systems over finite fields; by analyzing the transfer function of these systems using graph-theoretic concepts (such as the *Max-Flow-Min-Cut* theorem), linear network codes have been shown to achieve the maximum rate of transmission in multicast networks [4]. These concepts have also been extended to the problem of disseminating static initial values to some or all nodes via a gossip algorithm [22], [23]. For real-valued transmissions, [24], [25] showed that the problem of transmitting static initial values via a linear strategy is equivalent to the notion of linear system observability, and introduced *structured system theory* as a means of analyzing *linear dynamics* in networks.

Compelled by the fact that the multi-agent control and information dissemination problems are equivalent to the problems of controllability and observability in linear systems (over the field of complex numbers, and with appropriately defined nearest-neighbor rules) [7], [8], [9], [10], [11], [24], in this paper we ask the following question. Is it possible to maintain linear dynamics (and the paradigm of linear system controllability and observability) in multi-agent systems even when the state-space of the agents is constrained to be finite? Although this constraint precludes the use of analysis techniques for controllability of continuous-time continuous-state systems presented in previous works (as we will explain in further detail later in the paper), partly inspired by the finite-field paradigm adopted by the communications community, we show that linearity can be maintained by viewing the discrete states of the agents as elements of an appropriately chosen finite field. Specifically, we devise a nearest-neighbor rule whereby at each time-step, each agent updates its state to be a linear combination (over the finite field) of its state and those of its neighbors. We show that this approach allows the finite-state multi-agent system to be conveniently modeled as a *linear system over a finite field*. With this motivating insight, we start by considering the general problem of controllability of linear systems over finite fields, and develop a graph-theoretic characterization of controllability by extending existing theory on *structured linear systems* over the field of complex numbers to the finite-field domain. As we show, existing proof methods for structural controllability do not translate directly to

linear systems over finite fields, and thus we use a first-principles approach to establish this characterization.

Using the duality of control and estimation in linear systems, we also show how our results can be applied to the problem of quantized information dissemination in networks. In this setting, each agent is assumed to have an initial value in some finite set, and is only able to transmit or operate on values from that set. Certain "sink" agents wish to recover the initial values (or some function of them) by examining the transmissions of their neighbors over the course of the linear iterative strategy. Viewing the discrete set as a finite field, we show that the linear iterative strategy allows the sink agents to obtain the initial values of all other agents after at most N time-steps (where N is the number of agents in the network), provided that each agent has a path to at least one sink agent, and that the size of the discrete set is large enough. This guaranteed upper bound on accumulation time (in fixed and known networks) is a benefit of this strategy over the work on quantized consensus and gossip-based network coding, where the expected convergence time can be much larger[1] than the number of nodes in the network [19], [12], [22], [23].

The contributions of this paper are as follows. First, we develop a theory of structured linear systems over finite fields, providing graph-theoretic conditions for properties such as controllability and observability to hold. Second, we provide an improved upper bound on the generic controllability and observability indices of linear systems based on their graph representations. This characterization holds for standard linear systems over the field of complex numbers as well, and thus extends existing results on structured system theory [26]. Third, we introduce the notion of using finite fields as a means to represent finite state-spaces in multi-agent systems, and apply our results on structured system theory to analyze such systems.

II. Notation and Background

We use $\mathbf{e}_{i,l}$ to denote the column vector of length l with a "1" in its i-th position and "0" elsewhere, and $\mathbf{I}_N$ to denote the $N \times N$ identity matrix. The notation $\text{diag}\,(\cdot)$ indicates a block matrix with the diagonal blocks given by the quantities inside the brackets, and zero blocks

[1]It should be noted, however, that the higher cost in terms of convergence time is counter-balanced by the key benefit of gossip-based network coding, which is that it can operate in unknown and potentially time-varying networks.

elsewhere. The transpose of matrix $\mathbf{A}$ is denoted by $\mathbf{A}'$. The set of nonnegative integers is denoted by $\mathbb{N}$. For a sequence of vectors $\mathbf{u}[k]$, $k \in \mathbb{N}$, and two nonnegative integers k_1, k_2 with $k_2 \geq k_1$, we use $\mathbf{u}[k_1 : k_2]$ to denote $\begin{bmatrix}\mathbf{u}'[k_1] & \mathbf{u}'[k_1+1] & \cdots & \mathbf{u}'[k_2]\end{bmatrix}'$. We denote the cardinality of a set $\mathcal{S}$ by $|\mathcal{S}|$.

A. Graph Theory

A graph is an ordered pair $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ is a set of vertices, and $\mathcal{E}$ is a set of ordered pairs of different vertices, called directed edges.[2] The nodes in the set $\mathcal{N}_i = \{x_j | (x_j, x_i) \in \mathcal{E}\}$ are said to be neighbors of node x_i, and the in-degree of node x_i is denoted by $\deg_i = |\mathcal{N}_i|$. A *subgraph* of $\mathcal{G}$ is a graph $\mathcal{H} = \{\bar{\mathcal{X}}, \bar{\mathcal{E}}\}$, with $\bar{\mathcal{X}} \subseteq \mathcal{X}$ and $\bar{\mathcal{E}} \subseteq \mathcal{E}$ (where all edges in $\bar{\mathcal{E}}$ are between vertices in $\bar{\mathcal{X}}$).

A *path* P from vertex x_{i_0} to vertex x_{i_t} is a sequence of vertices $x_{i_0} x_{i_1} \cdots x_{i_t}$ such that $(x_{i_j}, x_{i_{j+1}}) \in \mathcal{E}$ for $0 \leq j \leq t-1$. The nonnegative integer t is called the length of the path. A graph is *strongly connected* if there is a path from every node to every other node. The *distance* between node x_j and node x_i is the length of the shortest path between node x_j and node x_i in the graph. A path is called a *cycle* if its start vertex and end vertex are the same, and no other vertex appears more than once in the path. A graph is called *acyclic* if it contains no cycles. A graph $\mathcal{G}$ is a *spanning tree rooted at* x_i if it is an acyclic graph where every node in the graph has a path from x_i, and every node except x_i has in-degree exactly equal to 1. The set of nodes with no outgoing edges are called the *leaves* of the tree. A *branch* of the tree is a subtree rooted at one of the neighbors of x_i. Similarly, a graph is a *spanning forest rooted at* $\mathcal{R} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ if it is a disjoint union of a set of trees, each of which is rooted at one of the vertices in $\mathcal{R}$. Examples of the above concepts are shown in Fig. 1. Analogously, a *spanning forest topped at* $\mathcal{R}$ is a forest that is obtained by reversing the direction of all edges in a spanning forest rooted at $\mathcal{R}$. In other words, all nodes in a spanning forest topped at $\mathcal{R}$ have a path to exactly one node in $\mathcal{R}$. Note that spanning forests in graphs can be easily found via a simple breadth- or depth-first search starting at any node in the root set, and proceeding through all other nodes in the root set until all nodes in the graph have been included.

[2]In this paper, we will be dealing with graphs with at most one edge from one vertex to another; later we will permit self-edges from a vertex to itself.

Definition 1: Let $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ denote a graph, and for any set $\mathcal{R} \subset \mathcal{X}$, consider a subgraph $\bar{\mathcal{H}}$ of $\mathcal{G}$ that is a spanning forest rooted at $\mathcal{R}$, with the property that the number of nodes in the largest tree in $\bar{\mathcal{H}}$ is minimal over all possible spanning forests rooted at $\mathcal{R}$. We call $\bar{\mathcal{H}}$ an *optimal spanning forest rooted at* $\mathcal{R}$. Similarly, an *optimal spanning forest topped at* $\mathcal{R}$ is obtained by reversing the directions of all edges in an optimal spanning forest rooted at $\mathcal{R}$. □

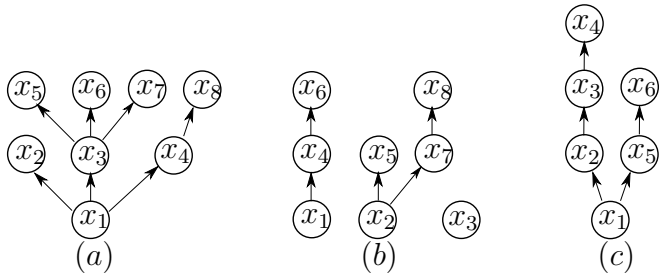

Fig. 1. (a) Spanning tree rooted at x_1. Nodes x_2, x_5, x_6, x_7 and x_8 are the leaves of the tree. The tree has three branches, consisting of the nodes $\{x_2\}, \{x_3, x_5, x_6, x_7\}$ and $\{x_4, x_8\}$. (b) A spanning forest rooted at $\{x_1, x_2, x_3\}$. (c) A spanning tree rooted at x_1 with two branches, both of which are paths.

B. Finite Fields

In this section, we briefly review the notion of a finite algebraic field. Further details can be found in standard texts, such as [27].

An algebraic *field* $\mathbb{F}$ is a set of elements, together with two operations written as addition ($+$) and multiplication[3] ($\times$), satisfying the following properties [28]:

1) Closure (i.e., $a + b \in \mathbb{F}$ and $ab \in \mathbb{F}$ for all $a, b \in \mathbb{F}$).
2) Commutativity (i.e., $a + b = b + a$ and $ab = ba$ for all $a, b \in \mathbb{F}$).
3) Associativity (i.e., $a + (b + c) = (a + b) + c$ and $a(bc) = (ab)c$ for all $a, b, c \in \mathbb{F}$).
4) Distributivity (i.e., $a(b + c) = ab + ac$ for all $a, b, c \in \mathbb{F}$).
5) The field contains an additive identity and a multiplicative identity, denoted 0 and 1, respectively (i.e., $a + 0 = a$ and $1a = a$ for all $a \in \mathbb{F}$).
6) For each element $a \in \mathbb{F}$, there is an additive inverse denoted by $-a \in \mathbb{F}$ such that $a + (-a) = 0$. Similarly, for each element $a \in \mathbb{F} \setminus \{0\}$, there is a multiplicative inverse denoted by $a^{-1} \in \mathbb{F}$ such that $aa^{-1} = 1$.

[3]We will denote the multiplicative operator by simply concatenating the operands (i.e., $a \times b$ is written as ab).

The closure property will play an important role in the scheme proposed in this paper. The number of elements in a field can be infinite (such as in the field of complex numbers), or finite. The finite field of size q is unique up to isomorphism, and is denoted by $\mathbb{F}_q$. When $q = p$ for some prime number p, the finite field $\mathbb{F}_p$ can be represented by the set of integers $\{0, 1, \ldots, p-1\}$, with addition and multiplication done modulo p. For example, the addition and multiplication tables for $\mathbb{F}_3 = \{0, 1, 2\}$ are given by

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\times$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

A key fact about finite fields is that they can only have sizes that are of the form $q = p^n$ for some prime p and positive integer n [27]. Every element of the field $\mathbb{F}_{p^n}$ can be represented by a polynomial[4] of degree $n-1$ in an arbitrary variable α, where each coefficient is an element of $\mathbb{F}_p$. Under this representation, addition or subtraction of two elements from $\mathbb{F}_{p^n}$ can be performed by adding or subtracting their polynomial representations, and reducing each of the coefficients modulo p. To multiply elements of $\mathbb{F}_{p^n}$, one first chooses an arbitrary polynomial $f(\alpha)$ of degree n that is *irreducible* over the field $\mathbb{F}_p$ (i.e., it does not factor into a product of polynomials of smaller degree over $\mathbb{F}_p$). For any elements $a, b \in \mathbb{F}_{p^n}$, the product ab is obtained by multiplying together their polynomial representations, reducing all coefficients modulo p, and then taking the remainder of the polynomial modulo $f(\alpha)$. This produces a new polynomial of degree $n-1$ or less with coefficients in $\mathbb{F}_p$, which corresponds to a unique element of $\mathbb{F}_{p^n}$.

III. Problem Formulation

Consider a network of agents (nodes) modeled by the directed graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ is the set of agents and the directed edge $(x_j, x_i) \in \mathcal{E}$ indicates that agent x_i can receive information from agent x_j. The state of each agent x_i is restricted to be an element from the finite set $\{0, 1, \ldots, q-1\}$, for some $q \in \mathbb{N}$. At each time-step k, each agent is allowed to update its state as a function of its previous state and those of its neighbors. We study two scenarios in this paper.

[4]For instance, any element of $\mathbb{F}_{2^3}$ can be represented by a polynomial $a_2\alpha^2 + a_1\alpha + a_0$, where $a_i \in \{0, 1\}$ for $i \in \{0, 1, 2\}$.

1) A set of leader agents $\mathcal{L} \subset \mathcal{X}$ wish to cooperatively update their states (within the confines of the discrete state-space) in order to make all of the other agents achieve a certain configuration (i.e., reach a certain state in $\{0, 1, \ldots, q-1\}^N$).
2) A set of sink agents $\mathcal{S} \subset \mathcal{X}$ wish to examine the state evolutions of their neighbors and use this information to collectively determine the initial states of all agents.

Throughout the paper, we will refer to the fact that the states of each agent must lie in the set $\{0, 1, \ldots, q-1\}$ as a *quantization constraint*. Thus, we will refer to the first scenario as the *Quantized Multi-Agent Control* (QMAC) problem, and the second scenario as the *Quantized Multi-Agent Estimation* (QMAE) problem.[5] We will provide a novel solution to these problems by viewing the discrete states of the agents as elements of a finite field, and performing all operations within that field.[6] Specifically, we assume (for now) that q is of the form p^n for some prime p and positive integer n, and treat the set $\{0, 1, \ldots, q-1\}$ as $\mathbb{F}_q$ (the finite field of size q); we will discuss generalizations of this later in the paper. To develop the theory, we will also assume that all agents have identical state-spaces, and that the network is fixed; we leave relaxations of these assumptions for future work.

To solve the above problems, we study a linear iterative strategy of the form

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij}x_j[k],$$

where $x_i[k]$ is the state of agent x_i at time-step k, and the w_{ij}'s are a set of weights[7] (constant elements) from the field $\mathbb{F}_q$. Note that *all operations in the above equation are done over the finite field* $\mathbb{F}_q$, guaranteeing that the state $x_i[k+1]$ will be in the set $\{0, 1, \ldots, q-1\}$ for all $k \in \mathbb{N}$ (by the closure property of finite fields). For ease of analysis, the states of all nodes at time-step k can be aggregated into the state vector $\mathbf{x}[k] = \begin{bmatrix} x_1[k] & x_2[k] & \cdots & x_N[k] \end{bmatrix}'$, so that

[5]Note that the QMAE problem can be viewed as an abstraction for the problem of *data accumulation* in networks [29]. For instance, the initial state of the agents can represent a certain piece of information (such as a temperature measurement, or a vote) which must be transmitted via the network to the sink agents. Similarly, the QMAC problem can be viewed as an abstraction for the problem of *broadcasting a different value to each agent* from the set of leaders.

[6]Note that this differs from the usual method of quantization, where all operations are first performed over the field of real numbers and then the result is mapped to the nearest quantization point. Instead, our approach will be to perform all operations within the confines of the discrete set; for convenience, we will use the term *quantization* as an allusion to the constrained nature of the system, keeping in mind the philosophical difference between the two methodologies.

[7]We will discuss appropriate ways to choose the weights later in the paper.

the nearest-neighbor update for the entire system can be represented as

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k], \tag{1}$$

for $k \in \mathbb{N}$, where entry (i,j) of matrix $\mathbf{W}$ is equal to the weight w_{ij} if $j \in \mathcal{N}_i$, the diagonal entries are equal to the self-weights w_{ii}, and all other entries are zero.

A. Quantized Multi-Agent Control Problem

Since each leader agent in the quantized multi-agent control problem is allowed to modify its state in arbitrary ways (subject to the quantization constraints), we can model this in the linear iterative strategy by simply including an "input" term[8] for each agent, i.e.,

$$x_l[k+1] = w_{ll}x_l[k] + \sum_{j \in \mathcal{N}_l} w_{lj}x_j[k] + u_l[k], \;\; x_l \in \mathcal{L} \;\; .$$

Letting $\mathcal{L} = \{x_{l_1}, x_{l_2}, \ldots, x_{l_{|\mathcal{L}|}}\}$, the system model (1) becomes

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k] + \underbrace{\begin{bmatrix}\mathbf{e}_{l_1,N} & \mathbf{e}_{l_2,N} & \ldots & \mathbf{e}_{l_{|\mathcal{L}|},N}\end{bmatrix}}_{\mathbf{B}_{\mathcal{L}}} \underbrace{\begin{bmatrix} u_{l_1}[k] \\ u_{l_2}[k] \\ \vdots \\ u_{l_{|\mathcal{L}|}}[k] \end{bmatrix}}_{\mathbf{u}_{\mathcal{L}}[k]} . \tag{2}$$

The explicit statement of the Quantized Multi-Agent Control Problem is as follows.

Problem 1: Find conditions on the network topology, a set of weights $w_{ij} \in \mathbb{F}_q$ (with the constraint that $w_{ij} = 0$ if $x_j \notin \mathcal{N}_i \cup \{x_i\}$), and a set of updates $\mathbf{u}_{\mathcal{L}}[k] \in \mathbb{F}_q^{|\mathcal{L}|}$, $k \in \mathbb{N}$, so that the state of the agents $\mathbf{x}[k]$ at some time-step k achieves some desired state $\bar{\mathbf{x}} \in \mathbb{F}_q^N$, starting from any given initial state $\mathbf{x}[0]$.

B. Quantized Multi-Agent Estimation Problem

Let $\mathbf{y}_s[k]$ denote the vector of states that sink agent $x_s \in \mathcal{S}$ views at the k–th time-step. Since x_s has access to its own state as well as the states of its neighbors, we can write

$$\mathbf{y}_s[k] = \mathbf{C}_s\mathbf{x}[k], \;\; x_s \in \mathcal{S} \;\; , \tag{3}$$

[8]We can leave the nearest-neighbor rule in the update for each leader without loss of generality because it can effectively be canceled out by choosing $u_l[k] \in \mathbb{F}_q$ appropriately.

where $\mathbf{C}_s$ is the $(\deg_s + 1) \times N$ matrix with a single "1" in each row denoting the positions of the vector $\mathbf{x}[k]$ that correspond to the neighbors of x_s, along with x_s itself. The overall system model for the Quantized Multi-Agent Estimation Problem is given by equations[9] (1) and (3).

The explicit statement of the Quantized Multi-Agent Estimation Problem is as follows.

Problem 2: Find conditions on the network topology, a set of weights $w_{ij} \in \mathbb{F}_q$ (with the constraint that $w_{ij} = 0$ if $x_j \notin \mathcal{N}_i \cup \{x_i\}$), and a strategy for the set $\mathcal{S}$ of sink nodes to follow so that they can collectively obtain the initial states of all of the other agents via $\{\mathbf{y}_s[0:L], s \in \mathcal{S}\}$, for some $L \in \mathbb{N}$.

C. Discussion

Problems 1 and 2 are precisely the notions of *controllability* and *observability* in linear systems, with the salient point being that we are working with systems over finite fields. In particular, we are interested in investigating how the *topology* of the network affects these properties, and therefore we will develop a graph-theoretic characterization of controllability and observability of linear systems over finite fields to solve Problems 1 and 2.

Note that the above problems assume that the nearest-neighbor rules are designed for the agents based on a given network. In other words, we allow different agents to potentially use different weights in their update; this is in contrast to previous work on controllability of multi-agent systems [7], [11], where each agent applies the *same* nearest-neighbor rule (i.e., with identical weights). Our approach will generally require more overhead to "setup" the system (by choosing the weights), but we will show that there is additional flexibility that is gained in return. We will first show how a network designer (with knowledge of the topology) can select a (deterministic) set of weights for each agent to apply, depending on the location of the agent in the network. We will then show how a *random* choice of weights can be used to solve the above problems; this will have several benefits, one of which is that the agents can choose their weights in a distributed manner (at the cost of requiring a larger number of states that each agent can occupy). We will discuss this issue further in Section VI.

Note also that the input sequence $u_l[k], x_l \in \mathcal{L}, k \in \mathbb{N}$ applied by leader x_l to solve Problem 1 will generally depend on the network topology, the weights w_{ij}, and the inputs applied by the

[9]One can also consider including input terms of the form in (2) into this model. If the inputs are known, their influence can readily be subtracted out from the values received by each sink agent.

other leaders. Knowledge of the network parameters will also be required by the sink nodes in order to solve Problem 2. Thus we will assume in this paper that the leaders and sink nodes know the matrix $\mathbf{W}$ and coordinate with each other to apply inputs, or recover the initial values, respectively. As we will discuss later in the paper, it is possible for the leaders and sink nodes to discover $\mathbf{W}$ via a distributed algorithm, under mild conditions on the network topology.

IV. Linear Systems Over Finite Fields

Consider a linear system of the form

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \ , \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k] \ , \end{aligned} \tag{4}$$

with state vector $\mathbf{x} \in \mathbb{F}^N$, input $\mathbf{u} \in \mathbb{F}^m$ and output $\mathbf{y} \in \mathbb{F}^r$ (for some field $\mathbb{F}$). The matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ (of appropriate sizes) also have entries from the field $\mathbb{F}$. Such systems have been extensively studied over several decades, both over the field of complex numbers by the control systems community (e.g., [30]), and over finite fields, particularly in the context of linear sequential circuits, convolutional error correcting codes and finite automata (e.g., [31], [32], [33], [34], [35], [36], [37], [38]). We will now review some important concepts for such systems, and explain how they differ from standard linear systems over the field of complex numbers. In the next section, we will use the intuition gained from this analysis to develop one of the key contributions of this paper, namely, a graph-theoretic characterization of controllability and observability for linear systems over finite fields.

Starting at some initial state $\mathbf{x}[0]$, the state of the system at time-step L (for some positive integer L) is given by

$$\mathbf{x}[L] = \mathbf{A}^L\mathbf{x}[0] + \underbrace{\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{L-1}\mathbf{B}\end{bmatrix}}_{\mathcal{C}_{L-1}} \mathbf{u}[0:L-1].$$

Similarly, when $\mathbf{u}[k] = 0$ for all k, the output of the system over L time-steps (for some positive integer L) is given by

$$\mathbf{y}[0:L-1] = \underbrace{\begin{bmatrix}\mathbf{C}' & (\mathbf{CA})' & \cdots & (\mathbf{CA}^{L-1})'\end{bmatrix}'}_{\mathcal{O}_{L-1}} \mathbf{x}[0].$$

If one wishes the state $\mathbf{x}[L]$ to be any arbitrary vector in $\mathbb{F}^N$, then one must ensure that the controllability matrix $\mathcal{C}_{L-1}$ has full rank over the field $\mathbb{F}$; in this case the pair $(\mathbf{A}, \mathbf{B})$ (or, more

loosely, the system) is said to be *controllable*.[10] Analogously, if one wishes to determine the initial state $\mathbf{x}[0]$ uniquely from the output of the system over L time-steps, one requires the observability matrix $\mathcal{O}_{L-1}$ to have rank N over the field $\mathbb{F}$, in which case the pair $(\mathbf{A}, \mathbf{C})$ (or the system) is said to be *observable*. Note that the ranks of $\mathcal{C}_{L-1}$ and $\mathcal{O}_{L-1}$ are nondecreasing functions of L, and bounded above by N. Suppose μ is the first integer for which $\text{rank}(\mathcal{C}_\mu) = \text{rank}(\mathcal{C}_{\mu-1})$. This implies that there exists a matrix $\mathbf{K}$ such that $\mathbf{A}^\mu\mathbf{B} = \mathcal{C}_{\mu-1}\mathbf{K}$. In turn, this implies that

$$\begin{aligned}\mathbf{A}^{\mu+1}\mathbf{B} = \mathbf{A}\mathbf{A}^\mu\mathbf{B} &= \mathbf{A}\mathcal{C}_{\mu-1}\mathbf{K} \\ &= \begin{bmatrix}\mathbf{AB} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^\mu\mathbf{B}\end{bmatrix}\mathbf{K} \ ,\end{aligned}$$

and so the matrix $\mathbf{A}^{\mu+1}\mathbf{B}$ can be written as a linear combination of the columns in $\mathcal{C}_\mu$. Continuing in this way, we see that the rank of $\mathcal{C}_L$ monotonically increases with L until $L = \mu - 1$, at which point it stops increasing. In the linear systems literature, the integer μ is called the *controllability index* of the pair $(\mathbf{A}, \mathbf{B})$. Similarly, the first integer ν for which $\text{rank}(\mathcal{O}_\nu) = \text{rank}(\mathcal{O}_{\nu-1})$ is called the *observability index* of the pair $(\mathbf{A}, \mathbf{C})$.

The above concepts and terminology hold regardless of the field $\mathbb{F}$ under consideration [39]. However, when one considers arbitrary fields, some of the further theory that has been developed to test controllability and observability of linear systems over the complex field will no longer hold. For example, consider the commonly used Popov-Belevitch-Hautus (PBH) test [30].

Theorem 1 (PBH Test): The pair $(\mathbf{A}, \mathbf{B})$ (over the field of complex numbers) is uncontrollable if and only if there exists a complex scalar λ_c such that $\text{rank}\begin{bmatrix}\lambda_c\mathbf{I}_N - \mathbf{A} & \mathbf{B}\end{bmatrix} < N$. The pair $(\mathbf{A}, \mathbf{C})$ (over the field of complex numbers) is unobservable if and only if there exists a complex scalar λ_o such that $\text{rank}\begin{bmatrix}\lambda_o\mathbf{I}_N - \mathbf{A} \\ \mathbf{C}\end{bmatrix} < N$. □

One might expect that this theorem will also apply to linear systems over finite fields, perhaps by taking the scalar λ to be an element of that field and then evaluating the rank of the resulting matrix over the field. However the following example shows that this is not necessarily the case.

Example 1: Consider the linear system operating over the finite field $\mathbb{F}_2 = \{0, 1\}$, with system matrices $\mathbf{A} = \begin{bmatrix}1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1\end{bmatrix}$, $\mathbf{B} = \mathbf{e}_{3,3}$. The controllability matrix for this system is $\mathcal{C}_{N-1} = \begin{bmatrix}0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1\end{bmatrix}$, which only has rank 1 over the field $\mathbb{F}_2$ (recall that multiplications and additions are performed

[10] If the system is controllable, an input (or control) sequence taking the system from any initial state $\mathbf{x}[0]$ to any desired final state $\mathbf{x}[L]$ can be found simply by solving the linear system of equations $\mathbf{x}[L] - \mathbf{A}^L\mathbf{x}[0] = \mathcal{C}_{L-1}\mathbf{u}[0 : L-1]$; note that this requires knowledge of the initial states of the system, as well as the controllability matrix $\mathcal{C}_L$.

modulo 2 in this field). However, the PBH matrix for this system is given by $\begin{bmatrix}\lambda\mathbf{I}_N - \mathbf{A} & \mathbf{B}\end{bmatrix} = \begin{bmatrix}\lambda{+}1 & 1 & 0 & 0\\ 1 & \lambda & 0 & 0\\ 0 & 0 & \lambda{+}1 & 1\end{bmatrix}$; note that $-1 = 1$ in $\mathbb{F}_2$. One can readily verify that the above matrix has full row rank (equal to 3) over $\mathbb{F}_2$ for any $\lambda \in \{0, 1\}$. In other words, the PBH condition is satisfied (over this field), but the system is clearly not controllable. The reason for the test failing in this case is that finite fields are not *algebraically closed*, which means that not every polynomial with coefficients from a finite field will have a root in that field (this also implies that not all $N \times N$ matrices in a finite field will have N eigenvalues) [40]. □

This PBH test plays a key role in much of the previous work on multi-agent controllability [7], [8], [9], [11]. It also features heavily in graph-theoretic characterizations of controllability and observability that have been developed in the structured linear systems literature [41], [26], [24]. However, since this test is not sufficient to treat linear systems over finite fields, we will now use a first-principles approach to derive a graph-theoretic characterization of controllability over finite fields.

V. Controllability of Structured Linear Systems Over Finite Fields

While much of linear system theory deals with systems with given (numerically specified) system matrices, there is frequently a need to analyze systems whose parameters are not exactly known, or where numerical computation of properties like controllability is not feasible. In response to this, control theorists have developed a characterization of system properties based on the *structure* of the system. Specifically, a linear system of the form (4) is said to be *structured* if every entry in the system matrices is either zero or an independent free parameter (traditionally taken to be real-valued) [26]. A property is said to hold structurally for the system if that property holds for at least one choice of free parameters. In fact, for real-valued parameters (with the underlying field of operation taken as the field of complex numbers), structural properties will hold generically (i.e., the set of parameters for which the property does not hold has Lebesgue measure zero). Previous works in this area typically rely on the PBH test to derive graph-theoretic characterizations of the controllability of structured systems (with real-valued parameters) [41], [26], [24], but as we have seen, such derivations do not directly extend to systems over finite fields.

To further illustrate the difference between structural controllability over finite fields and over the complex field, consider the pair $\mathbf{A} = \begin{bmatrix}0 & 0 & 0 & 0\\ a & b & 0 & 0\\ c & 0 & d & 0\\ e & 0 & 0 & f\end{bmatrix}$, $\mathbf{B} = \mathbf{e}_{1,4}$. The nonzero entries in $\mathbf{A}$

are independent free parameters, and thus $\mathbf{A}$ is a structured matrix. After some algebra, the controllability matrix $\mathcal{C}_3$ for this pair can be shown to have determinant $ace(f-d)(f-b)(d-b)$, and thus the system is structurally controllable if and only if a, c and e are nonzero, and b, d and f are all different. Clearly, one can satisfy this condition by choosing parameters from the field of complex numbers. However, one can also see that there does not exist any choice of parameters from the binary field $\mathbb{F}_2 = \{0, 1\}$ for which the system will be controllable (since this field only has two elements, at least two of b, d and f must be the same). Thus, this system is structurally controllable over $\mathbb{C}$, but not over $\mathbb{F}_2$.

In this section, we will develop a characterization of structural controllability over finite fields. We will start by investigating controllability of matrix pairs of the form $(\mathbf{A}, \mathbf{e}_{1,N})$, where $\mathbf{A}$ is an $N \times N$ matrix, and $\mathbf{e}_{1,N}$ is a column-vector of length N with a 1 in its first position and zeros elsewhere. Matrix $\mathbf{A}$ may be structured (i.e., every entry of $\mathbf{A}$ is either zero, or an independent free parameter to be chosen from a field $\mathbb{F}$), or it may be numerically specified. As in standard structured system theory [26], our analysis will be based on a graph representation of matrix $\mathbf{A}$, denoted by $\mathcal{H}$, which we obtain as follows. The vertex set of $\mathcal{H}$ is $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$, and the edge set is given by $\mathcal{E} = \{(x_j, x_i) \mid \mathbf{A}_{ij} \neq 0\}$. The weight on edge (x_j, x_i) is set to the value of $\mathbf{A}_{ij}$ (this can be a free parameter if $\mathbf{A}$ is a structured matrix).

A. Controllability of a Spanning Tree and Spanning Forest

Theorem 2: Consider the matrix pair $(\mathbf{A}, \mathbf{e}_{1,N})$, where $\mathbf{A}$ is an $N \times N$ matrix with elements from a field $\mathbb{F}$ of size at least N. Suppose that the following two conditions hold:

- The graph $\mathcal{H}$ associated with $\mathbf{A}$ is a spanning tree rooted at x_1, augmented with self-loops on every node.
- The weights on the self-loops are different elements of $\mathbb{F}$ for every node, and the weights on the edges between different nodes are equal to 1.

Then the pair $(\mathbf{A}, \mathbf{e}_{1,N})$ is controllable over the field $\mathbb{F}$, with controllability index equal to N.

□

Proof: Since the graph associated with $\mathbf{A}$ is a spanning tree rooted at x_1, there exists a numbering of the nodes such that the $\mathbf{A}$ matrix is lower-triangular, with the self-loop weights on the diagonal [42]. Denote the self-loop weight on node x_i by λ_i. Since all of the self-loop

weights are different, this matrix will have N distinct eigenvalues (given by $\lambda_1, \lambda_2, \ldots, \lambda_N$), with N corresponding linearly independent eigenvectors (this holds for arbitrary fields [43]).

Consider the eigenvalue λ_i. Let x_l be any leaf node in the graph such that the path from x_1 to x_l passes through x_i (if x_i is a leaf node, we can take $x_l = x_i$). Let N_i denote the number of nodes in this path, and reorder the nodes (leaving x_1 unchanged) so that all nodes on the path from x_1 to x_l come first in the ordering, and all other nodes come next. Let $\mathbf{P}_i$ denote the permutation matrix that corresponds to this reordering, and note that $\mathbf{P}_i \mathbf{A} \mathbf{P}_i'$ has the form

$$\mathbf{P}_i \mathbf{A} \mathbf{P}_i' = \begin{bmatrix} \mathbf{J}_i & \mathbf{0} \\ \bar{\mathbf{A}}_1 & \bar{\mathbf{A}}_2 \end{bmatrix} \quad , \tag{5}$$

for some matrices $\bar{\mathbf{A}}_1$ and $\bar{\mathbf{A}}_2$. The matrix $\mathbf{J}_i$ has the form $\mathbf{J}_i = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{N_i}) + \mathbf{S}_{N_i}$, where $\lambda_1, \lambda_2, \ldots, \lambda_{N_i}$ are different elements of $\mathbb{F}$, and $\mathbf{S}_{N_i}$ is an $N_i \times N_i$ matrix with ones on the main subdiagonal and zeros everywhere else. Note that there exists some $t \in \{1, 2, \ldots, N_i\}$ such that $\lambda_t = \lambda_i$ (where λ_i is the eigenvalue that we are considering in matrix $\mathbf{A}$). It is easy to verify that the left-eigenvector $\mathbf{v}_t$ of $\mathbf{J}_i$ associated with the eigenvalue λ_t is given by

$$\mathbf{v}_t = \Big[1 \quad (\lambda_t - \lambda_1) \quad (\lambda_t - \lambda_1)(\lambda_t - \lambda_2) \quad \cdots \quad \cdots \quad \prod_{s=1}^{t-1}(\lambda_t - \lambda_s) \quad 0 \quad \cdots \quad 0\Big] \quad ,$$

and thus the left-eigenvector corresponding to eigenvalue λ_t for the matrix $\mathbf{P}_i \mathbf{A} \mathbf{P}_i'$ in equation (5) is given by $\mathbf{w}_t = \Big[\mathbf{v}_t \quad \mathbf{0}\Big]$. Next, note that the left-eigenvector corresponding to eigenvalue λ_t (or equivalently, λ_i) for matrix $\mathbf{A}$ will be given by $\mathbf{w}_t \mathbf{P}_i$. Since $\mathbf{P}_i$ is a permutation matrix, and node x_1 was left unchanged during the permutation, the first column of $\mathbf{P}_i$ is given by the vector $\mathbf{e}_{1,N}$. This means that the first element of the eigenvector $\mathbf{w}_t \mathbf{P}_i$ will be "1" (based on the vectors $\mathbf{w}_t$ and $\mathbf{v}_t$ shown above). Since the above analysis holds for any eigenvalue λ_i, we can conclude that all left-eigenvectors for the matrix $\mathbf{A}$ will have a "1" as their first element. Let $\mathbf{V}$ be the matrix whose rows are these left-eigenvectors (so that each entry in the first column of $\mathbf{V}$ is "1"); since the eigenvectors are linearly independent, this matrix will be invertible over the field $\mathbb{F}$. We thus have $\mathbf{V}\mathbf{A}\mathbf{V}^{-1} = \Lambda$, where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$, and furthermore, $\mathbf{V}\mathbf{e}_{1,N} = \mathbf{1}_N$ (the length-N column vector of all 1's). The controllability matrix for the pair $(\Lambda, \mathbf{1}_N)$ is a *Vandermonde matrix* in the parameters $\lambda_1, \lambda_2, \ldots, \lambda_N$ [44]. Such matrices are invertible over a field $\mathbb{F}$ if and only if all of the parameters are distinct elements of that field [44], and thus the

above controllability matrix has rank N over $\mathbb{F}$. This means that the pair $(\mathbf{A}, \mathbf{e}_{1,N})$ will also be controllable.[11] Since the controllability matrix $\begin{bmatrix}\mathbf{e}_{1,N} & \mathbf{A}\mathbf{e}_{1,N} & \cdots & \mathbf{A}^{L-1}\mathbf{e}_{1,N}\end{bmatrix}$ only has L columns, it is obvious that the matrix obtains a rank of N at $L = N$. ■

Theorem 3: Consider the matrix pair $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A}$ is an $N \times N$ matrix with elements from a field $\mathbb{F}$, and $\mathbf{B}$ is a $N \times m$ matrix of the form $\mathbf{B} = \begin{bmatrix}\mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_m,N}\end{bmatrix}$. Suppose the graph $\mathcal{H}$ associated with the matrix $\mathbf{A}$ satisfies the following two conditions:

- The graph $\mathcal{H}$ is a spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$, augmented with self-loops on every node.
- No two nodes in the same tree have the same weight on their self-loops, and the weights on the edges between different nodes are equal to 1.

Let D denote the maximum number of nodes in any tree in $\mathcal{H}$. Then, the pair $(\mathbf{A}, \mathbf{B})$ is controllable over the field $\mathbb{F}$ with controllability index equal to D. □

The proof is directly obtained by noting that the system considered in the theorem corresponds to a set of decoupled subsystems, each of which is of the form described in Theorem 2.

B. Controllability of Arbitrary Graphs

Corollary 1: Consider the matrix pair $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A}$ is an $N \times N$ structured matrix, and $\mathbf{B}$ is a $N \times m$ matrix of the form $\mathbf{B} = \begin{bmatrix}\mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_m,N}\end{bmatrix}$. Suppose the graph $\mathcal{H}$ associated with $\mathbf{A}$ satisfies the following two conditions:

- Every node can be reached by a path from at least one node in the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$.
- Every node has a self-loop (i.e., the diagonal elements of $\mathbf{A}$ are free parameters).

Let $\bar{\mathcal{H}}$ be a subgraph of $\mathcal{H}$ that is a spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$. Let D denote the size of the largest tree in $\bar{\mathcal{H}}$. Then if $\mathbb{F}$ has size at least D, there exists a choice of parameters from $\mathbb{F}$ such that the controllability matrix corresponding to the pair $(\mathbf{A}, \mathbf{B})$ has rank N over that field, with controllability index equal to D. □

The proof of the above corollary is readily obtained by setting the values of all parameters corresponding to edges that are not in $\bar{\mathcal{H}}$ to zero, and then choosing the weights for edges in $\bar{\mathcal{H}}$ to satisfy Theorem 3. The strategy inherent in the above corollary is to decompose the system

[11]Here, we are using the well-known fact that the pair $(\mathbf{A}, \mathbf{e}_{i,N})$ is controllable if and only if the pair $(\mathbf{V}\mathbf{A}\mathbf{V}^{-1}, \mathbf{V}\mathbf{e}'_{1,N})$ is controllable, for any invertible matrix $\mathbf{V}$ [30]. It is easy to show that this fact also holds for arbitrary fields.

into a set of disjoint systems, each of which is controlled by a different input. This is intuitively appealing, and shows that one only needs a finite field of size D to control the system. Note that D will be no larger than $N-m+1$; this is because in the worst case, the spanning forest consists of one spanning tree (containing $N-(m-1)$ nodes) rooted at one node in $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ and $m-1$ isolated nodes (corresponding to the remaining nodes in the root set). By choosing $\bar{\mathcal{H}}$ to be an optimal spanning forest (see Definition 1 in Section II-A), one obtains the smallest value of D over all other choices of spanning forests.

C. Controllability using Arbitrary Fields

One can guarantee controllability of the pair $(\mathbf{A}, \mathbf{B})$ if the parameters of $\mathbf{A}$ are chosen from a finite field of size at least D, as long as the two conditions in Corollary 1 are satisfied. However, the following theorem shows that one can obtain controllability over finite fields of arbitrary size (including the binary field $\mathbb{F}_2 = \{0, 1\}$), if the graph of matrix $\mathbf{A}$ satisfies certain additional conditions. To maintain clarity, we will focus on the case where $\mathbf{B} = \mathbf{e}_{1,N}$ (i.e., a single input), but the result can be easily generalized to the case of multiple inputs, each of which is a root of a spanning tree of the form described in the theorem.

Theorem 4: Consider the matrix pair $(\mathbf{A}, \mathbf{e}_{1,N})$, where $\mathbf{A}$ is an $N \times N$ structured matrix. Suppose the graph $\mathcal{H}$ associated with $\mathbf{A}$ satisfies the following two conditions:

- $\mathcal{H}$ contains a subgraph that is a spanning tree rooted at x_1 with at most two branches.
- Each branch is a path, augmented with self-loops on every node.

Then, for any field $\mathbb{F}$, there is a set of parameters from $\mathbb{F}$ such that $(\mathbf{A}, \mathbf{e}_{1,N})$ is controllable. □

Proof: Consider the subgraph of $\mathcal{H}$ that is a spanning tree rooted at x_1 with at most two outgoing branches, both of which are paths. Set all of the weights corresponding to edges that are not in this spanning tree to zero, and set all edges between nodes in this spanning tree to 1. We will now describe how to choose the self-weights for the nodes.

Let $r-1$ denote the number of nodes in the first branch, and renumber the non-leader nodes so that the nodes in the first branch are $x_2, x_3, \ldots, x_r$, and the nodes in the second branch are $x_{r+1}, x_{r+2}, \ldots, x_N$. Set the self-weight w_{ii} for all nodes in the first branch (including x_1) to be 0, and the self-weight for all nodes in the second branch to be 1. The matrix $\mathbf{A}$ then has the form $\mathbf{A} = \left[\begin{smallmatrix} \mathbf{J}_0 & \mathbf{0} \\ \mathbf{F} & \mathbf{J}_1 \end{smallmatrix}\right]$, where $\mathbf{F} = \begin{bmatrix} \mathbf{e}_{1,N-r} & \mathbf{0} \end{bmatrix}$, $\mathbf{J}_0 = \mathbf{S}_r$, $\mathbf{J}_1 = \mathbf{I}_{N-r} + \mathbf{S}_{N-r}$, and $\mathbf{S}_j$ is a $j \times j$ matrix with ones on the main subdiagonal and zeros elsewhere.

Consider the matrix $\mathbf{P} = \left[\begin{smallmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{F} & \mathbf{J}_1 \end{smallmatrix}\right]$; note that this matrix is invertible over $\mathbb{F}_q$ since the matrix $\mathbf{J}_1$ is invertible over that field (it has determinant equal to 1). Also note that $\mathbf{F}\mathbf{J}_0 = \mathbf{0}$ (from the definition of these matrices given above). If we perform a similarity transformation on the pair $(\mathbf{A}, \mathbf{e}_{1,N})$ with $\mathbf{P}$, we obtain $\mathbf{P}\mathbf{A}\mathbf{P}^{-1} = \left[\begin{smallmatrix} \mathbf{J}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_1 \end{smallmatrix}\right]$ and $\mathbf{P}\mathbf{e}_{1,N} = \begin{bmatrix} \mathbf{e}'_{1,r} & \mathbf{e}'_{1,N-r} \end{bmatrix}'$. The controllability matrix for this transformed pair is

$$\begin{bmatrix} \mathbf{e}_{1,r} & \mathbf{J}_0\mathbf{e}_{1,r} & \mathbf{J}_0^2\mathbf{e}_{1,r} & \cdots & \mathbf{J}_0^{N-1}\mathbf{e}_{1,r} \\ \mathbf{e}_{1,N-r} & \mathbf{J}_1\mathbf{e}_{1,N-r} & \mathbf{J}_1^2\mathbf{e}_{1,N-r} & \cdots & \mathbf{J}_1^{N-1}\mathbf{e}_{1,N-r} \end{bmatrix}.$$

One can readily verify that for $\mathbf{J}_0$ as given above, we have $\begin{bmatrix} \mathbf{e}_{1,r} & \mathbf{J}_0\mathbf{e}_{1,r} & \mathbf{J}_0^2\mathbf{e}_{1,r} & \cdots & \mathbf{J}_0^{r-1}\mathbf{e}_{1,r} \end{bmatrix} = \mathbf{I}_r$ and $\mathbf{J}_0^k\mathbf{e}_{1,r} = \mathbf{0}$ for $k \geq r$. Thus, the above controllability matrix has the form $\left[\begin{smallmatrix} \mathbf{I}_r & \mathbf{0} \\ * & \mathbf{T} \end{smallmatrix}\right]$, where $*$ represents unimportant quantities and

$$\begin{aligned} \mathbf{T} &= \begin{bmatrix} \mathbf{J}_1^r\mathbf{e}_{1,N-r} & \mathbf{J}_1^{r+1}\mathbf{e}_{1,N-r} & \cdots & \mathbf{J}_1^{N-1}\mathbf{e}_{1,N-r} \end{bmatrix} \\ &= \mathbf{J}_1^r \underbrace{\begin{bmatrix} \mathbf{e}_{1,N-r} & \mathbf{J}_1\mathbf{e}_{1,N-r} & \cdots & \mathbf{J}_1^{N-r-1}\mathbf{e}_{1,N-r} \end{bmatrix}}_{\bar{\mathbf{T}}}. \end{aligned}$$

The matrix $\mathbf{J}_1^r$ is full rank (since $\mathbf{J}_1$ has determinant 1 over any field). One can also readily verify that the matrix $\bar{\mathbf{T}}$ is upper-triangular, with all diagonal entries equal to 1, and thus also has full rank over any field. Thus, the matrix $\mathbf{T}$ is invertible over the field $\mathbb{F}_q$, which means that the pair $(\mathbf{A}, \mathbf{e}_{1,N})$ is controllable over that field. ■

An example of the type of spanning tree discussed in the above theorem is shown in Fig. $1(c)$ (with self-loops omitted). The above theorem also encompasses topologies where the nodes are simply arranged in a path or a ring. For such systems, the proof of the theorem indicates that one only needs a field with elements "0" and "1" in order to ensure controllability – one simply finds the appropriate spanning tree, and assigns the self-loop parameters on one side of the tree to be "1", and the self-loop parameters on the other side to be "0". Note that the difference between Corollary 1 and Theorem 4 is that the latter focuses on graphs that contain a particular kind of spanning tree, but does not require a lower bound on the field size.

While we have been able to show that certain graph topologies can be controlled with finite fields of size smaller than D, the characterization of the smallest size required for controllability of arbitrary graphs (in terms of their topology) is an open problem for research.

D. Controllability with a Random Choice of Parameters

While the previous results allow us to obtain controllability over finite fields of relatively small size (no greater than D in Corollary 1 and of arbitrary size in Theorem 4), they require some manipulation of the system graph (i.e., to shape it into an appropriate tree or forest). We now consider what happens if we choose the parameters for the matrix randomly (i.e., uniformly and independently) from a field of sufficiently large size. The following theorem shows that this allows us to obtain controllability with high probability, and with a controllability index that is equal to (or better than) that provided by the optimal spanning forest. Furthermore, this will be achieved without requiring any detailed analysis of the graph (which will make it amenable to a decentralized implementation), but comes at the cost of working with finite fields of larger sizes. The proof of the theorem is provided in the Appendix.

Theorem 5: Consider the matrix pair $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A}$ is an $N \times N$ structured matrix, and $\mathbf{B}$ is a $N \times m$ matrix of the form $\mathbf{B} = \begin{bmatrix} \mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_m,N} \end{bmatrix}$. Suppose the graph $\mathcal{H}$ associated with the matrix $\mathbf{A}$ satisfies the following two properties:

- Every node can be reached by a path from at least one node in the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$.
- Every node has a self-loop (i.e., the diagonal elements of $\mathbf{A}$ are free parameters).

Let $\bar{\mathcal{H}}$ be a subgraph of $\mathcal{H}$ that is an optimal spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$. Let D denote the size of the largest tree in $\bar{\mathcal{H}}$. Then, if the free parameters in $\mathbf{A}$ are chosen uniformly and independently from the finite field $\mathbb{F}_q$ of size $q \geq (D-1)(N-m-\frac{D}{2}+1)$, then with probability at least $1 - \frac{1}{q}(D-1)(N-m-\frac{D}{2}+1)$, the pair $(\mathbf{A}, \mathbf{B})$ will be controllable and the controllability index will be upper bounded by D. □

Remark 1: While the lower bound on the field size specified by the above theorem is in terms of D, one does not actually need to know the value of D to apply it. More precisely, the quantity $(D-1)(N-m-\frac{D}{2}+1)$ is a concave function of D, and achieves its maximum value at $D = N-m+\frac{3}{2}$. However D is an integer and upper bounded by $N-m+1$, and substituting this into $(D-1)(N-m-\frac{D}{2}+1)$, we see that $\frac{(N-m)(N-m+1)}{2} \geq (D-1)(N-m-\frac{D}{2}+1)$. Thus, if one chooses entries from a field of size $q \geq \frac{(N-m)(N-m+1)}{2}$, one is guaranteed to obtain controllability with probability at least $1 - \frac{(N-m)(N-m+1)}{2q}$, without having to know the value of D. However, note that the controllability index of the resulting system will still be upper bounded by D (i.e., one can achieve the same, or better, controllability index as provided by the

optimal forest, without having to know anything about the forest). It is also worth noting that the upper bound of D on the controllability index also holds for systems over the field of real or complex numbers; the proof carries over almost directly, with the exception that one does not have to appeal to the Schwartz-Zippel Lemma (as we do for finite fields in the Appendix). Instead, the set of parameters for which the controllability index exceeds D lies on an algebraic variety, and thus has measure zero. In this case, the upper bound on the controllability index is *generic*, falling in line with the kinds of results that are typically obtained for structured systems over the field of complex and real numbers [26]. □

The above theorem has one apparent drawback in comparison to Corollary 1: the size of the field specified by this theorem is much larger than the size specified in the corollary. However, the theorem does provide some substantial benefits over the corollary. First, it does not require the network to be analyzed and processed (i.e., by decomposing the network into a spanning forest and setting non-tree edges to zero). Instead, each free parameter can simply be chosen randomly, and with high probability (with increasing q), the system will be controllable. Second, Theorem 5 shows that one can achieve a controllability index no greater than the one provided by the optimal spanning forest, without needing to actually find such a forest. In fact, as the following example shows, for certain topologies, one can obtain a controllability index that is strictly smaller than the one provided by the optimal forest.

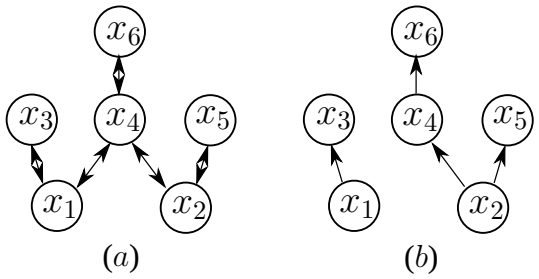

Fig. 2. (a) Graph of matrix $\mathbf{A}$ with self-loops omitted. (b) A subgraph of the original network that is an optimal spanning forest rooted at $\{x_1, x_2\}$.

Example 2: Consider the matrix pair $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A}$ is a structured matrix with all diagonal entries as free parameters and the remaining free parameters captured by the graph shown in Fig. 2(a), and $\mathbf{B} = \begin{bmatrix} \mathbf{e}_{1,6} & \mathbf{e}_{2,6} \end{bmatrix}$. An optimal spanning forest rooted at $\{x_1, x_2\}$ is shown in Fig. 2(b) (this forest is not unique), with $D = 4$. Corollary 1 indicates that by choosing the

parameters from a field of size $q \geq 4$ so that no two self-loops in the same tree have the same value, and by choosing the remaining parameters to adhere to the forest structure of Fig. $2(b)$, the system will be controllable with controllability index equal to 4.

Now let us consider a random choice of free parameters, in accordance with Theorem 5. Specifically, if each parameter is chosen uniformly and independently from a field of size $q \geq \frac{(N-m)(N-m+1)}{2} = 10$, the system will be controllable with probability at least $1 - \frac{10}{q}$. For example, with $q = 101$ (which is a prime number), we obtain a probability of success at least 0.91. One particular realization of random parameters from this field is $\mathbf{A} = \begin{bmatrix} 82 & 0 & 93 & 8 & 0 & 0 \\ 0 & 48 & 0 & 5 & 47 & 0 \\ 94 & 0 & 76 & 0 & 0 & 0 \\ 35 & 84 & 0 & 79 & 0 & 61 \\ 0 & 59 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 13 & 0 & 66 \end{bmatrix}$. One can verify that the controllability matrix $\mathcal{C}_2 = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} \end{bmatrix}$ for $(\mathbf{A}, \mathbf{B})$ has determinant 45 over the field $\mathbb{F}_{101}$ (recall that all multiplications and additions are done modulo 101 in this field). Thus, the system is controllable with a controllability index of 3, which outperforms the deterministic forest-based scheme described in Corollary 1. We will comment further on the implications of this observation in the next section. □

Choosing the parameters randomly also has benefits over a forest-based decomposition when one would like to obtain controllability from each input acting on its own. Specifically, the choice of weights indicated by Corollary 1 ensures that each input will control a subset of the states (i.e., those states that correspond to nodes in the tree rooted at that input), but will not influence any other states. However, in certain situations, it will be desirable for each input to be able to control all of the states on its own (e.g., when some of the inputs fail). The random choice of parameters described by Theorem 5 improves upon the forest-based decomposition of Corollary 1 because it maintains the ability of each input to affect more states than simply those in a particular tree. Starting with the bound provided by Theorem 5 (concerning the probability that the system will be controllable from all inputs acting *together*), one can obtain a bound on the probability that the system will be controllable from each input individually.

Theorem 6: Consider the matrix pair $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A}$ is an $N \times N$ structured matrix, and $\mathbf{B}$ is a $N \times m$ matrix of the form $\mathbf{B} = \begin{bmatrix} \mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_m,N} \end{bmatrix}$ for some distinct indices $\{i_1, i_2, \ldots, i_m\}$. Suppose the graph $\mathcal{H}$ associated with the matrix $\mathbf{A}$ satisfies the following two properties:

- Every node can be reached by a path from *each* node in the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$.
- Every node has a self-loop (i.e., the diagonal elements of $\mathbf{A}$ are free parameters).

If the free parameters in $\mathbf{A}$ are chosen uniformly and independently from the finite field $\mathbb{F}_q$ of size $q \geq \frac{mN(N-1)}{2}$, then with probability at least $1 - \frac{mN(N-1)}{2q}$, the pair $(\mathbf{A}, \mathbf{e}_{i_j,N})$ will be controllable for all $j \in \{1, 2, \ldots, m\}$. □

The proof is a straightforward consequence of Theorem 5 (with $\mathbf{B} = \mathbf{e}_{i_j,N}$), Remark 1 (with $m = 1$) and the union bound, and thus we omit the details here in the interest of space.

Remark 2: Note that the probability bounds obtained in Theorems 5 and 6 are potentially quite loose because of several conservative assumptions in our derivation, such as the union bound and the Schwartz-Zippel Lemma (which is used in the Appendix to derive Theorem 5). It may be the case that one can use finite fields of smaller sizes than those specified by the above theorems and still ensure that the system is controllable from every input with a specified probability (e.g., see [45] where an improvement on the Schwarz-Zippel Lemma is provided). A deeper investigation of the smallest field required to guarantee a certain probability of success (in terms of achieving controllability or observability) under a random choice of parameters is an open avenue for exploration. □

VI. Design of Nearest Neighbor Rules for the QMAC Problem

We now return to Problem 1 (the Quantized Multi-Agent Control Problem) stated in Section III. Recognizing that this is exactly a controllability problem over a finite field, we can immediately apply the theorems developed in the previous section. For the sake of pedagogy, we will demonstrate the application of Corollary 1. We will assume that each agent in the network can be reached by a path from at least one of the leader agents (if this is not true for some agent, it will be impossible for any leader to influence that particular agent). In the graph $\mathcal{G}$ of the network, let $\mathcal{H}$ be the subgraph of the network that is an optimal spanning forest rooted at the leader agents, and let D be the number of agents in the largest tree of that forest.

Theorem 7: Consider a multi-agent system with N agents described by the graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$. Let $\mathcal{L} \subset \mathcal{X}$ be a set of leader agents, and suppose that each agent in the network can be in one of q discrete states, where $q = p^n$ for some prime p and positive integer n. Then, if each agent in the network can be reached by a path from some leader agent, and if $q \geq D$, there is a set of weights $w_{ij} \in \mathbb{F}_q$, $j \in \mathcal{N}_i \cup \{x_i\}$, and a set of updates $\mathbf{u}_{\mathcal{L}}[k] \in \mathbb{F}_q^{|\mathcal{L}|}$, $k = 0, 1, \ldots, D-1$ in the nearest-neighbor rule (2) such that the state $\mathbf{x}[D]$ of the agents achieves any desired value starting from any initial condition $\mathbf{x}[0]$. □

Proof: First, note that the weight matrix $\mathbf{W}$ in (2) is a structured matrix (since every element is either identically zero or an independent free parameter). Since every agent in the network can be reached by a path from a leader agent, and since each agent has a "self-loop" (i.e., it can use its own current state in its update), we can appeal to Corollary 1. If the number of discrete states for each agent satisfies $q \geq D$, all of the conditions in this corollary are satisfied, and thus there exists a specific assignment of weights from $\mathbb{F}_q$ such that the pair $(\mathbf{W}, \mathbf{B}_{\mathcal{L}})$ in (2) is controllable over that field,[12] with controllability index D. Then, we have $\mathbf{x}[D] = \mathbf{W}^D\mathbf{x}[0] + \mathcal{C}_{D-1}\mathbf{u}_{\mathcal{L}}[0:D-1]$, and since we have shown that the matrix $\mathcal{C}_{D-1}$ has full rank over the field $\mathbb{F}_q$, the updates for the leader agents are

$$\mathbf{u}_{\mathcal{L}}[0:D-1] = \mathcal{C}_{D-1}^{\dagger}\left(\bar{\mathbf{x}} - \mathbf{W}^D\mathbf{x}[0]\right) \quad , \tag{6}$$

where $\bar{\mathbf{x}}$ is any desired vector in $\mathbb{F}_q^N$ and $\mathcal{C}_{D-1}^{\dagger}$ is any right-inverse of $\mathcal{C}_{D-1}$. Thus, all agents can be put into any desired configuration via the above set of updates by the leader, and by having all other agents follow nearest-neighbor rules with an appropriate set of updates. ■

Remark 3: The above control law only ensures that the state of all agents reaches the desired state at some time-step D. If one requires the agents to stay at that state for some period of time, then one can have all agents simply stop following the nearest neighbor rule after time-step D (assuming that all agents know the value of D). Alternatively, if for a given weight matrix $\mathbf{W}$, the desired state satisfies $\bar{\mathbf{x}} = (\mathbf{I}_N - \mathbf{W})^{-1}\mathbf{B}\bar{\mathbf{u}}$ for some vector $\bar{\mathbf{u}} \in \mathbb{R}^m$, then the leaders can maintain the state at $\bar{\mathbf{x}}$ by applying the input $\bar{\mathbf{u}}$ (as done in standard control problems). □

A. Controlling Agents When $q < D$

Note that Theorem 7 only provides a *sufficient* condition for multi-agent controllability over finite fields: as long as the number of states for each agent satisfies $q \geq D$, and the leaders have paths to every other node, then we can find a set of weights in $\mathbb{F}_q$ for each agent to use in its nearest-neighbor rule. However, there will often be cases where the number of possible states for each agent is less than D, preventing Theorem 7 from being applied directly. In these instances, one can appeal to Theorem 4 as long as the network topology satisfies the conditions

[12]Namely, the self-loop weight for each agent can be set to a different element of the field $\mathbb{F}_q$ and all other weights in the graph can be set to either 1 or 0 in order to obtain a spanning forest rooted at the leader nodes.

in that theorem. For such networks, there is no lower bound on the number of possible states for each agent (except for the constraint that $q = p^n$, which we will relax shortly).

Example 3: Consider a set of N agents arranged in a grid, where each agent can only be in one of two states, denoted by $\{0, 1\}$. For instance, these agents could represent a set of pixels that can only be white or black, or a set of cameras that can only face north or south. For any given agent x_i, the grid network contains a subgraph that is a spanning tree rooted at x_i with at most two branches, where each branch is a path; Theorem 4 indicates that the entire network can thus be controlled from x_i. Once the weights are chosen according to that theorem, the resulting matrix $\mathbf{W}$ can be used in equation (6) to produce the sequence of inputs to be applied by the leader in order to cause the other agents to enter any desired state in $\{0, 1\}^N$ (e.g., to make the cameras point in certain directions to ensure visual coverage of the environment, or to cause a picture to appear in the grid of pixels). □

The above analysis is worth comparing to previous works on multi-agent controllability in the continuous-time and continuous-state setting [7], [8], [10], [11]. These investigations have led to various characterizations of network topologies that are controllable by the leaders under a specific set of nearest-neighbor rules (e.g., when the dynamics of the overall system are given by the *Laplacian* of the graph [7]). The paper [7] showed that for these specific rules, certain topologies are controllable from a single leader, while others are not. The authors of [8] extended this result and showed that topologies that are "symmetric" from the perspective of the leader(s) are not controllable, the intuition being that both sides of the symmetric topology will be affected in exactly the same way by the leader nodes, and thus cannot be driven to different states. Recently, [11] further generalized the above studies on controllability of Laplacian dynamics in single-integrator networks by considering topological structures known as *equitable partitions*.

The Laplacian-based nearest-neighbor rules in those papers have the benefit of being uniform for all agents in the network (i.e., the behavior of each agent depends solely on the number of neighbors it has), but consequently does not break the symmetries in the network topology. In our setting, however, we are effectively breaking any symmetries by allowing different nodes to update their values in different ways, based on where they are in the network (i.e., they are allowed to have different weights in their nearest neighbor rules). In other words, we are *designing* the system in order to obtain controllability or observability from the leader and sink agents, whereas these previous works considered a given set of dynamics on networks, and then

analyzed those dynamics for controllability and observability. We will discuss a possible method for a distributed implementation of our scheme shortly.

B. Controlling Agents when $q \neq p^n$

Theorem 7 (or Theorem 4, when permitted by the network) is restricted to the case where the number of possible states for each agent is of the form $q = p^n$ for some prime p; this is due to the fact that finite fields only come in sizes of this form. However, one can also adapt the finite-field framework that we have developed to handle multi-agents systems where q is not of this form. First, factor q as $q = p_1^{n_1} p_2^{n_2} \cdots p_t^{n_t}$, where $p_1, p_2, \ldots, p_t$ are all distinct primes and $n_1, n_2, \ldots, n_t$ are positive integers. Now, each state $x_i[k] \in \{0, 1, \ldots, q-1\}$ can instead be viewed as a t-tuple of states $x_i[k] = (x_i^1[k], x_i^2[k], \ldots, x_i^t[k])$, where $x_i^j[k] \in \mathbb{F}_{p_j^{n_j}}$. Furthermore, the final desired state for each agent is also an element of $\prod_{j=1}^{t} \mathbb{F}_{p_j^{n_j}}$. Thus, each agent in the network can apply a linear iterative strategy of the form described in the previous sections (e.g., Theorem 7) to each element of the t-tuple corresponding to its state (performing operations over the field $\mathbb{F}_{p_j^{n_j}}$ for the j-th element), and can be placed in any desired state by the leader agents, as long as there is a path from the leaders to every other agent, and each $p_j^{n_j}$ is sufficiently large.

C. Controlling Agents with a Random Choice of Weights

Recall from Section V-D and Example 2 that a random choice of parameters for a structured system will potentially produce a smaller controllability index than a forest-based deterministic approach. This reveals that partitioning the agents into different sets, each of which is controlled by a different leader, might result in worse performance (in terms of time required to control the agents) than having all leaders control all of the agents together. A more detailed exploration of this phenomenon is a ripe area for future research.

It is also worth noting that a random choice of weights allows a completely distributed implementation of the linear iterative strategy. Specifically, suppose that there is no centralized designer to assign the weights in the network *a priori*, and to inform the leader agents of the controllability matrix to use in controlling the system (via (6)). In this case, if each agent in the network simply chooses the weights for itself and its incoming edges randomly (uniformly and independently) from a field of sufficiently large size, and if the network has the required paths from the leader agents, Theorem 5 indicates that the system will be controllable with high

probability. The agents can then use a simple flooding protocol [3], or alternatively, a variant of the distributed protocol described in [24],[13] in order to allow the leader agents to discover the weight matrix (and thus the controllability matrix). Note that this "discovery" phase has to be run only once in time-invariant networks, and the cost of executing this phase will be amortized over the number of times the linear iterative strategy is used to control the system.

A third benefit of a random choice of weights is that it can make the system fault-tolerant; to demonstrate this, suppose that up to f agents can fail, whereby they stop participating in the linear updates (i.e., they transmit the zero state to all their neighbors at each time-step). If the connectivity of the network is $f+1$ or higher (i.e., the graph remains strongly connected even when any f nodes are removed), then the leader agent(s) will still have a path to all other agents and the system will remain controllable with high probability. In this case, the fact that certain agents have dropped out of the network will need to be conveyed to the leaders via an appropriate mechanism, and this information can be used by the leaders to adjust their inputs appropriately. However, the remaining agents in the network will not need to change the weights that they have chosen for themselves (with high probability these weights will result in a full rank controllability matrix for the remaining correctly functioning nodes in the network). Note that we are assuming throughout that the leader agents are allowed to communicate with each other to apply the appropriate control inputs to the system.

VII. Design of Nearest Neighbor Rules for the QMAE Problem

We now turn our attention to Problem 2 (the Quantized Multi-Agent Estimation Problem) stated in Section III. As this is essentially an observability problem, we start by establishing graph-theoretic results for observability of structured linear systems over finite fields.

A. Observability of Structured Systems Over Finite Fields

Using the results in Section V, we can obtain equivalent results for observability of structured linear systems over finite fields based on the well known *duality* between control and estimation [30]. Specifically, since the pair $(\mathbf{A}, \mathbf{C})$ for the linear system (4) is observable if and only if

[13]The protocol in [24] was developed to allow nodes to distributively discover the observability matrix for linear iterative strategies with real-valued weights, but it also applies directly to linear iterative strategies over finite fields.

the pair $(\mathbf{A}', \mathbf{C}')$ is controllable, we can establish a graph-theoretic test for observability of the pair $(\mathbf{A}, \mathbf{C})$, where $\mathbf{A}$ is a structured matrix, by applying our results from Section V to the pair $(\mathbf{A}', \mathbf{C}')$. We can also make these tests more direct by noting that the graph associated with the matrix $\mathbf{A}'$ is simply the graph of matrix $\mathbf{A}$ with the directions of all edges reversed. Thus, we immediately obtain the following result (dual to Corollary 1).

Corollary 2: Consider the matrix pair $(\mathbf{A}, \mathbf{C})$, where $\mathbf{A}$ is an $N \times N$ structured matrix, and $\mathbf{C}$ is a $r \times N$ matrix of the form $\mathbf{C} = \begin{bmatrix} \mathbf{e}'_{i_1,N} & \mathbf{e}'_{i_2,N} & \cdots & \mathbf{e}'_{i_r,N} \end{bmatrix}'$. Suppose the graph $\mathcal{H}$ associated with $\mathbf{A}$ satisfies the following two conditions:

- Every node has a path to at least one node in the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_r}\}$.
- Every node has a self-loop (i.e., the diagonal elements of $\mathbf{A}$ are free parameters).

Let $\bar{\mathcal{H}}$ be a subgraph of $\mathcal{H}$ that is a spanning forest topped at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_r}\}$. Let D denote the size of the largest tree in $\bar{\mathcal{H}}$. Then if $\mathbb{F}$ has size at least D, there exists a choice of parameters from $\mathbb{F}$ such that the observability matrix corresponding to the pair $(\mathbf{A}, \mathbf{C})$ has rank N over that field, with observability index equal to D. □

One can immediately obtain the duals of Theorems 4, 5 and 6 in the same way.

B. Application to the Quantized Multi-Agent Estimation Problem

Theorem 8: Consider a multi-agent system with N agents described by the graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$. Let $\mathcal{S} \subset \mathcal{X}$ be a set of sink agents, and suppose that each agent in the network has an initial state from the set $\{0, 1, 2, \ldots, q-1\}$, where $q = p^n$ for some prime p and positive integer n. Also suppose every agent in the network has a path to some sink agent, and let D denote the size of the largest tree in some subgraph of $\mathcal{G}$ that is a spanning forest topped at $\bigcup_{x_s \in \mathcal{S}} \{\{x_s\} \cup \mathcal{N}_s\}$. Then, if $q \geq D$, there is a set of weights $w_{ij} \in \mathbb{F}_q$, with $w_{ij} = 0$ if $j \notin \mathcal{N}_i \cup \{x_i\}$, such that the sink agents can collectively determine the initial states of all agents after using the nearest-neighbor updates provided by (1) for D time-steps. □

Proof: First, note that the weight matrix $\mathbf{W}$ in (1) is a structured matrix (because every element is either identically zero or an independent free parameter). Since every agent in the network has a path to a sink agent, we can apply Corollary 2. Consider the spanning forest topped at $\bigcup_{x_s \in \mathcal{S}} \{\{x_s\} \cup \mathcal{N}_s\}$, and let D denote the number of agents in the largest tree. Then, from the field $\mathbb{F}_q$ (with $q \geq D$), assign the self-loop weights so that no two agents in the same tree have the same self-weight. Assign all other weights to be either 1 or 0, depending on whether

or not that edge appears in the forest. For each sink node $x_s \in \mathcal{S}$, let $\mathcal{T}_s \in \mathcal{X}$ correspond to all agents that are in trees topped at nodes in $\{x_s\} \cup \mathcal{N}_s$. If we rearrange the nodes $\mathcal{X}$ so that the nodes in $\mathcal{T}_s$ come first in the ordering, we obtain a weight matrix of the form $\mathbf{W} = \left[\begin{smallmatrix} \mathbf{W}_s & \mathbf{0} \\ \mathbf{0} & * \end{smallmatrix}\right]$, where $\mathbf{W}_s$ is a $|\mathcal{T}_s| \times |\mathcal{T}_s|$ submatrix containing the weights in the tree $\mathcal{T}_s$, and $*$ represents unimportant quantities. The matrix $\mathbf{C}_s$ in (3) for sink node x_s is of the form $\mathbf{C}_s = \begin{bmatrix} \bar{\mathbf{C}}_s & \mathbf{0} \end{bmatrix}$, where $\bar{\mathbf{C}}_s$ is a $(\deg_s + 1) \times |\mathcal{T}_s|$ matrix with a single "1" in each row denoting the nodes in $\mathcal{T}_s$ that are in $\{x_s\} \cup \mathcal{N}_s$; these are the nodes whose values are available to x_s at each time-step. Since the pair $(\mathbf{W}'_s, \bar{\mathbf{C}}'_s)$ satisfies the conditions in Theorem 3, the pair $(\mathbf{W}_s, \bar{\mathbf{C}}_s)$ is observable with observability index at most D. Thus sink node x_s can recover the initial values of all nodes in $\mathcal{T}_s$ after running the linear iteration for at most D time-steps. Specifically, the values seen by node x_s over those time-steps are given by $\mathbf{y}_s[0 : D-1] = \begin{bmatrix} \mathcal{O}_{s,D-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathcal{T}_s}[0] \\ \mathbf{x}_{\bar{\mathcal{T}}_s}[0] \end{bmatrix}$, where $\mathbf{x}_{\mathcal{T}_s}[0]$ denotes the vector of initial states of nodes in $\mathcal{T}_s$, $\mathbf{x}_{\bar{\mathcal{T}}_s}[0]$ denotes the vector of initial values of nodes not in $\mathcal{T}_s$, and $\mathcal{O}_{s,D-1}$ is the observability matrix for the pair $(\mathbf{W}_s, \bar{\mathbf{C}}_s)$. The latter matrix has full column rank (by the observability of the pair $(\mathbf{W}_s, \bar{\mathbf{C}}_s)$), and thus there exists a matrix Γ_s such that $\Gamma_s \mathcal{O}_{s,D-1} = \mathbf{I}_{|\mathcal{T}_s|}$. If node x_s left-multiplies the set of values $\mathbf{y}_s[0 : D-1]$ by Γ_s, it immediately obtains

$$\Gamma_s \mathbf{y}_s[0 : D-1] = \Gamma_s \begin{bmatrix} \mathcal{O}_{s,D-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathcal{T}_s}[0] \\ \mathbf{x}_{\bar{\mathcal{T}}_s}[0] \end{bmatrix} = \mathbf{x}_{\mathcal{T}_s}[0] \ .$$

The same analysis holds for all sink nodes in $\mathcal{S}$. Since every agent in the network belongs to some tree topped at a sink node or its neighbor, we see that all of the sink nodes together obtain the initial values of all agents in the network. ■

Remark 4: One can also obtain observability from the sink nodes (either collectively, or individually) via a random choice of free parameters by applying the duals of Theorems 5 and 6. The latter can be used to treat the problem of distributed consensus with quantized values that is commonly treated in the literature. Specifically, suppose $\mathcal{S} = \mathcal{X}$ (i.e., the set of sink agents is the set of all agents in the network), and assume that the network is strongly connected. Then, Theorem 6 states that if the weights in the nearest-neighbor rule (1) are chosen randomly (independently and uniformly) from a field of size $q \geq \frac{N^2(N-1)}{2}$, the resulting linear system will be observable from every agent (and its neighbors) with probability at least $1 - \frac{N^2(N-1)}{2q}$. Thus, each agent will be able to obtain the initial state of all of the other agents after a sufficiently large (but finite) number of time-steps, and therefore calculate any function of that state – if all

agents calculate the same function of the initial state, they will reach consensus. Of course, this assumes that the network is fixed and the nodes run a network discovery phase at initialization (as described in Section VI-C). These assumptions are certainly stricter than that required in other consensus protocols (e.g., [12], [16], [17], [13], [14], [15], [18], [19]), where the network can be time-varying and unknown. However, if satisfied, the assumptions in this paper allow the nodes to reach consensus on arbitrary functions of the initial values in a number of time-steps that will generally be smaller than that required by protocols that are fully agnostic of the network topology (as discussed in the Introduction). □

Remark 5: It is worth comparing the linear strategies studied in this paper to a simple flooding or broadcasting protocol. For instance, to place all agents in some desired state, the leader could transmit a vector of desired states to its neighbors, which subsequently set their states to the values denoted in the corresponding portion of the vector, and then pass the vector to their neighbors. After a number of time-steps equal to the largest distance from the leader to any other node, all agents will be in their desired positions. However, this scheme requires each agent to transmit a large amount of information to its neighbors at a given time-step (proportional to the number of agents in the network). If one restricts the amount of information that can be communicated per time-step (e.g., due to bandwidth constraints), the number of time-steps required will generally increase; a standard approach in this case is to decompose the network into a tree and schedule transmissions along the various branches [29]. In the extreme case where each node can only transmit a single value at each time-step, Example 2 shows that the linear strategy potentially allows the nodes to be placed in their desired states in a number of time-steps strictly smaller than that required by a tree-based scheme (corroborating analysis in the network coding literature showing that linear coding can potentially outperform simple routing schemes in terms of *rate* of information transmission [4]). Furthermore, the linear strategy also applies in cases where agents are not able to transmit information to each other, but instead directly sense each other's states. □

VIII. Summary

We showed how to formulate a linear iterative strategy for a network of quantized agents to follow so that they can be put into any desired configuration by a set of leader agents. We obtained this result by viewing the states of the agents as elements of a finite field, and then

developing a theory of structured system controllability (and observability) over these fields. For arbitrary topologies, we showed that the system will be controllable provided that the number of possible states for each agent is large enough, and that the leaders have a path to every agent. We also obtained similar results for the problem of estimating the initial states of the agents in the system by a set of sink agents. We provided both deterministic and randomized methods of choosing the weights in the linear iteration in order to obtain controllability and observability, and showed that random weights might outperform deterministic tree-based decompositions of the network in terms of the time required to control or estimate the system state.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their constructive comments and insights.

APPENDIX

To prove Theorem 5, we will make use of the following lemma that is commonly applied in the analysis of communication strategies over finite fields (e.g., see [45], [46]). In this lemma, the *total degree* of a multivariate polynomial $p(\xi_1, \xi_2, \ldots, \xi_n)$ is defined as the maximum sum of the degrees of the variables $\xi_1, \xi_2, \ldots, \xi_n$ in any term of the polynomial.

Lemma 1 (Schwartz-Zippel): Let $f(\xi_1, \xi_2, \ldots, \xi_n)$ be a nonzero polynomial of total degree d with coefficients in the finite field $\mathbb{F}_q$ (with $q \geq d$). If f is evaluated on an element $(s_1, s_2, \ldots, s_n)$ chosen uniformly and independently from $\mathbb{F}_q^n$, then $\Pr[f(s_1, s_2, \ldots, s_n) = 0] \leq \frac{d}{q}$. □

We will now prove Theorem 5.

Proof: Let the free parameters of matrix $\mathbf{A}$ be given by $\lambda_1, \lambda_2, \ldots, \lambda_l \in \mathbb{F}_q$. Note that these λ_i's now refer to all of the free parameters in $\mathbf{A}$ and not just the diagonal ones (as in the proof of Theorem 2). When convenient, we will aggregate these parameters into a vector $\lambda \in \mathbb{F}_q^l$. With this notation, the matrix $\mathbf{A}$ can also be denoted as $\mathbf{A}(\lambda)$ to explicitly show its dependence on the free parameters. Any particular choice of the free parameters will be denoted by $\lambda^* = \begin{bmatrix} \lambda_1^* & \lambda_2^* & \cdots & \lambda_l^* \end{bmatrix}$, with corresponding numerical matrix $\mathbf{A}(\lambda^*)$.

If the graph of matrix $\mathbf{A}$ satisfies the conditions in the theorem, then we know from Corollary 1 that there exists a choice of parameters $\lambda^* \in \mathbb{F}_q^l$ such that the controllability matrix

$$\mathcal{C}(\lambda^*)_{D-1} = \begin{bmatrix} \mathbf{B} & \mathbf{A}(\lambda^*)\mathbf{B} & \mathbf{A}^2(\lambda^*)\mathbf{B} & \cdots & \mathbf{A}^{D-1}(\lambda^*)\mathbf{B} \end{bmatrix}$$

has rank N over the field $\mathbb{F}_q$. This means that the controllability matrix $\mathcal{C}(\lambda^*)_{D-1}$ contains an $N \times N$ submatrix (denoted by $\mathbf{Z}(\lambda^*)$) whose determinant will be nonzero. Suppose (without loss of generality) that $\mathbf{Z}(\lambda^*)$ is constructed by taking the first N columns of $\mathcal{C}(\lambda^*)_{D-1}$ that form a linearly independent set. Note that the first m columns of $\mathbf{Z}(\lambda^*)$ are given by the matrix $\mathbf{B}$. Next, recall from Section IV that every set of columns of the form $\mathbf{A}^k(\lambda^*)\mathbf{B}$ must increase the rank of the controllability matrix by at least one. In the worst case, each such set of columns increases the rank by exactly one, and the last set of columns $\mathbf{A}^{D-1}(\lambda^*)\mathbf{B}$ contributes the rest of the rank. Thus, in the worst case, matrix $\mathbf{Z}(\lambda^*)$ has m columns from $\mathbf{B}$, one column from each of $\mathbf{A}(\lambda^*)\mathbf{B}, \mathbf{A}^2(\lambda^*)\mathbf{B}, \ldots, \mathbf{A}^{D-2}(\lambda^*)\mathbf{B}$, and $N - m - (D-2)$ columns from $\mathbf{A}^{D-1}(\lambda^*)\mathbf{B}$.

Next, consider the matrix $\mathbf{Z}(\lambda)$ (which is obtained by reverting the special choice of parameters λ^* back to the original symbolic parameters). The determinant of $\mathbf{Z}(\lambda)$ will therefore be a nonzero polynomial in these parameters (since this polynomial is nonzero after a specific choice of parameters). Specifically, the determinant of an $N \times N$ matrix is a sum of products, where no two entries in any product come from the same row or column of that matrix. Thus, each product in $\det \mathbf{Z}(\lambda)$ will consist of an entry from each column of $\mathbf{Z}(\lambda)$. Now, note that the matrix $\mathbf{A}^k(\lambda)\mathbf{B}$ is simply a set of columns from $\mathbf{A}^k(\lambda)$ (from the form of $\mathbf{B}$), and recall that entry (i,j) in $\mathbf{A}^k(\lambda)$ is a polynomial in λ where every term corresponds to the product of weights on a path of length k from node x_j to x_i (i.e., every term is a product of k parameters from λ) [42]. Since $\mathbf{Z}(\lambda)$ consists of at least one column from matrices of the form $\mathbf{A}(\lambda)\mathbf{B}, \mathbf{A}^2(\lambda)\mathbf{B}, \ldots, \mathbf{A}^{D-2}(\lambda)\mathbf{B}$, and at most $N-m-D+2$ columns from $\mathbf{A}^{D-1}(\lambda)\mathbf{B}$, we see that each term in $\det \mathbf{Z}(\lambda)$ will be a product of at most $1+2+\cdots+D-2+(N-m-D+2)(D-1) = (D-1)\left(N - m - \frac{D}{2} + 1\right)$ free parameters. Thus, $\det \mathbf{Z}(\lambda)$ is a polynomial of total degree no greater than $(D-1)\left(N - m - \frac{D}{2} + 1\right)$ and, as noted earlier, it is not identically zero over the field $\mathbb{F}_q$. Using Lemma 1, we see that for a random (uniform and independent) choice of parameters λ^* from the field $\mathbb{F}_q$ of size $q \geq (D-1)(N - m - \frac{D}{2} + 1)$, the polynomial $\det \mathbf{Z}(\lambda^*)$ will be nonzero with probability at least $1 - \frac{1}{q}(D-1)(N - m - \frac{D}{2} + 1)$. Thus the controllability matrix will also have full rank with at least this probability, and with controllability index no more than D. ■

REFERENCES

[1] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[3] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., 1996.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[5] A. Giridhar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, Apr. 2006.

[6] N. H. Packard and S. Wolfram, "Two-dimensional cellular automata," *Journal of Statistical Physics*, vol. 38, no. 5-6, pp. 901–946, March 1985.

[7] H. Tanner, "On the controllability of nearest neighbor interconnections," in *Proc. 43rd IEEE Conference on Decision and Control*, 2004, pp. 2467–2472.

[8] A. Rahmani and M. Mesbahi, "On the controlled agreement problem," in *Proc. American Control Conference*, 2006, pp. 1376–1381.

[9] S. Martini, M. Egerstedt, and A. Bicchi, "Controllability decompositions of networked systems through quotient graphs," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 5244–5249.

[10] R. Lozano, M. W. Spong, J. A. Guerrero, and N. Chopra, "Controllability and observability of leader-based multi-agent systems," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 3713–3718.

[11] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, Feb. 2009.

[12] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, July 2007.

[13] A. Savkin, "Coordinated collective motion of groups of autonomous mobile robots: analysis of Vicsek's model," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 981–983, June 2004.

[14] T. Aysal, M. Coates, and M. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4905–4918, Oct. 2008.

[15] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus by gossip algorithms with quantized communication," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 4837–4842.

[16] F. Benezit, P. Thiran, and M. Vetterli, "Interval consensus: from quantized gossip to voting," in *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3661–3664.

[17] M. Draief and M. Vojnovic, "Convergence speed of binary interval consensus," in *Proceedings of the IEEE Conference on Computer Communications*, 2010, pp. 1–9.

[18] M. E. Yildiz and A. Scaglione, "Differential nested lattice encoding for consensus problems," in *Proc. 6th International Conference on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 89–98.

[19] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 4825–4830.

[20] A. Fagiolini, E. M. Visibelli, and A. Bicchi, "Logical consensus for distributed network agreement," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 5250–5255.

[21] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory," in *Foundations and Trends in Communications and Information Theory*. Now Publishers Inc., 2005, vol. 2, no. 4/5, pp. 241–381.

[22] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2486–2507, June 2006.

[23] D. Mosk-Aoyama and D. Shah, "Information dissemination via network coding," in *Proc. 2006 IEEE International Symposium on Information Theory*, 2006, pp. 1748–1752.

[24] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications: Special Issue on Control and Communications*, vol. 26, no. 4, pp. 650–660, May 2008.

[25] ——, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Automatic Control*, vol. 56, no. 7, pp. 1495–1508, July 2011.

[26] J.-M. Dion, C. Commault, and J. van der Woude, "Generic properties and control of linear structured systems: a survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, July 2003.

[27] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, 2nd ed. Cambridge University Press, 1994.

[28] A. Papantonopoulou, *Algebra: Pure and Applied*. Prentice Hall, 2002.

[29] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger, *Dissemination of Information in Communication Networks*. Berlin, Germany: Springer-Verlag, 2005.

[30] P. J. Antsaklis and A. N. Michel, *Linear Systems*. Birkhauser Boston, 2006.

[31] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Transactions on Computers*, vol. C-17, no. 4, pp. 330–337, April 1968.

[32] G. Forney, "Convolutional codes I: Algebraic structure," *IEEE Transactions on Information Theory*, vol. 16, no. 6, pp. 720–738, Nov. 1970.

[33] M. Kociecki and K. M. Przyluski, "On the number of controllable linear systems over a finite field," *Linear Algebra and its Applications*, vol. 122-124, pp. 115–122, 1989.

[34] J. Reger and K. Schmidt, "Aspects on analysis and synthesis of linear discrete systems over the finite field GF(q)," in *Proc. European Control Conference*, 2003.

[35] J. Reger, "Linear systems over finite fields – modeling, analysis and synthesis," Ph.D. dissertation, University of Erlangen-Nuremberg, 2004.

[36] J. Rosenthal, "Connections between linear systems and convolutional codes," in *Codes, Systems and Graphical Models*, B. Marcus and J. Rosenthal, Eds. Springer-Verlag, 2001, vol. 123, pp. 39–66.

[37] C. N. Hadjicostis, "Non-concurrent error detection and correction in fault-tolerant linear finite-state machines," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2133–2140, Dec. 2003.

[38] R. Grosu, "Finite automata as time-inv linear systems: Observability, reachability and more," in *Hybrid Systems: Computation and Control*, R. Majumdar and P. Tabuada, Eds. Springer-Verlag, 2009, vol. LNCS 5469, pp. 194–208.

[39] R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in Mathematical System Theory*. McGraw-Hill, 1969.

[40] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[41] K. J. Reinschke, *Multivariable Control: A Graph-Theoretic Approach*. Springer-Verlag, 1988.

[42] D. B. West, *Introduction to Graph Theory*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 2001.

[43] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[44] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge University Press, 2003.

[45] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. 41st Allerton Conference on Communications, Control and Computing*, 2003.

[46] D. C. Kozen, *Theory of Computation*. Springer-Verlag London Ltd., 2006.