

# Control of Quantized Multi-Agent Systems with Linear Nearest Neighbor Rules: A Finite Field Approach

Shreyas Sundaram and Christoforos N. Hadjicostis

**Abstract**— We study the problem of controlling a multi-agent system where each agent is only allowed to be in a discrete and finite set of states. Each agent is capable of updating its state based on the states of its neighbors, and there is a leader agent in the network that is allowed to update its state in arbitrary ways (within the discrete set) in order to put all agents in a desired state. We present a novel solution to this problem by viewing the discrete states of the system as elements of a finite field. Specifically, we develop a theory of structured linear systems over finite fields, and show that such systems will be controllable provided that the size of the finite field is sufficiently large, and that the graph associated with the system satisfies certain properties. We then use these results to show that a multi-agent system with a leader node is controllable via a linear nearest-neighbor update as long as there is a path from the leader to every node, and that the number of discrete states for each node is large enough.

## I. INTRODUCTION

Multi-agent and -robot systems hold great promise in a variety of applications [1], [2], and for this reason, a tremendous amount of research has gone into the problems of controlling and coordinating such systems [3], [4]. These distributed systems have no coordinator who is able to command all agents directly, and thus the agents must rely on interactions with their neighbors in order to achieve the overall objective. For example, the topic of *distributed consensus* has received a great deal of attention, where all of the agents are expected to converge to a common decision (or *value*) after repeated interactions with their neighbors (e.g., see [4] and the references therein). The topic of *controlling* multi-agent systems with one or multiple leader agents has also attracted attention over the past few years; the objective in this case is to cause all of the agents to be in some desired state via a set of actions by the leader agents [5], [6], [7], [8], [9]. These investigations (which consider continuous-time systems with continuous state-spaces) have led to various characterizations of network topologies that are controllable

This material is based upon work supported in part by the National Science Foundation under NSF ITR Award 0426831 and 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 INFOS-ICT-223844 and PIRG02-GA-2007-224877. 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 or EC.

S. Sundaram is with the GRASP Laboratory and the Department of Electrical and Systems Engineering, University of Pennsylvania, USA. E-mail: ssund@seas.upenn.edu. C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of 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.

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 [5]). The papers [5], [6], [7] showed that topologies that are “symmetric” from the perspective of the leader(s) are not controllable, and developed control strategies for multi-agent systems with other topologies.

As the field of distributed control in multi-agent systems matures, researchers have started to investigate the problem of *quantization*, where the agents in the network can only exchange a finite number of bits with their neighbors, or can only occupy a fixed number of states [10], [11], [12]. These works have revealed that nearest neighbor rules can be adapted in various ways in order to reach consensus 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 they bring their values as close together as possible) [11], to incorporating quantization steps into (otherwise) linear update strategies for each agent [10]. Along similar lines, the topic of logical consensus (where agents are expected to reach agreement on certain Boolean functions of various Boolean inputs) has been studied in [13].

In contrast to the above works that focus on quantized consensus, we examine the problem of *controlling* a multi-agent system where the state-space of each agent is assumed to lie in a discrete and finite set (e.g., as would be the case with a finite number of quantization levels). We provide a novel solution to this problem by showing that multi-agent systems with discrete and finite state-spaces can be conveniently modeled as linear systems over *finite fields*. We then develop a characterization of the controllability of linear systems over finite fields based on a graph representation of the system. We use this to show that as long as each agent has a sufficiently large number of states that it can be in, and as long as the leader has a path to every node in the network, the leader will be able to put all agents in any desired state via a set of linear nearest neighbor rules.

## 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. The symbol  $\mathbf{1}_l$  denotes the column vector of length  $l$  with all entries equal to “1”, and  $\mathbf{I}_N$  denotes the  $N \times N$  identity matrix. We will also denote the cardinality of a set  $\mathcal{S}$  by  $|\mathcal{S}|$ , and use the notation  $\text{diag}(\cdot)$  to indicate a square matrix with the quantities inside the brackets on the diagonal, and zeros

elsewhere. The transpose of matrix  $\mathbf{A}$  is denoted by  $\mathbf{A}'$ . The set of nonnegative integers is denoted by  $\mathbb{N}$ .

*Graph Theory:* A graph is an ordered pair  $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ , where  $\mathcal{X} = \{x_1, \dots, x_N\}$  is a set of vertices, and  $\mathcal{E}$  is a set of ordered pairs of different vertices, called directed edges. 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  $x_{i_0}x_{i_1}\dots x_{i_t}$  such that  $(x_{i_j}, x_{i_{j+1}}) \in \mathcal{E}$  for  $0 \leq j \leq t-1$ . 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 originating 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 that have no outgoing edges are called the *leaves* of the tree. A *branch* of the tree is a subtree originating at one of the neighbors of  $x_i$ . Examples of spanning trees are shown in Figure 1. Further background on graph theory can be found in standard texts, such as [14].

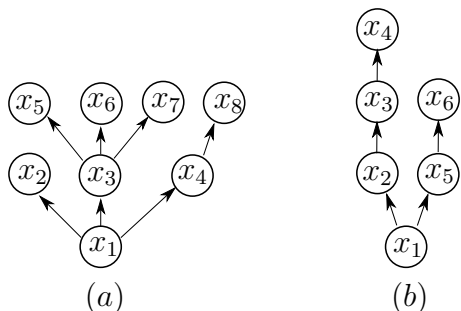


Fig. 1. (a) Spanning tree originating at  $x_1$ . Nodes  $x_2, x_5, x_6, x_7$  and  $x_8$  are the leaves of the tree. The tree has three branches, consisting of the nodes  $\{x_2\}$ ,  $\{x_3, x_5, x_6, x_7\}$  and  $\{x_4, x_8\}$ . (b) Spanning tree originating at  $x_1$  with two branches, both of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

*Finite Fields:* A *field*  $\mathbb{F}$  is a set of elements, together with the operations of addition and multiplication defined over those elements. These operations are associative, commutative, and satisfy the distributive laws. Addition is invertible with identity “0”, and multiplication is invertible with identity “1” (if the additive identity “0” is excluded). Fields are *closed*, meaning that addition or multiplication of two elements produces another element in that field. The number of elements in a field can be infinite (such as in the field of complex numbers), or finite. However, finite fields only come in sizes that are powers of a prime, i.e., of the form  $q = p^n$  for some prime  $p$  and positive integer  $n$ . The finite field of size  $q$  is unique (up to isomorphism) and is denoted by  $\mathbb{F}_q$ .

Every element of the finite field  $\mathbb{F}_{p^n}$  can be represented by a polynomial of degree  $n-1$ , with coefficients taking on one of  $p$  different values (which we denote by the integers  $\{0, 1, 2, \dots, p-1\}$ ). Addition or subtraction of two elements from the field can be performed by adding or subtracting

their polynomial representations, and reducing each of the coefficients modulo  $p$ . Multiplication of elements can be performed by multiplying their polynomials, and then taking the remainder modulo an irreducible polynomial over that field; the details can be found in texts such as [15]. Note that when  $n = 1$ , addition and multiplication in  $\mathbb{F}_p$  reduce to simply adding or multiplying integers modulo  $p$ .

### III. MULTI-AGENT COORDINATION VIA NEAREST NEIGHBOR RULES

Consider a network of agents modeled by the directed graph  $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ , where  $\mathcal{X} = \{x_1, \dots, x_N\}$  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  $x_j$ . Each agent  $x_i$  has a certain discrete and finite set of states that it can be in, and we will denote these states by the set  $\{0, 1, \dots, q-1\}$ . For now, we will assume that  $q$  is of the form  $p^n$  for some prime  $p$  and positive integer  $n$ , and thus treat the discrete states as elements of the finite field  $\mathbb{F}_q$ ; we will discuss generalizations of this later in the paper. We will assume that the possible state-space for all agents is identical, and that the network is fixed. At each time-step  $k$ , each agent in the network is allowed to update its state as a function of its previous state and those of its neighbors. The agent  $x_1$  is taken to be the leader in the network (without loss of generality), and it can update its state in arbitrary ways (within the confines of the discrete state-space) in order to make all of the other agents achieve a certain configuration. We will assume a single leader in this paper to maintain clarity, but the ideas can be extended to multiple leaders acting cooperatively in order to control the network.

We will investigate linear nearest neighbor rules 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 (constant elements) from the field  $\mathbb{F}_q$ . Due to the closure property of a finite field, this update rule guarantees that the state  $x_i[k+1]$  will be in the set  $\{0, 1, \dots, q-1\}$ . Since the leader agent is allowed to modify its state in arbitrary ways, we can model this by simply including an “input” term<sup>1</sup> for  $x_1$ , i.e.,

$$x_1[k+1] = w_{11}x_1[k] + \sum_{j \in \mathcal{N}_1} w_{1j}x_j[k] + u[k].$$

For ease of analysis, the states of all nodes at time-step  $k$  can be aggregated into the state vector  $\mathbf{x}[k] = [x_1[k] \ x_2[k] \ \dots \ x_N[k]]'$ , so that

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k] + \mathbf{e}_{1,N}u[k] \quad (1)$$

for  $k \in \mathbb{N}$ , where the  $(i, j)$  entry of  $\mathbf{W}$  is the weight  $w_{ij}$  if  $x_j \in \mathcal{N}_i$ , and zero otherwise.

*Problem Formulation:* Find conditions on the network topology, a set of weights  $w_{ij} \in \mathbb{F}_q$  (with  $w_{ij} = 0$  if  $x_j \notin$

<sup>1</sup>We leave the nearest neighbor rule in the update for the leader without loss of generality, because it can effectively be canceled out by choosing  $u[k]$  appropriately.

$\mathcal{N}_i$ ), and a set of updates  $u[k] \in \mathbb{F}_q$ ,  $k \in \mathbb{N}$ , 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  $\mathbf{x}[0]$ .

The above problem statement is evocative of the issue of *controllability* in multi-agent systems (and, more generally, linear systems), with the salient difference being that we are working with systems over finite fields. We will explore the implications of this in the next few sections.

#### IV. LINEAR SYSTEMS OVER FINITE FIELDS

Consider a linear system of the form

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k], \quad (2)$$

with state vector  $\mathbf{x} \in \mathbb{F}^N$  and input  $\mathbf{u} \in \mathbb{F}^m$  (for some field  $\mathbb{F}$ ). The matrices  $\mathbf{A}$  and  $\mathbf{B}$  (of appropriate sizes) have entries from the field  $\mathbb{F}$ . 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

$$\mathbf{x}[L] = \mathbf{A}^L \mathbf{x}[0] + \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \dots & \mathbf{A}^{L-1}\mathbf{B} \end{bmatrix}}_{\mathcal{C}_{L-1}} \underbrace{\begin{bmatrix} \mathbf{u}[L-1] \\ \mathbf{u}[L-2] \\ \vdots \\ \mathbf{u}[0] \end{bmatrix}}_{\mathbf{u}[0:L-1]}.$$

If one wishes the state  $\mathbf{x}[L]$  to be any arbitrary vector in  $\mathbb{F}^N$ , then one must ensure that the matrix  $\mathcal{C}_{L-1}$  has full rank; in this case the system is said to be *controllable*. The matrix  $\mathcal{C}_{N-1}$  is termed the *controllability matrix* for the pair  $(\mathbf{A}, \mathbf{B})$ . Note that the rank of  $\mathcal{C}_{L-1}$  is a nondecreasing function of  $L$ , and bounded above by  $N$ . The above concepts and terminology are all completely identical to the case when the underlying field is  $\mathbb{F} = \mathbb{C}$  (i.e., the field of complex numbers) that is commonly studied in linear system theory [16]. However, when one considers arbitrary fields, some of the further theory that has been developed to test controllability of linear systems over the complex field will no longer hold. For example, consider the commonly used Popov-Belevitch-Hautus (PBH) test (e.g., see [16]).

*Theorem 1 (PBH Test):* The pair  $(\mathbf{A}, \mathbf{B})$  (over the field of complex numbers) is controllable if and only if  $\text{rank}[\lambda \mathbf{I}_N - \mathbf{A} \quad \mathbf{B}] = N$  for all  $\lambda \in \mathbb{C}$ .  $\square$

One might expect that the above theorem will also apply to linear systems over finite fields, perhaps by taking the scalar  $\lambda$  to be an element of that field and then evaluating the rank of the resulting matrix over the field. However, as the following example illustrates, this is not 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 modulo 2 in this field). However, the PBH matrix for this system is given by  $[\lambda \mathbf{I}_N - \mathbf{A} \quad \mathbf{B}] = \begin{bmatrix} \lambda+1 & 1 & 0 & 0 \\ 1 & \lambda & 0 & 0 \\ 0 & 0 & \lambda+1 & 1 \end{bmatrix}$ ; note that  $-1 = 1$  in  $\mathbb{F}_2$ . One can readily verify that the above matrix has full row rank (equal to 3) for any  $\lambda \in \{0, 1\}$ ,

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) [17].  $\square$

Since this PBH test is not sufficient to characterize linear systems over finite fields, we will start in the next section by applying a first-principles approach to the problem of multi-agent controllability.

#### V. CONTROLLABILITY OF STRUCTURED SYSTEMS OVER FINITE FIELDS

The field of *structured system theory* deals with analyzing system properties based solely on the zero-nonzero structure of the system matrices. Specifically, a linear system of the form (2) 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) [18]. 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); this is the situation that is commonly considered in the literature on structured systems [18]. With this assumption, these previous works rely on tests such as the PBH condition to determine properties of real-valued matrix sets [18]; however, these proof techniques do not extend to the case where the parameters in the matrices are chosen from finite fields (as discussed in Section IV).

Here, we will investigate structural controllability of matrix pairs of the form  $(\mathbf{A}, \mathbf{e}_{1,N})$  over a finite field  $\mathbb{F}$ , 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. 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  $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ , 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  $\mathbf{A}_{ij}$ .

*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 originating 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  $\mathbb{F}$ .  $\square$

*Proof:* Since the graph associated with  $\mathbf{A}$  is a spanning tree originating at  $x_1$ , there exists a numbering<sup>2</sup> of the nodes

<sup>2</sup>This renumbering simply corresponds to performing a similarity transformation on  $\mathbf{A}$  with a permutation matrix, and thus does not change the eigenvalues of the matrix.

such that the  $\mathbf{A}$  matrix is lower-triangular, with the self-loop weights on the diagonal [14]. Denote the self-loop weight on node  $x_i$  by  $\lambda_i$ . Since all of the self-loop weights are different, this matrix will have  $N$  distinct eigenvalues (given by  $\lambda_1, \lambda_2, \dots, \lambda_N$ ), with  $N$  corresponding linearly independent eigenvectors [19].

Consider the eigenvalue  $\lambda_i$ . 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  $r_i$  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

$$\mathbf{P}_i \mathbf{A} \mathbf{P}_i' = \begin{bmatrix} \mathbf{J}_i & \mathbf{0} \\ \bar{\mathbf{A}}_1 & \bar{\mathbf{A}}_2 \end{bmatrix}, \quad (3)$$

for some matrices  $\bar{\mathbf{A}}_1$  and  $\bar{\mathbf{A}}_2$ . The matrix  $\mathbf{J}_i$  has the form  $\mathbf{J}_i = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{r_i}) + \mathbf{S}_{r_i}$ , where  $\lambda_1, \lambda_2, \dots, \lambda_{r_i}$  are different elements of  $\mathbb{F}$ , and  $\mathbf{S}_{r_i}$  is an  $r_i \times r_i$  square matrix with ones on the main subdiagonal and zeros everywhere else. The matrix  $\mathbf{J}_i$  has  $r_i$  distinct eigenvalues (given by the  $\lambda_t$ 's) in the field  $\mathbb{F}$ , and thus the matrix has  $r_i$  eigenvectors over  $\mathbb{F}$ . Note that there exists some  $t \in \{1, 2, \dots, r_i\}$  such that  $\lambda_t = \lambda_i$  (where  $\lambda_i$  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  $\lambda_t$  is given by

$$\mathbf{v}_t = \begin{bmatrix} 1 & (\lambda_t - \lambda_1) & (\lambda_t - \lambda_1)(\lambda_t - \lambda_2) & \dots \\ \dots & \prod_{s=1}^{t-1} (\lambda_t - \lambda_s) & 0 & \dots & 0 \end{bmatrix},$$

and thus the left-eigenvector corresponding to eigenvalue  $\lambda_t$  for the matrix  $\mathbf{P}_i \mathbf{A} \mathbf{P}_i'$  in Equation (3) is given by  $\mathbf{w}_t = [\mathbf{v}_t \ \mathbf{0}]$ . Next, note that the left-eigenvector corresponding to eigenvalue  $\lambda_t$  (or equivalently,  $\lambda_i$ ) 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  $\lambda_i$ , 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  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ , and furthermore,  $\mathbf{V} \mathbf{e}_{1,N} = \mathbf{1}_N$ . The controllability matrix for the pair  $(\Lambda, \mathbf{1}_N)$  is a *Vandermonde matrix* in the parameters  $\lambda_1, \lambda_2, \dots, \lambda_N$  [20]. Such matrices are invertible over a field  $\mathbb{F}$  if and only if all of the parameters are distinct elements of that field [20], 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 (since a similarity transformation does not affect the controllability of the system). ■

*Corollary 1:* Consider the matrix pair  $(\mathbf{A}, \mathbf{e}_{1,N})$ , where  $\mathbf{A}$  is an  $N \times N$  structured matrix (i.e., every entry of  $\mathbf{A}$  is either

a fixed zero or an independent free parameter from a field  $\mathbb{F}$ ). Suppose the graph  $\mathcal{H}$  associated with the matrix  $\mathbf{A}$  contains a path from  $x_1$  to every other node, and furthermore, every node has a self-loop (i.e., the diagonal elements of  $\mathbf{A}$  are free parameters). Then if  $\mathbb{F}$  has size at least  $N$ , there exists a choice of parameters from  $\mathbb{F}$  such that the pair  $(\mathbf{A}, \mathbf{e}_{1,N})$  is controllable over that field.

*Proof:* Since  $\mathcal{H}$  contains a path from  $x_1$  to every other node, it contains a subgraph that is a spanning tree originating from  $x_1$ . Set the values of all parameters corresponding to edges that are not in this spanning tree to zero, and set the values of all parameters corresponding to edges between different nodes in the spanning tree to "1". Finally, select the values of the parameters corresponding to self-loops to be such that no two nodes have the same value (this is possible since the size of the field is at least  $N$ ). This produces a matrix  $\mathbf{A}$  satisfying the conditions in Theorem 2, and thus the resulting pair  $(\mathbf{A}, \mathbf{e}_{1,N})$  is controllable. ■

## VI. DESIGN OF NEAREST NEIGHBOR RULES AND CONTROL LAW

Based on our analysis in the previous few sections, we are in place to prove the following result.

*Theorem 3:* Consider a multi-agent system with  $N$  agents given by the set  $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$  and with fixed interconnections described by the graph  $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ . Let  $x_1$  be a leader agent, 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 there is a path from  $x_1$  to every other agent in the network and  $q \geq N$ , there exists a set of weights  $w_{ij} \in \mathbb{F}_q$ ,  $j \in \mathcal{N}_i$  and a set of updates  $u[k] \in \mathbb{F}_q$ ,  $k = 0, 1, \dots, N-1$  such that the state  $\mathbf{x}[N]$  of the agents achieves any desired value  $\bar{\mathbf{x}} \in \mathbb{F}_q^N$  starting from any initial condition  $\mathbf{x}[0]$  when using the linear nearest neighbor updates provided by (1). □

*Proof:* First, note that the weight matrix  $\mathbf{W}$  in (1) is a structured matrix (since every element is either identically zero or an independent free parameter). Since the network contains a path from  $x_1$  to every other node, and the diagonal elements of  $\mathbf{W}$  are free parameters, we can appeal to Corollary 1. If the number of discrete states for each agent satisfies  $q \geq N$ , 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  $(\mathbf{W}, \mathbf{e}_{1,N})$  in (1) is controllable over that field. Then, we have  $\mathbf{x}[N] = \mathbf{W}^N \mathbf{x}[0] + \mathcal{C}_{N-1} \mathbf{u}[0 : N-1]$ , and since we have shown that the matrix  $\mathcal{C}_{N-1}$  is invertible over the field  $\mathbb{F}_q$ , the updates for the leader agent are

$$\mathbf{u}[0 : N-1] = \mathcal{C}_{N-1}^{-1} (\bar{\mathbf{x}} - \mathbf{W}^N \mathbf{x}[0]), \quad (4)$$

where  $\bar{\mathbf{x}}$  is the desired vector in  $\mathbb{F}_q^N$ . ■

### A. Controlling Agents When $q < N$

In this section, we show that networks with a certain topological structure are controllable using finite fields of any size.

*Theorem 4:* Consider a multi-agent system with  $N$  agents given by the set  $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$  and with fixed

interconnections described by the graph  $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ . Let  $x_1$  be a leader agent, 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  $n$ . Suppose  $\mathcal{G}$  contains a subgraph that is a spanning tree originating at  $x_1$  that has at most two branches, and each branch is a path. Then, there is a set of weights  $w_{ij} \in \mathbb{F}_q$ ,  $j \in \mathcal{N}_i$  and a set of updates  $u[k] \in \mathbb{F}_q$ ,  $k = 0, 1, \dots, N-1$  such that the state of the agents  $\mathbf{x}[N]$  achieves any desired value  $\bar{\mathbf{x}} \in \mathbb{F}_q^N$  starting from any initial condition  $\mathbf{x}[0]$  when using the nearest neighbor updates provided by (1).  $\square$

Note that the difference between Theorem 3 and Theorem 4 is that the latter focuses on graphs that contain a particular kind of spanning tree, but does not require the condition  $q \geq N$ . An example of the type of spanning tree discussed in the above theorem is shown in Fig. 1(b).

*Proof:* [Theorem 4] Consider the subgraph of  $\mathcal{G}$  that is a spanning tree originating 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  $x_2, x_3, \dots, x_r$ , and the nodes in the second branch are  $x_{r+1}, x_{r+2}, \dots, x_N$ . 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 weight matrix  $\mathbf{W}$  in (1) then has the form  $\mathbf{W} = \begin{bmatrix} \mathbf{J}_0 & \mathbf{0} \\ \mathbf{F} & \mathbf{J}_1 \end{bmatrix}$ , where  $\mathbf{F} = [\mathbf{e}_{1,N-r} \quad \mathbf{0}]$ ,  $\mathbf{J}_0 = \mathbf{S}_r$ ,  $\mathbf{J}_1 = \mathbf{I}_{N-r} + \mathbf{S}_{N-r}$ , and  $\mathbf{S}_j$  is a  $j \times j$  matrix with ones on the main subdiagonal and zeros elsewhere.

Consider the matrix  $\mathbf{P} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{F} & \mathbf{J}_1 \end{bmatrix}$ ; 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  $(\mathbf{W}, \mathbf{e}_{1,N})$  with  $\mathbf{P}$ , we obtain  $\mathbf{P}\mathbf{W}\mathbf{P}^{-1} = \begin{bmatrix} \mathbf{J}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_1 \end{bmatrix}$  and  $\mathbf{P}\mathbf{e}_{1,N} = [\mathbf{e}'_{1,r} \quad \mathbf{e}'_{1,N-r}]'$ . 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  $[\mathbf{e}_{1,r} \quad \mathbf{J}_0\mathbf{e}_{1,r} \quad \mathbf{J}_0^2\mathbf{e}_{1,r} \quad \cdots \quad \mathbf{J}_0^{r-1}\mathbf{e}_{1,r}] = \mathbf{I}_r$  and  $\mathbf{J}_0^k\mathbf{e}_{1,r} = \mathbf{0}$  for  $k \geq r$ . Thus, the above controllability matrix has the form  $\begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ * & \mathbf{T} \end{bmatrix}$ , where  $*$  represents unimportant quantities and

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

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  $(\mathbf{W}, \mathbf{e}_{1,N})$  is controllable over that field.  $\blacksquare$

Note that the above theorem encompasses topologies where the nodes are simply arranged in a path or a ring (or more generally, any network that contains a Hamiltonian path [14]). For such networks, the above result indicates that one only needs a field with elements “0” and “1” in order to ensure controllability from any node – one simply finds the appropriate spanning tree, and assigns the self-weights on one side of the tree to be the field element “1”, and the self-weights on the other side to be the field element “0”. Note that by assigning the self-weights in this way, we are effectively breaking any symmetries in the graph from the perspective of the leader agent. In other words, we allow different nodes to use different weights in their update (based on where they are in the network). This is in contrast to the previous works that have studied uncontrollability of path topologies in the continuous-state setting [6], [8], [9]. The nearest neighbor rules in those papers are based on the *Laplacian* of the graph, and have the benefit of being uniform for all agents in the network, but consequently do not break symmetries in the network topology.

While we have been able to show that certain graph topologies can be controlled with finite fields of size smaller than  $N$ , the characterization of the smallest size required for controllability of arbitrary graphs is an open problem.

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

To demonstrate how one can apply this finite-field framework to handle multi-agents systems where  $q$  is not of the form  $p^n$  for some prime  $p$ , consider the following example. Suppose that we have four agents arranged as  $x_3 \leftrightarrow x_2 \leftrightarrow x_1 \leftrightarrow x_4$ , and suppose that each agent can be in one of  $q = 6$  states (denoted by the set  $\{0, 1, \dots, 5\}$ ). The initial state of the agents is  $\mathbf{x}[0] = [3 \quad 4 \quad 2 \quad 4]'$ . The leader  $x_1$  would like to obtain a final state of  $\mathbf{x}[T] = [0 \quad 4 \quad 2 \quad 5]'$  for some  $T$ . We will achieve this objective by applying two phases of the nearest neighbor rule (1) (i.e., with  $T = 2N = 8$ ). Define  $\bar{0} \triangleq \{0, 1, 2\}$  and  $\bar{1} \triangleq \{3, 4, 5\}$ . During the first phase, whenever an agent is in one of the states  $\{0, 1, 2\}$ , the agent (and its neighbors) will just map it to the meta-state  $\bar{0}$ , and analogously for the states  $\{3, 4, 5\}$  and the meta-state  $\bar{1}$ . Now, the leader treats this as a multi-agent system with just two states  $\{\bar{0}, \bar{1}\}$ , and changes its objective (for now) to place itself in meta-state  $\bar{0}$  (since  $0 \in \bar{0}$ ), and agents  $x_2, x_3, x_4$  in meta-states  $\bar{1}, \bar{0}, \bar{1}$ , respectively (because their final desired real states lie within those meta-states). This can be done via the nearest neighbor rule (1) and Theorem 4 (since this topology is a tree consisting of two branches, each of which is a path). Specifically, following the proof of Theorem 4, we choose the self-weights on nodes  $x_1, x_2$  and  $x_3$  to be the element  $\bar{0}$ , and the self-weight on  $x_4$  to be  $\bar{1}$ . We set the other weights to  $\bar{1}$  or  $\bar{0}$  as needed to obtain a spanning tree originating at  $x_1$ . This produces the weight matrix  $\mathbf{W} = \begin{bmatrix} \bar{0} & \bar{0} & \bar{0} & \bar{0} \\ \bar{1} & \bar{0} & \bar{0} & \bar{0} \\ \bar{0} & \bar{1} & \bar{0} & \bar{0} \\ \bar{1} & \bar{0} & \bar{0} & \bar{1} \end{bmatrix}$ . The controllability matrix for the

pair  $(\mathbf{W}, \mathbf{e}_{1,4})$  (over the field  $\mathbb{F}_2$ ) is given by  $\mathcal{C}_3 = \begin{bmatrix} \bar{1} & \bar{0} & \bar{0} & \bar{0} \\ \bar{0} & \bar{1} & \bar{0} & \bar{0} \\ \bar{0} & \bar{0} & \bar{1} & \bar{0} \\ \bar{0} & \bar{1} & \bar{1} & \bar{1} \end{bmatrix}$ , with inverse  $\mathcal{C}_3^{-1} = \mathcal{C}_3$  over the field  $\mathbb{F}_2$  (note that all additions and multiplications are performed modulo 2 in this field). Let  $\bar{\mathbf{x}}[k]$  denote the vector of meta-states of the agents at time-step  $k$ , so that  $\bar{\mathbf{x}}[0] = [\bar{1} \ \bar{1} \ \bar{0} \ \bar{1}]'$ . Using the control law (4), we find that  $\bar{\mathbf{x}}[4] = [\bar{0} \ \bar{1} \ \bar{0} \ \bar{1}]'$  if  $x_1$  applies  $\bar{\mathbf{u}}[0 : 3] = [\bar{0} \ \bar{1} \ \bar{0} \ \bar{0}]'$ . Note that at each time-step  $k$ , each agent  $x_i$  updates its meta-state  $\bar{x}_i[k]$  based on its own meta-state and the meta-states of its neighbors (i.e., it does not care which of the real states comprising each of the meta-states is occupied by each agent). Thus, if the nearest neighbor rule specifies that an agent should move to meta-state  $\bar{0}$ , it can move to any of the real states within that meta-state; for convenience, we can assume that each agent occupies the first real state in its meta-state (i.e., state 0 in meta-state  $\bar{0}$  and state 3 in meta-state  $\bar{1}$ ).

At the conclusion of the first phase, each agent is in the meta-state that contains its final desired state. Now, note that each meta-state contains 3 real states. Since this is again a prime number, one can apply the nearest neighbor rule (1) to place each agent in their final desired state. From this point on, the agents ignore the meta-state that is occupied by each agent, and will instead only monitor the three real states within each meta-state. Furthermore, each agent will only move within the three real states that comprise its current meta-state. Mapping the three real states in any meta-state to the set  $\{\hat{0}, \hat{1}, \hat{2}\}$ , and denoting the corresponding state of each agent  $x_i$  by  $\hat{x}_i[k]$ , the leader agent's objective is to drive all agents to the state  $\hat{\mathbf{x}}[4] = [\hat{0} \ \hat{1} \ \hat{2} \ \hat{2}]'$ ; these values are obtained by noting that the real state 0 maps to state  $\hat{0}$  in meta-state  $\bar{0}$ , real state 4 maps to state  $\hat{1}$  in meta-state  $\bar{1}$  and so forth. To do this, note that the weight matrix considered above will also work over the field  $\mathbb{F}_3$  (by Theorem 4), except now all operations will be performed modulo 3, and elements  $\bar{0}$  and  $\bar{1}$  will be replaced by  $\hat{0}$  and  $\hat{1}$  (to correspond to the representations of those elements in the field of size 3). Once again, the controllability matrix for the

pair  $(\mathbf{W}, \mathbf{e}_{1,4})$  is full rank, but has inverse  $\mathcal{C}_3^{-1} = \begin{bmatrix} \hat{1} & \hat{0} & \hat{0} & \hat{0} \\ \hat{0} & \hat{1} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & \hat{1} & \hat{0} \\ \hat{0} & \hat{2} & \hat{2} & \hat{1} \end{bmatrix}$  over  $\mathbb{F}_3$ . The initial states of all agents are taken to be  $\hat{\mathbf{x}}[0] = \mathbf{0}$  (under the assumption that each agent occupies the first state inside its meta-state during the first phase of the algorithm). Using (4), we find that by applying the inputs  $\hat{\mathbf{u}}[0 : 3] = [\hat{0} \ \hat{1} \ \hat{2} \ \hat{2}]'$ , the leader can place all agents into the desired final state  $\hat{\mathbf{x}}[4]$ . At the conclusion of the two phases, we have  $\mathbf{x}[8] = [0 \ 4 \ 2 \ 5]'$ .

The above algorithm can be applied to any value<sup>3</sup> of  $q$  by first writing  $q = p_1^{n_1} p_2^{n_2} \cdots p_r^{n_r}$ , where  $p_1, p_2, \dots, p_r$  are distinct primes and  $n_1, n_2, \dots, n_r$  are positive integers. One can then use  $r$  different phases of the nearest neighbor rule (1) (as long as each  $p_i^{n_i}$  is sufficiently large to allow the use of Theorem 3 or Theorem 4) to have the leader place each

agent into increasingly refined regions. This idea is similar to that of “zooming-in” with a quantizer in order to stabilize systems with quantized measurements [21].

## VII. SUMMARY

We showed how to formulate a set of nearest neighbor rules for a network of quantized agents so that they can be put into any desired configuration by a leader agent. We obtained this result by viewing the discrete states of the agents as elements of a finite field, and then developed a theory of linear system controllability 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 leader has a path to every agent.

## REFERENCES

- [1] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, “Human-robot teaming for search and rescue,” *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 72–79, Jan. 2005.
- [2] E. Guizzo, “Three engineers, hundreds of robots, one warehouse,” *IEEE Spectrum*, vol. 45, no. 7, pp. 26–34, July 2008.
- [3] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009.
- [4] 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.
- [5] H. Tanner, “On the controllability of nearest neighbor interconnections,” in *Proc. 43rd IEEE Conference on Decision and Control*, 2004, pp. 2467–2472.
- [6] A. Rahmani and M. Mesbahi, “On the controlled agreement problem,” in *Proc. American Control Conference*, 2006, pp. 1376–1381.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] A. Kashyap, T. Başar, and R. Srikant, “Quantized consensus,” *Automatica*, vol. 43, no. 7, pp. 1192–1203, July 2007.
- [12] 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.
- [13] 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.
- [14] D. B. West, *Introduction to Graph Theory*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 2001.
- [15] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, 2nd ed. Cambridge University Press, 1994.
- [16] P. J. Antsaklis and A. N. Michel, *Linear Systems*. Birkhauser Boston, 2006.
- [17] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2003.
- [18] 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.
- [19] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
- [20] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge University Press, 2003.
- [21] R. W. Brockett and D. Liberzon, “Quantized feedback stabilization of linear systems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1279–1289, July 2000.

<sup>3</sup>This procedure can be avoided if one can design the system to have  $q = p^n$ ; however, in practice the number of states  $q$  for each agent may be given *a priori*, and cannot be changed to a more convenient form.