2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Distributed Function Calculation via Linear Iterations in the Presence of Malicious Agents – Part II: Overcoming Malicious Behavior

Shreyas Sundaram and Christoforos N. Hadjicostis

Abstract— **Given a network of interconnected nodes, each with a given initial value, we develop a distributed strategy that enables some or all of the nodes to calculate any arbitrary function of these initial values, despite the presence of some malicious (or faulty) nodes. Our scheme utilizes a linear iterative strategy where, at each time-step, each node updates its value to be a weighted average of its own previous value and those of its neighbors. We consider a node to be malicious if, instead of following the predefined linear iterative strategy, it updates its value arbitrarily at each time-step (perhaps by conspiring and coordinating with other malicious nodes). When there are up to f malicious nodes, we show that any node in the network is guaranteed to be able to calculate any arbitrary function of all initial node values if the graph of the network is at least $(2f+1)$-connected. Specifically, we show that under this condition, the nodes can calculate their desired functions after running the linear iteration for a finite number of time-steps (upper bounded by the number of nodes in the network) using almost any set of weights (i.e., for all weights except for a set of measure zero). Our approach treats the problem of fault-tolerant distributed consensus, where all nodes have to calculate the same function despite the presence of faulty or malicious nodes, as a special case.**

I. Introduction

In distributed systems and networks, it is often necessary for some or all of the nodes to calculate some function of certain parameters. For example, sink nodes in sensor networks may be tasked with calculating the average value of all the sensor measurements [1]. A special case of distributed function calculation is the *distributed consensus* problem, where all nodes in the network calculate the same function [2]. The notion of consensus has recently received extensive attention in the control literature, due to its applicability to topics such as cooperative control of multi-agent systems [3]. In these cases, the approach to consensus is to use a linear iteration, where each node in the network repeatedly updates its value to be a weighted linear combination of its own value and those of its neighbors (e.g. see [3] and the references therein). These works have revealed that if the network topology satisfies certain conditions, the weights for the linear iteration can be chosen so that all of the nodes asymptotically converge to the same value. Recently, it was shown in [4], [5] that this linear iterative strategy can actually be applied to the more general function calculation problem, allowing any node in the network to calculate an arbitrary function of the node values in a finite number of time-steps. In Part I of this paper [6], we studied the susceptibility of linear iterative schemes to (possibly coordinated) misbehavior by a subset of nodes. Specifically, we allowed for the possibility that some nodes in the network update their values at each time-step in an arbitrary manner, instead of following the predefined strategy of using a specific weighted linear combination of their neighbors' and their own values. We showed that if the graph of the network is $2f$ connected, then it is possible for f nodes to conspire to maliciously update their values in such a way that some nodes *cannot* calculate an arbitrary function of all node values [6].

In this paper, we present a method to inoculate linear iterative schemes against malicious behavior by nodes in the network. Specifically, the contribution of this paper is to show that, if the network topology has connectivity at least $2f+1$, then any node can work around the misbehaving nodes to calculate any desired function of the initial node values, even if up to f nodes conspire and coordinate their actions in an attempt to disrupt the network. Given a graph of $2f+1$ connectivity, our strategy is to have the nodes follow a linear iterative protocol with random weights, and to have each node simply observe its own value and those of its neighbors as they evolve over time. After doing this for a finite number of time-steps, we show that each node will have enough information (and a checking scheme) to correctly determine the initial values despite the possible presence of up to f malicious nodes. To prove the correctness of this strategy, we exploit concepts from classical linear system theory (such as structured system theory, strong observability, and invariant zeros of linear systems). The problem of fault-tolerant distributed consensus can be treated as a special case of our results, effectively narrowing the gap between linear iterative schemes and existing fault-tolerant consensus protocols (such as those described in [2]).

For relevant definitions and terminology on graph theory, we refer the reader to Part I of this paper [6]. We will use the following classical result in our derivations[1] (e.g., see [7]).

Theorem 1: Suppose graph $\mathcal{G}$, with vertex set $\mathcal{X}$, is κ-connected. Given any two subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{X}$, with $|\mathcal{X}_1| \geq \kappa$, $|\mathcal{X}_2| \geq \kappa$, there exists a κ-linking from $\mathcal{X}_1$ to $\mathcal{X}_2$. □

Note that some of the paths in this linking might have zero length (i.e., if $\mathcal{X}_1 \cap \mathcal{X}_2$ is nonempty).

This material is based upon work supported in part by the National Science Foundation under NSF Career Award 0092696 and NSF ITR Award 0426831. 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.

The authors are with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, 61801-2307, USA. E-mails: {ssundarm, chadjic}@uiuc.edu.

[1] Recall that a r-linking from a set of vertices $\mathcal{X}_1$ to a set of vertices $\mathcal{X}_2$ is a set of vertex disjoint paths, each with a start vertex in $\mathcal{X}_1$ and an end vertex in $\mathcal{X}_2$.

II. Linear Iterative Protocol and Modeling of Malicious Node Behavior

The interaction constraints in a distributed system or network can be modeled via a directed graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where $\mathcal{X} = \{x_1, \ldots, x_N\}$ is the set of nodes in the system and $\mathcal{E} \subseteq \mathcal{X} \times \mathcal{X}$ represents the communication constraints in the network (i.e., directed edge $(x_j, x_i) \in \mathcal{E}$ if node x_i can receive information directly from node x_j). The neighbors of node i are given by the set $\mathcal{N}_i = \{x_j | (x_j, x_i) \in \mathcal{E}\}$.

Suppose that each node i has some initial value, given by $x_i[0]$. At each time-step k, all nodes update and/or exchange their values based on some strategy that adheres to the constraints imposed by the network topology. The scheme that we study in this paper makes use of linear iterations; specifically, at each time-step, each node updates its value as

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

where the w_{ij}'s are a set of weights.[2] The term $u_i[k]$ is used to represent an additive error by node i at time-step k. Specifically, we introduce the following definition.

Definition 1: Suppose all nodes run the linear iteration for T time-steps in order to perform function calculation. Node i is said to be *malicious* (or *faulty*) if $u_i[k]$ is nonzero for at least one time-step k, $0 \leq k \leq T-1$. □

Note that the model for malicious nodes considered here is quite general, and allows node i to update its value in a completely arbitrary manner (via appropriate choices of $u_i[k]$ at each time-step). As such, this model encapsulates a wide variety of malicious behavior (including a conspiracy by a set of malicious nodes). Note, however, that we do not treat the case where malicious nodes try to influence the result of the computation by modifying their initial values; see Part I of this paper for a further discussion of this issue [6].

Let $\mathcal{S} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_f}\}$ denote the set of nodes that are malicious during a run of the linear iteration. For ease of analysis, the values of all nodes at time-step k can be aggregated into the value vector $\mathbf{x}[k] = \begin{bmatrix} x_1[k] & x_2[k] & \cdots & x_N[k] \end{bmatrix}'$, and based on the local updates (1), the update equation for the entire system can be represented as

$$\mathbf{x}[k+1] = W\mathbf{x}[k] + \underbrace{\begin{bmatrix} \mathbf{e}_{i_1} & \mathbf{e}_{i_2} & \cdots & \mathbf{e}_{i_f} \end{bmatrix}}_{B_{\mathcal{S}}} \underbrace{\begin{bmatrix} u_{i_1}[k] \\ u_{i_2}[k] \\ \vdots \\ u_{i_f}[k] \end{bmatrix}}_{\mathbf{u}_{\mathcal{S}}[k]}$$

$$\mathbf{y}_i[k] = C_i \mathbf{x}[k], \ \ 1 \leq i \leq N \ , \quad (2)$$

where $\mathbf{e}_l$ denotes a unit vector with a single nonzero entry with value 1 at its l–th position. The matrix W in the above equation is called the weight matrix for the linear iteration, with entry (i, j) containing the weight w_{ij} from the linear update (1); note that this imposes the constraint that the (i, j)-th entry is zero if node j is not a neighbor of node i. The quantity $\mathbf{y}_i[k]$ in equation (2) represents the outputs (node values) seen by node i during time-step k of the linear iteration. Specifically, C_i is a $(\deg_i + 1) \times N$ matrix with a single 1 in each row capturing the positions of the state-vector $\mathbf{x}[k]$ that are available to node i (these positions correspond to neighbors of node i, along with node i itself). The set of all values seen by node i during the first $L+1$ time-steps of the linear iteration is given by

$$\underbrace{\begin{bmatrix} \mathbf{y}_i[0] \\ \mathbf{y}_i[1] \\ \mathbf{y}_i[2] \\ \vdots \\ \mathbf{y}_i[L] \end{bmatrix}}_{\mathbf{y}_i[0:L]} = \underbrace{\begin{bmatrix} C_i \\ C_i W \\ C_i W^2 \\ \vdots \\ C_i W^L \end{bmatrix}}_{\mathcal{O}_{i,L}} \mathbf{x}[0] + \quad (3)$$

$$\underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ C_i B_{\mathcal{S}} & 0 & \cdots & 0 \\ C_i W B_{\mathcal{S}} & C_i B_{\mathcal{S}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_i W^{L-1} B_{\mathcal{S}} & C_i W^{L-2} B_{\mathcal{S}} & \cdots & C_i B_{\mathcal{S}} \end{bmatrix}}_{\mathcal{M}^{\mathcal{S}}_{i,L}} \underbrace{\begin{bmatrix} \mathbf{u}_{\mathcal{S}}[0] \\ \mathbf{u}_{\mathcal{S}}[1] \\ \mathbf{u}_{\mathcal{S}}[2] \\ \vdots \\ \mathbf{u}_{\mathcal{S}}[L-1] \end{bmatrix}}_{\mathbf{u}_{\mathcal{S}}[0:L-1]} .$$

The matrices $\mathcal{O}_{i,L}$ and $\mathcal{M}^{\mathcal{S}}_{i,L}$ will characterize the ability of node i to calculate the required function of the initial values. The matrix $\mathcal{O}_{i,L}$ resembles the *observability matrix* [8] for the pair (W, C_i) (it is exactly the observability matrix for $L = N-1$), and we will call $\mathcal{M}^{\mathcal{S}}_{i,L}$ the *fault matrix* for the triplet $(W, B_{\mathcal{S}}, C_i)$. Note that our earlier work in [5] focused on characterizing the conditions under which, for each node i, the matrix $\mathcal{O}_{i,L}$ allows node i to calculate the desired function of the initial values (with $\mathcal{M}^{\mathcal{S}}_{i,L} = 0$). We will call upon the following simple lemma in our development.

Lemma 1: Let $\mathcal{S}_1$ and $\mathcal{S}_2$ denote two subsets of $\mathcal{X}$, and define $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$. Then, the column space of $\mathcal{M}^{\mathcal{S}}_{i,L}$ is the same as the column space of $\begin{bmatrix} \mathcal{M}^{\mathcal{S}_1}_{i,L} & \mathcal{M}^{\mathcal{S}_2}_{i,L} \end{bmatrix}$.

Proof: Let $\mathcal{S}_1 = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{|\mathcal{S}_1|}}\}$, $\mathcal{S}_2 = \{x_{j_1}, x_{j_2}, \ldots, x_{j_{|\mathcal{S}_2|}}\}$, and $\mathcal{S} = \{x_{l_1}, x_{l_2}, \ldots, x_{l_{|\mathcal{S}|}}\}$, so that $B_{\mathcal{S}_1} = \begin{bmatrix} \mathbf{e}_{i_1} & \mathbf{e}_{i_2} & \cdots & \mathbf{e}_{i_{|\mathcal{S}_1|}} \end{bmatrix}$, $B_{\mathcal{S}_2} = \begin{bmatrix} \mathbf{e}_{j_1} & \cdots & \mathbf{e}_{j_{|\mathcal{S}_2|}} \end{bmatrix}$, $B_{\mathcal{S}} = \begin{bmatrix} \mathbf{e}_{l_1} & \mathbf{e}_{l_2} & \cdots & \mathbf{e}_{l_{|\mathcal{S}|}} \end{bmatrix}$. Since $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, we have $\mathcal{R}(B_{\mathcal{S}}) = \mathcal{R}\left(\begin{bmatrix} B_{\mathcal{S}_1} & B_{\mathcal{S}_2} \end{bmatrix}\right)$ (where $\mathcal{R}(\cdot)$ denotes the column space of the matrix). From the structure of $B_{\mathcal{S}_1}, B_{\mathcal{S}_2}$ and $B_{\mathcal{S}}$, we have $\mathcal{R}\left(\mathcal{O}_{i,L} B_{\mathcal{S}}\right) = \mathcal{R}\left(\mathcal{O}_{i,L} \begin{bmatrix} B_{\mathcal{S}_1} & B_{\mathcal{S}_2} \end{bmatrix}\right)$ for any $L \geq 0$. Using this in the expression for $\mathcal{M}^{\mathcal{S}}_{i,L}$ in (3), one obtains $\mathcal{R}\left(\mathcal{M}^{\mathcal{S}}_{i,L}\right) = \mathcal{R}\left(\begin{bmatrix} \mathcal{M}^{\mathcal{S}_1}_{i,L} & \mathcal{M}^{\mathcal{S}_2}_{i,L} \end{bmatrix}\right)$. ■

In the sequel, we will prove the following key result, thereby showing how to incorporate resilience to malicious nodes in linear iteration-based function calculation schemes.

Theorem 2: Let f denote the maximum number of malicious nodes that are to be tolerated in a given network $\mathcal{G}$, and let κ denote the connectivity of the network. If $\kappa \geq 2f+1$, then for almost any choice of weight matrix, every node in the network can calculate any function of the initial values

[2] The methodology for choosing the weights appropriately and the implications of this choice are discussed later in the paper.

after running the linear iteration for at most N time-steps, even when up to f malicious nodes update their values arbitrarily (and possibly in a coordinated manner) at each time-step. □

III. Calculating Functions in the Presence of Malicious Nodes when $\kappa \geq 2f+1$

The proof of the following theorem provides a procedure for each node to calculate its desired function in the presence of malicious nodes.

Theorem 3: Suppose that there exists a weight matrix W and an integer L such that, for all possible sets $\mathcal{S}$ of $2f$ nodes, the matrices $\mathcal{O}_{i,L}$ and $\mathcal{M}_{i,L}^{\mathcal{S}}$ satisfy

$$\rho\left(\begin{bmatrix}\mathcal{O}_{i,L} & \mathcal{M}_{i,L}^{\mathcal{S}}\end{bmatrix}\right) = N + \rho\left(\mathcal{M}_{i,L}^{\mathcal{S}}\right) \ . \tag{4}$$

Then, if the nodes run the linear iteration for $L+1$ time-steps with the weight matrix W, node i can calculate any arbitrary function of the values $x_1[0], x_2[0], \ldots, x_N[0]$, even when up to f nodes are malicious. □

Proof: Let W be a weight matrix that satisfies the conditions in the above theorem, and let the nodes run the linear iteration for $L+1$ time-steps. Suppose that the malicious nodes during the linear iteration are a subset of the set $\mathcal{S}_F = \{x_{j_1}, x_{j_2}, \ldots, x_{j_f}\}$. From (3), the values seen by node i over $L+1$ time-steps are given by

$$\mathbf{y}_i[0:L] = \mathcal{O}_{i,L}\mathbf{x}[0] + \mathcal{M}_{i,L}^{\mathcal{S}_F}\mathbf{u}_{\mathcal{S}_F}[0:L-1] \ . \tag{5}$$

Note that if there are fewer than f malicious nodes, the components of $\mathbf{u}_{\mathcal{S}_F}[0:L-1]$ corresponding to the non-malicious nodes in $\mathcal{S}_F$ can simply be set to zero. Let $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\binom{N}{f}}$ denote all possible sets of f nodes, and let $\mathcal{M}_{i,L}^{\mathcal{S}_1}, \mathcal{M}_{i,L}^{\mathcal{S}_2}, \ldots, \mathcal{M}_{i,L}^{\mathcal{S}_{\binom{N}{f}}}$ denote the corresponding fault matrices. With these matrices in hand, suppose node i finds the first $j \in \{1, 2, \ldots, \binom{N}{f}\}$ such that

$$\rho\left(\begin{bmatrix}\mathcal{O}_{i,L} & \mathcal{M}_{i,L}^{\mathcal{S}_j} & \mathbf{y}_i[0:L]\end{bmatrix}\right) = \rho\left(\begin{bmatrix}\mathcal{O}_{i,L} & \mathcal{M}_{i,L}^{\mathcal{S}_j}\end{bmatrix}\right). \tag{6}$$

In other words, node i finds the first j such that the vector $\mathbf{y}_i[0:L]$ is in the column space of the matrices $\mathcal{O}_{i,L}$ and $\mathcal{M}_{i,L}^{\mathcal{S}_j}$ (note that such a j is guaranteed to be found, since the above equation will be satisfied for $\mathcal{S}_j = \mathcal{S}_F$). This means that there exist vectors $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ such that $\mathcal{O}_{i,L}\bar{\mathbf{x}} + \mathcal{M}_{i,L}^{\mathcal{S}_j}\bar{\mathbf{u}} = \mathbf{y}_i[0:L]$. Using (5) and rearranging, we have $\mathcal{O}_{i,L}(\mathbf{x}[0] - \bar{\mathbf{x}}) = \mathcal{M}_{i,L}^{\mathcal{S}_j}\bar{\mathbf{u}} - \mathcal{M}_{i,L}^{\mathcal{S}_F}\mathbf{u}_{\mathcal{S}_F}[0:L-1]$. Letting $\mathcal{S} = \mathcal{S}_F \cup \mathcal{S}_j$, we note from Lemma 1 that the above expression can be written as $\mathcal{O}_{i,L}(\mathbf{x}[0] - \bar{\mathbf{x}}) = \mathcal{M}_{i,L}^{\mathcal{S}}\mathbf{u}_{\mathcal{S}}[0:L-1]$, for some appropriately defined vector $\mathbf{u}_{\mathcal{S}}[0:L-1]$. From equation (4) in the statement of the theorem, the observability matrix is assumed to be of full column rank, and all its columns are linearly independent of the columns of the fault matrix $\mathcal{M}_{i,L}^{\mathcal{S}}$ (since the set $\mathcal{S}$ has $2f$ or fewer nodes). This means that $\mathbf{x}[0] = \bar{\mathbf{x}}$ in the above expression, and so

$$0 = \mathcal{M}_{i,L}^{\mathcal{S}}\mathbf{u}_{\mathcal{S}}[0:L-1] = \mathcal{M}_{i,L}^{\mathcal{S}_j}\bar{\mathbf{u}} - \mathcal{M}_{i,L}^{\mathcal{S}_F}\mathbf{u}_{\mathcal{S}_F}[0:L-1] \ ,$$

or equivalently, $\mathcal{M}_{i,L}^{\mathcal{S}_F}\mathbf{u}_{\mathcal{S}_F}[0:L-1] = \mathcal{M}_{i,L}^{\mathcal{S}_j}\bar{\mathbf{u}}$. Let $\mathcal{N}_{i,L}^{\mathcal{S}_j}$ be a matrix whose rows form a basis for the left null space of $\mathcal{M}_{i,L}^{\mathcal{S}_j}$. Left-multiplying the above expression by $\mathcal{N}_{i,L}^{\mathcal{S}_j}$, we obtain $\mathcal{N}_{i,L}^{\mathcal{S}_j}\mathcal{M}_{i,L}^{\mathcal{S}_F}\mathbf{u}_{\mathcal{S}_F}[0:L-1] = 0$, which means that if we left-multiply (5) by $\mathcal{N}_{i,L}^{\mathcal{S}_j}$, we have $\mathcal{N}_{i,L}^{\mathcal{S}_j}\mathbf{y}_i[0:L] = \mathcal{N}_{i,L}^{\mathcal{S}_j}\mathcal{O}_{i,L}\mathbf{x}[0]$. Since the columns of $\mathcal{O}_{i,L}$ are all linearly independent of the columns of $\mathcal{M}_{i,L}^{\mathcal{S}_j}$ (from equation (4)), the matrix $\mathcal{N}_{i,L}^{\mathcal{S}_j}\mathcal{O}_{i,L}$ will have full column rank. Define

$$\mathcal{P}_{i,L}^{\mathcal{S}_j} = \left(\mathcal{N}_{i,L}^{\mathcal{S}_j}\mathcal{O}_{i,L}\right)^{\dagger}\mathcal{N}_{i,L}^{\mathcal{S}_j} \ , \tag{7}$$

where the notation $(\cdot)^{\dagger}$ indicates the left inverse of a matrix. One can verify that

$$\mathcal{P}_{i,L}^{\mathcal{S}_j}\mathbf{y}_i[0:L] = \mathbf{x}[0] \ , \tag{8}$$

and so node i can obtain the entire set of initial values $\mathbf{x}[0]$ from the above equation (and thereby calculate any function of those values). ■

In the sequel, we will show that when $\kappa \geq 2f+1$, one can find a weight matrix W and an integer L so that all columns of the observability matrix $\mathcal{O}_{i,L}$ will be linearly independent of each other, and of the columns in $\mathcal{M}_{i,L}^{\mathcal{S}}$, for every i and for any set $\mathcal{S}$ of up to $2f$ nodes (i.e., equation (4) will be satisfied for every node i). To find such a weight matrix, we will first require some concepts from classical control theory.

A. Strong Observability

Consider a linear system of the form

$$\begin{aligned}\mathbf{x}[k+1] &= A\mathbf{x}[k] + B\mathbf{u}[k] \\ \mathbf{y}[k] &= C\mathbf{x}[k] + D\mathbf{u}[k] \ ,\end{aligned} \tag{9}$$

with state vector $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^p$, with $p \geq m$. When the values of the inputs at each time-step are completely unknown and arbitrary, systems of the above form are termed *linear systems with unknown inputs* [9]. For such systems, the following terminology has been established in the literature (e.g., see [9], [10], [11]).

Definition 2: A linear system with unknown inputs (of the form (9)) is said to be *strongly observable* if $\mathbf{y}[k] = 0$ for all k implies $\mathbf{x}[0] = 0$ (regardless of the values of the unknown inputs $\mathbf{u}[k]$). □

Definition 3: For the linear system (9), the matrix $P(z) = \begin{bmatrix} A - zI_n & B \\ C & D \end{bmatrix}$ is called the *matrix pencil* of the set (A, B, C, D). □

Definition 4: The *normal-rank* of the matrix pencil $P(z)$ is defined as $\rho_n(P(z)) \equiv \max_{z_0 \in \mathbb{C}} \rho(P(z_0))$. □

Definition 5: The complex number $s \in \mathbb{C}$ is called an *invariant zero* of the system (9) if $\rho\left(P(s)\right) < \rho_n(P(z))$. □

Theorem 4 ([10], [12], [9]): The following statements are equivalent for the system (9):

- Denoting the output of the system over n time-steps by

$$\begin{bmatrix}\mathbf{y}[0] \\ \mathbf{y}[1] \\ \vdots \\ \mathbf{y}[n-1]\end{bmatrix} = \underbrace{\begin{bmatrix}C \\ CA \\ \vdots \\ CA^{n-1}\end{bmatrix}}_{\mathcal{O}_{n-1}}\mathbf{x}[0] +$$

$$\underbrace{\begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n-2}B & CA^{n-3}B & \cdots & D \end{bmatrix}}_{\mathcal{M}_{n-1}} \begin{bmatrix} \mathbf{u}[0] \\ \mathbf{u}[1] \\ \vdots \\ \mathbf{u}[n-1] \end{bmatrix},$$

the matrices $\mathcal{O}_{n-1}$ and $\mathcal{M}_{n-1}$ satisfy $\rho\left(\begin{bmatrix}\mathcal{O}_{n-1} & \mathcal{M}_{n-1}\end{bmatrix}\right) = n + \rho(\mathcal{M}_{n-1})$.

- The set (A, B, C, D) has no invariant zeros.
- The system is strongly observable.

□

The above theorem indicates that if we can choose the weight matrix W so that the set $(W, B_{\mathcal{S}}, C_i, 0)$ has no invariant zeros, for all possible sets $\mathcal{S}$ of $2f$ nodes, then the rank condition in equation (4) of Theorem 3 will be satisfied with $L = N - 1$ (note that the assumption $p \geq m$ will be trivially satisfied in networks that have $(2f+1)$-connectivity, because each node i will have at least $2f+1$ neighbors, and thus C_i will have at least $2f+2$ rows). Therefore, node i will be able to calculate any desired function of the initial values, even in the presence of up to f malicious nodes. Thus, we now focus on choosing W so that this is the case.

B. Invariant Zeros

Let $\mathcal{S} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{2f}}\}$ denote any set of $2f$ nodes, and let $\bar{\mathcal{S}} = \mathcal{X} - \mathcal{S}$. Define $B_{\mathcal{S}} = \begin{bmatrix}\mathbf{e}_{i_1} & \mathbf{e}_{i_2} & \cdots & \mathbf{e}_{i_{2f}}\end{bmatrix}$. For any choice of weights for the linear iteration, let $W_{\bar{\mathcal{S}}}$ represent the weight matrix corresponding to interconnections within the set $\bar{\mathcal{S}}$, $W_{\mathcal{S},\bar{\mathcal{S}}}$ represent the weight matrix corresponding to connections from the set $\mathcal{S}$ to the set $\bar{\mathcal{S}}$, $W_{\mathcal{S}}$ represent the weight matrix corresponding to interconnections within the set $\mathcal{S}$, and $W_{\bar{\mathcal{S}},\mathcal{S}}$ represent the weight matrix corresponding to connections from the set $\bar{\mathcal{S}}$ to the set $\mathcal{S}$. Note that $W_{\bar{\mathcal{S}}}$ has dimension $(N-2f)\times(N-2f)$, and $W_{\mathcal{S}}$ has dimension $(2f)\times(2f)$. Furthermore, for any node i, let $C_i = \begin{bmatrix}C_{i,\bar{\mathcal{S}}} & C_{i,\mathcal{S}}\end{bmatrix}$ denote a $(\deg_i + 1) \times N$ matrix with a single 1 in each row corresponding to the nodes in $\mathcal{X}$ that are neighbors of node i, along with node i itself (the first $N-2f$ columns correspond to nodes in $\bar{\mathcal{S}}$ and the last $2f$ columns correspond to nodes in $\mathcal{S}$).

Lemma 2: For any set $\mathcal{S}$ of $2f$ nodes, the invariant zeros of $(W, B_{\mathcal{S}}, C_i, 0)$ are exactly the invariant zeros of $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$.

Proof: The matrix pencil for the set $(W, B_{\mathcal{S}}, C_i, 0)$ is given by $P(z) = \left[\begin{smallmatrix} W - zI_N & B_{\mathcal{S}} \\ C_i & 0 \end{smallmatrix}\right]$. Without loss of generality, we can assume that W is of the form

$$W = \begin{bmatrix} W_{\bar{\mathcal{S}}} & W_{\mathcal{S},\bar{\mathcal{S}}} \\ W_{\bar{\mathcal{S}},\mathcal{S}} & W_{\mathcal{S}} \end{bmatrix},$$

since this is obtained simply by renumbering the nodes so that nodes in the set $\mathcal{S}$ have indices between $N-2f+1$ and N. This also means that $B_{\mathcal{S}}$ has the form $B_{\mathcal{S}} = \left[\begin{smallmatrix} 0 \\ I_{2f} \end{smallmatrix}\right]$, and substituting these equations (with $C_i = \begin{bmatrix}C_{i,\bar{\mathcal{S}}} & C_{i,\mathcal{S}}\end{bmatrix}$) into the above expression for $P(z)$, we see that

$$\rho(P(z)) = 2f + \rho\left(\begin{bmatrix} W_{\bar{\mathcal{S}}} - zI_{N-2f} & W_{\mathcal{S},\bar{\mathcal{S}}} \\ C_{i,\bar{\mathcal{S}}} & C_{i,\mathcal{S}} \end{bmatrix}\right).$$

Thus, the invariant zeros of the set $(W, B_{\mathcal{S}}, C_i, 0)$ are exactly the invariant zeros of the set $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$. ■

Lemma 2 and Theorem 4 reveal that in order to ensure that the rank condition (4) in Theorem 3 is satisfied for every node i, we can focus on the problem of choosing the weights so that the set $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$ will have no invariant zeros, for any choice of i and for any set $\mathcal{S}$ of $2f$ nodes. To choose a set of weights that accomplishes this, we will use techniques from a branch of control theory pertaining to *linear structured systems* [13]. Specifically, a linear system of the form (9) is said to be structured if each entry of the matrices A, B, C and D is either a fixed zero or an independent free parameter. It is known that linear systems have certain structural properties (such as observability, controllability, etc.), and these properties hold generically. In other words, if the structural property holds for some particular choice of free parameters, it will hold for almost any choice of parameters (i.e., the set of parameters for which the property does not hold has Lebesgue measure zero) [13]. It turns out that the number of invariant zeros of a linear system is also a structured property (i.e., a linear system with a given zero/nonzero structure will have the same number of invariant zeros for almost any choice of the free parameters) [11].

To analyze structural properties (such as the number of invariant zeros) of linear systems, one first associates a graph $\mathcal{H}$ with the structured set (A, B, C, D) as follows. The vertex set of $\mathcal{H}$ is given by $\mathcal{X} \cup \mathcal{U} \cup \mathcal{Y}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ is the set of state vertices, $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ is the set of input vertices, and $\mathcal{Y} = \{y_1, y_2, \ldots, y_p\}$ is the set of output vertices. The edge set of $\mathcal{H}$ is given by $\mathcal{E}_{xx} \cup \mathcal{E}_{ux} \cup \mathcal{E}_{xy} \cup \mathcal{E}_{uy}$, where $\mathcal{E}_{xx} = \{(x_j, x_i) | A_{ij} \neq 0\}$, $\mathcal{E}_{ux} = \{(u_j, x_i) | B_{ij} \neq 0\}$, $\mathcal{E}_{xy} = \{(x_j, y_i) | C_{ij} \neq 0\}$, and $\mathcal{E}_{uy} = \{(u_j, y_i) | D_{ij} \neq 0\}$. The following theorems characterize the generic number of invariant zeros of a structured system and the generic normal-rank of a structured matrix pencil in terms of the associated graph $\mathcal{H}$. The terminology *$\mathcal{Y}$-topped path* is used to denote a path with end vertex in $\mathcal{Y}$. Recall that a *linking* is a set of vertex disjoint paths.

Theorem 5 ([11], [13]): Let $P(z)$ be the matrix pencil of the structured set (A, B, C, D), and let the normal-rank of $P(z)$ be $n + m$, even after the deletion of an arbitrary row from $P(z)$. Then the generic number of invariant zeros of system (9) is equal to n minus the maximal number of vertices in $\mathcal{X}$ contained in the disjoint union of a size m linking from $\mathcal{U}$ to $\mathcal{Y}$, a set of cycles in $\mathcal{X}$, and a set of $\mathcal{Y}$-topped paths. □

Theorem 6 ([11]): Let $P(z)$ be the matrix pencil of the structured set (A, B, C, D). Then the normal-rank of $P(z)$ is generically equal to n plus the maximum size of a linking from $\mathcal{U}$ to $\mathcal{Y}$. □

To apply the above results to the problem of determining the number of invariant zeros of the set $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$, we note that all matrices in this set are essentially structured matrices, with the exception that the nonzero entries in $C_{i,\bar{\mathcal{S}}}$ and $C_{i,\mathcal{S}}$ are taken to be 1 rather than free parameters. However, one can easily show

that this fact does not affect the normal rank or the number of invariant zeros of the system (e.g., by using a technique similar to the one described in [5]). We can therefore treat $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$ as a structured set, and use the above results on structured systems to prove the following lemma for the linear iteration in (2).

Lemma 3: Let the graph of the network $\mathcal{G}$ have connectivity κ. Let x_i be any node in the network, and let $\mathcal{S}$ be any set of $2f$ nodes. If $\kappa \geq 2f+1$, then for almost any choice of weights, the set $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$ will have no invariant zeros.

Proof: We will show that the matrix pencil

$$P(z) = \begin{bmatrix} W_{\bar{\mathcal{S}}} - zI_{N-2f} & W_{\mathcal{S},\bar{\mathcal{S}}} \\ C_{i,\bar{\mathcal{S}}} & C_{i,\mathcal{S}} \end{bmatrix}$$

has full normal-rank (equal to N) even after the deletion of an arbitrary row. We will then use Theorem 5 to prove the lemma.

We start by constructing the graph $\mathcal{H}$ associated with the matrices $(W_{\bar{\mathcal{S}}}, W_{\mathcal{S},\bar{\mathcal{S}}}, C_{i,\bar{\mathcal{S}}}, C_{i,\mathcal{S}})$ in $P(z)$. For this set of matrices, note that the state vertices in the graph $\mathcal{H}$ are given by the set $\bar{\mathcal{S}}$, and the input vertices are given by the set $\mathcal{S}$. In particular, $\mathcal{H}$ can be obtained by first taking the graph of the network $\mathcal{G}$, and removing all incoming edges to nodes in $\mathcal{S}$ (since the nodes in $\mathcal{S}$ are treated as inputs in the above set of matrices). To this graph, add $\deg_i + 1$ output vertices (denoted by the set $\mathcal{Y}_i$), and place a single edge from the set $x_i \cup \mathcal{N}_i$ to vertices in $\mathcal{Y}_i$, corresponding to the single nonzero entry in each row of the matrices $C_{i,\bar{\mathcal{S}}}$ and $C_{i,\mathcal{S}}$. Furthermore, add a self loop to every state vertex to correspond to the nonzero entries on the diagonal of the weight matrix $W_{\bar{\mathcal{S}}}$.

Suppose we remove one of the rows of $P(z)$ corresponding to a vertex $v \in \bar{\mathcal{S}}$ (i.e., one of the top $N-2f$ rows of $P(z)$), and denote the resulting matrix by $\bar{P}(z)$. The generic rank of $\bar{P}(z)$ can be found by examining the associated graph, which we will denote by $\bar{\mathcal{H}}$. Note that $\bar{\mathcal{H}}$ is obtained from $\mathcal{H}$ simply by removing all incoming edges to vertex v in $\mathcal{H}$, since we removed the row corresponding to v from $P(z)$; however, all outgoing edges from v are still left in the graph (since the column corresponding to vertex v is left in matrix $\bar{P}(z)$). Thus, we see that vertex v can be treated as an input vertex in $\bar{\mathcal{H}}$, leaving $N-2f-1$ state vertices (corresponding to the set $\bar{\mathcal{S}} - v$). Since the set $\mathcal{S} \cup v$ has size $2f+1$, and since the graph $\mathcal{G}$ of the network has connectivity $\kappa \geq 2f+1$, Theorem 1 indicates that there will be a linking of size $2f+1$ from the set $\mathcal{S} \cup v$ to the set $x_i \cup \mathcal{N}_i$ in $\mathcal{G}$, and hence in $\bar{\mathcal{H}}$ (since such a linking would not use any incoming edges to $\mathcal{S} \cup v$). Furthermore, since each vertex in $x_i \cup \mathcal{N}_i$ has a one-to-one correspondence to a vertex in $\mathcal{Y}_i$, we see from Theorem 6 that the matrix pencil $\bar{P}(z)$ will generically have full normal-rank (equal to $(N-2f-1)+2f+1 = N$).

Suppose we remove one of the bottom $\deg_i + 1$ rows of $P(z)$ (corresponding to one of the output vertices in $\mathcal{Y}_i$) to form matrix $\bar{P}(z)$. The associated graph $\bar{\mathcal{H}}$ is obtained by simply removing the appropriate output vertex from $\mathcal{H}$, and since $\deg_i + 1 \geq \kappa + 1 \geq 2f + 2$, there will be at least $2f+1$ output vertices in $\bar{\mathcal{H}}$. Choose any $2f$ of the nodes in $x_i \cup \mathcal{N}_i$ such that none of the chosen nodes have an edge to the removed output vertex in $\mathcal{H}$ (this is possible since each output vertex only connects to a single vertex in $x_i \cup \mathcal{N}_i$). Denote these $2f$ nodes by $\mathcal{X}_i$. Once again, since the graph is at least $2f+1$ connected, there exists a linking of size $2f$ from $\mathcal{S}$ to $\mathcal{X}_i$, and therefore to the remaining output vertices in $\bar{\mathcal{H}}$. From Theorem 6, the matrix pencil will generically have full normal-rank (equal to $(N-2f)+2f = N$) even after deleting one of the bottom $\deg_i + 1$ rows of $P(z)$.

The above arguments show that $P(z)$ will generically have full column rank after deleting an arbitrary row. From Theorem 5, the number of invariant zeros of $P(z)$ is equal to $N-2f$ minus the maximal number of vertices in $\bar{\mathcal{S}}$ contained in the disjoint union of a size $2f$ linking from $\mathcal{S}$ to $\mathcal{Y}_i$, a cycle family in $\bar{\mathcal{S}}$, and a $\mathcal{Y}_i$-topped path family (in the graph $\mathcal{H}$). If we simply take all the self-loops in $\mathcal{H}$ (corresponding to the nonzero weights on the diagonal of the weight matrix $W_{\bar{\mathcal{S}}}$), we will have a set of disjoint cycles that covers all $N-2f$ vertices in $\bar{\mathcal{S}}$. Thus, the matrix pencil $P(z)$ will generically have no invariant zeros, thereby proving the lemma. ■

We now prove Theorem 2 (provided in Section II).

Proof: From Lemmas 2 and 3, we see that for almost any choice of weight matrix W, the set $(W, B_{\mathcal{S}}, C_i, 0)$ will have no invariant zeros, for any i and any set $\mathcal{S}$ of $2f$ nodes. Furthermore, since the set of weights for which this property does not hold has measure zero, it will hold generically (and therefore simultaneously) for all i and for all possible sets $\mathcal{S}$ of $2f$ nodes. From Theorem 4, we see that $\rho\left(\begin{bmatrix}\mathcal{O}_{i,N-1} & \mathcal{M}^{\mathcal{S}}_{i,N-1}\end{bmatrix}\right) = N + \rho\left(\mathcal{M}^{\mathcal{S}}_{i,N-1}\right)$, for all nodes i and all sets $\mathcal{S}$ of $2f$ nodes. Thus, Theorem 3 indicates every node can calculate any arbitrary function of the initial values after running the linear iteration for at most N time-steps, despite the actions of up to f malicious nodes. ■

IV. Example

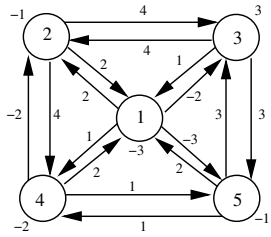

Fig. 1. Network with weights chosen from the set $\{-4,-3,-2,-1,1,2,3,4\}$.

Consider the network shown in Fig. 1. The objective in this network is for all nodes to calculate the function $\sum_{i=1}^{5} x_i^2[0]$, even if there is up to $f=1$ malicious node in the network.

A. Network Design

Examining the network, we see that the connectivity of the network is $\kappa = 3$ (since removing any two nodes still leaves a strongly connected network), and so Theorem 2 indicates that every node can calculate the desired function after running the linear iteration with almost any choice of weights for

at most $N = 5$ time-steps, despite the presence of up to one malicious node. For this example, we choose each of the edge and self weights as an independent random variable uniformly distributed in the set[3] $\{-4, -3, -2, -1, 1, 2, 3, 4\}$. The resulting weights are shown in Fig. 1, and produce the weight matrix W in (2). For brevity, we will focus on node 3 in the network. Since node 3 has access to its own value, as well as those of its neighbors (nodes 1, 2 and 5), at each time-step, the matrix C_3 is given by $C_3 = \begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&1 \end{bmatrix}$. With the above set of weights, one can verify that equation (4) is satisfied with $i = 3$ and $L = 1$ for all sets $\mathcal{S}$ of $2f = 2$ nodes, where $\mathcal{O}_{i,L}$ and $\mathcal{M}_{i,L}^{\mathcal{S}}$ are defined in equation (3). Based on Theorem 2, node 3 can find the candidate malicious node in the system after running the linear iteration for $L + 1 = 2$ time-steps if it has access to the matrices $\mathcal{O}_{3,1} = \begin{bmatrix} C_3 \\ C_3W \end{bmatrix}$ and $\mathcal{M}_{3,1}^{\mathcal{S}_j} = \begin{bmatrix} 0 \\ C_3 B_{\mathcal{S}_j} \end{bmatrix}$ for all possible sets $\mathcal{S}_j = \{x_j\}$ of one malicious node (note that there are $\binom{5}{1} = 5$ such sets). For each set $\mathcal{S}_j$, we also calculate the matrices $\mathcal{P}_{3,1}^{\mathcal{S}_j}$ given in (7), satisfying $\mathcal{P}_{3,1}^{\mathcal{S}_j}\mathcal{M}_{3,1}^{\mathcal{S}_j} = 0$ and $\mathcal{P}_{3,1}^{\mathcal{S}_j}\mathcal{O}_{3,1} = I_N$, and provide these matrices to node 3. We will omit the explicit values of these matrices in the interest of space. At this point, node 3 has all the information it needs to calculate the function $\sum_{i=1}^{5} x_i^2[0]$, even in the presence of one malicious node. Performing the same analysis for all nodes i, one finds that equation (4) is satisfied with $L = 1$ for all i and all sets $\mathcal{S}$ of $2f = 2$ nodes. Therefore, all nodes can calculate the function $\sum_{i=1}^{5} x_i^2[0]$ after running the linear iteration for $L + 1 = 2$ time-steps.

B. Performing Function Calculation

Suppose that the initial values of the nodes are $\mathbf{x}[0] = \begin{bmatrix} 3 & -1 & 4 & -4 & 7 \end{bmatrix}'$, and the nodes run the linear iteration with the weights shown in Fig. 1; however, suppose that node 1 is malicious and updates its value as

$$x_1[1] = -3x_1[0] + 2x_2[0] + 1x_3[0] + 2x_4[0] + 2x_5[0] - 8 \ ,$$

i.e., during the first time-step, it commits an additive error of $u_1[0] = -8$. All other nodes follow the predefined (correct) strategy of updating their values according to the weighted average specified by the weight matrix W. After one iteration, the values of all nodes become $\mathbf{x}[1] = \begin{bmatrix} -9 & 31 & 23 & 14 & -8 \end{bmatrix}'$. Since the values seen by node 3 at time-step k are given by $\mathbf{y}_3[k] = C_3\mathbf{x}[k]$, the values seen by node 3 over the two time-steps of the linear iteration are $\mathbf{y}_3[0:1] = \begin{bmatrix} 3 & -1 & 4 & 7 & -9 & 31 & 23 & -8 \end{bmatrix}'$. Node 3 can now use these values to eliminate the effects of the malicious node on the system. As discussed in Theorem 2, node 3 finds the first set $\mathcal{S}_j$ consisting of one node that satisfies equation (6); in this case, node 3 finds that this equation is satisfied for $\mathcal{S}_1 = \{x_1\}$. Node 3 then uses equation (8) to multiply the vector $\mathbf{y}_3[0:1]$ by the matrix $\mathcal{P}_{3,1}^{\mathcal{S}_1}$, and this produces $\mathcal{P}_{3,1}^{\mathcal{S}_1}\mathbf{y}_3[0:1] = \mathbf{x}[0]$. Node 3 can now calculate the function $\sum_{i=1}^{5} x_i^2[0]$, and obtains the value 91. All other nodes obtain the function value in the same way, and the system reaches consensus in 2 time-steps. Note that no scheme can produce consensus in this network in fewer than 2 time-steps (since the diameter of the graph is 2), and so the linear iterative strategy is time optimal for this graph, even in the presence of one malicious node.

[3] In general, the result in Theorem 2 will hold with high probability if one chooses the weights for the linear iteration from a continuous distribution over the real numbers (such as a Gaussian distribution). For this pedagogical example, however, it will suffice to consider a distribution on a small set of integers.

V. Summary

In this paper, we considered the problem of enabling nodes to calculate functions in a network in the presence of malicious or malfunctioning nodes. We utilized a linear iteration to provide this capability, and showed that if the network is at least $(2f + 1)$-connected, then for almost any choice of weights, it is possible for any node i to follow a checking/correction strategy that enables it to calculate any arbitrary function despite the presence of up to f malicious nodes (after running the linear iteration for at most N time-steps). This result complements the analysis in [6], where it was shown that if the connectivity of the network satisfies $\kappa \leq 2f$, it is possible for f malicious nodes to prevent some nodes from calculating any function of all values, regardless of the choice of weight matrix and the number of steps used in the linear iteration.

References

[1] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755–764, Apr. 2005.

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

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

[4] S. Sundaram and C. N. Hadjicostis, "Distributed consensus and linear functional calculation in networks: An observability perspective," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 99–108.

[5] ——, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, May 2008, to appear.

[6] ——, "Distributed function calculation via linear iterations in the presence of malicious agents – part I: Attacking the network," in *Proceedings of the American Control Conference*, 2008.

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

[8] C.-T. Chen, *Linear System Theory and Design*. Holt, Rinehart and Winston, 1984.

[9] M. L. J. Hautus, "Strong detectability and observers," *Linear Algebra and its applications*, vol. 50, pp. 353–368, Apr. 1983.

[10] D. Rappaport and L. M. Silverman, "Structure and stability of discrete-time optimal systems," *IEEE Transactions on Automatic Control*, vol. 16, no. 3, pp. 227–233, June 1971.

[11] J. van der Woude, "The generic number of invariant zeros of a structured linear system," *SIAM Journal on Control and Optimization*, vol. 38, no. 1, pp. 1–21, Nov. 1999.

[12] L. M. Silverman, "Discrete Riccati equations: Alternative algorithms, asymptotic properties and system theory interpretations," in *Control and Dynamic Systems*, C. T. Leondes, Ed. Academic Press, 1976, vol. 12, pp. 313–386.

[13] 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.