Linear Iterative Strategies for Transmitting Streams of Values Through Sensor Networks

Shreyas Sundaram and Christoforos N. Hadjicostis

***Abstract*— For any given fixed network of interconnected nodes, with some nodes designated as sources and some nodes designated as sinks, we investigate the problem of transmitting a stream of values from every source node to all of the sink nodes (possibly after some delay). We study linear iterative strategies for transmitting this information through the network, whereby at each time-step, each node in the network transmits a value that is a linear combination of the most recent transmissions of its neighbors. We show that this linear iterative strategy can be conveniently modeled as a linear dynamical system in state-space form. We then use techniques from control theory pertaining to dynamic system inversion and structured linear systems to show that each sink node can reconstruct the data streams if and only if there are node disjoint paths in the network from the set of all source nodes to each sink node. Furthermore, this reconstruction can be accomplished after a delay of at most $N - |\mathcal{S}| + 1$ time-steps (where N is the number of nodes in the network, and $|\mathcal{S}|$ is the number of sources). This holds true for almost any choice of weights in the linear iteration.**

I. Introduction

A common requirement in distributed systems and networks is to transmit streams of values between sets of nodes. For example, in networked control systems, a set of sensors measure certain quantities in a plant and send that information through a network to controllers and actuators in order to control the plant [1]. More generally, sensor networks typically contain a set of nodes that periodically sense data generated at sources (such as intruders [2], chemical or biological plumes [3], etc.); these sensing nodes are then required to send this stream of data through the network to a set of base-stations (or sinks) which will then sound an alarm or take other appropriate action [4].

Various protocols and algorithms have been developed for information dissemination in networks by the computer science, communication, and control communities [5], [6], [7]. A particular strategy that has attracted a large amount of attention in the control systems community is that of *linear iterations*; in this strategy, 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 [8] 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 (usually a weighted linear combination of the initial values of the nodes). Recently, it was shown in [9], [10] that this linear iterative strategy can actually be applied to the more general function calculation problem, allowing any node in networks with time-invariant topologies to calculate arbitrary functions of the initial node values in a finite number of time-steps (upper bounded by the size of the network), even when some of the nodes are potentially malicious or faulty.

In this paper, we develop linear iterative strategies for transmitting streams of values in networks (as opposed to the above works, which were concerned with transmitting only the initial values of the nodes). In this context, we show that the linear iterative strategy can be viewed as a *linear dynamical system with unknown inputs*. This insight allows us to draw upon results from control theory pertaining to *dynamic system inversion* [11] and *structured systems* [12] to derive necessary and sufficient conditions on the network topology in order for every sink node to be able to reconstruct the source streams. Furthermore, our approach immediately produces a decoding strategy for the sink nodes to follow in order to reconstruct the input streams, along with an upper bound on the delay required to do so. Later in the paper, we will connect this approach to *linear network codes* that have been developed in the communications literature for transmitting information through networks.

II. Notation and Background

In our development, we use $\mathbf{e}_i$ to denote the column vector with a 1 in its i-th position and 0's elsewhere. The symbol $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. The transpose of matrix $\mathbf{A}$ is denoted by $\mathbf{A}'$. We will denote the cardinality of a set $\mathcal{S}$ by $|\mathcal{S}|$.

We will require the following graph-theoretic terminology. Further background on graph theory can be found in standard texts, such as [13]. A graph is an ordered pair $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where $\mathcal{X} = \{x_1, \ldots, x_N\}$ is a set of vertices (or nodes), and $\mathcal{E}$ is a set of ordered pairs of 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 $\deg_i = |\mathcal{N}_i|$.

This material is based upon work supported in part by the National Science Foundation under NSF ITR Award 0426831 and NSF CNS Award 0834409. 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 research leading to these results has also received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements INFSO-ICT-223844 and PIRG02-GA-2007-224877.

S. Sundaram is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, USA. E-mail: ssundarm@illinois.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.

A *path* P from vertex x_{i_0} to vertex x_{i_t} is a sequence of vertices $x_{i_0}, x_{i_1}, \ldots, x_{i_t}$ such that $(x_{i_j}, x_{i_{j+1}}) \in \mathcal{E}$ for $0 \le j \le t-1$. Paths P_1 and P_2 are *vertex disjoint* if they have no vertices in common. A set of paths $P_1, P_2, \ldots, P_r$ are vertex disjoint if the paths are pairwise vertex disjoint. Given two subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{X}$, a set of r vertex disjoint paths, each with start vertex in $\mathcal{X}_1$ and end vertex in $\mathcal{X}_2$, is called an r*-linking* from $\mathcal{X}_1$ to $\mathcal{X}_2$.

III. Problem Formulation

Consider a network modeled by the 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). Let $\mathcal{S} \subseteq \mathcal{X}$ be a set of *source* nodes, and let $\mathcal{T} \subseteq \mathcal{X}$ be a set of *sink* nodes (note that $\mathcal{S}$ and $\mathcal{T}$ are not necessarily disjoint). We assume that the operation of the network proceeds in a series of time-steps. At each time-step k, each source node $x_i \in \mathcal{S}$ senses new information from an entity external to the network; for example, this new information could represent measurements of the environment by the node. The goal is for every source node to transmit this stream of values (i.e., the value that it receives at each time-step) to each sink node via the network. In order to disseminate information, all nodes in the network can transmit a value at each time-step k based on some strategy that adheres to the constraints imposed by the network topology; let $x_i[k]$ denote the value transmitted by the i-th node at time-step k. The scheme that we study in this paper makes use of linear iterations; specifically, the value transmitted by each node x_i at time-step $k+1$ is given by

$$x_i[k+1] = \begin{cases} w_{ii}x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij}x_j[k] & \text{if } x_i \notin \mathcal{S}, \\ w_{ii}x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij}x_j[k] + u_i[k] & \text{if } x_i \in \mathcal{S}. \end{cases} \quad (1)$$

where the w_{ij}'s are a set of weights, and $u_i[k]$ is the new value (information) obtained by source node x_i at time-step k. If we let $\mathcal{S} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{|\mathcal{S}|}}\}$ denote the set of source nodes, and aggregate the values transmitted by all nodes at time-step k into the value vector $\mathbf{x}[k] = \begin{bmatrix} x_1[k] & x_2[k] & \cdots & x_N[k] \end{bmatrix}'$, the transmission strategy for the entire system can be represented as

$$\mathbf{x}[k+1] = \underbrace{\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}}_{\mathbf{W}} \mathbf{x}[k] + \underbrace{\begin{bmatrix} \mathbf{e}_{i_1} & \mathbf{e}_{i_2} & \cdots & \mathbf{e}_{i_{|\mathcal{S}|}} \end{bmatrix}}_{\mathbf{B}_{\mathcal{S}}} \underbrace{\begin{bmatrix} u_{i_1}[k] \\ u_{i_2}[k] \\ \vdots \\ u_{i_{|\mathcal{S}|}}[k] \end{bmatrix}}_{\mathbf{u}_{\mathcal{S}}[k]} \quad (2)$$

for all nonnegative integers k. In the above equation, the weight $w_{ij} = 0$ if $x_j \notin \mathcal{N}_i$ (i.e., if $(x_j, x_i) \notin \mathcal{E}$), and we take $\mathbf{x}[0]$ to be the vector of all zeros.

At each time-step k, sink node $x_i \in \mathcal{T}$ has access to its own transmitted value as well as the values transmitted by its neighbors. Letting $\mathbf{y}_i[k]$ denote the transmitted values seen by node x_i during time-step k, we obtain the equation

$$\mathbf{y}_i[k] = \mathbf{C}_i\mathbf{x}[k], \quad (3)$$

where $\mathbf{C}_i$ is a $(\deg_i + 1) \times N$ matrix with a single 1 in each row capturing the positions of the vector $\mathbf{x}[k]$ that are available to node x_i (i.e., these positions correspond to neighbors of node x_i, along with x_i itself). The values $\mathbf{y}_i[k]$, $k = 0, 1, \ldots$, characterize the ability of x_i to reconstruct the source values. In the next section, we will discuss conditions on the network topology and ways to choose the weights for each node so that each sink node in $\mathcal{T}$ can recover the source vector $\mathbf{u}_{\mathcal{S}}[k]$ based on $\mathbf{y}_i[0], \mathbf{y}_i[1], \ldots, \mathbf{y}_i[k], \ldots, \mathbf{y}_i[k+L]$ for $k = 0, 1, \ldots$ (i.e., with a delay of L time-steps, for some L). Specifically, we will demonstrate the following key result.

Theorem 1: Let $\mathcal{G}$ denote the graph of a fixed network, and let $\mathcal{S}$ denote a set of source nodes, and $\mathcal{T}$ denote a set of sink nodes. Assume that each node in the network follows the linear iterative strategy (2). Then a sink node $x_i \in \mathcal{T}$ can reconstruct all of the streams of data originating at the source nodes if and only if there exists a $|\mathcal{S}|$-linking from $\mathcal{S}$ to $\{x_i\} \cup \mathcal{N}_i$. In addition, if this condition is satisfied for every sink node $x_i \in \mathcal{T}$, then each sink node can reconstruct the input streams from all sources. The reconstruction delay is at most $N - |\mathcal{S}| + 1$ time-steps and the iteration can be performed with almost any choice of weight matrix $\mathbf{W}$. $\square$

In the above theorem, the phrase "almost any" means that the set of weights for which the theorem does not hold is of Lebesgue measure zero.

IV. Reconstruction by Sink Nodes

To obtain the proof of Theorem 1, we start by making the observation that the linear iterative strategy given by equations (2) and (3) together form a linear system in state-space form [14]. Since the quantity $\mathbf{u}_{\mathcal{S}}[k]$ in (2) is unknown to the sink nodes, such systems are termed *linear systems with unknown inputs* (e.g., see [11]). Often, it is of interest to "invert" the system in order to reconstruct the unknown inputs; this problem has been studied under the moniker of *dynamic system inversion*.

A. Background on System Inversion

Consider a linear system of the form

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

where $\mathbf{u}[k] \in \mathbb{R}^m$ is an unknown input, $\mathbf{y}[k] \in \mathbb{R}^p$ is the output of the system, and $\mathbf{x}[k] \in \mathbb{R}^N$ is the system state (with $\mathbf{x}[0]$ known). The output of the system over $L+1$ time-steps (for some nonnegative integer L) is given by

$$\underbrace{\begin{bmatrix} \mathbf{y}[k] \\ \mathbf{y}[k+1] \\ \mathbf{y}[k+2] \\ \vdots \\ \mathbf{y}[k+L] \end{bmatrix}}_{\mathbf{y}[k:k+L]} = \underbrace{\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^L \end{bmatrix}}_{\mathcal{O}_L} \mathbf{x}[k] + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{L-1}\mathbf{B} & \mathbf{CA}^{L-2}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix}}_{\mathcal{M}_L} \underbrace{\begin{bmatrix} \mathbf{u}[k] \\ \mathbf{u}[k+1] \\ \mathbf{u}[k+2] \\ \vdots \\ \mathbf{u}[k+L-1] \end{bmatrix}}_{\mathbf{u}[k:k+L-1]} . \quad (5)$$

The transfer function matrix of system (4) is given by $\mathbf{T}(z) \equiv \mathbf{C}\left(z\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B}$, and is a $p \times m$ matrix of rational functions of z.

Definition 1: The system (4) is said to have an L-delay inverse if there exists a system with transfer function $\widehat{\mathbf{T}}(z)$ such that $\widehat{\mathbf{T}}(z)\mathbf{T}(z) = \frac{1}{z^L}\mathbf{I}_m$; note that $\widehat{\mathbf{T}}(z)$ is an $m \times p$ matrix. The system is invertible if it has an L-delay inverse for some finite L. The least integer L for which an L-delay inverse exists is called the inherent delay of the system. □

In order for the system to be invertible, its transfer function matrix must clearly have rank m over the field of rational functions in z. If the system has an L-delay inverse, it was shown in [11] (and other works) that one can construct an *inverse system* that operates on the outputs $\mathbf{y}[0], \mathbf{y}[1], \ldots, \mathbf{y}[k+L]$ and reconstructs the input $\mathbf{u}[k]$. In fact, it was shown in [11] that this inverse system can be designed purely in the time-domain with state-space models, rather than having to manipulate rational transfer function matrices. For instance, the following result from [11] and [15] provides a test for invertibility directly in terms of the system matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$.

Theorem 2 ([11], [15]): For any nonnegative integer L,

$$\text{rank}(\mathcal{M}_L) - \text{rank}(\mathcal{M}_{L-1}) \leq m \quad (6)$$

with equality if and only if the system has an L-delay inverse (note that rank(M_{-1}) is defined to be zero). If the system is invertible, its inherent delay will not exceed $L = N - m + 1$. □

Based on the form of the matrix $\mathcal{M}_L$ in equation (5), note that condition (6) in the above theorem indicates that the system is invertible if and only if the first m columns of $\mathcal{M}_L$ are linearly independent of each other and of all other columns in the matrix. We will discuss the construction of the inverse system in more detail when we apply the above result to design decoders for the sink nodes in the network. For now, note from the above discussion that the problem of transmitting streams of data via linear iterative strategies is equivalent to that of linear system inversion. In particular, if one can choose the weights in the weight matrix $\mathbf{W}$ such that the linear system given by equations (2) and (3) is invertible for each $x_i \in \mathcal{T}$, then each sink node will be able to recover the source vector $\mathbf{u}_\mathcal{S}[k]$ for all $k \geq 0$ by constructing an inverse system. The question thus becomes: When is it possible to choose the weight matrix to cause the system to be invertible, and how should it be chosen? To help answer this question, we make use of some results from a branch of control theory known as *structured system theory*.

B. Structured Systems

A linear system of the form (4) is said to be *structured* if each entry of the matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ is either a fixed zero or an independent free parameter [12]. Interestingly, such systems have certain properties that can be inferred purely from the zero/nonzero structure of the system matrices; these properties will hold for almost any choice of free parameters (i.e., the set of parameters for which the property does not hold has Lebesgue measure zero [12]), and thus these properties are called *generic*. Of particular relevance to this paper is the *generic normal rank* of the transfer function matrix of a structured system, which is the maximum rank (over the field of rational functions in z) of the transfer function matrix over all possible choices of free parameters.

To analyze structural properties of linear systems, one first associates a graph $\mathcal{H}$ with the structured set $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ 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 $\mathcal{E}_{xx} \cup \mathcal{E}_{ux} \cup \mathcal{E}_{xy}$, where $\mathcal{E}_{xx} = \{(x_j, x_i) | \mathbf{A}_{ij} \neq 0\}$, $\mathcal{E}_{ux} = \{(u_j, x_i) | \mathbf{B}_{ij} \neq 0\}$, and $\mathcal{E}_{xy} = \{(x_j, y_i) | \mathbf{C}_{ij} \neq 0\}$. The following theorem characterizes the generic normal rank of the transfer function of a structured linear system in terms of the associated graph $\mathcal{H}$.

Theorem 3 ([12]): Let the graph of a structured linear system be given by $\mathcal{H}$. Then the generic normal rank of the transfer function of the system is equal to the maximal size of a linking in $\mathcal{H}$ from $\mathcal{U}$ to $\mathcal{Y}$. □

C. Reconstructing the Source Streams

To apply these concepts to the problem of transmitting streams of values in networks, note that the tuple $(\mathbf{W}, \mathbf{B}_\mathcal{S}, \mathbf{C}_i)$ in equations (2) and (3) essentially defines a structured linear system, with the only exception being that the nonzero entries of $\mathbf{C}_i$ and $\mathbf{B}_\mathcal{S}$ are taken to be "1", rather than free parameters. However, this is easily remedied by defining the matrices $\Lambda_i = \text{diag}\left(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,\deg_i + 1}\right)$ and $\Gamma_i = \text{diag}\left(\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,|\mathcal{S}|}\right)$, where the $\lambda_{i,j}$'s and the $\gamma_{i,j}$'s are a set of independent free parameters. If we form the matrices $\bar{\mathbf{C}}_i = \Lambda_i \mathbf{C}_i$ and $\bar{\mathbf{B}}_{i,\mathcal{S}} = \mathbf{B}_\mathcal{S}\Gamma_i$, then matrix $\bar{\mathbf{C}}_i$ has a single independent free parameter in each row, and matrix $\bar{\mathbf{B}}_{i,\mathcal{S}}$ has a single independent free parameter in each column; these matrices therefore qualify as valid structured matrices. The transfer function for the system given by the tuple $(\mathbf{W}, \bar{\mathbf{B}}_{i,\mathcal{S}}, \bar{\mathbf{C}}_i)$ is

$$\begin{aligned}\bar{\mathbf{T}}(z) &= \bar{\mathbf{C}}_i \left(z\mathbf{I} - \mathbf{W}\right)^{-1} \bar{\mathbf{B}}_{i,\mathcal{S}} \\ &= \Lambda_i \mathbf{C}_i \left(z\mathbf{I} - \mathbf{W}\right)^{-1} \mathbf{B}_{\mathcal{S}} \Gamma_i \ \ ,\end{aligned}$$

and if the matrices Λ_i and Γ_i are invertible, the transfer function for the tuple $(\mathbf{W}, \bar{\mathbf{B}}_{i,\mathcal{S}}, \bar{\mathbf{C}}_i)$ will have the same rank as the transfer function for the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_i)$. Thus, for the purposes of analyzing the generic normal rank, we can assume without loss of generality that the nonzero entries in $\mathbf{B}_{\mathcal{S}}$ and $\mathbf{C}_i$ are all ones (i.e., take Λ_i and Γ_i to be the identity matrices).

In order to transmit streams via linear iterative strategies, the linear system with unknown inputs given by equations (2) and (3) must be invertible for each $x_i \in \mathcal{T}$, which means that its transfer function $\mathbf{C}_i(z\mathbf{I} - \mathbf{W})^{-1}\mathbf{B}_{\mathcal{S}}$ must have rank $|\mathcal{S}|$ over the field of rational functions in z. In particular, this requires that each of these transfer functions have generic normal rank equal to $|\mathcal{S}|$. To check this, we can construct graph $\mathcal{H}_i$ for each tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_i)$, $x_i \in \mathcal{T}$, as follows: first take the graph of the network $\mathcal{G}$, and add $\deg_i + 1$ output vertices (denoted by the set $\mathcal{Y}_i$) and $|\mathcal{S}|$ input vertices (denoted by the set $\mathcal{U}$). Next, 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 matrix $\mathbf{C}_i$, and a single edge from each input vertex to each source node (corresponding to the single nonzero entry in each column of the matrix $\mathbf{B}_{\mathcal{S}}$). Furthermore, add a set of self loops to every state vertex to correspond to the nonzero entries on the diagonal of the weight matrix $\mathbf{W}$. Now, according to Theorem 3, the structured system given by the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_i)$ is generically invertible if and only if there exists a linking of size $|\mathcal{S}|$ from $\mathcal{U}$ to $\mathcal{Y}_i$ in the graph $\mathcal{H}_i$. Since each input vertex connects to exactly one source node, and since each output vertex connects to exactly one vertex in $\{x_i\} \cup \mathcal{N}_i$, a linking of size $|\mathcal{S}|$ from $\mathcal{U}$ to $\mathcal{Y}_i$ corresponds to a linking of size $|\mathcal{S}|$ from the source nodes $\mathcal{S}$ to $\{x_i\} \cup \mathcal{N}_i$.

We are now ready to prove the main theorem of the paper (Theorem 1, given at the end of Section III).

Proof: (Sufficiency.) Suppose that for every $x_i \in \mathcal{T}$, the network contains a $|\mathcal{S}|$-linking from $\mathcal{S}$ to $\{x_i\} \cup \mathcal{N}_i$. As described above, the graph $\mathcal{H}_i$ for the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_i)$ will therefore contain an $|\mathcal{S}|$-linking from the inputs to the outputs, for every $x_i \in \mathcal{T}$. For any particular $x_i \in \mathcal{T}$, Theorem 3 then indicates that for almost any choice of weight matrix $\mathbf{W}$, the transfer function $\mathbf{C}_i(z\mathbf{I} - \mathbf{W})^{-1}\mathbf{B}_{\mathcal{S}}$ will have rank $|\mathcal{S}|$, and will therefore be invertible. Since this property holds generically (with respect to the choice of weight matrix $\mathbf{W}$), it will hold simultaneously for all sink nodes in $\mathcal{T}$. Using this fact, we can now construct a decoder for each sink node as follows.

From (2) and (3), the values seen by sink node x_i over $L_i + 1$ time-steps (for some nonnegative integer L_i) are given by

$$\mathbf{y}_i[k : k + L_i] = \mathcal{O}_{i,L_i}\mathbf{x}[k] + \mathcal{M}_{i,L_i}\mathbf{u}_{\mathcal{S}}[k : k + L_i - 1] \ \ , \quad (7)$$

where

$$\begin{aligned}
\mathbf{y}_i[k : k + L_i] &= \begin{bmatrix}\mathbf{y}_i'[k] & \mathbf{y}_i'[k+1] & \cdots & \mathbf{y}_i'[k+L_i]\end{bmatrix}' \\
\mathbf{u}_{\mathcal{S}}[k : k + L_i - 1] &= \begin{bmatrix}\mathbf{u}_{\mathcal{S}}'[k] & \cdots & \mathbf{u}_{\mathcal{S}}'[k+L_i-1]\end{bmatrix}' \\
\mathcal{O}_{i,L_i} &= \begin{bmatrix}\mathbf{C}_i' & (\mathbf{C}_i\mathbf{W})' & \cdots & (\mathbf{C}_i\mathbf{W}^{L_i})'\end{bmatrix}' \\
\mathcal{M}_{i,L_i} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{C}_i\mathbf{B}_{\mathcal{S}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_i\mathbf{W}^{L_i-1}\mathbf{B}_{\mathcal{S}} & \mathbf{C}_i\mathbf{W}^{L_i-2}\mathbf{B}_{\mathcal{S}} & \cdots & \mathbf{C}_i\mathbf{B}_{\mathcal{S}} \end{bmatrix}.
\end{aligned}$$

From Theorem 2, there exists an integer L_i (upper-bounded by $N - |\mathcal{S}| + 1$) such that the first $|\mathcal{S}|$ columns in the matrix $\mathcal{M}_{i,L_i}$ will be linearly independent of each other and of the remaining columns in $\mathcal{M}_{i,L_i}$. Therefore, there exists a matrix $\mathbf{K}_i$ for node x_i such that $\mathbf{K}_i\mathcal{M}_{i,L_i} = \begin{bmatrix}\mathbf{I}_{|\mathcal{S}|} & 0 & \cdots & 0\end{bmatrix}$. Left-multiplying (7) by $\mathbf{K}_i$ and rearranging, one obtains

$$\mathbf{u}_{\mathcal{S}}[k] = \mathbf{K}_i\mathbf{y}_i[k : k + L_i] - \mathbf{K}_i\mathcal{O}_{i,L_i}\mathbf{x}[k] \ \ , \quad (8)$$

and substituting this into equation (2), one gets the state update equation

$$\mathbf{x}[k+1] = \left(\mathbf{W} - \mathbf{B}_{\mathcal{S}}\mathbf{K}_i\mathcal{O}_{i,L_i}\right)\mathbf{x}[k] + \mathbf{B}_{\mathcal{S}}\mathbf{K}_i\mathbf{y}_i[k : k + L_i]. \quad (9)$$

Equations (9) and (8) together form the decoder for node x_i. If the decoder is initialized with state $\mathbf{x}[0] = 0$, the output equation (8) will produce the source values $\mathbf{u}_{\mathcal{S}}[k]$ for each $k \geq 0$ (based on the transmissions seen by node x_i up to time-step $k + L_i$). Each sink node x_i obtains a decoder in this way, and can therefore reconstruct the source values after a delay of L_i time-steps. This concludes the proof of sufficiency.

(Necessity.) This part of the proof follows straightforwardly from Theorem 3 and the fact that, if the transfer function for the system given by (2) and (3) is not of full column rank for some i, then there exists a nonzero sequence of inputs $\mathbf{u}_{\mathcal{S}}[k]$, $k = 0, 1, \ldots$, such that $\mathbf{y}_i[k] = 0$, $k = 0, 1, \ldots$. These inputs are indistinguishable from the input sequence consisting of all zeros, and thus node x_i cannot uniquely identify the inputs to the system. ■

Remark 1: Note from Theorem 1 that the invertibility of the system defined by the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_i)$ is dependent solely on the existence of a sufficiently large linking from the source nodes to the sink nodes (and their neighbors). This implies that the self-weights on the diagonal of the weight matrix are not strictly necessary in order to form an invertible system (since they do not appear in the linking anyway). For this reason, the self-weights can be set to zero, if desired. □

V. Comparisons to Existing Work in Network Coding

We now compare the results presented in this paper to information transmission schemes that have been proposed in the communications literature. The traditional approach to information dissemination in networks has been to treat data packets as distinct entities, and to route them through the network without performing intermediate operations on the packets (e.g., see the discussion in [16]). In recent years, however, the concept of *network coding* has generated a great deal of interest, as it discards the notion of treating packets as individual elements, and instead allows intermediate nodes in the network to mathematically combine several packets into a single packet for transmission [16], [17], [6], [18]. Of particular interest to this paper are *linear network codes*, where the packet that is transmitted by each node is a linear combination of the incoming packets. It has been shown that, for certain networks, network coding can be used to achieve the transmission capacity of the network even when traditional store-and-forward approaches fail [17].

When the network topology contains cycles, the input packets to some nodes in the network could be functions of previous output packets at those nodes. To deal with this phenomenon, the network coding literature introduces a delay parameter z into the network (e.g., by modeling either the links or the nodes as having delays), and views the packets generated by each node as a power series in z [6], [18], [16], thereby producing a *convolutional network code*. Decoders for such codes are designed by forming a transfer function matrix (with elements that are rational functions in z) from the source stream to the sink nodes, and then essentially inverting this matrix [6], [16]. For example, [19] discussed an algorithm for choosing the weights for convolutional codes so that the transfer function from the source nodes to the sink nodes will have full rank; the decoding procedure then involves solving a set of linear equations in the delay variable z. However, a drawback of these existing approaches is the need to construct and manipulate (potentially large) matrices of rational functions.

In contrast, our decoding procedure is based on the linear state-space model given by equations (2) and (3); inspired by results pertaining to dynamic system inversion, we show that every sink node only has to examine the values that it receives over a certain length of time in order to decode the source streams. Furthermore, the operations performed by the sink node involve purely numerical matrices, and do not require manipulation of transfer function matrices in the delay variable z. Indeed, this was the main motivation that drove the development of dynamic inversion schemes in the control literature [11], and there are various results in this area that can be leveraged to design *system inverters* (or decoders, in our context) purely in the state-space domain (e.g., see [11], [20]). This connection to the field of dynamic system inversion also opens up various areas for future research, some of which we now describe.

VI. Potential Extensions

There are a variety of interesting extensions that can be pursued within the framework that we have developed in this paper. For example, the decoding procedure specified by the proof of Theorem 1 essentially requires each sink node to reconstruct the values transmitted by all nodes in the network at each time-step (i.e., the decoder for each sink node has dimension equal to the number of nodes in the system). However, based on the connections that we have made in this paper between linear network coding and the theory of dynamic system inversion, it may be possible to design more efficient decoders by using results from the topic of *minimal dynamic inversion*, which consider the problem of designing system inverses of smallest possible dimension [21], [22].

As another extension, one can investigate conditions on the network topology that are needed for each sink node to reconstruct only the streams from *some* of the source nodes, or more generally, some function of the source streams. In this case, the condition that there exist an $|\mathcal{S}|$-linking from $\mathcal{S}$ to each sink node and its neighbors will no longer be necessary; however, with this relaxation, each sink node will no longer be able to determine the values transmitted by all nodes in the system, and thus designing a decoder for the sink nodes will become more complicated. One possible way to handle this could be to make use of results on *partial system inversion*, which deal with the topic of constructing a system inverse that recovers only some of the unknown inputs to a linear system [23], [24]. Extending such results and leveraging them to design decoders for the sink nodes would be an interesting avenue for future research.

VII. Example

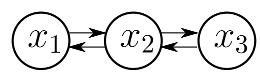

Fig. 1. Nodes x_1 and x_3 wish to exchange their streams of values via x_2.

Consider the network with three nodes $\{x_1, x_2, x_3\}$ shown in Fig. 1. The set of source nodes in the network is given by $\mathcal{S} = \{x_1, x_3\}$, and those nodes are also the sink nodes in the network (i.e., $\mathcal{T} = \mathcal{S}$). In other words, the objective in the network is for nodes x_1 and x_3 to exchange streams of values, via the intermediate node x_2. This is a standard example used to demonstrate the benefits of network coding in wireless networks (e.g., see [25]), and we revisit it here in order to demonstrate our design procedure. In order to transmit these streams, Theorem 1 indicates that there must exist a linking of size $|\mathcal{S}| = 2$ from the set $\mathcal{S}$ to node x_1 and its neighbor x_2, and also from $\mathcal{S}$ to node x_3 and its neighbor x_2. Both of these conditions are satisfied (e.g., a linking of size 2 from $\mathcal{S}$ to the set $\{x_1, x_2\}$ is given by the single node x_1 and the edge (x_3, x_2)), and so nodes x_1 and x_3 will be able to recover each other's streams (and clearly their own streams) for almost any choice of weight matrix in (2). For this pedagogical example, it will suffice to simply take the nonzero weights to be 1, which produces the weight matrix $\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$; recall that the self weights on the diagonal of $\mathbf{W}$ do not play any role in system invertibility (since Theorem 1 is only concerned with the existence of a linking), and so we can set those weights to be zero. To complete the state-space model in (2), we note that $\mathbf{B}_{\mathcal{S}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}'$.

Once the weight matrix for the code is chosen, one must design the decoders for each sink node. For brevity, we will focus on node x_1 in the network. Since node x_1 has access to its own transmitted value, as well as that of its neighbor (node x_2) at each time-step, we have $\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$.

The next step is to determine the smallest delay L_1 for which the system given by the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_1)$ is invertible. Following Theorem 2, we have to find the smallest integer L_1 for which $\text{rank}\,(\mathcal{M}_{1,L_1}) - \text{rank}\,(\mathcal{M}_{1,L_1-1}) = 2$. For $L_1 = 2$, we have

$$\mathcal{M}_{1,2} = \begin{bmatrix} 0 & 0 \\ \mathbf{C}_1\mathbf{B}_{\mathcal{S}} & 0 \\ \mathbf{C}_1\mathbf{W}\mathbf{B}_{\mathcal{S}} & \mathbf{C}_1\mathbf{B}_{\mathcal{S}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} ,$$

and we find that $\text{rank}\,(\mathcal{M}_{1,L_1}) - \text{rank}\,(\mathcal{M}_{1,L_1-1}) = 2$ (i.e., node x_1 can recover both streams after a delay of 2 time-steps). The matrix $\mathcal{O}_{1,2}$ in (7) is given by

$$\mathcal{O}_{1,2} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}' .$$

The matrix $\mathbf{K}_1$ satisfying $\mathbf{K}_1\mathcal{M}_{1,2} = \begin{bmatrix} \mathbf{I}_2 & 0 \end{bmatrix}$ is given by

$$\mathbf{K}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} .$$

The decoder for node x_1 is obtained by substituting the above matrices into the decoder equations (9) and (8), with $L_1 = 2$, which produces

$$\begin{aligned}
\mathbf{x}[k+1] &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}[k] + \\
&\qquad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1[k] \\ \mathbf{y}_1[k+1] \\ \mathbf{y}_1[k+2] \end{bmatrix} \\
\mathbf{u}_{\mathcal{S}}[k] &= \begin{bmatrix} 0 & -1 & 0 \\ 0 & -1 & 0 \end{bmatrix} \mathbf{x}[k] + \\
&\qquad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1[k] \\ \mathbf{y}_1[k+1] \\ \mathbf{y}_1[k+2] \end{bmatrix} .
\end{aligned}$$

The decoder for node x_3 is found by following an identical procedure.

VIII. Summary

In this paper, we considered the problem of transmitting streams of values in sensor networks via linear iterative strategies. We showed that such strategies can be compactly represented as a linear system with unknown inputs in state-space form. We then used concepts pertaining to system inversion and structured system theory to show that streams can be transmitted if and only if there exist vertex disjoint paths from the set of source nodes to each sink node and its neighbors. Furthermore, we showed that if this topographical condition is satisfied, then almost any choice of weights in the linear iteration will allow the sink nodes to reconstruct the source streams after a finite delay.

References

[1] C. L. Robinson and P. R. Kumar, "Optimizing controller location in networked control systems with packet drops," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 661–671, May 2008.

[2] O. Dousse, C. Tavoularis, and P. Thiran, "Delay of intrusion detection in wireless sensor networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2006, pp. 155–165.

[3] G. Nofsinger and G. Cybenko, "Distributed chemical plume process detection," in *Proceedings of the IEEE Military Communications Conference*, 2005, pp. 1076–1082.

[4] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4–5, pp. 436–463, 2005.

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

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

[7] 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.

[8] 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.

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

[10] ——, "Distributed function calculation via linear iterations in the presence of malicious agents," in *Proceedings of the American Control Conference*, 2008, pp. 1350–1361.

[11] M. K. Sain and J. L. Massey, "Invertibility of linear time-invariant dynamical systems," *IEEE Transactions on Automatic Control*, vol. AC-14, no. 2, pp. 141–149, Apr. 1969.

[12] 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.

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

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

[15] A. S. Willsky, "On the invertibility of linear systems," *IEEE Transactions on Automatic Control*, vol. 19, no. 2, pp. 272–274, June 1974.

[16] 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.

[17] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[18] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[19] E. Erez and M. Feder, "Efficient network codes for cyclic networks," in *Proceedings of the 2005 International Symposium on Information Theory*, 2005, pp. 1982–1986.

[20] L. M. Silverman, "Inversion of multivariable linear systems," *IEEE Transactions on Automatic Control*, vol. AC-14, no. 3, pp. 270–276, June 1969.

[21] F.-M. Yuan, "Minimal dimension inverses of linear sequential circuits," *IEEE Transactions on Automatic Control*, vol. AC-20, no. 1, pp. 42–52, Feb. 1975.

[22] E. Emre and L. Silverman, "Minimal dynamic inverses for linear systems with arbitrary initial states," *IEEE Transactions on Automatic Control*, vol. AC-21, no. 5, pp. 766–769, Oct. 1976.

[23] T. Yoshikawa and T. Sugie, "Inverse systems for reproducing linear functions of inputs," *Automatica*, vol. 17, no. 5, pp. 763–769, 1981.

[24] S. Sundaram and C. N. Hadjicostis, "Partial state observers for linear systems with unknown inputs," *Automatica*, vol. 44, no. 12, pp. 3126–3132, Dec. 2008.

[25] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proceedings of the 2005 Conference on Information Sciences and Systems*, 2005, pp. 113–122.