A MULTIDIMENSIONAL SYSTEMS APPROACH TO GRID SENSOR NETWORKS

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

M.G. Buddika Sumanasena

Peter H. Bauer, Director

Graduate Program in Electrical Engineering
Notre Dame, Indiana
April 2012

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

A MULTIDIMENSIONAL SYSTEMS APPROACH TO GRID SENSOR NETWORKS

Abstract

by

M.G. Buddika Sumanasena

A method for distributed information processing in rectangular grid based wireless sensor networks is presented, employing the Givone-Roesser and the Fornasini-Marchesini state space models for m-D systems. It can be used for distributed implementation of any general linear system on a grid sensor network. The method is highly scalable and requires only communication between immediate neighbors.

Usage of finite precision schemes for the representation of numbers and computations introduce nonlinearities to the otherwise linear m-D system models. Nonlinearities caused by fixed point and floating point number representation schemes used for in node computations and inter-node communication are modeled. Stability of the system is analyzed with special consideration given to the influence of inter-node communication on system dynamics. Necessary and sufficient conditions for the global asymptotic stability under both fixed point and floating point arithmetic is derived. It has been shown that the global asymptotic stability of the sensor networks is equivalent to that of a 1-D system for both the cases of fixed point and floating number representation.

Issues posed by communication time delay, in real-time implementation of the proposed method, are discussed. It is shown that, in order to implement a real-time sensor

network, system matrices of the state space models have to satisfy certain conditions. A necessary and sufficient condition for a transfer function to be realizable in the constrained state space models is established. Realization algorithms to derive state space models of the desired form given an admissible transfer function are also presented. Node and link failure introduce complications not encountered in centralized implementation of m-D systems. Givone-Roesser and the Fornasini-Marchesini state space models are extended to include node and link failure. Necessary and sufficient conditions for mean square stability are then derived with the help of these two state space models. Input output stability of the distributed systems under node and link failure is also discussed.

The utility of the proposed method is demonstrated by examples. In particular a distributed Kalman filter is proposed for grid sensor networks. Implementation of the proposed Kalman filter on grid sensor networks is discussed in some detail. A method for contaminant detection and its implementation using the proposed method is also presented.

CONTENTS

FIGURES

SYMBOLS

1-D	One dimensional
2-D	Two dimensional
3-D	Three dimensional
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^{m\times n}$	The set of real matrices of order $m \times n$
$\mathbb{Z}$	The set of integers
$\mathbb{Z}^{+}$	The set of non-negative integers
$[a, b]$	$\{x :\ x \in \mathbb{R}\ and\ a \leq x \leq b\}$
$[a, b)$	$\{x :\ x \in \mathbb{R}\ and\ a \leq x < b\}$
$A \otimes B$	Kronecker product of matrices A and B
$\rho(A)$	Spectral norm of matrix A
$\sigma(A)$	Spectral radius of matrix A
BIBO	Bounded input bounded output
BIBOMS	Bounded input bounded output stable in the mean square sense
FM model	Fornasini-Marchesini model
GR model	Givone-Roesser model
GAS	Globally asymptotically stable
GAS	Global asymptotic stability
I_n	Identity matrix of order n

LHS	Left hand side
m-D	Multidimensional
MSS	Mean square stable
MSS	Mean square stability
RF	Radio frequency
RHS	Right hand side

ACKNOWLEDGMENTS

The author would like to thank his advisor, Professor Peter Bauer for his continued support throughout this work. His patient guidance and encouragement was conducive for the smooth progress of this work.

The author is indebted to the committee members Professor Panos Antsaklis, Professor Ken Sauer and Professor Tracy L. Kijewski-Correa for time and effort spent amidst their busy schedules on short notice. The author owes his colleagues a debt of gratitude for their support granted in many an occasion.

This material is based upon research supported by the National Science Foundation (NSF) under grant IIS-0325252 (ITR Medium) at University of Notre Dame. Funding was also provided by Defense Threat Reduction Agency (DTRA) and Naval Surface Warfare Center (NSWC) - Crane division under grant N00164-07-C-8510. The author would like to take this opportunity to thank the funding agencies for their generosity.

CHAPTER 1

INTRODUCTION

Sensor networks, collections of spatially distributed sensor nodes cooperating to accomplish a common task, are promising to change the way we live our lives, Akyildiz et al. [2002]. Though the development of sensor networks was originally motivated by military applications, such as battlefield surveillance, they are used in numerous civilian applications including industrial process monitoring and control, agriculture, environmental and habitat monitoring, and traffic control.

A sensor network comprises of sensor nodes and a communication medium enabling inter node communication. A node consists of a processor, a sensor unit and a transceiver. The sensor unit may consist of different types of sensors such as seismic, magnetic, temperature, infrared radiation, acoustic and radar sensors, Chong and Kumar [2003]. Processors in typical sensor nodes have very limited computational capabilities and memory. The transceiver facilitates communication between nodes and possibly between the node and the outer world.

The most commonly used means of communication in sensor networks is wireless RF communication. Free space optical communication is proposed as an alternative in Kahn et al. [1999]. The main disadvantage of free space optical communication is the necessity of line-of-sight between the transmitter and the receiver. When a line-of-sight path is available, a well designed free space optical link requires significantly lower energy per bit than their RF counterparts Kahn et al. [1998].

Applications of sensor networks are many and varied, but typically involve some kind of monitoring, tracking, and controlling. They can be classified broadly into the following categories.

Military applications An early example of military applications of sensor networks is the Sound Surveillance System (SOSUS). It is an array of acoustic sensors (hydrophones) deployed on the ocean bottom at strategic locations to detect and track Soviet submarines. Over the years, other more sophisticated acoustic sensor networks have been developed for submarine surveillance. Sensor networks can be rapidly deployed, are fault tolerant and require little or no maintenance once deployed. Destruction of some nodes by hostile action does not affect the operation of the network as much as the destruction of a traditional sensor. These properties have rendered sensor networks promising candidates for military command, communication, intelligence, surveillance, and targeting systems.

Environmental applications Sensor networks have the potential to make a profound impact on the monitoring of natural environments, Hart and Martinez [2006]. Sensor networks have also been successfully employed for habitat monitoring, Mainwaring et al. [September 2002]. Other potential application areas include forest fire detection, flood detection and metrological and geological research.

Industrial applications Industrial process control is a potential application area of sensor networks. Information gathered by sensors can be used to reduce cost and improve performance and maintainability of machines and assembly lines.

Traffic control Sensor networks have been used for traffic control for quite a while. However the cost of sensors and the communication network that connects them have limited the deployment of traffic monitoring systems to few critical points.

Cheap sensors with embedded networking capabilities can change the landscape of traffic monitoring and control completely. A network of widely deployed sensors can estimate the flow of traffic and make real-time control decisions. Another approach is to equip vehicles with sensors Estrin et al. [1999]. Information on locations of traffic jams, the speed and density of traffic is exchanged between vehicles. This information can be used by both the drivers and traffic control systems.

Health applications Potential health applications of sensor networks include drug administration in hospitals, telemonitoring of human physiological data, diagnostics and tracking and monitoring patients inside a hospital, Akyildiz et al. [2002]. Sensor networks can allow a single system to be used for all the above mentioned applications.

Domestic applications A networked home where domestic appliances are networked and can interact with each other and external networks is envisioned in Petriu et al. [2000]. Remote monitoring and control of domestic devices would be made possible by such a network. Domestic energy saving and assisting impaired or elderly people are potential applications of such networks.

Despite the potential of sensor networks in diverse application areas, a number of unique technical challenges posed by them have impeded their deployment in envisioned applications.

Energy and power limitations Since the sensor nodes upon deployment are often inaccessible, the lifetime of a sensor network depends on the energy resources on the nodes. Power available for a sensor node is very limited due to size and cost constraints of a node.

Deployment Some applications demand sensor networks be deployed randomly in regions with little or no infrastructure, an example of which is military applications. Therefore sensor network protocols and algorithms must possess self organizing capabilities.

Cost The number of nodes in a sensor network may be very large. To justify the cost of the overall network, the cost of a sensor node has to be low. On the other hand the restrictions on the cost of a node have implications on the computational, power, energy and communication resources on board.

Reliability and fault tolerance Sensor networks may be deployed in hostile environments and left unattended. Lack of power, physical damage, or interference can cause some nodes to malfunction. The sensor network should therefore be able to sustain its functionalities under node failures and other faults.

1.1 Motivation

1.1.1 Grid Sensor Networks

Sensor networks can be broadly classified into three categories based on the sensor deployment strategy.

- Random deployment
- Regular Deployment
- Application specific deterministic deployment

A sensor network where sensors are placed in a regular pattern is called a regular sensor network. Such deployment of sensors may not be feasible in many applications. However, random placement of sensor nodes can be very expensive due to redundancy

required to overcome the uncertainty. Regular sensor networks have drawn considerable attention in the literature.

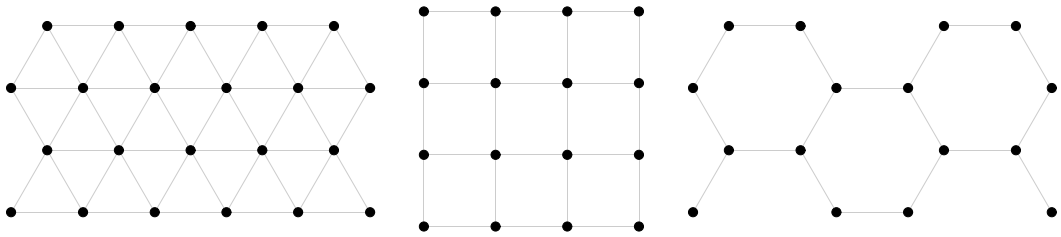


Figure 1.1. Equilateral triangular(a), square(b) and hexagonal(c) sensor deployment patterns

Three sensor placement patterns for regular sensor networks are discussed in Biagioni and Sasaki [2003]. The three patterns are derived from the three regular tessellations of the 2-D plane. For deployment of sensors in a triangular lattice, the region to be covered is tessellated by equilateral triangles, and sensor nodes are placed at the vertices of the triangles. The same procedure is followed for deployment of sensors in square and hexagonal patterns except the region to be covered is tiled by squares and hexagons respectively. Figure 1.1 illustrates the equilateral triangular, square and hexagonal sensor deployment patterns. Regular sensor networks with sensors deployed in a rectangular lattice are referred to as grid sensor networks.

1.1.2 Linear Algorithms on Sensor Networks

Vast and diverse applications of sensor networks have prompted a wide variety of signal processing algorithms to be implemented on them. Of particular relevance to the current work are algorithms that perform linear operations on the raw sensor data or functions of them. A few representative examples from the literature are discussed below.

1.1.2.1 Distributed Kalman Filters

Centralized implementation of Kalman filters on sensor networks has stringent communication, and hence energy, requirements. A strategy for distributed Kalman filtering was presented in [Cattivelli et al., 2008; Sumanasena and Bauer, 2010e]. Every node runs a local Kalman filter. Estimates from local Kalman filters of neighboring nodes are linearly combined to achieve asymptotically the performance of a global Kalman filter.

1.1.2.2 Consensus Filters

Distributed algorithms for computing the average of sensor data over a sensor network are proposed in [Olfati-saber and Shamma, 2005; Scherber and Papadopoulos, 2004, 2005]. The average is approximated at every node by iteratively performing local low pass filtering on data collected from neighboring nodes. Consensus filters have been used for distributed Kalman filtering [Olfati-Saber, 2005, 2007; Spanos et al., 2005]. In Pham et al. [2004] consensus filters are used for the distributed realization of sound source localization algorithms given in Pham et al. [2003].

1.1.2.3 Distributed Optimization

A sub-gradient descent algorithm for optimizing a convex function of sensor node measurements in a sensor network is given in Rabbat and Nowak [2004] . The update step in the proposed approach linearly combines data from all the sensors. Distributed optimization is possible by employing a distributed update step. Application of the algorithm for robust estimation, energy-based source localization and clustering density estimation is also discussed.

1.1.2.4 Filtering

Linear filtering based approaches for edge detection, noise reduction and contaminant front detection in sensor networks are proposed in Chintalapudi and Govindan [2003]; Devaguptapu and Krishnamachari [2003]; Sumanasena and Bauer [2008]. Methods discussed employ 2-D filters from image processing literature.

1.1.2.5 Multi Resolution Processing

When the sensor measurements in a sensor network exhibit redundancies, it is more efficient from a computational and communication perspective to process a sparser version of sensor measurements. Multi resolution signal processing techniques can be employed to make the data from the sensor network sparse. A wavelet-based approach for storage and search in sensor networks is proposed in Ganesan et al. [2005].

1.1.3 Local State Space Models for 3-D Systems

Multidimensional system theory is the theory concerning systems that evolve over multiple independent dimensions. Its applications include signal processing in radar, seismology, sonar, and image and video processing in general. The Givone-Roesser

model Givone and Roesser [1972] and the Fornasini-Marchesini model
Fornasini and Marchesini [1978] are widely used local state space models for m-D systems. The Givone-Roesser(GR) and the Fornasini-Marchesini(FM) models were originally proposed for 2-D systems but can be extended readily to higher dimensional systems.

Of particular importance to the current work are 3-D systems with two spatial dimensions and one temporal dimension. GR and FM models for such systems are discussed next. Details of dimensions of vectors and matrices are omitted since they are not necessary for the discussion at hand and will be treated when these models will be revisited in the next chapter.

1.1.3.1 Givone-Roesser Model

The GR model for 3-D systems is given by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = \boldsymbol{C}\boldsymbol{x}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t) \tag{1.1}$$

where $\boldsymbol{x}(n_1,n_2,t) = (\boldsymbol{x}^{h^T}(n_1,n_2,t), \boldsymbol{x}^{v^T}(n_1,n_2,t), \boldsymbol{x}^{t^T}(n_1,n_2,t))^T$ is the state vector, $\boldsymbol{u}(n_1,n_2,t)$ is the input vector and $\boldsymbol{y}(n_1,n_2,t)$ is the output vector. Here, $n_1 \in \mathbb{Z}$, $n_2 \in \mathbb{Z}$ and $t \in \mathbb{Z}$. Vectors $\boldsymbol{x}^h$, $\boldsymbol{x}^v$ and $\boldsymbol{x}^t$ are called the horizontal, vertical and temporal state vector components respectively.

1.1.3.2 Fornasini-Marchesini Model

There are two variations of the Fornasini-Marchesini model called the FM I and FM II models. The FM II model for 3-D systems is given by:

$$\begin{aligned}\boldsymbol{x}(n_1,n_2,t) &= \boldsymbol{A}_t\boldsymbol{x}(n_1,n_2,t\text{–}1) + \boldsymbol{A}_v\boldsymbol{x}(n_1,n_2\text{–}1,t) + \boldsymbol{A}_h\boldsymbol{x}(n_1\text{–}1,n_2,t)\\ &\quad + \boldsymbol{B}_t\boldsymbol{u}(n_1,n_2,t\text{–}1) + \boldsymbol{B}_v\boldsymbol{u}(n_1,n_2\text{–}1,t) + \boldsymbol{B_h}\boldsymbol{u}(n_1\text{–}1,n_2,t)\\ \boldsymbol{y}(n_1,n_2,t) &= \boldsymbol{C}\boldsymbol{x}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t). \end{aligned} \tag{1.2}$$

where $\boldsymbol{x}(n_1,n_2,t)$ is the state vector, $\boldsymbol{u}(n_1,n_2,t)$ is the input vector and $\boldsymbol{y}(n_1,n_2,t)$ is the output vector. Here, $n_1 \in \mathbb{Z}$, $n_2 \in \mathbb{Z}$ and $t \in \mathbb{Z}$. The FM I model for 3-D systems is given by:

$$\begin{aligned}\boldsymbol{x}(n_1,n_2,t) &= \boldsymbol{A}_t\boldsymbol{x}(n_1,n_2,t\text{–}1) + \boldsymbol{A}_v\boldsymbol{x}(n_1,n_2\text{–}1,t) + \boldsymbol{A}_h\boldsymbol{x}(n_1\text{–}1,n_2,t)\\ &\quad + \boldsymbol{A}_{tv}\boldsymbol{x}(n_1,n_2\text{–}1,t\text{–}1) + \boldsymbol{A}_{vh}\boldsymbol{x}(n_1\text{–}1,n_2\text{–}1,t) + \boldsymbol{A}_{ht}\boldsymbol{x}(n_1\text{–}1,n_2,t\text{–}1)\\ &\quad + \boldsymbol{A}_{tvh}\boldsymbol{x}(n_1\text{–}1,n_2\text{–}1,t\text{–}1) + \boldsymbol{B}_t\boldsymbol{u}(n_1,n_2,t\text{–}1) + \boldsymbol{B}_v\boldsymbol{u}(n_1,n_2\text{–}1,t)\\ &\quad + \boldsymbol{B}_h\boldsymbol{u}(n_1\text{–}1,n_2,t) + \boldsymbol{B}_{tv}\boldsymbol{u}(n_1,n_2\text{–}1,t\text{–}1) + \boldsymbol{B}_{vh}\boldsymbol{u}(n_1\text{–}1,n_2\text{–}1,t)\\ &\quad + \boldsymbol{B}_{ht}\boldsymbol{u}(n_1\text{–}1,n_2,t\text{–}1) + \boldsymbol{B}_{tvh}\boldsymbol{u}(n_1\text{–}1,n_2\text{–}1,t\text{–}1)\\ \boldsymbol{y}(n_1,n_2,t) &= \boldsymbol{C}\boldsymbol{x}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t). \end{aligned} \tag{1.3}$$

where $\boldsymbol{x}(n_1,n_2,t)$ is the state vector, $\boldsymbol{u}(n_1,n_2,t)$ is the input vector and $\boldsymbol{y}(n_1,n_2,t)$ is the output vector. A 3-D system representable in the FM-II model is also representable in the FM-I model. The converse is also true. Using the FM-II model makes the notation more concise since it has fewer matrices and vectors than the FM-I model. Therefore FM-II model is used throughout this work to represent 3-D systems.

1.1.4 The Proposed Approach

Consider a grid sensor network of size $N_1 \times N_2$. A linear process operating on sensor measurements of multiple nodes and over multiple sampling instances is a 3-D process. Let there be a spatially and temporally causal 3-D linear process to be implemented on the sensor network. Such a process is first octant causal and can be represented by models (1.1) and (1.2). State space models can be implemented on a sensor network as follows.

- In models (1.1) and (1.2), let (n_1, n_2) be the coordinates of sensor nodes in the grid and t denote the sampling time index.
- Every node computes its state vector using equations (1.1) or (1.2) depending upon which models is used, using the information received from preceding nodes[1] and the state vector of itself at the previous sampling time instance.
- Every node computes the output vector using the state vector and the input vector.
- Every node transmits the information required by the succeeding nodes to compute their state vectors.

The method described above is a distributed computational procedure and requires communication only between the neighboring nodes in a grid sensor network. Hence it is highly scalable. The sequence of operations has to be started at a node, preferably the node at the origin, and propagated in the positive directions of the spatial axes. Therefore the process realized is first octant causal. By selecting the origin and the direction of propagation appropriately, spatially second, third and fourth quadrant causal

[1]Node (n_1^1, n_2^1) is said succeed node (n_1^2, n_2^2) if $n_1^1 \geq n_1^2$ and $n_2^1 \geq n_2^2$. Node (n_1^1, n_2^1) is said to precede node (n_1^2, n_2^2) if the later succeeds the former

processes can be implemented. A spatially non-causal process can be realized as a combination of processes causal in each of the four quadrants.

If the system implemented is spatially invariant, every node runs identical code and over-the-air programming methods can be used to reconfigure the sensor network. Output at each node is computed by the node itself. Therefore the approach supports local actuation in response to local activity. The method assumes some kind of ordering of sensors in space. This results from the formulation of state space models (1.1) and (1.2). Hence there is no straightforward extension to irregular sensor networks, but the approach can be adapted to grids that are not rectangular.

1.2 Literature Survey

Existing literature on sensor networks and multidimensional systems is vast and diverse. In this section contributions pertinent for the discussion at hand are discussed.

General introductions to evolution, applications, routing algorithms, signal processing aspects and challenges of sensor networks are given in [Akyildiz et al., 2002; Chong and Kumar, 2003; Estrin et al., 2001; Romer and Mattern, 2004].

A sensor deployment strategy to deploy sensors in a grid when manual deployment is impossible is given in Leoncini et al. [2005]. Sensor drop strategies given in Leoncini et al. [2005] aim to meet a given degree of coverage with a minimum number of nodes under placement uncertainties. An algorithm to achieve connectivity and coverage for multiple target points in a grid sensor network is proposed in Wu et al. [2008]. The procedure given provides close to optimal results. In Xu et al. [2006] robustness of regular sensor networks against sensor placement errors is discussed. The work concentrates on the effect of both random and non-random deployment errors on the coverage in triangular lattice sensor networks. Triangular, square and hexagonal

lattice sensor networks are compared for the number of nodes required to cover a given area and robustness against node failure in Biagioni and Sasaki [2003]. Coverage and connectivity of grid sensor networks in the presence of node failure has been studied in Shakkottai et al. [2003]. A sufficient condition for the connectivity of the active nodes is also derived.

For the case where there are two types of sensor nodes with different coverage and cost, an optimal scheme for placing sensors on a grid, which minimizes the cost and assures coverage in the presence of node failure, is given in Chakrabarty et al. [2002].

A reliability measure for sensor networks, which considers a group of sensors to be operational if there exists an operational bidirectional path from the sink node to at least one operational sensor in the group, is formulated in AboElFotoh et al. [2005]. A reliability measure based on the aggregate flow of information from sensor nodes to the sink node is proposed in AboElFotoh et al. [2007]. It is also proven that evaluating this measure for an arbitrary sensor network is nP-hard. It has been shown that evaluating this reliability measure for grid sensor networks is also nP-hard see, AboElFotoh and Elmallah [2008]. An algorithm to evaluate the said reliability measure for a grid sensor network with uniformly generated traffic upon a particular routing algorithm is proposed in AboElFotoh and Elmallah [2008].

Network capacity limits and optimal routing algorithms for grid sensor networks for cases with no node failure and random node failure are studied in Barrenechea et al. [2004]. For sensor networks with no node failures, an upper bound for network capacity was derived, and a routing algorithm which achieves the upper bound is presented. A combination of two routing algorithms, the first being the optimal routing algorithm for the no node failure case and the second an algorithm suitable if the probability of node failure is high, is proposed for sensor networks with random node failure. Proto-

cols and algorithms that allow routing load to be distributed fairly in a sensor network are proposed in Akbar et al. [2006] for grid sensor networks. A node or link failure recovery scheme is also presented.

A power consumption model for sensor nodes in a sensor network was proposed in Wang et al. [2006]. In Wang and Yang [2007] an energy consumption model is proposed for sensor networks. Energy aware routing protocols have been proposed in [Muruganathan et al., 2005; Stojmenovic and Lin, 2001] and references therein.

Several local state space models for multidimensional systems have been proposed [Attasi, 1973; Fornasini and Marchesini, 1978; Givone and Roesser, 1972]. They were originally proposed for 2-D systems. A comparison of the three state space models presented in above work is given in Kung et al. [1977]. A procedure to derive the GR model from the FM model of a 2-D system is also presented and it is argued that the GR model is the most general model of the three. The local state space models in [Attasi, 1973; Fornasini and Marchesini, 1978; Givone and Roesser, 1972] are shown to be special cases of another local state space model in Eising [1978].

Contrary to the 1-D case, where minimal state space realizations can be derived for a given causal transfer function, minimal realizations can be derived only for special categories of m-D systems such as, continued fraction expandable systems, all-pole, and all-zero filters, product factorable transfer functions, discrete time lossless bounded real functions, separable and factorable systems, and first order all-pass filters Antoniou [2001]. A realization procedure to realize a causal transfer function in the GR model is presented in Eising [1978]. Realization procedure presented in Eising [1978] is generalized to a larger class of 2-D systems in Eising [1980]. It is shown that in general, causal 2-D systems do not have a causal inverse. Algorithms to realize a given 2-D transfer function in the GR model are given in [Antoniou, 2001; Mitra et al., 1975].

A 3-D realization algorithm for the GR model is given in Fan et al. [2006]. Realization of 2-D transfer functions in the FM model is addressed in [Bisiacco et al., 1989; Fornasini and Marchesini, 1978; Xu et al., 2008].

Stability of multidimensional systems has drawn much attention and a rich literature exists on the subject. It has been argued that direct extension of 1-D BIBO stability to multidimensional systems is overly restrictive Agathoklis and Bruton [1983]. A new external stability criterion, practical-BIBO stability which is less restrictive and more relevant for practical applications than the conventional one, is also introduced.

A set of 1-D conditions necessary for GAS of non-linear multidimensional systems is given in Bauer [1995b]. Conditions under which GAS of a system with quantization and overflow nonlinearities is equivalent to GAS of that with only the quantization nonlinearity was established in Leclerc and Bauer [1994]. A sufficient condition for GAS of a 2-D system realized in the FM model is established in Bose [1995]. Stability of 2-D systems realized in FM and GR models using two's complement arithmetic is studied in Bose [1995] and Bose [1994], respectively. Further results on GAS of 2-D systems realized in GR and FM models employing finite precision arithmetic are presented in [Kar and Singh, 2001a; Singh, 2008] and references therein.

1.3 The Big Picture

Though m-D systems theory and local state space models for m-D systems were known for a long time and linear systems were implemented on sensor networks in a variety of applications, the first efforts to realize m-D systems on grid sensor networks using the local state space models appeared in Dewasurendra and Bauer [2008] and Sumanasena and Bauer [2008, 2009, 2010a] to the best of the author's knowledge.

The approach proposed enables distributed implementations of linear systems on

grid sensor networks. Tools and techniques already developed in m-D systems theory can be used for implementation and analysis of properties such as stability of such systems. The m-D systems theory was developed in a centralized context and the migration to a distributed and more uncertain setting poses striking theoretical challenges and opportunities to explore.

Communication time delay mandates system matrices to have certain properties for them to be implementable in real-time. This makes it impossible to realize any given transfer function. Different number representation and quantization schemes may be used for in-node computations and communication among nodes. The effect of possibly different overflow and quantization nonlinearities on system dynamics also needs to be analyzed. On the other hand, the finite spatial extent of the sensor network enables stronger conditions for stability to be established, see Sumanasena and Bauer [2011e, 2009, 2010a].

Node and link failures in a sensor network introduce uncertainties to otherwise deterministic m-D system models [Sumanasena and Bauer, 2011c,d]. These uncertainties have to be modeled and their effect on the overall system performance has to be assessed. In a grid sensor network it may not be feasible to place sensors exactly on the grid. Non uniform sampling resulting from the irregular sensor placement can have adverse effects on the performance of the system.

The aim of the current work is to address these issues that naturally arise in distributed implementations of m-D systems in grid sensor networks.

1.4 Structure of the Thesis

The GR and FM local state space models for 3-D systems and their application to information processing in grid sensor networks are discussed in chapter 2. Issues

posed by time lag in data communication, in real-time implementation of the proposed method, are also discussed. Extension of the proposed method to a ring topology is also discussed in chapter 2.

In chapter 3 realizability of a given 3-D transfer function in real-time is studied. A necessary and sufficient condition, for a transfer function to be realizable as GR and FM models in real-time, is established. Realization algorithms that realize an admissible transfer matrix in GR and FM models of the desired form are also presented. Stability of the 3-D distributed systems in the face of quantization and overflow nonlinearities is treated in chapter 4. Nonlinearities resulting from quantization and overflow are modeled for both fixed point and floating point implementations. Necessary and sufficient conditions for GAS of GR and FM models are also established. Sufficient conditions for BIBO stability are derived for systems realized in both local state space models when fixed point implementations.

The effect of node and link failure on system dynamics is studied in chapter 5. GR and FM models for 3-D systems are extended to incorporate node and link failure. Internal and external stability of the distributed 3-D systems under node and link failure is also analyzed. Two examples are presented to demonstrate the utility of the proposed approach for signal processing in sensor networks in chapter 6. Open research issues are discussed and concluding remarks are given in chapter 7.

CHAPTER 2

MODELS FOR GRID SENSOR NETWORKS

GR and FM models are m-D counterparts of 1-D state space models for discrete time systems. They model the evolution of an m-D system driven by an input, over multiple independent dimensions. Similar to 1-D state space models where the current state is evaluated using the previous state and input, states are evaluated using immediately preceding states and inputs. The local nature of computation renders GR and FM state space models promising candidates for distributed signal processing in grid sensor networks.

2.1 GR Model Based Implementation

2.1.1 GR Model for 3-D Systems

The GR model for 3-D systems is given by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = \boldsymbol{C}\boldsymbol{x}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t) \tag{2.1}$$

where $\boldsymbol{x}(n_1, n_2, t) = (\boldsymbol{x}^{h^T}(n_1, n_2, t), \boldsymbol{x}^{v^T}(n_1, n_2, t), \boldsymbol{x}^{t^T}(n_1, n_2, t))^T$, $n_1 \in \mathbb{Z}$, $n_2 \in \mathbb{Z}$ and $t \in \mathbb{Z}$. Vectors $\boldsymbol{x}^h \in \mathbb{R}^a$, $\boldsymbol{x}^v \in \mathbb{R}^b$ and $\boldsymbol{x}^t \in \mathbb{R}^c$ are called the horizontal, vertical and temporal state vector components respectively. Let the input vector $\boldsymbol{u} \in \mathbb{R}^p$ and the output vector $\boldsymbol{y} \in \mathbb{R}^q$. Then $\boldsymbol{A}_1 \in \mathbb{R}^{a\times a}$, $\boldsymbol{A}_2 \in \mathbb{R}^{a\times b}$, $\boldsymbol{A}_3 \in \mathbb{R}^{a\times c}$, $\boldsymbol{A}_4 \in \mathbb{R}^{b\times a}$, $\boldsymbol{A}_5 \in \mathbb{R}^{b\times b}$, $\boldsymbol{A}_6 \in \mathbb{R}^{b\times c}$, $\boldsymbol{A}_7 \in \mathbb{R}^{c\times a}$, $\boldsymbol{A}_8 \in \mathbb{R}^{c\times b}$, $\boldsymbol{A}_9 \in \mathbb{R}^{c\times c}$, $\boldsymbol{B}_1 \in \mathbb{R}^{a\times p}$, $\boldsymbol{B}_2 \in \mathbb{R}^{b\times p}$, $\boldsymbol{B}_3 \in \mathbb{R}^{c\times p}$, $\boldsymbol{C} \in \mathbb{R}^{q\times(a+b+c)}$ and $\boldsymbol{D} \in \mathbb{R}^{q\times p}$. Let:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix}$$

and $\boldsymbol{B} = (\boldsymbol{B}_1^T, \boldsymbol{B}_2^T, \boldsymbol{B}_3^T)^T$. A first octant causal linear process can be represented using this model. Given a 2-D causal input-output transfer function, state space representation of the system in the GR model can be derived using methods given in [Eising, 1978; Kung et al., 1977; Mitra et al., 1975]. Algorithms to realize a given causal 3-D transfer function in the GR model are given in Fan et al. [2006]; Kanellakis et al. [1989]; Manikopoulos and Antoniou [1990]; Sumanasena and Bauer [2011d, 2010c]; Theodorou and Tzafestas [1984]. The 3-D realization algorithm proposed in Fan et al. [2006] was generalized for m-D realization in Xu et al. [2008].

2.1.2 Implementation in a Sensor Network

In model (2.1) let spatial variables n_1 and n_2 be the horizontal and vertical coordinates of a node respectively. The tuple (n_1, n_2) thus refers to a unique node in the sensor network. Vectors $\boldsymbol{x}(n_1, n_2, t)$, $\boldsymbol{y}(n_1, n_2, t)$ and $\boldsymbol{u}(n_1, n_2, t)$ are the state vector, the output vector and the input vector of node (n_1, n_2) at time slot t respectively. The following operations are performed by each node (n_1, n_2) at the time slot t.

- Receive state vector components $\boldsymbol{x}^h(n_1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2, t)$ from nodes $(n_1 - 1, n_2)$ and $(n_1, n_2 - 1)$ respectively.
- Use equation (2.1) to compute $\boldsymbol{x}^h(n_1+1, n_2, t)$, $\boldsymbol{x}^v(n_1, n_2+1, t)$ and $\boldsymbol{x}^t(n_1, n_2, t+1)$
- Transmit $\boldsymbol{x}^h(n_1 + 1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2 + 1, t)$
- Use equation (2.1) to compute the output.

For a sensor network of size $N_1 \times N_2$, $0 \leq n_1 \leq N_1 - 1$ and $0 \leq n_2 \leq N_2 - 1$. For convenience let $t \in \mathbb{Z}^+$. Figure 2.1 illustrates the operation of nodes and communication of state vectors between nodes.

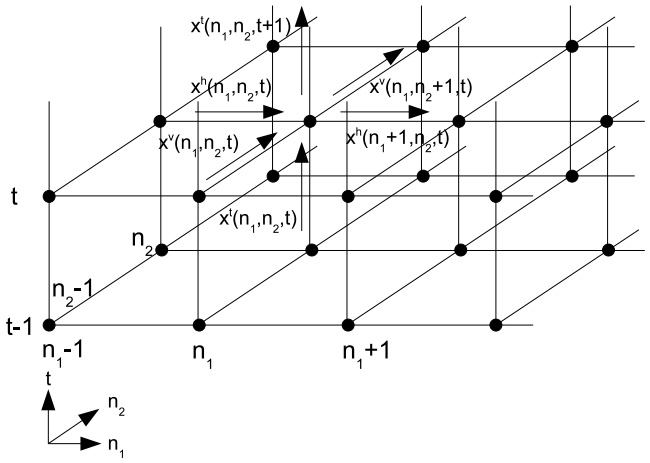


Figure 2.1. Communication of state vectors between nodes in the network for the GR model based implementation

2.2 FM Model Based Implementation

There are two variations of the FM model called the FM I and FM II models. In the current work FM II local state space models will be used and would be referred to as FM model.

2.2.1 FM Model for 3-D Systems

The FM model for 3-D systems is given by:

$$\begin{aligned}\boldsymbol{x}(n_1, n_2, t) &= \boldsymbol{A}_t\boldsymbol{x}(n_1, n_2, t–1) + \boldsymbol{A}_v\boldsymbol{x}(n_1, n_2–1, t) + \boldsymbol{A}_h\boldsymbol{x}(n_1–1, n_2, t) \\ &\quad + \boldsymbol{B}_t\boldsymbol{u}(n_1, n_2, t–1) + \boldsymbol{B}_v\boldsymbol{u}(n_1, n_2–1, t) + \boldsymbol{B}_h\boldsymbol{u}(n_1–1, n_2, t) \\ \boldsymbol{y}(n_1, n_2, t) &= \boldsymbol{C}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t) \end{aligned} \tag{2.2}$$

where $n_1 \in \mathbb{Z}$, $n_2 \in \mathbb{Z}$ and $t \in \mathbb{Z}$. Let the input vector $\boldsymbol{u} \in \mathbb{R}^p$ and the output vector $\boldsymbol{y} \in \mathbb{R}^q$. If the state vector $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{C} \in \mathbb{R}^{q\times n}$, $\boldsymbol{D} \in \mathbb{R}^{q\times p}$, $\boldsymbol{A}_t \in \mathbb{R}^{n\times n}$, $\boldsymbol{A}_v \in \mathbb{R}^{n\times n}$, $\boldsymbol{A}_h \in \mathbb{R}^{n\times n}$, $\boldsymbol{B}_h \in \mathbb{R}^{n\times p}$, $\boldsymbol{B}_v \in \mathbb{R}^{n\times p}$ and $\boldsymbol{B}_t \in \mathbb{R}^{n\times p}$.

Realization of 2-D causal input-output transfer functions in the FM state space model is discussed in [Bisiacco et al., 1989; Fornasini and Marchesini, 1978; Xu et al., 2005, 2007]. Algorithms to realize a given causal 3-D transfer function in the FM model are discussed in [Sumanasena and Bauer, 2011b, 2010b].

2.2.2 Implementation in a Sensor Network

The FM model can be implemented in a sensor network employing a similar approach to the implementation of the GR model. The following operations are performed by each node (n_1, n_2) at the time slot t.

- Receive state vectors $\boldsymbol{x}(n_1–1, n_2, t)$ and $\boldsymbol{x}(n_1, n_2–1, t)$ and input vectors $\boldsymbol{u}(n_1–$

$1, n_2, t)$ and $\boldsymbol{u}(n_1, n_2 - 1, t)$.

- Use equation (2.2) to compute $\boldsymbol{x}(n_1, n_2, t)$.
- Transmit $\boldsymbol{x}(n_1, n_2, t)$ and $\boldsymbol{u}(n_1, n_2, t)$
- Use equation (2.2) to compute the output.

For a sensor network of size $N_1 \times N_2$, $0 \leq n_1 \leq N_1 - 1$ and $0 \leq n_2 \leq N_2 - 1$. Let $t \in \mathbb{Z}^+$ for convenience. Figure 2.2 illustrates the operation of nodes and communication of state vectors among the nodes.

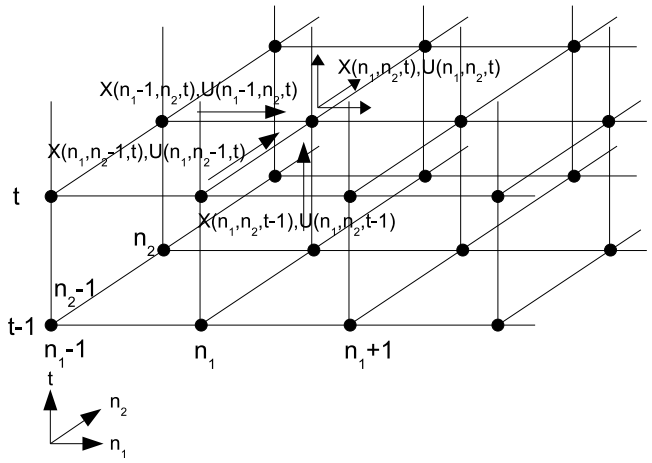


Figure 2.2. Communication of state vectors between nodes in the network for the FM model based implementation

2.3 Realization of Non-causal Systems

Systems realizable in models (2.1) and (2.2) are necessarily first octant causal. A non causal impulse response in the spatial plane can be decomposed into four quarter plane causal components. In [Lele and Mendel, 1987; Ntogramatzidis et al., 2007] a non causal 2-D impulse response is decomposed into four quarter plane causal impulse responses. Each component is realized in a generalized FM model Kaczorek [1988] in Ntogramatzidis et al. [2007]. The four local state space realizations are then combined to a single local state space model. Each quarter plane causal impulse response is realized by a GR model in [Lele and Mendel, 1987].

In a sensor network implementation, each component of the impulse response can be realized using the above models by appropriately selecting the origin and the direction of propagation. Outputs of the four quarter plane causal systems are summed to obtain the final output. Alternatively if the transfer function to be realized can be represented as a concatenation of quarter plane causal transfer functions, a series combination of quarter plane causal systems can be used to obtain the final output. Without loss of generality it is assumed in the rest of this work that the process implemented is first octant causal.

2.4 Real-time Implementation Issues

In a GR model based implementation state vector components $\boldsymbol{x}^h(n_1+1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2+1, t)$ of nodes (n_1+1, n_2) and (n_1, n_2+1) are evaluated at node (n_1, n_2) at time slot t. These components are required at time slot t by the nodes (n_1+1, n_2) and (n_1, n_2+1) to perform their computations. In a FM model based implementation the state vector $\boldsymbol{x}(n_1, n_2, t)$ is evaluated at the node (n_1, n_2) at time slot t. It is required by nodes (n_1+1, n_2) and (n_1, n_2+1) at time slot t to perform their computations. A

real-time implementation using either model thus requires data transmission with zero time delay, which is impossible. There are two options to work around this problem.

One is to allow a time lag between nodes which means the system is not real-time. This could be a good solution for small sized sensor networks. The other option is to modify the system matrices such that zero time delay data transmission is not required in the spatial dimensions to perform computations. This would limit the type of systems that can be implemented. These two options are discussed next.

2.4.1 Delayed Response Implementation

The problem is that nodes $(n_1 + 1, n_2)$ and $(n_1, n_2 + 1)$ do not receive state vector components $\boldsymbol{x}^h(n_1 + 1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2 + 1, t)$ at time slot t to perform their computations at time slot t. In a FM model based implementation nodes $(n_1 + 1, n_2)$ and $(n_1, n_2 + 1)$ do not receive the state vector $\boldsymbol{x}(n_1, n_2, t)$ at time slot t. A simple solution is to allow those nodes to do the computations they should do at time slot t at time slot $t+1$ instead. This means for each distance unit in either spatial direction there is a time lag of one time slot (one distance unit is equivalent to the distance between two nodes). This time lag could be significant in a moderate sized sensor network.

The time lag can be reduced significantly if the computation front of nodes propagates more than one distance unit in a sampling interval $[t, t + 1]$. Let the computation front propagate d distance units along either spatial axis in one time slot. Let the equivalence class E_k of nodes be defined by:

$$E_k = \{(n_1, n_2) : \lfloor \frac{n_1 + n_2}{d} \rfloor = k - 1\}$$

where $\lfloor\ \rfloor$ is the floor function. For example, consider the case where $d = 3$.

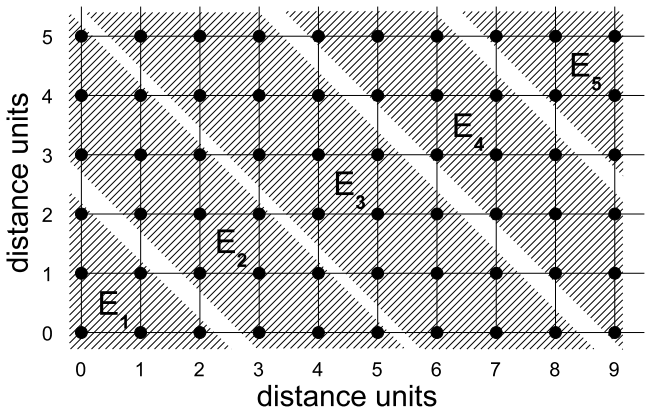


Figure 2.3. Equivalence classes of nodes

Equivalence classes of nodes for this case are shown in figure 2.3. Let computations start from node $(0, 0)$ at the first time slot. In the first time slot nodes in E_1 perform their computations and transmit the state vectors. At time slot 2 nodes in E_2 perform computations they should have done at time slot 1. At time slot 3 nodes in E_3 perform computations they should have done at time slot 1. In general, nodes in E_k do their first computation on the k-th time slot and the t-th computation on the $(k - 1 + t)$ th time slot. So the node (n_1, n_2) does its t-th computation on the $(\lfloor \frac{n_1+n_2}{d} \rfloor + t)$ th time slot. For a sensor network of size $N_1 \times N_2$ the maximum delay is $\lfloor \frac{N_1+N_2-2}{d} \rfloor$.

2.4.2 Real-Time Implementation

2.4.2.1 Using the GR Model

If the matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_4$ and $\boldsymbol{A}_5$ in (2.1) are zero, node (n_1, n_2) requires state vector components $\boldsymbol{x}^v(n_1, n_2, t)$ and $\boldsymbol{x}^h(n_1, n_2, t)$ only to compute $\boldsymbol{x}^t(n_1, n_2, t + 1)$. Since the state vector component $\boldsymbol{x}^t(n_1, n_2, t + 1)$ is used only by the node (n_1, n_2) the node can compute it at time slot $t+1$. Thus the node (n_1, n_2) does not require state vec-

tor components $\boldsymbol{x}^h(n_1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2, t)$ at time slot t to perform computations. So there is no need for data communication with zero time delay. The system (2.1) becomes:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2+1, t) \\ \boldsymbol{x}^t(n_1, n_2, t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_3 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2, t) \\ \boldsymbol{x}^t(n_1, n_2, t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1, n_2, t)$$

$$\boldsymbol{y}(n_1, n_2, t) = \boldsymbol{C}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t) \tag{2.3}$$

This modification to system matrix $\boldsymbol{A}$ makes a real-time implementation possible, but limits the impulse responses the system could have.

2.4.2.2 Using the FM Model

Let:

$$\begin{aligned} \boldsymbol{A}_v\boldsymbol{A}_v = \boldsymbol{A}_v\boldsymbol{A}_h = \boldsymbol{A}_h\boldsymbol{A}_v = \boldsymbol{A}_h\boldsymbol{A}_h = \boldsymbol{0} \\ \boldsymbol{A}_v\boldsymbol{B}_v = \boldsymbol{A}_v\boldsymbol{B}_h = \boldsymbol{A}_h\boldsymbol{B}_v = \boldsymbol{A}_h\boldsymbol{B}_h = \boldsymbol{0} \end{aligned} \tag{2.4}$$

in (2.2). Then node (n_1, n_2) can transmit $\boldsymbol{A}_t\boldsymbol{x}(n_1, n_2, t–1) + \boldsymbol{B}_t\boldsymbol{u}(n_1, n_2, t–1)$ at time slot t instead of $\boldsymbol{x}(n_1, n_2, t)$ which can't be computed with the information it has at time slot t. This doesn't affect computations at nodes (n_1+1, n_2) and (n_1, n_2+1) due to the condition (2.4). Node (n_1, n_2) can compute state vector $\boldsymbol{x}(n_1, n_2, t)$ and output vector $\boldsymbol{y}(n_1, n_2, t)$ at time slot $t+1$, when the necessary information is available. The constraint (2.4) on system matrices restricts the impulse responses the system could have.

2.5 Power and Energy Considerations

Implementation of the proposed scheme requires sensor nodes to transmit and receive data periodically. Receiver circuitry of current sensor nodes consumes a considerable amount of power, Wang et al. [2006], making it inefficient to keep it on all the time. Energy efficient implementation thus requires nodes to operate in synchronization with their neighbors. Synchronization can be achieved through synchronization schemes such as Li and Rus [2006]. This enables sensor nodes to operate on sleep-wake cycles resulting in significant energy savings.

State vectors of different nodes in a neighborhood may share common state vector elements albeit not in the same location of the state vector. This opens up the possibility of transmitting state vectors over multiple hops. It has been shown that transmission over multiple hops can reduce the energy consumption when 2-D systems are implemented in the FM model Sumanasena and Bauer [2008]. Realization algorithm proposed in Xu et al. [2005] was used to realize the 2-D transfer function. Since the repetition of state vector elements among nodes depend on the realization algorithm used, the merits of multi-hop data transmission depend on the realization algorithm used. Models derived in this chapter, for the distributed implementation of m-D systems, remain valid regardless of the number of hops over which information is transmitted.

2.6 Special Topologies

2.6.1 Infinite Grids

Though sensor networks are necessarily of finite size, the case in which the spatial extent of the system is infinite is of theoretical interest. Models derived in this chapter remain valid except that spatial variables can assume any integer value. In general,

conditions established for stability of finite grids are not applicable for infinite grids.

2.6.2 Cyclic Sensor Networks

In applications such as perimeter security sensors may be deployed in a ring encircling the target to be protected. A sensor network in which sensors are placed in a ring is called a cyclic sensor network. A system evolving on temporal and spatial dimensions in this setting can be considered a 2-D system. Spatial dimension of the system has a finite extent and is cyclic. Conventional definitions of linearity and spatial and temporal invariance can be extended to such systems.

Systems evolving on cyclic network dimensions have not been studied to the best of the author's knowledge. FM and GR models for 2-D systems with cyclic spatial dimension are proposed below. It is not understood yet, which types of transfer functions or impulse responses are realizable in the proposed models.

Let the sensor network has N nodes deployed on a ring. Arithmetic operator $\oplus$: $[0\ N-1] \times [0\ N-1] \rightarrow [0\ N-1]$ is defined by.

$$i \oplus j = \begin{cases} i+j & i+j \leq N-1 \\ i+j-N & else \end{cases}$$

2.6.2.1 GR Model

The GR model for the 2-D system is given by:

$$\begin{bmatrix} \boldsymbol{x}^s(n_1 \oplus 1, t) \\ \boldsymbol{x}^t(n_1, t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^s(n_1, t) \\ \boldsymbol{x}^t(n_1, t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} \boldsymbol{u}(n_1, t)$$

$$\boldsymbol{y}(n_1, t) = \boldsymbol{C}\boldsymbol{x}(n_1, t) + \boldsymbol{D}\boldsymbol{u}(n_1, t) \tag{2.5}$$

where $\boldsymbol{x}^s(n_1,t) \in \mathbb{R}^a$ and $\boldsymbol{x}^t(n_1,t) \in \mathbb{R}^c$ are the spatial and temporal state vector components respectively. Vectors $\boldsymbol{x}(n_1,t) = (\boldsymbol{x}^{s^T}(n_1,t), \boldsymbol{x}^{t^T}(n_1,t))^T$, $\boldsymbol{u}(n_1,t) \in \mathbb{R}^p$, and $\boldsymbol{y}(n_1,t) \in \mathbb{R}^q$ are the state, input and output vectors respectively. Matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$, $\boldsymbol{A}_4$, $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are of appropriate dimensions.

2.6.2.2 FM Model

The FM model for the 2-D system is given by:

$$\begin{aligned} \boldsymbol{x}(n_1 \oplus 1, t) &= \boldsymbol{A}_t \boldsymbol{x}(n_1 \oplus 1, t\text{–}1) + \boldsymbol{A}_h \boldsymbol{x}(n_1, t) + \boldsymbol{B}_t \boldsymbol{u}(n_1 \oplus 1, t\text{–}1) + \boldsymbol{B}_h \boldsymbol{u}(n_1, t) \\ \boldsymbol{y}(n_1, t) &= \boldsymbol{C}\boldsymbol{x}(n_1, t) + \boldsymbol{D}\boldsymbol{u}(n_1, t). \end{aligned} \tag{2.6}$$

where $\boldsymbol{x}(n_1,t) \in \mathbb{R}^n$, $\boldsymbol{u}(n_1,t) \in \mathbb{R}^p$, and $\boldsymbol{y}(n_1,t) \in \mathbb{R}^q$. Matrices $\boldsymbol{A}_t$, $\boldsymbol{A}_h$, $\boldsymbol{B}_t$, $\boldsymbol{B}_h$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are of appropriate dimensions.

CHAPTER 3

REALIZABILITY IN REAL-TIME

Real-time implementation demands system matrices of GR and FM models to satisfy conditions described in chapter 2. Not all causal rational transfer functions are realizable in the GR model (2.3) or the FM model (2.2) under the constraint (2.4). It is important to address the following issues.

- What are the transfer functions realizable in real-time using the GR and FM models?
- Given a realizable transfer function how to derive the GR or FM state space model of the desired form?

Since non-causal systems can be implemented as a combination of quarter plane causal systems, the solution to above issues for the case of causal systems can be generalized to non-causal systems.

3.1 Realizability in the GR Model

A necessary and sufficient condition, for a 3-D rational transfer matrix to be realizable in a GR model of the form (2.3), is established in this section. A realization algorithm to derive the GR model of the from (2.3), given an admissible transfer matrix, is presented as part of the proof.

3.1.1 Proper Transfer Matrices

Right matrix fraction description [Bose, 2003] is used to represent rational transfer matrices in this paper. Furthermore, let the Z-transform of a signal $s(n_1, n_2, t)$ denoted by $\mathbf{Z}[s(n_1, n_2, t)]$ be defined by:

$$\mathbf{Z}[s(n_1, n_2, t)] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} s(i, j, k) z_1^{-i} z_2^{-j} z_t^{-k}$$

Theorem 3.1.1 *Let an input-output transfer matrix be given by:*

$$\boldsymbol{H}(z_1, z_2, z_t) = \boldsymbol{N}_R(z_1, z_2, z_t) \boldsymbol{D}_R^{-1}(z_1, z_2, z_t) \tag{3.1}$$

where matrices $\boldsymbol{N}_R(z_1, z_2, z_t)$ and $\boldsymbol{D}_R(z_1, z_2, z_t)$ are of size $q \times p$ and $p \times p$ respectively. Let $\boldsymbol{N}_R(l, m)$ and $\boldsymbol{D}_R(l, m)$ denote $(l, m)th$ elements of matrices $\boldsymbol{N}_R(z_1, z_2, z_t)$ and $\boldsymbol{D}_R(z_1, z_2, z_t)$ respectively[1]*. Let:*

$$\boldsymbol{N}_R(l, m) = \sum_{i=0}^{N_{R1}} \sum_{j=0}^{N_{R2}} \sum_{k=0}^{N_{Rt}} n_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}$$

$$\boldsymbol{D}_R(l, m) = \sum_{i=0}^{D_{R1}} \sum_{i=0}^{D_{R2}} \sum_{i=0}^{D_{Rt}} d_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}$$

where N_{R1}, N_{R2} and N_{Rt} are the degrees of the polynomial matrix $\boldsymbol{N}_R(z_1, z_2, z_t)$ with respect to variables z_1, z_2 and z_t respectively. Furthermore D_{R1}, D_{R2} and D_{Rt} are the degrees of the polynomial matrix $\boldsymbol{D}_R(z_1, z_2, z_t)$ with respect to variables z_1, z_2 and z_t respectively. The transfer matrix (3.1) can be realized in a GR model of the form (2.3)

[1] In order to make the notation concise the (l, m)-th element of a polynomial matrix $\boldsymbol{P}(z_1, z_2, z_t)$ is denoted by $\boldsymbol{P}(l, m)$ in this work.

if and only if the conditions (3.2) and (3.3) are satisfied.

$$n_{ijk}^{lm} = 0 \;\; for \;\; i+j > k+1 \tag{3.2}$$

$$d_{ijk}^{lm} = 0 \;\; for \;\; i+j \geq k+1 \tag{3.3}$$

Proof Necessity is proved by showing that, if a transfer matrix is realizable in the GR model (2.3), it can be represented in the form (3.1) with conditions (3.2) and (3.3) satisfied. In order to prove sufficiency, a realization algorithm, which realizes the transfer matrix (3.1) in a GR model of the form (2.3), provided conditions (3.2) and (3.3) are satisfied, is presented[2]. The proposed algorithm is a modification of the algorithm given in Fan et al. [2006].

To prove the necessity let the input-output transfer matrix (3.1) be realized by the GR model (2.3). Let $\boldsymbol{X}^k(z_1, z_2, z_t) = \mathbf{Z}[\boldsymbol{x}^k(n_1, n_2, t)]$ for $k \in \{h, v, t\}$, $\boldsymbol{U}(z_1, z_2, z_t) = \mathbf{Z}[\boldsymbol{u}(n_1, n_2, t)]$, $\boldsymbol{Y}(z_1, z_2, z_t) = \mathbf{Z}[\boldsymbol{y}(n_1, n_2, t)]$, $\boldsymbol{Z}_1 = z_1 \boldsymbol{I}_a$[3], $\boldsymbol{Z}_2 = z_2 \boldsymbol{I}_b$ and $\boldsymbol{Z}_t = z_t \boldsymbol{I}_c$. Let $\boldsymbol{Z} = diag\{z_1 \boldsymbol{I}_a, z_2 \boldsymbol{I}_b, z_t \boldsymbol{I}_c\}$. We have:

$$\boldsymbol{H}(z_1, z_2, z_t) = \boldsymbol{C}\boldsymbol{Z}^{-1}(\boldsymbol{I}_{a+b+c} - \boldsymbol{A}\boldsymbol{Z}^{-1})^{-1}\boldsymbol{B} + \boldsymbol{D}$$

Elements of the transfer matrix $\boldsymbol{H}(z_1, z_2, z_t)$ are rational functions. The $(l, m) - th$ element of $\boldsymbol{H}(z_1, z_2, z_t)$ is denoted by $\boldsymbol{H}(l, m)$.

$$\boldsymbol{H}(l, m) = \frac{\sum\sum\sum a_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}}{1 - \sum\sum\sum b_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}}$$

[2]It can be shown by counterexamples, that none of the 3-D realization algorithms presented in [Fan et al., 2006; Kanellakis et al., 1989; Manikopoulos and Antoniou, 1990; Theodorou and Tzafestas, 1984] realizes a transfer matrix of the form (3.1) with conditions (3.2) and (3.3) satisfied in a GR model of the form (2.3), in general

[3]Identity matrix of order n is denoted by $\boldsymbol{I}_n$

$$\boldsymbol{x}^h(n_1+1,n_2,t) = \boldsymbol{A}_3\boldsymbol{x}^t(n_1,n_2,t) + \boldsymbol{B}_1\boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{Z}_1\boldsymbol{X}^h(z_1,z_2,z_t) = \boldsymbol{A}_3\boldsymbol{X}^t(z_1,z_2,z_t) + \boldsymbol{B}_1\boldsymbol{U}(z_1,z_2,z_t)$$

$$\boldsymbol{X}^h(z_1,z_2,z_t) = z_1^{-1}\boldsymbol{A}_3\boldsymbol{X}^t(z_1,z_2,z_t) + z_1^{-1}\boldsymbol{B}_1\boldsymbol{U}(z_1,z_2,z_t) \tag{3.4}$$

Similarly:

$$\boldsymbol{X}^v(z_1,z_2,z_t) = z_2^{-1}A_6\boldsymbol{X}^t(z_1,z_2,z_t) + z_2^{-1}\boldsymbol{B}_2\boldsymbol{U}(z_1,z_2,z_t) \tag{3.5}$$

$$\begin{aligned}\boldsymbol{X}^t(z_1,z_2,z_t) = {} & z_t^{-1}\boldsymbol{A}_7\boldsymbol{X}^h(z_1,z_2,z_t) + z_t^{-1}\boldsymbol{A}_8\boldsymbol{X}^v(z_1,z_2,z_t) \\ & + z_t^{-1}\boldsymbol{A}_9\boldsymbol{X}^t(z_1,z_2,z_t) + z_t^{-1}\boldsymbol{B}_3\boldsymbol{U}(z_1,z_2,z_t)\end{aligned}$$

$$\begin{aligned}\boldsymbol{X}^t(z_1,z_2,z_t) = {} & (\boldsymbol{I}_c - z_t^{-1}z_1^{-1}\boldsymbol{A}_7\boldsymbol{A}_3 - z_t^{-1}z_2^{-1}\boldsymbol{A}_8\boldsymbol{A}_6 - z_t^{-1}\boldsymbol{A}_9)^{-1} \\ & \times (z_t^{-1}z_1^{-1}\boldsymbol{A}_7\boldsymbol{B}_1 + z_t^{-1}z_2^{-1}\boldsymbol{A}_8\boldsymbol{B}_2 + z_t^{-1}\boldsymbol{B}_3)\boldsymbol{U}(z_1,z_2,z_t)\end{aligned} \tag{3.6}$$

Let:

$$\boldsymbol{U}(z_1,z_2,z_t) = [1\ 0.....0]^T \tag{3.7}$$

Then, elements of vectors $\boldsymbol{X}^h(z_1,z_2,z_t)$, $\boldsymbol{X}^v(z_1,z_2,z_t)$ and $\boldsymbol{X}^t(z_1,z_2,z_t)$ are rational functions. Monomials in the RHS of the equation (3.6) result from products of monomials $z_t^{-1}z_1^{-1}$, $z_t^{-1}z_2^{-1}$ and z_t^{-1}. Hence any monomial $z_1^{-i}z_2^{-j}z_t^{-k}$ on the RHS of (3.6) satisfies $i+j \leq k$. By equations (3.4) and (3.5), denominators of the elements of $\boldsymbol{X}^h(z_1,z_2,z_t)$ and $\boldsymbol{X}^v(z_1,z_2,z_t)$ have monomials that conform to the same property. Monomials in the numerators of the elements of $\boldsymbol{X}^h(z_1,z_2,z_t)$ and $\boldsymbol{X}^v(z_1,z_2,z_t)$ result from multiplication of monomials of $\boldsymbol{X}^t(z_1,z_2,z_t)$ by z_1^{-1} and z_2^{-1} respectively. Therefore monomials $z_1^{-i}z_2^{-j}z_t^{-k}$ in numerators of the elements of $\boldsymbol{X}^h(z_1,z_2,z_t)$ and

$\boldsymbol{X}^v(z_1, z_2, z_t)$ satisfy $i + j \leq k + 1$.

$$\boldsymbol{Y}(z_1, z_2, z_t) = \boldsymbol{C}\boldsymbol{X}(z_1, z_2, z_t) + \boldsymbol{D}$$

Therefore, monomials $z_1^{-i} z_2^{-j} z_t^{-k}$ in the numerators and denominators of the elements of $\boldsymbol{Y}(z_1, z_2, z_t)$ satisfy $i + j \leq k + 1$ and $i + j \leq k$, respectively. Due to (3.7), $\boldsymbol{Y}(z_1, z_2, z_t)$ is equal to the first column of $\boldsymbol{H}(z_1, z_2, z_t)$. Hence:

$$a_{ijk}^{l1} = 0 \;\; for \;\; i + j > k + 1$$
$$b_{ijk}^{l1} = 0 \;\; for \;\; i + j \geq k + 1$$

By choosing $\boldsymbol{u}(n_1, n_2, t)$ appropriately, the same argument can be used to show that:

$$a_{ijk}^{lm} = 0 \;\; for \;\; i + j > k + 1$$
$$b_{ijk}^{lm} = 0 \;\; for \;\; i + j \geq k + 1$$

Therefore, $\boldsymbol{H}(z_1, z_2, z_t)$ can be represented in the form (3.1) with conditions (3.2) and (3.3) satisfied. This completes the proof of necessity.

In order to prove the sufficiency, let the transfer matrix (3.1) satisfy the conditions (3.2) and (3.3). The realization algorithm given below realizes the transfer matrix in a GR model of the form (2.3). Without loss of generality it can be assumed that $\boldsymbol{N}_R(z_1, z_2, z_t) = 0$ and $\boldsymbol{D}_R(z_1, z_2, z_t) = \boldsymbol{I}_p$ when $z_1^{-1} = z_2^{-1} = z_t^{-1} = 0$. Let:

$$\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) = \boldsymbol{I}_p - \boldsymbol{D}_R(z_1, z_2, z_t)$$
$$\boldsymbol{F}(z_1, z_2, z_t) = \begin{bmatrix} \boldsymbol{N}_R(z_1, z_2, z_t) \\ \tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) \end{bmatrix}$$

Elements of $\boldsymbol{N}_R(z_1, z_2, z_t)$ and $\boldsymbol{D}_R(z_1, z_2, z_t)$ are polynomials in z_1^{-1}, z_2^{-1} and z_t^{-1}. The basic idea behind the algorithm is to construct a matrix $\boldsymbol{\Psi}$ of size $(a+b+c)\times p$ consisting of all the monomials required for the realization. Matrices $\boldsymbol{D}_{HT} \in \mathbb{R}^{p\times(a+b+c)}$, $\boldsymbol{N}_{HT} \in \mathbb{R}^{q\times(a+b+c)}$, $\boldsymbol{B} \in \mathbb{R}^{(a+b+c)\times p}$ and $\boldsymbol{A} \in \mathbb{R}^{(a+b+c)\times(a+b+c)}$ are constructed such that:

$$\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) = \boldsymbol{D}_{HT}\boldsymbol{Z}^{-1}\boldsymbol{\Psi} \tag{3.8}$$

$$\boldsymbol{N}_R(z_1, z_2, z_t) = \boldsymbol{N}_{HT}\boldsymbol{Z}^{-1}\boldsymbol{\Psi} \tag{3.9}$$

and

$$\boldsymbol{\Psi}\boldsymbol{D}_R^{-1}(z_1, z_2, z_t) = (\boldsymbol{I}_{a+b+c} - \boldsymbol{A}\boldsymbol{Z}^{-1})^{-1}\boldsymbol{B} \tag{3.10}$$

Then:

$$\begin{aligned}\boldsymbol{N}_R(z_1, z_2, z_t)\boldsymbol{D}_R^{-1}(z_1, z_2, z_t) &= \boldsymbol{N}_{HT}\boldsymbol{Z}^{-1} \times \boldsymbol{\Psi}\boldsymbol{D}_R^{-1}(z_1, z_2, z_t) \\ &= \boldsymbol{N}_{HT}\boldsymbol{Z}^{-1}(\boldsymbol{I}_{a+b+c} - \boldsymbol{A}\boldsymbol{Z}^{-1})^{-1}\boldsymbol{B} \\ &= \boldsymbol{C}\boldsymbol{Z}^{-1}(\boldsymbol{I}_{a+b+c} - \boldsymbol{A}\boldsymbol{Z}^{-1})^{-1}\boldsymbol{B}\end{aligned}$$

where $\boldsymbol{C} = \boldsymbol{N}_{HT}$. For reasons that would become obvious later, let $\boldsymbol{\Psi}$ have the follow-

ing structure:

$$\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1^T \ \boldsymbol{\Psi}_2^T \ \boldsymbol{\Psi}_t^T]^T$$

$$= \left[\begin{array}{ccc|c} \boldsymbol{\Psi}_{11} & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \boldsymbol{\Psi}_{1p} \\ \hline \boldsymbol{\Psi}_{21} & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \boldsymbol{\Psi}_{2p} \\ \hline \boldsymbol{\Psi}_{t1} & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \boldsymbol{\Psi}_{tp} \end{array}\right] \tag{3.11}$$

To construct $\boldsymbol{D}_{HT}$, $\boldsymbol{N}_{HT}$, $\boldsymbol{B}$ and $\boldsymbol{A}$ that conform to above conditions $\boldsymbol{\Psi}$ should have the following properties.

1. Entries in the $j-th$ column of $\boldsymbol{Z}^{-1}\boldsymbol{\Psi}$ should contain all the monomials in the $j-th$ column of $\boldsymbol{F}(z_1, z_2, z_t)$. This condition guarantees that $\boldsymbol{D}_{HT}$ and $\boldsymbol{N}_{HT}$ which satisfy (3.8) and (3.9) respectively, always exist.

2. For any non-unity entry $\boldsymbol{\Psi}(i,j)$ in the $j-th$ column of $\boldsymbol{\Psi}$ there should be another entry $\boldsymbol{\Psi}(h,j)$ in the same column such that $\boldsymbol{\Psi}(i,j) = z_k^{-1}\boldsymbol{\Psi}(h,j)$ where $k \in \{1,2,t\}$. As will be explained later this property enables the construction of $\boldsymbol{A}$ and $\boldsymbol{B}$ such that equation (3.10) is satisfied.

3. There is at least one unity entry in every column of $\boldsymbol{\Psi}$. This condition in necessitated by condition 2 above.

Construct $\boldsymbol{A}_0 \in \mathbb{R}^{(a+b+c)\times(a+b+c)}$ such that $\boldsymbol{A}_0(i,j) = 1$ if the only non zero element in the $i-th$ row of $\boldsymbol{\Psi}$, say $\boldsymbol{\Psi}(i,h)$ is equal to the $(j,h)-th$ entry of $\boldsymbol{Z}^{-1}\boldsymbol{\Psi}$ and $\boldsymbol{A}_0(i,j) = 0$ otherwise. Let $\boldsymbol{B} \in \mathbb{R}^{(a+b+c)\times p}$ and $\boldsymbol{B}(i,j) = 1$ if $\boldsymbol{\Psi}(i,j) = 1$ and $\boldsymbol{B}(i,j) = 0$ otherwise.

$$\boldsymbol{\Psi} - \boldsymbol{A}_0\boldsymbol{Z}^{-1}\boldsymbol{\Psi} = \boldsymbol{B}$$

Condition 2 above guarantees that $\boldsymbol{A}_0$ and $\boldsymbol{B}$ can be constructed to satisfy the above equation.

$$(\boldsymbol{I}_{a+b+c} - \boldsymbol{A}_0\boldsymbol{Z}^{-1})\boldsymbol{\Psi} = \boldsymbol{B}$$

Let $\boldsymbol{A} = \boldsymbol{A}_0 + \boldsymbol{B}\boldsymbol{D}_{HT}$. Then:

$$\begin{aligned}
\boldsymbol{B}\boldsymbol{D}_R(z_1,z_2,z_t) &= \boldsymbol{B}(1 - \tilde{\boldsymbol{D}}_R(z_1,z_2,z_t)) \\
&= (\boldsymbol{I}_{a+b+c} - \boldsymbol{A}_0\boldsymbol{Z}^{-1})\boldsymbol{\Psi} - \boldsymbol{B}\boldsymbol{D}_{HT}\boldsymbol{Z}^{-1}\boldsymbol{\Psi} \\
&= (\boldsymbol{I}_{a+b+c} - \boldsymbol{A}_0\boldsymbol{Z}^{-1} - \boldsymbol{B}\boldsymbol{D}_{HT}\boldsymbol{Z}^{-1})\boldsymbol{\Psi} \\
&= (\boldsymbol{I}_{a+b+c} - (\boldsymbol{A}_0 + \boldsymbol{B}\boldsymbol{D}_{HT})\boldsymbol{Z}^{-1})\boldsymbol{\Psi} \\
&= (\boldsymbol{I}_{a+b+c} - \boldsymbol{A}\boldsymbol{Z}^{-1})\boldsymbol{\Psi}
\end{aligned}$$

$$\boldsymbol{\Psi}\boldsymbol{D}_R^{-1}(z_1,z_2,z_t) = (\boldsymbol{I}_{a+b+c} - \boldsymbol{A}\boldsymbol{Z}^{-1})^{-1}\boldsymbol{B}$$

Vector $\boldsymbol{\Psi}$ must be constructed such that the matrix $\boldsymbol{A}$ resulting from the above procedure is in the desired form. Matrix $\boldsymbol{A}$ should be such that $\boldsymbol{A}(i,j) = 0$ for $(i,j) \in [1, a+b]\times[1, a+b]$. Matrix $\boldsymbol{A}$ would have the desired property if $\boldsymbol{A}_0$ and $\boldsymbol{D}_{HT}$ are such that $\boldsymbol{A}_0(i,j) = 0$ for $(i,j) \in [1, a+b] \times [1, a+b]$ and $\boldsymbol{D}_{HT}(i,j) = 0$ for $j \leq a+b$. Let $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1^T, \boldsymbol{\Psi}_2^T, \boldsymbol{\Psi}_t^T]^T$ where $\boldsymbol{\Psi}_1$, $\boldsymbol{\Psi}_2$ and $\boldsymbol{\Psi}_t$ are of size $a \times p$, $b \times p$ and $c \times p$

respectively. For the matrix $\boldsymbol{A}$ to be in the desired form the following properties must be satisfied by $\boldsymbol{\Psi}_1$, $\boldsymbol{\Psi}_2$ and $\boldsymbol{\Psi}_t$.

4. All the monomials in the $j-th$ column of $\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t)$ should be contained in the same column of $z_t^{-1}\boldsymbol{\Psi}_t$.

5. For every non-unity and non-zero entry $\boldsymbol{\Psi}_1(i,j)$ in $\boldsymbol{\Psi}_1$ there should be a term $\boldsymbol{\Psi}_t(h,j)$ in $\boldsymbol{\Psi}_t$ such that $\boldsymbol{\Psi}_1(i,j) = z_t^{-1}\boldsymbol{\Psi}_t(h,j)$

6. For every non-unity and non-zero entry $\boldsymbol{\Psi}_2(i,j)$ in $\boldsymbol{\Psi}_2$ there should be a term $\boldsymbol{\Psi}_t(h,j)$ in $\boldsymbol{\Psi}_t$ such that $\boldsymbol{\Psi}_2(i,j) = z_t^{-1}\boldsymbol{\Psi}_t(h,j)$

Condition 4 guarantees that $\boldsymbol{D}_{HT}$ is in the desired form. Properties 5 and 6 enable $\boldsymbol{A}_0$ to be constructed in the desired form. The algorithm given in figure 3.1 can be used to construct the matrix $\boldsymbol{\Psi}$ with the above properties.

Steps 2-6 of the algorithm ensure that all the monomials in $\boldsymbol{F}(z_1, z_2, z_t)$ and $\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t)$ are contained in $\boldsymbol{Z}^{-1}\boldsymbol{\Psi}$ and $z_t^{-1}\boldsymbol{\Psi}_t$ respectively. The monomials missing in $\boldsymbol{\Psi}$ that are needed to satisfy conditions 2, 5 and 6 are inserted in the loop starting from step 8. Note that the monomials $z_1^{-i}z_2^{-j}z_t^{-k}$ in $\boldsymbol{\Phi}_m$ for $m \in [1,p]$ have the property $i+j \leq k+1$. Hence for every monomial K inserted into $\boldsymbol{\Psi}_{1m}$ or $\boldsymbol{\Psi}_{2m}$, z_tK is inserted into $\boldsymbol{\Psi}_{tm}$. Further for every monomial K inserted into $\boldsymbol{\Psi}_{tm}$, z_kK is inserted to $\boldsymbol{\Psi}_{km}$, where $k \in \{1,2,t\}$. This ensures that properties 2, 5 and 6 are satisfied by $\boldsymbol{\Psi}_1$, $\boldsymbol{\Psi}_2$ and $\boldsymbol{\Psi}_t$. This completes the proof of the theorem.

1: **for** $j = 1$ to p **do**
2: Collect all the monomials $z_1^{-l} z_2^{-m} z_t^{-n}$ in the $j-th$ column of $\boldsymbol{N}_R(z_1, z_2, z_t)$ that satisfy $l+m = n+1$ and has at least one power of z_1^{-1} to $\boldsymbol{\Phi}_{1j}$ in the ascending total degree lexicographic order [4].
3: Collect all the monomials $z_1^{-l} z_2^{-m} z_t^{-n}$ in the $j-th$ column of $\boldsymbol{N}_R(z_1, z_2, z_t)$ and not in $\boldsymbol{\Phi}_{1j}$ that satisfy $l+m = n+1$ and has at least one power of z_2^{-1} to $\boldsymbol{\Phi}_{2j}$ in the ascending total degree lexicographic order.
4: Collect all the other distinct monomials in the $j-th$ column of $\boldsymbol{F}(z_1, z_2, z_t)$ to vector $\boldsymbol{\Phi}_{tj}$ in the ascending total degree lexicographic order.
5: $\boldsymbol{\Phi}_{kj} = z_k \boldsymbol{\Phi}_{kj} \quad k \in \{1, 2, t\}$
6: $\boldsymbol{\Psi}_{kj} = \boldsymbol{\Phi}_{kj} \quad k \in \{1, 2, t\}$
7: $\boldsymbol{\Phi}_j = [\boldsymbol{\Phi}_{1j}^T \; \boldsymbol{\Phi}_{2j}^T \; \boldsymbol{\Phi}_{tj}^T]^T$
8: **for** $i = 0$ to $Number\ of\ elements\ in\ \boldsymbol{\Phi}_j$ **do**
9: $K = \boldsymbol{\Phi}_j(i)$
10: **if** K is an element of $\boldsymbol{\Phi}_{1j}$ or $\boldsymbol{\Phi}_{2j}$ **then**
11: **if** There is at least one power of z_t^{-1} in K **then**
12: **if** $z_t K$ is not in $\boldsymbol{\Psi}_{tj}$ **then**
13: Insert $z_t K$ to $\boldsymbol{\Psi}_{tj}$ in the ascending total degree lexicographic order.
14: **end if**
15: $K = z_t K$
16: **end if**
17: **else**
18: **while** $K \neq 1$ and $orderz_1(K) + orderz_2(K) \leq orderz_t(K)$ [5] **do**
19: **if** There is at least one power of z_t^{-1} in K **then**
20: **if** $z_t K$ is not in $\boldsymbol{\Psi}_{tj}$ **then**
21: Insert $z_t K$ to $\boldsymbol{\Psi}_{tj}$ in the ascending total degree lexicographic order.
22: **end if**

Figure 3.1. Algorithm for constructing $\boldsymbol{\Psi}$

[4] $z_1^{-1} < z_2^{-1} < z_t^{-1} < z_1^{-2} < z_1^{-1} z_2^{-1} < z_2^{-2}$.....

[5] $orderz_1(z_1^{-l} z_2^{-m} z_t^{-n}) = l$, $orderz_2(z_1^{-l} z_2^{-m} z_t^{-n}) = m$ and $orderz_t(z_1^{-l} z_2^{-m} z_t^{-n}) = n$

$K = z_t K$
end if
end while
end if
while $K \neq 1$ **do**
if There is at least one power of z_1^{-1} in K **then**
if $z_1 K$ is not in $\mathbf{\Psi}_{1j}$ **then**
Insert $z_1 K$ to $\mathbf{\Psi}_{1j}$ in the ascending total degree lexicographic order.
end if
$K = z_1 K$
if There is at least one power of z_t^{-1} in K **then**
if $z_t K$ is not in $\mathbf{\Psi}_{tj}$ **then**
Insert $z_t K$ to $\mathbf{\Psi}_{tj}$ in the ascending total degree lexicographic order.
end if
$K = z_t K$
end if
else
if There is at least one power of z_2^{-1} in K **then**
if $z_2 K$ is not in $\mathbf{\Psi}_{2j}$ **then**
Insert $z_2 K$ to $\mathbf{\Psi}_{2j}$ in the ascending total degree lexicographic order.
end if
$K = z_2 K$
end if
if There is at least one power of z_t^{-1} in K **then**
if $z_t K$ is not in $\mathbf{\Psi}_{tj}$ **then**
Insert $z_t K$ to $\mathbf{\Psi}_{tj}$ in the ascending total degree lexicographic order.
end if
$K = z_t K$
end if
end if
end while
end for
end for
Using the vectors $\mathbf{\Psi}_{tj}$ where $k \in \{1, 2, t\}$ and $j \in [1, p]$ construct the matrix $\mathbf{\Psi}$ according to the structure given in (3.11).

Figure 3.1. continued

For the sake of simplicity, implications of the theorem would be discussed for the case of single input single output systems. In this case, the right matrix fraction description of the system given by (3.1), reduces to a rational transfer function. Let the impulse response of the system described by the transfer function (3.1) be $h(n_1, n_2, t)$. If conditions (3.2) and (3.3) are satisfied by the transfer function:

$$h(n_1, n_2, t) = 0 \ \ for \ \ n_1 + n_2 > t + 1 \tag{3.12}$$

Therefore an impulse response realizable in a GR model of the form (2.3) satisfies the condition (3.12). The GR model (2.3) is implementable in a grid sensor network in which information is conveyed over a single hop in one time slot. The effect of an input at a node propagates over a single hop in a time slot. Therefore the result is intuitive. Let a first octant causal impulse response satisfying the condition (3.12) have a rational Z-transform $\boldsymbol{H}(z_1, z_2, z_t)$. It can be easily shown that it satisfies the conditions (3.2) and (3.3). Hence any first octant causal impulse response satisfying (3.12) with a rational Z-transform is realizable in the GR model (2.3).

3.1.2 Non Proper Transfer Matrices

Systems described by the GR model (2.3) are necessarily first octant causal. Systems causal in any of the four quadrants of the spatial plane can be realized as:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+\alpha, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2+\beta, t) \\ \boldsymbol{x}^t(n_1, n_2, t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_3 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2, t) \\ \boldsymbol{x}^t(n_1, n_2, t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1, n_2, t)$$

$$\boldsymbol{y}(n_1, n_2, t) = \boldsymbol{C}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t) \tag{3.13}$$

by appropriately choosing α and β. Here $(\alpha, \beta) \in \{-1, 1\} \times \{-1, 1\}$. For example when $(\alpha, \beta) = (-1, -1)$ the region of support of the impulse response lies in the 3^{rd} quadrant of the spatial plane. The following lemma which is a generalization of the Theorem 3.1.1 can be used to derive the final result of this section.

Lemma 3.1.1 *Let an input-output transfer matrix be given by the right matrix fraction description:*

$$\boldsymbol{H}_{\alpha\beta}(z_1, z_2, z_t) = \boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)\boldsymbol{D}^{-1}_{\alpha\beta}(z_1, z_2, z_t) \tag{3.14}$$

where $(\alpha, \beta) \in \{-1, 1\} \times \{-1, 1\}$ *and where matrices* $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ *and* $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ *are of size* $q \times p$ *and* $p \times p$ *respectively. Let* $(l, m)-th$ *elements of matrices* $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ *and* $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ *be denoted by* $\boldsymbol{N}_{\alpha\beta}(l, m)$ *and* $\boldsymbol{D}_{\alpha\beta}(l, m)$ *respectively. Let:*

$$\boldsymbol{N}_{\alpha\beta}(l, m) = \sum_{i=0}^{N_{\alpha\beta1}} \sum_{j=0}^{N_{\alpha\beta2}} \sum_{k=0}^{N_{\alpha\beta t}} n^{lm}_{ijk(\alpha\beta)} z_1^{-\alpha i} z_2^{-\beta j} z_t^{-k}$$

$$\boldsymbol{D}_{\alpha\beta}(l, m) = \sum_{i=0}^{D_{\alpha\beta1}} \sum_{j=0}^{D_{\alpha\beta2}} \sum_{k=0}^{D_{\alpha\beta t}} d^{lm}_{ijk(\alpha\beta)} z_1^{-\alpha i} z_2^{-\beta j} z_t^{-k}$$

where $N_{\alpha\beta1}$, $N_{\alpha\beta2}$ *and* $N_{\alpha\beta t}$ *are the degrees of the polynomial matrix* $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ *with respect to variables* z_1, z_2 *and* z_t *respectively. Furthermore* $D_{\alpha\beta1}$, $D_{\alpha\beta2}$ *and* $D_{\alpha\beta t}$ *are the degrees of the polynomial matrix* $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ *with respect to variables* z_1, z_2 *and* z_t *respectively. The transfer matrix (3.14) can be realized in a GR model of the form (3.13) if and only if the conditions (3.15) and (3.16) are satisfied.*

$$n^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i + j > k + 1 \tag{3.15}$$

$$d^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i + j \geq k + 1 \tag{3.16}$$

Proof For the case of $(\alpha, \beta) = (1, 1)$ the lemma is a restatement of Theorem 3.1.1.

For other cases it can be proven using an identical approach to the proof of Theorem 3.1.1.

A parallel combination of systems causal in each quadrant can be used to realize non causal systems [Lele and Mendel, 1987].

Theorem 3.1.2 *A transfer matrix $\boldsymbol{H}(z_1, z_2, z_t)$ can be realized as a parallel combination of GR models of the form (3.13) if and only if it can be expressed as:*

$$\boldsymbol{H}(z_1, z_2, z_t) = \sum_{(\alpha,\beta)\in\{-1,1\}\times\{-1,1\}} \boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)\boldsymbol{D}^{-1}_{\alpha\beta}(z_1, z_2, z_t) \tag{3.17}$$

where elements of matrices $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ and $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ are polynomials of $z_1^{-\alpha}$, $z_2^{-\beta}$ and z_t^{-1} satisfying the following conditions: If the $(l, m) - th$ elements of $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ and $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ are given by:

$$\boldsymbol{N}_{\alpha\beta}(l, m) = \sum_{i=0}^{N_{\alpha\beta1}} \sum_{j=0}^{N_{\alpha\beta2}} \sum_{k=0}^{N_{\alpha\beta t}} n^{lm}_{ijk(\alpha\beta)} z_1^{-i\alpha} z_2^{-j\beta} z_t^{-k}$$

$$\boldsymbol{D}_{\alpha\beta}(l, m) = \sum_{i=0}^{D_{\alpha\beta1}} \sum_{j=0}^{D_{\alpha\beta1}} \sum_{k=0}^{D_{\alpha\beta t}} d^{lm}_{ijk(\alpha\beta)} z_1^{-i\alpha} z_2^{-j\beta} z_t^{-k}$$

then

$$n^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i + j > k + 1 \tag{3.18}$$

$$d^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i + j \geq k + 1 \tag{3.19}$$

Proof The result trivially follows from Lemma 3.1.1.

3.1.3 Example

The realization algorithm discussed in the above section will be illustrated by an example in this section. Let the transfer matrix to be realized be as follows:

$$\boldsymbol{H}(z_1, z_2, z_t) = \begin{bmatrix} a^{11}_{100}z_1^{-1} + a^{11}_{010}z_2^{-1} + a^{11}_{101}z_1^{-1}z_t^{-1} & a^{12}_{010}z_2^{-1} \\ a^{21}_{001}z_t^{-1} & a^{22}_{101}z_1^{-1}z_t^{-1} \end{bmatrix} \times \begin{bmatrix} 1 - b^{11}_{011}z_2^{-1}z_t^{-1} - b^{11}_{001}z_t^{-1} - b^{11}_{112}z_1^{-1}z_2^{-1}z_t^{-2} & -b^{12}_{001}z_t^{-1} \\ -b^{21}_{011}z_2^{-1}z_t^{-1} & 1 - b^{22}_{101}z_1^{-1}z_t^{-1} \end{bmatrix}^{-1} \tag{3.20}$$

According to Theorem 3.1.1, the above transfer matrix is realizable in a GR model of the form (2.3).

$$\boldsymbol{F}(z_1, z_2, z_t) = \begin{bmatrix} a^{11}_{100}z_1^{-1} + a^{11}_{010}z_2^{-1} + a^{11}_{101}z_1^{-1}z_t^{-1} & a^{12}_{010}z_2^{-1} \\ a^{21}_{001}z_t^{-1} & a^{22}_{101}z_1^{-1}z_t^{-1} \\ b^{11}_{011}z_2^{-1}z_t^{-1} + b^{11}_{001}z_t^{-1} + b^{11}_{112}z_1^{-1}z_2^{-1}z_t^{-2} & b^{12}_{001}z_t^{-1} \\ b^{21}_{011}z_2^{-1}z_t^{-1} & b^{22}_{101}z_1^{-1}z_t^{-1} \end{bmatrix}$$

Since the system has two inputs, the loop starting in step 1 of the algorithm given in figure 3.1 runs twice. The sequence of operations in the first iteration is given in table 3.1.

TABLE 3.1:

THE SEQUENCE OF OPERATIONS IN THE FIRST ITERATION OF THE ALGORITHM

step	operation
2	$\mathbf{\Phi}_{11} = [z_1^{-1}]^T$
3	$\mathbf{\Phi}_{21} = [z_2^{-1}]^T$
4	$\mathbf{\Phi}_{t1} = [z_t^{-1} \quad z_1^{-1}z_t^{-1} \quad z_2^{-1}z_t^{-1} \quad z_1^{-1}z_2^{-1}z_t^{-2}]^T$
5	$\mathbf{\Phi}_{11} = \begin{bmatrix} 1 \end{bmatrix}$ $\mathbf{\Phi}_{21} = \begin{bmatrix} 1 \end{bmatrix}$ $\mathbf{\Phi}_{t1} = \begin{bmatrix} 1 \\ z_1^{-1} \\ z_2^{-1} \\ z_1^{-1}z_2^{-1}z_t^{-1} \end{bmatrix}$
6	$\mathbf{\Psi}_{11} = \begin{bmatrix} 1 \end{bmatrix}$ $\mathbf{\Psi}_{21} = \begin{bmatrix} 1 \end{bmatrix}$ $\mathbf{\Psi}_{t1} = \begin{bmatrix} 1 \\ z_1^{-1} \\ z_2^{-1} \\ z_1^{-1}z_2^{-1}z_t^{-1} \end{bmatrix}$
7	$\mathbf{\Phi}_1 = [1 \quad 1 \quad 1 \quad z_1^{-1} \quad z_2^{-1} \quad z_1^{-1}z_2^{-1}z_t^{-1}]^T$
8	$i = 1$ This is the first iteration of the **for** loop starting at step 8.
9	$K = 1$ No elements are inserted to $\mathbf{\Psi}_{11}$, $\mathbf{\Psi}_{21}$ or $\mathbf{\Psi}_{t1}$ in this iteration.
8	$i = 2$ This is the second iteration of the **for** loop starting at step 8.
9	$K = 1$ No elements are inserted to $\mathbf{\Psi}_{11}$, $\mathbf{\Psi}_{21}$ or $\mathbf{\Psi}_{t1}$ in this iteration.
8	$i = 3$ This is the third iteration of the **for** loop

TABLE 3.1 continued

	starting at step 8.
9	$K = 1$ No elements are inserted to $\mathbf{\Psi}_{11}$, $\mathbf{\Psi}_{21}$ or $\mathbf{\Psi}_{t1}$ in this iteration.
8	$i = 4$ This is the fourth iteration of the **for** loop starting at step 8.
9	$K = z_1^{-1}$
10	K is not an element of $\mathbf{\Phi}_{11}$ or $\mathbf{\Phi}_{21}$
18	$orderz_1(z_1^{-1}) + orderz_2(z_1^{-1}) > orderz_t(z_1^{-1})$
27	$K \neq 1$
28	There is one power of z_1^{-1} in K
29	$z_1K = 1$ is in $\mathbf{\Psi}_{11}$
32	$K = z_1K$ Since $K = 1$ the iteration ends
8	$i = 5$ This is the fifth iteration of the **for** loop starting at step 8.
9	$K = z_2^{-1}$
10	K is not an element of $\mathbf{\Phi}_{11}$ or $\mathbf{\Phi}_{21}$
18	$orderz_1(z_1^{-1}) + orderz_2(z_1^{-1}) > orderz_t(z_1^{-1})$
27	$K \neq 1$
28	There is no power of z_1^{-1} in K
40	There is one power of z_2^{-1} in K
41	$z_2K = 1$ is in $\mathbf{\Psi}_{21}$
44	$K = z_2K$ Since $K = 1$ the iteration ends
8	$i = 6$ This is the sixth iteration of the **for** loop starting at step 8.

TABLE 3.1 continued

9	$K = z_1^{-1} z_2^{-1} z_t^{-1}$
10	K is not an element of $\mathbf{\Phi}_{11}$ or $\mathbf{\Phi}_{21}$
18	$order z_1(z_1^{-1}) + order z_2(z_1^{-1}) > order z_t(z_1^{-1})$
27	$K \neq 1$
28	There is one power of z_1^{-1} in K
29	$z_1 K = z_2^{-1} z_t^{-1}$ is not in $\mathbf{\Psi}_{11}$
30	Insert $z_2^{-1} z_t^{-1}$ into $\mathbf{\Psi}_{11}$
32	$K = z_1 K$ Now $K = z_2^{-1} z_t^{-1}$
33	There is one power of z_t^{-1} in K
34	$z_t K = z_2^{-1}$ is in $\mathbf{\Psi}_{t1}$
37	$K = z_t K$ Now $K = z_2^{-1}$
27	$K \neq 1$
28	There is no power of z_1^{-1} in K
40	There is one power of z_2^{-1} in K
41	$z_2 K = 1$ is in $\mathbf{\Psi}_{21}$
44	$K = z_2 K$ Since $K = 1$ the iteration ends

At the end of the first iteration of the loop starting at step 1 of the algorithm we have $\mathbf{\Psi}_{11}$, $\mathbf{\Psi}_{21}$ and $\mathbf{\Psi}_{t1}$. Vectors $\mathbf{\Psi}_{12}$, $\mathbf{\Psi}_{22}$ and $\mathbf{\Psi}_{t2}$ can be derived in the second iteration which is omitted for the sake of brevity. In accordance with the structure given in (3.11), $\mathbf{\Psi}$ can be constructed.

$$\mathbf{\Psi} = \begin{bmatrix} 1 & 0 \\ z_2^{-1} z_t^{-1} & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ z_1^{-1} & 0 \\ z_2^{-1} & 0 \\ z_1^{-1} z_2^{-1} z_t^{-1} & 0 \\ 0 & 1 \\ 0 & z_1^{-1} \end{bmatrix}$$

Matrices $\boldsymbol{D}_{HT}$ and $\boldsymbol{N}_{HT}$can be derived using (3.8) and (3.9).

$$\boldsymbol{D}_{HT} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & b_{001}^{11} & 0 & b_{011}^{11} & b_{112}^{11} & b_{001}^{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{011}^{21} & 0 & 0 & b_{101}^{22} \end{bmatrix}$$

$$\boldsymbol{N}_{HT} = \begin{bmatrix} a_{100}^{11} & 0 & 0 & a_{010}^{11} & a_{010}^{12} & 0 & a_{101}^{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{001}^{21} & 0 & 0 & 0 & 0 & a_{101}^{22} \end{bmatrix}$$

Matrix $\boldsymbol{C} = \boldsymbol{N}_{HT}$. Matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can be derived using the procedure given in

section 3.1.1.

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$\boldsymbol{A} = \left[\begin{array}{ccccc|cccccc} 0 & 0 & 0 & 0 & 0 & b_{001}^{11} & 0 & b_{011}^{11} & b_{112}^{11} & b_{001}^{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{011}^{21} & 0 & 0 & b_{101}^{22} \\ \hline 0 & 0 & 0 & 0 & 0 & b_{001}^{11} & 0 & b_{011}^{11} & b_{112}^{11} & b_{001}^{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{011}^{21} & 0 & 0 & b_{101}^{22} \\ \hline 0 & 0 & 0 & 0 & 0 & b_{001}^{11} & 0 & b_{011}^{11} & b_{112}^{11} & b_{001}^{12} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{011}^{21} & 0 & 0 & b_{101}^{22} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

In this realization $\boldsymbol{x}^h \in \mathbb{R}^3$, $\boldsymbol{x}^v \in \mathbb{R}^2$ and $\boldsymbol{x}^t \in \mathbb{R}^6$

3.2 Realizability in the FM Model

In this section, a necessary and sufficient condition for a 3-D rational transfer matrix to be realizable in an FM model, that satisfies condition (2.4), is established . A realization algorithm for the derivation of an FM model that satisfies the desired condition, given an admissible transfer matrix[6], is presented as part of the proof.

[6] A transfer matrix realizable in an FM model that satisfies condition (2.4)

3.2.1 Causal Transfer Matrices

Theorem 3.2.1 *Let an input-output transfer matrix be given by the right matrix fraction description Bose [2003]:*

$$\boldsymbol{H}(z_1, z_2, z_t) = \boldsymbol{N}_R(z_1, z_2, z_t)\boldsymbol{D}_R^{-1}(z_1, z_2, z_t) \tag{3.21}$$

where matrices $\boldsymbol{N}_R(z_1, z_2, z_t)$ and $\boldsymbol{D}_R(z_1, z_2, z_t)$ are of size $q \times p$ and $p \times p$ respectively. Let $\boldsymbol{N}_R(l, m)$ and $\boldsymbol{D}_R(l, m)$ denote (l, m)-th elements of matrices $\boldsymbol{N}_R(z_1, z_2, z_t)$ and $\boldsymbol{D}_R(z_1, z_2, z_t)$ respectively. Let:

$$\boldsymbol{N}_R(l, m) = \sum_{i=0}^{N_{R1}} \sum_{j=0}^{N_{R2}} \sum_{k=0}^{N_{Rt}} n_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}$$

$$\boldsymbol{D}_R(l, m) = \sum_{i=0}^{D_{R1}} \sum_{j=0}^{D_{R2}} \sum_{k=0}^{D_{Rt}} d_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}$$

where N_{R1}, N_{R2} and N_{Rt} are the degrees of the polynomial matrix $\boldsymbol{N}_R(z_1, z_2, z_t)$ with respect to variables z_1^{-1}, z_2^{-1} and z_t^{-1} respectively. Furthermore D_{R1}, D_{R2} and D_{Rt} are the degrees of the polynomial matrix $\boldsymbol{D}_R(z_1, z_2, z_t)$ with respect to variables z_1^{-1}, z_2^{-1} and z_t^{-1} respectively. The transfer matrix (3.21) can be realized in an FM model (2.2) *satisfying condition* (2.4)*, if and only if the conditions (3.22) and (3.23) are satisfied.*

$$n_{ijk}^{lm} = 0 \;\; for \;\; i + j > k + 1 \tag{3.22}$$

$$d_{ijk}^{lm} = 0 \;\; for \;\; i + j \geq k + 1 \tag{3.23}$$

Proof Necessity is proved by showing that, if a transfer matrix is realizable in an FM model (2.2) satisfying condition (2.4), it can be represented in the form (3.21) with conditions (3.22) and (3.23) satisfied. In order to prove sufficiency, a realization algorithm which realizes a transfer matrix (3.21) that satisfies conditions (3.22) and

(3.23), in an FM model satisfying condition (2.4), is presented. The proposed algorithm is a modification of the algorithm given in Cheng et al. [2010]; Xu et al. [2005, 2007].

To prove the necessity, let the input-output transfer matrix (3.21) be realized by the FM model (2.2) with condition (2.4) satisfied. From (2.2) we have:

$$\begin{aligned}\boldsymbol{x}(n_1-1,n_2,t)=&\boldsymbol{A}_h\boldsymbol{x}(n_1-2,n_2,t)+\boldsymbol{A}_v\boldsymbol{x}(n_1-1,n_2-1,t)+\boldsymbol{A}_t\boldsymbol{x}(n_1-1,n_2,t-1)\\&+\boldsymbol{B}_h\boldsymbol{u}(n_1-2,n_2,t)+\boldsymbol{B}_v\boldsymbol{u}(n_1-1,n_2-1,t)+\boldsymbol{B}_t\boldsymbol{u}(n_1-1,n_2,t-1)\end{aligned}$$

Due to condition (2.4):

$$\boldsymbol{A}_h\boldsymbol{x}(n_1-1,n_2,t)=\boldsymbol{A}_h\boldsymbol{A}_t\boldsymbol{x}(n_1-1,n_2,t-1)+\boldsymbol{A}_h\boldsymbol{B}_t\boldsymbol{u}(n_1-1,n_2,t-1)$$

$$\boldsymbol{A}_h\boldsymbol{X}(z_1,z_2,z_t)z_1^{-1}=\boldsymbol{A}_h\boldsymbol{A}_t\boldsymbol{X}(z_1,z_2,z_t)z_1^{-1}z_t^{-1}+\boldsymbol{A}_h\boldsymbol{B}_t\boldsymbol{U}(z_1,z_2,z_t)z_1^{-1}z_t^{-1} \tag{3.24}$$

Similarly:

$$\boldsymbol{A}_v\boldsymbol{X}(z_1,z_2,z_t)z_2^{-1}=\boldsymbol{A}_v\boldsymbol{A}_t\boldsymbol{X}(z_1,z_2,z_t)z_2^{-1}z_t^{-1}+\boldsymbol{A}_v\boldsymbol{B}_t\boldsymbol{U}(z_1,z_2,z_t)z_2^{-1}z_t^{-1} \tag{3.25}$$

$$\begin{aligned}\boldsymbol{X}(z_1,z_2,z_t)=&\boldsymbol{A}_h\boldsymbol{X}(z_1,z_2,z_t)z_1^{-1}+\boldsymbol{A}_v\boldsymbol{X}(z_1,z_2,z_t)z_2^{-1}+\boldsymbol{A}_t\boldsymbol{X}(z_1,z_2,z_t)z_t^{-1}\\&+\boldsymbol{B}_h\boldsymbol{U}(z_1,z_2,z_t)z_1^{-1}+\boldsymbol{B}_v\boldsymbol{U}(z_1,z_2,z_t)z_2^{-1}+\boldsymbol{B}_t\boldsymbol{U}(z_1,z_2,z_t)z_t^{-1}\\\boldsymbol{Y}(z_1,z_2,z_t)=&\boldsymbol{C}\boldsymbol{X}(z_1,z_2,z_t)+\boldsymbol{D}\boldsymbol{U}(z_1,z_2,z_t)\end{aligned} \tag{3.26}$$

By substituting from (3.24) and (3.25) in (3.26), we have:

$$\boldsymbol{H}(z_1, z_2, z_t) = [\boldsymbol{C}(\boldsymbol{I}_n - \boldsymbol{A}_h\boldsymbol{A}_t z_1^{-1} z_t^{-1} - \boldsymbol{A}_v\boldsymbol{A}_t z_2^{-1} z_t^{-1} - \boldsymbol{A}_t z_t)^{-1} \times (\boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1} + \boldsymbol{A}_h\boldsymbol{B}_t z_1^{-1} z_t^{-1} + \boldsymbol{A}_v\boldsymbol{B}_t z_2^{-1} z_t^{-1})] + \boldsymbol{D}$$

Elements of the transfer matrix $\boldsymbol{H}(z_1, z_2, z_t)$ are rational functions. The (l, m)-th element of $\boldsymbol{H}(z_1, z_2, z_t)$ is denoted by $\boldsymbol{H}(l, m)$.

$$\boldsymbol{H}(l, m) = \frac{\sum\sum\sum a_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}}{1 - \sum\sum\sum b_{ijk}^{lm} z_1^{-i} z_2^{-j} z_t^{-k}}$$

Assume

$$\boldsymbol{U}(z_1, z_2, z_t) = [1\ 0.....0]^T \tag{3.27}$$

$$\boldsymbol{Y}(z_1, z_2, z_t) = \boldsymbol{H}(z_1, z_2, z_t)\boldsymbol{U}(z_1, z_2, z_t)$$

Monomials in the denominators of the elements of $\boldsymbol{Y}(z_1, z_2, z_t)$ result from products of monomials z_t^{-1}, $z_2^{-1} z_t^{-1}$ and $z_1^{-1} z_t^{-1}$. Therefore monomials $z_1^{-i} z_2^{-j} z_t^{-k}$ in the denominators of the elements of $\boldsymbol{Y}(z_1, z_2, z_t)$ satisfy $i + j \leq k$. Monomials in the numerators of the elements of $\boldsymbol{Y}(z_1, z_2, z_t)$ result from products of monomials z_t^{-1}, z_1^{-1}, z_2^{-1}, $z_2^{-1} z_t^{-1}$ and $z_1^{-1} z_t^{-1}$. Therefore monomials $z_1^{-i} z_2^{-j} z_t^{-k}$ in the numerators of the elements of $\boldsymbol{Y}(z_1, z_2, z_t)$ satisfy $i + j \leq k + 1$. Due to (3.27), $\boldsymbol{Y}(z_1, z_2, z_t)$ is equal to the first column of $\boldsymbol{H}(z_1, z_2, z_t)$. Hence:

$$a_{ijk}^{l1} = 0 \ \ for \ \ i + j > k + 1$$

$$b_{ijk}^{l1} = 0 \ \ for \ \ i + j \geq k + 1$$

Select $\boldsymbol{u}(n_1, n_2, t)$ such that the $m\text{-}th$ element in the vector $\boldsymbol{U}(z_1, z_2, z_t)$ is unity and all other elements are zero. Then the same argument can be used to show that:

$$a_{ijk}^{lm} = 0 \;\; for \;\; i + j > k + 1$$

$$b_{ijk}^{lm} = 0 \;\; for \;\; i + j \geq k + 1$$

Therefore, $\boldsymbol{H}(z_1, z_2, z_t)$ can be represented in the form (3.21) while satisfying conditions (3.22) and (3.23). This completes the proof of necessity.

In order to prove sufficiency, let the transfer matrix (3.21) satisfy conditions (3.22) and (3.23). The realization algorithm given below realizes the transfer matrix (3.21) in the FM model (2.2) while satisfying condition (2.4). Without loss of generality it can be assumed that $\boldsymbol{N}_R(z_1, z_2, z_t) = \boldsymbol{0}$ and $\boldsymbol{D}_R(z_1, z_2, z_t) = \boldsymbol{I}_p$ when $z_1^{-1} = z_2^{-1} = z_t^{-1} = 0$. Let $\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) = \boldsymbol{I}_p - \boldsymbol{D}_R(z_1, z_2, z_t)$ and:

$$\boldsymbol{F}(z_1, z_2, z_t) = \begin{bmatrix} \boldsymbol{N}_R(z_1, z_2, z_t) \\ \tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) \end{bmatrix}$$

The basic idea behind the algorithm is to construct a matrix $\boldsymbol{\Psi}$ of size $n \times p$ containing all the monomials required for the realization. Matrix $\boldsymbol{\Psi}$ is of the following form:

$$\boldsymbol{\Psi} = diag\{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 ..... \boldsymbol{\Psi}_p\}$$

where each $\boldsymbol{\Psi}_l$ for $l \in \{1, 2, ...p\}$ is a column vector of size $n_l \times 1$ whose elements are monomials of the form $z_1^{-i} z_2^{-j} z_t^{-k}$ where i, j and k are non-negative integers. Matrices $\boldsymbol{D}_{HT} \in \mathbb{R}^{p \times n}$, $\boldsymbol{N}_{HT} \in \mathbb{R}^{q \times n}$, $\boldsymbol{A}_h \in \mathbb{R}^{n \times n}$, $\boldsymbol{A}_v \in \mathbb{R}^{n \times n}$, $\boldsymbol{A}_t \in \mathbb{R}^{n \times n}$, $\boldsymbol{B}_h \in \mathbb{R}^{n \times p}$,

$\boldsymbol{B}_v \in \mathbb{R}^{n\times p}$ and $\boldsymbol{B}_t \in \mathbb{R}^{n\times p}$ are constructed such that:

$$\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) = \boldsymbol{D}_{HT}\boldsymbol{\Psi} \tag{3.28}$$

$$\boldsymbol{N}(z_1, z_2, z_t) = \boldsymbol{N}_{HT}\boldsymbol{\Psi} \tag{3.29}$$

$$\boldsymbol{B}_v z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1} = (\boldsymbol{I}_n - \boldsymbol{A}_h z_1^{-1} - \boldsymbol{A}_v z_2^{-1} - \boldsymbol{A}_t z_t^{-1}) \times \boldsymbol{\Psi}\boldsymbol{D}_R^{-1}(z_1, z_2, z_t) \tag{3.30}$$

Then:

$$\begin{aligned}\boldsymbol{N}_R(z_1, z_2, z_t)\boldsymbol{D}_R^{-1}(z_1, z_2, z_t) &= \boldsymbol{N}_{HT}\boldsymbol{\Psi}\boldsymbol{D}_R^{-1}(z_1, z_2, z_t)\\ &= \boldsymbol{N}_{HT}(\boldsymbol{I}_n - \boldsymbol{A}_h z_1^{-1} - \boldsymbol{A}_v z_2^{-1} - \boldsymbol{A}_t z_t^{-1})^{-1}(\boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1})\\ &= \boldsymbol{C}(\boldsymbol{I}_n - \boldsymbol{A}_h z_1^{-1} - \boldsymbol{A}_v z_2^{-1} - \boldsymbol{A}_t z_t^{-1})^{-1}(\boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1})\end{aligned}$$

where $\boldsymbol{C} = \boldsymbol{N}_{HT}$.

In order to construct matrices $\boldsymbol{D}_{HT}$, $\boldsymbol{N}_{HT}$, $\boldsymbol{A}_h$, $\boldsymbol{A}_v$, $\boldsymbol{A}_t$, $\boldsymbol{B}_h$, $\boldsymbol{B}_v$ and $\boldsymbol{B}_t$ that satisfy the above properties, $\boldsymbol{\Psi}$ should have the following properties.

1. Entries in $\boldsymbol{\Psi}_j$ where $j \in \{1, 2...p\}$ should contain all the monomials in the j-th column of $\boldsymbol{F}(z_1, z_2, z_t)$. This condition guarantees that $\boldsymbol{D}_{HT}$ and $\boldsymbol{N}_{HT}$ always exist and satisfy conditions (3.28) and (3.29) respectively.

2. For every entry in $\boldsymbol{\Psi}_j$ say $\boldsymbol{\Psi}_j(i)$ which is not z_1, z_2 or z_t there should be another element in $\boldsymbol{\Psi}_j$ say $\boldsymbol{\Psi}_j(h)$ such that $\boldsymbol{\Psi}_j(i) = z_k^{-1}\boldsymbol{\Psi}_j(h)$, where $k \in \{1, 2, t\}$. As will be explained later this condition enables the construction of matrices $\boldsymbol{A}_h$, $\boldsymbol{A}_v$, $\boldsymbol{A}_t$, $\boldsymbol{B}_v$, $\boldsymbol{B}_v$ and $\boldsymbol{B}_t$ such that equation (3.30) is satisfied.

3. Vectors $\boldsymbol{\Psi}_j$ where $j \in \{1, 2...p\}$ should contain at least one of the terms z_1^{-1}, z_2^{-1}

and z_t^{-1}. This is necessitated by condition 2 above.

For $l \in \{1, 2...p\}$, matrices $\boldsymbol{A}_h^l \in \mathbb{R}^{n_l \times n_l}$, $\boldsymbol{A}_v^l \in \mathbb{R}^{n_l \times n_l}$, $\boldsymbol{A}_t^l \in \mathbb{R}^{n_l \times n_l}$, $\boldsymbol{B}_h^l \in \mathbb{R}^{n_l \times 1}$, $\boldsymbol{B}_v^l \in \mathbb{R}^{n_l \times 1}$ and $\boldsymbol{B}_t^l \in \mathbb{R}^{n_l \times 1}$ are constructed as follows.

I For $i \in \{1, 2...n_l\}$, $\boldsymbol{A}_t^l(i,j) = 1$ if $\exists\, j \in \{1, 2, ...n_l\}$ such that $\boldsymbol{\Psi}_l(i) = z_t^{-1}\boldsymbol{\Psi}_l(j)$. Otherwise $\boldsymbol{A}_t^l(i,j) = 0$.

II For $i \in \{1, 2...n_l\}$, $\boldsymbol{A}_v^l(i,j) = 1$ if $\boldsymbol{A}_t^l(i,k) = 0$ for $\forall\ k \in \{1, 2...n_l\}$ and $\exists\ j \in \{1, 2, ...n_l\}$ such that $\boldsymbol{\Psi}_l(i) = z_2^{-1}\boldsymbol{\Psi}_l(j)$. Otherwise $\boldsymbol{A}_v^l(i,j) = 0$.

III For $i \in \{1, 2...n_l\}$, $\boldsymbol{A}_h^l(i,j) = 1$ if $\boldsymbol{A}_t^l(i,k) = 0$ and $\boldsymbol{A}_v^l(i,k) = 0$ for $\forall\ k \in \{1, 2...n_l\}$ and $\exists\ j \in \{1, 2, ...n_l\}$ such that $\boldsymbol{\Psi}_l(i) = z_1^{-1}\boldsymbol{\Psi}_l(j)$. Otherwise $\boldsymbol{A}_h^l(i,j) = 0$.

IV $\boldsymbol{B}_t^l(i) = 1$ if $\boldsymbol{\Psi}_l(i) = z_t^{-1}$ and $\boldsymbol{B}_t^l(i) = 0$ otherwise.

V $\boldsymbol{B}_v^l(i) = 1$ if $\boldsymbol{\Psi}_l(i) = z_2^{-1}$ and $\boldsymbol{B}_v^l(i) = 0$ otherwise.

VI $\boldsymbol{B}_h^l(i) = 1$ if $\boldsymbol{\Psi}_l(i) = z_1^{-1}$ and $\boldsymbol{B}_h^l(i) = 0$ otherwise.

Let for $k \in \{h, v, t\}$:

VII $\bar{\boldsymbol{A}}_k = diag\{\boldsymbol{A}_k^1, \boldsymbol{A}_k^2, ..., \boldsymbol{A}_k^p\}$

VIII $\boldsymbol{B}_k = diag\{\boldsymbol{B}_k^1, \boldsymbol{B}_k^2, ..., \boldsymbol{B}_k^p\}$

It can be easily seen that:

$$(\boldsymbol{I}_n - \bar{\boldsymbol{A}}_h z_1^{-1} - \bar{\boldsymbol{A}}_v z_2^{-1} - \bar{\boldsymbol{A}}_t z_t^{-1})\boldsymbol{\Psi} = \boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1} \tag{3.31}$$

Let

$$\begin{aligned}\boldsymbol{A}_h =& \bar{\boldsymbol{A}}_h + \boldsymbol{B}_h \boldsymbol{D}_{HT} \\ \boldsymbol{A}_v =& \bar{\boldsymbol{A}}_v + \boldsymbol{B}_v \boldsymbol{D}_{HT} \\ \boldsymbol{A}_t =& \bar{\boldsymbol{A}}_t + \boldsymbol{B}_t \boldsymbol{D}_{HT}\end{aligned} \tag{3.32}$$

$$\begin{aligned}\{\boldsymbol{I}_n - \boldsymbol{A}_h z_1^{-1} - \boldsymbol{A}_v z_2^{-1} - \boldsymbol{A}_t z_t^{-1}\}\boldsymbol{\Psi} &= \{\boldsymbol{I}_n - \bar{\boldsymbol{A}}_h z_1^{-1} - \bar{\boldsymbol{A}}_v z_2^{-1} - \bar{\boldsymbol{A}}_t z_t^{-1}\}\boldsymbol{\Psi} \\ &\quad - \{\boldsymbol{B}_h \boldsymbol{D}_{HT} z_1^{-1} + \boldsymbol{B}_v \boldsymbol{D}_{HT} z_2^{-1} + \boldsymbol{B}_t \boldsymbol{D}_{HT} z_t^{-1}\}\boldsymbol{\Psi} \\ &= \boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1} \\ &\quad - (\boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1})\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) \\ &= (\boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1})(\boldsymbol{I}_p - \tilde{\boldsymbol{D}}_R(z_1, z_2, z_t)) \\ &= (\boldsymbol{B}_h z_1^{-1} + \boldsymbol{B}_v z_2^{-1} + \boldsymbol{B}_t z_t^{-1})\boldsymbol{D}_R(z_1, z_2, z_t)\end{aligned}$$

Matrices $\boldsymbol{A}_h$, $\boldsymbol{A}_v$, $\boldsymbol{A}_t$, $\boldsymbol{B}_h$, $\boldsymbol{B}_v$ and $\boldsymbol{B}_t$ satisfy condition (3.30). Therefore matrices $\boldsymbol{A}_h$, $\boldsymbol{A}_v$, $\boldsymbol{A}_t$, $\boldsymbol{B}_h$, $\boldsymbol{B}_v$, $\boldsymbol{B}_t$ and $\boldsymbol{C}$ realize the transfer matrix (3.21). In addition to realizing the transfer matrix (3.21), all system matrices are required to satisfy condition (2.4).

Let $\boldsymbol{\Psi}(i, j) = z_1^{-1}$. There is no monomial z_1^{-1} in any of the polynomial elements of $\boldsymbol{D}_R(z_1, z_2, z_t)$ due to (3.23). Therefore $\boldsymbol{D}_{HT}(h, i) = 0, \forall\, h \in \{1, 2, ..., p\}$. Hence:

$$\boldsymbol{D}_{HT}\boldsymbol{B}_h = \boldsymbol{0} \tag{3.33}$$

Similarly:

$$\boldsymbol{D}_{HT}\boldsymbol{B}_v = \boldsymbol{0} \tag{3.34}$$

Let $\boldsymbol{\Psi}$ be constructed such that, it doesn't contain monomials z_1^{-2}, $z_1^{-1}z_2^{-1}$ and z_2^{-2}. Due to (3.22) and (3.23) monomials z_1^{-2}, $z_1^{-1}z_2^{-1}$ and z_2^{-2} are not required by $\boldsymbol{\Psi}$ to satisfy (3.28) and (3.29). There is no element in $\boldsymbol{\Psi}$ such that $\boldsymbol{\Psi}(h,j) = z_1^{-1}\boldsymbol{\Psi}(i,j) = z_1^{-2}$. Therefore $\bar{\boldsymbol{A}}_h(h,i) = 0$ for $\forall\ h \in \{1,2,...,n\}$. Hence:

$$\bar{\boldsymbol{A}}_h \boldsymbol{B}_h = \boldsymbol{0} \tag{3.35}$$

Since there are no z_2^{-2} or $z_1^{-1}z_2^{-1}$ terms in $\boldsymbol{\Psi}$, using a similar argument:

$$\bar{\boldsymbol{A}}_h \boldsymbol{B}_v = \bar{\boldsymbol{A}}_v \boldsymbol{B}_h = \bar{\boldsymbol{A}}_v \boldsymbol{B}_v = \boldsymbol{0} \tag{3.36}$$

Due to (3.33), (3.34), (3.35) and (3.36) we can deduce that:

$$\boldsymbol{A}_v \boldsymbol{B}_v = \boldsymbol{A}_v \boldsymbol{B}_h = \boldsymbol{A}_h \boldsymbol{B}_v = \boldsymbol{A}_h \boldsymbol{B}_h = \boldsymbol{0} \tag{3.37}$$

Let $\exists\ \boldsymbol{\Psi}(h,j)$ such that $\boldsymbol{\Psi}(i,j) = z_t^{-1}\boldsymbol{\Psi}(h,j)$ when $\boldsymbol{\Psi}(i,j)$ is a monomial in the j-th column of $\tilde{\boldsymbol{D}}_R(z_1,z_2,z_t)$ and $\boldsymbol{\Psi}(i,j) \neq z_t^{-1}$. Let $\boldsymbol{D}_{HT}(l,i) \neq 0$ for some $l \in \{1,2,...,p\}$. By construction, there can be one and only one non-zero element in any row of $\boldsymbol{\Psi}$. Let $\boldsymbol{\Psi}(i,j)$ be the non-zero element in the i-th row. Due to (3.28) $\boldsymbol{\Psi}(i,j)$ is in the j-th column of $\tilde{\boldsymbol{D}}_R(z_1,z_2,z_t)$. If $\boldsymbol{\Psi}(i,j) \neq z_t^{-1}$, since $\boldsymbol{A}_t$ is constructed according to steps I and VII above, $\boldsymbol{A}_t(i,h) = 1$ for some $h \in \{1,2,...,n\}$. Therefore, by construction of $\bar{\boldsymbol{A}}_h$ and $\bar{\boldsymbol{A}}_v$(given by steps II and III and VII), $\bar{\boldsymbol{A}}_h(i,k) = 0$ and $\bar{\boldsymbol{A}}_v(i,k) = 0$ for $k \in \{1,2,...,n\}$, if $\boldsymbol{D}_{HT}(l,i) \neq 0$ for some $l \in \{1,2,...,p\}$. Hence:

$$\boldsymbol{D}_{HT}\bar{\boldsymbol{A}}_h = \boldsymbol{D}_{HT}\bar{\boldsymbol{A}}_v = \boldsymbol{0} \tag{3.38}$$

Assume that, if $\boldsymbol{\Psi}(h,j)$ is not equal to z_1^{-1}, z_2^{-1} or z_t^{-1} and satisfy $\boldsymbol{\Psi}(i,j) = z_1^{-1}\boldsymbol{\Psi}(h,j)$

or $\boldsymbol{\Psi}(i,j) = z_2^{-1}\boldsymbol{\Psi}(h,j)$, then $\exists\ \boldsymbol{\Psi}(k,j)$ such that $\boldsymbol{\Psi}(h,j) = z_t^{-1}\boldsymbol{\Psi}(k,j)$. Let $\bar{\boldsymbol{A}}_h(i,h) = 1$. Then by construction of $\bar{\boldsymbol{A}}_h$, $\boldsymbol{\Psi}(i,j) = z_1^{-1}\boldsymbol{\Psi}(h,j)$ for some $j \in \{1,2,...,n\}$. Hence, either $\bar{\boldsymbol{A}}_t(h,k) = 1$ for some $k \in \{1,2,...,n\}$ or $\boldsymbol{\Psi}(h,j) \in \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$. Therefore, by construction of $\bar{\boldsymbol{A}}_h$ and $\bar{\boldsymbol{A}}_v$, if $\bar{\boldsymbol{A}}_h(i,h) = 1$, then $\bar{\boldsymbol{A}}_h(h,k) = 0$ and $\bar{\boldsymbol{A}}_v(h,k) = 0$ for $k \in \{1,2,...,n\}$. Similarly if $\bar{\boldsymbol{A}}_v(i,h) = 1$, then $\bar{\boldsymbol{A}}_h(h,k) = 0$ and $\bar{\boldsymbol{A}}_v(h,k) = 0$ for $k \in \{1,2,...,n\}$. Therefore:

$$\bar{\boldsymbol{A}}_h\bar{\boldsymbol{A}}_h = \bar{\boldsymbol{A}}_h\bar{\boldsymbol{A}}_v = \bar{\boldsymbol{A}}_v\bar{\boldsymbol{A}}_h = \bar{\boldsymbol{A}}_v\bar{\boldsymbol{A}}_v = \mathbf{0} \tag{3.39}$$

From (3.37), (3.38) and (3.39), we can deduce that condition (3.40) is satisfied.

$$\begin{aligned} \boldsymbol{A}_v\boldsymbol{A}_v = \boldsymbol{A}_v\boldsymbol{A}_h = \boldsymbol{A}_h\boldsymbol{A}_v = \boldsymbol{A}_h\boldsymbol{A}_h = \mathbf{0} \\ \boldsymbol{A}_v\boldsymbol{B}_v = \boldsymbol{A}_v\boldsymbol{B}_h = \boldsymbol{A}_h\boldsymbol{B}_v = \boldsymbol{A}_h\boldsymbol{B}_h = \mathbf{0} \end{aligned} \tag{3.40}$$

Therefore, in addition to conditions 1, 2 and 3, if matrix $\boldsymbol{\Psi}$ satisfies the following conditions, the transfer matrix (3.21) can be realized in an FM model satisfying condition (2.4).

4 Monomials z_1^{-2}, $z_1^{-1}z_2^{-1}$ and z_2^{-2} are not contained in $\boldsymbol{\Psi}$.

5 When $\boldsymbol{\Psi}(i,j)$ is a monomial in $\tilde{\boldsymbol{D}}_R(z_1,z_2,z_t)$ and $\boldsymbol{\Psi}(i,j) \neq z_t^{-1}$, $\exists\ \boldsymbol{\Psi}(h,j)$ such that $\boldsymbol{\Psi}(i,j) = z_t^{-1}\boldsymbol{\Psi}(h,j)$.

6 For $\boldsymbol{\Psi}(h,j)$ that is not equal to z_1^{-1}, z_2^{-1} or z_t^{-1} and satisfies $\boldsymbol{\Psi}(i,j) = z_1^{-1}\boldsymbol{\Psi}(h,j)$, there exist a $\boldsymbol{\Psi}(k,j)$ such that $\boldsymbol{\Psi}(h,j) = z_t^{-1}\boldsymbol{\Psi}(k,j)$.

7 For $\boldsymbol{\Psi}(h,j)$ not equal to z_1^{-1}, z_2^{-1} or z_t^{-1} satisfying $\boldsymbol{\Psi}(i,j) = z_2^{-1}\boldsymbol{\Psi}(h,j)$ there exist a $\boldsymbol{\Psi}(k,j)$ such that $\boldsymbol{\Psi}(h,j) = z_t^{-1}\boldsymbol{\Psi}(k,j)$.

1: **for** $j = 1$ to p **do**
2: Collect all the monomials in the j-th column of $\boldsymbol{F}(z_1, z_2, z_t)$ into $\boldsymbol{\Phi}_j$ in the ascending total degree lexicographic order .
3: $\boldsymbol{\Psi}_j = \boldsymbol{\Phi}_j$
4: **for** $i = 1$ to $Number\ of\ elements\ in\ \boldsymbol{\Phi}_j$ **do**
5: $K = \boldsymbol{\Phi}_j(i)$
6: **if** $orderz_1(K) + orderz_2(K) \leq orderz_t(K)$ **then**
7: **while** $orderz_1(K) + orderz_2(K) \leq orderz_t(K)$ and $K \notin \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$ **do**
8: **if** z_tK is not in $\boldsymbol{\Psi}_j$ **then**
9: Insert z_tK to $\boldsymbol{\Psi}_j$ in the ascending total degree lexicographic order.
10: **end if**
11: $K = z_tK$
12: **end while**
13: **end if**
14: **while** $K \notin \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$ **do**
15: **if** There is at least one power of z_1^{-1} in K **then**
16: **if** z_1K is not in $\boldsymbol{\Psi}_j$ **then**
17: Insert z_1K to $\boldsymbol{\Psi}_j$ in the ascending total degree lexicographic order.
18: **end if**
19: $K = z_1K$
20: **if** There is at least one power of z_t^{-1} in K and $K \neq z_t^{-1}$ **then**
21: **if** $z_t^{-1}K$ is not in $\boldsymbol{\Psi}_j$ **then**
22: Insert z_tK to $\boldsymbol{\Psi}_j$ in the ascending total degree lexicographic order.
23: **end if**
24: $K = z_tK$
25: **end if**
26: **else**
27: **if** There is at least one power of z_2^{-1} in K and $K \neq z_2^{-1}$ **then**
28: **if** z_2K is not in $\boldsymbol{\Psi}_j$ **then**
29: Insert z_2K to $\boldsymbol{\Psi}_j$ in the ascending total degree lexicographic order.
30: **end if**
31: $K = z_2K$
32: **end if**

Figure 3.2. Algorithm for constructing $\boldsymbol{\Psi}$

33: **if** There is at least one power of z_t^{-1} in K and $K \neq z_t^{-1}$ **then**
34: **if** $z_t K$ is not in $\mathbf{\Psi}_j$ **then**
35: Insert $z_t K$ to $\mathbf{\Psi}_j$ in the ascending total degree lexicographic order.
36: **end if**
37: $K = z_t K$
38: **end if**
39: **end if**
40: **end while**
41: **end for**
42: **end for**
43: $\mathbf{\Psi} = diag\{\mathbf{\Psi}_1, \mathbf{\Psi}_1, ..., \mathbf{\Psi}_p\}$

Figure 3.2. continued

The algorithm given in figure 3.2 can be used to construct the matrix $\mathbf{\Psi}$ with the above properties.

Lines 2 and 3 of the algorithm ensure that all the monomials in $\boldsymbol{F}(z_1, z_2, z_t)$ are contained in $\mathbf{\Psi}$. Lines 6-13 ensure that $\mathbf{\Psi}$ satisfies condition 5. The monomials missing in $\mathbf{\Psi}$ that are needed to satisfy conditions 2, 6 and 7 are inserted in the loop starting from line 14. Due to conditions (3.22) and (3.23), monomials $z_1^{-i} z_2^{-j} z_t^{-k}$ in vectors $\mathbf{\Phi}_l$ satisfy $i + j \leq k + 1$ for $l \in \{1, 2, ...p\}$. Moreover, if monomial $z_1^{-i} z_2^{-j} z_t^{-k}$ satisfy $i + j = k + 1$, the algorithm starts inserting monomials to $\mathbf{\Psi}_l$ by inserting $z_1^{-i+1} z_2^{-j} z_t^{-k}$ or $z_1^{-i} z_2^{-j+1} z_t^{-k}$. Therefore, for every monomial $z_1 K$ inserted into $\mathbf{\Psi}_l$, $z_t z_1 K$ is inserted into $\mathbf{\Psi}_l$ unless $z_1 K \in \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$. For every monomial $z_2 K$ inserted to $\mathbf{\Psi}_l$, $z_t z_2 K$ is inserted to $\mathbf{\Psi}_l$ unless $z_2 K \in \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$. Therefore conditions 6 and 7 are satisfied by $\mathbf{\Psi}$. It is easily seen that, the matrix $\mathbf{\Psi}$ constructed by the algorithm satisfies condition 4 if conditions (3.22) and (3.23) hold. This completes the proof of the theorem.

For the sake of simplicity, implications of the theorem are discussed for the case of single-input single-output systems. In this case, the right matrix fraction description of the system given by (3.21), reduces to a rational transfer function. Let the impulse response of the system described by the transfer function (3.21) be $h(n_1, n_2, t)$. If conditions (3.22) and (3.23) are satisfied by the transfer function, then we have:

$$h(n_1, n_2, t) = 0 \ \ for \ \ n_1 + n_2 > t + 1 \tag{3.41}$$

Therefore an impulse response, realizable in an FM model (2.2) which satisfies condition (2.4), satisfies the condition (3.41). The FM model (2.2) (satisfying condition (2.4)) is implementable in a grid sensor network in which information is conveyed over a single hop in one time slot. The effect of an input at a node propagates over a sin-

gle hop in a single time slot. Let a first octant causal impulse response that satisfies the condition (3.41) have a rational Z-transform $\boldsymbol{H}(z_1, z_2, z_t)$. It can easily be shown that $\boldsymbol{H}(z_1, z_2, z_t)$ satisfies conditions (3.22) and (3.23). Hence any first octant causal impulse response satisfying (3.41) with a rational Z-transform is realizable in an FM model (2.2) that satisfies condition (2.4).

3.2.2 Summary of the Realization Algorithm

The algorithm to derive an FM model, that satisfies condition (2.4), given the right matrix fraction description (3.21) of an admissible transfer function, can be summarized as follows.

- Compute $\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t)$ using $\tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) = \boldsymbol{I}_p - \boldsymbol{D}_R(z_1, z_2, z_t)$
- Compute $\boldsymbol{F}(z_1, z_2, z_t)$ using:

$$\boldsymbol{F}(z_1, z_2, z_t) = \begin{bmatrix} \boldsymbol{N}_R(z_1, z_2, z_t) \\ \tilde{\boldsymbol{D}}_R(z_1, z_2, z_t) \end{bmatrix}$$

- Construct the matrix $\boldsymbol{\Psi}$ using the algorithm given in figure 3.2.
- Derive the matrices $\boldsymbol{N}_{HT}$ and $\boldsymbol{D}_{HT}$ that satisfies (3.28) and (3.29)
- Using the steps I-VIII derive matrices $\bar{\boldsymbol{A}}_h$, $\bar{\boldsymbol{A}}_v$, $\bar{\boldsymbol{A}}_t$, $\boldsymbol{B}_h$, $\boldsymbol{B}_v$, and $\boldsymbol{B}_t$
- Compute $\boldsymbol{A}_h$, $\boldsymbol{A}_v$ and $\boldsymbol{A}_t$ using (3.32).
- $\boldsymbol{C} = \boldsymbol{N}_{HT}$

3.2.3 Non Causal Transfer Matrices

Systems described by the FM model (2.2) are necessarily first octant causal. Systems causal in any of the four quadrants of the spatial plane can be realized as:

$$\begin{aligned}\boldsymbol{x}(n_1, n_2, t) &= \boldsymbol{A}_h\boldsymbol{x}(n_1\text{–}\alpha, n_2, t) + \boldsymbol{A}_v\boldsymbol{x}(n_1, n_2\text{–}\beta, t) + \boldsymbol{A}_t\boldsymbol{x}(n_1, n_2, t\text{–}1) \\ &\quad + \boldsymbol{B}_h\boldsymbol{u}(n_1\text{–}\alpha, n_2, t) + \boldsymbol{B}_v\boldsymbol{u}(n_1, n_2\text{–}\beta, t) + \boldsymbol{B}_t\boldsymbol{u}(n_1, n_2, t\text{–}1) \\ \boldsymbol{y}(n_1, n_2, t) &= \boldsymbol{C}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t) \end{aligned} \tag{3.42}$$

by appropriately choosing α and β. Here $(\alpha, \beta) \in \{-1, 1\} \times \{-1, 1\}$. For example when $(\alpha, \beta) = (-1, -1)$ the region of support of the impulse response lies in the 3^{rd} quadrant of the spatial plane. The following lemma which is a generalization of the Theorem 3.2.1 can be used to derive the final result of this paper.

Lemma 3.2.1 *Let an input-output transfer matrix be given by the right matrix fraction description:*

$$\boldsymbol{H}_{\alpha\beta}(z_1, z_2, z_t) = \boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)\boldsymbol{D}^{-1}_{\alpha\beta}(z_1, z_2, z_t) \tag{3.43}$$

where $(\alpha, \beta) \in \{-1, 1\} \times \{-1, 1\}$ *and where matrices* $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ *and* $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ *are of size* $q \times p$ *and* $p \times p$ *respectively. Let* (l, m)*-th elements of matrices* $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ *and* $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ *be denoted by* $\boldsymbol{N}_{\alpha\beta}(l, m)$ *and* $\boldsymbol{D}_{\alpha\beta}(l, m)$ *respectively. Let:*

$$\begin{aligned}\boldsymbol{N}_{\alpha\beta}(l, m) &= \sum_{i=0}^{N_{\alpha\beta1}} \sum_{j=0}^{N_{\alpha\beta2}} \sum_{k=0}^{N_{\alpha\beta t}} n^{lm}_{ijk(\alpha\beta)} z_1^{-\alpha i} z_2^{-\beta j} z_t^{-k} \\ \boldsymbol{D}_{\alpha\beta}(l, m) &= \sum_{i=0}^{D_{\alpha\beta1}} \sum_{j=0}^{D_{\alpha\beta2}} \sum_{k=0}^{D_{\alpha\beta t}} d^{lm}_{ijk(\alpha\beta)} z_1^{-\alpha i} z_2^{-\beta j} z_t^{-k} \end{aligned}$$

where $N_{\alpha\beta1}$*,* $N_{\alpha\beta2}$ *and* $N_{\alpha\beta t}$ *are the degrees of the polynomial matrix* $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ *with respect to variables* z_1^{-1}*,* z_2^{-1} *and* z_t^{-1} *respectively. Furthermore* $D_{\alpha\beta1}$*,* $D_{\alpha\beta2}$ *and*

$D_{\alpha\beta t}$ are the degrees of the polynomial matrix $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ with respect to variables z_1^{-1}, z_2^{-1} and z_t^{-1} respectively.

The transfer matrix (3.43) can be realized in an FM model of the form (3.42) while satisfying condition (2.4), if and only if the conditions (3.44) and (3.45) are satisfied.

$$n^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i + j > k + 1 \tag{3.44}$$

$$d^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i + j \geq k + 1 \tag{3.45}$$

Proof For the case of $(\alpha, \beta) = (1, 1)$ the lemma is a restatement of Theorem 3.2.1. For other cases it can be proven using an identical approach to the proof of Theorem 3.2.1.

A parallel combination of systems causal in each quadrant can be used to realize non causal systems Lele and Mendel [1987].

Theorem 3.2.2 *A transfer matrix $\boldsymbol{H}(z_1, z_2, z_t)$ can be realized as a parallel combination of FM models of the form (3.42) while satisfying condition (2.4), if and only if it can be expressed as:*

$$\boldsymbol{H}(z_1, z_2, z_t) = \sum_{(\alpha,\beta)\in\{-1,1\}\times\{-1,1\}} \boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)\boldsymbol{D}^{-1}_{\alpha\beta}(z_1, z_2, z_t) \tag{3.46}$$

where elements of matrices $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ and $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ are polynomials of $z_1^{-\alpha}$, $z_2^{-\beta}$ and z_{-t} satisfying the following conditions: If the (l, m)-th elements of $\boldsymbol{N}_{\alpha\beta}(z_1, z_2, z_t)$ and $\boldsymbol{D}_{\alpha\beta}(z_1, z_2, z_t)$ are given by:

$$\boldsymbol{N}_{\alpha\beta}(l, m) = \sum_{i=0}^{N_{\alpha\beta1}} \sum_{j=0}^{N_{\alpha\beta2}} \sum_{k=0}^{N_{\alpha\beta t}} n^{lm}_{ijk(\alpha\beta)} z_1^{-i\alpha} z_2^{-j\beta} z_t^{-k}$$

$$\boldsymbol{D}_{\alpha\beta}(l, m) = \sum_{i=0}^{D_{\alpha\beta1}} \sum_{j=0}^{N_{\alpha\beta2}} \sum_{k=0}^{N_{\alpha\beta t}} d^{lm}_{ijk(\alpha\beta)} z_1^{-i\alpha} z_2^{-j\beta} z_t^{-k}$$

then

$$n^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i+j > k+1 \tag{3.47}$$

$$d^{lm}_{ijk(\alpha\beta)} = 0 \;\; for \;\; i+j \geq k+1 \tag{3.48}$$

Proof The result trivially follows from Lemma 3.2.1.

3.2.4 Example

The realization algorithm discussed in the above section is illustrated by an example in this section. Let the transfer matrix to be realized be as follows:

$$\begin{aligned}\boldsymbol{H}(z_1, z_2, z_t) = &\begin{bmatrix} n^{11}_{100}z_1^{-1} + n^{11}_{010}z_2^{-1} + n^{11}_{101}z_1^{-1}z_t^{-1} & n^{12}_{010}z_2^{-1} \\ n^{21}_{001}z_t^{-1} & n^{22}_{101}z_1^{-1}z_t^{-1} \end{bmatrix} \\ &\times \begin{bmatrix} 1+ d^{11}_{011}z_2^{-1}z_t^{-1}+ d^{11}_{001}z_t^{-1}+ d^{11}_{112}z_1^{-1}z_2^{-1}z_t^{-2} & d^{12}_{001}z_t^{-1} \\ d^{21}_{011}z_2^{-1}z_t^{-1} & 1 + d^{22}_{101}z_1^{-1}z_t^{-1} \end{bmatrix}^{-1}\end{aligned} \tag{3.49}$$

The transfer function (3.49) satisfies conditions (3.22) and (3.23). Therefore, according to Theorem 3.2.1 the above transfer matrix is realizable in the FM model (2.2) with condition (2.4) satisfied. Matrix $\boldsymbol{F}(z_1, z_2, z_t)$ is given by:

$$\boldsymbol{F}(z_1, z_2, z_t) = \begin{bmatrix} n^{11}_{100}z_1^{-1} + n^{11}_{010}z_2^{-1} + n^{11}_{101}z_1^{-1}z_t^{-1} & n^{12}_{010}z_2^{-1} \\ n^{21}_{001}z_t^{-1} & n^{22}_{101}z_1^{-1}z_t^{-1} \\ -d^{11}_{011}z_2^{-1}z_t^{-1}- d^{11}_{001}z_t^{-1}- d^{11}_{112}z_1^{-1}z_2^{-1}z_t^{-2} & -d^{12}_{001}z_t^{-1} \\ -d^{21}_{011}z_2^{-1}z_t^{-1} & -d^{22}_{101}z_1^{-1}z_t^{-1} \end{bmatrix}$$

Since the system has two inputs, the loop starting at line 1 of the algorithm given in figure 3.2 runs twice. The sequence of operations in the first iteration is given in table 3.2.

TABLE 3.2:

THE SEQUENCE OF OPERATIONS IN THE FIRST ITERATION OF THE ALGORITHM

step	operation
2	$\mathbf{\Phi}_1 = [z_1^{-1} \;\; z_2^{-1} \;\; z_t^{-1} \;\; z_1^{-1}z_t^{-1} \;\; z_2^{-1}z_t^{-1} \;\; z_1^{-1}z_2^{-1}z_t^{-2}]^T$
3	$\mathbf{\Psi}_1 = \mathbf{\Phi}_1$
4	$i = 1$ This is the first iteration of the **for** loop starting at step 3.
5	$K = z_1^{-1}$ No elements are inserted to $\mathbf{\Psi}_1$ in this iteration.
4	$i = 2$ This is the second iteration of the **for** loop starting at step 3.
5	$K = z_2^{-1}$ No elements are inserted to $\mathbf{\Psi}_1$ in this iteration.
4	$i = 3$ This is the third iteration of the **for** loop starting at step 3.
5	$K = z_t^{-1}$ No elements are inserted to $\mathbf{\Psi}_1$ in this iteration.
4	$i = 4$ This is the fourth iteration of the **for** loop starting at step 3.
5	$K = z_1^{-1}z_t^{-1}$

TABLE 3.2 continued

6	$orderz_1(K) + orderz_2(K) = orderz_t(K)$
7	$K \notin \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$
8	$z_tK = z_1$ is in $\mathbf{\Psi}_1$
11	$K = z_tK$ Since $K = z_1^{-1}$ the iteration ends
4	$i = 5$ This is the fifth iteration of the **for** loop starting at step 3.
5	$K = z_2^{-1}z_t^{-1}$
6	$orderz_1(K) + orderz_2(K) = orderz_t(K)$
7	$K \notin \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$
8	$z_tK = z_2^{-1}$ is in $\mathbf{\Psi}_1$
11	$K = z_tK$ Since $K = z_2^{-1}$ the iteration ends
4	$i = 6$ This is the sixth iteration of the **for** loop starting at step 3.
5	$K = z_1^{-1}z_2^{-1}z_t^{-2}$
6	$orderz_1(K) + orderz_2(K) = orderz_t(K)$
7	$K \notin \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$
8	$z_tK = z_1^{-1}z_2^{-1}z_t^{-1}$ is not in $\mathbf{\Psi}_1$
9	Insert $z_1^{-1}z_2^{-1}z_t^{-1}$ into $\mathbf{\Psi}_1$
11	$K = z_tK$ Now $K = z_1^{-1}z_2^{-1}z_t^{-1}$
7	$orderz_1(K) + orderz_2(K) > orderz_t(K)$
14	$K \notin \{z_1^{-1}, z_2^{-1}, z_t^{-1}\}$
15	There is one power of z_1^{-1} in K
16	$z_1K = z_2^{-1}z_t^{-1}$ is in $\mathbf{\Psi}_1$
19	$K = z_1K$ Now $K = z_2^{-1}z_t^{-1}$

TABLE 3.2 continued

20	There is one power of z_t^{-1} in K and $K \neq z_t$
21	$z_t K = z_2^{-1}$ is in $\mathbf{\Psi}_1$
24	$K = z_t K$ Since $K = z_2^{-1}$ the iteration ends

At the end of the first iteration (of the loop starting at line 1 of the algorithm) we obtain $\mathbf{\Psi}_1$. Vector $\mathbf{\Psi}_2$ can be derived in the second iteration which is omitted for the sake of brevity. Using $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$, $\mathbf{\Psi}$ can be constructed as:

$$\mathbf{\Psi} = \begin{bmatrix} z_1^{-1} & z_2^{-1} & z_t^{-1} & z_1^{-1}z_t^{-1} & z_2^{-1}z_t^{-1} & z_1^{-1}z_2^{-1}z_t^{-1} & z_1^{-1}z_2^{-1}z_t^{-2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & z_1^{-1} & z_2^{-1} & z_t^{-1} & z_1^{-1}z_t^{-1} \end{bmatrix}^T$$

Matrices $\boldsymbol{D}_{HT}$ and $\boldsymbol{N}_{HT}$can be derived using (3.28) and (3.29) as:

$$\boldsymbol{D}_{HT} = \begin{bmatrix} 0 & 0 & -d_{001}^{11} & 0 & -d_{011}^{11} & 0 & -d_{112}^{11} & 0 & 0 & -d_{001}^{12} & 0 \\ 0 & 0 & 0 & 0 & -d_{011}^{21} & 0 & 0 & 0 & 0 & 0 & -d_{101}^{22} \end{bmatrix}$$

$$\boldsymbol{N}_{HT} = \begin{bmatrix} n_{100}^{11} & n_{010}^{11} & 0 & n_{101}^{11} & 0 & 0 & 0 & 0 & n_{010}^{12} & 0 & 0 \\ 0 & 0 & n_{001}^{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_{101}^{22} \end{bmatrix}$$

Matrix $\boldsymbol{C} = \boldsymbol{N}_{HT}$. Matrices $\boldsymbol{A}_h$, $\boldsymbol{A}_v$, $\boldsymbol{A}_t$, $\boldsymbol{B}_h$, $\boldsymbol{B}_v$ and $\boldsymbol{B}_t$ can be derived using the procedure given in section 3.2.1.

$$\boldsymbol{B}_h = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\boldsymbol{B}_v = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$$

$$\boldsymbol{B}_t = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$\boldsymbol{A}_h = \begin{bmatrix}
0 & 0 & -d_{001}^{11} & 0 & -d_{011}^{11} & 0 & -d_{112}^{11} & 0 & 0 & -d_{001}^{12} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -d_{011}^{21} & 0 & 0 & 0 & 0 & 0 & -d_{101}^{22} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\boldsymbol{A}_v = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -d_{001}^{11} & 0 & -d_{011}^{11} & 0 & -d_{112}^{11} & 0 & 0 & -d_{001}^{12} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -d_{011}^{21} & 0 & 0 & 0 & 0 & 0 & -d_{101}^{22} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\boldsymbol{A}_t = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -d_{001}^{11} & 0 & -d_{011}^{11} & 0 & -d_{112}^{11} & 0 & 0 & -d_{001}^{12} & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -d_{011}^{21} & 0 & 0 & 0 & 0 & 0 & -d_{101}^{22} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

The state vector $\boldsymbol{x} \in \mathbb{R}^{11}$ in this realization.

3.2.5 Comparison with GR Model Based Implementations

A causal transfer matrix of the form (3.21) is realizable in the constrained GR model if and only if conditions (3.22) and (3.23) are satisfied. Therefore a causal transfer matrix realizable in the constrained GR model is also realizable in the constrained FM model and vice versa. Similarly a transfer matrix realizable as a parallel combination of constrained GR models that are causal in each quadrant is also realizable as a parallel combination of constrained FM models causal in each quadrant.The converse is also true.

CHAPTER 4

STABILITY UNDER FINITE PRECISION ARITHMETIC

Systems (2.1) and (2.2) are implemented on wireless sensor networks using finite precision number representation schemes. Shorter word lengths may be used for numbers communicated between nodes due to bandwidth and power restrictions. The effect of quantization and finite precision arithmetic on system dynamics is modeled in this chapter. Internal and external stability of distributed systems implemented using finite precision arithmetic is analyzed. Emphasis is given to the effect of inter node communication on system stability, since short word length formats are an important means of reducing communication bandwidth and energy requirements. Floating point and fixed point computations result in different system models for the distributed systems. Therefore the cases of floating point and fixed point computation are treated separately in this chapter.

4.1 Fixed Point Arithmetic

4.1.1 Fixed Point Quantization and Overflow

Quantization and overflow operators widely used in fixed point arithmetic are discussed in the following. Let $\mathbb{S}$ be the set of numbers representable with the number representation scheme used. Quantization and overflow operators employed are denoted by Q and O respectively. If the difference between any two consecutive elements

in $\mathbb{S}$ is either q or $-q$, the quantization operator is said to be uniform and q is called the quantization step size. Widely used fixed point quantization schemes are discussed in the following paragraphs. It is assumed that the quantization is uniform with a step size of q. Input versus output plots for the quantization schemes shown in figures 4.1, 4.2 and 4.3 are for the case where $q = 1$.

Magnitude Truncation This scheme rounds the input to the closest representable quantization level in the direction of zero as shown in Figure 4.1. The quantization error is less than q in this scheme .

Rounding This scheme rounds the output value to the nearest quantization level as shown in figure 4.2. In the case of a tie, it rounds positive numbers to the closest quantization level in the direction of positive infinity, and negative numbers to the closest quantization level in the direction of negative infinity. The maximum quantization error that occurs under this scheme is $q/2$.

Two's Complement This scheme rounds the input to the closest representable quantization level in the direction of negative infinity as shown in Figure 4.3. The maximum quantization error is less than q in this scheme.

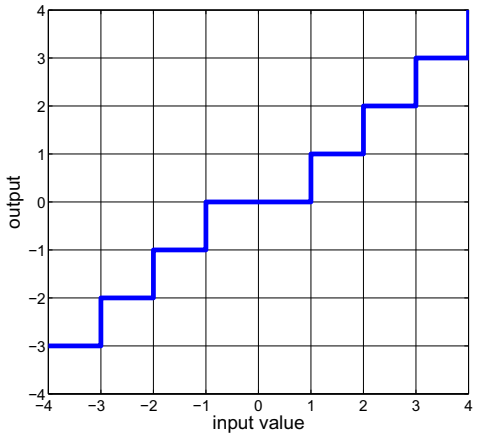


Figure 4.1. Fixed-point quantization schemes: Magnitude Truncation

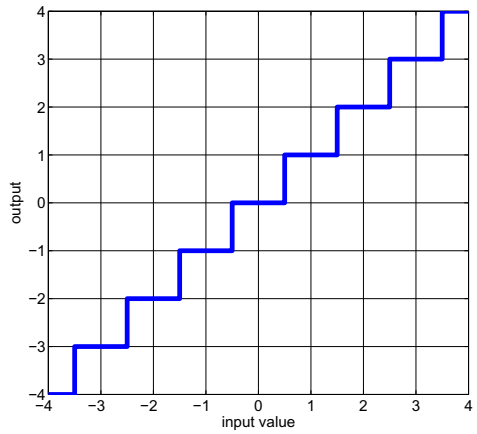


Figure 4.2. Fixed-point quantization schemes: Rounding

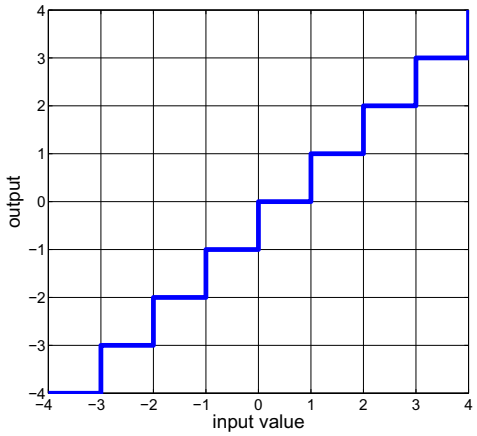


Figure 4.3. Fixed-point quantization schemes: Two's complement

Overflow occurs when a result of an arithmetic operation is either larger than the largest representable number or smaller than the smallest representable number of the number format used. Input-output behavior of the overflow nonlinearity depends on how the overflow is handled. Models pertaining to commonly used methods to handle overflow are discussed below. Input output curves for the overflow nonlinearities discussed are shown in figures 4.4, 4.5 and 4.6. for the case where the largest and smallest representable numbers are 4 and -4 respectively.

Saturation nonlinearity When an overflow occurs in an arithmetic operation, the closest representable number to the outcome of the arithmetic operation may be used instead of it. The resulting nonlinearity is illustrated in figure 4.4 and is called the saturation nonlinearity.

Wraparound (2's complement overflow) nonlinearity Input-output behavior of the wraparound nonlinearity is illustrated in figure 4.5. When two's complement

arithmetic is used, overflow of arithmetic operations result in the wraparound nonlinearity.

Zeroing nonlinearity Nonlinearity resulting from replacing the outcome of an arithmetic operation by zero when an overflow occurs is the zeroing nonlinearity. This nonlinearity is illustrated in figure 4.6.

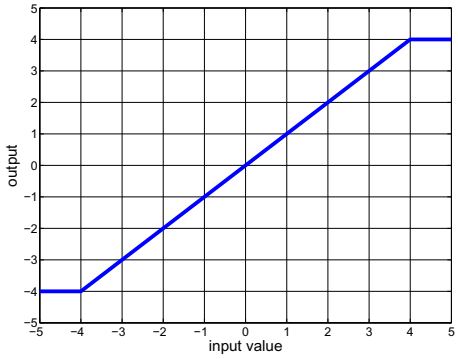


Figure 4.4. Fixed-point overflow nonlinearities: Saturation Nonlinearity

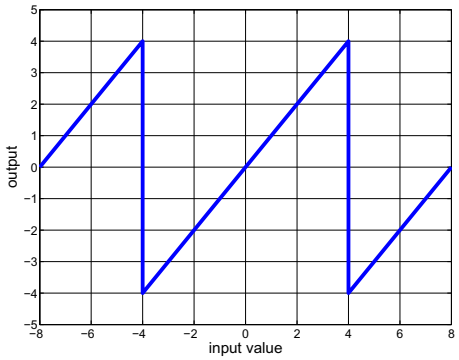


Figure 4.5. Fixed-point overflow nonlinearities: Wraparound Nonlinearity

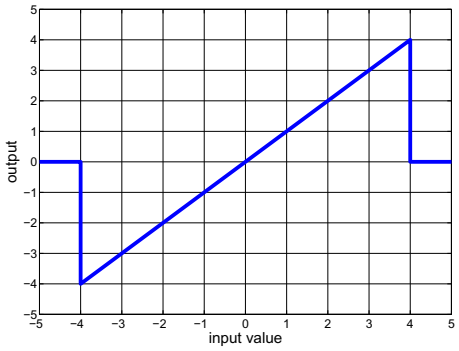


Figure 4.6. Fixed-point overflow nonlinearities: Zeroing Nonlinearity

4.1.2 Models for Quantization and Overflow Nonlinearities

Microprocessors used in most wireless sensor platforms use fixed point number representation schemes. Bandwidth and power restrictions may demand even shorter word

lengths for numbers communicated between nodes. So the error resulting from finite precision arithmetic could be significant. Finite word lengths, used for the representation of numbers, can result in errors due to three reasons:

- Coefficients of matrices must be quantized to numbers that are representable using the number format used.
- Results of arithmetic operations must be quantized.
- Overflow can occur during arithmetic operations.

Only the effects of quantization and overflow nonlinearities are considered in this work. For notational convenience the net effect of quantization and overflow nonlinearities will be modeled by a single nonlinear operator. The nonlinearity resulting from quantization and overflow can be considered as a concatenation of quantization and overflow nonlinearities.

Let $O : \mathbb{R} \rightarrow [L_{min},\ L_{max}]$ denote the overflow nonlinearity where L_{min} and L_{max} denote the smallest and the largest numbers representable with the number format used respectively. The quantization operator is denoted by $Q : [L_{min},\ L_{max}] \rightarrow \mathbb{S}$ where $\mathbb{S}$ is the set of numbers representable with the number format used. The nonlinearity resulting from quantization and overflow is denoted by $N : \mathbb{R} \rightarrow \mathbb{S}$ where $N = Q \cdot O$.

Models for three different quantization schemes are discussed in the following section. It will be assumed that the node's microprocessor uses a double length accumulator to sum the intermediate products and quantize only the final sum.

4.1.2.1 Model 1

Let $N_P : \mathbb{R} \rightarrow \mathbb{S}_p$ be the concatenation of quantization and overflow operators for computations within the node. Here $\mathbb{S}_p$ denotes the set of numbers representable with

the number format used by the node to store the final result. In a GR model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2+1, t) \\ \boldsymbol{x}^t(n_1, n_2, t+1) \end{bmatrix} = N_P\left[\boldsymbol{A}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{B}\boldsymbol{u}(n_1, n_2, t)\right] \tag{4.1}$$

In a FM model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \boldsymbol{x}(n_1, n_2, t) = N_P\left[\boldsymbol{A}_t\boldsymbol{x}(n_1, n_2, t\text{–}1) + \boldsymbol{A}_v\boldsymbol{x}(n_1, n_2\text{–}1, t) + \boldsymbol{A}_h\boldsymbol{x}(n_1\text{–}1, n_2, t)\right. \\ \left. +\boldsymbol{B}_t\boldsymbol{u}(n_1, n_2, t\text{–}1) + \boldsymbol{B}_v\boldsymbol{u}(n_1, n_2\text{–}1, t) + \boldsymbol{B}_h\boldsymbol{u}(n_1\text{–}1, n_2, t)\right] \end{aligned} \tag{4.2}$$

4.1.2.2 Model 2

Due to bandwidth and power limitations the word length of numbers communicated between nodes may be shorter than that for in-node computations. This result in coarser quantization for state vector components communicated between nodes. Let $N_C : \mathbb{R} \to \mathbb{S}_C$ be the concatenation of quantization and overflow operators used for state vector components communicated between nodes. Here $\mathbb{S}_C$ denotes the set of numbers representable with the number format used for communicated numbers. In a GR model based implementation, computation of state vectors within each node can be

modeled by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \end{bmatrix} = N_C \left[\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t) + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t) \right]$$
$$\begin{bmatrix} \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = N_P \left[\begin{bmatrix} \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t) + \begin{bmatrix} \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t) \right] \tag{4.3}$$

In a FM model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \boldsymbol{x}(n_1,n_2,t) = N_P \left[\boldsymbol{A}_t \boldsymbol{x}(n_1,n_2,t\text{–}1) + \boldsymbol{A}_v N_C[\boldsymbol{x}(n_1,n_2\text{–}1,t)] + \boldsymbol{A}_h N_C[\boldsymbol{x}(n_1\text{–}1,n_2,t)] \right. \\ \left. + \boldsymbol{B}_t \boldsymbol{u}(n_1,n_2,t\text{–}1) + \boldsymbol{B}_v N_C[\boldsymbol{u}(n_1,n_2\text{–}1,t)] + \boldsymbol{B}_h N_C[\boldsymbol{u}(n_1\text{–}1,n_2,t)] \right] \end{aligned} \tag{4.4}$$

$N_P : \mathbb{R} \rightarrow \mathbb{S}_p$ is the concatenation of quantization and overflow operators for computations within the node.

4.1.2.3 Model 3

If the required precision for the two directions of communication is different, computed values may be quantized to different precisions in the orthogonal spatial directions. Possible reasons for this could be that the size of the sensor network is different in the two directions or there is a preferred direction in which higher precision is required . In a GR model based implementation, computation of state vectors within each node

can be modeled by:

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \end{bmatrix} &= N_{C_h}\left[\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \end{bmatrix}\boldsymbol{x}(n_1,n_2,t) + \begin{bmatrix} \boldsymbol{B}_1 \end{bmatrix}\boldsymbol{u}(n_1,n_2,t)\right] \\
\begin{bmatrix} \boldsymbol{x}^v(n_1,n_2+1,t) \end{bmatrix} &= N_{C_v}\left[\begin{bmatrix} \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \end{bmatrix}\boldsymbol{x}(n_1,n_2,t) + \begin{bmatrix} \boldsymbol{B}_2 \end{bmatrix}\boldsymbol{u}(n_1,n_2,t)\right] \\
\begin{bmatrix} \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} &= N_P\left[\begin{bmatrix} \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix}\boldsymbol{x}(n_1,n_2,t) + \begin{bmatrix} \boldsymbol{B}_3 \end{bmatrix}\boldsymbol{u}(n_1,n_2,t)\right]
\end{aligned} \tag{4.5}$$

Here $N_{C_h} : \mathbb{R} \rightarrow \mathbb{S}_{C_h}$ and $N_{C_v} : \mathbb{R} \rightarrow \mathbb{S}_{C_v}$ denote concatenation of quantization and overflow operators used for horizontal and vertical state vector components respectively. Here $\mathbb{S}_{C_h}$ and $\mathbb{S}_{C_v}$ denote sets of numbers representable with the number formats used for horizontal and vertical state vector components respectively.

In a FM model based implementation each node transmits its state vector to its neighboring nodes. Both the neighboring nodes in the orthogonal spatial directions can receive the same transmission. Hence quantizing state vectors to different precisions for the two orthogonal spatial directions is not required.

4.1.3 Internal Stability

The definition of GAS for 1-D systems can be directly extended for m-D systems. GAS of a nonlinear system ensures the absence of zero-input limit cycles. It has been argued that the conventional definition of GAS for m-D systems is overly restrictive and less practically applicable Xu et al. [1996]. An alternative definition, practical internal stability is also introduced in Xu et al. [1996]. A m-D system is practically asymptotically stable if the norm of the state vector of the system, driven only by the initial conditions, tend to zero when each of the independent dimensions tend to infinity while keeping the other dimensions of the system finite.

GAS of m-D systems under finite precision arithmetic has drawn considerable attention and a vast literature exist on the subject. Sufficient conditions, for GAS of 2-D systems described by the FM model, under overflow and saturation nonlinearities are given in Kar and Singh [2001b] and Hinamoto [1997] respectively. 2-D systems realized in the FM model are studied for GAS when implemented using two's complement arithmetic in Bose [1995].

For 2-D systems described by the GR model, sufficient conditions for GAS under saturation and quantization nonlinearities are established in [Kar, 2008; Singh, 2008] and Bose [1994] respectively. Further results on GAS of 2-D systems realized in the GR model in the presence of quantization nonlinearities are reported in Kar and Singh [2000]. In Kar and Singh [2001a], sufficient conditions, for GAS of 2-D systems described by either local state space model, under both quantization and overflow nonlinearities are established. A set of necessary conditions for GAS of nonlinear m-D systems is reported in Bauer [1995b].

Since we are interested in GAS, the input to all the nodes is assumed to be zero for $t \geq 0$. Since the sensor network is assumed to be of size $N_1 \times N_2$ the spatial variables satisfy $(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$.

Definition 4.1.1 *A system implemented on a sensor network of size $N_1 \times N_2$ employing either local state space model is said to be GAS if:*

$$\lim_{t \to \infty} \| \boldsymbol{x}(n_1, n_2, t) \| = 0 \quad \forall (n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$$

where $\boldsymbol{x}(n_1, n_2, t)$ is the state vector, $\| \cdot \|$ is any vector norm and the only non-zero initial conditions are given by $\boldsymbol{x}(n_1, n_2, 0)$

Finite extent of the two spatial dimensions of the system and the discrete nature of

the quantization nonlinearity allow a stability condition, which is simple to check (but stronger than those available in the literature), to be derived.

4.1.3.1 GR Model Based Implementation

The system (4.5) is considered since it is the most general case and results derived for this case carry over to the other two cases. Model (4.5) describes the evolution of the system, when different quantization and overflow nonlinearities are applied to temporal, horizontal and vertical state vector components, possibly due to different number formats used to represent them.

Theorem 4.1.1 *The system (4.5) is GAS, if and only if the 1-D system:*

$$\boldsymbol{x}(t+1) = N_P\left[\boldsymbol{A}_9 \boldsymbol{x}(t)\right] \tag{4.6}$$

is GAS. Here $\boldsymbol{x} \in \mathbb{R}^c$ and N_P is the concatenation of quantization and overflow operators for in node computations.

Therefore global asymptotic stability of the 3-D distributed system is equivalent to that of a 1-D system. For any $K < \infty$, the set of state vectors of system (4.6) with a norm less than K is finite. If the state trajectory of the system (4.6) traverses the same non-zero state twice the system is not GAS. Therefore if the system (4.6) is GAS it reaches the origin in finite time, Premaratne et al. [1996].

Proof It is assumed that nodes initially have non zero states. The proof of necessity is trivial. To prove sufficiency, assuming that the system (4.6) is GAS, it will be shown that states of all the nodes reach the origin in finite time.

From the boundary conditions [1] $\boldsymbol{x}^h(0,0,t) = 0$ and $\boldsymbol{x}^v(0,0,t) = 0$ for $\forall t > 0$. For

[1] This simply follows from the fact that there are no nodes along $n_1 = -1$ or $n_2 = -1$.

$t > 0$:

$$\boldsymbol{x}^t(0,0,t+1) = N_P\left[\boldsymbol{A}_9\boldsymbol{x}^t(0,0,t)\right] \tag{4.7}$$

Since the system (4.7) is GAS, $\exists$ a finite M_{00} such that $\boldsymbol{x}^t(0,0,t) = 0$ for $\forall\ t \geq M_{00}$. For $t \geq M_{00}$, $\boldsymbol{x}(0,0,t) = 0$ and from (4.5) $\boldsymbol{x}^h(1,0,t) = 0$. From the boundary conditions $\boldsymbol{x}^v(1,0,t) = 0$ and hence:

$$\boldsymbol{x}^t(1,0,t+1) = N_P\left[A_9\boldsymbol{x}^t(1,0,t)\right] \tag{4.8}$$

Since the system (4.8) is GAS $\exists$ a finite $M_{10} \geq M_{00}$ such that $\boldsymbol{x}^t(1,0,t) = 0$ for $\forall\ t \geq M_{10}$. So, for $t \geq M_{10}$ $\boldsymbol{x}(1,0,t) = 0$. Following the same argument iteratively $\exists$ a finite $M_{N_1-1\ 0}$ such that $\boldsymbol{x}(N_1-1,0,t) = 0$ for $\forall\ t \geq M_{N_1-1\ 0}$. Since by construction $M_{N_1-1\ 0} \geq .... \geq M_{10} \geq M_{00}$, $\boldsymbol{x}(n_1,0,t) = 0$ for $0 \leq n_1 \leq N_1 - 1$ and $t \geq M_{N_1-1\ 0}$. Following a similar argument $\exists$ a finite M_{0N_2-1} such that $\boldsymbol{x}(0,n_2,t) = 0$ for $0 \leq n_2 \leq N_2 - 1$ and for $\forall\ t \geq M_{0N_2-1}$. We have proved that state vectors of the nodes in planes $n_1 = 0$ and $n_2 = 0$ become zero in finite time.

Let $M^0 = \max(M_{N_1-1\ 0}, M_{0N_2-1})$. From (4.5) $\boldsymbol{x}^h(1,1,t) = 0$ and $\boldsymbol{x}^v(1,1,t) = 0$ for $t \geq M^0$. Therefore for $t \geq M^0$:

$$\boldsymbol{x}^t(1,1,t+1) = N_P\left[\boldsymbol{A}_9\boldsymbol{x}^t(1,1,t)\right] \tag{4.9}$$

Since the system (4.9) is GAS $\exists$ a finite $M_{11} \geq M^0$ such that $\boldsymbol{x}^t(1,1,t) = 0$ for $\forall\ t \geq M_{11}$. We have $\boldsymbol{x}(1,1,t) = 0$ for $\forall\ t \geq M_{11}$. It can be seen that the argument used to show the GAS of the nodes in planes $n_1 = 0$ and $n_2 = 0$ extend to nodes in planes $n_1 = 1$ and $n_2 = 1$.

This can be repeated to show that $\exists$ a finite M such that $\boldsymbol{x}(n_1,n_2,t) = 0$ for all

$n_1 \in [0, N_1 - 1]$ and $n_2 \in [0, N_2 - 1]$, for $t \geq M$. This completes the proof.

4.1.3.2 FM Model Based Implementation

The system in (4.4) is considered since it is the most general case and results derived for this case carry over to the other cases. Evolution of the system, when different quantization and overflow operators are used for in node computations and communicated state vectors, is described by the model (4.4).

Theorem 4.1.2 *The system (4.4) is GAS, if and only if the 1-D system:*

$$\boldsymbol{x}(t+1) = N_P\left[\boldsymbol{A}_t \boldsymbol{x}(t)\right] \tag{4.10}$$

is GAS. N_P is the concatenation of quantization and overflow operators for in-node computations and $\boldsymbol{x} \in \mathbb{R}^n$.

Proof The proof is similar to the proof of Theorem 4.1.1.

For implementations using either local state space model, GAS of the 3-D system under quantization and overflow nonlinearities is equivalent to that of a 1-D system under quantization and overflow nonlinearities. Conditions, under which the GAS of a system under quantization and overflow nonlinearities is equivalent to the GAS of that under quantization only, are established in [Kar and Singh, 2001a; Leclerc and Bauer, 1994]. An exhaustive search algorithm to determine the GAS of 1-D systems under fixed point quantization is presented in Premaratne et al. [1996]. The algorithm reported in Premaratne et al. [1996] can be readily extended to determine the GAS of 1-D systems under both quantization and overflow nonlinearities. Therefore Theorems 4.1.1 and 4.1.2 allow to test the GAS, of 3-D systems implemented on grid sensor net-

works, using either local state space model under the occurrence of fixed point quantization and overflow nonlinearities.

GAS of the system is independent of its dynamics along horizontal and vertical dimensions. This is due to the finite extent of the system in the said dimensions. An important implication is that, GAS of the sensor network is independent of the quantization and overflow operations applied to communicated state vectors.

4.1.4 BIBO Stability

The definition of BIBO stability for 1-D systems can be directly extended for m-D systems. It has been argued, that the direct extension of the definition of BIBO stability for 1-D systems to m-D systems, is too restrictive for most applications
Agathoklis and Bruton [1983]; Lazar and Bruton [1993]. Practical BIBO stability, an alternative input-output stability criterion for m-D systems is introduced in
Agathoklis and Bruton [1983].

When finite precision arithmetic is used in computations, overflow nonlinearities are introduced to the system. Hence the output vector of a system, under finite precision arithmetic, necessarily has a bounded norm. BIBO stability of distributed 3-D systems, in the presence of only the quantization nonlinearity, is studied in the following.

4.1.4.1 GR Model Based Implementation

Assuming infinite precision arithmetic, a sufficient condition for the BIBO stability of a system implemented on a sensor network of size $N_1 \times N_2$, is established first.

Theorem 4.1.3 *The system*

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B_1} \\ \boldsymbol{B_2} \\ \boldsymbol{B_3} \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = \boldsymbol{C}\boldsymbol{x}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t) \tag{4.11}$$

where $0 \le n_1 \le N_1 - 1$ *and* $0 \le n_2 \le N_2 - 1$ *is BIBO stable, if the 1-D system,*

$$\boldsymbol{x}(t+1) = \boldsymbol{A}_9\boldsymbol{x}(t) \tag{4.12}$$

is GAS. Here $\boldsymbol{x} \in \mathbb{R}^c$*,*

Proof To prove sufficiency, assuming the system (4.12) is GAS and the input $\boldsymbol{u}(n_1,n_2,t)$ is bounded, it will be shown that the output $\boldsymbol{y}(n_1,n_2,t)$ is bounded. From the boundary conditions $\boldsymbol{x}^h(0,0,t) = 0$ and $\boldsymbol{x}^v(0,0,t) = 0$.

$$\begin{aligned} \boldsymbol{x}^t(0,0,t+1) &= \boldsymbol{A}_9\boldsymbol{x}^t(0,0,t) + \boldsymbol{B}_3\boldsymbol{u}(0,0,t) \\ \boldsymbol{y}(0,0,t) &= \boldsymbol{C}_3\boldsymbol{x}^t(0,0,t) + \boldsymbol{D}\boldsymbol{u}(0,0,t) \end{aligned} \tag{4.13}$$

Since the system (4.12) is GAS, system (4.13) is BIBO stable. Therefore, $\| \boldsymbol{y}(0,0,t) \|$ is bounded. Moreover $\| \boldsymbol{x}^t(0,0,t) \|$ and $\| \boldsymbol{x}(0,0,t) \|$ are bounded.

$$\begin{aligned} \boldsymbol{x}^t(1,0,t+1) = &\quad \boldsymbol{A}_9\boldsymbol{x}^t(1,0,t) + \boldsymbol{A}_7\boldsymbol{x}^h(1,0,t) + \boldsymbol{B}_3\boldsymbol{u}(1,0,t) \\ \boldsymbol{y}(1,0,t) = &\quad \boldsymbol{C}_3\boldsymbol{x}^t(1,0,t) + \boldsymbol{C}_1\boldsymbol{x}^h(1,0,t) + \boldsymbol{D}\boldsymbol{u}(1,0,t) \end{aligned} \tag{4.14}$$

Since $\boldsymbol{x}^h(1,0,t) = \boldsymbol{A}_3\boldsymbol{x}^t(0,0,t) + \boldsymbol{B}_1\boldsymbol{u}(0,0,t)$, $\| \boldsymbol{x}^h(1,0,t) \|$ is bounded. System (4.14) can be considered as a single dimensional system with a bounded input $\boldsymbol{A}_7\boldsymbol{x}^h(1,0,t) + \boldsymbol{B}_3\boldsymbol{u}(0,0,t)$. The system (4.14) is BIBO stable and $\| \boldsymbol{y}(1,0,t) \|$ is bounded. Moreover $\| \boldsymbol{x}^t(1,0,t) \|$ and $\| \boldsymbol{x}(1,0,t) \|$ are bounded. This argument can be used iteratively to show that $\| \boldsymbol{y}(n_1,0,t) \|$ and $\| \boldsymbol{x}(n_1,0,t) \|$ are bounded for $0 \leq n_1 \leq N_1 - 1$. Following a similar argument it can be shown that $\| \boldsymbol{y}(0,n_2,t) \|$ and $\| \boldsymbol{x}(0,n_2,t) \|$ are bounded for $0 \leq n_2 \leq N_2 - 1$.

$$\begin{aligned}
\boldsymbol{x}^t(1,1,t+1) =& \boldsymbol{A}_9\boldsymbol{x}^t(1,1,t) + \boldsymbol{A}_7\boldsymbol{x}^h(1,1,t) + \boldsymbol{A}_8\boldsymbol{x}^v(1,1,t) + \boldsymbol{B}_3\boldsymbol{u}(1,1,t) \\
\boldsymbol{y}(1,1,t) =& \boldsymbol{C}_3\boldsymbol{x}^t(1,1,t) + \boldsymbol{C}_1\boldsymbol{x}^h(1,1,t) + \boldsymbol{C}_2\boldsymbol{x}^v(1,1,t) + \boldsymbol{D}\boldsymbol{u}(1,1,t) \quad (4.15)
\end{aligned}$$

The system (4.15) can be considered as a single dimensional system with a bounded input. Therefore $\| \boldsymbol{y}(1,1,t) \|$ and $\| \boldsymbol{x}(1,1,t) \|$ are bounded. This can be repeated to show that $\| \boldsymbol{y}(n_1,n_2,t) \|$ is bounded for $0 \leq n_1 \leq N_1 - 1$ and $0 \leq n_2 \leq N_2 - 1$. This completes the proof of Theorem 4.1.3.

If the 1-D system (4.12) which describes the temporal dynamics of the 3-D system (4.11) is GAS, BIBO stability of the system is independent of its spatial dynamics. The result is due to the finite spatial extent of the 3-D system.

Above result can be used to derive a sufficient condition for the BIBO stability of a system, implemented on a sensor network, in the presence of quantization nonlinearity. The model (4.5) is used since it is the most general case. It can be modified to include

only the quantization nonlinearities as follows:

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \end{bmatrix} =& Q_{C_h}\left[\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \end{bmatrix}\boldsymbol{x}(n_1,n_2,t)+\begin{bmatrix} \boldsymbol{B}_1 \end{bmatrix}\boldsymbol{u}(n_1,n_2,t)\right] \\
\begin{bmatrix} \boldsymbol{x}^v(n_1,n_2+1,t) \end{bmatrix} =& Q_{C_v}\left[\begin{bmatrix} \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \end{bmatrix}\boldsymbol{x}(n_1,n_2,t)+\begin{bmatrix} \boldsymbol{B}_2 \end{bmatrix}\boldsymbol{u}(n_1,n_2,t)\right] \\
\begin{bmatrix} \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} =& Q_P\left[\begin{bmatrix} \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix}\boldsymbol{x}(n_1,n_2,t)+\begin{bmatrix} \boldsymbol{B}_3 \end{bmatrix}\boldsymbol{u}(n_1,n_2,t)\right] \\
\boldsymbol{y}(n_1,n_2,t) =& Q_P\left[\boldsymbol{C}\boldsymbol{x}(n_1,n_2,t)+\boldsymbol{D}\boldsymbol{u}(n_1,n_2,t)\right]
\end{aligned} \tag{4.16}$$

Here $Q_P : \mathbb{R} \to \mathbb{S}_P$, $Q_{C_h} : \mathbb{R} \to \mathbb{S}_{C_h}$ and $Q_{C_v} : \mathbb{R} \to \mathbb{S}_{C_v}$ denote quantization operators used for temporal, horizontal and vertical state vector components respectively. Here $\mathbb{S}_{C_h}$ and $\mathbb{S}_{C_v}$ denote sets of numbers representable with the number formats used for horizontal and vertical state vector components respectively. The set of numbers representable with the number format used for in-node computations is denoted by $\mathbb{S}_P$.

Theorem 4.1.4 *The system (4.16) is BIBO stable, if the 1-D system,*

$$\boldsymbol{x}(t+1) = \boldsymbol{A}_9\boldsymbol{x}(t) \tag{4.17}$$

is GAS, where $\boldsymbol{x} \in \mathbb{R}^c$.

Proof Let:

$$
\begin{aligned}
Q_{C_h}[[\boldsymbol{A}_1\ \boldsymbol{A}_2\ \boldsymbol{A}_3]\,\boldsymbol{x}(n_1,n_2,t)+[\boldsymbol{B}_1]\,\boldsymbol{u}(n_1,n_2,t)] &= [\boldsymbol{A}_1\ \boldsymbol{A}_2\ \boldsymbol{A}_3]\,\boldsymbol{x}(n_1,n_2,t) \\
&\quad + [\boldsymbol{B}_1]\,\boldsymbol{u}(n_1,n_2,t)+\boldsymbol{e}^h(n_1,n_2,t) \\
Q_{C_v}[[\boldsymbol{A}_4\ \boldsymbol{A}_5\ \boldsymbol{A}_6]\,\boldsymbol{x}(n_1,n_2,t)+[\boldsymbol{B}_2]\,\boldsymbol{u}(n_1,n_2,t)] &= [\boldsymbol{A}_4\ \boldsymbol{A}_5\ \boldsymbol{A}_6]\,\boldsymbol{x}(n_1,n_2,t) \\
&\quad + [\boldsymbol{B}_2]\,\boldsymbol{u}(n_1,n_2,t)+\boldsymbol{e}^v(n_1,n_2,t) \\
Q_{C_t}[[\boldsymbol{A}_7\ \boldsymbol{A}_8\ \boldsymbol{A}_9]\,\boldsymbol{x}(n_1,n_2,t)+[\boldsymbol{B}_3]\,\boldsymbol{u}(n_1,n_2,t)] &= [\boldsymbol{A}_7\ \boldsymbol{A}_8\ \boldsymbol{A}_9]\,\boldsymbol{x}(n_1,n_2,t) \\
&\quad + [\boldsymbol{B}_3]\,\boldsymbol{u}(n_1,n_2,t)+\boldsymbol{e}^t(n_1,n_2,t)
\end{aligned}
$$

where the error vectors $\boldsymbol{e}^h \in \mathbb{R}^a$, $\boldsymbol{e}^v \in \mathbb{R}^b$ and $\boldsymbol{e}^t \in \mathbb{R}^c$ are the errors introduced to horizontal, vertical and temporal state vector components due to quantization. Since fixed point number representations schemes are used, elements of vectors $\boldsymbol{e}^h$, $\boldsymbol{e}^v$ and $\boldsymbol{e}^t$ are bounded. Therefore the error vector $(\boldsymbol{e}^{h^T}(n_1,n_2,t), \boldsymbol{e}^{v^T}(n_1,n_2,t), \boldsymbol{e}^{t^T}(n_1,n_2,t))^T$ has a bounded norm.

$$
\begin{aligned}
\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t) \\
&\quad + \begin{bmatrix} \boldsymbol{e}^h(n_1,n_2,t) \\ \boldsymbol{e}^v(n_1,n_2,t) \\ \boldsymbol{e}^t(n_1,n_2,t) \end{bmatrix}
\end{aligned} \tag{4.18}
$$

The system (4.18) can be considered as a 3-D linear space and time invariant system. Let the system (4.17) be GAS. According to the Theorem 4.1.3 system (4.18) is BIBO stable. When $\| \boldsymbol{u}(n_1,n_2,t) \|$ is bounded the input to the system (4.18) is bounded. Therefore $\| \boldsymbol{x}(n_1,n_2,t) \|$ and hence $\| \boldsymbol{y}(n_1,n_2,t) \|$ are bounded. This completes the

proof of theorem (4.1.4).

4.1.4.2 FM Model Based Implementation

The system (4.4) is considered since it is the most general case. It can be modified to include only the quantization nonlinearities as follows:

$$\begin{aligned}\boldsymbol{x}(n_1,n_2,t) = Q_P\,[&\boldsymbol{A}_t\boldsymbol{x}(n_1,n_2,t\text{–}1) + \boldsymbol{A}_v Q_C[\boldsymbol{x}(n_1,n_2\text{–}1,t)] + \boldsymbol{A}_h Q_C[\boldsymbol{x}(n_1\text{–}1,n_2,t)] \\ &+\boldsymbol{B}_t\boldsymbol{u}(n_1,n_2,t\text{–}1) + \boldsymbol{B}_v Q_C[\boldsymbol{u}(n_1,n_2\text{–}1,t)] + \boldsymbol{B}_h Q_C[\boldsymbol{u}(n_1\text{–}1,n_2,t)]]\end{aligned} \tag{4.19}$$

Here $Q_P : \mathbb{R} \to \mathbb{S}_P$ and $Q_C : \mathbb{R} \to \mathbb{S}_C$ denote quantization operators used for computations within the node and communication among nodes respectively. $\mathbb{S}_P$ and $\mathbb{S}_C$ denote sets of numbers representable with the number formats used for in node computations and inter-node communication respectively.

Theorem 4.1.5 *The system (4.19) is BIBO stable if the 1-D system:*

$$\boldsymbol{x}(t+1) = \boldsymbol{A}_t\boldsymbol{x}(t) \tag{4.20}$$

is GAS, where $\boldsymbol{x} \in \mathbb{R}^n$

Proof Theorem can be proved using a similar line of argument as in the proof of Theorem 4.1.4.

The error introduced in quantization, in systems described by GR and FM models can be considered as a second input to the system. Quantization error in fixed point quantization schemes is bounded by the quantization step size. Therefore GAS of 1-D systems (4.17) and (4.20), which are sufficient conditions for the BIBO stability of the

corresponding un-quantized 3-D systems, are also sufficient conditions for the BIBO stability of the quantized 3-D systems

4.1.5 Example

In this section, an example implementation of a linear filter on a grid sensor network is presented. Let the transfer function of the single input single output filter be as follows:

$$H(z_1, z_2, z_t) = \frac{z_t^{-1}}{1 - az_t^{-1} - bz_1^{-1}z_t^{-1} - cz_2^{-1}z_t^{-1}} \tag{4.21}$$

4.1.5.1 GR Model Based Implementation

The above system can be realized using the GR model:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2+1, t) \\ \boldsymbol{x}^t(n_1, n_2, t+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & a_3 \\ 0 & 0 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2, t) \\ \boldsymbol{x}^t(n_1, n_2, t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \boldsymbol{u}(n_1, n_2, t)$$

$$\boldsymbol{y}(n_1, n_2, t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(n_1, n_2, t) \tag{4.22}$$

where $a_9 = a$, $a_3 a_7 = b$ and $a_6 a_8 = c$. According to Theorem 4.1.1, the system (4.22) is GAS if and only if the system,

$$\boldsymbol{x}(t+1) = N_P\left[a_9 \boldsymbol{x}(t)\right] \tag{4.23}$$

is GAS. To illustrate the convergence of state vectors to the origin, the system (4.22) was simulated on a sensor network of size 4×4. Coefficients a, b and c in the transfer function were set to 0.375, 0.25 and 0.25 respectively. Coefficients a_3, a_6, a_7 and a_8

were set to 0.5. The quantization scheme described by model 2 was used. State vector components computed for use within the node were quantized to 8 bit sign magnitude with 4 fractional bits. State vector components communicated between nodes were quantized to 4 bit sign magnitude with 3 fractional bits. The input given to the system is as follows:

$$\boldsymbol{u}(n_1, n_2, t) = \begin{cases} 1 & 1 \leq t \leq 5, \;\; n_1, n_2 \in [0, 3] \\ 0 & t > 5, \;\; n_1, n_2 \in [0, 3] \end{cases}$$

The input is used only to drive state vectors of nodes to non zero values. Figures 4.7 and 4.8 show plots of the Euclidean norm of state vectors of nodes $(1, 1)$ and $(4, 4)$ respectively versus time.

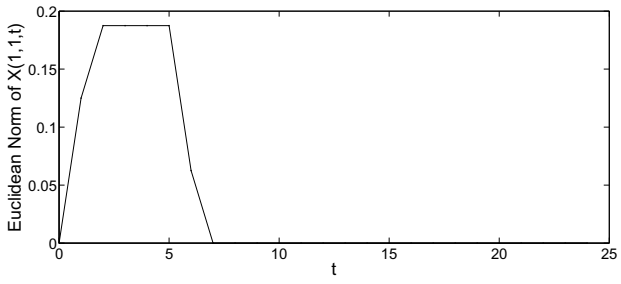


Figure 4.7. Euclidean norm of the state vector of the node $(1, 1)$ versus t for the GR model

Figures 4.9 and 4.10 show plots of the Euclidean norm of the state vectors versus time for all other nodes of the sensor network. Due to symmetry, state vectors of nodes (i, j) and (j, i) are equal.

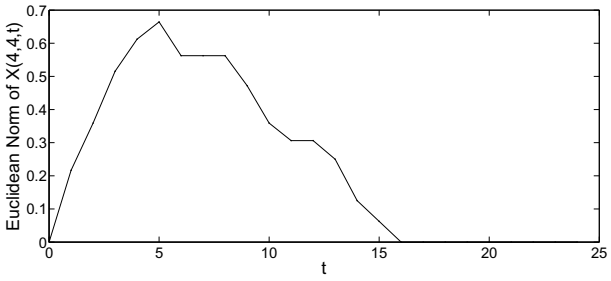


Figure 4.8. Euclidean norm of the state vector of the node $(4, 4)$ versus t for the GR model

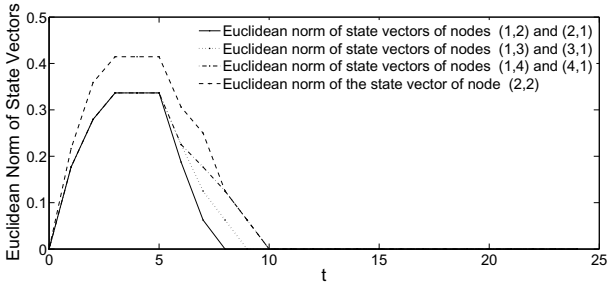


Figure 4.9. Euclidean norm of the state vectors versus t for the GR model

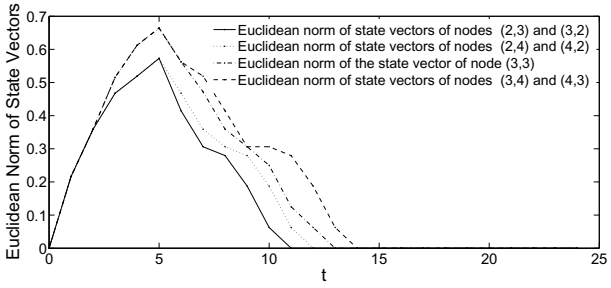


Figure 4.10. Euclidean norm of the state vectors versus t for the GR model

Using the algorithm given in Premaratne et al. [1996], it can be verified that the system (4.23) is GAS for the value of a_9 used in the example. According to the Theorem 4.1.1, the system (4.22) is also GAS. The simulation results are in accordance with the theoretical conclusion. Note that the state vector of node $(4, 4)$ reach the origin later than the state vector of node $(1, 1)$. This is due to the system being first octant causal.

4.1.5.2 FM Model Based Implementation

The input-output transfer function (4.21) can be realized using the FM model:

$$\begin{aligned}\boldsymbol{x}(n_1, n_2, t) &= \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}(n_1, n_2, t-1) + \begin{bmatrix} 0 & b \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(n_1 - 1, n_2, t) \\ &\quad + \begin{bmatrix} 0 & c \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(n_1, n_2 - 1, t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \boldsymbol{u}(n_1, n_2, t-1) \\ \boldsymbol{y}(n_1, n_2, t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}(n_1, n_2, t) \end{aligned} \tag{4.24}$$

where $\boldsymbol{x}(n_1, n_2, t) \in \mathbb{R}^2$. According to the Theorem 4.1.2, the system (4.24) is GAS if and only if the system,

$$\boldsymbol{x}(t+1) = N_P \left[\begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}(t) \right] \tag{4.25}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^2$ is GAS. To illustrate the convergence of state vectors to the origin, the system (4.24) was simulated on a sensor network of size 4×4. The input to the system and the coefficients a, b and c were the same as for the simulation of the GR

model based implementation described above. The quantization scheme described by model 2 was used. In node computations were done using 8 bit sign magnitude numbers with 4 fractional bits. Communicated state vectors were represented using 4 bit sign magnitude with 3 fractional bits.

Figures 4.11 and 4.12 show plots of the Euclidean norm of state vectors of nodes $(1,1)$ and $(4,4)$ versus time respectively.

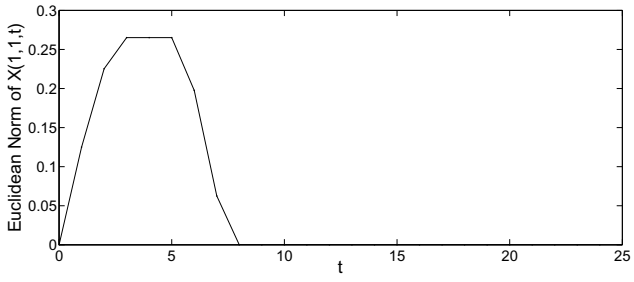


Figure 4.11. Euclidean norm of the state vector of the node $(1,1)$ versus t for the FM model

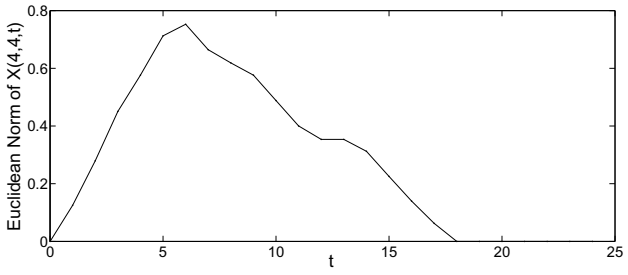


Figure 4.12. Euclidean norm of the state vector of the node $(4, 4)$ versus t for the FM model

Figures 4.13 and 4.14 show plots of the Euclidean norm of the state vectors versus time for all other nodes of the sensor network. State vectors of nodes (i, j) and (j, i) are equal due to symmetry in this example.

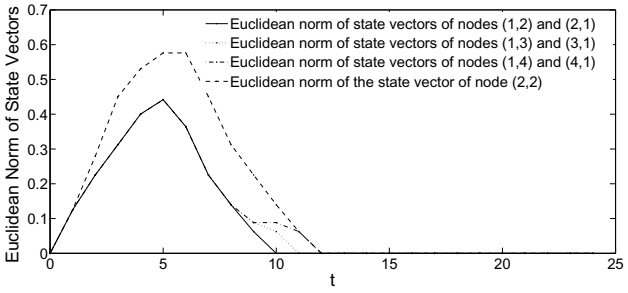


Figure 4.13. Euclidean norm of the state vectors versus t for the FM model

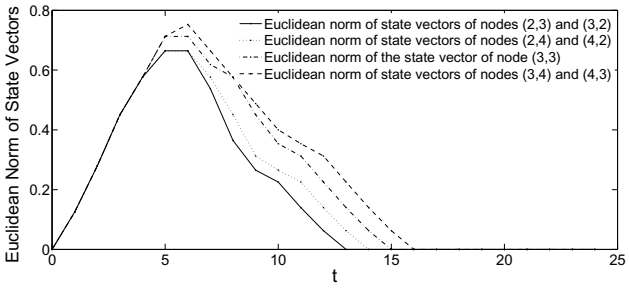


Figure 4.14. Euclidean norm of the state vectors versus t for the FM model

For the value of a used in the example the system (4.25) is GAS. Hence the system (4.24) is also GAS according to the theorem 4.1.2. Simulation results are in accordance with the theoretical findings.

4.2 Floating Point Arithmetic

Though most of the commercially available sensor nodes use fixed point processors for computations, sensor nodes capable of floating point computations have also appeared. Sun SPOT is an example for a sensor node capable of floating point computations [Labs, 2010] As computational capabilities of embedded processors improve, floating point processors can be expected to be used widely in sensor nodes in the future.

4.2.1 Floating Point Representation of Numbers

In floating point base 2 formats a real number x is represented as $x = sgn(x)m(x)2^{e(x)}$. Here $sgn(x) = -1$ if $x < 0$ and $sgn(x) = 1$ otherwise. Furthermore $m(x)$ is the mantissa of x and $e(x)$ is the exponent of x. The mantissa is

usually normalized such that $0.5 \leq m(x) < 1$. Sign of x, $sgn(x)$ can be represented by 1 bit. Mantissa and exponent are represented using fixed point schemes. Number of binary digits used to represent mantissa and exponent is determined based on the relative precision and the range of numbers required to be represented. For example in the IEEE 754 single precision floating point format the mantissa is represented by 23 bits and the exponent is represented by 8 bits.

4.2.1.1 Floating Point Multiplication

Let x_1 and x_2 be the two numbers, represented in floating point, that are to be multiplied. Their product is given by $sgn(x_1x_2)m(x_1)m(x_2)2^{e(x_1)+e(x_2)}$. Three different kinds of errors can occur in representing the product of x_1 and x_2 in a floating point format.

1. Overflow occurs if the product is larger than the largest representable number.

2. Underflow occurs if the product is smaller than the smallest representable number.

3. To represent the mantissa of the product exactly, word length of the mantissa should be equal to or larger than, the summation of the word lengths of the mantissas of the two multiplicands. Otherwise a quantization error is introduced in the multiplication.

Let $N^m : \mathbb{R} \rightarrow \mathbb{S}^m$ denote the overall non-linearity introduced in floating point multiplication. Here $\mathbb{S}^m$ denotes the set of numbers representable with the number format used.

4.2.1.2 Floating Point Addition

Let x_1 and x_2 be the two numbers, represented in floating point, that are to be added. Without loss of generality it can be assumed that $x_1 \leq x_2$. To add the two numbers, mantissa of the smaller number is denormalized such that exponents of the two numbers are equal. Their summation is given by $\{sgn(x_1)m(x_1)2^{e(x_1)-e(x_2)} + sgn(x_2)m(x_2)\}2^{e(x_2)}$. The mantissa of the summation may have to be normalized to represent it in a floating point format. Overflow, underflow and quantization errors can be introduced in floating point addition. Let $N^a : \mathbb{R} \rightarrow \mathbb{S}^a$ denote the overall non-linearity introduced in floating point addition. Here $\mathbb{S}^a$ denotes the set of numbers representable with the number format used.

4.2.2 Quantization Models

Floating point computations introduce nonlinearities to otherwise linear system models (2.1) and (2.2). System models (2.1) and (2.2) are modified to incorporate effects of different floating point quantization schemes in this section. The following notation is used to make the presentation of the quantization models more concise. Let the $(i,j) - th$ element of matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be a_{ij} and the $i - th$ element of vector $\boldsymbol{x} \in \mathbb{R}^n$ be x_i. The product of matrix $\boldsymbol{A}$ with vector $\boldsymbol{x}$ computed using floating point arithmetic is denoted by $N[\boldsymbol{Ax}]$, where $N[\boldsymbol{Ax}]$ is of the form given by (4.26).

$$N[\mathbf{Ax}] = \begin{pmatrix} N^a[N^m[a_{11}x_1] + N^a[N^m[a_{12}x_2] + N^a[N^m[a_{13}x_3] + \cdots + N^m[a_{1n}x_n]]]] \\ N^a[N^m[a_{21}x_1] + N^a[N^m[a_{22}x_2] + N^a[N^m[a_{23}x_3] + \cdots + N^m[a_{2n}x_n]]]] \\ \cdots\cdots \\ \cdots\cdots \\ N^a[N^m[a_{n1}x_1] + N^a[N^m[a_{n2}x_2] + N^a[N^m[a_{n3}x_3] + \cdots + N^m[a_{nn}x_n]]]] \end{pmatrix} \tag{4.26}$$

In (4.26), N^a and N^m denote nonlinearities caused by floating point addition and multiplication respectively.

4.2.2.1 Model 1

Let $N_P^a : \mathbb{R} \rightarrow \mathbb{S}_p^a$ and $N_P^m : \mathbb{R} \rightarrow \mathbb{S}_p^m$ denote nonlinearities caused by floating point addition and multiplication respectively within the node. Here $\mathbb{S}_p^a$ and $\mathbb{S}_p^m$ denote the sets of numbers representable with the number formats used by the node to store the results of addition and multiplication respectively. In a GR model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2+1, t) \\ \boldsymbol{x}^t(n_1, n_2, t+1) \end{bmatrix} = N_P^a \left[N_P[\boldsymbol{A}\boldsymbol{x}(n_1, n_2, t)] + N_P \left[\boldsymbol{B}\boldsymbol{u}(n_1, n_2, t) \right] \right] \tag{4.28}$$

It is assumed that $\boldsymbol{A}\boldsymbol{x}(n_1, n_2, t)$ and $\boldsymbol{B}\boldsymbol{u}(n_1, n_2, t)$ are computed first and then their summation is computed. The nonlinearity introduced in the on node computation of a product of a matrix and a vector is denoted by N_P. Nonlinear operator N_P is of the form given by (4.26). In a FM model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \boldsymbol{x}(n_1, n_2, t) = N_P^a \left[N_P[\boldsymbol{A}_t \boldsymbol{x}(n_1, n_2, t\text{–}1)] + N_P^a[N_P[\boldsymbol{A}_v \boldsymbol{x}(n_1, n_2\text{–}1, t)] \right. \\ + N_P^a[N_P[\boldsymbol{A}_h \boldsymbol{x}(n_1\text{–}1, n_2, t)] + N_P^a[N_P[\boldsymbol{B}_t \boldsymbol{u}(n_1, n_2, t\text{–}1)] \\ \left. + N_P^a[N_P[\boldsymbol{B}_v \boldsymbol{u}(n_1, n_2\text{–}1, t)] + N_P[\boldsymbol{B}_h \boldsymbol{u}(n_1\text{–}1, n_2, t)]]]]] \end{aligned} \tag{4.29}$$

4.2.2.2 Model 2

Due to bandwidth and power limitations the word length of numbers communicated between nodes may be shorter than that for in-node computations. This results in coarser quantization for state vector components communicated between nodes. Let $N_C : \mathbb{R} \rightarrow \mathbb{S}_C$ be the quantization operator used for state vector components communicated between nodes. Here $\mathbb{S}_C$ denotes the set of numbers representable with the number format used for communicated numbers. In a GR model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \end{bmatrix} =& N_C \left[N_P \left[\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t) \right] \right. \\
& \left. + N_P \left[\begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t) \right] \right] \\
\begin{bmatrix} \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} =& N_P^a \left[N_P \left[\begin{bmatrix} \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t) \right] \right. \\
& \left. + N_P \left[\begin{bmatrix} \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t) \right] \right]
\end{aligned} \tag{4.30}$$

In (4.30), it is assumed that communicated state vector components are computed using the same floating point scheme as the temporal state vector component and then quantized to shorter word lengths for communication. In a FM model based implementation,

computation of state vectors within each node can be modeled by:

$$\begin{aligned}\boldsymbol{x}(n_1,n_2,t) = N_P^a\left[N_P[\boldsymbol{A}_t\boldsymbol{x}(n_1,n_2,t–1)] + N_P^a[N_P[\boldsymbol{A}_v[N_C\left[\boldsymbol{x}(n_1,n_2–1,t)\right]]]\right.\\ + N_P^a[N_P[\boldsymbol{A}_h[N_C\left[\boldsymbol{x}(n_1–1,n_2,t)\right]]] + N_P^a[N_P[\boldsymbol{B}_t\boldsymbol{u}(n_1,n_2,t–1)]\\ \left. + N_P^a[N_P[\boldsymbol{B}_v[N_C\left[\boldsymbol{u}(n_1,n_2–1,t)\right]]] + N_P[\boldsymbol{B}_h[N_C\left[\boldsymbol{u}(n_1–1,n_2,t)\right]]]]]]]\end{aligned} \tag{4.31}$$

4.2.2.3 Model 3

If the required precision for the two directions of communication is different, computed values may be quantized to different precisions in the orthogonal spatial directions. Possible reasons for this could be the size of the sensor network is different in the two directions or there is a preferred direction in which higher precision is required. In a GR model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned}\left[\begin{array}{c}\boldsymbol{x}^h(n_1+1,n_2,t)\end{array}\right] =& N_{C_h}\left[N_P\left[\left[\begin{array}{ccc}\boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3\end{array}\right]\boldsymbol{x}(n_1,n_2,t)\right]\right.\\ &\left.+N_P\left[\left[\begin{array}{c}\boldsymbol{B}_1\end{array}\right]\boldsymbol{u}(n_1,n_2,t)\right]\right]\\ \left[\begin{array}{c}\boldsymbol{x}^v(n_1,n_2+1,t)\end{array}\right] =& N_{C_v}\left[N_P\left[\left[\begin{array}{ccc}\boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6\end{array}\right]\boldsymbol{x}(n_1,n_2,t)\right]\right.\\ &\left.+N_P\left[\left[\begin{array}{c}\boldsymbol{B}_2\end{array}\right]\boldsymbol{u}(n_1,n_2,t)\right]\right]\\ \left[\begin{array}{c}\boldsymbol{x}^t(n_1,n_2,t+1)\end{array}\right] =& N_P\left[N_P\left[\left[\begin{array}{ccc}\boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9\end{array}\right]\boldsymbol{x}(n_1,n_2,t)\right]\right.\\ &\left.+N_P\left[\left[\begin{array}{c}\boldsymbol{B}_3\end{array}\right]\boldsymbol{u}(n_1,n_2,t)\right]\right]\end{aligned} \tag{4.32}$$

Here $N_{C_h} : \mathbb{R} \rightarrow \mathbb{S}_{C_h}$ and $N_{C_v} : \mathbb{R} \rightarrow \mathbb{S}_{C_v}$ denote quantization operators used for horizontal and vertical state vector components respectively. Here $\mathbb{S}_{C_h}$ and $\mathbb{S}_{C_v}$ denote

sets of numbers representable with the number formats used for horizontal and vertical state vector components respectively. In (4.32), it is assumed that communicated state vector components are computed using the same floating point scheme as the temporal state vector component and then quantized to shorter word lengths for communication.

In a FM model based implementation each node transmits its state vector to its neighboring nodes. Both the neighboring nodes in the orthogonal spatial directions can receive the same transmission. Hence quantizing state vectors to different precisions for the two orthogonal spatial directions is not required.

4.2.3 Stability of the System

Global asymptotic stability of the quantized system will be considered in this work, while the input output stability of the system would be a subject for future research.

4.2.3.1 GR Model Based Implementation

The system (4.32) is considered since it is the most general case and results derived for this case carry over to the other two cases.

Theorem 4.2.1 *The system* (4.32) *is GAS, if and only if the 1-D system,*

$$\boldsymbol{x}(t+1) = N_P\left[\boldsymbol{A}_9\boldsymbol{x}(t)\right] \tag{4.33}$$

is GAS. Here $\boldsymbol{x} \in \mathbb{R}^c$ *and* N_P *denotes the nonlinearity introduced in computations within the node.*

Proof The proof is similar to the proof of Theorem 4.1.1.

4.2.3.2 FM Model Based Implementation

The system (4.31) is considered since it is the most general case and results derived for this case carry over to the other cases.

Theorem 4.2.2 *The system* (4.31) *is GAS, if and only if the 1-D system,*

$$\boldsymbol{x}(t+1) = N_P\left[\boldsymbol{A}_t \boldsymbol{x}(t)\right] \tag{4.34}$$

is GAS. Here $\boldsymbol{x} \in \mathbb{R}^n$.

Proof The proof is similar to the proof of Theorem 4.1.1.

Implications of theorems 4.2.1 and 4.2.2 are similar to those of theorems 4.1.1 and 4.1.2 for fixed point implementations. Global asymptotic stability of distributed systems implemented on the sensor network is independent of the quantization and overflow operations applied to communicated state vectors. Global asymptotic stability of the 3-D system under floating point arithmetic is equivalent to the global asymptotic stability of a 1-D system under floating point computations. Asymptotic stability of 1-D systems described by second order difference equations under floating point arithmetic is studied in Bauer and Wang [1993]. Asymptotic stability of 1-D systems described by state space models under floating point computations is analyzed in Bauer [1995a]; Ralev and Bauer [1999]. Note that in contrast to fixed point systems, in floating point systems an otherwise GAS system can cause an unbounded response to a bounded input or a zero input with non-zero initial conditions due to quantization nonlinearities.

4.2.4 Example

Theoretical results are illustrated using an example implementation of a linear filter on a grid sensor network. Let the transfer function of the single input single output filter

be given by (4.35)

$$H(z_1, z_2, z_t) = \frac{n(z_1, z_2, z_t)}{d(z_1, z_2, z_t)} \tag{4.35}$$

where

$$\begin{aligned} n(z_1, z_2, z_t) &= \frac{1}{8}z_1^{-1} - \frac{1}{128}z_1^{-1}z_2^{-1} - \frac{5}{256}z_1^{-1}z_t^{-1} + \frac{129}{4096}z_1^{-1}z_2^{-1}z_t^{-1} \\ d(z_1, z_2, z_t) &= 1 - \frac{1}{16}z_1^{-1} - \frac{1}{16}z_2^{-1} + \frac{1}{256}z_1^{-1}z_2^{-1} \\ &\quad - \frac{5}{4}z_t^{-1} + \frac{5}{64}z_1^{-1}z_t^{-1} + \frac{5}{64}z_2^{-1}z_t^{-1} - \frac{5}{1024}z_1^{-1}z_2^{-1}z_t^{-1} \\ &\quad + \frac{25}{128}z_t^{-2} - \frac{25}{2048}z_1^{-1}z_t^{-2} - \frac{25}{2048}z_2^{-1}z_t^{-2} + \frac{25}{32768}z_1^{-1}z_2^{-1}z_t^{-2} \end{aligned}$$

The sensor network is assumed to be of size 4×4. It is assumed that the exponent of the floating point representation can be any integer. Therefore overflow and underflow do not occur. In stable linear systems used in practical applications overflow is unlikely to occur when the maximum exponent allowed by the floating point representation is sufficiently large. The effect of underflow on global asymptotic stability of the system depends on how underflow is handled. Hence allowing the range of the exponent to be infinite is justified for the purpose of this example.

Errors are introduced due to quantization of the mantissa. In this example magnitude truncation is used to quantize the mantissa in all arithmetic operations.

4.2.4.1 GR Model Based Implementation

The transfer function (4.35) can be realized using the GR model:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \frac{1}{16} & 0 & \frac{1}{8} & 0 \\ 0 & \frac{1}{16} & 0 & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{5}{4} & \frac{5}{32} \\ 0 & 1 & \frac{5}{4} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = [\,1\;\;0\;\;0\;\;0\,]\boldsymbol{x}(n_1,n_2,t) \tag{4.36}$$

where $\boldsymbol{x}^h \in \mathbb{R}$, $\boldsymbol{x}^v \in \mathbb{R}$ and $\boldsymbol{x}^t \in \mathbb{R}^2$. According to Theorem 4.1.1, the system (4.36) under floating point computation is GAS if and only if the system,

$$\boldsymbol{x}(t+1) = N_P\left[\begin{bmatrix} \frac{5}{4} & \frac{5}{32} \\ \frac{5}{4} & 0 \end{bmatrix} \boldsymbol{x}(t)\right] \tag{4.37}$$

is GAS. Here $\boldsymbol{x} \in \mathbb{R}^2$ and N_P is the quantization operator used for in node computations. Global asymptotic stability of systems of the form (4.37) has been studied in Bauer [1995a]. It has been shown that quantization nonlinearities can result in four fundamental response types if the system is otherwise GAS. A sufficient condition on the length of the mantissa to ensure a granular periodic response in the underflow regime is established in Bauer [1995a]. For the system (4.37) a mantissa length of 11 bits is sufficient to ensure a granular periodic response in the underflow regime.

The system (4.36) was simulated on the sensor network using floating point computations with a mantissa length of 6 bits. The quantization scheme described by model 1 was used. The only non-zero initial conditions are given by $\boldsymbol{x}(n_1,n_2,0) = [0,0,5,14]^T$ for $0 \leq n_1 \leq 3$ and $0 \leq n_1 \leq 3$. Figure 4.15 shows plots of the Euclidean norm of state vectors of nodes $(1,1)$,$(1,4)$,$(4,1)$ and $(4,4)$ versus time. In this case, states of the nodes reach the origin as time tends to infinity. Figure 4.16 illustrates plots of the

Euclidean norm of state vectors of the same nodes versus time when a mantissa length of 5 is used for computations.

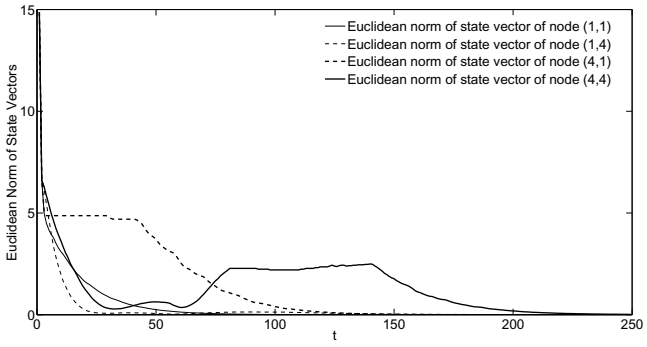


Figure 4.15. Euclidean norm of the state vectors versus time t for the GR model, when computations are done with a 6 bit mantissa

System (4.37) is not GAS when computations are done with a 5 bit mantissa. Therefore, when computations are done with a 5 bit mantissa, system (4.36) is also not GAS according to Theorem 4.2.1. Simulation results are in accordance with the theoretical findings. Simulation results, for the case when the quantization scheme described by model 2 was used, is given in figure 4.17. State vectors communicated between nodes were represented using a floating point scheme with a 5 bit mantissa. A 6 bit mantissa was used for floating point computations within the node. Figure 4.17 shows plots of the Euclidean norm of state vectors of nodes $(1,1)$,$(1,4)$,$(4,1)$ and $(4,4)$ versus time.

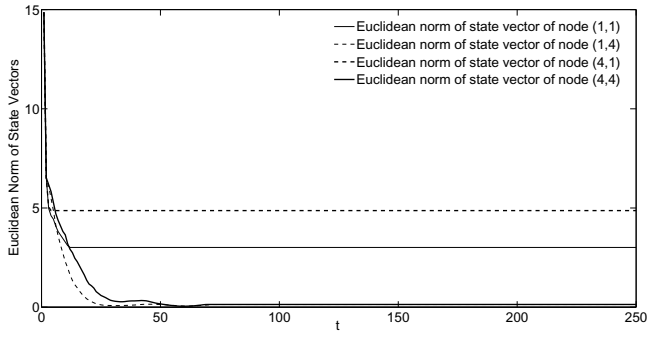


Figure 4.16. Euclidean norm of the state vectors versus time t for the GR model, when computations are done with a 5 bit mantissa

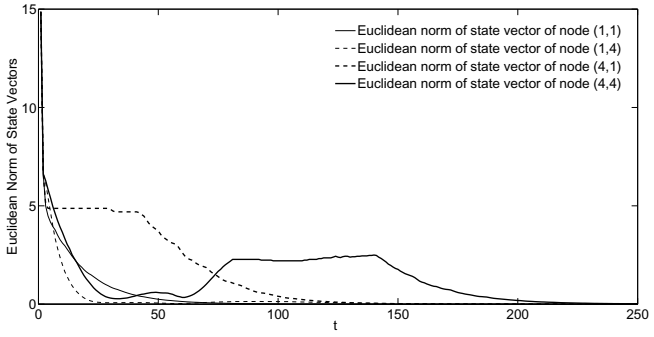


Figure 4.17. Euclidean norm of the state vectors versus time t for the GR model, when in node computations are performed with a 6 bit mantissa and communicated state vectors are quantized to a 5 bit mantissa.

In this case states of the nodes reach the origin as time tends to infinity, even though a 5 bit mantissa is used for communicated state vectors. As predicted by Theorem 4.2.1 the quantization scheme used for communicated state vectors does not affect the asymptotic stability of the system.

4.2.4.2 FM Model Based Implementation

The input output transfer function (4.35) can be realized using the FM model:

$$\boldsymbol{x}(n_1,n_2,t) = \begin{bmatrix} \frac{1}{8} & -\frac{1}{16} & \frac{1}{8} & 0 \\ \frac{1}{8} & -\frac{1}{16} & \frac{1}{8} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}(n_1-1,n_2,t) + \begin{bmatrix} -\frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{8} \\ -\frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}(n_1,n_2-1,t)$$
$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{5}{4} & \frac{5}{32} \\ -2 & 2 & \frac{5}{4} & 0 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t-1) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{u}(n_1-1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = [\ 2 \quad -1 \quad 0 \quad 0\ ]\boldsymbol{x}(n_1,n_2,t) \tag{4.39}$$

Here $\boldsymbol{x} \in \mathbb{R}^4$. According to Theorem 4.2.2, the system (4.39) under floating point computation is GAS if and only if the system,

$$\boldsymbol{x}(t+1) = N_P \left[\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{5}{4} & \frac{5}{32} \\ -2 & 2 & \frac{5}{4} & 0 \end{bmatrix} \boldsymbol{x}(t) \right] \tag{4.40}$$

is GAS. Here $\boldsymbol{x} \in \mathbb{R}^4$ and N_P is the quantization operator used for in node computations. It is evident that global asymptotic stability of system (4.40) is equivalent to that of system (4.37). The system (4.39) was simulated on the sensor network using floating point computations with a mantissa length of 6 bits. The quantization

scheme described by model 1 was used. The only non-zero initial conditions are given by $\boldsymbol{x}(n_1, n_2, 0) = [0, 0, 5, 14]^T$ for $0 \leq n_1 \leq 3$ and $0 \leq n_1 \leq 3$. Figure 4.18 shows plots of the Euclidean norm of state vectors of nodes $(1, 1)$,$(1, 4)$,$(4, 1)$ and $(4, 4)$ versus time. In this case states of the nodes reach the origin as time tends to infinity. Figure 4.19 illustrates plots of the Euclidean norm of state vectors of the same nodes versus time when a mantissa length of 5 is used for computations.

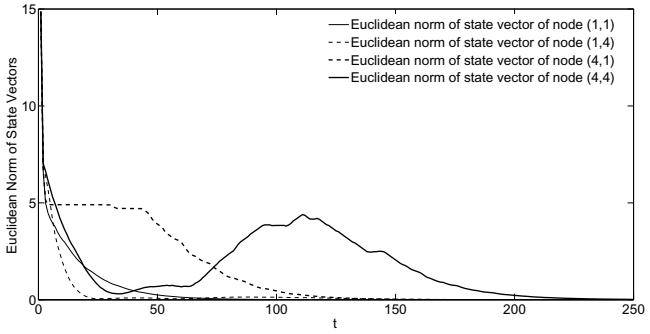


Figure 4.18. Euclidean norm of the state vectors versus time t for the FM model, when computations are done with a 6 bit mantissa

System (4.40) is not GAS when computations are done with a 5 bit mantissa. Therefore, when computations are done with a 5 bit mantissa, system (4.39) is also not GAS according to Theorem 4.2.2. Simulation results show that system (4.39) is not GAS. Simulation results, for the case when the quantization scheme described by model 2 was used, is given in figure 4.20. A floating point scheme with 5 bit mantissa was used to represent state vectors communicated between nodes. A 6 bit mantissa was used for floating point computations within the node. Figure 4.20 shows plots of the Euclidean

norm of state vectors of nodes $(1,1)$,$(1,4)$,$(4,1)$ and $(4,4)$ versus time.

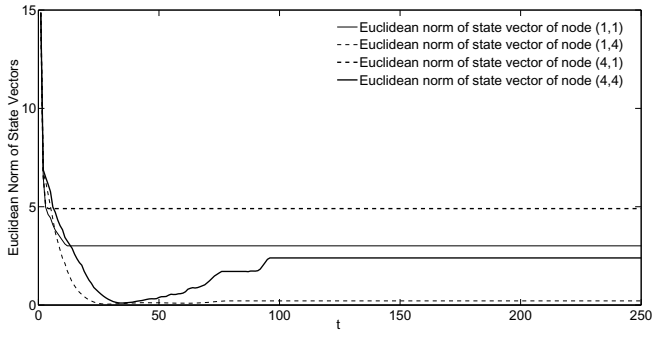


Figure 4.19. Euclidean norm of the state vectors versus time t for the FM model, when computations are done with a 5 bit mantissa

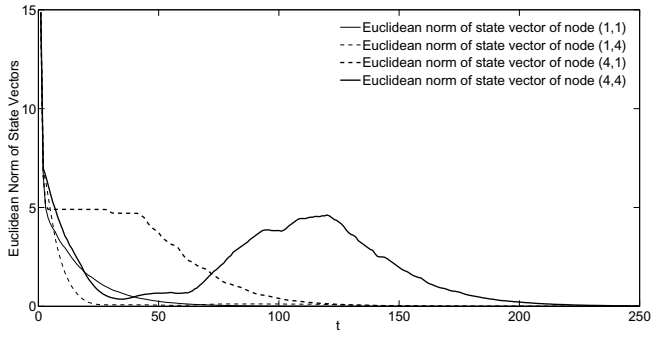


Figure 4.20. Euclidean norm of the state vectors versus time t for the FM model, when in node computations are performed with a 6 bit mantissa and communicated state vectors are quantized to a 5 bit mantissa.

In this case states of the nodes reach the origin as time tends to infinity, even though a 5 bit mantissa is used for communicated state vectors. As predicted by Theorem 4.2.2 the quantization scheme used for communicated state vectors does not affect the asymptotic stability of the system (4.39).

CHAPTER 5

NODE AND LINK FAILURE

Node and link failure is a common occurrence in sensor networks. Various aspects of sensor networks, under the occurrence of node and link failure, have been addressed in the literature. Coverage, connectivity, routing schemes and network capacity limits of sensor networks with node failure are studied in Akbar et al. [2006]; Barrenechea et al. [2004]; Shakkottai et al. [2003]. Preservation of generated data in a sensor network in case of node failures is discussed in Hamed Azimi et al. [2010]. A topology management scheme is proposed for sensor networks with node failure in Frye et al. [2006]. In Imamoglu and Keskinoz [2010] serial distributed detection in wireless sensor networks with node failure is studied. Impact of node failures and unreliable communication on decentralized detection in sensor networks is discussed in Tay et al. [2008].

State and input vectors of sensor nodes have to be communicated between nodes to implement the FM and GR models in sensor networks. A node may not receive information required for its computations if a neighboring node or the communication link with the same fails. In order to ensure uninterrupted functioning of the sensor network, when information required for a computation is not received by a node it has to carry on its operation with the information it can acquire. An approach to mathematically model such behavior is to assume that nodes estimate missing information with the information it can acquire. In this work, the said approach is used to extend FM and GR models for 3-D systems to include node and link failure. Resulting system

models are stochastic due to randomness of node and link failure. In this work, asymptotic and input-output stability criteria are proposed for sensor networks under node and link failure. Node failure and communication link failure are treated as separate cases throughout this work.

5.1 Models for 3-D Systems Under Link Failure

Let $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ be random processes such that:

$$q_1(n_1, n_2, t) = \begin{cases} 0 & communication\ link\ between\ nodes\ (n_1, n_2) \\ & and\ (n_1 - 1, n_2)\ failed\ at\ time\ t \\ 1 & otherwise \end{cases} \tag{5.1}$$

$$q_2(n_1, n_2, t) = \begin{cases} 0 & communication\ link\ between\ nodes\ (n_1, n_2) \\ & and\ (n_1, n_2 - 1)\ failed\ at\ time\ t \\ 1 & otherwise \end{cases} \tag{5.2}$$

Random processes $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ are success-failure processes of data blocks communicated between node (n_1, n_2) and its immediate predecessors. The node at the origin does not require any information from other nodes to perform its computations. Nodes on the axis receive information from only one neighboring node. Therefore $q_1(n_1, n_2, t) = 1$ for $n_1 = 0$ and $q_2(n_1, n_2, t) = 1$ for $n_2 = 0$.

5.1.1 FM Model

Node (n_1, n_2) receives state and input vectors of nodes $(n_1 - 1, n_2)$ and $(n_1, n_2 - 1)$. It is assumed that if there is a failure in the communication link between node (n_1, n_2)

and any of its immediate predecessors, node (n_1, n_2) receives neither the state vector nor the input vector from that node. When node (n_1, n_2) does not receive state or input vectors required for a computation it estimates the missing vectors. Let $\hat{\boldsymbol{x}}_{n_1 n_2}(n_1 - i, n_2 - j, t)$ and $\hat{\boldsymbol{u}}_{n_1 n_2}(n_1 - i, n_2 - j, t)$ be estimates of $\boldsymbol{x}(n_1 - i, n_2 - j, t)$ and $\boldsymbol{u}(n_1 - i, n_2 - j, t)$ made by node (n_1, n_2) at time t. The FM model for distributed 3-D systems under link failure is given by:

$$\begin{aligned}
\boldsymbol{x}(n_1, n_2, t) &= q_1(n_1, n_2, t)\{\boldsymbol{A}_h \boldsymbol{x}(n_1\text{–}1, n_2, t) + \boldsymbol{B}_h \boldsymbol{u}(n_1\text{–}1, n_2, t)\} \\
&\quad + (1 - q_1(n_1, n_2, t))\{\boldsymbol{A}_h \hat{\boldsymbol{x}}_{n_1 n_2}(n_1\text{–}1, n_2, t) + \boldsymbol{B}_h \hat{\boldsymbol{u}}_{n_1 n_2}(n_1\text{–}1, n_2, t)\} \\
&\quad + q_2(n_1, n_2, t)\{\boldsymbol{A}_v \boldsymbol{x}(n_1, n_2\text{–}1, t) + \boldsymbol{B}_v \boldsymbol{u}(n_1, n_2\text{–}1, t)\} \\
&\quad + (1 - q_2(n_1, n_2, t))\{\boldsymbol{A}_v \hat{\boldsymbol{x}}_{n_1 n_2}(n_1, n_2\text{–}1, t) + \boldsymbol{B}_v \hat{\boldsymbol{u}}_{n_1 n_2}(n_1, n_2\text{–}1, t)\} \\
&\quad + \boldsymbol{A}_t \boldsymbol{x}(n_1, n_2, t\text{–}1) + \boldsymbol{B}_t \boldsymbol{u}(n_1, n_2, t\text{–}1) \\
\boldsymbol{y}(n_1, n_2, t) &= \boldsymbol{C}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t)
\end{aligned} \tag{5.3}$$

Here $\boldsymbol{x}(n_1, n_2, t)$, $\boldsymbol{y}(n_1, n_2, t)$ and $\boldsymbol{u}(n_1, n_2, t)$ are the state vector, output vector and input vector of node (n_1, n_2) at time t respectively. Let the input vector $\boldsymbol{u} \in \mathbb{R}^p$, output vector $\boldsymbol{y} \in \mathbb{R}^q$ and state vector $\boldsymbol{x} \in \mathbb{R}^n$. Then $\hat{\boldsymbol{u}} \in \mathbb{R}^p$,$\hat{\boldsymbol{x}} \in \mathbb{R}^n$, $\boldsymbol{C} \in \mathbb{R}^{q \times n}$, $\boldsymbol{D} \in \mathbb{R}^{q \times p}$, $\boldsymbol{A}_h \in \mathbb{R}^{n \times n}$, $\boldsymbol{A}_v \in \mathbb{R}^{n \times n}$, $\boldsymbol{A}_t \in \mathbb{R}^{n \times n}$, $\boldsymbol{B}_h \in \mathbb{R}^{n \times p}$, $\boldsymbol{B}_v \in \mathbb{R}^{n \times p}$ and $\boldsymbol{B}_t \in \mathbb{R}^{n \times p}$. For a sensor network of size $N_1 \times N_2$, $n_1 \in [0, N_1 - 1]$, $n_2 \in [0, N_2 - 1]$ and $t \in [0, \infty)$.

5.1.2 GR Model

In the GR model based implementation, node (n_1, n_2) receives its horizontal and vertical state vector components from nodes $(n_1 - 1, n_2)$ and $(n_1, n_2 - 1)$ respectively. Input vectors of other nodes are not required for computations and hence not required to be estimated. Let estimates of $\boldsymbol{x}^h(n_1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2, t)$ made by node (n_1, n_2)

at time t be denoted by $\hat{\boldsymbol{x}}^h(n_1, n_2, t)$ and $\hat{\boldsymbol{x}}^v(n_1, n_2, t)$ respectively. Let:

$$\tilde{\boldsymbol{x}}^h(n_1, n_2, t) = q_1(n_1, n_2, t)\boldsymbol{x}^h(n_1, n_2, t) + (1 - q_1(n_1, n_2, t))\hat{\boldsymbol{x}}^h(n_1, n_2, t)$$
$$\tilde{\boldsymbol{x}}^v(n_1, n_2, t) = q_2(n_1, n_2, t)\boldsymbol{x}^v(n_1, n_2, t) + (1 - q_2(n_1, n_2, t))\hat{\boldsymbol{x}}^v(n_1, n_2, t) \quad (5.4)$$

The GR model for distributed 3-D systems under link failure is given by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1, n_2, t) \\ \boldsymbol{x}^v(n_1, n_2+1, t) \\ \boldsymbol{x}^t(n_1, n_2, t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}^h(n_1, n_2, t) \\ \tilde{\boldsymbol{x}}^v(n_1, n_2, t) \\ \boldsymbol{x}^t(n_1, n_2, t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1, n_2, t)$$
$$\boldsymbol{y}(n_1, n_2, t) = \boldsymbol{C}\tilde{\boldsymbol{x}}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t) \quad (5.5)$$

where $\tilde{\boldsymbol{x}}(n_1, n_2, t) = (\tilde{\boldsymbol{x}}^{h^T}(n_1, n_2, t), \tilde{\boldsymbol{x}}^{v^T}(n_1, n_2, t), \boldsymbol{x}^{t^T}(n_1, n_2, t))^T$. Let the input vector $\boldsymbol{u} \in \mathbb{R}^p$ and output vector $\boldsymbol{y} \in \mathbb{R}^q$. Then $\tilde{\boldsymbol{x}}^h \in \mathbb{R}^a$, $\tilde{\boldsymbol{x}}^v \in \mathbb{R}^b$,$\hat{\boldsymbol{x}}^h \in \mathbb{R}^c$,$\hat{\boldsymbol{x}}^v \in \mathbb{R}^b$, $\boldsymbol{A}_1 \in \mathbb{R}^{a\times a}$, $\boldsymbol{A}_2 \in \mathbb{R}^{a\times b}$, $\boldsymbol{A}_3 \in \mathbb{R}^{a\times c}$, $\boldsymbol{A}_4 \in \mathbb{R}^{b\times a}$, $\boldsymbol{A}_5 \in \mathbb{R}^{b\times b}$, $\boldsymbol{A}_6 \in \mathbb{R}^{b\times c}$, $\boldsymbol{A}_7 \in \mathbb{R}^{c\times a}$, $\boldsymbol{A}_8 \in \mathbb{R}^{c\times b}$, $\boldsymbol{A}_9 \in \mathbb{R}^{c\times c}$, $\boldsymbol{B}_1 \in \mathbb{R}^{a\times p}$, $\boldsymbol{B}_2 \in \mathbb{R}^{b\times p}$, $\boldsymbol{B}_3 \in \mathbb{R}^{c\times p}$, $\boldsymbol{C} \in \mathbb{R}^{q\times(a+b+c)}$ and $\boldsymbol{D} \in \mathbb{R}^{q\times p}$. For a sensor network of size $N_1 \times N_2$, $n_1 \in [0, N_1 - 1]$, $n_2 \in [0, N_2 - 1]$ and $t \in [0, \infty)$.

5.2 Asymptotic Stability under Link Failure

In this work asymptotic stability of systems (5.3) and (5.5) is studied. Systems (5.3) and (5.5) are stochastic systems. Therefore a stochastic notion of asymptotic stability has to be employed. Mean square stability of systems (5.3) and (5.5) is studied in this work. An important consideration that affects stability and the performance of the systems is how the missing vectors are estimated by nodes. In this work it is assumed that all estimates made are linear combinations of known state and input

vectors. It is also assumed that estimations are time invariant. Further it is assumed that for an estimate made by node (n_1, n_2) at time t, the following is true.

1. Only state and input vectors from its neighboring nodes are used.

2. State and input vectors at time less than $t-1$ are not used.

3. State vectors of its succeeding neighbors are not used.

Failure of a communication link between two nodes does not generate an additional information routing burden on the sensor network due to condition (1). This condition can be lifted if there is a means to acquire information from non neighboring nodes when a communication link fails. Results derived below can be extended in a straightforward manner to the case, where conditions (1) and (2) are lifted but a finite number of state and input vectors are included in the estimation. Condition (3) is necessitated due to causality and computability considerations. At time t, any of the available vectors from the set $S(n_1, n_2, t)$, defined in (5.6), can be used by node (n_1, n_2) to estimate the missing information.

$$\begin{aligned} S(n_1, n_2, t) = \{&\boldsymbol{x}(n_1, n_2, t-1), \boldsymbol{x}(n_1-1, n_2, t-1), \boldsymbol{x}(n_1, n_2-1, t-1), \\ &\boldsymbol{x}(n_1-1, n_2, t), \boldsymbol{x}(n_1, n_2-1, t), \boldsymbol{u}(n_1, n_2, t-1), \\ &\boldsymbol{u}(n_1-1, n_2, t-1), \boldsymbol{u}(n_1, n_2-1, t-1), \boldsymbol{u}(n_1+1, n_2, t-1), \\ &\boldsymbol{u}(n_1, n_2+1, t-1), \boldsymbol{u}(n_1, n_2, t), \boldsymbol{u}(n_1-1, n_2, t), \\ &\boldsymbol{u}(n_1, n_2-1, t), \boldsymbol{u}(n_1+1, n_2, t), \boldsymbol{u}(n_1, n_2+1, t)\} \end{aligned} \tag{5.6}$$

Since we are interested in asymptotic stability, the input to all the nodes is assumed to be zero for $t \geq 0$.

Definition 5.2.1 *The system is said to be mean square stable if:*

$$\lim_{t\to\infty} \|\boldsymbol{x}(n_1, n_2, t)\| = 0$$

$\forall(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$,*where:*

$$\|\boldsymbol{x}(n_1, n_2, t)\| = \sqrt{E(\boldsymbol{x}(n_1, n_2, t)^T \boldsymbol{x}(n_1, n_2, t))}$$

and the operator E denotes the expectation of a random variable. It is assumed that $\boldsymbol{x}(-1, n_2, t) = 0$, $\boldsymbol{x}(n_1, -1, t) = 0$ and the only non-zero boundary conditions are given by $\boldsymbol{x}(n_1, n_2, 0)$.

Mean square stability of the systems (5.3) and (5.5) depends on the statistical properties of random processes $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ which switch the system between its modes of operation. Statistical properties of the communication channel between nodes in the sensor network determine the statistical properties of the success-failure process of data blocks, $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$. A set of models that map attributes such as transmission power and the distance between nodes to packet reception probabilities is given in Cerpa et al. [2005]. A channel model that describes path loss in near-to-ground communication links in wireless sensor networks is given in Martinez-Sala et al. [2005]. First order Markovian chains have been proposed in the literature to model received signal strength of fading radio communication links Tan and Beaulieu [2000]; Turin and van Nobelen [1998]; Wang and Chang [1996] Wang and Moayeri [1995]. A finite state first order Markovian model is proposed for Rayleigh fading channels in Wang and Moayeri [1995]. Wang and Chang [1996] propose a mutual information metric to demonstrate that first order finite state Markov chain models are sufficient to model slow fading channels. Higher order finite state

Markovian models for fading communication channels are examined in Babich et al. [1997]; Turin and van Nobelen [1998]. Using a mutual information metric similar to the one used in Wang and Chang [1996], a binary first order Markov model is demonstrated to be sufficient to model the success failure process of data blocks over fading communication links in Zorzi et al. [1995]. Methods to determine the parameters of the Markov model are also discussed in Zorzi et al. [1995].

Therefore the success-failure process of data blocks can be assumed to be binary Markovian. In order to make the analysis general enough to accommodate higher order Makovian models for the success-failure process of data blocks, it is assumed that processes $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ are Markovian processes of order N. Therefore given $\{q_1(n_1, n_2, t-d) : 0 < d \leq N\}$ and $\{q_2(n_1, n_2, t-d) : 0 < d \leq N\}$, $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ are independent of $\{q_1(n_1, n_2, t-d) : d > N\}$ and $\{q_2(n_1, n_2, t-d) : d > N\}$.

5.2.1 FM Model

For notational simplicity let:

$$\boldsymbol{A}_h \hat{\boldsymbol{x}}_{n_1 n_2}(n_1\text{–}1, n_2, t) + \boldsymbol{B}_h \hat{\boldsymbol{u}}_{n_1 n_2}(n_1\text{–}1, n_2, t) = \boldsymbol{L}^1_{q_2(n_1,n_2,t)}(n_1, n_2, t) + \boldsymbol{K}^1_{q_2(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}(n_1, n_2, t-1) \tag{5.7}$$

$$\boldsymbol{A}_v \hat{\boldsymbol{x}}_{n_1 n_2}(n_1, n_2\text{–}1, t) + \boldsymbol{B}_v \hat{\boldsymbol{u}}_{n_1 n_2}(n_1, n_2\text{–}1, t) = \boldsymbol{L}^2_{q_1(n_1,n_2,t)}(n_1, n_2, t) + \boldsymbol{K}^2_{q_1(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}(n_1, n_2, t-1) \tag{5.8}$$

In the above, $\boldsymbol{L}^1_{q_2(n_1,n_2,t)}(n_1, n_2, t)$ and $\boldsymbol{L}^2_{q_1(n_1,n_2,t)}(n_1, n_2, t)$ denote linear combinations of the elements of $S(n_1, n_2, t)$ except $\boldsymbol{x}(n_1, n_2, t-1)$. Subscripts $q_1(n_1, n_2, t)$ and

$q_2(n_1, n_2, t)$ are used to indicate that weights given to available vectors in the estimation may depend upon random variables $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$. Consider the set of systems described by the state space model:

$$\begin{aligned}\boldsymbol{x}(n_1, n_2, t) = \boldsymbol{A}_t\boldsymbol{x}(n_1, n_2, t–1) + (1 - q_1(n_1, n_2, t))\boldsymbol{K}^1_{q_2(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}(n_1, n_2, t-1) \\ + (1 - q_2(n_1, n_2, t))\boldsymbol{K}^2_{q_1(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}(n_1, n_2, t-1) \qquad (5.9)\end{aligned}$$

where $(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$. Though the same notation as (5.3) is used for state vectors, equation (5.9) describes a set of 1-D systems which is not as same as the 3-D system (5.3). System matrices of systems described by (5.9) are subjected to random variations. Such systems are called jump linear systems. Systems described by (5.9) can have one, two or four modes of operation, and the mode of operation at time t is determined by random variables $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$. Stochastic stability of jump linear systems is examined in Bolzern et al. [2004]; Costa and Fragoso [1993]; Feng et al. [1992]; Kubrusly and Costa [1985]; Tejada et al. [2005]and references therein. A necessary and sufficient condition for mean square stability of discrete time stochastic bilinear systems is presented in Kubrusly and Costa [1985]. In Feng et al. [1992] a necessary and sufficient condition for mean square stability of continuous time Markovian jump linear systems is presented. Costa and Fragoso [1993] give a necessary and sufficient condition for mean square stability of discrete time Markovian jump linear systems.

Define an integer valued random process $\Psi(n_1, n_2, t)$ such that:

$$\Psi(n_1, n_2, t) = \sum_{i=Nt}^{(t+1)N-1} \{2q_2(n_1, n_2, i) + q_1(n_1, n_2, i)\}2^{2i-2Nt} \qquad (5.10)$$

Given $\Psi(n_1, n_2, t)$, $\{q_1(n_1, n_2, i) : Nt \leq i < N(t + 1)\}$ and $\{q_2(n_1, n_2, i) : Nt \leq i <$

$N(t+1)\}$ can be determined. It is easily seen that given $\Psi(n_1, n_2, t-1)$, $\Psi(n_1, n_2, t)$ is independent of $\{\Psi(n_1, n_2, t-d) : d \geq 2\}$. Therefore $\Psi(n_1, n_2, t)$ is a Markovian random process. Furthermore $\Psi(n_1, n_2, t) \in \mathbb{Z} \cap [0, 4^N - 1]$. Let the set of values that $\Psi(n_1, n_2, t)$ can take be denoted by $S_\Psi(n_1, n_2)$ and the number of distinct values $\Psi(n_1, n_2, t)$ can take be denoted by $N_\Psi(n_1, n_2)$. Let the transition probability matrix of the Markovian random process $\Psi(n_1, n_2, t)$ be $\mathbb{P}(n_1, n_2)$.

Lemma 5.2.1 *Let:*

$$\begin{aligned}\boldsymbol{A}^F_{q_1(n_1,n_2,t),q_2(n_1,n_2,t)}(n_1, n_2) = \boldsymbol{A}_t + (1 - q_1(n_1, n_2, t))\boldsymbol{K}^1_{q_2(n_1,n_2,t)}(n_1, n_2) \\ + (1 - q_2(n_1, n_2, t))\boldsymbol{K}^2_{q_1(n_1,n_2,t)}(n_1, n_2)\end{aligned} \tag{5.11}$$

$$\boldsymbol{A}^F_{\Psi(n_1,n_2,t)}(n_1, n_2) = \prod_{i=Nt}^{N(t+1)-1} \boldsymbol{A}^F_{q_1(n_1,n_2,i),q_2(n_1,n_2,i)}(n_1, n_2) \tag{5.12}$$

Then the system:

$$\bar{\boldsymbol{x}}(n_1, n_2, t+1) = \boldsymbol{A}^F_{\Psi(n_1,n_2,t)}(n_1, n_2)\bar{\boldsymbol{x}}(n_1, n_2, t) \tag{5.13}$$

is a Markovian jump linear systems. Furthermore the system (5.13) *is mean square stable if and only if the system* (5.9) *is mean square stable.*

Proof System (5.13) is switched between its modes of operation by the Markovian random process $\Psi(n_1, n_2, t)$. Hence it is a Markovian jump linear system. If the systems (5.9) and (5.13) have the same initial conditions state vector of system (5.13) at time t is equal to the state vector of system (5.9) at time Nt. Hence the system (5.13) is mean square stable if and only if the system (5.9) is mean square stable.

Lemma 5.2.2 *If a system described by* (5.9) *is mean square stable* $\exists\ \mu_{n_1n_2} > 0, \lambda_{n_1n_2} \in [0,1)$ *and* $k_{n_1n_2} \in \mathbb{Z}^+$ *such that* $\|\boldsymbol{x}(n_1,n_2,t)\| < \mu_{n_1n_2} t^{k_{n_1n_2}} \lambda_{n_1n_2}^t \|\boldsymbol{x}(n_1,n_2,0)\|$.

Proof Assume that systems described by (5.9) are mean square stable. The system (5.13) is also mean square stable. According to the proposition 8 in Costa and Fragoso [1993] root mean square norm of the state vector of system (5.13) bounded as below:

$$\|\boldsymbol{x}(n_1,n_2,t)\| < \tilde{\mu}_{n_1n_2} t^{\tilde{k}_{n_1n_2}} \tilde{\lambda}_{n_1n_2}^t \|\boldsymbol{x}(n_1,n_2,0)\|$$

where $\tilde{\mu}_{n_1n_2} > 0, \tilde{\lambda}_{n_1n_2} \in [0,1)$ and $\tilde{k}_{n_1n_2} \in \mathbb{Z}^+$. If systems (5.9) and (5.13) have the same initial conditions state vector of system (5.13) at time t is equal to the state vector of system (5.9) at time Nt. Hence root mean square norm of the state vector of system (5.9) has and upper bound of the form given above.

Theorem 5.2.1 *The system* (5.3) *is mean square stable, if and only if systems described by* (5.9) *are mean square stable* $\forall (n_1,n_2) \in [0, N_1-1] \times [0, N_2-1]$.

Proof The proof of necessity is trivial. To prove sufficiency it is assumed that, $\forall (n_1,n_2) \in [0, N_1-1] \times [0, N_2-1]$ systems described by (5.9) are mean square stable. Since, for $(n_1,n_2) = (0,0)$, the system described by (5.9) is mean square stable $\exists$ constants $\mu_{00} > 0, \lambda_{00} \in [0,1)$ and $k_{00} \in \mathbb{Z}^+$ such that $\|\boldsymbol{x}(0,0,t)\| < \mu_{00} t^{k_{00}} \lambda_{00}^t \|\boldsymbol{x}(0,0,0)\|$. Therefore:

$$\lim_{t\to\infty} \|\boldsymbol{x}(0,0,t)\| = 0$$

$$\boldsymbol{x}(0,1,t) = \left(\prod_{i=0}^{t-1} \boldsymbol{A}^F_{q_1(0,1,i),q_2(0,1,i)}(0,1)\right)\boldsymbol{x}(0,1,0) + \sum_{k=0}^{t}\left(\prod_{i=k}^{t-1} \boldsymbol{A}^F_{q_1(0,1,i),q_2(0,1,i)}(0,1)\right) q_1(0,1,k)\boldsymbol{A}_h\boldsymbol{x}(0,0,k) + \sum_{k=0}^{t-1}\left(\prod_{i=k+1}^{t-1} \boldsymbol{A}^F_{q_1(0,1,i),q_2(0,1,i)}(0,1)\right)\boldsymbol{G}^1(0,1,k)\boldsymbol{x}(0,0,k)$$

If the state vector $\boldsymbol{x}(0,0,t)$ has not been received by node $(0,1)$ at time t it may use the state vector $\boldsymbol{x}(0,0,t-1)$ to estimate the former state vector if $\boldsymbol{x}(0,0,t-1)$ is available. Therefore $\boldsymbol{x}(0,0,t-1)$ may be used by node $(0,1)$ at time t, in computing $\boldsymbol{x}(0,1,t)$ depending upon random variables $q_1(0,1,t)$ and $q_1(0,1,t-1)$. Matrix $\boldsymbol{G}^1(0,1,k) \in \mathbb{R}^{n\times n}$ is of the form:

$$\boldsymbol{G}^1(0,1,k) = (1-q_1(0,1,k+1))q_1(0,1,k)\boldsymbol{A}_h\boldsymbol{W}^1(0,1)$$

Here $\boldsymbol{W}^1(0,1) \in \mathbb{R}^{n\times n}$ is the weight given to $\boldsymbol{x}(0,0,k-1)$ by node $(0,1)$ in estimating $\boldsymbol{x}(0,1,k)$. We have:

$$\|\boldsymbol{x}(0,1,t)\| \leq \left\|\left(\prod_{i=0}^{t-1} \boldsymbol{A}^F_{q_1(0,1,i),q_2(0,1,i)}(0,1)\right)\boldsymbol{x}(0,1,0)\right\| + \sum_{k=0}^{t}\left\|\left(\prod_{i=k}^{t-1} \boldsymbol{A}^F_{q_1(0,1,i),q_2(0,1,i)}(0,1)\right) q_1(0,1,k)\boldsymbol{A}_h\boldsymbol{x}(0,0,k)\right\| + \sum_{k=0}^{t-1}\left\|\left(\prod_{i=k+1}^{t-1} \boldsymbol{A}^F_{q_1(0,1,i),q_2(0,1,i)}(0,1)\right)\boldsymbol{G}^1(0,1,k)\boldsymbol{x}(0,0,k)\right\|$$

Since, for $(n_1, n_2) = (0, 1)$, the system described by (5.9) is mean square stable:

$$\begin{aligned}\|\boldsymbol{x}(0,1,t)\| \leq\ & \mu_{01} t^{k_{01}} \lambda_{01}^{t} \|\boldsymbol{x}(0,1,0)\| \\ & + \sum_{k=0}^{t} \mu_{01}(t-k)^{k_{01}} \lambda_{01}^{t-k} \mu_{00} k^{k_{00}} \lambda_{00}^{k} \sigma(\boldsymbol{A}_h) \|\boldsymbol{x}(0,0,0)\| \\ & + \sum_{k=0}^{t} \mu_{01}(t-k-1)^{k_{01}} \lambda_{01}^{t-k-1} \mu_{00} k^{k_{00}} \lambda_{00}^{k} \sigma(\boldsymbol{G}^1(0,1,k)) \|\boldsymbol{x}(0,0,0)\|\end{aligned}$$

Therefore:

$$\lim_{t\to\infty} \|\boldsymbol{x}(0,1,t)\| = 0$$

Following the same reasoning it can be shown that $\|\boldsymbol{x}(0,2,t)\|$ tends to zero as time t tends to infinity. Furthermore using the same reasoning iteratively it can be shown that, $\forall (n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$:

$$\lim_{t\to\infty} \|\boldsymbol{x}(n_1,n_2,t)\| = 0$$

Therefore the system (5.3) is mean square stable.

Mean square stability of the distributed 3-D system (5.3) is equivalent to that of a set of 1-D systems described by (5.9).

Lemma 5.2.3 *Let:*

$$\boldsymbol{H}^F(n_1,n_2) = diag_{i\in S_\Psi(n_1,n_2)}(\boldsymbol{A}_i^F(n_1,n_2) \otimes \boldsymbol{A}_i^F(n_1,n_2))$$

and

$$\boldsymbol{A}^{\mathcal{F}}(n_1,n_2) = (\mathbb{P}^T(n_1,n_2) \otimes \boldsymbol{I}_{n^2}) \boldsymbol{H}^F(n_1,n_2)$$

Here $\boldsymbol{H}^F(n_1,n_2) \in \mathbb{R}^{N_\Psi(n_1,n_2)n^2 \times N_\Psi(n_1,n_2)n^2}$, $\boldsymbol{A}^{\mathcal{F}} \in \mathbb{R}^{N_\Psi(n_1,n_2)n^2 \times N_\Psi(n_1,n_2)n^2}$ *and oper-*

ator $\otimes$ denotes the Kronecker product of two matrices. System (5.13) *is mean square stable if and only if the spectral radius of the matrix $\boldsymbol{A}^{\mathcal{F}}(n_1, n_2)$, $\rho(\boldsymbol{A}^{\mathcal{F}}(n_1, n_2)) < 1$.*

Proof The lemma 5.2.3 is a direct result of proposition 8 in Costa and Fragoso [1993].

Theorem 5.2.2 *System* (5.3) *is mean square stable if and only if the spectral radius of the matrix $\boldsymbol{A}^{\mathcal{F}}(n_1, n_2)$, $\rho(\boldsymbol{A}^{\mathcal{F}}(n_1, n_2)) < 1 \; \forall (n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$.*

Proof The result follows directly from Lemma 5.2.1, Theorem 5.2.1 and Lemma 5.2.3.

For FM model based implementations checking the mean square stability of the sensor network involves evaluating the spectral radius of a matrix or a set of matrices. Since $\boldsymbol{A}^{\mathcal{F}} \in \mathbb{R}^{N_\Psi(n_1,n_2)n^2 \times N_\Psi(n_1,n_2)n^2}$ computational complexity of computing the spectral radius is dependant on:

- n, the size of the state vector.
- N the order of the Markovian processes $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$.

5.2.2 GR Model

For notational simplicity let:

$$
\begin{aligned}
\hat{\boldsymbol{x}}^h(n_1, n_2, t) &= \boldsymbol{L}^1_{q_2(n_1,n_2,t)}(n_1, n_2, t) + \boldsymbol{K}^{1h}_{q_2(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}^h(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{1v}_{q_2(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}^v(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{1t}_{q_2(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}^t(n_1, n_2, t-1) \\
\hat{\boldsymbol{x}}^v(n_1, n_2, t) &= \boldsymbol{L}^2_{q_1(n_1,n_2,t)}(n_1, n_2, t) + \boldsymbol{K}^{2h}_{q_1(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}^h(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{2v}_{q_1(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}^v(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{2t}_{q_1(n_1,n_2,t)}(n_1, n_2)\boldsymbol{x}^t(n_1, n_2, t-1)
\end{aligned} \tag{5.14}
$$

In the above, $\boldsymbol{L}^1_{q_2(n_1,n_2,t)}(n_1,n_2,t)$ and $\boldsymbol{L}^2_{q_1(n_1,n_2,t)}(n_1,n_2,t)$ denote linear combinations of the elements of $S(n_1,n_2,t)$ except $\boldsymbol{x}(n_1,n_2,t-1)$. Subscripts $q_1(n_1,n_2,t)$ and $q_2(n_1,n_2,t)$ indicate that weights given to available vectors in the estimation may depend upon random variables $q_1(n_1,n_2,t)$ and $q_2(n_1,n_2,t)$.

Consider the set of systems described by the state space model:

$$\begin{aligned}\boldsymbol{x}(n_1,n_2,t) = \boldsymbol{A}_9\boldsymbol{x}(n_1,n_2,t\text{–}1)\\ + (1-q_1(n_1,n_2,t))\boldsymbol{A}_7\boldsymbol{K}^{1t}_{q_2(n_1,n_2,t)}(n_1,n_2)\boldsymbol{x}(n_1,n_2,t-1)\\ + (1-q_2(n_1,n_2,t))\boldsymbol{A}_8\boldsymbol{K}^{2t}_{q_1(n_1,n_2,t)}(n_1,n_2)\boldsymbol{x}(n_1,n_2,t-1)\end{aligned} \tag{5.15}$$

where $(n_1,n_2)\in[0,N_1-1]\times[0,N_2-1]$.

Lemma 5.2.4 *Let:*

$$\begin{aligned}\boldsymbol{A}^R_{q_1(n_1,n_2,t),q_2(n_1,n_2,t)}(n_1,n_2) = \boldsymbol{A}_9 + (1-q_1(n_1,n_2,t))\boldsymbol{A}_7\boldsymbol{K}^{1t}_{q_2(n_1,n_2,t)}(n_1,n_2)\\ + (1-q_2(n_1,n_2,t))\boldsymbol{A}_8\boldsymbol{K}^{2t}_{q_1(n_1,n_2,t)}(n_1,n_2)\end{aligned} \tag{5.16}$$

$$\boldsymbol{A}^R_{\Psi(n_1,n_2,t)}(n_1,n_2) = \prod_{i=Nt}^{N(t+1)-1}\boldsymbol{A}^R_{q_1(n_1,n_2,i),q_2(n_1,n_2,i)}(n_1,n_2) \tag{5.17}$$

Then the system

$$\bar{\boldsymbol{x}}(n_1,n_2,t+1) = \boldsymbol{A}^R_{\Psi(n_1,n_2,t)}(n_1,n_2)\bar{\boldsymbol{x}}(n_1,n_2,t) \tag{5.18}$$

is a Markovian jump linear system. Furthermore the system (5.18) *is mean square stable if and only if the system* (5.15) *is mean square stable.*

Proof The proof follows the same line of argument as that of lemma 5.2.1 and is

omitted for the sake of brevity.

Lemma 5.2.5 *If a system described by* (5.15) *is mean square stable,* $\exists$ *constants* $\mu_{n_1n_2} > 0, \lambda_{n_1n_2} \in [0, 1)$ *and* $k_{n_1n_2} \in \mathbb{Z}^+$ *such that:*

$$\|\boldsymbol{x}(n_1, n_2, t)\| < \mu_{n_1n_2} t^{k_{n_1n_2}} \lambda_{n_1n_2}^t \|\boldsymbol{x}(n_1, n_2, 0)\|$$

Proof The proof follows the same line of argument as that of lemma 5.2.2 and would be omitted for the sake of brevity.

Theorem 5.2.3 *The system* (5.5) *is mean square stable, if and only if systems described by* (5.15)*, are mean square stable* $\forall (n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$.

Proof The proof is similar to the proof of Theorem 5.2.1.

Lemma 5.2.6 *Let:*

$$\boldsymbol{H}^R(n_1, n_2) = diag_{i \in S_\Psi(n_1,n_2)}(\boldsymbol{A}_i^R(n_1, n_2) \otimes \boldsymbol{A}_i^R(n_1, n_2))$$

and

$$\boldsymbol{A}^{\mathcal{R}}(n_1, n_2) = (\mathbb{P}^T(n_1, n_2) \otimes \boldsymbol{I}_{c^2})\boldsymbol{H}^R(n_1, n_2)$$

Here $\boldsymbol{H}^R(n_1, n_2) \in \mathbb{R}^{N_\Psi(n_1,n_2)c^2 \times N_\Psi(n_1,n_2)c^2}$ *and* $\boldsymbol{A}^{\mathcal{R}} \in \mathbb{R}^{N_\Psi(n_1,n_2)c^2 \times N_\Psi(n_1,n_2)c^2}$. *System* (5.18) *is mean square stable if and only if the spectral radius of the matrix* $\boldsymbol{A}^{\mathcal{R}}(n_1, n_2)$*,* $\rho(\boldsymbol{A}^{\mathcal{R}}(n_1, n_2)) < 1$.

Proof The lemma 5.2.6 is a direct result of proposition 8 in Costa and Fragoso [1993].

Theorem 5.2.4 *System* (5.5) *is mean square stable if and only if the spectral radius of the matrix* $\boldsymbol{A}^{\mathcal{R}}(n_1, n_2)$*,* $\rho(\boldsymbol{A}^{\mathcal{R}}(n_1, n_2)) < 1$*,* $\forall (n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$.

Proof The result follows directly from Lemma 5.2.4, Theorem 5.2.3 and and Lemma 5.2.6.

For GR model based implementations checking the mean square stability of the sensor network involves evaluating the spectral radius of a matrix or a set of matrices. Since $\boldsymbol{A}^{\mathcal{R}} \in \mathbb{R}^{N_\Psi(n_1,n_2)c^2 \times N_\Psi(n_1,n_2)c^2}$ computational complexity of computing the spectral radius is dependant on:

- c, the size of the temporal vector component.
- N the order of the Markovian processes $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$.

5.2.3 Example

Stability of an implementation of a linear filter on a grid sensor network is discussed to illustrate the theoretical results. Let the transfer function of the single input single output filter be as follows:

$$H(z_1, z_2, z_t) = \frac{z_t^{-1}}{1 - az_t^{-1} - bz_1^{-1}z_t^{-1} - cz_2^{-1}z_t^{-1}} \tag{5.19}$$

Success/failure processes of data blocks communicated between node (n_1, n_2) and its immediate predecessors, $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ are assumed to be first order Markovian processes. Furthermore it is assumed that random processes $q_1(n_1, n_2, t)$ and $q_2(n_1, n_2, t)$ are independent. For $n_2 \in [0, N_1 - 1]$, $q_1(0, n_2, t) = 1$ and for $n_1 \in [0, N_1-1]$, $q_2(n_1, 0, t) = 1$. For $n_1 \neq 0$ let the transition probability from 0 to 1 be $1-p_0$ and transition probability from 1 to 0 be $1 - p_1$ for the random process $q_1(n_1, n_2, t)$. Let the random process $q_2(n_1, n_2, t)$ have the same transition probabilities for $n_2 \neq 0$. Random process $\Psi(n_1, n_2, t)$ given by (5.10) is also first order Markovian. We have

$\Psi(0,0,t) = 3$, $\Psi(0,n_2,t) \in \{1,3\}$ for $n_2 \neq 0$, $\Psi(n_1,0,t) \in \{2,3\}$ for $n_1 \neq 0$ and $\Psi(n_1,n_2,t) \in \{0,1,2,3\}$ for $n_1 \neq 0$ and $n_2 \neq 0$. Let:

$$\mathbb{P}_0 = 1$$

$$\mathbb{P}_1 = \begin{bmatrix} p_0 & 1-p_0 \\ 1-p_1 & p_1 \end{bmatrix}$$

$$\mathbb{P}_2 = \begin{bmatrix} p_0 & 1-p_0 \\ 1-p_1 & p_1 \end{bmatrix}$$

$$\mathbb{P}_3 = \begin{bmatrix} p_0p_0 & p_0(1-p_0) & p_0(1-p_0) & (1-p_0)^2 \\ p_0(1-p_1) & p_0p_1 & (1-p_0)(1-p_1) & (1-p_0)p_1 \\ (1-p_1)p_0 & (1-p_0)(1-p_1) & p_0p_1 & (1-p_0)p_1 \\ (1-p_1)^2 & p_1(1-p_1) & p_1(1-p_1) & p_1p_1 \end{bmatrix}$$

Transition probability matrices for the random process $\Psi(n_1,n_2,t)$ are given below.

$$\mathbb{P}(0,0) = \mathbb{P}_0$$

$$\mathbb{P}(0,n_2) = \mathbb{P}_1 \; for \; n_2 \neq 0$$

$$\mathbb{P}(n_1,0) = \mathbb{P}_2 \; for \; n_1 \neq 0$$

$$\mathbb{P}(n_1,n_2) = \mathbb{P}_3 \; for \; n_1 \neq 0 \; and \; n_2 \neq 0$$

5.2.3.1 FM Model Based Implementation

The input-output transfer function (5.19) can be realized using the FM model:

$$\begin{aligned}\boldsymbol{x}(n_1,n_2,t) &= \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t-1) + \begin{bmatrix} 0 & b \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(n_1-1,n_2,t) \\ &+ \begin{bmatrix} 0 & c \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(n_1,n_2-1,t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t) \\ \boldsymbol{y}(n_1,n_2,t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}(n_1,n_2,t) \end{aligned} \tag{5.20}$$

where $\boldsymbol{x}(n_1,n_2,t) \in \mathbb{R}^2$. Let:

$$\begin{aligned}\boldsymbol{K}_0^1(n_1,n_2) &= \begin{bmatrix} f^1_{11(0)} & f^1_{12(0)} \\ f^1_{21(0)} & f^1_{22(0)} \end{bmatrix} \\ \boldsymbol{K}_1^1(n_1,n_2) &= \begin{bmatrix} f^1_{11(1)} & f^1_{12(1)} \\ f^1_{21(1)} & f^1_{22(1)} \end{bmatrix} \\ \boldsymbol{K}_0^2(n_1,n_2) &= \begin{bmatrix} f^2_{11(0)} & f^2_{12(0)} \\ f^2_{21(0)} & f^2_{22(0)} \end{bmatrix} \\ \boldsymbol{K}_1^2(n_1,n_2) &= \begin{bmatrix} f^2_{11(1)} & f^2_{12(1)} \\ f^2_{21(1)} & f^2_{22(1)} \end{bmatrix} \end{aligned} \tag{5.21}$$

$\forall(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$. Then from (5.11):

$$\boldsymbol{A}^F_{0,0}(n_1, n_2) = \begin{bmatrix} a + f^1_{11(0)} + f^2_{11(0)} & f^1_{12(0)} + f^2_{12(0)} \\ 1 + f^1_{21(0)} + f^2_{21(0)} & f^1_{22(0)} + f^2_{22(0)} \end{bmatrix}$$

$$\boldsymbol{A}^F_{0,1}(n_1, n_2) = \begin{bmatrix} a + f^1_{11(1)} & f^1_{12(1)} \\ 1 + f^1_{21(1)} & f^1_{22(1)} \end{bmatrix}$$

$$\boldsymbol{A}^F_{1,0}(n_1, n_2) = \begin{bmatrix} a + f^2_{11(1)} & f^2_{12(1)} \\ 1 + f^2_{21(1)} & f^2_{22(1)} \end{bmatrix}$$

$$\boldsymbol{A}^F_{1,1}(n_1, n_2) = \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix}$$

for $(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$. From (5.12), $\boldsymbol{A}^F_0(n_1, n_2) = \boldsymbol{A}^F_{0,0}(n_1, n_2)$, $\boldsymbol{A}^F_1(n_1, n_2) = \boldsymbol{A}^F_{0,1}(n_1, n_2)$, $\boldsymbol{A}^F_2(n_1, n_2) = \boldsymbol{A}^F_{1,0}(n_1, n_2)$ and $\boldsymbol{A}^F_3(n_1, n_2) = \boldsymbol{A}^F_{1,1}(n_1, n_2)$.

$$\begin{aligned} \boldsymbol{H}^F_0 &= \boldsymbol{A}^F_3(n_1, n_2) \otimes \boldsymbol{A}^F_3(n_1, n_2) \\ \boldsymbol{H}^F_1 &= diag_{i\in\{1,3\}}(\boldsymbol{A}^F_i(n_1, n_2) \otimes \boldsymbol{A}^F_i(n_1, n_2)) \\ \boldsymbol{H}^F_2 &= diag_{i\in\{2,3\}}(\boldsymbol{A}^F_i(n_1, n_2) \otimes \boldsymbol{A}^F_i(n_1, n_2)) \\ \boldsymbol{H}^F_3 &= diag_{i\in\{0,1,2,3\}}(\boldsymbol{A}^F_i(n_1, n_2) \otimes \boldsymbol{A}^F_i(n_1, n_2)) \end{aligned}$$

$$\begin{aligned} \boldsymbol{A}^{\mathcal{F}}(0, 0) &= (\mathbb{P}^T_0 \otimes \boldsymbol{I}_4)\boldsymbol{H}^F_0 \\ \boldsymbol{A}^{\mathcal{F}}(0, n_2) &= (\mathbb{P}^T_1 \otimes \boldsymbol{I}_4)\boldsymbol{H}^F_1 && for\ n_2 \neq 0 \\ \boldsymbol{A}^{\mathcal{F}}(n_1, 0) &= (\mathbb{P}^T_2 \otimes \boldsymbol{I}_4)\boldsymbol{H}^F_2 && for\ n_1 \neq 0 \\ \boldsymbol{A}^{\mathcal{F}}(n_1, n_2) &= (\mathbb{P}^T_3 \otimes \boldsymbol{I}_4)\boldsymbol{H}^F_3 && for\ n_1 \neq 0\ and\ n_2 \neq 0 \end{aligned}$$

System (5.20) is mean square stable under link failure if and only if $\rho(\boldsymbol{A}^{\mathcal{F}}(n_1, n_2)) < 1$, $\forall(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$.

5.2.3.2 GR Model Based Implementation

The input-output transfer function (5.19) can be realized using the GR model:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & a_3 \\ 0 & 0 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^h(n_1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = [0\ 0\ 1]\boldsymbol{x}(n_1,n_2,t) \tag{5.22}$$

where $a_9 = a$, $a_3a_7 = b$, $a_6a_8 = c$ and $\boldsymbol{x}(n_1,n_2,t) \in \mathbb{R}^3$. Let $\boldsymbol{K}_0^{1t}(n_1,n_2) = r^1_{11(0)}$, $\boldsymbol{K}_1^{1t}(n_1,n_2) = r^1_{11(1)}$, $\boldsymbol{K}_0^{2t}(n_1,n_2) = r^2_{11(0)}$ and $\boldsymbol{K}_1^{2t}(n_1,n_2) = r^2_{11(1)}$, $\forall(n_1,n_2) \in [0,N_1-1]\times[0,N_2-1]$. Then from (5.16):

$$\begin{aligned} \boldsymbol{A}^R_{0,0}(n_1,n_2) &= a_9 + a_7 r^1_{11(0)} + a_8 r^2_{11(0)} \\ \boldsymbol{A}^R_{0,1}(n_1,n_2) &= a_9 + a_7 r^1_{11(1)} \\ \boldsymbol{A}^R_{1,0}(n_1,n_2) &= a_9 + a_8 r^2_{11(1)} \\ \boldsymbol{A}^R_{1,1}(n_1,n_2) &= a_9 \end{aligned}$$

for $(n_1,n_2) \in [0,N_1-1]\times[0,N_2-1]$. From (5.17), $\boldsymbol{A}^R_0(n_1,n_2) = \boldsymbol{A}^R_{0,0}(n_1,n_2)$, $\boldsymbol{A}^R_1(n_1,n_2) = \boldsymbol{A}^R_{0,1}(n_1,n_2)$, $\boldsymbol{A}^R_2(n_1,n_2) = \boldsymbol{A}^R_{1,0}(n_1,n_2)$ and $\boldsymbol{A}^R_3(n_1,n_2) = \boldsymbol{A}^R_{1,1}(n_1,n_2)$.

$$\begin{aligned} \boldsymbol{H}^R_3 &= \boldsymbol{A}^R_3(n_1,n_2) \otimes \boldsymbol{A}^R_3(n_1,n_2) \\ \boldsymbol{H}^R_2 &= diag_{i\in\{2,3\}}(\boldsymbol{A}^R_i(n_1,n_2) \otimes \boldsymbol{A}^R_i(n_1,n_2)) \\ \boldsymbol{H}^R_1 &= diag_{i\in\{1,3\}}(\boldsymbol{A}^R_i(n_1,n_2) \otimes \boldsymbol{A}^R_i(n_1,n_2)) \\ \boldsymbol{H}^R_0 &= diag_{i\in\{0,1,2,3\}}(\boldsymbol{A}^R_i(n_1,n_2) \otimes \boldsymbol{A}^R_i(n_1,n_2)) \end{aligned}$$

$$
\begin{aligned}
\boldsymbol{A}^{\mathcal{R}}(0,0) &= (\mathbb{P}_0^T \otimes \boldsymbol{I}_4)\boldsymbol{H}_0^R \\
\boldsymbol{A}^{\mathcal{R}}(0,n_2) &= (\mathbb{P}_1^T \otimes \boldsymbol{I}_4)\boldsymbol{H}_1^R && for\ n_2 \neq 0 \\
\boldsymbol{A}^{\mathcal{R}}(n_1,0) &= (\mathbb{P}_2^T \otimes \boldsymbol{I}_4)\boldsymbol{H}_2^R && for\ n_1 \neq 0 \\
\boldsymbol{A}^{\mathcal{R}}(n_1,n_2) &= (\mathbb{P}_3^T \otimes \boldsymbol{I}_4)\boldsymbol{H}_3^R && for\ n_1 \neq 0\ and\ n_2 \neq 0
\end{aligned}
$$

System (5.22) is mean square stable under link failure if and only if $\rho(\boldsymbol{A}^{\mathcal{R}}(n_1,n_2)) < 1, \forall (n_1,n_2) \in [0, N_1-1] \times [0, N_2-1]$.

5.3 Models for 3-D Systems Under Node Failure

Node failure can be due to intrusion, faults in the node or battery exhaustion. Once a fault has occurred the faulty node may or may not be repaired. In this work the cases where a faulty node is repaired and not repaired is treated separately.

It is assumed that if the node (n_1,n_2) has failed at time slot t, it is unable to transmit any information in the time slot . Let $q(n_1,n_2,t)$ be a random processes such that:

$$
q(n_1,n_2,t) = \begin{cases} 0 & node\ (n_1,n_2)\ failed\ at\ time\ t \\ 1 & otherwise \end{cases} \tag{5.23}
$$

Random process $q(n_1,n_2,t)$ is a success-failure process of node (n_1,n_2) at time t.

5.3.1 Permanent Node Failure

When there are node failures in the sensor network, at a given time, only a subset of the originally deployed sensor nodes may be operational. If node failures are permanent, number of nodes operational at a given time t is a non increasing function of t.

Depending upon how failures are handled by the sensor network system matrices and interconnections between nodes may change after every new node failure. Furthermore it is reasonable to assume that as time tends to infinity the probability of at least one node remaining functional tends to zero. Therefore it is impossible to adopt the notion of asymptotic stability in the conventional sense to sensor networks with permanent node failures. The following notion of stability is introduced for sensor networks with permanent node failure.

Definition 5.3.1 *The sensor network is said to be asymptotically stable when configuration is frozen in time if:*

$$\lim_{t\to\infty} \|\boldsymbol{x}(n_1, n_2, t)\|_2 = 0 \quad \forall (n_1, n_2) \in O(t_0)$$

when the set of currently functional nodes remain unchanged. Here $\|\boldsymbol{x}(n_1, n_2, t)\|_2 = \sqrt{\boldsymbol{x}(n_1, n_2, t)^T \boldsymbol{x}(n_1, n_2, t)}$*,* t_0 *is the current time and* $O(t_0)$ *denotes the set of nodes operational at time* t_0*. There are no nodes along* $n_1 = -1$ *or* $n_2 = -1$*. Therefore it is assumed that* $\boldsymbol{x}(-1, n_2, t) = 0$*,* $\boldsymbol{x}(n_1, -1, t) = 0$*. The only non-zero boundary conditions are given by* $\boldsymbol{x}(n_1, n_2, 0)$*.*

To determine the asymptotic stability of a sensor network according to the above definition the set of operational nodes is fixed to be invariant. Therefore the system of which asymptotic stability when configuration is frozen in time has to be determined becomes a deterministic system.

5.3.1.1 FM Model under Permanent Node Failure

When node (n_1, n_2) does not receive state or input vectors required for computation, it estimates the missing vectors. Let $\hat{\boldsymbol{x}}_{n_1 n_2}(n_1 - i, n_2 - j, t)$ and $\hat{\boldsymbol{u}}_{n_1 n_2}(n_1 - i, n_2 - j, t)$

be estimates of $\boldsymbol{x}(n_1-i, n_2-j, t)$ and $\boldsymbol{u}(n_1-i, n_2-j, t)$ made by node (n_1, n_2) at time t. The FM model for distributed 3-D systems under permanent node failure is given by:

$$\begin{aligned}
\boldsymbol{x}(n_1, n_2, t) &= q(n_1-1, n_2, t)\{\boldsymbol{A}_h\boldsymbol{x}(n_1\text{–}1, n_2, t) + \boldsymbol{B}_h\boldsymbol{u}(n_1\text{–}1, n_2, t)\} \\
&\quad + (1-q(n_1-1, n_2, t))\{\boldsymbol{A}_h\hat{\boldsymbol{x}}_{n_1n_2}(n_1\text{–}1, n_2, t) + \boldsymbol{B}_h\hat{\boldsymbol{u}}_{n_1n_2}(n_1\text{–}1, n_2, t)\} \\
&\quad + q(n_1, n_2-1, t)\{\boldsymbol{A}_v\boldsymbol{x}(n_1, n_2\text{–}1, t) + \boldsymbol{B}_v\boldsymbol{u}(n_1, n_2\text{–}1, t)\} \\
&\quad + (1-q(n_1, n_2-1, t))\{\boldsymbol{A}_v\hat{\boldsymbol{x}}_{n_1n_2}(n_1, n_2\text{–}1, t) + \boldsymbol{B}_v\hat{\boldsymbol{u}}_{n_1n_2}(n_1, n_2\text{–}1, t)\} \\
&\quad + \boldsymbol{A}_t\boldsymbol{x}(n_1, n_2, t\text{–}1) + \boldsymbol{B}_t\boldsymbol{u}(n_1, n_2, t\text{–}1) \\
\boldsymbol{y}(n_1, n_2, t) &= \boldsymbol{C}\boldsymbol{x}(n_1, n_2, t) + \boldsymbol{D}\boldsymbol{u}(n_1, n_2, t)
\end{aligned} \tag{5.24}$$

where $(n_1, n_2) \in O(t)$. Dimensions of vectors and matrices in (5.24) are the same as those in (5.3).

5.3.1.2 GR Model under Permanent Node Failure

Input vectors of other nodes are not required for computations and hence not required to be estimated. Let estimates of $\boldsymbol{x}^h(n_1, n_2, t)$ and $\boldsymbol{x}^v(n_1, n_2, t)$ made by node (n_1, n_2) at time t be denoted by $\hat{\boldsymbol{x}}^h(n_1, n_2, t)$ and $\hat{\boldsymbol{x}}^v(n_1, n_2, t)$ respectively. Let:

$$\begin{aligned}
\tilde{\boldsymbol{x}}^h(n_1, n_2, t) &= q(n_1-1, n_2, t)\boldsymbol{x}^h(n_1, n_2, t) + (1-q(n_1-1, n_2, t))\hat{\boldsymbol{x}}^h(n_1, n_2, t) \\
\tilde{\boldsymbol{x}}^v(n_1, n_2, t) &= q(n_1, n_2-1, t)\boldsymbol{x}^v(n_1, n_2, t) + (1-q(n_1, n_2-1, t))\hat{\boldsymbol{x}}^v(n_1, n_2, t)
\end{aligned} \tag{5.25}$$

The GR model for distributed 3-D systems under link failure is given by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}^h(n_1,n_2,t) \\ \tilde{\boldsymbol{x}}^v(n_1,n_2,t) \\ \boldsymbol{x}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = \boldsymbol{C}\tilde{\boldsymbol{x}}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t) \tag{5.26}$$

where $\tilde{\boldsymbol{x}}(n_1,n_2,t) = (\tilde{\boldsymbol{x}}^{h^T}(n_1,n_2,t), \tilde{\boldsymbol{x}}^{v^T}(n_1,n_2,t), \boldsymbol{x}^{t^T}(n_1,n_2,t))^T$.
Here $(n_1,n_2) \in O(t)$. Dimensions of vectors and matrices in (5.26) are the same as those in (5.5).

5.3.1.3 Asymptotic Stability

How the missing vectors are estimated by nodes affects stability of the system. All the neighbors of a node could be non-functional after a certain time. Nodes from which information is gathered for estimation, once a neighboring node fails, can't be restricted to immediate neighbors of a node. Sensor networks with permanent node failure may require schemes to reconfigure the interconnections between nodes once a node failure is detected. It is assumed that all estimates made are linear combinations of known state and input vectors. It is further assumed that for a estimation made by node (n_1,n_2) at time t:

1. A node does not use its state and input vectors at times less than $t-1$.

2. State vectors of its succeeding nodes are not used.

Results derived in the following can be extended readily to the case where state and input vectors at a finite number of previous time slots are used for the estimation. Condition 2 is necessitated due to causality and computability considerations.

5.3.1.4 FM Model

For notational simplicity let:

$$\boldsymbol{A}_h\hat{\boldsymbol{x}}_{n_1n_2}(n_1\text{–}1,n_2,t)+\boldsymbol{B}_h\hat{\boldsymbol{u}}_{n_1n_2}(n_1\text{–}1,n_2,t)=\boldsymbol{L}^1_{q(n_1,n_2-1,t)}(n_1,n_2,t)+\boldsymbol{K}^1_{q(n_1,n_2-1,t)}(n_1,n_2)\boldsymbol{x}(n_1,n_2,t-1) \tag{5.27}$$

$$\boldsymbol{A}_v\hat{\boldsymbol{x}}_{n_1n_2}(n_1,n_2\text{–}1,t)+\boldsymbol{B}_v\hat{\boldsymbol{u}}_{n_1n_2}(n_1,n_2\text{–}1,t)=\boldsymbol{L}^2_{q(n_1-1,n_2,t)}(n_1,n_2,t)+\boldsymbol{K}^2_{q(n_1-1,n_2,t)}(n_1,n_2)\boldsymbol{x}(n_1,n_2,t-1) \tag{5.28}$$

In the above, $\boldsymbol{L}^1_{q(n_1,n_2-1,t)}(n_1,n_2,t)$ and $\boldsymbol{L}^2_{q(n_1-1,n_2,t)}(n_1,n_2,t)$ denote linear combinations of all the vectors used by node (n_1,n_2) for the estimation except $\boldsymbol{x}(n_1,n_2,t-1)$. Subscripts $q(n_1-1,n_2,t)$ and $q(n_1,n_2-1,t)$ are used to indicate that weights given to the vectors used in the estimation may depend upon random variables $q(n_1-1,n_2,t)$ and $q(n_1,n_2-1,t)$.

Theorem 5.3.1 *The system* (5.24) *is asymptotically stable when configuration is frozen in time, if and only if* $\rho(\boldsymbol{A}_F(n_1,n_2))<1\ \forall(n_1,n_2)\in O(t)$ *where:*

$$\boldsymbol{A}_F(n_1,n_2)=\boldsymbol{A}_t+(1-q(n_1,n_2-1,t))\boldsymbol{K}^2_{q(n_1-1,n_2,t)}(n_1,n_2)+(1-q(n_1-1,n_2,t))\boldsymbol{K}^1_{q(n_1,n_2-1,t)}(n_1,n_2) \tag{5.29}$$

Proof Proof of necessity is trivial. In order to prove sufficiency it is assumed that

$\rho(\boldsymbol{A}_F(n_1, n_2)) < 1, \forall (n_1, n_2) \in O(t)$. Let:

$$O_1(t) = \{(n_1, n_2) : (n_1, n_2) \in O(t) \; and \; node \; (n_1, n_2) \; doesn't \; have \; preceeding \; nodes\}$$

Since nodes in $O_1(t)$ do not have predecessors to receive state vectors from, state transition of nodes in the aforementioned set is described by the following equation.

$$\boldsymbol{x}(n_1, n_2, t+1) = \boldsymbol{A}_F(n_1, n_2)\boldsymbol{x}(n_1, n_2, t)$$

Therefore:

$$\lim_{t \to \infty} \|\boldsymbol{x}(n_1, n_2, t)\|_2 = 0 \quad \forall (n_1, n_2) \in O_1(t)$$

Let:

$$O_2(t) = \{(n_1, n_2) : (n_1, n_2) \in O(t) \; and \; all \; preceeding \; nodes \; of \; node \; (n_1, n_2) \; are \; in \; O_1(t)\}$$

Nodes in $O_2(t)$ receive state vectors, form nodes in $O_1(t)$, which are used in computing their own state vectors. But state vectors of nodes in $O_1(t)$ decay exponentially to origin as time t tends to infinity. Systems running on nodes in $O_2(t)$ are asymptotically stable and driven by exponentially decaying inputs. Therefore:

$$\lim_{t \to \infty} \|\boldsymbol{x}(n_1, n_2, t)\| = 0 \quad \forall (n_1, n_2) \in O_2(t)$$

By using the same argument iteratively it can be shown that:

$$\lim_{t\to\infty} \|\boldsymbol{x}(n_1, n_2, t)\| = 0 \quad \forall (n_1, n_2) \in O(t)$$

This completes the proof.

5.3.1.5 GR Model

For notational simplicity let:

$$\begin{aligned}
\hat{\boldsymbol{x}}^h(n_1, n_2, t) &= \boldsymbol{L}^1_{q(n_1,n_2-1,t)}(n_1, n_2, t) + \boldsymbol{K}^{1h}_{q(n_1,n_2-1,t)}(n_1, n_2)\boldsymbol{x}^h(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{1v}_{q(n_1,n_2-1,t)}(n_1, n_2)\boldsymbol{x}^v(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{1t}_{q(n_1,n_2-1,t)}(n_1, n_2)\boldsymbol{x}^t(n_1, n_2, t-1) \\
\hat{\boldsymbol{x}}^v(n_1, n_2, t) &= \boldsymbol{L}^2_{q(n_1-1,n_2,t)}(n_1, n_2, t) + \boldsymbol{K}^{2h}_{q(n_1-1,n_2,t)}(n_1, n_2)\boldsymbol{x}^h(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{2v}_{q(n_1-1,n_2,t)}(n_1, n_2)\boldsymbol{x}^v(n_1, n_2, t-1) \\
&\quad + \boldsymbol{K}^{2t}_{q(n_1-1,n_2,t)}(n_1, n_2)\boldsymbol{x}^t(n_1, n_2, t-1) \qquad (5.30)
\end{aligned}$$

In the above, $\boldsymbol{L}^1_{q(n_1-1,n_2,t)}(n_1, n_2, t)$ and $\boldsymbol{L}_{q(n_1,n_2-1,t)}(n_1, n_2, t)$ denote linear combinations of all the vectors used by node (n_1, n_2) for the estimation except $\boldsymbol{x}(n_1, n_2, t-1)$. Subscripts $q(n_1-1, n_2, t)$ and $q(n_1, n_2-1, t)$ indicate that weights given to vectors used in the estimation may depend upon random variables $q(n_1-1, n_2, t)$ and $q(n_1, n_2-1, t)$.

Theorem 5.3.2 *The system* (5.26) *is asymptotically stable when configuration is frozen*

in time if and only if $\rho(\boldsymbol{A}_R(n_1,n_2)) < 1$, $\forall (n_1,n_2) \in O(t)$ *where:*

$$\begin{aligned}\boldsymbol{A}_R(n_1,n_2) = \boldsymbol{A}_9 &+ (1-q(n_1-1,n_2,t))\boldsymbol{A}_7\boldsymbol{K}^{1t}_{q(n_1,n_2-1,t)}(n_1,n_2) \\ &+ (1-q(n_1,n_2-1,t))\boldsymbol{A}_8\boldsymbol{K}^{2t}_{q(n_1-1,n_2,t)}(n_1,n_2)\end{aligned} \tag{5.31}$$

Proof The proof is similar to the proof of Theorem 5.3.1 and would be omitted for the sake of brevity.

5.3.1.6 Example

Let the transfer function to be implemented be given by (5.19). FM model (5.20) is used to realize the transfer function. Let $\boldsymbol{K}^1_0(n_1,n_2)$,$\boldsymbol{K}^1_1(n_1,n_2)$,$\boldsymbol{K}^2_0(n_1,n_2)$ and $\boldsymbol{K}^2_1(n_1,n_2)$ be given by (5.21). Then from (5.29):

$$\boldsymbol{A}^F(n_1,n_2) = \begin{cases} \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} + \boldsymbol{K}^1_0(n_1,n_2) + \boldsymbol{K}^2_0(n_1,n_2) & \begin{matrix} q(n_1-1,n_2,t)=0 \\ q(n_1,n_2-1,t)=0 \end{matrix} \\ \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} + \boldsymbol{K}^1_1 & \begin{matrix} q(n_1-1,n_2,t)=1 \\ q(n_1,n_2-1,t)=0 \end{matrix} \\ \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} + \boldsymbol{K}^2_1 & \begin{matrix} q(n_1-1,n_2,t)=0 \\ q(n_1,n_2-1,t)=1 \end{matrix} \\ \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} & \begin{matrix} q(n_1-1,n_2,t)=1 \\ q(n_1,n_2-1,t)=1 \end{matrix} \end{cases}$$

System (5.20) is asymptotically stable when configuration is frozen in time if and only if $\rho(\boldsymbol{A}^{\mathcal{F}}(n_1,n_2)) < 1, \forall (n_1,n_2) \in O(t)$.

Let the GR model (5.22) be used to realize the transfer function. Let $\boldsymbol{K}^{1t}_0(n_1,n_2) =$

$r^1_{11(0)}$, $\boldsymbol{K}^{1t}_1(n_1,n_2) = r^1_{11(1)}$, $\boldsymbol{K}^{2t}_0(n_1,n_2) = r^2_{11(0)}$ and $\boldsymbol{K}^{2t}_1(n_1,n_2) = r^2_{11(1)}$, $\forall (n_1,n_2) \in [0, N_1-1] \times [0, N_2-1]$. Then from (5.31):

$$\boldsymbol{A}^R(n_1,n_2) = \begin{cases} a_9 + a_7 r^1_{11(0)} + a_8 r^2_{11(0)} & q(n_1-1,n_2,t) = 0 \\ & q(n_1,n_2-1,t) = 0 \\ a_9 + a_8 r^2_{11(1)} & q(n_1-1,n_2,t) = 1 \\ & q(n_1,n_2-1,t) = 0 \\ a_9 + a_7 r^1_{11(1)} & q(n_1-1,n_2,t) = 0 \\ & q(n_1,n_2-1,t) = 1 \\ a_9 & q(n_1-1,n_2,t) = 1 \\ & q(n_1,n_2-1,t) = 1 \end{cases}$$

System (5.22) is asymptotically stable when configuration is frozen in time if and only if $\rho(\boldsymbol{A}^R(n_1,n_2)) < 1, \forall (n_1,n_2) \in O(t)$.

5.3.2 Temporary Node Failure

Node failure in a sensor network may be temporary because either non functional nodes are repaired or nodes start to work again spontaneously. For example in a sensor network powered by solar power, sensor nodes that ceased to function due to lack of power may start to work again once batteries are recharged. The notion of asymptotic stability as defined in definition 5.2.1 can be used for sensor networks with temporary node failure.

5.3.2.1 FM Model under Temporary Node Failure

Let $\hat{\boldsymbol{x}}_{n_1n_2}(n_1-i,n_2-j,t-k)$, $\hat{\boldsymbol{u}}_{n_1n_2}(n_1-i,n_2-j,t-k)$ be estimates of $\boldsymbol{x}(n_1-i,n_2-j,t-k)$ and $\boldsymbol{u}(n_1-i,n_2-j,t-k)$ made by node (n_1,n_2) at time t. The FM

model for distributed 3-D systems under temporary node failure is given by:

$$\begin{aligned}
\boldsymbol{x}(n_1,n_2,t) &= q(n_1-1,n_2,t)\{\boldsymbol{A}_h\boldsymbol{x}(n_1\text{–}1,n_2,t)+\boldsymbol{B}_h\boldsymbol{u}(n_1\text{–}1,n_2,t)\}\\
&\quad+(1-q(n_1-1,n_2,t))\{\boldsymbol{A}_h\hat{\boldsymbol{x}}_{n_1n_2}(n_1\text{–}1,n_2,t)+\boldsymbol{B}_h\hat{\boldsymbol{u}}_{n_1n_2}(n_1\text{–}1,n_2,t)\}\\
&\quad+q(n_1,n_2-1,t)\{\boldsymbol{A}_v\boldsymbol{x}(n_1,n_2\text{–}1,t)+\boldsymbol{B}_v\boldsymbol{u}(n_1,n_2\text{–}1,t)\}\\
&\quad+(1-q(n_1,n_2-1,t))\{\boldsymbol{A}_v\hat{\boldsymbol{x}}_{n_1n_2}(n_1,n_2\text{–}1,t)+\boldsymbol{B}_v\hat{\boldsymbol{u}}_{n_1n_2}(n_1,n_2\text{–}1,t)\}\\
&\quad+q(n_1,n_2,t-1)\{\boldsymbol{A}_t\boldsymbol{x}(n_1,n_2,t\text{–}1)+\boldsymbol{B}_t\boldsymbol{u}(n_1,n_2,t\text{–}1)\}\\
&\quad+(1-q(n_1,n_2,t-1))\{\boldsymbol{A}_t\hat{\boldsymbol{x}}(n_1,n_2,t\text{–}1)+\boldsymbol{B}_t\hat{\boldsymbol{u}}(n_1,n_2,t\text{–}1)\}\\
\boldsymbol{y}(n_1,n_2,t) &= \boldsymbol{C}\boldsymbol{x}(n_1,n_2,t)+\boldsymbol{D}\boldsymbol{u}(n_1,n_2,t)
\end{aligned} \tag{5.32}$$

Dimensions of vectors and matrices in (5.32) are the same as those in (5.3). An important difference compared to the other two cases is that, nodes may have to estimate their own state and input vectors at the last time slot they were not functioning, when they start to function again.

5.3.2.2 GR Model under Temporary Node Failure

Input vectors of other nodes are not required for computations and hence not required to be estimated. Let estimates of $\boldsymbol{x}^h(n_1,n_2,t)$, $\boldsymbol{x}^v(n_1,n_2,t)$ and $\boldsymbol{x}^t(n_1,n_2,t)$ made by node (n_1,n_2) be denoted by $\hat{\boldsymbol{x}}^h(n_1,n_2,t)$, $\hat{\boldsymbol{x}}^v(n_1,n_2,t)$ and $\hat{\boldsymbol{x}}^t(n_1,n_2,t)$ respectively. Let:

$$\begin{aligned}
\tilde{\boldsymbol{x}}^h(n_1,n_2,t) &= q(n_1-1,n_2,t)\boldsymbol{x}^h(n_1,n_2,t)+(1-q(n_1-1,n_2,t))\hat{\boldsymbol{x}}^h(n_1,n_2,t)\\
\tilde{\boldsymbol{x}}^v(n_1,n_2,t) &= q(n_1,n_2-1,t)\boldsymbol{x}^v(n_1,n_2,t)+(1-q(n_1,n_2-1,t))\hat{\boldsymbol{x}}^v(n_1,n_2,t)\\
\tilde{\boldsymbol{x}}^t(n_1,n_2,t) &= q(n_1,n_2,t-1)\boldsymbol{x}^t(n_1,n_2,t)+(1-q(n_1,n_2,t-1))\hat{\boldsymbol{x}}^t(n_1,n_2,t)
\end{aligned} \tag{5.33}$$

The GR model for distributed 3-D systems under link failure is given by:

$$\begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \\ \boldsymbol{A}_4 & \boldsymbol{A}_5 & \boldsymbol{A}_6 \\ \boldsymbol{A}_7 & \boldsymbol{A}_8 & \boldsymbol{A}_9 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}^h(n_1,n_2,t) \\ \tilde{\boldsymbol{x}}^v(n_1,n_2,t) \\ \tilde{\boldsymbol{x}}^t(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_3 \end{bmatrix} \boldsymbol{u}(n_1,n_2,t)$$

$$\boldsymbol{y}(n_1,n_2,t) = \boldsymbol{C}\tilde{\boldsymbol{x}}(n_1,n_2,t) + \boldsymbol{D}\boldsymbol{u}(n_1,n_2,t) \tag{5.34}$$

where $\tilde{\boldsymbol{x}}(n_1,n_2,t) = (\tilde{\boldsymbol{x}}^{h^T}(n_1,n_2,t), \tilde{\boldsymbol{x}}^{v^T}(n_1,n_2,t), \tilde{\boldsymbol{x}}^{t^T}(n_1,n_2,t))^T$.
Dimensions of vectors and matrices in (5.34) are the same as those in (5.5). Unlike in the other two cases nodes may have to estimate their own temporal vector component, when they start to function again after having been non-functional.

5.3.2.3 Asymptotic Stability

When a node restarts to function it has to estimate is own state vector or temporal state vector component depending upon the state space model used. State vectors from succeeding nodes in the sensor networks may be used for the estimation. Causality and computability of the system must be considered if state vectors from succeeding nodes are used to compute the state vector of a node. Using state vectors from succeeding nodes for estimating the state vector of itself has far reaching implications on asymptotic stability of the system. Conditions on stability are not available for this case at the time of writing. The case where nodes abide by conditions (1) and (2) given in 5.3.1.3 in estimations is studied in the following treatise.

Let $P(n_1,n_2,t_0,t_0+t)$ denote the probability that node (n_1,n_2) has not failed between time t_0 and t_0+t. It is assumed that node failure is statistically independent of the initial conditions of the sensor network. Spectral norm of a square matrix $\boldsymbol{M}$ is denoted by $\sigma(\boldsymbol{M})$.

5.3.2.4 FM Model

Lemma 5.3.1 *Consider the system described by:*

$$\boldsymbol{x}(n_1, n_2, t) = \boldsymbol{A}_t^t \boldsymbol{x}(n_1, n_2, 0) \prod_{i=0}^{i=t} q(n_1, n_2, i) \tag{5.35}$$

If $\exists\ \mu_{n_1 n_2} > 0, \lambda_{n_1 n_2} \in [0, \gamma]$ *where* $0 \leq \gamma < 1$ *and* $k_{n_1 n_2} \in \mathbb{Z}^+$ *such that:*

$$\sigma(\sqrt[4]{P(n_1, n_2, 0, t)}\boldsymbol{A}_t^t) \leq \mu_{n_1 n_2} t^{k_{n_1 n_2}} \lambda_{n_1 n_2}^t \tag{5.36}$$

$\forall\ t$*, then*

$$\|\boldsymbol{x}(n_1, n_2, t)\| \leq \mu_{n_1 n_2} t^{k_{n_1 n_2}} \lambda_{n_1 n_2}^t \|\boldsymbol{x}(n_1, n_2, 0)\|$$

Proof

$$\|\boldsymbol{x}(n_1, n_2, t)\| = E\{\boldsymbol{x}(n_1, n_2, 0)^T (\boldsymbol{A}_t^t)^T \boldsymbol{A}_t^t \boldsymbol{x}(n_1, n_2, 0) \prod_{i=0}^{i=t} q(n_1, n_2, i) \prod_{i=0}^{i=t} q(n_1, n_2, i)\}$$

Since node failure is assumed to be statistically independent of initial conditions of the

sensor network, we have:

$$\begin{aligned}
\|\boldsymbol{x}(n_1,n_2,t)\| &= E\{\boldsymbol{x}(n_1,n_2,0)^T(\boldsymbol{A}_t^t)^T\boldsymbol{A}_t^t\boldsymbol{x}(n_1,n_2,0)\} \\
&\times E\,\{\prod_{i=0}^{i=t} q(n_1,n_2,i)\prod_{i=0}^{i=t} q(n_1,n_2,i)\,\} \\
&= E\{\boldsymbol{x}(n_1,n_2,0)^T\sqrt{P(n_1,n_2,0,t)}(\boldsymbol{A}_t^t)^T \\
&\times \sqrt{P(n_1,n_2,0,t)}\boldsymbol{A}_t^t\boldsymbol{x}(n_1,n_2,0)\} \\
&\leq E\{\boldsymbol{x}(n_1,n_2,0)^T\sqrt[4]{P(n_1,n_2,0,t)}(\boldsymbol{A}_t^t)^T \\
&\times \sqrt[4]{P(n_1,n_2,0,t)}\boldsymbol{A}_t^t\boldsymbol{x}(n_1,n_2,0)\} \\
&\leq \mu_{n_1n_2}t^{k_{n_1n_2}}\lambda_{n_1n_2}^t\|\boldsymbol{x}(n_1,n_2,0)\|
\end{aligned}$$

Theorem 5.3.3 *The system* (5.32) *is mean square stable if:* $\exists\ \mu_{n_1n_2} > 0, \lambda_{n_1n_2} \in [0,\gamma]$ *where* $0 \leq \gamma < 1$ *and* $k_{n_1n_2} \in \mathbb{Z}^+$ *such that:*

$$\sigma(\sqrt[4]{P(n_1,n_2,t_0,t_0+t)}\boldsymbol{A}_t^t) \leq \mu_{n_1n_2}t^{k_{n_1n_2}}\lambda_{n_1n_2}^t \tag{5.37}$$

$\forall\ t$ *and* $\forall(n_1,n_2) \in [0,N_1-1]\times[0,N_2-1]$.

Proof In order to prove sufficiency let condition (5.37) be satisfied $\forall(n_1,n_2) \in [0,N_1-1]\times[0,N_2-1]$. We have:

$$\boldsymbol{x}(0,0,t) = \boldsymbol{A}_t^t\boldsymbol{x}(0,0,0)\prod_{i=0}^{i=t} q(0,0,i)$$

Due to Lemma 5.3.1:

$$\|\boldsymbol{x}(0,0,t)\| \leq \mu_{00}t^{k_{00}}\lambda_{00}^t\|\boldsymbol{x}(0,0,0)\|$$

Therefore:

$$\lim_{t\to\infty} \|\boldsymbol{x}(0,0,t)\| = 0$$

If the node $(1,0)$ was non-functional at time $t-1$ and starts to function at time t, that is if $q(1,0,t-1) = 0$ and $q(1,0,t) = 1$, it uses state vector of node (0.0) at time t to estimate its own state vector at time $t-1$. Let the weight given to $\boldsymbol{x}(0,0,t)$ in estimating $\boldsymbol{x}(1,0,t-1)$ be $\boldsymbol{W}^1(0,1) \in \mathbb{R}^n$. State vector of node $(1,0)$ at time t is given by:

$$\boldsymbol{x}(1,0,t) = \boldsymbol{A}_t^t \boldsymbol{x}(1,0,0) \prod_{i=0}^{i=t} q(1,0,i) + \sum_{i=0}^{t} \boldsymbol{A}_t^{t-i} \boldsymbol{A}_h \boldsymbol{x}(0,0,i) \prod_{j=i}^{j=t} q(1,0,j) + \sum_{i=0}^{t-1} \boldsymbol{A}_t^{t-i} \boldsymbol{W}^1(0,1) \boldsymbol{x}(0,0,i)(1-q(1,0,i-1)) \prod_{j=i}^{j=t} q(1,0,j)$$

$$\|\boldsymbol{x}(1,0,t)\| \leq \|\boldsymbol{A}_t^t \boldsymbol{x}(1,0,0) \prod_{i=0}^{i=t} q(1,0,i)\| + \sum_{i=0}^{t} \|\boldsymbol{A}_t^{t-i} \boldsymbol{A}_h \boldsymbol{x}(0,0,i) \prod_{j=i}^{j=t} q(1,0,j)\| + \sum_{i=0}^{t-1} \|\boldsymbol{A}_t^{t-i} \boldsymbol{W}^1(0,1) \boldsymbol{x}(0,0,i)(1-q(1,0,i-1)) \prod_{j=i}^{j=t} q(1,0,j)\|$$

Since node failure is assumed to be statistically independent of the initial conditions:

$$\begin{aligned}\|\boldsymbol{A}_t^{t-i}\boldsymbol{A}_h\boldsymbol{x}(0,0,i)\prod_{j=i}^{j=t}q(1,0,j)\| &\leq \|\boldsymbol{A}_t^{t-i}\boldsymbol{A}_h\boldsymbol{A}_t^{i}\boldsymbol{x}(0,0,0)\| \\ &\quad\times E\{\prod_{j=0}^{j=i}q(0,0,i)\prod_{j=i}^{j=t}q(1,0,j)\} \\ &\leq \|\boldsymbol{A}_t^{t-i}\boldsymbol{A}_h\boldsymbol{A}_t^{i}\boldsymbol{x}(0,0,0)\| \\ &\quad\times \sqrt{P(0,0,0,i)}\sqrt{P(1,0,i,t)} \\ &\leq \mu_{10}(t-i)^{k_{10}}\lambda_{10}^{t-i}\sigma(\boldsymbol{A}_h)\mu_{00}i^{k_{00}}\lambda_{00}^{i}\|\boldsymbol{x}(0,0,0)\|\end{aligned}$$

Therefore:

$$\begin{aligned}\|\boldsymbol{x}(1,0,t)\| &\leq \mu_{10}t^{k_{10}}\lambda_{10}^{t}\|\boldsymbol{x}(1,0,0)\| \\ &\quad+\sum_{i=0}^{t}\mu_{10}(t-i)^{k_{10}}\lambda_{10}^{t-i}\sigma(\boldsymbol{A}_h)\mu_{00}i^{k_{00}}\lambda_{00}^{i}\|\boldsymbol{x}(0,0,0)\| \\ &\quad+\sum_{i=0}^{t-1}\mu_{10}(t-i)^{k_{10}}\lambda_{10}^{t-i}\sigma(\boldsymbol{W}^1(0,1))\mu_{00}i^{k_{00}}\lambda_{00}^{i}\|\boldsymbol{x}(0,0,0)\|\end{aligned}$$

It is easily seen that:

$$\lim_{t\to\infty}\|\boldsymbol{x}(1,0,t)\| = 0$$

By using the same argument iteratively is can be shown that:

$$\lim_{t\to\infty}\|\boldsymbol{x}(n_1,n_2,t)\| = 0$$

$\forall(n_1,n_2)\in[0,N_1-1]\times[0,N_2-1]$.

Corollary I f the system (5.32) is asymptotically stable when there are no node fail-

ures it is mean square stable under temporary node failure.

Proof If the system is asymptotically stable condition (5.37) is satisfied $\forall(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$. Hence the system (5.32) is mean square stable.

5.3.2.5 GR Model

Lemma 5.3.2 *Consider the system described by:*

$$\boldsymbol{x}^t(n_1, n_2, t) = \boldsymbol{A}_9^t \boldsymbol{x}^t(n_1, n_2, 0) \prod_{i=0}^{i=t} q(n_1, n_2, i) \tag{5.38}$$

If $\exists\ \mu_{n_1 n_2} > 0, \lambda_{n_1 n_2} \in [0, \gamma]$ *where* $0 \leq \gamma < 1$ *and* $k_{n_1 n_2} \in \mathbb{Z}^+$ *such that:*

$$\sigma(\sqrt[4]{P(n_1, n_2, t_0, t_0 + t)}\boldsymbol{A}_9^t) \leq \mu_{n_1 n_2} t^{k_{n_1 n_2}} \lambda_{n_1 n_2}^t \tag{5.39}$$

$\forall\ t$ *then*

$$\|\boldsymbol{x}^t(n_1, n_2, t)\| \leq \mu_{n_1 n_2} t^{k_{n_1 n_2}} \lambda_{n_1 n_2}^t \|\boldsymbol{x}^t(n_1, n_2, 0)\|$$

Proof The proof is similar to the proof of Lemma 5.3.1.

Theorem 5.3.4 *The system* (5.34) *is mean square stable, if* $\exists\ \mu_{n_1 n_2} > 0, \lambda_{n_1 n_2} \in [0, \gamma]$ *where* $0 \leq \gamma < 1$ *and* $k_{n_1 n_2} \in \mathbb{Z}^+$ *such that:*

$$\sigma(\sqrt[4]{P(n_1, n_2, t_0, t_0 + t)}\boldsymbol{A}_9^t) < \mu_{n_1 n_2} t^{k_{n_1 n_2}} \lambda_{n_1 n_2}^t \tag{5.40}$$

$\forall\ t$ *and* $\forall(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$

Proof The proof is similar to the proof of Theorem 3.1.

Corollary 2 If the system (5.34) is asymptotically stable when there are no node failures it is mean square stable under temporary node failure.

Proof If the system is asymptotically stable condition (5.40) is satisfied $\forall(n_1, n_2) \in [0, N_1 - 1] \times [0, N_2 - 1]$. Hence the system (5.34) is mean square stable.

5.3.3 Example

Let the transfer function (5.19) be realized using the FM model (5.20) and GR model (5.22) on a sensor network. Let the probability of node (n_1, n_2) not failing until time t, $P(n_1, n_2, t_0, t + t_0) = \gamma^t$ where $\gamma \in (0, 1)$. In the FM model based implementation:

$$\sigma(\sqrt[4]{P(n_1, n_2, t_0, t + t_0)}\boldsymbol{A}_t^t) = (\sqrt[4]{\gamma}a)^t$$

Therefore the system (5.20) is mean square stable if $\sqrt[4]{\gamma}a < 1$. In the GR model based implementation:

$$\sigma(\sqrt[4]{P(n_1, n_2, t_0, t + t_0)}\boldsymbol{A}_t^t) = (\sqrt[4]{\gamma}a_9)^t$$

Therefore the system (5.22) is mean square stable if $\sqrt[4]{\gamma}a_9 < 1$.

5.4 Input-Output Stability

Input-output stability of 3-D systems implemented on grid sensor networks with node and link failure and its relationship with the internal stability of the same is studied in this section. A stochastic notion of input output stability is employed due to randomness of systems under consideration.

Definition 5.4.1 *The sensor network is said to be bounded input bounded output stable in the mean square sense (BIBOMS), if for an input with a bounded mean square value output at every node has a bounded mean square value.*

Input-output stability of grid sensor networks under link failure and node failure are treated separately in this work.

5.4.1 Input-Output Stability Under Link Failure

It is assumed that link failure is statistically independent of the input signal.

Lemma 5.4.1 *Let the output at node* (n_1, n_2) *at time* t *due to the input at node* (n_1^1, n_2^1) *at time* t^1 *be denoted by* $\boldsymbol{y}_{(n_1^1,n_2^1,t^1)}(n_1, n_2, t)$. *If the sensor network is asymptotically stable in the mean square sense* $\exists\ \mu_{n_1n_2} > 0, \lambda_{n_1n_2} \in [0, 1)$ *and* $k_{n_1n_2} \in \mathbb{Z}^+$ *such that* $\|\boldsymbol{y}_{(n_1^1,n_2^1,t^1)}(n_1, n_2, t)\| < \mu_{n_1n_2}(t - t^1)^{k_{n_1n_2}}\lambda_{n_1n_2}^{t-t^1}\|\boldsymbol{u}(n_1^1, n_2^1, t^1)\|$.

Proof For implementations using the FM model the result follows directly from Lemma 5.2.2 and Theorem 5.2.1. For GR model based implementations the result follows directly from Lemma 5.2.5 and Theorem 5.2.3.

Theorem 5.4.1 *Systems* (5.3) *and* (5.5) *are BIBOMS if they are mean square stable:*

Proof Systems (5.3) and (5.5) are linear. Hence:

$$\boldsymbol{y}(n_1, n_2, t) = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=-\infty}^{t} \boldsymbol{y}_{(i,j,k)}(n_1, n_2, t) \tag{5.41}$$

$$\begin{aligned}\|\boldsymbol{y}(n_1, n_2, t)\| &\leq \sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=-\infty}^{t} \|\boldsymbol{y}_{(i,j,k)}(n_1, n_2, t)\| \\ &\leq \sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=-\infty}^{t} \mu_{n_1n_2}(t - k)^{k_{n_1n_2}}\lambda_{n_1n_2}^{t-k}\|\boldsymbol{u}(i, j, k)\|\end{aligned}$$

Since $\|\boldsymbol{u}(i, j, k)\|$ is bounded so is $\|\boldsymbol{y}(i, j, k)\|$. This completes the proof.

5.4.2 Input-Output Stability Under Node Failure

Input output stability under permanent and temporary node failure is treated separately in this work.

5.4.2.1 Under Permanent Node Failure

Due to the same considerations that led to adopting a notion of asymptotic stability when configuration is frozen in time, input-output stability when configuration is frozen in time is studied for sensor networks.

Definition 5.4.2 *The sensor network is said to be bounded input bounded output stable when configuration is frozen in time, if for a bounded input the output is bounded when the set of currently operational nodes remain unchanged.*

To determine the BIBO stability of a sensor network according to the above definition the set of operational nodes is fixed to be invariant. Therefore the system of which BIBO stability when configuration is frozen in time has to be determined becomes a deterministic system.

Theorem 5.4.2 *Systems* (5.24) *and* (5.26) *are BIBO stable when configuration is frozen in time if they are are asymptotically stable in the current mode of operation.*

Proof The result follows directly from the well know result in system theory, that if a linear time invariant system described by a state space model is asymptotically stable it is also BIBO stable.

5.4.2.2 Under Temporary Node Failure

Theorem 5.4.3 *System* (5.32) *and* (5.34) *are BIBOMS if conditions* (5.37) *and* (5.40) *are satisfied respectively.*

Proof Proof is similar to the proof of Theorem 5.3.3 and is omitted for the sake of brevity.

CHAPTER 6

EXAMPLE

Algorithms performing linear operations on sensor measurements or functions of them are used in a wide variety of applications. Therefore the approach proposed in this work is applicable in many distributed signal processing applications. Implementation of a distributed Kalman filter and a contaminant front detector will be discussed to illustrate the state space model based approach proposed in this work.

6.1 Distributed Kalman Filtering

Let the evolution of a dynamic process driven by zero mean Gaussian noise be described by:

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{w}(t) \tag{6.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times p}$. State vector $\boldsymbol{x}(t) \in \mathbb{R}^n$ and input $\boldsymbol{w}(t) \in \mathbb{R}^p$ is drawn from a zero mean multivariate white Gaussian noise process. Let $E\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T\} = \boldsymbol{Q}(t)$. Initial state $\boldsymbol{x}(0)$ is Gaussian distributed with mean zero and covariance matrix $\boldsymbol{P}(0)$.

The output of the dynamic system is measured by sensor nodes in a grid sensor

network of size $N_1 \times N_2$. Let the sensing model at node (n_1, n_2) be given by:

$$\boldsymbol{y}(n_1, n_2, t) = \boldsymbol{H}(t)x(t) + \boldsymbol{v}(n_1, n_2, t) \tag{6.2}$$

where $\boldsymbol{H}(n_1, n_2, t) \in \mathbb{R}^{m \times n}$ and output vector $\boldsymbol{y}(n_1, n_2, t) \in \mathbb{R}^m$. The measurement at node (n_1, n_2) is corrupted by zero mean white Gaussian noise $\boldsymbol{v}(n_1, n_2, t)$ with covariance matrix $\boldsymbol{R}(t)$. Noise $\boldsymbol{v}(n_1, n_2, t)$ at different nodes is assumed to be uncorrelated. Measurements made by nodes are linearly related to the state of the dynamic process by the matrix $\boldsymbol{H}(t)$ which is independent of the node.

The objective is to estimate, at each node, the state of the dynamic process in collaboration with other nodes in the sensor network. The collaboration strategy that can be used depends on inter-node communication allowed by the sensor network. Let inter-node communication be restricted such that:

1. Node (n_1, n_2) can communicate only with nodes $(n_1 - 1, n_2)$, $(n_1, n_2 - 1)$, $(n_1 + 1, n_2)$ and $(n_1, n_2 + 1)$.

2. There is no information relaying over multiple hops in a single time slot.

The DKF problem is to estimate, at each node, the state of the dynamic process $\boldsymbol{x}(t)$ using measurements up to time t when inter-node communication is restricted by constraints (1) and (2) above.

6.1.1 The Proposed Algorithm

Let $S = [0\ ,\ N_1 - 1] \times [0\ ,\ N_2 - 1]$ and $S(n_1, n_2, t) = \{\boldsymbol{y}(i, j, k) | (i, j) \in S, 0 \leq k \leq t\ and\ \mid n_1 - i \mid + \mid n_2 - j \mid \leq t - k + 1\}$. A measurement made by a node can be conveyed over only a single hop in a single time slot due to communication constraints (1) and (2). Therefore $S(n_1, n_2, t)$ is the set of measurements that can be made available

for a computation at node (n_1, n_2) at time slot t, when inter-node communication is restricted by constraints (1) and (2).

The linear minimum mean square error(LMMSE) estimate of $\boldsymbol{x}(t)$ given $S(n_1, n_2, t)$ is a linear combination of the elements of $S(n_1, n_2, t)$. Therefore in principle LMMSE estimator of $\boldsymbol{x}(t)$ at node (n_1, n_2) given $S(n_1, n_2, t)$ can be implemented as a linear filter operating on sensor measurements in $S(n_1, n_2, t)$. But even for simple examples, the filters required for estimation are complex.

Therefore the focus of this paper in not to obtain at each node (n_1, n_2) the LMMSE estimate of $\boldsymbol{x}(t)$ given $S(n_1, n_2, t)$. Rather a DKF algorithm that requires minimal inter-node communication is proposed in this paper.

In order to make the rationale behind the proposed algorithm more evident, LMMSE estimator of $\boldsymbol{x}(t)$ given measurements of all the nodes in the sensor network up to time t, is discussed briefly in the following. Let:

$$\boldsymbol{Y}(t) = [\boldsymbol{y}(1,1,t)^T \; \boldsymbol{y}(1,2,t)^T \; \boldsymbol{y}(1,3,t)^T \cdots \boldsymbol{y}(N_1,N_2,t)^T]^T$$

$$\boldsymbol{V}(t) = [\boldsymbol{v}(1,1,t)^T \; \boldsymbol{v}(1,2,t)^T \; \boldsymbol{v}(1,3,t)^T \cdots \boldsymbol{v}(N_1,N_2,t)^T]^T$$

and

$$\mathbb{H}(t) = [\boldsymbol{H}(t)^T \; \boldsymbol{H}(t)^T \; \boldsymbol{H}(t)^T \cdots \boldsymbol{H}(t)^T]^T$$

Then we have:

$$\boldsymbol{Y}(t) = \mathbb{H}(t)\boldsymbol{x}(t) + \boldsymbol{V}(t) \tag{6.3}$$

Let $G(t) = \{\boldsymbol{y}(i,j,k) | 0 \leq i \leq N_1 - 1 \; and \; 0 \leq j \leq N_2 - 1 \; and \; 0 \leq k \leq t\}$ and the LMMSE estimate of $\boldsymbol{x}(t_1)$ given $G(t_2)$ be denoted by $\hat{E}(\boldsymbol{x}(t_1) \mid G(t_2))$. Let

$\mathbb{R}(i) = E\{\boldsymbol{V}(t)\boldsymbol{V}(t)^T\}$. The Kalman filter algorithm given in figure 6.1 computes the LMMSE estimate of $\boldsymbol{x}(t)$ given $G(t)$.

1: **for** $k = 0$ to t **do**
2: **if** k=0 **then**
3: $\bar{\boldsymbol{x}}(k) = \mu_{\boldsymbol{x}(0)}$
4: **end if**
5: $\mathbb{M}(k) = (\boldsymbol{P}(k)^{-1} + \mathbb{H}(k)^T\mathbb{R}(k)^{-1}\mathbb{H}(k))^{-1}$
6: $\mathbb{K}(k) = \mathbb{M}(k)\mathbb{H}(k)^T\mathbb{R}(k)^{-1}$
7: $\hat{\boldsymbol{x}}(k) = \bar{\boldsymbol{x}}(k) + \mathbb{K}(k)(\boldsymbol{Y}(k) - \mathbb{H}(k)\bar{\boldsymbol{x}}(k))$
8: $\boldsymbol{P}(k+1) = \boldsymbol{A}\boldsymbol{M}(k)\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{Q}(k)\boldsymbol{B}^T$
9: $\bar{\boldsymbol{x}}(k+1) = A\hat{\boldsymbol{x}}(k)$
10: **end for**

Figure 6.1. Algorithm for global Kalman filtering

In the algorithm in figure 6.1, $\hat{\boldsymbol{x}}(k) = \hat{E}(\boldsymbol{x}(k) \mid G(k))$ and $\bar{\boldsymbol{x}}(k) = \hat{E}(\boldsymbol{x}(k) \mid G(k-1))$.

Theorem 6.1.1 *Let the algorithm given in figure 6.2 be run by every node in the sensor network. Then we have:*

$$\hat{\boldsymbol{x}}(k) = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\hat{\boldsymbol{x}}(i,j,k) \tag{6.4}$$

Proof Let the statement be true for $k = p - 1$. Then it is easily seen that:

1: **for** $k = 0$ to t **do**
2: **if** k=0 **then**
3: $\bar{\boldsymbol{x}}(n_1, n_2, k) = \frac{1}{N_1 N_2}\mu_{\boldsymbol{x}(0)}$
4: **end if**
5: $\mathbb{M}(k) = (\boldsymbol{P}(k)^{-1} + \mathbb{H}(k)^T \mathbb{R}(k)^{-1}\mathbb{H}(k))^{-1}$
6: $\boldsymbol{K}(k) = \mathbb{M}(k)\boldsymbol{H}(k)^T \boldsymbol{R}(k)^{-1}$
7: $\hat{\boldsymbol{x}}(n_1, n_2, k) = \bar{\boldsymbol{x}}(n_1, n_2, k) + \boldsymbol{K}(k)(\boldsymbol{y}(n_1, n_2, k) - N_1 N_2 \boldsymbol{H}(k)\bar{\boldsymbol{x}}(n_1, n_2, k))$
8: $P(k+1) = \boldsymbol{A}\boldsymbol{M}(k)\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{Q}(k)\boldsymbol{B}^T$
9: $\bar{\boldsymbol{x}}(n_1, n_2, k+1) = \boldsymbol{A}\hat{\boldsymbol{x}}(n_1, n_2, k)$
10: **end for**

Figure 6.2. Algorithm for local Kalman filtering at node (n_1, n_2)

$$\bar{\boldsymbol{x}}(p) = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \bar{\boldsymbol{x}}(i, j, p)$$

Substituting in step 7 of the algorithm in figure 6.1 we have:

$$\begin{aligned}
\hat{\boldsymbol{x}}(p) &= \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \bar{\boldsymbol{x}}(i, j, p) + \mathbb{K}(p)\{\boldsymbol{Y}(p) - \mathbb{H}(p)\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \bar{\boldsymbol{x}}(i, j, p)\} \\
&= \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \bar{\boldsymbol{x}}(i, j, p) + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \boldsymbol{K}(p)\boldsymbol{y}(i, j, p) \\
&- \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{r=1}^{N_1}\sum_{s=1}^{N_2} \boldsymbol{K}(p)\boldsymbol{H}(p)\bar{\boldsymbol{x}}(i, j, p) \\
&= \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \hat{\boldsymbol{x}}(i, j, p)
\end{aligned}$$

The statement holds for $k = p$. Since:

$$\bar{\boldsymbol{x}}(0) = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \bar{\boldsymbol{x}}(i, j, 0)$$

using the case of $p = 0$ in the above derivation it can be shown that the statement holds for $p = 0$. Therefore by the principle of mathematical induction statement (6.4) holds for any finite k. This completes the proof of the theorem.

The following result on the expected value of the estimates of local Kalman filters will be useful later.

Theorem 6.1.2 *The local Kalman filter algorithm given in figure 6.2, provides an unbiased estimate of* $\frac{1}{N_1 N_2}\boldsymbol{x}(t)$. *Equivalently:*

$$E\{\hat{\boldsymbol{x}}(n_1, n_2, t) - \frac{1}{N_1 N_2}\boldsymbol{x}(t)\} = 0 \tag{6.5}$$

Proof Let the statement (6.5) be true for $t = p - 1$. We have:

$$E\{\bar{\boldsymbol{x}}(n_1, n_2, p) - \frac{1}{N_1 N_2}\boldsymbol{x}(p)\} = 0$$

Since $E\{\boldsymbol{y}(n_1, n_2, p) - N_1 N_2 \boldsymbol{H}(p)\bar{\boldsymbol{x}}(n_1, n_2, p)\} = 0$ we have:

$$E\{\hat{\boldsymbol{x}}(n_1, n_2, p) - \frac{1}{N_1 N_2}\boldsymbol{x}(p)\} = 0$$

The statement (6.5) holds for $t = p$. Moreover $E\{\bar{\boldsymbol{x}}(n_1, n_2, 0) - \frac{1}{N_1 N_2}\boldsymbol{x}(0)\} = 0$. It can be easily seen that:

$$E\{\hat{\boldsymbol{x}}(n_1, n_2, 0) - \frac{1}{N_1 N_2}\boldsymbol{x}(0)\} = 0$$

Therefore by the principle of mathematical induction, statement (6.5) holds for any finite t. This completes the proof of the theorem.

The LMMSE estimate of $\boldsymbol{x}(t)$ given measurements of all the nodes up to time t can

be obtained by summing the local estimates, of all the nodes in the sensor network. Summation given by (6.4) cannot be computed under communication constraints (1) and (2). Therefore we seek a summation of local estimates that can be performed at every node under prevailing restrictions on inter-node communication. It is desirable for the summation of the estimates of local Kalman filters to have the following properties.

- For the computation at node (n_1, n_2), the latest possible estimate is taken from node $(n_1 + i, n_2 + j)$, where $i \neq 0$ or $j \neq 0$. It is easily seen that under communication constraints (1) and (2) the latest estimate that can be included from node (n_1+i, n_2+j) for a computation at node (n_1, n_2) at time t is $\hat{\boldsymbol{x}}(n_1+i, n_2+j, t-|i|-|j|+1)$.

- When computing the summation of local estimates at node (n_1, n_2) at time t, $\hat{\boldsymbol{x}}(n_1 + i, n_2 + j, t - |i| - |j| + 1)$, the local estimate obtained from node $(n_1 + i, n_2 + j)$ is weighted by $\boldsymbol{A}^{|i|+|j|-1}$. As would be explained in section 6.1.2.1, the intention of aforementioned weighting is to make the final estimate unbiased.

Estimate of $\boldsymbol{x}(t)$ at node (n_1, n_2) is computed as a weighted summation of estimates of local Kalman filters according to the following equation:

$$\begin{aligned}\hat{\boldsymbol{x}}_D(n_1, n_2, t) =& \frac{N_1 N_2}{C(n_1, n_2, t) + 1}\{\hat{\boldsymbol{x}}(n_1, n_2, t) \\ &+ \sum_{(i,j)\in S_n(n_1,n_2,t)} \boldsymbol{A}^{|i|+|j|-1}\hat{\boldsymbol{x}}(n_1 + i, n_2 + j, t - |i| - |j| + 1)\} \end{aligned} \quad (6.6)$$

where:

$$\begin{aligned} S_n(n_1,n_2,t) =& \{(i,j) : (n_1+i, n_2+j) \in [0,\ N_1-1] \times [0,\ N_2-1] \ and \\ & (i,j) \neq (0,0) \ and \ t-|i|-|j|+1 \geq 0\} \\ C(n_1,n_2,t) =& The \ number \ of \ elements \ in \ S_n(n_1,n_2,t) \end{aligned} \tag{6.7}$$

Values of i and j are restricted such that $t-|i|-|j|+1 \geq 0$ since local estimates $\hat{\boldsymbol{x}}(n_1,n_2,k)$ exist for time $k \geq 0$ only. When computing the estimate at node (n_1,n_2), summation of local estimates is taken over the entire sensor network, after a certain time t. Thereafter the scale factor $\frac{N_1N_2}{C(n_1,n_2,t)+1}$ becomes unity. Node (n_1,n_2) can compute $C(n_1,n_2,t)$ provided it knows its coordinates in the grid sensor network and the size of the sensor network.

The summation in (6.6) can be implemented as a 3-D linear filter operating on local Kalman filter estimates. Impulse response of the linear filter required to perform the summation in (6.6) is as follows:

$$h(n_1,n_2,t) = \begin{cases} \boldsymbol{I}_n & n_1=n_2=0 \\ \boldsymbol{A}^{|n_1|+|n_2|-1}\delta[n_1+n_2-1-t] & otherwise \end{cases} \tag{6.8}$$

where $\delta : \mathbb{Z} \rightarrow \mathbb{R}$ is the unit impulse function.

According to theorems 3.1.1 and 3.2.1 impulse response (6.8) can be realized in a grid sensor network using methods given in the chapter 2 under communication constraints (1) and (2).

The proposed algorithm for DKF in grid sensor networks is given in figure 6.3:

1: At each node (n_1, n_2) run the algorithm given in figure 6.2 using local measurements.

2: Perform computation given by (6.6) at each node (n_1, n_2) to obtain $\hat{\boldsymbol{x}}_D(n_1, n_2, t)$ the final estimate of $x(t)$ at node (n_1, n_2).

Figure 6.3. Algorithm for DKF

Since the computation given by (6.6) can be performed at every node using the methods given in chapter 2 the DKF algorithm given above can be implemented under communication constrains (1) and (2).

6.1.2 Mean and Mean Square Error Performance

6.1.2.1 Mean Error Performance of the Algorithm

Mean error performance of the algorithm is summarized in the following theorem.

Theorem 6.1.3

$$E\{\hat{\boldsymbol{x}}_D(n_1, n_2, t) - \boldsymbol{x}(t)\} = 0 \tag{6.9}$$

Therefore the DKF algorithm given in figure 6.2 is unbiased.

Proof From theorem 6.1.2 we have that:

$$E\{\hat{\boldsymbol{x}}(n_1, n_2, t) - \frac{1}{N_1 N_2}\boldsymbol{x}(t)\} = 0 \tag{6.10}$$

and

$$E\{\boldsymbol{A}^{|i|+|j|-1}\hat{\boldsymbol{x}}(n_1+i,n_2+j,t-|i|-|j|+1)-\frac{1}{N_1N_2}\boldsymbol{x}(t)\}=0 \tag{6.11}$$

The result (6.9) follows readily from (6.6), (6.10) and (6.11).

While $\boldsymbol{A}^{|i|+|j|-1}\hat{\boldsymbol{x}}(n_1+i,n_2+j,t-|i|-|j|+1)$ is an unbiased estimate of $\frac{1}{N_1N_2}\boldsymbol{x}(t)$, $\hat{\boldsymbol{x}}(n_1+i,n_2+j,t-|i|-|j|+1)$ is in general not. Therefore weighting $\hat{\boldsymbol{x}}(n_1+i,n_2+j,t-|i|-|j|+1)$ by $\boldsymbol{A}^{|i|+|j|-1}$ in the summation (6.6) enables the distributed estimate to be unbiased.

6.1.2.2 Mean Square Error Performance of the Algorithm

Let:

$$\hat{\boldsymbol{e}}(n_1,n_2,t,t-k)=\frac{1}{N_1N_2}\boldsymbol{x}(t)-\boldsymbol{A}^k\hat{\boldsymbol{x}}(n_1,n_2,t-k)$$
$$\bar{\boldsymbol{e}}(n_1,n_2,t,t-k)=\frac{1}{N_1N_2}\boldsymbol{x}(t)-\boldsymbol{A}^k\bar{\boldsymbol{x}}(n_1,n_2,t-k)$$

$\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,t,t-k)=E\{\hat{\boldsymbol{e}}(n_1,n_2,t,t-k)\hat{\boldsymbol{e}}(n_1,n_2,t,t-k)^T\}$ and $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,t,t-k)=E\{\bar{\boldsymbol{e}}(n_1,n_2,t,t-k)\bar{\boldsymbol{e}}(n_1,n_2,t,t-k)^T\}$ for $t\geq k$.

Lemma 6.1.1 *$\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,t,t-k)$ and $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,t,t-k)$ can be computed iteratively using the algorithm given in figure 6.4.*

Proof We have:

$$\begin{aligned}\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,0)&=E\{\frac{1}{N_1N_2}(\boldsymbol{x}(0)-\mu_{\boldsymbol{x}(0)})\frac{1}{N_1N_2}(\boldsymbol{x}(0)-\mu_{\boldsymbol{x}(0)})^T\}\\&=\left(\frac{1}{N_1N_2}\right)^2\boldsymbol{P}(0)\end{aligned} \tag{6.12}$$

1: **for** $p = 0$ to t **do**
2: **if** p=0 **then**
3: $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, 0, 0) = \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{P}(0)$
4: **end if**
5: **if** $p \leq t - k$ **then**
6: $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1, n_2, p, p) = (I - \boldsymbol{K}(p)\boldsymbol{H}(p))\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, p, p)(\boldsymbol{I} - \boldsymbol{K}(p)\boldsymbol{H}(p))^T + \boldsymbol{K}(p)\boldsymbol{R}(p)\boldsymbol{K}(p)^T$
7: $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, p+1, p+1) = \boldsymbol{A}\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1, n_2, p, p)\boldsymbol{A}^T + \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T$
8: **else**
9: $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1, n_2, p, t-k) = \boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, p, t-k+1)$
10: $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, p+1, t-k+1) = \boldsymbol{A}\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1, n_2, p, t-k)\boldsymbol{A}^T + \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T$
11: **end if**
12: **end for**

Figure 6.4. Algorithm for computing $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1, n_2, t, t-k)$ and $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, t, t-k)$

For $p \leq t - k$:

$$\begin{aligned}
\hat{\boldsymbol{e}}(n_1, n_2, p, p) &= \frac{1}{N_1 N_2}\boldsymbol{x}(p) - \hat{\boldsymbol{x}}(n_1, n_2, p) \\
&= \frac{1}{N_1 N_2}\boldsymbol{x}(p) - \bar{\boldsymbol{x}}(n_1, n_2, p) - \boldsymbol{K}(t)\boldsymbol{y}(n_1, n_2, p) \\
&+ N_1 N_2 \boldsymbol{K}(p)\boldsymbol{H}(p)\bar{\boldsymbol{x}}(n_1, n_2, p) \\
&= \frac{1}{N_1 N_2}\boldsymbol{x}(p) - \bar{\boldsymbol{x}}(n_1, n_2, p) - \boldsymbol{K}(p)\boldsymbol{v}(n_1, n_2, p) \\
&- \boldsymbol{K}(p)\boldsymbol{H}(p)\boldsymbol{x}(p) + N_1 N_2 \boldsymbol{K}(p)\boldsymbol{H}(p)\bar{\boldsymbol{x}}(n_1, n_2, p) \\
&= \{\boldsymbol{I} - N_1 N_2 \boldsymbol{K}(p)\boldsymbol{H}(p)\}\bar{e}(n_1, n_2, p) - \boldsymbol{K}(p)\boldsymbol{v}(n_1, n_2, p) \qquad (6.13)
\end{aligned}$$

From (6.13) we have:

$$P_{\hat{\boldsymbol{e}}}(n_1,n_2,p) = (\boldsymbol{I} - N_1N_2\boldsymbol{K}(p)\boldsymbol{H}(p))\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,t)(I - N_1N_2\boldsymbol{K}(p)\boldsymbol{H}(p))^T + \boldsymbol{K}(p)\boldsymbol{R}(p)\boldsymbol{K}(p)^T \tag{6.14}$$

$$\begin{aligned}\bar{\boldsymbol{e}}(n_1,n_2,p+1,p+1) =& \frac{1}{N_1N_2}\boldsymbol{x}(p+1) - A\hat{\boldsymbol{x}}(n_1,n_2,p) \\ =& \boldsymbol{A}\hat{\boldsymbol{e}}(n_1,n_2,p,p) + \frac{1}{N_1N_2}\boldsymbol{B}\boldsymbol{w}(p)\end{aligned} \tag{6.15}$$

From (6.15) we have:

$$\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,p+1,p+1) = \boldsymbol{A}\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,p,p)\boldsymbol{A}^T + \left(\frac{1}{N_1N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T \tag{6.16}$$

It is evident from equations (6.12), (6.14) and (6.16) that steps 1-7 of the algorithm given in figure 6.4 can be used to compute $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,p,p)$ and $\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,p,p)$ for $p \leq t-k$.

For $t \geq p > t-k$:

$$\begin{aligned}\hat{\boldsymbol{e}}(n_1,n_2,p,t-k) =& \frac{1}{N_1N_2}\boldsymbol{x}(p) - \boldsymbol{A}^{p-t+k}\hat{\boldsymbol{x}}(n_1,n_2,t-k) \\ =& \frac{1}{N_1N_2}\boldsymbol{x}(p) - \boldsymbol{A}^{p-t+k-1}\bar{\boldsymbol{x}}(n_1,n_2,t-k+1) \\ =& \bar{\boldsymbol{e}}(n_1,n_2,p,t-k+1)\end{aligned}$$

Therefore:

$$\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,p,t-k) = \boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,p,t-k+1) \tag{6.17}$$

$$\begin{aligned}\bar{\boldsymbol{e}}(n_1,n_2,p+1,t-k+1) =& \frac{1}{N_1N_2}\boldsymbol{x}(p+1) - \boldsymbol{A}^{p-t+k+1}\hat{\boldsymbol{x}}(n_1,n_2,t-k)\\ =& \boldsymbol{A}\hat{\boldsymbol{e}}(n_1,n_2,p,t-k) + \frac{1}{N_1N_2}\boldsymbol{B}\boldsymbol{w}(p)\end{aligned}$$

Therefore:

$$\boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1,n_2,p+1,t-k+1) = \boldsymbol{A}\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,p,t-k)\boldsymbol{A}^T + \left(\frac{1}{N_1N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T \quad (6.18)$$

Steps 9 and 10 of the algorithm given in figure 6.4 can be used to compute $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,p,t-k)$ for $t \geq p > t-k$. This completes the proof of lemma 6.1.1.

Since $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,t,t-k)$ is independent of n_1 and n_2, $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(t,t-k)$ would be used instead of $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1,n_2,t,t-k)$ in the rest of this work for notational simplicity.

Let:

$$\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l) = E\{\hat{\boldsymbol{e}}(n_1,n_2,t,t-k)\hat{\boldsymbol{e}}(i,j,t,t-l)^T\}$$

and

$$\bar{\boldsymbol{P}}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l) = E\{\bar{\boldsymbol{e}}(n_1,n_2,t,t-k)\bar{\boldsymbol{e}}(i,j,t,t-l)^T\}$$

where $(n_1,n_2) \neq (i,j)$, $t \geq k$ and $t \geq l$.

By the definition of $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l)$ we have:

$$\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l) = \boldsymbol{P}_{\check{\boldsymbol{e}}}(i,j,n_1,n_2,t,t-l,t-k)^T$$

Therefore, if $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l)$ can be computed for $l \geq k$ it can also be computed for $l < k$.

Lemma 6.1.2 *When $l \geq k$, $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l)$ and $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1,n_2,i,j,t,t-k,t-l)$ can be computed iteratively using the algorithm given in figure 6.5.*

Proof Proof is similar to the proof of lemma 6.1.1 and is omitted for the sake of brevity.

1: **for** $p = 0$ to t **do**
2: **if** $p = 0$ **then**
3: $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, 0, 0, 0) = \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{P}(0)$
4: **end if**
5: **if** $p \leq t - l$ **then**
6: $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, p, p, p) = (\boldsymbol{I} - \boldsymbol{K}(p)\boldsymbol{H}(p))\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1, n_2 i, j, p, p, p)(\boldsymbol{I} - \boldsymbol{K}(p)\boldsymbol{H}(p))^T$
7: $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1, n_2, i, j, p+1, p+1, p+1) = \boldsymbol{A}\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, p, p, p)\boldsymbol{A}^T + \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T$
8: **else if** $t - k \geq p > t - l$ **then**
9: $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, p, p, t-l) = (I - \boldsymbol{K}(p)\boldsymbol{H}(p))\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1, n_2, i, j, p, p, t-l+1)$
10: $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1, n_2, i, j, p+1, p+1, t-l+1) = \boldsymbol{A}\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, p, p, t-l)\boldsymbol{A}^T + \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T$
11: **else**
12: $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, p, t-k, t-l) = \boldsymbol{P}_{\bar{\boldsymbol{e}}}(n_1, n_2, i, j, p, t-k+1, t-l+1)$
13: $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1, n_2, p+1, t-k+1, t-l+1) = \boldsymbol{A}\boldsymbol{P}_{\hat{\boldsymbol{e}}}(n_1, n_2, i, j, p, t-k, t-l)\boldsymbol{A}^T + \left(\frac{1}{N_1 N_2}\right)^2 \boldsymbol{B}\boldsymbol{Q}(p)\boldsymbol{B}^T$
14: **end if**
15: **end for**

Figure 6.5. Algorithm for computing $\boldsymbol{P}_{\tilde{e}}(n_1, n_2, i, j, p, t-k, t-l)$ and $\boldsymbol{P}_{\check{\boldsymbol{e}}}(n_1, n_2, p+1, t-k+1, t-l+1)$

Since $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, t, t-k, t-l)$ is independent of n_1, n_2, i and j, $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(t, t-k, t-l)$ would be will be used instead of $\boldsymbol{P}_{\tilde{\boldsymbol{e}}}(n_1, n_2, i, j, t, t-k, t-l)$ in the rest of this work

for notational simplicity.

Estimation error of the DKF algorithm, at node (n_1, n_2) at time t is given by:

$$\boldsymbol{e}_D(n_1, n_2, t) = \boldsymbol{x}(t) - \boldsymbol{x}_D(n_1, n_2, t)$$

Let the covariance of the estimation error at node (n_1, n_2) at time t be $\boldsymbol{P}_D(n_1, n_2, t) = E\{\boldsymbol{e}_D(n_1, n_2, t)\boldsymbol{e}_D(n_1, n_2, t)^T\}$

Theorem 6.1.4

$$\begin{aligned}\boldsymbol{P}_D(n_1, n_2, t) = \left(\frac{N_1 N_2}{C(n_1, n_2, t) + 1}\right)^2 \{ &\sum_{(i,j)\in S} \boldsymbol{P}_{\hat{e}}(t, t - |i| - |j| + 1) \\ &+ \sum_{(i,j)\in S} \sum_{(p,q)\in S \ (p,q)\neq(i,j)} \boldsymbol{P}_{\hat{\boldsymbol{e}}}(t, t - |i| - |j| + 1, t - |p| - |q| + 1)\} \quad (6.19)\end{aligned}$$

Proof By substituting for $\boldsymbol{x}_D(n_1, n_2, t)$ from (6.6) we have:

$$\begin{aligned}\boldsymbol{e}_D(n_1, n_2, t) =& \frac{N_1 N_2}{C(n_1, n_2, t) + 1}\{\frac{1}{N_1 N_2}\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(n_1, n_2, t) \\ &+ \sum_{(i,j)\in S_n(n_1,n_2,t)} \frac{1}{N_1 N_2}\boldsymbol{x}(t) - \boldsymbol{A}^{|i|+|j|-1}\hat{\boldsymbol{x}}(n_1 + i, n_2 + j, t - |i| - |j| + 1)\} \\ =& \frac{N_1 N_2}{C(n_1, n_2, t) + 1} \sum_{(i,j)\in S_n(n_1,n_2,t)\bigcup(0,0)} \hat{\boldsymbol{e}}(n_1 + i, n_2 + j, t, t - |i| - |j| + 1)\end{aligned} \quad (6.20)$$

Equation (6.19) can be derived by taking the covariance of $\boldsymbol{e}_D(n_1, n_2, t)$ and replacing terms $E\{\hat{\boldsymbol{e}}(n_1, n_2, t, t-k)\boldsymbol{e}(n_1, n_2, t, t-k)^T\}$ and $E\{\hat{\boldsymbol{e}}(n_1, n_2, t, t-k)\hat{\boldsymbol{e}}(i, j, t, t-l)^T\}$ by $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(t, t-k)$ and $\boldsymbol{P}_{\hat{\boldsymbol{e}}}(t, t-k, t-l)$ respectively.

6.1.3 Simulation Results

Let the evolution of the system under observation be described by:

$$\boldsymbol{x}(t+1) = \begin{bmatrix} 0.9996 & -0.0300 \\ -0.0300 & 0.9996 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \boldsymbol{w}(t) \tag{6.21}$$

$\boldsymbol{x}(t) \in \mathbb{R}^2$. The input $\boldsymbol{w}(t)$ is drawn from a zero mean two dimensional white Gaussian noise process. Let $E\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T\} = 20\boldsymbol{I}_2$. The state space model describes the trajectory of an object on a circular path(not necessarily in the same space as the sensor network). Initial state is Gaussian distributed with mean $\mu_{\boldsymbol{x}(0)} = [15 \;\; 0]^T$ and covariance matrix $\boldsymbol{P}(0) = 20\boldsymbol{I}_2$. The dynamic system is observed by nodes of a grid sensor network of size 20×20. Sensing model at every node is given by:

$$\boldsymbol{y}(n_1, n_2, t) = [1 \quad 0]\boldsymbol{x}(t) + \boldsymbol{v}(n_1, n_2, t) \tag{6.22}$$

where $\boldsymbol{y}(n_1, n_2, t)$ and $\boldsymbol{v}(n_1, n_2, t)$ are scaler quantities. Measurement noise $\boldsymbol{v}(n_1, n_2, t)$ is drawn from a zero mean white Gaussian noise process with covariance
$E\{\boldsymbol{v}(n_1, n_2, t)\boldsymbol{v}(n_1, n_2, t)^T\} = 10$.

The proposed DKF algorithm was used to estimate the state of the system. The logarithm of the root mean square value of the estimation error, where the mean is taken over all the nodes, is plotted versus time in figure 6.6.

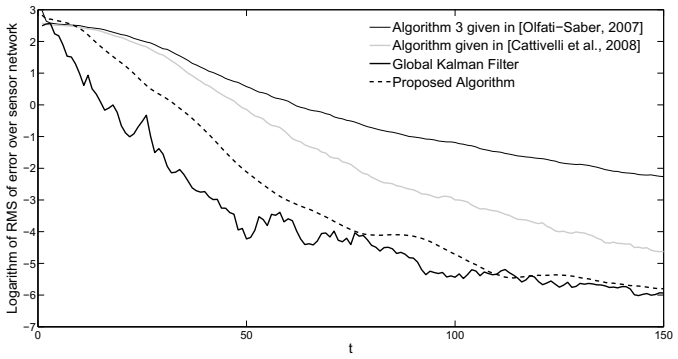


Figure 6.6. Logarithm of RMS of error over the sensor network

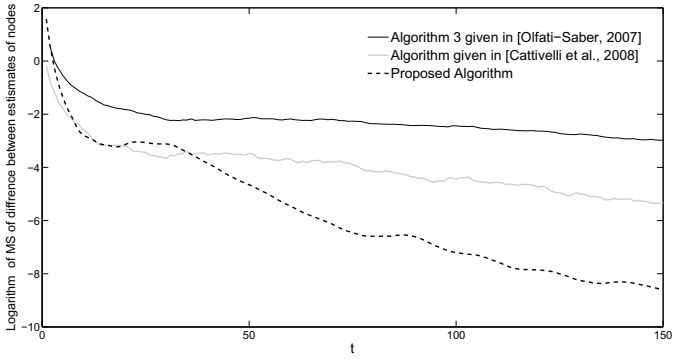


Figure 6.7. Logarithm of MS difference between estimates of nodes

The logarithm of the variance of estimates across the sensor network, is plotted versus time in figure 6.7. Variance of state estimates across the nodes is a measure of the mutual disagreement among nodes, on their estimates. This is an important

performance metric for a distributed estimation algorithm. The simulation was repeated with state transition equation changed to:

$$\boldsymbol{x}(t+1) = 0.95 \begin{bmatrix} 0.9996 & -0.0300 \\ -0.0300 & 0.9996 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \boldsymbol{w}(t) \tag{6.23}$$

The state space model (6.23) describes the trajectory of an object that converges to the origin along a logarithmic spiral. The logarithm of the root mean square value of the estimation error is plotted versus time in figure 6.8.

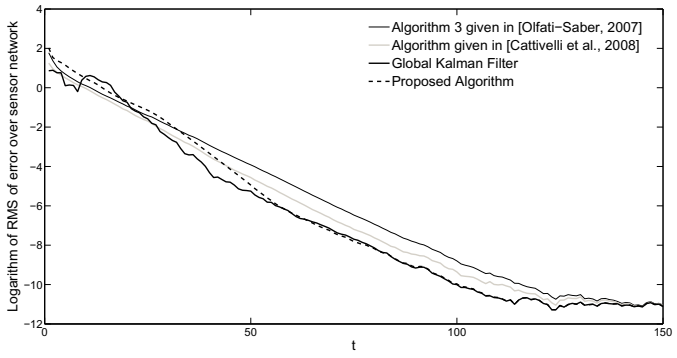


Figure 6.8. Logarithm of RMS of error over the sensor network

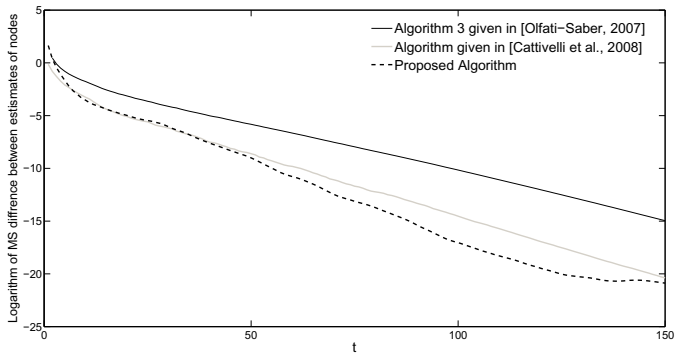


Figure 6.9. Logarithm of MS difference between estimates of nodes

The logarithm of the variance of estimates across the sensor network, is plotted versus time in figure 6.9.The state transition equation was changed to:

$$\boldsymbol{x}(t+1) = 1.05 \begin{bmatrix} 0.9996 & -0.0300 \\ -0.0300 & 0.9996 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \boldsymbol{w}(t) \tag{6.24}$$

and simulation was repeated again. The state space model (6.24) describes the trajectory of an object along a logarithmic spiral. The logarithm of the root mean square value of the estimation error is plotted versus time in figure 6.10.

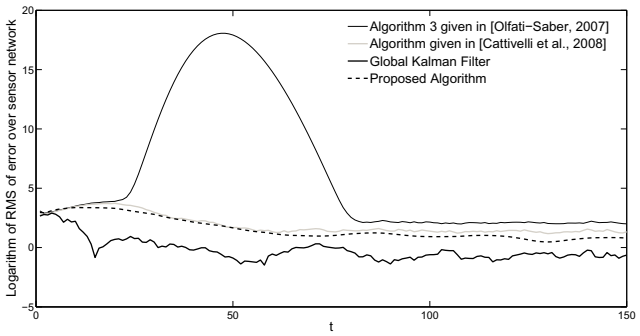


Figure 6.10. Logarithm of RMS of error over the sensor network

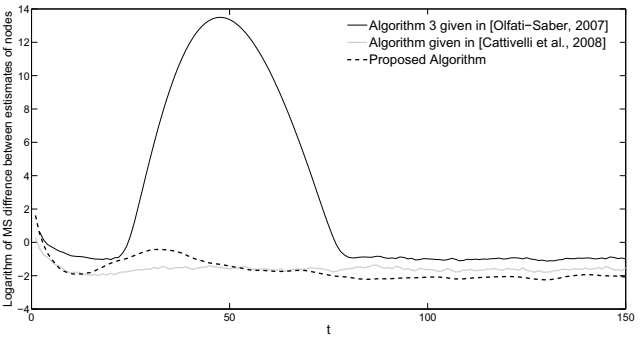


Figure 6.11. Logarithm of MS difference between estimates of nodes

The logarithm of the variance of estimates across the sensor network, is plotted versus time in figure 6.11.The proposed DKF algorithm outperforms existing DKF algorithms in simulated example applications.

Distributed Kalman filtering algorithms presented in the literature involve averaging

of local estimates across the sensor network to obtain a better estimate. Hence the performance of a DKF algorithm depends upon the strategy used to obtain network wide averages. In the algorithm proposed in this paper a 3-D linear filter is used to average estimates of local Kalman filters. In a grid sensor network a network wide average can be computed faster using linear filters than consensus filters used in existing DKF algorithms. Therefore authors expect the proposed DKF algorithm to outperform or at least match the performance of existing DKF algorithms even though a theoretical comparison of mean square error performance is not available yet.

6.2 Contaminant propagation detection

A method to detect the propagation of contaminants in air was proposed in Sumanasena and Bauer [2008]. The variation of intensity across the propagating front of the contaminant is detected using image processing techniques. The edge detection filter used, operates on sensor measurements at a single sampling instance. Therefore the system implemented on the sensor network is 2-D. It is assumed that the variation of intensity across the propagating front produce a unit step signal in sensor measurements. The impulse response of the filter used is given by:

$$h(n_1, n_2) = h_1(n_1)h_2(n_2)$$

$$h_1(n_1) = \begin{cases} \{0.511n_1(n_1+1) + 0.348n_1\}0.73^{n_1} & n_1 \geq 0 \\ -h_1(-n_1) & n_1 < 0 \end{cases}$$

$$h_2(n_2) = 0.0123\delta(n_2) + 0.3254(0.4)^{|n_2|} + 0.4020(0.44)^{|n_2|} - 0.7020(0.42)^{|n_2|}$$

The filter was derived by numerically maximizing the detection probability for a given worst case false detection probability within the class of third order separable 2-D fil-

ters. Impulse response of the filter can be decomposed into four quarter plane causal components. Let:

$$h_{1q}(n_1, n_2) = \begin{cases} h(n_1, n_2) & n_1 > 0 \; and \; n_2 > 0 \\ \frac{1}{2} h(n_1, n_2) & n_1 > 0 \; and \; n_2 = 0 \\ 0 & otherwise \end{cases}$$

We have:

$$h(n_1, n_2) = h_{1q}(n_1, n_2) - h_{1q}(-n_1, n_2) - h_{1q}(-n_1, -n_2) + h_{1q}(n_1, -n_2)$$

Each quarter plane causal filter can be realized in 2-D GR or FM state space models. Block diagram in figure 6.12 illustrates the implementation of the filter. Let

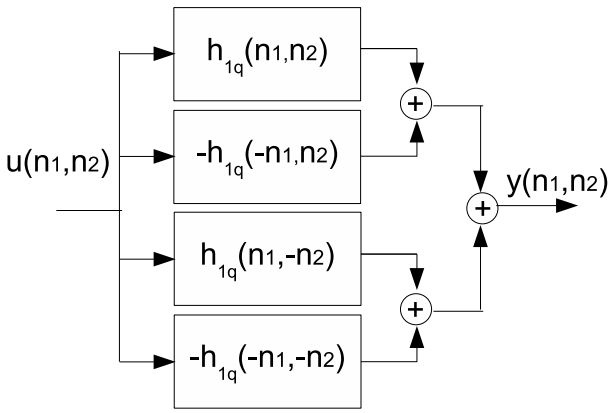
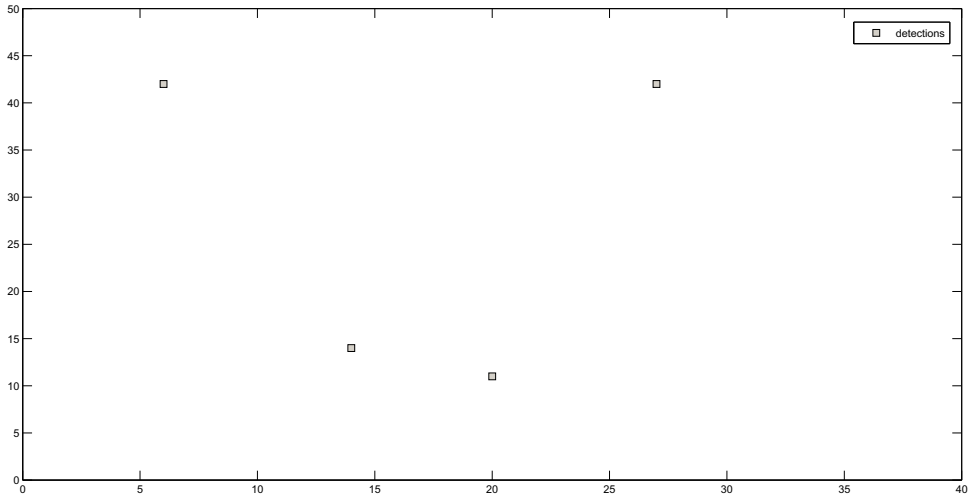


Figure 6.12. Implementation of the Filter

the output of the filter be $\boldsymbol{y}(n_1, n_2)$. Point (n_1, n_2) is marked as an edge point if

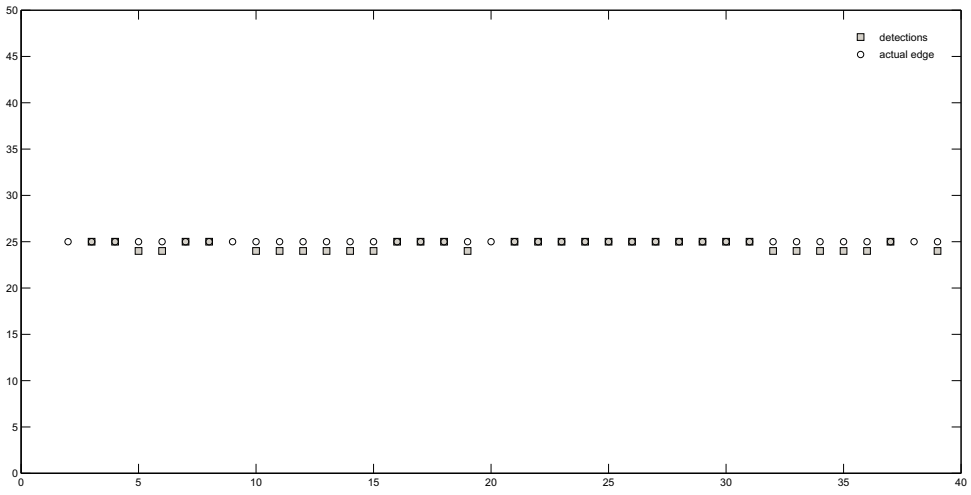
$\boldsymbol{y}(n_1, n_2) - \boldsymbol{y}(n_1 - 1, n_2) > 0.02$ and $\boldsymbol{y}(n_1, n_2) - \boldsymbol{y}(n_1 + 1, n_2) > 0.02$. The system was simulated on a sensor network of size 50×40. It is assumed that the input signal produced by the contaminant front is a unit step edge. Sensor readings are assumed to be contaminated by zero mean additive white Gaussian noise with variance of 0.5. False detection probability when there is no contaminant front is 0.0017.



$n_1 \uparrow$

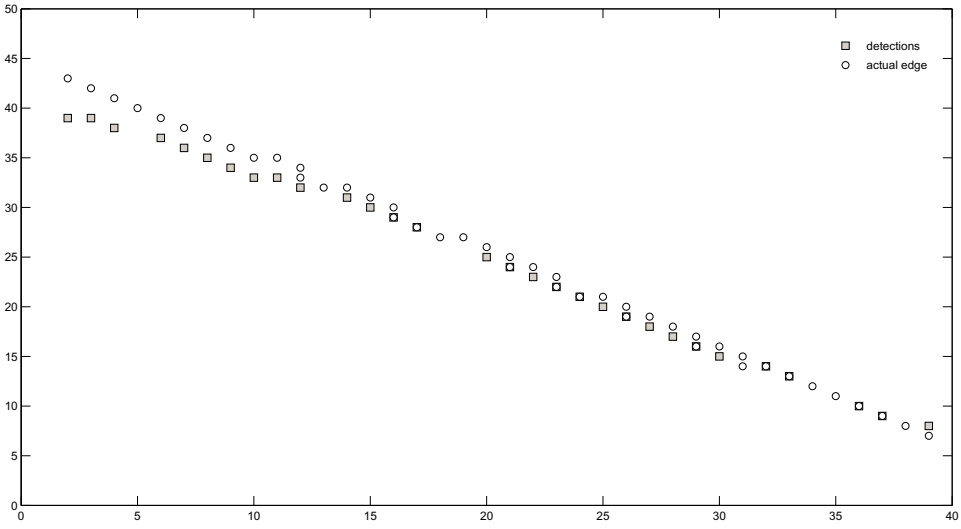
Figure 6.13. False detections with no input signal

False detections when there is no contaminant front is shown in figure 6.13. Figures 6.14 and 6.15 show the detection of contaminant fronts perpendicular to and having a 45^0 angle with the n_1 axis respectively.



$n_1 \uparrow$

Figure 6.14. Detection of the front perpendicular to the n_1 axis



$n_1 \uparrow$

Figure 6.15. Detection of the front making 45^0 to the n_1 axis

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Distributed implementation of m-D systems on sensor networks poses issues not raised in their centralized implementations. Open research issues that were not already addressed in chapters 2-6 are discussed in the following subsections.

7.1 Future Research Directions

7.1.1 Realization of Transfer Matrices

In the GR model based implementation node (n_1, n_2) transmits the horizontal and the vertical state vector components of nodes $(n_1 + 1, n_2)$ and $(n_1, n_2 + 1)$ respectively. Therefore per each time slot node (n_1, n_2) transmits $a + b$ state vector elements. In the FM model based implementation node (n_1, n_2) transmits its state vector and input vector. Hence n state vector elements and p input vector elements are transmitted by node (n_1, n_2) per time slot. Power required for the implementation of GR and FM models on the sensor network depends upon the order of the realization. Therefore realizing systems using state space models of lower order is beneficial for implementations in sensor networks.

Except for several special classes of transfer functions reported in [Kung et al., 1977; Lin et al., 2007] realizations algorithms capable of deriving the minimal realization of a given m-D transfer function are not available to the best of the author's

knowledge. Furthermore there is no method to determine the order of the minimal realization given a general m-D transfer function. Minimal realization of m-D transfer functions and matrices is an open research issue in itself.

Unlike in FM model based implementations, in GR model based implementations it is not required to transmit the entire state vector. Therefore in order to conserve power and communication bandwidth it is sufficient to realize a given transfer function such that the combined size of the horizontal and the vertical state vector components are as small as possible. Realizations algorithms presented in literature for the GR model such as [Fan et al., 2006; Kanellakis et al., 1989; Manikopoulos and Antoniou, 1990; Theodorou and Tzafestas, 1984; Xu et al., 2008] have put emphasis on minimizing the order of the realization. For implementations on sensor networks, algorithms that derive realizations with minimal horizontal and vertical state vector components even at the expense of a larger temporal vector component are desirable.

7.1.2 Power Efficient Implementations

In the procedure described in chapter 2 for implementing GR and FM models in sensor networks, inter-node communication is restricted to adjacent nodes in the sensor network. Depending on the characteristics of the transmitter and receiver circuitry of the sensor node it may be power efficient to transmit data over multiple hops in a sensor network [Haenggi, 2004; Sikora et al., 2004]. Realization algorithms for both GR and FM state space models such as [Bisiacco et al., 1989; Eising, 1978; Fan et al., 2006; Fornasini and Marchesini, 1978; Kanellakis et al., 1989; Kung et al., 1977; Manikopoulos and Antoniou, 1990; Mitra et al., 1975; Theodorou and Tzafestas, 1984; Xu et al., 2005, 2007, 2008] result in system matrices that have most of the rows containing zeros except for one unity element. Therefore state vectors of adjacent sen-

sor nodes may share the same state vector element albeit in different positions of the state vector. This opens up the possibility of transmission over multiple hops. It has been shown in Sumanasena and Bauer [2008] that, for a FM models based implementation of an edge detection filter in a sensor network, transmission of data over multiple hops results in lower power consumption than transmission over a single hop.

However a general treatment, of the condition under which multi-hop data transmission more power efficient than single-hop data transmission in implementing GR and FM state space models, is not available. Implications of multi-hop transmission on bandwidth requirements and the optimal number of hops over which data should be transmitted should be analyzed.

7.1.3 Robustness

Node and link failures in a sensor network can have adverse effects on the performance of the sensor network. Stability of distributed systems implemented on grid sensor networks under the occurrence of node and link failure was studied in chapter 5. It was shown that, adaptation of a sensor network to link and node failure can render an otherwise stable system unstable. Stability of 2-D distributed systems under communication delays has been studied in Bauer et al. [2001]. However the effect of node and link failure in a sensor network on the performance of the signal processing algorithm implemented on the same has not been analyzed. An approach to achieve optimal performance under node and link failure is given below.

- Propose a performance metric for the system implemented on the sensor network.
- Model the effect of node and link failure on system dynamics. The overall approach for handling link and node failure in the sensor network should be known to model the effect of the same on the system. System dynamics under link and

node failure can be modeled using models 5.3, 5.5, 5.24, 5.26, 5.32 and 5.33.

- Choose system parameters that represent the adaptation of the sensor network to node and link failure such that the performance metric is optimized.

The last step in the aforementioned approach in general requires the knowledge of statistics of node failure, link failure and input signal which may not be available. Due to the vast variety of signal processing applications implemented on sensor networks it may not be possible to propose an optimal criteria to handle node and link failure for all applications. But if criteria to handle node and link failure are restricted, for example to use linear estimates for unknown quantities, the author believes that methods to achieve optimal performance under node and link failure can be derived for classes of signal processing algorithms.

Sensors may not be placed exactly on the grid depending on the deployment strategy used Leoncini et al. [2005]. Deployment errors lead to non-unifrom sampling of the input signal. Deployment errors do not affect the stability properties of the system as long as the connection topology remains a grid. The effect of deployment errors on the system performance depends on the signal processing application as well. For example in the distributed Kalman filter example discussed in chapter 6, the signal to be estimated and the noise statistics were uniform over the sensor network. Therefore deployment errors do not affect the performance of the system provided that the grid topology is maintained. In a sound source localization application, deployment errors can introduce an error to the estimate of the location of the sound source [Pham et al., 2003, 2004]. But if the locations of the sensors are known deployment errors can be accounted for in signal processing. A treatment on the effect of sensor deployment errors on the performance of systems implemented on grid sensor networks and methods to counter the adverse effects of deployment errors is not available.

7.1.4 Extension to Random Sensor Networks

Of the three categories of sensor networks classified according to the deployment strategy in chapter 1, random sensor networks are the most widely studied in literature. The method proposed in this work to implement distributed systems in sensor networks assumes regular placement of sensor nodes. This is due to the formulation of state space models (2.1) and (2.2). Extending the method proposed in this work to random sensor networks will greatly enhance the utility of the proposed method and open up an entirely new approach for distributed signal processing in random sensor networks.

A potential approach to extend the current work to random sensor networks is as follows. Consider a rectangular area over which sensors are randomly deployed. The rectangular area can be hypothetically divided into $N_1 N_2$ rectangles. Sensor nodes in the (i, j)-th rectangle can be collectively considered to be the (i, j)-th node of the grid sensor network. This approach is illustrated in figure 7.1. A virtual grid sensor network of size 10×10 is constructed by randomly deploying 150 sensor nodes in the area to be covered.

In order to assign randomly deployed sensor nodes to grid nodes of the virtual grid sensor network, location of the nodes has to be determined. Sensor localization in sensor networks has been widely studied in literature, see [Ji and Zha, 2003; Khan et al., 2009; Patwari and Hero, 2003; Zhang et al., 2008] and references therein. An apparent issue in the above method is that some nodes in the virtual grid sensor network may not have sensors assigned to them. This is due to some rectangles in the area to be covered not having any nodes deployed in them. It is possible to treat the unavailability of the virtual node as a node failure using models proposed in chapter 5.

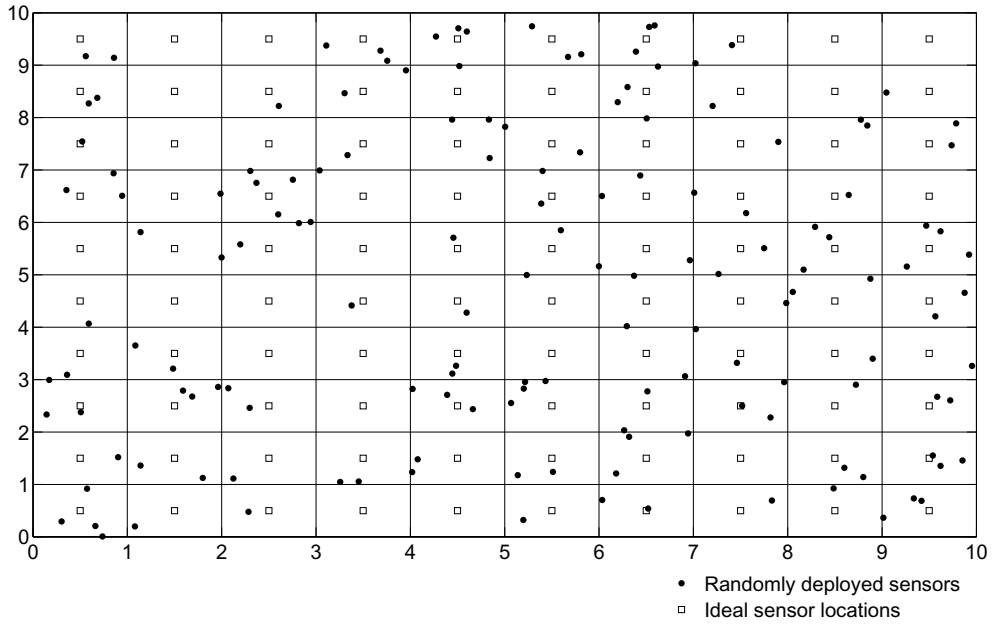


Figure 7.1. Virtual grid sensor network of randomly deployed nodes.

7.1.5 Extension to Broader Classes of Systems

Systems described by models (2.1) and (2.2) are necessarily linear and space-time invariant. They can be used only to implement linear space-time invariant systems. The GR model (2.1) and FM model (2.2) were extended for nonlinear and non space-time invariant systems in Sumanasena and Bauer [2011a]. The GR model for 3-D linear systems can be extended to non-linear systems as follows:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}^h(n_1+1,n_2,t) \\ \boldsymbol{x}^v(n_1,n_2+1,t) \\ \boldsymbol{x}^t(n_1,n_2,t+1) \end{bmatrix} &= f_{(n_1,n_2,t)}(\boldsymbol{x}(n_1,n_2,t),\boldsymbol{u}(n_1,n_2,t)) \\ \boldsymbol{y}(n_1,n_2,t) &= g_{(n_1,n_2,t)}(\boldsymbol{x}(n_1,n_2,t),\boldsymbol{u}(n_1,n_2,t)) \end{aligned} \tag{7.1}$$

where $\boldsymbol{x}(n_1, n_2, t) = (\boldsymbol{x}^{h^T}(n_1, n_2, t), \boldsymbol{x}^{v^T}(n_1, n_2, t), \boldsymbol{x}^{t^T}(n_1, n_2, t))^T$. Vectors $\boldsymbol{x}^h \in \mathbb{R}^a$, $\boldsymbol{x}^v \in \mathbb{R}^b$ and $\boldsymbol{x}^t \in \mathbb{R}^c$ are called the horizontal, vertical and temporal state vector components respectively. Let the input vector $\boldsymbol{u} \in \mathbb{R}^p$ and the output vector $\boldsymbol{y} \in \mathbb{R}^q$. Functions $f_{(n_1,n_2,t)} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $g_{(n_1,n_2,t)} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^q$ are in general non-linear. For a sensor network of size $N_1 \times N_2$, $n_1 \in [0, N_1 - 1]$ and $n_2 \in [0, N_2 - 1]$. The FM model for 3-D linear systems can be extended to non-linear systems as follows:

$$\begin{aligned}
\boldsymbol{x}(n_1, n_2, t) = f_{(n_1,n_2,t)}(&\boldsymbol{x}(n_1, n_2, t\text{–}1), \boldsymbol{x}(n_1, n_2\text{–}1, t), \boldsymbol{x}(n_1\text{–}1, n_2, t), \\
&\boldsymbol{u}(n_1, n_2, t\text{–}1), \boldsymbol{u}(n_1, n_2\text{–}1, t), \boldsymbol{u}(n_1\text{–}1, n_2, t)) \\
\boldsymbol{y}(n_1, n_2, t) = g_{(n_1,n_2,t)}(&\boldsymbol{x}(n_1, n_2, t), \boldsymbol{u}(n_1, n_2, t))
\end{aligned} \tag{7.2}$$

Let the state vector $\boldsymbol{x} \in \mathbb{R}^n$, the input vector $\boldsymbol{u} \in \mathbb{R}^p$ and the output vector $\boldsymbol{y} \in \mathbb{R}^q$. Functions $f_{(n_1,n_2,t)} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^n$ and $g_{(n_1,n_2,t)} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^q$ are in general non-linear.

Systems (7.1) and (7.2) can be implemented on grid sensor networks using methods described in chapter 2 for the implementation of systems (2.1) and (2.2). An important issue that has to be addressed is, given a processing algorithm, how to select functions $f_{(n_1,n_2,t)}$ and $g_{(n_1,n_2,t)}$ such that state space models (7.1) and (7.2) realize the said algorithm. To the best of the author's knowledge, realization of non-linear systems in local state space models has not been investigated in the literature even for restricted cases.

Models (2.1) and (2.2) were traditionally used to realize multidimensional systems in a centralized context. In the current work they were employed to implement distributed systems in grid sensor networks. State space models (2.1), (2.2), (7.1) and (7.2) provide a method to partition a potentially large computation into a sequence of smaller computations. Such partitioning of computations to a sequence of sub compu-

tations may be useful even in grid and cluster computing Sadashiv and Kumar [2011].

7.1.6 Applications

Based on the method proposed in this work to implement linear systems in grid sensor networks, in chapter 6 a DKF algorithm, which outperformed DKF algorithms proposed in literature such as [Cattivelli et al., 2008; Olfati-Saber, 2005; Spanos et al., 2005], was presented. Wide variety of signal processing algorithms implemented on sensor networks involves performing linear operations on sensor measurements [Ganesan et al., 2005; Pham et al., 2004; Rabbat and Nowak, 2004]. Potentially a variety of other signal processing algorithms that can improve the performance of algorithms existing in the literature can be developed based on the method proposed in this work .

7.2 Conclusions

A novel approach for distributed information processing in grid sensor networks was presented. The method based on the GR and the FM local state space models for 3-D systems can be used to implement any linear system on a regular grid sensor network. The method offers several advantages such as scalability and reconfigurability. Moreover, the output computed at each node can be used to decide on the execution of local actuation tasks, in response to local events.

Conditions on system matrices of the GR and FM models, for real-time implementation in a distributed sensor network, were derived. A necessary and sufficient conditions for a transfer function to be realizable in GR and FM models under the said constrains was established. If a transfer function is realizable in the constrained GR model it is also realizable in the constrained FM model. The converse is also true.

The effect of fixed point and floating point arithmetic on system dynamics was modeled. Models incorporate quantization and overflow nonlinearities introduced during computations inside the nodes and communication among nodes. Simple, necessary and sufficient conditions for global asymptotic stability of the system was derived for both fixed point and floating point implementations. Global asymptotic stability of the system is independent of the quantization and overflow nonlinearities applied to state vector components communicated between nodes. Sufficient conditions for the BIBO stability of the system under fixed point quantization nonlinearities were derived.

GR model and FM model were extended to incorporate the effects of node and link failure on system dynamics. Internal and external stability criteria were proposed for systems under node and link failure. Conditions for internal and external stability under node and link failure were established. It was shown that systems are externally stable if they are internally stable. Utility of the proposed method was demonstrated using two example applications.

This work is the first major effort to use state space models and methodologies developed in the multidimensional systems literature for distributed signal processing in sensor networks, to the best of the author's knowledge. It is the author's opinion that methods proposed in this work can be extended to broader classes of sensor networks and used in wide variety of signal processing applications. A significant amount of work has to be done in this regard, some of which has been outlined in this thesis.

BIBLIOGRAPHY

H. AboElFotoh and E. Elmallah. Reliability of wireless sensor grids. In *33rd IEEE Conference on Local Computer Networks*, pages 252–257, Montreal, Canada, October 2008.

H. AboElFotoh, S. Iyengar, and K. Chakrabarty. Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures. *IEEE Transactions on Reliability*, 54(1):145– 155, March 2005.

H. M. F. AboElFotoh, E. S. Elmallah, and H. S. Hassanein. A flow-based reliability measure for wireless sensor networks. *International Journal of Sensor Networks*, 2 (5/6):311–320, July 2007.

P. Agathoklis and L. Bruton. Practical-BIBO stability of n-dimensional discrete systems. *IEE Proceedings on Electronic Circuits and Systems*, 130(6):571–574, December 1983.

A. Akbar, W. Mansoor, S. Chaudhry, A. Kashif, and K. Kim. Node-link-failure resilient routing architecture for sensor grids. In *The 8th International Conference Advanced Communication Technology*, pages 131–135, Phoenix, USA, February 2006.

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

G. E. Antoniou. 2-D lattice discrete filters: Minimal delay and state space realization. *IEEE Signal Processing Letters*, 8(1):1097–1105, January 2001.

S. Attasi. Systems lineaires homogènes à deux indices. *Rapport Laboria*, 31, September 1973.

F. Babich, O. E. Kelly, and G. Lombardi. A variable-order discrete model for the fading channel. In *IEEE 9th Thyrrenian International Workshop on Digital Communications*, Levici, Italy, September 1997.

G. Barrenechea, B. Beferull-Lozano, and M. Vetterli. Lattice sensor networks: capacity limits, optimal routing and robustness to failures. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 186–195, Berkeley, USA, April 2004.

P. Bauer. Absolute response error bounds for floating point digital filters in state space representation. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(9):610–613, Sept. 1995a.

P. Bauer and J. Wang. Limit cycle bounds for floating point implementations of second-order recursive digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 40(8):493–501, August 1993.

P. Bauer, M. Sichitiu, and K. Premaratne. Stability of 2-D distributed processes with time-variant communication delays. In *The 2001 IEEE International Symposium on Circuits and Systems*, pages 497 – 500, Sydney, Australia, May 2001.

P. H. Bauer. A set of necessary stability conditions for m-D nonlinear digital filters. *Circuits, Systems, and Signal Processing*, 14(4):555–561, July 1995b.

E. Biagioni and G. Sasaki. Wireless sensor placement for reliable and efficient data collection. In *Proceedings of the 36 th Annual Hawaii International Conference on System Sciences (HICSS03)*, page 127b, 2003.

M. Bisiacco, E. Fornasini, and G. Marchesini. Dynamic regulation of 2-D systems: A statespace approach. *Linear Algebra and Its Applications*, 122/123/124:195–218, 1989.

P. Bolzern, P. Colaneri, and G. De Nicolao. On almost sure stability of discrete-time Markov jump linear systems. In *43rd IEEE Conference on Decision and Control*, volume 3, pages 3204 –3208, December 2004.

N. K. Bose. *Multidimensional systems theory and applications*. Kluwer Academic Publishers, 2003.

T. Bose. Asymptotic stability of two-dimensional digital filters under quantization. *IEEE Transactions On Signal Processing*, 42(5):1172–1177, May 1994.

T. Bose. Stability of the 2-D state-space system with overflow and quantization. *IEEE Transactions On Circuits And Systems-II*, 42(6):432–434, June 1995.

F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis. In *Proceedings of the Cognitive Information Processing*, Santorini, Greece, June 2008.

A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin. Statistical model of lossy links in wireless sensor networks. In *Fourth International Symposium on Information Processing in Sensor Networks*, pages 81 – 88, Los Angeles, USA, April 2005.

K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51 (12):1448– 1453, December 2002.

H. Cheng, T. Saito, S. Matsushita, and L. Xu. Realization of multidimensional systems in Fornasini-Marchesini state-space model. *Multidimensional Systems and Signal Processing*, pages 1–15, September 2010.

K. K. Chintalapudi and R. Govindan. Localized edge detection in sensor fields. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 59–70, May 2003.

C. Y. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. In *Proceedings of the IEEE*, pages 1247–1256, 2003.

O. L. V. Costa and M. D. Fragoso. Stability results for discrete-time linear systems with Markovian jumping parameters. *Journal of Mathematical Analysis and Applications*, 179(1):154 – 178, October 1993.

D. Devaguptapu and B. Krishnamachari. Applications of localized image processing techniques in wireless sensor networks. In *SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, April 2003.

D. A. Dewasurendra and P. H. Bauer. A novel approach to grid sensor networks. In *15th IEEE International Conference on Electronics, Circuits and Systems*, pages 1191–1194, Malta, August 2008.

R. Eising. Realization and stabilization of 2-D systems. *IEEE Transactions on Automatic Control*, 23(5), October 1978.

R. Eising. State-space realization and inversion of 2-D systems. *IEEE Transactions on Circuits and Systems*, CAS-27(7), July 1980.

D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the International Conference on Mobile Computing and Networking*, pages 263–270, 1999.

D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, pages 2033–2036, Salt Lake City, USA, May 2001.

H. Fan, L. Xu, and Z. Lin. A constructive procedure for three-dimensional realization. In *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pages 1893–1896, Dalian, China, June 2006.

X. Feng, K. Loparo, Y. Ji, and H. Chizeck. Stochastic stability properties of jump linear systems. *IEEE Transactions on Automatic Control*, 37(1):38 –53, January 1992.

E. Fornasini and G. Marchesini. Doubly-indexed dynamical systems: State-space models and structural properties. *Mathematical Systems Theory*, 12(1):59–72, December 1978.

L. Frye, L. Cheng, S. Du, and M. Bigrigg. Topology maintenance of wireless sensor networks in node failure-prone environments. In *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 886 –891, August 2006.

D. Ganesan, B. Greenstein, D. Estrin, J. Heidemann, and R. Govindan. Multiresolution storage and search in sensor networks. *ACM Transactions in Storage*, 1(3):277–315, August 2005. ISSN 1553-3077.

D. Givone and R. Roesser. Multidimensional linear iterative circuits- general properties. *IEEE Transactions on Computers*, C-21(10):1067–1073, October 1972.

M. Haenggi. Twelve reasons not to route over many short hops. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 5, pages 3130 – 3134, September 2004.

N. Hamed Azimi, H. Gupta, X. Hou, and J. Gao. Data preservation under spatial failures in sensor networks. In *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*, pages 171–180, 2010. doi: http://doi.acm.org/10.1145/1860093.1860117. URL `http://doi.acm.org/10.1145/1860093.1860117`.

J. K. Hart and K. Martinez. Environmental sensor networks:a revolution in the earth system science? *Earth-Science Reviews*, 78:177–191, 2006.

T. Hinamoto. Stability of 2-D discrete systems described by the Fornasini-Marchesini second model. *IEEE Transactions on Circuits and Systems I*, 44(3):254–257, March 1997.

M. Imamoglu and M. Keskinoz. Node failure handling for serial distributed detection in wireless sensor networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 1894 –1898, Istanbul, Turkey, September 2010.

X. Ji and H. Zha. Robust sensor localization algorithm in wireless ad-hoc sensor networks. In *ICCCN 2003, Proceedings of the 12th International Conference onComputer Communications and Networks*, Arlington,USA, October 2003.

T. Kaczorek. The singular general model of 2-D systems and its solution. *IEEE Transactions on Automatic Control*, 33(11):1060–1061, November 1988.

H. Kahn, V. S. Hsu, J. M. Kahn, and K. S. J. Pister. Wireless communications for smart dust. In *Electronics Research Laboratory Technical Memorandum M98/2*, 1998.

J. M. Kahn, R. H. Katz, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *Proceedings of the ACM MobiCom*, 1999.

A. J. Kanellakis, P. N. Paraskevopoulos, N. J. Theodorou, and S. J. Varoufakis. On the canonical state-space realization of 3-D discrete systems. *IEE Proceedings on Circuits, Devices and Systems*, 136:19–31, February 1989.

H. Kar. A new sufficient condition for the global asymptotic stability of 2-D state-space digital filters with saturation arithmetic. *Signal Processing*, 88(1):86–98, January 2008.

H. Kar and V. Singh. Stability analysis of 2-D state-space digital filters with overflow nonlinearities. *IEEE Transactions on Circuits and Systems I*, 47(4):598–601, April 2000.

H. Kar and V. Singh. Stability analysis of 1-D and 2-D fixed-point state-space digital filters using any combination of overflow and quantization nonlinearities. *IEEE Transactions On Signal Processing*, 49(5):1097–1105, May 2001a.

H. Kar and V. Singh. Stability analysis of 2-D digital filters described by the Fornasini-Marchesini second model using overflow nonlinearities. *IEEE Transactions on Circuits and Systems I*, 48(5):612–617, May 2001b.

U. Khan, S. Kar, and J. Moura. Distributed sensor localization in random environments using minimal number of anchor nodes. *IEEE Transactions on Signal Processing*, 57(5):2000 –2016, may 2009.

C. Kubrusly and O. Costa. Mean square stability conditions for discrete stochastic bilinear systems. *IEEE Transactions on Automatic Control*, 30(11):1082 – 1087, November 1985.

S. Y. Kung, B. Levy, M. Morf, and T. Kailath. New results in 2-D systems theory, part ii: 2-D state-space models realization and the notions of controllability, observability, and minimality. *Proceedings of the IEEE*, 65(6), June 1977.

Oracle Labs. Sun spot developer's guide. November 2010. URL `http://www.sunspotworld.com/docs/YellowSunSPOT-Programmers-Manual.pdf`.

M. Lazar and L. Bruton. On the practical BIBO stability of multidimensional filters. In *IEEE International Symposium on Circuits and Systems*, pages 571–574, Chicago, USA, May 1993.

L. J. Leclerc and P. H. Bauer. New criteria for asymptotic stability of one and multidimensional state-space digital filters in fixed-point arithmetic. *IEEE Transactions On Signal Processing*, 42(1):46–53, January 1994.

S. G. Lele and J. M. Mendel. Modeling and recursive state estimation for two-dimensional noncausal filters with applications in image restoration. *IEEE Transactions on Circuits and Systems I*, CAS-34(12):1507–1517, December 1987.

M. Leoncini, G. Resta, and P. Santi. Analysis of a wireless sensor dropping problem in wide-area environmental monitoring. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 239– 245, Los Angeles, California, 2005.

Q. Li and D. Rus. Global clock synchronization in sensor networks. *IEEE Transaction on Computers*, 55(2):214–226, February 2006.

Z. Lin, L. Xu, and Y. Anazawa. Revisiting the absolutely minimal realization for two-dimensional digital filters. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 597 –600, New Orleans, USA, May 2007.

A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM Int. Workshop on Wireless Sensor Networks and Applications*, September 2002.

C. N. Manikopoulos and G. E. Antoniou. State-space realization of three-dimensional systems using the modified Cauer form. *International Journal of Systems Science*, 12(21):2673 – 2678, December 1990.

A. Martinez-Sala, J.-M. Molina-Garcia-Pardo, E. Egea-Lopez, J. Vales-Alonso, L. Juan-Llacer, and J. Garcia-Haro. An accurate radio channel model for wireless sensor networks simulation. *Journal Of Communications And Networks*, 7(4):401–407, December 2005.

S. K. Mitra, A. D. Sagar, and N. A. Pendergrass. Realizations of two-dimensional recursive digital filters. *IEEE Transactions On Circuits And Systems*, CAS-22(3): 177–184, March 1975.

S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communications Magazine*, 43 (3):8–13, March 2005.

L. Ntogramatzidis, M. Cantoni, and R. Yang. On the partial realization of non-causal 2-D linear systems. *IEEE Transactions on Circuits and Systems I*, 54(8):1800–1808, August 2007.

R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 05)*, pages 8179 – 8184, Seville, Spain, December 2005.

R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *46th IEEE Conference on Decision and Control*, pages 5492-5498, New Orleans, LA, December 2007.

R. Olfati-saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 05)*, pages 6698–6703, Seville, Spain, December 2005.

N. Patwari and A. O. Hero, III. Using proximity and quantized RSS for sensor localization in wireless networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, WSNA '03, pages 20–29, 2003. ISBN 1-58113-764-8.

E. M. Petriu, N. D. Georganas, D. C. Petriu, D. Makrakis, and V. Z. Groza. Sensor-based information appliances. *IEEE Instrumentation and Measurement Magazine*, 3:31–35, 2000.

T. Pham, B. M. Sadler, and H. Papadopoulos. Energy-based source localization via ad-hoc acoustic sensor network. In *IEEE Workshop on Statistical Signal Processing*, pages 387– 390, October 2003.

T. Pham, D. S. Scherber, and H. Papadopoulos. Distributed source localization algorithms for acoustic ad-hoc sensor networks. In *Sensor Array and Multichannel Signal Processing Workshop Proceedings*, pages 613– 617, July 2004.

K. Premaratne, E. Kulasekere, P. Bauer, and L. Leclerc. An exhaustive search algorithm for checking limit cycle behavior of digital filters. *IEEE Transactions on Signal Processing*, 44(10):2405–2412, October 1996.

M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 20–27, Berkeley, California, USA, 2004. ISBN 1-58113-846-6. doi: http://doi.acm.org/10.1145/984622.984626.

K. Ralev and P. Bauer. Asymptotic behavior of block floating-point digital filters. *Circuits, Systems, and Signal Processing*, 18:75–84, 1999.

K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE*, 11(6):54– 61, December 2004.

N. Sadashiv and S. Kumar. Cluster, grid and cloud computing: A detailed comparison. In *6th International Conference on Computer Science Education (ICCSE)*, pages 477 –482, Singapore, Singapore, aug. 2011.

D. S. Scherber and H. C. Papadopoulos. Locally constructed algorithms for distributed computations in ad-hoc networks. In *Third International Symposium on Information Processing in Sensor Networks*, pages 11–19, Berkeley, California, April 2004.

D. S. Scherber and H. C. Papadopoulos. Distributed computation of averages over ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(4):776–787, April 2005.

S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. In *Proceedings of IEEE INFOCOM*, pages 1073–1083, San Francisco, April 2003.

M. Sikora, J. Laneman, M. Haenggi, J. Costello, D.J., and T. Fuja. On the optimum number of hops in linear wireless networks. In *Information Theory Workshop, 2004. IEEE*, pages 165 – 169, October 2004.

V. Singh. Global asymptotic stability of 2-D state-space digital filters with saturation arithmetic: Modified approach. *Signal Processing*, 88(5):1304–1309, May 2008.

D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 133–139, Los Angeles, California, April 2005. IEEE Press.

I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, November 2001.

M. G. B. Sumanasena and P. Bauer. Models for distributed computing in grid sensor networks. In *International Symposium on Distributed Computing and Artificial Intelligence*, volume 91 of *Advances in Intelligent and Soft Computing*, pages 151–158. Springer Berlin / Heidelberg, 2011a.

M. G. B. Sumanasena and P. Bauer. A distributed Kalman filter for grid sensor networks. *Under review, Journal of The Franklin Institute*, 2010e.

M. G. B. Sumanasena and P. Bauer. Realization using the Fornasini-Marchesini model for implementations in distributed grid sensor networks. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 58(11):2708 –2717, November 2011b.

M. G. B. Sumanasena and P. Bauer. Stability of distributed 3-D systems implemented on grid sensor networks - part i: Link failure. *Under review, IEEE Transactions on Signal Processing*, 2011c.

M. G. B. Sumanasena and P. Bauer. Stability of distributed 3-D systems implemented on grid sensor networks - part ii: Node failure. *Under review, IEEE Transactions on Signal Processing*, 2011d.

M. G. B. Sumanasena and P. Bauer. Stability of distributed 3-D systems implemented on grid sensor networks using floating point arithmetic. In *Accepted for 50th IEEE Conference on Decision and Control*, Orlando, USA, December 2011e.

M. G. B. Sumanasena and P. H. Bauer. Distributed m-D filtering for wave front detection in grid sensor networks. In *Proceedings of the 20th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 423–429, Orlando, Florida, November 2008.

M. G. B. Sumanasena and P. H. Bauer. A Roesser model based multidimensional systems approach for grid sensor networks. In *43rd Asilomar Conference on Signals Systems and Computers*, pages 2151 –2155, Pacific Grove, USA, November 2009.

M. G. B. Sumanasena and P. H. Bauer. Stability of distributed 3-D systems implemented on grid sensor networks. *IEEE Transactions on Signal Processing*, 58(8):4447–4453, August 2010a.

M. G. B. Sumanasena and P. H. Bauer. Realization using the Roesser model for implementations in distributed grid sensor networks. *Multidimensional Systems and Signal Processing*, 22:131–146, 2011d.

M. G. B. Sumanasena and P. H. Bauer. Realization using the FM model for implementations in distributed grid sensor networks. In *49th IEEE Conference on Decision and Control*, pages 389 –395, Atlanta, USA, December 2010b.

M. G. B. Sumanasena and P. H. Bauer. Realization using the Roesser model for implementations in distributed grid sensor networks. In *49th IEEE Conference on Decision and Control*, pages 382 –388, Atlanta, USA, December 2010c.

C. Tan and N. Beaulieu. On first-order Markov modeling for the Rayleigh fading channel. *IEEE Transactions on Communications*, 48(12):2032 –2040, December 2000.

W. P. Tay, J. Tsitsiklis, and M. Win. On the impact of node failures and unreliable communications in dense sensor networks. *IEEE Transactions on Signal Processing*, 56(6):2535 –2546, june 2008. ISSN 1053-587X. doi: 10.1109/TSP.2007.914343.

A. Tejada, O. Gonzalez, and W. Gray. Asymptotic and mean square stability conditions for hybrid jump linear systems with performance supervision. In *Proceedings of the American Control Conference 2005*, pages 569 – 574 vol. 1, Portland ,USA, June 2005.

N. J. Theodorou and S. G. Tzafestas. A canonical state-space model for three-dimensional systems. *International Journal of Systems Science*, 12(15):1353–1379, December 1984.

W. Turin and R. van Nobelen. Hidden Markov modeling of flat fading channels. *IEEE Journal on Selected Areas in Communications*, 16(9):1809 –1817, December 1998.

H. S. Wang and P.-C. Chang. On verifying the first-order Markovian assumption for a Rayleigh fading channel model. *IEEE Transactions on Vehicular Technology*, 45(2): 353 –357, May 1996.

H. S. Wang and N. Moayeri. Finite-state Markov channel-a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163 –171, February 1995.

Q. Wang and W. Yang. Energy consumption model for power management in wireless sensor networks. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 142–151, San Diego, USA, June 2007.

Q. Wang, M. Hempstead, and W. Yang. A realistic power consumption model for wireless sensor network devices. In *3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 286–295, Reston, USA, September 2006.

D. Wu, D. Xie, and L. Wang. A deployment algorithm to achieve both connectivity and coverage in grid sensor networks. In *The 9th International Conference for Young Computer Scientists*, pages 522–526, Hunan, China, November 2008.

K. Xu, G. Takahara, and H. Hassanein. On the robustness of grid-based deployment in wireless sensor networks. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 1183–1188, Vancouver, Canada, July 2006.

L. Xu, O. Saito, and K. Abe. Practical internal stability of n-D discrete systems. *IEEE Transactions On Automatic Control*, 41(5):756–761, May 1996.

L. Xu, L. Wu, Q. Wu, Z. Lin, and Y. Xiao. On realization of 2-D discrete systems by Fornasini-Marchecini model. *International Journal of Control, Automation, and Systems*, 3(4):631–639, December 2005.

L. Xu, Q. Wu, Z. Lin, and Y. Xiao. A new constructive procedure for 2-D coprime realization in Fornasini-Marchesini model. *IEEE Transactions on Circuits and Systems I*, 54(9):2061–2069, September 2007.

L. Xu, H. Fan, Z. Lin, and N. K. Bose. A direct-construction approach to multidimensional realization and LFR uncertainty modeling. *Multidimensional Systems Signal Processing*, 19(3-4):323–359, December 2008.

J. Zhang, J. Luo, and X. Luo. A robust localization algorithm for wireless sensor networks. In *WiCOM 2008, 4th International Conference on Wireless Communications, Networking and Mobile Computing,*, pages 1–4, Dalian, China, October 2008.

M. Zorzi, R. Rao, and L. Milstein. On the accuracy of a first-order Markov model for data transmission on fading channels. In *Fourth IEEE International Conference on Universal Personal Communications*, pages 211 –215, Tokyo, Japan, November 1995.