2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Stability of Distributed 3-D Systems Implemented on Grid Sensor Networks Using Floating Point Arithmetic

Buddika Sumanasena , Peter H. Bauer,

Abstract— **In this paper, stability of distributed 3-D systems implemented in sensor networks using the Givone-Roesser and the Fornasini-Marchesini state space models under floating point arithmetic is studied. Nonlinearities caused by 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 internode communication on system dynamics. A necessary and sufficient condition for global asymptotic stability under floating point arithmetic is established. Simulation results are presented to illustrate the theoretical results.**

I. Introduction

Wireless sensor networks consisting of a large number of resource-constrained embedded sensor nodes has recently become an emerging candidate for many distributed applications. Some of these applications require nodes regularly placed in a spatial grid. The research in [1] and [2] discuss contaminant propagation detection and structural health monitoring using grid sensor networks. Other application areas that often prefer grid or mesh topology include agriculture and environmental monitoring. Grid sensor networks are discussed in detail in [3], [4], [5], [6], [7], [8] highlighting sensor deployment strategies, robustness against deployment errors, reliability, routing schemes and network capacity limits. Coverage and connectivity of grid sensor networks in the presence of node failure has been studied in [9].

Distributed information processing methods are natural candidates for networks with regularly placed nodes, yielding significant benefits in terms of scalability, lower communication costs, and improved energy savings. Furthermore, applications requiring *local actuation in response to a local detection* [10] are best supported by such distributed algorithms, yielding minimum response delays as compared to centralized schemes.

Using the Givone-Roesser [11] and the Fornasini-Marchesini [12] local state space models, a method for distributed information processing in grid sensor networks has been presented in [13], [14]. The method can be used to implement linear systems in grid sensor networks. The system to be implemented is realized in either Givone-Roesser(Roesser) or Fornasini-Marchesini(FM-II) state space models and the state space model is implemented on the sensor network. Implementation of either state space model requires communication only between the adjacent sensor nodes. Therefore the scheme is scalable and lends itself readily for distributed signal processing.

A. Motivation and Goals

Using shorter word lengths for the representation of numbers during communication can result in significant energy and bandwidth savings. Furthermore limited processing power of sensor nodes may demand shorter word length number representation schemes. Therefore shorter word lengths are preferred for the representation of numbers in sensor networks.

Global asymptotic stability of 3-D systems implemented on grid sensor networks using the method presented in [13], [14] is studied in [15] for the case where numbers are represented using fixed point schemes. 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 [16]. As computational capabilities of embedded processors improve, floating point processors can be expected to be used widely in sensor nodes in the future. Global asymptotic stability of systems implemented in grid sensor networks, using the method presented in [13], [14], in the presence of floating point arithmetic is analyzed in this paper. Quantization nonlinearities resulting from quantizing results of arithmetic operations are modeled. A necessary and sufficient condition for system stability under floating point computations is derived. Special consideration is given to the role internode communication[1] plays in the global asymptotic stability of the sensor network.

B. Outline

The remainder of the paper is organized as follows: Floating point quantization is briefly discussed in section II. The Roesser and the FM-II state space models for 3-D systems are also presented in section II. In section III, system models for quantized systems are discussed. Stability of the system under quantization is analyzed in section IV. An example is provided in section V to illustrate the results. Concluding remarks are given in section VI.

II. Background

In this section floating point representation of numbers and quantization in floating point arithmetic operations will be briefly discussed. Roesser and FM-II state space models for 3-D systems will also be presented.

[1] Communication of state and input vectors between nodes.

978-1-61284-799-3/11/$26.00 ©2011 IEEE

A. Local State Space Models for 3-D Systems

Method proposed in [13], [14] for distributed signal processing in grid sensor networks is based on the Roesser and FM-II state space models for 3-D systems. In the method proposed in [13], [14], the system to be implemented is realized in either Roesser or FM-II local state space models. Then the state space model is implemented on the sensor network. Details of implementing Roesser or FM-II state space models on sensor networks are not discussed in this paper and the interested reader is referred to [13], [14]. Throughout the paper, the sensor network is assumed to be a 2-D regular rectangular grid sensor network of finite size. The Roesser and the FM-II state space models were originally proposed for 2-dimensional systems , but can be extended to higher dimensional systems in a straightforward manner.

1) The Roesser Model For 3-D Systems: The Roesser model for 3-D systems is given by,

$$\begin{bmatrix} \mathbf{x^h}(n_1+1,n_2,t) \\ \mathbf{x^v}(n_1,n_2+1,t) \\ \mathbf{x^t}(n_1,n_2,t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_4} & \mathbf{A_5} & \mathbf{A_6} \\ \mathbf{A_7} & \mathbf{A_8} & \mathbf{A_9} \end{bmatrix} \begin{bmatrix} \mathbf{x^h}(n_1,n_2,t) \\ \mathbf{x^v}(n_1,n_2,t) \\ \mathbf{x^t}(n_1,n_2,t) \end{bmatrix} + \begin{bmatrix} \mathbf{B_1} \\ \mathbf{B_2} \\ \mathbf{B_3} \end{bmatrix} \mathbf{u}(n_1,n_2,t)$$

$$\mathbf{y}(n_1,n_2,t) = \mathbf{Cx}(n_1,n_2,t) + \mathbf{Du}(n_1,n_2,t) \qquad (1)$$

where $\mathbf{x}(n_1,n_2,t) = (\mathbf{x^h}^T(n_1,n_2,t), \mathbf{x^v}^T(n_1,n_2,t), \mathbf{x^t}^T(n_1,n_2,t))^T$. For a sensor network of size $N_1 \times N_2$, $n_1 \in [1,N_1]$, $n_2 \in [1,N_2]$ and $t \in [0,\infty)$. Vectors $\mathbf{x^h} \in \mathbb{R}^a$, $\mathbf{x^v} \in \mathbb{R}^b$ and $\mathbf{x^t} \in \mathbb{R}^c$ are called the horizontal, vertical and temporal state vector components respectively. Let the input vector $\mathbf{u} \in \mathbb{R}^p$ and output vector $\mathbf{y} \in \mathbb{R}^q$. Then $\mathbf{A_1} \in \mathbb{R}^{a\times a}$, $\mathbf{A_2} \in \mathbb{R}^{a\times b}$, $\mathbf{A_3} \in \mathbb{R}^{a\times c}$, $\mathbf{A_4} \in \mathbb{R}^{b\times a}$, $\mathbf{A_5} \in \mathbb{R}^{b\times b}$, $\mathbf{A_6} \in \mathbb{R}^{b\times c}$, $\mathbf{A_7} \in \mathbb{R}^{c\times a}$, $\mathbf{A_8} \in \mathbb{R}^{c\times b}$, $\mathbf{A_9} \in \mathbb{R}^{c\times c}$, $\mathbf{B_1} \in \mathbb{R}^{a\times p}$, $\mathbf{B_2} \in \mathbb{R}^{b\times p}$, $\mathbf{B_3} \in \mathbb{R}^{c\times p}$, $\mathbf{C} \in \mathbb{R}^{q\times(a+b+c)}$ and $\mathbf{D} \in \mathbb{R}^{q\times p}$. Let

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_4} & \mathbf{A_5} & \mathbf{A_6} \\ \mathbf{A_7} & \mathbf{A_8} & \mathbf{A_9} \end{bmatrix}$$

and $\mathbf{B} = (\mathbf{B_1}^T, \mathbf{B_2}^T, \mathbf{B_3}^T)^T$.

2) The FM-II Model For 3-D Systems: The FM-II model for 3-D systems is given by:

$$\begin{aligned} \mathbf{x}(n_1,n_2,t) &= \mathbf{A_1x}(n_1-1,n_2,t) + \mathbf{A_2x}(n_1,n_2-1,t) \\ &\quad + \mathbf{A_tx}(n_1,n_2,t-1) + \mathbf{B_1u}(n_1-1,n_2,t) \\ &\quad + \mathbf{B_2u}(n_1,n_2-1,t) + \mathbf{B_tu}(n_1,n_2,t-1) \\ \mathbf{y}(n_1,n_2,t) &= \mathbf{Cx}(n_1,n_2,t) + \mathbf{Du}(n_1,n_2,t) \qquad (2) \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $n_1 \in [1,N_1]$, $n_2 \in [1,N_2]$ and $t \in [0,\infty)$. Let the input vector $\mathbf{u} \in \mathbb{R}^p$ and the output vector $\mathbf{y} \in \mathbb{R}^q$. Then, $\mathbf{C} \in \mathbb{R}^{q\times n}$, $\mathbf{D} \in \mathbb{R}^{q\times p}$, $\mathbf{A_1} \in \mathbb{R}^{n\times n}$, $\mathbf{A_2} \in \mathbb{R}^{n\times n}$, $\mathbf{A_t} \in \mathbb{R}^{n\times n}$, $\mathbf{B_1} \in \mathbb{R}^{n\times p}$, $\mathbf{B_2} \in \mathbb{R}^{n\times p}$ and $\mathbf{B_t} \in \mathbb{R}^{n\times p}$.

B. 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.

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. Overflow and underflow occurs if the product is larger than the largest or smaller than the smallest representable numbers respectively. 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.

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.

III. Quantization Models

Floating point computations introduce nonlinearities to otherwise linear system models (1) and (2). System models

$$N[\mathbf{Ax}] = \begin{pmatrix} N^a[N^m[a_{11}x_1] + N^a[N^m[a_{12}x_2] + N^a[N^m[a_{13}x_3] + \cdots N^a[N^m[a_{1n-1}x_{n-1}] + N^m[a_{1n}x_n]]]]] \\ N^a[N^m[a_{21}x_1] + N^a[N^m[a_{22}x_2] + N^a[N^m[a_{23}x_3] + \cdots N^a[N^m[a_{2n-1}x_{n-1}] + N^m[a_{2n}x_n]]]]] \\ \cdots\cdots \\ \cdots\cdots \\ N^a[N^m[a_{n1}x_1] + N^a[N^m[a_{n2}x_2] + N^a[N^m[a_{n3}x_3] + \cdots N^a[N^m[a_{nn-1}x_{n-1}] + N^m[a_{nn}x_n]]]]] \end{pmatrix} \qquad (3)$$

(1) and (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 $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a_{ij} and the $i-th$ element of vector $\mathbf{x} \in \mathbb{R}^n$ be x_i. The product of matrix $\mathbf{A}$ with vector $\mathbf{x}$ computed using floating point arithmetic is denoted by $N[\mathbf{Ax}]$, where $N[\mathbf{Ax}]$ is of the form given by (3). In (3), N^a and N^m denote nonlinearities caused by floating point addition and multiplication respectively.

A. Model 1

Let $N_P^a : \mathbb{R} \to \mathbb{S}_p^a$ and $N_P^m : \mathbb{R} \to \mathbb{S}_p^m$ 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 Roesser model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{bmatrix} \mathbf{x^h}(n_1+1,n_2,t) \\ \mathbf{x^v}(n_1,n_2+1,t) \\ \mathbf{x^t}(n_1,n_2,t+1) \end{bmatrix} = N_P^a\left[N_P[\mathbf{Ax}(n_1,n_2,t)]\right.$$
$$\left. + N_P\left[\mathbf{Bu}(n_1,n_2,t)\right]\right] \tag{4}$$

It is assumed that $\mathbf{Ax}(n_1,n_2,t)$ and $\mathbf{Bu}(n_1,n_2,t)$ are computed first and then their summation is computed. Nonlinearity introduced in the in node computation[2] of a product of a matrix and a vector is denoted by N_P. Nonlinear operator N_P is of the form given by (3). In a FM-II model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \mathbf{x}(n_1,n_2,t) = N_P^a & \left[N_P[\mathbf{A_t x}(n_1,n_2,t\text{–}1)]\right. \\ & + N_P^a[N_P[\mathbf{A_2 x}(n_1,n_2\text{–}1,t)] \\ & + N_P^a[N_P[\mathbf{A_1 x}(n_1\text{–}1,n_2,t)] \\ & + N_P^a[N_P[\mathbf{B_t u}(n_1,n_2,t\text{–}1)] \\ & + N_P^a[N_P[\mathbf{B_2 u}(n_1,n_2\text{–}1,t)] \\ & + N_P[\mathbf{B_1 u}(n_1\text{–}1,n_2,t)]]]]]] \end{aligned} \tag{5}$$

B. 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 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 Roesser model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \begin{bmatrix} \mathbf{x^h}(n_1+1,n_2,t) \\ \mathbf{x^v}(n_1,n_2+1,t) \end{bmatrix} = & N_C\left[N_P\left[\begin{bmatrix} \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_4} & \mathbf{A_5} & \mathbf{A_6} \end{bmatrix}\mathbf{x}(n_1,n_2,t)\right]\right. \\ & \left. + N_P\left[\begin{bmatrix} \mathbf{B_1} \\ \mathbf{B_2} \end{bmatrix}\mathbf{u}(n_1,n_2,t)\right]\right] \\ \begin{bmatrix} \mathbf{x^t}(n_1,n_2,t+1) \end{bmatrix} = & N_P^a\left[N_P\left[\begin{bmatrix} \mathbf{A_7} & \mathbf{A_8} & \mathbf{A_9} \end{bmatrix}\mathbf{x}(n_1,n_2,t)\right]\right. \\ & \left. + N_P\left[\begin{bmatrix} \mathbf{B_3} \end{bmatrix}\mathbf{u}(n_1,n_2,t)\right]\right] \end{aligned} \tag{6}$$

In (6), 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-II model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \mathbf{x}(n_1,n_2,t) = N_P^a & \left[N_P[\mathbf{A_t x}(n_1,n_2,t\text{–}1)]\right. \\ & + N_P^a[N_P[\mathbf{A_2}[N_C\left[\mathbf{x}(n_1,n_2\text{–}1,t)\right]]]] \\ & + N_P^a[N_P[\mathbf{A_1}[N_C\left[\mathbf{x}(n_1\text{–}1,n_2,t)\right]]]] \\ & + N_P^a[N_P[\mathbf{B_t u}(n_1,n_2,t\text{–}1)] \\ & + N_P^a[N_P[\mathbf{B_2}[N_C\left[\mathbf{u}(n_1,n_2\text{–}1,t)\right]]]] \\ & + N_P[\mathbf{B_1}[N_C\left[\mathbf{u}(n_1\text{–}1,n_2,t)\right]]]]]]]] \end{aligned} \tag{7}$$

C. 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 Roesser model based implementation, computation of state vectors within each node can be modeled by:

$$\begin{aligned} \begin{bmatrix} \mathbf{x^h}(n_1+1,n_2,t) \end{bmatrix} = & N_{C_h}\left[N_P\left[\begin{bmatrix} \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \end{bmatrix}\mathbf{x}(n_1,n_2,t)\right]\right. \\ & \left. + N_P\left[\begin{bmatrix} \mathbf{B_1} \end{bmatrix}\mathbf{u}(n_1,n_2,t)\right]\right] \\ \begin{bmatrix} \mathbf{x^v}(n_1,n_2+1,t) \end{bmatrix} = & N_{C_v}\left[N_P\left[\begin{bmatrix} \mathbf{A_4} & \mathbf{A_5} & \mathbf{A_6} \end{bmatrix}\mathbf{x}(n_1,n_2,t)\right]\right. \\ & \left. + N_P\left[\begin{bmatrix} \mathbf{B_2} \end{bmatrix}\mathbf{u}(n_1,n_2,t)\right]\right] \\ \begin{bmatrix} \mathbf{x^t}(n_1,n_2,t+1) \end{bmatrix} = & N_P\left[N_P\left[\begin{bmatrix} \mathbf{A_7} & \mathbf{A_8} & \mathbf{A_9} \end{bmatrix}\mathbf{x}(n_1,n_2,t)\right]\right. \\ & \left. + N_P\left[\begin{bmatrix} \mathbf{B_3} \end{bmatrix}\mathbf{u}(n_1,n_2,t)\right]\right] \end{aligned} \tag{8}$$

Here $N_{C_h} : \mathbb{R} \to \mathbb{S}_{C_h}$ and $N_{C_v} : \mathbb{R} \to \mathbb{S}_{C_v}$ 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 (8), 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-II 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.

IV. Stability of the System

The finite size of the sensor network makes the practical BIBO stability [17] the most suitable input-output stability criteria on the system under consideration. 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.

The sensor network is assumed to be of size $N_1 \times N_2$. Therefore $n_1 \in [1, N_1]$ and $n_2 \in [1, N_2]$. Since we are interested in global asymptotic stability, the input to all the nodes is assumed to be zero for $t \geq 0$.

[2] Computations done by nodes

Definition The system is said to be GAS[3] if:

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

where $\parallel \cdot \parallel$ is any vector norm. It is assumed that $\mathbf{x}(-1, n_2, t) = 0$, $\mathbf{x}(n_1, -1, t) = 0$ and the only non-zero boundary conditions are given by $\mathbf{x}(n_1, n_2, -1)$.

1) Roesser Model Based Implementation: The system (8) is considered since it is the most general case and results derived for this case carry over to the other two cases.

Theorem 1 The system (8) is GAS, if and only if the 1-D system,

$$\mathbf{x}(t+1) = N_P\left[\mathbf{A_9}\mathbf{x}(t)\right] \tag{9}$$

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

Proof: The proof is similar to the proof of Theorem (1) in [15]. ■

2) FM-II Model Based Implementation: The system (7) is considered since it is the most general case and results derived for this case carry over to the other cases.

Theorem 2 The system (7) is GAS, if and only if the 1-D system,

$$\mathbf{x}(t+1) = N_P\left[\mathbf{A_t}\mathbf{x}(t)\right] \tag{10}$$

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

Proof: The proof is similar to the proof of Theorem (1) in [15]. ■

An important implication of Theorems (1) and (2) is that the 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 [18]. Asymptotic stability of 1-D systems described by state space models under floating point computations is analyzed in [19], [20]. Global asymptotic stability of distributed 3-D systems implemented on grid sensor networks under fixed point arithmetic is independent of the quantization and overflow operations applied to communicated state vectors [15]. Furthermore global asymptotic stability of the 3-D system under fixed point arithmetic is equivalent to the global asymptotic stability of a 1-D system under fixed point computations [15]. Therefore conditions derived in this work for global asymptotic stability of 3-D systems under floating point arithmetic is analogous to the conditions derived in [15] for 3-D systems under fixed point arithmetic. Note that in contrast to fixed point systems, in floating point systems a linearly stable systems can cause an unbounded response to a bounded input or a zero input with non-zero initial conditions.

V. Examples

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 (13). 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.

A. Givone-Roesser Model Based Implementation

The transfer function (13) can be realized using the Givone-Roesser model:

$$\begin{aligned}
\begin{bmatrix} \mathbf{x^h}(n_1+1, n_2, t) \\ \mathbf{x^v}(n_1, n_2+1, t) \\ \mathbf{x^t}(n_1, n_2, t+1) \end{bmatrix} &= \begin{bmatrix} \frac{1}{16} & 0 & \frac{1}{8} & 0 \\ 0 & \frac{1}{16} & 0 & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{5}{4} & \frac{5}{32} \\ 0 & 1 & \frac{5}{4} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x^h}(n_1, n_2, t) \\ \mathbf{x^v}(n_1, n_2, t) \\ \mathbf{x^t}(n_1, n_2, t) \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \mathbf{u}(n_1, n_2, t) \\
\mathbf{y}(n_1, n_2, t) &= [1\ 0\ 0\ 0]\mathbf{x}(n_1, n_2, t)
\end{aligned} \tag{11}$$

where $\mathbf{x^h} \in \mathbb{R}$, $\mathbf{x^v} \in \mathbb{R}$ and $\mathbf{x^t} \in \mathbb{R}^2$. According to Theorem 1, the system (11) under floating point computation is GAS if and only if the system,

$$\mathbf{x}(t+1) = N_P\left[\begin{bmatrix} \frac{5}{4} & \frac{5}{32} \\ \frac{5}{4} & 0 \end{bmatrix} \mathbf{x}(t)\right] \tag{12}$$

is GAS. Here $\mathbf{x} \in \mathbb{R}^2$ and N_P is the quantization operator used for in node computations. Global asymptotic stability of systems of the form (12) has been studied in [19]. 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 [19]. For the system (12) a mantissa length

$$H(z_1, z_2, z_t) = \frac{\frac{1}{8}z_1 - \frac{1}{128}z_1z_2 - \frac{5}{256}z_1z_t + \frac{129}{4096}z_1z_2z_t}{1 - \frac{1}{16}z_1 - \frac{1}{16}z_2 + \frac{1}{256}z_1z_2 - \frac{5}{4}z_t + \frac{5}{64}z_1z_t + \frac{5}{64}z_2z_t - \frac{5}{1024}z_1z_2z_t + \frac{25}{128}z_t^2 - \frac{25}{2048}z_1z_t^2 - \frac{25}{2048}z_2z_t^2 + \frac{25}{32768}z_1z_2z_t^2} \tag{13}$$

[3] Globally asymptotically stable

of 11 bits is sufficient to ensure a granular periodic response in the underflow regime.

The system (11) 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 $\mathbf{x}(n_1, n_2, 0) = [0, 0, 5, 14]^T$ for $1 \leq n_1 \leq 4$ and $1 \leq n_2 \leq 4$. Figure 1 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 2 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.

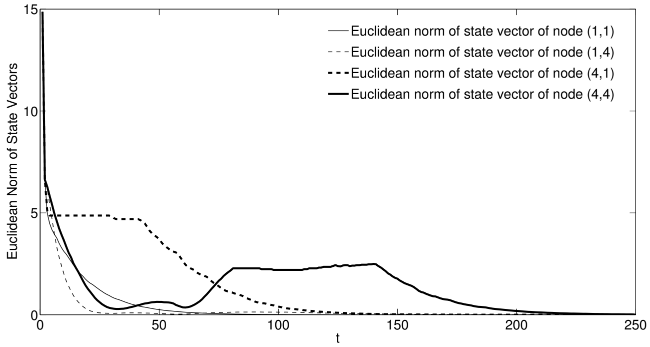

Fig. 1. Euclidean norm of the state vectors versus t for the Givone-Roesser model, when computations are done with 6 bit mantissa

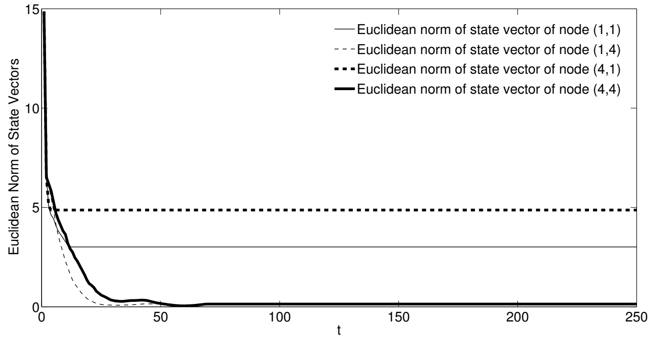

Fig. 2. Euclidean norm of the state vectors versus t for the Givone-Roesser model, when computations are done with 5 bit mantissa

System (12) is not GAS when computations are done with 5 bit mantissa. Therefore, when computations are done with 5 bit mantissa, system (11) is also not GAS according to Theorem 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 3. State vectors communicated between nodes were represented using a floating point scheme with 5 bit mantissa. A 6 bit mantissa was used for floating point computations within the node. Figure 3 shows plots of the Euclidean norm of state vectors of nodes $(1, 1)$,$(1, 4)$,$(4, 1)$ and $(4, 4)$ versus time.

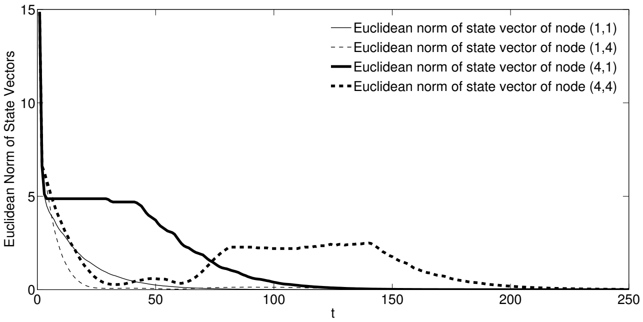

Fig. 3. Euclidean norm of the state vectors versus t for the Givone-Roesser model, when in node computations are performed with 6 bit mantissa and communicated state vectors are quantized to 5 bit mantissa.

In this case states of the nodes reach the origin as time tends to infinity, though a 5 bit mantissa is used for communicated state vectors. As predicted by Theorem 1 the quantization scheme used for communicated state vectors does not affect the asymptotic stability of the system.

B. FM-II Model Based Implementation

The input-output transfer function (13) can be realized using the FM-II model:

$$\begin{aligned}
\mathbf{x}(n_1, n_2, t) =& \begin{bmatrix} \frac{1}{8} & -\frac{1}{16} & \frac{1}{8} & 0 \\ \frac{1}{8} & -\frac{1}{16} & \frac{1}{8} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(n_1 - 1, n_2, t) \\
&+ \begin{bmatrix} -\frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{8} \\ -\frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(n_1, n_2 - 1, t) \\
&+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{5}{4} & \frac{5}{32} \\ -2 & 2 & \frac{5}{4} & 0 \end{bmatrix} \mathbf{x}(n_1, n_2, t - 1) \\
&+ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \mathbf{u}(n_1 - 1, n_2, t) \\
\mathbf{y}(n_1, n_2, t) =& [2 \ -1 \ 0 \ 0] \mathbf{x}(n_1, n_2, t)
\end{aligned} \tag{14}$$

Here $\mathbf{x} \in \mathbb{R}^4$. According to Theorem 2, the system (14) under floating point computation is GAS if and only if the system,

$$\mathbf{x}(t+1) = N_P \left[\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{5}{4} & \frac{5}{32} \\ -2 & 2 & \frac{5}{4} & 0 \end{bmatrix} \mathbf{x}(t) \right] \tag{15}$$

is GAS. Here $\mathbf{x} \in \mathbb{R}^4$ and N_P is the quantization operator used for in node computations. It is evident that global asymptotic stability of system (15) is equivalent to that of system (12). The system (14) 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 $\mathbf{x}(n_1, n_2, 0) = [0, 0, 5, 14]^T$ for $1 \leq n_1 \leq 4$ and $1 \leq n_2 \leq 4$. Figure 4 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 5 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.

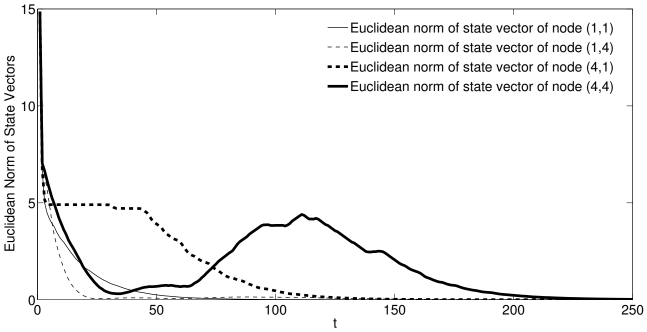

Fig. 4. Euclidean norm of the state vectors versus t for the FM-II model, when computations are done with 6 bit mantissa

System (15) is not GAS when computations are done with 5 bit mantissa. Therefore, when computations are done with 5 bit mantissa, system (14) is also not GAS according to Theorem 2. Simulation results show that system (14) is not GAS. Simulation results, for the case when the quantization scheme described by model 2 was used, is given in figure 6. 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 6 shows plots of the Euclidean norm of state vectors of nodes $(1,1)$,$(1,4)$,$(4,1)$ and $(4,4)$ versus time.

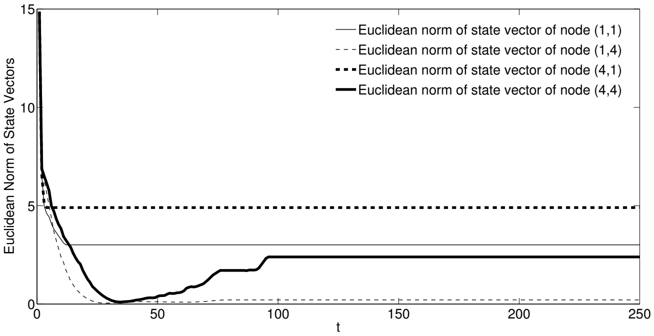

Fig. 5. Euclidean norm of the state vectors versus t for the FM-II model, when computations are done with 5 bit mantissa

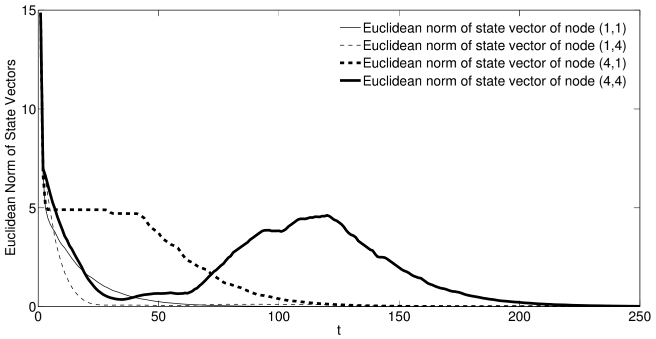

Fig. 6. Euclidean norm of the state vectors versus t for the FM-II model, when in node computations are performed with 6 bit mantissa and communicated state vectors are quantized to 5 bit mantissa.

In this case states of the nodes reach the origin as time tends to infinity, though a 5 bit mantissa is used for communicated state vectors. As predicted by Theorem 2 quantization scheme used for communicated state vectors does not affect the asymptotic stability of the system (14).

VI. Conclusions

Distributed implementation of 3-D systems described by the Roesser and the FM-II state space models in grid sensor networks under floating point computations was studied in this work. Nonlinearities introduced by floating point computations to otherwise linear systems were modeled. A necessary and sufficient condition for the global asymptotic stability of the system was derived. It was shown that global asymptotic stability of distributed 3-D systems implemented on grid sensor networks are independent of the quantization nonlinearities applied to communicated state and input vectors. An example was provided to illustrate the theoretical results.

References

[1] 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*, (Orlando, Florida), pp. 423–429, November 2008.

[2] Y. Huang, K. Loewke, K. Schaaf, and S. Nemat-Nasser, "Localized shm with embedded sensor network," in *Proceedings of the 5th International Workshop on Structural Health Monitoring*, (Stanford), pp. 1554–1561, September 2005.

[3] 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*, (Los Angeles, California), pp. 239– 245, 2005.

[4] 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*, (Vancouver, Canada), pp. 1183–1188, July 2006.

[5] 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*, vol. 54, pp. 145– 155, March 2005.

[6] 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*, vol. 2, pp. 311–320, July 2007.

[7] 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*, (Phoenix, USA), pp. 131–135, February 2006.

[8] 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*, (Berkeley, USA), pp. 186–195, April 2004.

[9] S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable sensor grids: Coverage, connectivity and diameter,” in *Proceedings of IEEE INFOCOM*, (San Francisco), pp. 1073–1083, April 2003.

[10] W. Hu, N. Bulusu, and S. Jhan, “A communication paradigm for hybrid sensor/actuator networks,” *International Journal of Wireless Information Networks*, vol. 12, pp. 47–59, October 2005.

[11] D. Givone and R. Roesser, “Multidimensional linear iterative circuits-general properties,” *IEEE Transactions on Computers*, vol. C-21, pp. 1067–1073, October 1972.

[12] E. Fornasini and G. Marchesini, “Doubly-indexed dynamical systems: State-space models and structural properties,” *Mathematical Systems Theory*, vol. 12, pp. 59–72, December 1978.

[13] D. A. Dewasurendra and P. H. Bauer, “A novel approach to grid sensor networks,” in *15th IEEE International Conference on Electronics, Circuits and Systems*, (Malta), pp. 1191–1194, August 2008.

[14] 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*, (Pacific Grove), November 2009.

[15] M. G. B. Sumanasena and P. H. Bauer, “Stability of distributed 3-d systems implemented on grid sensor networks,” *IEEE Transactions on Signal Processing*, vol. 58, pp. 4447–4453, August 2010.

[16] Oracle Labs, “Sun spot developer’s guide,’*http://www.sunspotworld.com/docs/Yellow/SunSPOT-Programmers-Manual.pdf* November 2010[Feb. 2011].

[17] P. Agathoklis and L. Bruton, “Practical-bibo stability of n-dimensional discrete systems,” *IEE Proceedings on Electronic Circuits and Systems*, vol. 130, pp. 571–574, December 1983.

[18] 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*, vol. 40, pp. 493–501, August 1993.

[19] 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*, vol. 42, pp. 610–613, Sept. 1995.

[20] K. Ralev and P. Bauer, “Asymptotic behavior of block floating-point digital filters,” *Circuits, Systems, and Signal Processing*, vol. 18, pp. 75–84, 1999.