Computer Systems Science & Engineering
DOI:10.32604/csse.2021.016357
Article

Model Implementation and Analysis of a True Three-dimensional Display System

Ye Tian[1,2], Yang Yang[1,*], Han Yang[3] and Ze Ji[4]

[1]Changchun University of Science and Technology, Changchun, 130022, China
[2]Tsinghua University, Beijing, 100084, China
[3]Academy of Military Sciences, Beijing, 100166, China
[4]Cardiff University, Cardiff, CF24 3AA, UK
[*]Corresponding Author: Yang Yang. Email: cloneyang@126.com
Received: 31 December 2020; Accepted: 11 March 2021

Abstract: To model a true three-dimensional (3D) display system, we introduced the method of voxel molding to obtain the stereoscopic imaging space of the system. For the distribution of each voxel, we proposed a four-dimensional (4D) Givone–Roessor (GR) model for state-space representation—that is, we established a local state-space model with the 3D position and one-dimensional time coordinates to describe the system. First, we extended the original elementary operation approach to a 4D condition and proposed the implementation steps of the realization matrix of the 4D GR model. Then, we described the working process of a true 3D display system, analyzed its real-time performance, introduced the fixed-point quantization model to simplify the system matrix, and derived the conditions for the global asymptotic stability of the system after quantization. Finally, we provided an example to prove the true 3D display system's feasibility by simulation. The GR-model-representation method and its implementation steps proposed in this paper simplified the system's mathematical expression and facilitated the microcontroller software implementation. Real-time and stability analyses can be used widely to analyze and design true 3D display systems.

Keywords: True 3D display system; method of voxel molding (MVM); Givone–Roessor (GR) model; asymptotic stability; Bounded-Input Bounded-Output (BIBO) stability; real-time display

1 Introduction

In recent years, with advances in optical and computer technology, stereoscopic display technology has undergone accelerated development. As a result, people are pursuing a more realistic display effect from the traditional two-dimensional (2D) display to three-dimensional (3D) display [1]. Among these developments, the most representative display is the true 3D display system.

For a true 3D display system, each voxel's brightness and color should be controllable, and the relative spatial positional relationship between voxels should also be truly embodied. These goals require finding a suitable model to build a state space for the true 3D display system. Furthermore, the implementation steps and the results of performance analyses must be detailed.

In multidimensional system theory, the Givone–Roessor (GR) [2] and Fornasini-Marchesini (FM) models [3] are two widely used local state-space models. Both have been employed in the state-space modeling of multidimensional systems, including wireless sensor networks [4–8]. Compared with the FM model, the GR model can quantitatively express each dimension's variables and then be used to analyze each dimension's influence on the entire system. Therefore, we established a 4D GR model by introducing 3D position coordinates and one-dimensional (1D) time coordinates to model the system's state space.

Galkowski [9] used the forward transfer operator to represent the transfer function and obtain the system implementation matrix through the matrix transformation for implementing the state-space model. The concept is simple and easily calculable. It can also be used to evaluate the influence of coefficient values on the realization matrix in implementing a multidimensional system. This flexibility, however, also can result in an infinite number of possible intermediate operations in constructing the feature matrix, making it difficult to obtain a general algorithm. Moreover, this method is not easily implemented by a computer program. Xu et al. [10] used the unit retardation factor to represent the transfer function and introduced the elementary operation approach (EOA) to give the implementation method of the system implementation matrix. In this method, however, only one order can be reduced in each supplementary operation. Although it is possible to increase the supplementary operation's efficiency by some decomposition, only a few transfer functions can be applied to this decomposition method. The matrix method is also used to obtain a GR model with a lower order of the matrix [11], but it cannot be used to analyze the coefficient value's influence on the implementation matrix. Xiong [12] proposed an improved 3D EOA algorithm. Through matrix operation, the state-space model of the GR model can be obtained from the transfer function, and the order of the model is significantly reduced. In this paper, the EOA algorithm is extended to four dimensions and applied to the modeling and implementation steps of a true 3D display system.

When a true 3D display system works, one needs each voxel to transfer the information to the microcontroller as soon as possible for data analysis; thus, it has high real-time and stability requirements. Therefore, one must analyze the real-time functionality and stability of the GR model. Kokil [13] and Xin et al. [14] introduced 2D discrete system stability. However, if it is directly extended to multidimensional systems, many limitations still exist. Agathoklis et al. [15] introduced the bounded-input, bounded-output stability of the traditional multidimensional system, but the practical application still has many limitations. In a true 3D display system with a finite size, the voxel calculation results often must be quantified to achieve the perfect display effect. Therefore, in the present study, we introduced a quantitative model to obtain the necessary and sufficient conditions for the true 3D display system's stability after quantization, and then provided an example to verify the model.

2 Establishment of GR Model with Method of Voxel Molding

2.1 Method of Voxel Molding

When a true 3D display system is working, a full-body cylinder-voxel space composed of several spatially discrete voxels will exist. It can be imagined that this space is a flask mold, and each voxel in the interior is sand. The sand in a sandbox has a unique 3D space coordinate and a 1D time coordinate. The 3D mathematical model of the displayed object is considered to be a mold. We assume that upon putting the mold into the sandbox, the mold will replace the sand's space in the original sandbox and form a mold cavity. Based on this assumption, we proposed a 3D model voxel generation method—that is, the method of voxel molding (MVM).

We converted the original space's image data to voxels according to the display's requirements, which conform to the display unit's geometric characteristics. Then we inputted the voxels to the display unit for calibration and calculation.

In the cylinder space generated by an LED screen's rotation, if each voxel's coordinates generated at a certain angle (such as 3°) are (x, y, z), then the time corresponding to each voxel is $t_0, t_1, t_2, t_3, \ldots, t_n$.

Assuming that the 3D model from the acquisition module is stored in the form of a point cloud, the MVM steps are as follows:

1. We put the 3D model into the voxel space, and the edges of the model coincided with or approximated some voxels in the space, as shown in Fig. 1.

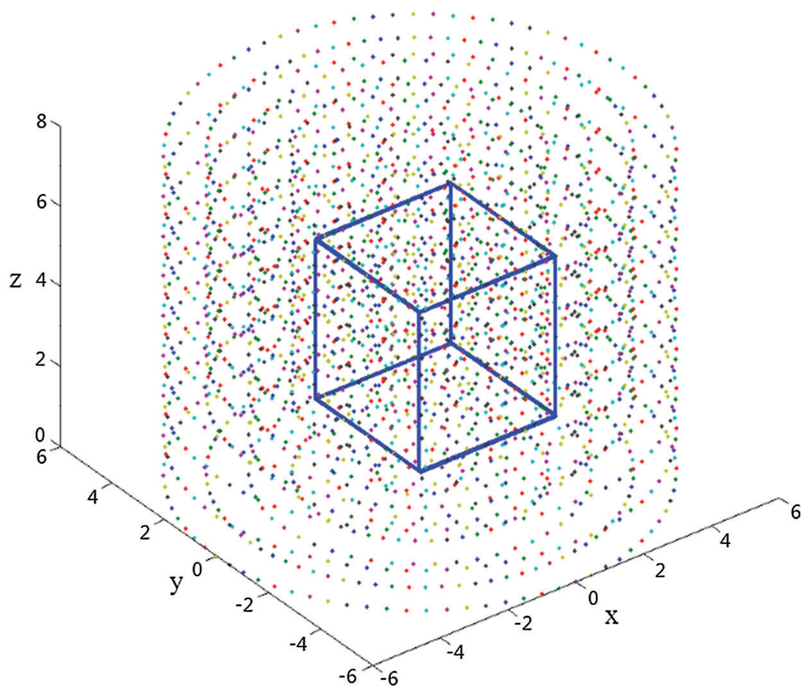

Figure 1: Schematic of 3D mold in voxel space

2. The voxel (x_i, y_j, z_k) with the same time $\{t \in [t_0, t_1, t_2, t_3, \ldots, t_n]\}$ was set as $A = \{(x_i, y_j, z_k, t_n)\}$, that is, $A_1 = (x_i, y_j, z_k, t_1)$, $A_2 = (x_i, y_j, z_k, t_2)$, …, $A_n = (x_i, y_j, z_k, t_n)$.

3. We reflected the voxels' data to the LED screen to obtain a set of 3D coordinates.

2.2 Establishment of GR State-space Model

The 3D position coordinates are combined with 1D time coordinates to create a 4D GR state-space model:

$$\begin{bmatrix} x^h(n_1+1, n_2, n_3, t) \\ x^v(n_1, n_2+1, n_3, t) \\ x^l(n_1, n_2, n_3+1, t) \\ x^t(n_1, n_2, n_3, t+1) \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ A_5 & A_6 & A_7 & A_8 \\ A_9 & A_{10} & A_{11} & A_{12} \\ A_{13} & A_{14} & A_{15} & A_{16} \end{bmatrix} \begin{bmatrix} x^h(n_1, n_2, n_3, t) \\ x^v(n_1, n_2, n_3, t) \\ x^l(n_1, n_2, n_3, t) \\ x^t(n_1, n_2, n_3, t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} u(n_1, n_2, n_3, t), \tag{1}$$
$$y(n_1, n_2, n_3, t) = Cx(n_1, n_2, n_3, t) + Du(n_1, n_2, n_3, t),$$

where

$$x(n_1,n_2,n_3,t) = (x^{h^T}(n_1,n_2,n_3,t), x^{v^T}(n_1,n_2,n_3,t), x^{l^T}(n_1,n_2,n_3,t), x^{t^T}(n_1,n_2,n_3,t))$$
$$= (x^{h^T}(n_1,n_2,n_3,t), x^{v^T}(n_1,n_2,n_3,t), x^{l^T}(n_1,n_2,n_3,t), x^{t^T}(n_1,n_2,n_3,t)),$$

and $n_1 \in Z$, $n_2 \in Z$, $n_3 \in Z$, and $t \in Z$. The vectors $x^h \in R^a$, $x^v \in R^b$, $x^l \in R^c$, and $x^t \in R^d$ are the vectors of the x-, y-, z-, and time t-axis, respectively. The input vector $u \in R^p$ and the output vector $y \in R^q$, $A_1 \in R^{a\times a}$, $A_2 \in R^{a\times b}$, $A_3 \in R^{a\times c}$, $A_4 \in R^{a\times d}$, $A_5 \in R^{b\times a}$, $A_6 \in R^{b\times b}$, $A_7 \in R^{b\times c}$, $A_8 \in R^{b\times d}$, $A_9 \in R^{c\times a}$, $A_{10} \in R^{c\times b}$, $A_{11} \in R^{c\times c}$, $A_{12} \in R^{c\times d}$, $A_{13} \in R^{d\times a}$, $A_{14} \in R^{d\times b}$, $A_{15} \in R^{d\times c}$, $A_{16} \in R^{d\times d}$, $B_1 \in R^{a\times p}$, $B_2 \in R^{b\times p}$, $B_3 \in R^{c\times p}$, $B_4 \in R^{d\times p}$, $C \in R^{q\times(a+b+c+d)}$, and $D \in R^{q\times p}$.

Assuming that

$$A = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ A_5 & A_6 & A_7 & A_8 \\ A_9 & A_{10} & A_{11} & A_{12} \\ A_{13} & A_{14} & A_{15} & A_{16} \end{bmatrix},$$

and letting r denote the order of the matrix, then $r = a + b + c + d$.

$B = \left(B_1^T, B_2^T, B_3^T, B_4^T\right)^T$. For the single input single output (SISO) system, $p = q = 1$; for the multiple input multiple output system, $p > 1$, $q > 1$. The voxel causal linear process in the first octant can be expressed in this model.

Considering the voxel space of size $N_1 \times N_2 \times N_3$, the display of each voxel is a linear process. Because the 3D mathematical model must be observable and realizable, the entire voxel space display is a process of linear causality in the first octant. The state-space model can represent the causality in the first octant.

3 Implementation Steps of GR Model with EOA Transformation

We performed EOA transformation to simplify the implementation of the GR model to the supplementary and transformation operations of the multidimensional characteristic polynomial matrix. This method featured easy calculation and could be used analyze the coefficient correlations on the system implementation matrix. We proposed a 4D EOA transformation and obtained the state-space model of the GR model through matrix operation.

3.1 Elementary Transformation of Matrices

Numerous elementary transformations of matrices are needed for obtaining the state-space matrix. Several of these transformations are defined below.

Let M denote a matrix and the following four kinds of row (column) transformations for a matrix denote the elementary row (column) transformations of a matrix.

1. $swaprow(M, i, j)$: Exchange the elements in the ith and jth rows of matrix M.

2. $addrow(M, i, j, k)$: Multiply all the elements in the ith row of matrix M by k, and add them to the corresponding elements in the jth row.

3. $swapcol(M, i, j)$: Exchange the elements in the ith and jth columns of matrix M.

4. $addcol(M, i, j, k)$: Multiply all the elements in the ith column of matrix M by k, and add them to the corresponding elements in the jth column.

In addition, define $augment(M) = \begin{bmatrix} 1 & 0 \\ 0 & M \end{bmatrix}$.

3.2 4D EOA Transformation

Next, we implemented the matrix transformation for the 4D GR model in the SISO case ($p = q = 1$) as follows:

Let $Z = diag\{z_1 I_a, z_2 I_b, z_3 I_c, z_4 I_d\}$ and define

$$M = \begin{bmatrix} x & B^T \\ (CZ)^T & (I_r - AZ)^T \end{bmatrix}$$

$$= \begin{bmatrix} x & B_1 & B_2 & B_3 & B_4 \\ C_1 z_1 & I_a - (A_1 z_1)^T & -(A_5 z_1)^T & -(A_9 z_1)^T & -(A_{13} z_1)^T \\ C_2 z_2 & -(A_2 z_2)^T & I_b - (A_5 z_2)^T & -(A_{10} z_2)^T & -(A_{14} z_2)^T \\ C_3 z_3 & -(A_3 z_3)^T & -(A_7 z_3)^T & I_c - (A_{11} z_3)^T & -(A_{15} z_3)^T \\ C_4 z_4 & -(A_4 z_4)^T & -(A_8 z_4)^T & -(A_{12} z_4)^T & I_d - (A_{16} z_4)^T \end{bmatrix}, \tag{2}$$

$$M_0 = \begin{bmatrix} x & 1 \\ n(z_1, z_2, z_3, z_4) & d(z_1, z_2, z_3, z_4) \end{bmatrix}. \tag{3}$$

As shown in Y. Xiong [12], the 4D GR model's implementation problem is to convert M_0 into matrix M by supplementing operations and elementary transformation without changing the determinant value.

The matrix M requires the following properties:

1. The first element on the diagonal can only be x.
2. Other elements on the diagonal can only be 1D linear polynomials about variables $z_k, k \in \{1, 2, 3, 4\}$, and the constant term can only be 1.
3. In addition to the first line, the off-diagonal element can only be a linear monomial about variables $z_k, k \in \{1, 2, 3, 4\}$.
4. In addition to x, the elements of the first row are constant terms.
5. The elements of the same row can only contain the same elements $z_k, k \in \{1, 2, 3, 4\}$, and from the second row all the rows are arranged in the order z_1, z_2, z_3, z_4.
6. The first element x on the diagonal is just a symbol, not a variable. During transformation, the position and expression of x cannot be changed.

The specific steps of the EOA transformation are as follows:

Step 1: Let any four-dimensional polynomial with no constant term be $p'(z_1, z_2, z_3, z_4)$. One of the variables $z_k, k \in \{1, 2, 3, 4\}$ can be decomposed into the following form:

$$p'(z_1, z_2, z_3, z_4) = p_1(z_k) + p_2(z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_4) + z_k p_3(z_1, z_2, z_3, z_4), \tag{4}$$

where $p_1(z_k)$ is a 1D linear polynomial containing only z_k; $p_2(z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_4)$ is a 4D linear polynomial without z_k. $p_1(z_k)$, $p_2(z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_4)$, and $p_3(z_1, z_2, z_3, z_4)$ do not include a constant term.

Letting $\hat{d}(z_1, z_2, z_3, z_4) = d(z_1, z_2, z_3, z_4) - 1$, the initial matrix M_0 is

$$M_0 = \begin{bmatrix} x & 1 \\ n(z_1, z_2, z_3, z_4) & 1 + \hat{d}(z_1, z_2, z_3, z_4) \end{bmatrix}. \tag{5}$$

Let $\hat{p} = \hat{d}(z_1, z_2, z_3, z_4)$, $\hat{q} = \hat{d}(z_1, z_2, z_3, z_4)$. According to Eq. (4), x and y are decomposed into

$$\hat{p} = p_1 + p_2 + z_1 p_3,$$
$$\hat{q} = q_1 + q_2 + z_1 q_3, \tag{6}$$

where p_1 and q_1 are 1D linear monomials that contain only z_1; p_2 and q_2 are 3D linear polynomials that do not contain z_1; and p_3 and q_3 are 4D polynomials.

The following operations are then performed:

$$M_1 = augment(M_0) = addrow(M_1, 3, 2, -z_1) = addcol(M_1, 3, 2, p_3) =$$
$$addcol(M_1, 3, 1, q_3) = \begin{bmatrix} x & 1 & 0 \\ q_1 + q_2 & 1 + p_1 + p_2 & -z_1 \\ q_3 & p_3 & 1 \end{bmatrix}. \tag{7}$$

Then, the new rows generated by each operation in p_3, q_3, and M_1 are sequentially operated so that terms with z_1 in M_1 become linear monomials concerning z_1.

Each row in M_1 performs the same operation on the variables z_1, z_2, z_3, and z_4 in turn, so that the diagonal elements other than x in M_1 are all 4D linear polynomials with the constant term of 1.

Step 2: Assume that the matrix M_1 obtained through Step 1 is

$$M_1 = \begin{bmatrix} x & 1 & 0 \\ a_1 z_1 + a_2 z_2 + a_3 z_3 + a_4 z_4 & 1 + b_1 z_1 + b_2 z_2 + b_3 z_3 + b_4 z_4 & c_1 z_1 + c_2 z_2 + c_3 z_3 + c_4 z_4 \\ * & \# & 1 \end{bmatrix}, \tag{8}$$

where * and # are both linear polynomials; a_i, b_i, c_i, and d_i are all coefficients, and $i = \{1,2,3,4\}$.

Convert matrix M_1 obtained in the first step to M_2 so that the diagonal elements in M_2 except x are 1D linear polynomials with a constant term of 1, and the non-diagonal elements except the first row are all linear monomials about the variables $z_k, k \in \{1, 2, 3, 4\}$.

Perform the following operation on M_1 in Eq. (8):

$$M_2 = augment(M_1) \to M_2 = addrow(M_2, 4, 2, -1) \to$$
$$M_2 = addcol(M_2, 4, 1, a_2 z_2) \to M_2 = addcol(M_2, 4, 2, b_2 z_2) \to$$
$$M_2 = addcol(M_2, 4, 3, c_2 z_2) = \begin{bmatrix} x & 1 & 0 & 0 \\ a_1 z_1 + a_3 z_3 + a_4 z_4 & 1 + b_1 z_1 + b_3 z_3 + b_4 z_4 & c_1 z_1 + c_3 z_3 + c_4 z_4 & -1 \\ * & \# & 1 & 0 \\ a_2 z_2 & b_2 z_2 & c_2 z_2 & 1 \end{bmatrix}. \tag{9}$$

In the same row, perform similar operations on variables z_3 and z_4. Then, matrix M_2 is finally obtained and the first three elements of the second row in M_2 are $a_1 z_1$, $1 + b_1 z_1$, and $c_1 z_1$, respectively; the remaining elements are -1:

$$M_2 = \begin{bmatrix} x & 1 & 0 & 0 & 0 & 0 \\ a_1 z_1 & 1 + b_1 z_1 & c_1 z_1 & -1 & -1 & -1 \\ * & \# & 1 & 0 & 0 & 0 \\ a_2 z_2 & b_2 z_2 & c_2 z_2 & 1 & 0 & 0 \\ a_3 z_3 & b_3 z_3 & c_3 z_3 & 0 & 1 & 0 \\ a_4 z_4 & b_4 z_4 & c_4 z_4 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Step 3: Through appropriate row and column transformations, each row in M_2 is arranged in the order z_1, z_2, z_3, and z_4, and all of the 1D linear polynomial elements are moved to the diagonal position. Then, the term -1 is eliminated through column transformation. Finally, matrix M_3 is obtained:

$$M_3 = \begin{bmatrix} x & 1 & 0 & 1 & 0 & 0 \\ a_1z_1 & 1+b_1z_1 & c_1z_1 & b_1z_1 & b_1z_1 & b_1z_1 \\ * & \# & 1 & \# & \# & \# \\ a_2z_2 & b_2z_2 & c_2z_2 & 1+b_2z_2 & b_2z_2 & b_2z_2 \\ a_3z_3 & b_3z_3 & c_3z_3 & b_3z_3 & 1+b_3z_3 & b_3z_3 \\ a_4z_4 & b_4z_4 & c_4z_4 & b_4z_4 & b_4z_4 & 1+b_4z_4 \end{bmatrix}, \tag{11}$$

and matrices A, B, and C are derived from Eq. (11).

3.3 Example

A strict causal transfer function is expressed as follows:

$$H(z_1, z_2, z_3, z_4) = \frac{z_1 + z_1z_2 + z_1z_3 + z_1z_4}{1 + z_1z_2 + z_1z_3 + z_2z_3 + z_1z_4}, \tag{12}$$

and, due to the strict causality, $D = H(0,0,0,0) = 0$.

The initial matrix is constructed as follows:

$$M_0 = \begin{bmatrix} x & 1 \\ z_1 + z_1z_2 + z_1z_3 + z_1z_4 & 1 + z_1z_2 + z_1z_3 + z_2z_3 + z_1z_4 \end{bmatrix} \text{ and} \tag{13}$$

transformed to give

$$\begin{aligned}
&M_1 = augment(M_0) \rightarrow M_1 = addrow(M_1, 3, 2, -z_1) \rightarrow M_1 = addcol(M_1, 3, 1, z_2 + z_3 + z_4) \rightarrow \\
&M_2 = augment(M_1) \rightarrow M_2 = addrow(M_2, 4, 2, -z_2) \rightarrow M_2 = addcol(M_2, 4, 2, z_3) \rightarrow \\
&M_3 = augment(M_2) \rightarrow M_3 = addrow(M_3, 5, 2, -1) \rightarrow M_3 = addcol(M_3, 5, 4, -z2) \rightarrow \\
&M_4 = augment(M_3) \rightarrow M_4 = addrow(M_4, 6, 3, -1) \rightarrow \\
&M_4 = addcol(M_4, 6, 1, z_3 + z_4) \rightarrow M_4 = addcol(M_4, 6, 2, z_3 + z_4) \rightarrow \\
&M_5 = augment(M_4) \rightarrow M_5 = addrow(M_5, 7, 6, -1) \rightarrow \\
&M_5 = addcol(M_5, 7, 1, z_4) \rightarrow M_5 = addcol(M_5, 7, 2, z_4) \rightarrow \\
&M_5 = swaprow(M_5, 4, 5) \rightarrow M_5 = swapcol(M_5, 4, 5) \rightarrow
\end{aligned}$$

Then,

$$M = \begin{bmatrix} x & 1 & 0 & 1 & 0 & 0 & 0 \\ z_1 & 1 & -z_1 & 0 & 0 & -z_1 & -z_1 \\ z_2 & z_2 & 1 & z_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -z_2 & 0 & 0 \\ 0 & z_3 & 0 & z_3 & 1 & 0 & 0 \\ z_3 & z_3 & 0 & z_3 & 0 & 1 & 0 \\ z_4 & z_4 & 0 & z_4 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

is obtained.

According to Eq. (2), $a = 1$, $b = 2$, $c = 2$, and $d = 1$. Therefore, the order of the realization matrix is $r = 6$. Then,

$$A = \begin{bmatrix} 0 & -1 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{15}$$

$C = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1], D = 0,$

are obtained. At time t, and letting the control variables

$$u(n_1, n_2, n_3) = \begin{bmatrix} n_1 + 5 \\ 0 \\ n_2 + 10 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{16}$$

the distribution of variables in the 3D volume space at t is simulated. The results are shown in Fig. 2.

4 Performance Analysis of GR model

We then studied the real-time performance and stability of the true 3D display system from the GR model's state-space matrix perspective.

4.1 Real-time Analysis

4.1.1 Working Process of True 3D Display System

In the GR model, (n_1, n_2, n_3) are the coordinates of the voxel in space, and t represents the sampling time index. $x(n_1, n_2, n_3, t)$, $y(n_1, n_2, n_3, t)$, and $u(n_1, n_2, n_3, t)$ are the state, output, and input vectors of the voxel, respectively. At time t, voxel (n_1, n_2, n_3) sends vector $x^h(n_1 + 1, n_2, n_3, t)$ to voxel $(n_1 + 1, n_2, n_3)$, $x^v(n_1, n_2 + 1, n_3, t)$ to voxel $(n_1, n_2 + 1, n_3)$, and $x^l(n_1, n_2, n_3 + 1, t)$ to voxel $(n_1, n_2, n_3 + 1)$. Then, we calculated the output vector $y(n_1, n_2, n_3, t)$.

In information processing, the problem of communication delay has always been difficult to solve [16]. For a true 3D display system with high real-time requirements and large scale, the communication between voxels will be seriously delayed, which is not conducive to the microcontroller's data calculation and analyses. Therefore, changing the system matrix to satisfy the real-time requirements of the system is critical.

4.1.2 Implementation of Delayed Response

In model (1), if matrices A_1, A_2, A_3, A_5, A_6, A_7, A_9, A_{10}, and A_{11} are all zero, the substate vector $x^h(n_1 + 1, n_2, n_3, t)$ of voxel $(n_1 + 1, n_2, n_3)$ in the x-axis direction, $x^v(n_1, n_2 + 1, n_3, t)$ of voxel $(n_1, n_2 + 1, n_3)$in the y-axis direction, and $x^l(n_1, n_2, n_3 + 1, t)$ of voxel $(n_1, n_2, n_3 + 1)$ in the z-axis direction are only related to $x^t(n_1, n_2, n_3, t)$. Thus, it is not necessary to calculate the substate vectors for voxel (n_1, n_2, n_3) at time t. The substate vector $x^t(n_1, n_2, n_3, t + 1)$ is used only by voxel (n_1, n_2, n_3), and we perform this calculation process at time $t + 1$. In this way, substate vectors $x^h(n_1 + 1, n_2, n_3, t)$, $x^v(n_1, n_2 + 1, n_3, t)$, and $x^l(n_1, n_2, n_3 + 1, t)$ cannot be calculated at time t, and thus there is no delay in data communication. Model (1) is then changed to

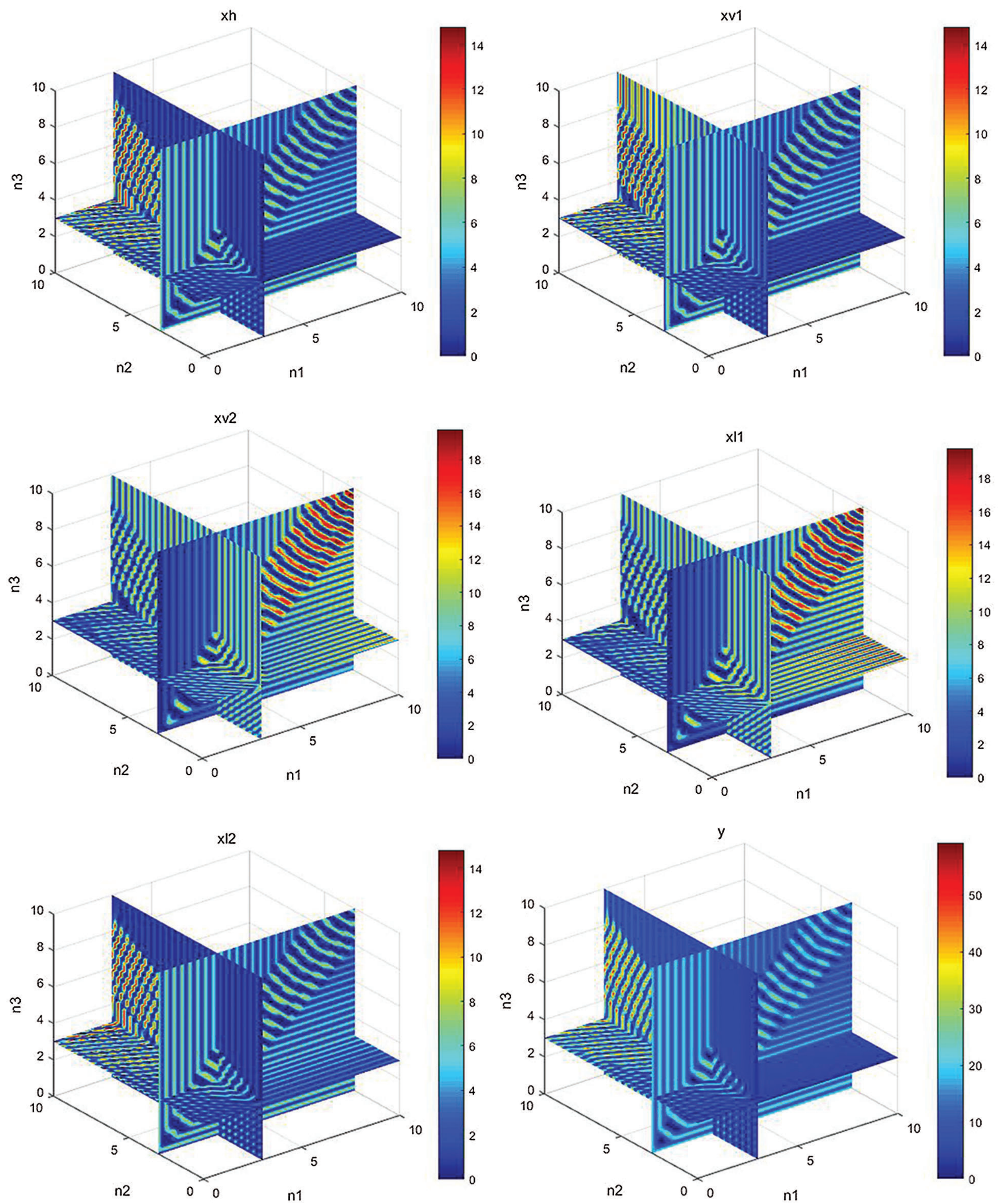

Figure 2: Simulation results in 3D space

$$\begin{bmatrix} x^h(n_1+1,n_2,n_3,t) \\ x^v(n_1,n_2+1,n_3,t) \\ x^l(n_1,n_2,n_3+1,t) \\ x^t(n_1,n_2,n_3,t+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & A_4 \\ 0 & 0 & 0 & A_8 \\ 0 & 0 & 0 & A_{12} \\ A_{13} & A_{14} & A_{15} & A_{16} \end{bmatrix} \begin{bmatrix} x^h(n_1,n_2,n_3,t) \\ x^v(n_1,n_2,n_3,t) \\ x^l(n_1,n_2,n_3,t) \\ x^t(n_1,n_2,n_3,t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} u(n_1,n_2,n_3,t). \tag{17}$$

4.2 Stability Analysis

4.2.1 Quantitative Model

In a true 3D display system, each state component represents the gray value in a specific direction that must be controlled between (0,255). Therefore, the calculation results of each voxel must be quantized after being transmitted to the microcontroller.

Since the display system may have dimensions of different sizes, or there may be a direction that requires higher accuracy, the calculation accuracy in each direction will be different when voxels are transmitting information. In this case, the following quantitative model is adopted:

$$\begin{aligned} &\left[x^h(n_1+1,n_2,n_3,t)\right] = Q_{ch}[[A_1 \quad A_2 \quad A_3 \quad A_4]x(n_1,n_2,n_3,t) + [B_1]U(n_1,n_2,n_3,t)], \\ &[x^v(n_1,n_2+1,n_3,t)] = Q_{cv}[[A_5 \quad A_6 \quad A_7 \quad A_8]x(n_1,n_2,n_3,t) + [B_2]U(n_1,n_2,n_3,t)], \\ &\left[x^l(n_1,n_2,n_3+1,t)\right] = Q_{cl}[[A_9 \quad A_{10} \quad A_{11} \quad A_{12}]x(n_1,n_2,n_3,t) + [B_3]U(n_1,n_2,n_3,t)], \\ &[x^t(n_1+1,n_2,n_3,t)] = Q_p[[A_1 \quad A_2 \quad A_3 \quad A_4]x(n_1,n_2,n_3,t) + [B_4]U(n_1,n_2,n_3,t)]. \end{aligned} \tag{18}$$

In this model, Q_{ch}, Q_{cv}, Q_{cl}, and Q_p denote the quantization operators of the x-, y-, z-, and t-axis, respectively.

4.2.2 Asymptotic Stability

In this paper, we studied the global asymptotic stability based on the 4D quantization model. Assuming that the volume space of the display system is $N_1 \times N_2 \times N_3$, then $n_1 \in [0, N_1-1]$, $n_2 \in [0, N_2-1]$, and $n_3 \in [0, N_3-1]$. First, the following conclusion is given.

Model (18) is asymptotically stable if and only if

$$x(t+1) = Q_p[A_{16}x(t)] \tag{19}$$

is asymptotically stable.

The detailed proof is shown in Yang et al. [17].

4.2.3 Example

The following is a concrete example of a true 3D display system. Let the GR model of a true 3D display system with a size of $4 \times 4 \times 4$ be expressed as follows:

$$\begin{bmatrix} x^h(n_1+1,n_2,n_3,t) \\ x^v(n_1,n_2+1,n_3,t) \\ x^l(n_1,n_2,n_3+1,t) \\ x^t(n_1,n_2,n_3,t+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.375 \end{bmatrix} \begin{bmatrix} x^h(n_1,n_2,n_3,t) \\ x^v(n_1,n_2,n_3,t) \\ x^l(n_1,n_2,n_3,t) \\ x^t(n_1,n_2,n_3,t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} U(n_1,n_2,n_3,t), \tag{20}$$

where $0 \le n_1 \le 4$, $0 \le n_2 \le 4$, $0 \le n_3 \le 4$, and $U(n_1,n_2,n_3,t) = \begin{cases} 1 & 0 \le t \le 5 \\ 0 & else \end{cases}$.

As explained in Section 4.2.2, system (20) is asymptotically stable if and only if the system

$$x(t+1) = Q_p[A_{16}x(t)] \tag{21}$$

is asymptotically stable.

The Euclidean norm of state vectors of voxels $(4, 4, 4)$ and $(1, 1, 1)$ are obtained by simulation, represented by $p1$ and $p2$, respectively. The simulation results are shown in Fig. 3.

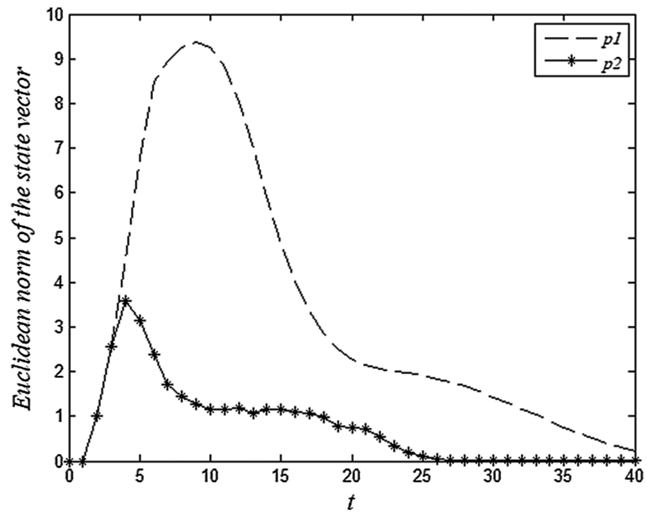

Figure 3: Euclidean norm of voxel $(4, 4, 4)$and $(1, 1, 1)$ state vectors in true 3D display system of size $4 \times 4 \times 4$

As shown in Fig. 3, after some time, the state vectors of voxels $(4, 4, 4)$ and $(1, 1, 1)$ tended to be 0, whereas the state vector of voxel $(4, 4, 4)$ reached 0 later than that of voxel $(1, 1, 1)$. This confirmed the causality of the system in the first octant.

5 Conclusions

For the modeling and analyses of a true 3D display system, we established a 4D GR model by combining the 3D azimuth coordinates and 1D time coordinates of voxels in the imaging space. Therefore, we obtained the state-space expression of the true 3D display system. We then proposed the implementation steps with 4D EOA transformation for the GR model's realization matrix; after describing the system working process, we analyzed the real-time and stability performance of the true 3D display system. By simplifying the system matrix, we derived the conditions for the global asymptotic stability of the system. Experimental results showed that in a true 3D display system with a size of $4 \times 4 \times 4$, the state vector of the voxel point converged to 0, and thus was asymptotically stable. The proposed GR-model-representation method and its implementation steps for a true 3D display system could simplify the system's mathematical expression and facilitate microcontroller software implementation. Real-time and stability analyses can be applied widely in the analysis and design of true 3D display systems.

Acknowledgement: We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

Funding Statement: This work was supported by the Key Research and Development Projects of Science and Technology Development Plan of Jilin Provincial Department of Science and Technology (20180201090gx).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Liu, "Review on research progress of true-3d volumetric display technology," *Advanced Display*, vol. 10, no. 5, pp. 30–40, 2010.

[2] R. N. Yang, W. X. Zheng and Y. R. Yu, "Event-triggered sliding mode control of discrete-time two-dimensional systems in Roesser model," *Automatica*, vol. 114, no. 5, pp. 108813, 2020.

[3] X. D. Z. Wu, "Realization method of multidimensional system and discussion of transformation between Roesser model and Fornasini-Marchesini model," M.S. theses. Wuhan University of Science and Technology, China, 2019.

[4] J. Yang, "Formation control and collaborative searching of swarm robotics," Ph.D. dissertation. Harbin Institute of Technology, China, 2018.

[5] B. Sumanasena and P. Bauer, "A Roesser model based multidimensional systems approach for grid sensor networks," in *IEEE Asilomar Conf. on Signals*, Pacific Grove, CA, USA, pp. 2151–2154, 2010.

[6] A. A. Hady, "Duty cycling centralized hierarchical routing protocol with content analysis duty cycling mechanism for wireless sensor networks," *Computer Systems Science and Engineering*, vol. 35, no. 5, pp. 347–355, 2020.

[7] M. E. Bayrakdar, "Cost effective smart system for water pollution control with underwater wireless sensor networks: a simulation study," *Computer Systems Science and Engineering*, vol. 35, no. 4, pp. 283–292, 2020.

[8] S. Kaur and V. K. Joshi, "Hybrid soft computing technique based trust evaluation protocol for wireless sensor networks," *Intelligent Automation & Soft Computing*, vol. 26, no. 2, pp. 217–226, 2020.

[9] K. Galkowski, "The state-space realization of ann-dimensional transfer function," *International Journal of Circuit Theory and Applications*, vol. 9, no. 2, pp. 189–197, 1981.

[10] L. Xu and S. Yan, "A new elementary operation approach to multidimensional realization and LFR uncertainty modeling: The MIMO case," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 3, pp. 638–651, 2012.

[11] L. Xu, H. J. Fan and Z. P. Lin, "A direct-construction approach to multidimensional realization and LFR uncertainty modeling," *Multidimensional Systems and Signal Processing*, vol. 19, no. 3, pp. 323–359, 2008.

[12] Y. Xiong, "Multidimensional Roesser state space model based on wireless sensor networks," M.S. theses. Wuhan University of Science and Technology, China, 2016.

[13] P. Kokil, "An improved criterion for the global asymptotic stability of 2d discrete state-delayed systems with saturation nonlinearities," *Birkhauser Boston Inc*, vol. 36, no. 1, pp. 2209–2222, 2017.

[14] G. Xin, W. Lin, C. Fuyong, H. Li and W. Zhang, "Reliability analysis of slope stability considering temporal variations of rock mass properties," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 263–281, 2020.

[15] P. Agathoklis and L. T. Bruton, "Practical-BIBO stability of n-dimensional discrete systems," *IEE Proceedings G (Electronic Circuits and Systems)*, vol. 130, no. 6, pp. 236–242, 1983.

[16] B. Nagaraj, D. Pelusi and J. I. Chen, "Special section on emerging challenges in computational intelligence for signal processing applications," *Intelligent Automation & Soft Computing*, vol. 26, no. 4, pp. 737–739, 2020.

[17] Y. Yang, Y. Tian, Z. Liu and G. L. Chen, "Roesser model of true 3d display system and its performance analysis," *Journal of Systems Science and Mathematical Sciences*, vol. 39, no. 4, pp. 534–544, 2019.