Numerical and Symbolic State Space Realizations of Linear Systems with Algebraic Loops

Masanobu Koga, Satoru Anan, and Kentaro Yano

***Abstract*— This paper proposes a method which gives a state space realization of the linear systems with algebraic loops not only in numerical format but also in symbolic format and guarantees that the order of the system is preserved. The method uses an adjacency matrix in graph theory for the modeling of the control systems and uses the matrix inversion lemma to calculate the inverse of matrix symbolically and to obtain the symbolic state space realizations. We have developed a new software platform for modeling and simulation of control systems using the proposed method.**

I. INTRODUCTION

In a typical control engineering course, students are required to develop skills in manipulating transfer functions and state space realizations of the systems. Although the underlying principles are easy to grasp, the manipulation involved, especially in large interconnected systems, are very tedious. Computer software tools for handling the symbolic state space realizations are useful for the education and aids to get formulas involving the parameters which we want to vary [1]. The symbolic realizations enable us to solve in a unified way a variety of control problems [2] and gives us the exact solutions, so there is no possibility of cumulative error due to rounding off.

It is easy to get a state space realization of an interconnected linear system, but the order of the obtained system may be higher than that of the original ones and the reduction procedure might cause the cancellation problems.

Although there are many tools which provides numerical state space realizations of the linear systems [3], [4], [5], few tool gives symbolic state space realizations [3]. And the symbolic state space realizations can be obtained only for the SISO systems without algebraic loops.

This paper proposes a method which gives a state space realization of the linear systems with algebraic loops not only in numerical format but also in symbolic format and guarantees that the order of the system is preserved. The method uses an adjacency matrix in graph theory for the modeling of the control system and uses the matrix inversion lemma to calculate the inverse of matrix symbolically and to obtain the symbolic state space realizations.

We have developed a new software platform for modeling and simulation of control systems using the proposed method. It provides an interactive graphical environment where a block diagram is created by dragging and drop blocks from the block library to the canvas and connecting them. If all the blocks are linear systems, we can obtain a state space realization of the whole system by selecting an appropriate menu.

In Section 2, the adjacency matrix representation of a proper linear system is defined. Section 3 gives the algorithm for the numerical and symbolic state space realization of system with algebraic loops. A new software platform for modeling and simulation of the control systems based on the proposed method is introduced with an illustrative example in Section 4 The related works are compared in Section 5 and Section 6 summarizes the paper.

II. ADJACENCY MATRIX REPRESENTATION

The adjacency matrix is a representation of a *graph* and the (i, j) element contains the adjacency information between the node i and the node j. When we regard an arrow of the block diagram as a node of graph, and a block and a summing point of the block diagram as an edge of graph, we can consider a causal block diagram as a weighted directed graph whose weights correspond to the values of the blocks. Those graphs are *simple graph* since they have no multi-edge and no self-loop. Therefore the data of a block digram can be represented as an adjacency matrix.

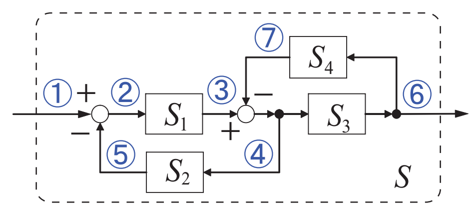

Fig. 1. Control System

For example, the adjacency matrix of the system S shown in Fig.1 is given by

$$
W = \begin{array}{c}
\begin{array}{ccccccc} \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} & \textcircled{5} & \textcircled{6} & \textcircled{7} \end{array} \\
\left[\begin{array}{c|c|c|c|c|c|c}
 & I & & & & & \\ \hline
 & & S_1 & & & & \\ \hline
 & & & I & & & \\ \hline
 & & & & S_2 & S_3 & \\ \hline
 & -I & & & & & \\ \hline
 & & & & & & S_4 \\ \hline
 & & & -I & & &
\end{array}\right]
\end{array}
\begin{array}{c} \\ \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \\ \textcircled{5} \\ \textcircled{6} \\ \textcircled{7} \end{array}
\qquad (1)
$$

, where S_i is an operator which represents input-output relationship of the system, I is an unit operator which transmits the input without modification, $-I$ is an unary minus operator whose output is opposite in sign to the

Masanobu Koga, Satoru Anan, and Kentaro Yano are with Department of Systems Design and Informatics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka,Fukuoka, Japan koga@ces.kyutech.ac.jp

input. Let v_i denotes the value of node i, then the following equation holds.

$$\begin{bmatrix} 0 \\ v_2 \\ \vdots \\ v_7 \end{bmatrix} = W^T \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_7 \end{bmatrix}$$

A. Adjacency Matrix of Linear Systems

Figure 2 shows the block diagram of a proper linear system, where $\frac{I}{q}$ is a diagonal matrix with integrators as diagonal elements for continuous-time systems and is a diagonal matrix with delay operators as diagonal elements for discrete-time systems. And I is an unit matrix.

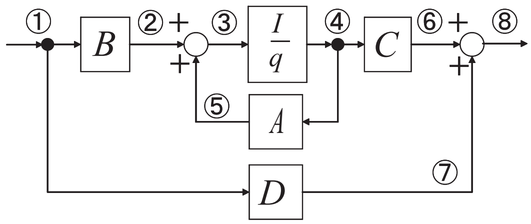

Fig. 2. Proper Linear System

The adjacency matrix of the linear system is given as follows.

$$\begin{array}{cc} \begin{array}{cccccccc} ① & ② & ③ & ④ & ⑤ & ⑥ & ⑦ & ⑧ \end{array} & \\ \left[\begin{array}{c|c|c|c|c|c|c|c} & B & & & & & D & \\ \hline & & I & & & & & \\ \hline & & & I/q & & & & \\ \hline & & & & A & C & & \\ \hline & & I & & & & & \\ \hline & & & & & & & I \\ \hline & & & & & & & I \\ \hline & & & & & & & \end{array}\right] & \begin{array}{c} ① \\ ② \\ ③ \\ ④ \\ ⑤ \\ ⑥ \\ ⑦ \\ ⑧ \end{array} \end{array}$$

In this paper, we call a *series-edge* with a constant weight as a *constant series-edge*. It is possible to contract a constant series-edge to an edge whose weight is the product of the weights of the original series-edge. Contracting all the constant series-edges of the graph in Fig. 2 gives the following matrix.

$$\begin{array}{cc} \begin{array}{cccc} ① & ③ & ④ & ⑧ \end{array} & \\ \left[\begin{array}{c|c|c|c} & B & & D \\ \hline & & I/q & \\ \hline & A & & C \\ \hline & & & \end{array}\right] & \begin{array}{c} ① \\ ③ \\ ④ \\ ⑧ \end{array} \end{array}$$

Exchanging the number of node-3 and node-4 (exchanging the 3rd row and the 4th row, and the 3rd column and the 4th column, respectively), leads to the next definition.

Definition 1: The adjacency matrix of a proper linear system is given by

$$\left[\begin{array}{c|c|c|c} & & B & D \\ \hline & & A & C \\ \hline & I/q & & \\ \hline & & & \end{array}\right] \tag{2}$$

and the equation

$$\begin{bmatrix} 0 \\ x \\ q \cdot x \\ y \end{bmatrix} = \left[\begin{array}{c|c|c|c} & & B^T & D^T \\ \hline & & A^T & C^T \\ \hline & I/q & & \\ \hline & & & \end{array}\right]^T \begin{bmatrix} u \\ x \\ q \cdot x \\ y \end{bmatrix}$$

holds, where u is the input, y is the output, and x is the state of the linear system.

III. NUMERICAL AND SYMBOLIC STATE SPACE REALIZATION

In this section we give an algorithm to obtain numerical and symbolic state space realizations of the interconnected linear systems.

A. State Space Realization of Linear Systems

Suppose that all the subsystems in a given block diagram are proper linear dynamic system or constant matrix, there are m external inputs and p external outputs, and the well-posedness condition of the loop are satisfied. The following algorithm gives the state space realization of the whole system as an analytic expression of $\{A_i, B_i, C_i, D_i\}$ $(i = 1, \ldots, N)$ which are coefficient matrices of the state space realization of the subsystems, where N denotes the number of subsystems. The order of the whole system is preserved and it is not required to do any pole zero cancellations.

Algorithm 1:

1) Get the adjacency matrix of the whole system using the adjacency matrices of the subsystems.
2) Contract all constant series-edges which don't connect with the node which has any self-loops.
3) If there exists any self-loop, i.e., there exists a nonzero diagonal element, collect up all nonzero diagonal elements to a diagonal submatrix by appropriately sorting the nodes. Let L_i denotes the submatrix. Multiply $P_i := (I - L_i)^{-1}$ to the elements in the the columns whose column numbers are equal to those of L_i from left and set O to L_i.
4) Sort the nodes (exchange both rows and columns) such that the integrators (delay operators) get lined up in diagonal in the left lower part of the adjacency matrix, where the first m columns and the last p rows are empty.
5) Sort the nodes which correspond to the external inputs and the external outputs by the number of those signals.
6) Get the coefficient matrices of the state space realization from the right upper part of the adjacency matrix using the structure defined in (2).

B. Elimination of Self loop (Algebraic loop)

In algorithm 1, we have to obtain the inverse of the submatrix which contains the weights of the self loops. If the numerical values of the weights are given, it is easy to calculate the inverse of matrix numerically. In this paper, we use the following matrix inversion lemma to calculate the inverse of matrix symbolically and to obtain the symbolic state space realizations.

Matrix inversion lemma:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1}+A^{-1}BE^{-1}CA^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}CA^{-1} & E^{-1} \end{bmatrix}$$

$$E := D - CA^{-1}B, \quad |A| \neq 0, \quad |E| \neq 0 \tag{3}$$

C. System with a Single Algebraic Loop

In this subsection, we deal with the system shown in Fig.3 where S_1 and S_2 are both bi-proper linear system.

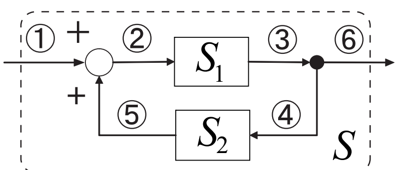

Fig. 3. System with Single Algebraic Loop

The adjacency matrix of the whole system is given by

$$\begin{array}{c c}
\begin{array}{cccccc} \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} & \textcircled{5} & \textcircled{6} \end{array} & \\
\left[\begin{array}{c|c|c|c|c|c}
 & I & & & & \\ \hline
 & & S_1 & & & \\ \hline
 & & & I & & I \\ \hline
 & & & & S_2 & \\ \hline
 & I & & & & \\ \hline
 & & & & &
\end{array}\right] & \begin{array}{c} \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \\ \textcircled{5} \\ \textcircled{6} \end{array}
\end{array} \tag{4}$$

Substitute the adjacency matrices of S_1 and S_2 to the matrix above and contract all constant series-edges and sort the number of the nodes (exchange both rows and columns) such that the integrators (delay operators) get lined up in diagonal in the left lower part of the adjacency matrix, where the first column and the last row are empty, then we have

$$\begin{array}{c c}
\begin{array}{ccccccc} \textcircled{1} & \textcircled{3} & \textcircled{7} & \textcircled{9} & \textcircled{4} & \textcircled{8} & \textcircled{10} \end{array} & \\
\left[\begin{array}{c|c|c|c|c|c|c}
 & & & D_2D_1 & B_1 & B_2D_1 & D_1 \\ \hline
 & & & D_2C_1 & A_1 & B_2C_1 & C_1 \\ \hline
 & & & C_2 & & A_2 & \\ \hline
 & & & \mathbf{D_2D_1} & B_1 & B_2D_1 & D_1 \\ \hline
 & I_1/q & & & & & \\ \hline
 & & I_2/q & & & & \\ \hline
 & & & & & &
\end{array}\right] & \begin{array}{c} \textcircled{1} \\ \textcircled{3} \\ \textcircled{7} \\ \textcircled{9} \\ \textcircled{4} \\ \textcircled{8} \\ \textcircled{10} \end{array}
\end{array} \tag{5}$$

When both two linear systems are bi-proper, i.e., $D_1 \neq 0$ and $D_2 \neq 0$, the matrix in Eq. (5) has a nonzero diagonal element $L := D_2D_1$ at (4,4). It means that there exists a *self-loop* at node 9, and the value of node-9 depends on the value of itself. Let v_i denotes the value of node i, then we have

$$v_9 = D_2D_1v_1 + D_2C_1v_3 + C_2v_7 + Lv_9$$

from Eq. (5). If the well-posedness condition is satisfied, then $(I - L)$ is nonsingular and we have

$$v_9 = PD_2D_1v_1 + PD_2C_1v_3 + PC_2v_7 \tag{6}$$

, where

$$P = (I - L)^{-1}$$

In the single feedback case, the submatrix which corresponds to the self loops contains only one element, so we don't have to use the matrix inversion lemma to get the inverse matrix of the submatrix. From Eq. (5) and Eq. (6), we have

$$\begin{array}{c c}
\begin{array}{ccccccc} \textcircled{1} & \textcircled{3} & \textcircled{7} & \textcircled{9} & \textcircled{4} & \textcircled{8} & \textcircled{10} \end{array} & \\
\left[\begin{array}{c|c|c|c|c|c|c}
 & & & PD_2D_1 & B_1 & B_2D_1 & D_1 \\ \hline
 & & & PD_2C_1 & A_1 & B_2C_1 & C_1 \\ \hline
 & & & PC_2 & & A_2 & \\ \hline
 & & & & B_1 & B_2D_1 & D_1 \\ \hline
 & I_1/q & & & & & \\ \hline
 & & I_2/q & & & & \\ \hline
 & & & & & &
\end{array}\right] & \begin{array}{c} \textcircled{1} \\ \textcircled{3} \\ \textcircled{7} \\ \textcircled{9} \\ \textcircled{4} \\ \textcircled{8} \\ \textcircled{10} \end{array}
\end{array}$$

Additional contraction of constant series-edges and sorting the number of some nodes leads to

$$\tilde{W} = \begin{array}{c c}
\begin{array}{cccccc} \textcircled{1} & \textcircled{3} & \textcircled{7} & \textcircled{4} & \textcircled{8} & \textcircled{10} \end{array} & \\
\left[\begin{array}{c|c|c|c|c|c}
 & & & \begin{array}{c} B_1+ \\ B_1PD_2D_1 \end{array} & \begin{array}{c} B_2D_1+ \\ B_2D_1PD_2D_1 \end{array} & \begin{array}{c} D_1+ \\ D_1PD_2D_1 \end{array} \\ \hline
 & & & \begin{array}{c} A_1+ \\ B_1PD_2C_1 \end{array} & \begin{array}{c} B_2C_1+ \\ B_2D_1PD_2C_1 \end{array} & \begin{array}{c} C_1+ \\ D_1PD_2C_1 \end{array} \\ \hline
 & & & B_1PC_2 & \begin{array}{c} A_2+ \\ B_2D_1PC_2 \end{array} & D_1PC_2 \\ \hline
 & I_1/q & & & & \\ \hline
 & & I_2/q & & & \\ \hline
 & & & & &
\end{array}\right] & \begin{array}{c} \textcircled{1} \\ \textcircled{3} \\ \textcircled{7} \\ \textcircled{4} \\ \textcircled{8} \\ \textcircled{10} \end{array}
\end{array}$$

Since matrix $\tilde{W}$ has the same structure as that of the adjacency matrix of a proper linear system defined in Eq. (2), the coefficient matrices of the state space realization of the whole linear system are given by

$$\begin{aligned}
A_S &= \begin{bmatrix} \tilde{W}(2,4)^T & \tilde{W}(2,5)^T \\ \tilde{W}(3,4)^T & \tilde{W}(3,5)^T \end{bmatrix}^T \\
&= \begin{bmatrix} A_1 + B_1PD_2C_1 & B_1PC_2 \\ B_2C_1 + B_2D_1PD_2C_1 & A_2 + B_2D_1PC_2 \end{bmatrix} \\
B_S &= \begin{bmatrix} \tilde{W}(1,4)^T & \tilde{W}(1,5)^T \end{bmatrix}^T \\
&= \begin{bmatrix} B_1 + B_1PD_2D_1 \\ B_2D_1 + B_2D_1PD_2D_1 \end{bmatrix} \\
C_S &= \begin{bmatrix} \tilde{W}(2,6)^T \\ \tilde{W}(3,6)^T \end{bmatrix}^T = \begin{bmatrix} C_1 + D_1PD_2C_1 & D_1PC_2 \end{bmatrix} \\
D_S &= \begin{bmatrix} \tilde{W}(1,6)^T \end{bmatrix}^T = \begin{bmatrix} D_1 + D_1PD_2D_1 \end{bmatrix}
\end{aligned}$$

D. Systme with a General Algebraic Loop

In this subsection, we deal with the systems which have a general algebraic loop shown in Fig.4, where K_i $(i = 1, ..., n)$ are constant matrices of appropriate dimensions.

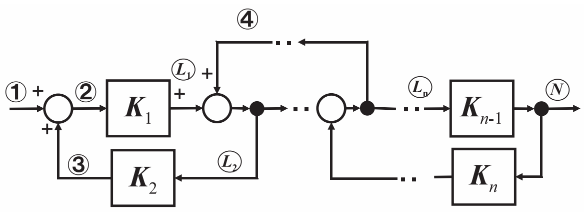

Fig. 4. System with a General Algebraic Loop

Contracting all constant series-edges which don't connect with the node which has any self-loops and sorting the nodes so that the nonzero diagonal elements in the adjacency matrix come together leads to the adjacency matrix

$$\begin{array}{c c}
\begin{array}{ccc} V_{\overline{\ell}} & V_{\ell} & V_{\underline{\ell}} \end{array} & \\
\left[\begin{array}{c|c|c} & \overline{L} & * \\ \hline * & L & * \\ \hline * & \underline{L} & \end{array}\right] & \begin{array}{c} V_{\overline{\ell}} \\ V_{\ell} \\ V_{\underline{\ell}} \end{array}
\end{array}$$

, where V_ℓ contains the values of the nodes in the algebraic loop, $V_{\overline{\ell}}$ and $V_{\underline{\ell}}$ contain the values of the nodes outside the algebraic loop, and L has nonzero diagonal elements. Then we can obtain V_ℓ as follows.

$$V_\ell = \begin{bmatrix} \overline{L}^T \\ L^T \\ \underline{L}^T \end{bmatrix}^T \begin{bmatrix} V_{\overline{\ell}} \\ V_\ell \\ V_{\underline{\ell}} \end{bmatrix}$$

and

$$(I - L)V_\ell = \begin{bmatrix} \overline{L}^T \\ \underline{L}^T \end{bmatrix}^T \begin{bmatrix} V_{\overline{\ell}} \\ V_{\underline{\ell}} \end{bmatrix} \tag{7}$$

We calculate

$$P = (I - L)^{-1}$$

symbolically by using the matrix inversion lemma, and multiply P to the both sides of (7) from left and we have

$$V_\ell = K \begin{bmatrix} V_{\overline{\ell}} \\ V_{\underline{\ell}} \end{bmatrix}, \quad K := \begin{bmatrix} \overline{K} & \underline{K} \end{bmatrix} = P \begin{bmatrix} \overline{L}^T \\ \underline{L}^T \end{bmatrix}^T$$

Then the adjacency matrix is given by

$$\begin{array}{c c}
\begin{array}{ccc} V_{\overline{\ell}} & V_{\ell} & V_{\underline{\ell}} \end{array} & \\
\left[\begin{array}{c|c|c} & \overline{K} & * \\ \hline * & & * \\ \hline * & \underline{K} & \end{array}\right] & \begin{array}{c} V_{\overline{\ell}} \\ V_{\ell} \\ V_{\underline{\ell}} \end{array}
\end{array}$$

Contracting the remaining constant series-edges and sorting the nodes according to Algorithm 1, we can obtain the symbolic state space realization of the system.

E. Systems with Multiple External Inputs and Outputs

In this subsection, we deal with a linear system with multiple external inputs and external outputs shown in Fig.5, where S_1 and S_2 are both proper linear system, and K_{11}, K_{12}, K_{21}, K_{22} are constant matrices of appropriate dimensions.

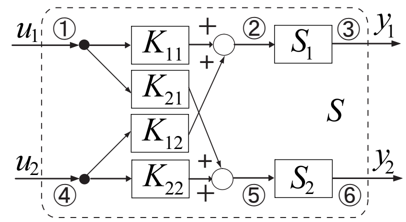

Fig. 5. Systems with Multiple External Inputs and Outputs

Contract all constant series-edges and sort the nodes (exchange both rows and columns) such that the integrators (delay operators) get lined up in diagonal in the left lower part of the adjacency matrix, where the first two columns and the last two rows are empty, then we have

$$\begin{array}{c c}
\begin{array}{cccccccc} ⑥ & ① & ③ & ⑧ & ④ & ⑨ & ⑩ & ⑤ \end{array} & \\
\left[\begin{array}{c|c|c|c|c|c|c|c}
 & & & & B_1K_{12} & B_2K_{22} & D_2K_{22} & D_1K_{12} \\ \hline
 & & & & B_1K_{11} & B_2K_{21} & D_2K_{21} & D_1K_{11} \\ \hline
 & & & & A_1 & & & C_1 \\ \hline
 & & & & & A_2 & C_2 & \\ \hline
 & & I_1/q & & & & & \\ \hline
 & & & I_2/q & & & & \\ \hline
 & & & & & & & \\ \hline
 & & & & & & &
\end{array}\right] & \begin{array}{l} ⑥(u_2) \\ ①(u_1) \\ ③ \\ ⑧ \\ ④ \\ ⑨ \\ ⑩(y_2) \\ ⑤(y_1) \end{array}
\end{array}$$

Sorting the nodes which correspond to the external inputs and the external outputs by the number of the ports, we obtain

$$\tilde{W} =$$

$$\begin{array}{c c}
\begin{array}{cccccccc} ① & ⑥ & ③ & ⑧ & ④ & ⑨ & ⑤ & ⑩ \end{array} & \\
\left[\begin{array}{c|c|c|c|c|c|c|c}
 & & & & B_1K_{11} & B_2K_{21} & D_1K_{11} & D_2K_{21} \\ \hline
 & & & & B_1K_{12} & B_2K_{22} & D_1K_{12} & D_2K_{22} \\ \hline
 & & & & A_1 & & C_1 & \\ \hline
 & & & & & A_2 & & C_2 \\ \hline
 & & I_1/q & & & & & \\ \hline
 & & & I_2/q & & & & \\ \hline
 & & & & & & & \\ \hline
 & & & & & & &
\end{array}\right] & \begin{array}{l} ①(u_1) \\ ⑥(u_2) \\ ③ \\ ⑧ \\ ④ \\ ⑨ \\ ⑤(y_1) \\ ⑩(y_2) \end{array}
\end{array}$$

Since matrix $\tilde{W}$ has the same structure as that of the adjacency matrix of a proper linear system defined in Eq. (2), the coefficient matrices of the state space realization of the whole linear system are given by

$$A_S = \begin{bmatrix} \tilde{W}(3,5)^T & \tilde{W}(3,6)^T \\ \tilde{W}(4,5)^T & \tilde{W}(4,6)^T \end{bmatrix}^T = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix}$$

$$B_S = \begin{bmatrix} \tilde{W}(1,5)^T & \tilde{W}(1,6)^T \\ \tilde{W}(2,5)^T & \tilde{W}(2,6)^T \end{bmatrix}^T = \begin{bmatrix} B_1K_{11} & B_1K_{12} \\ B_2K_{21} & B_2K_{22} \end{bmatrix}$$

$$C_S = \begin{bmatrix} \tilde{W}(3,7)^T & \tilde{W}(3,8)^T \\ \tilde{W}(4,7)^T & \tilde{W}(4,8)^T \end{bmatrix}^T = \begin{bmatrix} C_1 & \\ & C_2 \end{bmatrix}$$

$$D_S = \begin{bmatrix} \tilde{W}(1,7)^T & \tilde{W}(1,8)^T \\ \tilde{W}(2,7)^T & \tilde{W}(2,8)^T \end{bmatrix}^T = \begin{bmatrix} D_1 K_{11} & D_1 K_{12} \\ D_2 K_{21} & D_2 K_{22} \end{bmatrix}$$

IV. MODELING AND SIMULATION PLATFORM

We have developed a new software platform, Jamox (Java Agile MOdeling Tool for Control System), for modeling and simulation of control systems using the method described in this paper. Jamox provides an interactive graphical environment where a block diagram is created by dragging and drop blocks from the block library to the canvas and connecting them.

If all the blocks are linear systems, we can obtain the state space realization of the whole system not only in numerical format but also in symbolic format. Jamox also provides the facility to get the time responses for the specified inputs and the frequency responses between the specified inputs and outputs

Figure 6 shows the screen shot of Jamox which gives a numerical and symbolic state space realization of a linear system.

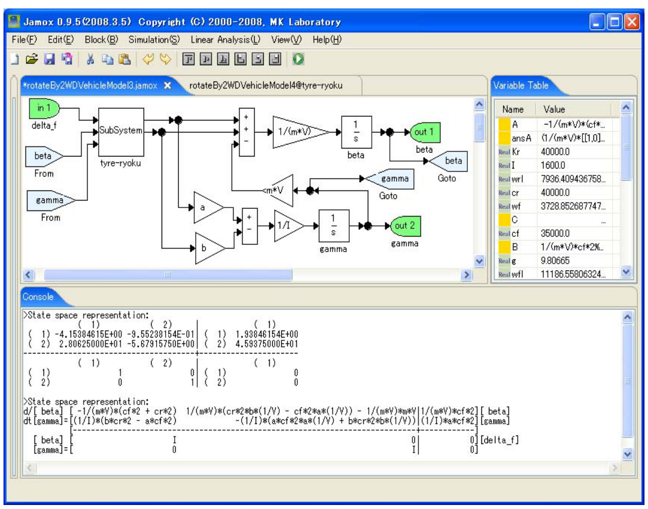

Fig. 6. State Space Realization (Jamox)

A. Example

Consider the mixed sensitivity problem whose block diagram is shown in Fig.7, where $P(s)$ is plant, $K(s)$ is controller, $W_S(s)$ and $W_T(s)$ are weighting functions, and

$$P(s) = \left[\begin{array}{c|c} A_p & B_p \\ \hline C_p & D_p \end{array}\right], K(s) = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array}\right]$$

$$W_S(s) = \left[\begin{array}{c|c} A_s & B_s \\ \hline C_s & D_s \end{array}\right], W_T(s) = \left[\begin{array}{c|c} A_t & B_t \\ \hline C_t & D_t \end{array}\right]$$

Figure 8 shows the symbolic state space realization of the generalized plant for the mixed sensitivity problem obtained by Jamox which is equal to the results in [1]. It is found that the order of the system is preserved.

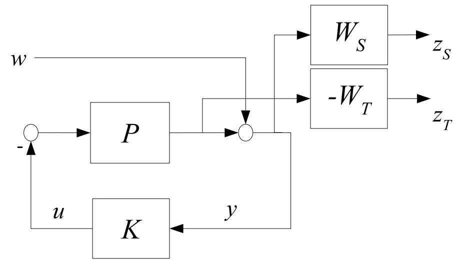

Fig. 7. Mixed Sensitivity Problem

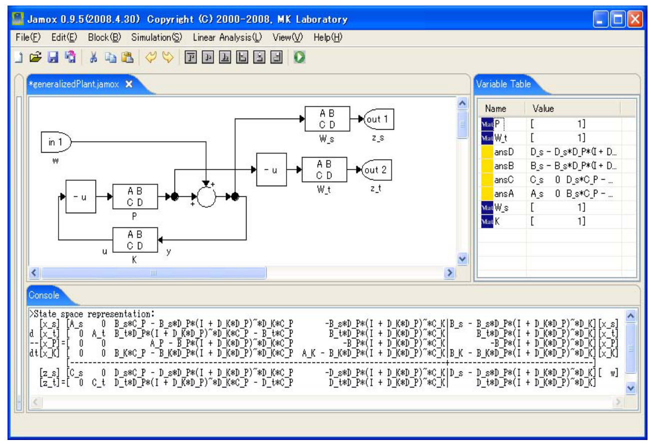

Fig. 8. Mixed Sensitivity Problem (Jamox)

V. RELATED WORKS

Table I shows the comparison of the feature among the related works for the modeling of the systems with algebraic loops. [4], [5], [3].

TABLE I
MODELING OF LINEAR SYSTEMS WITH ALGEBRAIC LOOPS

	Matlab/ Simulink	Scilab	Scicos	20-Sim	Jamox
MIMO	○	○	○	×	○
Numerical	○	○	×	△	○
Symbolic	×	△	×	△	○
Simulation	○	○	△	△	○

The following subsections describe in detail the modeling feature of the related works.

A. Matlab/Simulink

Figure 9 shows the numerical state space realization and simulation result by Simulink. Symbolic state space realization is not supported by the standard package of Matlab/Simulink as far as we know.

B. Scilab

Figure 10 shows the numerical state space realization by Scilab which supports the function **block2ss** to obtain the numerical state space realization. Although the function **bloc2exp** gives the symbolic transfer function of the specified system, it does not support the symbolic state space realizations.

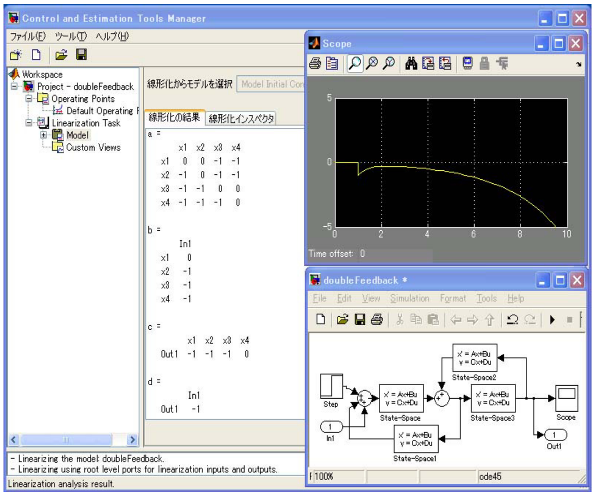

Fig. 9. Simulation (Simulink)

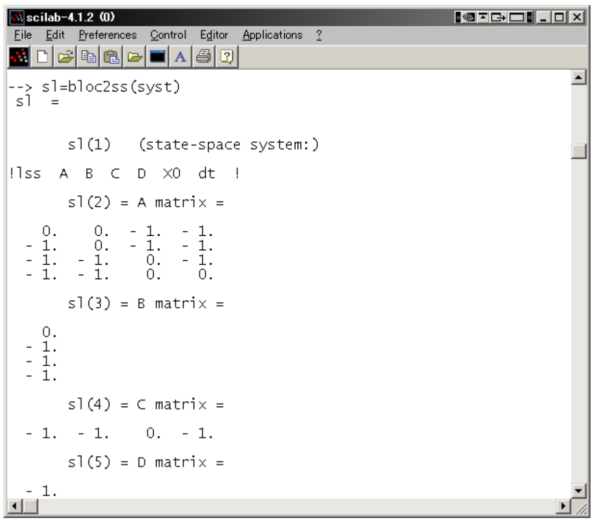

Fig. 10. Numerical State Space Realization (Scilab)

C. Scicos

Scicos provides an interactive graphical environment where a block diagram is created by dragging and drop blocks from the block library to the canvas and connecting them. In order to obtain the state space realization of the system in Scicos, we have to save the model to a file and invoke the function **lincos** in Scilab [6]. When we try to do the simulation of the systems with algebraic loops, the error message "**Algebraic Loop**" is displayed as shown in Fig.11.

D. 20-Sim

20-Sim deals only with SISO systems. It is impossible to do simulation of the system with coupled feedbacks. Figure 12 shows the symbolic state space realization of the system with single feedback.

VI. SUMMARY

This paper proposed a method which gives numerical and symbolic state space realizations of interconnected proper linear systems and guarantees that the order of the system is preserved. The method uses an adjacency matrix in graph theory for the modeling of the control systems.

We have developed a new software platform for modeling and simulation of control systems using the proposed method. If all the blocks are proper linear systems, we can obtain a state space realization of the whole system not only numerically but also symbolically. It is possible to solve in a unified way a variety of control system problems and get the exact solutions by applying the symbolic computation software such as Mathematica and Mapleto the obtained symbolic state space realizations.

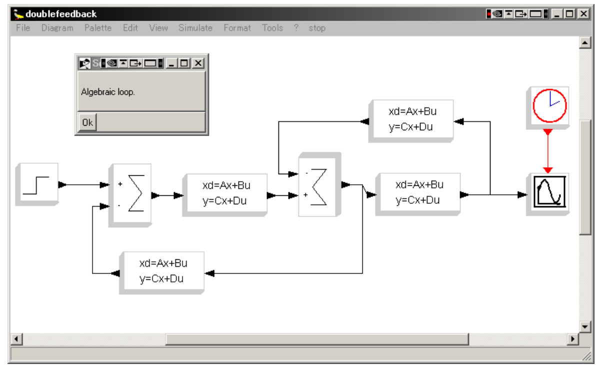

Fig. 11. Simulation (Scicos)

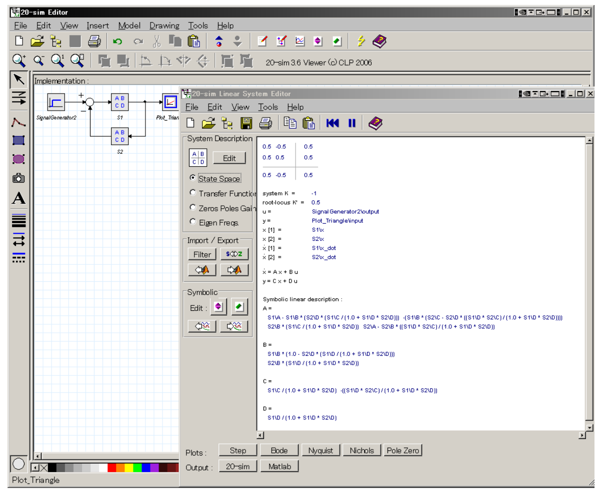

Fig. 12. Symbolic State Space Realization (20-Sim)

REFERENCES

[1] D. W. C. Ho, J. Lam, S. K. Tin, and C. Y. Han. Symbolic manipulation of transfer functions and state space realizations. *IEEE transaction on education*, 39(2), 1996.

[2] T. Umeno, S. Yamashita, O. Saito, and K. Abe. Symbolic computation application for the design of linear multivariable control systems. *J. Symbolic Computation*, 8:581–588, 1989.

[3] Controllab Products B.V. *20-sim*, 2007.

[4] Inc. The Math Works. *Matlab*, 2007.

[5] INRIA. *Scilab*, 2007.

[6] Stephen L. Campbell, Jean-Philippe Chancelier, and Ramine Nikoukhah. *Modeling and Simulation in Scilab/Scicos*. Springer, 2005.