Institute of Advanced Engineering and Science

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 2, No. 2, June 2013, pp. 59~72
ISSN: 2252-8938

Designing Stable intelligent Controller for Generalized Flow Shop Systems

Reza Ghasemi*, Sdjad Asad Beigi*, Mohammad Bagher Menhaj**
* Departement of Electrical Engineering, Damavand Branch, Islamic Azad Universdity
** Departement of Electrical Engineering, Amir Kabir University Technology

Article Info

Article history:

Received Aug 28,2012
Revised Dec 05, 2012
Accepted Jan 12,2013

Keyword:

Stability
Fuzzy System
Discrete Event System
Flow Shop System
Robustness

ABSTRACT

Designing a stable fuzzy controller for a class of generalized flow shop systems (GFSS) is addressed in this paper. An appropriate modeling of GFSS is proposed based on max-plus algebra. A multi-input single-output controller is discussed to stabilize the closed loop system. The merits of the proposed controller are robustness against uncertainties in the service times; stabilizing the closed loop system and withholding the blocking effect. An illustrative example is given to show the effectiveness of the proposed method.

Corresponding Author:

Reza Ghasemi,
Departement of Electrical Engineering,
Damavand Branch, Islamic Azad University,
Beheshti Road, Moalem Squar, Damavand, Iran.
Email: r_ghasemi@aut.ac.ir

1. INTRODUCTION

Flow shop production lines are very common in manufacturing systems such as car assemblies, manufacturing of electronic circuits, etc [1].

Based on fuzzy differential equation, [2] presented Timed Petri net where a fuzzy number is associated with each transition. The performance analysis considered in that paper is based on reachable graph where firing speed is fuzzy constant for each transition. Because of human's experts and uncertainties, [3] has extended crisp discrete event systems by introducing partial memberships to both states and event transitions. In addition, a method for considering the observability of discrete event system is presented.

A new formal way to solve the knowledge learning problem in expert systems is proposed in [4]. The generalized fuzzy Petri net in this paper is referred to as adaptive fuzzy timed Petri net because the system can learn. Modeling of discrete event systems using state vectors and event transition matrices in which their elements can be fuzzy numbers is presented in [5]. An optimal control is designed for this fuzzy model.

In [6] fuzzy discrete-event systems based on fuzzy language has been introduced in order to present uncertainty, imprecision, and vagueness of the model. This is concerned with the supervisory control for fuzzy discrete-event systems with partial observations. It is proven that there exist local fuzzy supervisors if and only if the proposed fuzzy language is both controllable and observable.

[7] presents fuzzy discrete-event systems as a generalization of conventional discrete event systems. Supervisory-control theory based on event feedback and state-based controller has been developed for proposed model. [8] Presents comprehensive approach to decision making based on fuzzy discrete event

systems. Optimal controller is designed for this model. The proposed method is used to HIV/AIDS treatment since AIDS is one of the most complex diseases to treat.

[9] has presented the application of discrete event system techniques to delay fault modeling and analysis in integrated circuit. The perturbation analysis and automation of the Discrete Event System and prediction of trajectory of new system has been developed in [10].

Supervisory control theory for a class of fuzzy discrete event systems, as a generalization of crisp discrete event systems have been introduced recently in order to represent possibility arising from the states and dynamics of a system in [11]. [12] considered the problem of supervisory control for a class of discrete event systems modeled by fuzzy automata. To use the fuzzy discrete event systems, fuzzy states and events have been introduced to describe uncertainties that occur often in practical problems, such as fault diagnosis applications in [13].

This paper is organized as follows. Section 2 gives a brief description about modeling of flow shop systems using max plus algebra. A new approach to design stable fuzzy controllers for generalized flow shop systems is proposed in section 3. Section 4 shows simulation results of the proposed controller and Section 5 concludes the paper.

2. MODELING OF GFSSWITH MAX-PLUS ALGEBRA

In this section modeling of generalized N-machine flow shop systemsis presented. Before proceeding further, the following definitions are needed.

Flow Shop System (FSS): a system that contains a single line of machines in which all jobs share the same processing order on these machines, each job visits all the machines, and each job may visit each machine at most once. For such systems, the production flow routing of jobs are identical.

Generalized Flow Shop System (GFSS): GFSS is a flow shop system in which a job does not have to visit all machines in the system. This means that certain jobs may skip certain machines. It is important to note that the production flow routing of jobs are not identical in this case although all jobs have the same processing order.

Interconnection Matrix (IM): IM is a matrix transformation over the max-plus algebra. It is an $r \times s$ matrix that operates on a s vector over $R \bigcup \{\infty\}$ to produce a r vector over $R \bigcup \{\infty\}$. The input vector to an IM is either the external input-time vector or a machine completion-time vector. The result of the transformation is either an input-time vector of a machine or an external output time vector of a machine.

Open-connection dynamics: It is the dynamics of any given machine in the system when considered working completely independent of any other machine in the system.

Max-plus algebra: The max-plus algebra is based on two operations $\oplus$ and $\circ$ defined over $\in R \cup \{-\infty\}$ where R is the set of real numbers. For scalars a and $b \in R \cup \{-\infty\}$, $\oplus$ and $\circ$ are defined as: $a \oplus b = \max(a,b)$ *and* $a \circ b = a + b$.

Furthermore, $\varepsilon = -\infty$ is a null element for $\oplus$ and e = 0 is the identity element for $\circ$.

Last Column matrix: an matrix is said to be in last column (LC) form if each of its first m-1 column is equal to $[\varepsilon]_{1\times n}$. If M is an $\mathrm{n} \times \mathrm{n}$ square matrix then it is called a square last column (SLC) matrix of size n.

Canonical form: A square matrix M is defined to be in canonical form if the blocks above its diagonal are equal to $[\varepsilon]_{q_1 \times q_2}$ for some integers q_1, q_2 , the blocks on the diagonal are SLC matrices, and the blocks below the diagonal are in last column forms.

For the system considered in this paper the following assumption are made.

The machines are numbered from left to right sequentially in an ascending order.

The inputs of the machine M_l , $l = 2,3,...,N$ are either the outputs from machine M_j where $j < l$ and/or the external inputs.

We are now ready to derive the dynamics of the generalized N-machine FSS shown in Figure 1. In particular, the objective is to derive the dynamics of the overall system using IM and the open-connection dynamics of the machines.

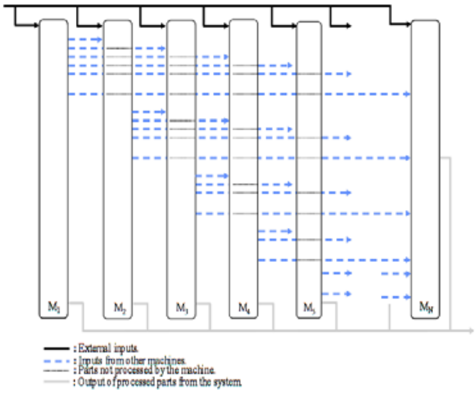

Figure 1. Generalized N-machine flow shop system

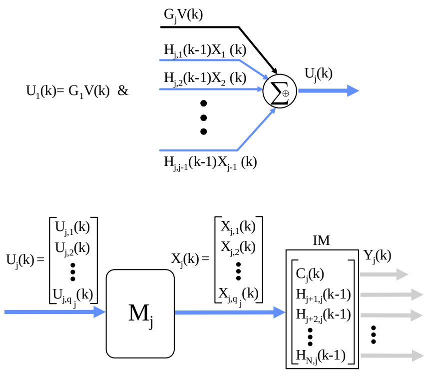

Figure 2. An equivalent representation of machine M_j of Figure 1 in the coupled connection mode

As shown in Figure 1 and Figure 2 the only inputs to the first machine are the external inputs. However, the inputs to machine M_l, l=1,2,…,N can come from any machine $M_j$where j<l and also may be external inputs. For the system of Figure 1 and Figure 2 the following notation are given.

k: A positive integer variable indicating the cycle index.

V(k): vector of external input-time that indicates the time at which the k^{th} external inputs are made available to the system.

G_j: An IM indicating the routing of external inputs to M_j, j=1, 2, …, N.

C_j (k): An IM indicating the external outputs of M_j, j=1, 2, …, N.

$H_{l,j}$ (k): An IM indicating the flow of parts from M_j to M_l, $j,l = 1,2,...,N \quad j < l$

q_j: Number of operations performed by M_j, j=1,2,…,N at each cycle.

$u_{j,r}$(k): Time at which the k^{th} open-connection input of type r is available to be processed by M_j in which j = 1, 2, …, N and r = 1, 2, …, q_j

U_j(k): A vector of open-connection input-time that indicates the time at which the k^{th} open-connection inputs are available for M_j , $j = 1,2,...,N$. It is defined as

$$U_j(k) = \left[u_{j,1}(k)\, u_{j,2}(k) ... u_{j,q_j}(k) \right]^T$$

$x_{j,r}$(k): Time at which machine M_j , $j = 1,2,...,N$ completes the processing of a part corresponding to $u_{j,r}(k)$, $r = 1,2,...,q_j$.

X_j(k): k^{th} vector of completion-time corresponding to M_j , $j = 1,2,...,N$ It is defined as

$$X_j(k) = \left[x_{j,1}(k) \quad x_{j,2}(k) \quad ... \quad x_{j,q_j}(k) \right]^T$$

Y_j(k): k^{th} external output-time vector corresponding to M_j , $j = 1,2,...,N$ .It is defined as

$Y_j(k) = C_j(k)X_j(k)$

$S_{j,r}(k)$: Service time required by machine M_j, $j = 1,2,...,N$ to process a part corresponding to $x_{j,r}(k)$ for $r = 1,2,...,q_j$.

It should be noted that how the interconnection matrices $G_j, H_{l,j}(k-1)$ and $C_j(k)$ are used to describe the routing of parts. Having stated the above definitions one can now proceed with the following major steps as a procedure for deriving the dynamics of the generalized N-machine FSS as shown in Figure 1 and Figure 2.

Obtain the IMs of the system.

Derive the open-connection dynamics of each machine

Using the IMs, obtain input-times of the machines in the coupled-connection mode.

Substitute the input-times of the machines obtained in step 3 into the open-connection dynamic equations of the machines obtained in step 2.

Construct the matrix form of the equations obtained in step 4. The result will be in the following form:

$$X(k+1) = \phi_s(k)X(k+1) \oplus \tilde{A}(k)X(k) \oplus \tilde{B}(k)V(k+1)$$

Do a recursion on the equation obtained in step 5 for N-1 times to eliminate X(k+1) from its right had side.

The above steps are carried out below. Just note that the IMs of the system (step 1) are shown in Figure 2. As stated before the open connection dynamics of each machine (step 2) is written as:

$$X_j(k+1) = A_j(k)X_j(k) \oplus B_j(k)U_j(k+1) \tag{1}$$

Where $A_j(k), B_j(k)$ have the following forms:

$$A_j(k) = \begin{bmatrix} \varepsilon & ... & \varepsilon & S_{j1}(k) \\ \varepsilon & ... & \varepsilon & S_{j1}(k)S_{j2}(k) \\ : & : & : & : \\ \varepsilon & ... & \varepsilon & S_{j1}(k)S_{j2}(k)...S_{jr}(k) \end{bmatrix} \tag{2}$$

$$B_j(k) = \begin{bmatrix} S_{j1}(.) & \varepsilon & ... & ... & \varepsilon \\ S_{j1}(.)S_{j2}(k) & S_{j2}(k) & \varepsilon & ... & \varepsilon \\ S_{j1}(k)S_{j2}(k)S_{j3}(k) & S_{j1}(k)S_{j2}(k) & S_{j3}(k) & ... & \varepsilon \\ : & : & : & : & \varepsilon \\ S_{j1}(k)S_{j2}(k)S_{jr}(k) & S_{j1}(k)S_{jr}(k) & ... & ... & S_{jr}(k) \end{bmatrix} \tag{3}$$

where $A_j(k)$ is in form of SLC.

From Figure 2 one can see that in the coupled-connection mode the input times of the machines (step 3) are given by

$$U_j(k+1) = G_jV(k+1) \oplus \sum_{l=1}^{j-1} \oplus H_{j,l}(k)X_l(k+1) \qquad j = 2,...N \tag{4}$$

The input times of the first machine (j=1) are the external input-time.

Substitution of (4) into (1) for j= 2,…N, yields (step 4)

$$\begin{aligned} X_j(k+1) = A_j(k)X_j(k) \oplus B_j(k)G_jV(k+1) \oplus \\ B_j(k)\sum_{l=1}^{j-1} \oplus H_{j,l}(k)X_l(k+1) \qquad j = 2,...,N \end{aligned} \tag{5}$$

Let us now define

$$X(k) = \left[X_1^T(k) \quad X_2^T(k) \quad \ldots \quad X_N^T(k) \right] \tag{6}$$

And

$$\phi_s(k) = \begin{bmatrix} [\varepsilon]_{q_1 \times q_1} & \ldots & \ldots & \ldots & [\varepsilon]_{q_1 \times q_N} \\ B_2(k)H_{2,1}(k) & \ldots & \ldots & \ldots & : \\ B_3(k)H_{3,1}(k) & B_3(k)H_{3,2}(k) & \ldots & \ldots & : \\ : & : & : & : & : \\ B_N(k)H_{N,1}(k) & B_N(k)H_{N,2}(k) & \ldots & B_N(k)H_{N,N-1}(k) & [\varepsilon]_{q_N \times q_N} \end{bmatrix} \tag{7}$$

Then we have

$$\tilde{A}(k) = \begin{bmatrix} A_1(k) & [\varepsilon]_{q_1 \times q_2} & \ldots & \ldots & [\varepsilon]_{q_1 \times q_N} \\ [\varepsilon]_{q_2 \times q_1} & A_2(k) & \ldots & \ldots & : \\ : & : & \ldots & \ldots & : \\ : & : & : & : & [\varepsilon]_{q_{N-1} \times q_N} \\ [\varepsilon]_{q_N \times q_1} & \ldots & \ldots & [\varepsilon]_{q_N \times q_{N-1}} & A_N(k) \end{bmatrix} \tag{8}$$

And

$$\tilde{B}(k) = \begin{bmatrix} B_1(k)G_1 \\ B_2(k)G_2 \\ : \\ B_N(k)G_N \end{bmatrix} \tag{9}$$

It can be shown that the matrix form of 1-5 (step 5) is given by

$$X(k+1) = \phi_s(k)X(k+1) \oplus \tilde{A}(k)X(k) \oplus \tilde{B}(k)V(k+1) \tag{10}$$

Note however that X(k + 1) not only appears on the right-hand side of (10) but it also appears on the left-hand side as well. To eliminate X(k + 1) from the right-hand side of (10) one needs to do a recursion on X(k + 1) for N-1 times (step 6) to obtain the following:

$$X(k+1) = \phi_s^N(k)X(k+1) \oplus \phi_s^*(k)\tilde{A}(k)X(k) \oplus \phi_s^*(k)\tilde{B}(k)V(k+1) \tag{11}$$

Where $\phi_s^*(k)$ is defined as

$$\phi_s^*(k) = \phi_s^{N-1}(k) \oplus \phi_s^{N-1}(k) \oplus \ldots \phi_s(k) \oplus E \tag{12}$$

It is shown in [1] that $\phi_s^N(k) = [\varepsilon]_{(q_1+q_2+\ldots+q_N)\times(q_1+q_2+\ldots+q_N)}$.

If we now define A(k) and B(k) as

$$A(k) = \phi_s^*(k)\tilde{A} \tag{13}$$

and

$$B(k) = \phi_s^*(k)\tilde{B} \tag{14}$$

then (11) simplifies to

$$X(k+1) = A(k)X(k) \oplus B(k)V(k+1) \tag{15}$$

It is now shown that A(k) is in canonical form. It is shown in [1] that $\phi_s^*(k)$ has the following structure.

$$\phi_s^*(k) = \begin{bmatrix} E & [\varepsilon]_{q_1 \times q_2} & & [\varepsilon]_{q_1 \times q_N} \\ & E & : & : \\ & & \cdots & : \\ & [.] & E & [\varepsilon]_{q_{N-1} \times q_N} \\ & & & E \end{bmatrix} \tag{16}$$

where E is the identity matrix defined over max-plus algebra.
Now, by using (7), (8), and (13), it can be shown that A(k) is given by

$$A(k) = \begin{bmatrix} A_1(k) & [\varepsilon]_{q_1 \times q_2} & \cdots & \cdots & \cdots & [\varepsilon]_{q_1 \times q_N} \\ [.]A_1(k) & A_2(k) & : & & & : \\ [.]A_1(k) & [.]A_2(k) & : & & & : \\ : & : & : & & & : \\ : & : & : & & & [\varepsilon]_{q_{N-1} \times q_N} \\ [.]A_1(k) & [.]A_2(k) & \cdots & \cdots & [.]A_{N-1}(k) & A_N(k) \end{bmatrix} \tag{17}$$

In (17) represents a matrix in which its elements are not significant for the result that we are after. It is important to note that since $A_j(k)$ is a SLC matrix for j = 1, 2, …, N then by a proposition given in [1], $[.]A_j(k)$ is in last column form for j = 1, 2, … , N. Furthermore, since the diagonal blocks of A(k) are SLC matrices and the blocks above the diagonal are equal to $[\varepsilon]_{(.)\times(.)}$, A(k) is in canonical form.

The system output-time vector defined as

$$Y(k) = \left[Y_1^T(k) \quad Y_2^T(k) \quad \cdots \quad Y_N^T(k)\right]^T \tag{18}$$

is obtained using X(k) given in equation (15) by

$$Y(k) = C(k)X(k) \tag{19}$$

Where

$$C(k) = \begin{bmatrix} C_1(k) & [\varepsilon]_{q_1 \times q_2} & & [\varepsilon]_{q_1 \times q_N} \\ & C_2(k) & : & : \\ & & \cdots & : \\ & [.] & C_{N-1}(k) & [\varepsilon]_{q_{N-1} \times q_N} \\ & & & C_N(k) \end{bmatrix} \tag{20}$$

It should be noted that the completion-time vector X(k) in (15) will grow without bound. This makes us obtain an alternate model in terms of boundedness of certain delay of parts. Before proceeding further, we define $\Gamma(k) = X(k) - U(k)$, $\Delta V(k) = V(k+1) - V(k)$ and use (15) to write the Γ state model as:

$$\Gamma(k+1) = \widehat{A}(k)\Gamma(k) \oplus \widehat{B}(k)\Delta V(k) \tag{21}$$

$\widehat{A}(k), \widehat{B}(k)$ are given below.

$$\begin{aligned} A(k) &= A(k) - \tilde{U}(k+1)\underbrace{\begin{bmatrix}0 & 0 & \dots & 0\end{bmatrix}}_{n} + \left(\tilde{U}(k)\underbrace{\begin{bmatrix}0 & 0 & \dots & 0\end{bmatrix}}_{n}\right)^T \\ \widehat{B}(k) &= B(k) + \left(V(k)\underbrace{\begin{bmatrix}0 & 0 & \dots & 0\end{bmatrix}}_{n}\right)^T - \tilde{U}(k+1)\underbrace{\begin{bmatrix}0 & 0 & \dots & 0\end{bmatrix}}_{m} \end{aligned} \tag{22}$$

Definition 1: the Γ state model given by (21) is Γ stable or $\Delta V(k)$ belongs to Γ stable set if the entire $\Gamma(k)$ are bounded.

Theorem1: Consider the GFSS given by

$$X(k+1) = A(k)X(k) \oplus B(k)V(k+1) \tag{23}$$

where $A(k)$ is in canonical form for all k and V (k + 1) is a p-vector. Now, suppose that the corresponding Γ -state model is given by:

$$\Gamma(k+1) = \widehat{A}(k)\Gamma(k) \oplus \widehat{B}(k)\Delta V(k) \tag{24}$$

where $\widehat{A}(k)$ is in canonical form for all k. Necessary conditions for $\Delta V(k)$ to belong to the Γ -stability set are:

1. The entries of $\widehat{A}(k)$ and $\widehat{B}(k)\Delta V(k)$ must be bounded above as functions of k.

2. The Bottom Right-Corner-Element (BRCE) of each block on the diagonal of $\widehat{A}(k)$ must be less than or equal to zero for at least one value of k.

Assuming that the necessary conditions are satisfied, a sufficient condition for $\Delta V(k)$ to belong to the Γ -stability set is stated as: the BRCE of every block on the diagonal of $\widehat{A}(k)$ must be less than or equal to zero for all k.

The proof of above stability theorem is given in [1].

It is clear that the complexity of the modeling is only for first time. Without any complexity, we can use this algorithm simply without any

3. FUZZY SYSTEM

Figure 3 shows the basic configuration of the fuzzy system considered in this paper. Here, we consider a multi-input, single-output fuzzy system: $x \in U \subset R \to y \in V \subset R$. Consider that a multi-output system can be separated into a group of single-output systems.

The fuzzifier performs a mapping from a crisp input vector to a fuzzy set, where the label of the fuzzy set are such as "small", "medium", "large", etc.

The fuzzy rule base consists of a collection of fuzzy IF-THEN rules. Assume that there are M rules, and the l^{th} rule is

$R^l(u):\ if\ (x_1\ is\ A_1^l\ \dots\ x_n\ is\ A_1^l) then\ (y\ is\ B^l)\ \ l = 1,2,\dots,M$

where $x = [x_1, x_2, \dots, x_n]^T$ and y are the crisp input and output of the fuzzy system, respectively. A_j^l and B^l are fuzzy membership function in U_j and V, respectively.

The fuzzy inference performs a mapping from fuzzy sets in U to fuzzy sets in V, based on the fuzzy IF-THEN rules in the fuzzy rule base.

The defuzzifier maps fuzzy sets in V to a crisp value in V. The configuration of Figure 3 represents a general framework of fuzzy systems, because many different choices are allowed for each block in Figure 3, and various combinations of these choices will construct different fuzzy systems [14]. Here we use the sum-product inference and the center-average defuzzifier. So the fuzzy system can be expressed as

$$y(x) = F(x) = \frac{\sum_{l=1}^{M} y^l \prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)} \quad (25)$$

where y(x) is the crisp output of fuzzy system, $\mu_{A_i^l}(x_i)$ is the membership degree of the input x_i to fuzzy set A_j^l and y^l is the point at which the membership function of fuzzy set B^l achieves its maximum value.

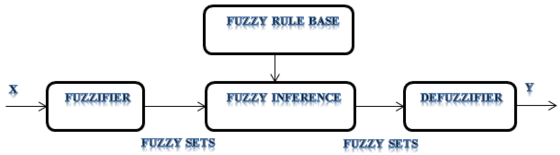

Figure 3. Configuration of fuzzy system.

The fuzzy system in the form of (25) is proven in [15] to be a universal approximator if its parameters are properly chosen.

4. STABLE FUZZY CONTROLLER FOR DISCRETE EVENT SYSTEM

This section presents a new method for designing a stable fuzzy controller for discrete event systems when the system has uncertainties in its service times. Figure (4) presents the closed loop system with the proposed fuzzy controller.

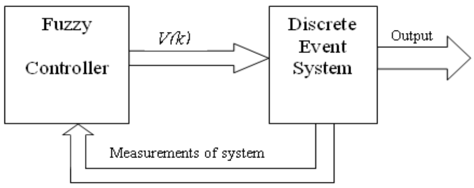

Figure 4. Closed loop system with fuzzy controller

Consider the Γ state model given in (24). Assume that the service times defined in (10) are bounded.

$$s_i(k) = s_i + \Delta s_i(k) \;\; i = 1, 2, ..., N \quad (26)$$

where s_i is a nominal service time and Δs_i presents the uncertainty in the service time which is bounded as:

$$|\Delta s_i| < m_i \quad (27)$$

In the above equation m_i is an unknown constant.

Now, for the system given by (24) a controller of the following form is proposed.

$$V(k+1) = V(k) + C + F(X(k), DP, Buffer) \tag{28}$$

where C is constant, $F(X(k), DP, Buffer)$ is the fuzzy controller and *Buffer* presents number of elements in the buffer. By substituting (26) and (28) into (24), (29) is obtained.

$$\Gamma(k+1) = \left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, F)\right)\Gamma(k) \oplus \left(\tilde{B} + \Delta\tilde{B}(k, \Delta s, F)\right)\left(C + F(X(k), DP, Buffer)\right) \tag{29}$$

where $\tilde{A}$ and $\tilde{B}$ are constant matrices, $\Delta\tilde{A}(k, \Delta s, F)$ and $\Delta\tilde{B}(k, \Delta s, F)$ are uncertainty matrices coming from the fuzzy controller and the uncertainties in service times.

The proposed fuzzy controller is presented as figure (5).

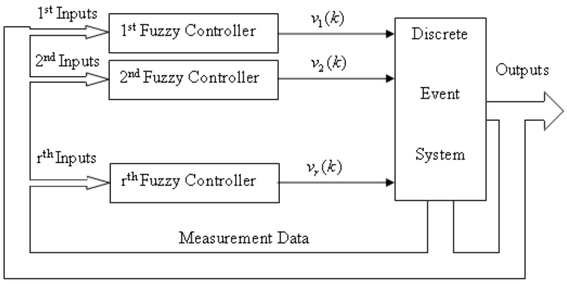

Figure 5. Proposed structure of fuzzy controller

The max plus algebra modeling for flow shop systems causes the controller to be separated in some local controllers. Finally, the closed loop of the whole system must be stable.

Theorem 2: The system modeled by (29) with the fuzzy controller given in equation (28) is Γ stable if and only if the fuzzy controller is BIBO and its upper bound, (C) and the uncertainties of the service times (Δs) satisfy the following:

BRCE of each block on diagonal of $\left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, C, f)\right)$ must be equal or lower than zero.

Proof:

"$\Rightarrow$": Based on Theorem 1, (29) is stable if $\left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, C, f)\right)$ and $\left(\tilde{B} + \Delta\tilde{B}(k, \Delta s, C, f)\right)$ are bounded functions of k and

$BRCE(each\ \ block\ \ on\ \ diagonal\ \ of\ \ \left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, C, f)\right)) \le 0$ for all k.

Now using equations (26) and (27) one can write the following

$$\Delta\tilde{A}(k) < A_{cte} \tag{30}$$

where A_{cte} is a constant matrix. Consequently $\left(\tilde{A} + \Delta\tilde{A}(k)\right)$ is bounded. Using similar reasoning one can see that $\left(\tilde{B} + \Delta\tilde{B}(k, \Delta s, C, f)\right)$ is bounded too.

Now consider a fuzzy system such that its l^{th} rule is defined as follows:

$R^l(u):\ if\ (x_1\ is\ A_1^l\ \ldots\ x_n\ is\ A_1^l) then\ (y\ is\ B^l)\ \ l = 1, 2, \ldots, M$

The proposed controller is considered to be the output of the fuzzy system with the above rule base form. Now, assume that the fuzzy system has the following properties as mentioned in previous section: product inference engine, singleton fuzzification, center average defuzzification

Furthermore, assume that $0 \le \mu_{A_i^l}(x_i) \le 1$ is satisfied and y^l is bounded.

Using the above assumptions $\left(C + F(X(k), DP, Buffer)\right)$ will be bounded and as a result $\left(\tilde{B} + \Delta\tilde{B}(k, \Delta s, C, f)\right)\left(C + F(X(k), DP, Buffer)\right)$ will be bounded function of k.

Now suppose f is an upper bound for $F(X(k), DP, Buffer)$ that is $F(X(k), DP, Buffer) < f$ where f is constant. If we now select *f, C,* and $s_i + m_i$ such that *BRCE of each block on diagonal of* $\left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, C, f)\right)$ *is equal or lower than zero* is satisfied, then the close loop system will be stable.

Proof of " $\Leftarrow$ " part

If fuzzy controller is BIBO and c is a constant, then it can be shown that the controller input is bounded for all k. Because of the boundedness in the fuzzy controller output and Δs, boundedness of $\left(\tilde{B} + \Delta\tilde{B}(k, \Delta s, C, f)\right)$ and $\left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, C, f)\right)$ are guaranteed. Now using the result of Theorem (1) one can see that if we satisfy

$$BRCE(each\quad block\quad on\quad diagonal\quad of\quad \left(\tilde{A} + \Delta\tilde{A}(k, \Delta s, C, f)\right)) \le 0$$

Then the closed loop system will be stable.

5. ALGORITHM OF IMPLEMENTATION

This paper presents a new method for modeling and designing stable controller. To clarify the steps of modeling and designing controller, the following algorithm is presented.

Step 1: construct the model of the system based on equation (1) and the parameters of the proposed model is as equation (2, 3).

Step 2: based on equation (22) and theorem 1, the proposed model in first step can be converted to Γ state model as shown in equation (21).

Step 3: the controller as shown in equation (28) can stabilize the perturbed model presented in the step 2.

Note: the parameter of the proposed controller should satisfy the theorem 2.

6. SIMULATION RESULTS

We will give an illustrative example to show the capability of the proposed fuzzy controller to stabilize a generalized flow shop system. Consider the generalized three-machine flow shop system shown in figure 6.The following example is a part of an assembling line of an Electrical factory in which M_1, M_2, M_3 are machines to assemble some electronic parts in some input chaises.

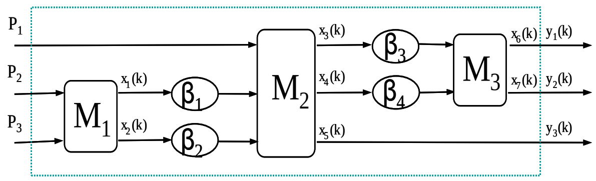

Figure 6. Generalized three-machine flow shop system

In the above, β_i refers to the i^{th} buffer capacity. The inputs of the system are the external parts (P_1, P_2, P_3). It is assumed that the process is cyclic. We further assume that all buffers have a capacity of size $\beta = 3$ that is at no time any given buffer can have more than β parts in it. Consequently, blockings could occur in the system if the inputs to the machines are not properly controlled. It is shown that the proposed fuzzy controller not only will prevent the blocking effect but also it will improve the throughput of the system compared to what is given in [1].

In the following equations, $x_j(k)$ represents the k^{th} completion time of activity of type j, $v_{i,k}$ is the k^{th} availability time of P_i and S_j represents the amount of time required for the part of type j to be processed at cycle k .

To formulate the Γ model, we denote the input time vector as $V(k)=\begin{bmatrix} v_1(k) & v_2(k) & v_3(k) \end{bmatrix}^T$, the completion time vector as

$$X(k)=\begin{bmatrix} X_1(k) & X_2(k) & X_3(k) \end{bmatrix}^T=\left[x_1(k) \quad x_2(k) \,\middle|\, x_3(k) \quad x_4(k) \quad x_5(k) \,\middle|\, x_6(k) \quad x_7(k) \right]^T$$

and the Γ -vector defined in (24) is rewritten as

$$\Gamma(k)=X(k)-\begin{bmatrix} v_2(k) & v_3(k) & v_1(k) & v_2(k) & v_3(k) & v_1(k) & v_2(k) \end{bmatrix}$$

Now, we can write the Γ model as follows:

$$\Gamma(k+1)=\tilde{A}(k)\Gamma(k)\oplus\tilde{B}(k)\Delta V(k) \tag{31}$$

where the matrices $\tilde{A}(k)$ and $\tilde{B}(k)$ are given by

$$\tilde{A}(k)=\begin{bmatrix} \tilde{A}_{1,1}(k) & [\varepsilon]_{2\times 3} & [\varepsilon]_{3\times 3} \\ \tilde{A}_{2,1}(k) & \tilde{A}_{2,2}(k) & [\varepsilon]_{3\times 2} \\ \tilde{A}_{3,1}(k) & \tilde{A}_{3,2}(k) & \tilde{A}_{3,3}(k) \end{bmatrix} \quad and \quad \tilde{B}(k)=\begin{bmatrix} \tilde{b}_1(k) & \tilde{b}_2(k) & \tilde{b}_3(k) \end{bmatrix} \tag{32}$$

The above matrices is function of both Input time vector and service times.We further assume that the service time uncertainties are given by

$$s_i(k)=s_i+\Delta s_i \quad i=1,...,7 \tag{33}$$

where Δs_i is the uncertainty in the ith service time and it is assumed that this uncertainty is bounded by $|\Delta s|<m_i$.

As discussed in the previous section, the entries of $\Gamma(k)$ are bounded. Knowing that the inputs of the proposed controller are the external input times such as $v_1(k)$, $v_2(k)$ and $v_3(k)$, the controller may have the following form

$$v_i(k+1)=v_i(k)+c+f_i(X(k),DP,Buffer) \tag{34}$$

In (34), $f_i(X(k),DP,Buffer)$ represents the fuzzy controller in which DP is the external part delay, *Buffer* presents number of the elements in the buffer and c is a constant which is properly calculated below.

For the flow shop system, we have three fuzzy rule bases. This fuzzy system is aimed to compensate for the uncertainty in system and stabilize the closed loop system.

The first fuzzy rule base that controls the first input $(v_1(k))$ has six inputs such as delay of part 1 (*DP1*), delay of part 2 (*DP2*), delay of part 3 (*DP3*), number of elements in buffer 1 (*buffer 1*), number of elements in buffer 2 (*buffer 2*), and difference of earliest times that machine 7 completes processing over the seventh activity, ($x_7(k+1)-x_7(k)$). It is named *deltax 7* in the rule base.

All of the membership functions in all rule bases are gaussian. The sample of the rule base related to the first input, $v_1(k)$ is shown in table 1.

Table 1. Rule base related to the first input

If (DP1 is low) and (DP2 is high)and (DP3 is high)and(deltax7 is high)then(f1 is low)

Controlling the second input $(v_2(k))$, we also need six inputs like those given above for the first input, but the input number six is replaced with the difference of earliest times that this machine finishes processing over the first activity, ($x_1(k+1)-x_1(k)$). It is named as *deltax1* in rule base. The sample of this rule base is shown in table 2.

Table 2. Rule base of the second input

if (DP1 is low)and (DP2 is low)and(DP3 is low)then (f2 is zero)

The fuzzy rule base of the third fuzzy controller for $(v_2(k))$ has three inputs such as delay of part 3, number of elements in buffer 2, and difference of earliest times that the machine finishes processing on the third activity, ($x_3(k+1)-x_3(k)$). It is called *deltax3* in the rule base. Table 3 represents the sample of the corresponding rule base.

Table3: rule base of the third inpu

if (DP3 is low)and(deltax3 is high)then (f3 is low)

By substituting (33) and (34) into (24), the closed loop of the system can be written as:

$$\Gamma(k+1)=\left(\tilde{A}+\Delta\tilde{A}(k)\right)\Gamma(k)\oplus\left(\tilde{B}+\Delta\tilde{B}(k)\right)\left(C+F(X(k),DP,Buffer)\right) \tag{35}$$

As shown in theorem 1, we can properly establish constant matrices $\tilde{A}$, $\tilde{B}$ and C , and have matrices $\Delta\tilde{A}(k)$, $\Delta\tilde{B}(k)$ and $F(X(k),DP,Buffer)$ bounded denoted as:

$$\left|F(X(k),DP,Buffer)\right|<f \quad and \quad \Delta\tilde{A}(k)<A_c \quad and \quad \Delta\tilde{B}(k)<\tilde{B}_c \tag{36}$$

where A_c , B_c and f are constant matrices.

From the above, we can easily say that $\tilde{A}\Delta\tilde{A}(k)$ and $\tilde{B}\Delta\tilde{B}(k)CF(X(k),DP,Buffer)$ are upper bounded. If we prove that the BRCE of diagonal elements of the $\tilde{A}(k)$ is equal or less than zero for all k, then we can say the closed loop system with the fuzzy controller is Γ stable or $\Delta V(k)$ belongs to Γ stability set. This leads to the following inequalities.

$$\begin{aligned} & s_1+s_2+\Delta s_1+\Delta s_2-c-F_3\le 0 \quad , \quad s_6+s_7+\Delta s_6+\Delta s_7-c-F_2\le 0 \\ & s_3+s_4+s_5+\Delta s_3+\Delta s_4+\Delta s_5-c-F_3\le 0 \end{aligned} \tag{37}$$

We are now ready to find the value of C properly as stated below. The following inequalities are easily derived from (37) and (33).

$$\begin{aligned} & s_1+s_2+m_1+m_2-f_3\le c \quad , \quad s_6+s_7+m_6+m_7-f_2\le c \\ & s_3+s_4+s_5+m_3+m_4+m_5-f_3\le c \end{aligned} \tag{38}$$

From (38), (39) can be easily derived.

$$c=\max\left(s_1+s_2+m_1+m_2-f_3, s_3+s_4+s_5+m_3+m_4+m_5-f_3, s_6+s_7+m_6+m_7-f_2\right) \tag{39}$$

This value of C will guarantee the closed stability (see theorem 2). For the purpose of simulation, it is assumed that $s_1=6, s_2=3, s_3=4$ $s_4=6, s_5=2, s_6=4, s_7=5$ and $m_i=2,\ i=1,2,...,7$ and initial values of all buffers are zero.

The MATLAB software has been used to implement the closed loop system. The simulation results are shown in figure 7. As observed, the number of elements in buffers will not exceed beta which is a pre-specified number chosen by user.

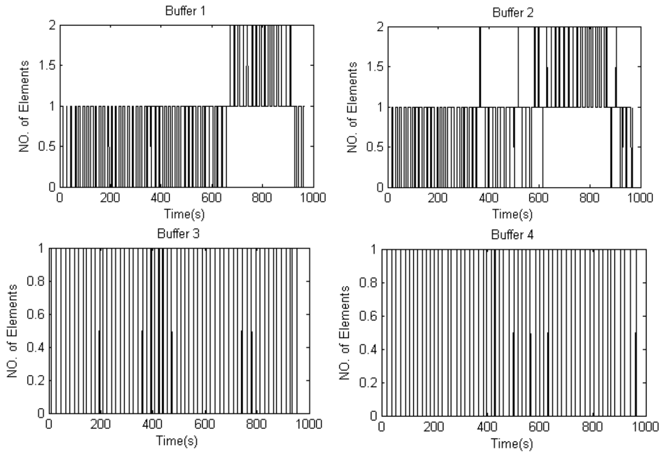

Figure 7. Number of element in all buffers

To compare the performance of the proposed method with the performance of the approach given in [1], we list the average delay of parts and the average throughput times in table (4) and (5).

Table 4. average throughput time over 60 performances

$\Delta x_i(k)$	Average throughput type	
	Approach[1]	Our Approach
i=1	16.76	16.0114
i=2	16.76	16.0809
i=3	17.01	16.3700
i=4	16.88	16.3429
i=5	16.89	16.3412
i=6	17.00	16.3666
i=7	16.86	16.3243

Table 5. Average delay of parts over 60 performances

Ppart type	Average delay of part types	
	Approach[1]	Our Approach
1	20.35	13.83
2	30.13	22.93
3	25.79	20.33

From these tables, it is obvious that our approach remarkably outperforms the approach given in [1] by showing 1) more robustness against uncertainties, 2) more reduction in the delay of part types and throughput times, and 3) ensuring stability of the closed loop system.

7. CONCLUSION

In this paper, we proposed a new method to model GFSS based on max plus algebra. The recommended intelligent controller can stabilize GFSS. The novelties of this paper are 1) modeling and control of GFSS with finite buffer capacities, 2) stabilization of the closed loop system, 3) robustness against uncertainties. Comparing to other paper, the performance of the presented model and controller are promising.

REFERENCES

[1] Ghasemi R, Doustmohammadi A, Menhaj MB. Stable fuzzy controller for discrete event systems. Control & Automation, MED'07 Mediterranean Conference Athens-Greece. 2007: 1-4.
[2] Ding Z. fuzzy timed Petri net – analysis and implementation. *Mathematical and Computer Modeling Journal.* 2006; 385-400.
[3] Li X, Rosana FL. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework. *IEEE Tran. On systems, Man, and Cybernetics – Part c: Application and Reviews*. 2000; 30(4): 100-104.
[4] Tang Y. Stable fuzzy adaptive control for class of nonlinear systems. *Fuzzy set and systems journal*. 1999: 279-288.
[5] Lin F, Ying H. Modeling and Control of Fuzzy Discrete Event Systems. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*. 2002; 32(4): 408-415.
[6] Cao Y, Ying M. Observability and Decentralized Control of Fuzzy Discrete-Event Systems. *IEEE Transactions on Fuzzy Systems*. 2006; 14(2): 202-216.
[7] Cao Y, Ying M, Chen G. State-Based Control of Fuzzy Discrete-Event Systems. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*. 2007; 37(2): 410-424.
[8] Lin F, ect. Decision making in fuzzy discrete event systems. *Information Sciences*. 2007; 177: 3749-3763.
[9] Westerman G, ect. *Discrete event system approach for delay fault analysis in digital circuits.* American Control Conference. IEEE, USA. 1998: 239-243.
[10] Cao X. the predictability of Discrete Event Systems. *IEEE Transactions on Automatic Control*. 1989; 34(11): 1168-1171.
[11] Li C, Cai Y, Ding F. *Product Supervisor for Fuzzy Discrete Event Systems*. IEEE computer society, Fifth International Conference on Fuzzy Systems and Knowledge Discovery. 2008: 391-395.
[12] Cao Y, Ying M. Supervisory Control of Fuzzy Discrete Event Systems. *IEEE Transactions on Systems, Man and Cybernetics—Part B*. 2005; 35(2): 366-371.
[13] Kilic E. Diagnosability of fuzzy discrete event systems. *Information Science*. 2008; 178: 858-870.
[14] Wang LX. A course in fuzzy systems and control.first edition. Prentice Hall PTR. 1997.
[15] Wang LX. Mendel JM. Fuzzy basis function, universal approximation and orthogonal least square learning. *IEEE Trans. Neural Network*. 1993; 3(5): 807-814.

BIOGRAPHIES OF AUTHORS

	**Reza Ghasemi** was born in Tehran, Iran in 1979. He received his B.Sc degrees in Electrical engineering from SemnanUniversity in 2000 and M.Sc. degrees and Ph.D. in control engineering from Amirkabir University of Technology, Tehran, Iran, in 2004 and 2009, respectively. His research interests include large-Scale Systems, Adaptive Control, Robust Control, Nonlinear Control, and Intelligent Systems. Dr. Reza Ghasemi joined the Department of Electrical Engineering, Islamic Azad University, Damavand Branch, Damavand, Iran, where he is currently an Assistant Professor of electrical engineering.
	**SadjadAsadBeigi** was born in Tehran, Iran in 1981. He received his B.Sc degrees in Bio-Electrical engineering from Science and Research Branch, Islamic Azad University in 2005 and M.Sc. degrees. in control engineering from Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2009. His interests are Bio-electric, control Engineering.
	**Mohammad B. Menhaj** received his Ph.D. degree from school of electrical and computer engineering at OSU in 1992. After completing one year with OSU as postdoctoral fellow, in 1993, he joined Amirkabir University of Technology, Tehran, Iran. December 2000 to Aug. 2003, he was with school of Electrical and Computer Engineering and Department of Computer Science at OSU as visiting faculty member and research scholar. He is author and co-author of more than 400 technical papers, and four books: Fundamentals of Neural Networks, 1998, Application of Computational Intelligence in Control, 1998, Neural Networks, 2000, and Fuzzy Computations, 2007, all in Persian. He has also been project director for many industrial projects in the areas such as crisis control management, communication traffic control, real time simulator design, flight control and navigation systems, and satellite attitude determination and control systems, sponsored by private and government institutions. His main research interests are: theory of computational intelligence, learning automata, adaptive filtering and their applications in control, power systems, image processing, pattern recognition, and communications, and other areas of interests are: theory of rough set and knowledge discovery.