Modelling flexible manufacturing systems using weighted Fuzzy Coloured Petri Nets[1]

Daniel S. Yeung, Simon C.K. Shiu and
Eric C.C. Tsang
Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Email: {csdaniel, cskshiu, csetsang} @comp.polyu.edu.hk

In this paper, a contribution was made in developing a Fuzzy Coloured Petri Nets (FCPNs) model for description and analysis of the dynamic behaviour and inexact production inference of Flexible Manufacturing Systems. The basic idea is to integrate the concepts of Coloured Petri Nets and Fuzzy Petri Nets, where the former is good at modelling complicated concurrent processes with embedded data structures, while the latter is good at modelling systems that involve approximate reasoning and uncertainty knowledge inference. Furthermore, a method of assigning a weight parameter to each proposition in the antecedent of a fuzzy production rule is developed which can be used to model the relative degree of importance of each proposition in the antecedent contributing to the consequent. A new weighted fuzzy production rule evaluation method is also devised. Our approach can be used to model and analyze Fuzzy Expert Systems that are used in controlling or supervising flexible manufacturing processes. Throughout this paper, a printed circuit board production system is used to demonstrate the model's capability in simulating the system behaviour with weighted fuzzy control rules.

1. Introduction

1.1. FMS and Petri Nets

According to [18], a flexible manufacturing system (FMS) is made up of hardware and software elements. Hardware elements are visible and tangible such as Computer Numerical Control (CNC) machine tools, pallet queuing carrousels, material handling equipment, central chip removal and coolant systems, tooling systems, and the like. Software elements include numerical control (NC) programs, traffic management software, tools control software, and the like. The main objective of FMS is to produce varieties of components or products within its stated capability and to a predetermined schedule. Such systems combine the advantages of very flexible but inefficient manual job shops with those highly productive but rigid transfer lines. Therefore, FMS is necessary to enable a manufacturer to compete efficiently in today's market.

In the context of using Petri Nets techniques for modelling FMS, it should be possible to be able to extend both the descriptive and analytical powers of the ordinary Petri Nets. At present, there are large numbers of researches in progress in this area. Nevertheless, they all share a common goal: to search for solutions to achieve higher speeds and more flexibility and thus increase manufacturing productivity. There are different techniques for modelling FMS, the most common one is based on techniques from operational research and mathematical programming. However, they cannot be satisfactorily used and even impossible to apply for a moderate complex system. Another promising approach is based on simulation. Petri Nets oriented simulator is one of the most popular simulation languages used for analyzing complicated FMS systems. The results from the simulation can be used to aid the study of optimal operational setting problems. In a typical FMS, there are some associated production strategies, many of which strategies are expressed in the form of fuzzy production rules, e.g. if the tool speed is high and the feed rate is small then the surface-quality is good. In order to have an appropriate modelling language for FMS of the given complexity, we provide an extension of Petri Nets to efficiently describe the concurrency, synchronization, and mutual exclusion of flexible manufacturing systems with fuzzy characterized behaviours.

---

[1] This project is supported by a research grant from the Hong Kong Polytechnic University. Grant No. 340/642.

Journal of Intelligent and Fuzzy Systems 7 (1999) 137–149
ISSN 1064-1246 / $8.00

1.2. Extensions of Petri Nets

Petri Nets have been widely used in various application domains such as expert system verification [15], VLSI chips, complex radar surveillance systems, communication protocol for digital telephone networks, flexible manufacturing systems [8, 21], control of electronic transfer of money and the like. The main objectives of these applications are for modelling and analysis of the asynchronous, discrete event dynamic systems behaviours.

Extensions of Petri Nets, such as High-Level Petri Nets (HLPNs) and Coloured Petri Nets (CPNs) [11] have been developed because ordinary Petri Nets are not always sufficient to represent and analyze complex system behaviours. The reasons for its success are: a graphical representation of the problem, well defined semantics, ability to describe a large variety of different systems, few but powerful primitives, explicit description of both states and actions, concurrent semantic, hierarchical description, integration and synchronization of control and data manipulation, stability towards minor system changes, interactive simulation, formal analysis methods and supporting computer tools.

Ordinary Petri Nets have a lot of limitations, the most problematic one is the inability of modelling incomplete, uncertain, and approximate information or states. As the popularity of fuzzy reasoning grows in certain kinds of manufacturing processes, it is necessary to extend Petri Nets to incorporate fuzzy logic and fuzzy set theories in such processes. Therefore, Fuzzy Petri Nets (FPNs), a model which is able to represent the fuzzy production rules of a rule-based system, is the ideal tool to aid such manufacturing system development [8]. It is noted that Fuzzy Petri Nets have been used in task sequencing in robotics system [2], knowledge acquisition and representation [28, 29], approximate reasoning and medical diagnosis. A summary of the applications of Fuzzy Petri Net can be obtained in [16].

However, until recently, there was no standardized definition of Fuzzy Petri Nets, therefore, in the present paper, we based on Yeung's Fuzzy Petri Nets [28, 29] concepts for production rule modeling and on Jensen's Coloured Petri Nets [11, 12] concepts for complex data structure modeling to formulate our extended model, Fuzzy Coloured Petri Nets (FCPNs).

The application problem we consider here is: if there is a complex system, such as Flexible Manufacturing Systems, with the following properties: concurrency, complex data structure and fuzzy rule-based production strategies that we need to model, it would be desirable to have a modeling tool which supports the abilities of both Fuzzy Petri Nets and Coloured Petri Nets.

In the next section, the definitions of FCPNs are described and an illustrative example is given. In section three, a flexible automatic printed circuit board component insertion assembly system [26] is described. Basically, the system can be divided into two subsystems that should be integrated together. One of the subsystems is the robot arms operation sequencing system that can be efficiently modeled by CPNs. Another subsystem is the fuzzy robotics wrist control system. It cannot be modeled by CPNs itself without the use of FPN model. Using FCPNs, the two subsystems can be nicely integrated into one system where the modeling, simulation and analysis can be performed effectively. In section four, a weighted fuzzy production rule is defined and explained. The evaluation method for this rule is also developed. Section five explains our enhanced fuzzy reasoning algorithm. The last section gives the summary and the future work.

2. Fuzzy Coloured Petri Nets

2.1. Definition

Fuzzy Petri Net described by D.S. Yeung and E.C.C. Tsang in [28, 29] and Coloured Petri Nets given by K. Jensen in [11, 12] are the foundation of our proposed Fuzzy Coloured Petri Net (FCPN). A generalized non-hierarchical FCPN, instead of the 12-tuple given in [26], can be defined as an 18-tuple : **FCPN** = $(\Sigma, P, T, D, A, N, C, G, E, Th, F, W, CF, \alpha, \beta, \gamma, \varphi, \theta)$, where

$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_l\}$, a finite set of non-empty types, called *colour sets*, $l \leq 0$,

$P = \{P_C, P_F\}$, a finite set of *places*,

$P_C = \{pc_1, pc_2, \ldots, pc_m\}$, a finite set of places that model the dynamic control behaviour of system, called *control places*, $m \geq 0$,

$P_F = \{pf_1, pf_2, \ldots, pf_n\}$, a finite set of places that model the fuzzy production rules, called *fuzzy places*, $n \geq 0$, $P_C \cap P_F = \varnothing$,

$T = \{T_C, T_F\}$, a finite set of *transitions*,

$T_C = \{tc_1, tc_2, \ldots, tc_i\}$, a finite set of transitions that are connected to and from *control places*, called *control transition*, $i \geq 0$,

$T_F = \{tf_1, tf_2, \ldots, tf_j\}$ a finite set of transitions that are connected to or from fuzzy places, called *fuzzy transition*, $j \geq 0$, $T_C \cap T_F = \varnothing$,

$D = \{d_1, d_2, \ldots, d_h\}$, a finite set of *propositions*, $|P_F| = |D|$,

$A = \{a_1, a_2, \ldots, a_k\}$, a finite set of *arcs*, $k \geq 0$, $P \cap T = P \cap A = T \cap A = \varnothing$,

$N: A \rightarrow P \times T \cup T \times P$, a *node function*, it maps each arc into a pair where the first element is the source node and the second is the destination node, the two nodes have to be of different kind,

In: an *input function* that maps each node, *x*, to the set of nodes that are connected to *x* by an input arc of *x;*

Out: an *output function* that maps each node, *x*, to the set of its nodes that are connected to *x* by an output arc of *x*,

$C: (P \cup T) \rightarrow \Sigma_{ss}$ a *colour function*, i.e. it maps each place and transition to a super-set of colour set,

$G: T \rightarrow$ expression, a *guard function*, $\forall t \in T$: $[Type(G(t)) = Boolean \wedge Type(Var(G(t))) \subseteq \Sigma]$, where *Type*(Vars) to denote the set of types $\{Type(v) \mid v \in \text{Vars}\}$, Vars is a set of variables, *Var*(*G*(*t*)) denotes the variables used in *G*(*t*),

$E: A \rightarrow$ *expression, an arc expression function*, $\forall a \in A$:

$[Type(E(a)) = C(p(a))MS \wedge Type(Var(E(a))) \subseteq \Sigma]$ where *p*(*a*) is the place of *N*(*a*), *MS* stands for multi-set,

$Th = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$ denotes a set of threshold values,

$F = \{f_1, f_2, \ldots, f_n\}$ denotes a set of fuzzy sets,

$W = \{w_1, w_2, \ldots, w_p\}$ denotes a set of weights,

$CF = \{cf_1, cf_2, \ldots, cf_m\}$ denotes a set of certainty factors,

$\alpha: P_F \rightarrow [0, 1]$ is an association function, a mapping from fuzzy transitions to certainty factors,

$\beta: P_F \rightarrow D$ is a bijective mapping between the proposition and the fuzzy place for each node,

$\gamma: P_F \rightarrow Th$ is an association function, a mapping from fuzzy places to threshold values. Each proposition in the antecedent is assigned a threshold value,

$\varphi: P_F \rightarrow F$ is an association function that assigns a fuzzy set to each fuzzy place,

$\theta: P_F \rightarrow W$ is an association function that assigns a weight to each fuzzy place.

2.2. A Fuzzy Coloured Petri Net example

To illustrate our definition of non-hierarchical FCPNs, the net in Fig. 1 can be represented as the followings:

$\Sigma = \{H, I, J\}$. There are three colour sets used.

$P = \{P1, P2, P3, P4a, P4b, P5\}$. Six places are used namely from *P*1 to *P*5.

$P_C = \{P1, P2, P3\}$. The first three places are for modelling control processes.

$P_F = \{P4a, P4b, P5\}$. The last three places are for modelling fuzzy rule.

$T = \{T1, T2, T3, T4\}$. Four transitions used.

$T_C = \{T1, T2, T4\}$. These three transitions are for modelling control processes.

$T_F = \{T3\}$. The transition *T*3 is for modelling fuzzy rule.

$D = \{d1, d2, d3\}$. There is only one fuzzy rule with three linguistic propositions. E.g. IF the tool-speed is high AND the feed-rate is small THEN the surface-quality is good.

A = There are twelve arcs used. e.g. arc1, arc2 ... arc12 where {arc1:*P*1-*T*1, arc2:*P*2-*T*1, arc3:*T*1-*P*3, arc4:*P*3-*T*2, arc5:*T*2-*P*4*a*, arc6:*T*2-*P*4*b*, arc7:*P*4*a*-*T*3, arc8:*P*4*b*-*T*3, arc9:*T*3-*P*5, arc10:*P*5-*T*4, arc11:*T*4-*P*1, arc12:*T*4-*P*2}. ":" means from.

N(*a*) = A node function, which maps each arc into a pair where the first element is the source node and the second is the destination node, e.g. *P*1to*T*1, *P*2to*T*1, *T*1to*P*3, *P*3to*T*2, *T*2to*P*4*a*, *T*2to*P*4*b*, *P*4*a*to*T*3, *P*4*b*to*T*3, *T*3to*P*5, *P*5to*T*4, *T*4to*P*1, *T*4to*P*2.

$$C(p \cup t) = \begin{cases} \{H\} & \text{if} \quad p \in \{P1\}, \\ \{I\} & \text{if} \quad p \in \{P2\}, \\ \{J\} & \text{if} \quad p \in \{P3,\ P4a,\ P4b,\ P5\}, \\ \{H, I\} & \text{if} \quad t \in \{T1\}, \\ \{j\} & \text{if} \quad t \in \{T2,\ T3,\ T4\}. \end{cases}$$

The colour sets for each place and transition are defined above.

G(t) = True. All Guards are defined "True" i.e. no additional constraints in the firing of transitions.

$$E(a) = \begin{cases} h & \text{if} \qquad a \in \{P1\text{to}T1, T4\text{to}P1\}, \\ i & \text{if} \qquad a \in \{P2\text{to}T1, T4\text{to}P2\}, \\ j & \text{otherwise.} \end{cases}$$

Fuzzy Rule R: IF the tool-speed is high AND the feed-rate is small
THEN the surface-quality is good (CF=0.85)

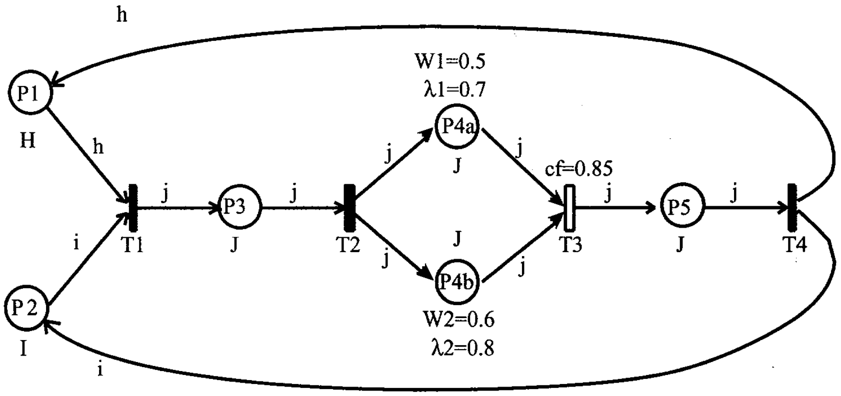

Fig. 1. A fuzzy coloured Petri net example.

The arc expressions are defined as simply passing different types of token (h, i and j).

$Th = \{0.7, 0.8\}$.

F = {"Tool-Speed", "Feed-rate" and "Surface-Quality"}, three fuzzy sets are defined in this example. The first one is the fuzzy set of Tool-Speed and the second one is feed-rate and the third one is Surface-Quality.

$W = \{0.5, 0.6\}$.

$CF = \{0.85\}$.

$\alpha(t) = \{0.85 \text{ if } t \in (T3)\}$.

$$\beta(p_f) = \begin{cases} d1 & \text{if} \quad p_f = P4a, \\ d2 & \text{if} \quad p_f = P4b, \\ d3 & \text{if} \quad p_f = P5. \end{cases}$$

where $d1$ is "Tool-Speed is high", $d2$ is "Feed-Rate is small" and $d3$ is "Surface-Quality is Good".

The rule's three prepositions are mapped to place for which $P4a$ and $P4b$ are the antecedents and $P5$ is the consequent.

$$\gamma(p_f) = \begin{cases} 0.7 & \text{if} \quad p_f = P4a, \\ 0.8 & \text{if} \quad p_f = P4b. \end{cases}$$

We may assign 0.7 and 0.8, for example, as the threshold value of $P4a$ and $P4b$ respectively.

$\varphi = \{f_1, f_2, f_3\}$ where f_1 is the fuzzy set of "Tool-Speed", which may be composed of elements of {very slow, slow, normal, fast, very fast...etc.}

$$\theta(p_f) = \begin{cases} 0.5 & \text{if} \quad p_f = P4a, \\ 0.6 & \text{if} \quad p_f = P4b. \end{cases}$$

3. A case study

3.1. System description

One of the applications of Petri Net is to aid the production engineers to develop Flexible Manufacturing Systems (FMSs). Basically, the applications of Petri Net in FMSs could be classified into five different domains: modelling, qualitative and quantitative analysis, performance evaluation, scheduling and control implementations. In recent years, the development of automation in many production processes, especially in electronic production, have grown rapidly. However, in some processes, automated machines may not be able to achieve the high degree of accurate performance which is comparable to that of skilled workers. Electronic components insertion on printed circuit boards is one typical process. According to [25], in-

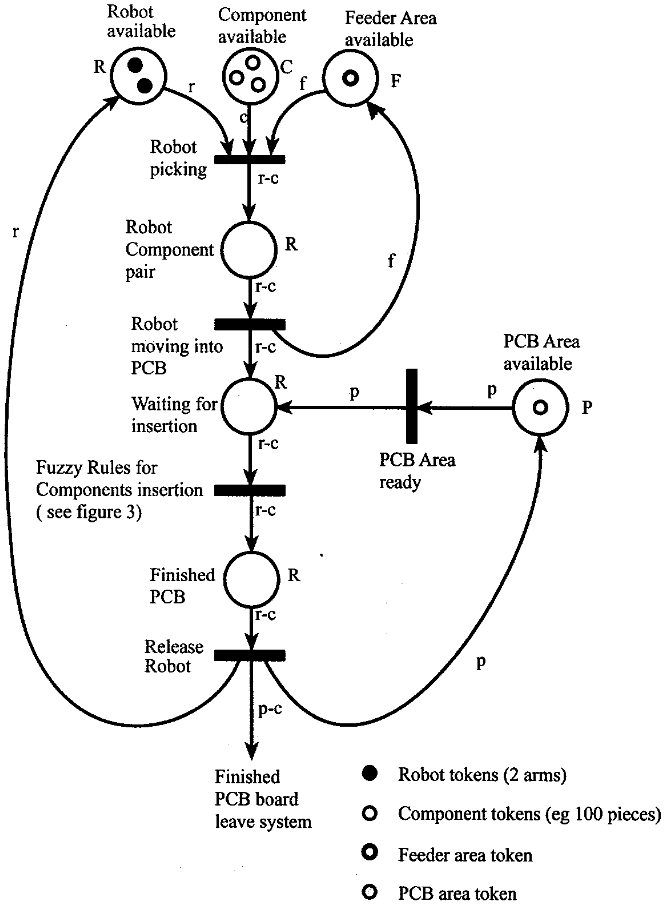

Fig. 2. FCPN for the flexible PCB assembly workstation.

corporating fuzzy rules in the automated insertion process will enable a smoother performance, and result in a faster process towards a full insertion.

Multi-robot systems for printed circuit board (PCB) assembly have the following three features which are distinct from conventional systems for automated assembly of mechanical parts [34]:

- Numerous insertions are required for a moderately complex printed circuit board. The component count ranges from 100 to 200 per board. The activities for each board assembly are highly repetitive.
- There is no strict sequence which has to be followed. This means that the insertion order of components can be largely altered without affecting the outcome. While in many mechanical assembly tasks, assembly possibilities are often quite limited due to heavy precedence constraints.
- Each insertion is performed by one robot manipulator even in a multi-robot system. While in mechanical assembly, two or more robots may be involved in fulfilling a single operation.

The assembly station that we study is AT&T FWS-200 flexible workstation. It will be briefly described here. The system consists of the following major components: frame structure, control cabinet, Robot Arms, AT&T PC386 computer, Drive Electronics Drawers, Control Panel, Vision System, and Peripheral Equipment.

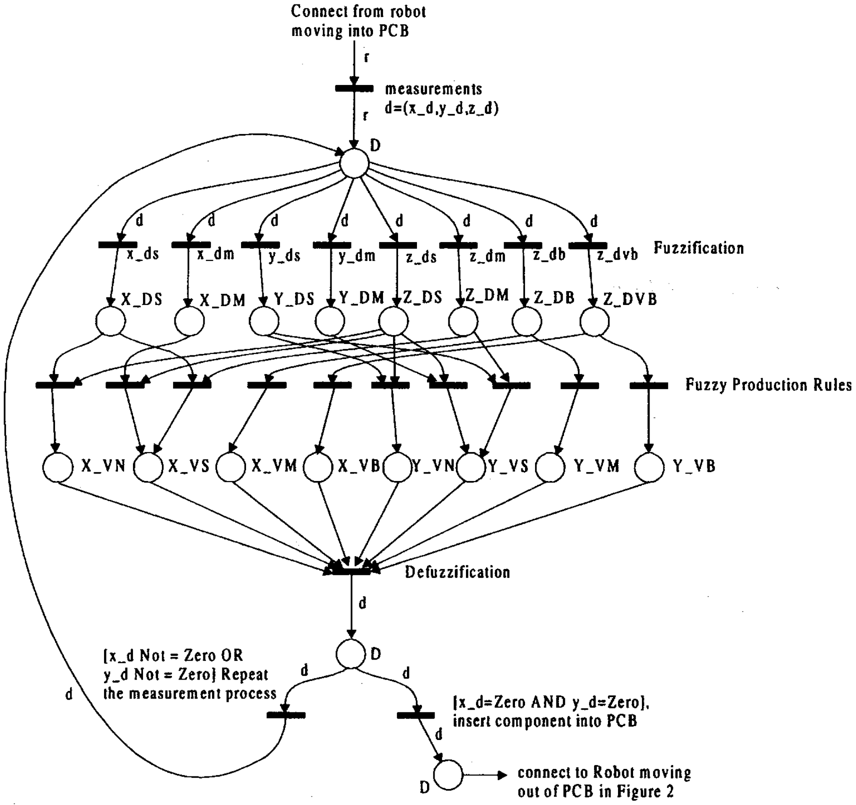

Fig. 3. FPN for the fuzzy production rules of 'Robot inserting' transition.

To improve the flexible workstation performance, we apply a robotic compliant wrist device which combines passive compliance and displacement sensor for robots to facilitate various complex manipulation tasks. When robots make contact with a workpiece, the wrist provides the necessary flexibility to accommodate transitions to correct the positioning error of robots and geometric tolerance of parts, and to avoid high impact forces normally generated in manufacturing operations. The sensing from the wrist device is used in the feedback loop for actively controlling contact forces and compensating for positioning errors during motion and contact. Since insertion operations when the wrist device is installed in robotic assembly, is inherently an ill-defined and complex process, fuzzy controllers might be more effective other than conventional controllers. The controller is comprised of four parts, fuzzification interface, knowledge base, decision making logic, and defuzzification interface. Membership functions and decision rules are deduced from operator experience and task requirement. Compared with the method of evaluating exact force zones, the fuzzy control approach presented a smoother performance, and yielded a fast process towards a full insertion.

In Fig. 2, the processes of flexible printed circuit board assembly workstation are modelled using our proposed Fuzzy Coloured Petri Nets methodology. There are eight activities: Robot Available, Component Available, Feeder Area Available, Robot Picking, Robot Moving into PCB, Robot Inserting, Robot Moving out PCB, PCB Available.

3.2. Fuzzy Coloured Petri Net modelling

At the initialization, there are two tokens, "*r*1" and "*r*2", where "*r*1" represents the left robot arm while "*r*2" represents the right robot arm. These two tokens are inside the CPN place Robot Available, with colour *R*, where the number of tokens represents the number of components available. In this study, 100 components are chosen for illustration purposes. For reasons of simplicity, we assume that there is only one type of component with colour *C*. (In a practical situation, the number of components is normally more than ten.) Besides, there is only one Feeder Area Available and one PCB Area Available, therefore, one feeder token is put inside the Feeder Area Available place and one PCB area token in the PCB Available place, respectively. The mutually exclusive property of the feeder area and PCB area can be maintained during the simulation process. After the initialization, the first transition will consume one "*r*" token (either "*r*1" or "*r*2"), one "*c*" token and one "*f*" token, then a "*r-c*" token will be created in place Robot Picking, meaning that the robot arm is holding the component. The robot arm will move to the PCB position and wait for insertion. If the feeder area is available, then the robot arm will insert the component into the printed circuit board. During this insertion process, the robot arm must rotate in an angle such that the legs of the components are perpendicular to the PCB board. The rotation of the robot arm is controlled by a set of ten fuzzy production rules (see Fig.3). Typical rules may be:

Rule1 : IF X axis's displacement is Small AND
Z axis's displacement is Small
THEN X's velocity is Zero.

Rule2 : IF X axis's displacement is Medium AND
Z axis's displacement is Small
THEN X's velocity is Small.

In this part, we will utilize the special feature of the proposed Fuzzy Coloured Petri Nets to model the operation sequencing and the fuzzy control rules together. First, we have some readings about the angles of the component to be inserted with respect to the PCB board (i.e. the x-axis, y-axis, z-axis displacements). These three measurements are then fuzzified into eight fuzzy variables: x_ds, x_dm, y_ds, y_dm, z_ds, z_dm, z_db and z_dvb (ds = displacement small, dm = displacement medium, db = displacement big and dvb = displacement very big). These eight fuzzy variables are used as inputs to ten fuzzy production rules. The rules' output will be defuzzified which gives the x_d and y_d values. If these two outputs are not zero, then they are used to feedback to the control of the robot arm. The new readings will be used to feed in the fuzzy control rules again until the x_d and y_d is zero, then the component will be inserted into the PCB and the finished PCB will leave the system.

The robot arm will move out from the PCB available area and the robot arm and the PCB area will again be available for the next component insertion. This process will keep on operating until all the components are used up.

4. Weighted fuzzy production rules

4.1. Description

In practical situations for applying Fuzzy Coloured Petri Nets modelling technique, we need to pay special attention to the evaluation method of fuzzy rules since they are the "expert" in controlling and supervising the manufacturing production processes. Traditionally, the applications of fuzzy production rules are rather limited because the relative degree of importance of each proposition in the antecedent contributing the consequent is ignored or assumed to be equal. This is unfortunately the case for many existing fuzzy expert system development shells or environments, therefore, in the following sections, we propose to assign a weight parameter to each proposition in the antecedent of a fuzzy production rule. We also devise a new fuzzy production rule evaluation method and a multi-level weighted fuzzy reasoning algorithm.

In general, the computation of fuzzy set f'_c including both the certainty factor and the weight is as follows:

First, in order to determine whether the rule *R* will be executed or not and the values for fuzzy set f'_c of c', a method based on the degree of subsethood [28, 29] will be used to compare the similarity between the two fuzzy sets f_{ai} and f'_{ai}. The formula is shown as follows:

$$S_{ai} = S(f_{a'i}, f_{ai}) = \frac{M(f_{a'i} \cap f_{ai})}{f_{a'i}} \qquad (1)$$

where M(f'_{ai}) is the sigma-count of the fuzzy set f'_{ai} which is defined by [33] as the size or cardinality of f'_{ai}, and M($f'_{ai} \cap f_{ai}$) is the sigma-count of the intersection of the two fuzzy sets f'_{ai} and f_{ai}. It is observed that S(f'_{ai}, f_{ai}), which is the degree of subsethood of f'_{ai} in f_{ai} differs from $S(f_{ai}, f'_{ai})$ and $0 \leq S(f'_{ai}, f_{ai}) \leq 1$. For instance, if f'_{ai} is a subset of f_{ai}, then $S(f'_{ai}, f_{ai}) = 1$. If f_{ai} has all its fuzzy membership values equal to zero, then $S(f'_{ai}, f_{ai}) = 0$.

Let

$$w_{\max} = \max[w_1, w_2, \ldots, w_n]$$

and

$$S_{a_{\max}} = S_a \text{ of } w_{\max},$$

$$S_w = \min[S_{a1} + (S_{a_{\max}} - S_{a1}) * \frac{w_{\max} - w_1}{w_{\max}},$$

$$S_{a2} + (S_{a_{\max}} - S_{a2}) * \frac{w_{\max} - w_2}{w_{\max}}, \ldots$$

$$S_{an} + (S_{a_{\max}} - S_{an}) * \frac{w_{\max} - w_n}{w_{\max}}]; \tag{2}$$

$$f'_c = \min[1, \text{fuzzy set in } f_c / (S_w * \mu)].$$

The above fuzzy production rule evaluation method generalises the traditional method of computing the fuzzy set of the deduced consequent by including the assigned weights and treats the traditional method as a limiting case of our method. So when all the weights assigned to the propositions in the antecedent of a fuzzy production rule are assumed equal, the traditional method will have $S_w = \min[S_{a1}, S_{a2}, \ldots, S_{an}]$ for a conjunctive rule and $S_w = \max[S_{a1}, S_{a2}, \ldots, S_{an}]$ for a disjunctive rule.

For a composite fuzzy disjunctive rule:

Let

$$w_{\max} = \max[w_1, w_2, \ldots, w_n]$$

and

$$S_{a_{\max}} = S_a \text{ of } w_{\max},$$

$$S_w = \max[S_{a1} + (S_{a_{\max}} - S_{a1}) * \frac{w_{\max} - w_1}{w_{\max}},$$

$$S_{a2} + (S_{a_{\max}} - S_{a2}) * \frac{w_{\max} - w_2}{w_{\max}}, \ldots$$

$$S_{an} + (S_{a_{\max}} - S_{an}) * \frac{w_{\max} - w_n}{w_{\max}}]. \tag{3}$$

In the case of a conjunctive rule (disjunctive rule), one may notice that the overall similarity measure is taken to be the minimum (maximum) of each individual similarity measure which has been adjusted by the assigned weights. When a proposition a_i in the antecedent of a conjunctive (disjunctive) fuzzy production rule has the highest weight w_i and the S_{ai} is the lowest (highest) value, this S_{ai} is used together with the certainty factor of the rule to compute the fuzzy set of the deduced consequent. On the other hand, if a_i does not have the highest assigned weight w_i, S_{ai} is adjusted according to Eq. 2 (Eq. 3) to reflect the degree of importance of the assigned weights. After the adjustment is done the minimum (maximum) one will be chosen.

Furthermore, the maximum function used in the case of a disjunctive rule can also be used to solve the rule conflict problems, i.e., when multiple rules modify the same attribute in their respective consequents.

4.2. *Firing (evaluating) a weighted fuzzy production rule (fuzzy transition)*

Since the firing rules for a fuzzy transition are quite complicated, therefore, we would like to illustrate our idea by the printed circuit board in section three above.

Given the following fuzzy rule for controlling the robotic arm:

Rule R: IF X axis's displacement is large
AND Y axis's displacement is low
THEN
Z's velocity is low ($CF = 0.85$).

Rule R can be represented as follows:

Rule R: IF x is f_{a1} AND y is f_{a2}
THEN z is f_c ($CF = \mu$).

Given X axis's displacement is *quite large* and its threshold value and weight are 0.6, 0.9, respectively, Y axis's displacement *fairly low* and its threshold value and weight are 0.5, 0.6 respectively, what can be drawn for velocity of Z (i.e. f'_c)?

Here we have: f_{a1} = *large*, f'_{a1} = *quite large*, f_{a2} = *low*, f'_{a2} = *fairly low*, f_c = *low* and f'_c = ?, $x = X$ axis's displacement, $y = Y$ axis's displacement, and z = Z's velocity.

Assume $X = \{1, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000\}$ to be the universe of discourse for X axis's displacement, and the universe of discourse of the Y axis's displacement is $Y = \{500, 1000, 1500, 2000, 2500, 3000, 3500, 4000\}$.

The fuzzy sets of f_{a1}, f'_{a1}, f_c, f_{a2} and f'_{a2} are given as follows:

$$\begin{aligned} f_{a1} = {} & 0.0/1 + 0.2/500 + 0.3/1000 + 0.4/1500 \\ & + 0.5/2000 + 0.6/2500 + 0.7/3000 \\ & + 0.8/3500 + 0.9/4000, \end{aligned}$$

$$\begin{aligned} f'_{a1} = {} & 0.0/1 + 0.3/500 + 0.4/1000 + 0.5/1500 \\ & + 0.6/2000 + 0.7/2500 + 0.8/3000 \\ & + 0.9/3500 + 0.95/4000, \end{aligned}$$

$$\begin{aligned} f_c = {} & 0.95/5 + 0.8/10 + 0.7/15 + 0.6/20 + 0.5/25 \\ & + 0.4/30 + 0.3/35 + 0.2/40 + 0.1/45, \end{aligned}$$

$$\begin{aligned} f_{a2} = {} & 1.0/1 + 0.8/500 + 0.7/1000 + 0.6/1500 \\ & + 0.5/2000 + 0.4/2500 + 0.3/3000 \\ & + 0.2/3500 + 0.1/4000, \end{aligned}$$

$$\begin{aligned} f'_{a2} = {} & 1.0/1 + 0.9/500 + 0.85/1000 + 0.78/1500 \\ & + 0.65/2000 + 0.5/2500 + 0.45/3000 \\ & + 0.37/3500 + 0.26/4000. \end{aligned}$$

The degree of subsethood between f_{a1} and f'_{a1}; f_{a2} and f'_{a2} is given by:

$$S(f'_{a2}, f_{a1}) = 4.4/5.15 = 0.85$$

and

$$S(f'_{a2}, f_{a2}) = 4.6/5.76 = 0.80.$$

Since $0.85 \geq 0.6$ and $0.8 \geq 0.5$, and

$$\begin{aligned} S_w = {} & \mathrm{Min}[0.85 + (0.85 - 0.85) * (0.9 - 0.9)/0.9, \\ & 0.8 + (0.85 - 0.8) * (0.9 - 0.6)/0.9] \\ = {} & 0.82, \end{aligned}$$

the fuzzy set $f'_c = \min\{1, f_c/(0.83 * 0.82)\}$ and the result is:

$$\begin{aligned} f'_c = {} & 1.0/5 + 1.0/10 + 1.0/15 + 0.89/20 + 0.74/25 \\ & + 0.59/30 + 0.44/35 + 0.30/40 + 0.15/45. \end{aligned}$$

From this fuzzy set, one can infer that f'_c is *fairly low*. In this example, one may observe that $S_{a2} = 0.8$ has been adjusted due to its assigned weight 0.6 less than 0.9 of a_1.

5. An enhanced fuzzy reasoning algorithm

5.1. Description

By fuzzy reasoning algorithm we mean that it is a reasoning mechanism (an inference engine) used in a fuzzy expert system, which makes use of working memory, given facts, a set of weighted fuzzy production rules, and a fuzzy production rule evaluation method to draw the final goals (conclusions). The fuzzy production rule evaluation method refers to an approximate reasoning method which draws a conclusion of a single weighted fuzzy production rule from given fact(s). Fuzzy reasoning algorithm therefore involves multiple weighted fuzzy production rules whereas a fuzzy production rule evaluation method involves only a single weighted fuzzy production rule.

The multi level weighted fuzzy reasoning algorithm presented here is an enhancement of that in [31], i.e., weights are incorporated in fuzzy production rules. It employs the fuzzy production rule evaluation method described in the previous section. It is different from the algorithm proposed by [7].

In the evaluation process, we make the following assumptions:

(i) The fuzzy knowledge is represented by fuzzy production rules.

(ii) The weighted fuzzy production rules are mapped into the FCPN model as described in Section 2.

(iii) The integrity checking of the weighted fuzzy production rules is done by Fuzzy Petri Nets [1, 7, 15, 19]. Cyclic rules in the knowledge base are detected by the algorithm which builds the reversed immediate reachable place table.

In the subsequent paragraph, a multi level weighted fuzzy reasoning algorithm is presented. The other supplementary algorithms used to build the *goal set, reversed immediate reachable place, adjacent place, and user input set,* will be described here briefly. The detailed algorithms are presented in the appendix. A *goal set* (GS) is a set whose elements are the places (propositions) representing the final goals. A *reversed immediate reachable place* (RIRP) is represented by a 2-column table, with the 1st column containing all the places (consequents) and the 2nd column containing their antecedent places in the knowledge base. An *adjacent place* (AP) is a 3-column table. The 1st column comprises all the places which have adjacent antecedent(s), and the 2nd column comprises the consequent of the place in the 1st column while the 3rd column comprises the adjacent antecedents of the place in the 1st column. The *user input set* (UIS) is a set whose elements are provided by the users.

5.2. A multi-level weighted fuzzy reasoning algorithm

Step 1. Build GS, RIRP, AP and UIS by calling their respective algorithms (see Appendix).

Step 2. Get an element p_i from GS.

Goto Step 6 if GS is empty.

Step 3. Get the antecedent place(s) of p_i from the RIRP table.

Step 4. For each of the antecedent place a_i:

4.1. If a_i does not have a token and it is not an element of UIS, put a_i's consequent and a_i onto the stack. Get the antecedent place(s) of a_i.

4.2. Request user's input if a_i is an element of UIS.

4.3. If a_i and its adjacent places have values (tokens), compute a value for the consequent of a_i using the fuzzy production rule evaluation method. Delete the consequent of a_i from the stack. Goto Step 5 if the stack is empty

4.4. If a_i has a value (a token) and its consequent has other arc incident on it, compute a value for the consequent of a_i using the fuzzy production rule evaluation method. Delete the consequent of a_i from the stack. Goto Step 5 if the stack is empty.

Step 5. Display the conclusion for p_i. Continue to get other possible conclusions if the user requests to do so. Delete p_i from GS and goto Step 2

Step 6. End.

6. Conclusion and discussion

In this paper, a contribution was made in the development of the Fuzzy Coloured Petri Net (FCPN) model. This technique integrated the modelling power of both Fuzzy Petri Net and Coloured Petri Net while preserving the advantages of both nets. The ability of FCPN modelling technique is demonstrated using an electronic components insertion process in Flexible PCB Manufacturing System. The developed model is able to contribute two significant advantages. The first advantage is that the dynamic and fuzzy control behaviour of the two robotic arms can be modelled by a single FCPN. Since CPN is unable to model fuzzy concept and FPN is unable to model the dynamic behaviour of the system, FCPN is a natural and powerful tool to model a system with fuzzy and dynamic processes. It also enables the whole system to be analyzed in one single net. The second advantage is that using FCPN will significantly reduce the graphical complexity. As we have shown in the case study, if only FPN is used alone, two identical FPNs are needed to model the fuzzy production rules associated with two identical robot arms. Therefore, FCPN not only provides a more concise graphical representation, it also demonstrates the power to use one kind of Petri Net to model the whole

FMS which cannot be done the otherwise.

Furthermore, by incorporating weights into the propositions of the antecedent of a fuzzy production rule, meaningful and important knowledge can be captured and applied to practical situations. The conventional fuzzy production rules are enhanced to include weights in each proposition of the antecedent. A new fuzzy production rule evaluation method has been proposed to generalize the traditional rule evaluation method. The proposed multi-level weighted fuzzy reasoning algorithm which incorporates the new fuzzy production rule evaluation method is devised to act as an inference engine for fuzzy expert systems that control or supervise manufacturing processes.

Extensions of research are suggested here. First, a large industrial scale simulation experiment is needed in order to demonstrate how the technique can improve flexibility in design, system efficiency, cost effectiveness and the quality of products. Secondly, a state-space reachability analysis is required in order to assess the practical performance of our methodology. Thirdly, a user interface should be developed to allow users to quickly determine how to improve efficiency, reduce the cost, and pin down the bottleneck. Furthermore, some system validation and verification effort should be investigated to strengthen the reliability and robustness of the FCPN model.

Appendix A

In this appendix, algorithms used to build the *goal set, reversed immediate reachable place, adjacent place, and user input se*t tables are presented.

A.1. A goal set (GS) algorithm

```
1. Initialise GS as an empty set.
2. Get an element $p_i$ from $P$.
3. If $P$ is empty goto 7.
4. If $p_i$ has no consequent place (immediate reachable place), add $p_i$ to GS.
5. Delete $p_i$ from $P$.
6. Goto 2.
7. End.
```

A.2. A reversed immediate reachable place (RIRP) algorithm

The RIRP table is assumed to have two columns: consequent place p_i and antecedent place p_j.

```
1. Get an element $p_i$ from $P$.
2. If $P$ is empty goto 8.
3. Search the FPN for places which have $p_i$ as its immediate reachable place (consequent).
4. For each found place (node) which has $p_i$ as its immediate reachable place (consequent), add $p_i$ to the consequent place $p_i$ column, add the found place(s) to the antecedent place $p_i$ column.
5. If no node has $p_i$ as its immediate reachable place (consequent), add $p_i$ to the consequent place $p_i$ column, add the $\varnothing$ to the antecedent place $p_i$, column.
6. Delete $p_i$ from $P$.
7. Goto 1.
8. For each place $p_i$ in column consequent place $p_i$, connect it with a directed arc from its antecedent place(s). If a closed walk (i.e., a cycle) exists, alert knowledge engineers and terminate the program.
9. End.
```

A.3. An adjacent place (AP) algorithm

```
1. Get an element $p_i$ from $P$.
2. If $P$ is empty goto 7.
3. Get the antecedent place(s) of $p_i$ from the RIRP table.
4. If there is more than one antecedent place of $p_i$: for each of its antecedent place, add it to the place $p_i$ column, add $p_i$ to the place $p_k$ column, add other antecedent places to $AP_{ik}$ column.
5. Delete $p_i$ from $P$.
6. Goto 1.
7. End.
```

A.4. An user input set (UIS) algorithm

```
1. Initialise UIS as an empty set.
2. Get an element $p_i$ from $P$.
3. If P is empty goto 7.
4. If antecedent of $p_i$ is $\varnothing$ (obtained from the RIRP table), add $p_i$ to UIS.
5. Delete $p_i$.
6. Goto 2.
7. End.
```

References

[1] R. Agarwal and M. Tanniru, A Petri Net based approach to verify the integrity of production systems, *International Journal of ManMachine Studies* (1992), 447–465.

[2] T. Cao and A.C. Sanderson, Task sequence planning using Fuzzy Petri Nets, *IEEE Transactions on Systems, Man, and Cybernetics* **25**(5) (1995), 755–768.

[3] S.M. Chen, A weighted fuzzy reasoning algorithm for medical diagnosis, *Decision Support Systems* **11** (1994), 37-43.

[4] S.M. Chen, I.S. Ke and J.F. Chang, Knowledge representation using Fuzzy Petri Nets, *IEEE Transaction on Knowledge and Data Engineering* **2** (1990), 311–319.

[5] R. Cossins and P. Ferreira, CELERITAS: A Coloured Petri Net approach to simulation and control of flexible manufacturing systems, *International Journal of Production Research* **30**(8) (1992), 925–1956.

[6] J. Ezpeleta, J.M. Colom and J. Martinez, A Petri Net based deadlock prevention policy for flexible manufacturing systems, *IEEE Transactions on Robotics and Automation* **11**(2) (1995), 173–184.

[7] M.L. Garg, S.I. Ahson and P.V. Gupta, A Fuzzy Petri Net for knowledge representation and reasoning, *Information Processing Letters* **39** (1991), 165–171.

[8] M. Hanna, A. Buck and R. Smith, Fuzzy Petri Nets to control vision system and robot behaviour under uncertain situations within an FMS cell, in: *Proc. of the 3rd IEEE Conference on Fuzzy Systems: IEEE World Congress on Computational Intelligence,* 1994, pp. 1889–1894.

[9] M. Hanna and M. Moheb, Fuzzy Petri Nets with neural networks to model products quality from a CNC-milling machining centre, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* **26**(5) (1996), 638–645.

[10] I. Hatono, K. Yamagata and H. Tamura, Modeling and online scheduling of flexible manufacturing systems using stochastic Petri Nets, *IEEE Transactions on Software Engineering* **17**(2) (1991), 126–132.

[11] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use,* Vol.1, Springer-Verlag, Berlin, Heidelberg, 1992.

[12] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use,* Vol.2, Springer-Verlag, Berlin, Heidelberg, 1995.

[13] A. Knoar and A.K. Mandal, Uncertainty management in expert systems using Fuzzy Petri Nets, *IEEE Transactions on Knowledge & Data Engineering* **8**(1) (1996), 96–105.

[14] D.Y. Lee and F. DiCesare, FMS scheduling using Petri Nets and heuristic search, in: *Proc. of the 1992 IEEE international Conference on Robotics and Automations*, 1992, pp. 1057-1062.

[15] N.K. Liu and T.S. Dillon, Formal description and verification of production systems, *International Journal of Intelligent Systems* **10** (1995), 399–442.

[16] C.G. Looney, Fuzzy Petri Nets and applications, in: *Fuzzy Reasoning in Information, Decision and Control Systems*, Kluwer Academic, 1994, pp. 511–527.

[17] C.G. Looney, Fuzzy Petri Nets for rulebased decision making, *IEEE Transactions on System, Man and Cybernetics* **18** (1988), 178–183.

[18] W.W. Luggen, *Flexible Manufacturing Cells and Systems*, Prentice-Hall, 1991.

[19] H. Scarpelli and F. Gomide, A high level net approach for discovering potential inconsistencies in fuzzy knowledge bases, *Fuzzy Sets and Systems* **64** (1994) , 175–193.

[20] H. Scarpelli and F. Gomide, A reasoning algorithm for high-level Fuzzy Petri Nets, *IEEE Transactions on Fuzzy Systems* **4**(3) (1996), 282–294.

[21] H. Scarpelli and F. Gomide, Fuzzy reasoning and fuzzy Petri Nets in manufacturing systems modelling, *Journal of Intelligent and Fuzzy Systems* **1** (1993), 225–242.

[22] S.C.K. Shiu, J.N.K. Liu and D.S. Yeung, Modelling hybrid rule/frame-based expert systems using Coloured Petri Nets, in: *Proc. of the Eight International Conference of Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 95)*, 1995, pp. 525–531.

[23] M. Silva and R. Valetta, Petri Nets and flexible manufacturing, in: *Advances in Petri Net '89,* G. Rozenberg, ed., Springer-Verlag, 1990, pp. 375–417.

[24] N. Viswanadham, Y. Narahari and T.L. Johnson, Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri Net models, *IEEE Transactions on Robotics and Automation* **6**(6) (1990), 713–723.

[25] Y. Xu, R.P. Paul and H.Y. Shum, Fuzzy control of robot and compliant wrist system, in: *Conference Record of the 1991 IEEE Industry Applications Society Annual Meeting*, 1991, pp. 1431–1437.

[26] D. Yeung, J. Liu, S. Shiu and G. Fung, Fuzzy Coloured Petri Nets in modelling flexible manufacturing systems, in: *Proc. of the Ninth International Symposium on Artificial Intelligence: Industrial Fuzzy Control and Intelligent Systems, ISAI/IFIS* (November 12–15, 1996), Cancun, Mexico, 1996, pp. 100–107.

[27] D.S. Yeung and E.C.C. Tsang, A comparative study on similarity based fuzzy reasoning methods, *IEEE Transactions on System, Man, and Cybernetics, Part B: Cybernetics* **27**(2) (1997), 216–227.

[28] D.S. Yeung and E.C.C. Tsang, Fuzzy knowledge representation and reasoning using Petri Nets, *International Journal of Expert Systems with Applications* **7** (1994), 281–290.

[29] D.S. Yeung and E.C.C. Tsang, Improved fuzzy knowledge representation and rule evaluation using Fuzzy Petri Nets and degree of subsethood, *International Journal of Intelligent Systems* **9** (1994), 1083–1100.

[30] D.S. Yeung and E.C.C. Tsang, A multi-level weighted fuzzy reasoning algorithm for expert systems, *IEEE Transactions on System, Man and Cybernetics, Part A: System amd Human* **28**(2) (1998), 149–158.

[31] D.S. Yeung, J.W.T. Lee and E.C.C. Tsang, A new fuzzy reasoning algorithm for fuzzy expert system, in: *Proc.of '94 Korean/Japan Joint Conference on Expert Systems,* 1993, pp. 115–118.

[32] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems, Man and Cybernetics* **3** (1973), 28–44.

[33] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-1, *in: Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, R.R. Yager et al., eds, 1987.

[34] M. Zhou and M.C. Leu, Modelling and performance analysis of a flexible pcb assembly station using Petri Nets, *Transactions of the ASME: Journal of Electronic Packaging* (1991), 410–416.