'89 KACC 1989. 10. 27~28

A MODELLING METHODOLOGY FOR ROBOTIC WORKCELLS THROUGH KNOWLEDGE BASE

°Dae-Won Kim, Myoung-Sam Ko, and Bum-Hee Lee

Robotics and Intelligent Systems Laboratory
Dept. of Control and Instrumentation Engineering
Seoul National University, Seoul 151-742, KOREA

In this paper, a modelling methodology for a robotic workcell is proposed and copmared with the conventional Petri nets model. Also, a method for managing the cell operation is described through the knowledge base. The knowledge bases for state transition and assembly job information are obtained from the state transition map(STM) and the assembly job tree(AJT), respectively. Using the knowledge base, the system structure is discussed in both managing the cell operation and evaluating the various performance. Finally, a simulation algorithm is presented with the simulation results to show the effectiveness of the proposed modelling approach.

1. Introduction

Although a number of the FMS problems have already been solved and more than hundreds have been implemented so far, the FMS concept is too complex to be handled mathematically through appropriate modelling. The FMS problems depend upon the selected modelling methods, problem domains, and viewpoints.

The FMS may be viewed as one of the discrete event dynamic system(DEDS). The evolution of the DEDS depends upon the interactions of discrete and asynchronous events, and the state transition occurs only at discrete instants. However, we do not have a good analytical model for the DEDS, while we do have for the continuous variable dynamic systems. In 1989, Ho[14] scanned the issue for DEDS in which the model classification was made and the modelling desiderata for DEDS were summarized.

Up to date, a number of approaches to the FMS problems have been proposed, which can be classified into the performance-related, logically-based, and algebraically-based models, etc. according to Ho's claims. In the performance-related model, there are the queueing network(QN) model, the discrete event simulation, and the perturbation analysis(PA)[1,10,12,14].To the logically-based model, belong the Petri nets(PN) model, the finite state machine(FSM), and the extended state machine(ESM), and so forth[3,4,5,6,14,15]. In the algebraically-based model, there are the Boolean model and the min-max algebraic model[2,14]. Due to the development of the artificial intelligence technology, many kinds of knowledge-based systems have become gradually applicable to the real world. Lately, various knowledge-based control algorithms[7,8,11,13,16]have been proposed for controlling the FMS, and their application domain tends to be enlarged.

In this paper, we propose a methodology for modelling of robotic workcells through knowledge base. This approach lays emphasis on the cell flexibility complied with task requirements, the provision of modelling procedures using the concept of state variables, and the construction of systematic knowledge bases for modelling the robotic workcell. The proposed modelling approach is based on the concept of state variable which we define as the state of the cell element. The cell elements include both the active and the passive elements. Since the cell elements are closely coupled and interacted with each other, their relations can be obtained only through synthetic observations and analyses. This paper focuses on this problem, and also presents a solving approach through the knowledge base.

2. Problem Formulation

2.1 Problem Descriptions

We assume that the target system is deterministic and the system states are finite or countable. Also, each robot is driven in the collision-free mode.

In order to model a robotic workcell for operation management, we formalize the frame to specify the cell structure. Also, we identify the elementary sets in the structure, and define the set mapping for the dynamical cell behaviors.

A robotic assembly cell can be represented by the following structure, Σ^9.

$$\Sigma : (E, X, M, Y; \Phi, \Omega, \tau) \quad (1)$$

where,

- E : the event set involving the exogenous and endogenous events
- X : the state variable vector set
- M : the measured environmental condition set
- Y : the output vector set
- Φ : the state transition mapping under the environmental condition set M
- Ω : the output mapping under the environmental condition set M
- τ : the time advance function

Here, we define the state transition mapping Φ which is the mapping from $(X \times E)$ into X with the restriction of the environmental condition set M as,

$$\Phi : [(X \times E) \rightarrow X]_M \quad (2)$$

Also, the output mapping is defined as,

$$\Omega : [(X \times E) \rightarrow Y]_M \quad (3)$$

The time advance function τ is defined as the mapping from x into the nonnegative real value with infinity, and $\tau(x)$ denotes the time that the cell remains in the state x. Thus,

$$\tau : x \rightarrow R^{+} \quad (4)$$

In this paper, we make the state transition mapping graphically by the nodes and the links. Then, we transform the graphic information into the knowledge. Here, the graphic information is defined as a state transition map(STM), from which we can get all information about the transition of the state variables in the defined cell structure, including dynamic information of the part to be assembled in the specified cell operation. The advantage of this approach lies in the good changeability to add and delete the knowledge easily, and the reasoning power to yield the better results.

2.2 Problem Solving Approach

The logical behavior of a DEDS can be modelled graphically by either the conventional Petri nets or the proposed approach. We can compare them in terms of basic elements and modelling structure. In Table 1, some elements corresponding to three basic functions are compared. The place represents the condition for progressing a process or progressing state of the process, while the node in the STM represents the state of the system element. The link in the STM is used instead of the transition and the arc in the Petri nets model, and represents the operation for state transition. Also, the link connotes the link condition for state transition. To designate the discrete state of the system, the marking is used in the Petri nets model, while the dynamic knowledge is managed in the proposed model.

Table 1. Comparison of Basic Elements in two Modelling Tools

Petri Nets Model	the Proposed Model
Place: conditions or states	Node: states
Transition: events or operations Arc: connection of place to transition	Link: operations for state transition; connection from node to node
Marking: a state of the system being modelled	Dynamic Knowledge: current states of state variables, EC flags, and Assembly Job Pointer

Here, we want to emphasize the difference of the modelling structure between the proposed scheme and the Petri nets. Since the Petri nets model is process-oriented, the operations obtained by decomposing a process, and the resource states or conditions required for each operation, are represented graphically in the process flow. Thus, it provides an easy visualization for complex systems and enables a hierarchical modelling. However, a change in the system's functioning induces a substantial change in the whole net's structure[17].

On the contrary, the proposed model is structured in a modular and functionally-distributed form. That is, the STMs corresponding to the cell elements are constructed and managed independently of the job assignments. Thus, the advantages are that good modularity can be provided in modelling structure, and the system sensitivity to the system element can be also easily obtained. Additionally, it is easy to store and analyze the historical data from the knowledge base constructed. However, a visual understandability is not provided for the given process.

In simulating the system model, the synchronous timing method[20] is adopted, where the simulation timing is advanced by an appropriately chosen sampling time and the system states are changed at every sampling time, is adopted. We denote the sampling time as the Global Sampling Time(GST) from now on.

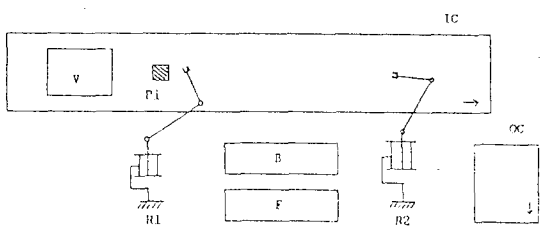
If a system with an assembly job is given, we can carry out the system modelling and the system analysis through the following problem solving approach.

[step1] Establish the state variables and define the representative states for each state variable.
[step2] Describe the relationships among the state variables.
[step3] Construct the state transition maps(STMs) graphically.
[step4] Establish the environment conditions(ECs) and define the EC flags.
[step5] Transform the graphic information into the knowledge and construct the knowledge base for STM.
[step6] Define global variables and store them with the ECs.
[step7] Classify the predetermined information for the assigned assembly job and store it.
[step8] Transform the information into the knowledge and construct the knowledge base for AJT.
[step9] Classify the knowledge and grasp the interactions among the knowledge bases.
[step10] Design the simulation driver and construct the inference engine for operation of knowledge-basec system.
[step11] Make decisions for the operation strategy of robot manipulator and calculate the control input for the robot manipulators.
[step12] Analyze the system operation through the sensitivity analysis.
[step13] Design the system and evaluate its performance before operating it, then redesign it from the reviewing results.

3. System Modelling

3.1 System Descriptions

We consider a robotic workcell as a simple discrete event dynamical system for system modelling. The cell is composed of two robot manipulators, a robot vision system, an input and an output conveyor system, a buffer as a waiting place for assembly sequence, and a fixture as an assembly station. The overall configuration is depicted in Fig.1.



V : the robot vision system
P_i : the part to be assembled
R1,R2 : two robots
B : the buffer
F : the fixture
IC : the input conveyor
OC : the output conveyor

Fig.1 Overall Configuration of a Robotic Workcell.

We now describe the cell operation and the function of the cell elements. If a part arrives randomly on the input conveyor system, the position and orientation information for the part is obtained through the robot vision system. We now describe the function of two robot manipulators and the cell operation. First, we assume that we call two robot manipulators as R1 and R2, respectively. The role of two robots R1 and R2 is different from each other. In other words, the R1 works for a classification job, in which the R1 transfers one part on the input conveyor system to the buffer, classifies it according to information from the robot vision system, and stacks it separately. The R2 works for an assembly job in which the R2 transfers one part in a proper assembly sequence from the buffer to the fixture and assembles it. If the assembly product is completed in the fixture, the R2 transfers it to the output conveyor system.

3.2 State Variable Representation for Part

The model we develop is based on the concept of state variable in state space control theory. We define the state variable as the state of the cell element, which is innovative compared with the existing concept of the state variable. In this paper, the state variable representation for the part only and the corresponding state transition map are described.

Since the parts arrived in the cell are to be considered in terms of transition, we need to represent the part state as a state vector. The part state is defined as a vector consisting of each part state variable P^i_n, where i=1,2, ... ,m and m is the number of parts arrived in the cell. The vector of the part state is denoted as Eq.(5), and the status of all parts in the cell is denoted as one point in an m-dimensional coordinate system.

$$P_n = [\ P^1_n,\ P^2_n,\ \ldots\ ,\ P^m_n\ ]^T \qquad (5)$$

where,

P_n : the current part state vector
P^i_n and i=1,2, ... ,m : each part state
m : the number of parts arrived in the cell

The transition of one element P^i_n in the part state vector depends on the past information such as, the previous robot and part states, the previous gripping, assembly and recognition failure indicators, and the part arrival indicator. The functional dependency is described in an implicit form as Eq.(6).

$$P^i_{n+1} = G\ (\ P^i_n,\ R_n,\ \alpha_n,\ \beta_n,\ \gamma_n,\ \delta_n\ ) \qquad (6)$$

where,

P^i_{n+1} : the next state of the i-th part with 6-dimension
P^i_n : the current state of the i-th part
$R_n = [R^1_n,\ R^2_n]^T$: the robot state vector
$\alpha_n = [\alpha^1_n,\ \alpha^2_n]^T$: the gripping failure indicator vector
$\beta_n = [\beta^1_n,\ \beta^2_n]^T$:the assembly failure indicator vector
γ_n : the current recognition failure indicator in vision processing
δ_n : the current part arrival indicator

According to the vector P_n, we can identify the cell status for the purpose of the job requirement and calculate the control inputs for robots at each decision instant.

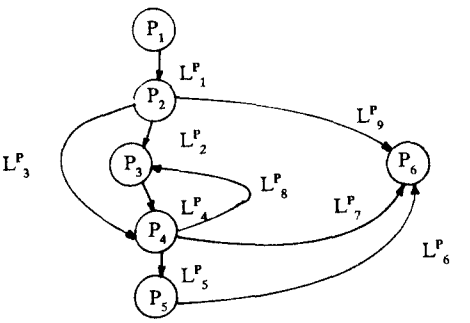
3.3 State Transition Map for Part

The part state can have six different states: the out-of-cell state indicating that a part is not entered in the cell, the pre-executable state indicating that the part has been arrived at the input conveyor and is not transferred to the buffer, the waiting state indicating that the part in the buffer is not in the proper assembly sequence, the just-executable state indicating that the part in the buffer is in the proper and is transferred to the fixture, the execution state indicating that the part is being assembled, and the failure-generation state indicating that the a failure has occurred.

The STM for a part is depicted in Fig.2, where P_i, i=1,2,...,6 represent the six possible states of the part, L^P_j, j=1,2,...,9 represent the link conditions for state transition, and superscript P is used to emphasize the part. In Table 2, the STT for part is provided, and in particular, four environment condition(EC) flags have an effect on the state transition for part.

In Fig.2, the link conditions for state transition are described in detail as follows:

L^P_1 : if a part has been arrived at the input conveyor and is not yet transferred to the buffer.
L^P_2 : if the part in the buffer is not in the proper assembly sequence.
L^P_3 :if the part recognized is in the proper assembly sequence.
L^P_4 :if the part in the buffer become in the proper assembly sequence.
L^P_5 : if the fixture is ready for assembling and gripping is not failed.
L^P_6 : if assembly is failed.
L^P_7 : if gripping is failed.
L^P_8 :if the part in the buffer becomes in the improper assembly sequence.
L^P_9 : if a recognition or a gripping failure has occurred.



P_1 : the out-of-cell state
P_2 : the pre-executable state
P_3 : the waiting state
P_4 : the just-executable state
P_5 : the execution state
P_6 : the failure-generation state

Fig.2 The STM for the Part.

Table 2. The STT for Part

Link	P_n	R^1_n	R^1_{n+1}	α^1_n	β^1_n	γ_n	δ_n	CF	PF	P-BUF	FF	P_{n+1}
L^P_1	P_1	*	*	*	*	*	1	1	0,1	0	*	P_2
L^P_2	P_2	*	*	*	*	*	*	*	*	1,3	*	P_3
L^P_3	P_3	*	*	*	*	*	*	*	*	2	*	P_4
L^P_4	P_4	R^2_4	R^2_3, R^2_5	*	*	*	*	*	*	-1	1,3	P_5
L^P_5	P_5	R^2_5	*	*	1	*	*	*	*	*	4	P_6
L^P_6	P_4	R^2_4	*	1	*	*	*	*	*	*	*	P_6
L^P_7	P_4	*	*	*	*	*	*	*	*	1	*	P_5
L^P_8	P_4	*	*	*	*	1	*	*	*	*	*	P_3
L^P_9	P_2	R^1_3	1	*	*	*	*	*	*	*	0	P_6

EC Flags :
CF : Input Conveyor Flag
PF : Robot Vision Processing Flag
P-BUF : Part-in-Buffer Flag
FF : Fixture Flag

4. Knowledge Base Representation and Management

4.1 Knowledge Base Repersentation

The knowledge for a specified AJT consists of the static knowledge that is fixed according to the operation requirement of the robotic workcell. For knowledge representation of the AJT, lists in LISP are used. Since a list may have an element that is itself a list[18,19], the knowledge base for the AJT is represented by the nesting structure of lists in which lists can occur within lists. Since lists are easily understandable and implemented in terms of the binary tree, we can easily construct the knowledge base for the AJT from the specific assembly graph or tree. Dynamic information from the knowledge base for the AJT is required in the cell operation in order to represent the state of assembly progress. To store the dynamic information, global variables are used, the contents of which form into dynamic global knowledge base together with the EC flags and the current state of each state variable.

The state transition for a state variable and the corresponding link conditions are organized to construct a STM. Here, a link condition in the STM is represented as a production-rule. In particular, the slot-filler form of knowledge representation is used to store the knowledge for the STM. The slot-filler structure is a generalization of the frame concept[11] in the sense that an arbitrary degree of nesting is allowed. In Fig. 3, the structure for a rule in the knowledge base is shown.

```
(RULE ROB1-1 (if   (CF 1)
                   (PF 0)
                   (u1 1)
                   (R1 R11))
             (then   (R1+ R12))
             (time   (value 1))
             (description "[R1.1] A part is arrived at the
             input conveyor and the vision processing is not
             finished."))
```

Fig.3 A Rule Stored in the Knowledge Base for a STM.

Each rule in the knowledge base for the STM is tested by the matching algorithm in the inference engine at every period of the sampling time, and the matched rules from which useful information can be obtained in system analysis are stored into a rule history as the dynamic knowledge.

4.2 Operation Management

In order to manage the cell operation, the overall system is structured as shown in Fig.4, where the system is composed of the user interface system, the knowledge-based system, the controlled environment, the observation environment, and the simulation driver. The user interface system is responsible for the input commands from the users, which are based on the monitored outputs. The knowledge-based system is hierarchically structured from the cell operation manager(COM) to the knowledge base manager(KBM), the failure handler(FH), and the environment manager(EM), taking charge of the operations of the knowledge base for STM, the knowledge base for AJT and the dynamic global knowledge. Here, the dynamic global knowledge includes all dynamic information such as the EC flags, the failure indicators, and the current state of the state variables. The COM is responsible for managing all the parts in the knowledge-based system as a cell supervisor, and the KBM is responsible for managing all the knowledge. The FH manages the failure indicators when the failures occur. The robot and the conveyor system in the controlled environment are controlled by the EM, and the environmental information is transferred from the observation environment to the EM. The simulation driver is responsible for driving the cell simulation and composed of the exogenous event generator and the failure generator.

In reality, the EC(environment condition) information is obtained from the observation environment as the real-time information. However, it is changed according to the specified scenarios and reflected in the EC flags in simulation. The following algorithm simulates the cell operation through the knowledge base, and collects the statistics in order to evaluate the cell performance.

COM Algorithm for Simulation

Input : An AJT, an event list, and the EC information.
Output: Decisions of an operating strategy for robot manipulator, and the corresponding control inputs for the controlled environment.

[Step 1] Initialization.
Initialize the GST(global sampling time), the state variables, the EC flags, the failure indicators, and the analysis variables. Input an event list and an AJT.

[Step 2] Set the status flags related to the exogenous events.
Augment the GST by one. From the event list and AJT, set the EC flags.

[Step 3] Update the status flags related to the endogenous events.
According to the EC information, update the current EC flags at every GST; the sequence flag, the vision processing flag, the buffer flag, the part-in-buffer flag, the fixture flag, the output conveyor flag, and the inter-node time flag.

[Step 4] Set the decision variables for robot manipulators in the cell.
Make a decision for the operation strategy of robot manipulator considering the current status of cell and the assembly processing state.

[Step 5] Update the state variables.
Find the matched rules in the knowledge base for STM through the inference engine, then update the state variables from the results. Store the matched rules into the rule history. If there is no match, go to step 2; otherwise, continue.

[Step 6] Check the completion of the assigned job.
If the number of finished product is equal to the required number of product, then continue; otherwise, go to step 2.

[Step 7] Calculate the statistics.
From the time analysis variables and the matched rule history, obtain the production time, the work-in-process information, the part waiting time, and the robot idling time.

END of Algorithm

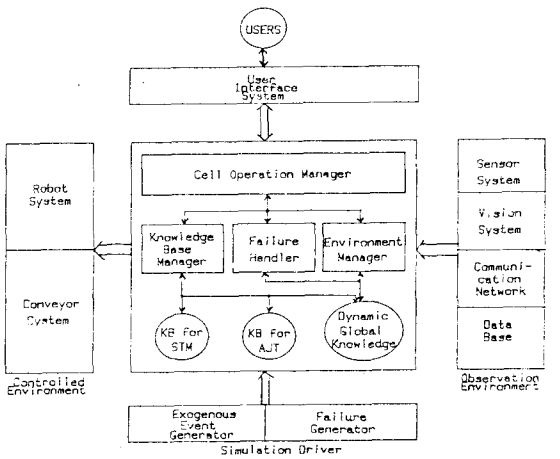


Fig.4 The System Structure for Cell Operation.

5. Simulation

Computer simulation is performed in the GCLISP on the IBM-PC, and driven in the failure-free mode. When an AJT is assigned and an event list as the part arrival information is given, we can obtain the historical statistics from the matched rule history through the simulation, and analyze them for additional information.

The input variables for simulation are divided into the policy input variables and the non-policy input variables. The policy input variables include the buffer service discipline, the assembly job tree(AJT), and the required number of finished product. The non-policy input variables include the interarrival time of part, the speed of robots, the inter-node process

time, and so forth. In the simulation, FCFS(First Come First Service) is assumed as the buffer service discipline. Also, many kinds of event lists are provided for system analysis, and a specified AJT, AJT1 is assigned. The AJT1 is assumed to be composed of three different parts, A, B, and C, the assembly sequence of which is A →B →C. The assembly time ratio of these three parts is denoted as [a:b:c].

As the performance measures for cell operation, the Total Production Time(TPT), the Average Production Time(APT), the Average Waiting Time(AWT), the Average Work-In-Process(AWIP), the Robot Idling Ratio(RIR), the Required Buffer Size(RBS), and the Throughput are introduced, where the TPT is defined as the total production time for the required number of finished product, the APT as the average production time per unit product, the AWT as the average waiting time per unit part in the buffer, the AWIP as the average process time per unit part, the RIR as the percentage ratio of the robot idling time to the TPT, the RBS as the minimum of required buffer size, and the Throughput as the average number of products completed per unit GST(global sampling time).

If the part movement and the operation of two robots are considered together, a timing graph for the system described in Fig.1 can be drawn as shown in Fig.5, where the inter-node process time is defined by the link indices depicted in Table 3. In Table 3, TF1_C depends upon the speed of R1, TF2_C and TF3_C depend upon the speed of R2, and AJT-time depends upon both the assembly complexity and the speed of R2. In the simulation, the consuming times for TF1_C, TF2_C, and TF3_C are assumed as one GST when the robot manipulator is driven at the full speed, that is, at 100 percentage speed. The consuming time for AJT-time is assumed as one GST when the assembly time rate is 1 and the robot R2 is driven at the full speed. This assumption is only for the simulation and can be relaxed as desired.

In this paper, two cases are described. In CASE 1, the AJT1 is assigned and the comparison of performance measures is discussed according to the various interarrival time. In CASE 2, the AJT1 is also assigned, and the speed bounds of two robots and the sensitivity of the Throughput to the speeds of two robots are discussed. These two cases are described briefly in the following.

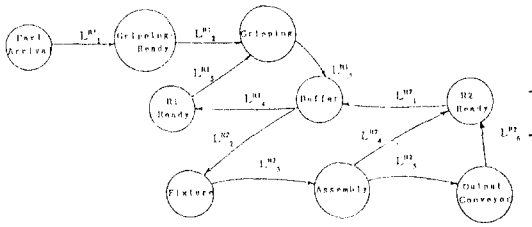


Fig.5 A Timing Graph.

Table 3. Definition of the Link Index

UNIT:Global Sample Time

Link Index	Consuming Time[GST]	Moving Robot	Descriptions
	1		vision processing time
	2	R1	gripping process
	TF1_C	R1	stacking in the buffer
	1	R1	rapid return to the R1 ready
	1	R1	going to the gripping-ready
	1	R2	going to the buffer
	TF2_C	R2	picking up a part
	TF3_C	R2	preparing for the assembly
	AJT-Time	R2	assembly except the final part
	AJT-Time	R2	assembly for the final part
	2	R2	rapid return to the R2-ready

CASE 1:

We assume that the speed of R1 is three times faster than that of R2, and the speed of R1 is set to the maximum. The assembly time ratio of three parts is assumed as [1:2:2], the sequence of part arrival is assumed to be in order compared with the assembly sequence, such as A B C A B C A ..., and the required number of finished product is set to three. In Table 4, the analysis data for comparison of performance measures are collected according to the variation of interarrival time. The graphic description of several performance measures is shown in Fig.6.

As indicated, the APT and the RIR are found to be the minimum when the interarrival time is set to 12. When the interarrival time is shorter than 12, the AWT and the AWIP increase as the decrease of the interarrival time, and the APT and the RIR fluctuate slightly. The increases of the AWT and the AWIP are due to the high frequency of the part arrival, and the fluctuations of the APT and the RIR are due to the existence of the local minimum. Although the minimum bound of the interarrival time is not indicated in Table 10, it is found to be 5 for effective operation of the cell. When the interarrival time is longer than 12, the APT and the RIR increase gradually as the increase of the interarrival time, and the AWT and the AWIP come up to the minimum constant values. The lower limit of the AWT and the AWIP comes from the excessiveness of the interarrival time. Consequently, we can reasonably select the best interarrival time for the cell operation and the required buffer size under the given conditions.

CASE 2:

In CASE 2, the same AJT and the event list as that in Table 4 are used, and the interarrival time is fixed at 9. When the speed of two robots is set to 100, 50, 33, 25, 20, or 17 percentage of the full speed, we can find the speed bound from the performance measures.

In Fig.7 and Fig.8, the TPT and the Throughput are depicted with the speed variation of two robots. The TPT and the Throughput can not be obtained when the speed of R1 is less than 20 percentage of the full speed. The reason is that the speed of R1 has the constrained range which depends upon the interarrival time, the robot vision processing time and the speed of the input conveyor. In Fig.8, the Throughput is scaled up as much as 100 times, and is found to be more sensitive to the speed of R2 than that of R1.

Table 4. Comparison of performance measures according to the variation of interarrival time.

Interarrival Time [GST]	TPT [GST]	APT [GST]	AWT [GST]	AWIP [GST]	RIR [%]	RBS [SIZE]	Throughput [PRODUCT]
9	119	39.67	22.00	33.11	34.0	3	0.0252
10	120	40.00	18.22	29.33	34.6	3	0.0250
11	118	39.33	13.33	24.44	33.4	2	0.0254
12	117	39.00	8.89	20.00	32.9	1	0.0256
13	118	39.33	5.22	16.33	33.4	1	0.0254
14	126	42.00	5.00	16.11	37.6	1	0.0238
15	134	44.67	5.00	16.11	41.4	1	0.0224

6. Conclusion

This paper presented a modelling methodology for a robotic workcell, which is based on the concept of the state variable, and a method for managing the cell operation through the knowledge base. The state variables were defined as the representative states of the cell element for modelling, which was innovative compared with the existing concept. Since the transition of each state was related to several variables, the relationships of states were described implicitly, and the state transition map(STM) was also described for the corresponding state variable. From the STMs and the AJTs, we obtained the knowledge bases for the cell operation. The

system structure for managing the cell operation was discussed through the knowledge base, and a simulation algorithm was presented. In simulation, the various performance analysis was done with the variation of system input variables. In particular, the variations of the performance measure were shown according to the various interarrival time in the event list, and the speed bounds of two robot manipulators.

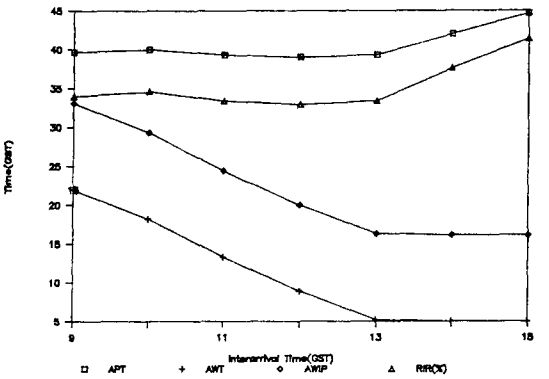


Fig.6 Comparison of four Performance Measures in Table 4.

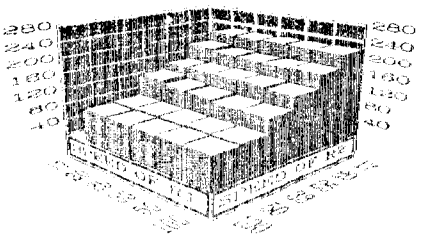


Fig.7 Total Production Time with the Speed Variation of two Robots.

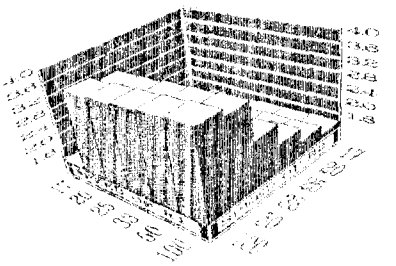


Fig.8 Throughput with the Speed Variation of two Robots.

REFERENCES

[1] David D. Yao and J. A. Buzacott," Modelling the performance of flexible manufacturing systems", *Int. J. Prod. Res.*, Vol.23, No. 5, pp. 945-959 (1985)

[2] Robert L. Aveyard,"A Boolean model for a class of discrete event system", *IEEE Trans. on System, Man, and Cybernetics*, Vol.SMC-4, No. 3, May (1974)

[3] M. Kamath and N. Viswanadham," Application of Petri net based models in the modelling and analysis of flexible manufacturing systems", *Proc. of the 1986 IEEE of Int. Conf. on Robotics and Automation*, pp. 312-317 (1986)

[4] N. Viswanadham and Y. Narahari, " Colored Petri net models for automated manufacturing systems", *Proc. of the 1987 IEEE Int. Conf. on Robotics and Automation*, pp. 1985-1990 (1987)

[5] Carolyn L. Beck and Bruce H. Krogh," Models for simulation and discrete control of manufacturing systems", *Proc. of the 1986 IEEE Int. Conf. on Robotics and Automation*, pp. 305-310 (1986)

[6] J. S. Ostroff and W. M. Wonham," State Machines, Temporal Logic and Control: a Framework for Discrete Event Systems", *Proc. of the 28th Conf. on Decision and Control*, pp. 681-686 (1987)

[7] David H. Ben-Arieh, Colin L. Moodie, and Chi-Chung Chu," Control Methodology for FMS ", *IEEE Journ. of Robotics and Automation*, Vol.4, No.1, pp. 53-59, Feb. (1988)

[8] A. Kusiak and A. Villa, " Architectures of expert systems for scheduling flexible manufacturing systems", *Proc. of the 1987 IEEE Int. Conf. on Robotics and Automation*, pp. 113-117 (1987)

[9] Bernard P. Zeigler, *Theory of Modelling and Simulation* (John Wiley & Sons, 1976)

[10] Y. C. Ho, " Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems ", *IEEE Trans. Automat. Contr.*, Vol. AC-32, pp. 563-572 (1987)

[11] A.C. Kak, K.L. Boyer, C.H. Chen, R.J. Safranek, and H.S. Yang," A Knowledge-BasedRobotic Assembly Cell", *IEEE Expert*, pp. 63-83, Spring (1986)

[12] Ricardo F. Garzia et al., " Discrete-Event Simulation ", *IEEE Spectrum*, pp. 32-36 (1986)

[13] David Ben-Arieh et al., " Knowledge Based Control System for Automated Production and Assembly", *Toward the Factory of the Future*, Springer-Verlag (1985)

[14] Y. C. Ho, "Scanning the Issue", *Proc. of the IEEE*, pp. 3-6, Jan. (1989)

[15] Steven H. Kim, " An Automata - Theoretic Framework for Intelligent Systems", *Robotics and Computer-Integrated Manufacturing*, Vol. 5, No. 1, pp. 43-51 (1989)

[16] Dae-Won Kim, Myoung-Sam Ko, and Bum-Hee Lee, " An Approach to Modelling for Operation Management of Robotic Assembly Cells through Knowledge Base ", *Proc. of the 28th SICE Annual Conf.*, Vol.II, pp. 961-964, Matsuyama, Japan (1989)

[17] J. Martinez, H. Alla, and M. Silva, " Petri Nets for the Specification of FMSs", *Modeling and Design of Flexible Manufacturing Systems*, pp. 389-406 (1986)

[18] V. Daniel Hunt, *Artificial Intelligence & Expert Systems Source Book* (Chapman & Hall,1986)

[19] *Golden Common LISP* (Gold Hill Computers, Inc., 1983)

[20] H. Kobayashi, *Modeling and Analysis*, Addison-Wesley Pub. Co.(1978)