A Study on Controls System of FMS Using Petri net Modelling

페트리네트 모델링을 이용한 FMS 통제 시스템에 관한 연구

김 만 균*
Kim, Man-Kyun
함 효 준**
Hahm, Hyo-Joon

Abstract

최근 제조업체들은 고객의 다양한 요구와 제품의 Life Cycle 단축에 따른 제조환경의 변화에 직면하고 있다. 이러한 변화에 능동적으로 대처하기 위한 수단으로 효율성과 유연성이 크게 부각된 유연생산시스템(FMS)의 역할은 대단히 크다. 따라서, 본 연구에서는 유연생산시스템을 평가하고 모델링하는 도구인 Petri net 모델링 기법을 이용하여 효율성이 뛰어난 흐름생산방식을 적용한 복잡한 유연생산 시스템의 통제시스템 모델 설정과 시스템정지(deadlock) 예방에 대한 분석을 하였으며, AGVs의 셀이동 생산시스템을 제어할수 있는 FMS의 계층적 TPN 모델링을 구축하였다.

1. Introduction

In the present competitive market, a greater flexibility is required for the automated manufacturing system in order to respond to the needs of different product types from the market. But the greater flexibility implies a greater complexity in the manufacturing processes. Today's manufacturing environment involves great flexibility and fast change over in manufacturing systems. The control of these systems is complicated and is similar to the control of discrete even dynamic systems. Modelling and evaluating these systems become essential to the investigation of manufacturing operation and to the design of manufacturing systems. Many companies have recently responded to competition by investing in a FMS(Flexible manufacturing system) and emphasizing quality, delivery and flexibility to meet customer's requirement in their objectives.[1] The control of systems needs to be accurate and first enough to correct the problems occurred during system operation in order to avoid any undesired downtime. In fact, most of the research attention has been devoted to the hardware requirements of flexibility with little concern for the system management and control mechanisms that are necessary for its realization. We focus on the study of Petri nets(PNs) which are gaining more popularity in the analysis of the FMS performance such as scheduling, throughput, boundness and deadlocks. PNs are a suitable tool for analysing these problems and a modelling tool, will be used investigate the composition of a manufacturing system model because of the tool's ability to represent the concurrent operations in a complex manufacturing system.

---

* 대우그룹 회장비서실

** 아주대학교 산업공학과

2. Definition of PNs Approach

PNs(a powerful tool for modelling and analysis of concurrent systems) were originally introduced by Petri in his Ph. D. dissertation(1962) where he gave a theory of communication between asynchronous components of a computer system. Since then considerable work has been done in both the theory and application of PNs, mainly in the areas of computers science (software and hardware) communication protocols, operating systems and manufacturing systems. PNs have proven to be valuable tools for analyzing the dynamic behavior of manufacturing systems, in particular of FMSs. On the theoretical side, the emphasis was on extending the modelling power of PNs and on the formulation of a mathematical framework in which different properties of Petri net can be analysed. However, because of their high complexity, most Petri net models of FMSs cannot be studied analytically. Indeed, practically useful theoretical results can only be obtained for special cases of PNs. Therefore, on the application side, the application of PNs to the analysis of the behavior of FMSs most of the requires the development of sophisticated simulation programs.

2.1 PNs terminology

A Petri net structure is composed of four parts; a set of places, a set of transitions, an input function, an output function defined as follows;

$PN = < P, T, A, M >$

where, $P = \{ P_1, P_2, \cdots, P_n \}$ the set of *n* place,

$T = \{ T_1, T_2, \cdots, T_m \}$ the set of *m* transition. A is a mapping,

$A : (P \times T) \cup (P \times T) \longrightarrow N$

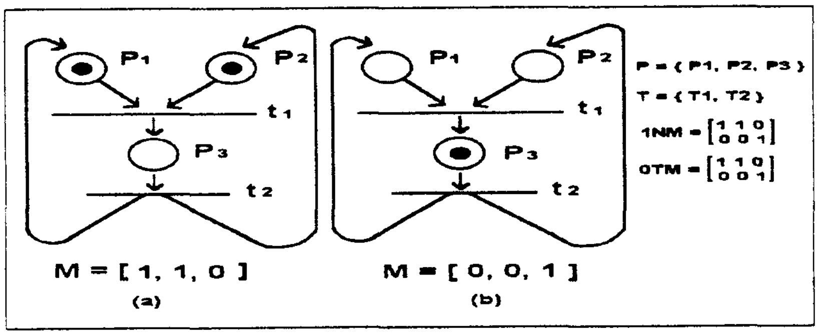

Fig 1. A Petri net. (a) Before firing of t_1. (b) After firing of t_1.

the set of weighted arcs (N is the set of all Positive integers). M is a mapping

$M : P \longrightarrow N$, the marking of the PN.

The input function and the output function are used to linked set of places and set of transitions in a PN.

Graphically, a PN is represented (Fig. 1) as a directed bipartite graph in which places are represented by circles, transitions by bars and arcs by directed edges. A marking of a PN indicates the distribution of tokens in its places. The number of tokens in a given place 'P' is the marking of that place. Tokens are represented by black dots in a PN graph. In Fig. 1, n=3, m=2, $T=\{ T_1, T_2 \}$, $P=\{ P_1, P_2, P_3 \}$ and $M=\{ 1, 1, 0 \}$.

2.1.1 Input and output places

All place that have arcs leading into(out of) a transition are said to be that transition's input(output) places. In Fig. 1, T1 has to input places, P1 and P2, and one output place P3

2.1.2 Input and out matrix

Consider a PN with n places and m transitions. The input matrix (INM) of a PN is an $n \times m$ matrix defined as;

INM(Ti, Pj) = W if Pj is an input place of Tj,
where, W is the weight of the arc connection Pj and Ti.
= 0 otherwise

The output matrix(OTM) is an $n \times m$ matrix defined as;

OTM(Ti, Pj) = W if Pj, is an output place of Ti,
where, W is the weight of the arc connecting Pj and Ti.
= 0 otherwise

2.1.3 Execution of a PN

The execution of a PN proceeds as follows. A transition is said to be enabled it each of its input places has a number of tokens at least equal to the weight of the arc leading form that place to the transition. an enabled transition fires by removing from each of its input places the number of tokens equal to the weight of the corresponding incoming arcs and depositing tin each of its output places the number of tokens equal to the corresponding weights of output arcs. Fig. 1(a) and (b) show the firing of a transition

2.1.4 Timed PNs

A timed PN(TPN) associates deterministic firing durations to its transition. The TPN is defined as;

TPN = < P, T, A, M, D>

where, $D : T \rightarrow \{O, R^{+}\}$ is the set of firing durations, and $\{O, R^{+}\}$ is the set of all positive read number

The execution of a TPN is different form that of an untimed PN. In an untimed PN, an enabled transition fires instantaneously. The removal of tokens form the input places and the deposition of token is the enabled transition's output places occurs at the completion of its firing duration. The state of the TPN is defined by the instantaneous description(ID) of the net. An ID of a TPN is a four-tuple,

ID= <M, F, R, T>

where, M is a marking function, $M : P \rightarrow \{0, 1, 2, \cdots\}$;
F is a enabled transition function, $F : T \rightarrow \{0, 1\}$;
R is a remaining firing time function, $R : T \rightarrow \{0, R^{+}\}$;
T is a cumulative time function.

2.1.5 PNs for modelling

It is worth while to summarize briefly the features which make PNs particularly suitable in modelling complex concurrent system. PNs represent parallel process and synchronization events between them very easily, without the need of over-specifying the behaviour of the global system; PNs are non-deterministic, in the sense that the sequence of transitions firings is not specified in the model and can be constrained or fully determined by external policies, added to the model itself; There have been developed more powerful variations of PNs, namely coloured, predicate / transitions and timed PNs and this allows choice of the most suitable modelling technique according to the particular characteristics of the system under examination. The correspondence still holds for coloured PNs and predicate transition PNs; in the former case data are added to facts, like colors are added to tokens; in the latter case more pattern are added to the left-hand-side(LHS) of rules, as predicated are associated with transitions. Table 1 Points out the correspondence between a real production system, the PN-based model representing it and the rule-based software simulator.

Real system (Production system)	PN (Model)	Rule-based system (Simulator)
Event or operation	Transition	Rule
Structure	Complete net	Base of rules
State of unit or an object in the system	Token	Fact
Global state	Marking	Base of facts
Evolution	Transition firings	Rules execution

Table 1. Correspondence between the real production system and the PN model

3. Analysis of Manufacturing Systems

3.1 A Flow-line system for FMSs

In flow line a part goes through an ordered sequence of operations. Each operation is performed by a dedicated machine or workstation. In a flow line FMSs, while the part is still restricted to a fixed sequence of operations. This situation is explained in question as follows;

① What is the effect of varying flexibility in either the part production requirements or the machines processing capabilities on system performance measure such as part flow time, levels of work-in-process inventory, production rate, and the number of bottlenecks.

② How much of the flexibility should be allowed and / or can be afforded.

③ How can optimal levels of flexibility be determined so that any associated costs and benefits are appropriately traded off.

In the following discussion as above, insights as to how each of these questions can be answered are presented based on a series of simple probabilistic and empirical models.

Let M_1, M_2, …… M_n represent the sequence of operations carried out along the flow line FMS. Each of these operations is associated with a set of workstations on which it can be performed with each of the workstation possibly having a different processing cost and processing time. Let τ_f represent the total time a part spends on the line or simply part flow time. The value of τ_f can then be calculated as;

$$\tau_f = \sum_{i=1}^{n} \tau_w(i) + \sum_{i=1}^{n} \tau_p(i) = \sum_{i=1}^{n} \tau_s(i) \quad \text{-----------------------} \quad (1)$$

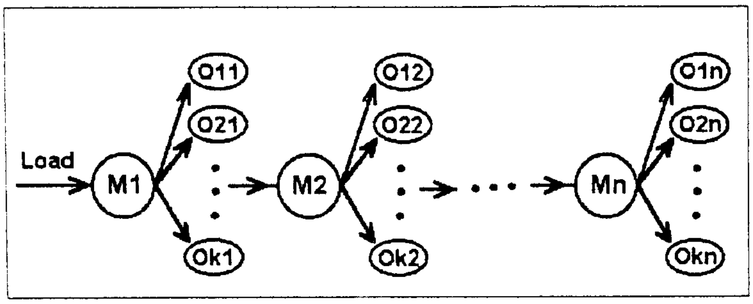

Fig 2. A flow-line FMS

where, $\tau_w(i)$: The waiting time for the selected workstation of operation *i*.

$\tau_p(i)$: The processing time on the selected workstation for operation *i*.

$\tau_s(i)$: The sum $\tau_w(i)$ + $\tau_p(i)$ and sojourn time at production stage *i*.

Then, assuming that management is interested in minimizing part sojourn time at each stage of line, the expected total flow time can be calculated as;

$$E(\tau_f) = \sum_{i=1}^{n} E(\min(\tau_s(M_i, O_{1_i}), \tau_s(M_i, O_{2_i}), \cdot\cdot\cdot\cdot\cdot\ \tau_s(M_i, O_{k_i}))) \text{ ---- } (2)$$

Where, O_{ji} : The *j*th machine at stage *i* and *ki* the number of machines at state *i*. Sojourn time is minimized at each of the stage by assigning parts to the workstation with the shortest expected sojourn time. The value of $E(\tau_f)$ can be shown to be a non-decreasing function of the number of workstations at each stage, that is flexibility. In fact, under most conditions $E(\tau_f)$ is a strictly decreasing function of flexibility is maximum. The marginal effect of flexibility on performance can also be shown to be an increasing function of variability in the sojourn time. Let the sojourn times at each stage be independent and identically and uniformly distributed with upper and lower bounds *Ui* and *Li* respectively. The upper limit *Ui* is used to represent the maximum allowable queue size in front of each workstation while the lower bound *Li* represents the minimum processing time in which operation *Mi* can be performed. The expected sojourn time at a stage *i* can then be found as;

$$E(\tau_f) = Li + \frac{(Ui - Li)}{(Ki + 1)} \text{ --------------------------------- } (3)$$

Where, *ki* : The number of machines that can perform operation *Mi*.

Clearly, The above expression for sojourn time consists of tow factors, the first is constant and represents the minimal achievable sojourn time, while the second is variable and is a decreasing function of flexibility. In the case where flexibility is zero (*i.e.* ki=1), sojourn time is simply the mean of the distribution $(Ui+Li)/2$ while in the case where flexibility is maximum, *i.e.* when $ki \to \infty$, expected sojourn time is reduced to the lower bound *Li*. This means that a part never waits for a workstation and is always assigned to the workstation with the shortest processing time. If the width of the distribution $(Ui-Li)$ is taken as a measure of the degree of variability in the sojourn time, then it can be said that the need for flexibility increased with increasing variability. Alternatively, flexibility gas an effect on performance only in the presence of some degree of variability (*i.e.* when $Ui \neq Li$). Note also that the relationship between flexibility and sojourn time is of the diminishing return type. Much of the reduction in sojourn time is achieved with relatively little flexibility with further increased bringing only marginal improvements. If the processing time for each of the operations are constant and/or significantly smaller than the waiting time, then the two factors, Li and $(Ui-Li)/(ki+1)$, can be viewed as corresponding respectively to processing time and waiting time. Waiting time is in this case a decreasing function of flexibility which reduces to zero when flexibility is maximum. Reducing waiting time should also result in a reduction of the level of work-in-process inventory and the possibility of a bottleneck.

3.2. PN approach for FMS

Petri net mothed is a graphical can mathematical modelling tool, PNs can be used to describe and study systems with characteristics such as concurrent, asynchronous, distributed, parallel, non-deterministic, and/or stochastic operations. This system consists of three machines M_1, M_2 and M_3, and two buffers B_1 and B_2, with capacity n_1 and n_2, respectively. Parts will be processed by these three machines with input from the loading station and output to the unloading station. A PN representation of the system operation is shown Fig 3. The notation in the net is explained by the following notes;

JR : job ready f M_1
Fj : Occupied slots in buffer Bj
MRi : Machine i ready to process
Ej : empty slots in buffer Bj
Pi : machine i processing to job
Tk : the processing operation

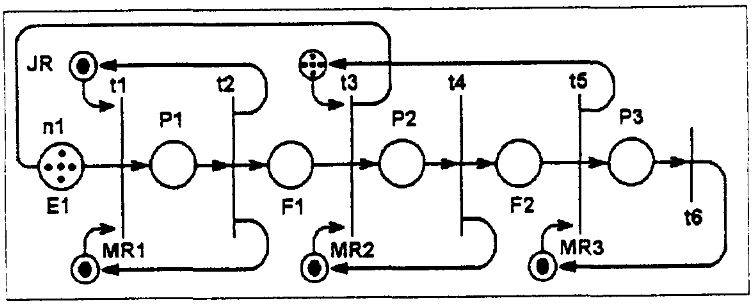

Fig 3. PN of the system

The purpose of decomposing a PN is to divide a system net into several subnets so that each subnet can be executed independently. Checking each column, transitions t_1, t_3 and t_5 have synchronization operations. Transition t_1 is with MR_1, JR, and E_1 (each one has a negative value). Transition t_3 is with MR_2, E_2, and F_1. Transition t_5 is with MR, and F_2. They are listed in the synchronization table of Table 1. At step 3. no row has more than one negative sign, so no conflict occurs among the transitions and the system net is a marked graph. The incidence matrix of this system is shown below. Let us begin with the first column of the incidence matrix. Since t_1 column has more than one negative sign.

Entry (MR_1, t_1) is chosen first. The following stages show the entry selection process.

① Select the entry (MR_1, t_1), Where $i = MR_1$, $j = t_1$;

t	P	Subnet 1	2	3	4	5	6
t_1	MR_1	×					
	JR		×				
	E_1					×	
t_3	MR_2			×			
	E_2						×
	F_1				×		
t_5	$MR3$			×			
	$F2$						×

Table 1. The marked graphs synchronizations table

② Select (MR_1, t_2), Where $i = MR_1$, $k = t_2$;

③ Select (P_1, t_2), Where $i = P_1$, $k = t_2$;

④ Select (P_1, t_1), Where $i = P_1$, $r = t_1$;

Factors	t_1	t_2	t_3	t_4	t_5	t_6
MR_1	-1	1	0	0	0	0
MR_2	0	0	-1	1	0	0
MR_3	0	0	0	0	-1	1
JR	-1	1	0	0	0	0
E_1	-1	0	1	0	0	0
E_2	0	0	-1	0	1	0
F_1	0	1	-1	0	0	0
F_2	–	0	0	1	-1	0
P_1	1	-1	0	0	0	0
P_2	0	0	1	-1	0	0
P_3	0	0	0	0	1	-1

After ④ stage is completed, r is equal to j and each row and each column has only one positive and one negative sign. Therefore, the formation of one subnet is complete. This subnet is denoted as subnet 1 in the synchronization table. The related information of synchronization is listed in the table with conjunction of t_1, MR_1 and subnet 1. The next starting point is (JR, t_1) which has a negative sign and is in t_1 column. The subnet 2 can be formed with two rows (JR and P_1) and two columbus (t_1 and t_2). The third and last starting point in column t_1 is (E_1, t_1).

3.3 Control system description and modelling

The layout of a FMS based on a case study[16] is considered for the present investigation. It consists of three machining centres and a pallet station connected by (Automated Guided Vehicles) AGVs, as shown in Fig. 4. Local storage is provided at each machining centre and at the pallet station to serve as the buffer for input / output. Processing times are considered to be deterministic due to thehigher degree of automation involved in FMSs. In the PN modelling of a system, the following modelling conventions are used;

① Places represent conditions in the system.(machine free, job available in the buffer)

② Transitions represent activities occurring in the system.

③ Token represent the true value of conditions and/or physical objects such as pallets, machines and AGVs. pallets are assumed to e different for different product types. It is also assumed that only one pallet is employed for all machine operations on a given part

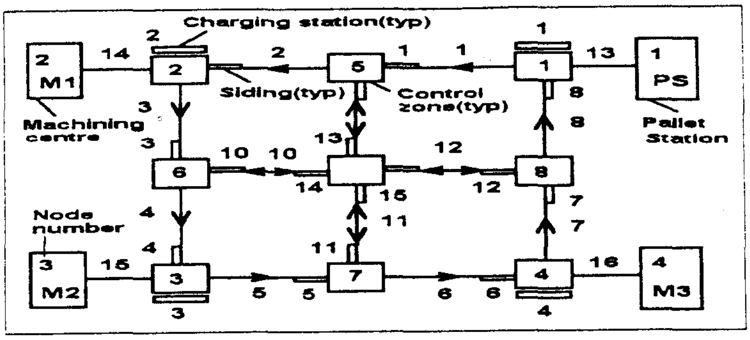

Fig. 4 Layout of the FMS

type. A pant on a pallet after completing its processing sequence is moved to the pallet station, where it is unloaded. The AGV track layout consisting of both bidirectional and unidirectional segments is shown in Fig. 4. The main advantage of the PNs is that they ofter modularity in the modelling, which makes the modelling of complex systems easier.

4. Control system modelling for FMS

The manufacturing system control includes machine control, manufacturing cell control, shop floor control and factory management control. A FMS processes a manufacturing plan consisting of a set of job. To understand the complex interaction of job, workstation, and transporter for a given manufacturing plan, replicating system behaviour under varying operating conditions. By analysing the control model off-line, properties leading to deadlocks are detected and the throughput rate is estimated. such feed-back information is useful to the production engineer designing controls for the FMS since it reduces the overall setup time. Mathematical models for deadlock detection and avoidance in computer operating system have been studied extensively. These deadlock occur when multitasks are performed concurrently in order to increase resource utilization. However, such models are not directly applicable to manufacturing systems, where resources are allocated to a process in a sequence specified by the process plan.

4.1 Control System Structure

The control model presented consists of algorithms for generating a PN representing the process flow, detecting deadlocks, and evaluating the performance of a deadlock-free net. The PN is generated in matrix definitional form(MDF) using the process plans as input. the PN is checked for deadlocks at each state generated by the state equation. Performance is evaluated for a deadlock-free PN by computing mean cycle time for the flow of tokens. Throughput rate is computed from the mean cycle time. The FMS consists of workstations having a single machine, or a machine. robot unit. These are called the subsystem of a FMS. Subsystems are often required for assembling jobs independently. Similar applications are noted on a machining workstation, and a machining cell consisting of CNC machine and robot. The FMS consists of a set of workstations arranged around a common transporter(Fig.5). The FMS assembles a job sequentially on a set of workstations specified in the process plan. Due to the inherent difference in the functioning of the subsystems of the FMS, separate control models have been developed for each. The control model basically consists of three modules, each having a function as noted below;

- ▶ Module 1. The manufacturing plan specified by a job schedule is transformed into a PN in MDF. The PN represents the process sequence of jobs, along with operation time and buffer capacity at each workstation.
- ▶ Module 2. The PN is analysed for deadlocks by determining a transition sequence which allows the PN to reach its initial state, or some predefined goal state.
- ▶ Module 3. The deadlock-free PN is evaluated for performance by computing mean cycle time for tokens to flow in the operation circuit.

4.2 Control system Modelling

PN modelling is a powerful tool for modelling and analysis of asynchronous, concurrent systems which are different to model using simulation and queuing networks. It is convenient to model non-product from characteristics, such as multiple workstation holding, blocking, synchronization, and prioritization, which are common in a FMS[19]. By representing the control model as a PN, properties leading to deadlock are analysed, and performance measures evaluated by assigning a firing time to each transition. In a PN, deadlock are detected when a sequence of transition firings results in a state from which no further transition can be fired. In a deadlock, no set of places having input arcs to transitions can enable any of the transitions. In a FMS, this state occurs when restriction on buffer capacity prevents flow of a job to the next workstation. Researchers have modelled automated manufacturing systems using PNs. When changes occurs in the manufacturing plan, models need updating, and re-analysis for deadlock and performance. The analyst is required to monitor the model continuously during real-time applications, which is time consuming for large size systems. Deadlock in a PN are commonly analysed using the reachability tree technique. The reachability tree is generated from the graphical from of PNs which allows the analyst to trace the flow of tokens through each path for detecting deadlock conditions. An exhaustive search of all paths of the tree represents all possible sequence of transition firings, it could become an infinite tree even for a finite reachability set. The PN of a manufacturing plan is evaluated for performance by assigned to each transition of computing the performance measures[1]. Cycle time for a token to return back to its initial place is used as a measure of performance. This method is applicable to small size PNs. Large PNs are evaluated by software package using Markov chain analysis methods. Once again, the PN in graphical form is generally used as input to the package. PN models in graphical from offer a good visual description, and ar convenient for modelling controls of a FMS. Such models become unmanageably complex to develop and analyse for large system. Researchers have developed a simplified approach of recasting a PN in matrix from. Deadlock detection is efficiently carried out on this from of PN using matrix algebra. Control models have been developed generation and analysis procedure.

5. Conclusion

Based on fairly simple models, several fundamental results regarding manufacturing flexibility using PN have been described in this paper. This methodology is based on the TPN which is developed by using the concept of recursion. Flexibility can be an effective mechanism in dealing with the disruptive potential of variability in a dynamic environment.

Under conditions of variability it is possible to establish direct relationships between flexibility and system performance. The advantages of the PN include its ease of understanding and its readability. The proposed methodology is illustrated with the help of a FMS consisting of three machine centres and a pallet station connected by AGVs. The control model generates a PN for a manufacturing plan processed by subsystems of the FMS. The PN is checked for deadlock and evaluated for performance. Deadlock detected in the PN can be avoided by reallocating buffer capacities at critical workstation. Controls systems of FMS play an important role in this transition. For future research an PN, one can try to combine PNs and real-time control system.

Reference

[1] 김만균, "*자동차 생산을 위한 유연생산 시스템 구축에 관한 연구*", 한양대, 석사논문, 1992.

[2] 안재홍, 노인규, "퍼트리 네트를 이용한 FMS 스케줄링에 대한 발견적 해법", *한국경영학회지*, 제21권 제2호, pp111-124, 1996.

[3] 이강수, "퍼트리 네트의 분류법", *한국정보과학회지*, 제21권 제8호, 1379-1389, 1994.

[4] 김고중 외2명, "Petri-Net을 이용한 FMS 제어 및 모니터링", *한국산업공학회지*, Vol.21 No. 1, pp 137-151, 1995.

[5] 이광식 외1명, "추계적 페트리네트를 이용한 대기시스템의 제어모형" *산업공학회지*, 제8권 제2호, pp161-169, 1995.

[6] 이강수, "*A study on Petri nets and its simulation method*", 서울대, 석사학위논문, 1982.

[7] Kim, Man-kyun & Hahm, Hyo-joon, "A study on optimal evaluation model for FMSs using Activity-Based Costing", *Journal of the society of Korea Industrial and Systems Engineering*, Vol. 19 No. 38, Jun. 30, pp. 147-148, 1996.

[8] Johns, A., and McLean, C., "A proposed hierarchical control model for automated manufacturing system", *Journal of manufacturing systems*, 5(1), pp15-25, 1986.

[9] Jafari, M., "An architecture for a shop-floor controller using colored petri nets", *International Journal of flexible manufacturing system*, 4, pp159-181, 1992.

[10] Peterson, J.L., "*Petri Net Theory and Modeling of systems*", Englewood cliffes NJ, Prentice-Hall, 1981.

[11] Narahari, Y., & viswanadham, N., "A petri net approach to the modelling and analysis of flexible manufacturing systems", *Annals of operations Research*, 3, pp449-472, 1985.

[12] Teng, S., and Black, J., "Petri nets for manufacturing cell control operations", *Manufacturing Review*, 5(2), pp127-136, 1992.

[13] Teng, S., and Black, J., " Cellular manufacturing systems modelling; the petri net approach," *Journal of manufacturing systems*, 9(1), pp45-54, 1990.

[14] Park, T. Y., "*Modular approach to Petri net modeling of automated manufacturing system*", Ph. D. dissertation, Dep't of Inderstrial Engineering, KAIST, 1992.

[15] Jeng, M. D., "Modular synthesis of Petri net for modeling flexible manufacturing system", *The International Journal of flexible manufacturing system*, Vol.7, No.3, 1995.

[16] Ozden, M., " A simulation study of multiple load carrying AGV in FMS", *International Journal of Production Research*, Vol.26, pp1353-1366, 1988.

[17] boucher, T.O., Jafari, M.A., and Meredith, G.A., "Petri net Control of an Automated Manufacturing Cell", *Computer Inderstrial Engineering* Vol. 17(4), pp459-463, 1989.

[18] Wu, F.C., "*Study of Petri Net and its Application to FMS*", Msc. Thesis, Indu. Eng. Dep't, National Tsing Hua University, Taiwan, 1991.

[19] Narahari, Y. and Viswandham, N., "A Petri net approach to the modeling and analysis FMS", *Annals of Operation Research*, Vol 4(5), pp119-131, 1986.