

## Integrated analytical models for flexible manufacturing systems

Y NARAHARI, N VISWANADHAM, C R MEENAKSHISUNDARAM  
and P HANUMANTHA RAO

Department of Computer Science & Automation, Indian Institute of  
Science, Bangalore 560 012, India

**Abstract.** Product form queueing networks (PFQN) and generalized stochastic Petri nets (GSPN) have emerged as the principal performance modelling tools for flexible manufacturing systems (FMS). In this paper, we present integrated PFQN-GSPN models, which combine the computational efficiency of PFQN and representational power of GSPN by employing the principle of flow-equivalence. We show that FMS that include non-product form characteristics such as dynamic routing and synchronization can be evaluated efficiently and accurately using the integrated models.

**Keywords.** Integrated analytical models; flexible manufacturing systems; product form queueing networks; generalised stochastic Petri nets.

### 1. Introduction

Flexible manufacturing systems (FMS) (Ranky 1983) represent an important recent development in manufacturing automation, on account of improved productivity, quality and resilience to demand fluctuations. An FMS is essentially a discrete event dynamic system (Ho 1987) and therefore Markov chains constitute the basic stochastic model of FMS. Queueing networks (QN) (Buzacott & Yao 1986) and generalized stochastic Petri nets (GSPN) (Viswanadham & Narahari 1988) have emerged as the principal analytical modelling tools for FMS. Product form queueing network (PFQN) models of FMS have gained prominence because of their computational efficiency, however, they cannot capture FMS characteristics such as priority decision rules, blocking of resources, dynamic routing, multiple resource holding, and synchronization. To model these (non-product form) features, researchers have often employed approximations (Chandy & Sauer 1978). GSPN provide an exact way of modelling non-product form features and as natural models of concurrency and synchronization, GSPN have emerged as a strong alternative to PFQN in many modelling situations. However, as FMS grow in size and complexity, the GSPN models lead to large state space, making them intractable.

The concept of flow-equivalence in queueing networks (Chandy *et al* 1975) can be used in building realistic and efficiently solvable performance models that include two parts: one a queueing network part and the other a Petri net (PN) part. In particular, we shall consider models that include a PFQN part and a GSPN part. Such

models were first considered by Balbo *et al* (1986) to evaluate computer systems with non-product form features such as priorities, blocking, forking and joining, dynamic routing, and multiple resource holding. Meenakshisundaram (1990) and Ramesh (1990) have looked at the application of this modelling technique in the context of parallel and distributed computing systems. In this paper, we bring out the use of such models, which we shall refer to as integrated PFQN-GSPN models (IQP models for short) in the FMS context.

In the next section, we introduce the notation of GSPN and explain the basic mechanism of IQP modelling. In §3, we present two illustrative examples. All the examples are concerned with the closed central server models of FMS. In example 1, we present an IQP model of a central server FMS where parts are routed to machines dynamically using the SQR (shortest queue routing) policy. In example 2, we consider a central server FMS in which fresh raw parts are released into the FMS only when an explicit demand is pending for a finished part and develop an IQP model. In §4, we discuss the accuracy of IQP modelling and show with numerical results that the performance estimates are remarkably accurate.

## 2. Integrated PFQN-GSPN modelling

Here, we first introduce GSPN and then explain integrated PFQN-GSPN models.

### 2.1 Generalized stochastic Petri nets

In the following definitions (Murata 1989),  $N$  denotes the set of non-negative integers.

#### DEFINITION 1

A Petri net  $G$  is a four-tuple  $(P, T, IN, OUT)$  where

$$\begin{aligned} P &= \{p_1, p_2, p_3, \dots, p_n\} \text{ is a set of places,} \\ T &= \{t_1, t_2, \dots, t_m\} \text{ is a set of transitions,} \\ P \cup T &\neq \emptyset; P \cap T = \emptyset; \end{aligned}$$

$IN: (P \times T) \rightarrow N$  is a function called the input function and associates directed arcs from places to transitions, and

$OUT: (P \times T) \rightarrow N$  is an output function that defines directed arcs from transitions to places. ●

Pictorially, places are represented by circles and transitions by horizontal bars. If  $IN(p_i, t_j) = k$ , where  $k \geq 1$  is an integer, a directed arc from place  $p_i$  to transition  $t_j$  is drawn with label  $k$ . If  $IN(p_i, t_j) = 0$ , no arc is drawn from  $p_i$  to  $t_j$ . Similarly, if  $OUT(p_i, t_j) = k$ , a directed arc is included from transition  $t_j$  to place  $p_i$ , with label  $k$  if  $k > 1$  and without label if  $k = 1$ . If  $k = 0$ , no arc is included from  $t_j$  to  $p_i$ .

#### DEFINITION 2

Let  $2^P$  be the powerset of  $P$ . We then define functions

$$IP: T \rightarrow 2^P \text{ and } OP: T \rightarrow 2^P \text{ as follows.}$$

$$\begin{aligned} IP(t_j) &= \{p_i \in P: IN(p_i, t_j) \neq 0\} \quad \forall t_j \in T, \\ OP(t_j) &= \{p_i \in P: OUT(p_i, t_j) \neq 0\} \quad \forall t_j \in T, \end{aligned}$$

$IP(t_j)$  is called the set of input places of  $t_j$ , and  $OP(t_j)$ , the set of output places of  $t_j$ .

#### DEFINITION 3

A marking  $M$  of a Petri net  $G$  is a function  $M: P \rightarrow N$ . A marked Petri net  $W$  is a Petri net  $G$  together with a marking defined on it. We denote it by  $(G, M)$  and write  $W = (G, M)$ . We always associate an initial marking  $M_0$  with a given PN.  $M_0$  will represent the initial state of the system which the PN is modelling.

It can be noted that a marking of a Petri net with  $n$  places is an  $n$ -dimensional vector and associates with each place a certain number of tokens which are represented by means of dots inside the places.

#### DEFINITION 4

A transition  $t_j$  of a Petri net is said to be enabled in a marking  $M$  iff

$$M(p_i) \geq IN(p_i, t_j) \quad \forall p_i \in IP(t_j).$$

An enabled transition  $t_j$  can fire at any instant of time. When a transition  $t_j$ , enabled in a marking of  $M$  fires, a new marking  $M'$  is reached according to the equation,

$$M'(p_i) = M(p_i) + OUT(p_i, t_j) - IN(p_i, t_j) \quad \forall p_i \in P,$$

we say marking  $M'$  is reachable from  $M$  and write  $M \xrightarrow{t_j} M'$ .

#### DEFINITION 5

The set of all markings reachable from an initial marking  $M_0$  of a Petri net is called the reachability set of  $M_0$  and is denoted by  $R[M_0]$ .

#### DEFINITION 6

A GSPN is a six-tuple  $(P, T, IN, OUT, M_0, F)$  where

- (a)  $(P, T, IN, OUT, M_0)$  is a marked Petri net;
- (b)  $T$  is partitioned into two sets:  $T_I$  of immediate transitions and  $T_T$  of timed transitions;
- (c)  $F$  is a function with domain  $R[M_0] \times T_T$ , which associates to each  $t \in T_T$  in each  $M \in R[M_0]$ , a continuous random variable that indicates the firing time of  $t$  in  $M$ ; and
- (d) each  $t \in T_I$  has zero firing time in all reachable markings.

In the graphical representation of GSPN, a horizontal line represents an immediate transition and a rectangular bar represents a timed transition. GSPN markings are classified into two types: vanishing markings (those in which at least one immediate transition is enabled) and tangible markings (those in which only timed transitions are enabled). In vanishing markings, as a rule, only an immediate transition is selected to fire even if timed transitions are enabled.

In a vanishing marking, if two or more conflicting immediate transitions are enabled, then any of them may fire, according to a predefined probability distribution. The set of conflicting transitions together with associated probabilities is referred to as a

random switch. A random switch may be static or dynamic. In a static random switch, the switching probabilities are constant whereas in a dynamic random switch, the probabilities will depend on the current marking of the GSPN. Also, note that the firing rates of exponential transitions are in general marking dependent. Dynamic random switches and marking dependent firing rates add significantly to the modelling power of GSPN.

Often, special arcs called inhibitor arcs are employed to test for the absence of a token and capture priorities. An inhibitor arc from a place  $p$  to a transition  $t$  is indicated by a directed arc with a black dot at the end.  $t$  will now be enabled only if  $p$  has no token and all its other (normal) input places have adequate numbers of tokens.

## 2.2 Integrated PFQN-GSPN models

IQP modelling is based on the concept of flow-equivalence. The basic idea is to isolate one or more subsystems of a given system and compute the flow-equivalents for these subsystems. The overall model (also called the high level model) of the given system will then have these subsystems represented by the flow-equivalents. To compute the flow-equivalent of a subsystem, one has to evaluate a model (PFQN or GSPN) of the subsystem for all possible populations (i.e., number of jobs in the subsystem) and compute the throughput in each case. Thus an IQP model can be of two types: (i) IQP model with GSPN as the high level model and (ii) IQP model with PFQN as the high level model. If the high level model is a GSPN, then some of its transitions represent flow-equivalents of subsystems computed using PFQN models of the subsystem. If the high level model is a PFQN, some of its individual nodes correspond to flow-equivalents of subsystems computed using GSPN models of these subsystems.

It is not necessary that the high level model and the models for computing flow-equivalents be of different types (i.e., one is a PFQN and the other a GSPN). If the high level model and the model for flow-equivalence are both PFQN, then we have the classical flow-equivalence of queueing networks (Chandy *et al* 1975). If one is a PFQN and the other a GSPN, we have the integrated PFQN-GSPN models (IQP models) which are the main focus of this paper.

## 3. IQP models for central server FMS

The closed central server network model of figure 1 is a well-formulated representation for an FMS with a transport mechanism such as automated guided vehicles (AGV) and several versatile numerically controlled (NC) machines. The network has two subsystems which we shall call the AGV subsystem and the machine subsystem. The transport mechanism serves as the central server and the probabilities  $q_0, q_1, \dots, q_m$  help capture the routing of parts inside the system. The closedness of the network is ensured by the availability of a limited number of fixtures for parts inside the system. Thus the population of jobs inside the network is equal to the total number of fixtures, assuming a perennial supply of raw parts. The central server network is an easily solvable PFQN but in this section, we shall consider variants of this model by including several non-product form features. Also, for ease of discussion, we consider a model with one AGV and two machines  $M_1$  and  $M_2$ .

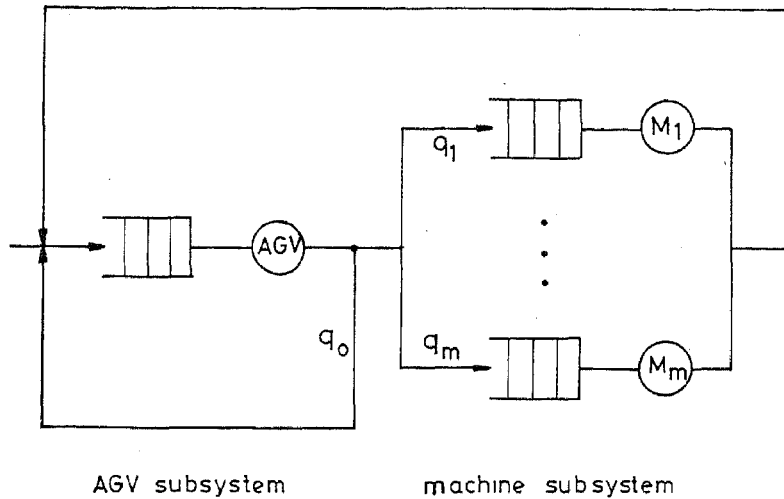


Figure 1. Closed central server model of an FMS with  $m$  machines and one AGV.

3.1 Example 1: FMS with dynamic routing

In this example, we construct an IQP model of a central server FMS with parts routed to the machines  $M_1$  and  $M_2$  dynamically. The dynamic routing policy we consider is the SQR (shortest queue routing) which has a good deal of intuitive appeal. To construct an IQP model, we first observe that the non-product form feature, SQR in this case, is localized in the subsystem comprising the machines  $M_1$  and  $M_2$ . Therefore, a GSPN model of this subsystem can be used to compute the flow-equivalent. The AGV node together with this flow-equivalent will make up a high level PFQN model as shown in figure 2. The GSPN model for the subsystem is shown in figure 3, with description of the model in table 1.

In the GSPN model for computing the flow-equivalent, observe that the AGV is short circuited. The transitions  $t_1$  and  $t_2$  will constitute a dynamic random switch that implements SQR routing, as follows.

$$\begin{aligned} \text{Prob}(t_1) &= 0, && \text{if } M(p_2) > M(p_3), \\ &= 1, && \text{if } M(p_2) < M(p_3), \\ &= 0.5, && \text{if } M(p_2) = M(p_3). \\ \text{Prob}(t_2) &= 0, && \text{if } M(p_2) > M(p_3), \\ &= 1, && \text{if } M(p_2) < M(p_3), \\ &= 0.5, && \text{if } M(p_2) = M(p_3). \end{aligned}$$

Note in the above that ties are resolved with equal probabilities.

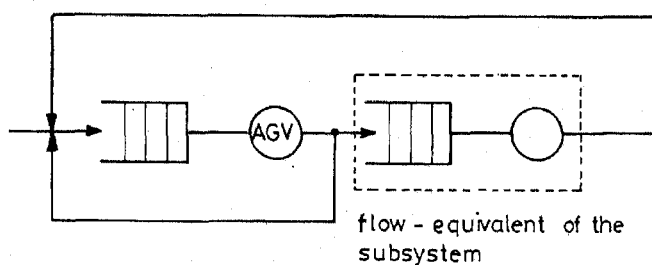


Figure 2. High level PFQN model for example 1.

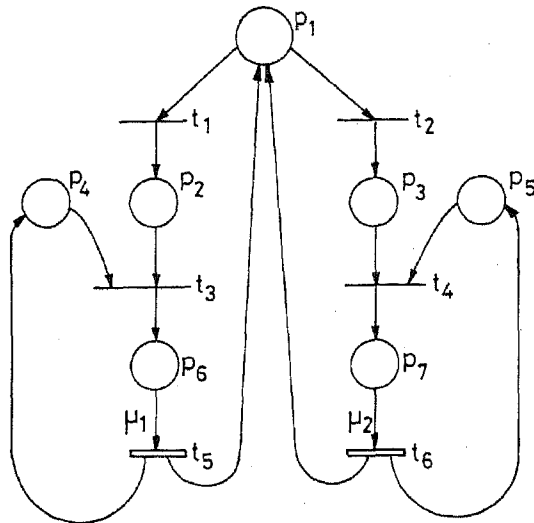


Figure 3. GSPN model for computing the flow-equivalent in example 1.

To compute the flow-equivalent, we observe that the number of jobs in the subsystem at any time can only vary from 0 to  $N$  where  $N$  is the total number of jobs in the closed network model of our FMS. The GSPN model is to be run for each of the populations  $0, 1, 2, \dots, N$  and to compute the service rate of the flow-equivalent server, we have to sum the throughput rates of the transitions  $t_5$  and  $t_6$ . The PFQN high level model of figure 2 can now be solved using standard computational algorithms. It would now be very efficient to study the effect of variations of parameters connected with the AGV node on the performance of the overall system because the experimentation is required on a 2-node PFQN.

To see the computational advantage obtained, notice that an exact GSPN model of the entire system will have  $O(N)$  states where  $N$  is the total number of jobs. Solving

Table 1. GSPN model for computing the flow-equivalent in example 1. (See figure 3.)

<i>Places</i>	
1	Parts ready to be routed to $M_1$ or $M_2$
2	Queue of parts waiting for $M_1$
3	Queue of parts waiting for $M_2$
4	$M_1$ available
5	$M_2$ available
6	$M_1$ processing a part
7	$M_2$ processing a part
<i>Immediate transitions</i>	
1	Part routed to queue for $M_1$
2	Part routed to queue for $M_2$
3	$M_1$ starts processing a part
4	$M_2$ starts processing a part
<i>Exponential transitions</i>	
5	Processing of part by $M_1$
6	Processing of part by $M_2$
<i>Dynamic random switch</i>	
The transitions $t_1$ and $t_2$ constitute a dynamic random switch to model shortest queue routing.	

the exact GSPN model will therefore entail solving a Markov chain with  $O(N)$  states. On the other hand, the GSPN submodel of figure 3 has at most two states for any value of  $N$ , because of SQR routing. In particular, when the population  $n$  is even, the submodel has only one tangible state and for odd  $n$ , there are exactly two tangible states. This would mean that the flow-equivalent can be obtained by solving  $(N + 1)$  Markov chains of at most 2 states, each. Finally, the solution of the PFQN high level model is also a very trivial affair. We can therefore conclude that the exact GSPN model leads to a time complexity of  $O(N^3)$  whereas the IQP model leads to  $O(N)$  complexity.

### 3.2. Example 2: FMS with finite demand for products

Here, we employ a GSPN high level model and compute the flow-equivalent using a PFQN submodel. The system considered is the central server FMS with a non-product form feature in the AGV subsystem rather than in the machine subsystem.

The FMS in question, depicted in figure 4, has a total of  $N$  fixtures, thus limiting the maximum number of in-process jobs to  $N$ . Raw parts are always available, however, a fixture released before the unloading of a finished part is not utilized for a waiting raw part until a specific demand for a finished part is posted in the demand queue. These demands are random and are assumed to constitute a Poisson arrival stream. When  $k$  ( $k > 0$ ) demands are waiting unfulfilled, no further demands are entertained and the demands that arrive when the demand buffer is full are assumed to be lost. There are two potential non-product form features in the AGV subsystem here, namely finiteness of the demands buffer and the joining operation for demands and fixtures. Assuming product-form characteristics for the machine subsystem, we then have a model that is congenial for IQP modelling. Note that the overall model here is not a closed queueing network, but a network in which a maximum of  $N$  jobs can circulate.

A high level GSPN model of the above system is shown in figure 5 (description of the model in table 2). In this model,  $t_7$  is an exponential transition that aggregately represents the machine subsystem. Needless to say,  $t_7$  has a marking dependent firing rate that can be computed by solving the PFQN model shown in figure 6.

Thus the solution of the IQP model in this case proceeds in the following way. In stage 1, the flow-equivalent of the PFQN model of figure 6 is computed by solving this simple model for different populations  $n$  ranging from 0 to  $N$ . This determines

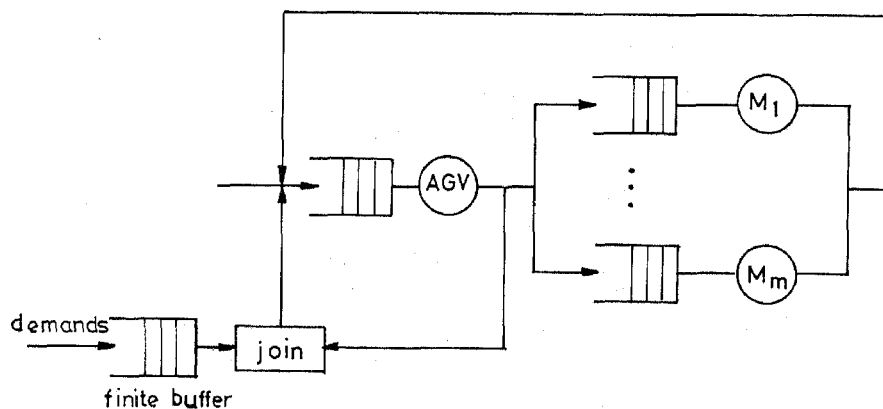


Figure 4. Central server FMS model with a finite external demand for finished products.

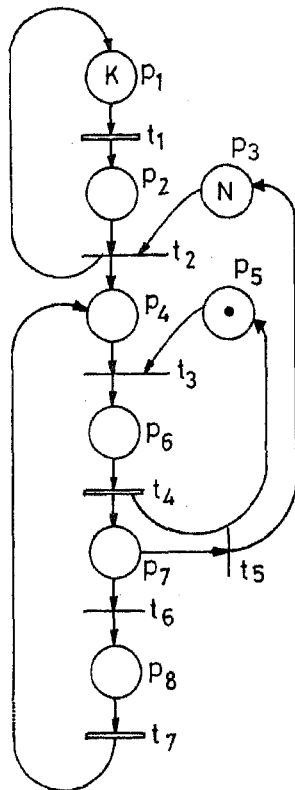


Figure 5. High level GSPN model for example 2.

the marking dependent firing rates of the transition  $t_7$ . In stage 2, the GSPN model of figure 5 is analysed using the standard techniques. This GSPN model has only 3 exponential transitions whereas the exact GSPN model for the entire system has  $m + 2$  exponential transitions where  $m$  is the number of machines.

Table 2. High level GSPN model for example 2. (See figure 5.)

Places	
1	Future demands for products
2	Queue of waiting demands for products
3	Fixtures available
4	Queue of parts waiting for AGV
5	AGV available
6	AGV transporting a part
7	A part just transported by AGV
8	Service in progress in the machine subsystem
Immediate transitions	
2	A fixtured part is released into the system
3	AGV starts transporting a part
5	Part carried by AGV is for unloading
6	Part carried by AGV is for some more machining
Exponential transitions	
1	Arrival of demands for finished products
4	Transport operation by AGV
7	Service in the machine subsystem



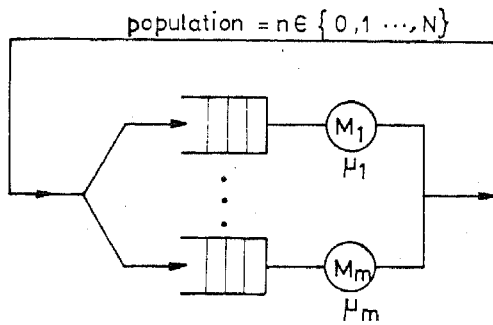


Figure 6. PFQN model for computing the flow-equivalent in example 2.

#### 4. Error analysis

The integrated PFQN-GSPN modelling technique is essentially an approximate technique for obtaining the mean values of performance measures. It has the advantages of computational efficiency of PFQN and representational power of GSPN. The accuracy of performance estimates using the integrated technique will depend on the degree of coupling between the subnetworks. The method leads to highly accurate estimates if this degree of coupling is very small. We shall see this with a simple example abstracted from the central server model we discussed in § 3.

*Example 3.* Consider a performance model with two subnetworks A and B, as shown in figure 7. Let the network B contain some non-product form features whereas the network A is product form. If we employ the IQP modelling, we typically compute the flow-equivalent of network B using a GSPN model which is run for different populations in the network. The computation of the flow-equivalent assumes that the duration for which a population in network B remains constant is long enough to offset any interference from network A. That is, between two successive interactions between the subnetworks A and B, network B reaches an equilibrium. If this condition is not met, then the flow-equivalent is bound to be inaccurate. In the specific case of the model of figure 6, this condition is met when either the value of  $p$  is close to 1 or the relative throughput is much higher than that of A.

We shall now present some numerical experiments conducted with examples 1 and 2, to see the accuracy of the IQP modelling technique. Table 3 shows the mean throughput and mean lead time values for example 1, with the following parameters.

- $\mu_0 = \text{AGV service rate} = 50 \text{ parts per hour};$
- $\mu_1 = M_1 \text{ service rate} = 10 \text{ parts per hour};$
- $\mu_2 = M_2 \text{ service rate} = 5 \text{ parts per hour};$
- $q_1 = q_2 = (1 - q_0)/2;$
- $q_0 \text{ varies from } 0.01 \text{ to } 0.99.$

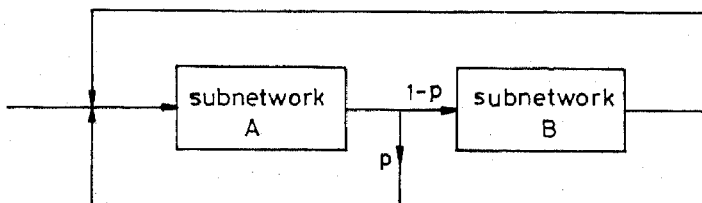


Figure 7. Performance model for example 3.

**Table 3.** Mean throughput and mean lead time for example 1.

$\mu_0 = 50$  parts per hour;  $\mu_1 = 10$  parts per hour;  $\mu_2 = 5$  parts per hour;  $q_1 = q_2 = (1 - q_0)/2$ ;  
Number of fixtures = 8.

$q_0$	Mean throughput		Mean lead time	
	Exact GSPN model	IQP model	Exact GSPN model	IQP model
0.99	49.50	49.50	0.16161	0.16161
0.95	47.49992	47.49993	0.16842	0.16842
0.90	44.98879	44.98999	0.17782	0.17781
0.80	39.14079	39.21808	0.20439	0.20398
0.70	30.37088	30.60465	0.26341	0.26139
0.60	21.44702	21.63114	0.37301	0.36983
0.50	14.71864	14.81111	0.54352	0.54013
0.40	9.91265	9.95348	0.80705	0.80374
0.30	6.39908	6.41611	1.25017	1.24686
0.20	3.74009	3.74664	2.13898	2.13524
0.10	1.66394	1.66592	4.80784	4.80215
0.05	0.78844	0.78921	10.14656	10.13663
0.01	0.15134	0.15147	52.85555	52.85521

Note that the throughput is the number of finished parts per hour and the lead time is the average time spent by a job inside the system. The numerical results in table 3 show the estimates using the IQP modelling and exact GSPN modelling. The estimates obtained through IQP models are found to be remarkably accurate, specially for small values of  $q_0$ .

**Table 4.** Mean throughput and mean lead time for example 2.

$\mu_0 = 50$  parts per hour;  $\mu_1 = 10$  parts per hour;  $\mu_2 = 5$  parts per hour;  $q_1 = q_2 = (1 - q_0)/2$ ;  
Number of fixtures = 4.

$q_0$	Mean throughput		Mean lead time	
	Exact GSPN model	IQP model	Exact GSPN model	IQP model
0.99	19.86499	19.86499	0.03396	0.03396
0.90	19.62480	19.62429	0.05668	0.05671
0.80	18.50368	18.49833	0.09763	0.09784
0.70	15.84976	15.83368	0.16315	0.16376
0.60	12.29868	12.27539	0.26070	0.26186
0.50	8.92581	8.90384	0.40232	0.40405
0.40	6.17814	6.16185	0.61550	0.61781
0.30	4.04035	4.03002	0.96859	0.97164
0.20	2.37657	2.37095	1.67004	1.67440
0.10	1.06081	1.05850	3.76461	3.77304
0.05	0.50316	0.50211	7.94671	7.96081
0.01	0.09664	0.09645	41.38727	41.47170

Table 4 shows similar results for example 2. The accuracy of IQP modelling is again illustrated by these results.

## 5. Concluding remarks

### 5.1 *Advantages and limitations*

In this paper, we have shown through illustrative examples, that integrated PFQN-GSPN models provide an efficient and accurate performance modelling paradigm for flexible manufacturing systems. We collect together below the main advantages of the IQP models.

- (1) IQP models have a solution efficiency comparable to that of PFQN and a representational power equalling that of GSPN.
- (2) The flow-equivalents computed in the integrated technique are exact since they are computed by solving the PFQN or GSPN.
- (3) The accuracy of solution given by IQP models is very good, especially when the subsystems in the given system are loosely coupled.
- (4) In the area of queueing networks, different kinds of approximations have been developed for different non-product form features. The integrated models provide a unified technique for all non-product form features. Also, flow-equivalents computed using queueing networks are approximate and not exact.
- (5) When two or more non-product form features are present in a subsystem, there are few techniques available in queueing theory to handle them. The IQP technique can handle any combination and any number of non-product features in a subsystem, in a unified way.

A limitation of the integrated technique is that it could be used only to compute average performance measures. However, this is true of any approximation technique based on flow-equivalence. Another limitation arises when the non-product form features are distributed across all subsystems. In such a case, the efficiency of IQP models suffers. The efficiency of these models will be appealing if the non-product form features are localized in a single subsystem.

### 5.2 *Use of other modelling paradigms*

In this paper, we have specifically looked at integrated models where PFQN and GSPN are employed. It is needless to add here that other analytical models could be used in the place of the above two models. For example, the flow-equivalent for the SQR subsystem in example 1, which was computed using a GSPN model (figure 3), can as well be computed by visualizing the underlying Markov chain model and solving it. In fact, in this particular case, it is easy to write down the generator matrix of the underlying Markov chain in a straightforward way. In a general situation where it is hard to visualize let alone compute the generator matrix of the underlying Markov chain, GSPN provide an easy-to-use, high level paradigm for automatic computation of the Markov model.

As another example, let us consider the central server FMS model of figure 4 and assume that the routing is SQR in the machine subsystem. In this situation, both the

AGV subsystem and the machine subsystem have a non-product form feature. Three ways of solving it become immediately obvious: (1) Exact GSPN modelling; (2) compute flow-equivalent of AGV subsystem using GSPN and then solve a high level GSPN model; (3) compute flow-equivalent of machine subsystem using GSPN and then solve a high level GSPN model. In methods 2 and 3, all the models are GSPN. For numerical results using methods 2 and 3, see Hanumantha Rao (1991). If one wishes, one could use any combination of analytical or simulation models.

## References

- Balbo G, Bruell S C, Ghanta S 1986 Combining queueing networks and generalized stochastic Petri nets for the analysis of software blocking phenomena. *IEEE Trans. Software Eng.* SE-12: 561-576
- Buzacott J A, Yao D D 1986 On queueing network models of flexible manufacturing systems. *Queueing Systems* 1: 5-27
- Chandy K M, Herzog V, Woo L 1975 Parametric analysis of queueing networks. *IBM J. Res. Dev.* 19: 36-42
- Chandy K M, Sauer C H 1978 Approximate methods for analysis of queueing network models of computer systems. *ACM Comput. Surv.* 10: 263-280
- Hanumantha Rao P 1991 *Integrated analytical models for automated manufacturing systems*, M E Project Report, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore
- Ho Y C 1987 Performance evaluation and perturbation analysis of discrete event systems. *IEEE Trans. Autom. Control* AC-32: 563-572
- Meenakshisundaram C R 1990 *Integrated analytical models for parallel and distributed computing systems*, M Sc (Eng.) thesis, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore
- Murata T 1989 Petri nets: Properties, analysis, and applications. *Proc. IEEE* 77: 541-580
- Ramesh N R 1990 *Integrated performance models based on queueing networks and Petri nets*, M E Project Report, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore
- Ranky P G 1983 *The design and operation of flexible manufacturing systems* (Amsterdam: IFS and North-Holland)
- Viswanadham N, Narahari Y 1988 Stochastic Petri nets for performance evaluation of automated manufacturing systems. *Inf. Decision Technol.* 14: 125-142