STOCHASTIC MODELLING OF FLEXIBLE MANUFACTURING SYSTEMS

N. VISWANADHAM, Y. NARAHARI
Department of Computer Science and Automation
Indian Institute of Science, Bangalore 560 012, India
T.L. JOHNSON
Control Systems Laboratory, General Electric CRD Center
P.O. Box 8, Schenectady, NY 12301, U.S.A.

(Received March 1990)

Abstract—Mathematical modelling plays a vital role in the design, planning and operation of flexible manufacturing systems (FMSs). In this paper, attention is focused on stochastic modelling of FMSs using Markov chains, queueing networks, and stochastic Petri nets. We bring out the role of these modelling tools in FMS performance evaluation through several illustrative examples and provide a critical comparative evaluation. We also include a discussion on the modelling of deadlocks which constitute an important source of performance degradation in fully automated FMSs.

1. INTRODUCTION

This paper presents a tutorial introduction to the important topic of stochastic modelling of flexible manufacturing systems (FMSs). In this area, three main modelling tools, namely Markov chains, queueing networks (QNs), and stochastic Petri nets (SPNs) have gained prominence in recent times. The paper focuses on these three modelling tools and presents illustrative examples to bring out the useful role stochastic models have come to play in logical analysis, performance evaluation and real-time control. We also include a discussion on the modelling and performance analysis of FMSs with deadlocks, which represent an important source of performance degradation in FMSs.

1.1. Flexible Manufacturing Systems

FMSs [1-4] have evoked considerable attention in recent times in the area of manufacturing automation, because of their high productivity. This technology is especially attractive for medium volume industries such as the automobile, aircraft, steel, and electronics industries. The philosophy of an FMS is ideally suited to today’s unpredictable market environment which demands low cost solutions providing for (a) quick product start-up, (b) adaptability, (c) responsiveness to changes in demand, and (d) the capacity to easily resurrect out-of-production designs.

An FMS can be defined [3] as a computer controlled configuration of numerically controlled machine tools and a material handling system (MHS) designed to simultaneously manufacture low to medium volumes of high quality products at low cost. The architecture of a typical FMS is constituted by

(a) versatile numerically controlled (NC) machines which can carry out a variety of machining operations,
(b) an automated MHS comprising one or more of carousels, conveyors, carts, robots, or automated guided vehicles (AGVs), to move parts and tools between machines,
(c) a hierarchical control system that coordinates the actions of the machines, the MHS, and the workpieces,
(d) a load/unload workstation through which the entry and exit of a part occurs; a fixturing station where the parts entering the system are fixtured onto pallets; inspection stations; coordinate measurement machines; and
(e) a buffer-storage in the form of local storage or central storage or both to store raw and semi-finished workpieces.

Often, an FMS is organized as a collection of flexible manufacturing cells (FMCs) which are interconnected by the automated MHS. A typical FMC comprises a few NC machines, tool magazines, and one or more material handling robots. The machines and the robots have individual programmable logic controllers (PLCs) and each FMC has a cell controller. All such controllers are interconnected via a local area network (LAN) to a host computer. Thus one can visualize two kinds of integration in an FMS: material integration provided by the automated MHS and information integration provided by the LAN and a database management system. Figure 1 depicts the architecture of a fully integrated FMS.

In Figure 2, we show a typical sequence of operations in an FMS. Initially, incoming raw workpieces are fixtured onto pallets at a load/unload station. These workpieces move to queues at the workstations of FMCs via the MHS. The sequence of operations on a workpiece is given by the routing table (see, for example, Table 1) which also specifies the choice of machines for each operation of each part type. As a result, there are several alternative routes that a workpiece of a given type can take while being processed. In a well designed system, machines, conveyors, and control elements combine to achieve high productivity, maximum machine utilizations, and minimum in-process inventory. Some of the important decisions that the hierarchical controller takes in an FMS are: scheduling and dispatching (which part to introduce into the FMS next and which machine to be assigned to a part), scheduling of the MHS, tool management, system monitoring and diagnostics, and reacting to disruptions such as machine breakdowns, tool breakages and deadlocks. There are also several off-line decisions taken such as fixing the number of machines, number of AGVs, machine layout, machine grouping, batching and balancing, etc.

Mathematical modelling plays an important role in the on-line and off-line decision making involved in the design and operation of FMSs. Mathematical programming techniques, such as linear programming and integer programming, and heuristic techniques have been used in the prescriptive modelling of FMSs [5] to obtain optimal policies for design and operation. In this
paper, we are concerned with the use of analytical evaluative models, such as Markov chains, queuing networks (QNs) and stochastic Petri nets (SPNs) in the design and operation of FMSs.

1.2. Organization of the Paper

In the next section, we present an FMS as a discrete event dynamical system and give a brief overview of FMS models considered in the literature.

Sections 3, 4, and 5 are devoted to a detailed discussion of Markov models, QN models, and SPN models respectively. In each section, we present an illustrative example to bring out the special features of each modelling tool. We also provide pointers to relevant research papers. To understand the models presented in these sections, the reader may have to refer to relevant books such as the ones by Trivedi [6] and Marasn et al. [7].
In Section 6, we consider the specific issue of the modelling of deadlocks in FMSs. We present an illustrative example using SPNs and discuss performance evaluation of FMSs with deadlocks. We conclude the paper with certain observations on future research in the important area of stochastic modelling of FMSs.

2. FMS MODELLING

An FMS can be described as a discrete event dynamical system [4,8] because changes in the system state are caused by the occurrence of events at discrete instants of time instead of completely described by partial or ordinary differential equations, as in the case of continuous variable dynamical systems. Examples of discrete events in an FMS are: entry/exit of a part, starting/finishing of a part transfer by a robot, starting/finishing of processing by a machine, a robot failure, and a machine breakdown. The number of such activities in a typical FMS is large and there are numerous interactions involving these activities. Further, these interactions exhibit concurrency, contention for resources, synchronization, and randomness. The activities are concurrent because machines and conveyors could be busy at the same time. There is a limited number of resources such as machines, tools, buffers, robots, conveyors, etc. and this causes contention for these resources and synchronization constraints. Machine failures, tool breakages, and unanticipated breakdowns occur at random points of time and this introduces randomness in activities. These interactions might also lead to a deadlocked state, in which the system is crippled and produces no output. Any effective model of an FMS should therefore capture the above characteristics.

The need for and the role of mathematical models in FMSs has been clearly brought out by Buzacott [9], Suri [5,10], and by Buzacott and Yao [11]. FMS models can be used in a variety of ways such as in logical analysis, performance prediction, optimization, control, and for gaining insight into the system.

There are two basic types of models of discrete event dynamical systems and hence FMSs: qualitative models and quantitative models. Qualitative models such as Petri nets [19], finite state machines [13], and general algebraic discrete event models [14] are useful for investigating logical aspects of FMS behavior such as boundedness, fairness, absence/existence of deadlocks, mutual exclusion of resources, and correctness of control logic. Quantitative models are basically stochastic models that address quantitative issues of performance such as production rate of parts, manufacturing lead times, queueing characteristics, machine/robot/AGV utilizations, and reliability. According to Buzacott [9], there are three basic types of quantitative modelling tools: simulation models, analytical models, and hybrid models. We now give a brief review of these three types of quantitative models.

Simulation models can be used to construct a detailed representation of the system operation, generally using simulation languages such as GPSS, SIMSCRIPT, SIMULA, or SLAM. Many simulation models have been developed in industry as design aids; for example, see references [15–17]. A simulation model enables a detailed representation of the characteristics of jobs, machine behaviour, flow of parts, and job routing, and complex control and sequencing rules. Thus, predictions of performance made by the model are potentially very accurate. This ability of simulation models to capture the complexity of the system makes it a very expensive tool computationally, because the simulation run can be lengthy. Also, the time to develop and adequately validate a detailed simulation model can be substantial.

Compared to simulation, analytical models of an FMS cannot capture all the details of the system. It is necessary to decide how much detail to include because too much makes the model intractable and too little makes the model unrealistic. Once a solution of an analytical model has been developed, it can be subjected to the full power of tests of mathematical correctness and precise statements can be made about the validity of the solution for the given assumptions. The process of model development and solution leads to new insight into the system and suggests ways in which assumptions might be relaxed. This in turn can lead to new ideas about how the system should be operated and controlled.

A third approach to FMS modelling uses a hybrid scheme in which simulation and analytical models are combined. Here there are two variants. In the first scheme, the modelling approach follows the framework of an analytical model but for some parts, where analysis is difficult or
impossible, a simulation submodel is constructed. From the results of simulation, an equivalent component for an analytical model is developed and the integrated analytical model is solved. See, for example, reference [18]. The other hybrid approach, called perturbation analysis [19,20] begins with the results of a simulation model. Using analytical techniques, the effect of a perturbation in the value of a parameter is found. Thus it is not necessary to repeat simulation runs to carry out parametric analysis. In this paper, we are concerned with analytical models only. We do not consider simulation and hybrid models.

3. MARKOV MODELS

In this section, we present Markovian modelling of FMSs. Markov and semi Markov processes are an important subclass of stochastic processes, which constitute the basis of performance evaluation using any analytical tool. The following are the advantages of Markov models. First, the data to be collected can be explained easily, since each element of a Markov transition matrix can be thought of as the relative frequency with which a transition occurs from one state to another. Second, the computations involve well-known and easily implemented matrix operations. Finally, the quantities computed from a Markov analysis can be easily interpreted. The major disadvantages of Markov analysis are:

(1) when the size of the physical system grows, the number of states in the Markov chain grows exponentially and this makes Markov analysis expensive;
(2) with increase, in the number and complexity of interactions in the physical system, it becomes difficult to visualize the Markov chain states and the transitions among the states;
(3) existence of two or more time scales lead to increased computational difficulties. In the sequel of this section, we shall illustrate the construction of Markov models and discuss performance evaluation using these models.

3.1. Markov Model of a Central Server FMS

In the FMS that we consider here, we have an AGV and two machines, $M_1$ and $M_2$, processing parts of a single type. There are two fixtures available and consequently at most two parts can be inside the system. We assume that raw parts are always available and hence when a finished part is unloaded from the system, the fixture that is released is immediately assigned to a fresh part. Each part undergoes a certain number of operations on the two machines. The AGV is required for loading, unloading, and at the end of every machine operation, transporting the part to an appropriate destination.

A natural representation of the above is provided by a queueing network model, as shown in Figure 3. There are three queues corresponding to the three service centers, namely AGV, $M_1$ and $M_2$. The queues are interconnected in the way shown. If we assume that the service time at each center is exponentially distributed, then we have a classical queueing network often called as the closed central server network [6] and consequently we shall call this the central server FMS model. The closed nature of the network is ensured by the constant number of fixtures available and a perennial supply of raw parts. Designating the service centers AGV, $M_1$, and $M_2$ as service
centers 0, 1, and 2 respectively, the routing matrix would be given by

\[
\begin{bmatrix}
  Q_0 & q_1 & q_2 \\
  1 & 0 & 0 \\
  1 & 0 & 0 
\end{bmatrix}
\]

The choice of the routing probabilities \( q_0, q_1, \) and \( q_2 \) will depend on the physics of processing. In this system, a fresh part is fixtured immediately after loading and is defixtured just before unloading. The released fixture is used for fixturing another fresh part. Since new parts are always available, the exit of a finished part always leads to the entry of a newly fixtured part. Thus the arc labelled by \( q_0 \) signals the exit of a finished part as well as the entry of a fresh part. Intuitively, it can be seen that \((1/q_0)\) is the average number of times the AGV is required to transport a typical part inside the system. Also \((q_1/q_0)\) and \((q_2/q_0)\) give the average numbers of times a typical part is processed by \( M_1 \) and \( M_2 \), respectively, before the part leaves the system.

Since the service times at all the nodes in the queueing network model of Figure 3 are exponentially distributed, the underlying stochastic process is a continuous time Markov chain. Thus, by enumerating the set of all states, one can obtain a Markov chain representation for the above FMS. If the number of fixtures is two, there are exactly six states in this Markov chain, as shown below:

- **State 1**: \( M_1 \) and \( M_2 \) processing one part each (only AGV free)
- **State 2**: \( M_2 \) processing a part and AGV transporting the other part (only \( M_1 \) free)
- **State 3**: \( M_1 \) processing a part and AGV transporting the other part (only \( M_2 \) free)
- **State 4**: AGV transporting a part and the other part waiting for the AGV (\( M_1 \) and \( M_2 \) both free)
- **State 5**: \( M_2 \) processing a part and the other part waiting for \( M_2 \) (AGV and \( M_1 \) both free)
- **State 6**: \( M_1 \) processing a part and the other part waiting for \( M_1 \) (AGV and \( M_2 \) both free).

The transitions among these states and the corresponding transition rates are shown in Figure 4. The transition rate matrix \( Q \) of the Markov chain model is shown in Table 2.
Since this Markov chain is finite and irreducible, it is ergodic. If \( \Pi = (\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6) \) is the vector of steady state probabilities of the above Markov chain, then by the standard solution to the Kolmogorov differential equations [6], it is known that

\[
\Pi Q = 0 \quad \text{and} \quad \sum_{i=1}^{6} \pi_i = 1.
\]

Ergodicity of the above Markov chain ensures that a unique solution exists for the above set of equations, and we can compute desired performance measures from these probabilities, as shown below:

Utilization of AGV = sum of the steady state probabilities of all states in which AGV is busy

\[
= \pi_2 + \pi_3 + \pi_4
\]

Similarly,

Utilization of \( M_1 \) = \( \pi_1 + \pi_3 + \pi_6 \)

Utilization of \( M_2 \) = \( \pi_1 + \pi_2 + \pi_6 \)

Expected number of parts in the

- AGV subsystem = \( \pi_2 + \pi_3 + 2\pi_4 \)
- \( M_1 \) subsystem = \( \pi_1 + \pi_3 + 2\pi_6 \)
- \( M_2 \) subsystem = \( \pi_1 + \pi_2 + 2\pi_5 \).

Other performance measures can be similarly obtained.

### 3.2. Review of Markov Models

In the above example, let us say we increase the number of fixtures to 3. That is, there are now three jobs circulating inside the network. In this case, we can say with some difficulty that the number of states in the resulting Markov chain is 10. If we further increase the number of fixtures, two things happen. First, the complexity of the Markov chain grows and its construction is no longer trivial. Second, the number of states in the Markov chain rises rapidly and makes computation of steady state probabilities and performance measures very expensive. Other changes in the system, such as increasing the number of AGVs or machines or both, also lead to the same complications. Note, however, that changes in routing probabilities or service rates or both will only change the transition rate matrix and leave the state space unaffected. The first of the above limitations can be effectively overcome by stochastic Petri nets which enable automatic generation of state space and transition rate matrix, using an elegant graphical representation and precise rules for dynamic evaluation. The second limitation is completely avoided by the queueing networks which do not at all generate the state space or transition rate matrix and instead use efficient polynomial time computational algorithms for computing performance measures directly. In fact, the above closed queueing network is one of the most efficiently solved queueing network models. For the above reasons, QNs and SPNs are almost always preferred to Markov chains. It should be noted that the underlying stochastic process is still a Markov or semi-Markov process, whether it be a QN model or an SPN model. To choose between SPNs and QNs, one has to weigh other considerations, some of which will be presented in subsequent sections.

Markov and semi-Markov models for manufacturing systems have been investigated by several researchers: Gershwin and Berman [21], Gershwin and Schick [22], Buzacott and Shantikumar [23], Foster and Garcia-Diaz [24], Alam et al. [25], and Ammar [26]. Most of these researches are on traditional flow lines. Real-life phenomena such as blocking, starving, and machine failures have been modelled using Markov chains. Also in some cases efficient techniques have been presented for Markov chain analysis. For example, see [21,22]. Ammar [26] has included the effect of control and communication equipment in his Markov chain model. If we want to include failure times, processing times, and message communication times in the same Markov chain model, then we are confronted with stiff Markov chains [27] because of large orders of magnitude difference among these times.
4. QUEUEING NETWORK MODELS

Queueing networks are by far the most popular and efficient analytical modelling tool for FMSs. An overview of QN models of FMSs can be obtained from the papers by Buzacott and Yao [28], Suri and Hildebrant [29], and Seidman et al. [30]. QN models capture dynamics, interactions, and uncertainties in an FMS in an aggregate way. The performance measures computed are average values which assume a steady-state operation of the system. QN-based models of FMSs can be broadly classified into the following three types [28]:

2. Reversible Networks: Fixed routing model, fixed loading model, and dynamic routing model.
3. Approximate QN models for handling non-product form features such as priorities, blocking, multiple resource holding, and synchronization.

In this section, we present a closed queueing network (CQN) model of a simple FMS to illustrate QN models.

4.1. Closed Queueing Network Models

The basic CQN model of an FMS was first developed by Solberg [31]. Since then several researchers have based their models on CQNs. The main assumptions of the model are as follows [11]:

1. The total number of jobs in the system is a fixed constant \( N \), which can be viewed as the total number of fixtures or pallets available in the system. This implies that when a completed job leaves the system, a fresh job is ready and enters the system.
2. At all stations with FCFS queueing discipline, service time distributions are exponential and all job classes must have the same service rate parameter at a station.
3. All stations have a local storage large enough to accommodate all \( N \) jobs in the system. This prevents occurrence of blocking.
4. Machines are always available for processing jobs. That is, there are no breakdowns and any set up or tool changing time is included in the service time.

A CQN with the above assumptions has an equilibrium distribution that is "product form". Let \( n = (n_1, n_2, \ldots, n_M) \) denote the state of the system where \( n_i (i = 1, 2, \ldots, M) \) is the number of jobs at station \( i \). The state space of the system is the set of \( n \) such that

\[
\sum_{i=1}^{M} n_i = N.
\]

The steady state probability that the system is in state \((n_1, n_2, \ldots, n_M)\) is given by

\[
\frac{1}{G(N)} \sum_{i=1}^{M} f_i(n_i),
\]

where \( G(N) \) is the normalizing constant and \( f_i(n_i) (i = 1, 2, \ldots, M) \) are simple functions of given parameters. The central problem is to compute \( G(N) \), and performance measures can be directly obtained from \( G(N) \). Details about efficient computational algorithms in this context may be obtained from [6].

The above model has been widely used for preliminary design of FMSs and for studying some of the issues in production planning, such as machine grouping [32]. Consider the FMS depicted in Figure 5. The system comprises four machines \( M_1, M_2, M_3 \) (tool machines) and \( L/U \) (load/unloaded machine), and a cart moving along a rail. A buffer is also available to store those workpieces that have not yet been completely processed. The workpieces are of two types. Type one parts have a two-phase operation schedule, while type two parts need only one machining operation before leaving the FMS. The first phase on type one parts can be accomplished only
Modelling manufacturing systems

Figure 5. A simple FMS.

Table 3. Details of machining times for the FMS example.

<table>
<thead>
<tr>
<th>Class</th>
<th>Required production mix</th>
<th>Process number</th>
<th>Mean machining times in minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>M1</td>
</tr>
<tr>
<td>1</td>
<td>20%</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>80%</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 6. A multiclass CQN model of the FMS of Figure 5.

on $M_1$, while the second phase can be performed either by $M_2$ or by $M_3$. Type two parts can undergo their single operation on $M_2$ or on $M_3$. There is a requirement to be met concerning production mix: of the total finished parts, 20% must be parts of type 1. Table 3 gives the data concerning the working schedules.

Inside the system, the workpieces are mounted on pallets for proper presentation and orientation during processing. Different pallet types are needed for different part types. The available number of pallets thus constrains the number of workpieces of various types flowing through the system. In fact, this means that the number of parts of each type in the system is fixed and this leads to a CQN model.

Figure 6 shows a multiclass CQN model for the above system. The three machines $M_1$, $M_2$, and $M_3$, the cart ($TR$), and the $L/U$ station are numbered from 1 to 5 in that order. The solid lines and the dotted lines have been used in the figure to distinguish between the two classes. Routing probabilities for the QN model can be computed using the following arguments. First note that each part of type 1 uses the cart three times during its processing, whereas each part of type 2 uses the cart twice during its processing. Let $V_{i,r}$ denote the visit ratio of parts of type $r(r = 1, 2)$ to station $(i = 1, 2, 3, 4, 5)$ and $P_{r,s}$ represent the fraction of times process number $s(s = 1, 2)$ is chosen to perform the machining of parts of type $r(r = 1, 2)$. It can be observed
that the following relations hold.

\[ V_{1,1} = 1; \quad V_{2,1} = p_{1,1}; \quad V_{3,1} = p_{1,2}; \]
\[ V_{4,1} = 0; \quad V_{5,1} = 1; \]
\[ V_{1,2} = 0; \quad V_{2,2} = p_{2,1}; \quad V_{3,2} = p_{2,2}; \]
\[ V_{4,2} = 2; \quad V_{5,2} = 1. \]

Given the structure of the model, these visit ratios uniquely identify the routing probabilities of the parts flowing through the network:

\[ q_{4,1,1} = q_{4,5,1} = 1/3, \]
\[ q_{4,2,1} = p_{1,1}/3, \]
\[ q_{4,3,1} = p_{1,1}/3, \]
\[ q_{4,1,2} = 0, \]
\[ q_{4,2,2} = p_{2,1}/2, \]
\[ q_{4,3,2} = p_{2,2}/2, \]
\[ q_{4,5,2} = 1/2, \]

where \( q_{i,j,r} \) represents the probability that a part of class \( r \) moves to station \( j \) upon completion of processing at station \( i \). The probabilities \( P_{r,i} \) can be computed using the procedure outlined in [33] and can be shown to be

\[ P_{1,1} = 0; \quad P_{1,2} = 1 \]
\[ P_{2,1} = .65; \quad P_{2,2} = .35 \]

Assuming that the queueing discipline is FCFS and that the processing times are exponentially distributed (with means as shown in Table 3), the above CQN becomes product form and can be analyzed using standard computational algorithms for closed queueing networks [6].

4.2. Review of QN Models

QN modelling of FMSs has been pursued at various institutions and there is a rich body of literature available in this area and notable contributions may be found in [34–39]. QN models of FMSs have been well reviewed in the literature and the reader is referred to the excellent survey papers by Buzacott and Yao [11,28]. Techniques based exclusively on CQN models have been reviewed by Seidman, Schweitzer, and Shalev-Oren [30]. For this reason, we do not attempt a fresh review of QN models here.

Under suitable assumptions, QN models of FMSs become product-form queueing networks (PFQNs) [6] and can therefore be solved using efficient computational algorithms. However, features such as priorities, synchronization, multiple resource holding, and blocking make QN models non-product form and entail approximate techniques to be used. The use of approximate analysis for non-product form QN models for FMSs has been investigated by Suri and Hildebrant [29], Cavaille and Dubois [34], Shalev-Oren et al. [35], Kamath and Sanders [39], and several others. A useful alternative model that can be used in such situations is stochastic Petri nets, discussed in the next section.

5. STOCHASTIC PETRI NET MODELS

Among analytical techniques for modelling FMSs, timed Petri nets in general and stochastic Petri nets in particular have emerged as a major tool in recent times. The major reasons for the popularity of SPN-based models are: (a) the elegant graphical nature of the model, (b) analysis techniques that can be completely automated, (c) the power of the model in representing non-product form queueing network features such as blocking, synchronization, priorities, and multiple resource holding, and (d) the excellent recognition classical Petri nets have already gained
as models of logical behaviour of FMSs. Dubois and Stecke [40] first employed Petri nets with deterministic timed transitions to model typical manufacturing systems and applied the results of Cohen et al. [41] to compute performance measures. However, the techniques presented by Dubois and Stecke are applicable to only FMSs with deterministic decision making and deterministic processing times and consequently cannot handle random phenomena. Stochastic Petri nets and, especially, generalised stochastic Petri nets (GSPNs) [7] are now being extensively used by various research groups as a performance modelling tool of FMSs. Notable contributions are reported in [33, 42-46]. In this section, we present two illustrative examples to bring out the major aspects and advantages of performance modelling of FMSs by SPN-based models. We shall use GSPNs as the modelling tool.

5.1. Central Server FMS Model

We revisit here the example in Section 3. Recall that the system comprises an AGV and two machines (Figure 3). Figure 7 shows a GSPN model for the FMS under study. The circles in the diagram are called places and represent conditions or resources in the system. There are 10 places in the model. Their interpretation is given in Table 4. The horizontal bars \( t_1, t_3, t_4, t_5, t_6, t_7 \) are called immediate transitions and represent logical changes in the system. The rectangular bars \( t_a, t_s, t_g \) are called exponential transitions and represent timed activities such as processing by machines and transportation by AGV. These three transitions are associated with exponential random variables with rates \( \mu_0, \mu_1, \mu_2 \) respectively. The interpretations of immediate and exponential transitions are also given in Table 4. The presence of black dots called "tokens" inside the places \( p_z, p_g, p_l \) indicates that the AGV is free, that a part is getting processed by \( M_1 \), and that a part is getting processed by \( M_2 \). This is the initial state or initial marking of the system (the words "state" and "marking" are used interchangeably in the sequel). Note that the immediate transitions \( t_3, t_4, t_5 \) are conflicting in the sense that only one of them fires at any point of time, disabling the others. The probability with which each can fire is specified by the probability distribution \( q_0, q_1, q_2 \) defined on this set of transitions. Note that these are the routing probabilities. The transitions \( t_3, t_4 \) and \( t_5 \) together with the associated probabilities is designated as a random switch.

The dynamic evolution of the GSPN model is governed by the firing rules for GSPN transitions. The evolution of markings and the transitions that lead to these markings are depicted in Figure 8. There are two types of markings here, namely vanishing markings and tangible markings. Vanishing markings are those in which at least one immediate transition is enabled and tangible markings are those in which only exponential transitions are enabled. The marking process of the GSPN model (the stochastic process that describes the dynamic evolution of the GSPN model) stays for zero time in vanishing markings because the firing time of each immediate transition is zero. On the other hand, the sojourn time of the marking process in the tangible markings is an exponential random variable. In Figure 8, each marking is represented by the set of places having a "token". For example, tangible marking \( T_1 \) is \( p_2p_9p_{10} \) meaning that there is a token each in the places \( p_2, p_9, \) and \( p_{10} \), and no token elsewhere. The following may be noted:

(i) There are 6 tangible markings (shown by double circles in Figure 8) and 11 vanishing markings (shown by single circles in the same figure). Tangible markings are precisely the six states enumerated in the continuous time Markov chain of Section 3. Vanishing markings are intermediate markings which lead to vanishing or tangible markings because of logical changes in the system.

(ii) The set of tangible and vanishing markings constitutes a discrete time Markov chain under an appropriate interpretation [7]. The transition probabilities are in fact shown in Figure 8, as the labels on the arcs.

(iii) By systematic elimination of vanishing markings as detailed in [7], a "reduced" embedded Markov chain comprising tangible markings alone may be obtained and will be the same as the Markov chain of Figure 4. Note that this Markov chain is the same as that we had obtained in Section 3.

The reader is referred to [7] for the procedure to compute the steady state probabilities \( \pi_1, \pi_2, \ldots, \pi_6 \) of the six tangible markings. Using these, we can compute several performance
Table 4. Description of the GSPN model of a simple FMS with an AGV and two machines.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>Queue of parts waiting to be transported by the AGV</td>
</tr>
<tr>
<td>$p_2$</td>
<td>AGV available</td>
</tr>
<tr>
<td>$p_3$</td>
<td>AGV transporting a part</td>
</tr>
<tr>
<td>$p_4$</td>
<td>A part that has just been transported by an AGV</td>
</tr>
<tr>
<td>$p_5$</td>
<td>Parts waiting to get serviced by $M_1$</td>
</tr>
<tr>
<td>$p_6$</td>
<td>Parts waiting to get serviced by $M_2$</td>
</tr>
<tr>
<td>$p_7$</td>
<td>$M_1$ available</td>
</tr>
<tr>
<td>$p_8$</td>
<td>$M_2$ available</td>
</tr>
<tr>
<td>$p_9$</td>
<td>$M_1$ processing a part</td>
</tr>
<tr>
<td>$p_{10}$</td>
<td>$M_2$ processing a part</td>
</tr>
<tr>
<td>$t_1$</td>
<td>AGV starts transporting a part</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Transport of a part by AGV</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Finished parts gets unloaded from the system and a fresh part is carried into the system by the AGV</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Part joins the queue for $M_1$</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Part joins the queue for $M_2$</td>
</tr>
<tr>
<td>$t_6$</td>
<td>$M_1$ starts processing a part</td>
</tr>
<tr>
<td>$t_7$</td>
<td>$M_2$ starts processing a part</td>
</tr>
<tr>
<td>$t_8$</td>
<td>Processing of a part by $M_1$</td>
</tr>
<tr>
<td>$t_9$</td>
<td>Processing of a part by $M_2$</td>
</tr>
</tbody>
</table>

$t_1, t_2, t_3, t_4, t_5, t_7$: Immediate transitions  
$t_3, t_4, t_5$: Random switch with probabilities $q_0, q_1$ and $q_2$ respectively  
$t_2, t_6, t_9$: Exponential transitions with rates $\mu_0, \mu_1$ and $\mu_2$ respectively.
measures as shown in Section 3. The computation of the performance measures can be completely automated, unlike in the case of Markov chains.

In fact, the analysis of a GSPN model, starting with the generation of tangible and vanishing markings till the computation of performance measures, can be completely automated and this is a major consideration in using GSPNs.

5.2. Central Server FMS Model with Priorities

Consider again the central server based FMS considered above. Suppose now we change the operating rule in this system as described in the following: If a part finds both $M_1$ and $M_2$ free, then it is assigned to $M_2$ (probably because $M_2$ is faster) and if it finds both busy, the earlier probabilistic routing is used. Also if the part finds exactly one of the machines available, it is assigned to that machine. To capture the effect of the above decision rule using a queueing network, we need a mechanism to represent priorities. We shall now see how the above can be represented using a GSPN model. Figure 9 depicts such a GSPN model. $p_{11}$ and $p_{12}$ are two additional places and transitions $t_{11} - t_{15}$ are five additional transitions, in comparison with the GSPN model of Figure 7. There are “inhibitor arcs” in this model (Figure 9) indicated by a black dot as the end marker on the transition. For example, consider transition $t_{10}$, which now fires only if there is a token in $p_{11}$ and there is no token in either $p_7$ or $p_8$. This is because of the existence of a “normal” arc from $p_{11}$ to $t_{10}$ and inhibitor arcs from $p_7$ and $p_8$ to $t_{10}$. Inhibitor arcs provide a mechanism to represent priorities in the system. Note that in this GSPN model, there are two random switches $[t_3, t_4]$ and $[t_{14}, t_{15}]$ instead of the single random switch $[t_3, t_4, t_5]$ in the case of Figure 7. This is required in order to model the prioritized decision rule. Also note the changes made in the associated probabilities of these random switches.

Now, let us complicate the decision rule further, by stipulating that if a part finds both machines busy, it will be assigned to the queue with the smaller number of waiting parts. Further if the two queues comprise the same number of waiting parts, we would route based on the probabilities $q_1$ and $q_2$. This new decision rule can be modelled by the same GSPN (Figure 9), with the “static” random switch $[t_{14}, t_{15}]$ replaced by a “dynamic” random switch, as shown below. The probabilities of firing $t_{14}$ and $t_{15}$ will now depend on the current marking $M$ of the GSPN model.
and will be defined as under:

$$\text{Prob}(t_{14}) = 0 \quad \text{if} \quad M(P_5) > M(P_6)$$

$$= \frac{q_1}{q_1 + q_2} \quad \text{if} \quad M(P_5) = M(P_6)$$

$$= 1 \quad \text{if} \quad M(P_6) < M(P_5)$$

$$\text{Prob}(t_{15}) = 0 \quad \text{if} \quad M(P_6) > M(P_8)$$

$$= \frac{q_2}{q_1 + q_2} \quad \text{if} \quad M(P_6) = M(P_8)$$

$$= 1 \quad \text{if} \quad M(P_8) < M(P_6).$$

The above example shows the power and elegance of GSPNs in modelling features that cannot be easily handled using QNs and Markov chains.

5.3. Review of SPN Models

GSPNs provide a compact, graphical framework as a modelling tool. It is easy to construct and understand GSPN models and their analysis can be completely automated. In comparison to Markov chains, GSPNs offer the advantages of automatic generation of the transition rate matrix. Also, aggregated GSPN models can lead to a significant reduction in the number of states of the Markov chain. Compared to product form QNs, GSPNs are more powerful because of their ability to represent non-product form features. However, product form QNs are certainly much more efficient wherever applicable, and approximate analysis of non-product form queueing networks.
is also a good alternative to GSPNs. An excellent comparison between the relative virtues of GSPNs and product form QNs may be found in [47]. The integrated technique proposed by Balbo, Bruell, and Ghanta [48], combining the best features of product form QNs and GSPNs in the same modelling framework, should prove to be quite useful in the FMS context.

6. MODELLING OF DEADLOCKS

Deadlock is one of the major issues which has received wide attention in the context of automated manufacturing. Deadlocks [49] represent a highly undesirable phenomenon in resource sharing concurrent systems. A deadlock is a situation where each of a set of two or more jobs keeps waiting indefinitely for the other jobs in the set to release resources. The occurrence of a deadlock can potentially stall all production in an automated factory. In this section, we present how analytical modelling of deadlocks can be done using stochastic models.

6.1. A Simple Manufacturing Deadlock

To visualize an example of a deadlock in a manufacturing system, consider a transfer line comprising two machines, $m_1$ and $m_2$, and an automated guided vehicle (AGV), shown in Figure 10. The AGV can carry only one part at a time. Assume that a given part undergoes the operations in the following sequence: (i) A fresh part, say $p$, enters the system when $m_1$ is free. (ii) After the machining by $m_1$ is complete, the AGV picks up $p$ and delivers it to $m_2$ if $m_2$ is free, otherwise the part waits on the AGV. (iii) When $m_2$ finishes processing $p$, the AGV unloads the part from the system.
Table 5. Description of the GSPN model of Figure 11.

<table>
<thead>
<tr>
<th>Places:</th>
<th>Transitions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$: At least one raw part waiting for processing by $M_1$</td>
<td>$t_1$: $M_1$ starts processing a part</td>
</tr>
<tr>
<td>$p_2$: $M_1$ idle</td>
<td>$t_2$: Processing of a part by $M_1$</td>
</tr>
<tr>
<td>$p_3$: $M_1$ processing a part</td>
<td>$t_3$: $AGV$ starts transporting a part from $M_1$ to $M_2$</td>
</tr>
<tr>
<td>$p_4$: $M_1$ waiting for $AGV$, after finishing processing</td>
<td>$t_4$: Traveling of $AGV$ from $M_1$ to $M_2$</td>
</tr>
<tr>
<td>$p_5$: $AGV$ transporting a part, from $M_1$ to $M_2$</td>
<td>$t_5$: $M_2$ starts processing a part</td>
</tr>
<tr>
<td>$p_6$: $AGV$ waiting for $M_2$, after bringing a part to $M_2$</td>
<td>$t_6$: Processing of a part by $M_2$</td>
</tr>
<tr>
<td>$p_7$: $M_2$ idle</td>
<td>$t_7$: $AGV$ starts transporting a part from $M_2$ to unload station</td>
</tr>
<tr>
<td>$p_8$: $AGV$ idle</td>
<td>$t_8$: Traveling of $AGV$ from $M_2$ to unload station</td>
</tr>
<tr>
<td>$p_9$: $M_2$ processing a part</td>
<td>$t_1, t_3, t_5, t_7$: Immediate transitions</td>
</tr>
<tr>
<td>$p_{10}$: $M_2$ waiting for $AGV$, after finishing processing</td>
<td>$t_{12}, t_{13}, t_{14}$: Exponential transitions</td>
</tr>
<tr>
<td>$p_{11}$: $AGV$ transporting a part from $M_2$ to unload station</td>
<td></td>
</tr>
</tbody>
</table>

Figure 11 shows a GSPN model for the above system and Table 5 contains a description of the places and transitions in this model. Now, imagine the following sequence of events, starting with an initial state in which $M_1$ and $M_2$ are both free and $AGV$ is available: (i) A fresh part, say $p_1$, enters the system, gets processed by $M_1$ and is delivered at $M_2$ by the $AGV$; $M_2$ starts processing $p_1$. (ii) A second part, say $p_2$, enters the system, gets processed by $M_1$, is put on the $AGV$ and transported to $M_2$, but the part waits on the $AGV$ since $M_2$ is busy. (iii) A third part, $p_3$, enters the system and gets processed by $M_1$. Meanwhile, $M_2$ finishes processing $p_1$. At this stage, $M_1$ is blocked because $AGV$ is not available; $AGV$ will not become available unless $M_2$ is free; $M_2$ cannot be free until $AGV$ is available. This situation is called a deadlock and the whole system comes to a standstill as shown by the event sequence diagram in Figure 12. In the GSPN model, the firing sequence of transitions that would lead to this deadlock is $t_1, t_2, t_3, t_4, t_5, t_6$. When the transitions fire in that sequence, we reach a marking in which none of the transitions is enabled, indicating a deadlock in the underlying FMS. Such situations can potentially occur in manufacturing systems because of limited number of resources, concurrent activities, and random nature of failures.

Now, consider two transfer lines, $T_1$ and $T_2$, of the type discussed above, and interconnected by a buffer. Assume that the parts processed by $T_1$ are deposited in the buffer and $T_2$ takes its input parts from the buffer. Now, if there occurs a deadlock of the type discussed above, in the transfer line $T_2$, it would be a partial deadlock for the overall system. Now, $T_1$ processes more parts and deposits them in the buffer. Once the buffer becomes full, the entire system is deadlocked. The above shows how a partial deadlock eventually leads to a (full) deadlock in the entire system. The occurrence of a deadlock or a partial deadlock can cripple the entire system and can potentially render automated operation impossible. The problem of deadlocks is therefore an important one to be tackled in FMSs.

Most studies of FMS deadlocks (see for example [12,50]) have focused on proving the existence/absence of deadlocks using Petri net models. Here, we show how a stochastic model such as a GSPN model can capture deadlocks in an FMS.
6.2. Analysis of Deadlocks

The GSPN model of Figure 11 can be used to study the performance of the transfer line, in the presence of deadlocks. The reachability graph of this GSPN model is shown in Figure 13. There are 8 vanishing markings and 14 tangible markings in the reachability set. The vanishing markings are shown by single circles and the tangible markings by double circles. While constructing the GSPN model, we have assumed that all processing times and AGV transportation times are exponential random variables. Using this reachability graph, the following issues can be studied:

1. The marking \( p_1p_4p_6p_{10} \) corresponds to a deadlock since none of the transitions of the GSPN model is enabled in this marking. As soon as the marking process reaches this state, it would get absorbed and remain there forever. So, the marking process here is a semi-Markov process with single absorbing state. In general, there will be as many
absorbing states as the number of deadlocks. By doing an extensive path analysis of the reachability graph, we can enumerate the set of all transition firing sequences that lead to the deadlock. An analysis of these sequences can lead to operating rules for preventing the occurrence of deadlocks. For example, in Figure 13, we should not allow the marking process to go from $V_4$ to $T_3$ and from $V_7$ to $V_5$. This can be ensured, respectively by keeping machine $m_1$ idle when AGV is transporting a part from $m_1$ to $m_2$ and by keeping the AGV idle when machine $m_2$ is busy. Such operating rules may often lead to underutilization of resources and may also be difficult to formulate if the number of markings is very large. However, whenever feasible, this approach guarantees complete prevention of deadlocks.

(2) By eliminating all the vanishing markings from the above reachability graph as outlined in [7], we obtain a continuous time Markov chain with absorbing states. Using the theory of such Markov chains [6], we can evaluate the performance of an FMS in the presence of deadlocks. The performance measures of interest here would be: probability of occurrence of a particular deadlock (among several possible deadlocks), expected time to deadlock, and expected number of parts produced before the occurrence of a deadlock.

(3) If deadlock recovery mechanisms have been incorporated in the FMS, then we can again use GSPN models to capture deadlock recovery. We can then compute the FMS performance in the presence of deadlocks and the overheads caused by recovery schemes.

In summary, stochastic modelling can provide useful information about the performance of an FMS in the presence of deadlocks and also can suggest deadlock prevention schemes.

7. CONCLUSIONS

We have surveyed the principal modelling tools for FMSs and also presented, through examples, their use in the evaluation of performance of FMSs. Each tool by itself currently constitutes a large area of active research.

The research surveyed here concentrates most on analysis after a configuration has been specified. Arriving at configurations which meet a set of stated specifications is more an art than science as it stands now. Efforts for developing systematic methods of configuring FMSs for given part types, throughputs, lead time, quality and cost would be worthwhile.

As we have commented often in the text of the paper, FMSs lead to large models with varying time scales. Computational methods for solving such models in a reasonable time and effort is an issue. A time honored way of conquering largeness is through decomposition and aggregation. Systematic methods are needed which use system physical structure to naturally decompose systems and use the aggregated models for total system performance evaluation. Solutions of these models on parallel machines are slowly emerging. Tool boxes, which integrate various performance evaluation tools, would help in managing a large number of distinct models suitable for different settings. Integrated modelling techniques, combining the best features of product form QNs and GSPNs, as in [48], also hold promise.

Blocking, deadlock, and failures are all generic issues in the FMS context. Performance evaluation of FMSs in the presence of these needs attention.

REFERENCES

4. Y.C. Ho, Editor, Dynamics of discrete event systems, Special Issue of Proceedings of the IEEE 77 (1) (January, 1989).


