Monte Carlo Simulation to Compare Markovian and Neural Network Models for Reliability Assessment in Multiple AGV Manufacturing System

Mohammad Saidi-Mehrabad$^{a,\ast}$, Hamed Fazlollahtabar$^b$

$^a$ Professor, Faculty of Industrial & Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
$^b$ Assistant Professor, Faculty of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Received 20 May, 2014; Revised 09 April, 2015; Accepted 24 June, 2015

Abstract

We compare two approaches for a Markovian model in flexible manufacturing systems (FMSs) using Monte Carlo simulation. The model, which is a development of Fazlollahtabar and Saidi-Mehrabad (2013), considers two features of automated flexible manufacturing systems equipped with automated guided vehicle (AGV), namely, the reliability of machines and the reliability of AGVs in a multiple AGV jobshop manufacturing system. The current methods for modeling reliability of a system involve determination of system state probabilities and transition states. Since the failure of the machines and AGVs could be considered in different states, a Markovian model is proposed for reliability assessment. The traditional Markovian computation is compared with a neural network methodology. Monte Carlo simulation has verified the neural network method having better performance for Markovian computations.

Keywords: Reliability assessment, Markovian model, Neural network, Monte Carlo simulation

1. Introduction

Traditional manufacturing has relied on dedicated mass-production systems to achieve high production volumes at low costs. As living standards improve and the demands for new consumer goods rise, manufacturing flexibility gains prominence as a strategic tool for the rapidly changing markets. Flexibility, however, cannot be properly incorporated in the decision-making process if it is not well defined and measured in a quantitative manner. Flexibility in its most rudimentary sense is the ability of a manufacturing system to respond to changes and uncertainties associated with the production process (Miettinen et al., 2010; Kumar and Sridharan, 2009; Das et al., 2009). A comprehensive classification of eight flexibility types was proposed in Browne et al. (1984). Flexible manufacturing systems (FMS) are crucial for modern manufacturing to enhance productivity involved with high product proliferation (Paraschidis et al., 1994). As one of the critical components of the FMS, the flexible material handling system (MHS) plays a strategic role in the implementation of the FMS (Beamon, 1998). According to Tompkins et al. (2002), about 20–50% of the total production cost is spent on material handling. This makes the subject of material handling increasingly important. In addition, all the complexity of manufacturing is passed on to the MHS. Therefore, the flexible MHS has been vital for improving the FMS to fulfill the requirements of high product proliferation (Zhao et al., 2011).

Automated manufacturing systems (AMS), which are equipped with several computer numerical control (CNC) machines and AGV-based material handling system are designed and implemented to gain the automation and efficiency of production. To make use of all features of AMS, the planning in the AMS decision making process is critical because the planning decision has influence on the subsequent decision processes such as scheduling, dispatching, etc. The planning in automated manufacturing systems can be characterized as being online and short-term nature to respond to frequently changing production orders. Given a production order, manufacturing planning function is responsible to establish a plan by decomposing the production task into a set of subtasks. An analysis of AMS dealing with changing demand can be found in Terkaj et al. (2009). An extensive review of the loading problem for an FMS can be found in Grieco et al. (2001). An early stochastic programming approach to address the short-term production planning for an FMS can be found in Terkaj and Tolio (2006). Automated Guided Vehicle System (AGVS) is becoming popular in many industrial fields because of its flexibility, reliability, safety, and contribution to the increase of productivity and to the

* Corresponding author Email address: mehrabad@qiau.ac.ir
improvement of housekeeping. However, the performance of the material handling system is significantly influenced by several operating policies. One of the important operating policies is the positioning strategy of idle vehicles on the guide path (Egbelu, 1993; Kim, 1995; Fazlollahtabar and Saidi-Mehrabad, 2013).

The objective of Aizel (2009) was to model and maximize performance of an integrated Automated Guided Vehicle System (AGVS), which is embedded in a pull type multi-product, multi-stage and multi-line flexible manufacturing system (FMS). The researcher examined the impact of guide-path flexibility on system performance through the development of three different guide-path configurations which range from dedicated to flexible relationships between automated guided vehicles (AGVs) and machine/assembly station resources. The system was modelled using coloured Petri net method (CPN) and the simulation results led to identify the resource redundancy which can be rectified to achieve lower overall cost of the system through the development of flexible guide-path configurations. The study was extended to seek global near-optimal conditions for each guide-path configuration using response surface method, which yields improvements in system throughput and cycle time along with a decrease in the numbers of AGVs. Material handling in manufacturing systems is becoming easier as the automated machine technology is improved. Nowadays, most of the research aims at increasing the flexibility and improving the performance of the automatic guided vehicle (AGV). Yahyaei et al. (2010) designed and made AGV in the Industrial Control Laboratory in Royce Lab at the University of Manchester Institute of Science and Technology. For controlling the navigation of the AGV, a newly developed controller integrated fuzzy logic with programmable logic controller was used. By using integrated fuzzy logic controller with programmable logic controller (IFPLC), the flexibility of AGV was increased and they achieved great advantages. Since that AGV used programmable logic controller and fuzzy logic controllers together, it proved usefulness for factories which implement flexible manufacturing system (FMS). Online maintenance and sending the commands to other machines from AGV and so on were the advantages that can be used in FMS.

In most manufacturing systems, decision making is worked out at several stages of design, planning and operation. The role of performance modeling is significant in advanced manufacturing systems from economic viewpoints (Yang, 2011). However, events such as machine breakdown, changes in part type and volume, tool replacement, raw material and other short interruptions are effective on the desired performance of a manufacturing system. This problem is critical due to its impacts on the capacity of the system (Stoop and Wiers, 1996). Researches on the automated manufacturing systems imply that the machine failure is the major problem in analyzing system performance in comparison with other factors like raw material, equipment, software and workers (Sanchez, 1994). Therefore, reliability considerations should be taken into account for manufacturing system analysis. Researchers who studied this problem include Hildebrandt (1980), Kimemia (1982), Liberopoulos (1993), Viswanadham and Narahari (1992), Perkins et al. (1994), Vinod (1983), Vinod and Solberg (1984), Choi and Lee (1998). Salehipour et al. (2011) presented a new solution framework to locate the workstations in the tandem automated guided vehicle (AGV) systems. So far, the research has focused on minimizing the total flow or minimizing the total AGV transitions in each zone. The authors focused on minimizing total cumulative flow among workstations. This objective allocates workstations to an AGV route such that total waiting time of workstations to be supplied by the AGV is minimized. They developed a property which simplified the available mathematical formulation of the problem. Also a development in a heuristic algorithm was proposed for the problem. Computational results showed that the heuristic could yield very high-quality solutions and in many cases optimal solutions.

An automated manufacturing system (AMS) is a complex network of processing, inspecting, and buffering nodes connected by system of transportation mechanisms. For an AMS, it is desirable to be capable to increase or decrease the output with the rise and fall of demand. Such specifications show the complexity of decision making in the field of AMSs and the need for concise and accurate modeling methods. Therefore, Fazlollahtabar et al. (2010) proposed a flexible jobshop automated manufacturing system to optimize the material flow. The flexibility was on the multi-shops of the same type and also multiple products that can be produced. An automated guided vehicle was applied for material handling. The objective was to optimize the material flow regarding the demand fluctuations and machine specifications.

Fazlollahtabar and Mahdavi-Amiri (2012) proposed an approach for finding an optimal path in a flexible jobshop manufacturing system considering two criteria of time and cost. A network was configured in which the nodes are considered to be the shops with arcs representing the paths among the shops. An automated guided vehicle functioned as a material handling device through the manufacturing network. The expert system for cost estimation was based on fuzzy rule backpropagation network to configure the rules for estimating the cost under uncertainty. A multiple linear regression model was applied to analyze the rules and find the effective rules for cost estimation. The objective was to find a path minimizing an aggregate weighted unscaled time and cost criteria. A fuzzy dynamic programming approach was presented for computing a shortest path in the network. Then, a comprehensive economic and reliability analysis was worked out on the obtained paths to find the optimal producer’s behavior.

Fazlollahtabar and Mahdavi-Amiri (2013) proposed an approach for finding an optimal path in a flexible jobshop manufacturing system considering two criteria of time
and cost. With rise in demands, advancement in technology and increase in production capacity, the need for more shops persists. Therefore, a flexible jobshop system has more than one shop with the same duty. The difference among shops with the same duty is in their machines with various specifications. A network was configured in which the nodes were considered to be the shops with arcs representing the paths among the shops. An automated guided vehicle (AGV) functioned as a material handling device through the manufacturing network. To account for uncertainty, the authors considered time to be a triangular fuzzy number and applied an expert system to infer cost. The objective was to find a path minimizing both the time and cost criteria, aggregatedly. Since time and cost have different scales, a normalization procedure was proposed to remove the scales. The model being biobjective, the analytical hierarchy process weighing method was applied to construct a single objective. Finally, a dynamic programming approach was presented for computing a shortest path in the network. The efficiency of the proposed approach was illustrated by a numerical example. Since the manufacturing systems experience different failure states, therefore considering these states in modeling a reliability problem is of importance. The best way for considering system states in modeling is to employ Markovian property. Reibman (1990) stated the problem in estimating the probability of failure in different state is vital for reliability computations. The increasing demand for the reliability assessment in manufacturing systems under several random parameters has been investigated by several approaches facilitating the computations of probability estimations.

2. Statement of the Problem

The substantial economic importance of advanced manufacturing systems stimulates researchers to consider the optimal design and maintenance as well. Despite adding flexibility and more accuracy via automation, availability of the system is crucial to retain the investments and obtain a reliable performance. We consider a jobshop manufacturing system having multiple AGVs for material handling purpose. In each shop several machines perform the part processing according to a process plan. To transfer the parts among different shops, AGVs are employed. Availability of the whole system depends on the performance of machines and AGVs. Reliability is a suitable merit for the advanced technology availability and performance. The reliability of the whole manufacturing system is concerned with the reliability of the machines in shops and the reliability of the AGVs. The failure of the machines and AGVs could be considered in different states. The failure causes for machines are (Fazlollahtabar and Saidi-Mehrabad, 2013):
- Equipment deficiency
- Inappropriate part specifications

Also, the failures of AGVs are due to:
- Carrier overload
- Guide path fracture

Using Markovian modelling, we can configure the transition diagram and the corresponding matrix. The result of the Markovian process is the failure probability for machines and AGVs. These probabilities are applied in reliability computations. For reliability, first we conceptualize different scenarios exist in the proposed manufacturing system. The shops are in parallel since the parts are disseminated through the system according to the process plan. The sequence of machines in a shop may be important or unimportant, i.e., the part processing in a shop should be performed sequentially on the machines or the sequence is not important and parallel machining is possible. Therefore, two separate cases of series and parallel should be modeled. AGVs are in series since if one AGV break down then the whole system should wait until the AGV is repaired or taken out of the system (Fazlollahtabar and Saidi-Mehrabad, 2013).

The aim of the decision maker is to maximize the performance of the whole system. To achieve the aim, two objectives namely maximizing the total reliability of machines in shops in the whole jobshop system and the maximizing the total reliability of the AGVs, should be investigated. Also, for the economic viewpoint of the system performance the third objective is to minimize the total repair cost in the system. As a unit (machine or AGV) in the system is broken down, the repair should be performed on it for preparing it to function.

2.1. Markovian Reliability

It is necessary to incorporate reliability into the model to ensure the level of service for each machine in each shop and the AGVs. For modeling reliability, the approach of Ball and Lin (1993) is adopted and further extended. The reliability is defined as the probability that the system works until time t. If a machine in a shop is broken down, it can be regarded as a failure. A desired level of reliability can be achieved by limiting the failure probabilities. This approach for handling reliability is called the method of chance constraints and was initially suggested by Charnes and Cooper (1959) in the context of mathematical programming. The use of chance constraints in vehicle routing problem was illustrated in Stewart and Golden (1983). Carbone (1974) used chance constraints for selecting multiple facilities under normally distributed demand. The model minimized an upper bound on the total demand-weighted distance while ensuring that constraint was satisfied with specified chance or probability. Shiode and Drezner (2003) used a similar approach in a competitive location problem on a tree network.

It is assumed that the reliability of each machine type and the AGV are independently according to Exponential distributions. Also, J is total types of machines, i.e., drilling machines, turning machines, bending machines
show three machine types. We discuss the reliability based model as follows:

\[ R_j(t) : \text{The probability that machine type } j \text{th works until time } t. \]

\[ R(t)_{\text{ave}} = \begin{cases} 
1 - \prod_{j=1}^{J} (1 - R_j(t)), & \text{when machines in each shop are in parallel case} \\
\prod_{j=1}^{J} R_j(t), & \text{when machines in each shop are in series case} 
\end{cases} 
\]

(1)

In our proposed problem, AGVs are series and the machine types in each shop may be in parallel or series cases and the shops are parallel, i.e., a composite system is configured. Therefore, the reliability of the system is as follows (Fazlollahtabar and Saidi-Mehrabad, 2013):

\[ 1 - \prod_{j=1}^{J} (1 - R_j(t)) \geq \alpha , \]

where \( \alpha \) is the lower bound for a desirable reliability of the system until time \( t \). The desirable reliability level is obtainable having a capable maintenance team using different maintenance policies. As previously assumed, the reliability of each machine type and AGV are independently according to Exponential distribution:

\[ R_j(t) = e^{-\theta_j t}, \]

(3)

where \( \theta_j \) is the exponential parameter for machine type or AGV breakdown’s failure rate. Then,

\[ 1 - \prod_{j=1}^{J} (1 - e^{-\theta_j j}) \geq \alpha . \]

(4)

It is obvious that to obtain a higher level of reliability, more cost is incurred to the system. Hence, a cost function \( (C_j(t)) \) is defined to keep machine type \( j \)th reliable until time \( t \). For the whole system we have:

\[ \sum_{j=1}^{J} C_j(t). \]

(5)

3. Mathematical Formulation

In this section, we construct the proposed failure state diagrams and matrices for machines and AGVs' using Markov system, separately. A Markov system is a system that can be in one of several (numbered) states, and can pass from one state to another each time step according to fixed probabilities. If a Markov system is in state \( i \), there is a fixed probability, \( p_{ij} \), of it going into state \( j \) the next time step, and \( p_{ij} \) is called a transition probability. A Markov system can be illustrated by means of a state transition diagram, which is a diagram showing all the states and transition probabilities. The entries in each row add up to 1.

First, we configure the machines’ state diagram. As stated before, the machines may be broken down in three states, namely, (A) amateur operator, (B) equipment deficiency, and (C) inappropriate part specifications. The state transition diagram for machines is shown in Figure 1.

As a result the corresponding transition matrix \( P_{ij} \) is,

\[ P_{ij} = \begin{bmatrix} 1 - \alpha - \beta & \alpha & \epsilon \\ \beta & 1 - \beta - \gamma & \gamma \\ \nu & \delta & 1 - \delta - \nu \end{bmatrix}. \]

(6)

where \( \alpha, \beta, \gamma, \delta, \epsilon, \) and \( \nu \) are the transition probability from the three states given in Figure 1. Using the probability transition matrix and the limiting probability we obtain each state’s occurrence probability as follows.

\[ \pi_j = \sum_{i=1}^{3} \pi_i P_{ij}, \text{ for } j=1,2,3 \]

(7)

\[ \sum_{j=1}^{3} \pi_j = 1 \]

(8)

Using these probabilities, we can compute the reliability of each state helping us to assess the total reliability of the system. We also can compute the long run probability for each state using steady state distribution given below.
having A+B+C=1, and note that A, B and C are the states.

The same computations exist for AGVs different failure states (as shown in Figure 2), while we stated 2 states, i.e., we have two state probability and a 2x2 transition matrix.

\[
\begin{bmatrix}
A & B & C
\end{bmatrix}
= \begin{bmatrix}
1 - \alpha - \delta & \alpha & \delta \\
\beta & 1 - \beta - \gamma & \gamma \\
\nu & \delta & 1 - \delta - \nu
\end{bmatrix}
\] (9)

Now, for reliability we have, \( R(t) = 1 - F(t) \),

\[
P_s = \begin{bmatrix}
1 - \vartheta & \vartheta \\
\omega & 1 - \omega
\end{bmatrix}.
\] (10)

where, \( F(t) \) is the failure probability computed above as states’ probabilities. Note that, we can compute the reliability in two cases, first for current state, and second for steady state. Having the current state of the system by Markovian model and by means of neural network, we can compute the steady state probabilities. Next, we review the artificial neural network and the backpropagation neural network for our proposed work.

The aim of computing the steady state probability and reliability is to obtain an estimation of the system availability for long-run planning horizon. Therefore, it is significant for a decision maker to determine steady state reliability using the corresponding probability accurately.

3.1. Artificial neural network

Neural networks are being widely used in many fields of study. This could be attributed to the fact that these networks attempt to model the capabilities of human brains. Since the last decade, neural networks have been used as a theoretically sound alternative to traditional statistical models. Although neural networks (NNs) originated in mathematical neurobiology, the rather simplified practical models currently in use have moved steadily towards the field of statistics. A number of researchers have illustrated the connection of neural networks to traditional statistical models. For example,

Gallinari et al. (1991) presented analytical results establishing a link between discriminant analysis and multilayer perceptrons (MLP) used for classification problems. Cheng and Titterington (1994) made a detailed analysis and comparison of various neural network models with traditional statistical models. They showed strong associations of the feed-forward neural networks with discriminant analysis and regression models, and unsupervised networks such as self-organizing neural networks with clustering. Neural networks are being used in areas of prediction and classification, areas where regression models and related statistical techniques have traditionally been used. Ripley (1994) discusses the statistical aspects of neural networks and classifies neural networks as one of a class of flexible nonlinear regression models. Sarle (1994) translates neural network terminologies into statistical ones and shows the relationship between neural networks and statistical models such as generalized linear models, projection pursuit and cluster analysis. He explains that neural networks and statistical approaches are not competing methodologies for data analysis and there is a considerable overlap between the two. Warner and Misra (1996) present a comparison between regression analysis and neural network computation in terms of notation and implementation. They also discuss when it would be advantageous to use a neural network model in place of a parametric regression model, as well as some of the difficulties in implementation. Schumacher et al. (1996) and Vach et al. (1996) present a comparison between feed-forward neural networks and the logistic regression. The conceptual similarities and discrepancies between the two methods are also analyzed.

Artificial neural networks have been applied successfully to many manufacturing and engineering areas. Zhengrong et al. (1996) used quadratic regression to assess the results of neural network for improving the efficiency of fermentation process development. The results show that different sizes of neural nets within a certain range give an equally good prediction by using the “stopping training” technique, while quadratic regressions are sensitive to the size of the data sets. Smith and Mason (1997) mentioned that regression and neural network modeling methods have become two competing empirical model-building methods. They compared the predictive capabilities of NNs and regression methods in manufacturing cost estimation problems.

3.1.1. The Backpropagation Neural Network

The backpropagation algorithm trains a given feed-forward multilayer neural network for a given set of input patterns with known classifications. When each entry of the sample set is presented to the network, the network examines its output response to the sample input pattern. The output response is then compared to the known and desired output and the error value is calculated. Based on the error, the connection weights are adjusted. The
backpropagation algorithm is based on Widrow-Hoff delta learning rule in which the weight adjustment is done through mean square error of the output response to the sample input (Abdi et al., 1996). The general steps of backpropagation are given below.

1. Propagate inputs forward in the usual way, i.e., all outputs are computed using sigmoid thresholding of the inner product of the corresponding weight and input vectors. All outputs at stage n are connected to all the inputs at stage n+1.

2. Propagate the errors backwards by apportioning them to each unit according to the amount of the error the unit is responsible for.

We now discuss how to develop the stochastic backpropagation algorithm for the general case. The following notations and definitions are needed:

\( \vec{x}_j \): input vector for unit j (\( x_{ji} \) = ith input to the jth unit)

\( \vec{w}_j \): weight vector for unit j (\( w_{ji} = \) weight on \( x_{ji} \))

\( z_j = \vec{w}_j \cdot \vec{x}_j \): the weighted sum of inputs for unit j

\( o_j \): output of unit j (\( o_j = \sigma(z_j) \))

\( t_j \): target for unit j

Downstream(j) : set of units whose immediate inputs include the output of \( j \)

Output: Set of output units in the final layer.

Since we update after each training example, we can simplify the notation somewhat by assuming that the training set consists of exactly one example and so the error can simply be denoted by E.

We want to calculate \( \frac{\partial E}{\partial w_{ji}} \) for each input weight \( w_{ji} \) for each hidden unit \( j \). Note that \( w_{ji} \) influences just \( z_j \) which influences \( o_j \) which influences \( z_k \), \( \forall k \in \text{Downstream}(j) \), each of which influences E. So, we can write,

\[
\frac{\partial E}{\partial w_{ji}} = \sum_{k \in \text{Downstream}(j)} \frac{\partial E}{\partial z_k} \frac{\partial z_k}{\partial o_j} \frac{\partial o_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ji}} = \sum_{k \in \text{Downstream}(j)} \delta_k w_{kj} o_j (1-o_j) \]

Again, note that all the terms except \( x_{ji} \) in (16) are the same regardless of which input weight of unit \( j \) we are trying to update. Like before, we denote this common quantity by \( \delta_j \). Also, note that \( \frac{\partial E}{\partial z_k} = \delta_k \sigma(z_k) w_{kj} \) and \( \frac{\partial o_j}{\partial z_j} = o_j (1-o_j) \). Substituting them in (14),

\[
\delta_j = \sum_{k \in \text{Downstream}(j)} \delta_k w_{kj} o_j (1-o_j) \]

we obtain:

\[
\delta_k = o_j (1-o_j) \sum_{k \in \text{Downstream}(j)} \delta_k w_{kj} \]

To adapt the backpropagation algorithm on our proposed model, consider the failure causes for machines and AGVs as inputs and the current state failure probability of machines and AGVs as outputs. We train the network collecting data in different time periods and compute the importance weight for each input resulting in the
corresponding output. A configuration of the proposed neural network is shown in Figure 3.

![Diagram of Neural Network](image)

**Fig. 3. A configuration of the proposed neural network**

We are now in a position to state the backpropagation algorithm formally.

**Algorithm 1: Formal statement of stochastic backpropagation.**

(Training examples, $\eta$, $n_i$, $n_h$, $n_o$)

Each training example is of the form $\langle \vec{x}, \vec{t} \rangle$, where $\vec{x}$ is the input vector and $\vec{t}$ is the target vector, $\eta$ is the learning rate (e.g., 0.05), $n_i$, $n_h$ and $n_o$ are the number of input, hidden and output nodes, respectively. Input from unit i to unit j is denoted by $x_{ji}$ and its weight is denoted by $w_{ji}$. Create a feed-forward network with $n_i$ inputs, $n_h$ hidden units, and no output units.

**Initialize** all the weights to small random values (e.g., between -0.05 and 0.05).

**While termination condition is not met Do**

For each training example $\langle \vec{x}, \vec{t} \rangle$,

1. Input the instance $\vec{x}$ and compute the output $o_k$ of every unit.
2. For each output unit $k$, calculate
   
   $$\delta_k = o_k (1 - o_k) (t_k - o_k)$$  
   
   (19)

3. For each hidden unit $h$, calculate
   
   $$\delta_h = o_h (1 - o_h) \sum_{k \in \text{Downstream}(h)} \delta_k w_{kh}$$  
   
   (20)

4. Update each network weight $w_{ji}$ as follows:
   
   $$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$  
   
   (21)

where,

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$  

(22)

This way, we can compare the performance of backpropagation neural network and limiting distribution model for computing the steady state probabilities using the current state probabilities. To do that, a Monte Carlo simulation is applied that is explained below.

### 3.2. Monte Carlo simulation

While the results from the backpropagation neural network and limiting distribution model may be different in computing the steady state probabilities using the current state probabilities, we apply Monte Carlo simulation to verify the more appropriate model.

Monte Carlo simulation is a comprehensive approach for analyzing the behavior of some activities, plans or processes that involve uncertainty. If we face uncertain or variable market demands, fluctuating costs, variations in a manufacturing process, effects of weather on operations, or stochastic activity time, we can benefit from using Monte Carlo simulation to understand the impact of uncertainty, and to develop plans to mitigate or otherwise cope with the risk. Whenever we need to make an estimate, forecast or decision where there is significant uncertainty, we would be well advised to consider Monte Carlo simulation (Metropolis and Ulam, 1949). Also, Fazlollahtabar et al. (2010) recently applied Monte Carlo simulation in manufacturing systems for line balancing with stochastic task times.

81
Monte Carlo simulation is a method for iteratively evaluating a deterministic model using sets of random numbers as inputs. This method is often used when the model is complex, nonlinear, or involves more than just a couple uncertain parameters. The Monte Carlo method is just one of many methods for analyzing uncertainty propagation, where the goal is to determine how random variation, lack of knowledge, or error affects the sensitivity, performance, or reliability of the system that is being modeled. Monte Carlo simulation is categorized as a sampling method because the inputs are randomly generated from probability distributions to simulate the process of sampling from an actual population. So, we try to choose a distribution for the inputs that most closely matches data we already have, or best represents our current state of knowledge (Fazlollahtabar et al., 2010). The data generated from the simulation can be represented as probability distributions (or histograms) or converted to error bars, reliability predictions, tolerance zones, and confidence intervals. (see Figure 4).

4. Computational Results and Analysis

Here, a numerical example is worked out to imply the effectiveness and applicability of the proposed model. Using machines’ (amateur operator, equipment deficiency, and inappropriate part specifications) and AGVs’ (carrier overload and guide path fracture) failure probability transition matrices, and by the means of limiting probability the states occurrence probabilities for each machine or AGV can be computed.

Example 1:

Also, note that the number of machines is 10, number of shops is 4, number of AGVs is 4, and number of jobs to be processed on a product is 8. Since each of the failure states mentioned cause break down of the system, therefore the failure states are parallel. Using these probabilities, we can compute the reliability of each state using equations (1) and (2) helping us to assess the total reliability values of the system as follows:

Reliability (machine 1): 0.0124
Reliability (machine 2): 0.0303
Reliability (machine 3): 0.0435
Reliability (machine 4): 0.0054
Reliability (machine 5): 0.0641
Reliability (machine 6): 0.00398
Reliability (machine 7): 0.0287
Reliability (machine 8): 0.00703
Reliability (machine 9): 0.00239
Reliability (machine 10): 0.0197
Reliability (AGV 1): 0.0753
Reliability (AGV 2): 0.0639
Reliability (AGV 3): 0.0458
Reliability (AGV 4): 0.389

As stated earlier, another way to compute the steady state probabilities is backpropagation neural network. The input of the system is given by a one dimensional vector and the output is given by a two/three dimensional matrix. To facilitate the computations of backpropagation neural network, MATLAB 7.1 user interface, NNtool, is applied. A feedforward network is programmed with one input, ten hidden units with logistics activation function, and two/three output. Using the MATLAB 7.1 user interface NNtool, we insert the data and perform the required settings to train the data to obtain an appropriate pattern. Then, using the pattern we can approximate the output of the proposed neural network.

Outputs:

Reliability (machine 1): 0.0133
Reliability (machine 2): 0.031
Reliability (machine 3): 0.045
Reliability (machine 4): 0.0063
Reliability (machine 5): 0.0628
Reliability (machine 6): 0.00377
Reliability (machine 7): 0.0281
Reliability (machine 8): 0.00723
Reliability (machine 9): 0.00243
Reliability (machine 10): 0.0184
Reliability (AGV 1): 0.0761
Reliability (AGV 2): 0.0652
Reliability (AGV 3): 0.0463
Reliability (AGV 4): 0.396
Clearly, the back propagation computations are slightly different from the steady state equation ones. Therefore, we use Monte Carlo simulation to verify the more appropriate method. Using Monte Carlo simulation based on the steps given in section 3.2, we obtain,
Reliability (machine 1): 0.0113
Reliability (machine 2): 0.0307
Reliability (machine 3): 0.0453
Reliability (machine 4): 0.0061
Reliability (machine 5): 0.0623
Reliability (machine 6): 0.0038
Reliability (machine 7): 0.0284
Reliability (machine 8): 0.0073
Reliability (machine 9): 0.00239
Reliability (machine 10): 0.0179
Reliability (machine 11): 0.0079
Reliability (machine 12): 0.0648
Reliability (machine 13): 0.047
Reliability (machine 14): 0.391
The deviation of the limiting distribution (LD) computations and neural network (NN) with the results of Monte Carlo simulation (MC) is given in Table 1.

<table>
<thead>
<tr>
<th>Example 2:</th>
<th>The deviation comparison for example 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC vs NN</td>
<td>MC vs LD</td>
</tr>
<tr>
<td>Reliability (machine 1)</td>
<td>0.00496115 0.0032415 sum of error</td>
</tr>
<tr>
<td>Reliability (machine 2)</td>
<td>5.024302E-05 6.3939E-05 Reliability (AGV 8)</td>
</tr>
<tr>
<td>Reliability (machine 3)</td>
<td>4.80243E-05 4.3959E-05 Reliability (AGV 7)</td>
</tr>
<tr>
<td>Reliability (machine 4)</td>
<td>4.86683E-05 4.7426E-05 Reliability (AGV 6)</td>
</tr>
<tr>
<td>Reliability (machine 5)</td>
<td>5.3425E-05 5.7691E-06 Reliability (AGV 5)</td>
</tr>
<tr>
<td>Reliability (machine 6)</td>
<td>4.69815E-06 4.5873E-06 Reliability (AGV 4)</td>
</tr>
<tr>
<td>Reliability (machine 7)</td>
<td>4.325E-05 3.4691E-06 Reliability (AGV 3)</td>
</tr>
<tr>
<td>Reliability (machine 8)</td>
<td>6.81436 E-06 5.7946E-06 Reliability (machine 20)</td>
</tr>
<tr>
<td>Reliability (machine 9)</td>
<td>8.81006 E-07 7.3966E-05 Reliability (machine 19)</td>
</tr>
<tr>
<td>Reliability (machine 10)</td>
<td>3.98363E-06 4.723E-07 Reliability (machine 18)</td>
</tr>
<tr>
<td>Reliability (machine 11)</td>
<td>7.0379E-05 5.574E-06 Reliability (machine 17)</td>
</tr>
<tr>
<td>Reliability (machine 12)</td>
<td>6.18046E-07 6.3196E-07 Reliability (machine 16)</td>
</tr>
<tr>
<td>Reliability (machine 13)</td>
<td>4.54691E-07 5.63946E-07 Reliability (machine 15)</td>
</tr>
<tr>
<td>Reliability (machine 14)</td>
<td>5.63946E-07 4.53469E-07 Reliability (machine 14)</td>
</tr>
<tr>
<td>Reliability (machine 15)</td>
<td>5.723E-07 4.3966E-05 Reliability (machine 13)</td>
</tr>
<tr>
<td>Reliability (machine 16)</td>
<td>6.9863E-06 5.3425E-05 Reliability (machine 12)</td>
</tr>
<tr>
<td>Reliability (machine 17)</td>
<td>7.23146E-07 5.3425E-05 Reliability (machine 11)</td>
</tr>
<tr>
<td>Reliability (machine 18)</td>
<td>6.3039E-05 5.024302E-05 Reliability (machine 10)</td>
</tr>
<tr>
<td>Reliability (machine 19)</td>
<td>3.5469E-07 2.868E-06 Reliability (machine 9)</td>
</tr>
<tr>
<td>Reliability (machine 20)</td>
<td>6.81436 E-06 5.7946E-06 Reliability (machine 8)</td>
</tr>
</tbody>
</table>

The results show better performance of neural network method. The reason could be the constraints of the limiting distribution model. For example, the transition matrix should not have zero elements, ergodic state and closed state. Since, neural network employs several training data sets to adapt the appropriate pattern; its results are more valid. Thus, using Monte Carlo simulation we could verify the proposed backpropagation neural network for the Markovian steady state computations. But, the computational efforts and past data are the significant requirements of NN making its application questionable. Therefore the condition of the system under study implies employing the appropriate method. If past data exist and no limit of computational effort is allowed, then NN is useful since having more confident results. On the other hand, if past data does not exist and time limit for reporting the reliability computations to management is obliged, then using steady state Markovian method is encouraged.

Table 2
The deviation comparison for example 2

<table>
<thead>
<tr>
<th>MC vs NN</th>
<th>MC vs LD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability (machine 1)</td>
<td>3.431E-06 4.7693E-05</td>
</tr>
<tr>
<td>Reliability (machine 2)</td>
<td>4.831E-06 6.2173E-06</td>
</tr>
<tr>
<td>Reliability (machine 3)</td>
<td>2.868E-06 3.5232E-05</td>
</tr>
<tr>
<td>Reliability (machine 4)</td>
<td>5.574E-06 7.0379E-05</td>
</tr>
<tr>
<td>Reliability (machine 5)</td>
<td>3.128E-06 5.264E-05</td>
</tr>
<tr>
<td>Reliability (machine 6)</td>
<td>6.3684E-07 5.63E-06</td>
</tr>
<tr>
<td>Reliability (machine 7)</td>
<td>7.469E-06 6.6901E-06</td>
</tr>
<tr>
<td>Reliability (machine 8)</td>
<td>3.723E-07 4.9836E-06</td>
</tr>
<tr>
<td>Reliability (machine 9)</td>
<td>6.6946E-07 5.4569E-07</td>
</tr>
<tr>
<td>Reliability (machine 10)</td>
<td>4.3966E-05 4.181006E-07</td>
</tr>
<tr>
<td>Reliability (machine 11)</td>
<td>5.7123E-07 3.8463E-06</td>
</tr>
<tr>
<td>Reliability (machine 12)</td>
<td>6.39046E-07 3.5469E-07</td>
</tr>
<tr>
<td>Reliability (machine 13)</td>
<td>6.3196E-05 6.18046E-07</td>
</tr>
<tr>
<td>Reliability (machine 14)</td>
<td>4.723E-07 3.9836E-06</td>
</tr>
<tr>
<td>Reliability (machine 15)</td>
<td>5.63946E-07 4.54691E-07</td>
</tr>
<tr>
<td>Reliability (machine 16)</td>
<td>2.3966E-05 3.18116E-07</td>
</tr>
<tr>
<td>Reliability (machine 17)</td>
<td>5.723E-07 6.9863E-06</td>
</tr>
<tr>
<td>Reliability (machine 18)</td>
<td>3.6946E-07 4.53469E-07</td>
</tr>
<tr>
<td>Reliability (machine 19)</td>
<td>7.969E-05 8.81006E-07</td>
</tr>
<tr>
<td>Reliability (machine 20)</td>
<td>5.7946E-06 6.81436E-06</td>
</tr>
<tr>
<td>Reliability (AGV 1)</td>
<td>2.1873E-06 2.29815E-06</td>
</tr>
<tr>
<td>Reliability (AGV 2)</td>
<td>3.4691E-06 3.25E-05</td>
</tr>
<tr>
<td>Reliability (AGV 3)</td>
<td>3.0426E-05 4.06383E-05</td>
</tr>
<tr>
<td>Reliability (AGV 4)</td>
<td>4.5873E-06 4.69815E-06</td>
</tr>
<tr>
<td>Reliability (AGV 5)</td>
<td>5.7691E-06 5.3425E-05</td>
</tr>
<tr>
<td>Reliability (AGV 6)</td>
<td>4.7426E-05 4.86683E-05</td>
</tr>
<tr>
<td>Reliability (AGV 7)</td>
<td>4.3959E-05 4.80243E-05</td>
</tr>
<tr>
<td>Reliability (AGV 8)</td>
<td>6.3039E-05 5.024302E-05</td>
</tr>
</tbody>
</table>

| sum of error | 0.0032415 0.00496115 |

The results show better performance of neural network method. The reason could be the constraints of the limiting distribution model, e.g., the transition matrix should not have zero elements, ergodic state and closed state. Since, neural network employs several training data sets to adapt the appropriate pattern; its results are more valid. Thus, using Monte Carlo simulation we could verify the proposed backpropagation neural network for the Markovian steady state computations. But, the computational efforts and past data are the significant requirements of NN making its application questionable. Therefore the condition of the system under study implies employing the appropriate method. If past data exist and no limit of computational effort is allowed, then NN is useful since having more confident results. On the other hand, if past data does not exist and time limit for reporting the reliability computations to management is obliged, then using steady state Markovian method is encouraged.
5. Conclusions

We proposed a Markovian model for flexible manufacturing systems (FMSs). The model considered two features of automated flexible manufacturing systems equipped with automated guided vehicle (AGV), namely, the reliability of machines and the reliability of AGVs in a multiple AGV jobshop manufacturing system. We made use of current state transition matrix for the failure of the machines and AGVs in different states. Therefore, a Markovian model was proposed for reliability assessment. Also, for steady state probability computations, the limiting theorem was compared with adapted backpropagation neural network showing neural network’s effectiveness. We verified the confidence of NN computational method using Monte Carlo simulation. The computational results illustrated the applicability of our proposed model. The results of two examples implied the application of the two discussed approaches is fully based on the condition of the system under study and the limitation obliged by the management. The limitation of the study was the availability of required data to handle the reliability computations for various purposes (considering past data or real time computations). The managerial implications of the model can be treated as the capability of the proposed methodology to provide the reliability of the system at any time, help the decision maker to make suitable maintenance policy and determine the most defected AGVs and machines for replacement or specification improvement. As future research developing a hybrid NN-Markovian method for reliability computations is forehead.

References


