International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

Analysis of Flexible Manufacturing System using Petri Nets to design a Deadlock Avoidance Policy

Mr. Shashank Satish More
Student of M.E., Department of Mechanical-Production Engineering,
KIT's college of Engineering, Gokul Shirgaon, Kolhapur, Maharashtra, India.

Prof. Dr. S. G. Bhatwadekar
Associate Professor, Department of Production Engineering,
KIT's college of Engineering, Gokul Shirgaon, Kolhapur, Maharashtra, India.

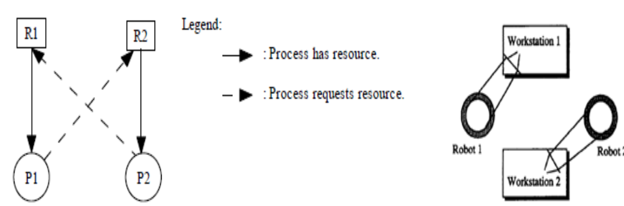
Prof. Murali M.Ambekar
Assistant Professor, Department of Mechanical Engineering,
KLE's college of Engineering, Chikodi, Belagavi, Karnataka, India.

Abstract
This paper illustrates the Petri net modeling of a conceptual model of flexible manufacturing system (FMS) and addresses the problem of deadlock by establishing deadlock prevention policy. The first part introduces the problem of deadlock in FMS, deadlock handling strategies and basic concepts of petri net modeling. Literature review is presented in the second part and third part shows an illustration of Petri net modeling of a case study model of FMS consists ofthree robots, two machines, two conveyors and an inspection center processing two parts.Type A product undergoes milling operation only whereas Type B product undergoes drilling and then milling. Both the parts are inspected before moving to conveyor out. Reachability graph is then generated for this model and is used to detect the possible deadlock states. Then deadlock prevention policy is established for this conceptual FMS model based on reachability graph analysis. Finally deadlock avoidance policy is generated with help of MATLAB.

Keywords: Deadlock, FMS, Petri net, Reachability graph, deadlock prevention, deadlock avoidance, MATLAB, GUI.

Introduction
Manufacturers must adapt to changes in the production environment as well as in the market in order to achieve and maintain competitiveness. Effectively designing and operating an automated manufacturing system (AMS) is important for manufacturers to reach this goal. An AMS conglomerates of machine tools, robots, buffers, fixtures, automated guided vehicles (AGVs), and other material-handling devices. Different types of parts enter the system at discrete points of time and are processed concurrently; these parts cause a high degree of resource sharing [11]. It is a difficult to predict the behaviour of manufacturing systems without modelling, analyzing, and control techniques. Therefore, several techniques have been developed to describe the behaviour of manufacturing systems. One of which are Petri nets (PNs). Petri nets are a powerful graphical tool for modeling and analyzing concurrent, parallel, simultaneous, synchronous, distributed, and resource sharing systems. There are manyadvantages of using Petri nets such as enable an easy visualization of complex systems, can model a system hierarchically and can analyze qualitative and quantitative aspects of the system, qualitative analysis searches for structural properties like the absence of deadlocks, the absence of overflows or the presence of certain mutual exclusions in case of resource sharing. Quantitative analysis looks for performance properties such as throughput, utilization rates, average queue lengths, or average completion times.

Deadlocks in FMS
Deadlock is a state where a set of parts are in "Circular waiting" situation, i.e., a situation where a set of two or more processes keep waiting indefinitely for other processes in the same set to release resources. But in context of FMS resources refer to machines, buffers, AGVs, robots, fixtures etc. and process refer to operations to be performed on the raw material at the work centers.



(a) (b)

Figure 1: A typical deadlock scenario.

Figure 1shows an example of a typical deadlock scenario. The deadlock situation can disrupt the entire system and makes the automated system stop working. To recover from deadlock one has to shutdown the system where deadlock has occurred and forcibly remove the part which is being processed and then restart the system in order to return to normalcy. Thus occurrence of deadlock causes unpredictably long down-time and causes under utilization of resources and may also lead to catastrophic results in safety-critical systems. Deadlocks in a FMS are in general considered to be a result of: [9]

1) Deficiency of system resources (Excessive sharing of resources).
2) An inappropriate order of process execution, and
3) Improper resource allocation logic.

There are four necessary conditions for a deadlock to occur [12], known as the Coffman conditions:

1) Mutual exclusion condition: a resource can only get involved in one process at a time.
2) Hold and wait condition: processes (operation) already holding resources may request new resources;
3) No preemption condition: no resource can be forcibly released from a process holding it, when processing is under progress and resources can be released only by the explicit action of the process;
4) Circular wait condition: two or more processes form a circular chain where each process waits for a resource that the next process in the chain holds.

A deadlock will never occur if one of these conditions is not satisfied. The physical characteristics and technical background of an FMS which is a discrete event system, show that the first three deadlock conditions always hold and any method which guarantees that the fourth condition (the circular wait) does not hold, is a valid solution to tackle problem of deadlock in FMS.

Deadlocks Handling Strategies

There are three major approaches to manage a deadlock situation [12] namely deadlock prevention, deadlock avoidance, and deadlock detection and recovery. Deadlock prevention refers, to static resource allocation policies to eliminate completely the possibility of deadlock occurrence. This approach is too much conservative and leads to underutilization of system. Deadlock avoidance involves dynamic resource allocation, so as to avoid deadlock just before it occurs. This approach is widely accepted because it leads to better utilization of resources than prevention policies. In deadlock detection and recovery, however, resources are granted to a process without any check. A deadlock detection algorithm determines whether a set of processes is deadlocked. If a deadlock is found, the system recovers from it by aborting one or more deadlocked processes. A good deadlock detection algorithm is the one, which detects all possible deadlocks and does not report nonexistent deadlocks.

Petri Nets as a Modeling Tool

Petri nets are bipartite directed graphs, in which there are two types of vertices known as places and transitions. The places are represented by hollow circle and transitions are represented by vertical bar. Places usually represent a state of the system, conditions or a resource and transitions represent events. A token is a small solid dot (or solid circle), presence of token in a place means the condition represented by the place is satisfied or a resource represented by the place is available. Places are connected to transitions by directed arcs and they represent input and output function. A firing of transition indicate occurrence of an event represented by it and transition is said to be enabled for firing, if all the input places of a transition are having tokens in them, in other words an event can occur only if all the conditions are satisfied or all the resources required for an event are available. Petri nets, as graphical and mathematical tools, provide a uniform environment for modeling, analysis and design of discrete event systems [13]. It is their simplicity and ability to model sequential execution, concurrency, conflict, synchronization and merging like situations which makes them suitable to model FMS.

Literature review

Coffman et. al [1] derives that the deadlock recovery might be costly one, in deadlock prevention the resources may remain in underutilization so deadlock avoidance might be quite efficient than above two policies as it has property of temporary resource allocation which is feasible and less costly. Narahari and Viswanadham [2] present an approach for modelling and analyzing flexible manufacturing systems (FMSs) using Petri nets. In this approach, they first build a (Petri net model ~PNM) of the given FMS in a bottom-up fashion and then analyze important qualitative aspects of FMS behaviour such as existence/absence of deadlocks and buffer overflows. They illustrate their approach using two typical manufacturing systems: an automated transfer line and a simple FMS. Viswanadham and Narahari [5] show that prevention and avoidance of FMS deadlocks can be implemented using Petri net models. For deadlock prevention, they used the reachability graph of a Petri net model of the given FMS, whereas for deadlock avoidance, they also proposed a Petri net-based on-line real time controller. Wu and Zhuang [7] study that Flexible Manufacturing Systems (FMS's), belong to the class of discrete event dynamic systems. The problem of collision and deadlock avoidance can be investigated using Petri nets which are powerful techniques suitable for modeling concurrent processes. A Petri net based approach is employed in this paper to model, detect and avoid collisions and deadlocks in FMS's.

Lawley et al. (1998) [9] describes Configuration flexibility, the ability to quickly modify manufacturing system components and their logical relationships, requires automatic generation of control executables from high level system specifications. The objective of their research is to develop rapidly configurable control policies that guarantee deadlock free FMS operation. Li et al. [16] investigate research surveys the state-of-the-art deadlock control strategies for automated manufacturing systems by reviewing the principles and techniques involved in preventing, avoiding, and detecting deadlocks. Ambekar and Bhatwadekar [19] illustrate the Petri net modeling of a conceptual model of flexible manufacturing

International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

system (FMS) and address the problem of deadlock by establishing deadlock prevention policy. This work illustrates that the Reachability graph technique can be effectively use for analyzing and establishing a deadlock prevention policy for an FMS.Abdul-Hussin [20] describes that the deadlock prevention method is caused by the unmarked siphons, during the Petri nets are an effective way to model, analyze, simulation and control deadlocks in FMS is present in this work. Kaid et. al [21] demonstrate a comprehensive overview for applications of Petri nets (PN) and their extensions in modeling, analyzing, and control of manufacturing systems are present.This paper presents an efficient siphon for control structure analysis of PN in MATLAB Toolbox.

From the literature review one can conclude that most of the works addressing deadlock problem using Petri nets are based on structural analysis of Petri nets and few works are based on reachability graph analysis and are restricted to very small systems. Structural methods eliminate the derivation of the state space, so they avoid the "state explosion" problem, but they cannot provide as much information as the reachability approach does.

Petri net modeling of a Case Study of FMS

A case study of FMS consisting three robots, two machines, two conveyors and an inspection center processing two parts types is considered for illustration using Petri net. The Figure 2shows the schematic layout of the case study FMS model.

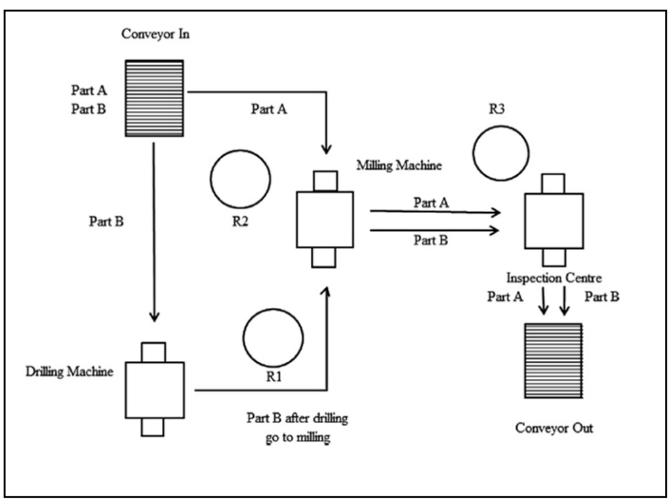


Figure 2: FMS model from Case study[18]

The Figure 2 shows block diagram of FMS in this study. It consists of three robots, two machines, i.e. a drilling machine and a milling machine and an inspection center. The system also consists of two conveyors, conveyor in and conveyor out. The system has operational flexibility in the sense that two types of products are produced in the system using two different process operations. Type A product undergoes milling operation only whereas Type B product undergoes drilling and then milling. Both the parts are inspected before moving to conveyor out. Robot 2 loads the milling and drilling machine from conveyor in, and Robot 1 loads the part B onto the milling machine after it completes the drilling operation. Robot 3 unloads milling machine and loads the inspection center it also unloads it to conveyor out. The Figure 3 shows the Petri net model of the FMS and Table 1 shows the interpretations of places and transitions.

Table 1: Definition of Places and Transition in PN model

Places
R1: Robot 1 is available R2: Robot 1 is available R3: Robot 1 is available Milling: Milling machine is available Drilling: Drilling machine is available Inspection: Inspection machine is available Place A: Raw material A is available Place 1: Robot 2 is loading raw material A on milling machine. Place 2 : Machining operation on milling machine is under progress. Place 3: Robot 3 is loading material A on inspection machine. Place 4: Inspection of material A is under progress. Place 5: Robot 3 is unloading finished material A on conveyor out. Place B: Raw material B is available Place 6: Robot 2 is loading raw material B on drilling machine. Place7: Machining operation on drilling machine is under progress. Place 8: Robot 1 is loading raw material B on milling machine. Place 9: Machining operation on milling machine is under progress. Place 10: Robot 3 is loading material B on inspection machine. Place 11: Inspection of material B is under progress. Place 12: Robot 3 is unloading finished material B on conveyor out.
Transitions
T1: Robot 2 starts loading raw material A on milling machine. T2: Milling machine starts machining raw material A. T3: Machining is over; Robot 3 unloads it and starts loading material A on Inspection center. T4: Inspection of material A starts at Inspection center. T5: Inspection is over Robot 3 starts unloading Inspection center. T6: Unloading of material A completed by Robot 3 from Inspection center. T7: Robot 2 starts loading raw material B on drilling machine. T8: Drilling machine starts machining raw material B. T9: Machining is over; Robot 1 unloads it and starts loading material B on milling machine. T10: Milling machine starts machining material B. T11: Machining is over; Robot 3 starts unloading part B from milling machine and loads it on Inspection center. T12: Inspection of material B starts at Inspection center. T13: Inspection is over; Robot 3 starts unloading material B from Inspection center. T14: Unloading of material B completed by Robot 3 from Inspection center.

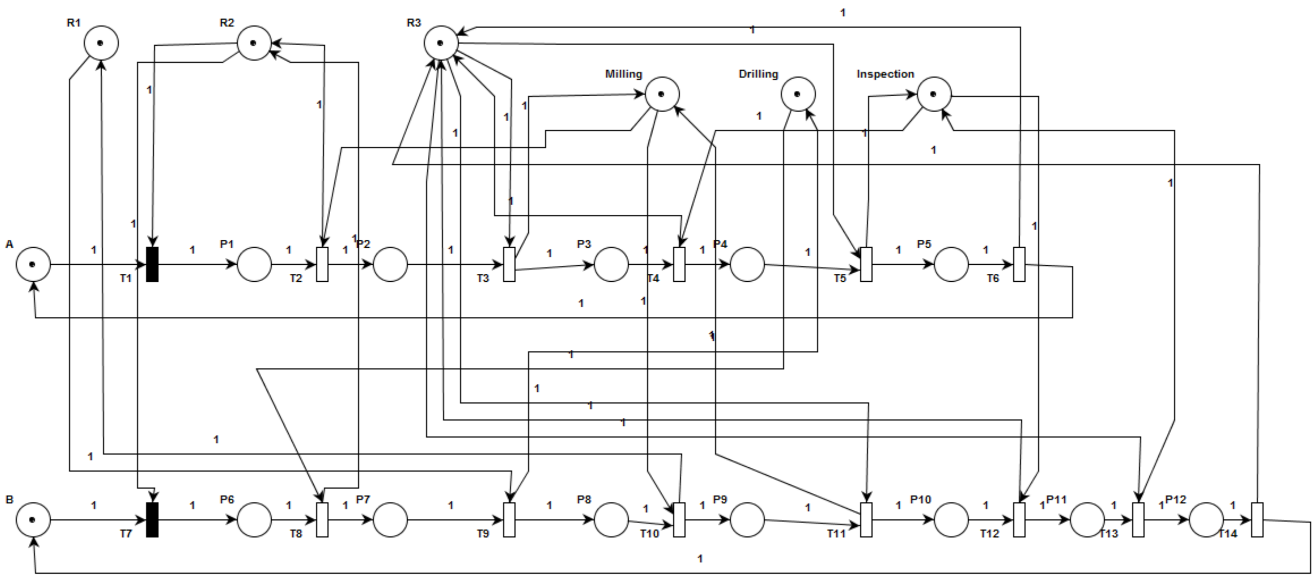


Figure 3: Petri Net model of the case study of FMS

Results and Discussions

The Figure 4 shows the reachability graph of Petri net model of FMS of case study 2. From the reachability graph, one can easily detect two such deadlock states namely, S29 and S30.The corresponding critical states are S24 and S25. As explained in chapter 4, our deadlock prevention policy is when system reaches critical state, avoid firing those transitions which leads to deadlock states. For example when the system is in state S24, two transitions T11 and T5 are enabled. If firing of T11 occurs, then system reaches deadlock state S29. Hence our deadlock prevention policy would be to avoid firing T11. The physical interpretation of this is, when system is at state S24, actually both milling machine and inspection center are busy, after they complete the process they need to be unloaded. Firing of T11 indicates, Robot 3 unloads part B from milling machine and starts loading on Inspection center. Firing of T5 indicates, robot 3 unloads finished part A from inspection center. Suppose firing of T11 occurs, i.e. if robots 3 unloads millingmachine and starts loading it on inspection center, however inspection center is waiting for the same robot 3 to unload part A from it and since robot can handle only one part at a time the system will be deadlocked.

Thus deadlock prevention policy would be at state S24, robot 3 should always be assigned to unload inspection center, rather than unloading milling machine. Similarly at state S25, there are two enabled transitions e.g., T13 and T3. And similar to previous case both milling center and inspection center are busy.Firing of T3 indicates robot 3 unloading part A from milling machine and start loading part A on to inspection center and firing of T13 indicates robot 3 unloading finished part from the inspection center, Suppose, if firing of T3 takes place, the robot 3 will unload part A from milling machine and will have to wait for inspection center to become free. But even if, inspection center completes its process it will be free only if the same robot 3 unloads the part. However robot 3 is already holding part which it unloaded from the milling machine and thus there is a deadlock situation. If we want at this stage to recover system then we have to make robot 3 free by human intervention. Thus deadlock prevention policy would be, at stage S25 robot 3 should always be assigned to unload inspection center. Table 5.4 shows the deadlock prevention policy for all possible deadlocks in the system. The first column in the table shows one possible sequence of transition firing for a particular deadlock state; however there can be many possible paths or sequence of transition firing which will lead to the same deadlock state.

Table 2 shows the deadlock prevention policy for all possible deadlocks in the system. The first column in the table shows one possible sequence of transition ring for a particular deadlock state; however there can be many possible paths or sequence of transition ring which will lead to the same deadlock state. A state in a reachability graph is represented by a vector having elements equal to number of places and the value of each element corresponds to the number of tokens held by respective places at that instant.

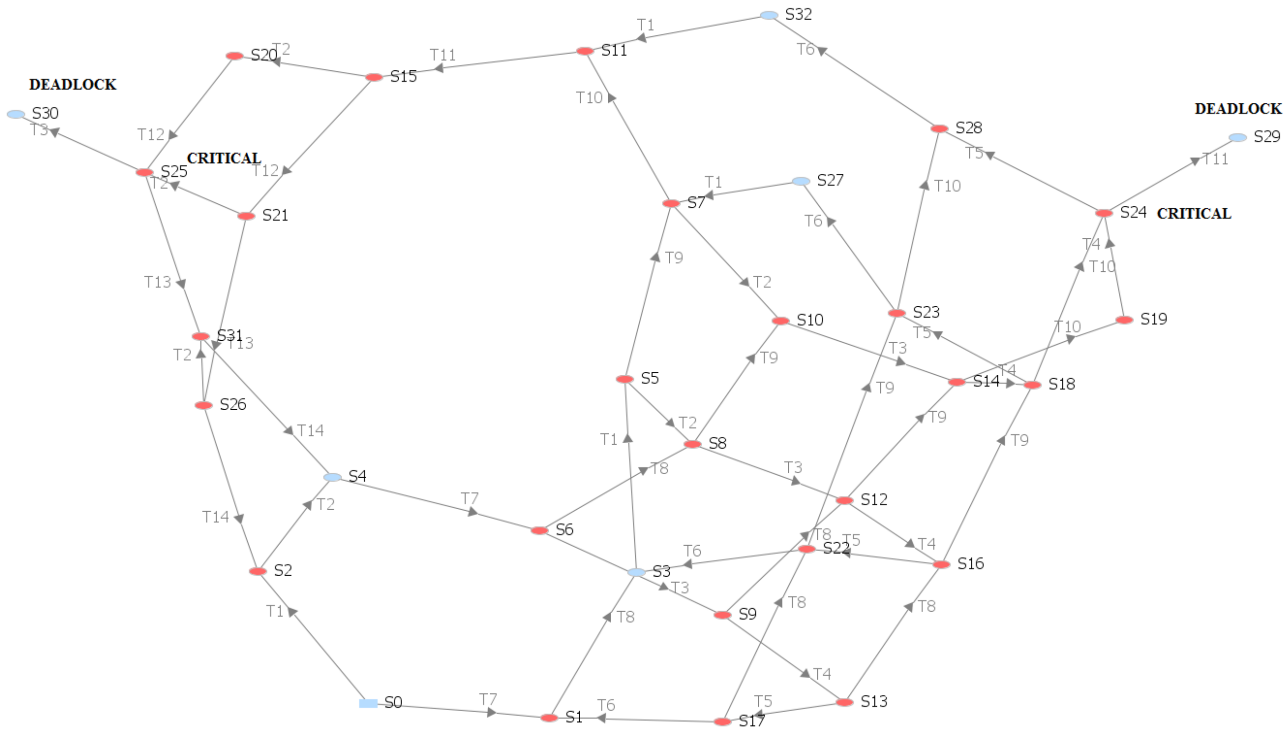


Figure 4: Reachability graph of Petri net model of FMS

Table 2: Deadlock Prevention Policy for FMS

Sequence of transition firing	Deadlock state /Critical state	Deadlock Prevention Policy	Physical Interpretation
T7,T8,T1,T9,T2,T3,T10,T4,T11	S29/S24	At S24, do not fire T11 instead fire T5	When system is at state S24 there may be a situation to choose between 1. Robot 3 assigned to unload part A from inspection center and 2. Robot 3 assigned to unload part B from Milling machine. Deadlock prevention policy is, at this state in order to prevent deadlock always choose option 1.
T7,T8,T1,T9, T2,T3,T10,T4, T5,T6,T1,T11, T2,T12,T3	S30/S25	At S25, do not fire T3 instead fire T13	When system is at state S25 there may be a situation to choose between 1. Robot 3 assigned to unload part B from inspection center and 2. Robot 3 assigned to unload part A from Milling machine. Deadlock prevention policy is, at this state in order to prevent deadlock always choose option 1.

International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

Deadlock Avoidance policy

In deadlock avoidance, the controller will be continuously collecting the status of each elements or happenings in the system and accordingly keepdirecting appropriately to the operator in order to avoid deadlock just before its occurrence.

In the present work the deadlock avoidance is achieved by deadlock avoidance algorithm. This algorithm has been implemented using MATLAB (Matrix Laboratory) with help of GUI (Graphical User Interface). For GUI we fetch data from an excel sheet which consists of enabled and fired transitions, lead state, look ahead message, machines etc.

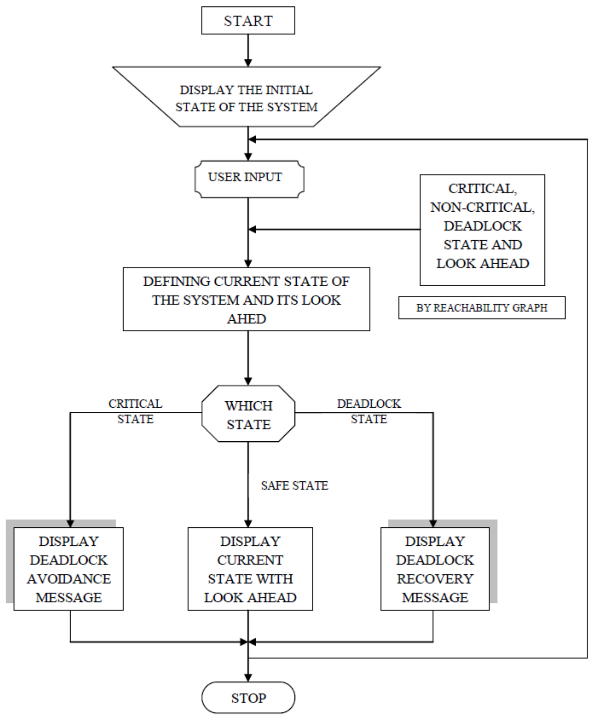


Figure 5: The flow chart of deadlock avoidance algorithm

The Figure 6 shows the screen shot of controller at initial state of the system. One canobserve that, the status of all the machines /inspection center and the robot is displayedas 'Free' and transition T1, T7 are displayed as the enabled transitions. The transitions firing sequence to obtain part A is displayed as the transition T1 to T6 and for part Bas T7 to T14, which confirms with the Petri net model shown in Figure 3. The screenalso shows the fired transitions at any given time/state by this one can get to know thestatus of processing of part A or part B. However in the initial state there will be no fired transitions. Look ahead of this state is displayed as 'SAFE' which means nextimmediate state is neither a critical or deadlock state.The Figure 7 shows the resulting screen shots of controller when transitions T1 andT2 are fired. At S0 when transition T1 is fired it reaches state S2 one can observe thisin figure 7, the fired transition now is displayed as T1, indicating thatrobot has started loading raw part Aon to the

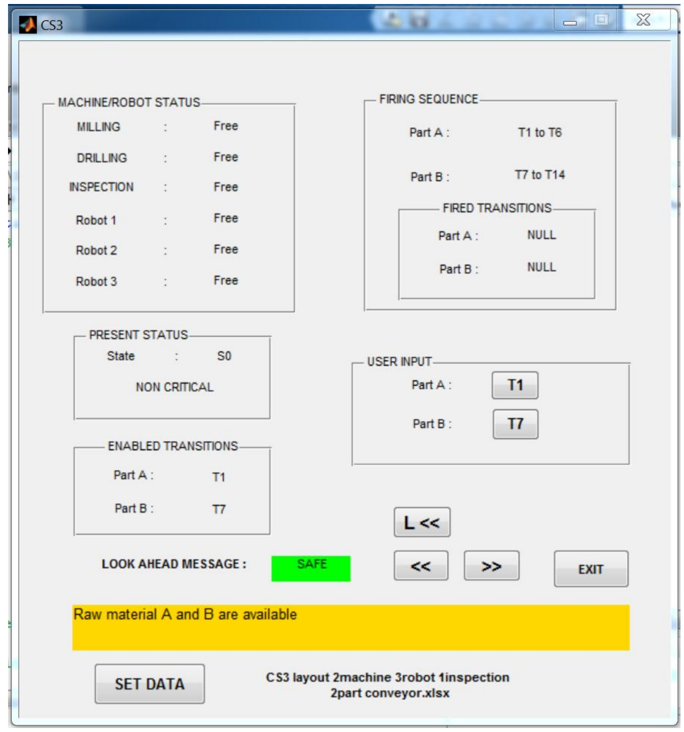


Figure 6: GUI screen at the initial state S0

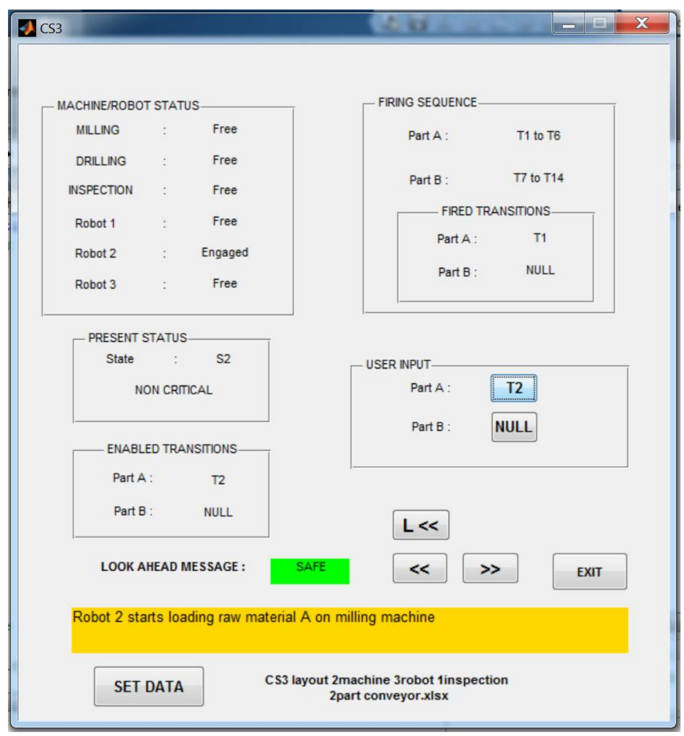


Figure 7: Screen shot of controller at the state S2 when transition T1 fired

Milling machine. Thus the status of robot is displayed as 'Engaged'. Similarly at S2 when transition T2 is fired it reaches state S4, one can observe this in Figure 8, the fired transition now is displayed as T2 for part A whereas it is NULL for part B because it has not started yet. This indicates that robot has completed loading raw part A on to the milling machine. Thus the status of robot is displayed as 'Free' whereas the status of milling machine is displayed as 'Engaged'.

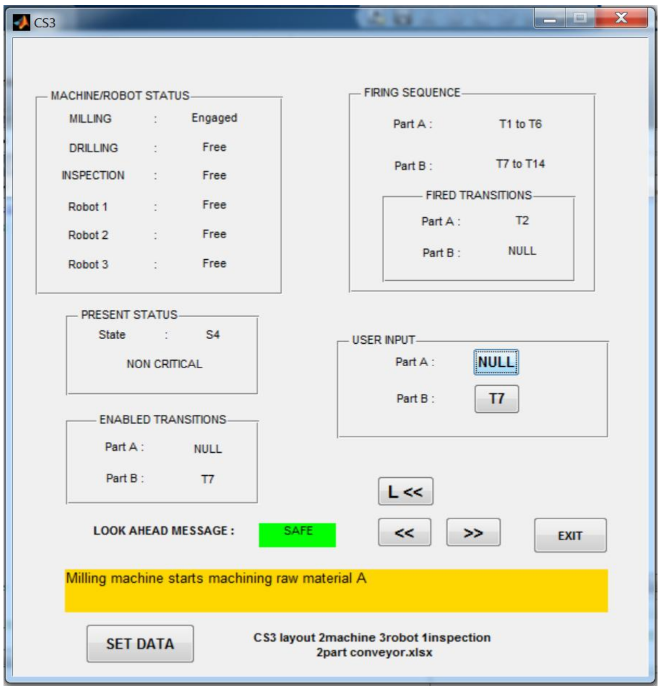


Figure 8: Screen shot of controller at the state S4 when transition T2 fired

When system is at state S24, which is a critical state there is a chance for systemto enter in to deadlock state S29. According to deadlock prevention policy established in the Table 2 robot 3 should be always assigned to unload part A from inspection center, However if Milling machine completes machining before inspection center, thensince robot 3 is free, deadlock is bound to happen. Thus there is need to avoid deadlockbecause we are not able to prevent it. The Figure 9 shows the screen shot of thecontroller when system is at critical state S24. One can observe that look ahead is displayed as 'DEADLOCK' and message displayed directs the operator to unload partfrom milling machine to buffer, in case it completes machining thus avoiding the deadlock.In case if deadlock is not avoided it results deadlock state S29, the Figure 10 shows thecorresponding screen shot in which the controller is directing the operator to unload theinspection center manually in order to resolve deadlock. The Figure 11 shows the screenshots of remaining deadlock state S30 and its critical state S25. Thus it can be concludedthat the developed deadlock avoidance algorithm is capable of avoiding deadlock in thesystem.

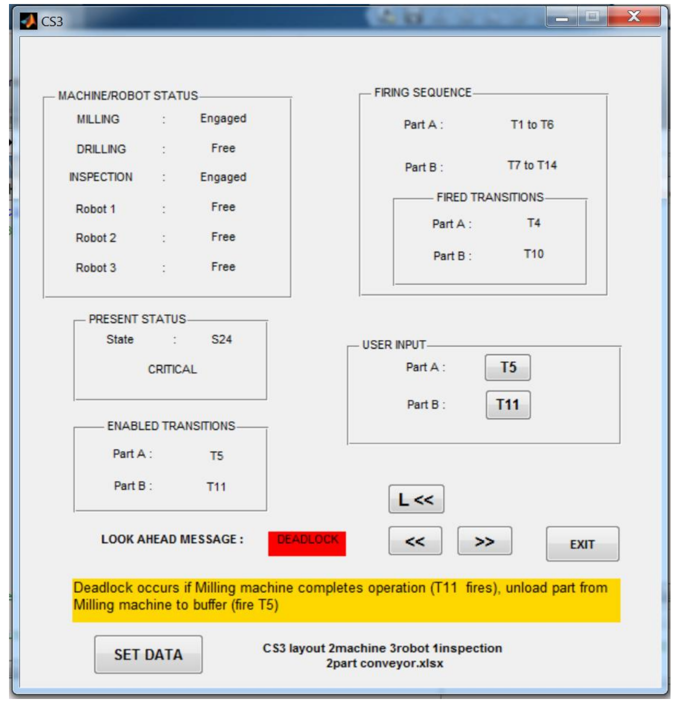


Figure 9: Screen shot of controller at critical state S24

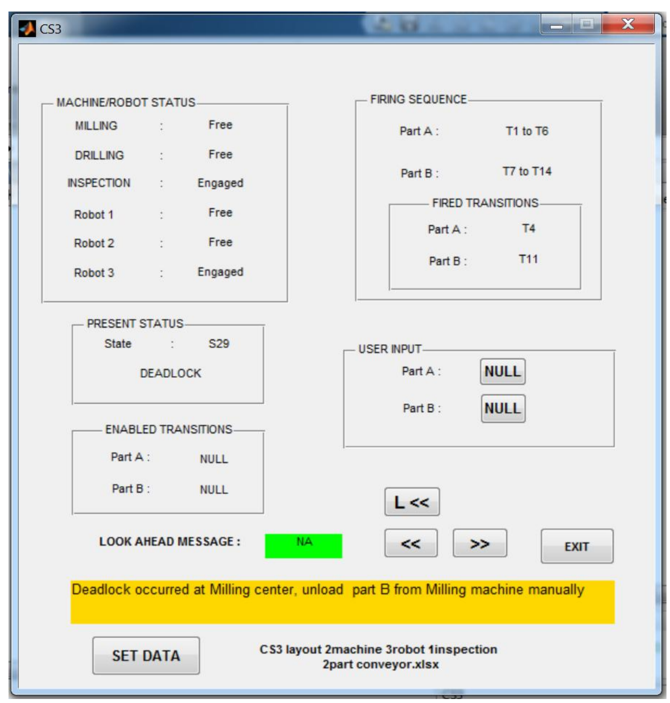


Figure 10: Screen shot of controller at deadlock state S29

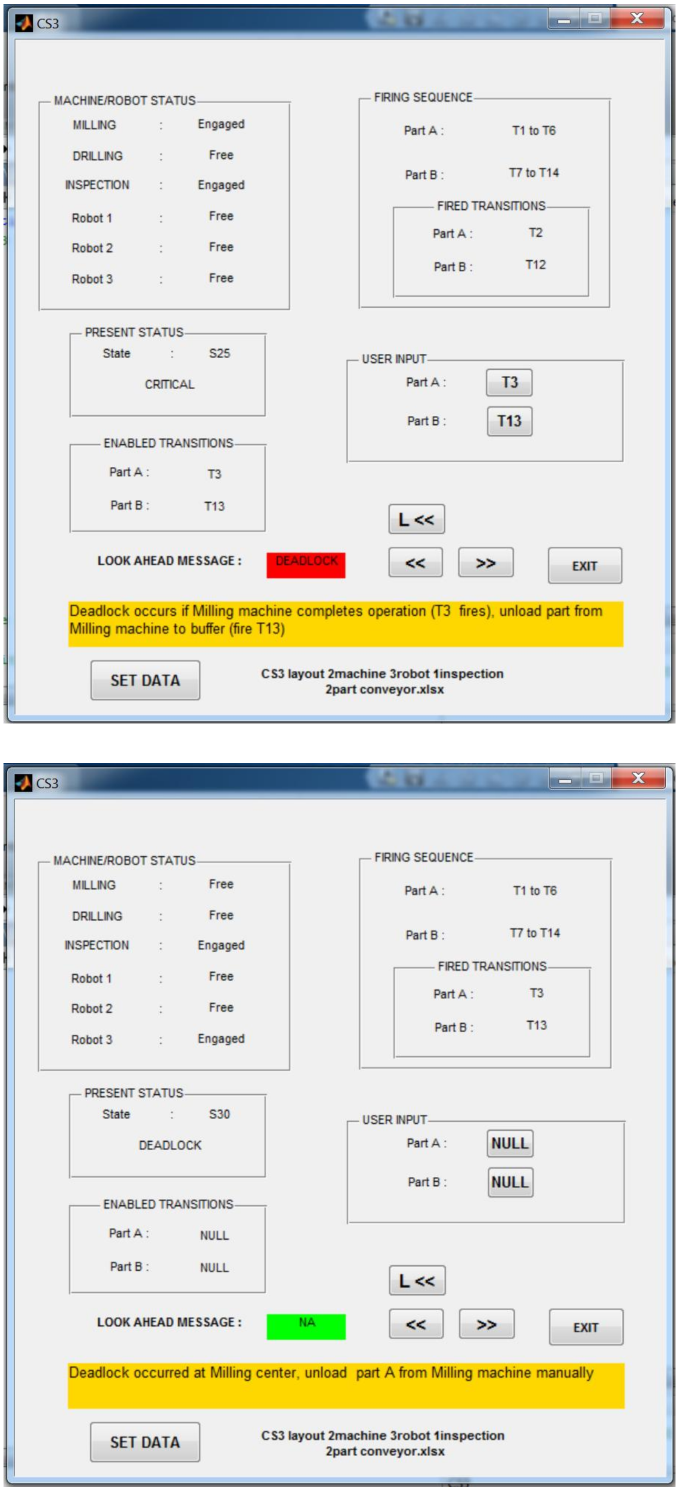


Figure 11: Screen shots of deadlock state S30 and its critical state S25 in a case study

Conclusion

The paper presented an overview of the deadlock problem in FMS and Petri net as a modeling tool. A case study of FMS having three robots, two machines and an inspection center processing two parts was analyzed for deadlock using reachability graph of Petri net model. Then deadlock prevention policy was established which gives deadlock free resource allocation method. This work illustrates that the reachability graph technique can be effectively used for analyzing and establishing a deadlock prevention policy for an FMS.

Finally for a deadlock avoidance algorithm has been generated and simulated over the screen with help of a GUI tool in MATLAB, the display of message can be converted into control signal which can bring about control action using sensors and actuators.It is possible to avoid a deadlock state in the FMS which can lead to improve productivity, reduce setup time and process time and saving’s in energy. It is easy to execute and establish a deadlock avoidance policy in MATLAB as compare to ‘c’ language.

References

[1] E. G. Coffman, M. J. Elphick and A. Shoshani, “System Deadlocks”, Computing Surveys, Vol. 3, No. 2, 1971, pp. 67-78.

[2] Y. Narahari and N. Viswanadham, “A petri net approach to the modelling and analysis of flexible manufacturing systems”, Analysis of Operations Research, Vol.3, 1985, pp. 449-472.

[3] Y. Narahari and N. Viswanadham, “Coloured petri net models for automated manufacturing systems”, IEEE, IIS, Bangalore, CH2413-3/87/0000, 1987, pp. 1985-1990.

[4] Tadao Murata, “Petri Nets: Properties, Analysis and Applications”, IEEE, Vol.77/4, 1989, pp. 541-580.

[5] Y. Narahari and N. Viswanadham et al., “Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models”, IEEE transactions on robotics and automation, Vol.6, 1990, pp. 713-723.

[6] Zurawski Richard, Zhou MengChu., “Petri Nets and Industrial Applications: A Tutorial”, IEEE. Transactions on industrial electronics, VOL. 41, NO. 6, December 1994, pp. 567-583.

[7] Jie Wu, Hanqi Zhuang, “A petri-net-based collision and deadlock Avoidance scheme for FMS”, IEEE 0-7803-2535-4, 1995, pp. 511-520.

[8] Joaquin Ezpeleta, JosC Manuel Colom, and Javier Martinez, “A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems”, IEEE, Vol.11, NO. 2, 1995, pp. 173-184

[9] Mark A. Lawley, Spyros A. Reveliotis, and Placid M. Ferreira, “A Correct and Scalable Deadlock Avoidance Policy for Flexible Manufacturing Systems”, IEEE transactions on robotics and automation, Vol.14, 1998, pp. 796-809.

[10] Sridhar Mohan, Ali Yalcin, Suresh Khator, “Controller design and performance evaluation for deadlock avoidance in automated flexible manufacturing cells”, Elsevier Ltd. 0736-5845, 2004, pp. 1-11.

[11] Sridhar Mohan, "Deadlock avoidance in mixed capacity flexible manufacturing systems", Graduate Theses and Dissertations 1164, 2004, pp. 1-69.

[12] Alexandre da Silva Ribeiro, Eduard Montgomery Meira Costa and Eduardo José Lima, “Flexible manufacturing systems modelling using high level petri nets”, ABCM Symposium Series in Mechatronics, Vol.3, 2008, pp. 405-413.

[13] Tauseef Aized , “Future Manufacturing Systems”, University Campus, Croatia. ISBN 978-953-307-128-2, 2010, pp. 1-16.

[14] Farooq Ahmad, Hejiao Huang and Xiao-long Wang, “Analysis of Parallel Manufacturing Processes with Resource Sharing”, International Journal of Computer Theory and Engineering, Vol.2/2, 2010, pp. 250-257.

[15] Wensong Hu, Yuyuan Zhu and Jie Lei, “The Detection and Prevention of Deadlock in Petri Nets”, ELSVIER, 2011, pp. 656 – 659.

[16] Zhiwu Li, Naiqi Wu and Mengchu Zhou, “Deadlock Control of Automated Manufacturing Systems Based on Petri Nets-A Literature Review”, IEEE Transactions on Systems Man and Cybernetics Part C 2011.2160626, 2012, pp. 1-50.

[17] Gunjan yadav and Dr A. K. Bajpai, “Overview of Modelling Of Production (FMS/CIM) System Using Petri Nets”, IJERT, Vol.1, Issue 8, 2012, pp. 1-4.

[18] Abdulziz M. El-Tamimi , Mustufa H. Abidi ,S. Hammad Mian and Javed Aalam “Analysis of performance measures of flexible manufacturing system”, Journal of King Saud University, Vol.24, 2012, pp. 115-129.

[19] Murali Ambekar and S. G. Bhatwadekar, “Petri net based modeling of FMS and its deadlock Prevention”, International Journal of Mechanical and Production Engineering Vol.2/1, 2013, pp. 65-71.

[20] Mowafak Hassan Abdul-Hussin, “Petri Nets approach to simulate and control of Flexible Manufacturing Systems”, International Journal on Software Engineering, Vol.1, 2014, pp. 1-10.

[21] Husam Kaid, Abdulaziz M. El-Tamimi, Emad Abouel Nasr, and Abdulrahman Al-Ahmari, “Applications of Petri nets Based Models in-Manufacturing Systems: A Review”, International Conference on Operations Excellence and Service Engineering Orlando (IEOM Society), 2015, pp. 516-528.

[22] Ranbir Singh, Rajender Singh and B. K. Khan, “A Critical Review of Machine Loading Problem in Flexible Manufacturing System”, World Journal of Engineering and Technology, 2015, Vol.3, pp. 271-290.

[23] Quang-Vinh Danga and Lam Nguyena, “A Heuristic Approach to Schedule Mobile Robots in Flexible Manufacturing Environments”, ELSVIER, 13th Global Conference on Sustainable Manufacturing, 2016, pp. 390-395.

[24] Y. Narahari and N. Viswanadham, “Performance modeling of automated manufacturing Systems”, Prentice Hall of India, New Delhi. ISBN 81-203-0870-0.

[25] Sreekaanth S. Isloor and T. Anthony Marsland, “The Deadlock Problem: An Overview”, Article, University of Alberta. pp. 1-30.

[26] Platform Independent Petri Net editor version 4.3 (PIPE v4.3) software is a freeware which can be downloaded at http://space.dl.sourceforge.net/project/pipe2/PIPEv4/PIPEv4.3/PIPEv4.3.zip <accessed on 08 Aug 2016>

[27] MATLAB (Matrix Laboratory- Version R2014a) software is obtained from college lab <accessed on 10 Nov 2016>