Journal of Rajasthan Academy of Physical Sciences
ISSN: 0972-6306; URL: http://raops.org.in
International Conference on Mathematical and Statistical Computation (ICMSC-2022)
Swami Keshvanand Institute of Technology (SKIT), Jaipur, Rajasthan, (India) 3rd-5th March 2022
October, 2022, 25-36

PETRI NET MODELING AND ANALYSIS OF AUTOMATED GUIDED VEHICLE SYSTEM

Priya[1] and **Sunita Kumawat**[2]
Department of Mathematics, Amity School of Applied Sciences,
Amity University Haryana, Gurugram, India.
E-mail: [1]priyayadav17894@gmail.com,
[2]skumavat@ggn.amity.edu (Corresponding author)

Abstract: The use of theory of Petri net is to detect deadlock, prevention and avoidance for Automated Guided Vehicle for System of Simple Sequential Processes (S^3PR) is explained. The live and deadlock free system of S^3PR is observed in this context. This is the efficient method to minimize the siphons in the system. When the liveness of the net is not shown then the proposed algorithm is used to make it deadlock free. The performance of the proposed algorithm is explained by several examples. Petri net is a graphical and mathematical tool which is used to model and simulate the discrete event system and AGV follows the rules of discrete event system. In this paper, we take AGV problem and model it through Petri net. Petri nets show several applications to check the behavior of the system. Due to these applications, deadlock can be easily detected and then treat their fault by re-model the system, so that the system becomes deadlock free. In this paper, we propose an algorithm to detect and treat the fault of AGV. This paper helps to overcome the deadlock occurs in AGV to produce a particular product. AGV follows deadlock free path i.e., liveness. So, to overcome the deadlock situation Petri net is used to model a particular system, so that we can get deadlock free path and system.

Keyword: Automated Guided Vehicle, Deadlock, Fault Detection, Fault Treatment, Petri net, S^3PR.

1. Introduction

Automated Guided Vehicle (AGV) is a wide area used often in industries to manufacture several types of products. AGVs are mainly robots, machines, buffer stocks and many other tools. AGV follows the pre-defined commands. Resources are shared for processing the system. The sharing of these resources sometime causes the deadlock situation. So, the system could not be live and stop working and AGV fails to complete their tasks. AGV's starting and ending stations are same. When AGV complete its task, it will store on its initial place or at idle state. Different parameters or different elements used in control system, while implementing automation to complete a particular task. In automated guided vehicle system, generally the basic properties [26], [19], are analyzed: liveness [4], [14], deadlock detection [11], deadlock avoidance, deadlock prevention [9], deadlock control [15], [12],[25], [29], deadlock recovery, reach-ability graph [27], minimal siphons [9], [7], siphons and traps [24]. These properties are analyzed in Petri net to get a fault free system. When deadlock occur, deadlock prevention policy [9] are applied, so that deadlock will not be occur. Petri net is basically a graph theory approach [8], [13], [18] which is used in discrete event system. The applicative area for flexible manufacturing system, neural network system [21], [11], automated manufacturing system [29], [7], wireless sensor network [2], biological network [2], [17] etc. Fuzzy Petri net [1], [23] is also a wide area for optimization of the parameters used in the system.

In order to prevent the deadlock in the AGV, an effective and efficient algorithm for deadlock control is implemented [29]. The computational complexity of the deadlock-control technique or approach has become lower than it may be implemented in the large-scale system. On the other hand, behavioral permissiveness requires high re- source utilization in the controlled or managed Petri net [22]. In addition to this, the deadlock control techniques developed on the basis of structural analysis approach (siphons) for the system s of automation with the use of Petri net framework. It may be implemented by inserting or adding the control places and arcs linked to original nets so the siphons are non-empty permanently. Different milling machines, turning machines, input and output buffers as well as assembly and inspection sectors are situated in it. Each machine owns a particular task at a particular time. The parts have similar routes and assembled within the assembling sectors. The assembled part is then inspected in the inspection sector. After the completion of the inspection, the final product is released. The working procedures of the two machines remain mutually exclusion. The first machine is used for loading or unloading the second part in the second route of the conveyor (AGV). On the other hand, there remains a third machine that works for conducting the assembly and inspections.

In this paper, automated guided vehicle [5] processing is explained using Petri net for modeling [5], [20] and simulation [6]. In AGV, the resources [22], [10] are shared through machines, robots and workstations. Deadlock occurs due to sharing of re sources.

The summary of the paper is as follows: section 2 explains the preliminaries and basic properties of Petri net in S^3PR net. Deadlock detection and deadlock recovery in deadlock resolution are explained in section 3. Deadlock recovery policy (algorithm) and an example is provides in section 4. Section 5 contains the conclusion of the paper.

2. Preliminaries

This section covers the basic properties of Petri nets, the simulation, and analysis of reach-ability, invariants, recovery transition and reach-ability graph.

Petri net is a directed, bipartite, weighted graph which is firstly introduced by Sir Carl Adam Petri in his doctorate work in 1962 [19]. It is a graphical and mathematical method which is used to simulate and check the behavior of a particular system. Petri nets have wide applicative areas such as distributive system, data-flow computing system, biological network, neural networks, manufacturing systems, etc. It is used for all the stages of system development such that modeling, analysis, simulation. It contains Places and Transitions. Places are represented by circle, denoted by P and transitions are represented by rectangular box, denoted by T. Places and transitions are connected with each other by arcs. No place to place or transition to transition path exists. Only place to transition and transition to place arcs exist. Tokens are denoted by black dots which are holds in the places.

Definition 2.1 *Petri net is a sequential process with resource sharing. It is a bipartite, weighted and directional graph which consists mainly 5-tuples i.e., (P, T, W, F, M_o), where $P = (p_1, p_2, p_3, \ldots\ldots p_n)$ are number of Places,*

$T = (t_1, t_2, t_3, \ldots\ldots, t_m)$ are number of Transitions,

W: $F \rightarrow (1,2,3,\ldots)$, W is the weight of corresponding area between Place and Transition or between Transition and Place,

M_o is the initial marking,

F is the flow of the system i.e., $F \rightarrow (P * T) \cup (T * P)$ and $F = (P * T) \rightarrow N$, where N is the set of natural numbers.

The union of P and T never be null and intersection will always null, i.e., $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$

A system of simple sequential processes with resources (S^3PR) Petri net is a 5-tuple net i.e., PN = (P_c, T, F, W, M_0)

where $P_c = (P_0 \cup P_A \cup P_R)$. P_0 be the idle places, P_A is the allocation places and P_R be the resource places.

2.1 Properties and Simulation

For simulation, the following properties should be followed to make the net deadlock free/live, through this the system works continuously.

Definition 2.2 *Idle state/place: This is the state at which the system is at initial state or it can also known as waiting state. It is denoted by P_o i.e., idle place and T_o i.e., idle Transition.*

Definition 2.3 *Resource place: This is the place where all the resource are settled which are used to share for the proper processing. It is denoted by P_R.*

Definition 2.4 *Let N=($P_o \cup P_A$ P_R, T, F, W) which is ordinary Petri net at initial state.*

Definition 2.5 *Preset of nodes: Let N = (P_B, T, W, F, M_0) be an S^3PR Petri net where $sP_B = (P_0 \cup P_A \cup P_R)$ and $(a, b)\ \varepsilon\ (P_c \cup T)$. Then $\bullet_a = b\ \varepsilon\ (P_c \cup T) | (b, a)\ \varepsilon\ F$.*

Definition 2.6 *Post set of nodes: Let N = (P_B, T, W, F, M_0) be an S^3PR Petri net where $P_B = (P_0 \cup P_A \cup P_R)$ and $(a, b) \varepsilon P_c \cup T$. Then $a^{\circ} = b\ \varepsilon\ (P_c \cup T) | (a, b) \varepsilon\ F$ of the net N and $M_0(r) \geq 1$, $M_0(P)=0\ \forall P_0, P_r$, P_A are the idle places.*

Definition 2.7 *Initial marking: Let N be a S^3PR Petri Net and M_o is the initial marks state respectively.*

Definition 2.8 *Liveness and Deadlock: In Petri-net, when all the transition fire and token moves from one place to another place then the system is live otherwise not live. In a system, if any one of the transition does not fix then it occurs deadlock and make the system non-live. There are some properties of liveness which are as follow:*

L_o - *Non-live i.e., if any transition never be fire in any sequence.*

L_1 - *when any transition t fire at-least one time.*

L_2 - *when any transition t fire at-least k times where k is a positive integer.*

L_3 - *when any transition t fire infinitely.*

L_4 - *when ant transition t is L_1-Live for each marking.*

3. Deadlock Resolution by Recovery Transition

Petri net can be used as a recovery net to recover or undo all the losses of the tokens which are moved from the places to the transitions. At a certain point, the primary protection of the tokens fails and at that point the Petri net is used as a recovery to backup all the lost tokens. In addition to this, the token of a place is deposited or to moved to the other place by adding recovery transition/ transitions. The loss has been taken place as the primary relay is unable to respond the activating current relay coil.

3.1 Deadlock Detection

We know that whenever a deadlock will occurs the progress of the system will get suspended, so the progress of the system can get also suspended by the deadlock. So, before going to resolve the situation, why these particular suspensions has taken place. Let us go for the deadlock detection and then if you can ensure yourself, the progress of the system has suspended only due to the occurrence of deadlock, then we can go for the deadlock recovery now time to execute the system out from deadlock situation.

Resource allocation graph: It is having 4 components: Process, Resources, Some edges will be directed from the process to the resource is known as the request edge. Some edges will be directed from the resources to the process and is known as the allocation edge. These are 4 components. We know that, in case of resource allocation graph, if there is no cycle (close path) i.e., cycle free that means deadlock will not occur, but if the resources allocation graph is having a cycle, then deadlock may or may not occur. Resource allocation graph can predict but can't ensure but wait for graph can ensure that the deadlock can take place or not and if the deadlock take place it has been detected properly then deadlock recovery routines will are and that will retrieve the sys- tem from the deadlock situation.

3.2 Deadlock Recovery

When a detection algorithm determines as that from the deadlock, there are two approaches of breaking a deadlock:

1) Process Termination: Simply kill or terminate one or more processes in order to break the circular wait.
2) Resources Pre-emption: To eliminate deadlock, we can prompt some resources and give these resources to other processes until the deadlock cycle breaks. It involve three issues:
    a) Selecting a victim: We must determine which resources and which processes are to be prompted and in which order to minimize cost.
    b) Rollback: We must determine what should be done with the process from which resources are prompted. However it is more efficient to rollback the process only as far as necessary to break the deadlock. This method requires the system to keep more information about the state of all the running processes.
    c) Starvation: In a system, it may happen that resources are always prompted from some process. As a result, this process never be complete its task. This situation is called starvation and needs to be avoided. When the deadlock is detected, then it is being necessary to remove it because it makes fault to processing a manufacturing system.

4. Deadlock recovery policies and illustrative example

To remove deadlock of AGV in manufacturing system, the following algorithm is followed to make the system live.

4.1 Deadlock Recovery Algorithm

This section explains the deadlock recovery algorithm to make the system live by design recovery transitions according to the dead places. An example is explained to illustrate the proposed algorithm. The following terms are used in the algorithm to solve a dead situation.

Let PN be the initial S^3PR net. M_0 be the initial marking. P_0 is the idle place. P_A

are the allocation places. P_R be the resources places in S^3PR net. M_L be the legal marking. M_D be the dead marking. T_r be the recovery transition. T_D be the dead transition. P_i be the dead place. T_{Di} be the dead transition according to P_i. PN_r be the recovered S^3PR net.

$\zeta_t rj$ =($M_d\, s\, M_D$ i.e., M_D is recovered by t_{rj}.

In proposed algorithm, a recovery transition is added to the dead place/places as required to make the net deadlock free in each iteration. This process continues until all of the dead marking becomes legal marking and all the recovered path should follow minimal siphons (*i.e.,* all the places should be active throughout the process.)

Input: A Petri net model *(PN ,M₀)* of AGV in manufacturing system with

$PN = (P,T,F,W)$ where $P= (P_0 \cup P_A \cup P_R)$.

Output: A live Petri net AGV system (PN_r,M_0). **Step 1:** Compute a S^3PR Petri net model (N,M_0). **Step 2:** Simulate model using Snoopy tool.

Step 3: Compute legal marking M_L and dead marking M_D of (N,M_0).

Step 4: Initially T_r=0 /* T_r= Recovery transition to be computed*/.

Step 5: Compute T_D /* Dead transition*/.

Step 6: While M_D =0.

Find P_i's and t_{Di}'s where deadlock occur.

Compute recovery transition t_{rj}.

$T_r=T_r \cup (t_{rj})$ and $M_D = M_D - \zeta_{t}rj$.

Add recovery arcs A_r from dead place to recovery transitions and from recovery transitions to corresponding places, so that it make minimal siphon system. End while.

Step 7: Add all recovery transitions T_{ri} in (PN ,M_0) and denote the resulting net as (PN_r, M_0).

Step 8: Output (PN_r, M_0).

Step 9: End.

Theorem 4.1 *Recovery transition/transitions should follow minimal siphon property.*

Proof: According to proposed algorithm, after step 6, there should be recovered paths through the dead points to recovery transition/transitions and from recovery transition to the next proceeding places. In figure 1, the system become live by adding t_{r2}, but it contains a number of siphons. To minimize the siphons, add another recovery path to make the system live as well as live. So that the system contain minimal siphon live net.

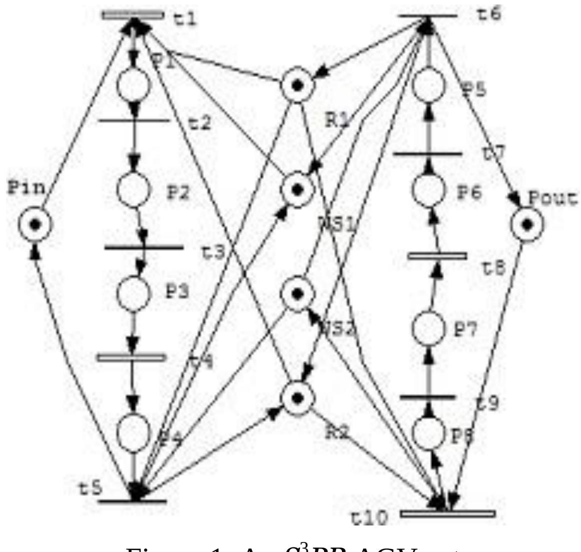


Figure 1: An S^3PR AGV net

An example is used to explain the proposed algorithm. Figure 1 shows a S^3PR Petri net model of AGV system which contains two idle places(P_0), eight allocation places (P_A) and four resource places (P_R). The partition of the places are: $P_0 = (P_{in}, P_{out})$, $P_A = (P_1\text{-}P_8)$, $P_R = (R_1, R_2, WS_1, WS_2)$. The legal marking of the net is $M_L = (P_1+\mathrm{P}_2+P_3+P_4, P_5+P_6+P_7+P_8)$ and the dead marking $M_D = (P_{in}, R_1, WS_1, R_2)$ through transition t_5 as shown in figure 1.

According to algorithm, a recovery transition t_{r1} is added to the dead place and arcs from recovery transition to the forward places are joined to make the system live. Red color arc and transition shows the first iteration step of algorithm *i.e.*, t_{r1} is first recovery transition corresponding to the path ($P_4 \rightarrow t_{r1}$) and ($t_{r1} \rightarrow R_2$) So, the recovered paths are ($P_4 \rightarrow t_r$) and ($t_r \rightarrow R_2$). But after 25th steps, the deadlocks arise at t_1 and t_{10}.

According to algorithm, initially $T_r = \varphi$ and $M_D = M_D - \zeta_{trj}$.

After applying algorithm, $T_r = (\varphi\ t_{r1}) \cup = t_{r1}$

and $M_D = M_D - \zeta_{tr}\ 1$.

Now, the deadlock removed from t_5, but stored at t_1 and t_{10}. To remove these deadlock from these transitions, again apply the algorithm (as according to iteration rule). Add another recovery transition t_{r2} in the net.

Now, $T_r = (T_r \cup t_{rj})$

$T_r = (t_{r1} \cup t_{r2})$.

So, number of recovered transitions are 2 *i.e.*, $n(T_r) = 2$. Also check M_D after simulation from figure 2.

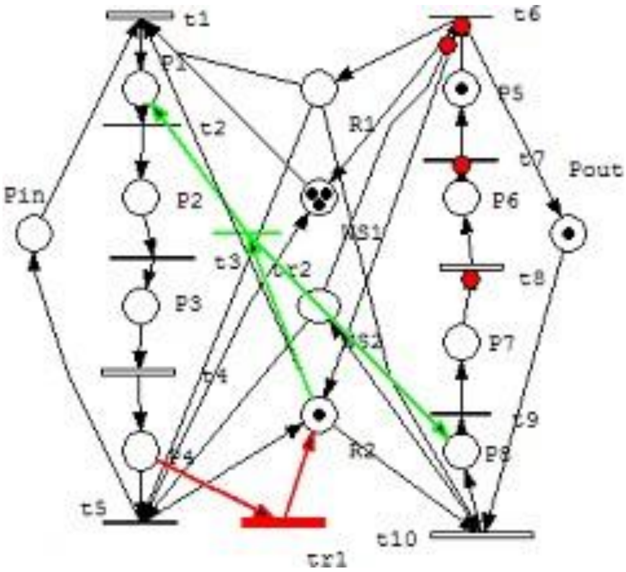


Figure 2: Live net but attain maximal no. of siphons

So choose some legal recovery path through which the AGV system contains minimal siphons. From figure 3, the recovery path is as follows :

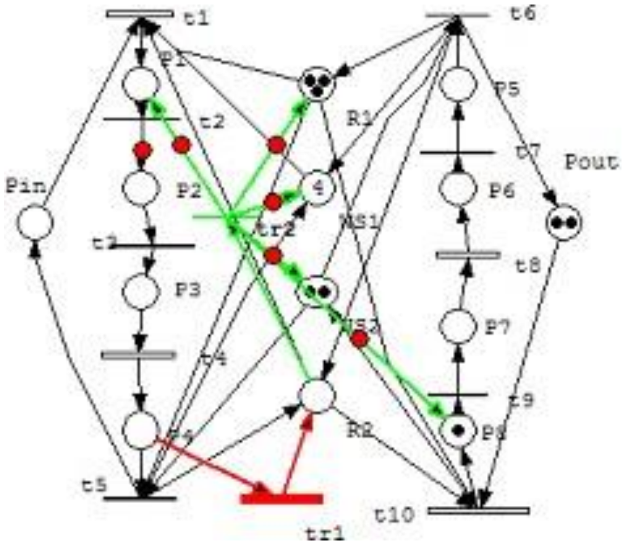
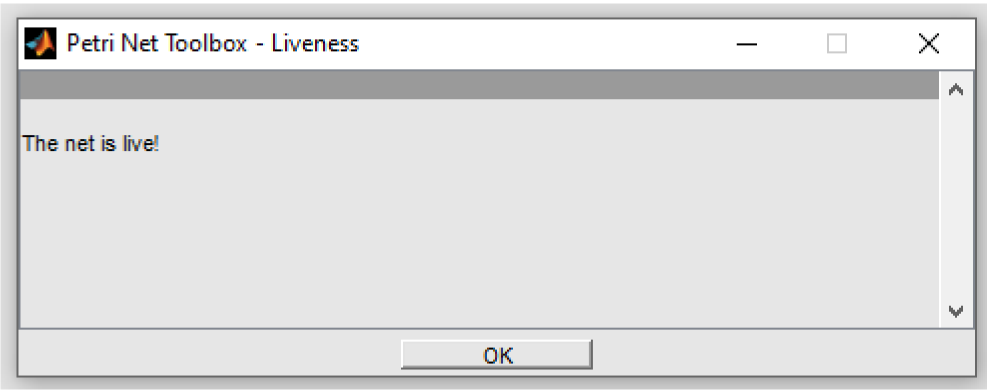


Figure 3: Live net with minimal siphons

($t_{r2} \rightarrow P_1$), ($t_{r2} \rightarrow WS_1$), ($t_{r2} \rightarrow R_1$), ($t_{r2} \rightarrow P_8$), ($t_{r2} \rightarrow WS_2$) and ($R_2 \rightarrow t_{r2}$).

Now, all the places and transitions work and the tokens starts storing on R_1, WS_1, WS_2, P_{in} and P_{out}. So, we get a deadlock free live net with minimal siphon. The liveness of the above live net with minimal siphons is as follows :



5. Conclusion

The Proposed Algorithm (PA) deals with the deadlock and with their recovery using Petri net in Snoopy tool, which is used in modeling and simulation. A recovered model is design by adding recovery transition. The main aim of the algorithm is to find a live AGV system with minimal siphons. The conceptual example has two idle states, two robots, two workstations and eight processing machines. By resource sharing through

AGV in the system and add recovery transitions, remove deadlock and make the system live. Number of Recovery Transitions ($n(t_{ri})$), Number of Recovery Arcs (n (A_r)), No. of Monitors ($n(P_i)$) parameters are taken for comparison of the results. Comparison results of deadlock recovery methods are shown in Table 5 and in figure **4**.

Parameters	[6]	[14]	[19]	[25]	[26]	[31]	[38]	[PA]
$n(t_{ri})$	0	0	0	7	0	0	0	2
$n(A_r)$	37	12	40	69	42	23	32	8
$n(P_i)$	8	2	5	0	9	5	6	0

Table 1: Comparison table for PA and existing result.

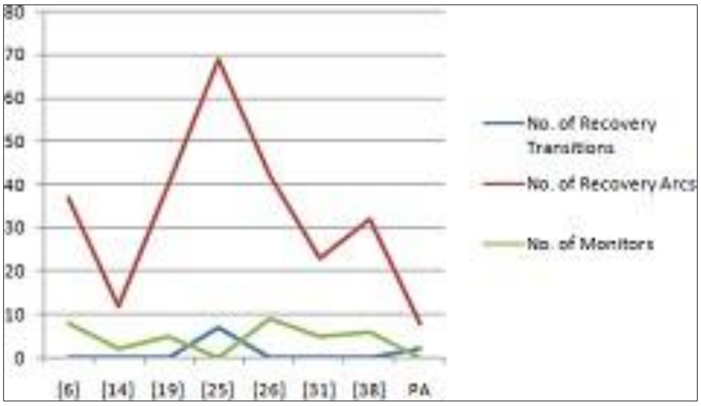


Figure 4: Comparison Graph for PA and existing result.

References

[1] Ahmari A., Kaid H., Li Z. and Davidrajuh R. (2020). Strict minimal siphon-based colored Petri net supervisor synthesis for automated manufacturing systems with unreliable resources. IEEE Access, **8**, 22411-22424.

[2] Aggarwal C. C. (2018). Neural networks and deep learning. Springer, **10**, 978-3.

[3] Bala T., Bhatia V., Kumawat S., Jaglan V. (2018). A Survey: issues and challenges in wireless sensor network. Int. J. Eng. Technol., **7**(2), 53-55.

[4] Chen Y. F., Li Z. W., Khalgui M., and Mosbahi O. (2011). Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems. IEEE Transactions on Automation Science and Engineering, **8**(2), 374-393.

[5] Chin-I. Liu, Loannou P. A. (2002). Petri net modeling and analysis of Automated Container Terminal using Automated Guided Vehicle Systems. Transportation Research Record Journal of the Transportation Research Board, **1782(1)**, 73-83.

[6] Erden Z, Aized T. (2018). Modeling and Simulation of AGVs using Petri Nets. Robotics and Automation Engineering Journal, **3**(5), ISSN. 2577-2899.

[7] Fatma, L. Ghabi J. and Dhouibi H. (2020). Applying interval fuzzy Petri net to failure analysis. International Journal of Service Science, Management, Engineering and Technology (IJSSMET), **11**(1), 14-30.

[8] Gupta S., Kumawat S., Singh G. P. (2019). Fuzzy Petri net representation of fuzzy production Propositions of a rule based system. Advances in Computing and Data Sciences, 197-210.

[9] Gupta S., Kumawat S, Singh G. P. (2021). Validation and analysis of Metabolic Pathways using Petri nets. Advances in Intelligent System and Computing, 1380.

[10] Gupta S., Singh Gajendra Pratap, Kumawat S. (2019). Petri net recommender system to model metabolic pathway of polyhydroxyalkanoates. International Journal of Knowledge and Systems Science, **10**(2), 42:59.

[11] Jiang W., Chen J. and Xu Y. (2018). A network celebrity identification and evaluation model based on hybrid trust elation. Tehnickivjesnik, **25**(4), 1136-1143.

[12] Kaid H., Ahmari A., Li Z. and Davidrajuh R. (2020). Single contoller-based Colored Petri nets for deadlock control in automated manufactruing systems. Processes, **8**(1), 21.

[13] Kaid H., Ahmari A., Nasr E.A., Shayea A., Kamrani A. K., Noman M.A. and Mohd. H.A. (2020) Petri net model based on neural network for deadlock control and fault detection and treatment in automated manufacturing systems. IEEE Access, **8,** 103219-103235.

[14] Kaid H., Ahmari A., Tamimi A. M., Nasr E.A., and Li. Z. (2019). Design and implementation of deadlock control for automated manufacturing systems. South Afr. J. Ind. Eng., **30**(1), 1-23.

[15] Kanayake E. Dewasurendra T., Abeyratne D., S. Ma. L. and Yarlagadda P. (2019)Model-based fault dignosis and prognsis of dynamic systems: A review. Proce-dia Manufacturing, **30**, 435-442.

[16] Kumawat S. (2011). A graph theoretic approach: Petri net. International Journal of Mathematical Sciences and Applications, **1**(3), 1637-1641.

[17] Kumawat S. (2012) A Graph Theoretic Approach: Petri Net. ISBN, **978-3-659-00254-0**.

[18] Kumawat S. (2012). Weighted directed graph: A Petri net-based method of extraction of closed weighted directed euler trail. Int. J. of Services, Economics and Management, **4**(3).

[19] Kumawat S. (2021) Chinese Postman Problem:A Petri Net based approach. Computational Methods and Data Engineering, **DOI. 10.1007/978-981-15-7907-3 16**, 203-222.

[20] Kumawat S, Purohit G. N. (2009). Travelling Salesman's Problem in Weighted directed graph: A Petri net Approach. Proceeding of APORS, 42-48.

[21] Kumawat S., Purohit G. N. (2011). Modeling of Access Control System using Petri nets. International Journal of Computer Science Engineering and Technology (IJCSET), 1(10), 605-616.

[22] Kumawat S., Purohit G. N. (2011). Modeling and analysis of Production System using operational Petri nets with resources sharing. International Journal of Networks and Mobile Technology, **2**(2), 61-80.

[23] Kumawat S, Purohit G. N. (2017). Total span of farm work flow using Petri net with resource sharing. Int. J. of Business Process Integration Management. **8(3)**, 160.

[24] Li Z. and Zaho M. (2018). On controllability of dependent siphons for deadlock prevention in generalized Petri nets. IEEE Trans. Syst. Man. Cybern. A, Syst. Humans, **38(2**), 369-384.

[25] Liu G., Zhabg L., Chang L. Ahmari A. and Wu N. (2020). Robust deadlock control for automated manufacturing systems based on elementary siphon theory. Inf. Sci., **510**, 165-182.

[26] Ma Z, Li Z and Gina A. (2017). Characterization of admissible marking sets in Petri nets with conflicts and synchronizations. IEEE T Automat Contr, **62**, 1329-1341.

[27] Ma Z, Tong Y, Li Z, Guia A. (2017). Basic marking representation of Petri net reachability spaces and its application to the reachability problem. IEEE T Automat Contr, **62**, 1078-1093.

[28] Muratha T. Petri nets: Properties, analysis and applications, IEEE, 77(4).

[29] Tamimi A. M., Nasr E.A., Ahmari A., Kaid H. and Li. Z. (2015). Evaluation of deadlock control designs in automated manufaacturing systems. in Proc. Int. Conf. Ind. Eng. Oper. Manage (IEOM), 1-10.