International Journal of Engineering Research and General Science Volume 4, Issue 4, July-August, 2016
ISSN 2091-2730

# Modeling and Simulation of a Manufacturing System by Using Time Petri Net

Chavan Vishwajit D[a*], Dr. Rajesh B. Buktar[b], Dr. Kiran Bhole[c]

Mechanical Engineering Department, Sardar Patel College of Engineering, Mumbai, India[a b c]

[*]Email: chavan.vishwajitd@gmail.com Tel.No. +91 8983706639

**Abstract**— The Petri Nets (PN) are being used for decision making in the manufacturing system using Petri Nets (PN). In this paper, PN has been applied successfully for modeling of discrete event dynamic systems such as manufacturing system that are characterized by conflicts, concurrency, and synchronization. An example of the manufacturing system of an XYZ industry consisting of four machines with operators has been taken into consideration and its PN model construction is explained. The model is then simulated in the MATLAB software using Petri Net toolbox. Furthermore, the modeled system is a Time Petri Net (TPN) model which is a well-known subclass of PNs for checking the coverability of the system. Analysis of the model using TPN depicts efficient performance evaluation and qualitative analysis.

**Keywords**— Petri Nets (PN), Manufacturing System, Time Petri Nets (TPN), Discrete Event Dynamic Systems (DEDS), Coverability, MATLAB, Petri Net Toolbox.

## INTRODUCTION

Automated manufacturing systems belong to the domain of discrete event dynamical systems (DEDS) in which the evolution of the system in time depends on the complex interactions of the timing of different discrete events, such as the entry of raw workpiece, the exit of finished parts, failure of a machine etc. The state of DEDS changes only at these discrete instants of time, instead of continuously. Several classes of models have emerged in this context and the models are broadly classified as qualitative or quantitative. Qualitative models capture logical aspects of systems evolution such as controllability, stability, the existence of deadlocks in system operations etc. This paper mainly concentrates on Quantitative Models which is more specifically named as "Performance Modeling." The paper highlights the quantitative system performance in terms of the throughput and the lead time. Quantitative models include discrete event simulation, min-max algebra, Markov chains, Stochastic Petri nets, queues and queuing networks. [1] [6] [8]

The concept of the Petri net has originated by Carl Adam Petri's in 1962. [1] [6] [8]

The area or system which can be described graphically like flow charts and which needs some means of representing parallel or concurrent activities, PNs can be applied. Petri Net is also used to solve complex problems coupled with other tools such as Genetic Algorithms, Linear Programing, C++ etc. To apply PNs, it is necessary to add some special modifications and restrictions which suit the particular application.

PNs provide a graphical notation for the formal description of the dynamic behavior of systems. PNs are particularly well suited to systems which exhibit synchronization, concurrency, mutual exclusion, and conflict. [1] [6] [8]

PN consists of four types of components places, transitions, tokens and arcs (Refer Fig.1). Places denoted as circles, are used to represent conditions or local system states, e.g. place may relate to one phase in the behavior of a particular component or Places represent possible states of the system. Transitions denoted as bars, are events or actions which cause the change of state or transitions are used to describe the events that occur in the system; these will usually result in a modification to the system state. Arcs specify the relationships between local states or conditions (places) and events (transitions). An arc from place to transition is termed as input arc and from the transition to place is termed as output arc. Tokens denoted as a dark dot, are identity-less markers that reside in places. The token in a place represents the condition or local state holds. Each transition has some number of input and output places. These input and output places represent the pre and post conditions of an event (transition).

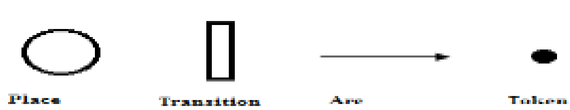


Fig.1. Components of Petri net

In section 2, transition enabling and the firing rule of PN is described along with an example of a simple manufacturing tool to show how transition enabling and firing rule executes. Section 3 gives an overview of the various properties of the PN of the modeled manufacturing system. In section 4, different analysis methods for the modeled PN are discussed by giving an example for the reachability analysis. Section 5 shows the actual creation of a model of a manufacturing system in PN toolbox in MATLAB environment following by doing a simulation of the modeled manufacturing system. Finally, section 6 concludes this paper.

### ENABLING TRANSITION AND FIRING RULE FOR THE MANUFACTURING SYSTEM

A transition with the absence of any input place is called a *source transition*, and one with the absence of any output place is called a *sink transition*. The sink transition consumes tokens but doesn't produce any. The source transition is frequently used in a manufacturing system model, so as to represent the entry of raw material or semi-finished parts in the system, while sink transitions are frequently used to represent the exit of finished or semi-finished parts. The source transition is unconditionally enabled, and on the firing of a sink transition tokens get consumed but does not produce any.

If p is both an input and output place of t, then the pair of place p and a transition t is called a self-loop. A Petri net is said to be pure if it has no self-loops. A Petri net with self-loops can be converted into a pure PN by adding a place and transition to each self-loop in the original Petri net. [3] [7]

The formal definition of a Petri net is, the PN is a 5-tuple, PN = $(P, T, I, O, M_0)$; where,

$P = \{P_1, P_2, \ldots, Pm\}$ is a finite set of places,

$T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions,

$I = (P \times T) \rightarrow N$ is an input function that defines directed arcs from places to transitions,

$O = (T \times P) \rightarrow N$ is an output function which defines directed arcs from transitions to places,

$M_0 = P \rightarrow (0, 1, 2, \ldots)$ is the initial marking.

Where, N is a set of non-negative integer i.e. $N = \{0, 1, 2, \ldots\}$.

To simulate the dynamic behavior of a system, a marking or state in PN is changed according to the following firing rule; [1]

a) A transition t is said to be enabled if each input place p of t is marked with at least one or more tokens or the weight of the directed arc from place p to transition t.
b) An enabled transition may or may not fire, depending on whether or not the event actually takes place.
c) On the firing of an enabled transition t, tokens get removed from the each input place p as the weight of the directed arc from place p to transition t and gets deposited equally to each output place.

Imposed by the transitions, tokens move between places according to the firing rules.

To fire a transition, each of the input places connected to it has at least one token; when transition fires, it removes a token from each of these input places and deposits a token in each of the output places connected. This is the firing rule. Sometimes a transition requires an input place which contains two or more tokens before it can fire. In this case, we denote the multiplicity of the arc by a small number written next to the arc instead of drawing a number of arcs between the place and the transition, similarly for output arcs. [1]

From an initial marking and followed by the firing rule, one can progress through the states of the model. Sometimes it is called playing the *token game*. On continuing we obtain all the possible states of the model, on recording all the states we see and stopping only when we can reach no states that we have not already seen. This is called the reachability set, which is the set of all possible markings that a net may exhibit, starting from the initial marking and following the firing rules. Various initial markings might lead to different reachability sets as we will see in the example below. That is why the initial marking is an important part of the model definition.

Example: For the simple Pent net shown in Fig. 2 the initial marking is (1, 0, 0) and the final marking is (0, 1, 1) on firing transition $t_1$. These are the only possible markings as it has only one transition to be fired. If the initial marking of the example is changed to (3,2,1) the set of reachable markings will be: {(3,2,1), (2,3,2), (1,4,3), (0,5,4)}. Because, the input places connected to transition $t_1$ has more than one tokens in it, so when the $t_1$ will fire only one token will remove from the input place $p_1$ as the weight of the arc is a unity. But still there are two tokens in the input place $p_1$, therefore the transition $t_1$ will remain enabled and it will fire again two times. Finally, all the tokens from input place removed and deposited to each output places and reached to a certain state of the system.

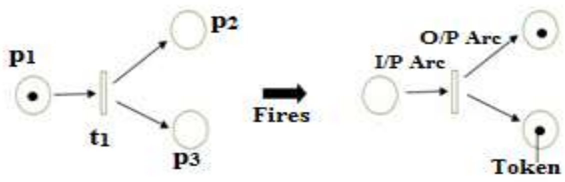


Fig. 2. Illustration Showing a Simple Petri Net Firing

#### EXAMINING PETRI NET PROPERTIES FOR MODELED MANUFACTURING SYSTEM

The Petri Net properties are the pre-requisite for validating the modeled system. On simulation, these properties get checked and if those gets satisfied then the modeled system meets the requirements.

Petri net is a mathematical tool which possesses a number of properties. These properties allow the system designer to identify under the design, the existence or absence of the application domain of specific functional properties of the system under. These properties can be distinguished as, behavioral and structural properties.

**A. Reachability**

In designing event-driven systems, an important issue is whether a system can reach a specific state, or exhibit a specific functional behavior. In general, the question arises is, whether the modeled system with Petri nets presents all prudent properties as specified in the requirement specification and no undesirable properties.

To find out the modeled system can reach a specific state as a result of a required functional behavior, it is important to find such a transition firing order which would transform a marking $M_0$ to $M_i$, where $M_i$ is the specific state, and the firing order is the required functional behavior. The existence of additional sequences of transition firings in the Petri net model which transform $M_0$ to $M_i$ shows that the Petri net model may not exactly reflect the structure and dynamics of the underlying system. A marking $M_i$ is reachable from a marking $M_0$ if there exists an order of transitions firings which changes a marking $M_0$ to $M_i$. [4]

**B. Safeness**

In a Petri net, places are often used to depict product and tool storage areas in manufacturing systems. It is important to determine the proposed control strategies which prevent from the overflows of these storage areas.

The property, boundedness is the concept which helps to identify the existence of overflows in the modeled system in the Petri net [4].

A place p is k-bounded if the number of tokens in a place p is should be less than or equal to k for every marking M reachable from the initial marking M0 where k is a non-negative integer number. It is safe if it is 1-bounded.

A Petri net N = (P, T, I, O, M0) is k-bounded (safe) if each place in p is k-bounded (safe).

**C. Liveness**

Liveness is the concept which is closely related to the deadlock situation.

A Petri net must be live which is modeled as a deadlock-free system. This indicates that for any reachable marking M, it is possible to fire any transition in the modeled net by progressing through some firing order. This requirement, however, might be too tough to represent some real systems that show a deadlock-free behavior. For some cases, the initialization of a system can be modeled by a set of transitions which fire a finite number of times. After initialization, the system may show a deadlock-free behavior, while the Petri net representing this system is no longer live as mentioned above. For this reason, various levels of liveness for transition t and marking M were defined. [4]

#### ANALYSIS OF MODELED TIME PETRI NET SYSTEM

It is important to analyse the modeled system. This analysis will lead to important insights into the behavior of the modeled system. There are three common approaches to Petri net analysis; reachability analysis, the matrix equation approach, and simulation. The first approach involves the computation of all reachable markings, but there is an issue of the state-space explosion. The matrix equations technique is strong but in many cases, it is applicable only to special situations or special subclasses of Petri nets. Discrete-event simulation is an option to check the system properties for complex Petri net models. [4]

### A. Reachability Analysis

Reachability analysis is done by construction of reachability tree and is also called a coverability tree. We can get the number of *new* markings as the number of the enabled transitions (firing transitions) from the initial marking $M_0$ in a given Petri net PN. From every new marking, we can reach more markings. On repeating the procedure over and over we will get the results in a tree representation of the markings. Markings generated from $M_0$ and its successors represent *nodes*, and every arc represents firing of transition, which will transform one marking to another. [4]

The tree will grow infinitely if the net is unbounded.

Example: Consider the Petri net shown in Fig. 3, all reachable markings are: $M_0 = (2, 0, 0, 0)$, $M_1 = (0, 2, 1, 0)$, $M_2 = (0, 1, 1, 1)$, $M_3 = (0, 2, 0, 1)$, $M_4 = (0, 0, 1, 2)$, $M_5 = (0, 1, 0, 2)$, and $M_6 = (0, 0, 0, 3)$.

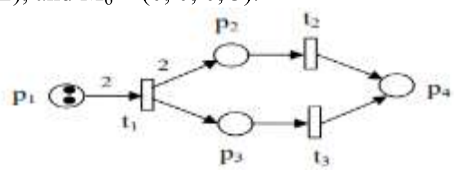


Fig. 3 Simple Petri Net

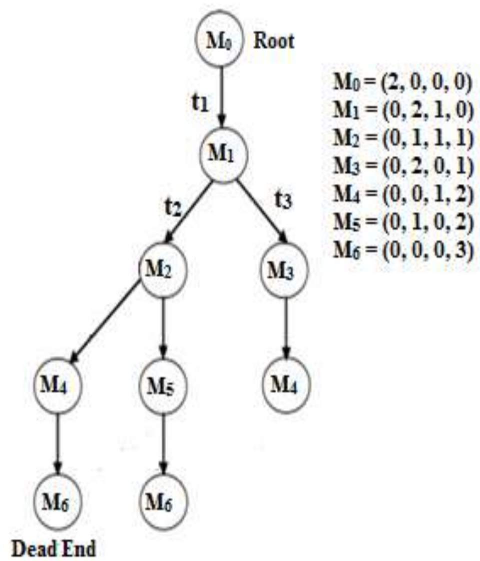


Fig. 4 Reachability Tree

### B. Simulation

To check the system properties, simulation is another way for complex Petri net models. The idea of simulation is simple, that is, by using the execution algorithm to run the net. Simulation is time-consuming and an expensive technique. Simulation can show the presence of undesirable properties but it cannot prove the correctness of the modeled system in general case [4]. Regardless of this, Petri net simulation is truly a convenient and straightforward yet effective approach for engineers to prove the desired properties of a discrete event system. The algorithm is given as follows;

- Initialization: choose the initial marking and the set of all enabled transitions in the marking.
- If the number of predetermined simulation steps or certain stopping criteria is met, stop. Else, if there is no transition enabled, report a deadlock marking and either stop or go to Step (1).
- Randomly pick a transition to fire. Remove the same number of tokens from each of its input places as the number of arcs from that place to the transition and deposit the same number of tokens to each of its output places as the number of arcs from the transition to that place.
- Remove enabled transitions which are modified at the new marking by checking the output transitions of the input places used in Step (3). If the output transitions of the output places in Step (3) become enabled, add those enabled ones. Go to Step (2).

To simulate extended Petri nets such as timed ones, the above-given algorithm can be modified. The benefit of the simulation methods is to allow one to derive the temporal performance of a system under very realistic assumptions. A list of Petri net simulation tools along with their feature descriptions can be found in the Petri Nets world website [5].

**MODELING AND SIMULATION OF A MANUFACTURING SYSTEM**

The flow process of the manufacturing system consists of one Traub semi-automatic lathe machine ($M_1$), one Band Saw machine ($M_2$), one CNC lathe machine ($M_3$), one Thread Rolling machine ($M_4$) and two operators $B_1$ and $B_2$ as shown in figure 5.

The main objective of modeling this manufacturing system is to get the coverability question satisfied,

If, $M_0 = \{P_1\}$ and $M = \{P_{13}\}$, then $M' \geq M$

The manufacturing system modeled by using PN modeling tool as Petri net toolbox in MATLAB environment as shown in Fig. 6. Initially, the author considered fifty parts to be processed from the store.

By doing so it is easy to understand the behavior of the system. To assess the system, various properties of Petri nets such as reachability, safeness, and liveness needs to be obtained. If these properties get satisfy after the simulation process then we can say that the modeled system meets the expectations.

By seeing the properties of the Petri nets, we can check the questions like is the net or system is bounded, is the net or system is repetitive, is the net or system is reachable, is the net or system is coverable, is the net or system is live, is it deadlock-free, using the Petri net toolbox in MATLAB environment. If we get these questions results in the software then we can say that the modeled system satisfies the pre-conditions. In Petri net toolbox in MATLAB environment, the author created a net of the considered manufacturing system as a T*ime Petri Net* model as shown in the Fig.6.

Basically, there are two conditions one is clocks with *transitions* that is called Time Petri Nets (TPN) and clocks with *tokens* which are known as Timed Petri Nets (TDPN). Here, the author has used Time Petri Net. To use TPN, the modeled system must be a bounded network. After creating the net, to check the properties of the modeled system, need to run the simulation in Petri net toolbox in MATLAB environment, as the simulation is one of the analysis methods of the Petri net.

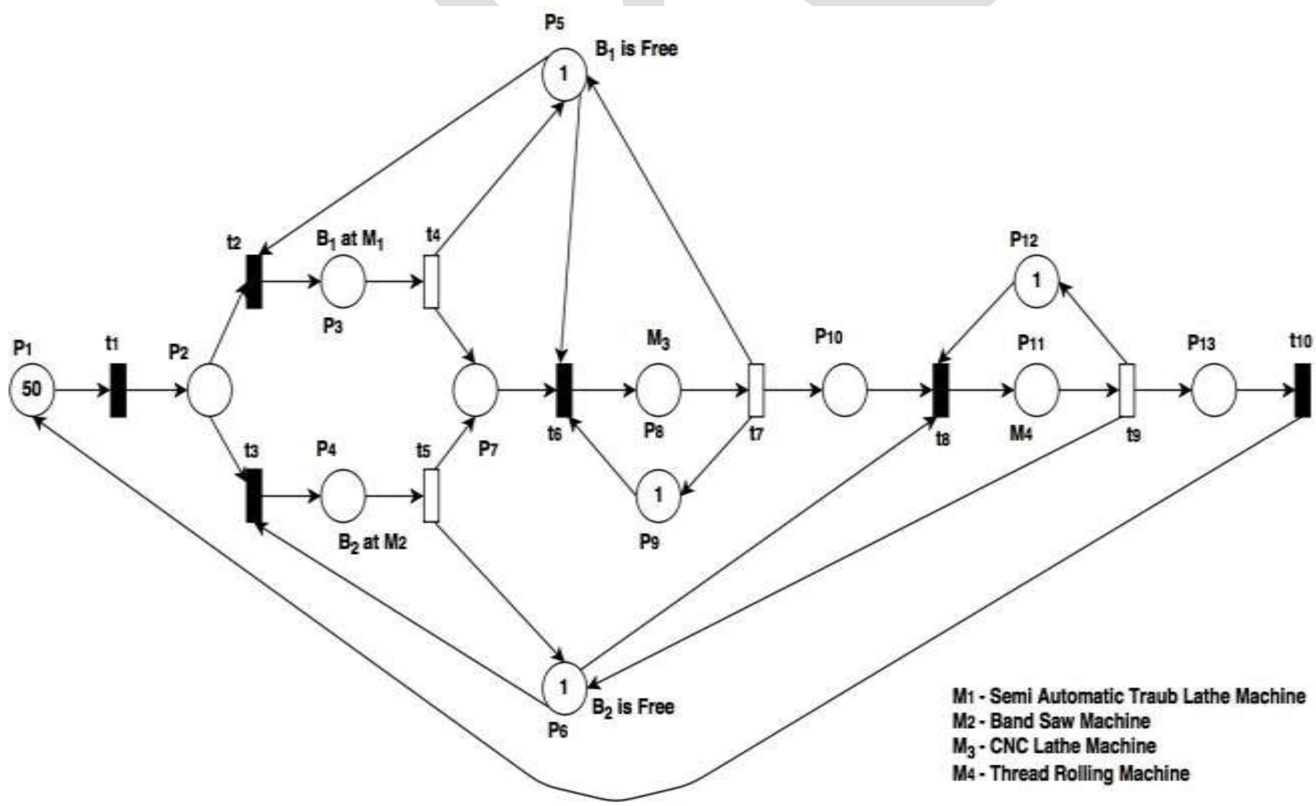


Fig. 6 Time Petri Net Model of a Manufacturing System

On doing simulation analysis author has got the results as shown in Table 1,

| Properties | Result Status |
|---|---|
| Structural Boundedness | The Net is Bounded |
| Structural Repetitiveness | The Net is Repetitive |
| Liveness | The Net is Live |
| Reachability | The Net is Reachable |

As the modeled manufacturing system is live so that the system is completely deadlock-free. If the initial marking is able to reach the final marking then we can say that the net is coverable too or if the net is reachable then it is coverable too.

**CONCLUSION**

In this study, author modeled the manufacturing system as a Time Petri Net (TPN) model with clocks with transitions a well-known subclass of PN. PN is a powerful tool and has been successfully employed to model manufacturing system as they capture various characteristics of manufacturing system such as concurrency, synchronization, and conflicts. A manufacturing system of an XYZ industry has been modeled using TPN approach and the construction has been explained in detail. From the reachability analysis and simulation analysis, it is evident that the Petri Net Model created for the Manufacturing System satisfies the pre-conditions for the Petri Net and which will assist the planner.

The concept has opened a new avenue for the diagnosis and performance measurement of complex manufacturing systems consists of the batch production system of a different variety of products.

**REFERENCES:**

[1] N. Viswanadham and Y. Narahari. Performance Modeling of Automated Manufacturing System, Pearson Education, Inc., Prentice Hall.

[2] Jain A., Jain P. K. & Singh I. P. Performance Modeling Of FMS With Flexible Process Plans A Petri Net Approach. International journal of simulation modeling 2006, vol. 5, no3, pp. 101-113.

[3] Ozkan Basak, Y. Esra Albayrak (2015). Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing system, Computers & Industrial Engineering 86 (2015) 116—126.

[4] J. Wang, Petri nets for dynamic event-driven system modeling, in Handbook of Dynamic System Modeling, Ed: Paul Fishwick, CRC Press, 2007.

[5] http://www.informatik.uni-hamburg.de/TGI/PetriNets/

[6] Viswanadham N., Narahari Y., & Johnson T. (1990). Deadlock prevention and deadlock-avoidance in flexible manufacturing systems using Petri net models. IEEE Transactions on Robotics and Automation, 6(6), 713-723.

[7] Tavana M. (2008). Dynamic process modeling using Petri nets with applications to nuclear power plant emergency management. International Journal Simulation and Process Modeling, 4(2), 130-138.

[8] Viswanadham N., Narahari Y., & Johnson T. L. (1992). Stochastic modeling of flexible manufacturing systems. Mathematical and Computer Modelling, 16(3), 15-34.

[9] Rongming Zhu (2012). A Deadlock Prevention Approach for Flexible Manufacturing Systems with Uncontrollable Transitions In Their Petri Net Models.

[10] Manuel Silva and Robert Valette (2005) Petri Nets and flexible Manufacturing. Advances in Petri nets 1989, 1990.

[11] Abdulziz M., E.-T. S., H. Mian and M. H. Abidi, "Analysis of performance measures of the flexible manufacturing system," Journal of King Saud University - Engineering Sciences, pp. 115-129, 2012.

[12] T. Aized, "Modeling and performance maximization of an integrated automated guided vehicle system using colored Petri net and response surface methods.," Computers & Industrial Engineering, vol. 57(3), pp. 822-831, 2009.