

Analysis and Optimization of Discrete Event Systems using Petri Nets

^{1,2}V. Padma and ¹K. Bhuvaneswari

Dept.of Mathematics

¹ Mother Teresa Women's University

² SRM University, Tamil Nadu, India

padmagurumoorthy@gmail.com

Abstract

Petrinets are a family of powerful discrete event models whose interest has grown with the development of the theory of discrete event systems. This has created needs for modeling, analysis and optimization that can help to evaluate the discrete events system. In this paper, we discuss about the advantages of petrinets as modeling, analysis and optimization tool of manufacturing system. In manufacturing system, systems are very complex and need in depth analysis before implementing them. Discrete Event System (DES) are much more difficult to study than linear system. Petrinets are a tool for study of these systems, petrinet graph models the static properties of the system. As a mathematical tool, it is possible to set up state equation, algebraic equations and other mathematical models governing the behaviour of the system.

AMS Subject Classification: 68Q85, 93B20, 90B10, 93B30, 93B11.

Key Words and Phrases: Petri Net, optimization, identification, and incidence matrix.

1 Introduction

Discrete Event System (DES) are very complex, often large systems of many interacting components. Each component can be quite complex, as can its interaction with other components in the system. This also true for many other systems which all involve many individual components interacting with other components, possibly in complex ways.

As a formal and graphical modeling petrinets have been successfully used for modeling and analysis of many discrete event dynamic systems including manufacturing system, communication systems, distributed database systems, workflow analysis and industrial control systems. The wide and successful applications of petrinets rely on its appealing graphic that enables both qualitative and quantitative analysis.

The paper is structured as follows:

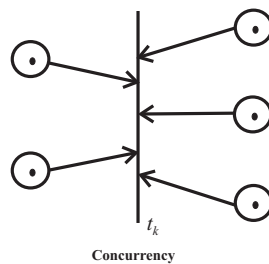
Section 2 introduce the modeling of physical systems with petrinet system. In section 2 the main analysis techniques of petrinets are discussed with some examples. In section 3 the definition of petrinets and its rules are given. In section 4 the basic information about discrete event systems are presented. Section 5 & 6 presents the representation of petrinets as matrix, algebraic equations and coverability tree. Concluding remarks are given in section 7. In this paper we considered the basic petrinet model called place/transition net (PN for short).

2 Analysis Techniques

Petrinets are a formal model that allows one to describe many interesting features of concurrent systems. Once a physical system has been modeled by a petrinet, the properties of interest of the system map fairly well into the properties of the corresponding model.

Methods of analysis for petrinets may be classified into the following three groups.

1. The coverability (reachability) tree method.
2. The matrix - equation approach and
3. Reduction (or) decomposition techniques



The first method involves essentially the enumeration of all reachable marking or their coverable markings. It should be able to apply to all classes of nets, but is limited to "small" nets due to the complexity of the state-space explosion. On the other hand, matrix equations and reduction techniques.

Among the methods that have been developed for the analysis of a petrinet, those based on reachability analysis are of particular importance. Typically they require solving the marking reachability problem. (i.e.) Determining if a given marking is reachable from the initial one. It plays an important role in petrinet theory since many properties liveness and dead lock freeness require the reachability analysis of the system and many other problem like deadlock avoidance and controllability analysis are equivalent (or) can be reduced to the marking reachability problem. In some restrictive subclasses of petrinets such as acyclic nets, the marking reachability problem can be determined by solving an integer linear programming problem and in state machines and marked graphs it can be solved in polynomial time.

3 Optimization

As a powerful modeling tool of DES petrinets are well adopted for describing the dynamic evolution and behavior of the system. Given that a system can be designed and controlled to achieve some desirable performance it is a nature task "how can we control it so as to achieve the best possible performance". This goal requires the development of additional analytical (or) experimental techniques for efficiently determining the optimal system behavior and the means for attaining it.

One of the main advantages of petrinets is that the state is a vector of non-negative integers, while it is usually non-numerical in other discrete models such as automata.

Example 1:

Let us consider the system composed of two machines whose automaton and petrinet model are shown in Fig. 1 (a) and Fig. 1 (b).

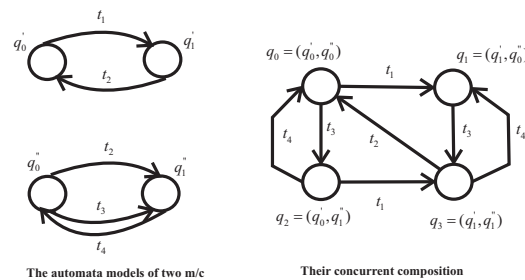


Fig. 1(a)

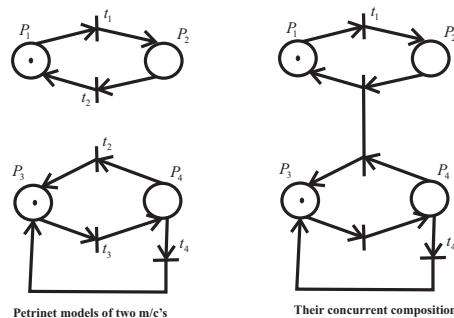


Fig. 1(b)

State q_0' (respectively q_0'') denotes first (respectively second) machine is idle, state q_1' (respectively q_1'') denotes that the first (respectively second) machine is working.

the petrinet state is represented by a non-negative vector. Marking $[1010]^T$ corresponds to the state in which both machines are idle. The marking $[0110]^T$ corresponds to the state in which the first machine is working and second is idle and so on. Using a petrinet model the state space of this system, that is a series of labels with no algebraic structure can be described by a set of vectors (i.e.) by highly structured net.

This also allows to describe logical specifications in a numerical form. Assume for instance, that we want to impose that the first

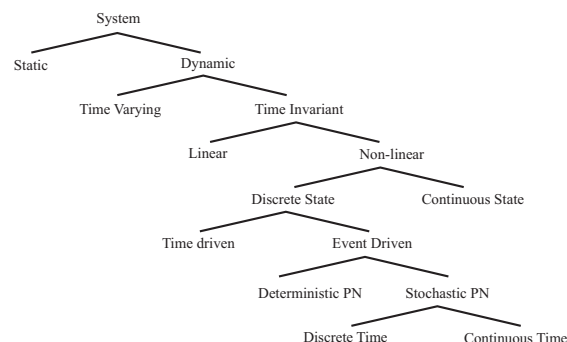
machine should never be working of the second machine is idle. Using the notation in diagram 1 (b) such a constraint can be imposed forcing the constraint $M(P_2) + M(P_3) \leq 1$.

The possibility offered by petrinets to describe the state space of a discrete event system that may have absolutely no algebraic structure with a set of integers vectors has an important implication. In fact it is possible to apply algebraic formations such as integer programming for the analysis and control of these symbols. A final class of problems arises from optimization considerations. If a petrinet exhibits a certain behavior as indicated by its set of transition firing sequences and its reachability set.

4 Discrete Event Systems (DES)

DES are systems whose state space is discrete and whose state can only change as a result of asynchronously occurring instantaneous events over time. Sample paths of DES are typically piecewise constant functions of time. Conventional differential equations are not suitable for describing such "discontinuous behavior". Classical control theory deals with time driven systems modeled by difference / differential equations. However, many properties of dynamical systems have been defined in very good terms that are model independent.

In system theory a DES is composed by actuators, the system itself and sensors. Actuator signals and sensor signals will be modeled as input and output alphabets respectively. It is represented by the following diagram:



5 Petrinets

A petrinet is usually defined a graph with two kinds of nodes called places and transitions and an interconnecting flow relation F which does not connect two nodes of the same kind. F is the union of disjoint relations $PT \subseteq (P \times T)$ and $TP \subseteq (T \times P)$ where P denotes the set of places and T denotes the set of transitions.

Definition 6.1:

A petrinet (N) is a bipartite directed graph usually represented by tuple $N = (P, T, F)$ where P : a finite set of places that is $P = \{P_1, P_2, \dots, P_n\}$ for $n > 0$.

T : a finite set of transitions that is $T = \{t_1, t_2, \dots, t_m\}$ for $m > 0$.

So that $P \cap T = \phi$.

$F \subseteq (P \times T) \cup (T \times P)$ is the incidence relation representing a set of directed arcs connecting places to transition and transition to places, corresponding to the input function and output function (flow relation).

6 Graphical Representation

(i) Place

It is denoted by a circle (or) ellipse. It is commonly used to represent the condition that exists in the modeled system. Input places of transition denote the conditions for the occurrence of the event whereas the conditions after the occurrence of the event represented by the output places of transition.

(ii) Transition

It is denoted by a solid bar (or) rectangle. It represents action, events, activity (or) task that is required in the system.

(iii) Arc

It is denoted by an arrow, between places and transitions and is always directed. An arc can connect only two nodes that belong to different types. Input arcs connect places to transitions, while output arcs start at a transition and end at a place. There cannot be an arc from a place to place (or) transition to transition. The symbols of place, transition and arc are given in Fig. 6.1.

(iv) Token

It is represented by a black dot. Tokens are assigned to/and residing inside the circles representing the places. In classical/ordinary

petrinets, the tokens do not represent any specific information and are not distinguishable. They are only markers indicating the presence (or) absence of data (eng) a signal, a piece to be machined etc. At any time places can hold arbitrary number of tokens or they can be restricted as to the number they can hold.

(v) Marking

The marking of a petrinet denoted by M , is the mapping $P \rightarrow \{0, 1, 2, \dots\}$ that is the number of tokens in the place. Generally, if a place is marked (i.e.) hold one (or) more tokens in it, the condition represented by the place is regarded as true. The marking, that is distribution of tokens on places represents the state of a modeled system. The state of the system changes when the distribution of tokens changes. The marking can be represented by an n -dimensional column vector $M = \{M_1, M_2, \dots, M_n\}$ where $n = |P|$ is the number of places. A marked petrinet $(FN)PN = (N, M)$ is a net structure $N = (P, T, F)$ with a marking M . So petrinets are also defined as $PN = (P, T, F, M)$.

(vi) Petrinet Execution

Enability Rule: A transition 't' is said to be enabled iff there is atleast one token each input place of 't'.

Firing Rule: When a transition enabled it may fire (or) occur. On firing it removes the token from all of its input places and deposits a token in each of petrinet its output places.

The execution of a petrinet is done by the number and distribution of token in its places. A petrinet executes when the enabled transition fires and consequently change the state of the model system.

Example: 2

Consider the PN namely PN_1 of the following fig.3 the incidence matrix the S-invariant vector is represented in the following equations:

$$\text{Incidence matrix of } PN_1 \text{ is } C_1 = \begin{bmatrix} & t_1 & t_2 & t_3 \\ P_1 & 1 & -1 & 1 \\ P_2 & 1 & 0 & -1 \\ P_3 & -1 & 1 & 0 \\ P_4 & -1 & 0 & 1 \end{bmatrix}$$

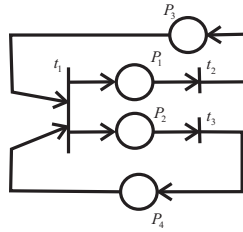


Fig. 6.2

The vector $X = [X_1 X_2 X_3 X_4]^T$ will be an S-invariant iff $C_1^T X = 0$ (or) equivalently if

$$\left. \begin{aligned} X_1 + X_2 - X_3 - X_4 &= 0 \\ -X_1 + X_3 &= 0 \\ -X_2 + X_4 &= 0 \end{aligned} \right\} \quad (1)$$

Solving the simultaneous system (1), we get

$$X_1 = X_3, X_2 = X_4 \quad \therefore X = [X_1 X_2 X_3 X_4]^T.$$

The two minimal support S-invariants are obtained by setting X_1 equal to one and X_2 equal to zero (or) X_1 equal to zero and X_2 equal to one.

And $X_1 = [1010]^T$ with $\langle X_1 \rangle = \{P_1, P_3\}$.

$X_2 = [0101]^T$ with $\langle X_2 \rangle = \{P_2, P_4\}$.

The corresponding S components are shown in fig. 4.

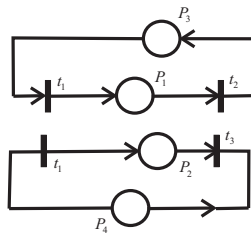
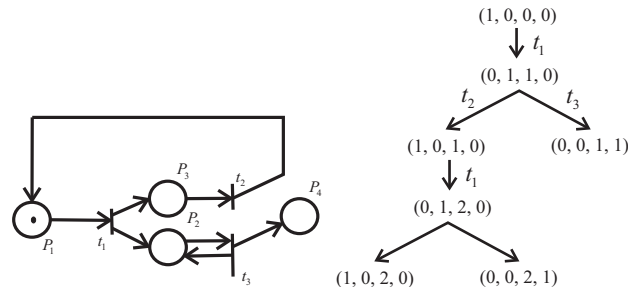


Fig. 6.3

The two S-components of PN were constructed from minimal support S-invariants, they are minimal S-components, they do not contain any other S-components by themselves. For example consider another petrinet PN_2 where the token is assigned in P_1 . The following fig. shows the firing of transitions and its reachability tree of the petrinet PN_2 .

Reachability Tree:**7 Conclusion**

Petrinet based approach of analysis and optimization of DES were discussed with some simple examples. The limitation in the optimization of systems by petrinets is generally accepted. The firing of a transition is considered to be an instantaneous event taking zero time and the occurrences of two events cannot happen simultaneously. The events are called primitive events, this need not cause problems in the optimization of a system. But non-primitive events cannot be properly modeled by transitions in a petrinet but it can be modeled by timed petrinets.

References

- [1] T. Murata, Petrinets: Properties, Analysis and Applications, *Proc. of the IEEE*, **77(4)**, April 1989, 541-580.
- [2] David R and H. Alla, Petrinets and Grafset: Tools for Modeling Discrete Event Systems, Prentice Hall, London, 1992.
- [3] Peterson, James L., Petrinet Theory and the Modeling of Systems, Prentice Hall, Englewood Cliffs, NJ, 1981.
- [4] Reisig, Wolfgang, Petrinets: An Introduction, Springer-verlag, Berlin, 1985.
- [5] Ramadge, P.J., Wonkium, W.M., The Control of Discrete Control Systems, *Proceedings of IEEE*, **77(18)**, 1989, 81-98.

