Petri-Net Robot Models for Robot Networks

Hussain Alayan and Robert W. Newcomb**
Microsystems Laboratory
Electrical Engineering Department
University of Maryland
College Park, MD 20742 USA
Phone: (301) 454-6869

Abstract:

This paper introduces Petri-Net models of robots for the analysis and design of robot networks.

Several levels of robot models are introduced. The simplest meaningful model is a two-transition four-place one which can be used for quick "executive" modeling for first or quick designs. Beyond this a detailed technician level model is introduced which can model fine points of the robot operation. The state equations for these models are given and application to networks of robots illustrated by the modeling of a ring network.

I. Introduction

Robot networks for industrial needs can be very extensive and involve quite complicated operations elaborate connections, as for example can be seen by examining the Gardner testing line [1] for automated testing of printed circuit boards. Consequently, there is a need for mathematical descriptions of interconnected robots such that computer aided analysis and design of such elaborate robot systems can be undertaken. For such a purpose we introduce in this paper place-transition network models, otherwise called Petri-Nets and abbreviated PTN, for describing the connection of robots. The models we introduce include the simplest possible ones for executive design and very elaborate ones for technician level design. These models provide a framework for determining the dynamic behavior of the robot network system structure, throughput, communication cost and a basis for determining control.

* Supported by the US-Spain Joint Committee for Scientific and Technological Cooperation Grant No. CCB-84-02 and NSF Grant ECS-8317877 and its extension. ECS-85-06924

* also with Grupo de Trabajo Sistemas (PARCOR)
Facultad de Informatica
Universidad Politecnica de Madrid
Ctra. de Valencia, Km. 7,00
28031 Madrid, Spain

Petri-Nets are ideal for describing interconnected objects such as robots in networks since they can abstract the operations into a graph theory framework and at different levels of complexity for different purposes. For PTN background we refer to the papers of Peterson and Murata where the concepts are clearly formulated [2] - [5]. But to establish our notation we mention that a PTN is taken as a directed graph with a node being either a place or a transition. Thus, there is a set of m places $(P_1, \ldots, P_m)$, with each place allowing the presence or absence of a token, and a set of n transistions $(T_1, \ldots, T_n)$, with each transition capable of either firing or not firing. Along with these at time λ are a marking m-vector $M(\lambda)$, an input m-vector $In(\lambda)$, and a firing n-vector $F(\lambda)$ all with entries being binary numbers, with a 1 representing the presence of a token (in the case of M or In) or the firing of a transition (in the case of F). The PTN has a set of directed arcs connecting nodes where an arc a_{ij} can be drawn from place i to transition j or from transition i to place j (with the entity at i the input and that at j the output). To represent these connections there is an nxm connection matrix, $C=[c_{ij}]$, where $c_{ij}=-1$ if node i is an output from node j, =0 if nodes i & j are disconnected, =+1 if node i is an input to node j. Then one can write [5] for all positive integer λ (using a superscript T to denote matrix transposition and a dotted equality sign to denote replacement of any positive component on the right by 1)

$$M(\lambda) \doteq M(\lambda-1) + In(\lambda-1) + C^T F(\lambda) \qquad (1)$$

Here λ represents the time at which a transition is considered and its increments are normalized to be unity, this representing the least time needed for a robot activity (it being assumed that all transitions occur at multiples of this least time). For (1) the dotted equality sign is needed to reduce all entries in $M(.)$ to be binary numbers. In Section II we obtain this representation for the models given and in Section III we use it on a ring network of robots.

II. PTN Robot Representation

In this section we demonstrate that a PTN can be used to describe a robot network in a top down fashion at various levels of abstraction and detail. An entire network may be replaced by a single place or transition for modeling at executive level, or places and transitions may be

CH2255-8/86/0000-0996 $1.00 © 1986 IEEE

replaced by more elaborate configurations to give more detailed modeling.

II.1. Executive Levels

Accordingly, a robot with input and output can be represented as in Fig. 1 by a single place called R; communication and connections to other robots are represented by single transitions. This simple represention of a robot is what we call level zero modeling with Figs. 2 & 3 using it to represent a three robot network in ring and feedforward forms, respectively. These two figures could be used for quick designs but since level zero is very primative, it is much more useful to advance to level one, which is shown in Fig. 4 and which can still be used for first or quick executive designs. In the level one model the fact that a robot takes some internal actions is incorporated with the places and transitions representing the following:

- P_1: the robot has an input on hand for processing
- P_2: the robot is taking action on the received input
- P_3: processing is idle
- P_4: the robot has an output to deliver
- T_1: robot input is transferred for action
- T_2: robot activity is complete

If there is only this one robot in a network then the level one model gives for (1)

$$\begin{bmatrix} P_1(\lambda) \\ P_2(\lambda) \\ P_3(\lambda) \\ P_4(\lambda) \end{bmatrix} = \begin{bmatrix} P_1(\lambda-1) \\ P_2(\lambda-1) \\ P_3(\lambda-1) \\ P_4(\lambda-1) \end{bmatrix} + \begin{bmatrix} In_1(\lambda-1) \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ -1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_1(\lambda) \\ F_2(\lambda) \end{bmatrix} \quad (2a)$$

If the initial marking is as shown in Fig. 4 then

$$M(0) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (2b)$$

And if the robot is part of a larger network, we note that (2a) is a subportion of a larger description.

II.2. Technician Level

In many cases the level one model is too primative to yield the detail needed about the operation. Then it is important to have more detailed robot models. In essence refinement to any desired detail can be made but the one we concentrate on here, called the technician level model, should usually be sufficient and in any event illustrates the salient points. And, although robots come in a variety of architectures, they all have some kind of input sensors (such as a camera), some kind of actuator devices for output (such as arms that deliver finished products), and a microprocessor with its own local memory. Further, the microprocessors of two robots in a network may communicate directly with each other. Incorporating these concepts leads to the technician level model of Fig. 5 where 16 places and 8 transitions are used. In Fig. 5 a correspondence of some of the nodes can be made with those of Fig. 4, for example the input P_1 is the same in both cases while the (product) output P_8 of the technician level model corresponds to P_4 of Fig. 4. In Fig. 5 P_9 & P_{13} are direct microprocessor signal inputs from adjacent robot processors (on the left and right, say) while P_{10} & P_{14} are similar signal outputs that allow for direct communication between robot microprocessors. Table I gives the meanings we ascribe to the places and transitions of this technician level model.

III. Robot Network Representation

Given a robot network we can replace every robot in it by one of the PTN models given here along with other connection transitions to represent communication links, as has already been illustrated in Figs. 2 & 3 using level zero models. In turn we can replace a given level model by one of a different level. For example, consider a ring network of robots for which the zero level model is that of Fig. 2; on replacement by level one models the PTN graph of Fig. 6 results. In Fig. 6 the second subscript on a place or transition denotes the robot number while the first subscript refers to the transistion or place as it occurs in the level one model of Fig. 4. The state equations (1) are readily found for which we note that different initial markings M(0) can represent significant differences in use of the network. For example, if the initial marking is that shown in Fig. 6 then the network can be considered as progressively taking one item through the network, finishing it completely before another item is started (that is, with that marking two robots and their processors are idle while the other one is doing its job). By adding tokens to the input places of the second and third robots a more efficient use of the robots is undertaken since all of the robots will be working simultaneously.

IV. Discussion

Given a robot network one can represent it via Petri-Nets as we have described here. This representation can take different forms by using different levels of PTN models for the robots contained. Conversely, given a job that it is desired to set up a robot network to perform, one can design the systems via PTN's by laying down a suitable PTN, first at the executive level to set the desired structure and then at technician level to do computer aided analysis to check that the design performs as needed. In this design area more research is needed to be able to proceed in a more synthesis type of manner, but the paper of Suzuki & Murata [6] seems a good step and one to develop further. Also it is important to develop equations for the firing vector $F(\lambda)$ since it controls the operation. Because of the laws of the transitions, there is a Boolean expression for the entries of the firing vector (albeit a rather messy one) and it is this which needs to be programmed on a computer to complete equation (1) for computer aided analysis of robot networks. Among the results of computer aided analyses using the models presented here would be time for parts to go through a production line, checking for deadlock, cost effectiveness of different configurations, effects of robot component failures, and (through the introduction of probabilistic quantities) reliability measures.

References

[1]. L. B. Gardner, "Managing the Integration of Robotics with ATE," Electronics Test, Vol. 7, No. 2, February 1984, pp. 89 - 99.

[2]. T. Murata, "Petri Nets and Their Application, An Introduction," in Management and Office Information Systems, edited by S.-K. Chong, Plenum Publishing Co., 1984, pp. 351 - 368.

[3]. J. L. Peterson, "An Introduction to Petri Nets," Proceedings of the National Electronics Conference, Vol. 32, October 1978, pp. 144 - 148.

[4]. R. Johnsonbaugh, "Petri Nets and Marked Graphs - Mathematical Models of Concurrent Computation," American Mathematical Monthly, Vol. 89, No. 8, October 1982, pp. 552 - 566.

[5]. T. Murata, "State Equation, Controllability, and Maximal Matchings of Petri Nets," IEEE Transactions on Automatic Control, Vol. AC-22, No. 3, June 1977, pp. 412 - 416.

[6]. I. Suzuki and T. Murata, "A Method for Stepwise Refinement and Abstraction of Petri Nets," Journal of Computer and System Sciences, Vol. 27, No. August 1983,pp.51-76.

Table I
Meaning of Places and Transitions for Technician Level Model

P_1: the robot has an input
P_2: the processor is receiving data from the input sensor
P_3: the processor has finished receiving data from the sensor
P_4: the robot is ready to receive data from the sensor
P_5: sensor data is stored and available for the processor
P_6: processor data for the actuator is stored
P_7: processor data for the actuator is transmitted
P_8: the robot has an output to deliver
P_9: the robot receives a message from a neighboring robot
P_{10}: the message for P_9 is finished
P_{11}: the message for P_9 is stored
P_{12}: the processor has stored its data
P_{13}: the processor is idle and ready to work
P_{14}: a message for a neighboring robot is stored
P_{15}: the message of P_{14} is ready to send
P_{16}: the neighboring robot ready to receive data

T_1: the sensor transmits data to the processor
T_2: the processor is receiving data from the sensor
T_3: the processor acknowledges receipt of sensor data
T_4: the processor generates actuator commands
T_5: the processor is ready to transfer data to the actuator
T_6: the actuator has received a message from the processor
T_7: the processor is receiving a message from a neighbor
T_8: the processor has prepared a message for a neighbor

Figure 1. Robot Model Level Zero

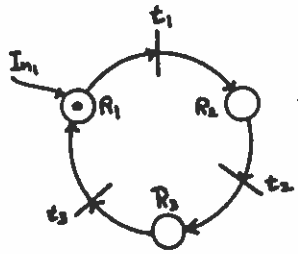

Figure 2. Three Robot Ring Network

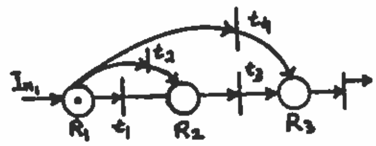

Figure 3. Three Robot Feedforward Network

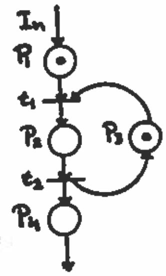

Figure 4. Robot Model Level One

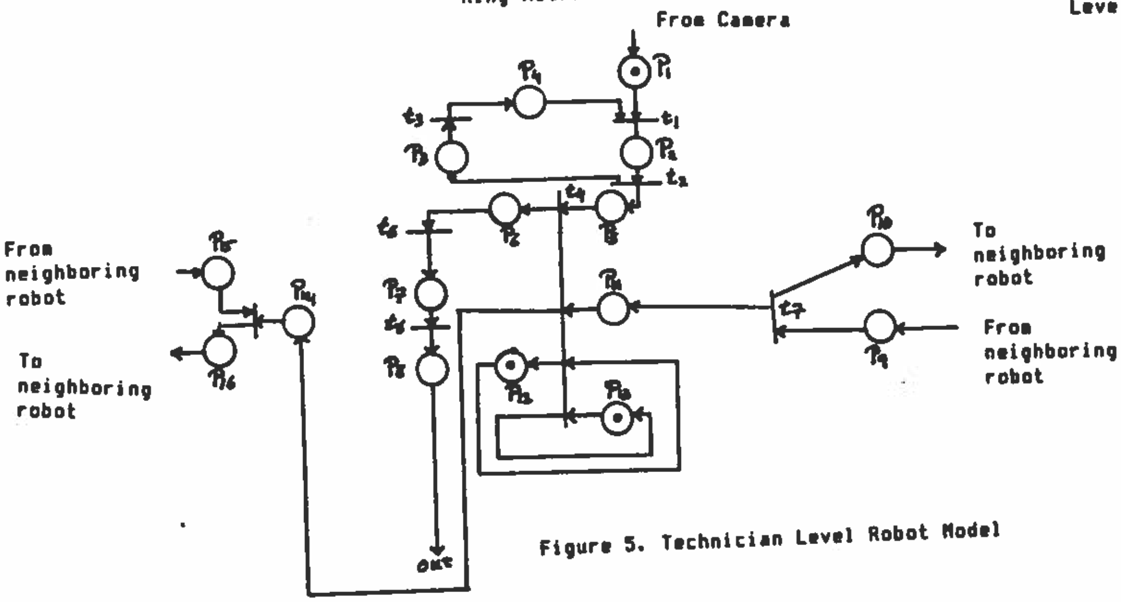

Figure 5. Technician Level Robot Model

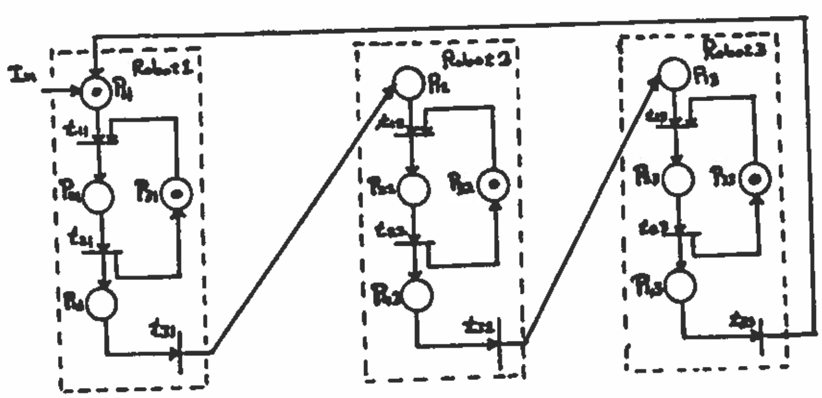

Figure 6. Three Robot Ring Using Level One Models

1986 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS

Volume 3 of 3

Le Baron Hotel
San Jose, California
May 5 – 7, 1986

86CH2255-8