Int J Adv Manuf Technol (1991) 6:346–363

The International Journal of Advanced Manufacturing Technology

BUILDING AGV TRAFFIC-CONTROL MODELS WITH PLACE-TRANSITION NETS

S. Hsieh and K.-H. M. Lin

Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan 10764

Owing to the lack of robust design and control algorithms, most current applications of automated guided vehicle systems (AGVSs) employ simple control methods despite the fact that the system is far from efficient. Recently, Petri nets have evolved into a powerful tool for modelling complex manufacturing systems. One of the advantages of the use of Petri nets is that analysis, simulation and on-line control can all be done on the same model once the model is built. The purpose of this paper is to establish the research fundamentals in the field of the Petri-net modelling of an AGVS. The main contribution of the paper is to define basic traffic-control nets which can be used directly to model an AGVS without too much thinking. Some basic AGV Petri-net control elements are described and illustrated in detail. Also, difficulties in the use of bidirectional flows are discussed, and it has been determined how Petri nets can be used to solve these problems.

Keywords: AGVS; PTNs; Synthesis; Zone control

1. Introduction

Flexible manufacturing systems (FMSs) evolving from traditional manufacturing systems have been of great interest to industry. The area consists of several subfields, such as flexible manufacturing cells (FMCs), automated storage/retrieval systems (AS/RSs) and automated material-handling systems.

Accepted for publication: 15 July 1990

The automated guided vehicle system (AGVS) has been proven to be the most popular material-handling system employed in FMSs [1]. This is because of its flexibility and capability when compared with other systems. However, the AGVS does raise more serious and challenging operational control problems. The degree of sophistication of the system is dependent on the number of vehicles, the level of system automation, the use of uni- or bidirectional paths, etc. Most current applications in manufacturing-shop environments employ unidirectional guide paths for vehicle routeing despite the fact that bidirectional vehicles exist [2]. One of the justifications for the use of unidirectional flows is their simplicity.

AGVSs present different aspects from design to implementation. The study of the dispatching vehicles employs queuing networking theory as an analysis tool [3]. The study of their layout employs the method of discrete-event system simulation [4]. Methods of operations research, such as transportation theory, are applied to the study of the system design [5]. Optimisation methods are used to determine the number of vehicles needed in the system [6]. The Markov decision process is adopted for the AGVS's control rules [7]. From past research, it seems that a model developed for the study of the system design will not be suitable for the system simulation, and the approach of building a simulation model is different from that of building an on-line control model; the effort made towards one aspect is useless for another aspect. Another phenomenon is that few attempts have been made towards the study of the basic traffic-control elements or substructure. When the basic traffic-control elements are established, and each of them represents a special qualitative property, they may break down a huge and complicated AGVS problem into many simple and straightforward subproblems. Therefore, the AGVS model can be developed step by step with ease.

Petri-nets tools are suitable for modelling the dynamic behaviour of discrete concurrent systems. There are many advantages in modelling a system using Petri nets, for example:

1. At the very beginning of the design stage, the use of Petri nets to model a system can easily be understood, since they are presented graphically.
2. Later in the design stage, a systematic and complete qualitative and quantitative analysis of the model is made possible by the well-developed Petri-net analysis technique [8].
3. At the design-validation stage, with the existence of the Petri-net model, off-line system simulation can be done to validate system design and synthesis [9].
4. At the implementation stage, the simulation model can be used directly as the on-line system-control model.
5. Performance evaluation of systems is possible if timed Petri nets are used.
6. Any Petri-net model can be treated as a submodel of another Petri-net model, i.e. Petri nets can model a system hierarchically.

Generally speaking, the use of Petri nets to model a system can be divided into two parts: the building of the net structure and the marking of the net. In AGVS applications, the net structure means the traffic-control structure, and the marking of the net means the routeing of the automated guided vehicles (AGVs).

The purpose of this paper is to establish the research fundamentals in the

field of the Petri-net modelling of an AGVS. The main contribution of the paper is to define basic traffic-control nets which can be used directly to model an AGVS without too much thinking. The routeing problem will be investigated in another paper. Some basic AGV Petri-net control structures are described and illustrated in detail in this paper. Also, difficulties in the use of bidirectional flows are discussed, and it will be determined how Petri nets can be used to solve these problems.

2. Review of Petri Nets

Petri nets, also known as place-transition nets (PTNs), were first described in Petri's dissertation "Communication with automata" (1962). Several researchers [9–11] have pointed out that Petri nets are suitable tools for the description and analysis of manufacturing systems that exhibit both asynchronous and concurrent properties.

The structure of a PTN is a bipartite directed graph consisting of a set of places, a set of transitions, and a set of directed arcs which connect places to transitions or vice versa. Pictorially, places are represented by circles and transitions by bars. Places may contain tokens (drawn as dots): a PTN with tokens is a marked PTN. A marking of a marked PTN is a vector, the elements of which give the distribution of tokens in the places of the net. A marking represents a state of the system being modelled. Generally, places represent conditions, and transitions represent events. If a place represents a condition, then the presence (or absence) of tokens means the truth (or falsity) of the condition. A place is an input (or output) place of a transition if an arc exists from the place (or transition) to the transition (or place).

Dynamic Behaviour

The dynamic behaviour of a system is modelled as follows. The occurrence of an event is represented by the firing of the corresponding transition. The movement of tokens in the net resulting from the firing of one or more transitions represents a change in the system state. The following are the firing rules for marked PTNs:

1. A transition is firable when each of its input places contains at least one token.
2. A transition can fire only if it is firable.
3. When a transition fires,
   a token is removed from each of its input places, and
   a token is deposited into each of its output places.

Note that it is not necessary to fire all firable transitions, although only firable transitions may be fired.

Qualitative Properties [8, 12]

A PTN is said to be pure, or self-loop free, if and only if no place is an input place and an output place for the same transition. A pure PTN can be completely defined by its incidence matrix. Several qualitative properties can be obtained by checking the incidence matrix of a PTN model. These are defined as follows:

1. *Boundedness*. A place in a marked PTN is said to be k-bounded if and only if there are no more than k tokens in the place at the same time for all markings contained in the reachability set of the PTN, where k is a positive integer.
2. *Liveness*. A transition in a marked PTN is said to be live if and only if for all reachable markings there exists a sequence of firings that results in a marking in which the transition is firable.
3. *Conservation*. A PTN is said to be conservative if and only if the total number of tokens in each reachable marking is the same.

These three qualitative properties can be checked out by analysis of the net's incidence matrix and S-invariants [8].

3. Single-Loop Traffic-Control Structure

In this section, a single-loop unidirectional guide-path AGVS with zone control is used to demonstrate the problem. Petri nets are used to model the system. Before the Petri-net structure is built, several assumptions have to be made:

1. Vehicles move loads without human operators.
2. Paths are divided into segments.
3. Each segment represents a zone.
4. The length of a zone is sufficient to hold one vehicle, plus an allowance for safety and other necessary considerations.

The operation rules of the zone control are as follows:

1. Every zone must have at least one stop station.
2. Only one vehicle is permitted in a zone (this implies that the PTN is 1-bounded).
2. When a vehicle occupies a given zone, any other vehicle which needs to occupy or pass through the zone can get into the next unoccupied zone behind the lead vehicle, where it waits.

A four-zone two-vehicle system is illustrated in Fig. 1. At the beginning, one vehicle, V_a, is at zone 1, and the other, V_b, is at zone 2. (Assuming that the characters of vehicle 1 are different from those of vehicle 2, the identification of vehicles is necessary.) Each vehicle has four possible states, represented by the symbols Z_{ia} and Z_{ib}, $= 1, \ldots, 4$; the Z represents the zone, the i represents the zone number, and the a or b represents the vehicle. If V_a (or V_b) is at zone i, then a token will be in state Z_{ia} (or Z_{ib}) to indicate the current status of the system.

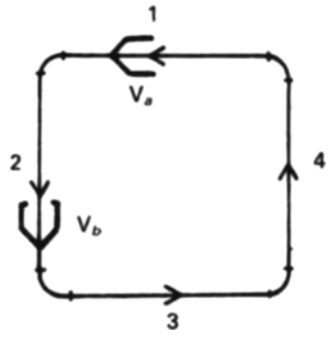

Fig. 1.

The availability of the zone has to be modelled. In this case, there are four zones and two possible states – available or not available – for each of them. When a token is placed in the zone, it means that the zone is available; otherwise, the zone is occupied by a vehicle. The PTN traffic-control structure is thus developed in Fig. 2 [13]. P_1, ..., P_4 and P_9, ..., P_{12} are used to represent the states of Z_{ia} and Z_{ib}, where $i = 1, \ldots, 4$, respectively. P_5, ..., P_8 represent the availability of zones 1, 2, 3 and 4. The transitions t_1, ..., t_8 represent the execution of the vehicle-move command. For example, if a command is given for V_a to move from zone 1 (i.e. P_1 contains a token) to zone 2, t_1 needs to be fired. Since zone 2 is not available (P_6 does not contain a token), t_1 cannot be fired, and hence the command is overridden.

As another example V_b is asked to move to zone 4, so t_6 and t_7 need to be fired one after the other. In this example t_6 has two inputs – P_{10} and P_7 – and both of the input places contain a token. Hence, t_6 can be fired immediately. After t_6 is fired, tokens will be distributed to P_{11} and P_6, which are the output places of t_6. In this moment, the token distributions have been changed: P_1,

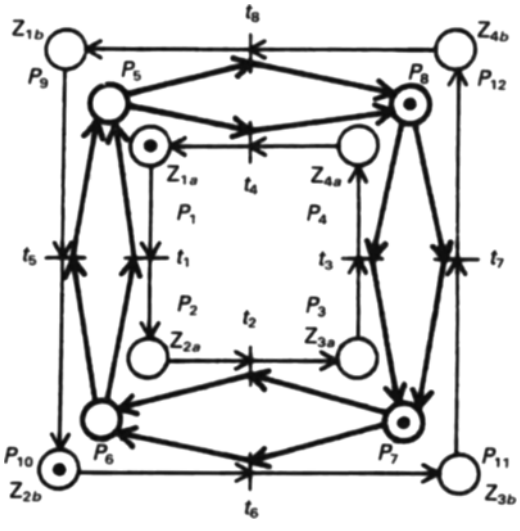

Fig. 2.

P_6, P_{11} and P_8 have tokens, and others do not. The t_7 transition is able to be fired. After t_7 is fired, V_b is moving towards zone 4, and zone 3 will become available.

The Petri-net model in Fig. 2 is pure. The corresponding incidence matrix N is

$$N = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{11} & P_{12} \end{array} \\ \left[\begin{array}{cccccccccccc} -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{array}\right] \end{array}$$

The initial marking M_0 is

$$M_0 = (1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0)$$

According to the firing rules, only t_6 is firable. This indicates that only V_b in zone 2 can move.

In the second example, after the firing of t_6, V_b moves into zone 3, and the resulting marking M_1 is

$$M_1 = (1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0)$$

After the firing of t_7, V_b moves into zone 4, and the resulting marking M_2 is

$$M_2 = (1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1)$$

This can be restated as follows:

$$\begin{aligned} M_1^{\mathrm{T}} &= M_0^{\mathrm{T}} + N \cdot f_1^{\mathrm{T}} \\ M_2^{\mathrm{T}} &= M_1^{\mathrm{T}} + N \cdot f_2^{\mathrm{T}} \\ &= M_0^{\mathrm{T}} + N \cdot f^{\mathrm{T}} \end{aligned}$$

where f_1, f_2 and f are the firing vectors

$$f_1 = (0, 0, 0, 0, 0, 0, 1, 0, 0)$$
$$f_2 = (0, 0, 0, 0, 0, 0, 0, 1, 0)$$

and $f = f_1 + f_2$

The above deduction confirms that the PTN model can emulate the dynamic behaviour of the system by token plays. Also, a comparison of the operation rules of the zone control with the firing rules of the PTN indicates that the Petri-net model does synthesise the zone-control functions.

When vehicles are identical or the vehicle identification is not a subject of the study, the above PTNs can be folded into simpler zone-control PTNs as shown in Fig. 3 [14]. It is important to ascertain that the features or properties of the model in Fig. 2 are the same as those of Fig. 3. Both Figs. 2 and 3 show that the model is pure or self-loop free. Hence, they can both be represented by their corresponding incidence matrices. By calculating the S-invariants of the two models, it is verified that both models share the same qualititative properties [15]. The S-invariants of Fig. 2 are

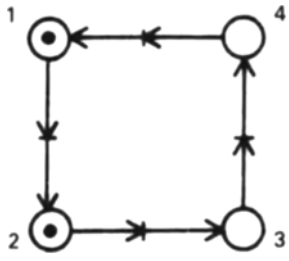

Fig. 3.

$I_1 = (1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$
$I_2 = (0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0)$
$I_3 = (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1)$

and the S-invariant of Fig. 3 is

$I = (1, 1, 1, 1)$

The meaning of the S-invariant of a PTN is that it conserves the number of tokens in itself [15]. The physical meaning in this study is that the total number of vehicles always remains constant during the operation. In fact, Fig. 2 is a coloured Petri-net (CPN) model, and it is an extension of Fig. 3. When the number of vehicles is one, both PTN models will be the same. As the number of vehicles increases, the PTN in Fig. 2 is extended proportionally with the addition of connecting nodes, and the incidence matrix grows quickly. The system becomes complex and difficult to analyse. More details about CPNs are given by Lautenbach [14].

It is worthwhile mentioning that the time factor needs to be incorporated into the PTN model when the system is no more a single loop. In the next section, merge and intersection problems will be discussed. The PTN model built in this section will not be adequate to synthesise the functions of zone control for systems with merges or intersections, and modified petri-net (MPN) traffic-control structures will need to be developed.

4. Complex Traffic-Control Structures

In this section, a more complex layout of the AGVS is discussed. Unidirectional, mixed and bidirectional guide paths are employed for structuring the merge and intersection PTNs.

Merge Structure for a Unidirectional System

Fig. 4 shows the layout of a unidirectional AGVS in which there are seven zones and two vehicles. Initially, the vehicles are at zones 2 and 7, respectively. According to the previous study, the corresponding CPN structure is as in Fig. 5. The tokens in places Z_{2a} and Z_{7b} represent the current positions of vehicles V_a and V_b, and those in places Z_1, Z_3, Z_4, Z_5 and Z_6 represent the availability

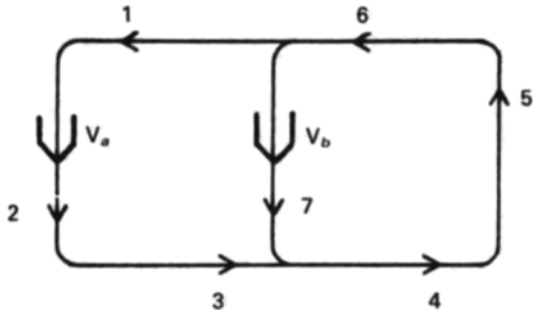

Fig. 4.

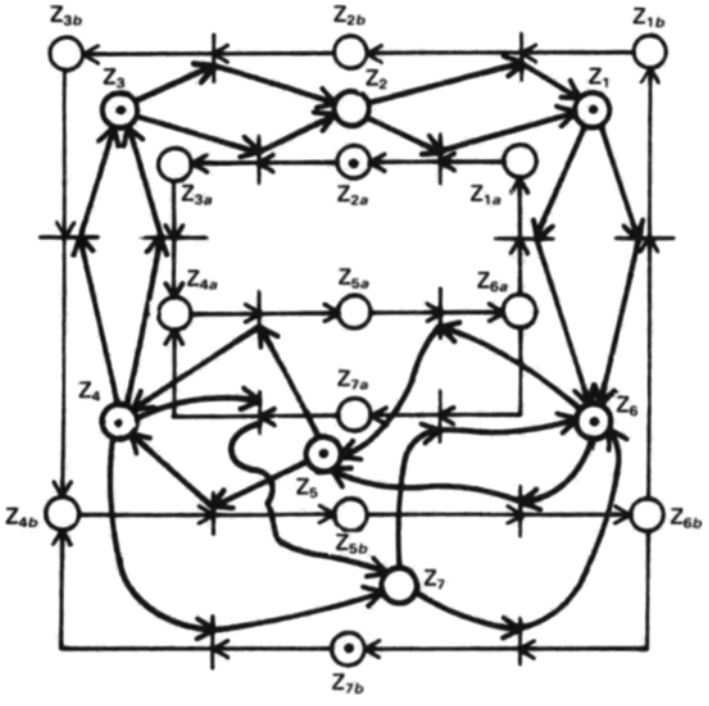

Fig. 5.

of those zones at this moment. Since the CPN structure is too complex to be built and analysed efficiently (the incidence matrix is huge), a condensed-zone CPN structure is again built, and is shown in Fig. 6. Comparing these two figures, it is found that the plain net structure will not be able to detect the possible collision when both vehicles are moving towards zone 4 at the same time. A more detailed picture will now be given. Suppose that vehicle V_a is assigned to move to zone 6 and vehicle V_b to zone 1. V_a is allowed to move into zone 3 when transition t_2 is fired. At this moment, V_a is in zone 3. Since zone 4 is still empty, according to Fig. 6, both vehicles V_a and V_b will be allowed into the same zone and will violate the zone-control regulations. Then

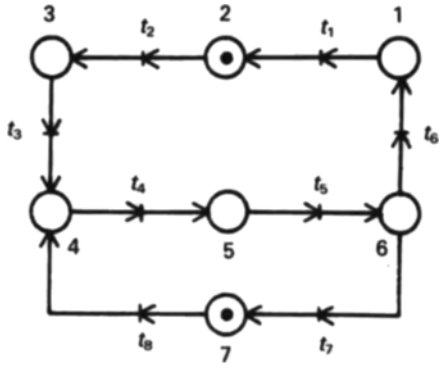

Fig. 6.

V_a and V_b may collide with each other. This situation comes from the fact that "time" is not incorporated into the model. Another control node, P_8 is now added to Fig. 6 to give the modified PTN model shown in Fig. 7. From Fig. 7, it can be seen that t_3 and t_8 are in conflict, i.e. that only one of them is allowed to be fired even though both of them are firable. The incidence matrix, N, of Fig. 7 is

$$
\begin{array}{c}
 \\
t_1 \\
t_2 \\
t_3 \\
t_4 \\
t_5 \\
t_6 \\
t_7 \\
t_8
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \end{array} \\
\left[\begin{array}{cccccccc}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & -1
\end{array}\right]
\end{array}
$$

and the S-invariant, I, is

$$I = (1, 1, 1, 1, 1, 1, 1, 0)$$

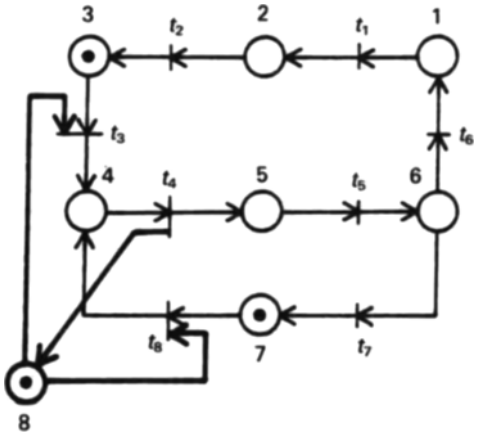

Fig. 7.

The S-invariants, together with the initial marking of the net, facilitate the identification of the zone-control operation rule by elementary deduction.

Merge Structure for a Mixed System

Fig. 8 shows the same layout as that in Fig. 4, except that zone 7 has become a bidirectional guide path. The possible confusing situations arise in zones 1, 4 and 7. After careful investigation, three more control nodes need to be added into the net structure. The MPN model is thus built and is shown in Fig. 9. The bold parts of the figure represent the added control nodes which restrict the number (only one) of vehicles in the zone.

The reason for the use of a bidirectional path is to improve the system performance. However, from Fig. 9 it can be seen that extra control nodes have to be considered in the model, i.e. it may complicate the AGVS and increase the once-off installation cost and the everyday control cost. If the traffic load in zone 7 is heavy and the workshop space is large enough, a two-way traffic path may be suggested (see Fig. 10). The corresponding PTN model is shown in Fig. 11, and is simpler than that in Fig. 9.

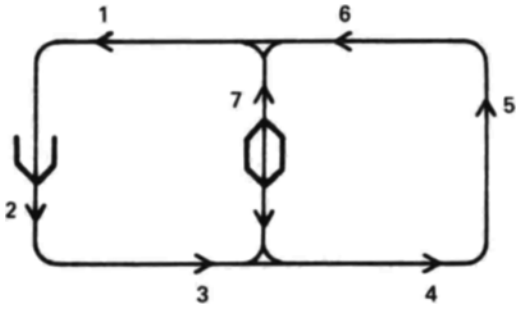

Fig. 8.

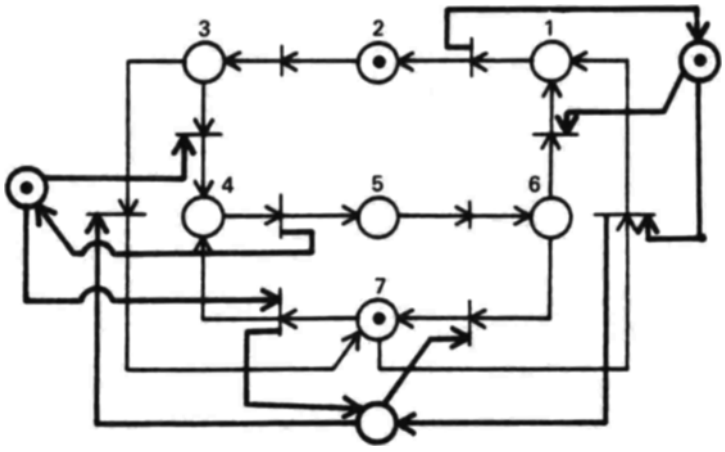

Fig. 9.

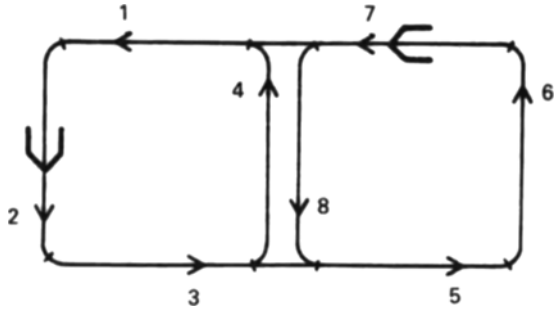

Fig. 10.

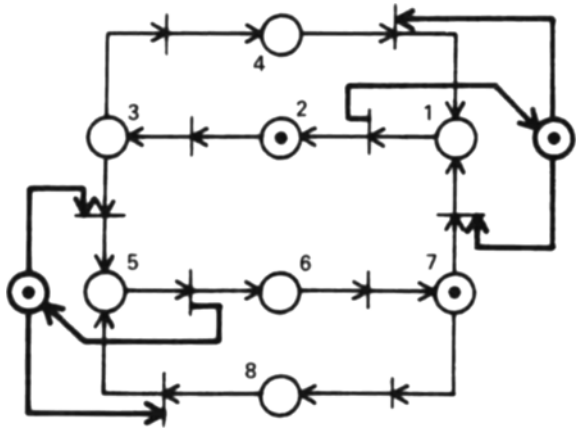

Fig. 11.

Merge Structure for a Bidirectional System

The same layout is again used, except that each segment has become bidirectional (see Fig. 12). The condensed plain PTN is shown in Fig. 13. Obviously, since the time factor has not been added into the net, this plain model will not be able to synthesise the zone-control function completely. A large number of additional control nodes must be included in the model to comply with the zone-control rules. Three kinds of traffic-control problems arise [16]:

1. *Merging*. Two vehicles on different paths move into the same zone in the same direction. This also happens in the unidirectional and the mixed-direction systems. Assuming that both the zone-1 vehicle and the zone-3 vehicle merge to zone 2, just as before, one additional control node, P_5, which can bring the time factor into the model, has to be included. The corresponding net structure is shown in Fig. 14. This situation can satisfactorily be resolved by defining a right of way of the vehicles (such as that the righthand vehicle has the right of way, or that the vehicle on the main road has the right of way).
2. *Blocking*. One vehicle is standing in front of another vehicle and blocks its way. The two vehicles may be moving in the same direction or in opposite

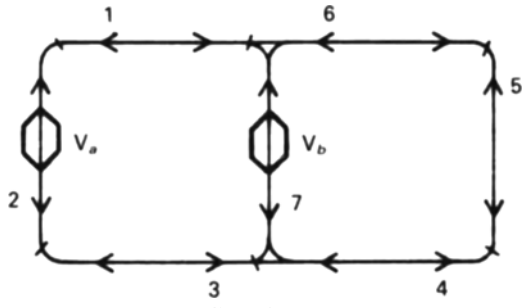

Fig. 12.

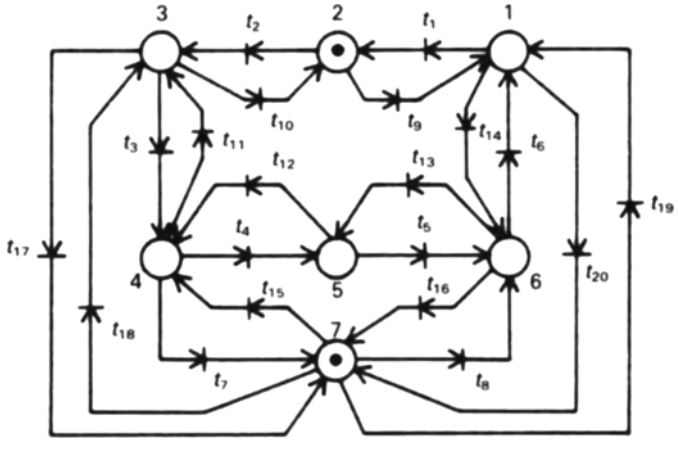

Fig. 13.

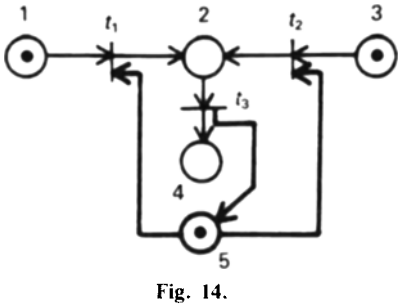

Fig. 14.

directions. The example shown in Fig. 15 is used to demonstrate one of the possible situations. The zone-2 vehicle wants to move to zone-3, and the zone-3 vehicle wants to move to zone 2. A new place, P_5, is added to control the availability of zone 2, and P_6 to control the availability of zone 3. Since both zones 2 and 3 are occupied, there are no tokens in places 5 and 6, i.e. both t_3 and t_4 can never be fired. In a robust AGVS, this situation should never occur if a proper vehicle-routeing algorithm is used.

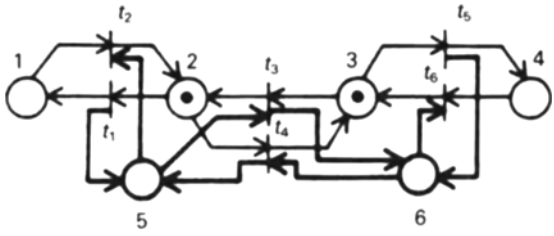

Fig. 15.

3. *Confusing*. Two vehicles travelling in opposite directions move into the same zone at the same moment. A special case is demonstrated in Fig. 16, which is described as follows: the zone-2 vehicle wants to move to zone 4 and the zone-4 vehicle wants to move to zone 1 at the same time. If t_8 fires first, the final state will be reached; if t_3 fires first, then a deadlock will arise, and the system becomes non-live. Problems such as this cannot be solved at the system-building stage. However, they can be prevented at the on-line control stage by adopting a robust vehicle-routeing algorithm.

Referring to Figs. 13 to 16, the complexity of the bidirectional traffic control is reflected. A complete PTN model for the system described in Fig. 12 can be structured by the composition of those figures. One can imagine the complexity of the PTN for the bidirectional system. The once-off installation cost could be many times larger than that of the unidirectional system, and the everyday on-line control cost could be unaffordable.

Intersection Structure for a Unidirectional System

Consider another possible structure of the AGVS – intersection, shown in Fig. 17. The plain PTN model is shown in Fig. 18. Since only one vehicle is allowed in each zone, additional zone-control nodes which indicate the availability of the zones have to be added to the model. The resulting model is shown in Fig. 19. Although this model does release the vehicle-merge problem, the vehicles may possibly collide with each other when one vehicle is moving from

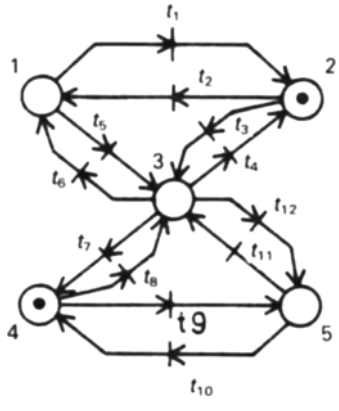

Fig. 16.

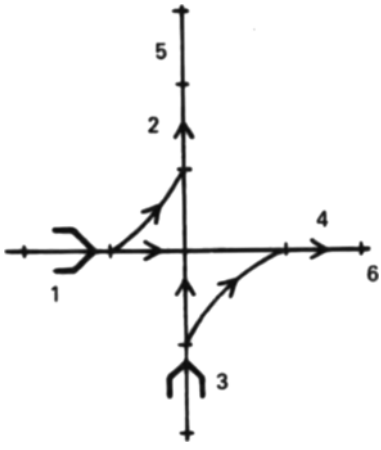

Fig. 17.

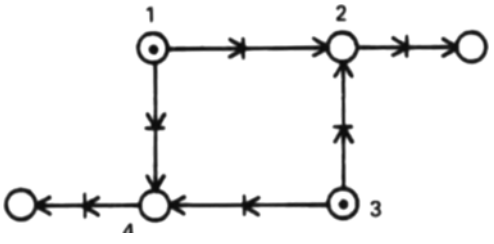

Fig. 18.

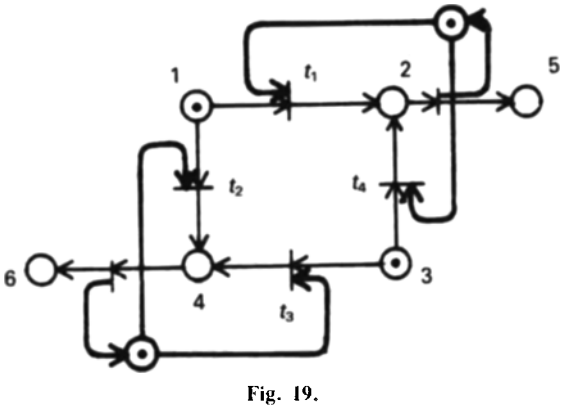

Fig. 19.

zone 3 to zone 2 and the other vehicle is moving from zone 1 to zone 4, since the firings of t_2 and t_4 are legal in the PTN rules. Hence, the PTN model in Fig. 19 cannot synthesise the complete dynamic behaviour of the zone-control function. Another MPN structure is shown in Fig. 20. One can find that a virtual place representing the intersection point has been added to the model,

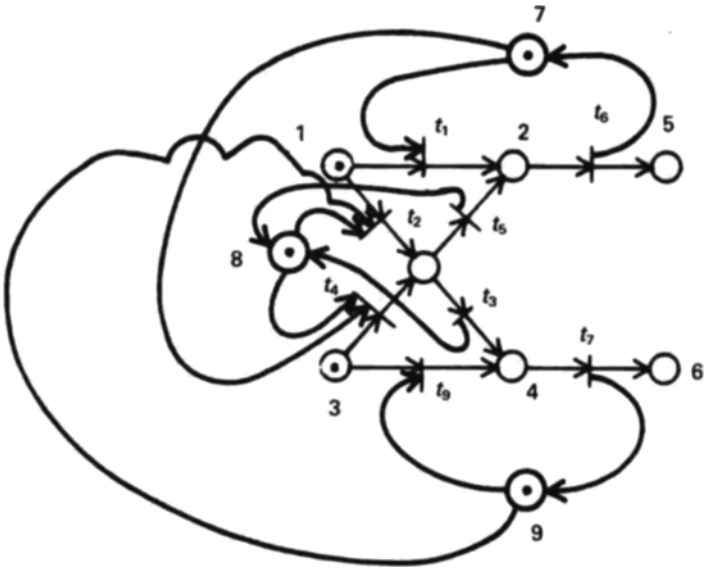

Fig. 20.

and a token in the virtual place indicates the right of way. Three possible cases will be used to validate the functionality of the model:

1. A vehicle is assigned to move from zone 1 to zone 2 (i.e. t_1 has to be fired), and another vehicle in zone 3 is assigned to move to zone 2 (i.e. t_4 and t_5 have to be fired in order). By playing the tokens, one can see that only one of t_1 and t_4 can be fired, since they share the same input place, P_7, in this model.
2. A vehicle is assigned to move from zone 1 to zone 4 (i.e. t_2 and t_3 have to be fired in order), and another vehicle is assigned to move from zone 3 to zone 4 (i.e. t_8 has to be fired). Since t_2 and t_8 share the same input place, P_9, only one of them can be fired.
3. A vehicle is assigned to move from zone 1 to zone 4 (i.e. to fire t_2 and then t_3), and another vehicle is assigned to move from zone 3 to zone 2 (i.e. to fire t_4 and then t_5). Since t_2 and t_4 share the same input place, P_8, t_4 is not firable if t_2 is fired, unless t_3 is fired after t_2. If t_4 is fired, t_2 is not firable unless t_5 is fired after t_4.

These three cases show that the PTN model in Fig. 20 does guarantee that the vehicles do not collide and that they comply with the zone-control regulations.

Intersection Structure for a Bidirectional System

Bidirectional guide paths are employed in the intersection part of the AGVS (see Fig. 21). The corresponding zone-control PTN structure is shown in Fig. 22. One can judge for oneself how practical it is to use such a structure in the real world.

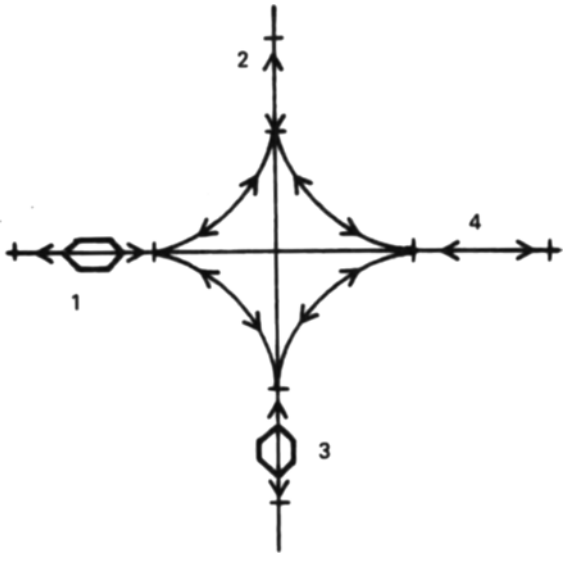

Fig. 21.

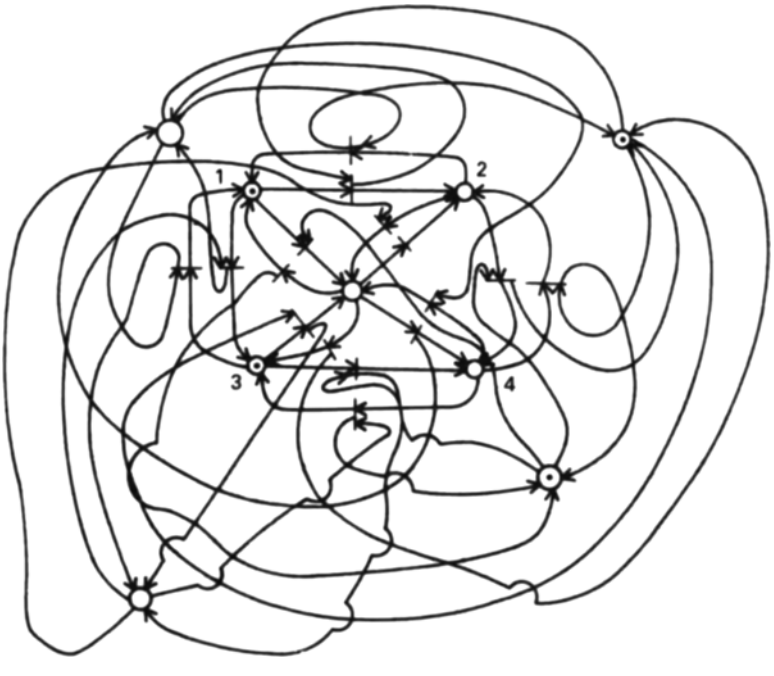

Fig. 22.

5. Conclusions

To model a manufacturing system, such as the AGVS, with regular tools is a difficult task. PTNs have been proven to be a powerful tool for modelling such a system. In this paper, emphasis has been placed on the development of basic traffic-control PTN structures. Based on these basic net structures, a huge and complex AGVS can thus be established with ease. In contrast to other researches, plain nets are suggested instead of CPNs for use in the early system-design stage. This is because:

1. The CPN size increases in proportion to the number of vehicles, which makes the system design more complex.
2. The incidence matrix and the S-invariants of the plain nets have the same qualitative properties as those of the CPNs.

Structures for merge and intersection in uni-, mixed, or bidirectional systems have been established, and synthesise the zone-control functions one by one. The results show that:

1. The unidirectional system is relatively simple; however, its performance may sometimes not be satisfactory to the users.
2. The bidirectional system is much more complex than the unidirectional one, and the use of the bidirectional system may be costly; hence, it is not practical to use the bidirectional system.
3. Owing to the above two conclusions, a mixed system could be a good solution for those who want to pay a reasonable price for a system with better performance.

Up to now, only the net structure of the AGV traffic control has been studied. The marking of the net should be incorporated into the net structure for the fulfilment of the zone-control function. An elegant and powerful graphical modelling tool for system design, analysis, simulation and on-line control would be the ultimate goal of this study.

References

1. G. A. Koff, "Automatic guided vehicle systems: applications, controls and planning", *Material Flow*, **4**(4), pp. 3–16, 1987.
2. P. J. Egbelu and J. M. A. Tanchoco, "Potentials for bi-directional guidepath for automated guided vehicle based systems", *International Journal of Production Research*, **24**(5), pp. 1075–1097, 1986.
3. P. J. Egbelu and J. M. A. Tanchoco, "Characterization of automatic guided vehicle dispatching rules", *International Journal of Production Research*, **22**(3), pp. 359–374, 1984.
4. G.-C. Vosniakos and B. J. Davies, "On the path layout and operation of an AGV system serving an FMS", *International Journal of Advanced Manufacturing Technology*, **4**(3), pp. 243–262, 1989.
5. W. L. Maxwell and J. A. Muckstadt, "Design of automatic guided vehicle systems", *IIE Transactions*, **14**(2), pp. 114–124, 1982.
6. L. C. Leung, S. K. Khator and D. L. Kimbler, "Assignment of AGVS with different vehicle types", *Material Flow*, **4**(4), pp. 65–72, 1987.

7. T. J. Hodgson, R. E. King and S. K. Monteith, "Developing control rules for an AGVS using Markov decision processes", *Material Flow*, **4**(4), pp. 85–96, 1987.
8. J. L. Peterson, *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
9. M. Kamath and N. Viswanadham, "Applications of Petri net based models in the modelling and analysis of flexible manufacturing systems", report no. CH2282-2/86/000/0312, IEEE, 1986.
10. J. Martinez, H. Alla and M. Silva, "Petri nets for the specification of FMSs", *Modelling and Design of Flexible Manufacturing Systems*, Elsevier Science Publishers BV, Amsterdam, pp. 389–406, 1986. Editor: Andrew Kusiak.
11. D. Crockett, A. Desrochers, F. DiCesare and T. Ward, "Implementation of a Petri net controller for a machining workstation", report no. CH2413-3/87/0000/1861, IEEE, 1987.
12. T. Murata, "State equation, controllability and maximal matchings of Petri nets", *IEEE Transactions on Automatic Control*, **AC-22**(3), June, 1977.
13. H. J. Genrich, "Predicate-transition nets", *Lecture Notes in Computer Science*, Springer-Verlag, **254**, pp. 207–247, September 1986. Editors: W. Brauer, W. Reisig, G. Rozenberg.
14. K. Jensen, "Colored Petri nets and the invariant method", *Theoretical Computer Science*, **14**(14), pp. 317–336, 1981.
15. K. Lautenbach, "Linear algebraic techniques for place/transition nets", *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 254, pp. 142–167, September 1986. Editors: W. Brauer, W. Reisig, G. Rozenberg.
16. W. Reisig, "A strong part of concurrency", *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 266, pp. 238–272, 1987. Editors: G. Rozenberg, 1987.

Correspondence and offprint requests to: Professor Suhma Hsieh, Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan 10764, Republic of China.