Design of Automated Guided Vehicle Systems Using Petri Net Models

by

Yogeswaran Theivanayagam

Thesis Submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE
in
INDUSTRIAL AND SYSTEMS ENGINEERING

Approved:

Dr. M. P. Deisenroth, Chairman

Dr. S. C. Sarin

Dr. J. E. Kobza

April, 1996

Blacksburg, Virginia

Design of Automatic Guided Vehicle Systems Using Petri Net Models

by

Yogeswaran Theivanayagam

Commitee Chairman : Dr. M. P. Deisenroth

Industrial and Systems Engineering

(ABSTRACT)

The analytical models for the design of automatic guided vehicle systems (AGVs) available in the literature are limited to the computations of the fleet size and the design of the guide path layout. Another major design factor which affects the performance of an AGV system is the locations and the dimensions of the control zones. The control zones facilitate congestion free flow of AGVs along the guide path. A proper design of the control zones is extremely important for the functioning of the AGV system. When the effects of the control zones are included in an analytical model, its solution procedure becomes complicated and difficult to be solved. This is due to the various interactions between the work stations and AGVs caused by the control systems. In this thesis, Petri net theory is introduced as an alternative method for the modeling of AGV systems. It is a graph theoretic tool which can easily capture synchronization and resource sharing situations which are common in AGV systems. Further, well developed mathematical procedures are available to analyze them. A systematic Petri net modeling procedure for AGV systems using sub nets is provided in this thesis. The Petri net model is used to compute the fleet size and the control zone dimensions of the AGV system.

ACKNOWLEDGMENTS

First, I thank Dr. M. P. Deisenroth for accepting me as his student and all the advice and guidance he provided during the course of this research. My thanks also to Dr. S. C. Sarin for his suggestions on the mathematical aspects of this thesis. I thank Dr. R. Reasor for serving as a committee member during the early part of this research. I thank Dr. Kobza for all the valuable corrections he made during a very short period of time. My thanks goes to my family members who helped me pursuing graduate studies at VPI&SU. Finally, I am greatly indebted to my wife Sahayam and my son Sanjeevan for their patience and encouragement during the period of this research.

DEDICATION

I dedicate this work to my mother in law, Juvanpillai Sebamalai Gabriel who passed away during the preparation of this thesis.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 BACKGROUND

Material handling today is an integral part of any manufacturing system. It is an important component which acts as a circulatory system in a plant, distributing vital material to all its departments. The increase in automation in the processing equipment has lead to the need for automation in material handling as well. Automatic guided vehicles (AGVs) have proved to be a globally accepted form of material handling automation during the recent years. Some of the many advantages of an AGV system include its ability to provide flexibility in the paths for material flow and the use of floor space, its ability to provide computer integration and control of the materials handling function and the extensive reduction in material handling related damage, injuries and equipment noise [24].

However, AGV systems involve high capital expenditure. Therefore, exhaustive planning is mandatory in the design stage of such systems. The major issues which have attracted the attention of many researchers include the estimation of the fleet size, the construction of the layout of the guide path and the design of control zones. Another area of interest is the routing and scheduling of the vehicles. Since the early 1980s, various studies have been conducted on the above issues and their effects on the performance of AGV systems. One of the pioneering efforts in representing an AGV system in the form of a mathematical model for the purpose of studying its design issues is due to the work of Maxwell and Muckstadt [36]. They cleverly formulated a transportation model which represents the movement of the AGVs between work stations. The objective of the model was to minimize the empty vehicle movement. They showed how the optimal solution of the transportation model is used to estimate the minimum fleet size to meet a given material flow requirement. Later, the 1987 issue of the well known journal titled, Material Flow contained several interesting articles related to the design of AGV systems. Notable contributions in this journal are the non simulation approaches of Egbelu [17], the queuing theory based *CAN_Q* method of Tanchoco, Egbelu and Tagaboni [52] and the mixed integer programming formulation of Leung, Khator and Kimbler [33]. These articles mainly focused on the estimation of the fleet size for an AGV system. Even though the above methods are approximate in nature based on several assumptions, the results obtained from them provide useful information for detailed simulation analysis.

However, an important factor which was not taken into consideration in these models is the zoning restrictions imposed in a realistic AGV system. The existing analytical models become complicated when the constraints related to zone control are included. One of the very few analytical models incorporating zone control in an AGV system is due to the research by Malmborg [34]. His method is based on the assumption that the vehicles arrive at each control zone following Poisson distributions. The special characteristics of Poisson distribution are used to develop the related Markov chains and thus the steady state probabilities of the system. The objective of his analytical model was to reduce the extent of simulation modeling necessary to design a zone control AGV system. The control zone model of Malmborg [34], provides the performance measures of an AGV system for a given set of variables such as guide path layout, fleet size, buffer capacity, vehicle dispatching rules and control zone parameters.

1.2 PROBLEM STATEMENT

This thesis introduces a relatively new and promising approach for the modeling and analysis of AGV systems with control zone restrictions. It views the AGV system as a network and identifies the applicability of graph theoretic methods in its modeling and analysis. However, a more versatile form of net theory than the classical node-arc networks is necessary to capture complex interactions between various elements of a realistic and dynamic AGV system. Thus, this thesis introduces an elegant net theoretic approach known as Petri net theory.

Petri net theory is a recent and promising methodology which can capture the complex interactions of an automated manufacturing system and be analyzed without simulation. The graphical nature allows easy visualization and the ability to represent complex interactions of concurrent systems have made Petri net theory to be increasingly recognized in the analysis of manufacturing systems. However, prior applications of Petri net theory in modeling AGV systems is very limited.

One of the earliest Petri net models used in the design and analysis of AGV systems was by Archetti and Scimachen [5] in 1989 and the most recent research is due to Wang and Hafeez [55] in 1994. During this five year period not more than ten research articles have been published in the area of the application of Petri nets in modeling AGV systems. Further, the major drawback in the existing Petri net models for AGV systems available in the literature is that they do not provide a systematic procedure of constructing them. A systematic procedure for constructing a Petri net model is vital to ensure that it guarantees the desired properties such as boundedness, liveness and reversibility. It is essential that a Petri net model representing a dynamic system should have these properties. A model which does not have these properties will not represent a system which functions properly. Another limitation in the existing models is that they do not take zone control into consideration. The only model which includes zoning restrictions appears in the research article by Dong Soon Yim and Linn [13]. However, the inclusion of zoning restriction in their model makes it complicated and difficult to be understood. Another major limitation in the existing Petri net models is that they are analyzed by simulation and not by using the mathematical background of Petri nets.

The purpose of this research is to overcome the above deficiencies. This research introduces a systematic methodology of constructing and analyzing Petri net models representing AGV systems. Unlike the existing models, the method introduced in this thesis uses a systematic top down approach and includes the zoning restrictions without much complication. This procedure is based on three sub nets representing three different functionality of an AGV system. The first sub net is known as the transportation sub net. The transportation sub net is used to model the movement of the AGVs between the work stations. The next sub net is the demand sub net and it models the demand for AGVs between the work stations. Finally, the control sub net is used to represent the restrictions imposed by the zone control. These sub nets provide a frame work for the systematic construction of the final Petri net.

Simulation is not used as a tool to evaluate the performance measures and the design parameters of the AGV system. Instead, the procedure introduced in this thesis, uses the well developed mathematical foundations of Petri nets. A deterministic timed Petri net model is used to model the AGV system and it is analyzed using the results of Sifakis [47]. The fleet size and the parameters of the control zones of the AGV system are computed using the analysis.

1.3 PRESENTATION METHODOLOGY

This thesis is divided into eight chapters. The first chapter is the introduction to the thesis. The second chapter is the literature review which is divided into three major sections. The first section focuses on the research related to AGV systems. The next section provides an extensive survey of the research articles on Petri net theory and its applications in manufacturing systems. The third section is on the application of Petri nets in modeling AGV systems.

The third chapter introduces the concepts of Petri net theory. In this chapter some of the definitions and the theoretical back ground of Petri nets which are useful in understanding this thesis are presented. The fourth chapter presents the methodology of modeling the AGV systems as sub nets. This chapter further shows the way of merging the sub nets in constructing the final Petri net. The following chapter is on the theoretical developments related to the three sub nets representing the components of an AGV system. These theoretical results are necessary to determine the fleet size and the design parameters of the control system.

Once the parameters of the control zones are computed, it is necessary a procedure is developed to show how those control zones are designed in a guide path layout. The sixth chapter explains the methodology of designing the control zones. In the seventh chapter, examples of a single and muliti loop AGV system are given to illustrate the procedure of computing the fleet size and the control zone parameters. This chapter also explains the method of designing the control zones once the dimensions are computed. The final chapter presents the contributions of this research and recommendations for future study in this area.

2. LITERATURE REVIEW

The literature survey comprises of three major sections. The first section focuses on the research papers related to the design issues and the routing and scheduling of AGVs. The second section is on Petri net theory. It identifies the contributions of various researchers in developing Petri nets as a powerful tool. It also focuses on its application in manufacturing systems. The final section introduces the few articles available in the literature on the modeling AGV systems using Petri nets.

2.1 RESEARCH PAPERS ON AGV SYSTEMS

The research papers in AGV system design examine issues such as the number and type of vehicles required and the design of guide path layout. The design issues of AGV systems were first addressed by Maxwell and Mucksdatt [36] in 1982. They formulated the movement of AGVs for a given material flow requirement as a transportation problem. The objective of the formulation was to minimize the empty vehicle travel time and thus minimize the vehicle requirement. They further proposed methods to minimize vehicle blocking and inventory blocking by proper assignment of drop off and pick up points to the segments of the guide path layout. Finally they presented a vehicle dispatching heuristic.

Egbelu [17] introduced four analytical methods to estimate the number of vehicles needed by an AGV system. The estimations were based on the loaded and unloaded travel times. He further adjusted the loaded and unloaded times for blocking, delays due to battery charging, etc. The results were tested using simulation under various dispatching rules. The choice of dispatching rules were found to have a definite effect on the requirement of the number of vehicles.

Shortest travel time dispatching rule tended to promote deadlock. However, if anti-locking features were incorporated STT/D rule provided a good estimate of the vehicle requirement.

Tanchoco, Egbelu and Tashabon [52] treated the AGV system as a queuing network. They used a queuing network analyzer called *CAN-Q* to determine the number of vehicles. The results were analyzed using simulation and was observed that *CAN-Q* underestimated the number of vehicles required. Sinriech and Tanchoco [49] developed a multi-criteria optimization model which combines the systems performance and costs related to that performance. This model was used to determine the number vehicles needed based on the combined measure. The throughput performance was measured by the number of completed parts and the total number of parts that arrived. They showed that by using a trade off ratio between the cost and throughput performance, several options can be considered for implementation.

The question of finding optimum guide path layout problem was first formulated as a mixed integer programming problem by Gaskins and Tanchoco [21] in 1987. Their formulation assumed unidirectional flow and that the vehicles travel along the shortest path between any two nodes. However, this formulation did not take into account of the travel of unloaded vehicles, vehicle blocking and congestion. Their approach is best suited for an environment in which the flow path changes over time. As the material flow between work stations vary, the formulation and solving the mixed integer programming problem will be faster than doing a new simulation study.

In 1989, Gaskins, Tanchoco and Tanghaboni [22] developed a similar integer programming formulation to determine the optimal flow paths for free ranging AGVs. The major weakness of the above formulation was the computational difficulty in solving the integer programming problem. Further, vehicle blocking and traffic congestion were not taken into consideration. Goetz and Egbelu [23] built on the work of Gaskins and Tanchoco [21] to model AGV system guide path layout formulation to include optimal locations for pick-up and drop-off points for inward and outward bound parts as well as for the selection of optimal guide paths. They introduced a heuristic algorithm to reduce the size of the problem by considering only the major flows from one department to another.

Egbelu and Tanchoco [18] addressed the possibility of designing an AGV system layout with bi-directional travel. They proposed different buffering areas for vehicles to avoid collision and congestion. Their simulation studies showed that installations requiring fewer vehicles have a potential for increased productivity through bi-directional travel path layout. Bozer and Srinivasan [8] proposed an alternative layout approach. Their tandem configuration composed of a set of interconnected loops, each loop having one vehicle. A buffer is used at the interconnection of the loops to transfer the loads from one loop to another. The advantage of this approach is the elimination of blocking and the ease of traffic control. The disadvantage of tandem configurations is that the throughput of the system cannot be increased by adding vehicles.

Malmborg [34] was the first to incorporate the effects of zone control in an analytical model. He developed a closed form analytical model which could predict the effects of AGV fleet size, the input and output buffer sizes and the control zones on the performance and the risks of deadlock in an AGV system.

Egbelu and Tanchoco [19] tested heuristic rules for dispatching AGVs in a job shop environment for a given guide path layout. They classified the dispatching rules as work center initiated rules and vehicles initiated rules. In Work center initiated rule a vehicle is selected from a set of idle vehicles and in a vehicle initiated rule a work center is selected from a set of work centers simultaneously requesting for vehicles. The work center dispatching rules used by them were random vehicle rule, farthest vehicle rule, nearest vehicle rule, least utilized vehicle rule and longest idle vehicle rule. The vehicle initiated rules were random work center rule, maximum outgoing queue size rule, minimum outgoing queue size rule, shortest travel time rule, longest travel time rule, longest travel time rule and the modified first come first serve rule.

They performed various simulation runs on the dispatching rules and observed that, there was no significant difference among the work center initiated rules in a busy job shop situation. Another observation was that, among vehicle initiated rules, the shortest travel time and longest travel time rules lead the system to deadlock in some instances. The explanation for the deadlock is that when these rules were applied, some work centers never qualified to be considered for assignment and resulted in build up of input and output queue. They further observed that, the rules associated with distance measures had draw backs if the appropriate layout conditions were not met.

Egbelu [16] in a separate study, classified the dispatching rules as source driven rules and demand driven rules. A source driven rule is a push based dispatching rule where a part is selected first and then the destination to which it should be removed is determined. Once the source and destination are determined, an AGV is selected to perform the selected load movement. In a demand driven or a pull based dispatching rule, a work station that should receive a part is first selected. Then a list of parts that can be moved to this work station from the output buffers of other work stations are created. Finally, a part selection rule is applied to this list to select the appropriate part to be transported.

2.2 RESEARCH RELATED TO PETRI NET THEORY

Good tutorials in Petri nets can be found in Agerwala [1], and Murata [38]. The text books by Peterson [42] and Reisig [46] are also important sources of information. Two books published recently by DiCesare et al. [12] and Derochers and Al-Jaar [11] provide excellent information on the application of Petri nets in manufacturing systems. The December 1994 issue of the journal titled IEEE Transactions on Industrial Electronics is a special issue on Petri nets. It contains many interesting articles on the application of Petri nets in manufacturing systems.

The original tool for conducting the properties of Petri nets was the reachability tree analysis. This was first studied by Karp and Miller [28]. Lautenbach [31] developed algebraic approaches in which place/transition invariants were used to study liveness, boundedness and reversibility of Petri nets. The structural properties of Petri nets using matrix algebra was developed by Sifakis [48].

Ramachandani [44] was the first to incorporate time into Petri nets. Appropriate incorporation of time in transitions or places result in a powerful tool to evaluate the quantitative performance measures of systems. Ramamoorthy and Ho [45] studied system performance measures by associating time to transitions. The study of Ramamoorthy and Ho was limited to decision free Petri nets. Sifakis [47] associated time to places and developed the conditions for a periodic functioning system, to perform at maximal rate. The work of Sifakis [47] was not confined to decision free Petri nets. It could be applied to situations where conflicting situations also arise.

The idea of associating exponential time duration was first explored by Molly [37]. He showed that the reachabilty tree of an exponentially distributed timed

Petri net can transformed into a continuos timed Markov chain. The major drawback of stochastic Petri nets proposed by Molly [37] is that the state space increases drastically with the increase in the number of places, transitions and tokens in a Petri net model. Therefore, the reachability tree and thus the Markov chain become very large and difficult to be analyzed.

Marson, Balbo and Conte [35] proposed Generalized timed Petri nets which facilitated the reduction of the size of the reachability tree to a great extent. Generalized stochastic Petri nets are obtained by allowing for transitions which can fire instantaneously and transitions with exponential firing rates. Transitions which can fire instantaneously are given priority over exponentially timed transitions. They further introduced inhibitor arcs which eliminated the usage of excessive places in a Petri net model. By doing so, the size of the reachability graph of the Petri net is reduced. Reduction in the size of the reachability tree results in the reduction of the number of states associated to the Markov chain model.

Another limitation in stochastic Petri nets is that the analytical procedure is applicable to transitions with exponential firing times and instantaneous transitions which fire immediately. However, in many systems, the firing duration can be deterministic or of non exponential distributions. Ajmone and Choila [3] developed a class of stochastic Petri nets known as deterministic and stochastic Petri nets. This form of Petri nets can accommodate exponential and deterministic transitions with the limitation that only one deterministic transition can be enabled at any time. Deterministic and stochastic Petri nets have been used in the modeling of traffic control systems. In a traffic control system vehicles arrive at traffic lights following an exponentially distributed inter arrival time and time taken for each vehicle to clear the traffic lights is deterministic. Further only one set of vehicles is allowed to clear the traffic lights.

Dugan, Trivedi, Geist and Nicola [15] presented class of Petri nets known as extended stochastic Petri nets. Extended stochastic Petri nets allow for exponentially, deterministic, immediate and non exponential transitions. They derived the conditions under which Markov chains can be used and the conditions under which one has to resort to simulation.

Watson and Desrochers [56] presented sub nets known as throughput subnets to represent non exponential transitions in a Petri net. For example, they showed how a transition with Erlangian distribution can be represented by a series of transitions with exponentially distributed firing times. Another non exponential timed transition presented by them was the hyper exponential transition which is modeled using a subnet with two transitions in parallel and a probability switch.

Guo, DiCesare, and Zhou [25] proposed the application of moment generating functions for the analysis of extended stochastic Petri nets. They defined transfer functions for the transitions with non exponential times based on their moment generating functions. They used the transfer functions to evaluate the system performance. They further introduced three fundamental structures which can be reduced to a single transition.

Colored Petri nets was proposed by Jensen [27]. In colored Petri nets, each transition and place is associated with a set of colors. A transition fires when the right combination of colored tokens is present. When a transition fires in a colored Petri net, the tokens are removed from the input places and added to the output places according to a functional dependency of the transition firing and the color of the tokens involved. Colored Petri nets allow modeling of a system in a concise manner. Invariant analysis of colored Petri nets was studied by Narahari and Viswanadham [40].

2.3 REVIEW OF PETRI NETS IN MANUFACTURING SYSTEMS

A good tutorial on the application of Petri nets in manufacturing systems appears in the paper by Zurawski and Zhou [62]. The first part of this paper is on the concepts of Petri nets and its applicability to manufacturing systems. The second part of the paper focuses on the performance evaluation of manufacturing systems using deterministic and stochastic Petri nets. The paper concludes with the emphasis on the need for more powerful Petri net tools such as temporal Petri nets and fuzzy Petri nets.

Modeling of manufacturing systems using Petri nets began with the work of Dubois and Stecke [14] in 1983. They showed how the processing times and setup times of a flexible manufacturing systems with parts flowing along fixed routes can be modeled using deterministic timed Petri nets. They used a Petri net based simulation procedure to identify bottleneck stations and the minimum possible cycle time of the FMS.

Agerwala and Choed-Amphai [2] proposed a synthesis technique to construct Petri nets to model concurrent systems. Similar work was performed by Narahari and Viswanadham [39] to model and analyze flexible manufacturing systems. In their approach, a sub net was constructed for every basic operation for a product. They constructed the final net by combining the sub nets along common places. The limitation in the above approaches is that the liveness of the final net cannot be guaranteed.

Krogh and Beck [30] introduced a different bottom-up synthesis technique which guaranteed a live and safe final Petri net. They modeled the activities of each resource as an elementary circuit and combined these circuits along common paths. They defined these common paths as solitary transition-transition path and solitary place-place path. In a solitary transition-transition path, the first and the last nodes are transitions. In a solitary place-place path, the first and the last nodes are places.

Valette [53] was the first to study top-down synthesis methods for Petri nets. In the top-down approach, an aggregate model of the system is developed. Low level details are ignored in the aggregate model. Then more details are included step by step. He defined well formed blocks which can substitute transitions to add more detail. Suzuki and Murata [51] generalized the concept of well formed blocks and presented ways in which individual transitions and places can be replaced with those blocks to add more detail to the aggregate Petri net.

Zhou, Dicesare and Desrochers [58] made use of the results of Suzuki and Murata [51] to define special blocks called sequential Petri net blocks, parallel Petri nets, conflict Petri nets and decision free Petri nets. These blocks are used to replace places in an aggregate Petri net to add more detail. In a separate study Zhou and Dicesare [59] introduced the concepts of parallel and sequential mutual exclusion, which are useful in modeling shared resources in manufacturing systems. The concepts of special blocks, parallel mutual exclusion and sequential mutual exclusion are useful in modeling manufacturing systems. The combination of the concepts of special blocks and the concepts of parallel/sequential mutual exclusion along with the top down and bottom up approaches form a powerful tool known as the hybrid methodology for the synthesis of Petri net models. The hybrid methodology was presented in a separate paper by Zhou, DiCesare, and Desrochers [60].

Zurawski [63] presented a different approach in modeling manufacturing systems using Petri nets. The procedure introduced in this paper focuses on the representation of the functional abstractions of the various components of the manufacturing systems using sub nets. The functional abstractions represent the external behavior of the components. Using this class of Petri nets four typical components of an AGV based FMS system re presented.. The four components were the machining, load and unload stations, and the buffer of empty carts. The advantage of such sub nets is that the validation of the final Petri net is computationally easy.

Hillion and Proth [26] showed an interesting way of modeling a job shop system functioning periodically. They represented the flow of jobs through the different machines in the job shops by circuits known as " processing circuits". The order in which each machine processed the jobs was represented by another set of circuits known as "command circuits". The complete net of the job shop system was a special class of Petri net known marked graphs. The conditions for optimum functioning was characterized by the full utilization of the bottle neck stations with minimum work in progress. They formulated the above condition as an integer programming problem and introduced a heuristic approach to obtain a good solution.

Bruno and Biglia [9] were among the first to use GSPN for evaluating performance measures of manufacturing systems. They addressed the problem of tool handling in a flexible manufacturing system where tools are gathered in a common warehouse and are carried by means of a limited number of conveyors. They analyzed the optimum balance between the number of conveyors and machine utilization. Balbo, Choila, Franceschinis and Roet [6] compared the application of generalized stochastic Petri nets and queuing networks in modeling and performance evaluation of automated manufacturing systems. They observed that GSPN models were much easier to analyze systems which represent synchronization and conflicting situations. They showed how a flexible manufacturing system with different scheduling policies and pallet mixes can be modeled as a generalized stochastic Petri net. However, the draw back in the GSPN models was the growth of the reachability tree which makes the solution procedure difficult. They further pointed out that the GSPN model can be used to develop the simulation code for detailed analysis. Al-Jaar and Desrochers [4] used generalized stochastic Petri net modules to build models to represent manufacturing systems. They analyzed the performance measures such as machine utilization, average production rate, etc. for transfer lines with unreliable machines and buffers.

Another useful application of Petri net theory is the implementation of deadlock controllers in flexible manufacturing systems. Narahari, Viswanadham, and Johnson [41] presented a look ahead policy to identify the possibility of future deadlocks. The implementation of deadlock avoidance relies on a Petri net based on-line monitoring and control system. In a similar study, Kasturia, Dicesare and Desrochers [29] showed how a work cell controller can be implemented using a colored Petri net model. Banaszak and Krogh [7] presented a resource allocation policy based on a deadlock avoidance algorithm for flexible manufacturing systems with concurrently competing for resources. Their deadlock avoidance algorithm is developed using Petri net models for the concurrent flow of jobs. Venkatesh, Zhou, and Caudill [54] presented a comparison between the ladder logic diagram and Petri net representations for sequence controller designs in manufacturing systems. They presented the equivalent Petri net representations of ladder logic diagrams for different manufacturing sequences. They proposed a new class of Petri nets named real time Petri nets for the above comparisons. The conclusions in this study were that the real time Petri net representation can be easily understood when compared to the ladder logic diagrams and the control logic can be checked for deadlocks and re initializability. Further, the number of basic elements in a real time Petri net representation, was significantly smaller when compared to the basic elements needed in the equivalent ladder logic diagram.

2.4 PETRI NET MODELS FOR AGV SYSTEMS

One of the earliest Petri net models for AGV systems was developed by Archetti and Sciomachen [5] in 1989. They modeled an AGV system with three machines with input and output buffers for each machine and wait stations for the AGVs. Petri net modules were constructed for the part flow and the routing of AGVs. The modules were then merged to form the final Petri net. They used invariant analysis to show that the final Petri net had the desired properties. Petri net based simulation runs were used by them to evaluate the performance measures of the AGV system. The effect of the fleet size and buffer capacities on the throughput, machine utilization and the vehicle utilization were studied using the results from the simulation runs.

Don Soon Yim and Linn [13] developed a Petri net to compare the performance measures of an FMS system under the push and pull rules proposed by Egbelu [16] for dispatching automated guided vehicles. This is the only model which incorporates zone control in the system. However, the final Petri net appears to be complicated and difficult to understand. A Petri net based simulation program was used to compare the performance measures of the AGV system under the two different dispatching rules. A more recent study related to control zones of AGV systems using Petri nets is by Cui and Wang[10]. This paper deals with a formal design method for the Petri net controllers for AGV systems with shared paths.

Wang and Hafeez [55] compared the performance measures of a conventional guide path layout with a tandem configuration by modeling them as generalized stochastic Petri nets. An FMS system with three machines and three buffers was used in their analysis. A stochastic Petri net software called SPNP was used to evaluate the throughput, machine utilization and AGV utilization of the two configurations. Their results showed that the tandem configuration resulted in higher throughput. No significant difference was observed in the utilization of the AGVs.

Zurawski and Dillon [61] presented a modular approach to model an AGV based manufacturing system using Petri nets. These modules were used to model machining stations, unload stations and loading stations. Invariant analysis was used to confirm the validity of the modules. The final Petri net model is used as a control module for simulation analysis. Zeng, Wang and Jin [57] developed a Petri net model to represent an AGV system using a bi-directional travel guide path layout. A colored Petri net model was used by them to identify potential conflicts in the routing of the vehicles. The simulation program based on the Petri net model was used to identify the conflicts in advance.

More recent research articles on the application of Petri nets in AGV systems focus on the scheduling of the vehicles in the flexible manufacturing environment. Lee and DiCesare [32] presented a global search procedure using the reachability graph of the Petri net representation of the manufacturing system. A Similar procedure is presented by Sun, Cheng, and Fu [50]. However, these procedures are applicable to very small systems where the Petri net representation is simple When the Petri net representation becomes larger, the reachability tree grows steadily and the global search becomes almost impossible.

3. CONCEPTS OF PETRI NETS

Petri net theory was first introduced by Carl Adam Petri [43] in 1965 as a modeling tool for the analysis of communication systems. Since then, this elegant net theoretic tool has proven to be extremely useful in modeling and analyzing systems exhibiting characteristics such as concurrency, synchronization and mutual exclusion. Automated manufacturing systems is a recent and promising area of application of Petri net theory. This chapter focuses on the definitions, basic concepts and the applications of Petri net theory in a manufacturing environment.

A Petri net consists of two major components. The first is the structure of the net and the second is the marking. The structure of a Petri net models the static part of a system and the markings describe its dynamic behavior. The structure of the Petri net is represented by a directed bi partite graph. It consists of two sets of nodes known as places and transitions.

The places are indicated by circles and the transitions by horizontal or vertical bars. The input and output relationships between the places and transitions are described by directed arcs. A Petri net with a marking is known as a marked Petri net. The marking of a Petri net is described by the tokens residing in its places. The tokens are represented graphically by dots. The initial placement of these dots in the places of a Petri net is known as its initial marking. The initial marking represents the initial state of the system and the movement of the tokens between the places describe its dynamic behavior.

3.1 DEFINITION OF A MARKED PETRI NET

A marked Petri net is represented by a five tuple as: $PN=\{ P, T, IN, OUT, M_0 \}$.

Where :

$P=\{p_1, p_2,..., p_m\}$	is a finite set of places of cardinality m.
$T=\{t_1, t_2,..., t_n\}$	is a finite set of transitions of cardinality n.
$IN: PXT \rightarrow \{0,1,2...\}$	is an input function.
$OUT: TXP \rightarrow \{0,1,2...\}$	is an output function.
$M_0: P \rightarrow \{0,1,2...\}$	is the initial marking.

The input and output functions are written as mxn matrices and the marking as a $mx1$ vector. The non negative integer numbers associated to the arcs are known as weights. When there is no arc between a place and a transition, the weight of the input arc from that place to the transition is zero. A weight of more than one associated to an input or output arc indicates that there are multiple arcs between that place and the transition. The representation of these multiple arcs in a Petri net are avoided by indicating that weight on the arc. The input and output functions of the Petri net is generally combined and expressed as a single matrix known as the incidence matrix. The incidence matrix A_{mxn} is defined as $A_{mxn}=OUT_{mxn}-IN_{mxn}$. Figures 3.1 and 3.2 illustrate a simple unmarked and marked Petri net respectively. The places, transitions, input/output functions, initial marking and the incidence matrix are given below.

$P=\{p_1,p_2,p_3,p_4\}$ $\quad$ $T=\{t_1,t_2,t_3\}$

$$IN=\begin{pmatrix}1&0&0\\1&0&0\\0&0&1\\0&1&0\end{pmatrix}, OUT=\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\\1&0&0\end{pmatrix}, M_0=\begin{pmatrix}1\\1\\0\\0\end{pmatrix}, \text{ and } A=\begin{pmatrix}-1&0&1\\-1&1&0\\1&0&-1\\1&-1&0\end{pmatrix}.$$

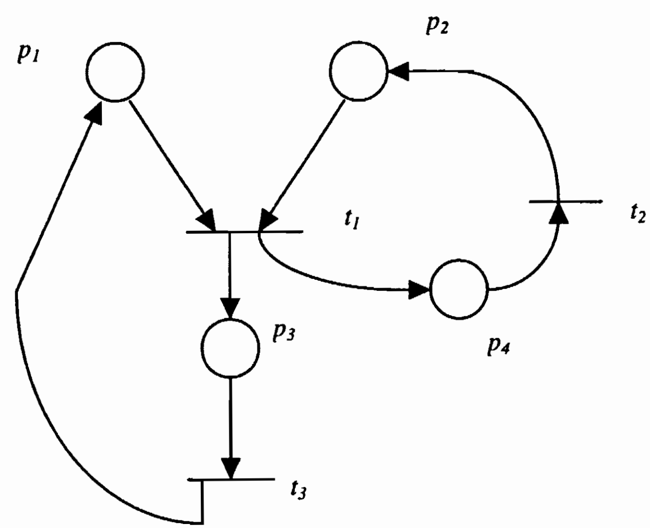


Figure 3.1 Illustration of an unmarked Petri net

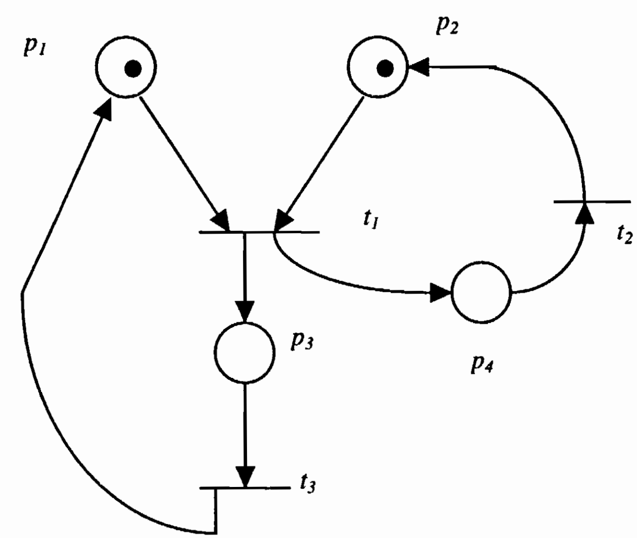


Figure 3.2 Illustration of a marked Petri net

3.2 EXECUTION OF A PETRI NET

As described in the previous section, the structure of a Petri net represents the static part of a system and the marking represents its dynamic state. The initial distribution of tokens in the places of a Petri net is known as its initial marking. The initial marking indicates the initial state of a system. It is possible to move from the initial marking to other markings. The process of moving from one marking to another marking is known as the execution of the Petri net. The execution of the net is governed by rules known as the firing rules.

A transition is said to be enabled if all of its input places are marked with at least as many tokens equal to the weight of the arcs joining them. An enabled transition can fire. When an enabled transition fires, tokens are removed from its input places and deposited in its output places. The number of tokens removed and deposited in those places is equal to the weights of the arcs joining them with that particular transition. A concise form of the firing rules is given below.

Rule 1 : A transition $t_j \in T$ is said to be enabled in a particular marking M, if each input place $p_i \in P$ of that transition t_j is marked with at least $IN(p_i, t_j)$ tokens.

Rule 2 : When an enabled transition $t_j \in T$ fires, $IN(p_i, t_j)$ tokens are removed from each input places $p_i \in P$ of t_j and $OUT(p_k, t_j)$ tokens are added to each output place $p_k \in P$ of t_j.

Figures 3.3 and 3.4 demonstrate the execution of a simple Petri net. The initial marking is shown in Figure 3.3. In the initial marking, transition t_1 is enabled. When transition t_1 fires, the new marking results as shown in Figure 3.4.

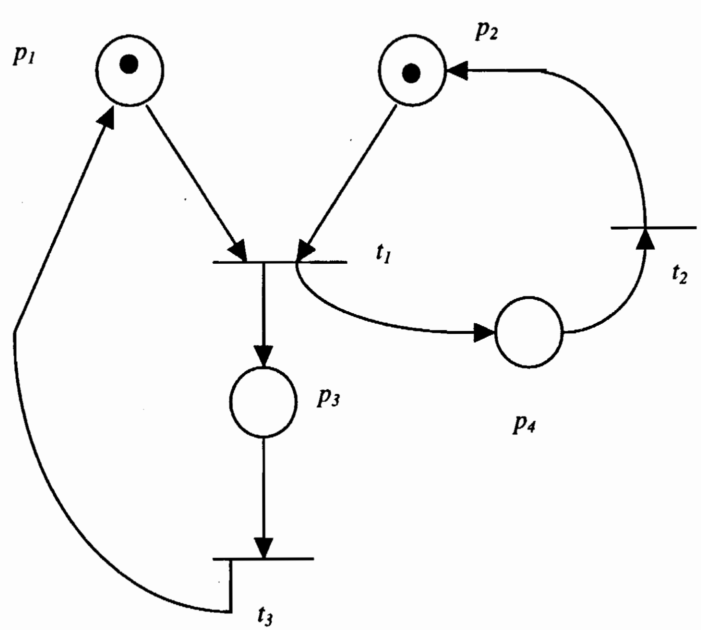


Figure 3.3 The initial marking of the Petri net

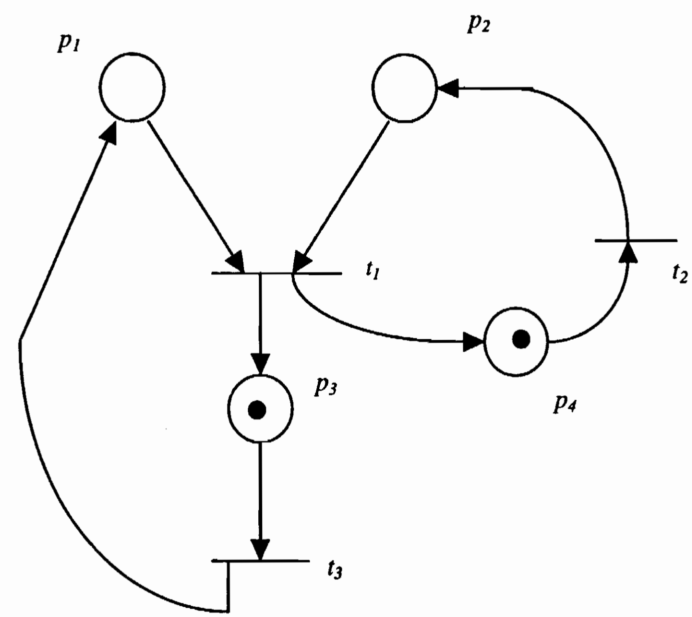


Figure 3.4 The marking of the Petri net after transition t_1 fires

3.3 REACHABILITY

Reachability is a basis for studying the dynamic properties of a system using Petri nets. It is the ability of a Petri net to reach different markings from its initial marking. This section provides the concepts of reachability and its usefulness in the analysis of Petri nets.

A marking M_i is said to be reachable from the initial marking M_0, if there exists a series of transitions $\sigma_j = t_{j1}\ t_{j2}....t_{jr}$, such that the firing of this series of transitions transforms the marking of the Petri net from M_0 to M_i. Where, t_{jk} represents the k^{th} transition fired in firing sequence j. The relationship between M_0, M_i, and σ_j is written as $M_0[\sigma_j{>}M_i$. Figure 3.5 illustrates the different firing sequences and the corresponding markings that can be reached from the initial marking of the Petri net illustrated in Figure 3.3. A set of all possible markings reachable from an initial marking is known as its reachability set. The reachability set of a Petri net depends on its initial marking and is denoted as $R(M_0)$. The graphical representation of a reachability set is known as the reachability graph or reachability tree. In a reachability graph, the nodes represent the markings and the arc between two nodes represent the firing of the transition which causes the transformation of the markings. For example, if $M_i[t_k{>}M_j$, then an arc representing t_k is drawn from node M_i to node M_j. The reachability tree of the example Petri net is illustrated in Figure 3.6. Reachability analysis is mainly used for the performance evaluation of stochastic Petri nets with exponential firing times [4], [6] and [35]. The underlying Markov chain derived from the reachability tree is used to compute the steady state probabilities of the system. The major disadvantage of reachability analysis is that the number of nodes of the reachability tree can become very large and cause difficulties in the computations.

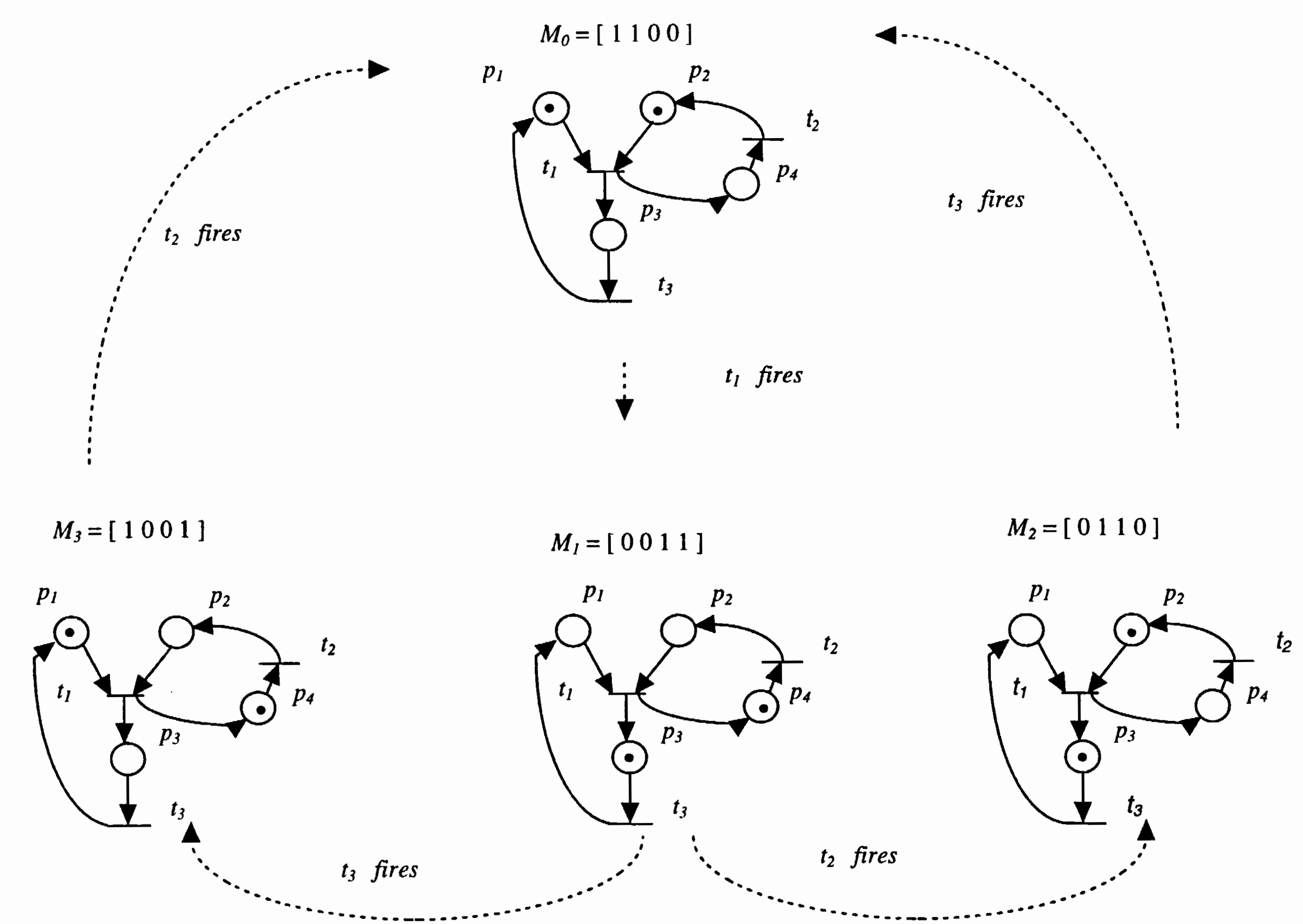


Figure 3.5 The different marking reachable from the original marking of the example Petri net

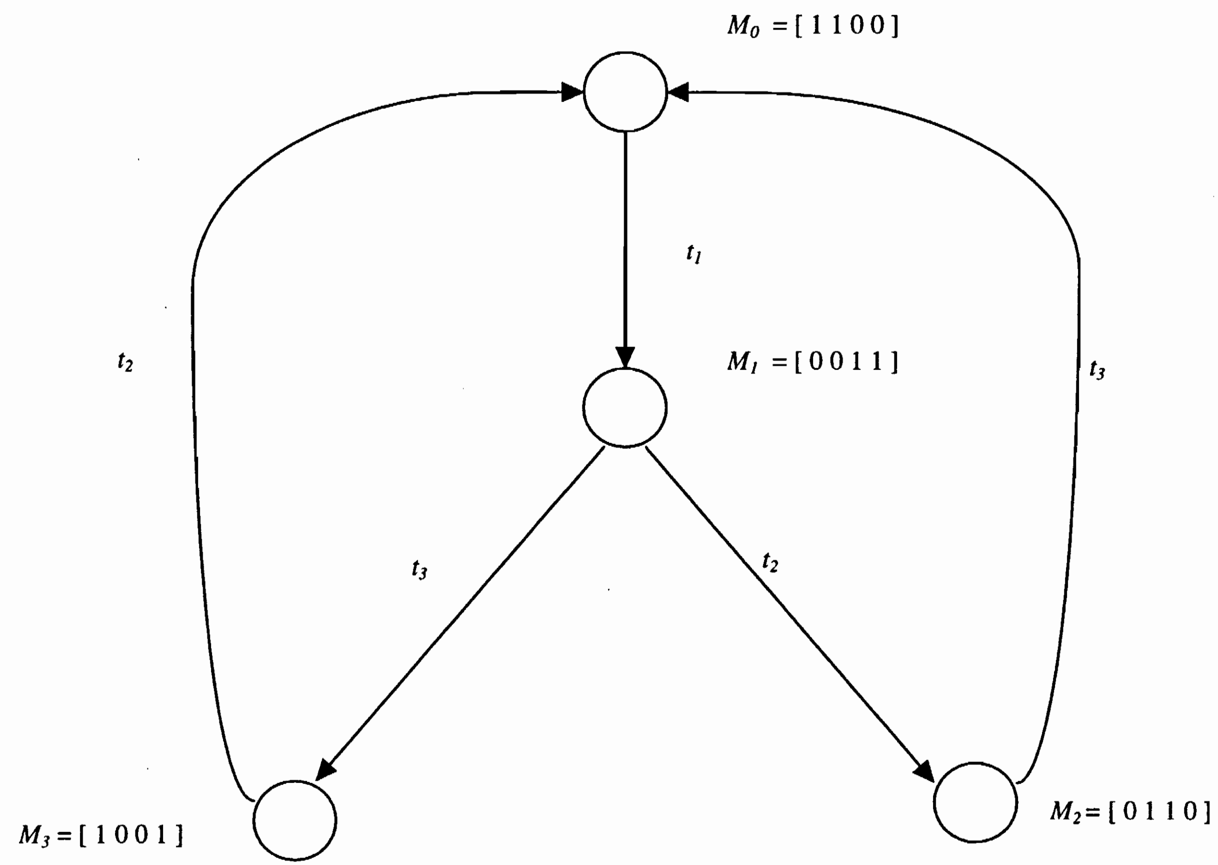


Figure 3.6 The reachability tree of the example Petri net

3.4 MATRIX ANALYSIS OF PETRI NETS

Matrix analysis is another important tool in Petri net theory. It is mainly used in analyzing the structural properties of Petri nets [48]. Another important application of matrix analysis is the performance evaluation of deterministic timed Petri net models [47]. Matrix analysis is based on the relationship between the markings of a Petri net, prior and after a series of firings of transitions. This relationship is written in terms of the incidence matrix of the Petri net and is known as the state equation. If there exists a transition $t_j \in T$ such that, $M_k[t_j>M_{k+1}$, the relationship between M_{k+1} and M_k is written as $M_{k+1}=M_k+A.U_k$. Where, A is the incidence matrix of the Petri net and U_k is a $(nx1)$ column vector U_k is known as the firing vector and $U_k(x)=1$ for $x=j$ and 0 otherwise. This expression can be extended to a sequence of firing of transitions as follows: If there exists a firing sequence such that $M_0[\sigma>M_k$, then the relationship between M_k and M_0 is $M_k=M_0+A.U$. $U=\sum_{i=1}^{k} U_i$ is known as the firing count vector. The element $U(j)$ of U represents the number of times transition t_j fires in the firing sequence σ. As an example, let us consider the simple Petri net shown in Figure 3.7. The incidence matrix and the initial marking and a marking reachable through the firing sequence σ are given below.

$$\mathrm{A} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}, \text{ and} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \text{ For } \sigma=t_1t_2,\ U=\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

The relationship between M_2 and M_0 is written as follows: $M_2=M_0+A.U$.

$$\text{i.e. } M_2=M_0+A.U=\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}+\begin{pmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}=\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

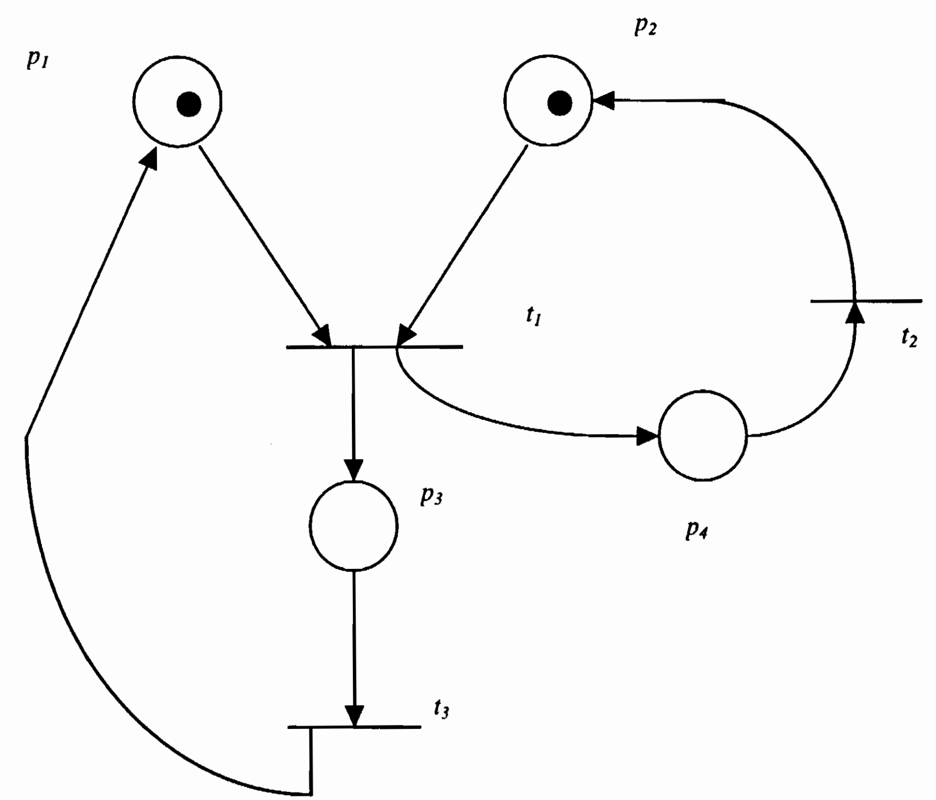


Figure 3.7 A simple Petri net example to illustrate the matrix representation

3.5 PETRI NETS IN A MANUFACTURING ENVIRONMENT

In order to represent a manufacturing system using Petri net models the places, the transitions and the tokens must be assigned proper meanings. Places represent resources such as machines, transportation equipment, or buffers. A place can also represent an operation such as machining or the transportation of a part. The existence of a token in a place represents the availability of a resource or an operation in progress. The firing of a transition represents the beginning or the end of an operation. It can also represent the beginning or the end of the availability of a resource.

Figure 3.8 provides a simple example where an AGV is used to transport parts from a storage area. Places p_1 and p_2 represent the availability of an AGV and a part for pick up respectively. Transitions t_1 and t_3 represent the beginning and the end of the transportation of a part. Place p_3 models the part being transported by the AGV and the return of the empty AGVs. Place p_4 represents the unavailability of a part after being picked up by the AGV. The enabling and firing of each transition are explained as follows: Initially, places p_1 and p_2 are marked with tokens. This indicates that the AGV and a part are available. Therefore, the input places of t_1 (i.e. p_1 and p_2) are marked with tokens and it can fire. When transition t_1 fires, the tokens in places p_1 and p_2 are removed and tokens will be deposited in places p_3 and p_4. This means that the AGV is being used and the part is removed from the pick up point. Now, at this point both transitions t_2 and t_3 are enabled. Firing of transition t_3 before transition t_2 means that the AGV has completed transporting the part before a new part arrived at the pick up point. Similarly, the firing of transition t_2 before transition t_3 indicates the arrival of a part before the completion of transportation.

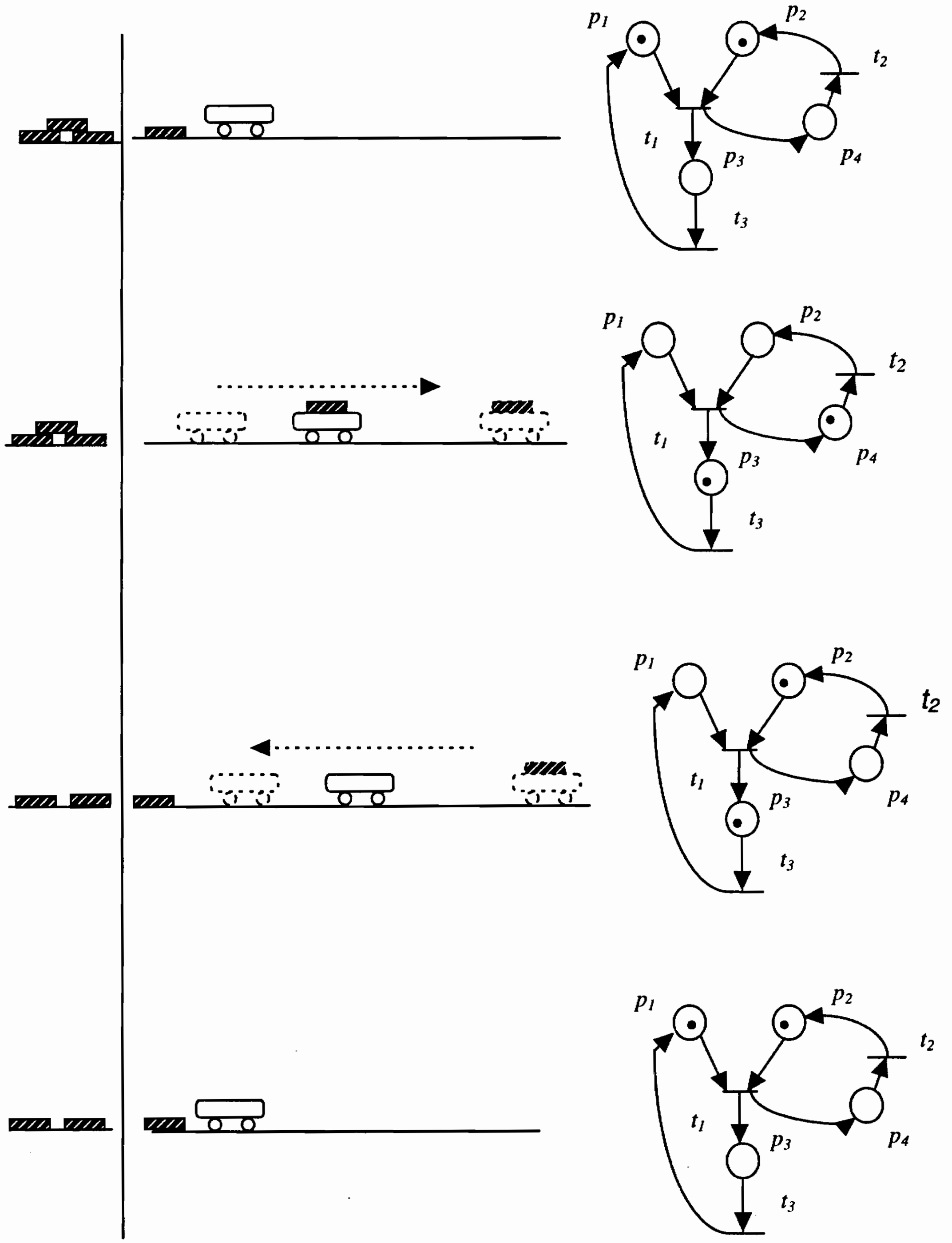


Figure 3.8 Petri net representation of a simple AGV based transportation system

3.6 IMPORTANT PROPERTIES OF PETRI NETS

One of the benefits of Petri nets is the ability to analyze them for properties related to manufacturing systems. Some of the important properties of Petri nets are boundedness, liveness, reversibility, conservativeness and consistency. These properties have important relevance to the behavior of a manufacturing system. This section describes the above mentioned properties and shows the relevance they have in the functioning of manufacturing systems.

3.6.1 BOUNDEDNESS

Boundedness is a property which guarantees the constraints on the number of each type of resource in a manufacturing system is satisfied. For example the number of space available in a buffer may be finitely limited. Therefore, the number of tokens residing in the place modeling the buffer cannot be more than that finite number. A Petri net is said to be bounded if the number of tokens in any of its places is less than a finite number for any marking reachable from the initial marking. If this number is k, then the Petri net is said to be k-bounded. Safeness is a special case of k boundedness where the value of k is 1.

Figure 3.9 illustrates a Petri net and its reachability graph which is not bounded. The number of tokens in places p_2 and p_4 increase steadily, with continious firing of transitions t_1, t_2, t_3 and t_4. The reachability graph of an unbounded Petri net always tend to grow and the number of nodes increases steadily towards infinity. A modification of the reachability graph that can handle unbounded Petri net models is known as the coverability graph. In a coverability graph, the labeling scheme of the nodes are modified such that the unbounded place is identified and no further branching is performed from that place.

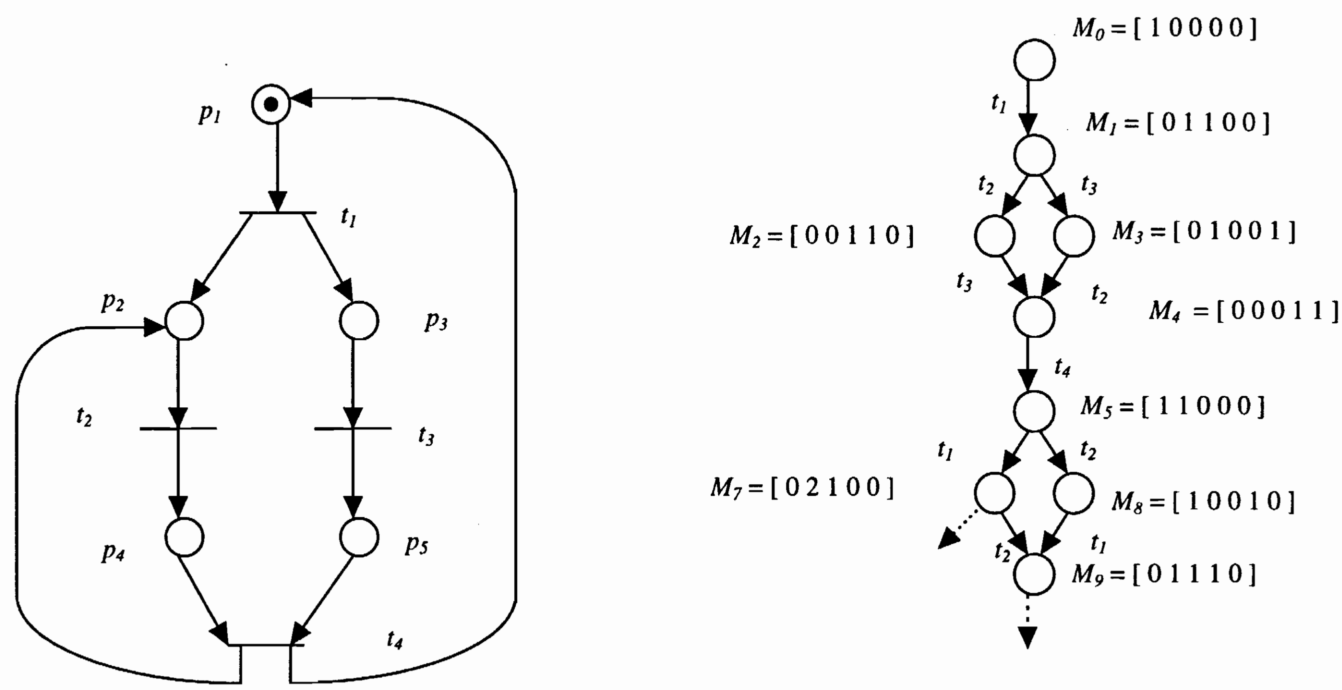


Figure 3.9 A Petri net which is not bounded and a portion of its reachability tree

3.6.2 LIVENESS

Liveness is a property which guarantees the absence of deadlocks in a manufacturing system. Deadlock is a situation where manual intervention is necessary to restore the functioning of an automated manufacturing system. A transition in a Petri net is said to be live if, for any marking reachable from the initial marking, there exists a firing sequence which enables that transition. The net is said to be live under an initial marking, if all of its transitions are live. Figures 3.10 gives an example of a Petri net and its reachability graph that is not live. It can be observed in the reachability graph that none of the transitions are enabled in marking M_3.

3.6.3 REVERSIBILITY

Reversibility is another important property of Petri net models. A Petri net is said to be reversible if its initial marking can be reached, from any marking, reachable from that initial marking. Reversibility guarantees error recovery of an automated manufacturing system without manual intervention. Figure 3.11 illustrates a Petri net and its reachability graph which is not reversible. It can be noted in the reachability graph that, marking M_0 cannot be reached again once transition t_1 is fired.

Boundedness, liveness and reversibility are known as behavioral properties. Behavioral properties depend on the initial marking. For example a Petri net which does not have the above desired properties might be made to be bounded, live and reversible under a different initial marking. Research related to the appropriate initial marking which ensures the desired properties are found in the work by Zhou, DiCesare, and Desrochers [59], [60].

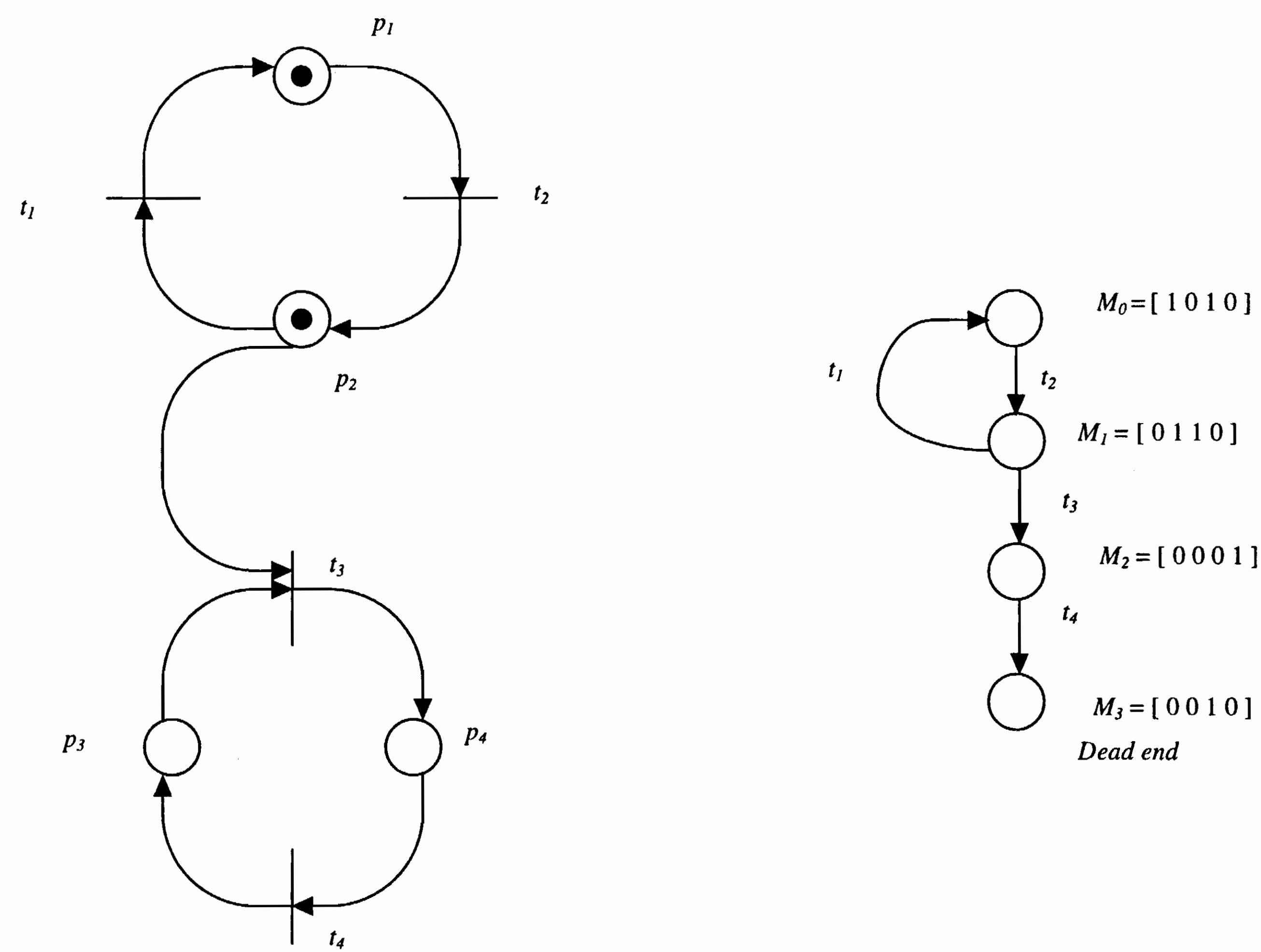


Figure 3.10 An example of a Petri net which is not live

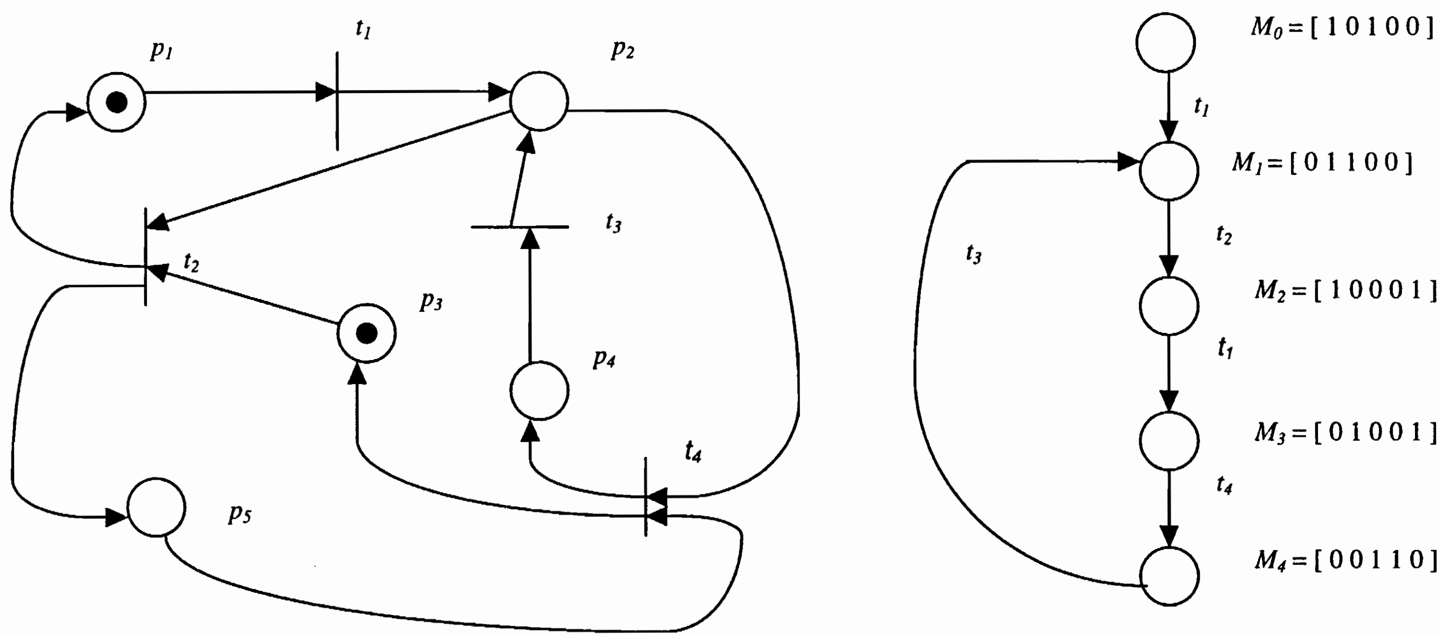


Figure 3.11 An example of a Petri net and its reachability tree which is not reversible

3.7 STRUCTURAL PROPERTIES OF PETRI NETS

The behavioral properties of Petri nets are evaluated using the reachability or coverability graphs. For example a Petri net can be shown to be unbounded if there exists a pair of markings $M_j(p)$ and $M_i(p)$, such that $M_j(p)>M_i(p)$ and $M_i(p)[\sigma >M_j(p)$. A Petri net is not live if there exists a node which is a dead end in its reachability graph. The Petri net is reversible if there exists a directed path from any node to the initial node in its reachability graph. However, the reachability graph of a Petri net can be very large and difficult to be analyzed. The process of evaluating the properties of a Petri net is simplified to some extent by checking its structural properties. The structural properties are independent of the initial marking and are evaluated using the incidence matrix of a Petri net. Sifakis [48], developed many useful results in providing the conditions for a Petri net to guarantee some structural properties. The following sections introduce some of his results.

3.7.1 STRUCTURAL BOUNDEDNESS

A Petri net is said to be structurally bounded if it is bounded for any initial marking. Therefore, structural boundedness is a stronger property than boundedness. If A is the incidence matrix of a Petri net, then the Petri net is structurally bounded if and only if there exists a vector x of positive integers such that $x^T.A \leq 0$. Therefore, if a vector which satisfies the above conditions can be found, the Petri net will be bounded for any initial marking.

3.7.2 STRUCTURAL CONSERVATIVENESS

Conservativeness is a special case of boundedness. A Petri net is structurally strictly conservative, if the number of tokens in its places is constant for any marking reachable from any initial marking. It is structurally conservative if the weighted number of tokens is constant for any marking reachable from any initial marking. The property conservativeness has good relevance to the modeling of manufacturing systems. The places in the Petri net model of a manufacturing system represent resources like machines, pallets and material handling equipment like robots and AGVs. When all of the places of the Petri net model represent the resources, the model should ensure that the conservation of the resources is maintained. In other words, the total number of tokens in the net should be the same for any marking reachable from the initial marking. Therefore the Petri net model of such a manufacturing system should be strictly conservative.

However, in general, the places in a Petri net of a manufacturing system represent resources as well as operations. Therefore, it should ensure that the weighted number of tokens in the model is constant. The weights depend on the resources needed for the different operations (e.g. a part and a machine needed for machining). When the weighted number of tokens in a Petri net is the same for any marking, the Petri net is said to be conservative. The weighted sum of the tokens is constant if there exists a non zero vector x, such that $x^T.M_0=x^T.M$, for any $M \in R(M_0)$. A Petri net is structurally conservative if it is conservative for any initial marking. Therefore it is structurally conservative if there exists a non zero vector x such that such that $x^T.A=0$. The Petri net is structurally strictly conservative if there exists a unit vector x such that $x^T.A=0$.

3.7.3 STRUCTURAL LIVENESS

A Petri net is said to be structurally live if it is live for any initial marking. Unfortunately there is no characterization of structural liveness in linear algebraic terms.

3.7.4 STRUCTURAL CONSISTENCY

Generally, automated manufacturing systems function in a cyclic manner. Therefore, a Petri net of an automated manufacturing should have the property of functioning repetitively. This property of a Petri net is known as consistency. A Petri net is said to be structurally consistent, if there exists an initial marking such that, it can be reached by firing all of the transitions of the net at least once. A Petri net will be consistent if and only if there exists a vector y such that $A.y=0$.

3.8 INVARIANT ANALYSIS OF PETRI NETS

Invariant analysis another important theoretical development in the study of Petri nets. This analytical procedure makes use of the state equation discussed in an earlier section. The state equation is written as $M_k=M_0+A.U$.

3.8.1 PLACE INVARIANT

A place invariant represents a set of places in which the weighted sum of the tokens is constant for any reachable marking from any initial marking. Therefore, a place invariant is a non negative integer vector x, satisfying the condition $x^T.A=0$. When this condition is satisfied, $x^T.M_k=x^T.M_0$ for any marking M_k reachable from the initial marking M_0. There can be more than one non negative integer vector satisfying the condition stated above. Therefore, a Petri net can have more than one place invariant. A linear combination of two or more place invariants is also a place invariant. A minimal place invariant is a place invariant which is not a linear combination of other place invariants. A simple Petri net and its minimal place invariants of a simple Petri net are shown in Figure 3.12 and 3.13 respectively.

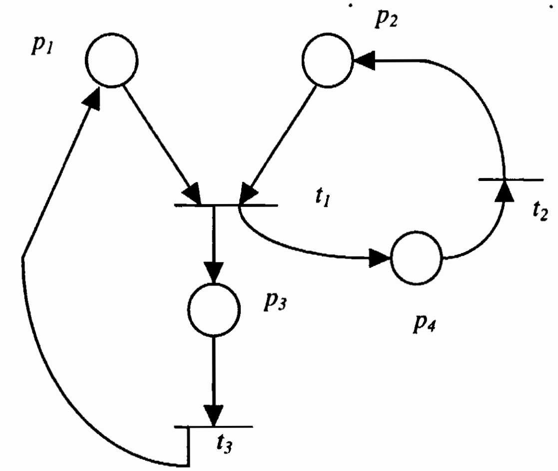


Figure 3.12 The original Petri net

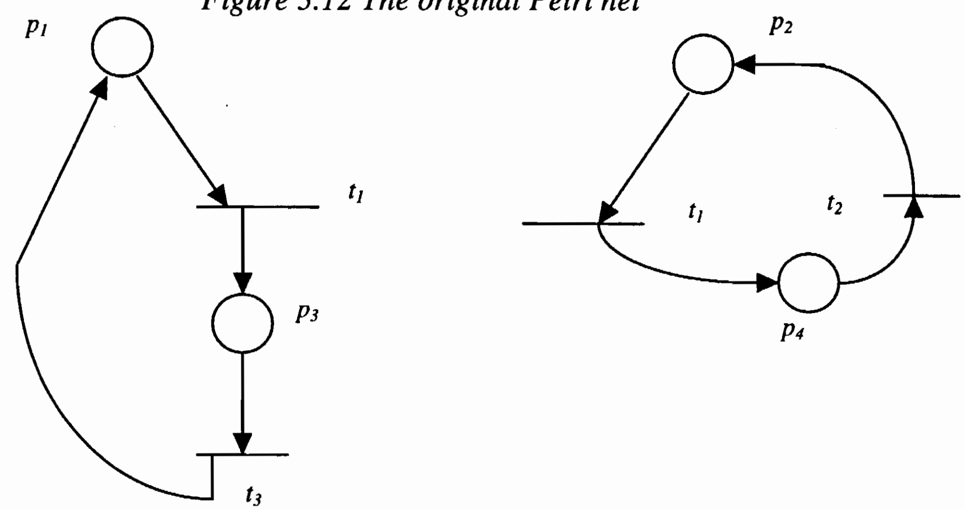


Figure 3.13 The place invariants of the original Petri net

The minimal place invariants are [1 0 1 0] and [0 1 0 1]. Therefore, the total number of tokens in p_1 and p_3 are the same for any marking reachable from the initial marking. Similarly, the total number of tokens in p_2 and p_4 will be constant for any marking reachable from the initial marking. Further, the vector [1 1 1 1] is also a place invariant of the above Petri net. This is because it is a linear combination of the previous place invariants. In this example, all the places of the Petri net are covered by the place invariants and hence the weighted sum of the tokens in this Petri net remains constant.

3.8.2 TRANSITION INVARIANT

A transition invariant represents a set of transitions which can be represented by a non zero vector y satisfying the condition $A.y=0$. A transition invariant gives the series of firing of transitions which brings the marking of the Petri net to its initial marking. This property is related to the cyclic functioning of a manufacturing system. The only transition invariant of the previous example is [1 1 1].

3.9 SPECIAL CLASSES OF PETRI NETS

In general many manufacturing systems can be modeled using Petri nets with special properties. These Petri nets are known as special classes of Petri nets. They fall into a major category known as ordinary Petri nets. In an ordinary Petri net, the weights of the input and output arcs between the places and transitions is either 0 or 1. Four important special classes of Petri nets exist in the literature. They are known as marked graphs, state machines, free choice nets and simple nets. This section focuses on the two most important special classes of Petri nets known as marked graphs and state machines.

3.9.1 MARKED GRAPHS

A marked graph is an ordinary Petri net in which, each place has exactly one input transition and exactly one output transition. An example of a marked graph is illustrated in Figure 3.14. A marked graph is ideally suited for the modeling of manufacturing systems with synchronization and without conflicting situations. For a marked graph to be live, it should not contain source or sink places or transitions. A source place or a source transition is a node which does not contain any incoming arcs and a sink place or a sink transition is a node which does no contain any out going arcs. In a live marked graph, different circuits can be formed. A circuit is closed path. A path is a set of distinct set of nodes and arcs such that, the terminal node of each arc is the initial node of the preceding arc. For example, in the marked graph shown in Figure 3.14, four different circuits can be found. These circuits are (p_1 t_1 p_3 t_2 p_4 t_3 p_1), (p_1 t_1 p_3 t_2 p_5 t_3 p_1), (p_2 t_1 p_3 t_2 p_4 t_3 p_2) and (p_2 t_1 p_3 t_2 p_5 t_3 p_2). A marked graph is live, bounded and reversible if each of its circuits is marked with at least one token.

3.9.2 SATE MACHINES

A state machine is an ordinary Petri net where each of its transition has exactly one input place and exactly one output place. Figure 3.15 illustrates a state machine. State machines are used to model conflicts but not concurrency or synchronization. A state machine is live, bounded and reversible if it is strongly connected and marked with at least one token. A strongly connected Petri net is one in which there exists a directed path from every node to every other node. The firing of a transition of a state removes exactly one token from input place and generates exactly one token in its output place. Therefore, the total number of tokens in a state machine will be the same for any marking reachable from the initial marking. Therefore, a strongly connected state machine is strictly conservative.

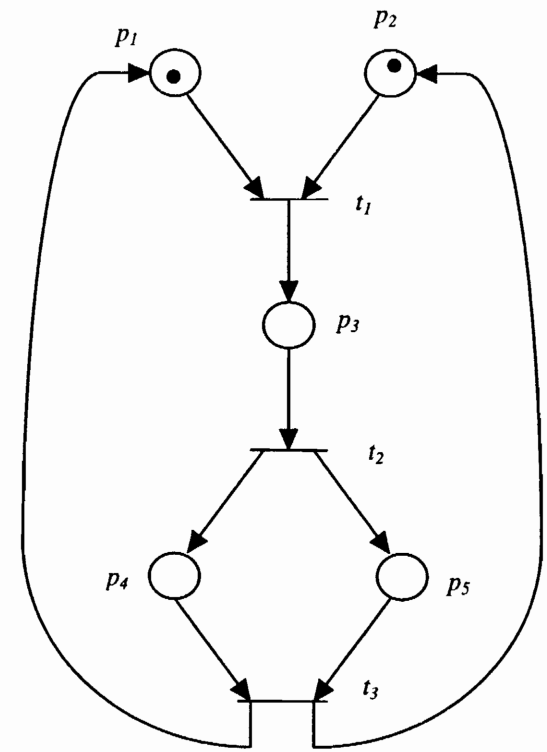


Figure 3.14 Illustration of a marked graph

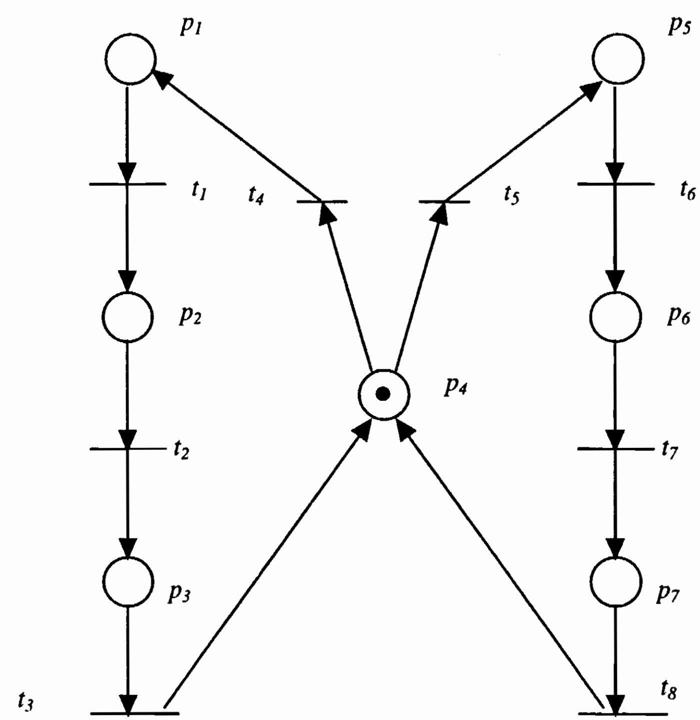


Figure 3.15 Illustration of a state machine

3.10 SYNTHESIS PROCEDURES FOR PETRI NETS

When Petri net models are constructed, it is necessary that the desired properties such as boundedness, liveness and reversibility are guaranteed. The two major approaches for the systematic synthesis technique of Petri nets are available in the literature. The first is known as the bottom up approach [2], [28] and [39]. The second is the top down approaches [53], [51], and [58]. Another useful procedure is based on the concepts of parallel and sequential exclusion [59]. This is mainly used for the modeling of manufacturing systems with shared resources. The hybrid technique [60] is a method of modeling manufacturing systems using the concepts of top down approach, bottom up approach and the parallel and sequential mutual exclusion in a systematic manner. These techniques are explained in the following sections

3.10.1 BOTTOM UP APPROACH

The bottom up synthesis technique was first introduced by Agerwala and Choed Amphai [2] in 1978. They proposed a systematic way of building the Petri net model by constructing sub nets for each operation or process. The sub nets are then merged along common places to construct the final net. Narahari and Viswanadham [39] in 1985, proposed a similar approach to model a flexible manufacturing system. The bottom up approach is illustrated using the example given in Figure 3.16. This figure shows an AGV based transportation system with one AGV and two machines. Parts are moved from the storage area to machine 1 by the AGV. Once the machining is completed on machine 1, the AGV unloads the part and loads it on machine 2. A new part will be available at the receiving area only after machine 2 is free. Figure 3.17 is an illustration of the sub net modeling the loading of the part on machine 1 and its machining operation.

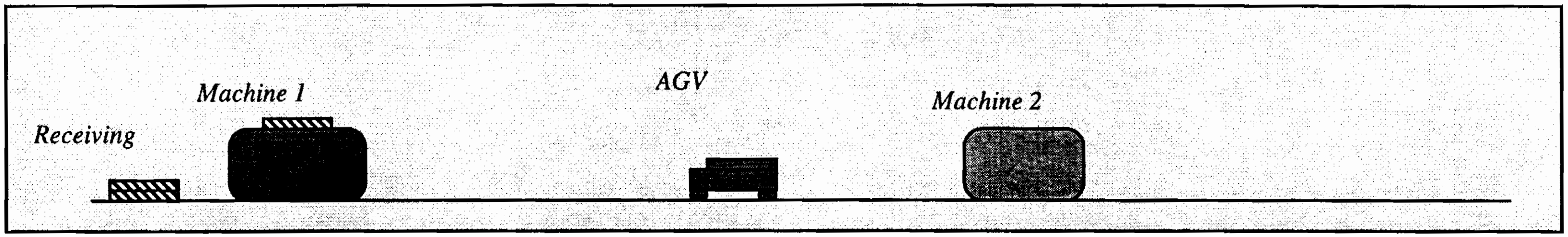


Figure 3.16 Illustration of a simple AGV based manufacturing system

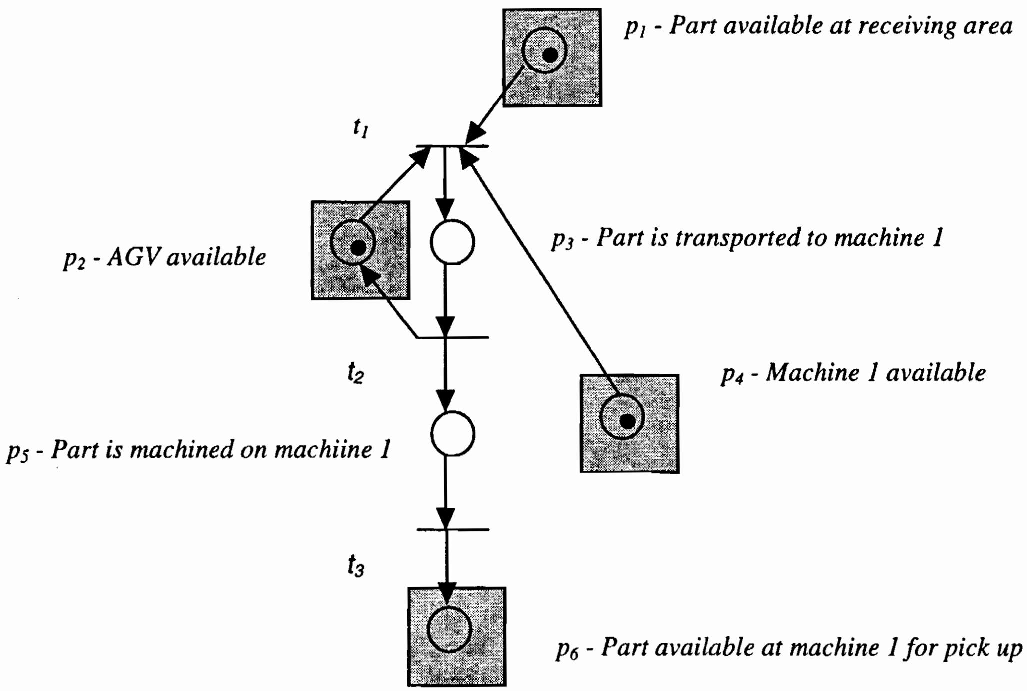


Figure 3.17 Illustration of the sub net describing the loading and machining of a part on machine 1

The sub net describing the loading and machining on machine 2 is illustrated in Figure 3.18. The common places of these sub nets are p_1, p_4 and p_6. When the two sub nets are merged along these common places, the final Petri net model will be as illustrated in Figure 3.19. The advantage of the above bottom up approach is that the place invariants of the merged nets can be computed easily at every stage. However, the usefulness of invariants are limited only to the verification of the boundedness of the Petri net. Therefore, the liveness of the final net remains to be verified.

Later, in 1986 Krogh and Beck proposed a different bottom up approach. Their method is suitable for the construction of live and safe Petri nets. However, it is limited to the modeling of manufacturing systems which function in a cyclic manner. It begins with the construction of elementary circuits to represent each process in a manufacturing system. Once all the circuits are constructed, they are merged along special paths known as solitary place path and solitary transition path. A solitary place path is one which terminates on both ends by a place and each transition having exactly one input and exactly one output place. A solitary transition path is one which terminates on both ends by a transition and each place in the path is an input place for exactly one transition and an output place for exactly one transition.

Figures 3.20, 3.21 and 3.22 illustrate the circuits modeling each operation for example problem given earlier. When these circuits are merged along the solitary place and solitary transition paths, the final net as shown in Figure 3.19 can be constructed.

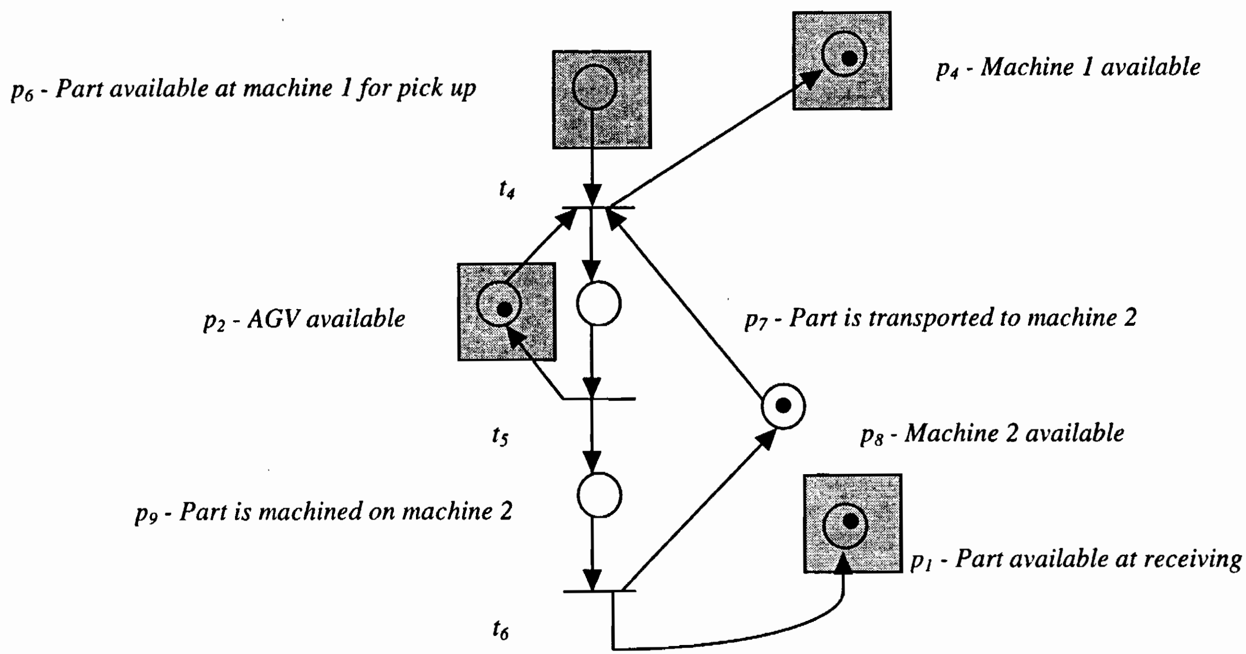


Figure 3.18 Illustration of the sub net describing the loading and machining of a part on machine 2

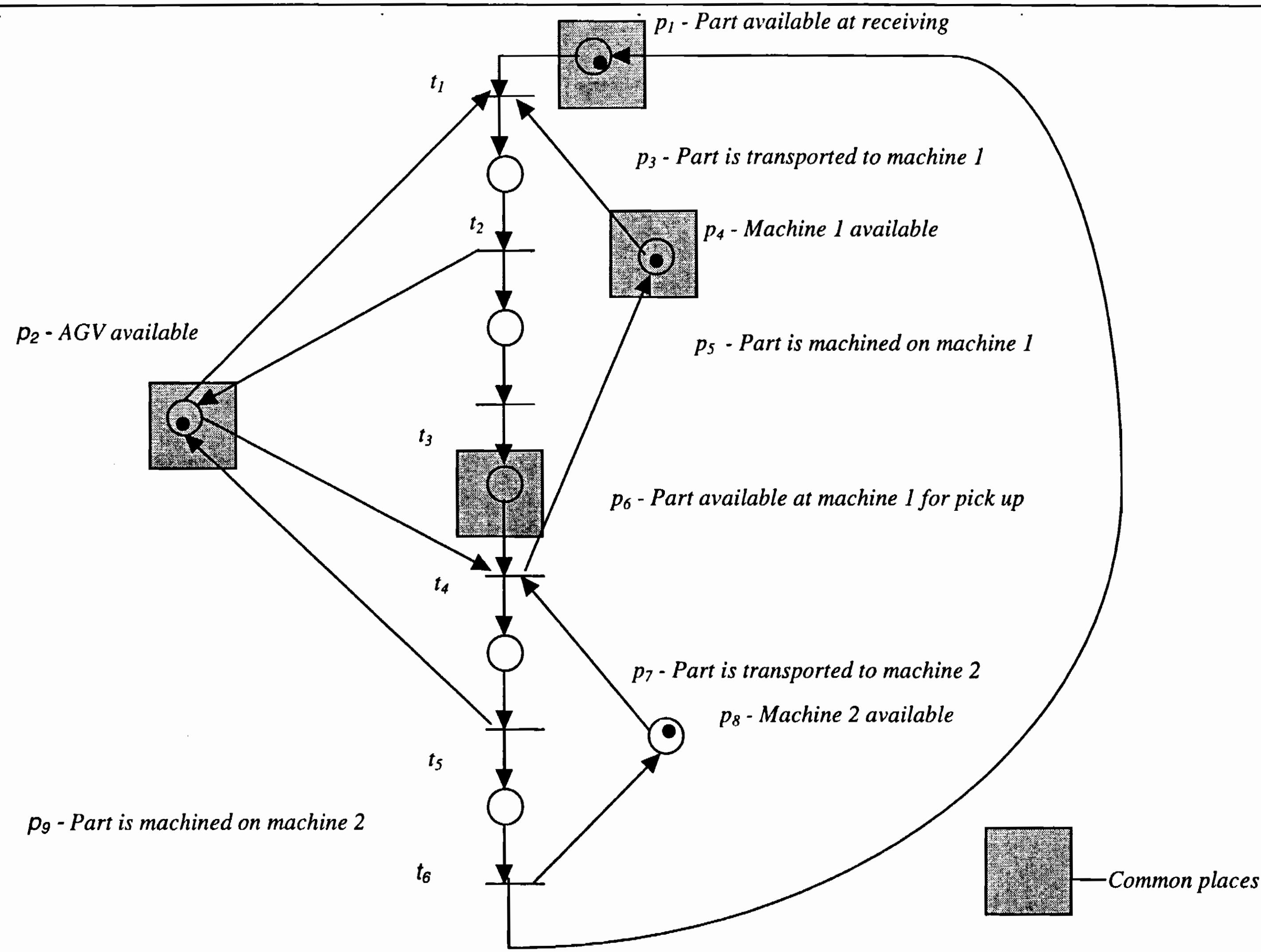


Figure 3.19 Illustration of the complete Petri net model of the AGV based manufacturing system

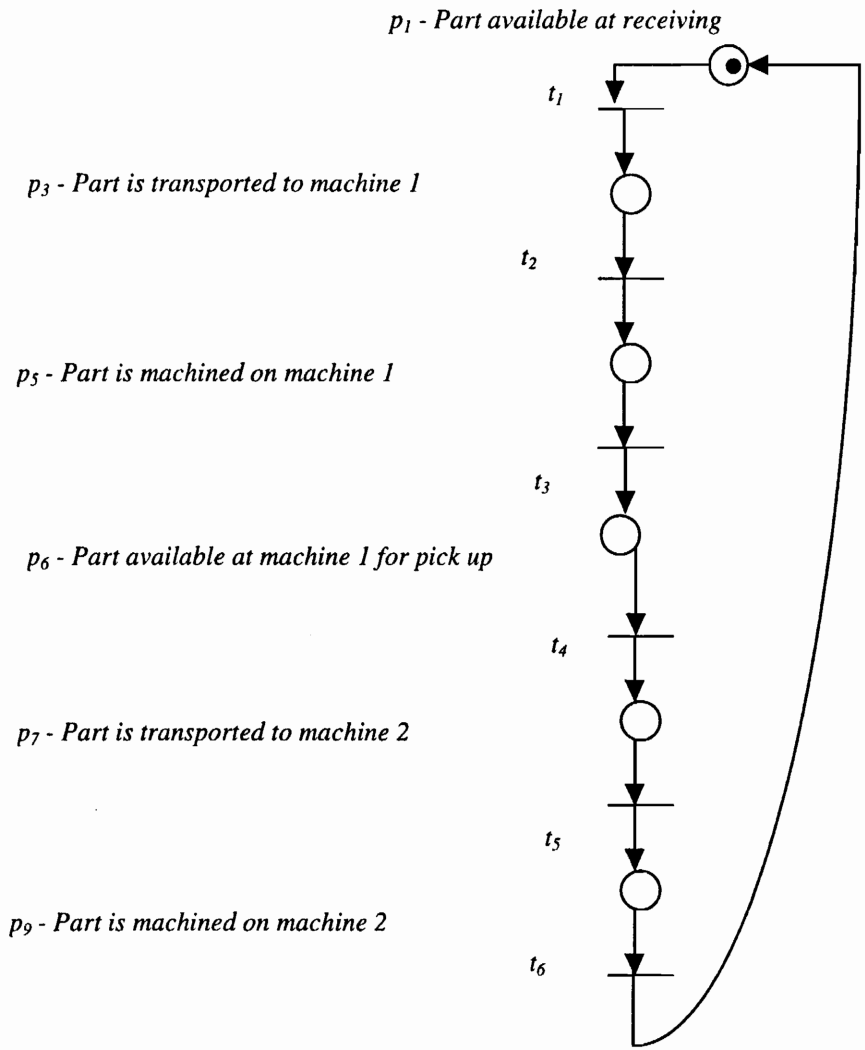


Figure 3.20 The circuit modeling the entire processing of a part

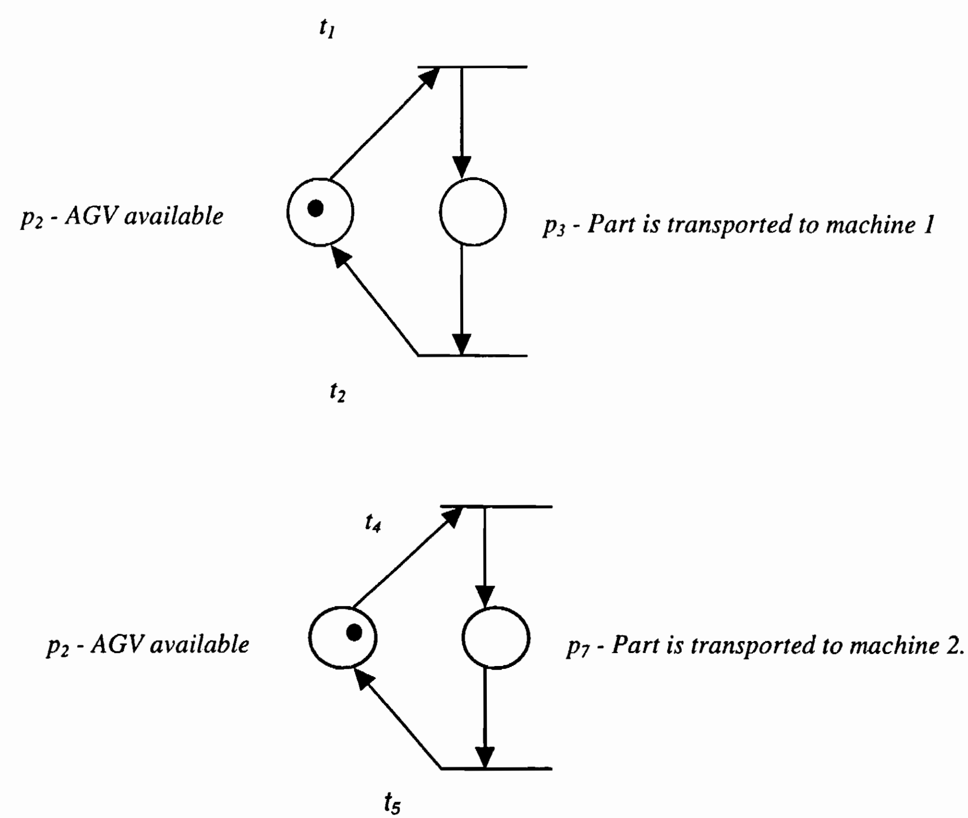


Figure 3.21 Circuits modeling the operations of the AGVs

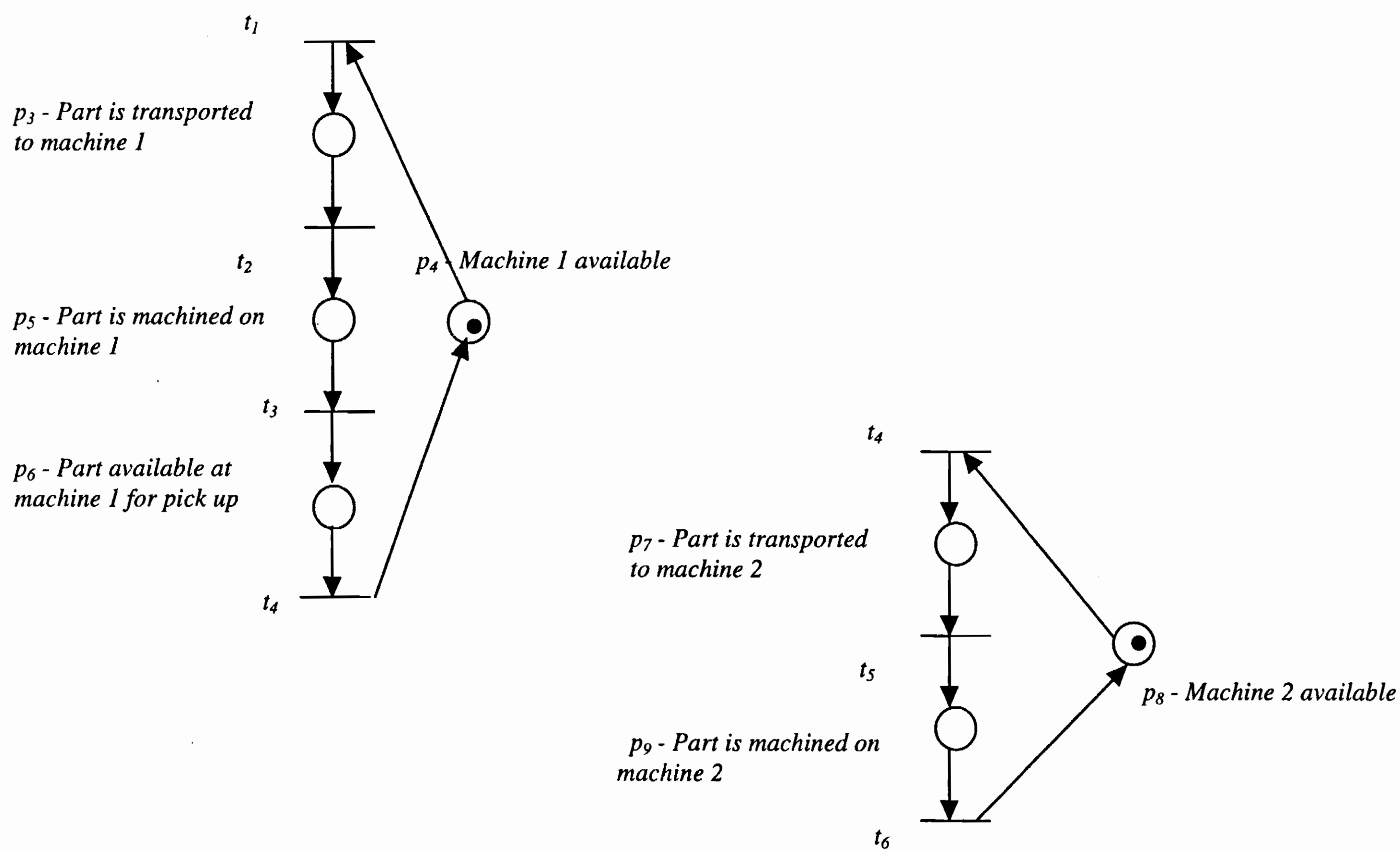


Figure 3.22 Circuits modeling the operations by machines 1 and 2

3.10.2 TOP DOWN APPROACH

The top down approach for constructing Petri net models is based on the pioneering work of Valette [53]. His concept of well formed blocks was a major break through in the construction of Petri net models. The idea of this concept is based on a sub net termed a block. A block is a Petri net which has one and only one transition that is named initial transition t_{ini} and one and only one transition that is termed, the final transition t_f. An associated Petri net of a block is a net obtained by adding an idle place p_0 such that, t_{ini} is the only output and t_f is the only input transition of p_0. A block and its associated Petri net are illustrated in Figures 3.23 and 3.24.

A block is well formed, if the following conditions are satisfied: (1) the associated Petri net is live, (2) the initial marking of the associated Petri net is the only marking such that the idle place is not empty, and (3) the only transition enabled by the initial marking is t_{ini}. When a transition is replaced by a well formed block in a live and bounded Petri net, the resulting Petri net also will be live and bounded. Figure 3.25 illustrates a simple Petri net where transition t_A is to be replaced with a block. The final Petri net after replacing transition t_A with a well behaved block is illustrated in Figure 3.26.

Suzuki and Murata [51] generalized the results of Valette [53] in 1983. They showed how a transition or a place of a Petri net can be replaced with special blocks known as well behaved blocks while preserving the desired properties of the original net. Figure 3.27 and 3.28 illustrate the initial modification for the replacement of a place p_0 in the aggregate Petri net. Place p_0 is first replaced with two places p_{01}, p_{02} and a transition t_0. Then transition t_0 is replaced with a well behaved block as shown in Figure 3.29.

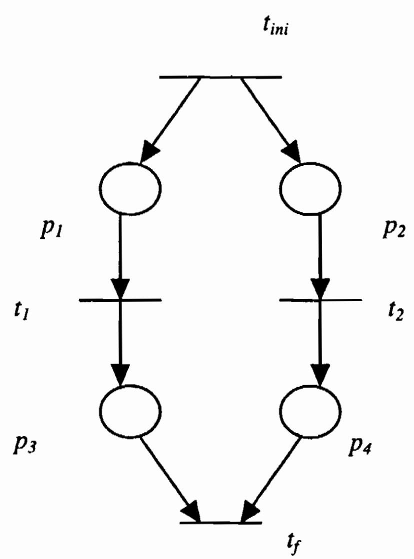


Figure 3.23 The well behaved block

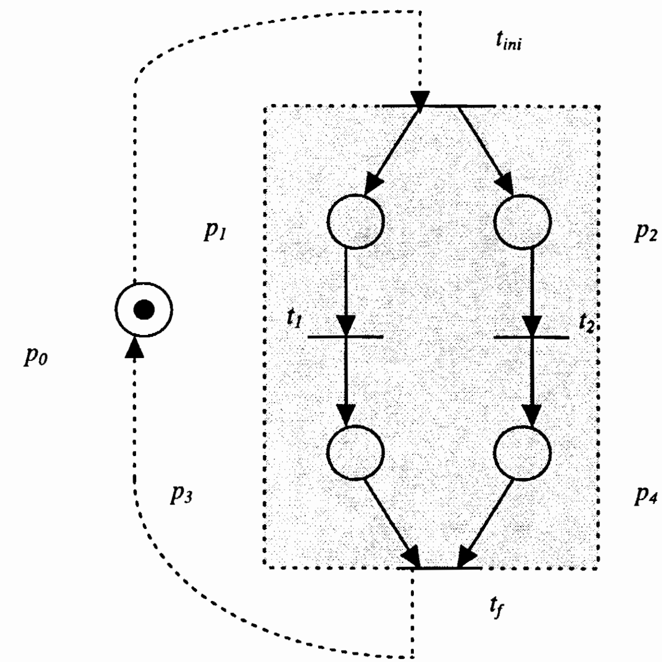


Figure 3.24 The associated Petri net of the block

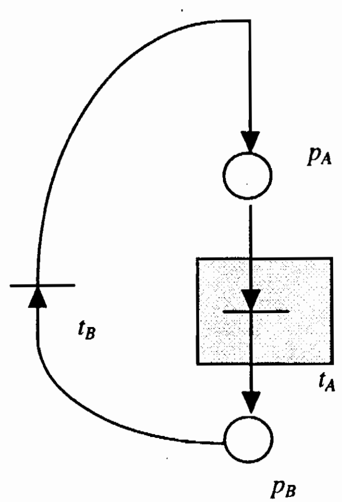


Figure 3.25 The original net where transition t_A is to be replaced with a well formed block

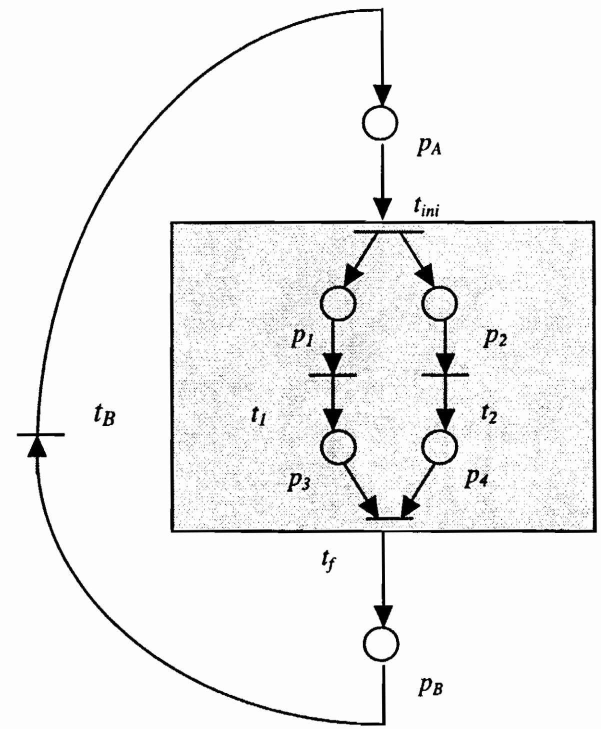


Figure 3.26 The final Petri net after replacing the transition with a well behave block

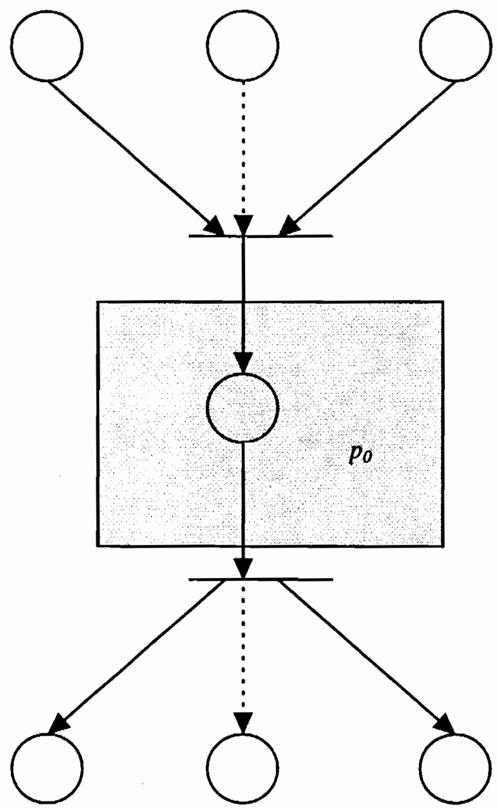


Figure 3.27 Illustration of a place which is to be replaced with a well behaved block

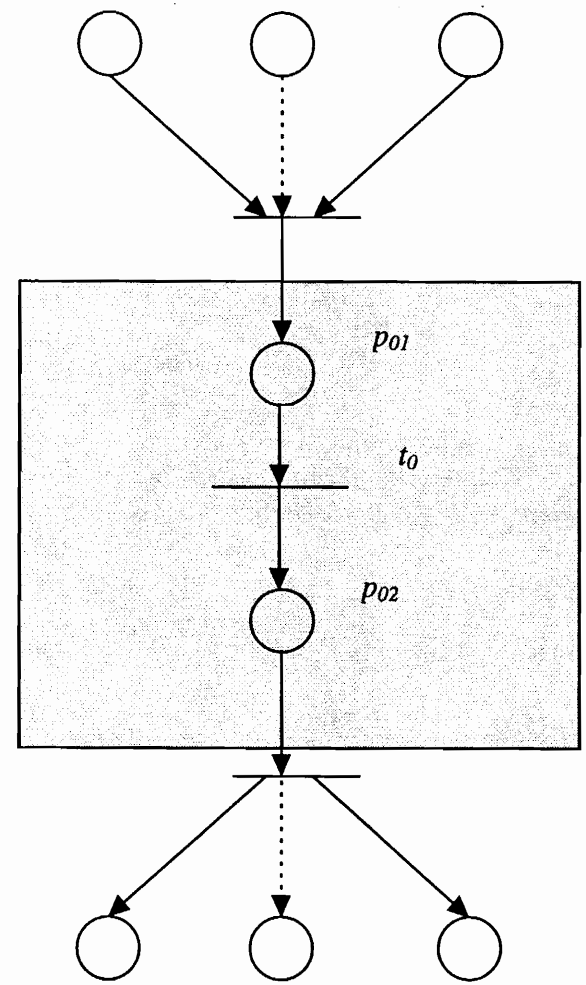


Figure 3.28 Illustration of a well behaved block

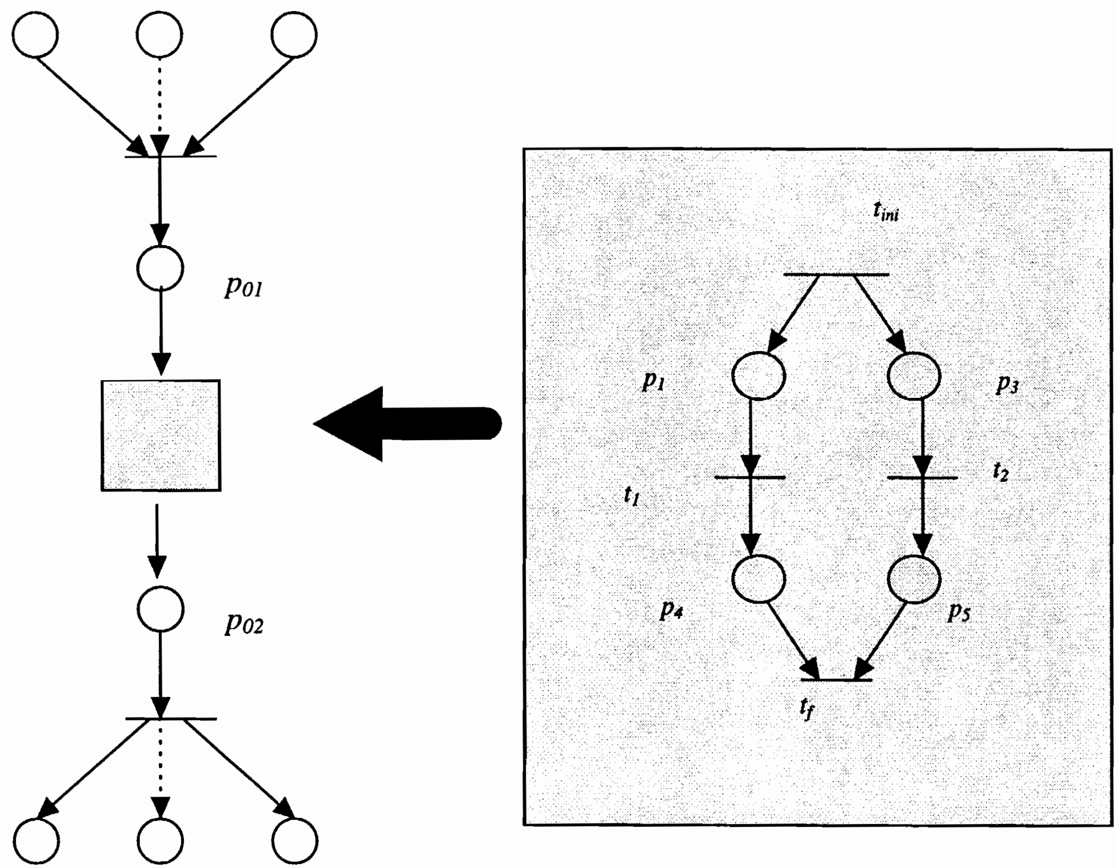


Figure 3.29 Illustration of transition t_0 being replaced by a well behaved block

3.10.3 THE BASIC DESIGN MODULES FOR MANUFACTURING SYSTEMS

In 1989, Zhou, DiCesare and Desrochers [58] used of the results of Valette [48] and Suzuki and Murata [51] to develop a top-down approach. Their method is ideally suited for the modeling of relatively large manufacturing systems. In this method, an aggregate Petri net model is developed with as few detail as possible to represent the manufacturing system. The aggregate Petri net is constructed such that it is live, bounded and reversible. Then the aggregate operations which are represented by the places are replaced with design modules to add more detail.

The four common design modules are named as sequential Petri nets, parallel Petri nets, choice Petri nets and decision free Petri nets. The sequential Petri net describes a series of successive operations, the parallel Petri net models parallel operations which can start simultaneously and the subsequent operations can start only after all the operations in the parallel Petri net are completed. In a conflict Petri net, a series of many different operations compete to start at the same time. However, not all of these operations can start at the same time. A decision free Petri net represents a situation where different operations which start based in a pre specified order. Figures 3.30, 3.31, 3.32 and 3.33 illustrate the above mentioned basic design modules.

3.11 PARALLEL AND SEQUENTIAL MUTUAL EXCLUSION

The concepts of parallel and sequential mutual exclusion were developed by Zhou and DiCesare [59] for the modeling of shared resources in a manufacturing system. The concepts of parallel and sequential mutual exclusions consist of extensive theoretical back ground This section provides the basic ideas of these concepts using simple examples.

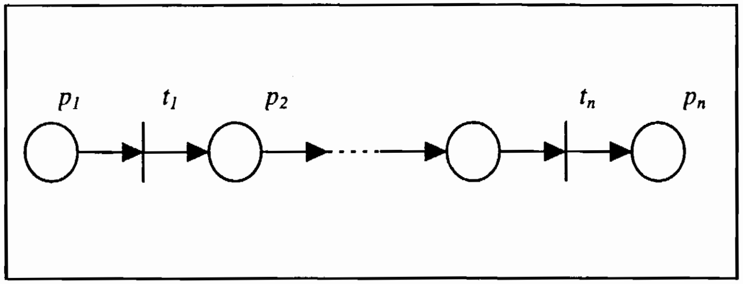


Figure 3.30 Series Petri net

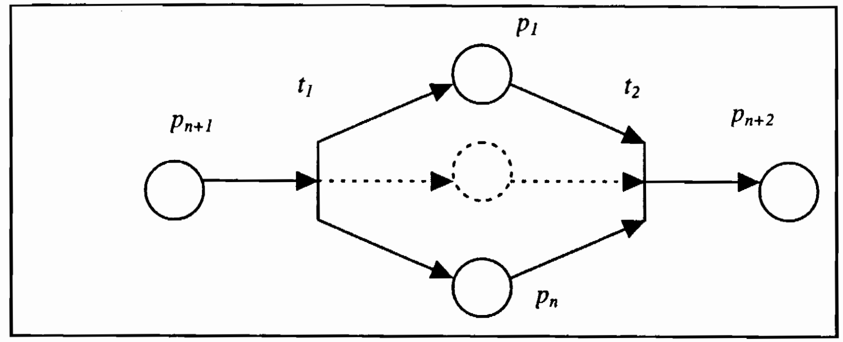


Figure 3.31 Parallel Petri net

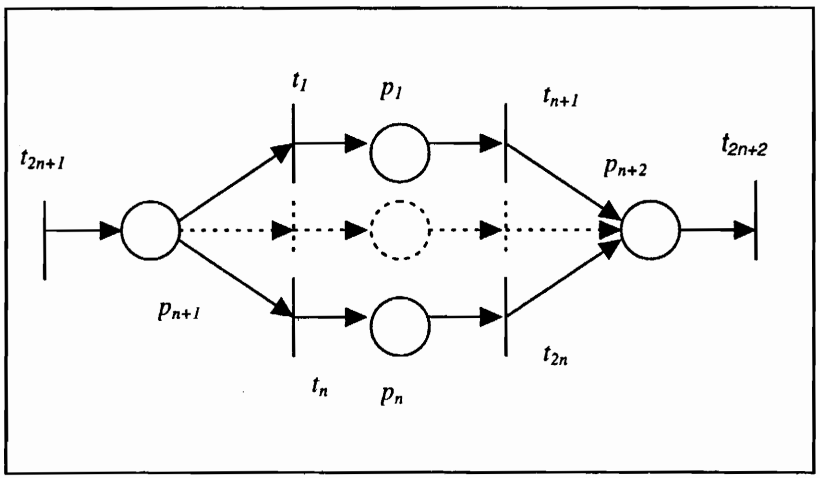


Figure 3.32 Conflict Petri net

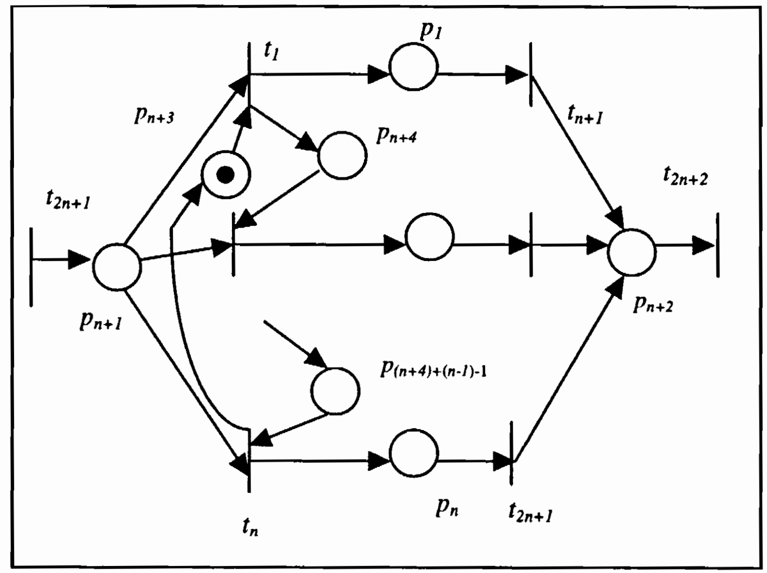


Figure 3.33 Decision free Petri net

A k parallel mutual exclusion is a situation where k independent process or operations share a single resource. In a k parallel mutual exclusion the shared resource is represented by a place. The input and output arcs are added to the place representing the shared resource in such a way that the k parallel processes have equal access to the shared resource. Figure 3.34 illustrates an example of a 2 parallel mutual exclusion. In this example, process 1 and process 2 compete for a single resource at the same time. A sequential mutual exclusion is a situation where, a single resource is shared by two or more operations or processes which are performed in a sequential manner. The subsequent process gets access to the shared resource after its preceding process releases that resource. An example of a sequential mutual exclusion is given in Figure 3.35.

The important consideration in modeling shared resources using parallel or sequential mutual exclusion is that the input and out arcs from the places representing the shared resources should be constructed in such a way that the desired properties of the Petri net are preserved. Further, the initial distribution of the tokens in the Petri net also should be appropriate such that these desired properties of the net are preserved. Detail analysis of the concepts of parallel and sequential mutual exclusion are found in Zhou and DiCesare [59].

3.12 THE HYBRID SYNTHESIS PROCEDURE

The hybrid synthesis procedure proposed by Zhou, DiCesare. and Desrochers [60] is a combination of the top down, bottom up approaches and the concepts of parallel and sequential mutual exclusion. It is a stepwise refinement procedure for the modeling of bounded, live and reversible Petri net models for manufacturing systems involving shared resources.

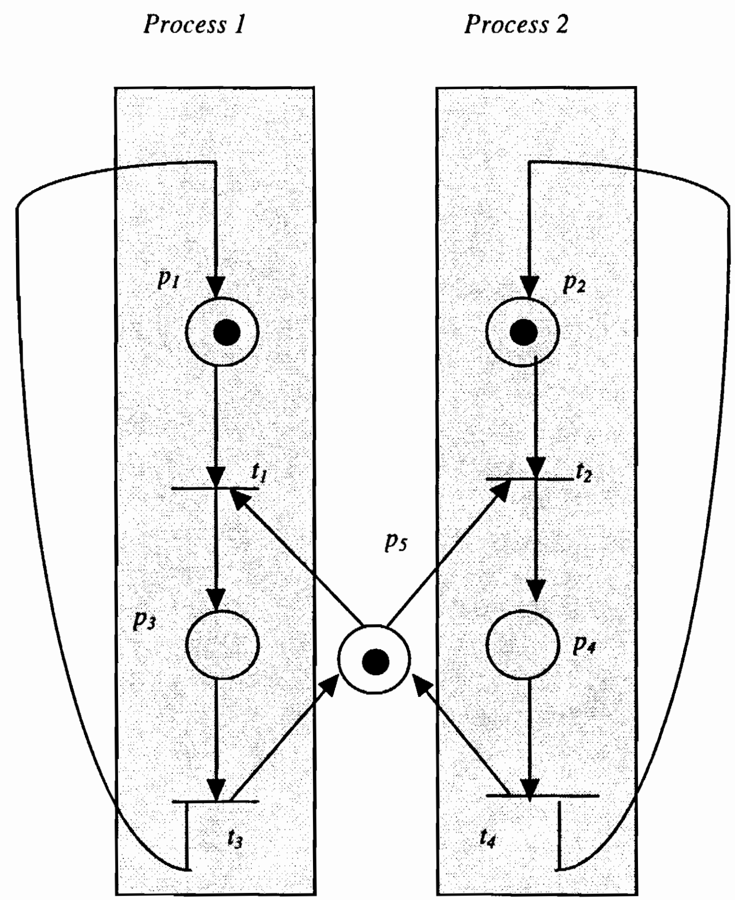


Figure 3.34 Illustration of a parallel mutual exclusion

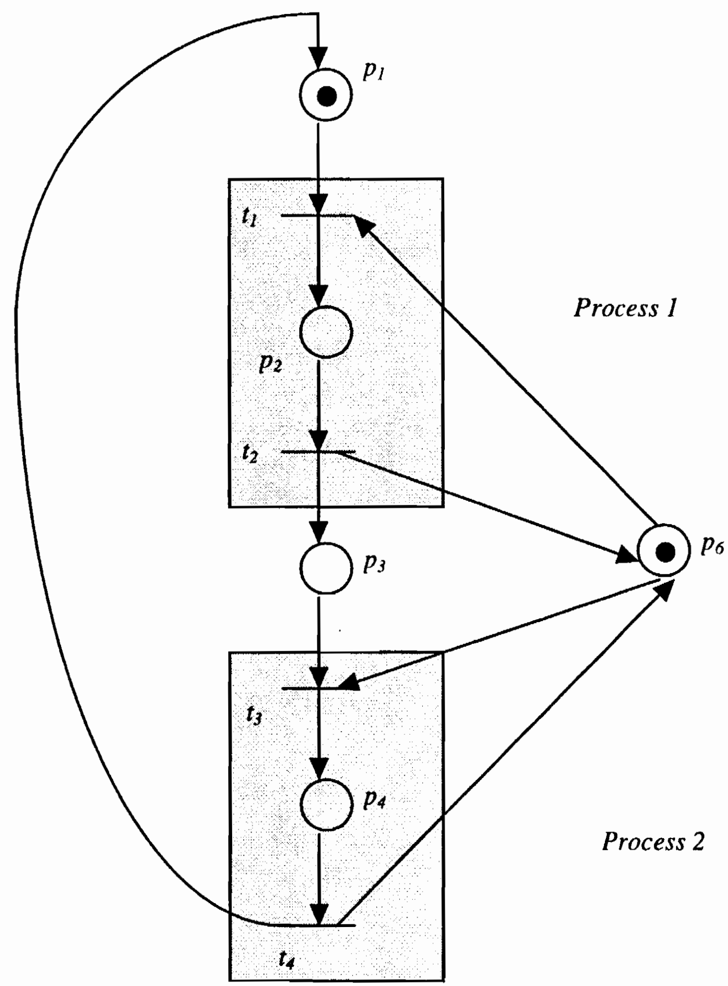


Figure 3.35 Illustration of a sequential mutual exclusion

Initially, in the hybrid procedure, an aggregate Petri net is built with as few detail as possible. Then, the appropriate transitions and the places of the aggregate Petri net are replaced with the basic design modules to add more detail. In the third step the non shared resources are added. In the final step the shared resources are added according to the concepts of parallel and mutual exclusion.

Figure 3.36 illustrates a simple AGV based transportation system. This figure is used to explain the hybrid methodology of constructing a Petri net model. Figure 3.37 illustrates the aggregate Petri net model representing the AGV based transportation system. Transitions t_{01} and t_{02} model the beginning and end of the transportation of a part. Place p_{01} represents the transportation and machining of a part. In Figure 3.38, place p_{01} is replaced with a series Petri net. The series Petri net models the sequence of operations involving the transportation of the part to machine 1, the machining on machine 1, unloading of the part from machine 1, loading the part on machine 2, machining the part on machine 2 and the unloading of the part from machine 2. The non shared resources in this manufacturing system are machines 1 and 2. Figure 3.39 shows how the two non shared resources are included the Petri net model of Figure 3.38. Tokens residing in the places representing the two machines are removed when they are in operation and deposited back after the parts are unloaded after the machining. Figure 3.40 shows the inclusion of the shared resource. The shared resource in the manufacturing system is the AGV. The AGV is used by two processes functioning in a sequential order. The first process is the loading, machining and unloading taking place in machine 1. The second process is the loading, machining and unloading from machine 2. The AGV is shared by these two processes. In Figure 3.40, place p_2 models the AGV and it is included in the Petri net model as a shared resource for two sequentially operating processes.

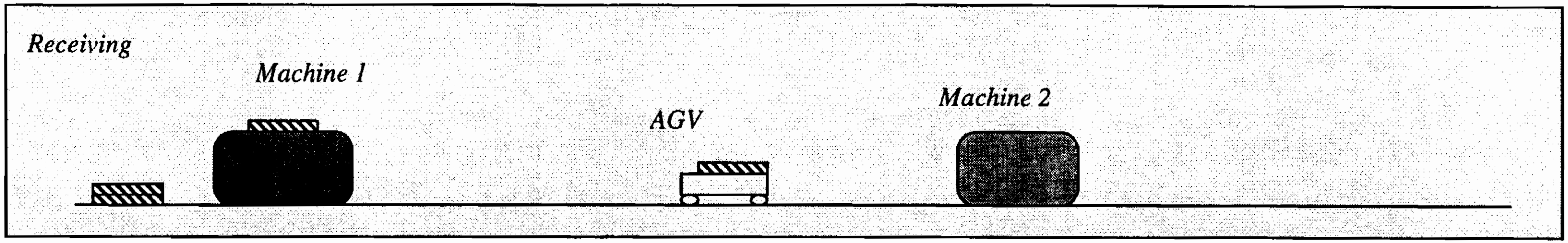


Figure 3.36 Illustration of the example manufacturing system

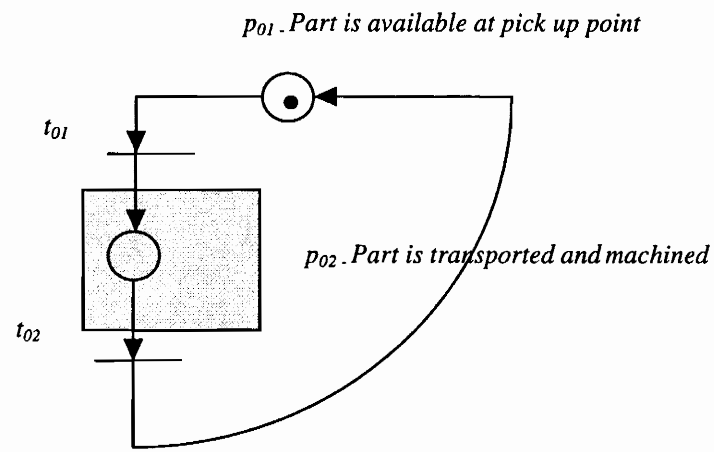


Figure 3.37 The illustration of the aggregate Petri net

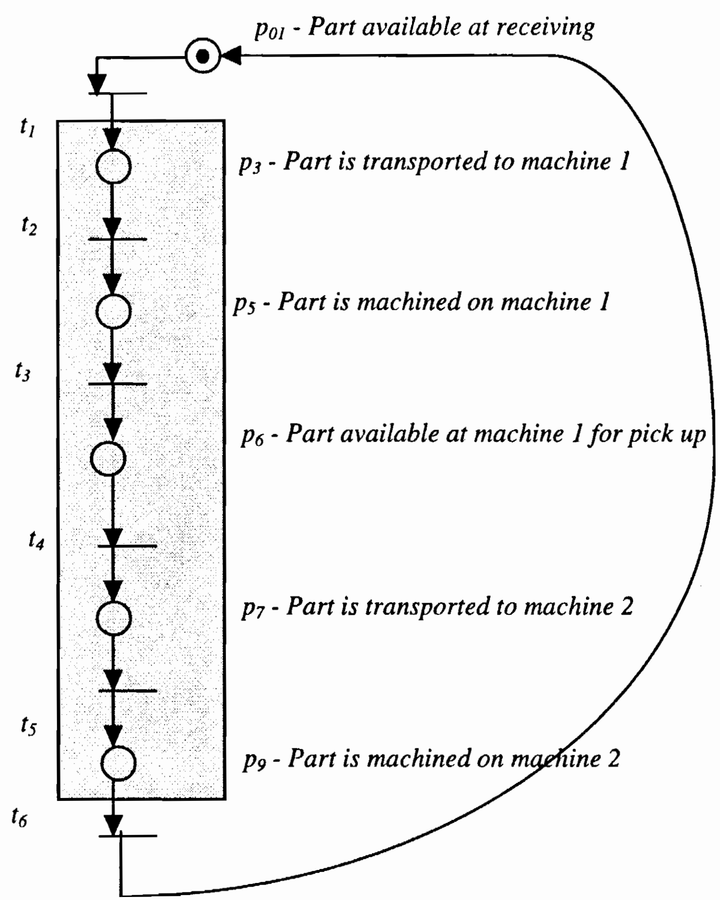


Figure 3.38 The Petri net after replacing a place with a sequential Petri net

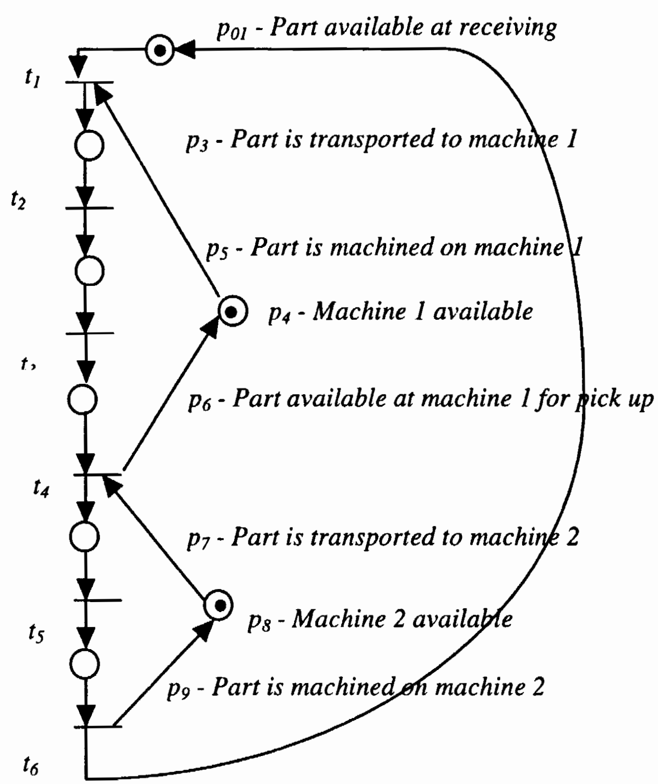


Figure 3.39 The Petri net after the non shared resources are added

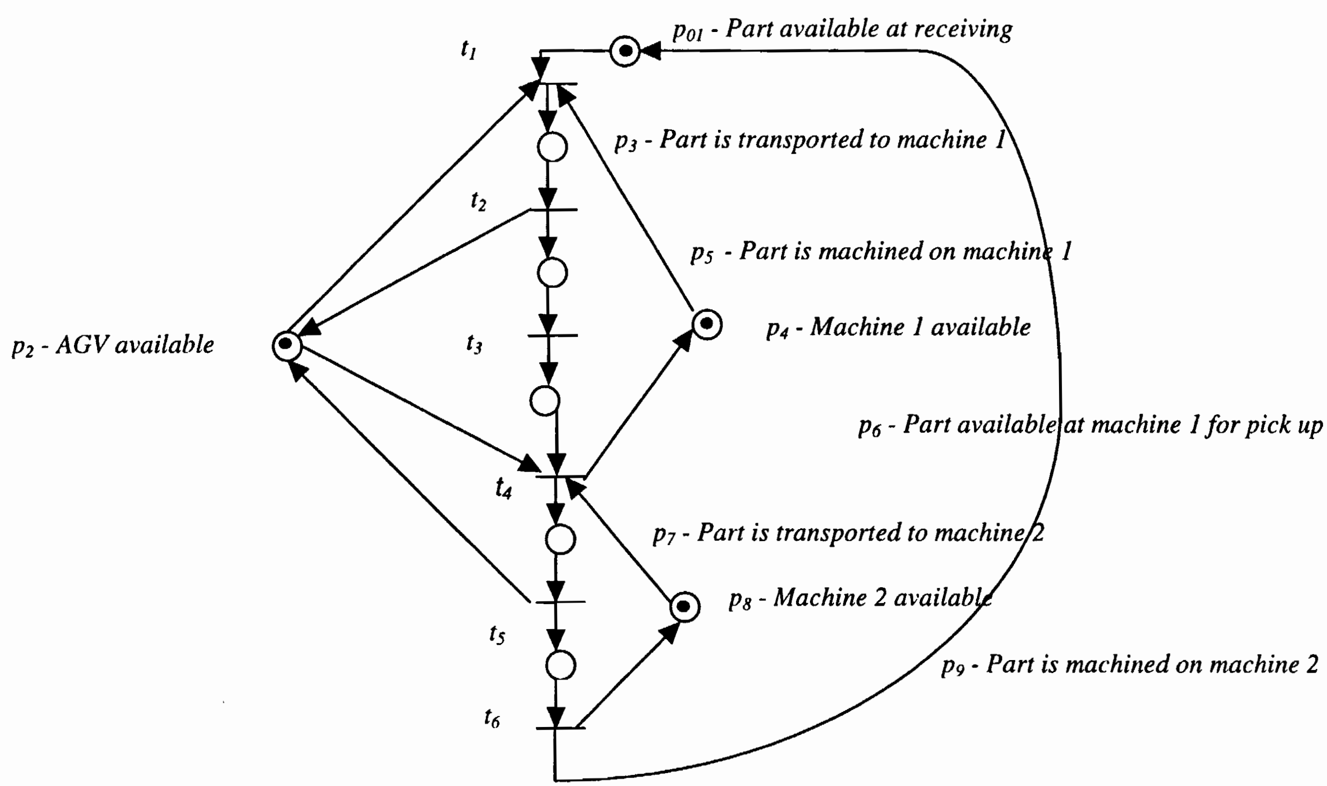


Figure 3.40 The final Petri net after the shared resources are added

3.13 TIMED PETRI NETS

When Petri net theory was introduced by Carl Adam Petri in 1965 in his doctoral dissertation, the notion of time was not included. Therefore, the original form of Petri nets was only used to study the logical behavior of a system. However, when the quantitative performance measures are to be evaluated, time becomes an essential component of any Petri net model. This was the motivational factor for Ramachandni's doctoral dissertation in 1973. He was the first to provide a way of associating time to Petri net models. By associating deterministic times to the transitions of a Petri net he showed how the performance measures of a system can be evaluated.

Since then, many forms of timed Petri nets have been developed by researchers. These forms include deterministic timed Petri nets, stochastic Petri nets with exponentially distributed timed transitions, generalized stochastic Petri nets which is a modified version of the stochastic Petri net modes, stochastic Petri nets with exponentially and deterministic timed Petri nets and stochastic Petri nets with non exponentially distributed timed transitions.

3.13.1 DETERMINISTIC TIMED PETRI NETS

Deterministic timed Petri nets are mainly studied using the matrix algebra. Two useful results in the study of deterministic timed Petri are due to the work of Ramamoorthy and Ho [45] and Sifakis [47]. Ramamoorthy and Ho [45] showed how the cycle time of a decision free Petri nets or marked graphs can be computed. They further showed performance evaluation of a general system which is not confined to marked graphs is NP complete. Sifakis [47] developed the conditions for a general system to function at its maximum rate. His results are useful in computing the number of different types of resources necessary to meet the throughput of a system. In this thesis we model AGV systems with deterministic travel times and deterministic demand for AGVs between work stations. The two important results related to deterministic timed Petri nets developed by Ramamoorthy and Ho [45] and Sifakis [47] are discussed in this section.

3.13.1.1 RESULTS OF RAMAMOORTHY AND HO

Ramamoorthy and Ho [45] developed two important results related to timed marked graphs. In the first theorem they showed that the cycle time of any transition in a marked graph is the same. The cycle time of a transition t_i is defined as $\lim_{n_i \to \infty} \frac{S_i(n_i)}{n_i}$. Where, $S_i(n_i)$ is the time at which the n_i^{th} firing of transition t_i is initiated.

In the next result, they showed that the minimum cycle time of any system represented by a marked graph is $C \geq \max\left\{\frac{T_k}{N_k}, k = 1,2,...q\right\}$. Where, T_k is the sum of the execution times of the transitions in circuit k, N_k is the sum of the tokens in the places in circuit k; and q is the number of circuits in the net.

These results are explained using the example illustrated in Figures 3.41 and 3.42. The four different circuits of the marked graph are ($p_1\ t_1\ p_3\ t_2\ p_4$), ($p_1\ t_1\ p_5\ t_2\ p_1$), ($p_2\ t_1\ p_3\ t_2\ p_2$) and ($p_2\ t_1\ p_5\ t_2\ p_2$). The values of T_k/N_k are $(\tau_1+\tau_2+\tau_3+\tau_4)/2$, $(\tau_1+\tau_2+\tau_3+\tau_4)$, $(\tau_1+\tau_2+\tau_3)/2$ and $(\tau_1+\tau_2+\tau_3)$. Therefore, the minimum cycle time of the system is Max $\{(\tau_1+\tau_2+\tau_3+\tau_4)/2, (\tau_1+\tau_2+\tau_3+\tau_4), (\tau_1+\tau_2+\tau_3)/2, (\tau_1+\tau_2+\tau_3)\}$.

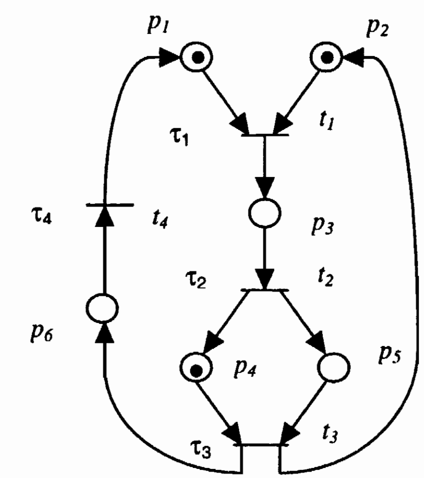


Figure 3.41 The original marked graphs

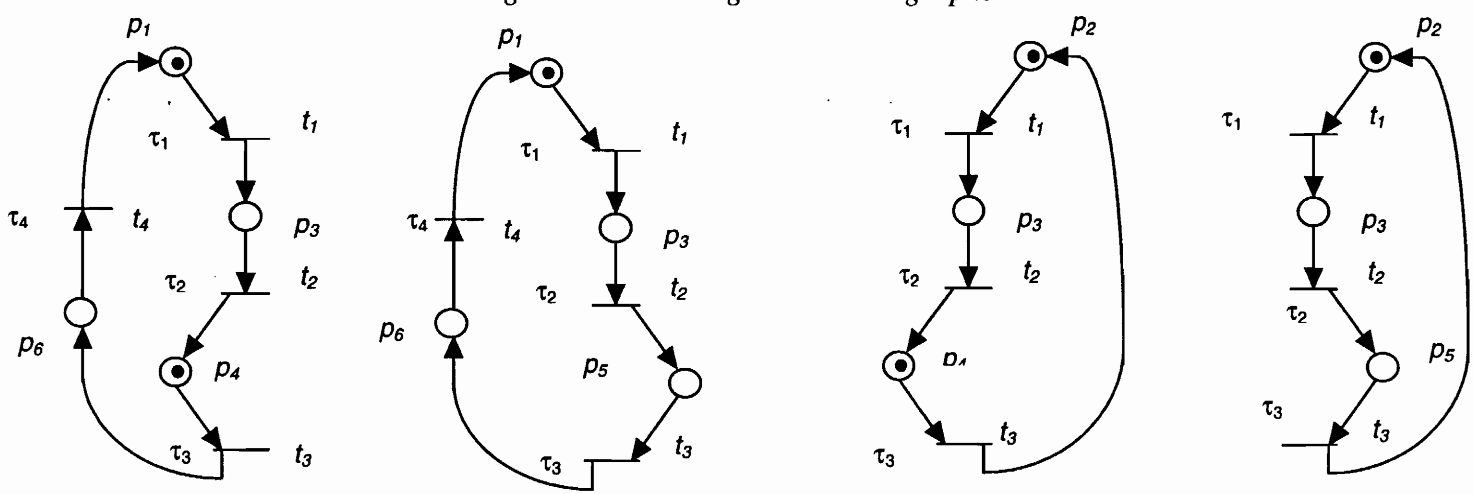


Figure 3.42 The four different circuits of the marked graph

3.13.1.2 RESULTS OF SIFAKIS

An important observation in the results of Ramamoorthy and Ho is that the computation of the minimum cycle time for a Petri net which is not decision free is intractable. They showed that the set partitioning problem can be reduced to this problem and thus it is NP complete. Realizing the limitations in analyzing decision free Petri nets, Sifakis [47] developed the conditions for a periodically functioning system to operate at its maximum rate. In the timed Petri nets of Sifakis, the delay times were associated to the places. However, he showed that an equivalent Petri net can be constructed with the delay times associated to the transitions. His results are applicable even to decision free Petri nets. The conditions for a system to function at its maximum operating rate is derived in terms of the initial marking, the time delays associated with places, and the firing frequencies of the transitions

The following condition should be satisfied for all place invariants of the Petri net for the system to function at its maximum rate: $A{\cdot}I{=}0$, $I > 0$ and $X^T{\cdot}M_0 \geq X^T.D_p.OUT.I$. Where A is the incidence matrix of the Petri net, X is one of its place invariants, M_0 is the initial marking, OUT is the output function, I is a vector representing the firing frequencies of the transitions and D_p is the delay times associated to the places.

When delay times are associated to the transitions, the above mentioned relationships is written as $A.I{=}0$, $I{>}0$ and $X^T.M_0{=}X^T.IN.D_T.I$. These conditions for a system to function at its maximum rate are explained using the example illustrated in Figure 3.43.

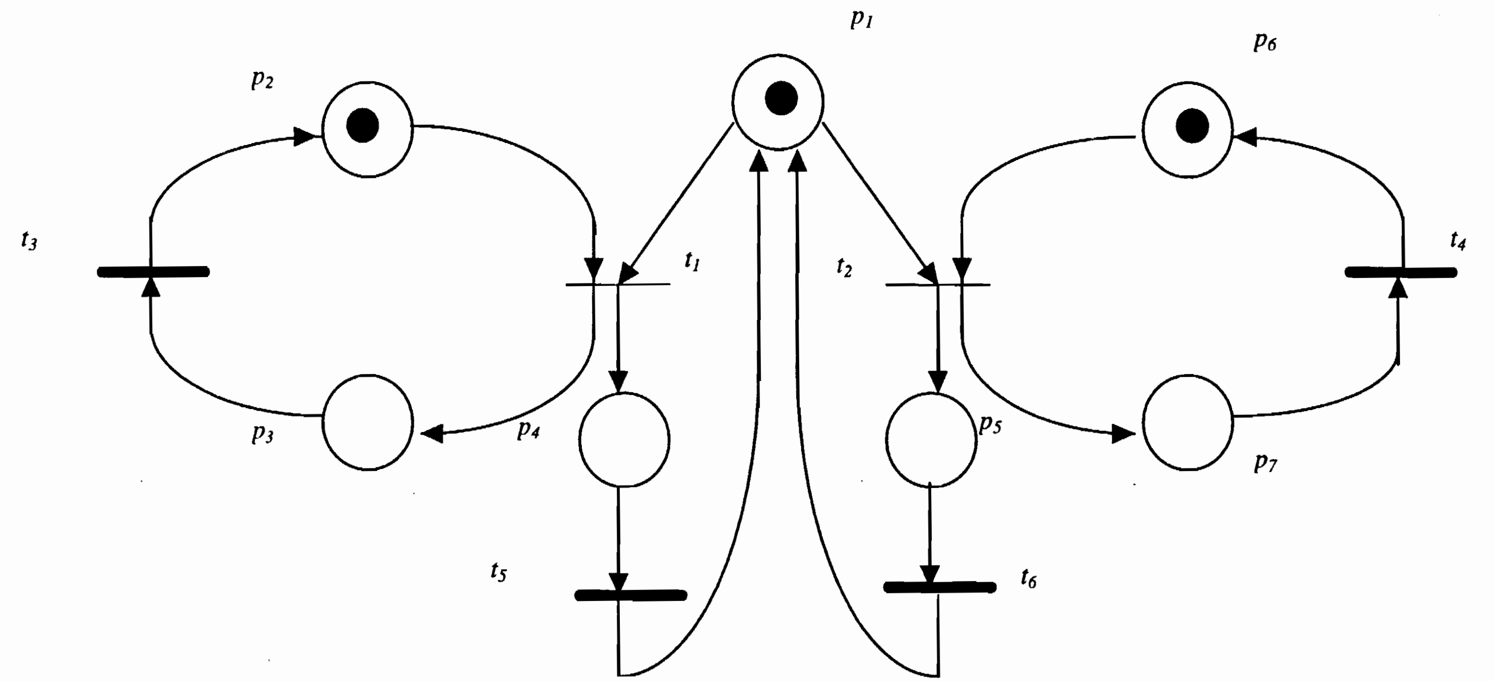


Figure 3.43 An illustration of a Petri net which is not a decision free net

The incidence matrix A and the input function of the Petri net are given below.

$$A = \begin{pmatrix} -1 & -1 & 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{pmatrix} \text{ and the input function is, } IN = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

If X is a place invariant, then $X^T A=0$. The place invariants for the above Petri net are [0 1 1 0 0 0 0], [1 0 0 1 1 0 0] and [0 0 0 0 0 1 1]. These place invariants are illustrated in Figure 3.44.

The next step is to compute the firing rates of each transition. The firing rate vector I should satisfy the following conditions: The first condition is that $A.I=0$ and the next is that $I > 0$. If $I=[\ I_1\ I_2\ I_3\ I_4\ I_5\ I_6\ ]$, then it can be easily show that $I_1=I_2=I_3=I_4=I_5=I_6$.

The matrix of delays associated to the transitions is given below.

$$D_T = \begin{pmatrix} d_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & d_6 \end{pmatrix}.$$

Making use of the above information the conditions for the system to function at its maximum operating rate can be written as: $n_1=I_1.d_1+I_3.d_3$, $n_2=I_1.d_1+I_2.d_2+I_5.d_5+I_6.d_6$ and $n_3=I_2.d_2+I_4.d_4$. Since $I_1=I_2=I_3=I_4=I_5=I_6$, $n_1 /(d_1+d_2)=n_2 /(d_1+d_2+d_4+d_5)=n_3 /(d_2+d_4)$. The above expression provides the ratio between the number of resources represented by places p_1, p_2 and p_6 to operate at maximum rate.

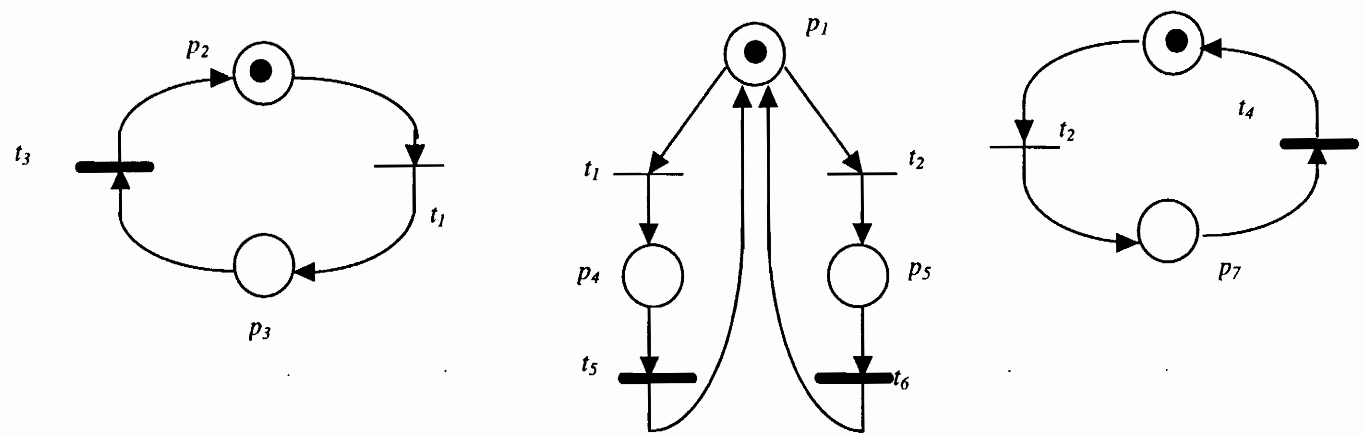


Figure 3.44 The place invariants of the above Petri net

3.13.2 STOCHASTIC PETRI NETS

The second group of timed Petri nets is known as stochastic Petri nets. In 1982 Molly [37] presented a method of evaluating the performance measures of systems with transitions exponentially distributed times. The reachability graph of the Petri net is used to define the different possible states and develop the Markov chain. The steady state probabilities of the system are then computed using the Markov chain. The draw back in the above procedure is that the reachability tree can be very large and create difficulties in storing all the data.

3.13.2.1 GENERALIZED STOCHASTIC PETRI NETS

Generalized stochastic Petri nets (GSPN) were developed in order to reduce the size of the reachability tree. In the generalized stochastic Petri net models, the immediate transitions are given priority over the exponentially timed transitions. The immediate transitions fire in zero time. When multiple immediate transitions are enabled they are allowed to fire simultaneously. Another important concept in GSPN models is the tangible and vanishing markings. In a tangible marking, only the exponentially distributed timed transition are enabled. A marking with at least one immediate transition enabled is known as a vanishing marking. The Markov chain after all the vanishing markings are removed is known as the reduced embedded Markov chain. The reduced embedded Markov chain is used for the computation of the steady state probabilities. Another useful feature in the GSPN models is the inhibitor arcs. The inhibitor arcs prevent transitions from firing when certain conditions are true. The inhibitor arcs avoids the inclusion of multiple places and transitions in representing certain conditions in a Petri net model. Inclusion of the above features facilitate the reduction of computational efforts to a great extent.

3.13.2.2 EXPONENTIAL AND DETERMINISTIC TIMED PETRI NETS

Exponentially distributed and deterministic timed Petri nets Ajmone and Choila

[3], are useful in analyzing systems in which the time taken for some of the events are exponentially distributed and the others are deterministic. However, the limitation of such Petri nets is that the performance evaluation of such systems are possible if only one event whose duration is deterministic can take place at any particular time.

This form of Petri net models have been successfully used in traffic control systems. The arrival of vehicles at an intersection are assumed to be exponentially distributed and the time taken for them to clear the intersection is deterministic. Further, only one set of vehicles will be using the intersection at any particular time. Therefore, this situation fits perfectly to be modeled as a exponentially and deterministic distributed timed Petri nets.

3.13.2.3 NON EXPONENTIAL TIMED PETRI NETS

The third group of Petri nets are those systems whose probability distribution need not be exponential or deterministic Guo, DiCesare, and Zhou [25]. Not much research has been performed in this field. The most common approach is to develop a simulation code using the Petri net model and then derive the results from the simulation runs. As indicated earlier most of the Petri net models of AGV systems in the literature use simulation runs for their analysis.

4. THE MODELING METHODOLOGY

One of the major objectives of this thesis is to develop a systematic modeling procedure for AGV systems using Petri nets. This chapter focuses on this modeling procedure in detail. The Petri net is constructed using three sub nets representing three major components of an AGV system. The three major components considered in modeling the AGV system are, the vehicles and the guided path, the information flow related to the demand for AGVs between the work stations and the control system. The control system avoids traffic congestion and collision of the vehicles. The guide path and the movement of AGVs are modeled using a sub net known as the transportation sub net. The information flow related to the demand for AGVs between the work stations is modeled using the demand sub net. The control sub net models the control system that facilitates the flow of the vehicles avoiding congestion and collisions. The three sub nets are constructed in such a way that they can be merged without much complication in forming the final net. The following sections describe the three sub nets and the process of merging them.

4.1 THE TRANSPORTATION SUB NET

AGVs move between the work stations along a guide path. The transportation sub net models the movement of AGVs along this guide path. In the transportation sub net, the work stations are represented by places. The movement of an AGV between two work stations is represented by a transition between these two places representing the two work stations. Figure 4.1 illustrates two work stations and the movement of an AGV between them. The Petri net representation of this simple system is shown in Figure 4.2.

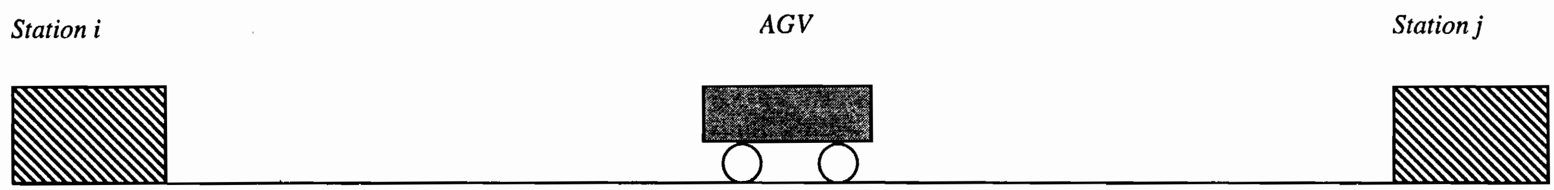


Figure 4.1 A simple example of the movement of an AGV between two work stations

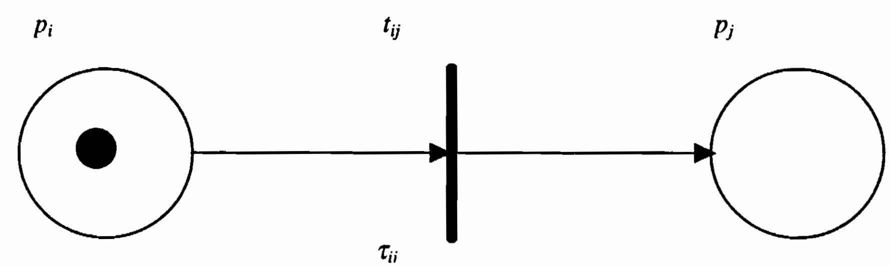


Figure 4.2 The Petri net representation of the movement of an AGV between two work stations

Places p_i and p_j model the two work stations i and j. Transition t_{ij} represents the movement of an AGV from station i to station j. The token residing in place p_i indicates that the AGV is at work station i. Time τ_{ij} associated to transition t_{ij} is the time taken for the AGV to move from work station i to work station j. Since transition t_{ij} is a timed transition with its firing time equal to t_{ij}, a token resides in its input place p_i for a time period τ_{ij} after that transition being enabled. This represents the time taken for the AGV to reach work station j after beginning to move away from work station i.

An interesting property of any transition t_{ij} in the transportation sub net is that, it has one and only one input place (i.e. p_i) and one and only one output place (i.e. p_j). The input place of the transition models the work station from where the AGV leaves and the output place represents the work station to which the AGV moves. Since each transition of the transportation sub net has one and only one input place and one and only one output place, it is in a special class of Petri nets known as state machines.

4.1.1 TRANSPORTATION SUB NET OF A SINGLE LOOP SYSTEM

In the previous section, the procedure of constructing the Petri net representation of the movement of an AGV between two work stations was explained. In this section, the above procedure is used to construct the transportation sub net for a single loop AGV system. In a single loop AGV system, the AGVs move along a single route. Therefore, the preceding and succeeding stations of any work station is known and fixed. If for example, $.....\rightarrow i\rightarrow j\rightarrow k\rightarrow.....$ is a portion of the route of the single loop AGV system, then, station i is the only preceding station of j. Similarly, station k is its only succeeding station.

This property has good relevance to the representation of the transportation sub net of the single loop AGV system. For example, if place p_j represents work station j, t_{ij} is the only input transition of p_j. Similarly, since station k is the only succeeding station of j, the only output transition of p_j is t_{jk}. Therefore, every place of the transportation sub net of a single loop AGV system has one and only one input transition and one and only one output transition. A Petri net with each place having only one input and only one output transition is a marked graph. It was shown in the previous chapter that the transportation sub net of any AGV system with predefined routes is a state machine. Therefore, the transportation sub net of a single loop AGV system is a marked graph as well as a state machine.

An example of a single loop AGV system is illustrated in Figure 4.3. The AGVs move along the guide path from work station 1 to work station 2, then from work station 2 to work station 3 and then return to work station 1 from work station 3. Therefore, the route followed by the AGVs is $(1 \rightarrow 2 \rightarrow 3 \rightarrow 1)$. The transportation sub net modeling the flow of AGVs in the above system is given in Figure 4.4. Places p_1, p_2 and p_3 represent the three work stations and transitions t_{12}, t_{23} and t_{31} model the movement of AGVs between the three work stations.

The time necessary for the AGVs spend between the work stations are associated to the transitions. In this example τ_{12}, τ_{23}, τ_{31} are the times the AGVs spend between work stations $(1,2)$, $(2,3)$ and $(3,1)$ respectively. These times include the time for the AGVs to travel between the work stations and the loading and unloading times. The sum of the times associated to the transitions is the time taken for an AGV to complete the route. The number of tokens in the places of the transportation sub net represents the number of AGVs in the system.

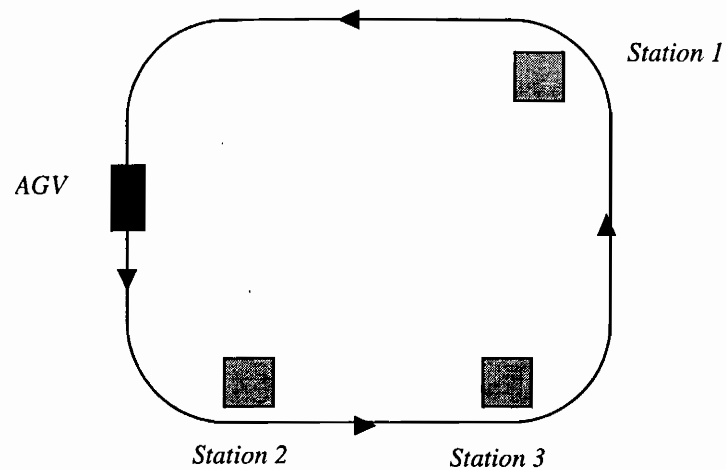


Figure 4.3 An example of a single loop AGV system

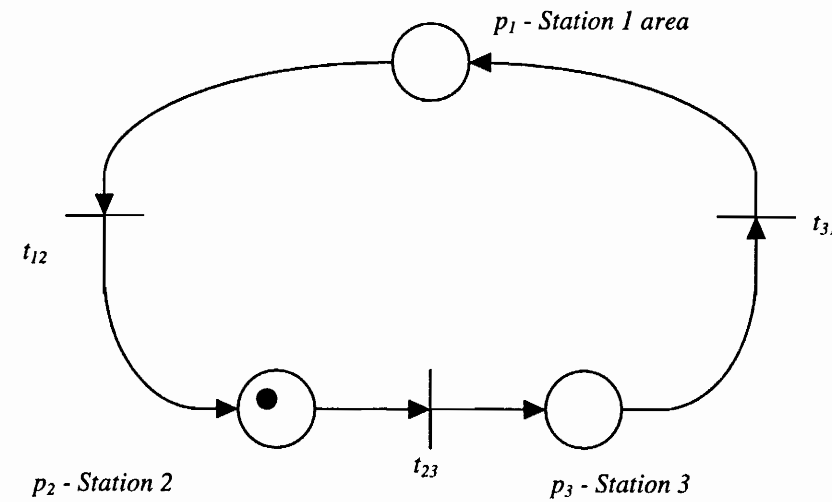


Figure 4.4 The transportation sub net for the single loop AGV system and a single route 1→2→3

4.1.2 TRANSPORTATION SUB NET OF A TWO LOOP SYSTEM

In this section, the construction of two looped AGV system is considered. As shown earlier, the route followed by an AGV in a single loop system is unique. However, when there is more than one loop, the number of routes in the system is also more than one. It was shown in the previous section that the transportation sub net of a single route AGV system is a marked graph. However, when there is more than one route in an AGV system, its transportation sub net cannot be represented by a marked graph. For example, let $.....\rightarrow i\rightarrow j\rightarrow k\rightarrow ....$ and $.....\rightarrow i\rightarrow l\rightarrow m\rightarrow .....$ be two portions of the routes of a two loop AGV system. In the above case, when an AGV is at work station i it can either move to work station j or work station l. Therefore, if place p_i is used to represent work station i in the transportation sub net for the above system, its output transitions can either be t_{ij} or t_{il}. Since a place of the transportation sub net can have more than one output transition, the transportation sub net is not a marked graph. A simple AGV system with two loops is illustrated in Figure 4.5. It is a system with a provision for two different routes. The two routes in this layout are $(1\rightarrow 2\rightarrow 1)$ and $(1\rightarrow 3\rightarrow 1)$. The transportation sub net of the two routes is illustrated in Figures 4.6. Places p_1, p_2 and p_3 represents work stations 1, 2 and 3. Transitions t_{12}, t_{13}, t_{21} and t_{31} model the movement of the AGVs between work stations $(1,2)$, $(1,3)$, $(2,1)$ and $(3,1)$. When an AGV is at work station 1, it can either move to work station 2 or work station 3. Therefore, place p_1 has two output transitions t_{12}, t_{13}. However, places p_2 and p_3 have only one output transition each and those are t_{21} and t_{31} respectively. This represents the situation that, when an AGV is at work station 1, its destination can either be work station 2 or 3. However, when it is at either work station 2 or work station 3, its destination is fixed and it is work station 1.

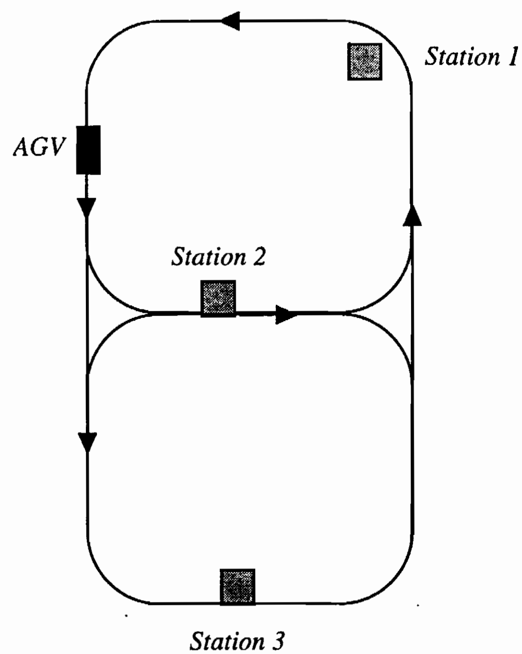


Figure 4.5 An illustration of a two loop AGV system with two different routes

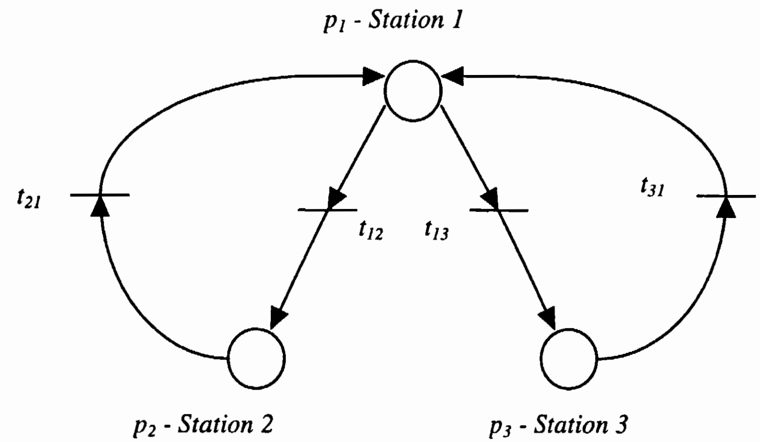


Figure 4.6 The transportation sub net of the two loop AGV system with two routes

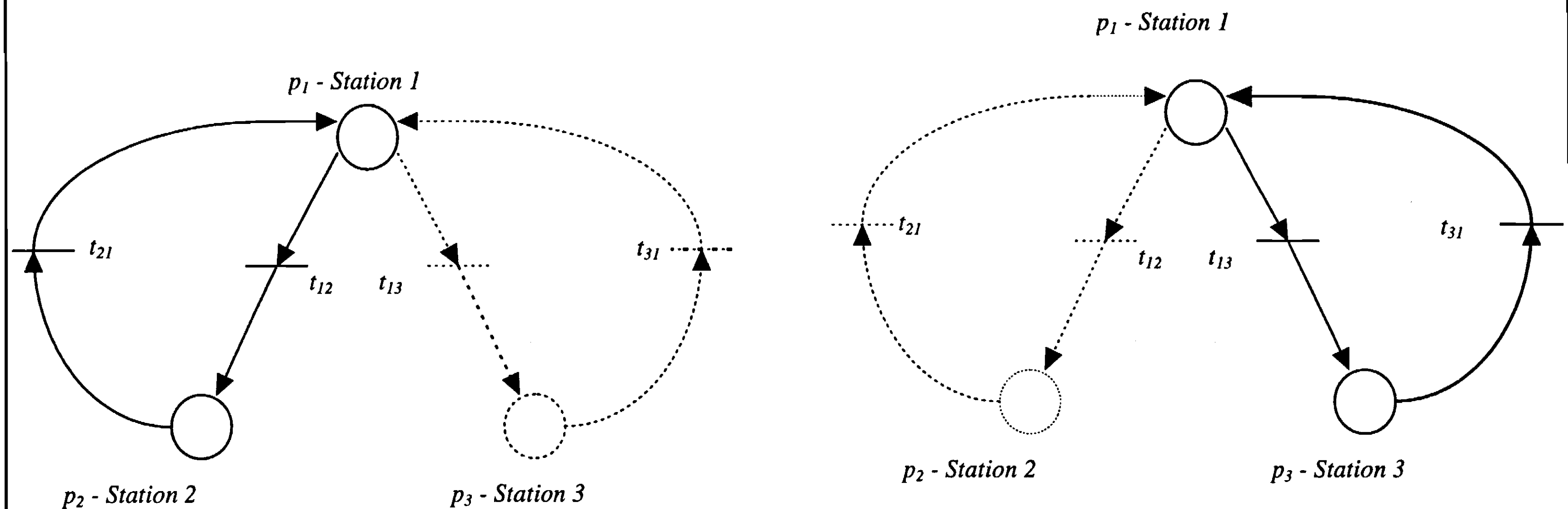


Figure 4.7 The portions of the transportation sub net showing the two routes

4.1.3 TRANSPORTATION SUB NET OF A MULTI LOOP SYSTEM

This section introduces the general form of the transportation sub net with multiple loops and multiple routes. The construction of the transportation sub net of a multi loop AGV system is similar to the construction of the two loop system discussed in the previous section. Initially, the work stations of the AGV system are modeled using the places. The places in this transportation sub net represent only the work stations. Then, the destination work station for each work station is identified using the predefined routes. Each place representing a work station is then connected to the places representing its destination work station through the transition modeling the movement of the AGV between the two work stations. A multi loop AGV system is illustrated in Figure 4.8. This AGV system consists of five work stations and four different routes. The four routes are $(1 \rightarrow 2 \rightarrow 3 \rightarrow 1)$, $(1 \rightarrow 2 \rightarrow 5 \rightarrow 1)$, $(1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 1)$, and $(1 \rightarrow 5 \rightarrow 1)$. Figure 4.9 illustrates the transportation sub net of the AGV system. The five work stations are represented by places p_1, p_2, p_3, p_4 and p_5. The eight transitions of the net modeling the flow of AGVs between the work stations are t_{12}, t_{15}, t_{23}, t_{24}, t_{45}, t_{25}, t_{31}, and t_{51}. The thick continuous lines of Figure 4.10 illustrate routes $(1 \rightarrow 2 \rightarrow 3 \rightarrow 1)$ and $(1 \rightarrow 5 \rightarrow 1)$.respectively.

A notable feature in the routes defined in the AGV systems considered so far is that, each route begins at a particular work station and ends in the same work station. Therefore, each route can be represented by a circuit in the transportation sub net. For example, the number of circuits in the AGV system considered earlier is four. These four circuits are $(p_1\, t_{12}\, p_2\, t_{23}\, p_3\, t_{31}\, p_1)$, $(p_1\, t_{12}\, p_2\, t_{25}\, p_5\, t_{51}\, p_1)$, $(p_1\, t_{12}\, p_2\, t_{24}\, p_4\, t_{45}\, p_5\, t_{51}\, p_1)$ and $(p_1\, t_{15}\, p_5\, t_{51}\, p_1)$. The number of tokens in the transportation sub net represent the number of AGVs in the system and the number of tokens in each circuit represent the number of AGVs moving along each route.

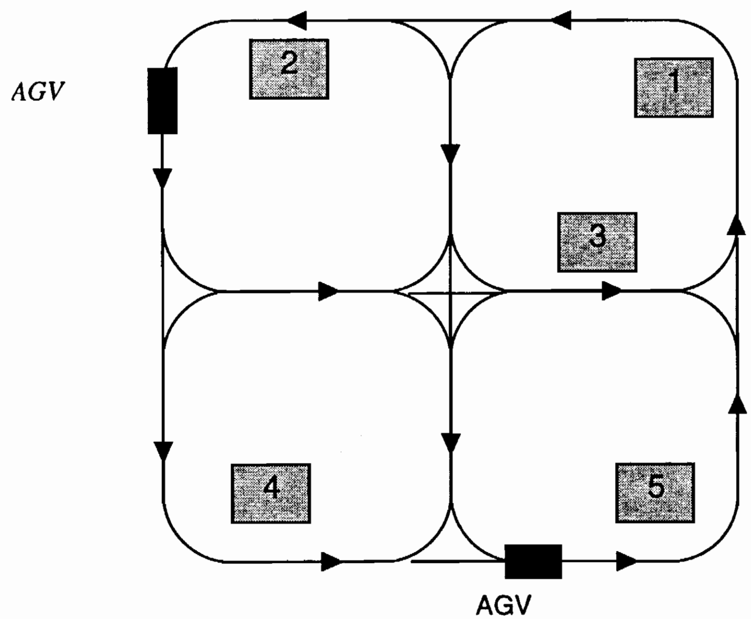


Figure 4.8 An illustration of an AGV system with five work stations

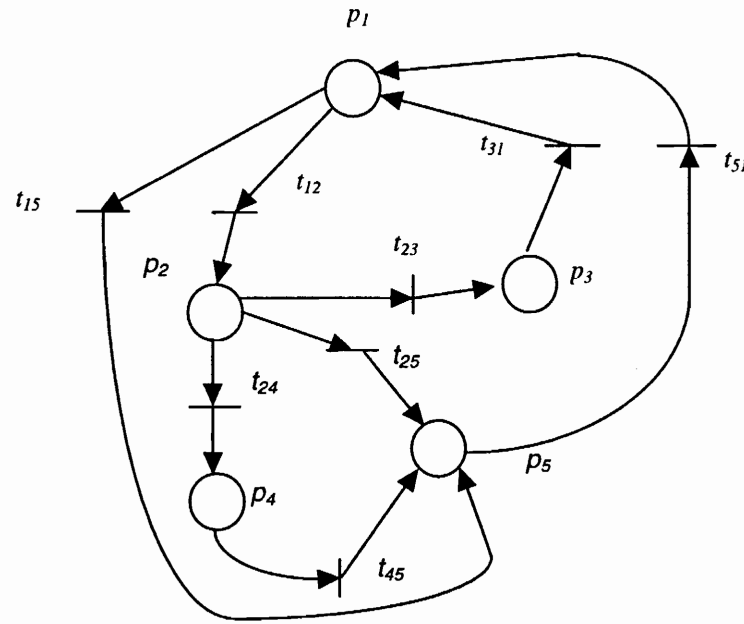


Figure 4.9 A transportation sub net of four routes

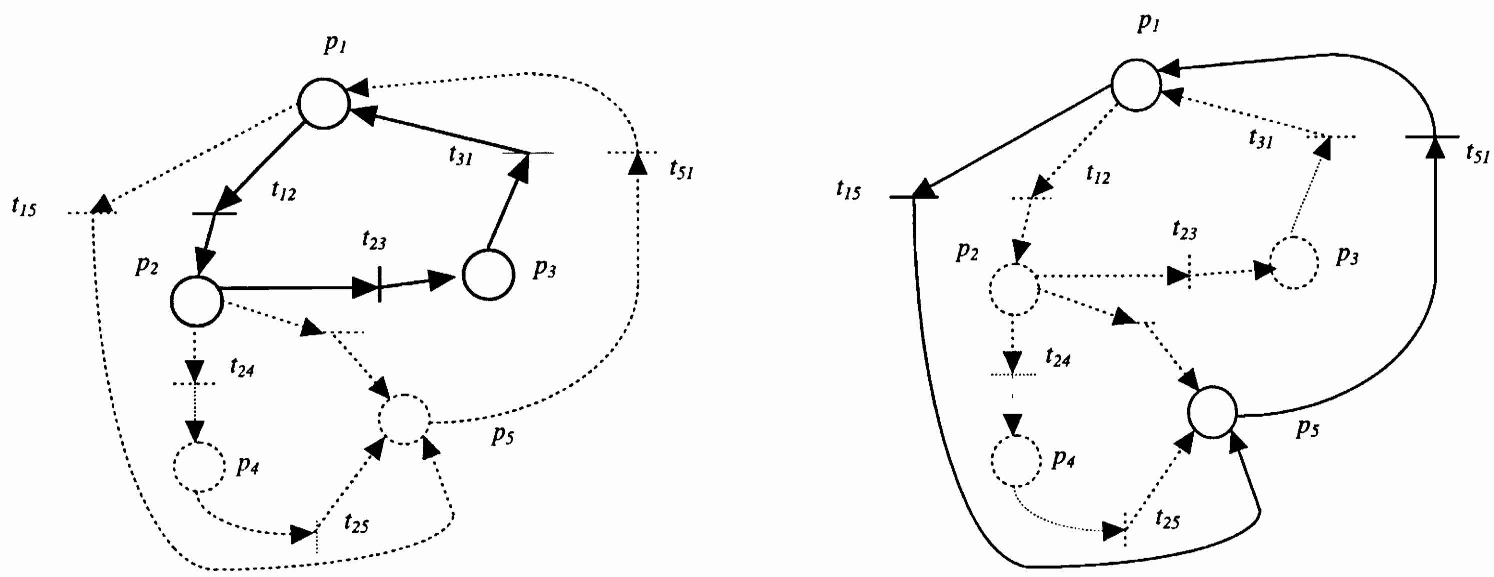


Figure 4.10 The portion of the transportation sub net sowing routes 1→2→3→1 and 1→5→1

4.1.4 VALIDATION OF THE PROPERTIES OF THE TRANSPORTATION SUB NET

The transportation sub net of an AGV system described earlier is proved to be bounded, live and reversible in this section. The proof is based on the fact that the transportation sub net is always a state machine for an AGV system with predefined routes. It is easy to show that a state machine is bounded, live and reversible as long as it is marked with a finite number of tokens and is strongly connected. In a state machine, each transition consists of exactly one input place and exactly one output place. Therefore, the firing of any transition in a state machine will remove exactly one token from its input place and place exactly one token in its output place. Since each firing of the transition in a state machine removes a token and creates another token in the net, the total number of tokens remain to be the same. Therefore, the total number of tokens in a state machine remains unchanged in any marking reachable from the initial marking. Hence, a state machine is bounded as long as it is marked with a finite number of tokens. A Petri net is said to be strongly connected, if there is a directed path from each node (a place or a transition) to any other node in the net. When a state machine is strongly connected and is marked with at least one token, there should be a firing sequence in any marking which enables any of its transitions. Therefore a state machine which is strongly connected and marked with at least one token is live. Further, when the state machine is strongly connected, there should exist a firing sequence which brings its marking back to its original marking. Therefore a strongly connected state machine marked with at least one token is reversible as well.

Now, the tokens in the transportation sub net represent the AGVs. Therefore, the transportation sub net should consist of a finite number of tokens. Further, each route defined in the AGV system considered in this study starts from a work station and ends in the same work station. Therefore, there exists a path between every pair of nodes of the transportation sub net. Hence, the transportation sub net is a strongly connected state machine. Since a transportation sub net is strongly connected and is marked with at least one token it is bounded, live and reversible.

4.2 THE DEMAND SUB NET

The demand sub net models the flow of information related to the demand for AGVs between the work stations. An AGV is dispatched from one work station to another work station only when there is a demand between them. For each pair of work stations, a demand sub net is defined which represents the requirement of AGVs between them. A demand sub net consists of two places and two transitions. The first place models the availability of the demand for an AGV between the two work stations. A token residing that place indicates that there is a demand between the two work stations. The next place models the unavailability of the demand between the two work stations. A token residing in this place indicates that there is no demand for an AGV between the two work stations. The input transition to the place modeling the demand for an AGV represents the beginning of the demand for an AGV between the two work stations. In this study, it is assumed that the time between consecutive demands between any two work stations is known and is fixed. The time associated to this transition models the time between the consecutive demands for an AGV between them.

A demand sub net is illustrated in Figure 4.11. The demand between work stations i and j is modeled using places p_{ijd} and p_{nijd} and two transitions t_{ijs} and t_{ijd}. A token residing in place p_{ijd} indicates that there is a demand for an AGV from work station i to j. Transition t_{ijs} models the beginning of the movement of the

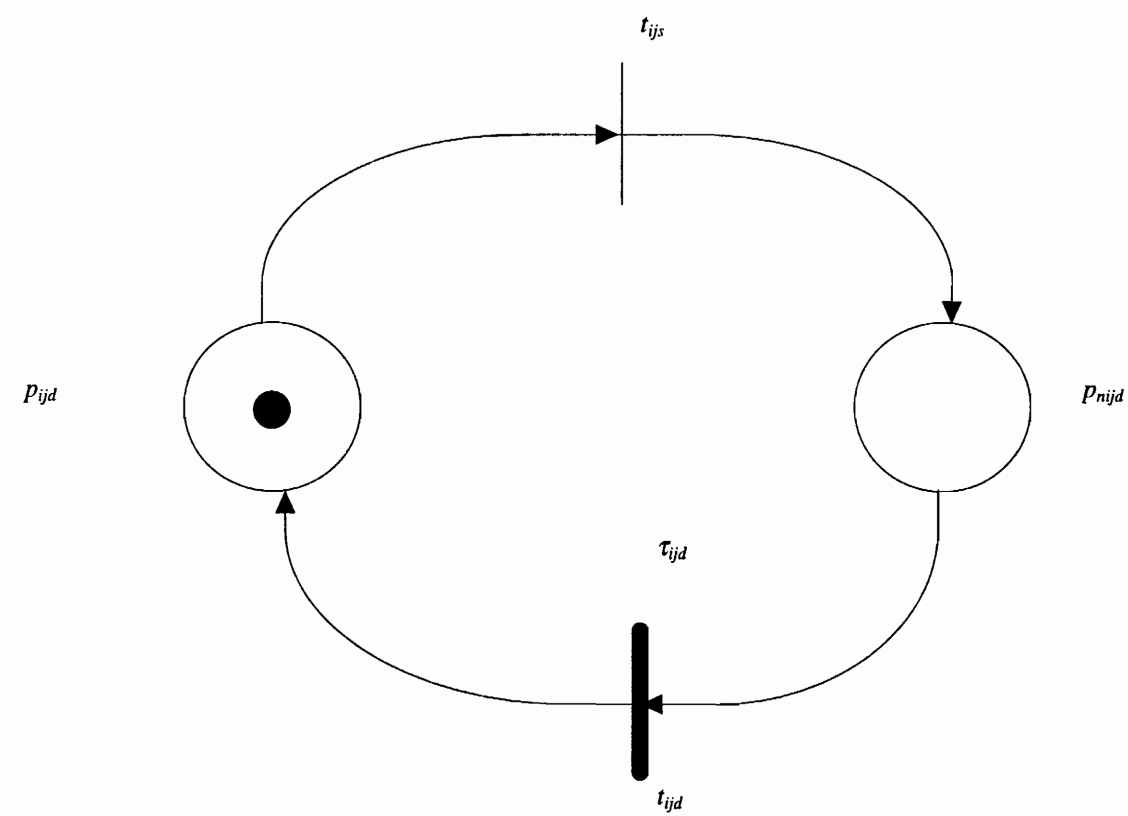


Figure 4.11 The demand sub net modeling the demand for an AGV from station i to station j

AGV from station i to station j to meet the required demand. The AGV moves from work station i to work station j as soon as there is a demand between them. Therefore, the firing of transition t_{ijs} is instantaneous and the time associated to it is zero. Once transition t_{ijs} fires, a token is deposited in place p_{nijd} indicating that a demand for an AGV from i to j is awaited. Transition t_{ijd} represents the beginning of the demand for an AGV from work station i to j. There is a time associated with the firing of transition t_{ijd}. This is the time between consecutive demands for an AGV between work stations i and j.

4.2.1 MERGING THE TRANSPORTATION AND DEMAND SUB NETS

This section focuses on the method of merging the transportation and demand sub nets. A transition in the transportation sub nets represents the movement of an AGV between two work stations. For example the transition t_{ij} illustrated in Figure 4.12 represents the movement of an AGV between work station i and j. It is not possible to merge the transportation and demand sub nets through this transition and describe the logical behavior of the system. In the procedure introduced in this study, each transition of the transportation sub net is replaced with two transitions and a place to facilitate the merger. The transition of the transportation sub net and the sub net replacing that transition are shown in Figures 4.12 and 4.13. The transition in the shaded area of Figure 4.12 is replaced with the sub net in the shaded area of Figure 4.13. In Figure 4.13, transition t_{ijs} models the beginning of the movement of an AGV from work station i to work station j. The end of the movement of the AGV between the two work stations is represented by transition t_{ije}. An AGV can begin its movement from a work station to another work station only if the following two conditions are satisfied. The first is that an AGV should be available at the origin work station and the second is that there should be a demand for an AGV between the origin and destination work stations.

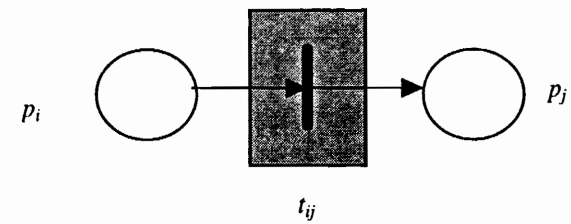


Figure 4.12 A portion of the transportation sub net modeling the movement of an AGV from station i to station j

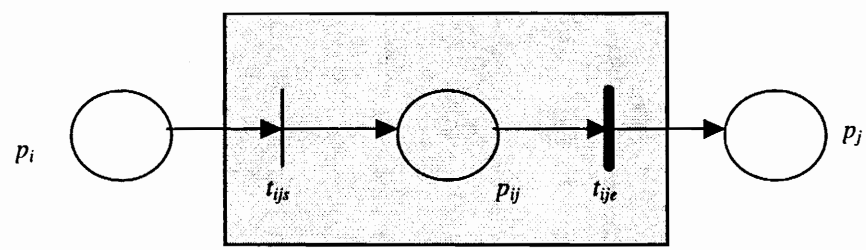


Figure 4.13 A portion of the transportation sub net modified to facilitate its merger with the demand sub net

Therefore, the beginning of the movement of an AGV from work station i to work station j is the result of the sychronization of the availability of an AGV at work station i and the availability of a demand between work stations i and j. Therefore transition t_{ijs} should have two input places namely p_i and p_{ijd} representing the availability of an AGV at work station i and the availability of a demand between work stations i and j. Figure 4.14 illustrates the Petri net representation of this synchronization.

The output place of transition t_{ijs} is p_{ij} and this place represents the actual movement of the AGV between the two work stations. The output place of transition t_{ijs} is p_{nijd} this place represents the end of the demand between the two work stations. When transition t_{ijs} fires, tokens will be deposited in places p_{ij} and p_{nijd}. Tokens residing in these two places indicate that an AGV is moving from work station i to j and the demand between the work stations is satisfied and a new demand is awaited.

The output transition of place p_{ij} is transition t_{ije}. This transition models the end of the movement of an AGV between work stations i and j. The time taken for the AGV to move between work stations i and j is associated with this transition. When transition t_{ije} fires, a token is deposited in place p_j indicating that the AGV has reached work station j.

Similarly the output transition of place p_{nijd} is transition t_{ijd}. This transition models the beginning of the demand between the two work stations. The time associated to this transition represents the time between consecutive demands for an AGV between work stations i and j. When transition t_{ijd} fires, a token is deposited in place p_{ijd} and it indicates that a demand for an AGV between the two work stations is available.

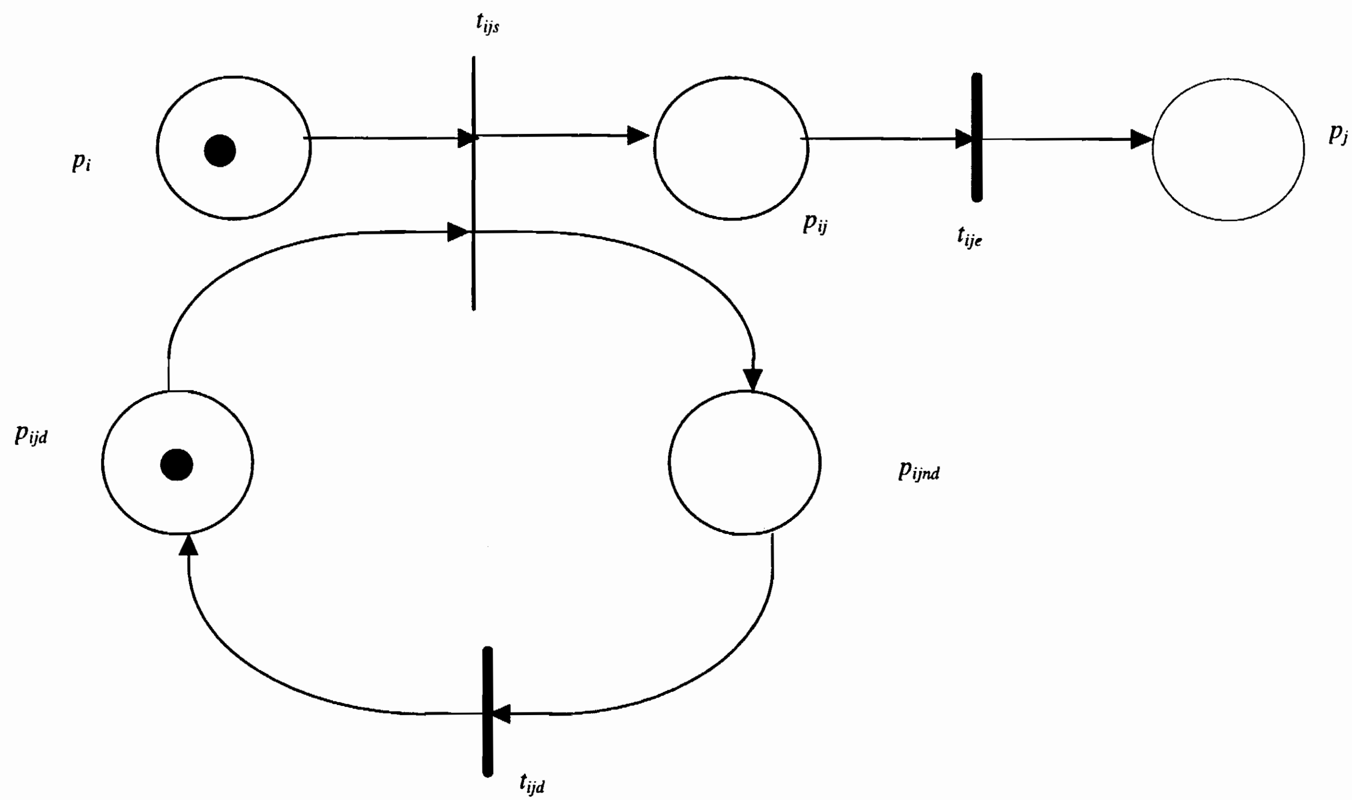


Figure 4.14 The merged transportation and demand sub net

4.2.2 VALIDATION OF THE PROPERTIES OF THE MERGED SUB NETS

The demand sub net simply consists of two places and two transitions between the two places. The number of tokens in any demand sub net is always one. Therefore it is easy see that it is bounded, live and reversible. This section, shows that the merged transportation and demand sub net is also bounded, live and reversible. The proof is based on the analysis of the firing of any transition t_{ij} and its equivalent sub net consisting of transitions t_{ijs}, t_{ije} and the place p_{ij} in the merged sub net. Figures 4.15 and 4.16 are used for this purpose.

In Figure 4.15, it is easy to see that the movement of a token from place p_i to place p_j is achieved through the firing of transition t_{ij}. This identical movement of tokens in the merged sub net in Figure 4.16 is attained through the consecutive firing of transitions t_{ijs} and t_{ije}. Further, transition t_{ijs} can fire again only after the firing of transition t_{ijd}. Therefore, the consecutive firing of transitions t_{ijs}, t_{ijd} and t_{ije} in the merged transportation and demand sub nets is equivalent to the firing of transition t_{ij} in the transportation sub net. In the previous section the transportation sub net was proved to be bounded, live and reversible. Hence, the merged transportation and demand sub net is also bounded, live and reversible.

Another property of interest related to the transitions of the merged transportation and demand sub nets are their firing rates. As shown earlier, once transition t_{ijs} fires, it can fire again only after the firing of transition t_{ijd}. Therefore the number of times transition t_{ijs} fires is equal to the number of times transition t_{ijd} fires in one cycle. Therefore, the firing rates of transitions t_{ijs} and t_{ijd} are equal. Using similar arguments it can be shown that the number of times transitions t_{ijs} and t_{ije} fire in one complete cycle are equal. Hence, the firing rates of transitions t_{ijs} and t_{ije} are equal.

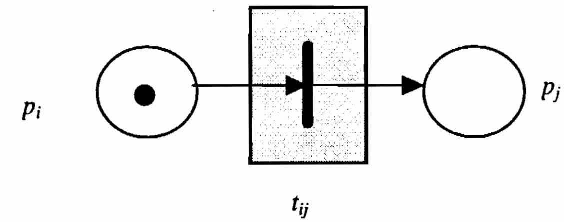


Figure 4.15 The portion of the transportation sub net

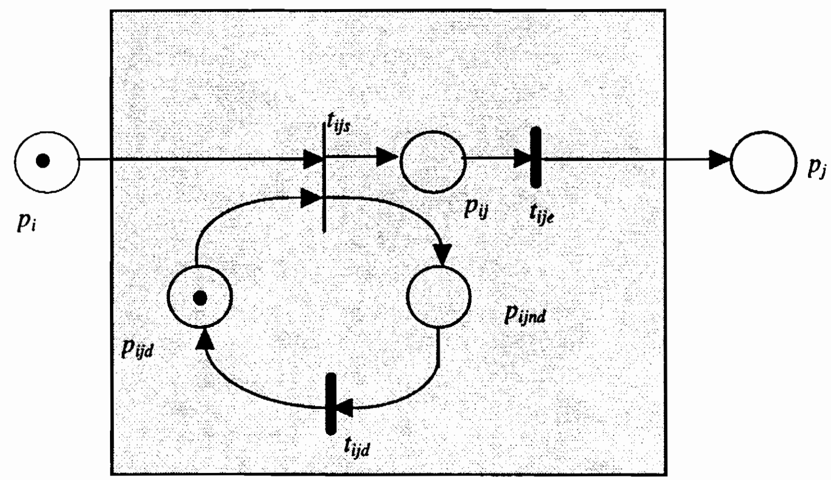


Figure 4.16 The merged transportation and demand sub net

4.3 THE CONTROL SUB NET

The control sub nets models the control zones that facilitate the movement of AGVs avoiding congestion and collision along the guide path. Each control zone is represented by a solitary place. A token residing in that place indicates that the control zone is free. It is necessary that the places modeling the control zones are merged with the transportation and demand sub nets to construct the final Petri net of the AGV system. This section illustrates the procedure of merging the places representing the control zones with the transportation and demand sub nets.

As discussed in the previous section, each transition of the transportation sub net was replaced with a sub net. This sub net consists of two transitions and a place. It facilitates the merger of the transportation sub net with the demand sub net. This section describes the procedure of expanding the transitions of the transportation sub net to facilitate the merger with the places representing the control zones.

Figure 4.17 illustrates how a transition of the transportation sub net is replaced with a sub net consisting of three transitions and two places to facilitate the merger with the demand and control sub nets. The top most figure illustrates places p_i, p_j representing work stations i and j and transition t_{ij} representing the movement of an AGV between the two work stations. The figure in the middle shows how transition t_{ij} is replaced with a sub net consisting of two transitions and a place to facilitate the merger with a demand sub net. The bottom figure shows how the place p_{ij} and t_{ije} are expanded and replaced with a sub net consisting of two places and two transitions to facilitate the merger with the control sub net.

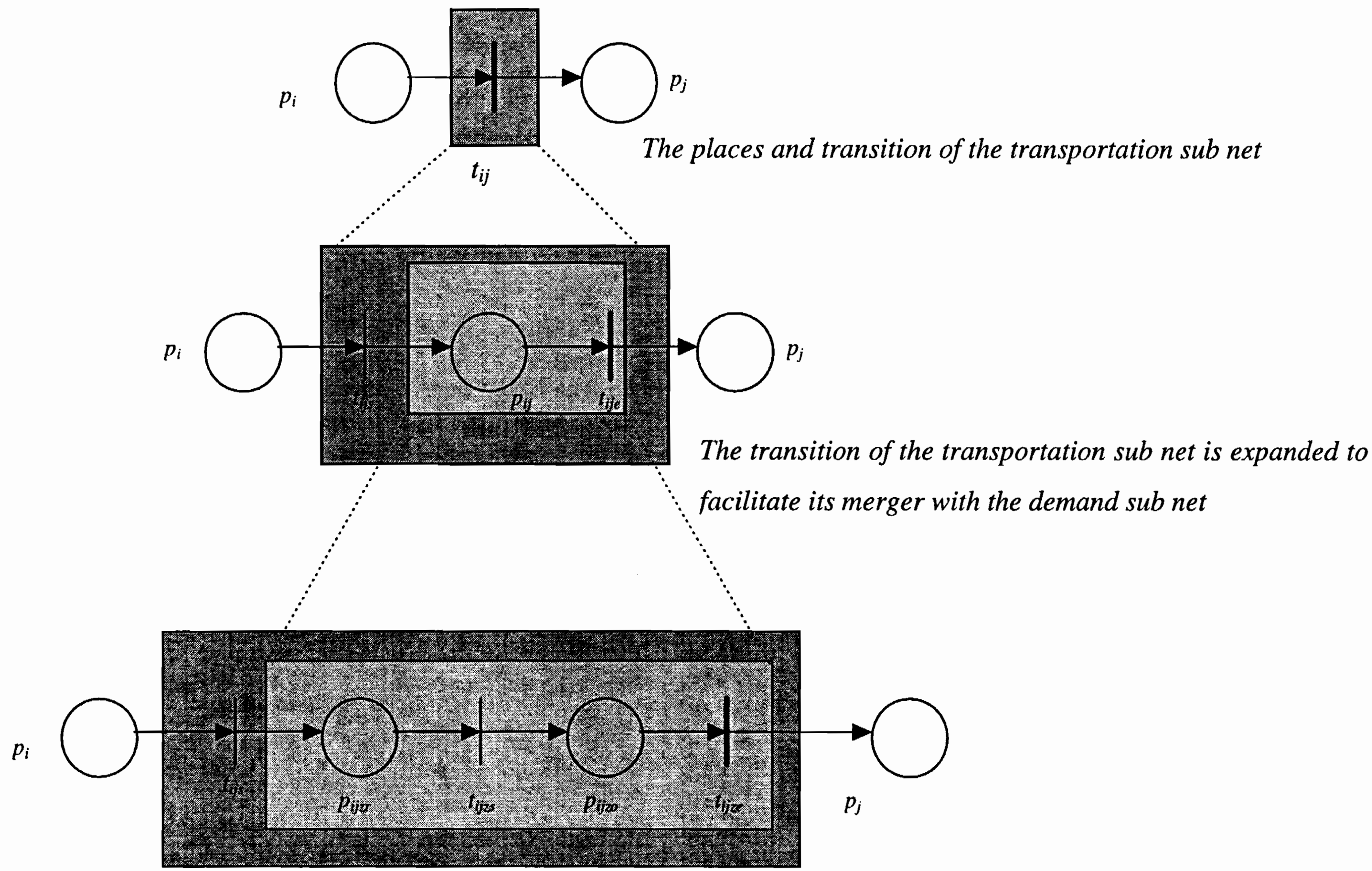


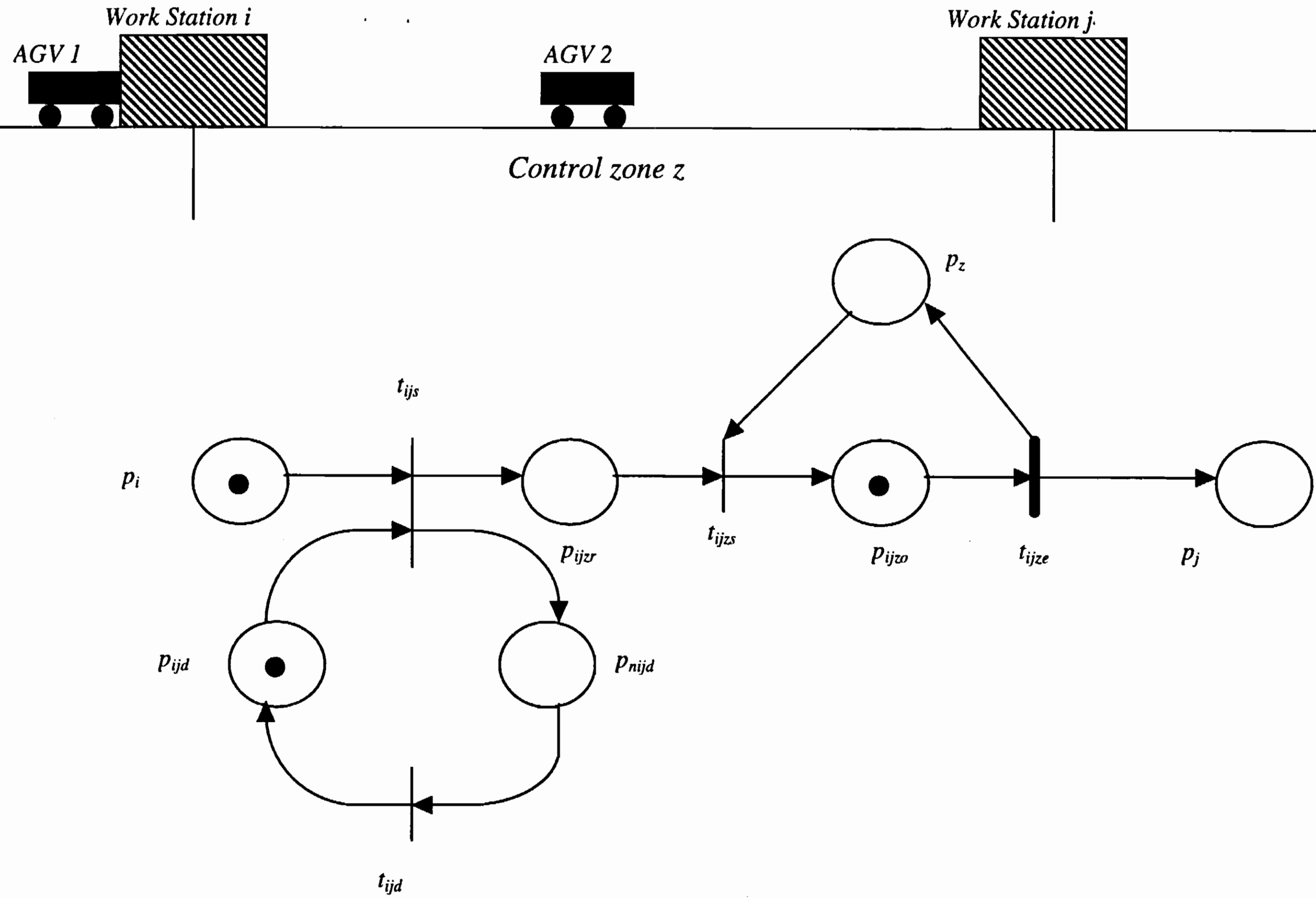
Figure 4.17 Illustration of the modification of the transportation sub net to merge the control sub net

The two places included in this sub net are p_{ijzr} and p_{ijzo} and the two transitions are t_{ijzs} and t_{ijze}. Place p_{ijzr} represents the condition that the AGV is ready to enter control zone z and p_{ijzo} indicates that the AGV is occupying that control zone. Transitions t_{ijzs} and t_{ijze} represent the beginning and the end of the movement of the AGV in the control zone. The firing of transition t_{ijzs} is the result of the synchronization of the availability of the control zone z and the availability of an AGV being ready to enter the control zone. Therefore transition t_{ijzs} has two input places namely p_{ijzr} and p_z. Transition t_{ijzs} is an immediate transition and no time is associated to it. Transition t_{ijze} represents the end of the movement of the AGV in the control zone and the time associated to this transition is the time taken for a vehicle to clear the control zone.

The design of control zones in an AGV system can be of different forms. The Petri net representations will vary according to the way in which the control zones are designed. In subsequent sections, the modeling procedures of the control zones according to their design are discussed. These sections present five different designs of control zones. The first design is one with control zones along a straight line between two work stations. The second is about a control zone located at the point where two lanes diverge The third design shows merger of two lanes and the fourth design is for intersecting lanes. The final design is a combination of merging and diverging lanes.

4.3.1 CONTROL ZONES ALONG A STRAIGHT LINE

The simplest form of a control zone is to have the zone between two work stations. A vehicle at one work station can enter a control zone only when that control zone is free. Figure 4.18 illustrates a situation where a control zone is placed between two work stations. In this example, the two work stations are work station i and work station j. Control zone z lies in between work stations i and j. One AGV is at work station i and another AGV occupies control zone z.



gure 4.18 The basic control zone design and the corresponding control sub net

In the Petri net representation of the above situation, place p_i indicates the availability of an AGV is at work station i and Place p_{ijd} represents the availability of a demand between work stations i and j. When an AGV is available at station i and there is a demand for it to move to work station j, transition t_{ijs} fires. When this transition fires a token is deposited in place p_{ijzr} indicating that the AGV is ready to enter the control zone z and a token is deposited in place p_{nijd} indicating that the demand for the AGV between the work stations is satisfied.

The AGV can enter the control zone only when the control zone is free. A token residing in place p_z indicates that the control zone is free. Transition t_{ijzs} models the beginning of the movement of the AGV into the control zone. The AGV moves into the control zone when it is free. Therefore, transition t_{ijzs} is an immediate transition and no time is associated to it. Transition t_{ijze} represents the end of the movement of the AGV in the control zone. The time associated to this transition t_{ijze} indicates the time necessary for the AGV to move thrgoughb and clear the control zone.

It may be necessary that more than one control zone is present between a pair of work stations along a straight line. The necessity for multiple zones may be due to the use of multiple number of AGVs to be present in the system. Figure 4.19 illustrates a system where two control zones are present between a pair of work stations. By having two control zones between the work stations, it is possible to have two AGVs traveling between the two work stations.

The Petri net representation of this system is similar to the one with a single control zone between a pair of work stations. In a Petri net representation of such a system, three different places are defined to represent the three different stages of an AGV in a control zone. Place p_{ijz1r} represents that an AGV is ready to enter control zone z_1 and place p_{ijz1o} represents that an AGV is occupying

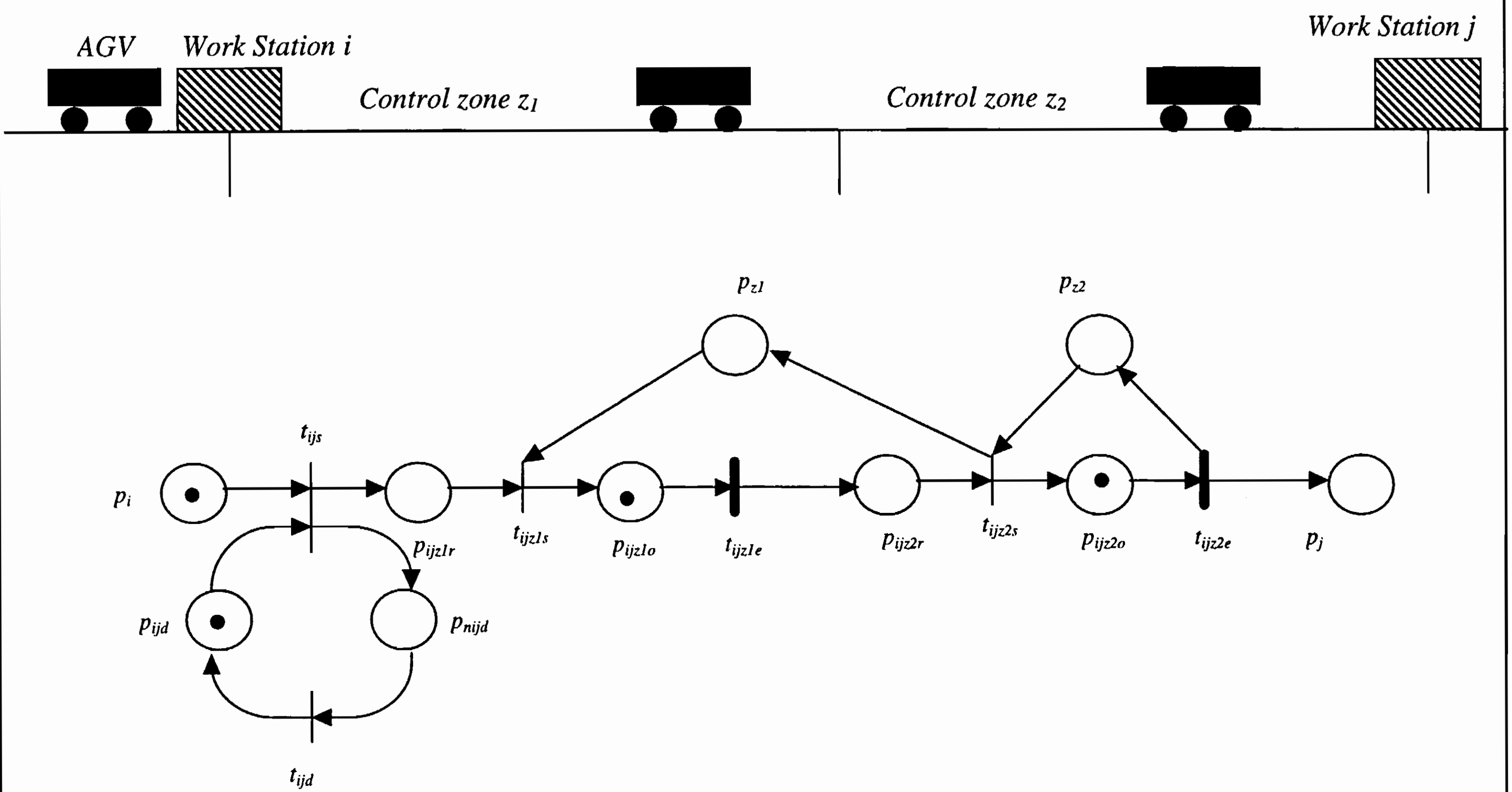


Figure 4.19 Illustration of a guide path layout with two control zones between a pair of work station and its Petri net representation.

control zone z_1. Place p_{ijz2r} indicates that the AGV is ready to enter control zone z_2.

Transition t_{ijz1s} models the beginning of the movement of an AGV into control zone z_1. An AGV can enter the control zone as long as the control zone is free and the AGV is ready to enter that control zone. Therefore, transition t_{ijz1s} has two input places namely p_{z1} and p_{ijz1r}. These two places represent the availability of the control zone and the availability of an AGV to enter the control zone. There is no time associated to this transition. Transition t_{ijz1e} represents the end of the movement of an AGV in control zone z_1. The time associated to this transition represents the time necessary for an AGV to clear that control zone.

4.3.2 CONTROL ZONE LAYOUT WITH DIVERGING LANES

Diverging lanes are necessary when AGVs from a work station move to different work stations located in different segments of the guide path. Figure 4.20 illustrates a situation where an AGV moves from one work station to two different work stations on two different segments of the guide path. The AGV moves straight along the guide path to go from work station i to j. While moving from i to k it makes a right turn at the point where the two lanes diverge.

While an AGV moves along the curved portions of the guide path, the speed of the vehicle is reduced to facilitate smooth turns. Therefore, a control zone at the point where the segments of the guide path diverge is necessary to avoid a faster moving trailing vehicle colliding with a slower vehicle that is on the curve. The portion of the layout consisting two diverging lanes is illustrated in Figure 4.20. The Petri net representation of this figure illustrates how the places modeling the control zones are merged with the transportation and demand sub nets.

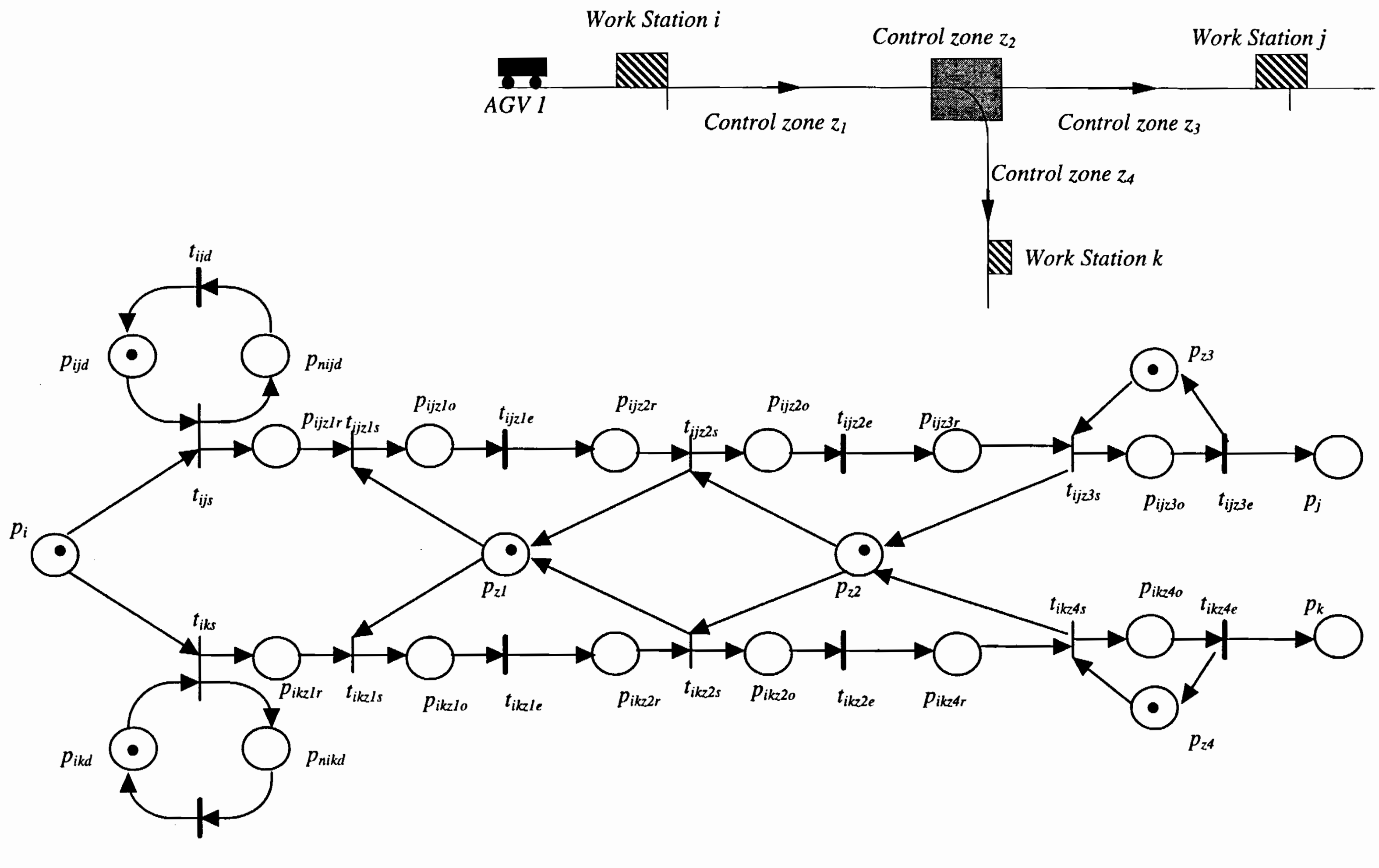


Figure 4.20 An illustration of diverging lanes and its Petri net representation

The AGV at work station i moves along path ij to reach work station j. Similarly it moves along path ik to reach work station k from work station i.

In Figure 4.20, place p_i represents work station i. The two arcs from place p_i along with the two transitions t_{ijz1s} and t_{ikz1s} represent the beginning of the movement of the AGV along paths ij and ik respectively. These transitions indicate that the paths ij and ik use control zone z_1 while moving along these two paths. Once an AGV moving along either path ij or ik clears control zone z_1, it enters control zone z_2. This is the control zone at the point where the two segments of the guide paths diverge. The control zones are represented by places p_{z1} and p_{z2}. Appropriate arcs are drawn from places p_{z1} to transitions t_{ijz1s}, t_{ikz1s}. Similarly, arcs are drawn from p_{z2} to t_{ijz2s}, and t_{ikz2s}.

Once an AGV moving along path ij clears control zone z_2, it enters control zone z_3. The AGV moving along path ik enters control zone z_4 from control zone z_2. Control zones z_3 and z_4 are represented by places p_{z3} and p_{z4}. The arcs from these places to transitions t_{ijz3s} and t_{ikz4s} indicate that the AGV moving along path ij enters control zone z_3 and the AGVs moving long path ik enters control zone z_4.

4.3.3 CONTROL ZONE LAYOUT CONSISTING OF MERGING LANES

In merging lanes AGVs move to a common work station from different work stations located at different segments of a guide path. An example of a segment of a guide path with merging lanes is illustrated in Figure 4.21. The Petri net representations containing the transportation sub net, demand sub net and the control zones are also represented in this figure. Control zones z_1 and z_2 are for the AGVs moving along paths ik and jk respectively. Places p_{z1} and p_{z2} model the two control zones.

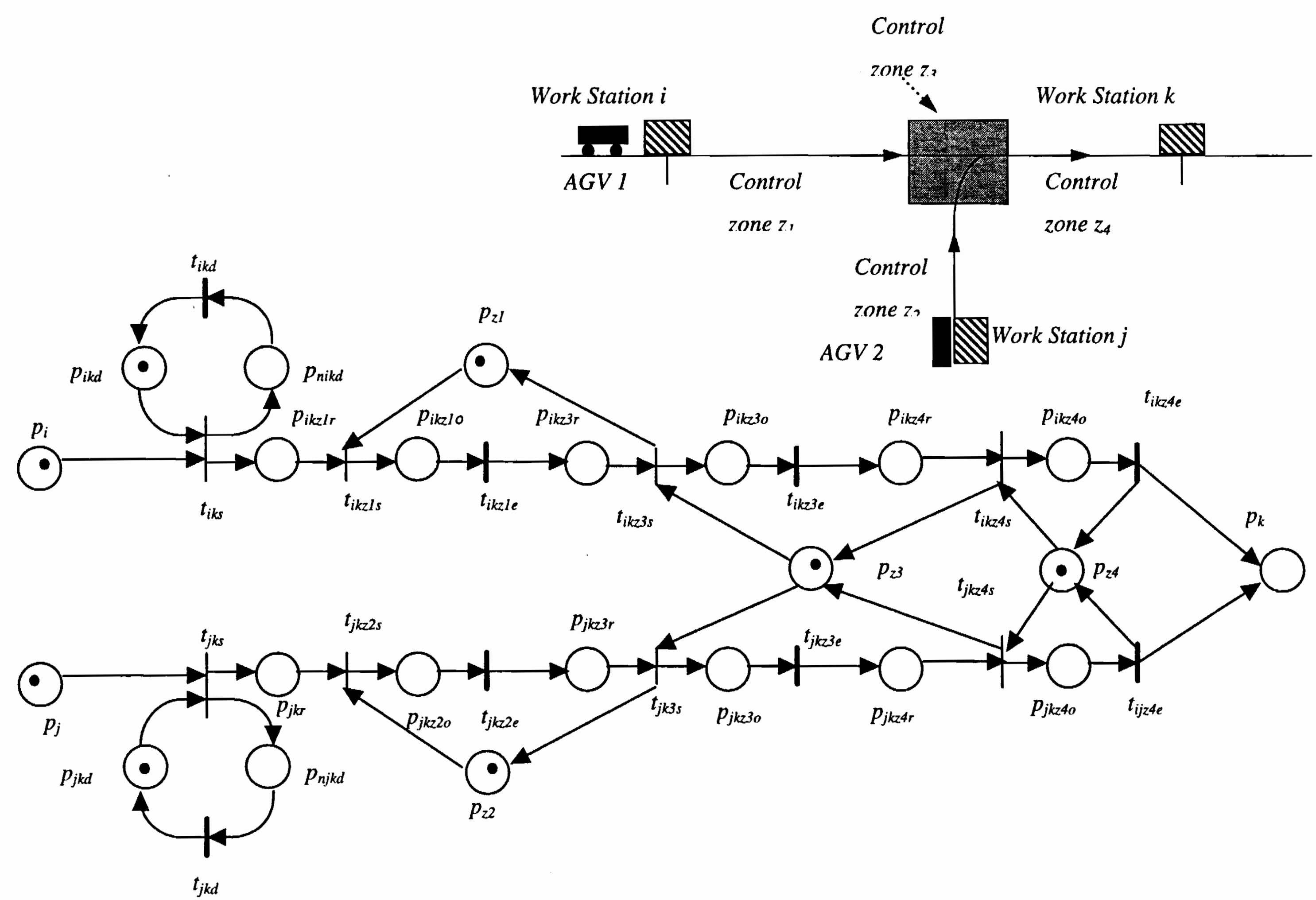


Figure 4.21 An illustration of merging lanes and its Petri net representation

Work station k is located on the merged portion of the guide path. The Petri net representation of this control design is illustrated in the same figure. The control zone at the intersection of the two merging lanes is z_3 and it is represented by place p_{z3}. The vehicles move along a common lane after clearing the intersection to reach work station k. Control zone z_4 on this common lane is represented by place p_{z4}.

Transitions t_{ikz1s} and t_{jkz2s} represent the beginning of the movement of the AGVs into control zone z_1 and z_2 respectively. Transition t_{ikz3s} and t_{jkz3s} represent the beginning of the movement of the AGVs traveling along paths ik and jk into the common control zone z_3 at the point where paths ik and jk merges. Similarly t_{ikz4s} and t_{jkz4s} represent the beginning of the movement of the vehicles along paths ik and jk into the common control zone z_4.

4.3.4 CONTROL ZONE WITH INTERSECTING LANES

In a pair of intersecting lanes, AGVs move along the segments of the guide path that are perpendicular to each other. The lanes cross at the point of intersection. Therefore, it is necessary that a control zone is present at the point of intersection of the lanes in order to avoid collisions of the vehicles. An example of an intersecting lane and its Petri net representation is illustrated in Figure 4.22. Places p_{z1} and p_{z2} represent the two control zones which lead into the point of intersection of two segments of a guide path. The control zone at the intersection of the two segments is z_3 and it is represented by place p_{z3}. AGV 1 moves through control zone z_1 along path il and enters control zone z_3. It enters control zone z_5 after clearing control zone z_3. Similarly, AGV 2 moving along path jk moves through control zone z_2 and enters control zone z_3. After clearing control zone z_3, this AGV enters control zone z_4.

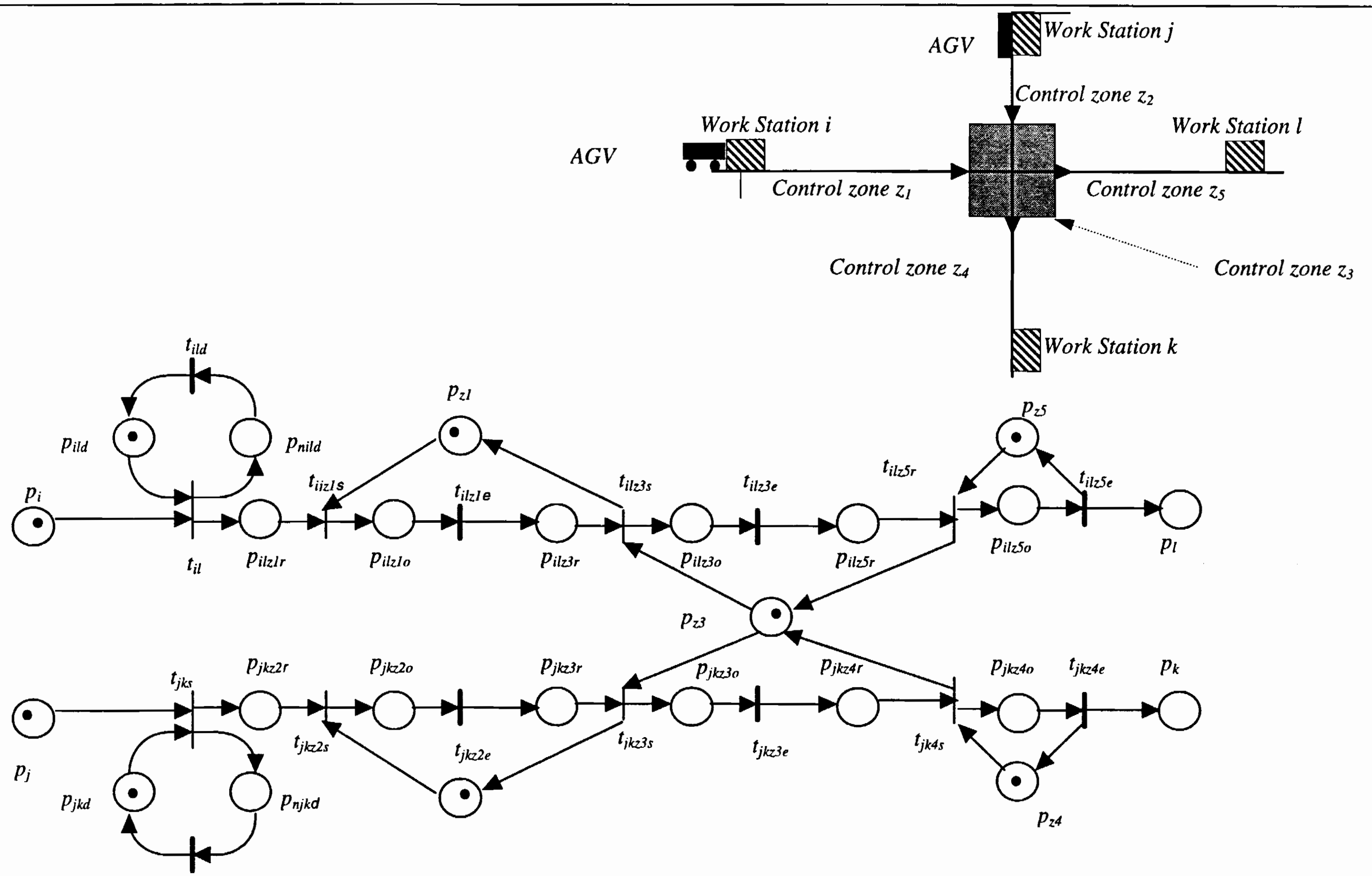


Figure 4.22 An intersection and its Petri net representation

Transitions t_{ilz1s} and t_{jkz2s} represent the beginning of the movement of the AGVs along paths il and jk into control zones z_1 and z_2 respectively. The beginning of the movement of these AGVs into the common control zone z_3 are represented by transitions t_{ilz3s} and t_{jkz3s}. The AGVs moving along path il enter control zone z_5 after clearing the common control zone z_3. The beginning of the movement of the AGV into control zone z_5 is represented by t_{ilz5s}. Similarly, the AGVs moving along path jk enter control zone z_4 after clearing the common control zone z_3 at the intersection. The beginning of the movement of an AGV into this control zone is represented by t_{ilz4s}.

4.3.5 CONTROL ZONES ALONG THE COMBINATION OF CONVERGING AND DIVERGING LANES

A combination of converging and diverging lanes appear when the lanes from two work stations converge at an intersecting points and then diverge from that intersecting point leading to two other work stations. A combination of converging and diverging lanes is illustarted in Figure 4.23. The lanes from work stations

4.3.6 VALIDATION OF THE PROPERTIES OF THE MERGED TRANSPORTATION, DEMAND AND CONTROL SUB NETS

It was shown in Sections 4.1.4 that the transportation sub net is bounded, live and reversible . In Section 4.2.2, it was shown that the merged demand and transportation sub nets had all the desired properties. In this section, the merged transportation, demand and control sub nets are proved be bounded, live and reversible. Figures, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22 and 4.23 show the different forms of incorporating the control zones in a Petri net form. It can be observed in these figures that the demand sub net is isolated from the places representing the control zones and the respective arcs leading to the transitions

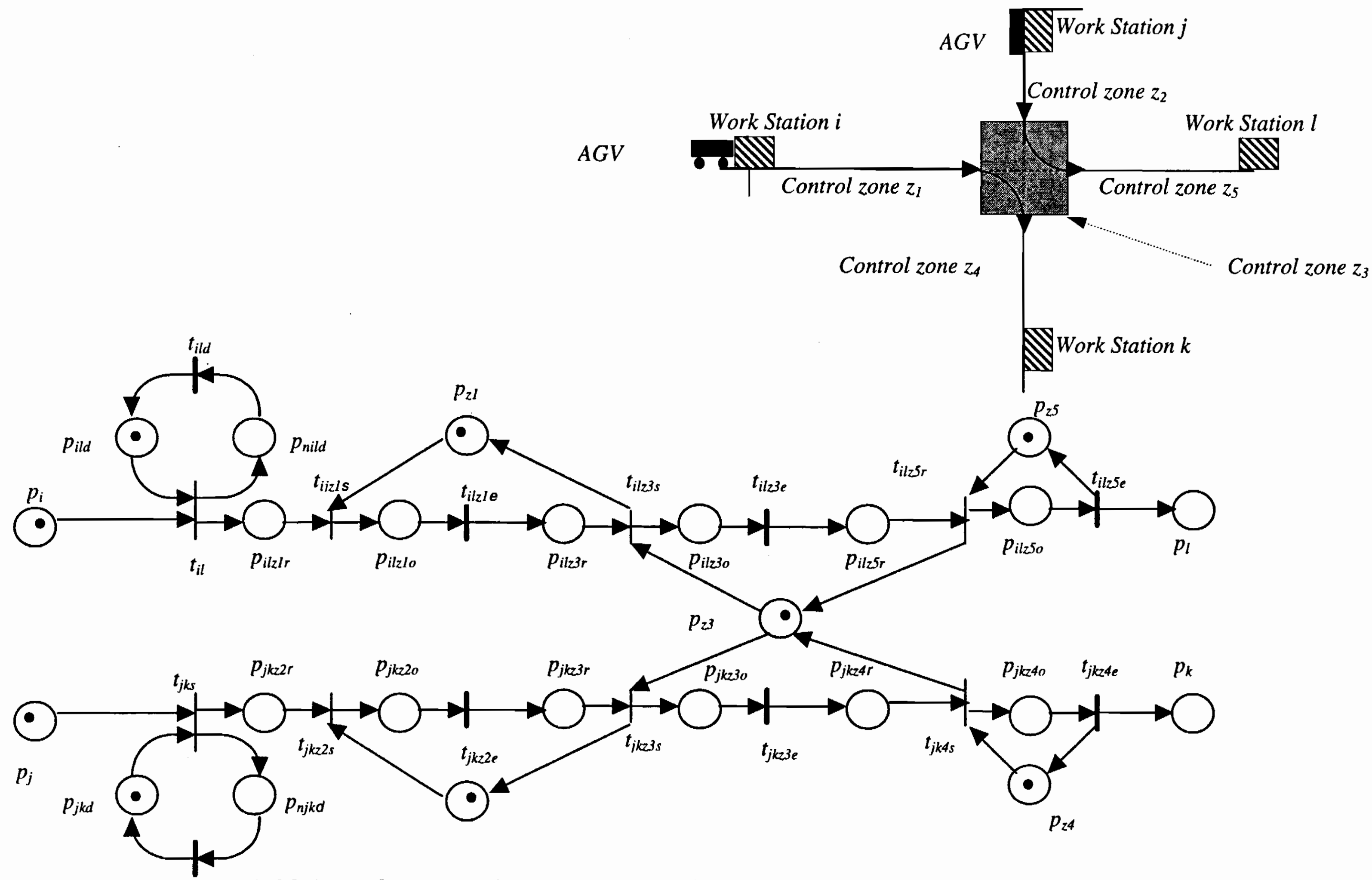


Figure 4. 23 A combination of converging and diverging lanes and its Petri net representation

in the transportation sub net. Therefore, therefore the demand sub nets and the control sub nets are independent of each other as far as the properties of the final Petri net are concerned. Hence, it is only necessary to analyze the effects of incorporating the places representing the control zones in the transportation sub net.

As discussed in the previous sections, a control zone is represented by a solitary places. It is merged with the transportation sub net through arcs leading from that place to the transitions of transportation sub net. Figure 4.23 illustrates the Petri net representation of control zone z being used by AGVs moving along four different paths namely ij, kl, mn and qr receptively. The transition t_{ijz1s}, t_{klz1s}, t_{mnz1s}, and t_{qrz1s} model the beginning of the entry of the respective AGVs into the control zone. It is easy to see that, whenever, any one of the above transitions fires the other transitions are disabled. They will be enabled only after the corresponding output transition to p_{zl} fires and a token is deposited in that places. For example if transition t_{ijz1s} fires first, then transitions t_{ijz1s}, t_{klz1s}, t_{mnz1s}, and t_{qrz1s} will be disabled. These transitions will be enabled again only after the firing of transition t_{ijz2s}. Therefore, there is no possibility of accumulation tokens in any of the places in the merged sub net and hence it is bounded.

The incorporation of the places representing the control zones and the addition of the input and output arcs to and from those places does not alter the behavior of the original transportation sub net. It simply regulates the firing of the transition of the transportation sub net. For example, if more than one paths use a particular control zone, then the incorporation of the control sub net ensures that only one transition corresponding to the paths using that control zone fires at any time. Hence the liveness and the reversibility of the transportation sub net is maintained even after being merged with the control sub net.

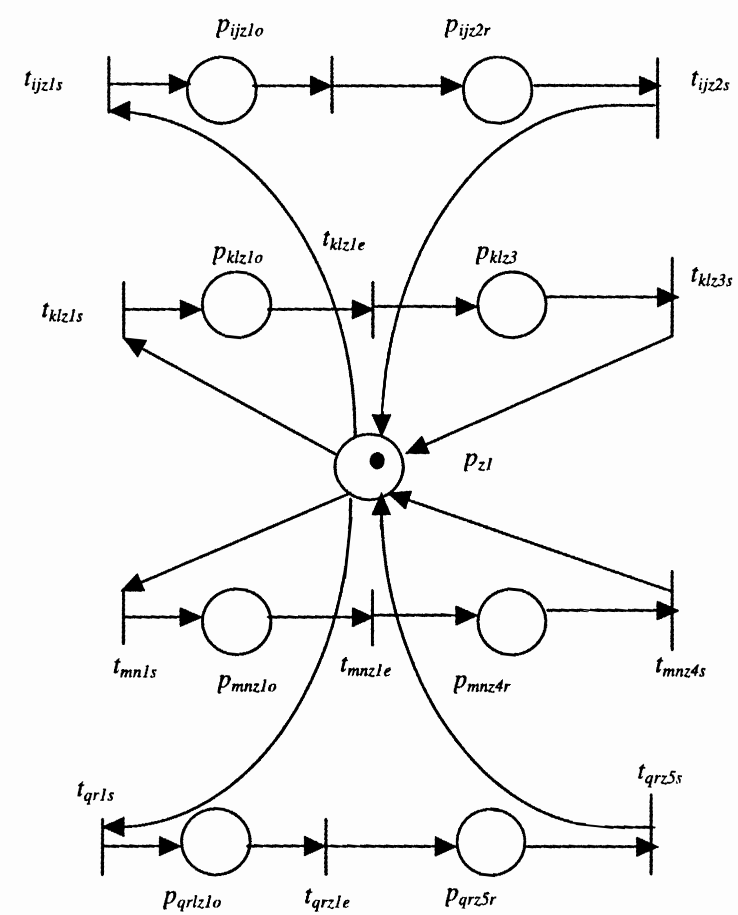


Figure 4.24 An illustration of the Petri net representation of a control zone of an intersection

4.4 MICRO ANALYSIS OF THE TRANSPORTATION AND CONTROL SUB NETS

In the previous sections, the transportation, demand and the control sub nets were described at the macro level. The overall pictures of the transportation sub net and the control sub net in particular were described. In this section, a few minor details of these control sub nets are described. These descriptions are essential for the understanding of the next two chapters. The first point of consideration is that in an AGV system, the pick up and delivery points of the work stations are not necessarily located at the same point. Theses two points are generally located separately. Now, in the transportation sub net, each place represents a work station. Therefore, it should be clear as to whether the pick up point or the drop off point is represented by the places of the transportation sub net. However, when an AGV moves from one work station to another, it moves from the pick up point of the origin work station to the delivery point of the destination work station. Once it unloads the part at the delivery point of the destination work station, it moves to the pick up point of the destination work station. The destination work station for an AGV is decided at the pick up point of the origin work station. Figure 4.24 illustrates a work station at the pick up point of work station i. It can either move to the delivery point of work station j or to the delivery point of work station k. The transportation sub net is constructed such that the places represent the points at which the decision is made on the destination work stations. Therefore, the pick up points of the work stations should be represented by the places of the transportation sub net.

The Petri net models representing the control zones are constructed on the assumption that, there is at least one control zone between each pair of work stations. The control sub net models show that an AGV leaves a work station

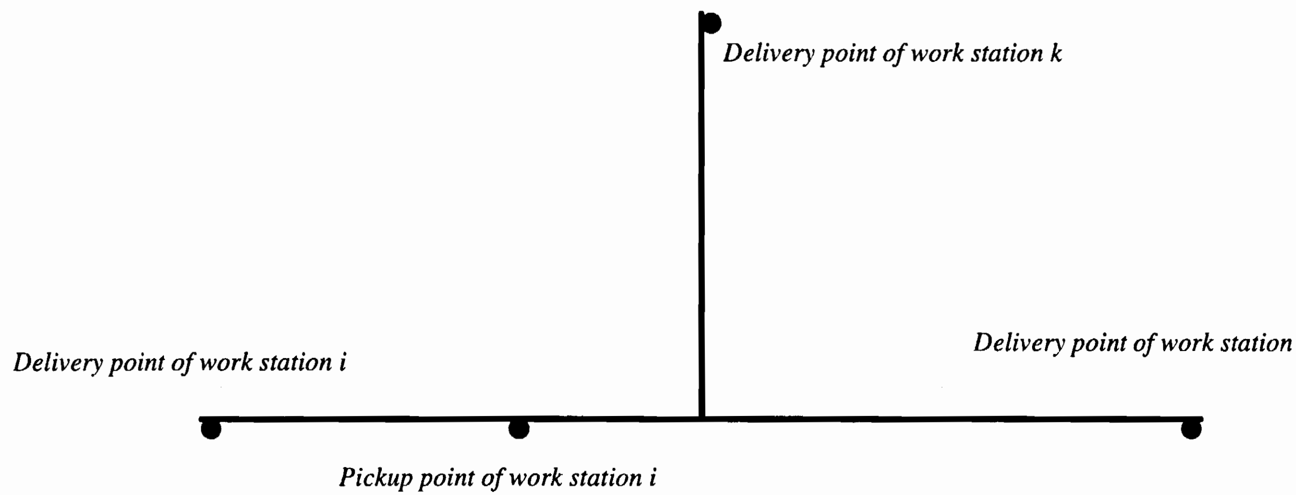


Figure 4.25 Illustration of the pick up and delivery points of the origin and destination work stations

and enters a control zone immediately when the control zone is available. It moves along one or more control zones and then enters the destination work station. As shown earlier, the work stations in the Petri net models are the pick up points of those work stations. Hence, between each pair of pick up points of the work stations there should be at least one control zone. Figure 4.25 illustrates the pick up points of work stations i and j and a control zone between these two pick up points. It is noted in this figure that the delivery point of work station j is in the control zone. Therefore, the time an AGV spends in this control zone is the time taken for it to move through the control zone plus the time taken to unload the part at work station j. The shaded area in the figure shows the control zone at the intersection and the thicker line from the shaded area to the delivery point of work station k. The notable feature in this figure is that the pick up point of each work station is located on the boundary of a control zone. This is in accordance with the control sub nets representing the control zones. The important design consideration in the control zones is that the pick up point of every work station should be at the boundary of a control zone. The objective of the design of the control system is to minimize number of control zones used in the layout.

Chapter 5 of this thesis provides the theoretical back ground for the computation of the dimensionss of each control zone. It is based on the speed of the AGVs and the demand for the vehicles between each pair of work station. These results provide the frame work for the design of the control zones. The location of the control zones is based on the location of the pick up points of the work stations and the dimensions of the control zones. Chapter 6 gives a systematic procedure for the location of the control zones.

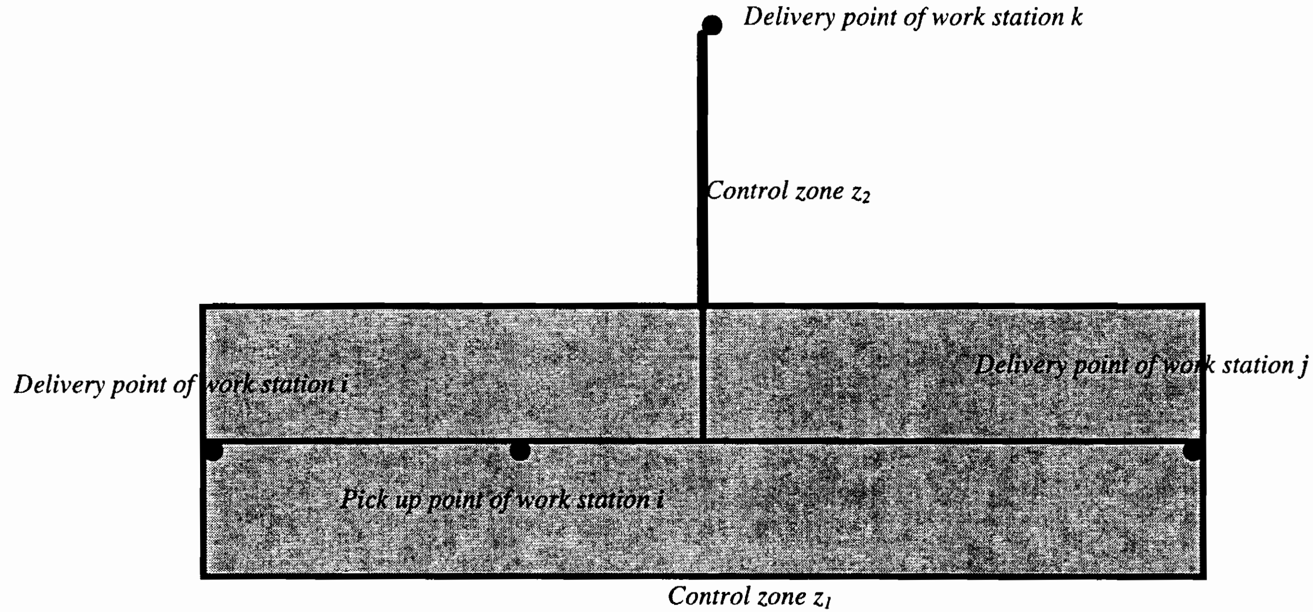


Figure 4.26 Illustration of the control zones between the origin and destination work stations

5. THEORETICAL DEVELOPMENTS

This chapter provides the necessary theoretical background to apply the results of Sifakis [47] in evaluating the design parameters of the AGV system. The design parameters are evaluated using the three sub nets introduced in Chapter 4. The three sub nets are analyzed for the performance measures using the results of Sifakis [47] introduced in Chapter 3. According to his results, for a system represented by a Petri net to function at its maximum rate, the following conditions should be satisfied: $X^T.M_0 \geq X^T.IN.D_T.I$ and $A.I=0,\ I>0$. In the above expression, X is a place invariant of the Petri net representing that system. The vector M_0 is its initial marking, matrix IN is the input function. The square matrix is D_T represents the delay times of the transitions. The vector I gives firing rates associated to the transitions. The condition shown above should be satisfied by all the place invariants of the Petri net. The following sections explain as to the components in the expression mentioned above are computed in the three sub nets representing an AGV system.

5.1 PLACE INVARIANTS OF THE PETRI NET REPRESENTING THE AGV SYSTEM

A place invariant of a Petri net is a set of places of that Petri net where the total number of tokens in that set of places is the same for any marking reachable from the initial marking. A heuristic way of finding the place invariants of the Petri net representing the AGV system is as follows: First, the tokens in the transportation sub net represent the vehicles in the AGV system. Since the number of AGVs in the system remain the same, the number of tokens in the transportation sub net should be the same for any marking reachable from the initial marking. Therefore, the places of the transportation sub net should form a place invariant of the Petri net representing the AGV system.

Similarly, when a demand sub net representing the demand between a pair of work stations is considered, the total number of tokens in the places of the demand sub net is always one. A token in one of the two places represent the availability of the demand for an AGV between the two work stations. A token in the other place represents the non availability of a demand between the two work stations. Hence a token should be in either one of the two places but not in both. Therefore the total number of tokens in the two places is exactly one for any marking. Therefore the two places representing the demand between a pair of work stations also forms a place invariant of the final Petri net.

Finally, in the control sub net, the control zone is represented by a solitary place. The control sub net consists of the place representing the control zone and the places and transitions between input and output transitions of that place modeling the control zone. For example, Figure 5.1 illustrates the control sub net related to control zone z_1 used by two paths ij and ik. Transitions t_{ijz1s} and t_{ikz1s} are the input transitions of place p_{z1} representing control zone z_1. Its output places are t_{ijz2s} and t_{ikz2s}. The places and transitions between the pairs of transitions (t_{ijz1s} ,t_{ikz1s}) and (t_{ijz2s} ,t_{ikz2s}) form the control sub net. It is easy to see in Figure 5.1 that, the total number of tokens in the places between the pairs of transitions (t_{ijz1s},t_{ikz1s}) and (t_{ijz2s},t_{ikz2s}) is always one. To be more precise, the token in the control sub net remains in place p_{z1} to represent that the control zone is free. If it is any of the other places of the control sub net, it indicates that the control zone is used by an AGV moving along one of the paths. Therefore, the places of the control sub net form a place invariant. The following sections show how the merged sub nets are decomposed into different place invariants and how they are also the transportation, demand and control sub nets.

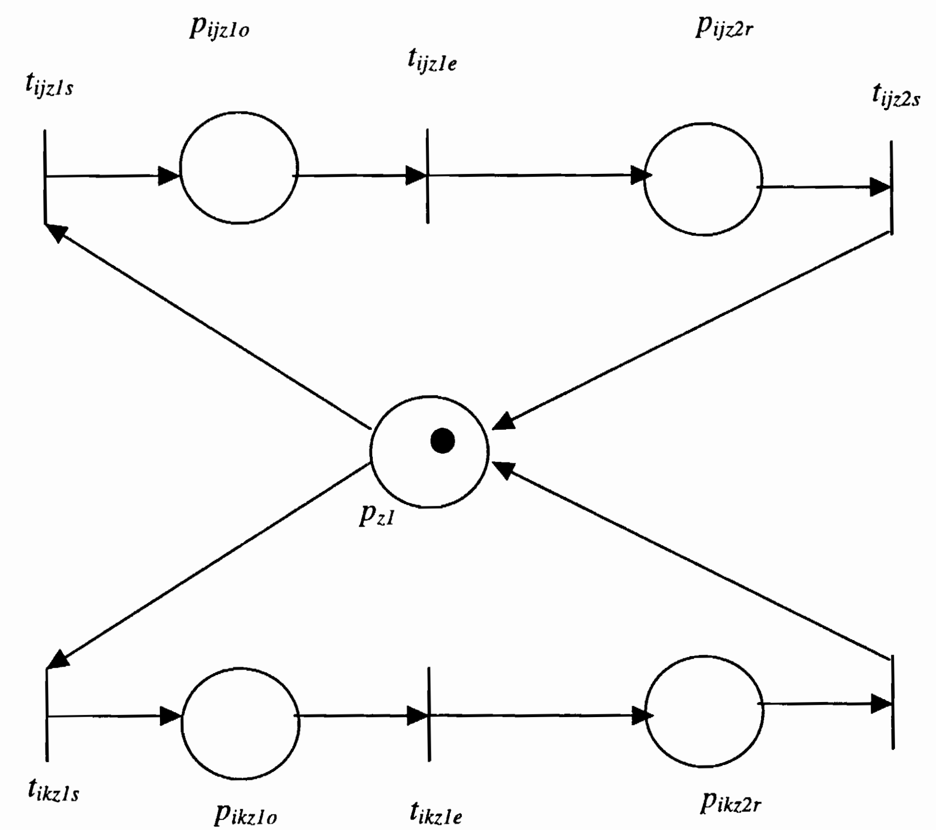


Figure 5.1 The place invariant formed by a control zone

5.2 PLACE INVARIANTS OF THE MERGED TRANSPORTATION AND DEMAND SUB NETS

In order to study the place invariants of the merged transportation and demand sub nets, it is necessary the structure of this merged net is analyzed. Figure 5.2 illustrates this merged sub net. The two transitions of importance in this analysis are t_{ijs} and t_{ijd}. When transition t_{ijs} fires, tokens are removed from places p_i, p_{ijd} and deposited in places p_{ij}, p_{nijd}. Therefore, the firing of transition t_{ijs} does not cause any change in the number of tokens in either the transportation or demand sub net. Similarly, when transition t_{ijd} fires, a token is removed from place p_{nijd} and deposited in place p_{ijd}. Therefore the firing of transition t_{ijd} also does not affect the number of tokens in either the transportation or the demand sub net. The firing of any of the other transitions does not change the number of tokens in the transportation or demand sub nets. Therefore the transportation and demand sub nets form two place invariants. These two place invariants are shown in Figures 5.3.

5.3 PLACE INVARIANTS OF THE MERGED TRANSPORTATION, DEMAND AND CONTROL SUB NETS.

The procedure of finding the place invariants of the merged transportation, demand and control sub nets is also based on the movement of tokens along the merged sub net. Figure 5.4 illustrates the Petri net representation of two merging lanes in a guide path. The shaded areas in this figure show the four control zones facilitating the movement of AGVs along the merging lanes. Places p_{z1}, p_{z2}, p_{z3} and p_{z4} model the control zones. The other places and the transitions in the shaded areas are the nodes necessary to merge the places modeling the control zone to the transportation sub net. It is easy to see that the

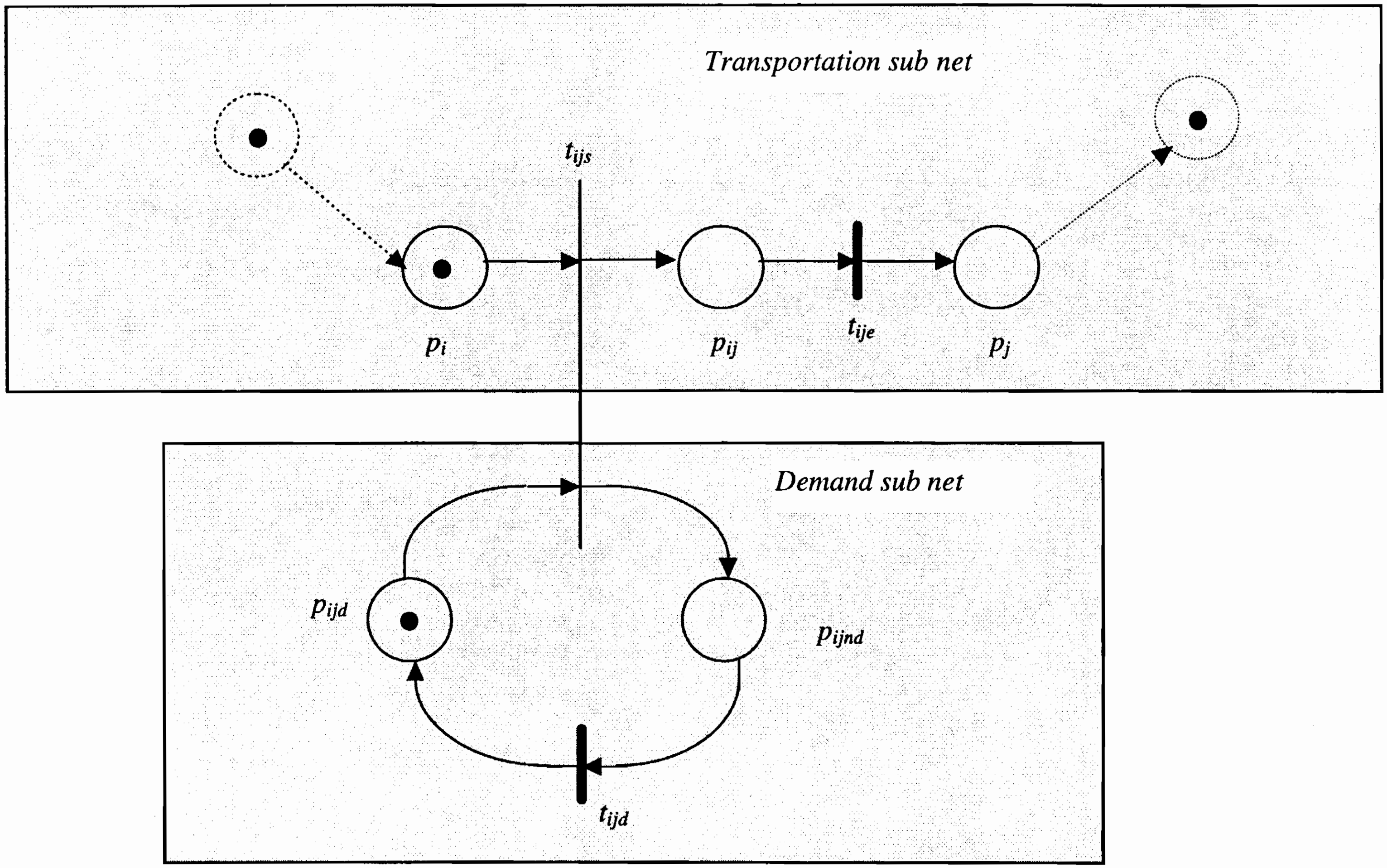


Figure 5.2 The merged transportation and demand sub net

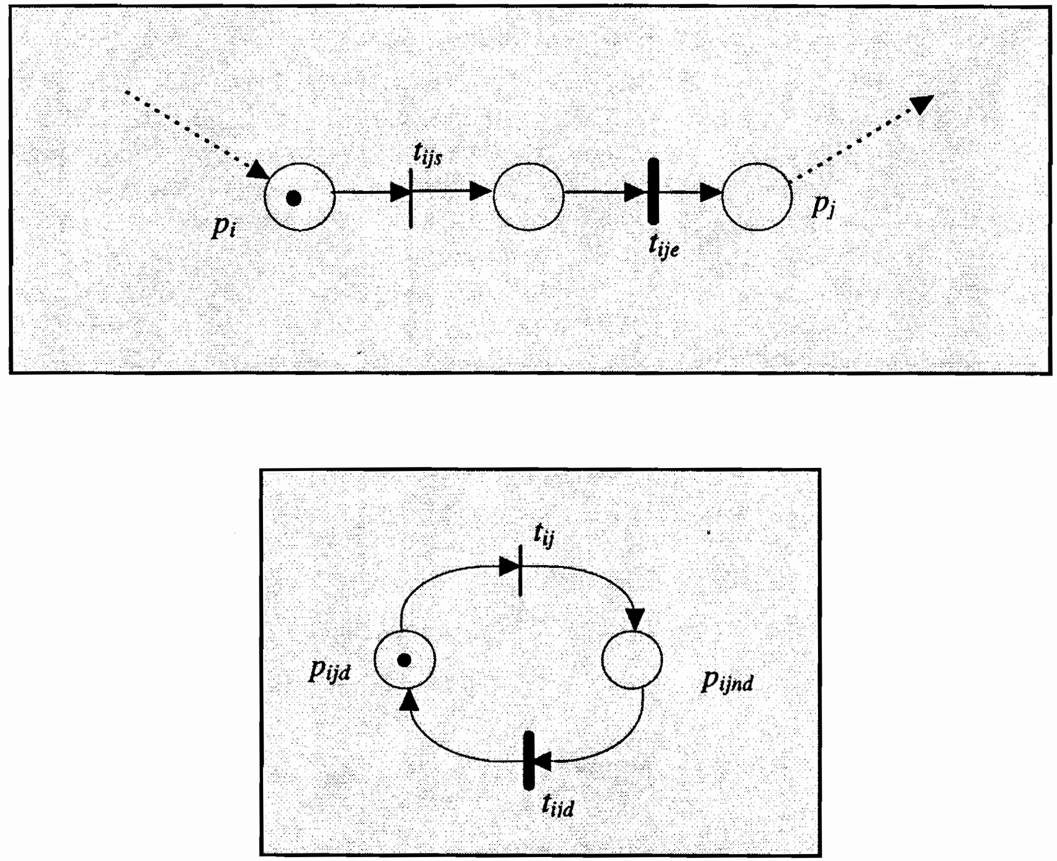


Figure 5.3 Illustration place invariants of the merged transportation and demand sub nets

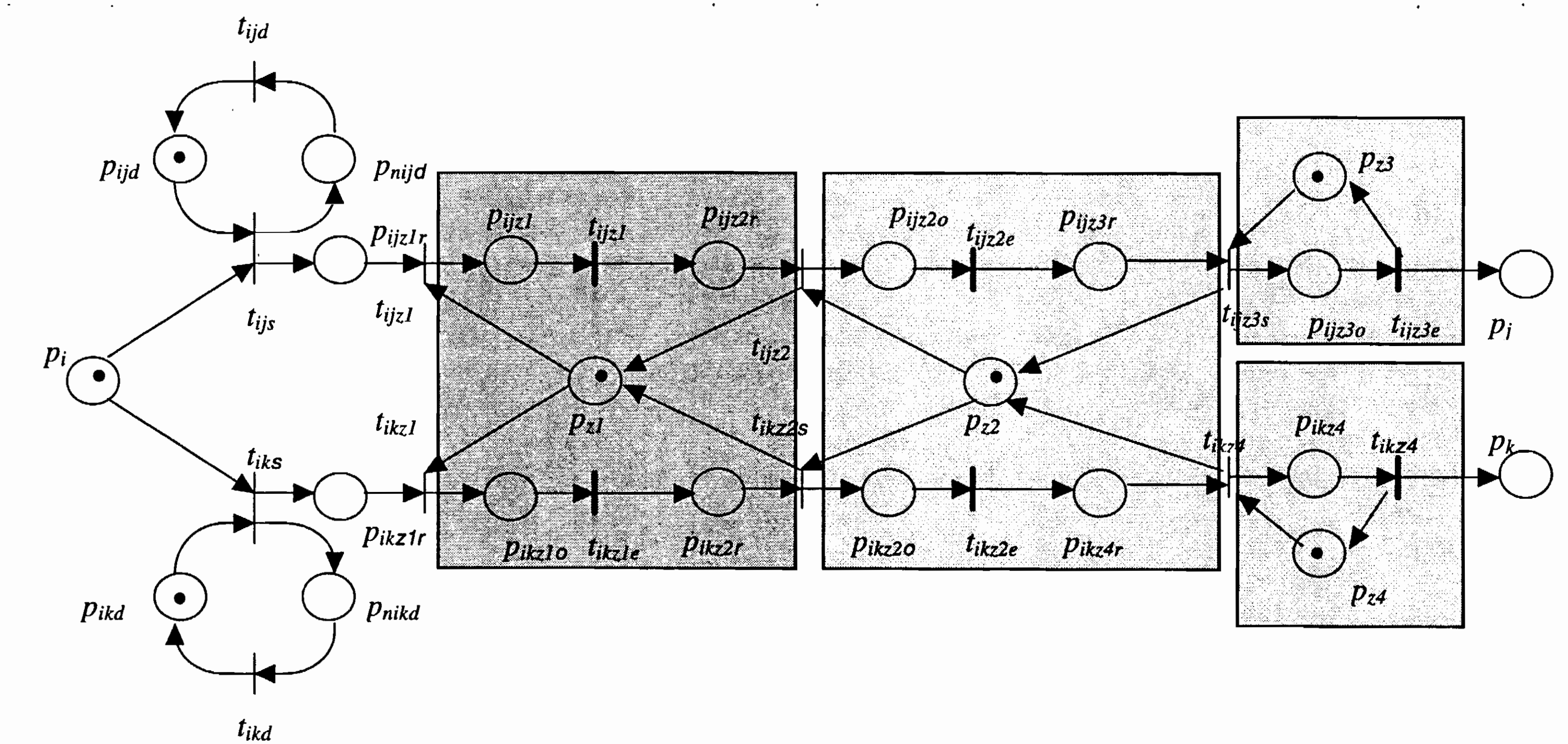


Figure 5.4 An illustration of merging lanes and its Petri net representation

number of tokens in each of the shaded areas of Figure 5.4 is exactly equal to one. Therefore, each set of places in the shaded area form a place invariant of the final Petri net. The four sets of places that form the place invariants are shown in Figure 5.5.

5.4 THE INITIAL MARKING

The initial marking in the transportation sub net represents the initial set up of the AGVs in the system. For example if place p_i is marked with a token in the initial marking M_0, then an AGV is available at work station i in the initial set up of the AGV system. The demand sub net representing the demand between two work stations has two places. The initial marking represents the availability of a demand between the two work stations. At time zero, it is assumed that there is a demand between each pair of work stations. Therefore, the initial marking of each demand sub net between two work stations consists of a single token to represent the availability of a demand between the two work stations. Similarly, at time zero, the AGVs are assumed to at the work stations and the control zones are assumed to be free. Therefore, in the initial marking, a token is available in each place representing the control zones.

5.5 THE INPUT FUNCTION

The input function was defined in Chapter 3 with the introduction of Petri nets. It is represented as a mxn matrix with the rows representing the places and the columns representing the transitions. Each element of the matrix represents the weight of the arc from the place to the transition which corresponds to the row and column of that element. Another important characteristic of the input function, is its special structure. As discussed earlier, the place invariants are the transportation, demand and control sub nets. These sub nets were shown to be state machines. In a state machine each place has one and only one input

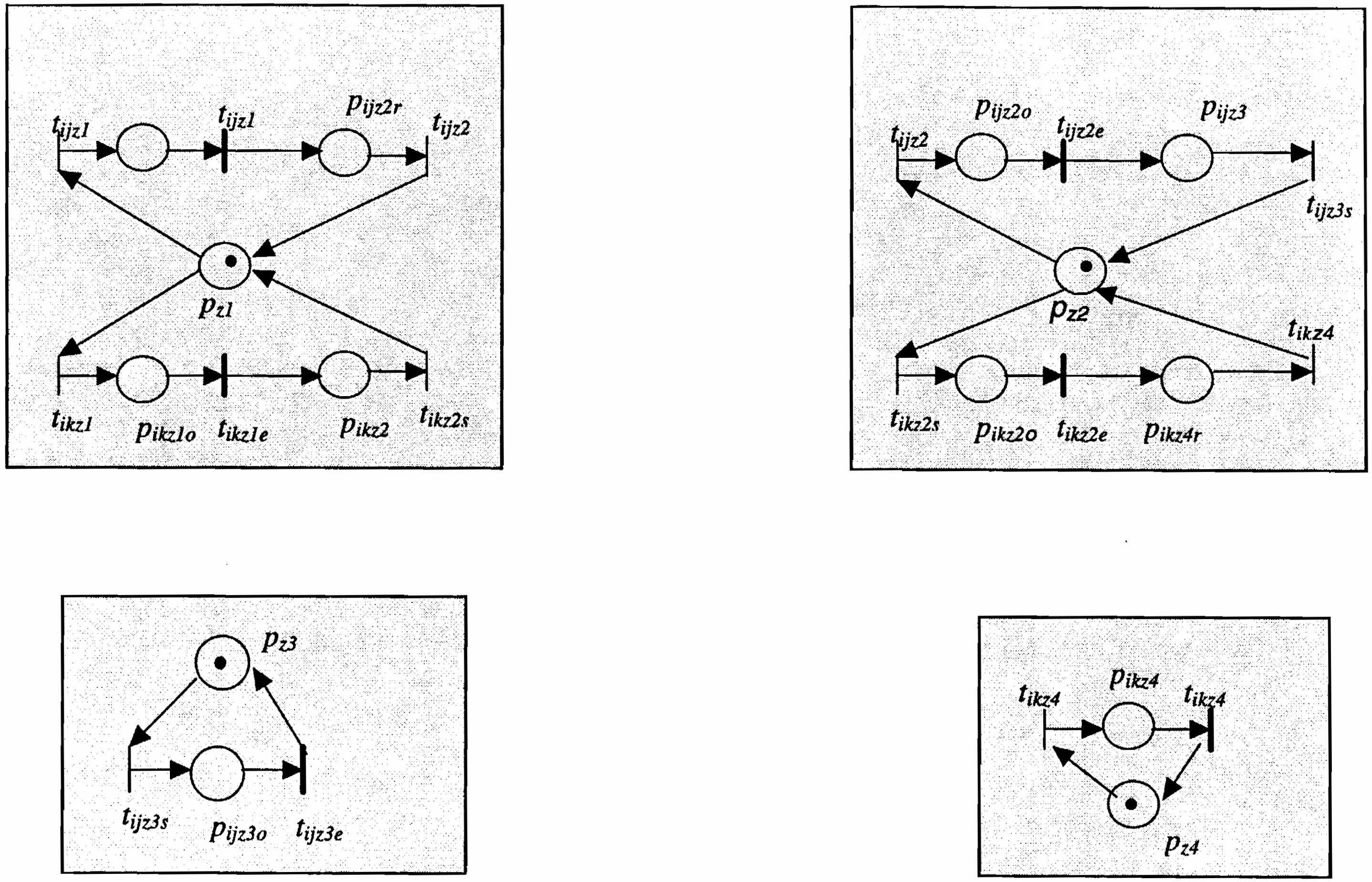


Figure 5.5 Illustration of the place invariants related to the control sub nets

place and one and only one output place. Therefore, the input functions related to these place invariants form a special type of matrix where each row consists of exactly one element which is 1 and all the other elements being equal to 0.

5.6 THE FIRING RATES OF THE TRANSITIONS OF THE TRANSPORTATION SUB NET

The firing rate of a transition of a Petri net modeling a system functioning repetitively is the number of times that transition fires during one cycle. The firing rates of the transitions of the Petri net is represented as a vector known as the firing rate vector. The firing vector is written as I_{nx1} for a Petri net with n transitions. As mentioned earlier the firing vector should satisfy the following three conditions for a system to function at its maximum rate: (1) $I > 0$, (2) $A.I=0$, and (3) $X^T.M_0 \geq X^T.IN.D_T.I$.

The first condition simply states that the firing rates of each transition should be greater than zero. In other words, each transition representing the system should fire at least once in a cycle. The transportation, demand and the control sub nets are constructed in such a way that this condition is satisfied. The second condition, mathematically written as $A.I = 0$, ensures that the system returns to its initial state after a sequence of firing of transitions. This condition produces some interesting relationships between the firing rates of the transitions of the transportation, demand and control sub nets. These relationships are described in the following sections.

5.6.1 RELATIONSHIP BETWEEN THE FIRING RATES OF THE TRANSITIONS OF THE TRANSPORTATION SUB NET

In the expression $A.I = 0$ mentioned above, A is the incidence matrix of the Petri net. Each row of the incidence matrix of the transportation sub net consists of s and zeros. The +1 s correspond to the output transitions and the -1 s correspond to the input transitions of the place represented by the row of the incidence matrix. Therefore, according $A.I = 0$, the summation of the firing rates of the input transitions of any place of the transportation sub net is equal to the summation of the firing rates of its output transitions.

Figures 5.6 (a), (b), (c) and (d) illustrate the different possible configurations in a transportation sub net. Figure 5.6 (a) shows a situation where a place has one and only one input transition and one and only one output transition. In this situation, the number of times the input transition t_1 fires in one cycle is equal to the number of times the output transition t_2 fires. Therefore, if I_1 and I_2 are the firing rates of the two transitions, then $I_1 = I_2$.

Figure 5.6 (b) illustrates a situation where place exactly one input transition t_1 and two output transitions t_2 and t_3. In this case, the number of times transition t_1 fires is equal to the sum of the times transitions t_2 and t_3 fire. Therefore $I_1 = I_2 + I_3$. Similarly in Figure 5.6 (c), the relationship between the firing rates of the transitions t_1, t_2 and t_3 is $I_1 + I_2 = I_3$. Figure 5.6 (d) illustrates a general case where, a place has multiple input and output transitions. In this case, the sum of the number of times the input transitions fire is equal to the sum of the number of times the output transition fires. Therefore the relationship between the firing rates of the input and output transitions is written as $\sum_{i=1}^{i=r} I_i = \sum_{j=r+1}^{s} I_j$. Where, I_1, I_2,...,I_r are the firing rates of the input transitions and I_{r+1}, I_{r+2},..., I_s are the firing rates firing of the output transitions of place p_i.. The interpretation of this result is that the number of time the AGVs enter a work station is equal to the number of times the vehicles leave that work station in one complete cycle.

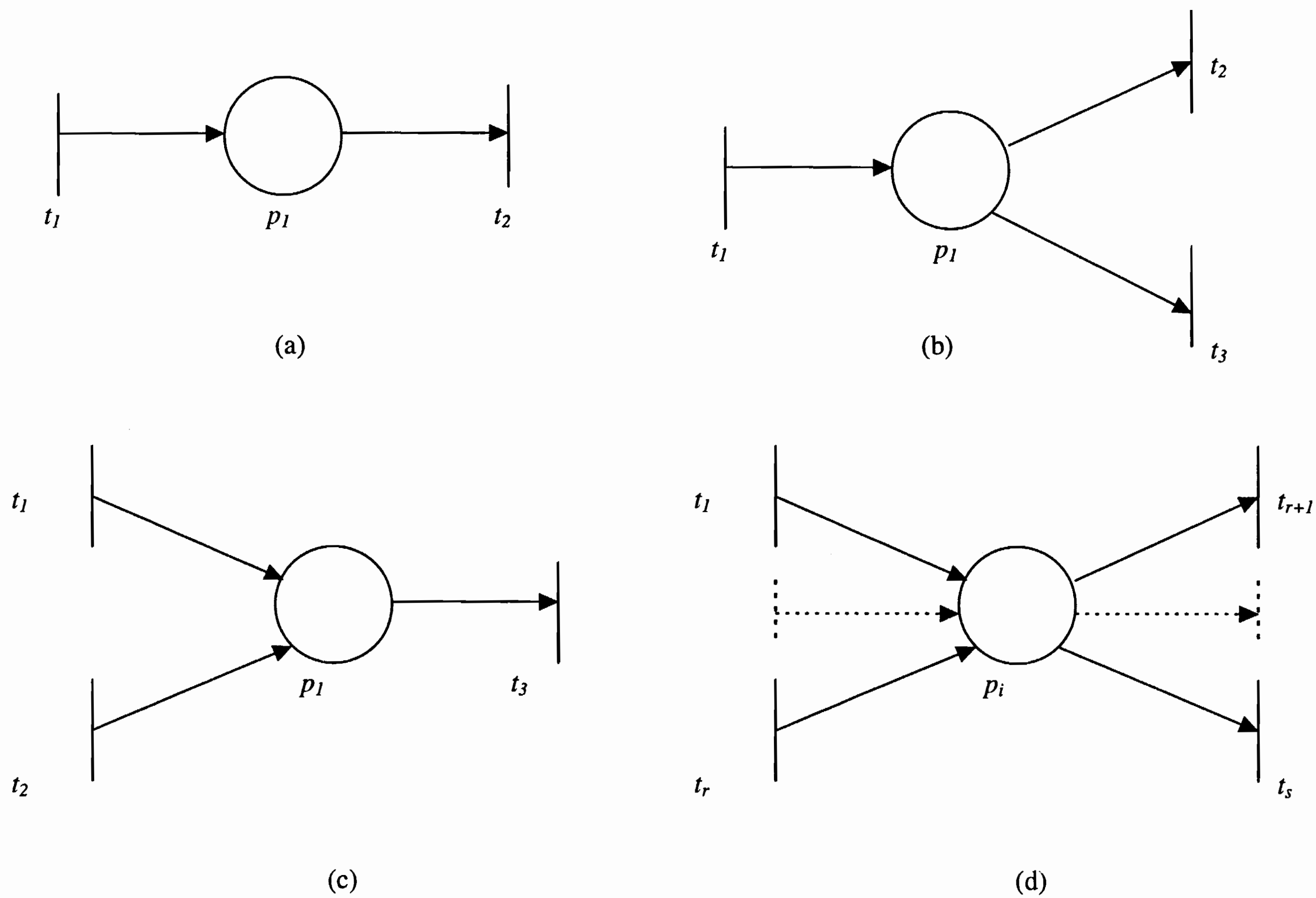


Figure 5.6 Illustration of the different possible configurations in a transportation sub net

5.6.2 THE RELATIONSHIP BETWEEN THE FIRING RATES OF THE TRANSITIONS OF THE MERGED TRANSPORTATION AND DEMAND SUB NETS

This section explains the relationship between the firing rates of the transitions of the transportation and demand sub nets. Figures 5.7 and 5.8 illustrate the merger of the transportation and demand sub nets. The movement of a token from place p_i to place p_j in Figure 5.7 is achieved through the firing of transition t_{ij}. Figure 5.8 illustrates merged transportation and demand sub nets. In the merged sub net, once transition t_{ijs} fires, it can fire again only after the firing of transition t_{ijd}. The movement of a token from place p_i to place p_j in the merged transportation and demand sub net occurs through the firings of transitions t_{ijs} and t_{ijd}. Therefore, the firing rate of t_{ij} in the transportation sub net is equal to the firing rates of t_{ijs} and t_{ijd} in the merged transportation and demand sub nets. Hence $I_{ij}=I_{ijs}=I_{ijd}$.

5.6.3 THE RELATIONSHIP BETWEEN THE FIRING RATES OF THE TRANSITIONS OF THE MERGED TRANSPORTATION, DEMAND AND CONTROL SUB NETS

In this section, the relationship between the firing rates of the transitions of the transportation, demand and the control sub nets are determined. Figure 5.9 and 5.10 illustrate the merged transportation, demand and control sub nets for a path ij using control zone z. As stated in the previous section, the movement of a token from place p_i to place p_j takes place during each firing of transition t_{ij}. In Figure 5.10, it can be observed that, once the transitions in the shaded area are fired, they can fire again only after the next firing of transition t_{ijd}. Hence the number of times the transitions t_{ijzs} and t_{ijze} fire in one cycle is equal to the

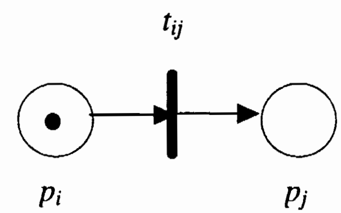


Figure 5.7 Places and transition of the transportation sub net

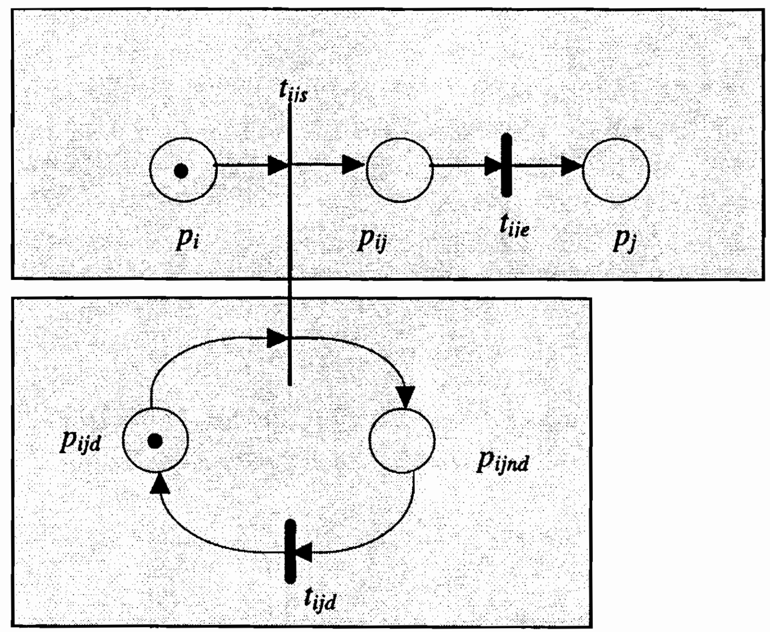


Figure 5.8 The places and transitions of the merged transportation and demand sub nets

5.7 COMPUTING THE FLEET SIZE OF AN AGV SYSTEM USING THE TRANSPORTATION AND DEMAND SUB NETS

The fleet size of the AGV system is computed using the results of Sifakis [47]. The place invariants of the merged transportation and demand sub nets are used in this computation. The method of finding the place invariants of the merged transportation and demand sub nets was explained in Section 5.2. The first place invariant consists of the set of places of the transportation sub net and the other place invariants consist of the places of each demand sub net. The condition for the AGV system to function at its maximum rate is by $X^T.M_0 \geq X^T \cdot IN.D_T.I$ and $A \cdot I = 0$, $I > 0$. These sets of conditions are applied on both the transportation and demand sub nets.

5.7.1 APPLICATION OF THE EXPRESSION FOR THE MAXIMUM OPERATING CONDITIONS ON THE TRANSPORTATION SUB NET

Let *m* be the number of places in the place invariant formed by the transportation sub net and *n* be its number of transitions. Then this place invariant can be represented by a vector X such that $X^T = (1 \; 1 \; 1 \; . \; . \; . \quad 1 \; 1)_{1xm}$. Hence, $X^T.M_0$ represents the number of tokens in the transportation sub net. The numbers of tokens in the transportation sub net represent the number of AGVs in the system. Let this quantity be N.

The right hand side of the expression can be broken down into two portions. The first portion is $X^T.IN$. As shown earlier, the input function IN is a special type of matrix with only one element in each row being 1 and all the others being equal to 0. The transpose of the vector representing the place invariant related

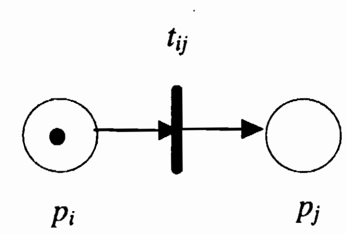


Figure 5.9 Places and transition of the transportation sub net

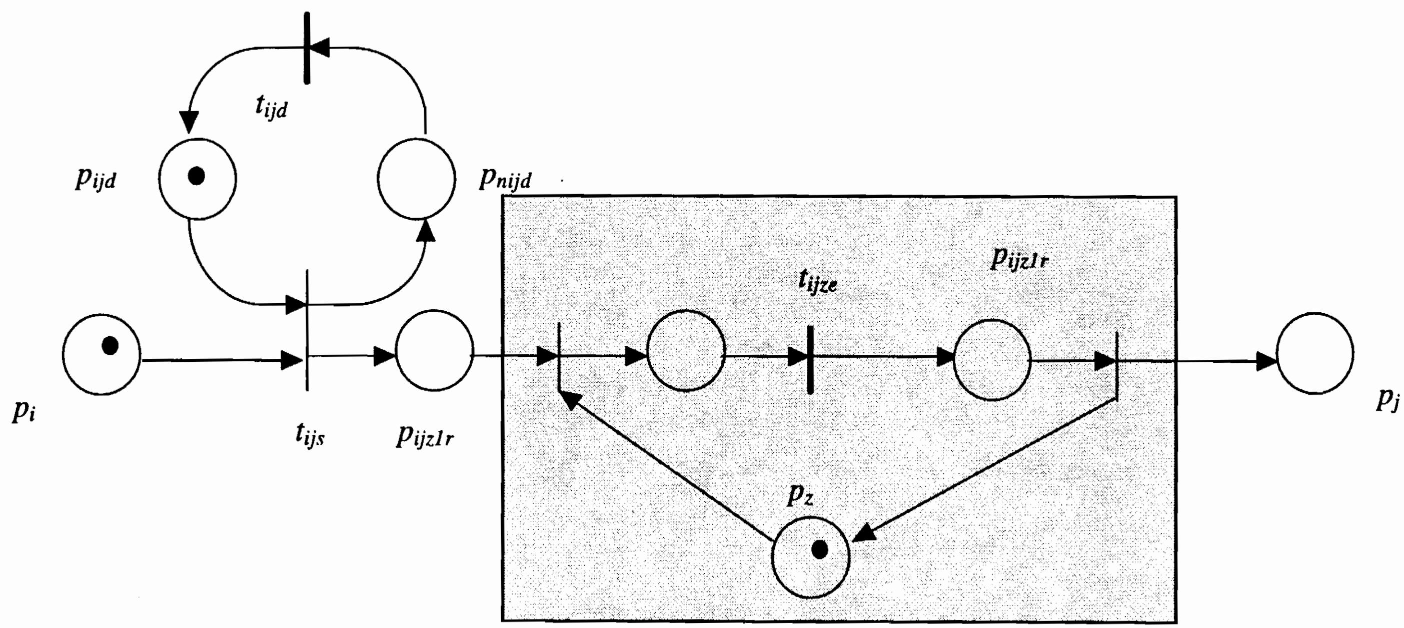


Figure 5.10 Illustration of a control sub net of a control zone used by two different paths

to the transportation sub net is a unit row vector. Therefore $X^T.IN=\begin{pmatrix}1 & 1 & 1 & . & . & . & & 1 & 1\end{pmatrix}_{1xn}$.

The next portion is $D_T.I$. Matrix D_T represents the delays associated to each transition of the transportation sub net. It is a diagonal matrix with each diagonal element representing the firing each transition. All the other elements of this matrix are zeros.

Therefore $D_T = \begin{pmatrix} d_1 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & d_n \end{pmatrix}_{nxn}$. Hence, $X^T.IN.D_T=\begin{pmatrix}d_1 & d_2 & & d_n\end{pmatrix}_{1xn}$. If the firing rate vector of the transitions in the place invariant is represented by $\begin{pmatrix}I_1 & I_2 & & I_n\end{pmatrix}_{1xn}$, then $X^T.IN.D_T.I=\sum_{r=1}^{r=n} I_r \cdot d_r$. Hence, the requirement for the maximum operating conditions of the AGV system is $N \geq \sum_{r=1}^{r=n} I_r \cdot d_r$. This expression states that the total number of tokens in the transportation sub net should be greater than the summation of the terms obtained by the firing rates and the delay times of transitions of the transportation sub net.

Hence, if t_{ij} is a transition of the transportation sub net, then above condition is written as $N \geq \sum_{t_{ij} \in TSN} I_{ij} \cdot \tau_{ij}$. The symbol $t_{ij} \in TSN$ means that t_{ij} is a transition of the transportation sub net and I_{ij} and τ_{ij} are the firing rate and the delay time associated to that transition.

5.7.2 COMPUTING THE FLEET SIZE OF THE AGV SYSTEM

Similar results are obtained for the place invariants containing the places of the demand sub nets as well. As shown earlier, the number of tokens in the demand sub net should be exactly 1. Now, performing similar operations on the control sub net, it is easy to show that the expression for the maximum operating conditions of the system as $1 \geq I_{ijd} \cdot \tau_{ijd}$. However, if the demand for vehicles should be met exactly, then the above condition should be the binding constraint and therefore $I_{ijd} \cdot \tau_{ijd} = 1$. Where, I_{ijd} is the firing rate of transition t_{ijd} and τ_{ijd} is the time between consecutive demand for an AGV between work stations i and station j.

It was shown earlier that $I_{ij}=I_{ijd}$. By making use of the above results the minimum number of AGVs needed for system as $N \geq \sum_{t_{ij} \in TSN, t_{ijd} \in DSN} \frac{\tau_{ij}}{\tau_{ijd}}$.

5.8 FINDING THE CONTROL ZONE PARAMETERS TO MEET THE AGV FLOW REQUIREMENT

Finding the control zone parameters is similar to the evaluation of the fleet size of the AGV system. The place invariant consisting of the places of the control zone are used to find the conditions for the AGV system to function at its maximum rate. Let ij be a path along which AGV moves and z be a control zone in this path. Then, the transitions in the place invariants related to the control zones are t_{ijzs}, t_{ijze} and t_{ijz1s}. These transitions are illustrated in Figure 5.10. Transitions t_{ijzs} and t_{ijz1s} are immediate transitions. The time associated to them a zero. The time associated to transition t_{ijze} is the time taken for an AGV moving along path ij to clear control z. If this time is denoted by τ_{ijze}, then the condition for the AGV system to operate at its maximum rate is given by $1 \geq \sum_{ij \in z} I_{ijze} \cdot \tau_{ijze}$.

It was shown earlier in Section 5.6 that, $I_{ijd}=I_{ijze}$ and in Section 5.7.2 that $I_{ijd}=1/\tau_{ijd}$. Making use of the above results, the condition for the AGV system to function at its maximum rate can be shown to be $1 \geq \sum_{ij \in z} \frac{\tau_{ijze}}{\tau_{ijd}}$. The symbol $ij \in z$ indicates that control zone z is used by AGVs moving along path ij.

6. THE DESIGN METHODOLOGY

In the previous chapter, necessary expressions were derived that are used for the computation of the fleet size and the dimensions of the control zones. This chapter explains the procedure of designing the control zones once their dimensions are computed. The first section of this chapter gives the procedure for a single loop AGV system and the second section illustrates the procedure for a multi loop AGV system with intersecting lanes.

The dimensions of the control zones are computed using the results of Section 5.8. According to the results of Section 5.8, the condition for the system to operate at its maximum rate is given by $1 \geq \sum_{ij \in z} \frac{\tau_{ijze}}{\tau_{ijd}}$. Where τ_{ijze} is the time taken for the AGV moving between work station i and j to clear control zone z and τ_{ijd} is the time between consecutive demands for AGVs from work station i to j. The control zones should be constructed in such a way that the demand for AGVs between each pair of work stations is satisfied. Therefore, the term τ_{ijd} is known and fixed for the pair of work stations i and j. For the above expression to be satisfied, the value of the term τ_{ijz} should be as low as possible. Since the speed of the AGV is known and fixed, this expression gives an upper bound for the length of each control zone. This chapter focuses on this result and provides systematic procedures for the design of the control zones.

6.1 DESIGN OF CONTROL ZONES FOR A SINGLE LOOP AGV SYSTEM

The computation of the dimensions and the location of the control zones for a single loop AGV system is quite simple and straight forward. The computation is based on the fact that the time between consecutive demands between the work stations of a single loop AGV system is same for any transition t_{ij} of its transportation sub net. Therefore the term τ_{ijd} is independent of ij. If L_{max} is the maximum length of the control zone in between work stations i and j, it is easy to show that $L_{max} = V^{*}\tau_{ijd}$. Where, V is the speed of the AGV. As stated in Section 4.4, there should be at least one control zone between the pick up points of each work station. Therefore, if there are n work stations in a single loop guide path layout, there should be at least n-1 control zones. If the length between the pick up point of work stations i and j is L_{ij}, then the number of control zones that should be present between the two pick up points is $\left\lceil \frac{L_{ij}}{L_{\max}} \right\rceil$.

6.2 DESIGN OF CONTROL ZONES FOR A MULTI LOOP AGV SYSTEM

Designing the control zones for a multi loop AGV system is more complicated than that of the single loop system. A multi loop system contains two types of control zones. The first type is the ordinary control zones that exist along the straight portion of the guide path. The second type is the control zone at the intersecting points of merging, diverging or intersecting lanes. This section gives a systematic procedure of designing these two types of control zones.

6.2.1 DESIGN OF CONTROL ZONES AT THE INTERSECTING POINTS

In designing the control zones for the multi loop AGV system, the control zones at the intersecting lanes are considered first. Once these control zones are fixed the control zones along the linear portions of guide path are designed. Figure 6.1 illustrates a control zone at an intersecting point of guide path layout. The x_1, x_2, x_3, and x_4 are the distances along which the control zone expands in the

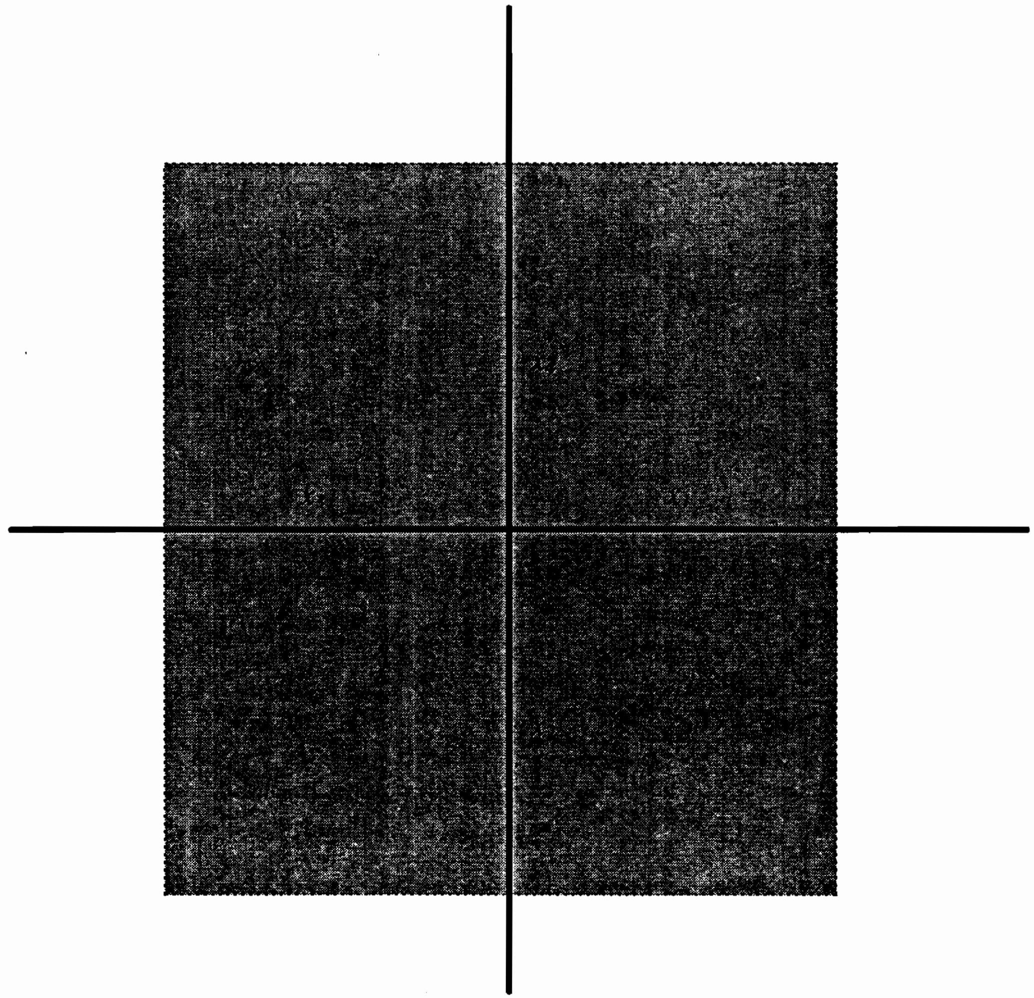


Figure 6.1 Illustration of a control zone at an intersection point

four directions originating from the intersecting point. It was shown earlier, that the condition for the system to maximum rate is depended on the time an AGV takes to clear a control zone and the time between demand for AGVs. The demand for AGVs between the work stations is fixed and known. Therefore, the performance of the AGV system is depended on the time an AGV takes to clear the control zone. The time the AGV takes to clear the control zones depends on the distance it has to travel inside that control zone.

Therefore for a control zone at an intersecting point, the condition for the operating conditions depends on the values of x_1, x_2, x_3, and x_4 mentioned earlier. However, there can be infinite number of combinations between the x_1, x_2, x_3, and x_4 that give the same result for a control zone. Figure 6.2 provides two different shapes of the control zone that may produce the same result. The control zone on the left has all four boundaries at equal distances from the origin. The control zone at the right has its vertical boundaries shortened and the horizontal boundaries lengthened. This is the first design consideration for the control zones.

The next constraint is that, the pick up points of each work station should be on the boundary of a control zone. Therefore, the boundaries of the control zones should be adjusted such that they cover one or more pick up points if possible. The designs of the control zones are based on the above two considerations. The following sections provide four design rules that are helpful in deciding the dimensions and locations of the control zones.

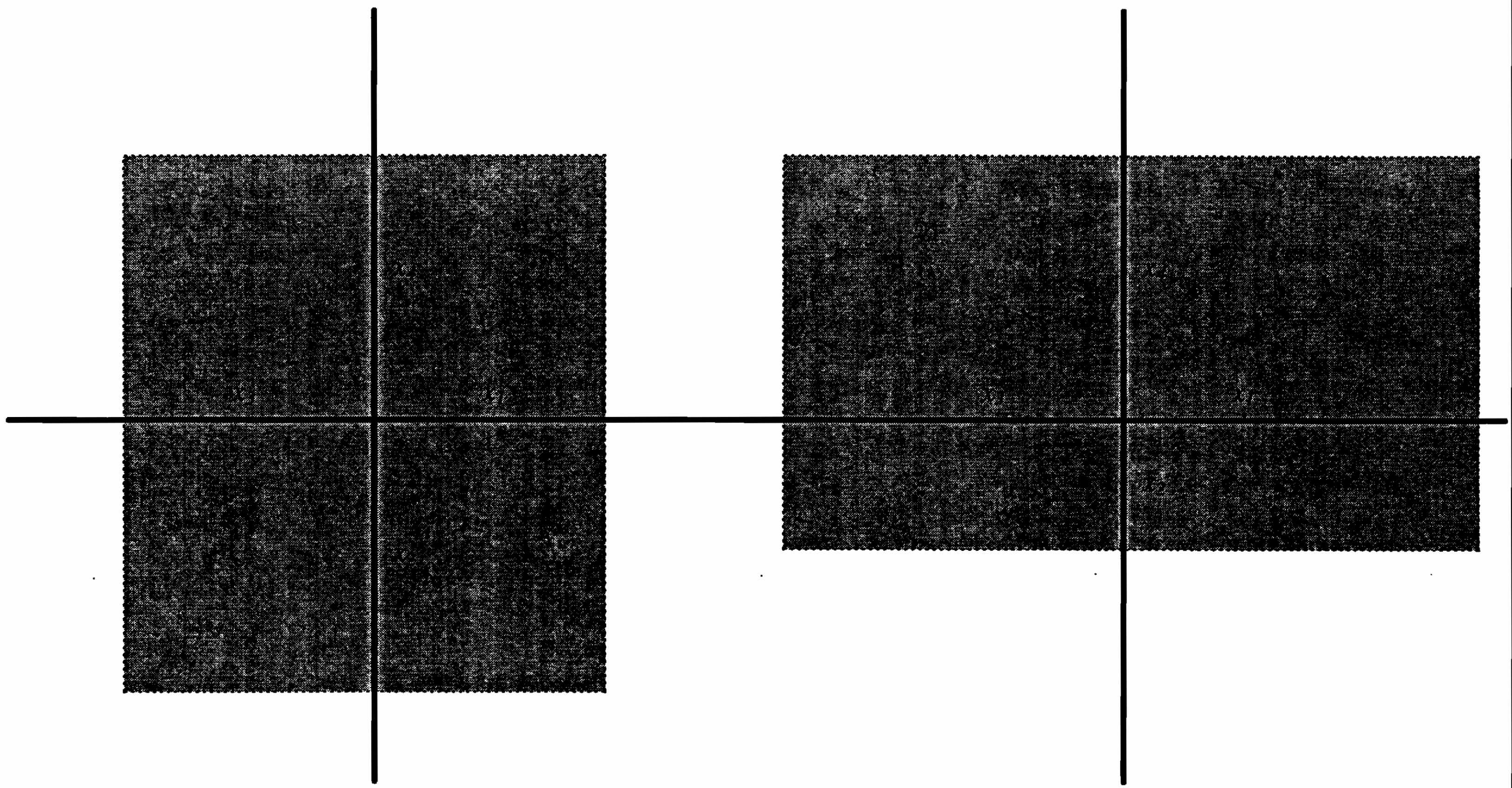


Figure 6.2 Illustration of two control zones which can satisfy the identical conditions for the maximum operating conditions of the AGV system

6.2.1.1 Design rule 1

Define the control zone at the intersecting point as square surrounding the intersecting point. In other words define the control zone with x_1, x_2, x_3, and x_4 equal to each other. This will help the control zone to be of a uniform shape. Once x_1, x_2, x_3, and x_4 are made equal to each other, their values are computed which satisfy the condition for the maximum operating conditions of the system..

6.2.1.2 Design rule 2

When the control zone dimensions are computed using rule 1, it may be possible that pick up points of one or more control zones lie within the control zone. In this case the control zone is adjusted such that the pick up points fall on its boundary. This is illustrated in Figure 6.3. In this figure, the control zone on the left was designed according to rule 1. When the control zone was designed according to this rule, pick up points of two work stations fall inside this control zone. The figure on the right shows the new design after adjusting the control zone so that the pickup points fall on the control zone boundary.

When the control zone is adjusted, the distances x_1, and x_3 become shorter. Therefore, the distances x_2 and x_4 can be made longer without violating the condition for the system to operate at its maximum rate. In this example, the new distances are calculated by setting x_1, and x_3 equal to the distances of the pick up points from the intersecting point. The distances x_2, and x_4 are set equal to each other to facilitate the computation of their values.

6.2.1.3 Design rule 2

In some cases, when a control zone is designed according to rule 1, one or more pick up points may fall outside its boundary. In this case, the boundary of the control zone is extended in the direction of the pick up points while shortening

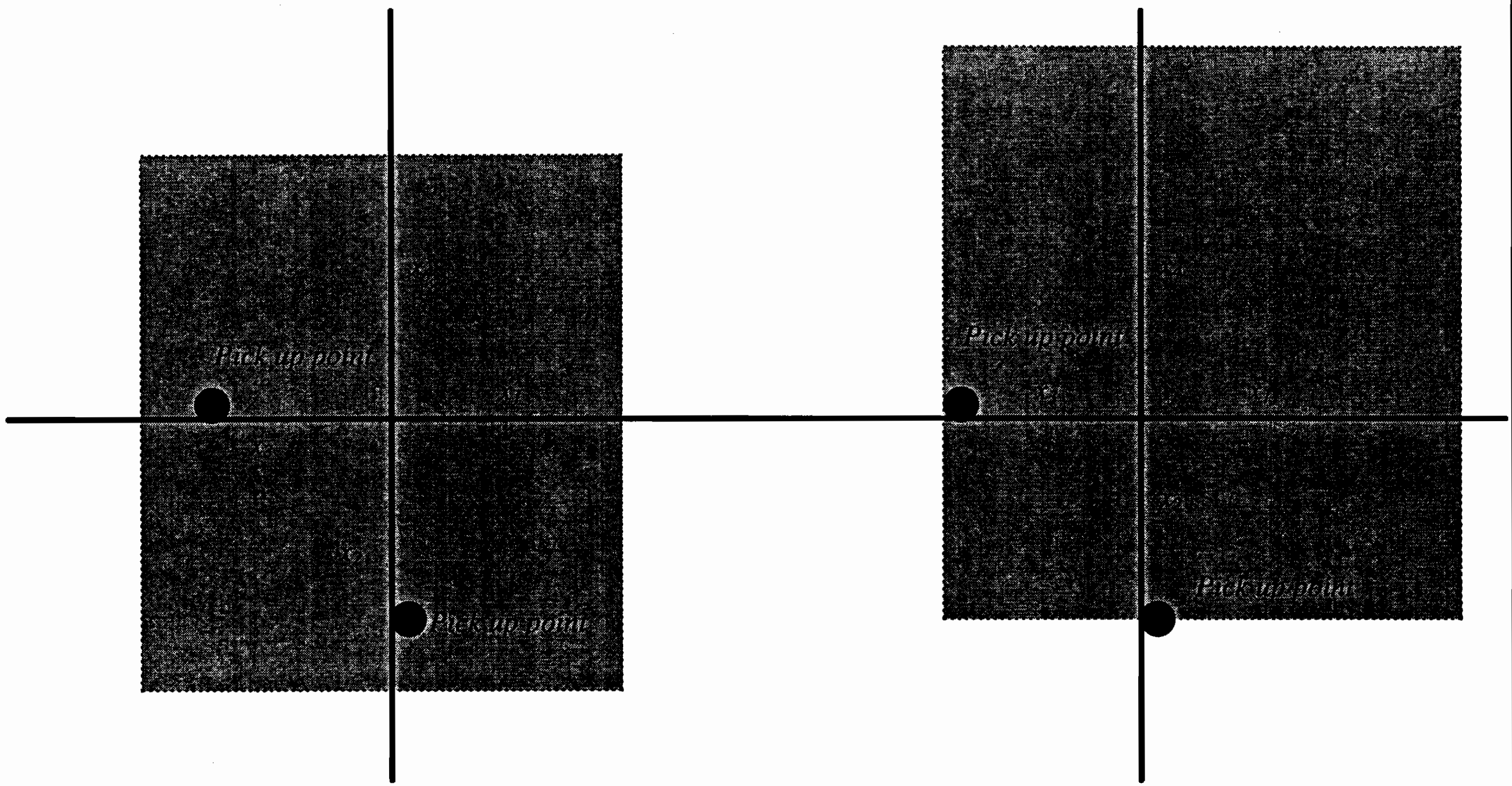


Figure 6.3 Illustration of the adjustment of a control zone according to design rule 2

the boundaries in the remaining directions. However, this should not be done at the expense of the of the uniform shape of the control zone. Therefore a limit should be set for the distance to which a control zone may be extended to cover a pick up point. This rule is illustrated in Figure 6.4. The pick up points of two work stations fall outside the control zone designed according to rule 1. The boundaries of the control zone in the directions of the pick up points are extended. In order to extend these boundaries, x_2, and x_4 are shortened so that the condition for the maximum operating condition of the system is satisfied.

6.2.1.4 Design rule 4

When two intersecting points of a guide path are very close to each other, the control zones should be designed so that they do not overlap. This restriction is made to facilitate a control zone between the two intersecting points. This design rule is illustrated in Figure 6.5. This figure shows two intersecting points of a guide path that are close to each other. In this case, the distances x_{11} and x_{23} are limited such that a control zone can be incorporated between these two control zones. The control zone between the intersecting points is shown by the thicker line in Figure 6.5. As a rule of thumb, these distances may to be set equal to one third of the distance between the two intersecting points.

6.2.2 DESIGN OF CONTROL ZONES ON THE STRAIGHT PORTIONS OF THE GUIDE PATH

The control zones along the straight portions of the guide path are the segments of the guide path that connect the control zones surrounding the intersecting point. Figure 6.6 illustrates a control zone between two control zones at two intersecting points. In this figure, the control zone covers a distance L. The maximum distance a control zone can cover is computed using the expression for the maximum operating condition of the system. If this distance is L_{max} and if $L_{max} < L$, more than one control zone is necessary to cover this distance.

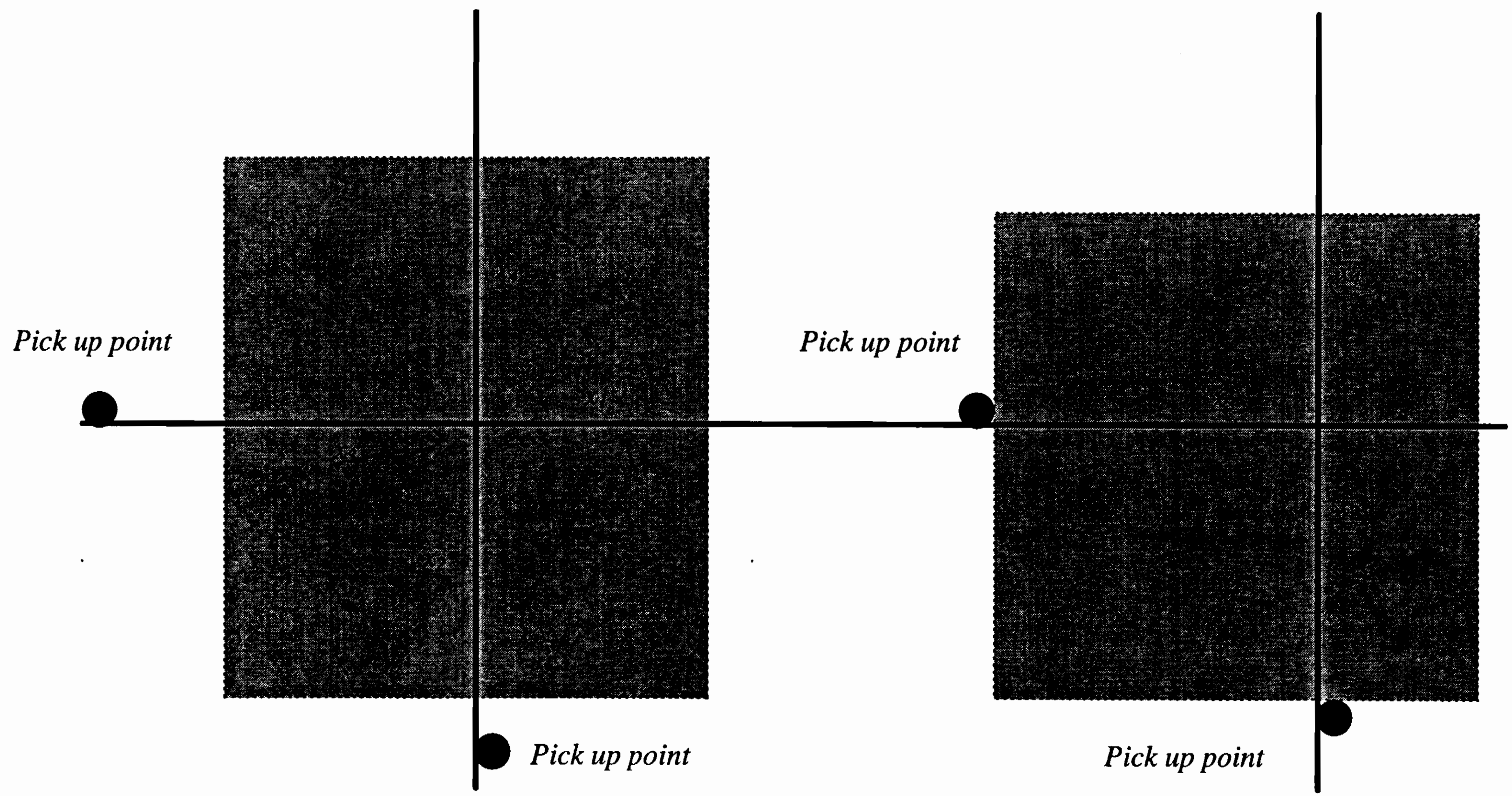


Figure 6.4 Illustration of the adjustment of a control zone according to design rule 3

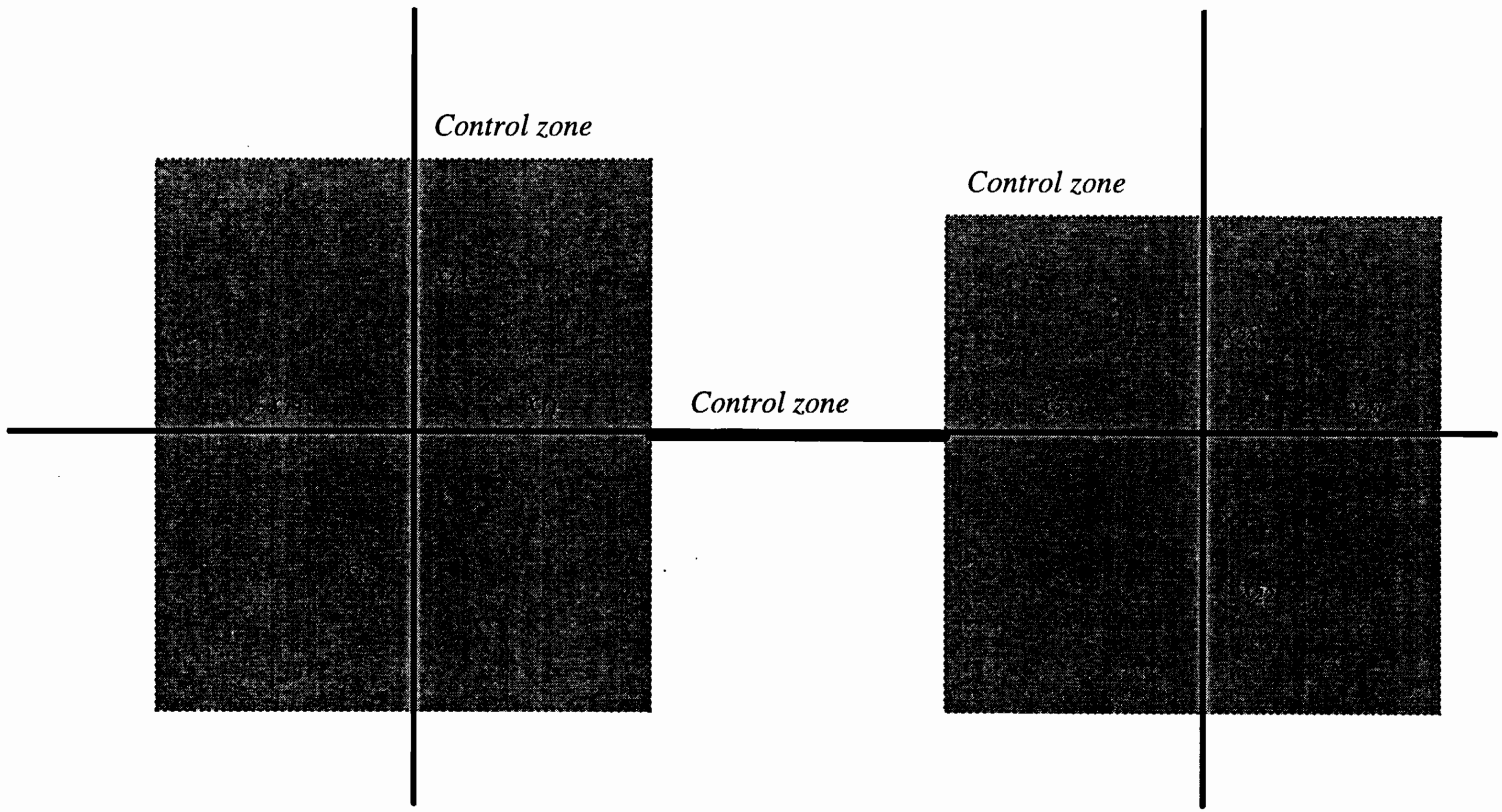


Figure 6.5 Illustration of two control zones which are close to each other in a guide path

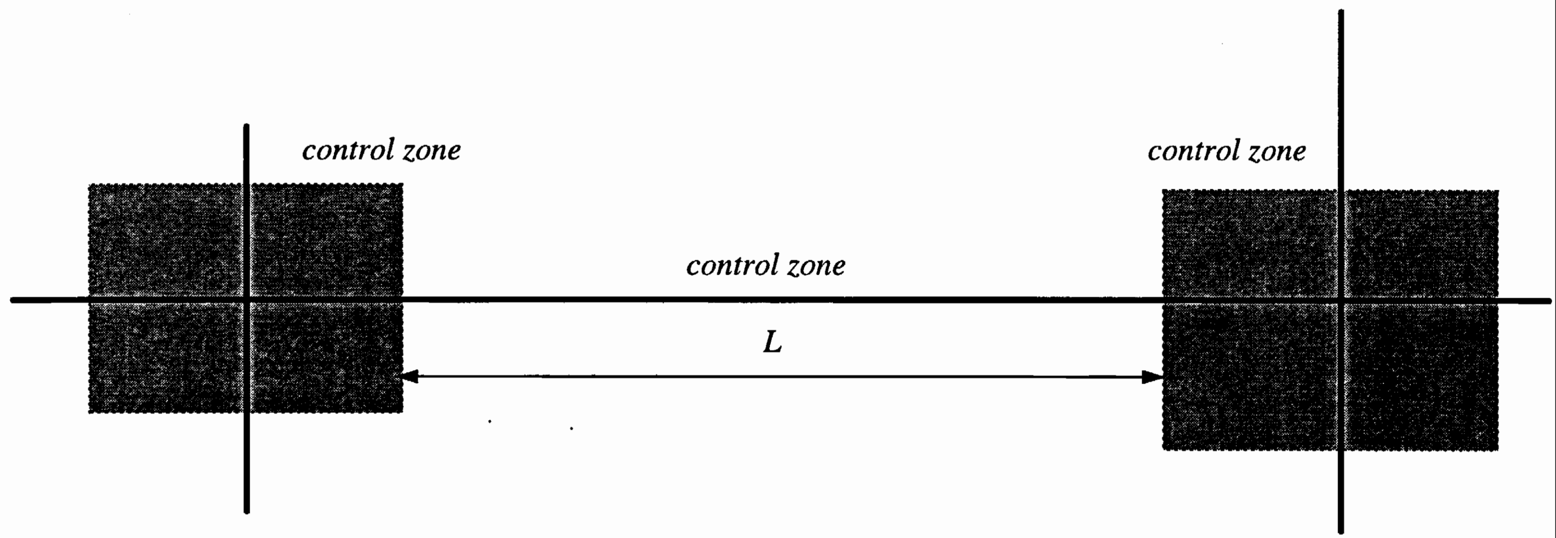


Figure 6.6 Illustration of a control zone between two intersecting points of the guide pat

7. COMPUTATIONAL RESULTS

This chapter focuses on the application of the theoretical results developed so far. In this chapter, two examples of AGV systems are given to show the applications of the expressions developed in Chapter 5 in computing the fleet size and the control zone dimensions. In Section 7.1, the method of computing the fleet size of a single loop AGV system is presented. The next section shows how the dimensions of the control zones are computed for the single loop AGV system.

Sections 7.3 and 7.4 show the method of computing the fleet size and the control dimensions for a multi loop AGV system. In these two sections, an example of an AGV system appearing in the research paper by Egbelu [17] is chosen for the computations. This example consists of 7 work stations with 5 different routes. In this paper Egbelu [17] introduced 4 analytical procedures to compute the fleet size of an AGV system. The results of the 4 analytical procedures of Egbelu [17] are compared with the results obtained using the Petri net model.

7.1 DETERMINING THE FLEET SIZE AND CONTROL ZONE PARAMETERS FOR A SINGLE LOOP AGV SYSTEM

In this section, a simple AGV system with four work stations is considered. The layout of the AGV system is illustrated in Figure 7.1. It is a layout with a single loop with the work stations located along this loop. A single type of product is produced with the product being processed on all four machines. The fleet size and the control zone dimensions are computed for the AGV system for a given material flow requirement. The fleet size and the control zone dimensions are computed using the expression derived in Sections 5.7.2 and 5.8 respectively.

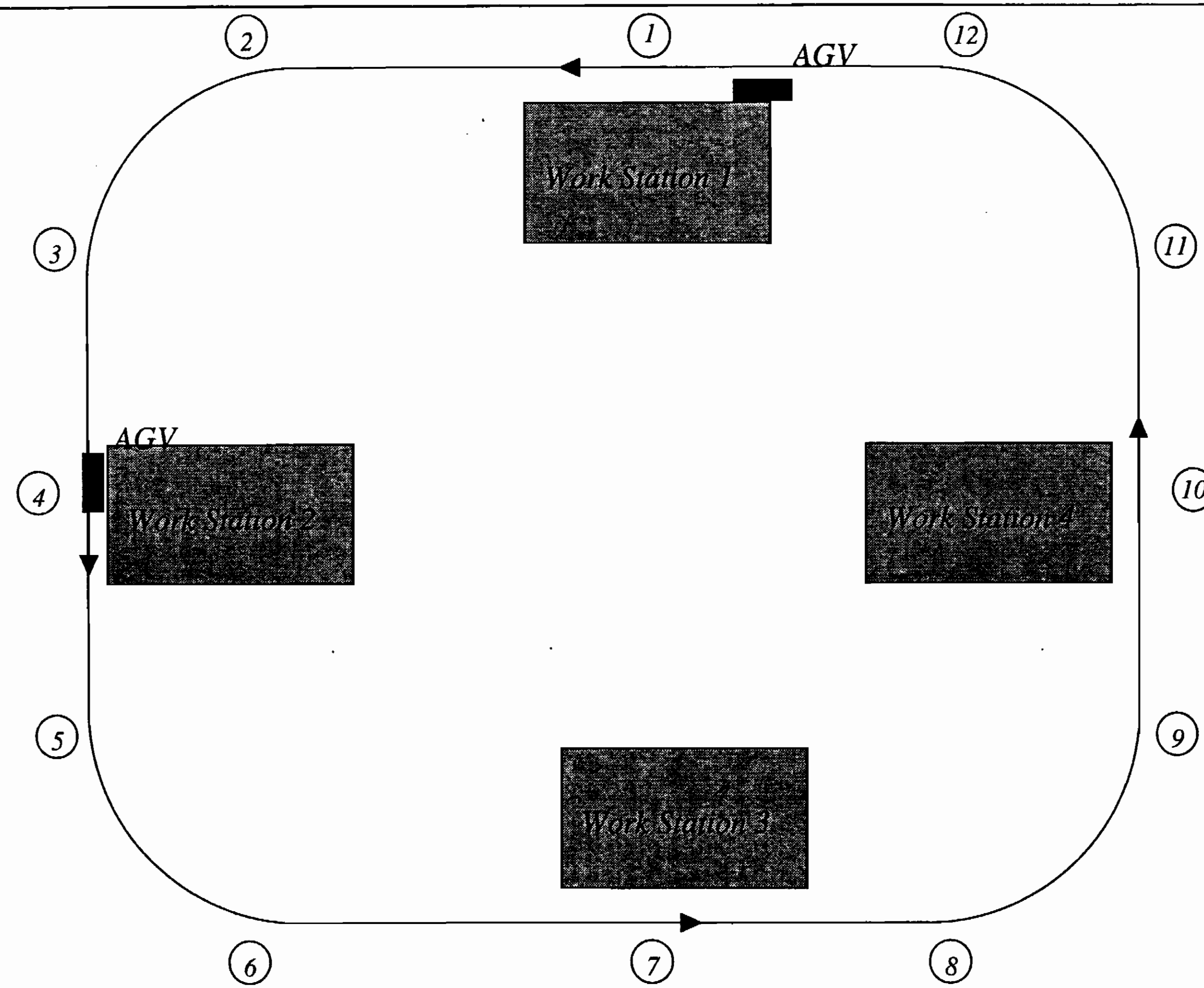


Figure 7.1 Illustration of a single loop AGV system used in the example of Section 7.1

7.1.1 DESCRIPTION OF THE EXAMPLE PROBLEM

The work stations in this example are numbered as 1, 2, 3, and 4. The AGVs move along the route 1→2→3→4→1. In order to meet the material flow requirement, this route should be completed every 12 minutes. The AGVs move at a speed of 200 feet per minute along the straight portions of the guide path and at a slower speed of 50 minute per minute along the curved portions. The time taken to pick up and deliver a part at each work station takes 20 minutes. It is necessary to compute the fleet size and the control zones dimensions.

7.1.2 COMPUTATION OF THE FLEET SIZE

The expression used for the computation of the fleet size is based on the expression $N \geq \sum_{t_{ij} \in TSN, t_{ijd} \in DSN} \frac{\tau_{ij}}{\tau_{ijd}}$. In this expression, N represents the number of AGVs needed in the system, τ_{ijd} is the time between consecutive demands for AGVs between the two work stations and τ_{ij} is the time taken for the AGV to move between the two work stations. Therefore, these two components should be computed for each pair of work stations between which the AGVs move.

As mentioned earlier, the route should be completed once every 12 minutes. Hence, there will be a demand for an AGV between each pair of work stations every 12 minutes. Therefore, τ_{ijd} is equal to 12 minutes for any pair of work stations ij.

Next, the time an AGV spends between each pair of work stations should be computed. Since the AGVs travel at 200 feet per minute along the straight portions and at 50 feet per minute along the curved portions, the guide path should be divided accordingly. As shown in Figure 4.1, the guide path is divided into 12 sections. The eight sections (1,2), (2,3) (3,4), (4,5), (5,6), (6,7), (7,8), (8,9), (9,10), (10,11), (11,12) and (12,1) represent the straight portions and the four sections (2,3), (5,6), (8,9) and (11,12) form the curved portions.

Sections (1,2), (2,3) and (3,4) are between work stations 1 and 2, sections (4,5), (5,6) and (6,7) are between work stations 2 and 3, sections (7,8), (8,9) and (9,10) are between work stations 3 and 4 and sections (10,11), (11,12) and (12,1) are between work stations 4 and 1. The time taken for an AGV to travel between these work stations are computed using the distances of these sections and the speed at which the AGVs travel along these sections.

Table 7.1 provides the length of each section defined in the guide path and the speed at which the AGVs travel along those sections. The data in Table 7.1 is translated into Table 7.2 which gives the time an AGV spends between each pair of work stations. The total time spent between each pair of work stations is the sum of the travel time and the pick up and delivery time. This is the time τ_{ij} associated to transition t_{ij} of the transportation sub net. For this example, the total time an AGV spends between each pair of work stations is 3 + 20 minutes. = 23 minutes. Figure 7.2 illustrates the transportation sub net for the example problem.

Now, the minimum number of AGVs needed for this system is computed using the expression $N \geq \sum_{t_{ij} \in TSN, t_{ijd} \in DSN} \frac{\tau_{ij}}{\tau_{ijd}}$. Hence, the minimum number of vehicles needed is 4x23/12 = 7.6 $\approx$ 8.

Table 7.1 The distance and travel time between each pair of points in the layout of the example AGV system of Section 7.1

Origin point	*Destination point*	*Distance between origin and destination points in feet*	*Speed of the AGV in feet per minute*	*Time taken for an AGV to cover the distance in minutes*
1	2	100	200	0.5
2	3	100	50	2.0
3	4	100	200	0.5
4	5	100	200	0.5
5	6	100	50	2.0
6	7	100	200	0.5
7	8	100	200	0.5
8	9	100	50	2.0
9	10	100	200	0.5
10	11	100	200	0.5
11	12	100	50	2.0
12	1	100	200	0.5

Table 7.2 The time an AGV spends between each pair of work stations in the example AGV system of Section 7.1

Origin work station	*Destination work station*	*Origin point*	*Destination point*	*Total travel time in minutes*	*Pick up and delivery time in minutes*	*Total time spent between the two work stations in minutes*
1	2	1	4	3.0	20	23
2	3	4	7	3.0	20	23
3	4	7	10	3.0	20	23
4	1	10	12	3.0	20	23

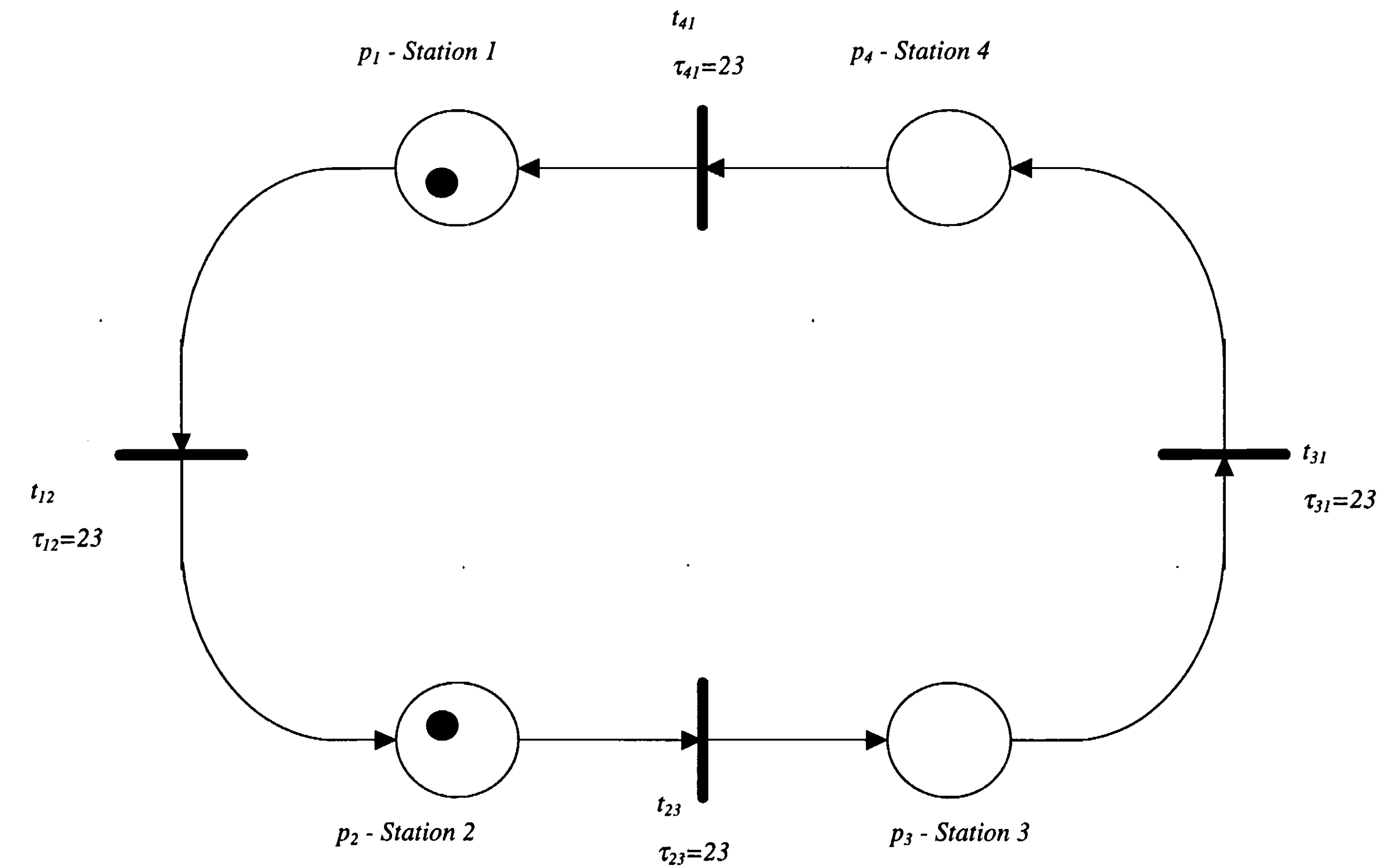


Figure 7.2 The transportation sub net of a single loop AGV system of the example of Section 7.1

7.1.3 COMPUTATION OF CONTROL ZONE DIMENSIONS

The dimensions of the control zones are computed using the control sub nets. Figure 7.3 illustrates the Petri net representation of a control zone between a pair of work stations i and j. Initially, the minimum number of control zones are computed based on the fact that there should be at least one control zone between each pair of work stations. Since there are four work stations, there should be at least four control zones. The time taken for an AGV to move between each pair of work stations is 3 minutes (ref. Table 7.1).

The condition for the maximum operating conditions of the AGV system is given by $1 \geq \sum_{ij \in z} \frac{\tau_{ijze}}{\tau_{ijd}}$. Where, τ_{ijze} is the time taken for the AGV to clear the control zone and τ_{ijd} is the time between consecutive demands for an AGV to be moved from work station i to work station j.

The time between consecutive demands between each pair of work stations is 12 minutes. Hence, the condition for each control zone is $\sum_{ij \in z} \frac{\tau_{ijze}}{\tau_{ijd}}$ is 3/12 = 0.25. Since this value is less than 1, the condition for the operating conditions of the AGV system is satisfied. Therefore, the four control zones between the four work stations will facilitate proper flow of AGVs and at the same time meet the demand of 12 minutes for each trip.

This example establishes the fact that the number of AGVs needed in the AGV system is independent of the control zone restrictions under deterministic conditions. On the other hand the control zone dimensions play a vital role in meeting the material flow requirements.

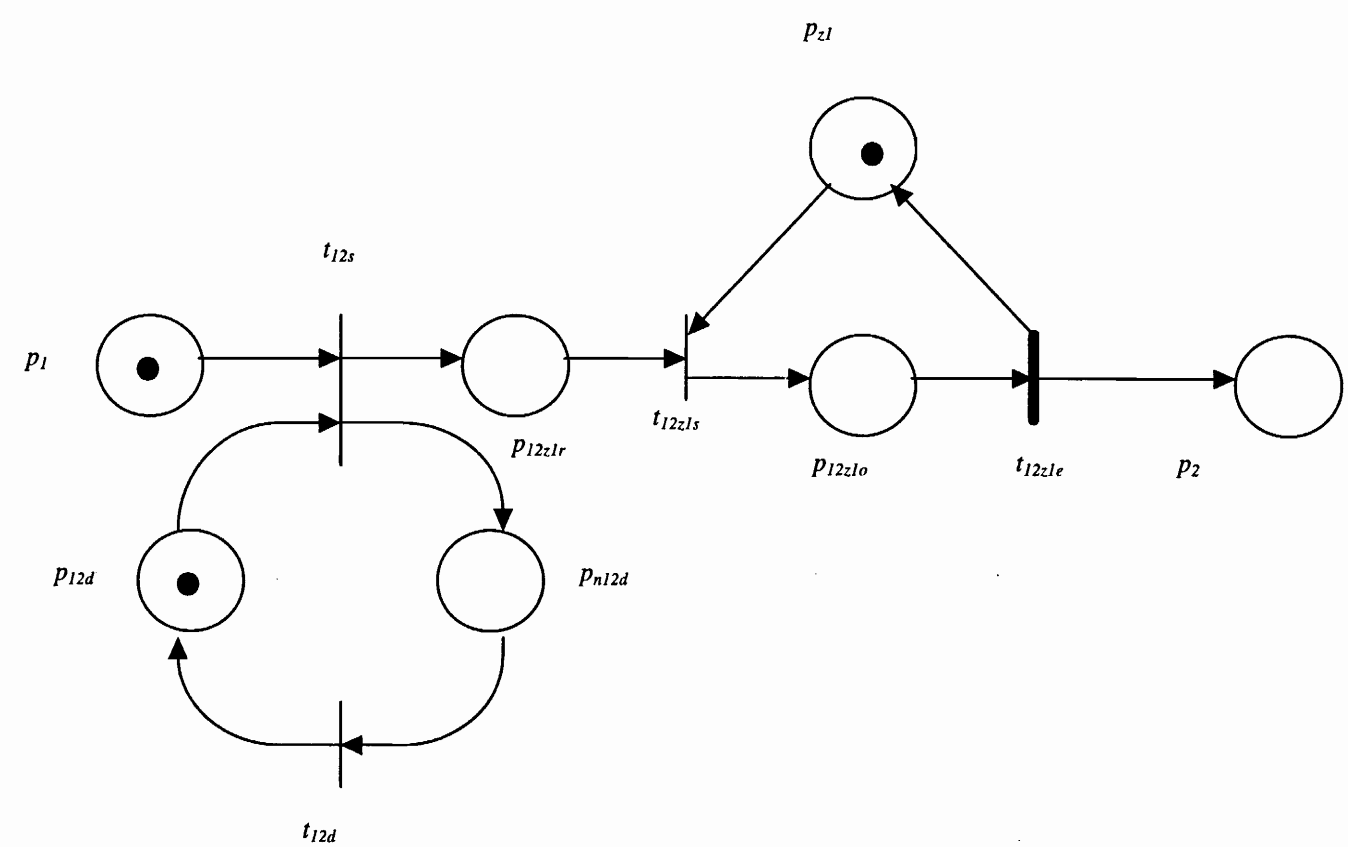


Figure 7.3 Illustration of a control subnet of a single loop AGV system of the example of Section 7.1

7.2 COMPUTING THE FLEET SIZE AND THE CONTROL ZONE PARAMETERS FOR MULTI LOOP AGV SYSTEM

The example studied in this section is from the research paper by Egbelu [17] used for the application of four non simulation approaches for the computation of the fleet size of an AGV system.

7.2.1 DESCRIPTION OF THE EXAMPLE PROBLEM

The guide path layout of this example is illustrated in Figure 7.5. This AGV system consists of eight work stations . Five different routes are employed for the production of five types of products. Table 7.3 gives the five different routes of this AGV system.

The distances between the pick up and delivery points of the work stations are given in Table 7.4 which is reproduced from the research paper by Egbelu [17].. In Table 7.4, α_i represents the delivery point of work station i and b_i represents its pick up point. Hence, $d(\beta_3,\alpha_4)$ is the distance between the pick up point of work station 3 and the delivery point of work station 4. The distances provided in this paper are given up to the second decimal place. The reasons for these values are that the pick up and delivery points in the layout had spurs which allow the loading and unloading AGVs to move away from the main guide oath and thus do not obstruct the other vehicles. The spurs are very short sections of the guide path with curved ends and the distances are computed up to the second decimal. The trip between two work station commences at the pick up point of the origin work station and ends at the pick up point of the destination work station. The total travel in this trip includes the movement from the pick up point of the origin work station to the delivery point of the destination work station and the

Table 7.3 Details of the routes of the example AGV system

Route #	*Route*	*Expected number of loads per 8 hour shift*
1	1→3→2→5→8→1	40
2	1→6→5→4→7→8→1	40
3	1→4→6→8→1	40
4	1→7→2→3→8→1	40
5	1→2→6→3→5→7→4→8→1	40

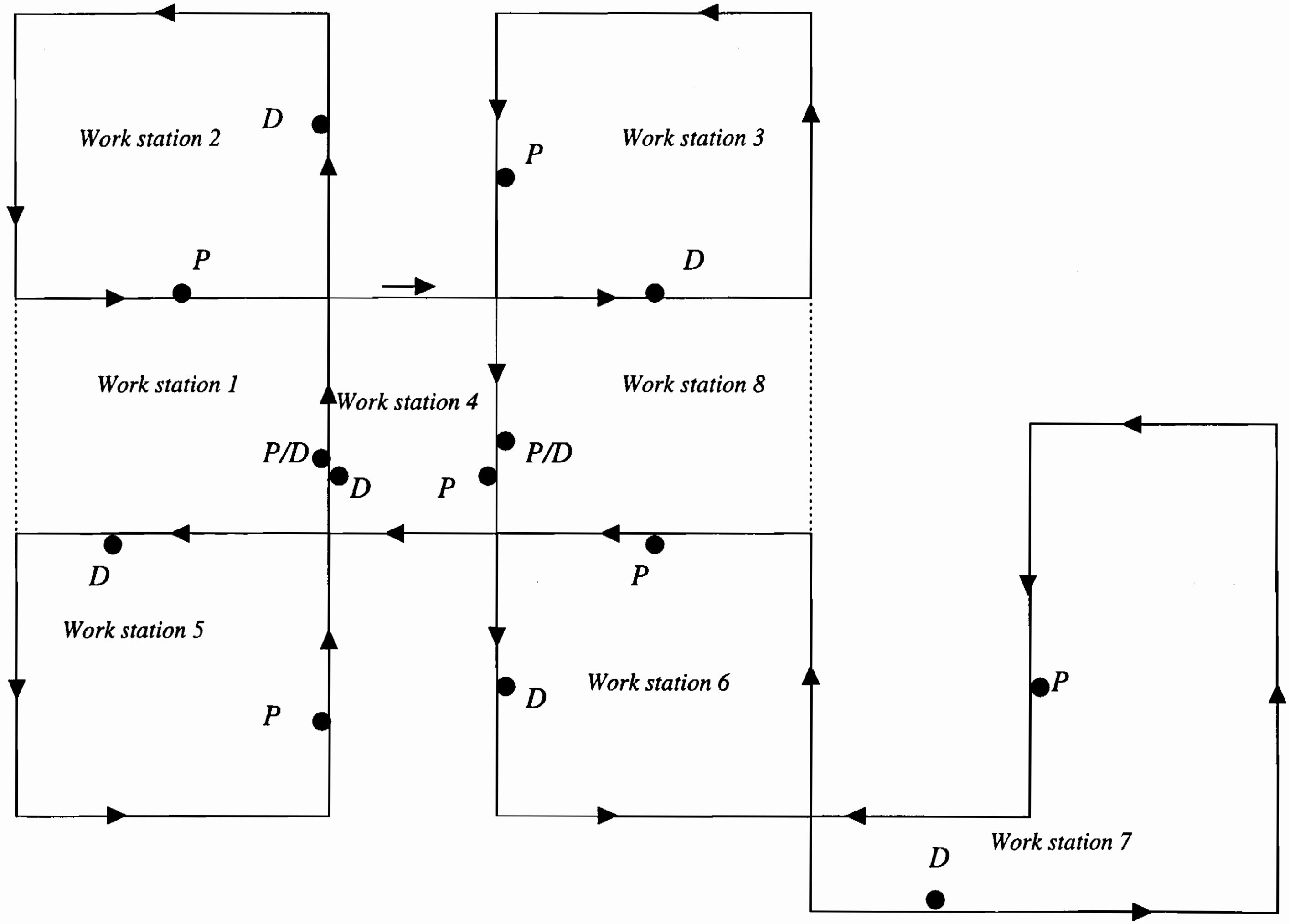


Figure 7.4 Illustration of the guide path layout of the example AGV system of Section 7.2

Table 7 1 The from to matrix of distance between load transfer stations of the example problem of Section 7.2
Source : Egbelu [17]

		WS 1		WS 2		WS 3		WS 4		WS 5		WS 6		WS 7		WS 8	
		α_1	β_1	α_2	β_2	α_3	β_3	α_4	β_4	α_5	β_5	α_6	β_6	α_7	β_7	α_8	β_8
WS																	
1	α_1	0.0	0.0	136.84	418.97	161.55	453.68	330.39	181.55	340.39	592.52	315.68	552.52	477.81	809.94	181.55	181.55
	β_1	0.0	0.0	136.84	418.97	161.55	453.68	330.39	181.55	340.39	592.52	315.68	552.52	477.81	809.94	181.55	181.55
2	α_2	582.52	582.52	0.0	294.84	413.68	705.81	582.52	433.68	592.52	844.65	567.81	804.65	729.94	1062.1	433.68	433.68
	β_2	300.39	300.39	106.84	0.0	131.55	423.68	300.39	151.55	310.39	565.52	285.68	522.52	477.81	809.94	151.55	151.55
3	α_3	557.81	557.89	681.94	964.07	0.0	304.84	557.81	408.97	567.81	819.94	543.10	779.94	705.23	1037.9	408.97	408.97
	β_3	265.68	265.58	389.91	671.94	96.84	0.0	265.68	116.84	275.68	527.81	250.97	487.81	413.10	745.23	116.84	116.84
4	α_4	330.39	330.39	136.84	418.97	161.55	453.68	0.0	181.55	340.39	592.52	315.68	552.52	447.81	809.94	181.55	181.55
	β_4	161.55	161.55	285.68	567.81	310.39	602.52	161.55	0.0	171.55	423.68	146.84	383.86	308.97	641.10	330.39	330.39
5	α_5	378.97	378.97	503.10	785.23	527.81	819.94	378.97	547.81	0.0	264.84	681.94	918.78	844.07	1176.2	547.81	547.81
	β_5	126.84	126.84	250.97	533.10	275.68	567.81	126.84	295.68	136.84	0.0	429.81	666.65	591.94	924.07	295.68	295.68
6	α_6	388.89	388.89	512.52	794.65	537.23	829.36	388.39	557.23	398.39	650.52	0.0	249.55	174.84	506.97	557.23	557.23
	β_6	151.55	151.55	275.68	557.81	300.39	592.52	151.55	320.39	161.55	413.68	136.84	0.0	298.97	631.10	320.39	320.39
7	α_7	690.52	690.52	814.65	1096.8	839.36	1131.5	690.62	859.36	700.52	952.65	675.81	551.68	0.0	344.84	859.36	859.36
	β_7	358.39	358.39	482.52	764.65	507.23	799.36	358.39	527.23	658.39	620.52	343.68	219.55	144.84	0.0	527.23	527.23
8	α_8	161.55	161.55	285.68	567.81	310.39	602.52	161.55	330.39	171.55	423.68	146.84	383.68	308.97	641.10	0.0	0.0
	β_8	161.55	161.55	285.68	567.81	310.39	602.52	161.55	330.39	171.55	423.68	146.84	383.68	308.97	641.10	0.0	0.0

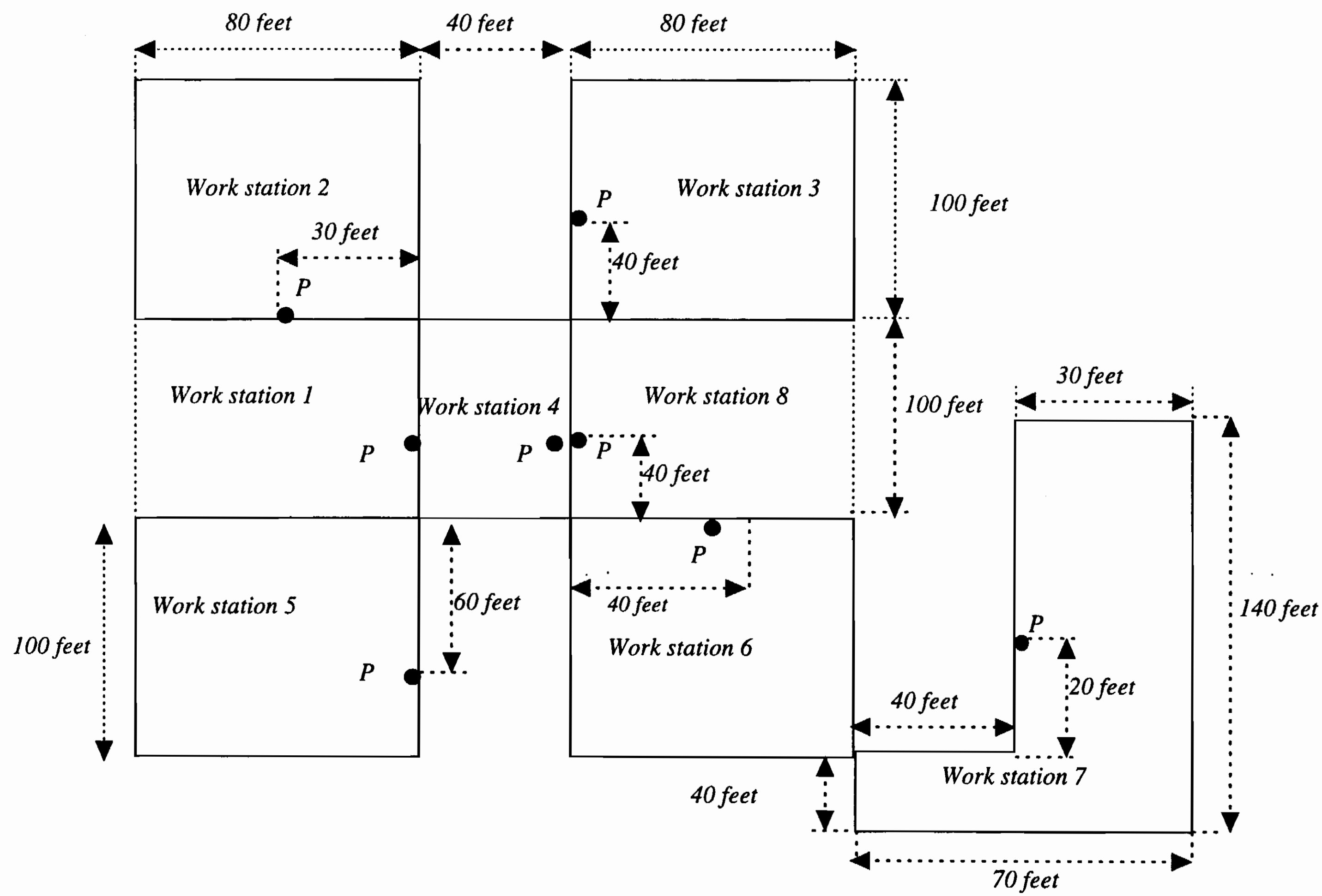


Figure 7.5 Illustration of the dimensions of the layout of the example AGV system

7.2.2 COMPUTATION OF THE FLEET SIZE

The fleet size for the AGV system is computed using the expression $N \geq \sum_{t_{ij} \in TSN, t_{ijd} \in DSN} \frac{\tau_{ij}}{\tau_{ijd}}$. The value of τ_{ijd}. is the time between demands for AGVs between work stations i and j. The time between demands for each pair of work stations is computed by preparing a from-to matrix as shown in Table 7.4. This table is constructed using the data provided in Table 7.3 which gives the work stations to which the AGVs move while traveling along each of the 5 routes. In Table 7.4, it is seen that there are 24 non-zero entries. These 24 non-zero entries correspond to the 24 pairs of work stations between which the AGVs are required to move. The total number of work stations in this AGV system is 8. Therefore, the transportation sub net representing this AGV system should have 8 places to represent the work stations The 24 transitions to represent the movement of AGVs between the work stations. This transportation sub net of the AGV system is illustrated in Figure 7.6. Each non-zero element of Table 7.4 gives the number of times the AGVs move between each pair of work stations in an 8 hour shift. For example, the AGVs move 40 times per 8 hour shift from work station 1 to work station 2. Hence the time between demands from work station 1 to 2 is 8x60/40 = 12 minutes. In other words the value of τ_{12d} = 12 minutes.

The next component needed in the computation of the fleet size is the time an AGV spends between each pair of work stations. This time is computed using the distance between the pick up points of the work stations, the speed of the AGVs, and the pick up and delivery times.

Table 7.5 The from to matrix representing flow of AGVs between the work stations Source: Egbelu [17]

	To							
	1	2	3	4	5	6	7	8
From								
1	0	40	40	40	0	40	40	0
2	0	0	40	0	40	40	0	0
3	0	40	0	0	40	0	0	40
4	0	0	0	0	0	40	40	40
5	0	0	0	40	0	0	40	40
6	0	0	40	0	40	0	0	40
7	0	40	0	40	0	0	0	40
8	200	0	0	0	0	0	0	0

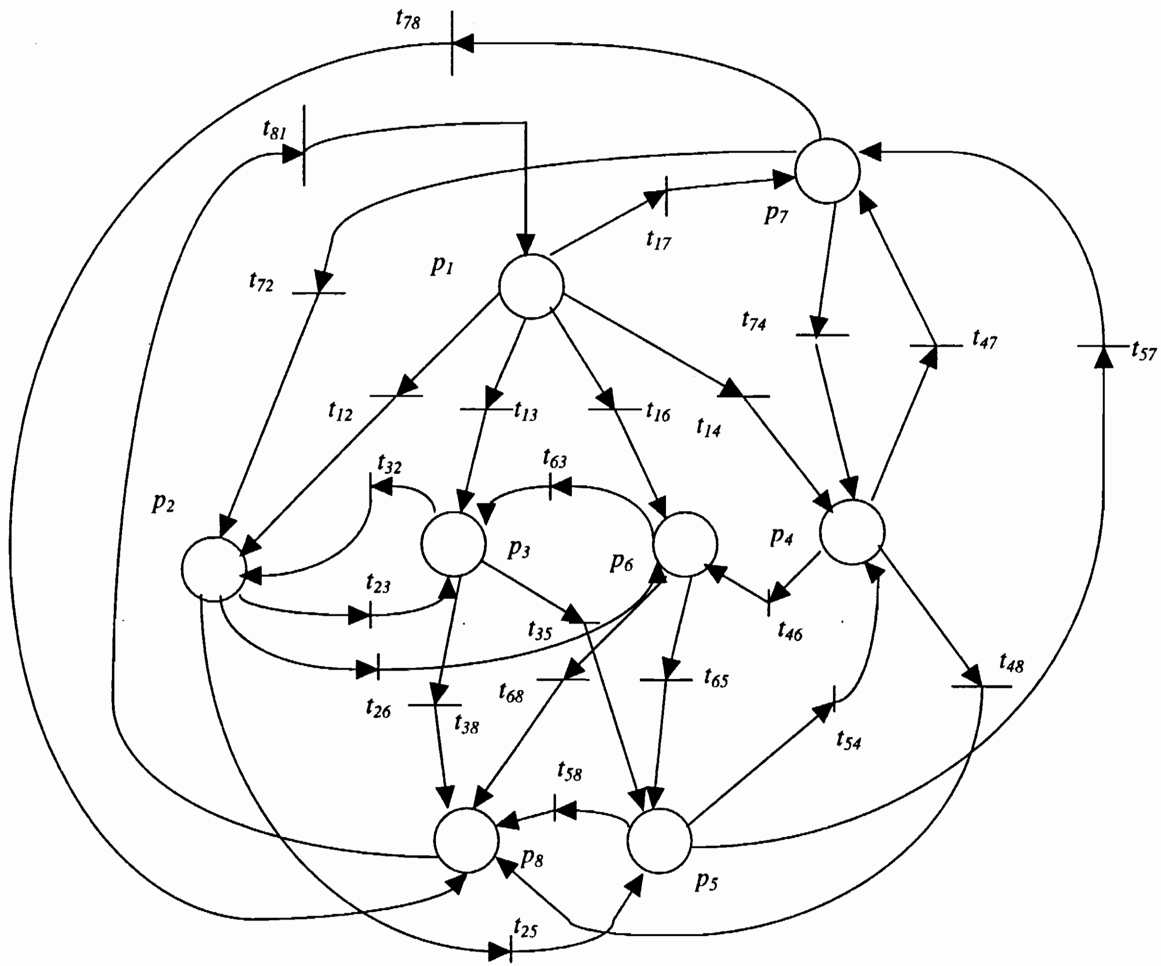


Figure 7.6 Illustration of the transportation sub net of the example problem of Section 7.2

The distances between the pick up points of the work stations are computed using Table 7.4. These distances are computed and are tabulated in Table 7.6. Table 7.7 gives The values of τ_{ijd} and τ_{ij} . The summation of the last column of this table gives the value of the expression $\sum_{t_{ij} \in TSN, t_{ijd} \in DSN} \frac{\tau_{ij}}{\tau_{ijd}}$. For the example of this section, this value is 8.49. Hence, fleet size needed by the system is 9.

7.2.3 COMPARISON OF THE RESULTS OBTAINED USING THE PETRI NET ANALYSIS WITH THE NON SIMULATION APPROACHES OF EGBELU [17]

In this section, the four non simulation approaches of Egbelu [17]are presented and compared with the Petri net approach.

7.2.3.1 Method 1

The expression used in this method for the computation of the fleet size is as follows

$: N = \left[2\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{f_{ij}d(\beta_i, \alpha_j)}{V} + \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}(t_l + t_u) \right] \left((60T - t)e\right)^{-1}$. Where f_{ij} is the expected number of loaded trips required between work stations i and j, $d(\beta_i,\alpha_j)$ is the distance between the pick up point of work station i and the drop off point of work station j, V is the velocity of the AGV, t_l and t_u are the pick up and delivery times, T is the length of the period during which the f_{ij} trips occur, t is the expected time lost due to battery change and e efficiency of the vehicle. The number of vehicles computed using this method 8.61 ≈ 9. The major draw back in this method is that the loaded and unloaded travel times are assumed to be the same. The loaded travel is multiplied by 2 to compute the total distance the AGVs in meeting the material flow requirement.

Table 7.6 The from to matrix illustrating the travel distances between the work station pick up points

	To							
	1	2	3	4	5	6	7	8
From								
1	N/A	432	466	512	N/A	802	823	N/A
2	N/A	N/A	436	N/A	575	535	N/A	N/A
3	N/A	685	N/A	N/A	541	N/A	N/A	409
4	N/A	N/A	N/A	N/A	N/A	396	654	330
5	N/A	N/A	N/A	308	N/A	N/A	937	296
6	N/A	N/A	605	N/A	426	N/A	N/A	320
7	N/A	777	N/A	N/A	N/A	N/A	N/A	527
8	162	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 7.7 The firing rates and times associated to the transitions of the transportation sub net

Transition t_{ij}	Time associated to t_{ij} (τ_{ij})	Time associated to t_{ijd} (τ_{ijd})	τ_{ij}/ τ_{ijd}
t_{12}	432/150 + 2x0.25 = 3.38	12	0.28
t_{13}	466/150 + 2x0.25 = 3.61	12	0.31
t_{14}	512/150 + 2x0.25 = 3.91	12	0.33
t_{16}	802/150 + 2x0.25 = 5.85	12	0.49
t_{17}	823/150 + 2x0.25 = 5.99	12	0.49
t_{23}	436/150 + 2x0.25 = 3.4	12	0.28
t_{25}	575/150 + 2x0.25 = 4.3	12	0.36
t_{26}	535/150 + 2x0.25 = 4.1	12	0.34
t_{32}	685/150 + 2x0.25 = 5.1	12	0.42
t_{35}	541/150 + 2x0.25 = 4.1	12	0.34
t_{38}	409/150 + 2x0.25 = 3.2	12	0.27
t_{46}	396/150 + 2x0.25 = 3.1	12	0.26
t_{47}	654/150 + 2x0.25 = 4.9	12	0.41
t_{48}	330/150 + 2x0.25 = 2.7	12	0.23
t_{54}	308/150 + 2x0.25 = 2.6	12	0.21
t_{57}	937/150 + 2x0.25 = 6.7	12	0.56
t_{58}	296/150 + 2x0.25 = 2.5	12	0.21
t_{63}	605/150 + 2x0.25 = 4.5	12	0.38
t_{65}	426/150 + 2x0.25 = 3.3	12	0.28
t_{68}	320/150 + 2x0.25 = 2.6	12	0.22
t_{72}	777/150 + 2x0.25 = 5.7	12	0.47
t_{74}	540/150 + 2x0.25 = 2.5	12	0.34
t_{78}	527/150 + 2x0.25 = 4.0	12	0.34
t_{81}	162/150 + 2x0.25 = 1.6	2.4	0.67

7.2.3.2 Method 2

The second method first computes average distance per loaded trip. This average distance is given by the expression : $\bar{D} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d(\beta_i, \alpha_j)}{\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}}$. The average distance per loaded trip and the speed of the AGV is used to compute the mean travel time per trip. This mean travel time is given by is given by $t_A = \frac{\bar{D}}{V}$. The mean travel time is then adjusted for blocking and idleness and then added to the pick up and delivery times. This total time is $\bar{t} = \frac{(1+b+c)}{e}.t_A + t_l + t_u$. Where b is the blocking time factor, c is the idle time factor and e is the vehicle efficiency.

The expression for the total number of vehicles is now given by

$$N = \left(\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}}{T} \right) . \left(\frac{60}{\bar{t}} \right)^{-1}.$$

The numerator of this expression gives the average number of trips per hour and the denominator gives the number of trips made per hour per vehicle. The total number of vehicles computed using this was 5.93 ≈6.

7.2.3.3 *Method 3*

This method is based on the computation of the net traffic flow into a work station. The net flow into work station i is given by $f_i = \sum_{j=1}^{n} f_{ji} - \sum_{j=i}^{n} f_{ij}$. The total empty vehicle run between the work stations is computed using the expression $D_1 = \left[\dfrac{\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d(\beta_i, \alpha_j)}{\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}} \right] \cdot \left(\sum_{\forall i, f_i > 0} f_i \right)$.

When the pick up and delivery points of a work station physically separated, the distance due to the empty vehicle transfer between the work station is given by $D_2 = \sum_{i=1}^{n} \left[\min\left\{ \sum_{j=1}^{n} f_{ij}, \sum_{j=1}^{n} f_{ji} \right\} . d(\alpha_i, \beta_i) \right]$ and the total loaded travel distance is $D_3 = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} . d(\beta_i, \alpha_j)$. The values of D_1, D_2 and D_3 are then used to computes the fleet size as given by the following expression

$N = \left[\dfrac{D_1 + D_2 + D_3}{V} + \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}(t_u + t_l) \right] . (60T - t)^{-1}$. The total number of vehicles for this method was 7.97 ≈ 8.

7.2.3.4 *Method 4*

This method is based on the assumption that the number of times a vehicle is re assigned to work station j, given that it has completed a delivery task at work station i is a function of the number of deliveries to work station i, the number of pick up from j and the total number of pick ups during the given period T. The estimate of the total number of empty runs from work station i to j is given by the expression $g_{ij} = \frac{\left(\sum_{k=1}^{n} f_{ki}\right)\cdot\left(\sum_{k=1}^{n} f_{jk}\right)}{\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}}$. The total empty and loaded vehicle travel distances between work stations i and j are computed using the expressions $D_{ij}^{'} = g_{ij}.d(\alpha_i,\beta_j)$ and $\bar{D}_{ij} = f_{ij}.d(\alpha_i,\beta_j)$. The total distance traveled between the two work stations is given by $D_{ij} = D_{ij}^{'} + \bar{D}_{ij}$. The total number of vehicles is calculated according to $N = \left[\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}}{V} + \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}(t_u + t_l)\right].(60T - t)^{-1}$. The total number of vehicles using this method was 12.34 ≈ 13.

7.2.3.5 The Petri net method

As stated earlier, the fleet size of the AGV system is computed using the expression derived from the timed Petri net model. The expression is as follows: $N = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\tau_{ij}}{\tau_{ijd}}$. According to the notations used by Egbelu [17], τ_{ijd} = $60.T/f_{ij}$ (if computed in minutes) and $\tau_{ij} = [d(\beta_i,\alpha_j)+d(\alpha_j,\beta_j)]+(t_u+t_l)]\ V^{1}$.

Hence, the expression for the minimum total number of vehicles can be written as $N = \left[\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{f_{ij}\cdot\left[d(\beta_i,\alpha_j) + d(\alpha_j,\beta_j)\right]}{V} + \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}\cdot[t_u + t_l]\right].(60T)^{-1}$. As it can be observed, the expression obtained using the timed Petri model is very similar to those of the expressions of Egbelu [17]. However, the non simulation approaches of [17] are very approximate in nature. On the other hand the Petri net method provides an exact solution for the fleet size. However, when the dispatching rules are employed under stochastic conditions, this expression will not reflect the actual requirement of the fleet size.

7.3 COMPUTING THE CONTROL ZONE DIMENSIONS

The computation of the dimensions and the locations of the control zones requires much more detail analysis. This section deals with the analysis related to the control zones. Before explaining the computational procedure some aspects related to the control zones of an AGV system are discussed in this section. The information provided below are vital for the design of the control zones.

A control zone can be either at an intersection where two or more lanes meet at a particular point or along the linear segment of the guide path between two work stations. The control sub nets for these two types of control zones were discussed in detail in Section 4.3. The computation of the location and the dimensions of these two types of control zone are performed separately. The location and the dimensions of the control zones at the intersection are performed first. The control zones involving the linear portion of the guide path are done later.

7.3.1.1 The initial set of control zones

The first step in computing the design parameters of the control zone is to identify their possible locations. The intersections of the guide path form the first set of control zones. The next set of control zones are the linear segments of the guide path. The linear segments can either be the portion of the guide path between two control zones at the intersection points or the segment surrounding a work station. The shaded areas of Figure 7.7 illustrate the five intersections of the AGV system layout of this example problem. These intersections are defined by the sequence of points (1,2,3,4), (5,6,7,8), (9,10,11,12), (13,14,15,16), and (17,18,19,20) traced in the clock wise direction in Figure 7.7. The other control zones are the segments between the pairs of points (1,4), (6,5), (13,3), (7,9), (2,8), (12,14), (11,20), (17,10), (16,15), and (19,18). These 15 portions of the guide path layout will form the initial set of control zones.

7.3.2 IDENTIFYING THE CONTROL ZONES USED BY THE VEHICLES WHILE MOVING BETWEEN EACH PAIR OF WORK STATIONS

The next step is to identify the control zones used by the vehicles traveling between the pairs of work stations. For example, as illustrated in Figure 7.8, while an AGV moves from work station 1 to work station 2, it moves through control zones z_9, z_1, and z_{10} in that order. All the control zones related to the example problem are tabulated in Table 7.8.

Figures A.1 to A.23 in the appendix, illustrate all the other movements of AGVs between each pair of work stations and the sequence of control zones they pass through during these movements. These figures are useful in computing the firing rates of the transitions of the control sub nets.

7.3.2.1 Computation of the firing rates of the transitions of the control sub nets

It was shown in the previous sections, that the firing rates of the transitions of the control sub nets is equal to the firing rates of the corresponding transitions of the transportation sub net (i.e. $I_{ijze} = I_{ij}$). Making use of this expression and Figure 7.8 in this chapter and the Figures A.1 to A.23 in the appendix, a table which gives the firing rates of the transitions of each control sub net can be computed. For example, from Figure 7.8, it can be shown that the firing rates of transitions

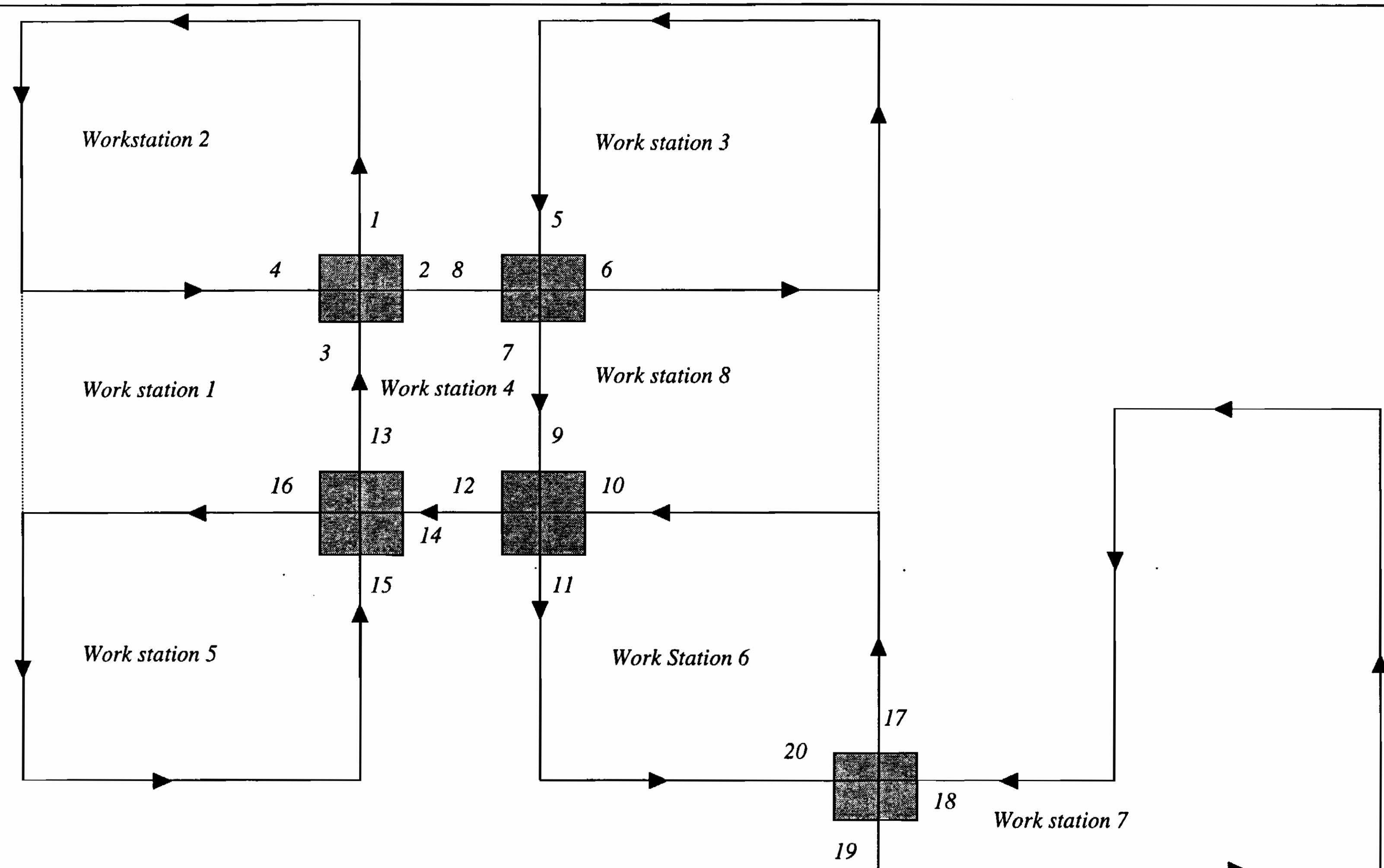


Figure 7.7 Illustration of the control zones at the intersections for the example AGV system of Section 7.2

Table 7. 8 Description of the control zones for the guide path layout of the example problem

Control zone	*The points defining the control zones in Figure 7.7*	*Description*
z_1	1,2,3,4	Intersection
z_2	5,6,7,8	Intersection
z_3	9,10,11,12	Intersection
z_4	13,14,15,16	Intersection
z_4	17,18,19,20	Intersection
z_6	2,8	Between control zones z_1, z_2
z_7	7,9	Between control zones z_2, z_3
z_8	11,14	Between control zones z_3, z_4
z_9	13,3	Between control zones z_4, z_5
z_{10}	1,4	Surrounding work station 1
z_{11}	6,5	Surrounding work station 3
z_{12}	17,10	Between control zones z_5, z_3
z_{13}	11,20	Between control zones z_3, z_5
z_{14}	16,15	Surrounding work station 5
z_{15}	19,18	Surrounding work station 7

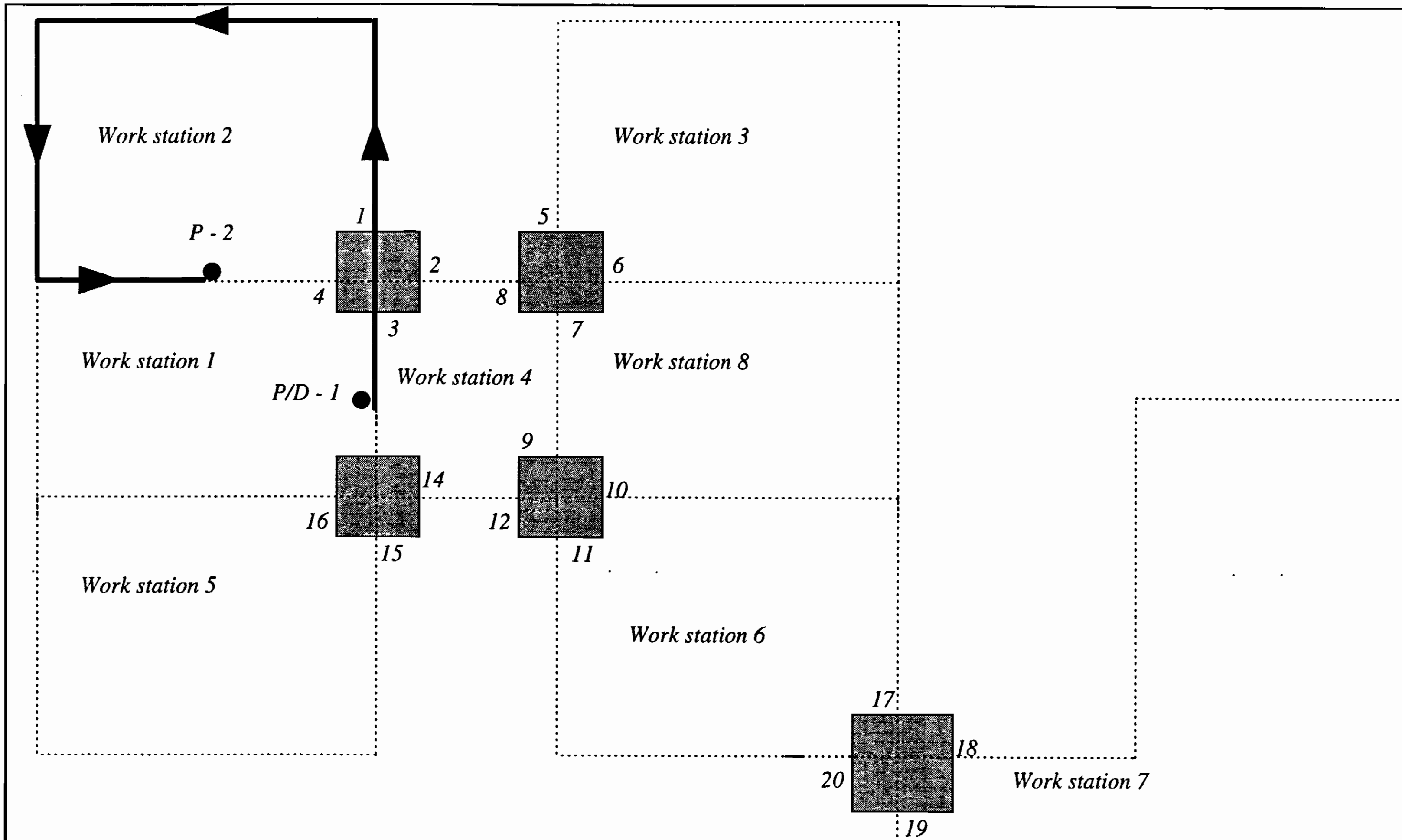


Figure 7.8 Illustration of the movement of an AGV between work stations 1 and 2

t_{12z1e}, t_{12z9e}, and t_{12z10e} as equal to 5 units per hour. This is because the AGVs are expected to complete each route 40 times during a 8 hour shift.

The firing rates corresponding to the transitions of the control sub nets representing the 15 control zones are tabulated in a matrix form as shown in Table 7.9. The rows of this table correspond to the pairs of work stations between which the AGVs move and the columns correspond to the control zones. For example if row (12) is the row representing the pair of work stations 1 and 2 and column(z_3) is the column corresponding to control zone z_3, then the element corresponding to that row and the column represents the firing rate of transition t_{12z3}.

7.3.2.2 Computation of the times associated to the transitions of the control sub nets

Each control zone consists of two sets of transitions. These transitions are denoted as t_{ijzs} and t_{ijze}. These two transition indicate the beginning and the end of the movement of an AGV along the control zone. Transition t_{ijzs} is an immediate transition and no time is associated to it and the time associated to transition t_{ijze} is the time taken for the AGV to clear that control zone This can be expressed mathematically as follows: The time taken to clear control zone z_k = $L_k/V + m_{ik}.D_i$.. Where L_k is the length traveled in the control zone, V is th speed of the AGV and m_{ik} is a variables which takes on a value of 1 if the delivery point of work station i in control zone k and 0 other wise. D_i is the delivery time at work station i.

Table 7. 9 The firing rates of the transitions of the control sub nets corresponding to each control zone of the example AGV system

	The control zones														
	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}	z_{12}	z_{13}	z_{14}	z_{15}
(ij)															
12	5	0	0	0	0	0	0	0	5	5	0	0	0	0	0
13	5	5	0	0	0	5	0	0	5	0	5	0	0	0	0
14	5	5	0	0	0	5	5	0	5	0	0	0	0	0	0
16	5	5	5	0	5	5	5	0	5	0	0	5	5	0	0
17	5	5	5	0	5	5	5	0	5	0	0	0	5	0	5
23	5	5	0	0	0	5	0	0	0	5	5	0	0	0	0
25	5	5	5	5	0	5	5	5	0	5	0	0	0	5	0
26	5	5	5	0	5	5	5	5	0	0	0	5	5	0	0
32	5	5	5	5	0	0	5	5	5	5	5	0	0	0	0
35	0	5	5	5	0	0	5	5	0	0	5	0	0	5	0
38	0	5	0	0	0	0	5	0	0	0	5	0	0	0	0
46	0	0	5	0	5	0	5	0	0	0	0	5	5	0	0
47	0	0	5	0	5	0	5	0	0	0	0	0	5	0	5
48	5	5	5	5	0	5	5	5	5	0	0	0	0	0	0
54	5	5	0	5	0	5	5	0	5	0	0	0	0	5	0
57	5	5	5	5	5	5	5	0	5	0	0	0	5	5	5
58	5	5	0	5	0	5	5	0	5	0	0	0	0	5	0
63	5	5	5	5	0	5	0	5	5	0	5	5	0	0	0
65	0	0	5	5	0	0	0	5	0	0	0	5	0	5	0
68	5	5	5	5	0	5	5	5	5	0	0	5	0	0	0
72	5	0	5	5	5	0	0	5	5	5	0	5	0	0	5
74	5	5	5	5	5	5	5	5	5	0	0	5	0	0	5
78	5	5	5	5	5	5	5	5	5	0	0	5	0	0	5
81	0	0	25	25	0	0	25	25	25	0	0	0	0	0	0

7.3.3 DESIGN OF THE CONTROL ZONES

Once the locations of the initial set of control zones are found, the values of the maximum possible dimensions of each control zone are computed. For the control zones at an intersection, it is the maximum length and width. For a control zone along a linear segment, it is its maximum length. The following sections illustrate the procedure of computing these dimensions. This section shows how control zones z_1 and z_{10} are designed based on the procedure described in this section. Control zone z_1 is at an intersection point and control zone z_{10} is along the linear portion of the guide path. The procedure of designing the rest of the control zones is similar to those of these two control zones.

7.3.3.1 Computing the dimensions of the control zones at the intersecting points

The condition for the system to operate at its maximum rate is given by the expression $1 \geq \sum_{ij \in z} I_{ijze} \cdot \tau_{ijze}$. Where, I_{ijze} and τ_{ijze} are the firing rates and the times associated to the transitions in the control sub nets that represent the end of the movement of an AGV in control zone z.

The initial dimensions of the control zone are computed according to design rule 1 explained in Section 6.2.1. According to this rule, the control zone at an intersecting is set as a square surrounding that intersection. In other words, the distances x_1, x_2, x_3, and x_4 are all set equal to each other. Hence, the time taken for the AGV to clear the control zone is independent of the path it takes in moving inside that control zone. Therefore τ_{ijze} is independent of ij and hence, it may be taken out of the summation sign of the expression stated above and the expression now becomes $\tau_{ijze} \leq \left(\sum_{ij \in z} I_{ijze} \right)^{-1}$. The term on the left hand side of the expression is the time an AGV spends between the pair of work stations. The term on the right hand side is obtained from the inverse of the summation of each column of Table 7.8. As mentioned earlier, the initial dimensions of a control zone are computed using design rule 1, then its dimensions are adjusted, according to design rules 2, 3 or 4, whichever is appropriate.

According to design rule 1, when the dimensions of control zone z are computed, the values of x_1, x_2, x_3, and x_4 are all set equal to each other (say x), their values can be computed using the expression $\tau_{ijze} \leq \left(\sum_{ij \in z} I_{ijze} \right)^{-1}$. For this control zone τ_{ijze} $= 2x/V$. Where, V is the speed of the AGV. The value of the term $\left(\sum_{ij \in z} I_{ijze} \right)^{-1}$ is equal to the sum of the elements of the column corresponding to this control zone in Table 7.8.

7.3.3.2 Computing the dimensions of the control zones along the linear segments

The computation of the maximum possible length of a control zone along the linear segment of the guide path is easier than that of a control zone at an intersecting point. The dimensions of the control zone at an intersection point has four variables as mentioned in the previous section. The AGVs can move along more than one direction at an intersection. However, the only variable related to linear section is its length. The AGVs move along only one direction in a control zone along a linear segment. If L_{max} is the maximum possible length of a control zone along a linear segment, then the expression for the proper operating conditions of the system is $L_{max}/V + m_{ik}.D_i \leq \left(\sum_{ij \in z} I_{ijze} \right)^{-1}$. Where, V is th speed of the AGV and m_{ik} is a variables which takes on a value of 1 if the delivery point of work station i in control zone k and 0 other wise. D_i is the delivery time at work station i. If L is the actual length of the control zone selected in the initial step and if $L \leq L_{ma}$, then the control zone selected initially will not need any modification. However, if $L > L_{ma}$, the control zone will have to be split into different segments so that the condition mentioned above is satisfied.

7.3.4 DESIGN OF CONTROL ZONE Z_1

This control zone is at an inter section point. The initial design is performed according to design rule 1 described in Section 6.2.1. According to this design rule, the control zone at this intersection points is defined as a square with the intersection point as the center. The initial design of this control zone is illustrated in Figure 7.9. The AGVs can move along three paths in this control zone. These three paths are $b \rightarrow o \rightarrow a$, $b \rightarrow o \rightarrow d$ and $c \rightarrow o \rightarrow a$. The distances traveled along these paths are x_1+x_2, x_2+x_4, and x_1+x_3 respectively. According to design rule 1, when the values of x_1, x_2, x_3, and x_4 are all set equal to each other (say x), the time an AGV spends in the control zone is $(2x)/V$.

For control zone z_1, the sum of the elements of column 1 of Table 7.8 is 90 hr $^{-1}$ or 1.5 min^{-1}. Therefore the expression for the maximum operating conditions of the system is $(2x)/V \leq 1/1.5$ and hence $x \leq 50$ feet. In other words, the maximum length or breadth of control zone 1 can be 100 feet. Therefore, according to design rule 1, the values of x_1, x_2, x_3 and x_4 are all equal to 50 feet.

When this control zone is designed according to design rule 1, the pick up points of work station 1 falls 10 feet outside this control zone. The pick up point of work station 3 falls within the control zone. Another issue of concern is that the intersection point to the right will be covered by this control zone. These distances can be observed by inspecting Figure 7.9. These are three typical situations where, design rules 3, 2 and 4 are applicable in that order.

According to the notations used in Section 6.2.1, the values of x_1 is set at 20 feet. This is based on design rule 4. The value of x_2 is set at 60 feet according to design rule 3 and the value of x_3 is set at 30 feet according to design rule 2. The new value of x_4 is computed using the updated values x_1, x_2 and x_3 and the expression for the maximum operating conditions of the system. This is done as follows: After adjusting the control zone to according to design rules 2,3 and 4, the distances and the frequency at which the distances are covered in the control zone are given below.

Distance	*Paths covered*	*Number of times per hour*
$60 + x_4$	*(1,2), (3,2), 7,2)*	*15*
50	*(2,3), (2,5), (2,6)*	*10*
80	*(1,3), (1,4), (1,6), (1,7), (4,8), (5,4), (5,7), (5,8), (6,3), (6,8), (7,2), (7,4), (7,8)*	*65*

When the condition for the system to operate at its maximum rate is applied, the value of x_4 is computed using $\{(60+x_4)/150\}x15/60+\{50/150\}x10/60+\{80/150\}65/60 \leq 1$ and hence $x_4 \leq 160$. The design of this control zone can include the drop off point of work station if necessary. In that case, the expression for the maximum operating conditions is $\{(60+x_4)/150\}x15/60 + \{50/150\}x10/60+\{80/150\}65/60+ 0.25x15/60 \leq 1$ and hence $x_4 \leq 122$. These two results give the upper limit on the value of x_4.

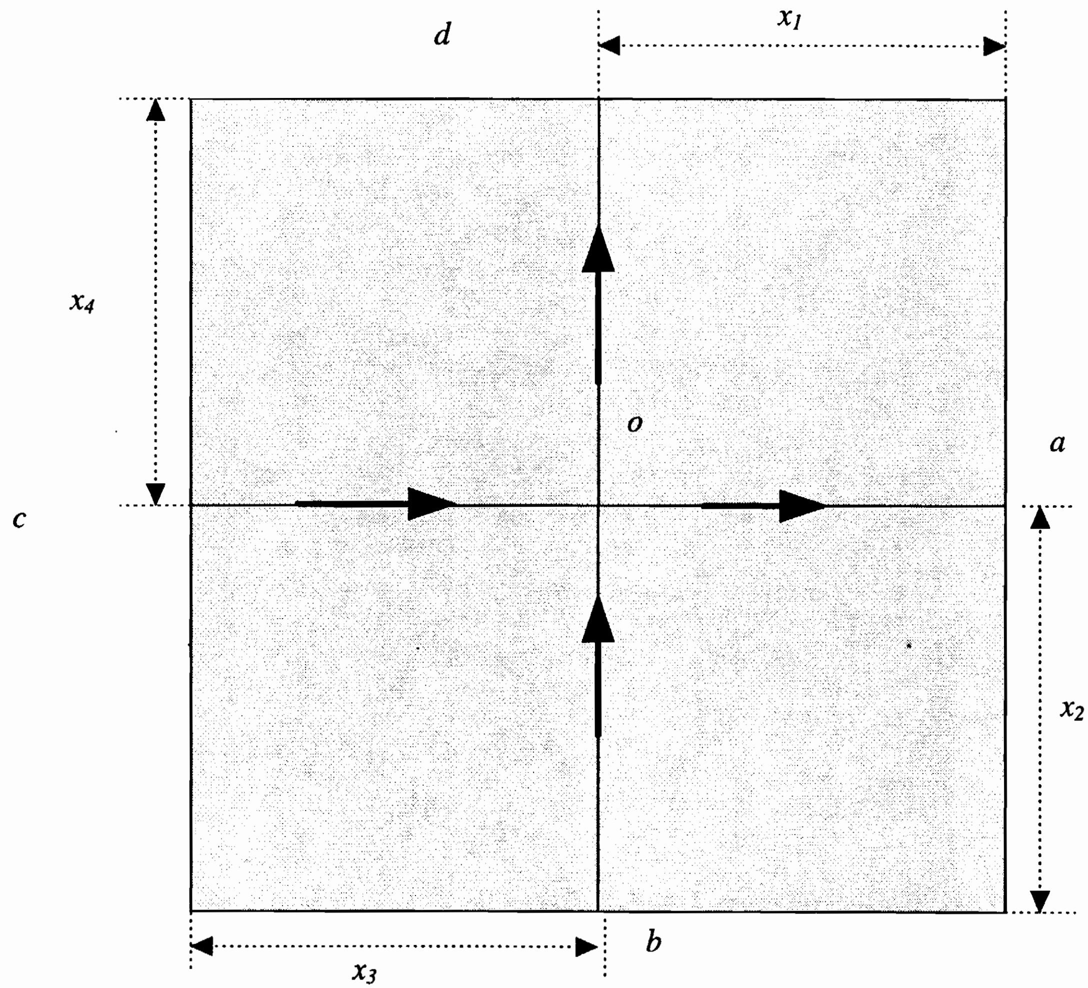


Figure 7.9 The initial design for control zone z_1

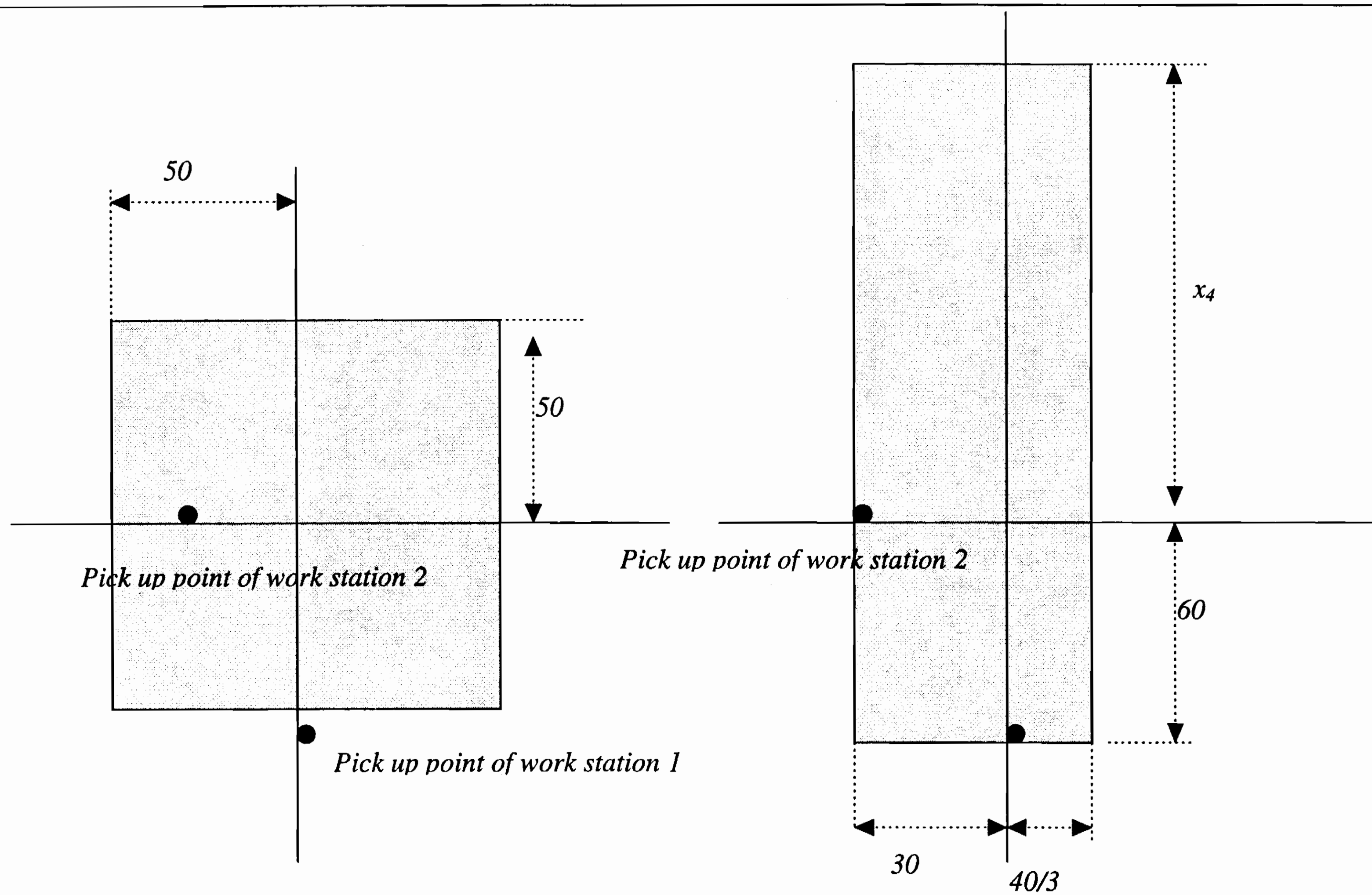


Figure 7.10 Illustration of the design of control zone 1

7.3.4.1 Design of control zone z_{10}

Control zone z_{10} is the segment of the guide path which surrounds work station 2. This control zone is illustrated in Figure 7.11. The total distance around work station 2 is 360 feet. The rate at which the AGVs enter this control zone is obtained from Table 7.8. This value is 35 per hour. This is the summation of the elements corresponding to this control zone in Table 7.8. Hence, the maximum distance L_{max} that can be covered by this control zone is given by $L_{max}/V \leq (35/60)^{-1}$. Where the V is the speed of the AGV. Therefore, $L_{ma} \leq (60/35) x 150$ = 257 feet. The total distance around work station 2 is 103 feet more than the maximum allowable distance. However, as it can be seen in Figure 7.11, the distances 30 feet and x_4 feet are covered by control zone z_1. The distance x_4 can be up to a maximum of 122 feet. Therefore, the designer has the option fixing the value of x_4 so that control zone z_1 can be used along the guide path surrounding work station 2.

7.3.5 REMARKS

The other control zones at the intersections and the ones along the linear portions of the guide path are designed in a similar fashion mentioned in the previous sections. The designer will have various options based on the dimensions of the control zones obtained using the above analysis. For example more analysis can be performed to explore the possibility of merging the control zones selected in the initial step. Prior research has not been found to analytically determine the size of the control zones so as to minimize the number of zones being employed. This example illustrates how the Petri net model can be used in establishing control zone dimensions that will permit the network to perform as desired under the specified demand load.

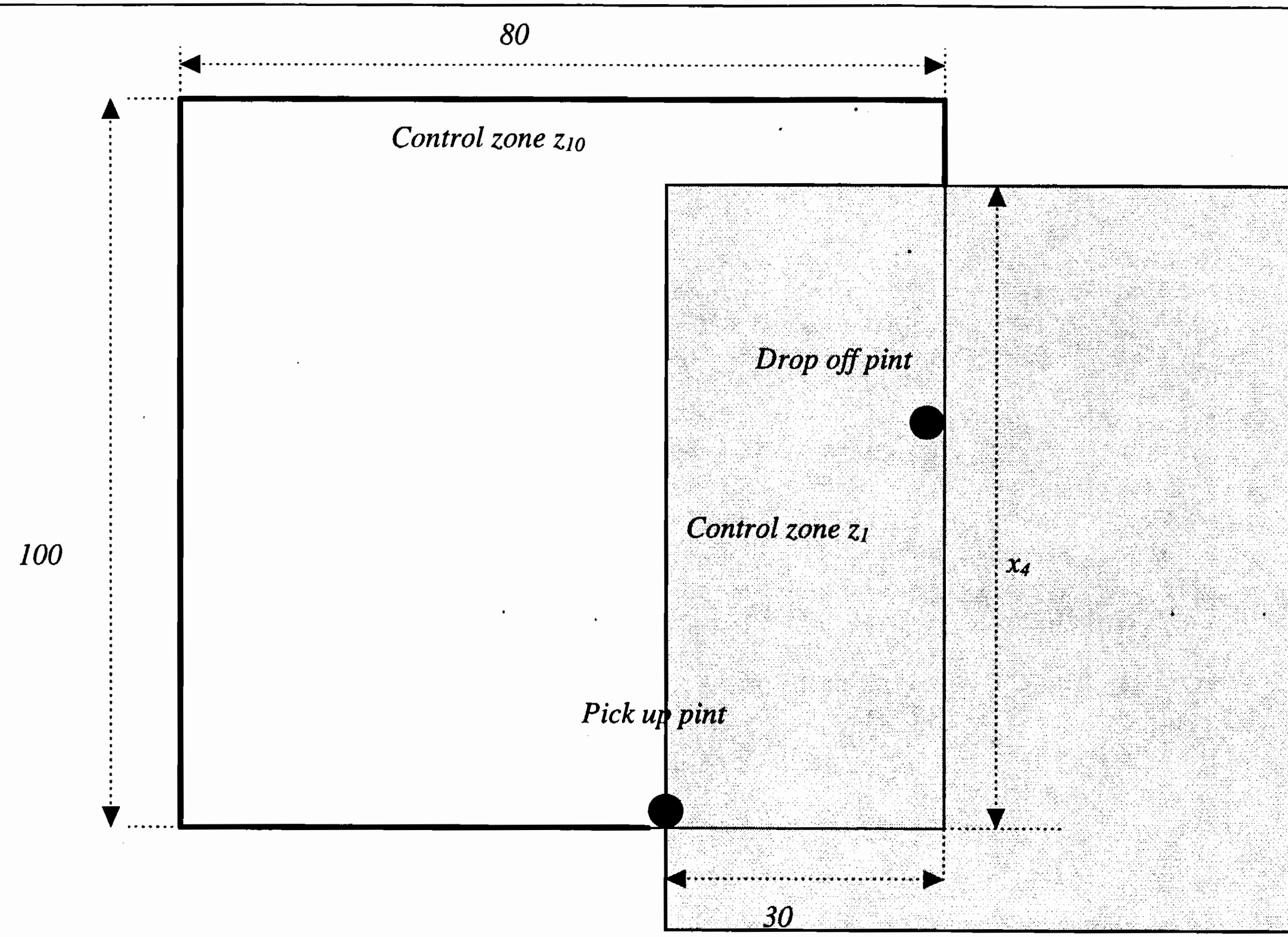


Figure 7.11 Illustration of control zone z_1

8. CONCLUSIONS AND REMARKS

Analytical procedures for AGV system have been restricted to the computation of the fleet size and the design of guide path layout over past several years. The design and location of control zones been avoided as it was believed that the inclusion of control zones complicates the analytical procedure. This research is one of the first attempts focused on capturing the effects of control zone on an AGV system.

The power of Petri net theory has been effectively used to capture all the interactions between the various components of an AGV system. The strong mathematical back ground of Petri net is used for the computation of the fleet size and the control zone dimensions. The existing models for AGV system appear to be complicate and difficult to be understood. The procedure of modeling the AGV system is based on the well known top down, bottom up approach and the concepts parallel mutual exclusion. The final Petri net is built using sub nets representing the components of an AGV system. The sub nets are easily understood and the merged final net is not necessarily needed for the analytical procedure. Unlike the simulation procedures employed by the existing Petri net models, this thesis employs the strong mathematical background of Petri net theory.

Even though the AGV system is assumed to operate under absolute deterministic conditions, the results obtained using the deterministic timed Petri net models can be a good input for detail simulation analysis. Further, this modeling procedure can be extended to the creation of stochastic Petri nets and analyzed using powerful Petri net packages such as GSPN.

This thesis is a contribution in employing Petri net theory in manufacturing systems. Many areas of possible application of Petri net models remain unexplored. AS/RS systems is a very promising area. Real-time control of AGV systems and AS/RS systems is another area future researchers can look for. Scheduling and sequencing of job shop and flow shop systems have been barely explored and remain un-attended. Extensive research in these areas would make Petri net theory an even more powerful tool benefiting the analysis of manufacturing systems.

REFERENCES

[1]. Agerwala, T., “Putting Petri Nets to Work,” Computer, Vol. 12, pp. 85-94.

[2]. Agerwala, T. and Choed Amphai, Y., “A Synthesis Rule for Concurrent Systems,” Proceedings of the 15th Design Automation Conference, pp. 305-311.

[3]. Ajmone, M. and Choila, G., “On Petri Nets with Deterministic and Exponentially Distributed Firing Times,” Advances in Petri Nets, New York : Springer-Verlag, Vol. 266, pp. 132-145.

[4]. Al-Jaar, R. Y. and Derochers, A. A., “Performance Evaluation of Automated Manufacturing Systems Using Generalized Stochastic Petri Nets,” IEEE Transactions on Robotics and Automation, Vol. 6, pp. 621-639.

[5]. Archetti, F. and Scimachen, A., “Development, Analysis and Simulation of Petri Net Models: An Application to AGV Systems,” Operation Research Models in Flexible Manufacturing Systems, Springer Verlag, Vol. 306, pp. 91-114.

[6]. Balbo, G., Choila, G., Franceschinis, G., and Roet, G. M., “Generalized Stochastic Petri Nets for the Performance Evaluation Performance of FMS,” Proceedings of the IEEE Conference on Robotics and Automation, Raleigh, NC, pp. 1013-1018.

[7]. Banaszak, Z. A. and Krogh, B. H., “Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows,” IEEE Transactions on Robotics and Automation, Vol. 6, pp. 724-734.

[8]. Bozer, Y. A. and Srinivasan, M. M., “Tandem Configuration for Automated Guided Vehicle Systems and Analysis of Single Vehicle Loops,” IIE Transactions, Vol. 23.

[9]. Bruno, G and Biglia, P., "Performance Evaluation and Validation of Tool Handling in Flexible Manufacturing Systems Using Petri Nets," Proceeding of IEEE International Workshop on Timed Petri Nets, Torino, Italy, pp. 64-71.

[10]. Cui, Y. and Wang, J., "The Formal Design Method of Petri Net Controllers for AGV Systems With Shared Paths," Journal of Xidian University, China. pp. 118-124.

[11]. Derochers, A. A. and Al-Jaar, R. Y., Application of Petri Nets in Manufacturing Systems, IEEE Press, New York.

[12]. Di Cesare, F. Harhalakis G, Proth J. M, Silva M. and Vernadat F. B., Practice of Petri Nets: in Manufacturing, Chapman and Hall, New York.

[13]. Dong, S. Y. and Linn, R. J., "Push and Pull Rules for Dispatching Automated Guided Vehicles in a Flexible Manufacturing Systems," International Journal of Production Research, Vol. 31, pp. 43-57.

[14]. Dubois, D. and Stecke, K., "Using Petri Nets to Represent Production Processes," Proceedings of the 22nd IEEE Conference on Decision and Control, San Antonio, TX, pp. 1062-1067.

[15]. Dugan, J. B., Trivedi, K. S., Geist, R. M., and Nicola, V. F., "Extended Stochastic Petri Nets: Application and Analysis," Performance 84, Paris, France, pp. 507-519.

[16]. Egbelu, P. J., "Pull Versus Push Strategy for Automated Guided Vehicles Load Movement in a Batch Manufacturing System," Journal of Manufacturing Systems, Vol. 6, 271-280.

[17]. Egbelu, P. J., "The Use of Non-simulation Approaches in Estimating Vehicle Requirements in an Automated Guided Vehicle Based Transport System," Material Flow, Vol. 4, pp. 17-32.

[18]. Egbelu, P. J. and Tanchoco, J. M. A., "Potentials for Bi-directional Guide Path for Automatic Guided Vehicle System," International Journal of Production Research Vol. 24 , pp. 1075-1099.

[19]. Egbelu, P. J. and Tanchoco, J. M. A., "Characterization of Automated Guided Dispatching Rules," International Journal of Production Research, Vol. 22, pp. 359-374.

[20]. Ezpeleta, J. and Colom, J. M, and Martinez, J., "A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems," IEEE Transactions on Robotics and Automation, Vol. 11, pp. 173-184.

[21]. Gaskins, R. J. and Tanchoco, J. M. A., "Flow Path Design for Automated Guided Vehicle Systems," International Journal of Production Research, Vol. 25, pp. 667-676.

[22]. Gaskins, R. J., Tanchoco, J. M. A., and Tanghaboni, F., "Virtual Flow Paths for Free Ranging Automated Guided Vehicle Systems," International Journal of Production Research, Vol. 27, pp. 91-100.

[23]. Goetz, W. J., and Egbelu, P. J., "Guided Path Design and Location of Load Pick-up/drop-off Points for an Automated Guided Vehicle System," International Journal of Production Research, Vol. 28, pp. 927-941.

[24]. Gould, L., "AGVs in America," Modern Material Handling, Sept.1990, pp. 56-60.

[25]. Guo, D., Di Cesare, F., and Zhou, M. C., "Moment Generating Function Approach to Performance Analysis of Extended Stochastic Petri Nets," Proceedings of the 1991 IEEE International Conference On Robotics and Automation, Sacramento, CA, pp. 1309-1314..

[26]. Hillion, H. P. and Proth, J. M., "A Heuristic Algorithm for the Periodic Scheduling and Sequencing Job-Shop Problem," Proceedings of the 26th IEEE Conference on Decision and Control, Los Angeles, CA, pp. 612-617.

[27]. Jensen, K., "Colored Petri Nets," Advances in Petri Nets, New York : Springer-Verlag, Vol. 254, pp. 248-299.

[28]. Karp, R. M. and Miller, R. E., "Parallel Program Schemata," Journal of Computers and Systems Science, Vol. 3, pp. 147-195.

[29]. Kasturia, E., DiCesare, F., and Desrochers, A. A., "Real Time Control of Multi-Level Manufacturing Systems Using Colored Petri Nets," Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, pp. 1114-1119.

[30]. Krogh, B. H. and Beck, L., "Synthesis of Place/Transition Nets for Simulation and Control of Manufacturing Systems," Proceedings of the 4th IFAC/IFORS Symposium on Large Scale Systems, Aug. 1986.

[31]. Lautenbach, K., "Linear Algebraic Techniques for Place/Transition Nets," Advances in Petri Nets, New York : Springer-Verlag, Vol. 254, pp. 142-167.

[32]. Lee, D. Y. and DiCesare, F., "Integrated Scheduling of Flexible Manufacturing Systems Employing Automated Guided vehicles," IEEE Transactions on Industrial Electronics, pp. 602-610.

[33]. Leung, L.C., Khator, S. K., and Kimbler, D. L., "Assignment of AGVs with Different Vehicle Types," Material Flow, Vol. 4, pp. 65-72.

[34]. Malmborg, C. J., "A Model for the Design of Zone Control Automated Guided Vehicle Systems," International Journal of Production Research , Vol. 28, pp. 1741-1758.

[35]. Marson, M. A., Conte, G., and Balbo, G., "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems," ACM Transactions on Computer Systems, Vol. 2, pp. 93-122.

[36]. Maxwell, W. L. and Muckstadt, J. A., Design of Automated Guided Vehicle Systems," IIE Transactions, Vol. 14, pp. 114-124.

[37]. Molly, M. K., "Performance Analysis Using Stochastic Petri Nets," IEEE Transactions on Computers, Vol. 31, pp. 913-917.

[38]. Murata, T., "Petri Nets : Properties, Analysis and Application," Proceedings of the IEEE, Vol. 77, pp. 541-579.

[39]. Narahari, Y. and Viswanadham, N., "A Petri Net Approach to the Modeling and Analysis of Flexible Manufacturing Systems," Annals of Operations Research, Vol. 3, pp. 449-472.

[40]. Narahari, Y. and Viswanadham, N., "On the Invariants of Colored Petri Nets," Advances in Petri Nets, New York : Springer-Verlag, Vol. 222, pp. 330-345.

[41]. Narahari, Y., Viswanadham, N, and Johnson, T. L., "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models," IEEE Transactions on Robotics and Automation, Vol. 6, pp. 713-723.

[42]. Peterson, J. L., Petri Net Theory and Modeling of Systems , Prentice, Inc., Englewood Cliffs, NJ.

[43]. Petri, C. A., "Kommunication mit Automaten," Ph.D. dissertation, University of Bonn, West Germany, 1962.

[44]. Ramachandani, C., "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1973.

[45]. Ramamoorthy, C. V. and Ho, G. S., "Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets," IEEE Transactions on Software Engineering, Vol. 6, pp. 440-449.

[46]. Reisig, W. L., Petri Nets: An Introduction, New York., Springer-Verlag.

[47]. Sifakis, J., "Performance Evaluation of Systems Using Nets," Net Theory and Applications, Springer-Verlag, Vol. 84, pp. 307-319.

[48]. Sifakis, J., "Structural Properties of Petri Nets," Mathematical Foundations of Computer Science, Vol. 64, New York : Springer-Verlag, pp. 474-483.

[49]. Sinriech, D. and Tanchoco, J. M. A., "An Economic Model for Determining AGV Fleet Size," International Journal of Production Research , Vol. 30, pp. 1255-1268.

[50]. Sun, T. H., Cheng, C. W., and Fu, L. C., "A Petri Net Based Approach to Modeling and Scheduling for an FMS and a Case Study," IEEE Transactions on Industrial Electronics, pp. 593-601.

[51]. Suzuki, I. and Murata, T., "A Method for Stepwise Refinements and Abstractions of Petri Nets," Journal of Computer and System Sciences, Vol. 27, pp. 51-76.

[52]. Tanchoco, J. M. A., Egbelu, P. J. and Taghaboni, F., "Determination of the Total Number of Vehicles in an AGV Based Material Handling System," Material Flow, Vol. 4, pp. 33-51.

[53]. Valette, R., "Analysis of Petri Nets by Stepwise Refinements," Journal of Computer and System Sciences, Vol. 18, pp. 35-46.

[54]. Venkatesh, K., Zhou, M. C., and Caudill, R. J., "Comparing Ladder Logic Diagrams and Petri Nets for Sequence Controller Design Through a Discrete Manufacturing Systems," IEEE Transactions on Industrial Electronics, pp. 611-619.

[55]. Wang, H. P. B. and Hafeez, S. A., "Performance Evaluation of Tandem and Conventional AGV Systems Using Generalized Petri Nets," International Journal of Production Research, Vol. 32, pp. 917-932.

[56]. Watson, J. F. and Derochers, A. A., "Applying Generalized Stochastic Petri Nets to Manufacturing Systems Containing Nonexponential Transition Functions," IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, pp. 1008-1017.

[57]. Zeng, L., Wang, H. P., and Jin, S., "Conflict Detection of Automated Guided Vehicles: A Petri Net Approach," International Journal of Production Research, Vol. 29, pp. 865-879.

[58]. Zhou, M. C. , DiCesare, F., and Desrochers, A. A., "A Top-Down Approach to Systematic Synthesis of Petri Net Models for Manufacturing Systems," Proceedings of the IEEE Conference on Robotics and Automation, Scottsdae, AZ, pp. 534-539.

[59]. Zhou, M. C. and DiCesare, F., "Parallel and Sequential Mutual Exclusions for Petri Net Modeling for Manufacturing Systems," IEEE Transactions on Robotics and Automation, Vol. 7, pp. 515-527.

[60]. Zhou, M. C., DiCesare, F., and Desrochers, A. A., "A Hybrid Methodology for Synthesis of Petri Net Models for Manufacturing Systems," IEEE Transactions on Robotics and Automation, Vol. 8, pp. 350-361.

[61]. Zurawski, R. Z and Dillon, T. S., "Specification, Verification and Performance Evaluation of Flexible Manufacturing Systems," Computer Applications in Production and Engineering, pp. 517-525.

[62]. Zurawski, R. Z and Zhou, M. C., "Petri Nets and Industrial Applications," IEEE Transactions on Industrial Electronics, pp. 567-583.

[63]. Zurawski, R. Z., "Systematic Construction of Functional Abstractions of Petri Net Models of Flexible Manufacturing Systems," IEEE Transactions on Industrial Electronics, pp. 584-592.

APPENDIX

This appendix illustrates the movement of the AGVs between work stations with respect to the control zones established in Section 7.3.

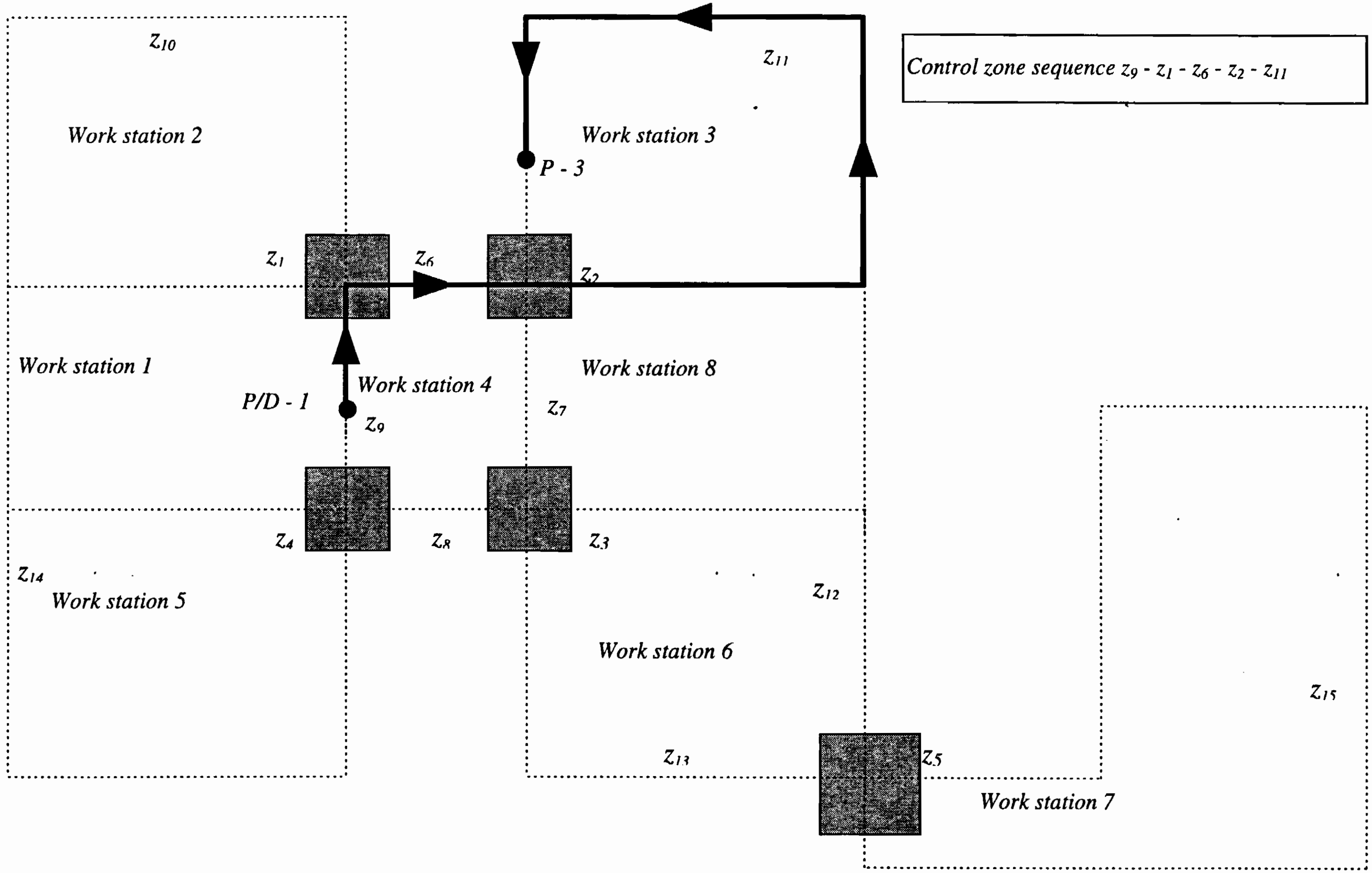


Figure A.1 Illustration of the movement of an AGV between work stations 1 and 3.

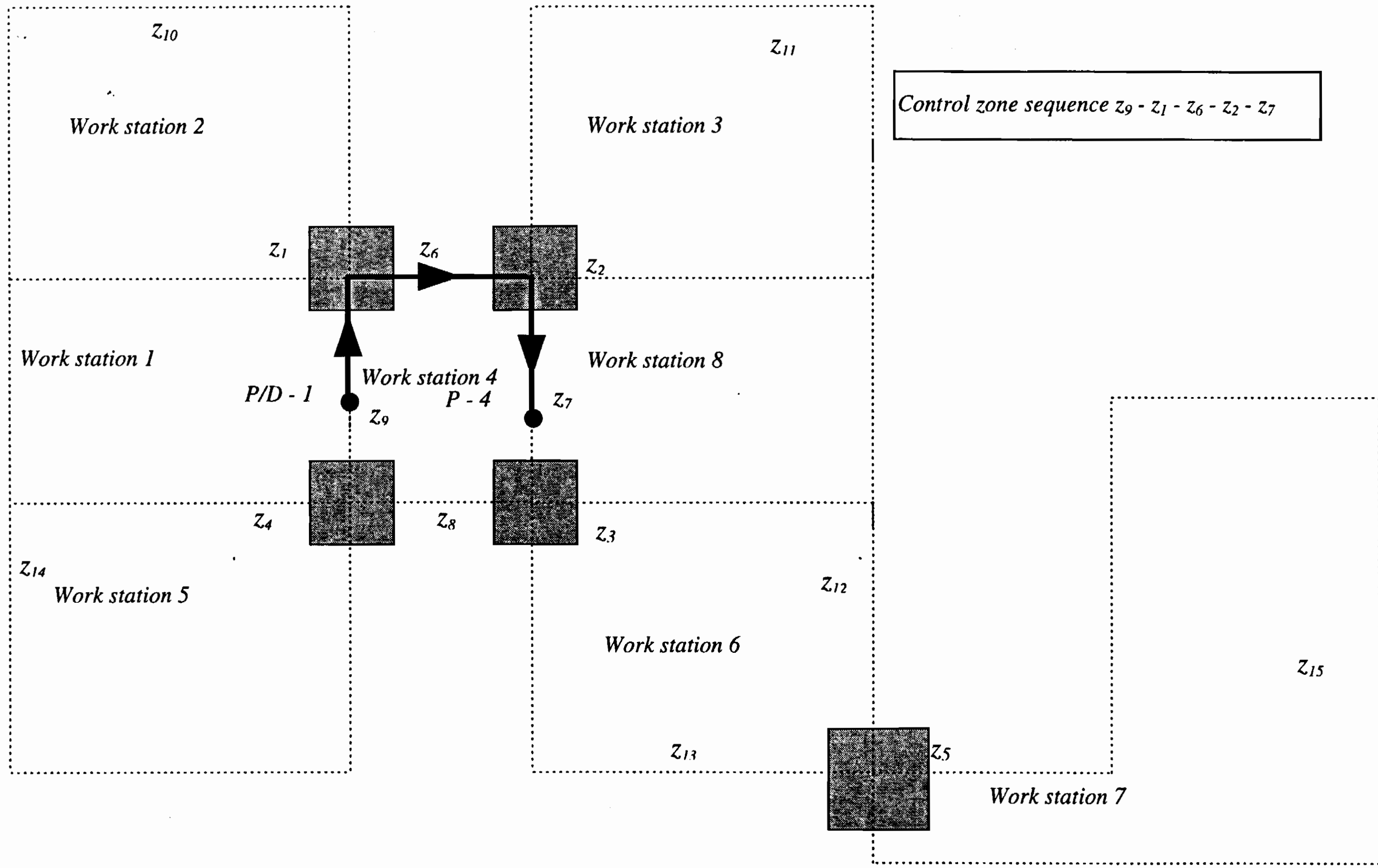


Figure A.2 Illustration of the movement of an AGV between work stations 1 and 4

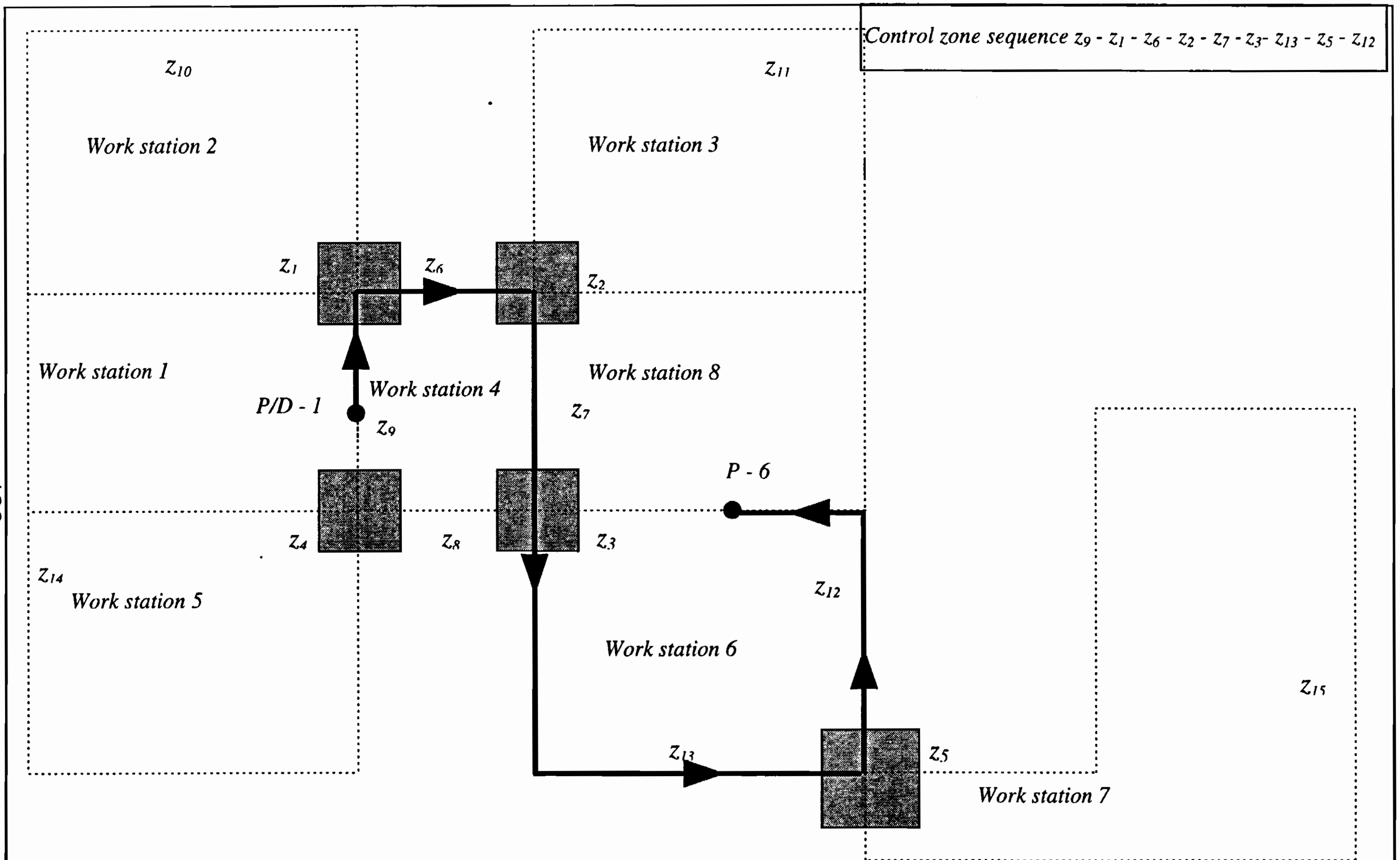


Figure A.3 Illustration of the movement of an AGV between work stations 1 and 6

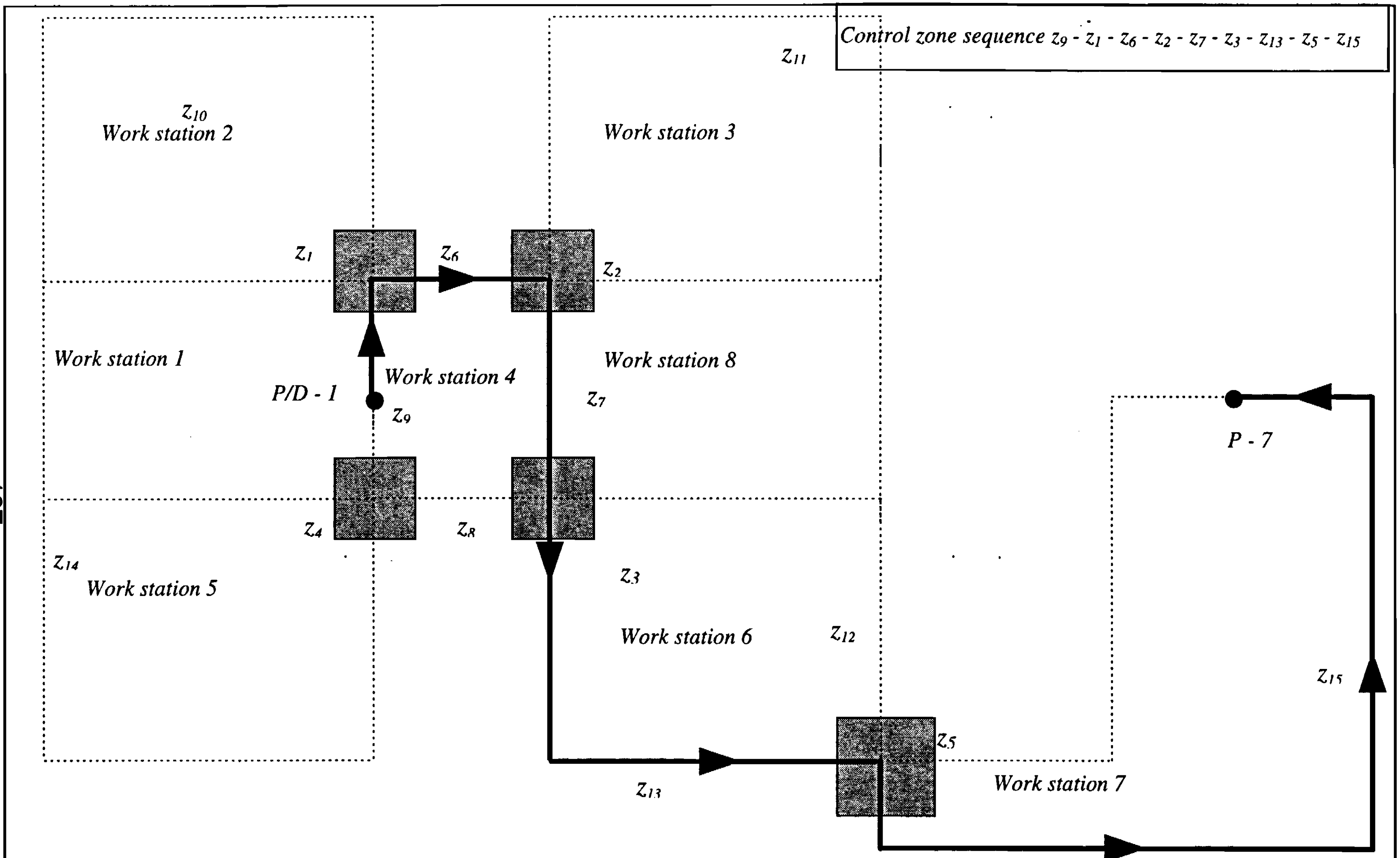


Figure A.4 Illustration of the movement of an AGV between work stations 1 and 7

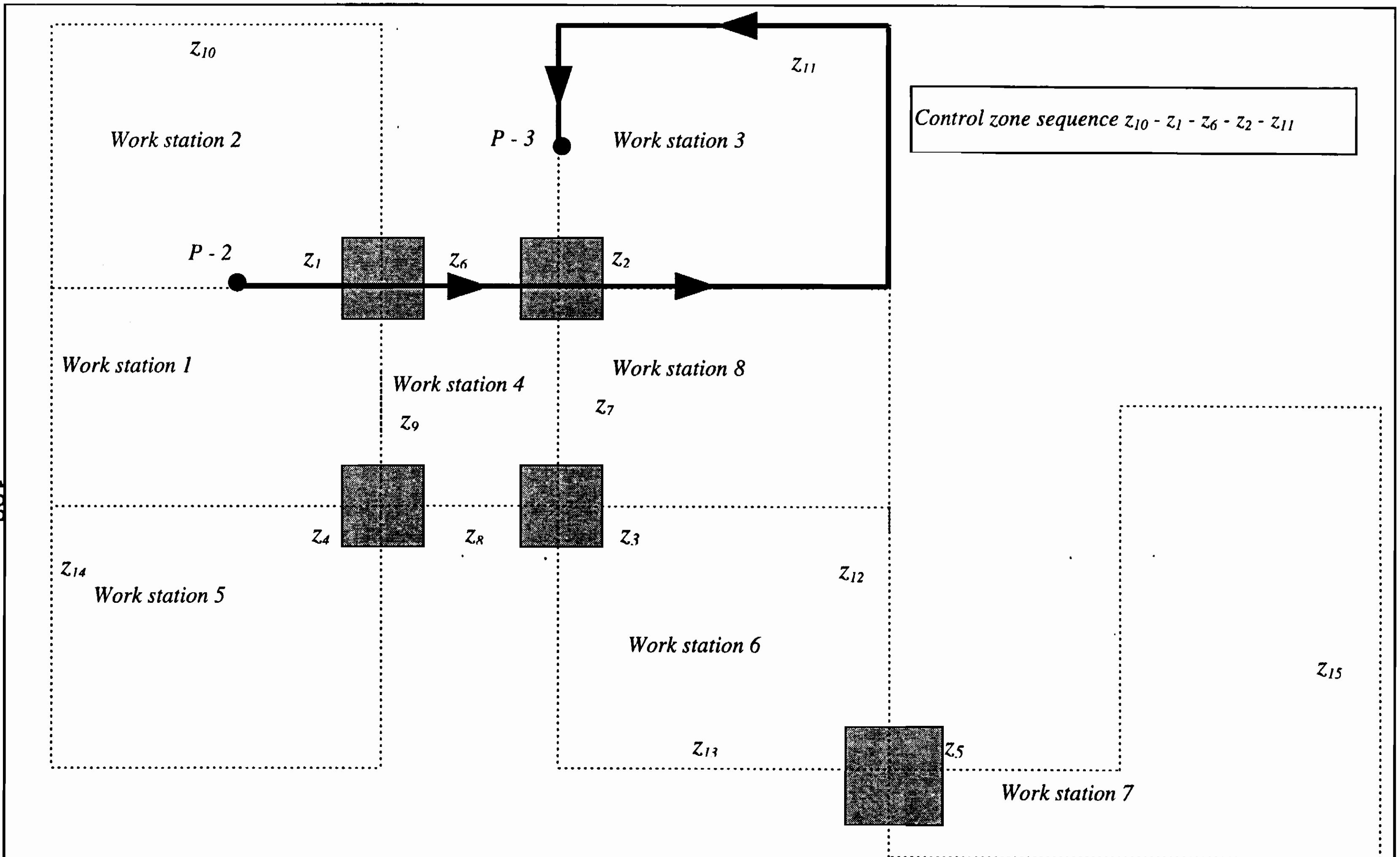


Figure A.5 Illustration of the movement of an AGV between work stations 2 and 3

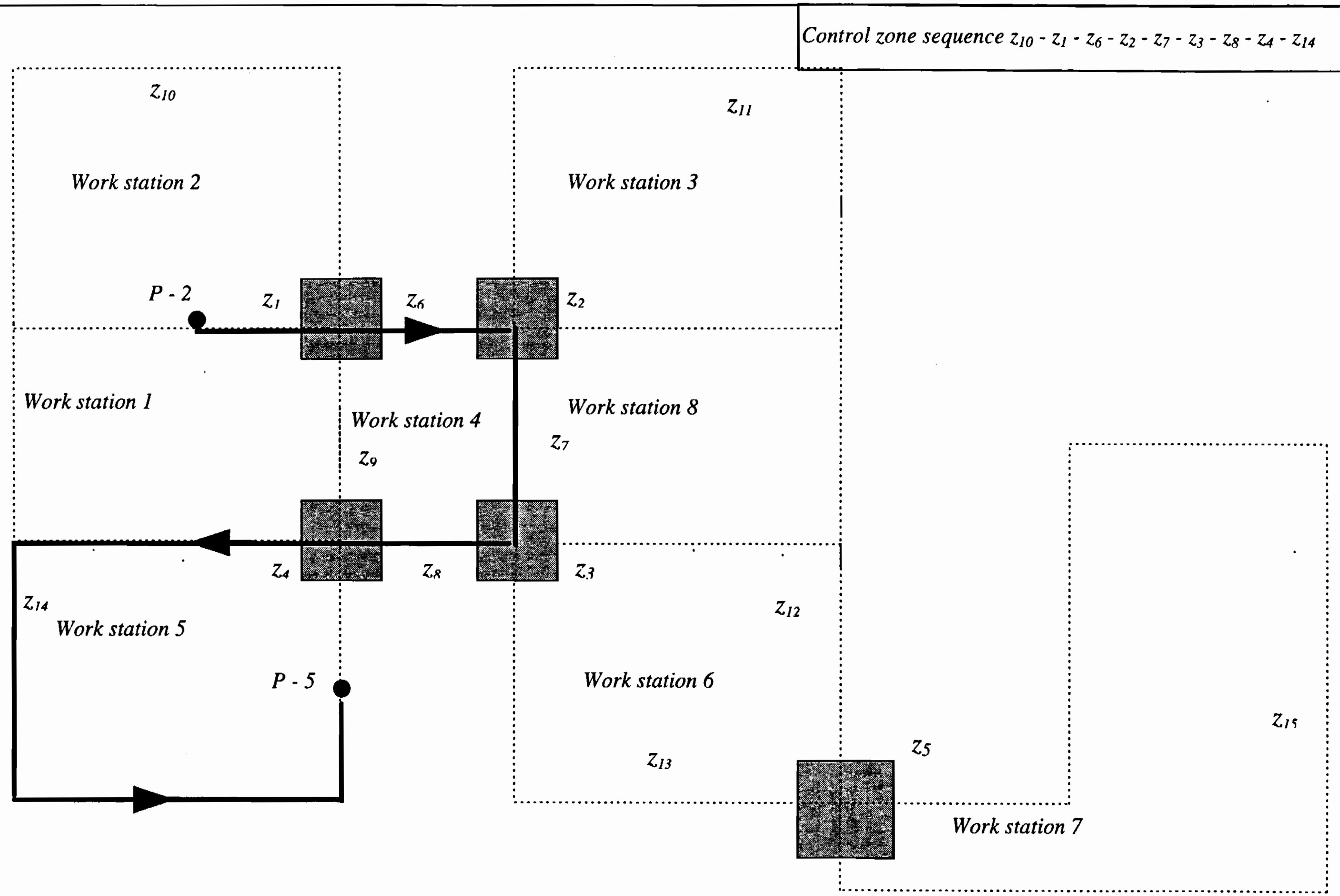


Figure A.6 Illustration of the movement of an AGV between work stations 2 and 5

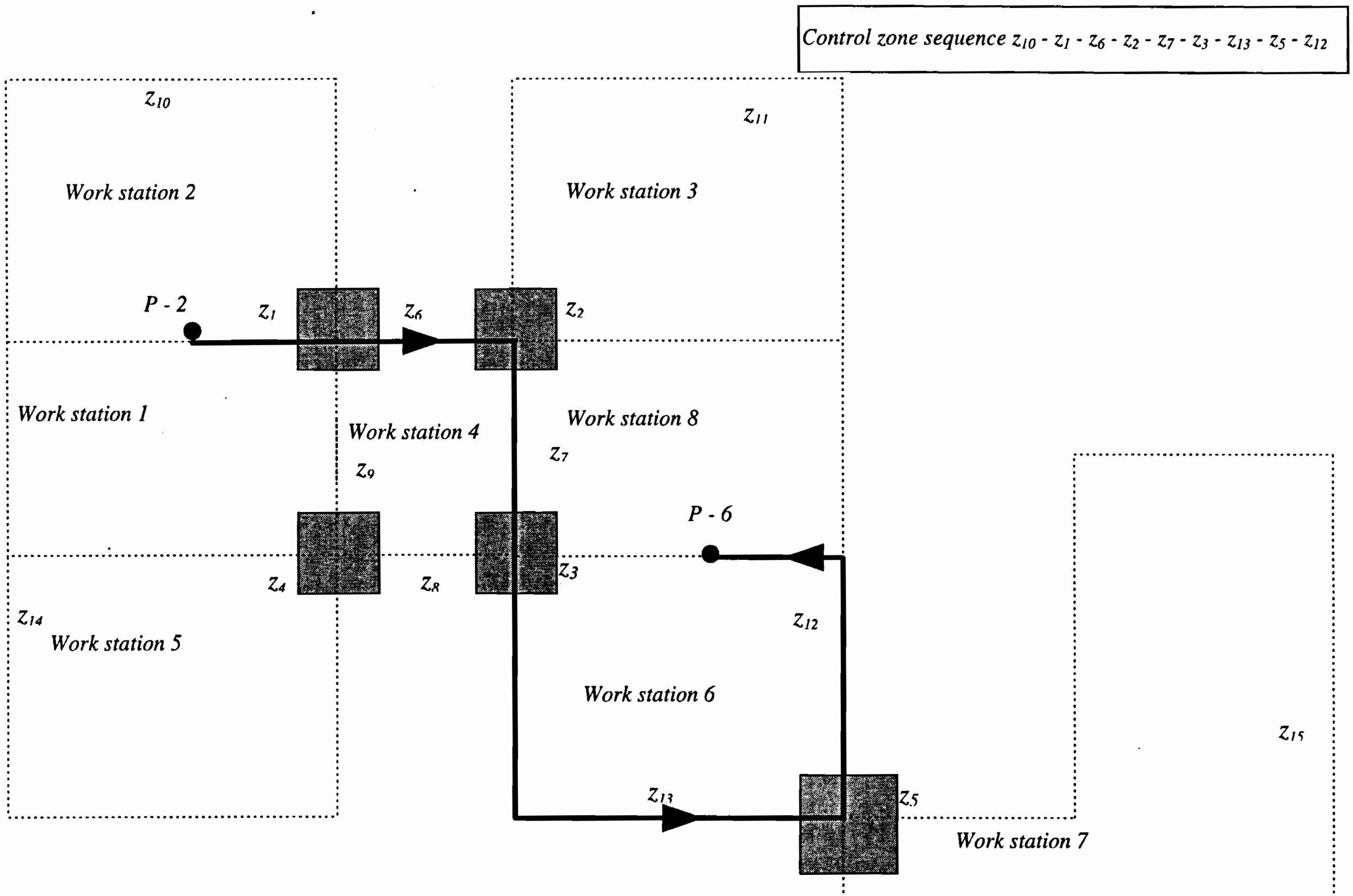


Figure A.7 Illustration of the movement of an AGV between work stations 2 and 6

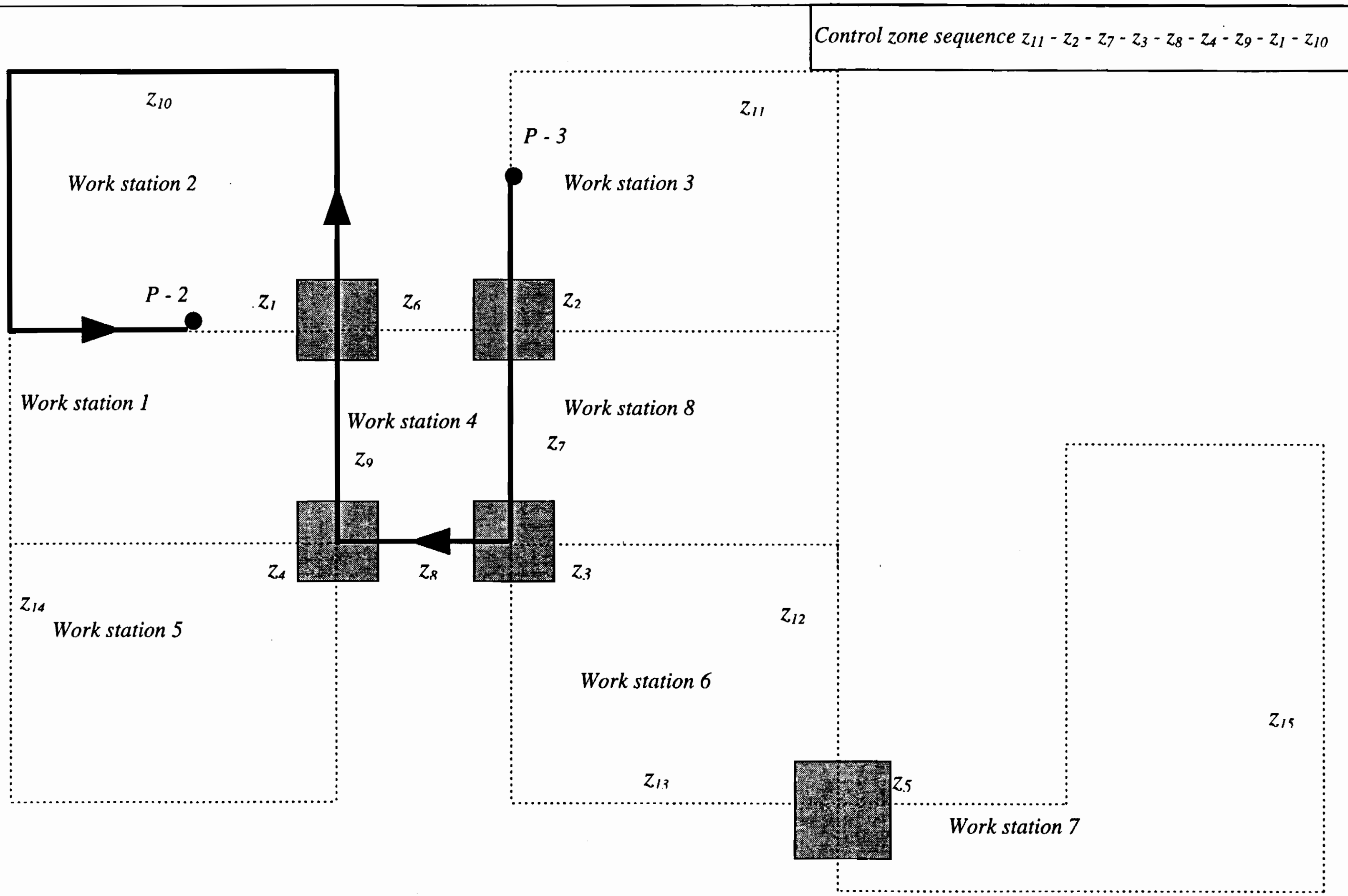


Figure A.8 Illustration of the movement of an AGV between work stations 3 and 2

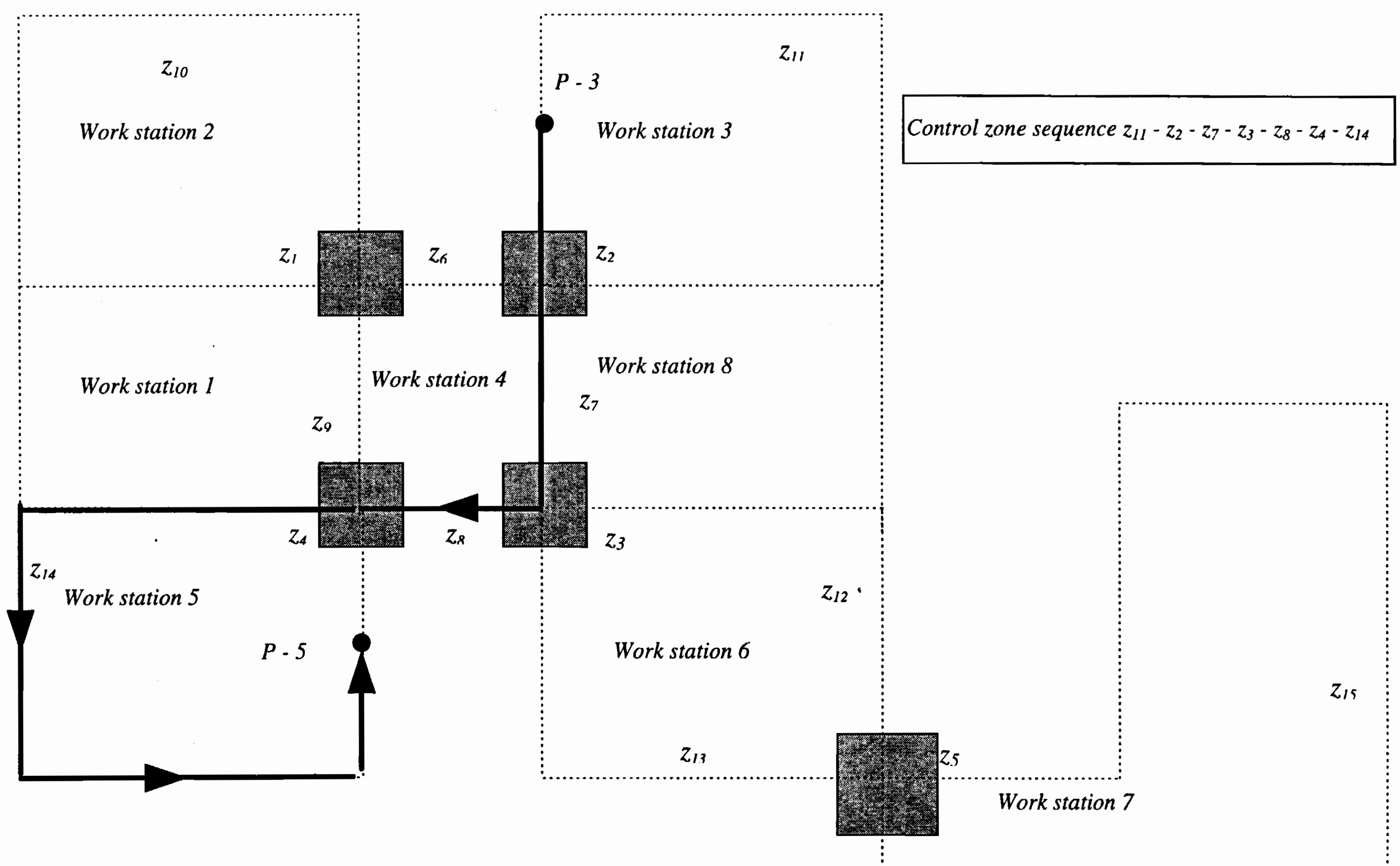


Figure A.9 Illustration of the movement of an AGV between work stations 3 and 5

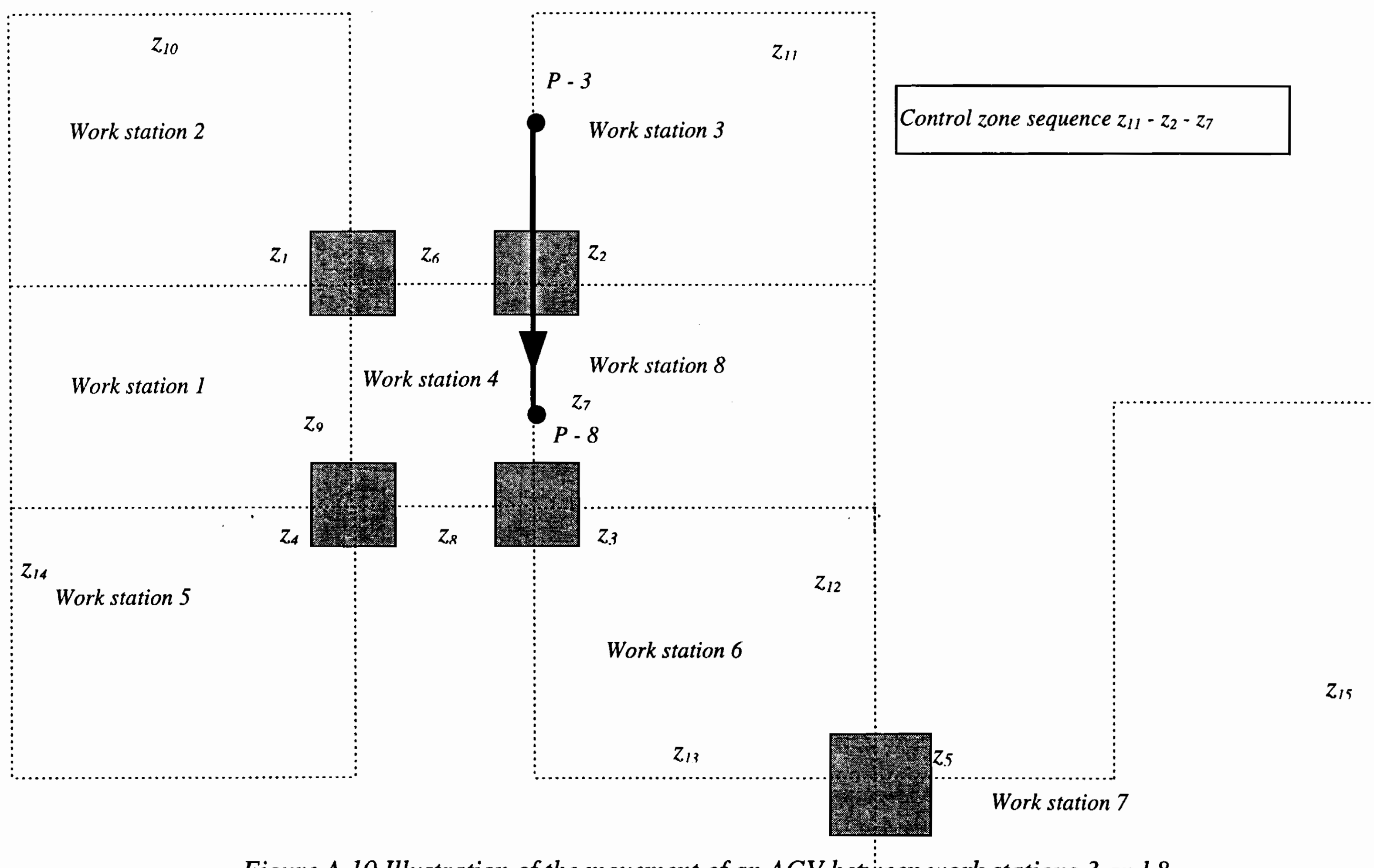


Figure A.10 Illustration of the movement of an AGV between work stations 3 and 8

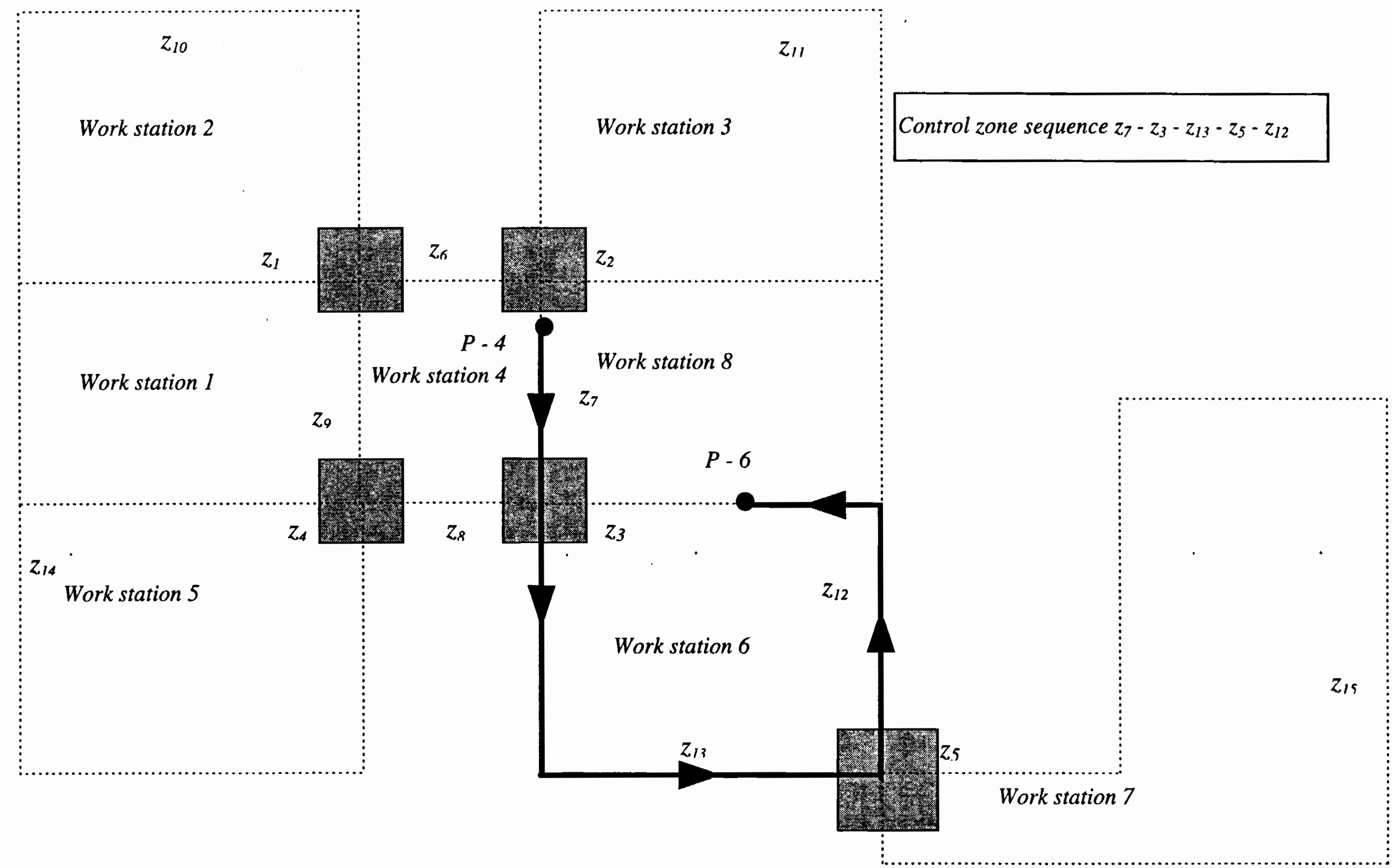


Figure A.11 Illustration of the movement of an AGV between work stations 4 and 6

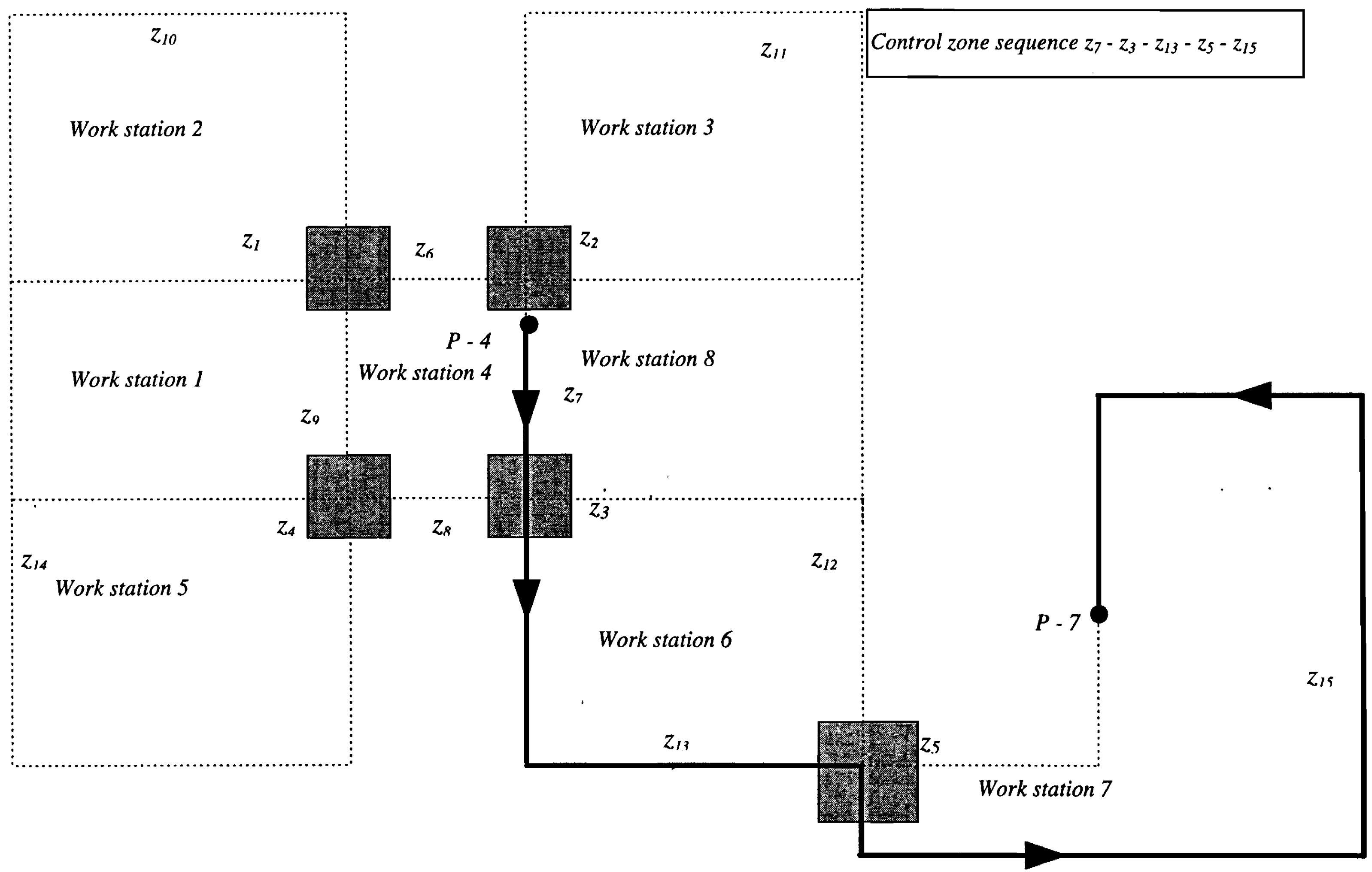


Figure A.12 Illustration of the movement of an AGV between work stations 4 and 7

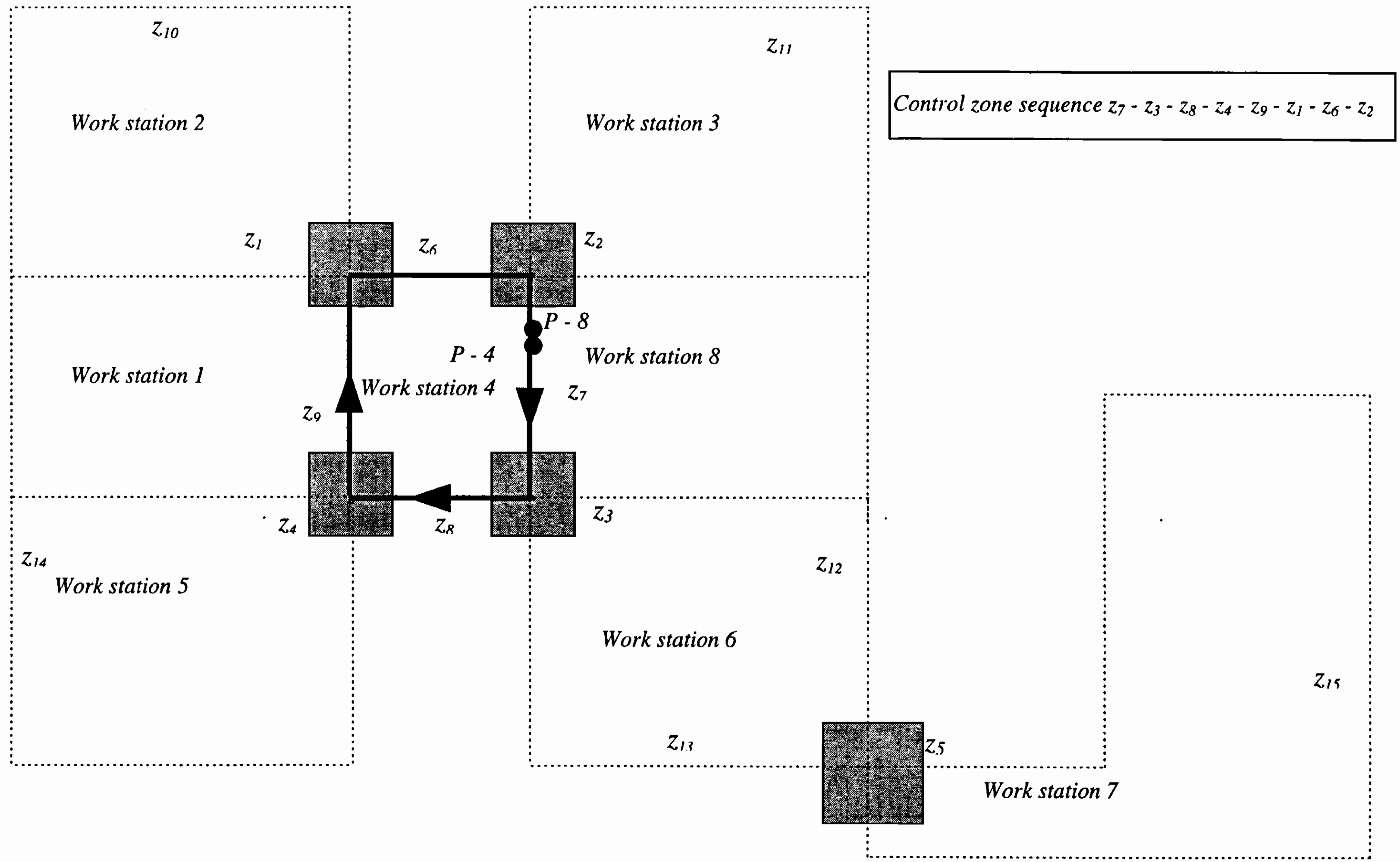


Figure A.13 Illustration of the movement of an AGV between work stations 4 and 8

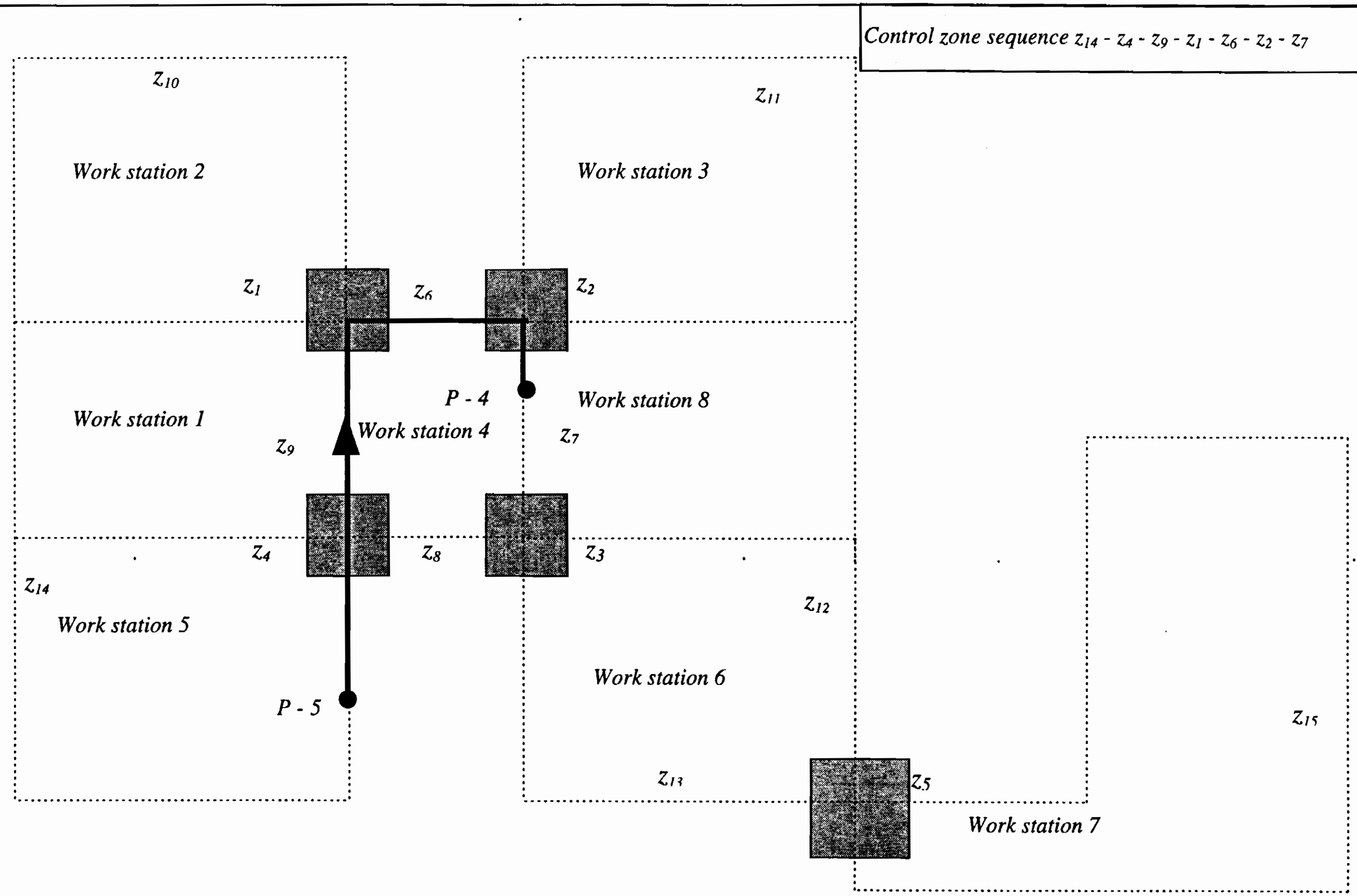


Figure A.14 Illustration of the movement of an AGV between work stations 5 and 4

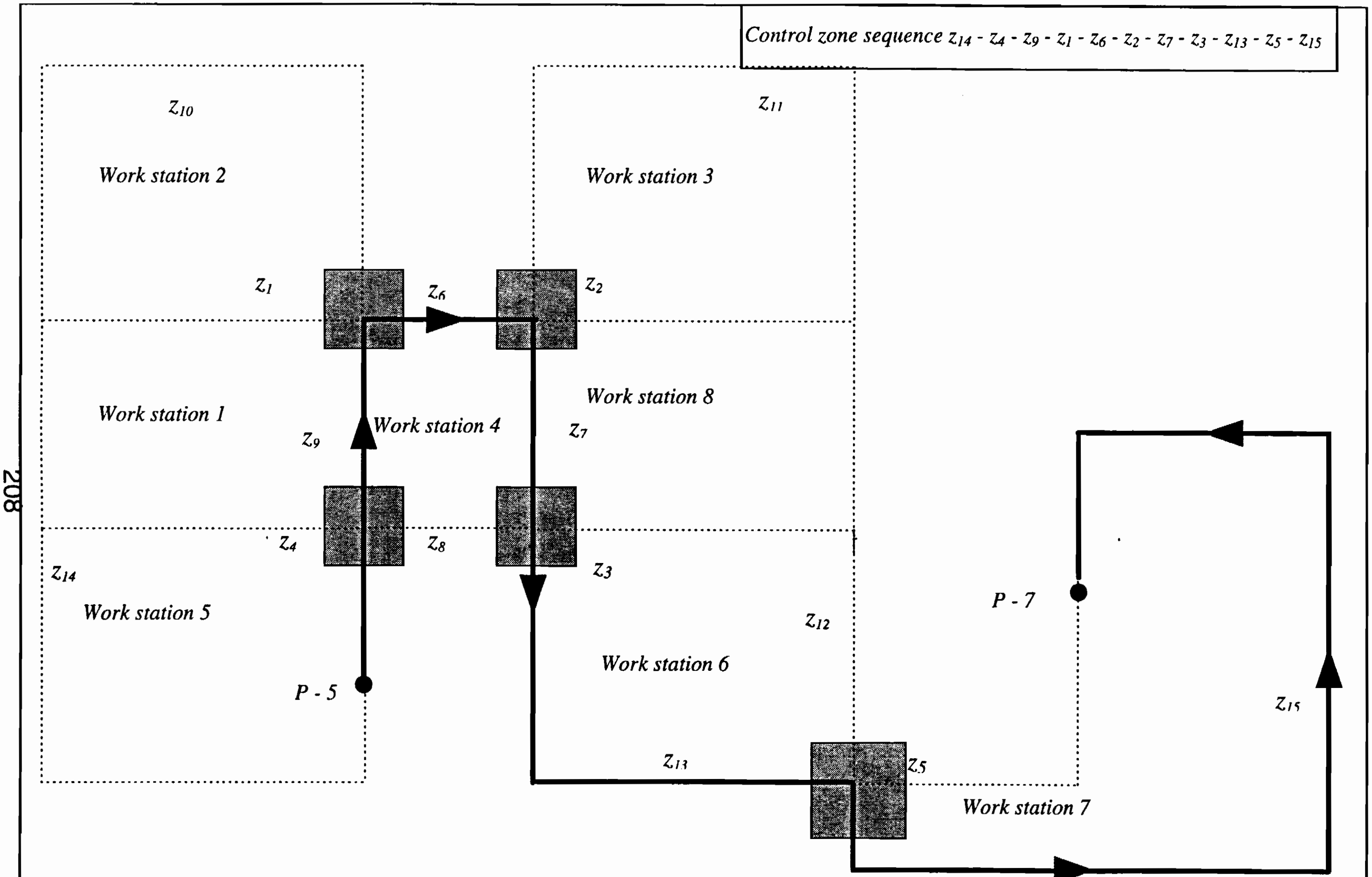


Figure A.15 Illustration of the movement of an AGV between work stations 5 and 7

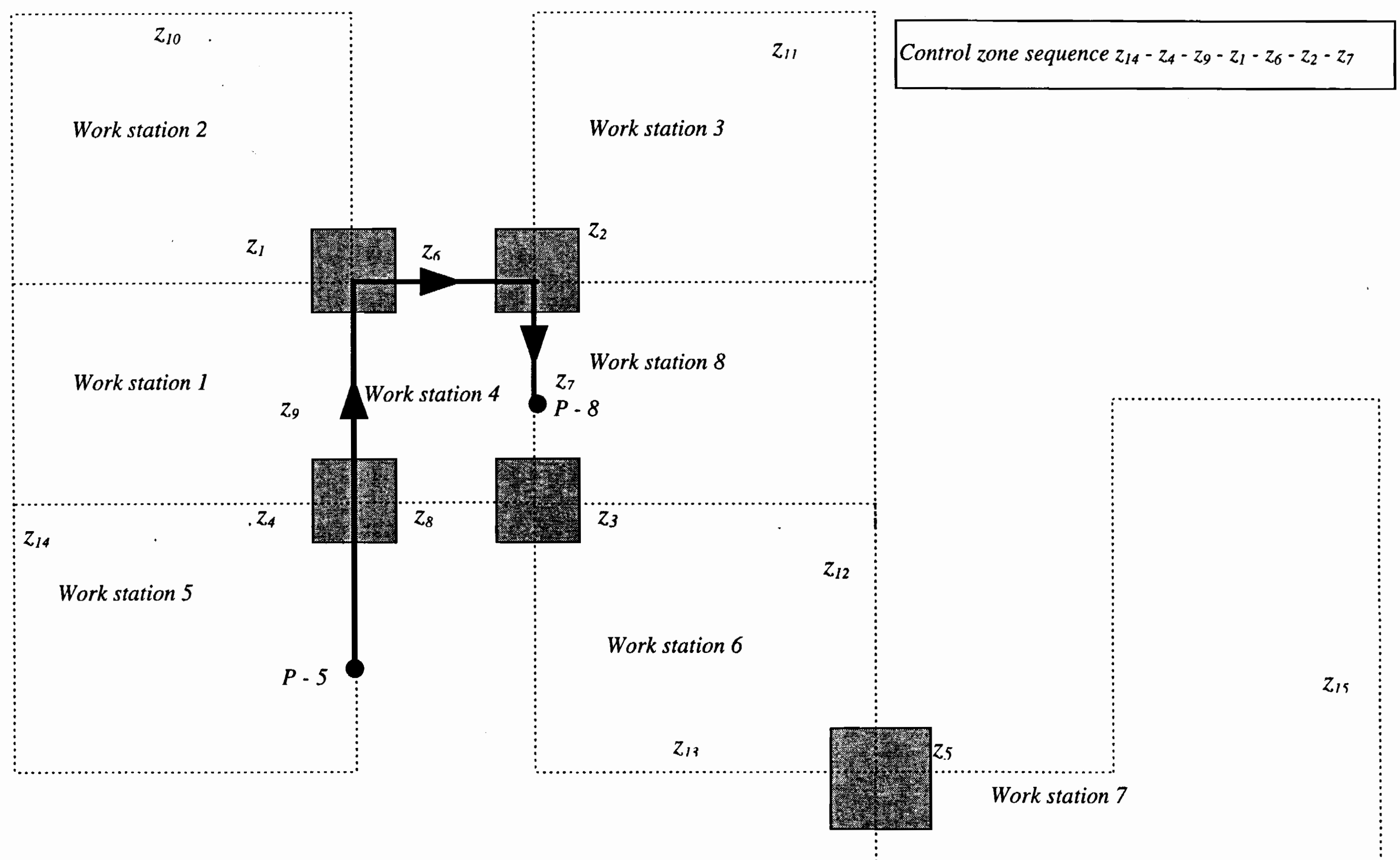


Figure A.16 Illustration of the movement of an AGV between work stations 5 and 8

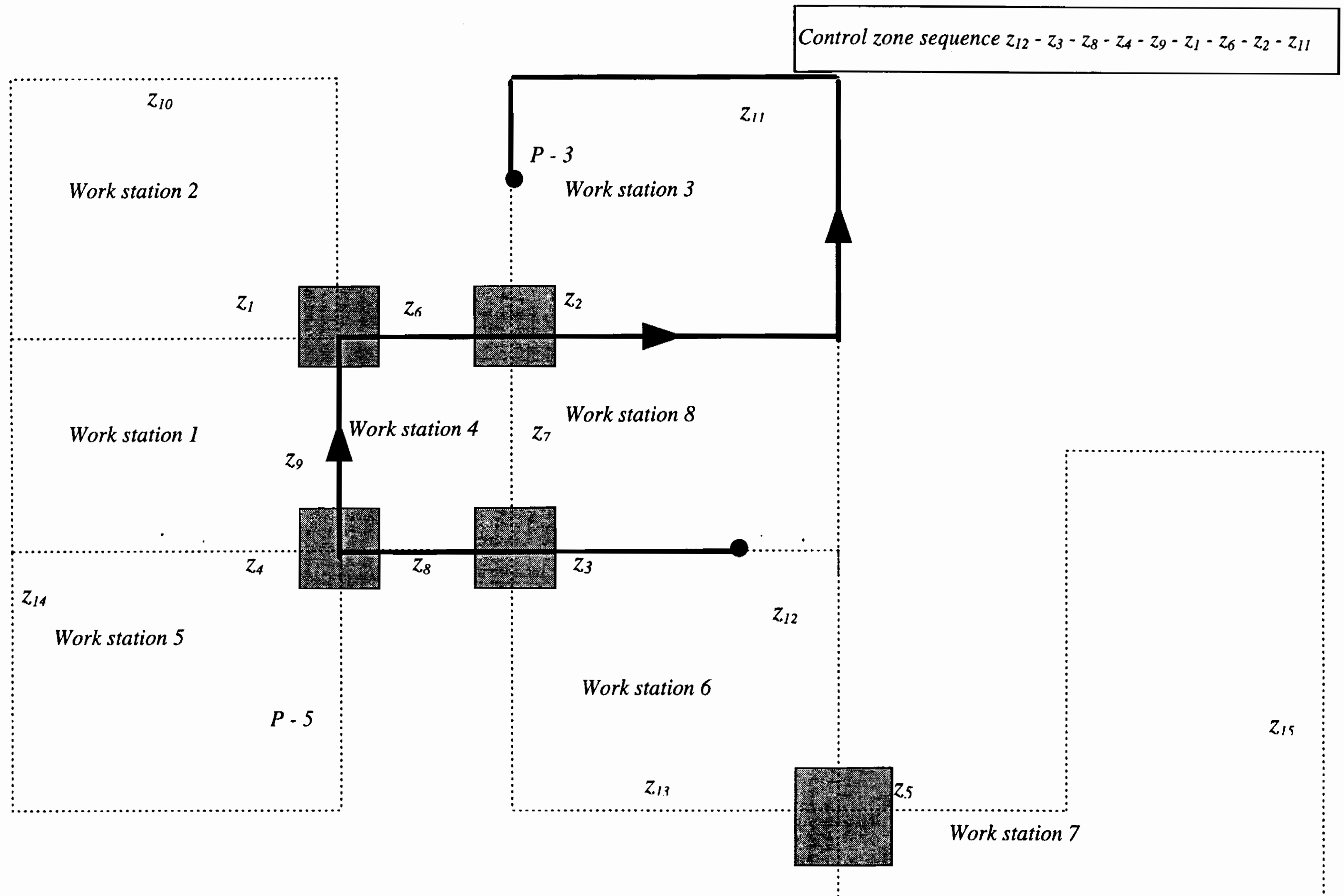


Figure A.17 Illustration of the movement of an AGV between work stations 6 and 3

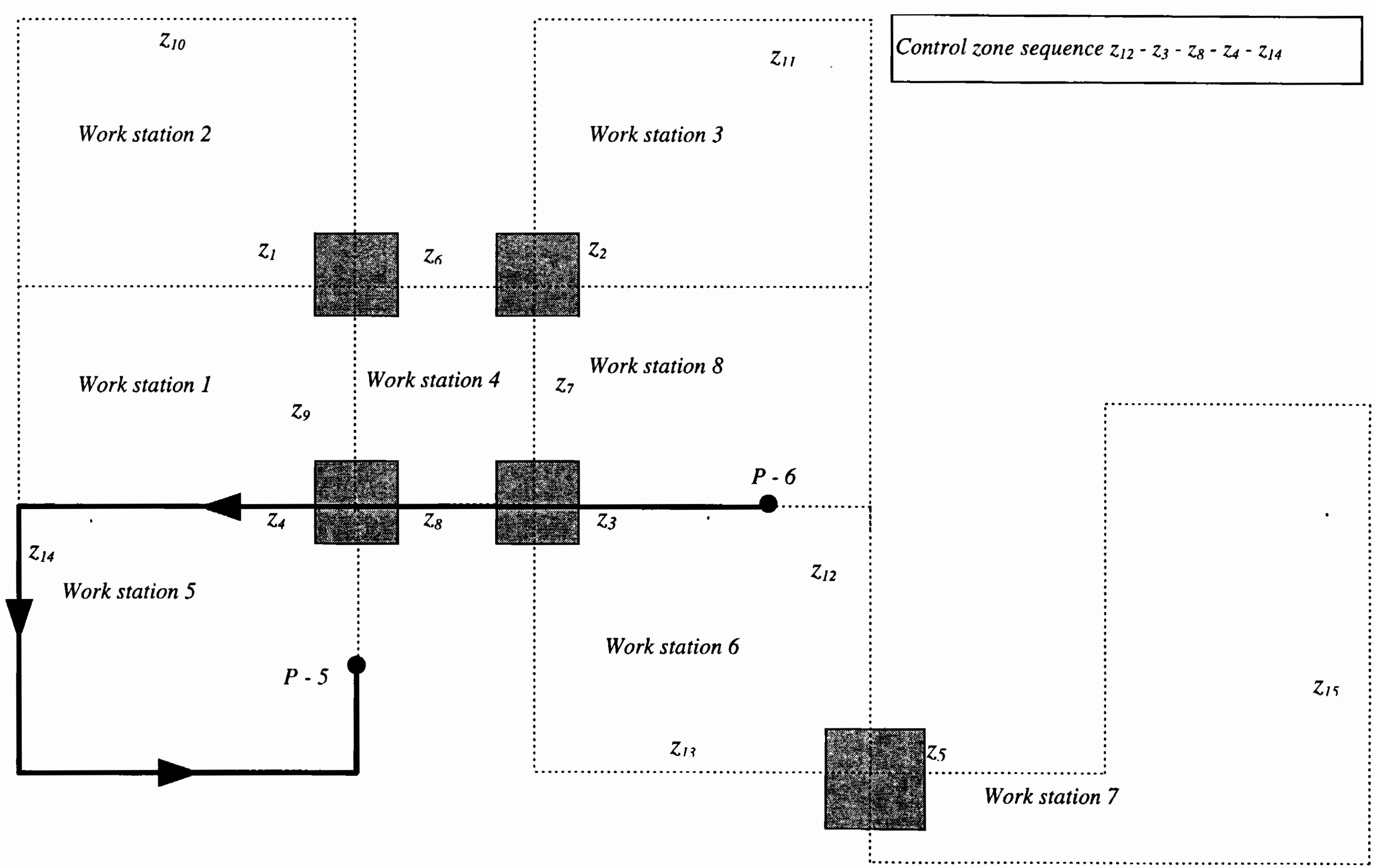


Figure A.18 Illustration of the movement of an AGV between work stations 6 and 5

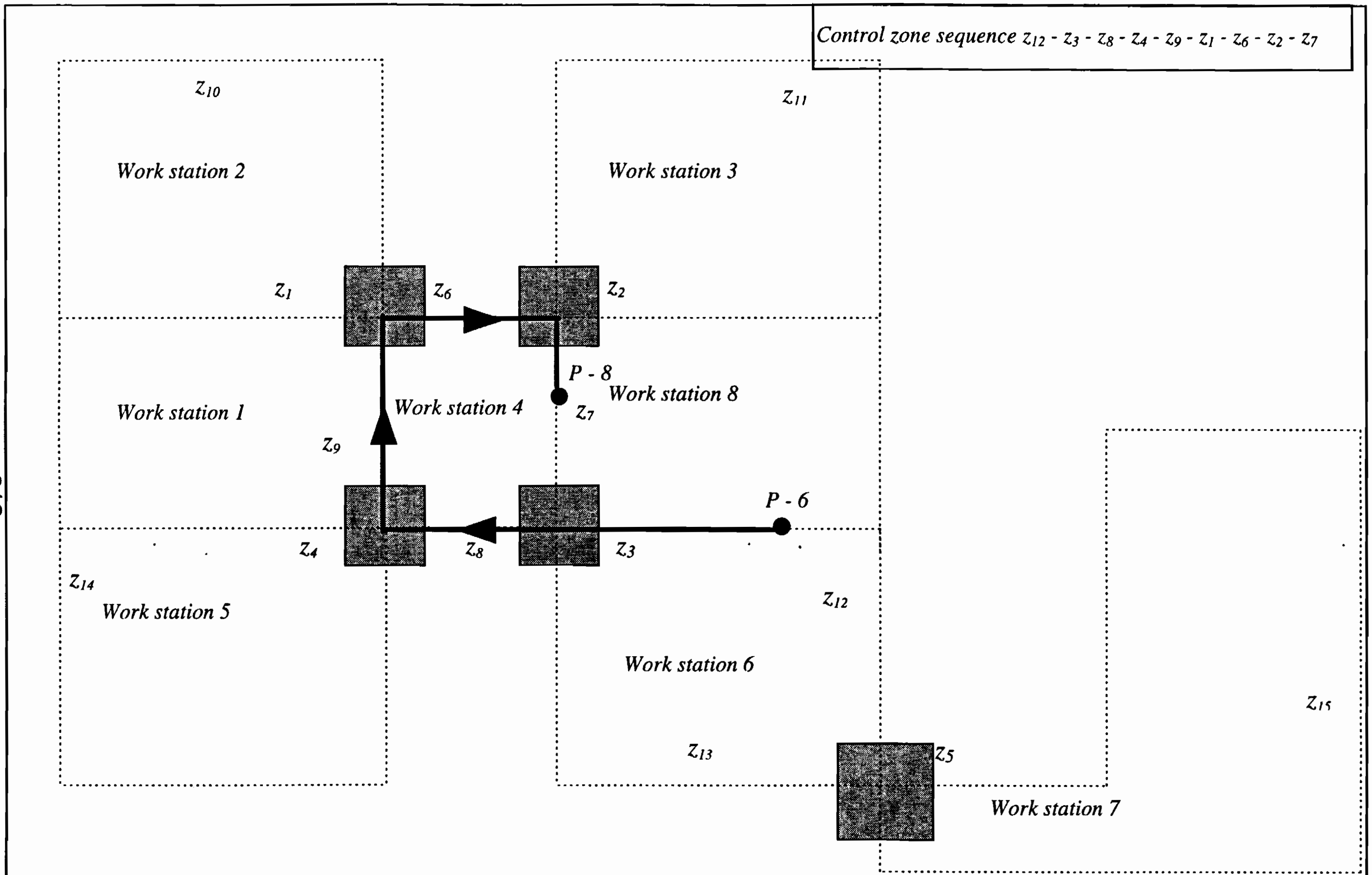


Figure A.19 Illustration of the movement of an AGV between work stations 6 and 8

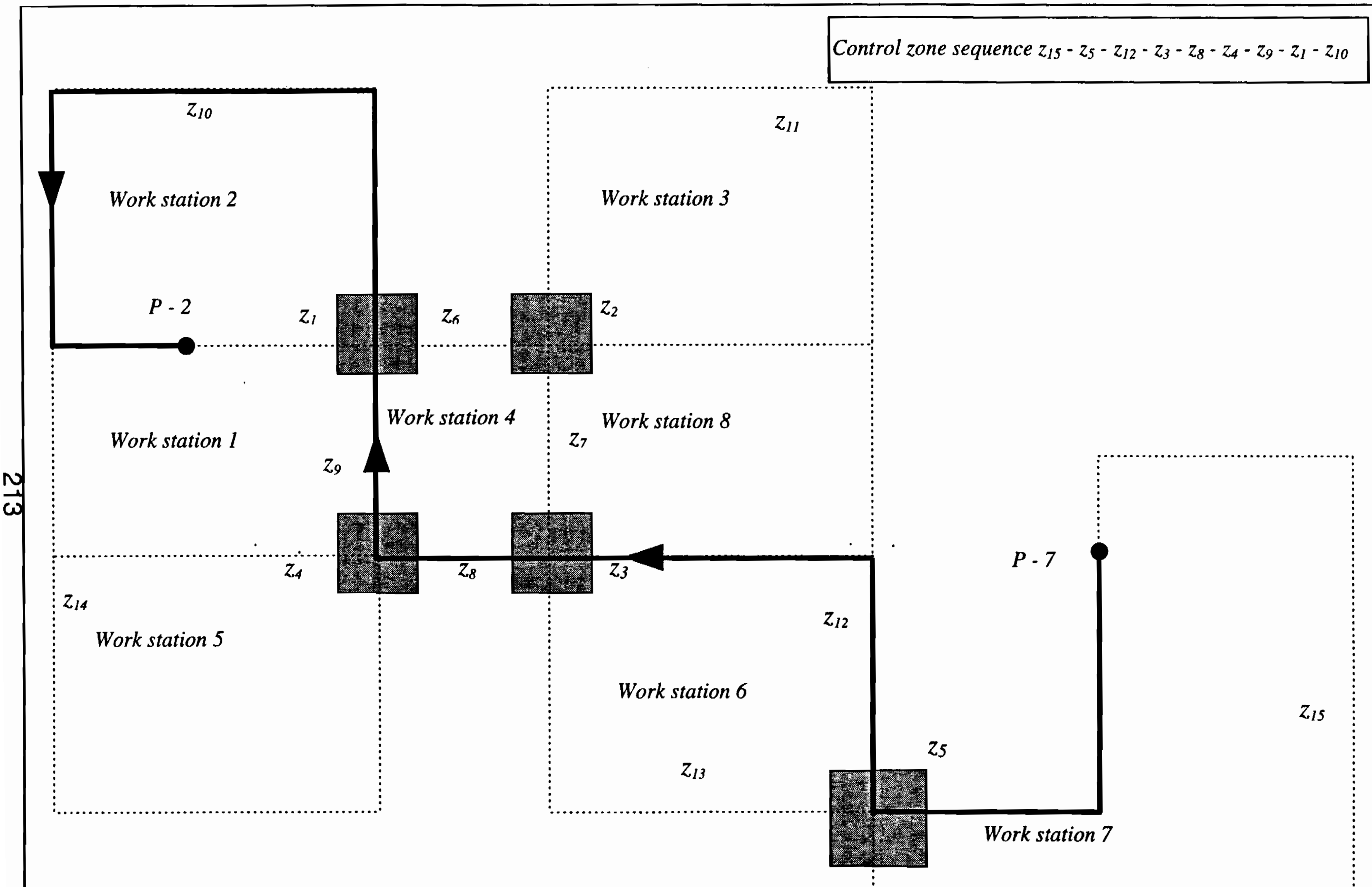


Figure A.20 Illustration of the movement of an AGV between work stations 7 and 2

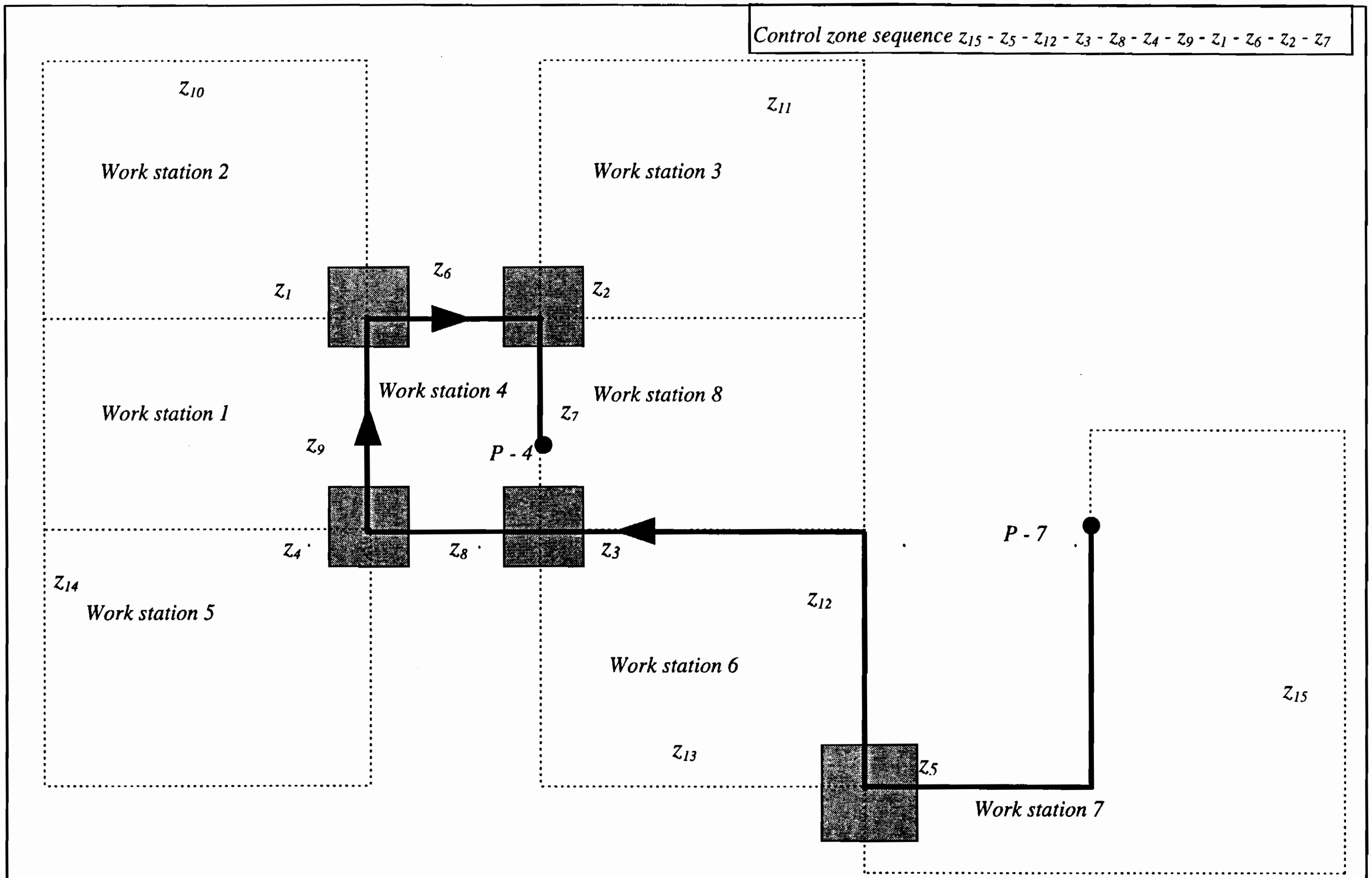


Figure A.21 Illustration of the movement of an AGV between work stations 7 and 4

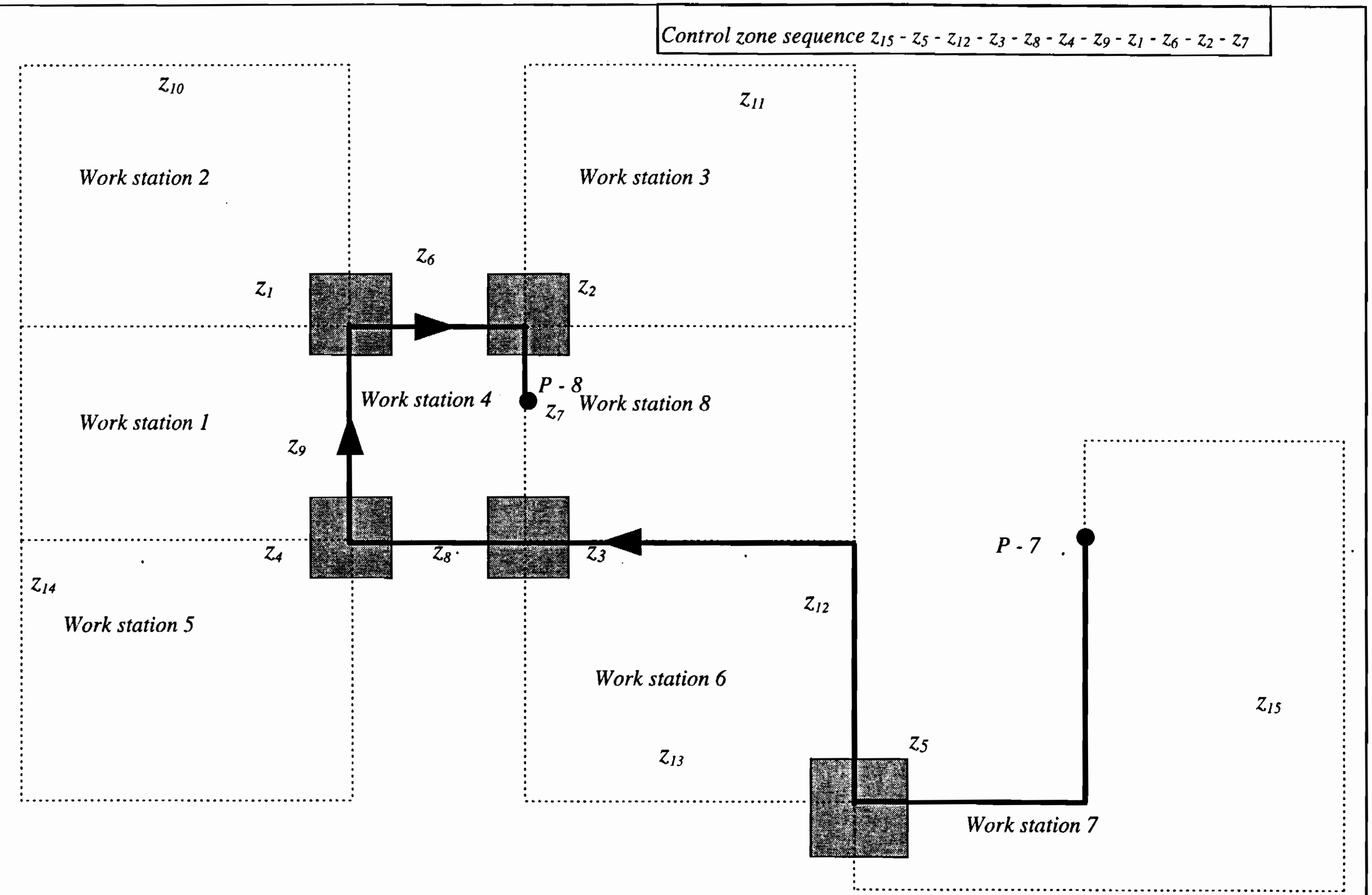


Figure A.22 Illustration of the movement of an AGV between work stations 7 and 8

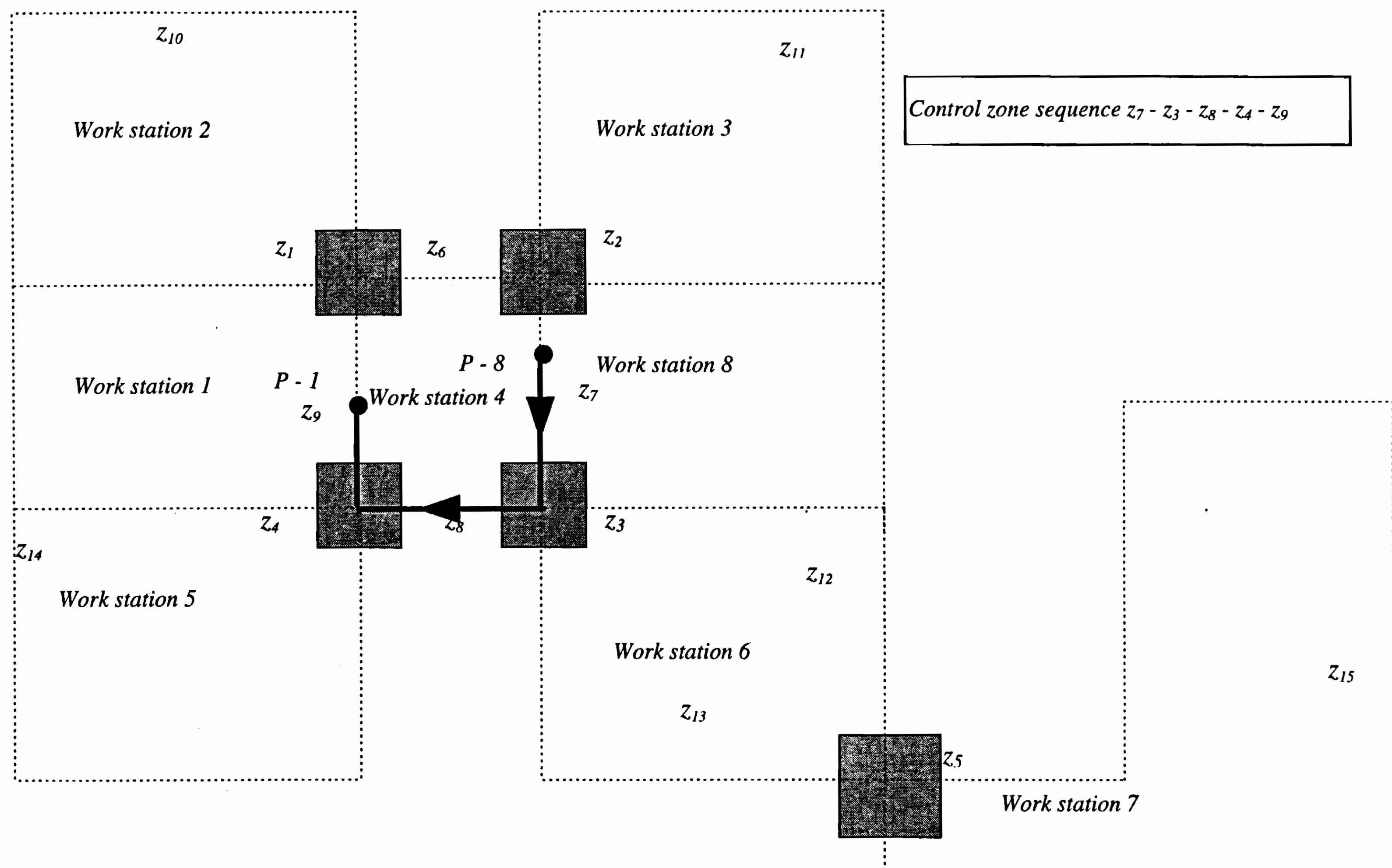


Figure A.23 Illustration of the movement of an AGV between work stations 8 and 1

VITA

The author received his Bachelors of Engineering from the University of Sri Lanka in 1976. He subsequently joined the National Paper Company of Sri Lanka as a mechanical engineer. After serving the National Paper Company in a variety of positions, including maintenance engineer, workshop engineer and mills engineer for five years, he pursued graduate studies at the Asian Institute of Technology in Industrial Engineering and Management in 1982 and received his MS degree in 1983. After the completion of his MS program at AIT, he rejoined the National Paper Company as the chief engineer and later served as an assistant mills manager. In 1990 he left the paper company and joined the University of Sri Lanka as a senior lecturer in Production Engineering. In 1992, he joined Virginia Tech to pursue a MS degree in Industrial and Systems Engineering. He continues with a Ph.D. program in Manufacturing Engineering under the guidance of Dr. M. P. Deisenroth.