This is to certify that the
thesis entitled

REAL TIME ADAPTIVE TASK SCHEDULE FOR A FLEXIBLE MANUFACTURING SYSTEM

presented by

QI ZHU

has been accepted towards fulfillment
of the requirements for the

M.S. degree in ECE

Satish Udpa for Prof Ning Xi
Major Professor's Signature

August 27, 2004
Date

DATE DUE	DATE DUE	DATE DUE

REAL TIME ADAPTIVE TASK SCHEDULE FOR A FLEXIBLE MANUFACTURING SYSTEM

BY

QI ZHU

A THESIS

Submitted to

Michigan State University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

2004

ABSTRACT

REAL TIME ADAPTIVE TASK SCHEDULE FOR A FLEXIBLE MANUFACTURING SYSTEM

By

Qi Zhu

Manufacturers always need flexible and reliable systems to control and monitor the manufacturing process. The Flexible Manufacturing System (FMS) is one of the answers. By using FMS, manufacturers can produce customized products faster than ever to meet market requirements. The problem of tasks scheduling is one of main research topics for the FMS. The Timed Petri net model introduces the possibility of applying a set of mathematical results, mainly based on the use of Max –Plus algebra, for performance analysis. This thesis aims to build a new scheduling method for a Flexible Manufacturing Work-Cell by merging the Timed Petri net model and Max-Plus algebra. These results can be computed as functions of a certain set of decision parameters. These functions can be used to schedule, plan and control the Flexible Manufacturing Work-Cell, so that it can adapt to machine faults, automatically in real time.

For my mother, father and sisters

Your love and supports make my dream come true. Thank you.

ACKNOWLEDGEMENTS

Foremost, I would like to acknowledge all the invaluable help from my advisor, Dr. Ning Xi, without whom this would not have been possible. Not only the knowledge, but also the research experience and friendship I gained here will be beneficial for the rest of my life.

I would like to thank all my committee members, Dr. Erik Goodman, Dr. Tongtong Li, for showing their interest in my study and my academic career.

I would like to thank Dr. Weihua Shang from Kettering University. The discussions with Dr. Weihua Sheng substantially contributed to my work and broadened my knowledge.

Lastly I would like to thank my family, especially my sisters and my parents. Without their years of encouragement and continuous support, I would not have reached this point.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Historical Perspective on Manufacturing Systems

In the history of the manufacturing systems development, the production systems evolving have four stages: Labor-intensive Production Systems, Mass-Production Lines, Job-shops, and Flexible Manufacturing Systems (FMS)

Before the Industrial Revolution, Labor-intensive Production systems remained as a primary way. Highly-skilled craftsman made a complex product, by using crude tools and materials [1]. These functions restrict the large-scale manufacturing; also the cost is very high. Mass Production system was invented in the beginning of the 20^{th} century. A complex product process is broken down a lot of simple steps, therefore reduce the cost and improve the efficiency, such as the automobiles assembly lines. The mass production lines could produce large volumes of a product at a reasonable cost, but were limited to the production of one, two, or very few different parts. The job shop type systems were capable of producing a variety of product, but at a high cost [3]. Along with the development of electrical devices and computer technologies, the workman is substituted by the computer numerically controlled (CNC) machines and the robots, and the product line is managed by the programmable logic controllers (PLC). These improvements brought about the invention of the Flexible Manufacturing System (FMS). Manufacturers can produce customized products faster than ever to meet market requirements.

1.2 Definitions of Flexible Manufacturing System

Ranky [Ranky, 1983] defines an Flexible Manufacturing System (FMS) as a system dealing with high-level distributed data processing and automated material flow using computer-controlled machines, assembly cells, industrial robots, inspection machines and so on, together with computer-integrated material handing and storage systems. [1] In fact, the scope and variety of flexible manufacturing are commonly disputed and are the focus of many research efforts. However, the components and characteristics of an FMS, as described by different authors and researchers, are generally as follows [1] (Figure 1.2) [Davis *et al.*, 1989]:

- Potentially independent NC machine tools
- An automated material handling system, and
- An overall method of control that coordinates the functions of both the machine tools and material handling system so as to achieve flexibility.

The specific manufacturing situations that would be suitable for the adoption of FMS were identified as early as 1973. The following are the production situations that are encompassed by FMS [Darrow, 1987]:

- A variety of high-precision parts are machined (typically job shop)
- A relatively large number of direct numerical control (DNC) machines are required.
- Some form of automated material handling system (MHS) is used to move the work pieces into, within, and out of the FMS.

- On-line computer control is used to manage the entire FMS under conditions of varying parts production mixes and priorities.

It can be concluded from the above that an FMS involves a number of machining centers and material handling systems integrated by a hierarchy of computer control. Furthermore, FMS is capable of handling flexible routing of parts instead of running parts in straight line through work stations. In a flexible manufacturing system; numerically controlled (NC) machines are controlled by computers; parts are handled by robots; and finished products are carried to specific destinations via automatically guided vehicles (AGVs).

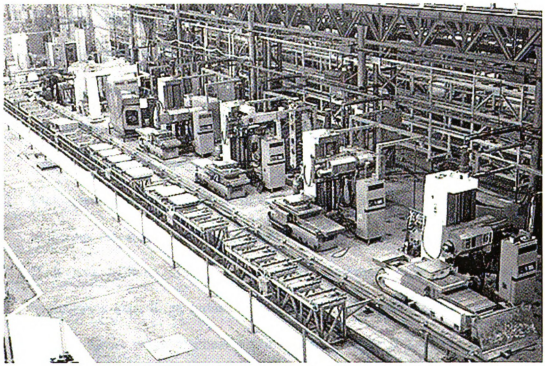


Figure 1.1 Scheme of a FMS

1.3 The Scheduling problem of FMS

Scheduling problems arise when multiple kinds of job types are processed by multiple kinds of shared resources according to their technological precedence constraints. We need to determine the optimal input sequence of jobs and resource usage for a given job mix. The required ordering of operations within each job must be preserved. Production scheduling problem are very complex. Several major approaches to production planning and scheduling are as follows:

1. Heuristic dispatching rules which are widely used in practice. Good rules are obtained based on one's experience. They work but often not optimally. They can also be developed based on the system simulation models. The disadvantage lies in that most comprehensive models and results are difficult to develop and take tremendous computation. Moreover, simulation models are often too specific to particular situations and thus the obtained result cannot be very well generalized.
2. Mathematical programming methods have been extensively studied by numerous researchers and can produce optimum results for some systems [Luh and Hoitomt, 1993; Chen, 1994]. However, only a few real applications exist in an industrial environment. The mathematical models have to ignore many practical constraints in order to solve these models efficiently. These practical constraints, such as material handling capacity, complex resource sharing and routing, and sophisticated discrete-event dynamics, are very difficult to understand for industrial engineers and

management. The optimality will not hold if any parameters or structures change during an operational stage.

3. Computational intelligence-based approaches include expert or knowledge-based systems, genetic algorithms, and neural networks. Knowledge-based systems have difficulty in acquiring the efficient rules and knowledge. The results cannot be guaranteed the best. Both genetic algorithms and neural networks require considerable computation and also have formulation difficulties.
4. Other methods such as algebraic models and control theoretic methods are difficult to use to offer efficient solution methodologies. The methods based on CPM/PERT and queueing networks can provide efficient solution methodologies but cannot describe shared resources, synchronization, and lot sizes easily.

FMS can process multiple products at the same time, employing various resources such as robots, Computer Numerical Controlled (CNC) machines, etc. In a flexible manufacturing system, a process consists of many controlled actions, such as moving robot, picking or placing a part or a raw material. A finite number of resources are shared by several tasks. Scheduling problems arise whenever there are shared resources or alternative routes in an FMS. The resources are shared at the physical level and the job level. At the physical level the robots, convey automatic guided vehicles and related transportation systems are shared by the CNC machines. At the job level, all the flexible machines are shared by different types of jobs. The problem of task scheduling is one of the main research topics for the FMS. The most important problems are how to assign

given resources to different processes required in making each product, to accomplish the best efficiency, and to eliminate the deadlock. The FMS is a kind of discrete event system, and Timed Petri Nets (TPN) is a scheduling and modeling tool for discrete event systems.

Timed Petri Nets (TPN) is a special class of Petri Nets. Petri Nets are graphical and mathematical modeling tools applicable to many systems. Petri Nets were named after Carl A Petri, who created a net-link mathematical tool for the study of communication with automation in 1962. The book [David and Alla, 1992], which is the first of its kind, presents Petri Nets as a modeling tool for discrete event systems. The book [Zhou and DiCesare, 1993] focused on modeling and scheduling [1].

Petri net theory has been applied for modeling, performance analysis and discrete event control of manufacturing systems. Their advantages to represent the complex discrete event dynamics and all the important FMS characteristics have motivated several researchers to investigate their usage for planning and scheduling. Several features of the schedules obtained are:

1. They are event driven. This facilitates the real-time implementation.
2. They are deadlock-free. Since a Petri net model of the system can be a detailed representation of all the operations and resource-sharing cases, a generated schedule spans from the system's initial condition to the final desired one. It thus avoids any deadlock.
3. The completion time is the optimization objective.

The disadvantages of this approach include:

1. A huge search space is required for large complex systems. [2] The speed depends heavily on a heuristic function selected. Study on the best heuristic functions is needed.
2. Most heuristic functions cannot guarantee the optimality of the obtained schedules. The question remains open on how good the resulting schedules are.
3. It remains difficult to use criteria other than makespan [2] in the search process. Other useful criteria include tardiness and due dates.

Although Petri Nets are easy to explain the sequence in a FMS, that can not calculate real-time. Normally, a Petri net only consider the precedence relationships among activities and does not include the time concept. To overcome this disadvantage, Timed Petri Nets (TPN) have been defined and used for performance analysis. [6][7][8] The Timed Petri nets scheduling method first constructs a Timed Petri net model for the FMS with the initial marking, and then generates an optimal schedule in terms of firing sequence of transitions. [9] How to optimize the TPN model is an essential challenge.

The Max-plus Algebra $\Re_{max}$ is the set of real numbers endowed with the operation max and plus. The Max-plus algebra plays a crucial role in at least two fields:

1. Path algebra (research of the path of maximal weight in a graph).
2. Performance evaluation of Discrete Event Dynamic Systems (DEDS)

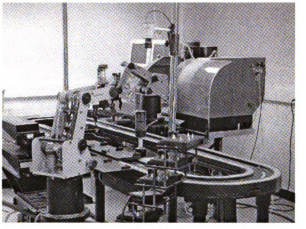
If the corresponding TPN model is an event graph (i.e. each place has exactly one input and one output transition), it is possible to apply a Max-Plus mathematical model for this system. [10] In mathematical performance analysis a complex system can be changed to a few equations in a certain set of decision parameters. Then these functions can be used to optimize the TPN model.

Traditionally, the schedule or the activation sequences are fixed and sequential, and the FMS has to follow the schedule step by step. If there is failure in any step, it will affect the following processes and the full system will be shut down. But in a FMS, the work-cell can usually process multiple products at the same time, so some products just needs some steps, not full steps. Obviously the traditional scheduling method can not obtain the best efficiency. An un-necessary step will affect its process.

In this research, we build the Max-plus Algebra model via the TPN model, and using this model to calculate the value of every step state. The scheduling method presented in this thesis first formulates a scheduling problem using a Timed Petri Net model for each kind of product, and generates the firing sequences for each kind of them. For the full work-cell, the parallel sequence (i.e. each product is processed according to its sequence) is used. Then Max-Plus algebra is used to analyze the performance and design the controllers to avoid the conflicts. The stability of the FMS is proved.

1.4 **Framework of the Work-Cell**

In my research, a robotic manufacturing work-cell consist two robots, each with five joints, two different kinds of CNC machines, and a disc conveyor. (Figure 1.2) This work-cell can process two kinds of run material, block and cylinder. Robot 1 and robot 2 pick up the run material and the parts; the CNC mill and CNC lathe process the run material; the conveyor transfers the raw material and parts from position 1 to position 2 or from position 2 to position 1. In each of the tasks of the FMS, a sequence of operations of the robots and the disc conveyor is performed under a discrete event dynamic system (DEDS). Scheduling these operations is my research topic.



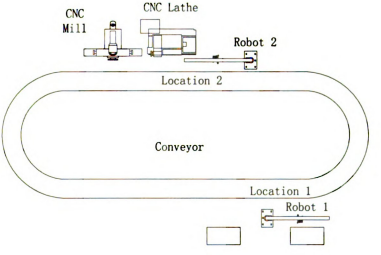


Figure 1.2 The Structure of the FMS Work-cell

CHAPTER 2

Modeling

2.1 Petri Net Model

2.1.1 Definition

We use Petri Nets to model the flow of production in an FMS.

Definition 2.1: *A Petri Net is a quintuple* $G = \{P, T, I, O, m_0\}$ *where P is a finite set of places, T is a finite set of transitions,* $I \subseteq P \times T$ *and* $O \subseteq T \times P$ *are the sets of directed arcs connecting places and transitions. Set I is from places to transitions. We denote these as input arcs, and the places are input places. Set O is from transitions to places. We denote these as output arcs, the places are output places.* $m_0 : P \rightarrow N$ *is the initial marking of* G, *where* N *is the set of nonnegative integers.*

In the graphical representation of a Petri Net, places are drawn as circles and transitions as bars. A directed arc goes from a place to a transition. There is no arc from a place to a place or a transition to a transition. The marking (token) m of a Petri Net G indicates the number of tokens in each place, which is the current state of the system. The set of all markings is denoted as M.

In order to represent the change of FMS states, we identify the marking of a Petri Net with the state. A marking or state of a Petri Net is changed according to the following transition (firing) rule:

1. A transition is said to be enabled if each input place has at least w tokens, and w is the weight of the arc from the input place to the transition.(Figure 2.1)

2. A firing of an enabled transition moves n tokens into each output place, and moves w tokens out from each input place. The n is the weight of the arc from the transition to the output place.(Figure 2.2)

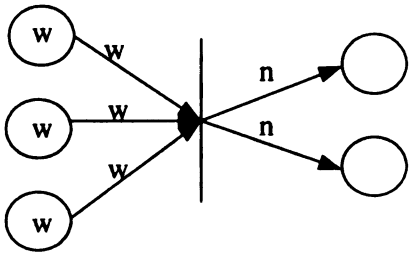


Figure 2.1 Before the firing (PN model)

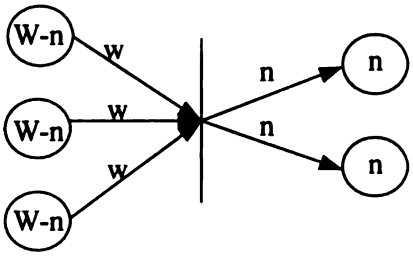


Figure 2.2 After the firing (PN model)

The original theory of Petri Nets deals with the ordering of events. My research is related to performance evaluation that is, how fast can a work cell produce a part? So, just the state information is not enough. It is necessary to introduce time. There are two basic ways to introduce time. One is with transition firing time the other is the sojourn time of token in place.

We using associated with firing time to represent production times in a FMS. We adopt the following definition.

Definition 2.2: *The firing time of a transition is the time that elapses between the starting and the completion of the firing of the transition. (1)*

By introducing the definition of firing time, the original Petri Nets become the Timed Petri Net. In a timed Petri Net, after each input place receives w tokens, all the input places are available. (Figure 2.3) If the firing time of a transition is τ , after τ the n tokens are deposited in each output place by the enable transition. (Figure2.4)

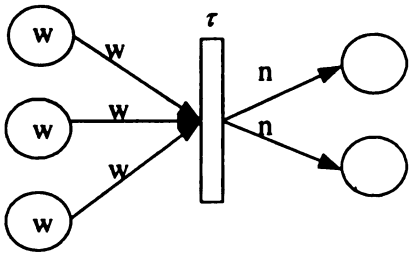


Figure 2.3 Before the firing (TPN model)

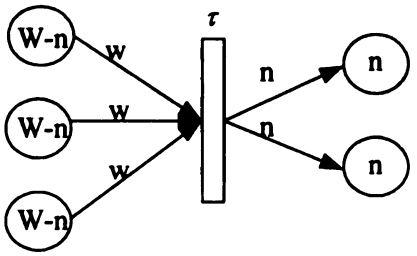


Figure 2.4 After the firing (TPN model)

2.1.2 Timed Petri Nets Model for One Robot Conveying Manipulation

My research involves an FMS work cell with many tasks. The Robot Conveying manipulation is the basic task. We can draw a Timed Petri Net as shown in the figure 2.5. The input is a part or run material. If the processing is finished we call that is part. If not finished we call that is run material.

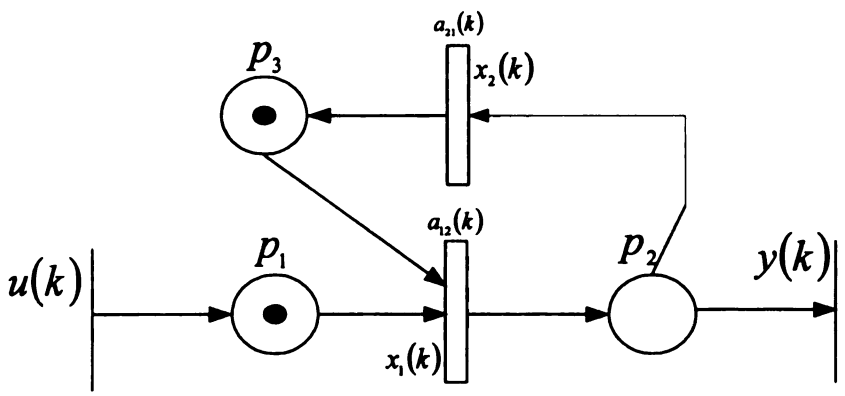


Figure 2.5 The TPN model for one robot conveying manipulation

- p_1 represents that the *kth* input is arriving.
- p_2 represents that the *kth* input is in the specific place and the robot will move back for the k+1th input.
- p_3 represents that the robot is available.

- $u(k)$ represents the time that the *kth* input is arriving
- $x_1(k)$ represents the time that the robot picking up the *kth* input.
- $x_2(k)$ represents the time that the robot is moving back.
- $y(k)$ represents the time that the *kth* input is leaving

The firing time can be expressed by $\{a_{12}(k), a_{21}(k)\}$, where k is the *kth* input. In the graph, the firing time is drawn as a box and the dot is the token. Since for each weight for each arc is 1, only one token is allowed in each place. The initial marking m_0 is $[1,0,1]^T$ .The token in the first place p_1 represents "the input is ready". The token in the place p_3 represents "the robot is available to pick up the input". Then the transition T_1 is enabled at time $x_1(k)$. After time delay $a_{12}(k)$, the token is moved to the place p_2. Now the transition T_2 is enabled at time $x_2(k)$, with time delay $a_{21}(k)$ a token is deposited in the place p_3. Associating with the token in the p_1, a circle as above starts again. So, this timed Petri Nets model can be depicted as blow.

- The initial state is that the robot is available to pick up the first input, and the first input is arriving
- At time $x_1(1)$, the robot picks it up and moves to anther place with time delay $a_{12}(1)$
- The robot moves back at time $x_2(1)$, with time delay $a_{21}(1)$.
- When another part or run material arrives, a new cycle will start.

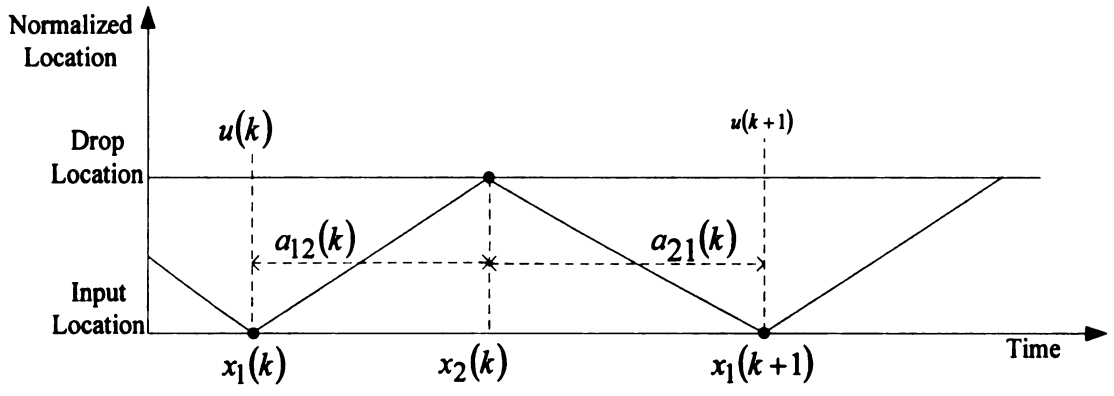


Figure 2.6 Timing Diagram representing one robot conveying manipulation

This is the Timed Petri Net model for the one robot conveying manipulation. We can use this model to build the model for the full system.

2.1.3 Timed Petri Net Model

This FMS can process two different kinds of raw material; one is a block the other is cylinder. We can build up two different timed Petri Net models respectively for those two different kinds of raw material.

I. Timed Petri Net Model for the Block Process

The raw material of the block process is a block. As shown in the Figure 2.7, the robot 1 picks up the block and then loads it on the conveyer of location 1. When the block is transported to the location 2, the robot 2 loads it into the mill. After the processing is finish, the robot 2 loads the part on the conveyor. When the part arrives at the location 1, the robot 1 loads it into part storage.

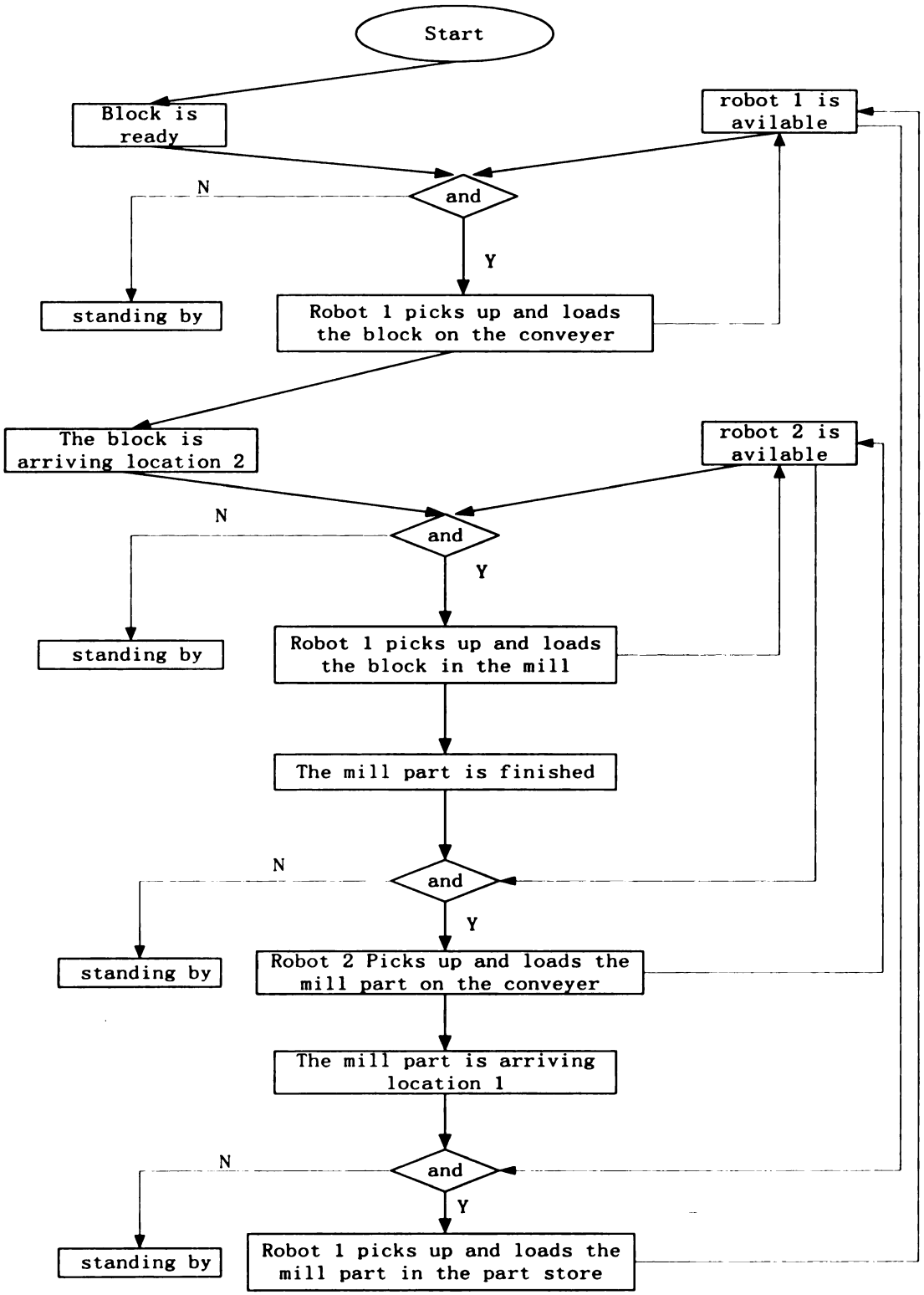


Figure 2.7 The logical flow chart of block process

Based on the tasks of the block process, we can draw the graph of the Timed Petri Net model as shown Figure 2.8.

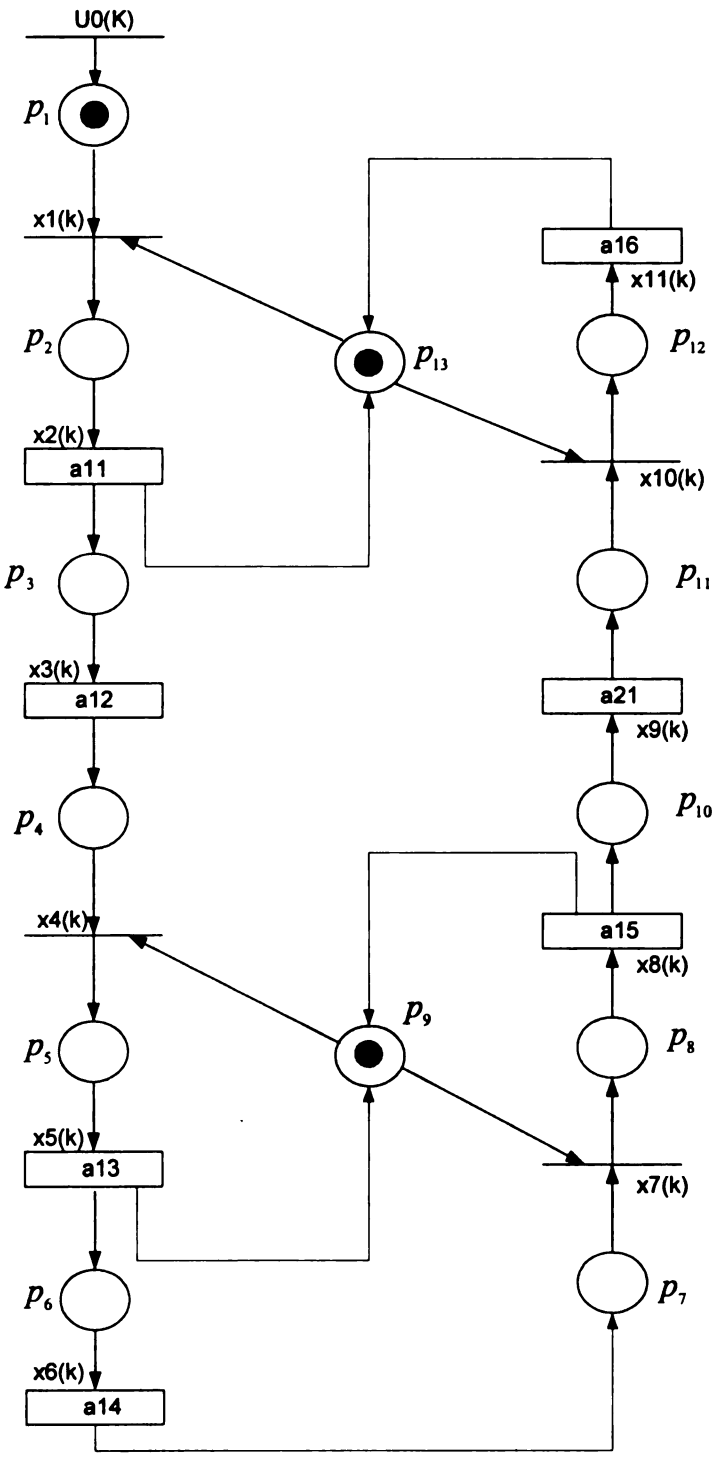


Figure 2.8 The TPN Model for the Block Process

- p_1 represents that the *kth* block is arriving.
- p_2 represents that robot 1 picks up the *kth* block.
- p_3 represents that robot 1 loads the *kth* block on the conveyor.(Location 1)
- p_4 represents that the *kth* block arrives at the CNC machine side (location 2) by the conveyor.
- p_5 represents that robot 2 picks up the *kth* block.
- p_6 represents that robot 2 loads the *kth* block into the mill.
- p_7 represents that the *kth* milled part is finished.
- p_8 represents that robot 2 picks up the *kth* milled part.
- p_9 represents that robot 2 unloads the *kth* milled part on the conveyor.
- p_{10} represents that robot 2 is available.
- p_{11} represents that the *kth* milled part arrives at the location 1.
- p_{12} represents that robot 1 picks up the *kth* milled part and loads it in the part store.
- p_{13} represents that robot 1 is available.
- $x_1(k)$ represents the time of robot 1 picks up the *kth* block
- $x_2(k)$ represents the time of robot 1 loads the *kth* block on the conveyor
- $x_3(k)$ represents the time of the *kth* block arrives at the location 2
- $x_4(k)$ represents the time of robot 2 picks up the k*th* block
- $x_5(k)$ represents the time of robot2 loading the *kth* block in the mill
- $x_6(k)$ represents the time of the *kth* milled part finished

- $x_7(k)$ represents the time of robot 2 picks up the *kth* milled part
- $x_8(k)$ represents the time of the robot 2 and loading the *kth* mill part on the conveyor
- $x_9(k)$ represents the time of the *kth* mill part arriving on the location 1
- $x_{10}(k)$ represents the time of robot 1 picking up the *kth* mill part
- $x_{11}(k)$ represents the time of robot 1 loading the *kth* mill part in the part store

In this model, we denote the firing time is $\{a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{21}, a_{16}\}$. The initial marking m_0 is $[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$. Each place just one token is allowed.

The flow is that

$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_1 and P_{13} either one has a token
T_1 is firing at $x_1(k)$
A token is deposited in P_2

$[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$

P_2 has a token
T_2 is firing at $x_2(k)$ with firing time a_{11}
A token is deposited in P_3 and P_{13}

$[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_3 has a token
T_3 is firing at $x_3(k)$ with firing time a_{12}
A token is deposited in P_4

$[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_3 with a token

T_3 is firing at $x_3(k)$ with firing time a_{12}

A token is deposited

$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_4 and P_9 each one with a token

T_4 is firing at $x_4(k)$

A token is deposited in P_5

$[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$

P_5 with a token

T_5 is firing at $x_5(k)$ with firing time a_{13}

A token is deposited in P_6 and P_9

$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_6 with a token

T_6 is firing at $x_6(k)$ with firing time a_{14}

A token is deposited in P_7

$[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_7 and P_9 each one with a token

T_7 is firing at $x_7(k)$

A token is deposited in P_8

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1]^T$

P_8 with a token

T_8 is firing at $x_8(k)$ with firing time a_{15}

A token is deposited in P_{10} and P_9

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1]^T$

P_{10} with a token

T_9 is firing at $x_9(k)$ with firing time a_{21}

A token is deposited in P_{11}

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1]^T$

P_{11} and P_{13} each one with a token

T_{10} is firing at $x_{10}(k)$

A token is deposited in P_{12}

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0]^T$

P_{12} with a token

T_{11} is firing at $x_{11}(k)$ with firing time a_{16}

A token is deposited in P_{13}

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

Associating with the token in P_1

The circle will be started again

$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

This Petri nets describes that

- The initial condition is robot 1 is available and the first block is arriving.
- Robot 1 picks up the first block at time $x_1(1)$.
- Robot 1 loads the first block on the conveyor at time $x_2(1)$ with time delay a_{11}, and then robot 1 moves back.
- At time $x_3(1)$, the first block is moved to the location 2 by the conveyor, with time delay a_{12}.

- At time $x_4(1)$, the first block is arriving at location 2, and robot 2 picks up the first block at the same time.
- At time $x_5(1)$, robot 2 load the first block into the mill with the time cost a_{13}, and then robot 2 go back.
- The mill starts processing the first block at time $x_6(1)$, with the processing time a_{14}.
- The first mill part will be finished at time $x_7(1)$, and robot 2 picks it up at the same time.
- At time $x_8(1)$, robot 2 loads it on the conveyor, with time cost a_{15}, then robot 2 moves back.
- At time $x_9(1)$, the first mill part start transfer from location 2 to location 1, with time delay a_{21}.
- At time $x_{10}(1)$, the first mill part is arriving at location 1, and robot 1 picks it up.
- At time $x_{11}(1)$, robot 1 loads it into the part store with time deal a_{17},and then the robot 1 moves back.

This is the timed Petri nets model for the block process.

II. Timed Petri Net Model for the Cylinder Process

This model looks the same as the block process model, so the flow chart is the same. But the definitions of the places and the transitions are different (figure 2.9).

- p_1 represents that the *Lth* cylinder arrives
- p_2 represents that robot 1 picks up the *Lth* cylinder
- p_3 represents that robot 1 loads the *Lth* cylinder on the conveyor.(Location 1)
- p_4 represents that the *Lth* cylinder is arriving on the CNC machine side (location 2) by the conveyor.
- p_5 represents that robot 2 picks up the *Lth* cylinder
- p_6 represents that robot 2 loads the *Lth* cylinder into the lathe
- p_7 represents that the *Lth* lathe part is finished.
- p_8 represents that robot 2 picks up the *Lth* lathe part
- p_9 represents that robot 2 unloads the *Lth* lathe part on the conveyor
- p_{10} represents that robot 2 is available.
- p_{11} represents that the *Lth* lathe part is arriving on the location 1 (the part store side)
- p_{12} represents that robot 1 picks up the *Lth* lathe part and loads it in the part store.
- p_{13} represents that robot 1 is available.
- $x_1(L)$ represents the time of robot 1 picking up the *Lth* cylinder

- $x_2(L)$ represents the time of robot 1 loading the *Lth* cylinder on the conveyor
- $x_3(L)$ represents the time of the *Lth* cylinder arriving on the location 2
- $x_4(L)$ represents the time of robot 2 picking up the *Lth* lathe part
- $x_5(L)$ represents the time of robot2 loading the Lth cylinder in the lathe
- $x_6(L)$ represents the time of the *Lth* lathe part finished
- $x_7(L)$ represents the time of robot 2 pick up the *Lth* lathe part
- $x_8(L)$ represents the time of the robot 2 and loading the *Lth* lathe part
- $x_9(L)$ represents the time of the *Lth* lathe part arriving on the location 1
- $x_{10}(L)$ represents the time of robot 1 picking up the *Lth* lathe part
- $x_{11}(L)$ represents the time of robot 1 loading the Lth lathe part in the part store

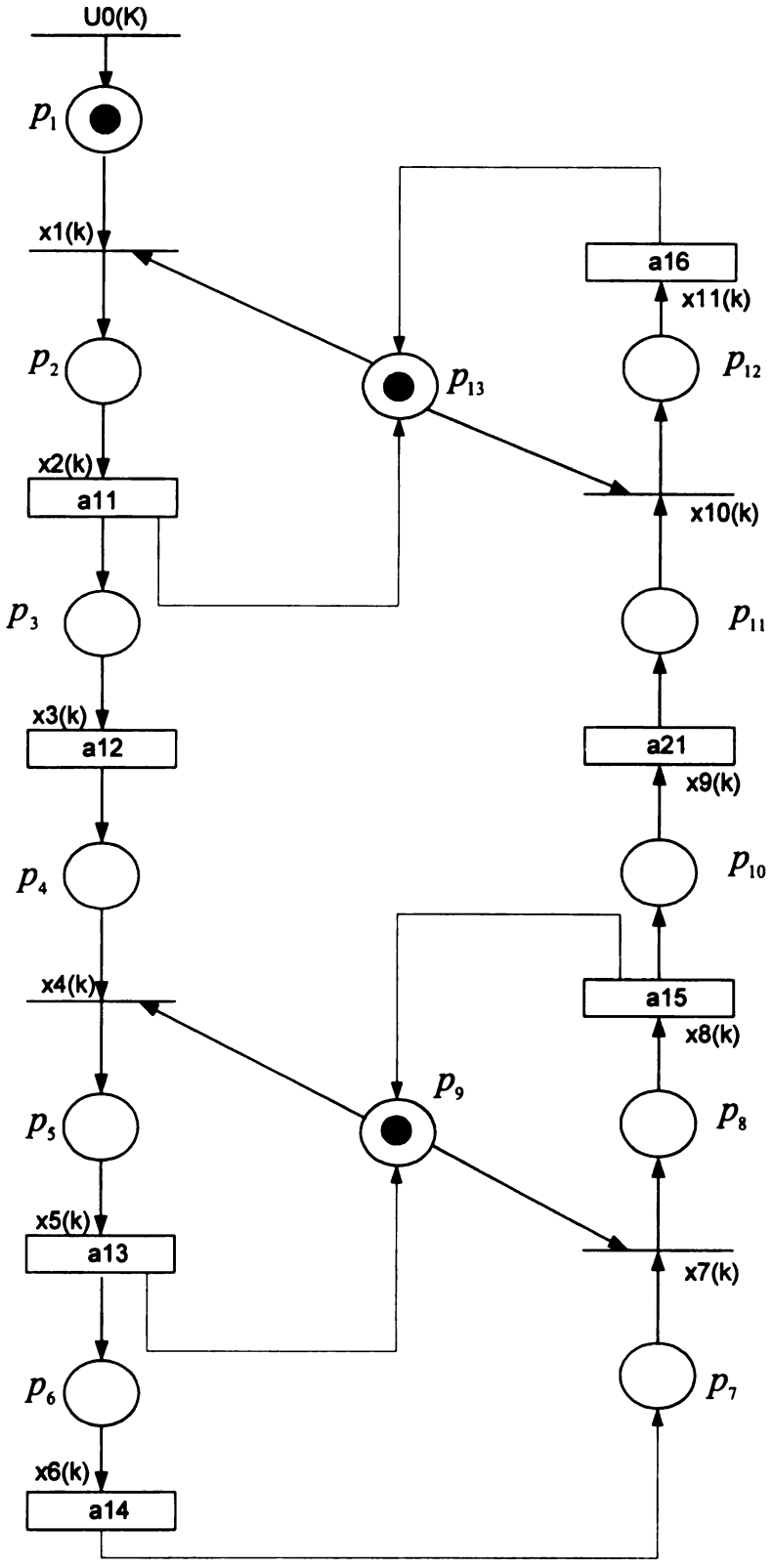


Figure 2.9 The TPN model for cylinder process

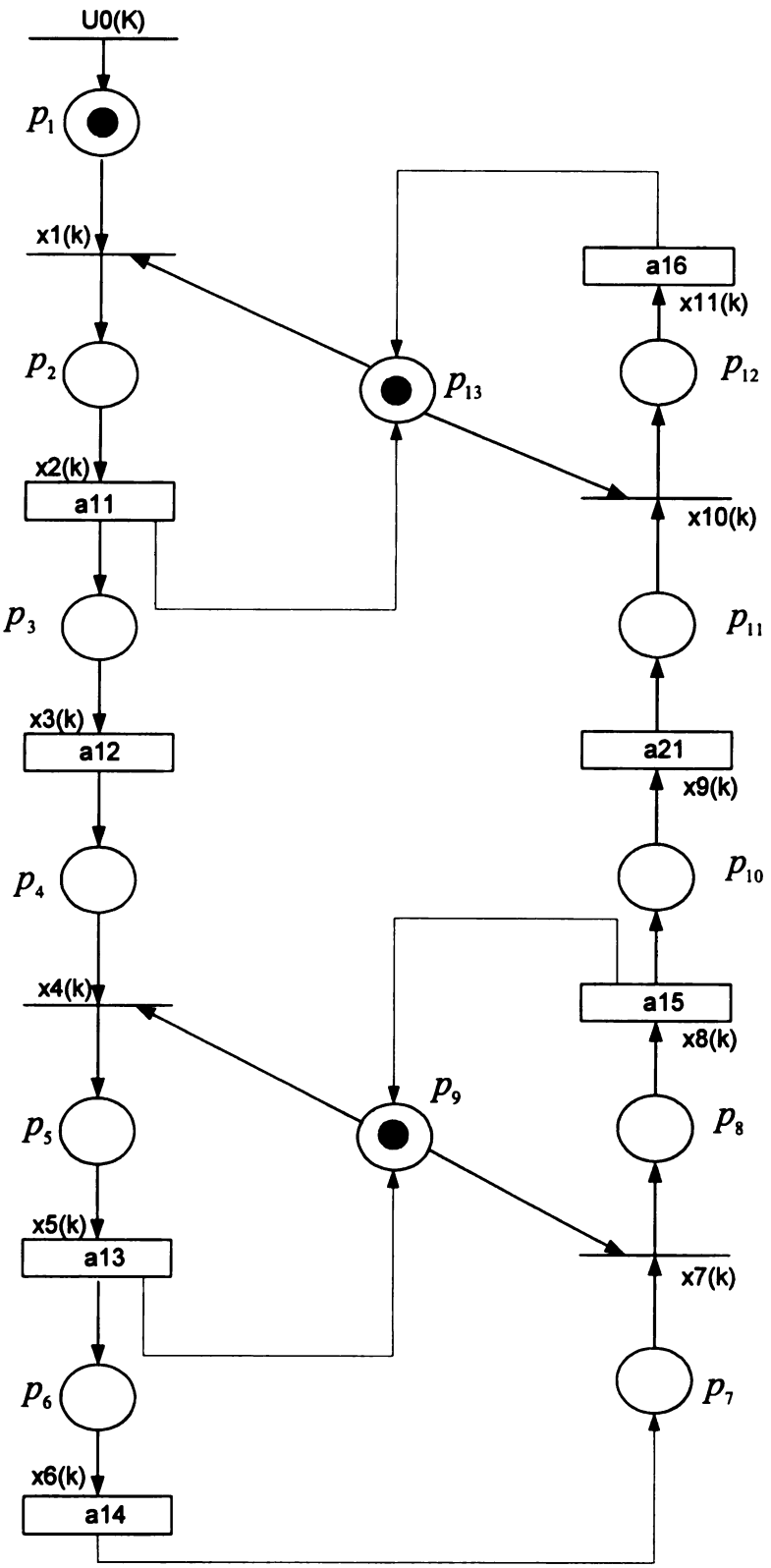
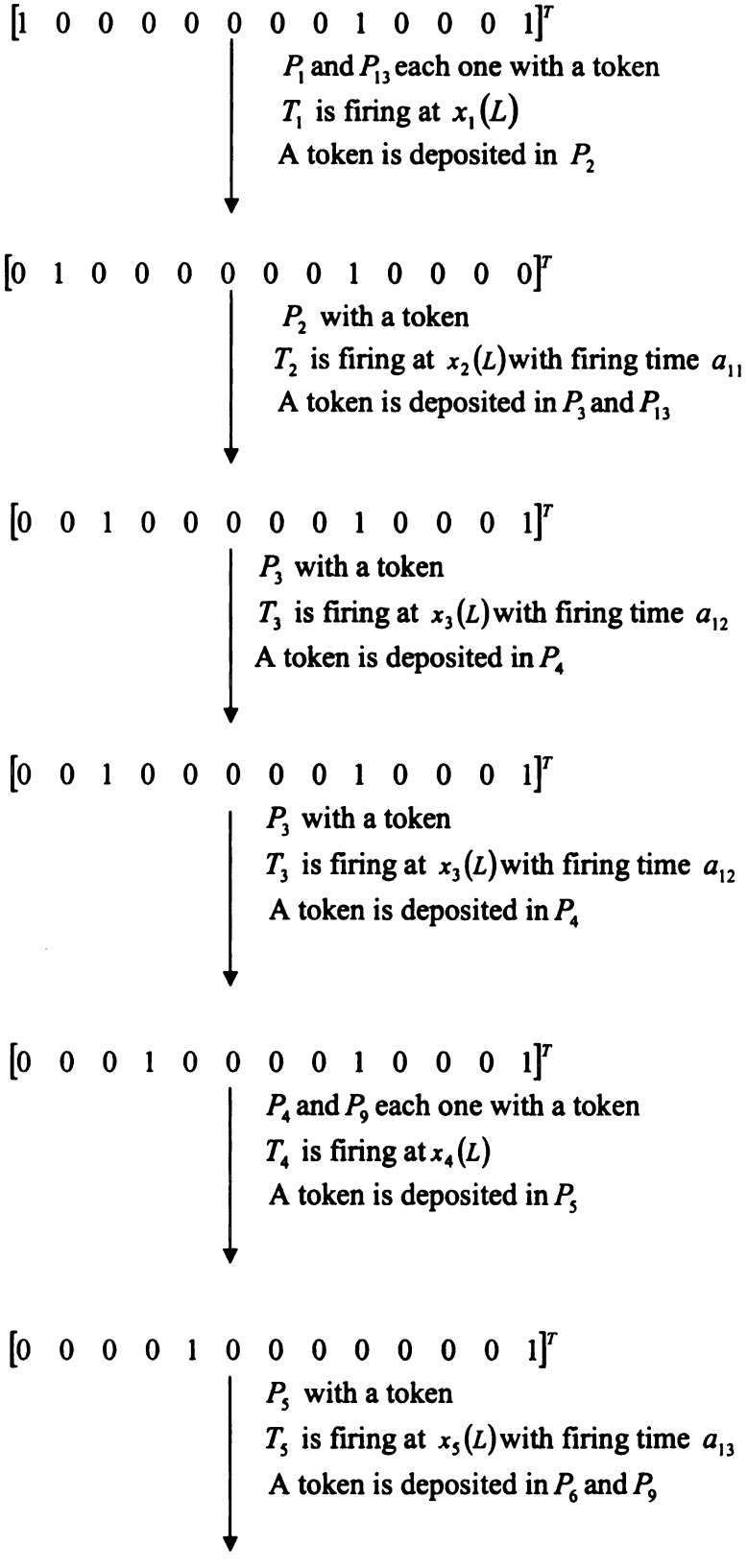


Figure 2.9 The TPN model for cylinder process

The flow is that



$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_6 with a token

T_6 is firing at $x_6(L)$ with firing time a_{14}

A token is deposited in P_7

$[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1]^T$

P_7 and P_9 each one with a token

T_7 is firing at $x_7(L)$

A token is deposited in P_8

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1]^T$

P_8 with a token

T_8 is firing at $x_8(L)$ with firing time a_{15}

A token is deposited in P_{10} and P_9

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1]^T$

P_{10} with a token

T_9 is firing at $x_9(L)$ with firing time a_{21}

A token is deposited in P_{11}

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1]^T$

P_{11} and P_{13} each one with a token

T_{10} is firing at $x_{10}(L)$

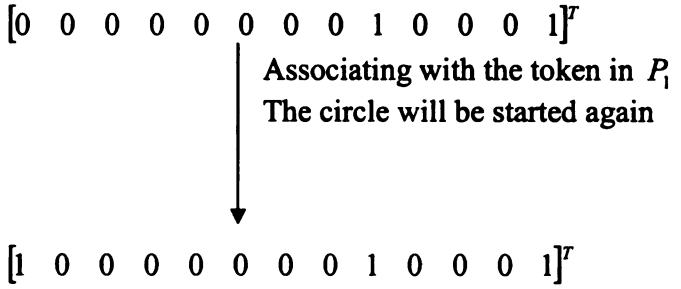
A token is deposited in P_{12}

$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0]^T$

P_{12} with a token

T_{11} is firing at $x_{11}(L)$ with firing time a_{16}

A token is deposited in P_{13}



This Petri nets describes that

- The initial condition is robot 1 is available and the first cylinder is arriving.
- Robot 1 picks up the first block at time $x_1(1)$.
- Robot 1 loads the first cylinder on the conveyor at the time $x_2(1)$with the time delay a_{11}, and then robot 1 moves back.
- At the time $x_3(1)$, the first cylinder is moved at location 2 by the conveyor, with the time delay a_{12}.
- At time $x_4(1)$, the first cylinder is arriving on the location 2, and robot 2 picks up the first block at the same time.
- At time $x_5(1)$, robot 2 loads the first cylinder into the lathe with the time cost a_{13}, and then robot 2 goes back.
- The mill starts process the first cylinder at time $x_6(1)$, with processing time a_{14}.
- The first mill part will be finished at time $x_7(1)$, and robot 2 picks it up at the same time.

- At the time $x_8(1)$, robot 2 loads it on the conveyor, with time cost a_{15}, then robot 2 moves back.
- At the time $x_9(1)$, the first lathe part starts transfer from location 2 to location 1, with time delay a_{21}.
- At the time $x_{10}(1)$, the first lathe part is arriving at location 1, and robot 1 picks it up.

At the time $x_{11}(1)$, robot 1 loads it into the part store with time delay a_{17},and then the robot 1 moves back.

This is the timed Petri nets model for the cylinder process

III. Timed Petri Net model for the full system

The TPN model for the full system is as shown on figure 2.10. The left line is for the block process, and the right line is for the cylinder process. The middle of two places represent the state of two robots. If robot 1 is available, the top one has a token, and similarly if robot 2 is available, the one below has a token.

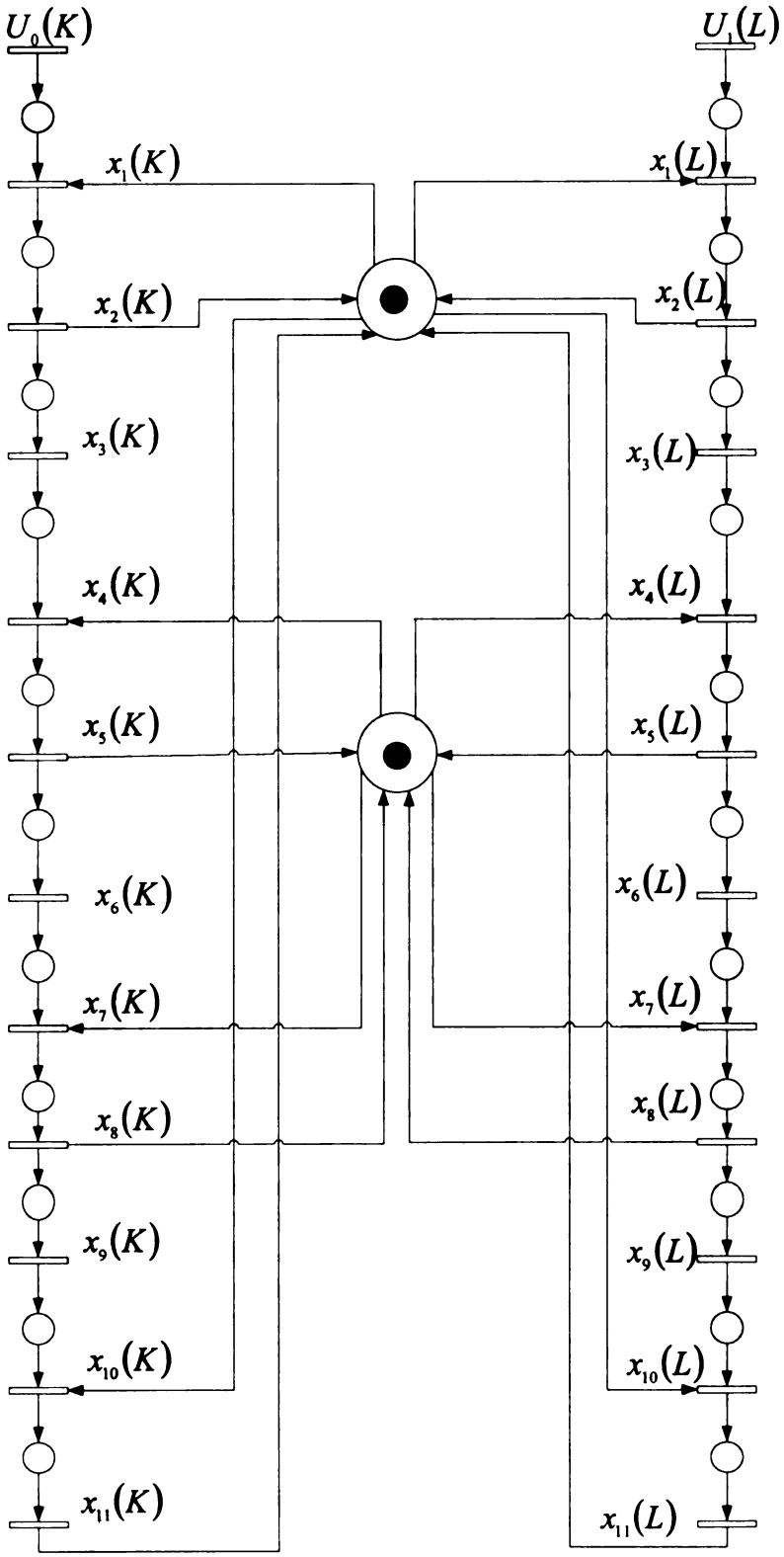


Figure 2.10 The TPN model for the full system

2.2 Max plus Algebra Model

2.2.1 Definitions

Definition 2.3 (semifield) *A semifield K is a set endowed with two operations* $\oplus$ *and* $\otimes$ *such that:*

- *The operation* $\oplus$ *is associating, commutative and has a zero element* ε
- *The operation* $\otimes$ *defines a group on* $K_{\bullet} = K \setminus \{\varepsilon\}$, *it is distributive with respect to* $\oplus$ *and its identity element e satisfies* $\varepsilon \otimes e = e \otimes \varepsilon = \varepsilon$.

We say that the semifield is

- *Idempotent if the first operation is idempotent, that is, if* $a \oplus a = a$, $\forall a \in K$;
- *Commutative if the group is commutative.*

Definition 2.4 (the algebra structure $\Re_{max}$**)** *The symbol* $\Re_{max}$ *denotes the set* $\Re \cup \{-\infty\}$ *with max and + as the two binary operations* $\oplus$ *and* $\otimes$, *respectively.*

We call this structure the max plus algebra.

Please note that, if we choose a, $b \in \Re$,

$$a \otimes b = a + b$$

and

$$a \oplus b = \max(a,b) = \begin{cases} a & a \geq b \\ b & a < b \end{cases}$$

In addition, we can define that

$$e = 0 \text{ and } \varepsilon = -\infty.$$

Theorem 2.1 The zero element ε of an idempotent semifield is absorbing for the second operation, that is $\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon$, $\forall a \in K$.

Proof We have that

$$\varepsilon = \varepsilon e = \varepsilon(\varepsilon \oplus e) = \varepsilon^2 \oplus \varepsilon = \varepsilon^2$$

and then,

$$\forall a \in K_*, \quad \varepsilon = \varepsilon e = \varepsilon a^{-1} a = \varepsilon(a^{-1} \oplus \varepsilon) a = \varepsilon a^{-1} a \oplus \varepsilon^2 a = \varepsilon^2 a = \varepsilon a$$

2.2.2 Max-Plus Algebra Model

I. Max-plus Algebra Model for the Block Process

From the TPN model, $X_1(K+1)$ is the time of robot1 picking up the K+1th block at the location 1. If robot1 needs to pick up the K+1th block, robot1 and the K+1th block should be available. We denote the $R_{11}(K+1)$, $R_{12}(K+1)$ are robot 1 's first and the second ready time in the K+1th block process. The $R_{21}(K+1)$ and $R_{22}(K+1)$ is for robot 2.

So:

$$X_1(K+1) = R_{11}(K+1) \oplus U_0(K)$$

From the timing diagram (Figure 2.11) for the block process:

$$X_2(K+1) = X_1(K+1) \otimes a_{11}$$

$$R_{12}(K+1) = X_2(K+1)$$

$$X_3(K+1) = X_2(K+1) \otimes a_{12}$$

Just as for the $X_1(K+1)$, if robot 2 want to pick up the K+1th block, that should be wait for robot 2 and the K+1th block to be available.

$$X_4(K+1) = X_3(K+1) \oplus R_{21}(K+1)$$

And then:

$$X_5(K+1) = X_4(K+1) \otimes a_{13}$$

$$R_{22}(K+1) = X_6(K+1)$$

$$X_6(K+1) = X_5(K+1) \otimes a_{14}$$

Just like $X_4(K+1)$:

$$X_7(K+1) = X_6(K+1) \oplus R_{22}(K+1)$$

$$X_8(K+1) = X_7(K+1) \otimes a_{15}$$

$$R_{21}(K+2) = X_9(K+1)$$

$$X_9(K+1) = X_8(K+1) \otimes a_{21}$$

Same as the $X_1(K+1)$:

$$X_{10}(K+1) = X_9(K+1) \oplus R_{12}(K+1)$$

$$X_{11}(K+1) = X_{10}(K+1) \otimes a_{17}$$

$$R_{11}(K+2) = X_{11}(K+1)$$

We need rewrite the above equations into matrix representation as

$$X(K+1) = A_0 \otimes X(K+1) \oplus A_1 \otimes R(K+1) \oplus \overline{B} \otimes U_0(K+1)$$

$$X(K+1)=\begin{bmatrix} X_1(K+1) \\ \cdot \\ \cdot \\ \cdot \\ X_{11}(K+1) \end{bmatrix} \qquad \overline{B}=\begin{bmatrix} e \\ \varepsilon \\ \cdot \\ \cdot \\ \varepsilon \end{bmatrix} \qquad R(K+1)=\begin{bmatrix} R_{11}(K+1) \\ \varepsilon \\ \varepsilon \\ R_{21}(K+1) \\ \varepsilon \\ \varepsilon \\ R_{22}(K+1) \\ \varepsilon \\ \varepsilon \\ R_{12}(K+1) \\ \varepsilon \end{bmatrix}$$

$$A_0=\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ a_{11} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & a_{12} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & a_{13} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & a_{14} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & a_{15} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & a_{21} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & a_{17} & \varepsilon \end{bmatrix}$$

$$A_1=\begin{bmatrix} e_{(1,1)} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e_{(4,4)} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e_{(7,7)} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e_{(10,10)} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

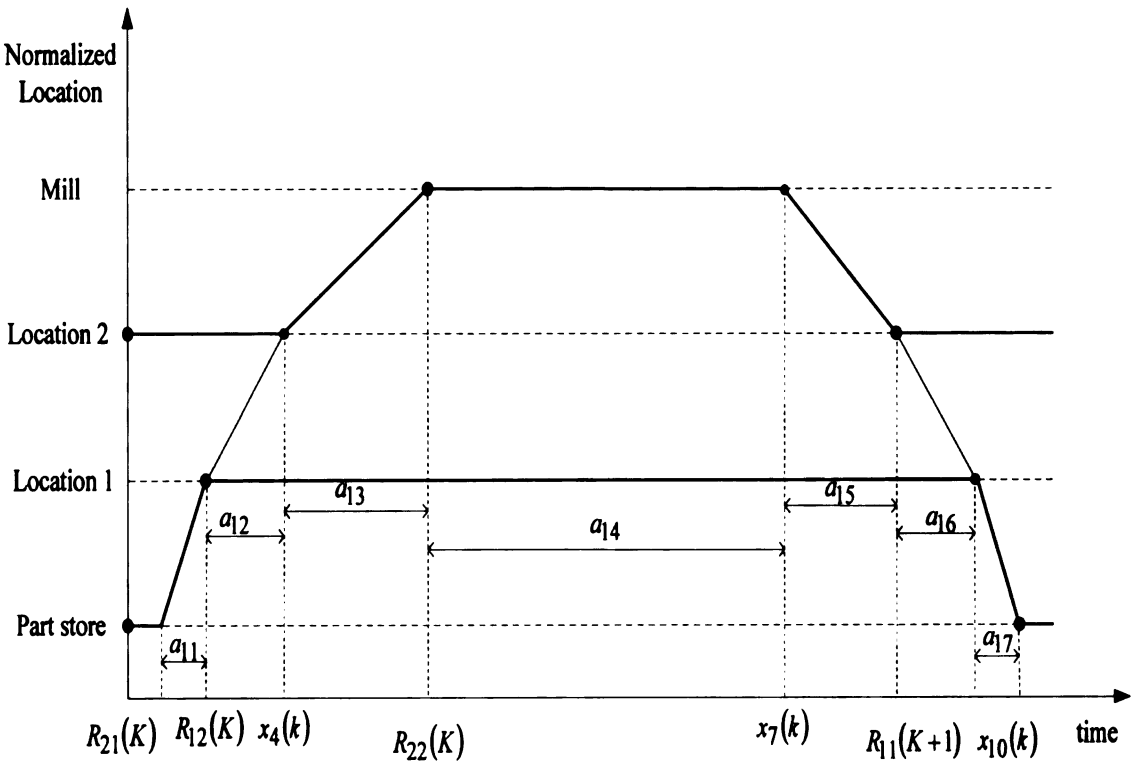


Figure 2.11 Timing Diagram for the block process

II. Max-plus Algebra Model for The Cylinder Process

The timed equations:

$$X_1(L+1) = R_{11}(L+1) \oplus U_0(L+1)$$

$$X_2(L+1) = X_1(L+1) \otimes a_{11}$$

$$X_3(L+1) = X_2(L+1) \otimes a_{12}$$

$$X_4(L+1) = X_3(L+1) \oplus R_{21}(L+1)$$

$$X_5(L+1) = X_4(L+1) \otimes a_{13}$$

$$X_6(L+1) = X_5(L+1) \otimes a_{18}$$

$$X_7(L+1) = X_6(L+1) \oplus R_{22}(L+1)$$

$$X_8(L+1) = X_7(L+1) \otimes a_{15}$$

$$X_9(L+1) = X_8(L+1) \otimes a_{21}$$

$$X_{10}(L+1) = X_9(L+1) \oplus R_{12}(L+1)$$

$$X_{11}(L+1) = X_{10}(L+1) \otimes a_{17}$$

The matrix representation:

$$X(L+1) = A_0 \otimes X(L+1) \oplus A_1 \otimes R(L+1) \oplus \overline{B} \otimes U_0(L+1)$$

$$X(L+1) = \begin{bmatrix} X_1(L+1) \\ \cdot \\ \cdot \\ \cdot \\ X_{11}(L+1) \end{bmatrix} \qquad R(L+1) = \begin{bmatrix} R_{11}(L+1) \\ \varepsilon \\ \varepsilon \\ R_{21}(L+1) \\ \varepsilon \\ \varepsilon \\ R_{22}(L+1) \\ \varepsilon \\ \varepsilon \\ R_{12}(L+1) \\ \varepsilon \end{bmatrix}$$

These are the Max-plus algebra models for the full systems. Based on these mathematical models, we can optimize them to schedule and plan the tasks of this work-cell.

CHAPTER 3

Scheduling

3.1 Problem Formulation

In the real system the robots are used by both the block and cylinder processes, so maybe the same robot will be shared by two different actions. This thing is called conflict. The robot just can do one action at a time, so when the robot meets a conflict, it must be choose one. How to choose is a big problem.

3.2 Optimize the Models

The Max-plus algebra model and the TPN model have already been built. In order to let the robots to choose one action when they meet a conflict, we need plan and schedule the tasks. For that we need optimize the TPN model and Max-plus algebra model. We add additional controls to the TPN model; based on the new TPN model the new Max-plus model will be built. It will control the additional controls available time to schedule the tasks.

3.2.1 Optimize the TPN Model

Additional controls are added to the TPN model to plan the robot action sequence (Figure 3.1). Now we add feedback from $X_{11}(K)$ to $U_0(K+1)$, and $X_{11}(L)$ to $U_0(L+1)$ in the TPN model. We can find the $U_0(K+1)$ firing time equal to T_{11} firing time $X_{11}(K)$, so we get the equation $U_0(K+1)=X_{11}(K)$. For the same reason, we get the equation

$U_0(L+1) = X_{11}(L)$. From the firing rule of Petri Nets and the operation rule of Max-plus algebra, the transition $X_1(K)$ firing time $X_1(K) = U_0(K) \oplus R_{11}(K) \oplus TU_{11B}(K)$. The $TU_{11B}(K)$ is U_{11B} the available time in the *Kth* block process. If $TU_{11B}(K) \geq U_0(K) \oplus R_{11}(K)$, $X_1(K) = TU_{11B}(K)$. We can schedule the time when $U_{11B}(K)$ will be available, to control whether to fire T_1. For the real system, we can control whether robot 1 picks up the *Kth* block when the *Kth* block and the robot1 are available.

Based on this idea, we add many additional controls $U_{11B}, U_{12B}, U_{21B}, U_{22B}, U_{11C}, U_{12C}$, U_{21C}, U_{22C} to the TPN model. The additional controls is U , and for the *Kth* block process the $U(K) = [U_{11B}(K) \;\; U_{12B}(K) \;\; U_{21B}(K) \;\; U_{22B}(K)]^T$; for the *Lth* cylinder process. It is $U(L) = [U_{11C}(L) \;\; U_{12C}(L) \;\; U_{21C}(L) \;\; U_{22C}(L)]^T$. If one of them is available we denote that as 1, if unavailable we denote that as 0. These controls can control the time when the robots will be activated. It will activate different controls at the same or the same controls at the different times, to schedule the tasks.

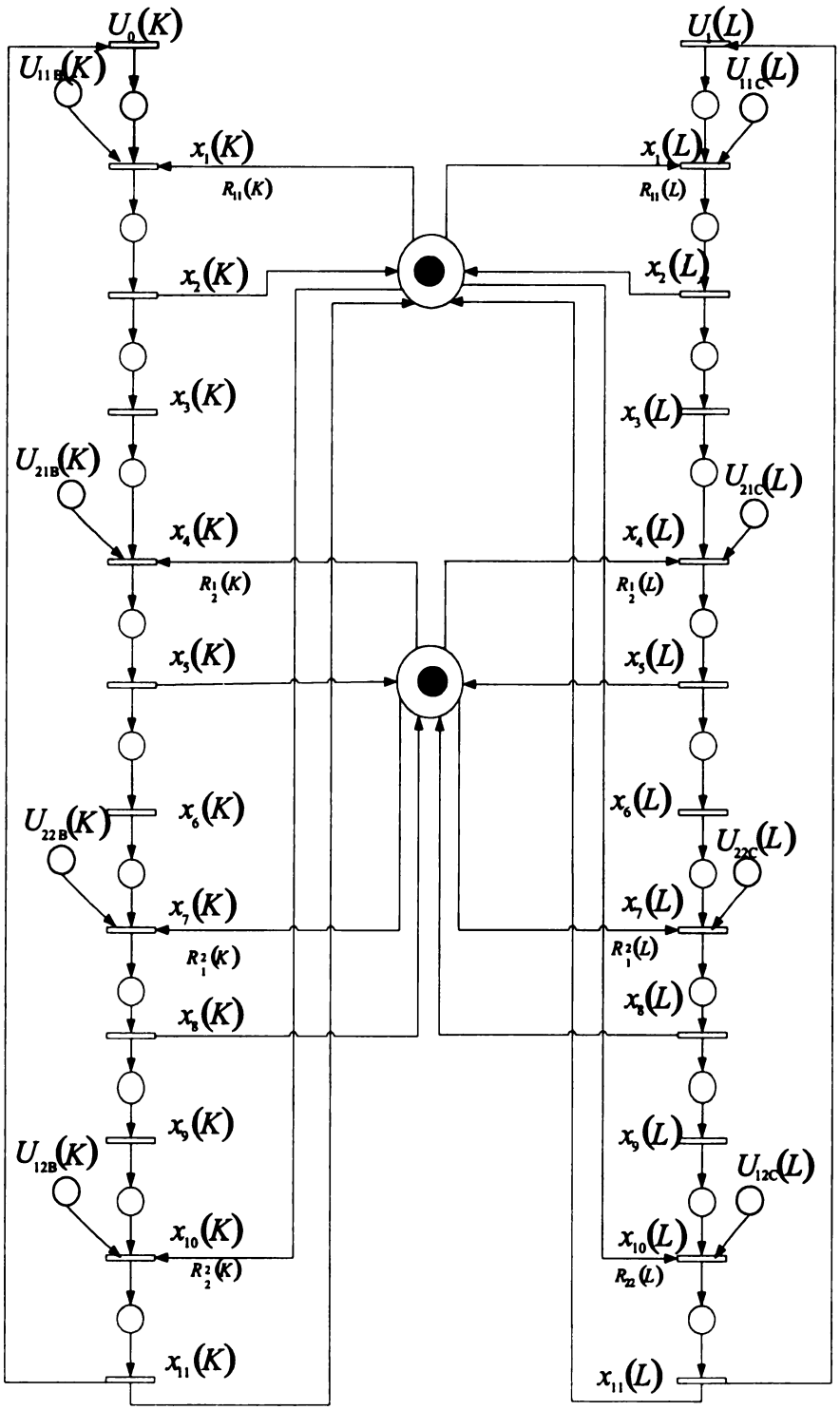


Figure 3.1 The TPN model with the additional controls

For example, we have a tasks sequence like the figure 3.2.

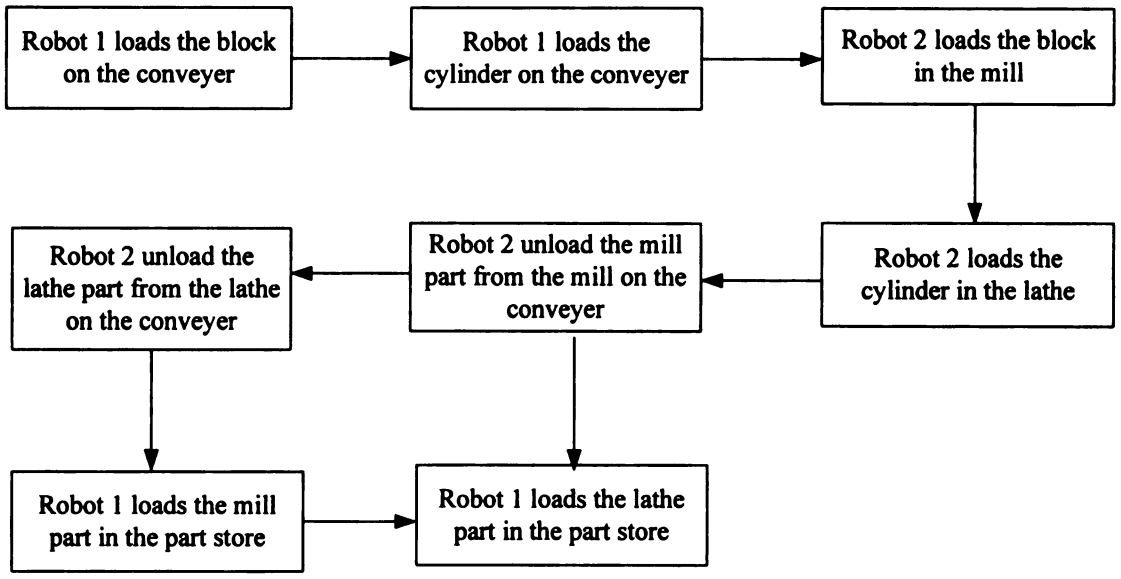


Figure 3.2 The task sequence of the example

To achieve this schedule, we active the additional controls as below:

When the *Kth* block, *Lth* cylinder and robot1 are available

$$U(K)=[1\ \ 0\ \ 0\ \ 0]\quad U(L)=[0\ \ 0\ \ 0\ \ 0]$$

↓ Robot 1 loads the *kth* block on the conveyer

When the robot 1 is available

$$U(K)=[0\ \ 0\ \ 0\ \ 0]\quad U(L)=[1\ \ 0\ \ 0\ \ 0]$$

↓ Robot 1 picks the *Lth* cylinder on the conveyer

When the *kth* block is arriving on the location 2 and the robot 2 is available

$$U(K)=[0\ \ 0\ \ 1\ \ 0]\quad U(L)=[0\ \ 0\ \ 0\ \ 0]$$

↓ Robot 2 loads the *Kth* block into the mill

When the *Lth* cylinder is arriving on the location 2 and the robot 2 is available

$U(K)=[0\ \ 0\ \ 0\ \ 0]\ U(L)=[0\ \ 0\ \ 1\ \ 0]$

↓ Robot 2 loads the *Lth* cylinder into the lathe

When the *Kth* mill part is finished and the robot 2 is available

$U(K)=[0\ \ 0\ \ 0\ \ 1]\ U(L)=[0\ \ 0\ \ 0\ \ 0]$

↓ Robot 2 unload the *Kth* mill part on the conveyer

When the *Lth* Lathe part is finished and the robot 2 is available

$U(K)=[0\ \ 0\ \ 0\ \ 0]\ U(L)=[0\ \ 0\ \ 0\ \ 1]$

↓ Robot 2 unload the *Lth* lathe part on the conveyer

When the *Kth* mill part is arriving on the location 1 and robot 1 is available

$U(K)=[0\ \ 1\ \ 0\ \ 0]\ U(L)=[0\ \ 0\ \ 0\ \ 0]$

↓ Robot 1 loads the *Kth* mill part in the part store

When the *Lth* lathe part is arriving on the location 1 and robot 1 is available

$U(K)=[0\ \ 0\ \ 0\ \ 0]\ U(L)=[0\ \ 1\ \ 0\ \ 0]$

↓ Robot 1 loads the *Lth* lathe part in the part store

Start the K+1th block and the L+1th cylinder processes

From that we can know that the additional controls can control the task sequence.

3.3 Scheduling and Planning

Now we need to define a rule to control the additional controllers U_A, U_B, U_C and U_D (figure 3.3). These controllers can control the place PAB, PAC, PBB, PBC, PCB, PCC, PDB and PDC when they are available. In the Max-plus algebra model, if one of them is available we denote that as 1, if none is available we denote that as 0. In the TPN model, if one of them is available, a token is added in the place. These controllers can control the additional controls when will be active to fire the tasks. The target of scheduling is to achieve the high efficiency.

A FMS can process n kind of parts during same time. Each process time is T1, T2.....Tn-1, Tn.

The total process time $T = T_1 \oplus T_2 ..... \oplus T_{n-1} \oplus T_n$

Assuming T_n is maximum, then $T = T_n$.

During the full process

$$T_n = T_{fn} \otimes T_{rn} \otimes T_{in}$$

Where

T_{fn} is the nth part time used and T_{rn} is the remaining time of the nth part if the part doesn't have any other time delay. If the nth part needs to wait for a sharing resource available or to wait for fixing of a fault, the T_n needs to be incremented by this idle time.

The T_{in} is this time.

The objective function of the scheduling is $T \rightarrow \min$.

Since $T = T_n$

So $T_{\min} = T_{n(\min)}$

We know $T_n = T_{fn} \otimes T_{rn} \otimes T_{in}$, and T_{in} is the idle time for each part, if the part does not need to wait for the sharing resource to be come available, it equals to zero.

So $T_{n(\min)} = T_{fn} \otimes T_{rn}$

For the real process, if the nth kind of part has the highest efficiency, they do not need to wait for the sharing resource available. They only need process the part step by step, no any idle time between any two steps.

In order to achieve the high efficiency, we setup a different priority for each kind of part. If the $T_{rn} > T_{r(n-1)}$, the nth kind of part has higher priority than the (n-1)th kind of part. When the parts meet conflict, the sharing resource must process the part which has the highest priority. This is the firing rule.

Obviously, how long the work cell needs to process the products depends on the amount of the products and the process time for each kind of product. Whether the efficiency is high or low depends on whether the process time is short or long. For this

work cell, if it processes two kinds of parts (mill part and lathe part), the total process time is determined by the part whose process time is the longest. Assume that the robot resource is enough, so R_K and R_L are ignored.

The new Max-plus model is:

$$X(L) = A_0 \otimes X(L) \oplus A_1 \oplus \overline{B} \otimes U_0(L)$$

And

$$X(K) = A_0 \otimes X(K) \oplus A_1 \oplus \overline{B} \otimes U_0(K)$$

Since for this work-cell the controller firing time is the only one unknown time, the process time for each part can be generated by the new Max-Plus algebra model. By the Max-Plus algebra model, the total process time, $T1$, and how long the process has already taken $T2$ can be known. Now define the remainder process time $T = T1 - T2$. TB is mill part remainder process time and TC is for lathe part.

Based on the firing rule for the U_A transfer equation $f(x)_1$:

$$f(x)_1 = \begin{cases} \begin{cases} U_{11}(K) = U_O(K) \\ U_{11}(L) = U_1(L) \otimes T \end{cases} & \text{if} \quad \begin{cases} U_0(K) = U_1(L) \\ T_B \geq T_C \end{cases} \\ \begin{cases} U_{11}(K) = U_O(K) \otimes T \\ U_{11}(L) = U_1(L) \end{cases} & \text{if} \quad \begin{cases} U_0(K) = U_1(L) \\ T_B \leq T_C \end{cases} \\ \begin{cases} U_{11}(K) = U_O(K) \\ U_{11}(L) = U_1(L) \end{cases} & \text{if} \quad U_0(K) \neq U_1(L) \end{cases}$$

Based on the same idea:

$$f(x)_2 = \begin{cases} \begin{cases} U_{21}(K) = X_3(K) \\ U_{21}(L) = X_3(L) \otimes T \end{cases} & \text{if} \quad \begin{cases} X_3(K) = X_3(L) \\ T_B \geq T_C \end{cases} \\ \begin{cases} U_{21}(K) = X_3(K) \otimes T \\ U_{21}(L) = X_3(L) \end{cases} & \text{if} \quad \begin{cases} X_3(K) = X_3(L) \\ T_B \leq T_C \end{cases} \\ \begin{cases} U_{21}(K) = X_3(K) \\ U_{21}(L) = X_3(L) \end{cases} & \text{if} \quad X_3(K) \neq X_3(L) \end{cases}$$

$$f(x)_3 = \begin{cases} \begin{cases} U_{22}(K) = X_6(K) \\ U_{22}(L) = X_6(L) \otimes T \end{cases} & \text{if} \quad \begin{cases} X_6(K) = X_6(L) \\ T_B \geq T_C \end{cases} \\ \begin{cases} U_{22}(K) = X_6(K) \otimes T \\ U_{22}(L) = X_6(L) \end{cases} & \text{if} \quad \begin{cases} X_6(K) = X_6(L) \\ T_B \leq T_C \end{cases} \\ \begin{cases} U_{22}(K) = X_6(K) \\ U_{22}(L) = X_6(L) \end{cases} & \text{if} \quad X_6(K) \neq X_6(L) \end{cases}$$

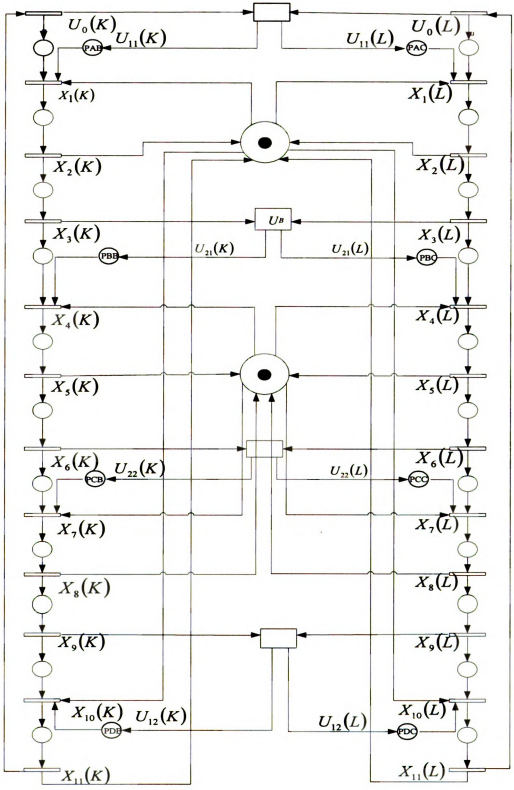


Figure 3.3 The TPN model with the additional controllers

CHAPTER 4

Experiment and Data Analysis

4.1 The Experiment

4.1.1 The Equipment of the Experiment

The system for this research is installed at Michigan State University. It was purchased in early 1996 from Eshed Robotec, Israel. The system itself consists of two robots, where one's base is movable on a slidebase, one conveyor, and one vision control system for the quality check, and finally of two third-party CNC-machines. Every device is controlled by a separate PC and the whole system is run by a CIM-manager, which runs on the supervisor PC. The PCs are connected via a local area network, using Microsoft Windows NT 4 as operating system.

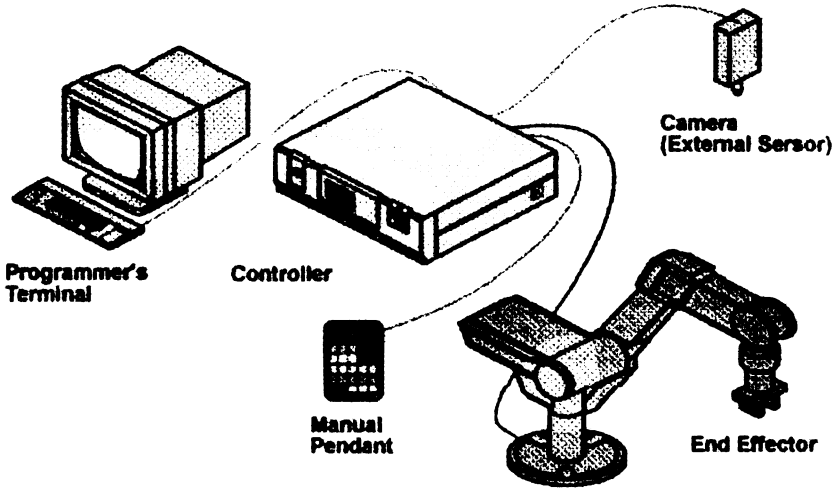


Figure 4.1 Components of the robot system

Figure 1.2 shows the robot system. The robot system includes the manipulator arm, the end effector (the gripper or tool mounted on the end of the arm), the robot controller and a computer for programming the robot. Another helpful device is the hand-held control pendant, which is used for manual operation of the robot and recording of positions of the robot.

The robot systems contain internal feedback devices, (such as encoders) and also include external sensing devices (such as a vision system). The controller is a stand-alone, real-time, multi-tasking controller. It allows simultaneous and independent operation of several programs, grouping of axes for multiple device control, and program editing while others are running.

The robots used for this system, SCORBOT-ER V PLUS are continuous path robots designed for training, research and laboratory applications. The maximum weight to be carried with the gripper is 1 kg.

This open, vertically articulated 5-axis robot (Figure 4) is controlled by an internal controller. The robot is equipped with internal force and torque sensors, which provide data on the position or motion of the arm joints and protect it against mechanical damage if it hits an unexpected obstacle. Without these sensors the controller cannot position the arm accurately. Achievable speed, accuracy and repeatability are good for a robot driven by small DC motors transmitting the movement over belts.

The effector at the end of the manipulator is used to hold the tool, moving in response to signals sent to the actuator. The mechanical arm's function is to move the end effector to different positions in space and change its orientation (the direction it faces). The arm

must be strong enough to carry workpieces, but also flexible and precise enough to accurately position the end effector.

A robot must be properly equipped for the kind of task it has to perform. The most common end effector is a gripper for grasping and picking up objects. The characteristics of the actuator and effector, combined with the load, determine the system performance. The nature of the task to be performed by the equipment will determine the preferred design for the actuator. As for this project only two different raw materials are considered, a simple gripper is used, as this is the best compromise.

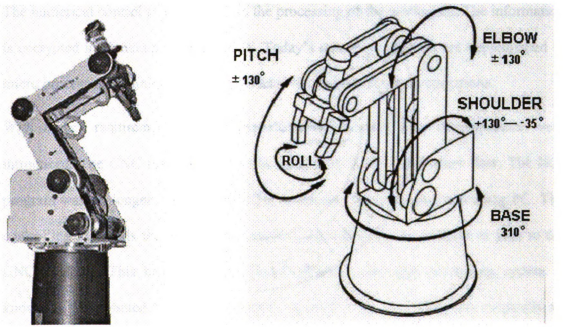


Figure 4.2 Scorbot ER Vplus

The manufacturing environment has changed dramatically in the last few years. The manufacturing environment has evolved from manual operation, where workers operated individual machines, to semiautomatic operation, where the machines were able to perform a few steps in automatic sequence, to a high degree of automation making

extensive use of computers and other automated equipment. It goes sometimes as far that no workers, beside for maintenance, are needed.

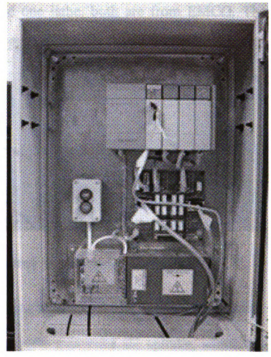
Major factors in this change are computer numerically controlled (CNC) machines. Computer numerical control is the logical extension of numerical control (NC). Numerical control is the control of a machine tool by means of a program. The Electronic Industries Association (EIA) defines NC equipment as a system in which actions are controlled by the direct insertion of numerical data at some point. The system must automatically interpret at least some portion of this data.

The numerical control program defines the processing of the workpiece. The information is encrypted in alphanumerical symbols. Today's numerical controls are implemented in micro processing technique, as they also have to calculate complex operations.

With higher requirements in the manufacturing industry new developments were introduced. The CNC machines had to be integrated in the information flow. The NC-program was no longer stored on the CNC machines, but on some controlling PC. The controlling PC sends the needed NC-program, dependent on the production plan to the CNC machine. This hierarchical structured program- and data processing system is known as "Distributed Numerical Control", as DNC. The term DNC was originally an acronym for "Direct Numerical Control" and used to describe systems in which a number of CNC machines were connected with a computer, which sent programs to the machines via serial communications using an interface connected to their controller. With the usage of DNC-systems the time to change a NC-program was reduced dramatically. Today DNC are controlled from a centralized shop level system. A communication network

linking the shop-level control system and the different cells is crucial with this type of control. Based on this a high level physical and organizational diverseness was achieved.

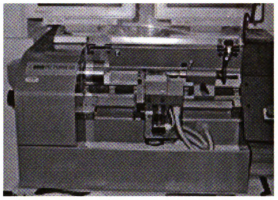
Many of the tasks a CNC machine has to fulfill are controlled by a PLC. They were initial introduced in the 70s, and now used in nearby all industrial application and are the main part of the automation technique. With the better performance of the available hardware, PLC's are used in more and more complex control tasks. The main task of a PLC in CNC machines is the monitoring and controlling of the mechanical operating units. This includes the logical connections and bolting functions as well as time and plausibility monitoring of the single product unit. Furthermore some of the functions of the operating system like turning on and off the motor and the coolant system are controlled by the PLC. In the CNC machine a PLC has to interact with the numerical control of the machine. While the numerical control extracts and processes the geometrical information of the used NC-program to move the axis (coordinates and moving speed), the switching information (e.g. changing the tool) are forwarded to the PLC. Other signals between the CNC-machine and PLC are used for the synchronization of both systems and the exchange of status messages.



The syntax of an NC-program is defined in DIN 66025. Sometimes NC-programs are called "G-code". It is structured in steps. Every step defines an order for the machine.

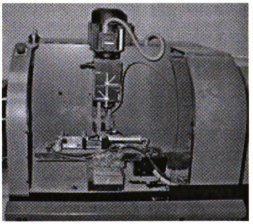
The orders are encoded with a combination of alphabetical symbols and numbers. The NC-program starts with "%". Each step begins with an "N", and ends with the "line feed" (LF) command.

The system used is equipped with one mill and one lathe, both are from EMCO. The software used is WINNC, Version 1.4. Both CNC machines have no internal controller, as a dedicated PC on which the G-code is processed controls them. Each CNC machine is connected with the controlling PC using an RS-486 interface, which has more possibilities and a higher speed than the normal serial interface, using an RS-232.



The mill is a PCMILL 50, which is a 3-axis milling machine. It has helical interpolation and a reversible spindle. It is equipped with a spindle motor for the spindle, which has 0.45 kW at 60 Hz. The milling head has a clamping device for the milling spindle and is mounted on a slide, which enables it to move in z-coordinates.

Small stepper motors are moving the milling table in x and y directions. An automated pneumatic device to hold the part, as well as a pneumatic moveable door is installed and can be accessed from the controlling PC.



The lathe is a PCTURN 50, which is a 2-

axis slant bed lathe, having a tailstock and a collision control. It is further equipped with a 3-jaw pneumatic chuck for the spindle. In the tool system three different turning tools can be clamped in. The automation devices are the same as on the mill.

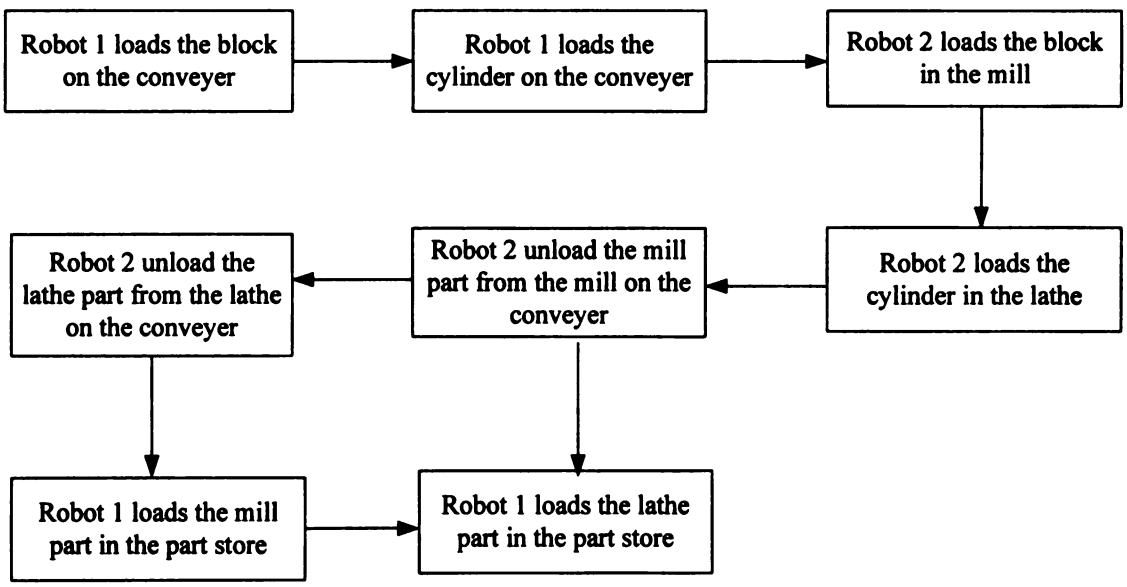
4.1.2 The Experiments

In these experiments we use this work cell to process two kind of part: one is a mill part and the other is a lathe part. The raw materials are blocks and cylinders. The process time of each block is 5 minutes and 45 seconds. The process time of each cylinder is 10 minutes and 45 seconds.

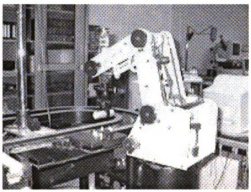
Experiment 1: Set up a schedule for the work cell

In this experiment we input the different $U(K)$ and $U(L)$ at the various time, to find the schedule which we want.

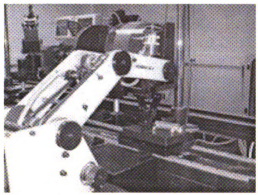
We have a schedule as blow.



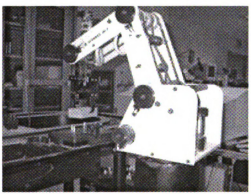
We find the experiment rouses as blow



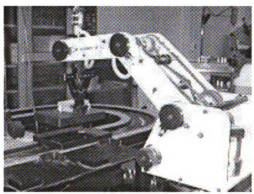
The Robot 1 picks up the cylinder



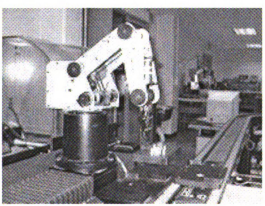
The Robot 1 loads the cylinder on the conveyer



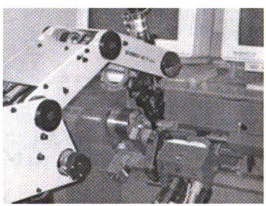
The Robot 1 picks up the block



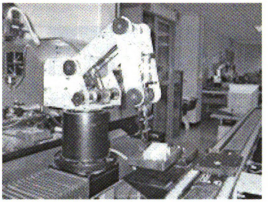
The Robot 1 loads the block on the conveyer



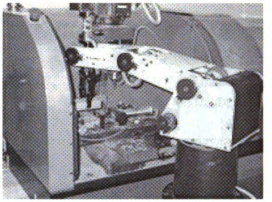
The Robot 2 picks up the cylinder



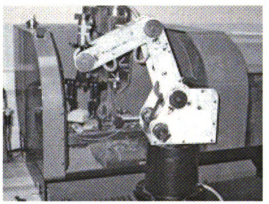
The Robot 2 loads the cylinder into the lathe



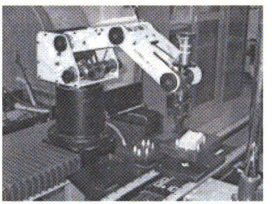
The Robot 2 picks up the block



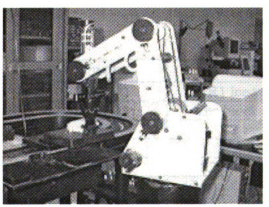
The Robot 2 loads the block into the mill



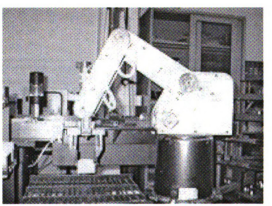
When the mill part is done, the Robot 2 unloads it from the mill



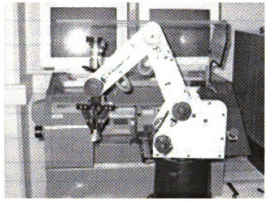
Robot 2 loads the mill part on the conveyer



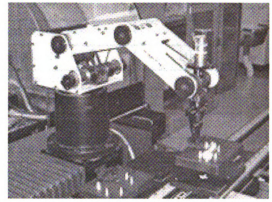
Robot 1 unloads the mill part form the conveyer



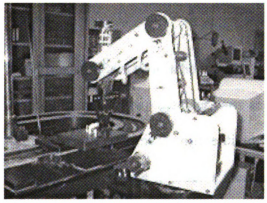
Robot 1 loads the mill part into the part store



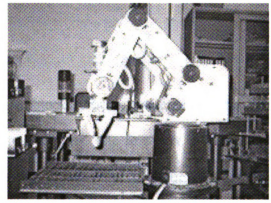
When the lathe part is done, the Robot 2 unloads it from the lathe



Robot 2 loads the lathe part on the conveyer



Robot 1 unloads the lathe part form the conveyer



Robot 1 loads the lathe part into the part store

option	The task of the option	operation time
W1(k)	R1 loads block on conveyor	30"
W2(k)	The block transfer to Location 2	10"
W3(k)	R2 loads block into mill	30"
W4(k)	Mill process the mill part	3'25"
W5(k)	R2 loads mill part on conveyor	30"
W6(k)	The mill part arriving on location 1	10"
W7(k)	R1 load the mill part into part store	30"

Table 4.1 The operation time of block process

option	The task of the option	operation time
W1(L)	R1 loads cylinder on conveyor	30"
W2(L)	The cylinder transfer to Location 2	10"
W3(L)	R2 loads cylinder into lathe	40"
W4(L)	lathe process the lathe part	8'15"
W5(L)	R2 loads lathe part on conveyor	30"
W6(L)	The lathe part arriving on location 1	10"
W7(L)	R1 load the lathe part into part store	30"

Table 4.2 The operation time of cylinder process

Experiment 2 Real time scheduling

In this experiment, two cylinder parts and three block parts are processed by this Work-Cell, and this experiment will be divided into two parts. The first part is following the experiment 1 schedule to process these parts. In the second part, a fault is setup during processing. Then the fault is fixed and the process continues the processing until the amount of the parts are accomplished.

	Total process time
Part 1	22 minutes and10 seconds
Part 2	22 minutes and 30 seconds

Table 4.3 The result of the experiment 2

Experiment 3 Real time adapting

In this experiment, still as in the experiment 2, two cylinder parts and three block parts are processed by this Work-Cell. This experiment includes two parts. In both experiments,

the same conflict time instant is set up, and the results are compared. In order to design the conflict time instant, after the first block is loaded into the mill, we shut down the mill for 4 minutes 30 seconds. When let the first mill part and the first lathe part finish at the same time.

Since the remaining time of the cylinder T_C is bigger than the remaining time of the block T_B, the controller asks robot 2 to pick up the lathe part first and then picks up the mill part. But in the first part , we force robot 2 to pick up the mill part first. In the second part, we ask the work-cell to follow the controller to pick up the lathe part first. The total process time is shown in the Table 2. From the Table 2 the part 2 process time is shorter than of part 1, so the rule is effective, and using this can get higher efficiency.

	Total process time
Without optimal schedule	25 minutes and 40 seconds
With optimal schedule	19 minutes and 30 seconds

Table 4.4 The result of the experiment 3

4.2 The Data Analysis

Based on the results of the experiment 1, the additional control can control the task sequence. We can activate the different control at the same or different to achieve the task

sequence which we want. The experiment 1, we get the operation time for each task and total process time for each part.

In the experiment 2, we set up a fault during the processing. If part processing does not need the section which has a fault, this processing is not be affected by this fault. If a part processing is affected by this fault, the processing stops there until the fault is removed. From the result Table4.3, an un-necessary step will not affect its process. In the experiment 3, the real time adaptive mechanism was active. From the Table4.4, we can see the efficiency is very high.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

This thesis presents a new method for the scheduling of flexible manufacturing work-cell. The method is based on the Timed Petri net model and the Max-plus algebra model.

In the TPN model, the parallel schedule is built, and these two schedules are independent if there is no conflict. This parallel scheduling is better than the series-wound scheduling. The processing is not effected by any un- necessary step. The Max-Plus algebra model is used to analyze the full system, and design controls and firing rules to avoid the conflict.

As shown in the experiments, even when the system has a problem, under the parallel schedule, the work-cell still can run and the real time adapting rule can get high

efficiency. But this method has only be demonstrated on one kind of work-cell, so it may has some limits. In the future this method will be used on other work-cells.

5.2 Future Work

This method can let the collateral processing system achieve a high efficiency simply and easily. But we just use this method in a work-cell, not to get the general result for all FMS systems.

In the future, we will extend to other kinds of FMS, to prove this method validity. This method will be used in some kind of computer system possibly. We will prove this probability.

BIBLIOGRAPHY

[1] Mumin Song, Tzyh Jong Tarn, Ning Xi, "Integrated Hybrid System Approach for Planning and Control of Concurrent Tasks in Manufacturing Systems" ICRA 1998: 1192-1197

[2] E.Bowman, "The schedule-sequence problem," Operations Research, vol. 7, no. pp. 621-624, 1959.

[3] R. Conway, W. Maxwell, and L. Miller, Theory of Scheduling, Reading, MA: Addison-Wesley, 1967.

[4] A. Manne, "On the job-shop scheduling problem," Operation Research, vol. 8, no. pp. 219-233,1960.

[5] Doo Yong Lee and Frank DiCesare, "Scheduling Flexible Manufacturing Systems Using Petri Nets and Heuristic Search" IEEE Transaction on Robotics and Automaton, vol.10 no.2.1994.

[6] H. P. Hillion and J. Proth, "Performance Evaluation of Job-Shop Systems using Timed Event-Graph." IEEE Transactions on Automatic Control, vol.34 no. 1, pp. 3-9 Jan. 1989.

[7] M. Silva and R. Valette , "Petri Nets and Flexible Manufacturing," Advances in Petri Nets,Berlin: Springer-Verlag, 1989, pp.374-417.

[8] Zhou, M. C. and K.Venkatesh, "Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach" World Scientific, 1999.

[9] K. R. Baker, introduction to Sequence and Scheduling, New York: John Wiley & Sons, 1974.

[10] A.Di Febbraro, R. Minciardi, G.Parodi, A.Prati, M.Profumo, and S.Sacone, "Performance Optimization in Manufacturing Systems by use of Max-Plus Algebraic Techniques" 4/1994 IEEE 1995-2001

[11] Tadao Murate "Petri Nets: Properties, Analysis and Applications" Proceeding of the IEEE, vol. 77, No. 4, 4/1989

[12] S. Ahmad and B. Li, "Robot control computation in microprocessor systems with multiple arithmetic processors using a modified DF/HIS scheduling algorithm ," IEEE transactions on systems, man, and cybernetics, vol. 19, No. 5, pp. 1167-1178, 1989.

[13] E. Bowman, "The scheduling –sequencing problem," Operations Research, vol. 7, No. pp. 621-624, 1959

[14] S. French, Sequencing and scheduling: an introduction to the mathematics of the job-shop. Ellis Horwood, 1982

[15] E. Horowirz and S. Sahni, Fundamentals of computer algorithms, Rockville, MD: Computer Science Press, 1978

[16] J.Carlier and P. Chretienne, "Timed Petri net schedules," in advances in Petri nets, Rozenberg, Ed, Berlin: Spring-Verlag 1988, pp.62-84

[17] M. A. Holiday and M. K. Vemon, "A generalized timed Petri net model for performance analysis," proceeding of the IEEE international Workshop on Timed Petri Nets, Torino, Italy, July 1-3, 1985, pp. 181-190

[18] J. Pearl, Heuristics: Intelligent search strategies for computer problem solving. Reading, MA: Addison-Wesley, 1984

[19] A. Manne, "On the job-shop scheduling," Operation research quarterly, vol. 17, No. 2, pp. 161-171, June 1966.

[20] N. Nilsson, Principles of artificial intelligence, Palo Alto, CA: Tioga, 1980

[21] J. O'Brien, Scheduling handbook, New York: McGraw-Hill, 1969

[22] J. Rickel, "Issues in the design of scheduling systems," in expert systems an intelligent manufacturing , Oliff, Ed. Elsevier 1988, pp. 70-89

[23] Y. Zhang, "Solution to job-shop scheduling of FMS by neural networks," Proceedings of the 1991 IFAC Workshop on Discrete Event System Theory and Applications in Manufacturing and Social Phenomena, Shengyang, China, June 25-27, 1991, pp. 261-266