MODELLING AND ANALYSIS OF LASER TRANSFER MACHINE BY USING PETRI NETS

Yüksek Lisans Tezi

Selçuk ERGİNEL

ESKİŞEHİR 2019

MODELLING AND ANALYSIS OF LASER TRANSFER MACHINE BY USING PETRI NETS

Selçuk ERGİNEL

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği
Kontrol ve Otomasyon Anabilim Dalı
Doç. Dr. Hanife APAYDIN ÖZKAN

Eskişehir
Eskişehir Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Ekim 2019

FINAL APPROVAL FOR THESIS

This thesis titled "Modelling And Analysis Of Laser Transfer Machine By Using Petri Nets" has been prepared and submitted by Selçuk ERGİNEL in partial fullfillment of the requirements in "Eskişehir Techinal University Directive on Graduate Education and Examination" for the Degree of Master of Science in Electrical And Electronics Engineering Department has been examined and approved on …/…/20… .

Commitee Members		**Title, Name and Surname**	**Signature**
Member (Supervisor)	:	....	....
Member	:	....	....
Member	:	....	....
Member	:	....	....
Member	:	....	....

Director of Gradute School

Prof Dr. Murat TANIŞLI

ÖZET

LAZER TRANSFER MAKİNESİNİN PETRİ AĞLARI KULLANILARAK MODELLENMESİ VE ANALİZİ

Selçuk ERGİNEL

Kontrol ve Otomasyon Anabilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ekim 2019

Danışman: Doç. Dr. Hanife APAYDIN ÖZKAN

Bu çalışmada, kesikli olay sistemlerinin modellenmesi ve analizinde kullanılan Petri Ağları modelleme yaklaşımı ele alınmıştır. Bu kapsamda, cam ev eşyası sektöründe lazer markalama için kullanılan pozisyon kontrollü bir transfer makinesinin “Emergency Stop, Manual, Reference ve Automatic” prosedürleri Petri Ağları ile modellenmiştir. Petri Ağı modellerinin sınırlılık, güvenlilik, tıkanıklık ve geri dönülebilirlik analizleri tamamlanmıştır. Bu analizler için Platform Independent Petri Net Editor (PIPE) kullanılmıştır. Analiz sonuçlarına göre üretim sahasındaki makinede gerçekleştirilen prosedürün çalışmasının doğruluğu; varsa riskleri tespit edilmiştir.

ABSTRACT

MODELLING AND ANALYSIS OF LASER TRANSFER MACHINE BY USING PETRI NETS

Selçuk ERGİNEL

Department of Control and Automation

Eskişehir Technical University, Institute of Graduate Programs, October 2019

Supervisor : Assoc. Prof. Hanife APAYDIN ÖZKAN

In this study, Petri Nets modelling approach is handled which is used for modelling and analysis of discrete event systems. In this scope, "Emergency Stop, Manual, Reference and Automatic" procedures of a position controlled transfer machine which is used in glassware production sector for laser marking. "Boundedness, Safeness, Deadlock and Revesibility" analysis are completed. Platform Independent Petri Net Editor (PIPE) is used for these analysis. According to analysis results, accuracy and if exists risks of the operation of the procedure on the machine on site are identified.

TEŞEKKÜR

Bu çalışma uzun ve zahmetli bir öğrencilik döneminin ürünüdür. Uzun yıllar önce başlamış olduğum yüksek lisans macerası bu tez ile sona ermektedir. Derslerimi bitirdikten sonra hayatın bana sundukları ve çalışma hayatının getirdiği zorluklar ile sekteye uğryan ve en sona kalan bu çalışmanın bu tez ile nihayete ulaşacağını umuyorum.

Tezin yazımı esnasında değerli rehberliği ile beni yönlendiren danışman hocam Sayın Doç. Dr. Hanife Apaydın Özkan'a teşekkürlerimi sunarım. Bu çalışmanın şekillenmesindeki değerli yorumları ve katkılarını her zaman aklımda tutacağım.

Ayrıca yüksek lisans eğitimim boyunca beni hep destekleyen ve tavsiyeleri ile beni hep yönlendiren değerli hocam Sayın Prof. Dr. Altuğ İftar'a da teşekkürlerimi sunarım.

Nihayetinde beni her zaman, hangi şart altında olursa olsun hep seven, bağrına basan, gölgeleri altında yaşamaktan huzur duyduğum, bazı durumlarda yanımda olamasalar bile zaman ve mesafe tanımaksızın sevgilerini hep hissettiğim pek kıymetli Annem Fahire Erginel'e ve pek kıymetli Babam Yavuz Erginel'e şükranlarımı sunar, ellerinden öperim.

Selçuk ERGİNEL

..../ .... / 20....

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical legal consequences that are involved.

Selçuk ERGİNEL

CONTENTS

Page

TABLES

FIGURES

Page

IMAGES

1. INTRODUCTION

Day after day technology is getting involved in human's life step by step. Glassware production is being effected from the improvement of the technology. Many manufacturing and production machines and systems were established by using the new technologies especially in the area of servo control and automation softwares.

In a glassware production plant, the laser tranfer machine which is the main topic of this study was build in order to transfer the glassware products to under laser marking machine by using position controlled axis.

Due to the dynamic structure of production methods in glassware industry, analysis and tests of the systems are mostly ignored. Because the top administration wants to commission such kind of systems immediately. However without tests and analysis, there may be occured some kind of problems and the producers may not be sure of the consistency and stability of the machines or systems. This is the main motivation of preparing this study to make a close analysis and see the problems which may be occur on site.

The software structure of the laser transfer machine is very suitable to express in a Petri Net (PN) model. Its program basis was established on a cause and effect relationship which is formed by the discrete event structures. So using PN models gave what is needed for this study.

Until this study many other thesis or paper were published. Some similar related studies were examined before. One of them is PLC Application Of Railway Interlocking Systems With Petri Net Approach And Simulation. In this thesis, signalling and interlocking systems were modelled by using PNs (Behmeranrezai, 2019). The other thesis is Petri Net Based Control Of Robotic Assembly Systems. In this thesis, it is emphasized the importance of robotic assembly lines which improve rapidly and presented a model which can be used for controlling the robotic lines by using PN (İçmez, 2015).

In the contents of this thesis Discrete Event Systems (DES) will be given primarily. After the definition and the some features of DES will be explained, The Markov Chains will be given briefly as a DES model.

In Chapter 2 PNs will be presented as another and effective way of modelling and analysing tool for DES. After the definition, firing rules, structural and behavioral features of PNs will be given, the methods which are used for modelling in PNs will be explained.

The Chapter 3 will begin by giving the main parts of a glassware production plant. Then the location and the mission of the system which is the topic of this study will be mentioned. The reason of the necessity of this machine will be clarified.

The hardware and software features will be given. The components of the main control system and their purpose will be mentioned. The software structure of the system and how it works will be explained .

The programs which will be analysed will be presented. The missions of each of them and which tasks they belong will be explained. There will be an individual section for each program which consists of general information, flowchart of the program which explains how it works, the definitions of places and transitions, PNs model of the related program.

After giving the basics and the models of the programs, the analysis section will be given. There will be two analysing ways. The first section is related with the behavioral properties of the PNs model of the related programs. In this section analyse results will be given which are obtained from PIPE and also will be evaluated by means of the real system. The second analysis method is about the evaluting the cause and effect relations in some spesific situations during the run sessions within the scope of operator's demands.

In Chapter 4 the conclusion will be given for a general evaluation of this study. Analysis results are emphasized and some advices are given for studies for future.

2. PETRI NETS

2.1. Basic Definition of a System

It can be found three basic definitions in the literature describing what a system is.

An aggregation or assemblage of things so combined by nature or man as to form an integral or complex whole acording to Encyclopedia Americana (Cassandras and Lafortune, 2008).

A regularly interacting or interdependent group of items forming a unified whole according to Webster's Dictionary (Cassandras and Lafortune, 2008).

A combination of components that act together to perform a function not possible with any of the individual parts according to IEEE Standard Dictionary of Electrical and Electronic Terms (Cassandras and Lafortune, 2008).

There are two specific features about these definitions of a system. They are interacting components and a function intended to perform. A system does not only have to point out, it deals with a physical objects and natural laws. Economic models, human behaviors and population dynamic can also be defined as a system (Cassandras and Lafortune, 2008).

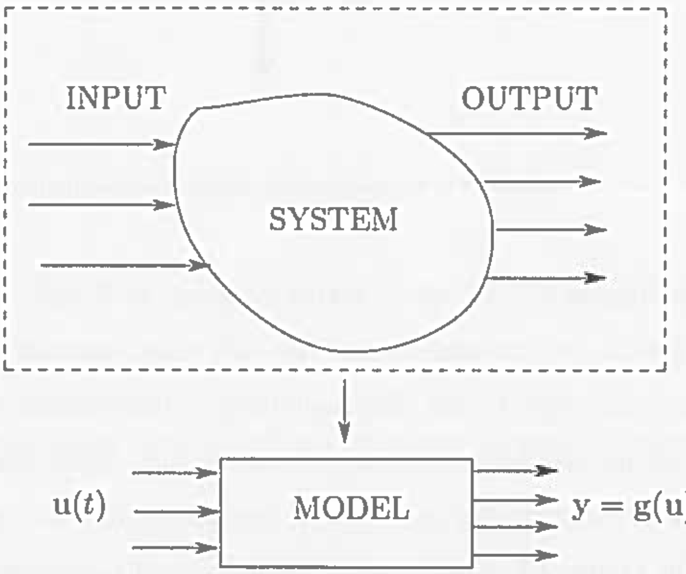

Figure 2.1. Basic depiction of a system

2.2. Basic Definition of Discrete Event Systems and Features

An event can be described as occuring instantaneously and causing transitions from one state to another. An event can be defined with some meanings such as physical action or behaviors of society. Or it may be a result of several conditions come real simultaneously. In a brief perspective an event might be both an action or a result. Random walk may be a good example understanding event concept and DES. Take a reference point in the origin of two dimensional coordinate system. The participant can only move randomly and only one step further from previous point, to one of four directions among North, South, East and West (Cassandras and Lafortune, 2008).

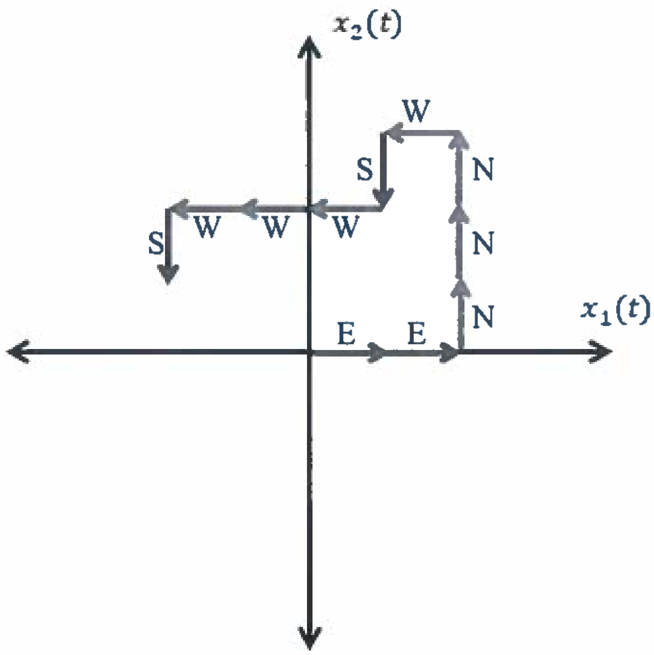

Figure 2.2. Example of two dimensional random walk

In the Figure 2.2. the set of events is shown as E ={N, S, W, E} when the state transitions are (x_1 , x_2) as state space. The only thing necessary to change the states is just the next event come to real. The participant of this random walk game is located on (0,0) point of the two dimensional coordinate system. When the next event appears its location changes step by step according to amount of the move and the direction. It can be seen in the Figure 2.3. step by step (Cassandras and Lafortune, 2008).

(0,0)	East
(1,0)	East
(2,0)	North
(2,1)	North
(2,2)	North
(2,3)	West
(1,3)	South
(1,2)	West
(0,2)	West
(-1,2)	West
(-2,2)	South
(-2,1)	

Figure 2.3. States changes according to preceeded events

Each event happen the position changes. After a sequence of events which is shown in the table above the position changes from (0,0) to (-2,1). Here, discrete set is described as $X = \{(i, j) : i, j = ..., -1, 0, 1, ...\}$. In this situation the discrete set is in the set of integers.

After random walk example the definition of discrete event systems can be done as follows;

A DES is an *event - driven system* that, its states change according to occurrence of asynchronous discrete events over time.

DES are composed of events and states. These transitions deal with the events. When any event occurs, the related state of the system changes in to its next situation from current (previous) one. But this happens asynchronously in time pattern. The total behavior of a DES is described assequences of occuring *events* and *states*.

DES based compunds can be seen in wide range scope vary from manufacturing systems to computer algorithms.

The main concept of this thesis rely on the modelling of a manufacturing machine in a glassware production base by PNs which is a tool for establishing and modelling of a DES. But before starting to metion about the PNs, there is another modelling method for DES. It is Markov Chains.

A Markov Chain is simply and basicly described as a stochastic model describing a sequence of possible events. The probability of each event depends only on the state attained in the previous event. But for a deeper understanding of Markov Chains, Markov Property must be given as a prior.

Briefly Markov Property refers to the memoryless property of a stochastic process. A stochastic process has the Markov property if the conditionla probability distribution of future states of the process depends only on the present state.

Under these circumstances if a process satisfies the Markov Property, one can make predictions for future state of related process by only taking care of its present situation, not its past.

A disrete time Markov Chain is a sequence of random variables with the Markov Property which the probability of moving to next step depends on the present state and not on the previous states.

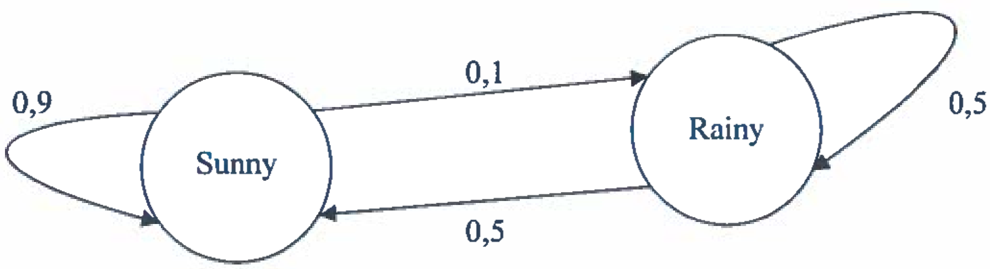

Figure 2.4. An example of weather forecasting by using Markov Chains

The best way to explain how Markov Chain works with an example. The Figure 2.4. shows a simple weather forecasting. According to this figure a sunny day is to be followed by 90% sunny day and by 10% rainy day. A rainy day is to be followed by 50% rainy day and 50% sunny day. In this perspective the transition matrix can be written as follows.

$$P = \begin{bmatrix} 0{,}9 & 0{,}1 \\ 0{,}5 & 0{,}5 \end{bmatrix}$$

According to today's certain information it is 100% sunny day. The problem is what the weather prediction is for 2 days later.

The initial matrix is written as follow $X^{(0)} = [1 \;\; 0]$ to describe current information.

To predict the situation for 2 day's later a stochastic row vector is written as follows.

$X^{(n+2)} = X^{(n+1)}P = (X^{(n)}P)P = X^{(n)}P^2$

In that situation $X^{(2)} = X^{(1)}P = (X^{(0)}P)P = X^{(0)}P^2$

$X^{(2)} = X^{(0)}P^2$

$X^{(2)} = [1\ 0] \begin{bmatrix} 0,9 & 0,1 \\ 0,5 & 0,5 \end{bmatrix} \begin{bmatrix} 0,9 & 0,1 \\ 0,5 & 0,5 \end{bmatrix}$

$X^{(2)} = [0,86\ 0,14]$

According to the matrix above which implies to show the prediction for 2 days later, the weather will be 86% sunny and 14% rainy according to only current information.

https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRXhhbXBsZXNfb2ZfTWFya292X2NoYWlucw (Access Date : 10.07.2019)

2.3. Petri Nets

PNs was first appeared by *Carl Adam Petri* in 1962 who worked as a scientist at the faculty of Mathematics and Physics of Technical University of Darmstadt (Raisch, 2009).

It is a very useful tool for modelling and analysing the discrete event systems. It main benefit is to allow modelling in graphical perspective so ease of access.

There are four types of components form the basics of PNs. They are places, transitions, arcs and weights.

$$PN = (P, T, A, w, M_0)$$

P : indicates the finite numbered set of *places.*

T : indicates the finite numbered set of *transitions.*

A : indicates the *arcs* are directed from places to transitions or transitions to places.

w : indicates the *weight of arcs.*

M_0 : indicates the *initial marking* of places (Bogdan, Lewis, Kovacic and Mireles, 2006).

The set of all input and output places for transitions are shown below

$$I(t_j) := \{p_i \in P \mid (p_i, t_j) \in A\}$$
$$O(t_j) := \{p_i \in P \mid (t_j, p_i) \in A\}$$

The set of all input and output transitions for places are shown below

$$I(p_i) := \{t_j \in T \mid (t_j, p_i) \in A\}$$
$$O(p_i) := \{t_j \in T \mid (p_i, t_j) \in A\}$$

Of course this equations are valid $p_i \in I(t_j)$ if and only if $t_j \in O(p_i)$ and $t_j \in I(p_i)$ if and only if $p_i \in O(t_j)$.

In the PNs representation places, transitions, arcs and weights are shown in Figure 2.5.(Raisch, 2009).

Figure 2.5. (a) Place (b) Transition (c) Arc (d) Weighted arc

In an ordinary PN, the weights of arcs equal to 1. Otherwise it is represented as in Figure 2.5. (d) with amount of its weight.

There is only one rule which is very important for further understanding of PNs. It is firing rule.

A PN can be described briefly as the combination of binding places and transition by arcs with weights. The places are representation of conditions or status, the transitions are representation of events. A transition has a certain number of inputs and outputs under the representation of places. These input and output places are the pre-conditions and consequences of transitions (Bogdan, Lewis, Kovacic and Mireles, 2006).

A particular property that differentiates a PN from an ordinary graph. It is marking shown as m which assingns a non-negaitve integer to each PN place. A marking is shown as a black dot which is defined as token. Representation of token inside a place means the related place has a marking.

In a PN whole whole places may have some amount of markings. This is shown with the marking vector of related PN.

$$M = [m(p_1), m(p_2), \ldots ,m(p_n)]$$

Another important property of a PN assocates with the capacity of a place ($K(p_i)$). The capacity of a place refers to the number of tokens which can be held by a place. $K(p_i) < \infty$ for each place in a PN, it is said that the PN has a *finite capacity* which mean it is *bounded* but otherwise it has an infinite capacity. If for each places $K(p_i) = 1$, the PN is *safe*.

After giving some fundamentals of PNs, it is better to define two basic rules which are very essential for further understanding. First one is *enabled transition*. If each input place of transition is marked at least one token, it is said that transition enabled. Second one is *firing of transition*. An enabled transition can be fired if the event that is represents occurs. In that situation $w(p,t)$ tokens are removed from each input place of transition and $w(t,p)$ tokens are added in each output place of transition. (Bogdan, Lewis, Kovacic and Mireles, 2006)

The behaviour of a PN is determined by exercising enabling and firing rules.

1) A transition $t \in T$ is enabled iff $m(p) \geq I(p,t), \forall p \in P$

2) An enabled transition t may fire at marking m', yielding the new marking

$m(p_i) = m'(p_i) + O(p_i, t) - I(p_i, t), \forall i = 1,2,..., |P|$ (Liu, 2009)

2.3.1. Structural Properties of PNs

There are two important classes of PNs. They are event graph and state machine. The definitions about both of them will be given below for further understanding.

2.3.1.1. Event Graph

If each place has only one input and one output transitions with the all connection arcs have weight 1, it is called as *event graph*. Event graphs allow representation of concurrency but not decisions (conflicts).

$|I(p_i)| = |O(p_i)| = 1 \ \forall p_i \in P$, and if all arcs have weight 1, i.e.

$w(p_i, t_j) = 1 \ \forall (p_i, t_j) \in A$ and $w(t_j, p_i) = 1 \ \forall (t_j, p_i) \in A$ (Raisch, 2009)

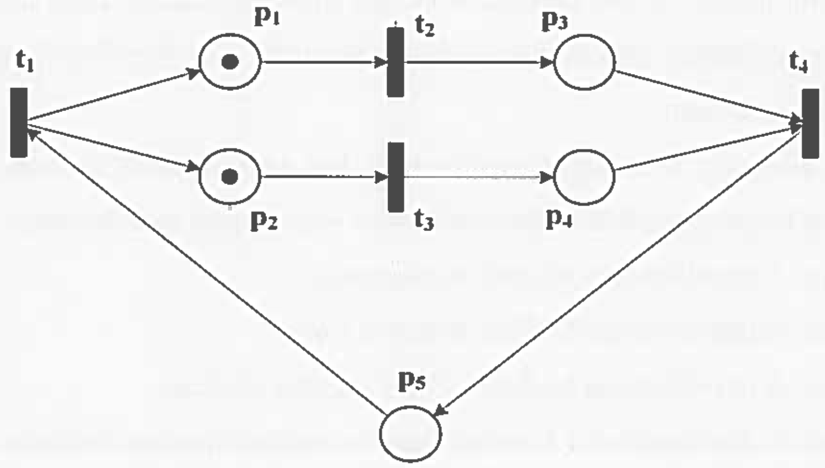

Figure 2.6. Event graph

2.3.1.2. State Machine

If each transition has only one input and one output places with the all connection arcs have weight 1, it is called as *state machine*. State machines allow the representation of decisions but not the synchronization of parallel activities.

$|I(t_i)| = |O(t_i)| = 1 \ \forall t_i \in T$, and if all arcs have weight 1, i.e.

$w(p_i, t_j) = 1 \ \forall (p_i, t_j) \in A$ and $w(t_j, p_i) = 1 \ \forall (t_j, p_i) \in A$ (Raisch, 2009)

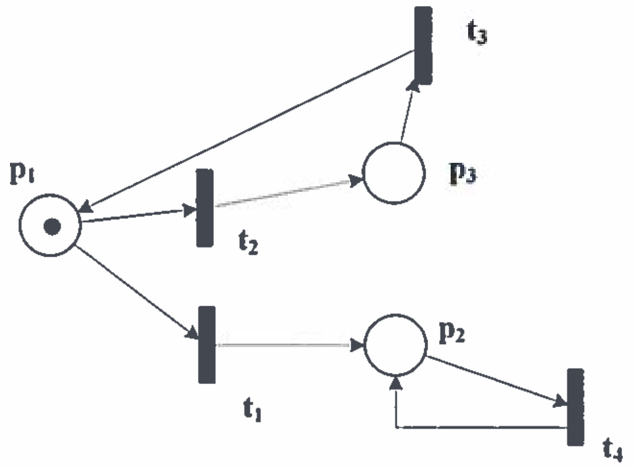

Figure 2.7. State machine

2.3.2. Behavioral Properties of PNs

2.3.2.1. Liveness

PNs with initial marking is live if there exists a firing sequence such that any transition in the PN can be fired from any marking reached from initial marking. The liveness absolutely proves that no deadlock in the operating PN, no matter what firing sequence is chosen.

Liveness is a strong requirement but very difficult to test. Because of that liveness is categorized in to four subclasses with respect to transitions.

L1 Live: A transition can be fired at least once

L2 Live: A transition can be fired at least k times.

L3 Live: A transition can be fired infinite number of times.

L4 Live: A transition is L1 Live for every m reached from m_0 (Murata, 1989).

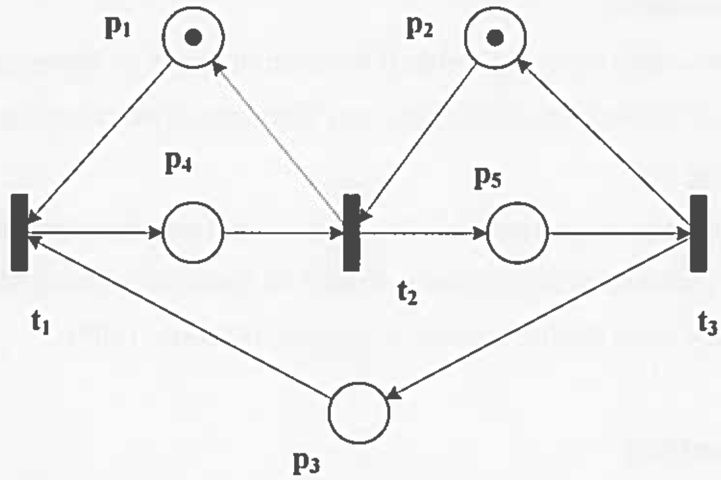

Figure 2.8. Liveness

2.3.2.2. Reachability

A marking m_j is reachable from any marking m_i if there exist a firing sequence such that it eadsa marking vector from m_i to m_j. A set of all markings reachable from m_i is denoted by $R(m_i)$. Reachability is determined by a listing of all markings that can be reached from a particular marking. Enabled transitions produce new marking and each new marking generates more markings. This kind of analysis could lead to enormous number of states. It is limited to a PN with a relatively few number of places.

Reachability analysis of a PN results in a graphical structure called as a coverability tree. For bounded PN, a coverability tree becomes a reachability tree and it contains all reachable states of the related PN.

It is evident that reachability analysis offers a solutin to many questions for PNs. However algebric analysis is proven to be more convenient for PNs. Furthermore, matrixbased modelling of manufacturing systems has much in common with the state representation of PNs (Murata, 1989).

2.3.2.3. Boundedness

A PN is bounded if the number of tokens in each place does not exceed a finite number for any marking reachable from initial marking (Murata, 1989).

2.3.2.4. Reversibility

A PN is said to be reversible if for each marking in $R(m_0)$, m_0 is reachable from each marking. It means in a reversible way from any of marking state can get back to its initial marking state.

In practice, it is required for most manufacturing systems to exhibit cyclic behavior. PN models of such systems should be reversible, hence, chacking reversibility is an important issue for the systems in practice (Murata, 1989).

2.3.2.5. Persistency

A PN is persistent if the firing of one transition will not disable the other in the case of two enabled transitions. A transition in a persistent net will stay enabled until it gets fired. Persistence is useful in the ontext of parallel program and speed independent asynchronous circuits. Persistence is related with the conflict free nets and a safe persistent net can be transformed into a marked graph by duplicating some transitions and places. All marked graphs are persistent but not all persistent nets are marked graphs (Murata, 1989).

2.3.3. Modelling by Using PNs

Some kind of methods are used in a very wide scope while modelling via PNs. These features are shown and explained briefly.

2.3.3.1. Concurreny

This structure may be used for happening events individually which are independent from each other. The events which are modelled by using this structure may realize at the exactly same time (https://polen.itu.edu.tr/bitstream/11527/4953/1/2824.pdf , (Access date: 21.05.2019)).

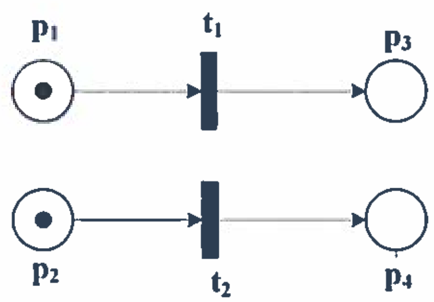

Figure 2.9. Concurrency

2.3.3.2. Synchronization

This structure is used under the circumstances of which more than one preconditions must be provided in order to be triggered for one transition (https://polen.itu.edu.tr/bitstream/11527/4953/1/2824.pdf , (Access date: 21.05.2019)).

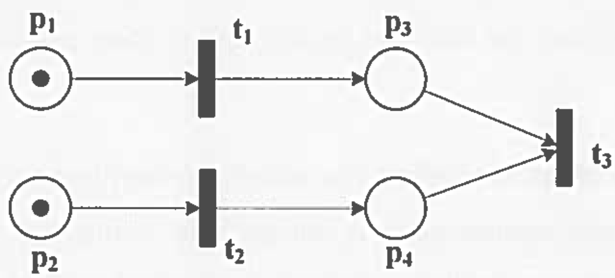

Figure 2.10. Synchronization

2.3.3.3. Conflict

If there are two different events which share the same source, they must not be allowed to access to source at the same time. It means only one event must be accessed to source each exact time. In this situation it is called that *conflict* (https://polen.itu.edu.tr/bitstream/11527/4953/1/2824.pdf , (Access date: 21.05.2019)).

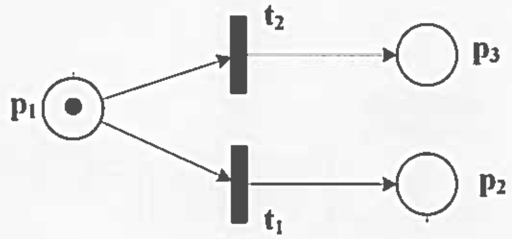

Figure 2.11. Conflict

2.3.3.4. Sequentiality

This property is used if the sequentialty is important for some events (https://polen.itu.edu.tr/bitstream/11527/4953/1/2824.pdf , (Access date: 21.05.2019)).

Figure 2.12. Sequentiality

3. MODELLING AND ANALYSIS OF LASER TRANSFER MACHINE

3.1. Laser Transfer Machine

The main motivation of this study is to model and analysis of Laser Transfer Machine (LTM) by using PNs in order to be sure that confirming the programs which are used for running the machine on site and if there exists some kind of faults, to appear them.

Because of the restrictions in a manufacturing plant such as time and labor force usage, the enough concentration is not paid for testing and commissioning the new systems and machines. In this perspective some problems and troubles may occur after commissioning these tools.

LTM was designed and established in a glassware production plant which is located in Eskişehir where is a industrially developed province of Turkey. A glassware production plant is composed of the departments which is described in Figure 3.1. basically.

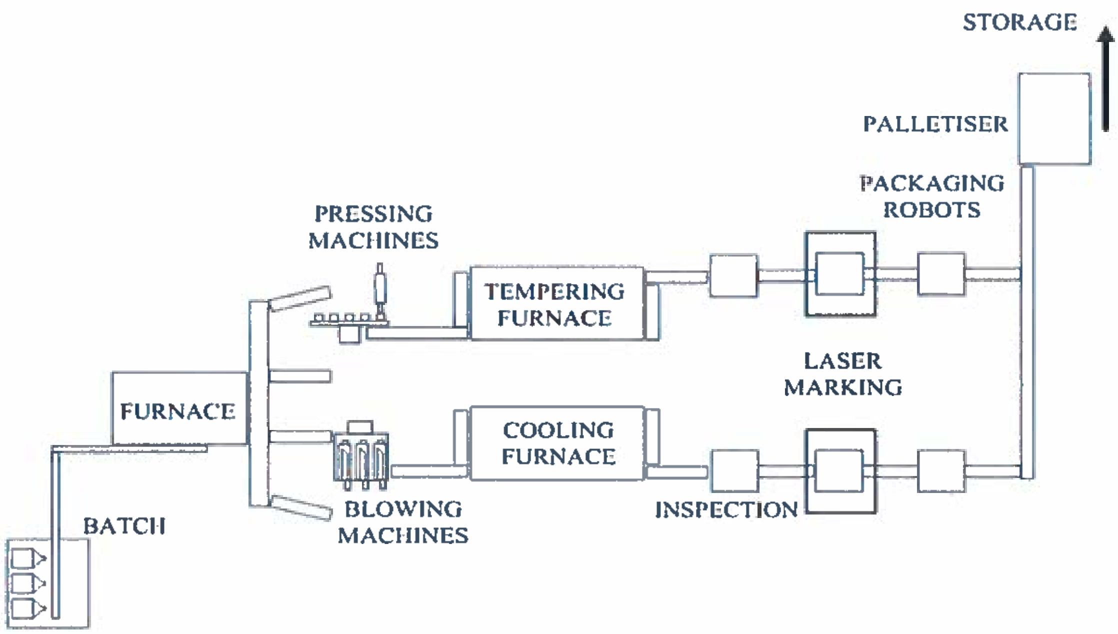

Figure 3.1. An overview of a glassware production plant

The glassware manufacturing process starts with the batch department. Several components come together with precise poriton in the mixer. The batch is transferred then to furnace to be melted in almost 1500 °C. Melted glass continues its journey with the canals which is used for feeding the machines. Shaping process starts here. Feeder system forms the gobs of melted glass for the spects of related product and cuts every gob. Gobs falls down the mould pots which are located on the forming machines. These machines are two different types. First of them is pressing machines and the second of them is blowing mahines. After get formed, hot glasswares proceed to cooling furnaces. During the process in cooling furnace the temperature of hot glassware is decreased gradually step by step. At the end of the cooling furnace the major portion of the forming process is done. There are only few steps left.

Glassware products aheads to inspection machines in order to be proven all those ones are under the certain specs by quality department decides. After inspection machines products go to laser marking process. In this process products are marked by laser machines. Marking means that putting the logo of company at the bottom of products. LTM is used in the location in order to reduce the losses because of velocity of moving products on the conveyor.

Packaging robots first form the packages and then put the products in them. Then the packages get closed and transferred to palletiser machine to be stacked in an order. Whole process ends here. Products directly are sent to storage to be waited just before to be transferred to customers.

After cooling furnace, the production wideness gets narrowed because of the composition of conveyor systems. One conveyor enables to pass only one product and there is no chance to increase the number of conveyors and of course the number of machines because of planned lay out of production site. When the huge bulk of glassware products rush in to the conveyors, there occurs some many losses.

Marking process also gets affected from this rush. Products stream in to marking machines with high velocity. Because of that situation, there are so many losses during marking process. These losses are two kinds. First one is the not marked ones and the other one is not marked in the right position of the product. Streaming with high velocity to laser marking machine may cause that laser machine can not catch every product which pass under its marker. If machine catches the product in order to mark them, high velocity of glassware on the conveyor does not allow to be marked in the right position of the glass. In order to prevent these two losses, LTM was designed, established and commissioned between the inspection machines and packaging robot on production site.

Image 3.1. General view of the real system on site

LTM is used for preventing such losses due to not to mark and not to mark in the right position of products. How LTM does that? LTM consists of two conveyors. Coming products is seperated one by one to the conveyor on the right and to the conveyor on the left. Each product is directed to one of the two conveyors on the machine. So the velocity of the coming products decreased to half of the main conveyor. During relatively short journey to main conveyor, the products are catched by the positioners which are driven by servo axis. When a positioner catches a product, it rounds the specified angle which is decided by the operator according to the number of gaps on the positioner. The shape of positioner can be resembled to a daisy of which its gaps look like the gaps between two of the leaves on the flower. This way of movement is described by the on site technical personel is indexed working. Axis which runs the positioner should take the product is coming signal from the photocell sensors which are located on each conveyor. Otherwise positioners do not move. When product arrives under laser marking machine, LTM produces a signal which means the product is under marker and laser machine starts to run in order to put the logo of the company on the bottom of product.

When marking process is finished laser marking machine produces a signal to LTM which means that marking process is done. This signal is counted by the LTM itself to save the number of products which is marked since the first run of the machine. After marking process is completed, positioner moves to get out the products from under the laser marker and gives them on the conveyor again.

Mechanical design and structure are not the concern of this thesis. Electrical design and the components will be mentioned in this chapter. There is one main electrical panel which consists of drivers, input – output units, cpu of control system, filter and dc power source and two local panels which are dedicated to each axis in order to be used at operator's will.

The important part of the electrical panel is just the control system and its components which are used for making the desired movements of axis possible. The whole parts of control system belong to Siemens. Hence the programming interface is also a product of Siemens which is commonly known as Simotion Scout and TIA (Totally Integrated Automation) Portal.

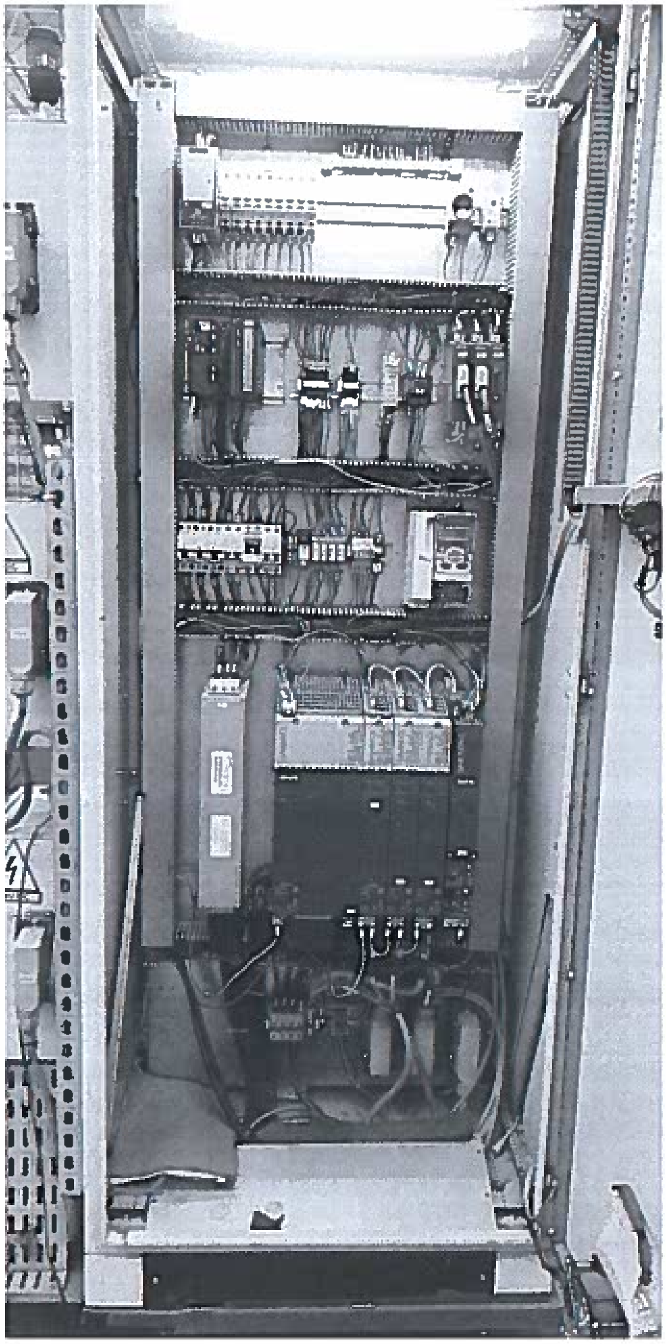

Image 3.2. Inside view of the LTM' s control panel

In Table 3.1. whole components which belong to control system is fully described. It can also be seen short description of mission of each component, their number of amounts and identity numbers.

Table 3.1. Components used in LTM's control panel

Component	Amount	Mission
D435-2 DP/PN	1	Main CPU of the control system. Enables Profibus and Profinet communication. Gathers up whole information from the site and other components and gives the orders to drivers and infeed control supply.
IM 153-4 PN	1	Distributed input and output unit. The signals from the site equipments come to this module. Processed signals in the CPU is sent from output card of this module to related points. Only Profinet communication enable.
KTP 900 Basic PN	1	Human Machine Interface (HMI) Screen. Several informations are broadcasted from here. This tool provides the possibility to control the some features of axis such as velocity, direction, etc. to operator. Only Profinet communication enable.
Smart Infeed Control	1	Basically an AC / DC converter. Supplies 600V DC voltage and 36 kW power to motor modules (drivers). Operation specs are 400 V AC, 50 Hz, 52A. Only Profinet communication enable.
Single Motor Module	2	Dummy driver modules. Supplies DC voltage and power to axis motors. Takes feedback from the encoder of motor and sends position information to CPU. 9 A and 4,3 kW.
1FK7 Synchronous Motor	2	Axis motors. Used for moving the positioner. Gives feedback via its encoder to driver module. 3000 rpm, 7,3 Nm, 5,6A. It has no break.
Line Filter	1	Used for fixing the troubles which occur on the main voltage. Supplies a perfect sinusiodal voltage signal to Smart Infeed Control.

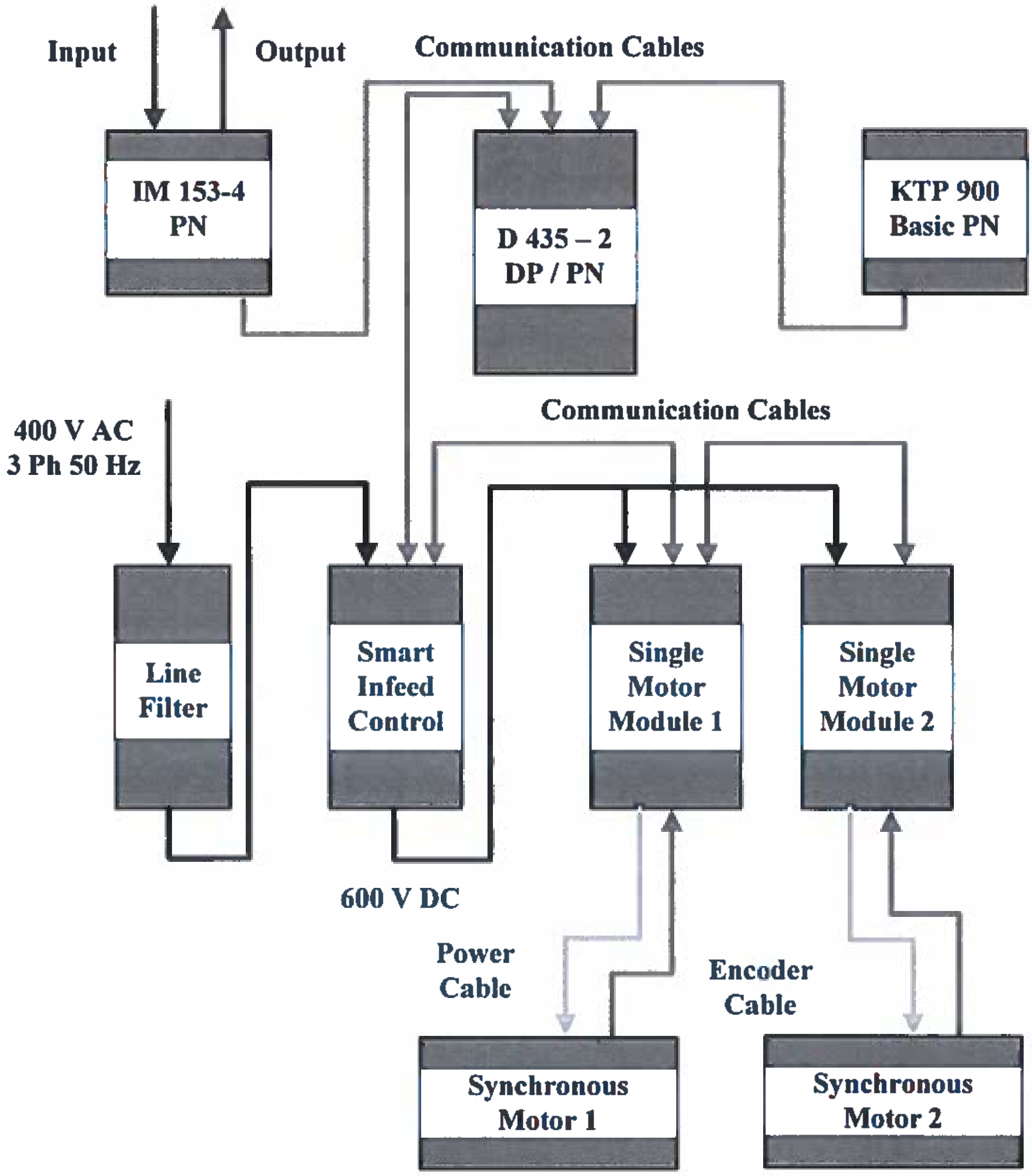

Figure 3.2. General concept of the control system of LTM

Working principle of hardware of control system is quite easy. Whole information on site comes to D 435 – 2 DP/PN by Profinet communication network. These are position information coming from the encoders of synchronous motors, on site sensor information which sense the existence of some desired inputs, informations coming from KTP 900 Basic Mono and the parameters of both Single Motor Modules and Smart Infeed Control. Profinet communication network allows transmitting the information on both sides which means that sending and receiving communication signals can be transferred with the same cable. This feature enables transmitting the orders coming from CPU to drivers and AC/DC converter module.

After an operator arrange the suitable set up, the system starts running according to desired specifications. These specifications may be running in automatic mode, turning to right side with a specific velocity and a specific acceleration, stopping at the exact position which the operator gives to the system.

All information comes from outside world and sends to motors are evalutaed by the programs which are formed of a specific software. Used software is TIA Portal which is a product of Siemens used for programming automation projects. The hardware configuration is formed in TIA Portal which means selecting the right components which are used in project in the libraries of software and establishing the communication network. By selecting the D435 – 2 DP/PN CPU, TIA Portal enables programming motion control tool which is called as Simotion Scout.

Simotion Scout is a special subbranch of TIA Portal which is only used for programming motion control of automation projects. The main frame of the software consists of the composition of the tasks which is simply called as Execution System. The tasks on the Execution System are the different branches which have different missions in order to execute different specifications of main frame. These tasks get cycled in different periods which are specified by Simotion Scout's software itself. During their own cycles they are executed by the Execution System. An overview of Execution System can be seen in Figure 3.3.

The Execution System provides various execution levels. Execution levels are assigned to user program tasks. Programs are assigned to the user program tasks. All programs, thus also tasks can contain PLC and motion control tasks. The following execution levels are available; synchronous, time – driven, event – driven, interrupt – controlled, sequential and free – running execution levels.

Programs, written by the programmer are assigned to suitable tasks which are needed to be executed to realize the desired missions. One or more tasks are available in each execution level for user programming. There are several tasks which can be used in the Execution System but to be modelled and analysed programs in this study are only assigned to System Interrupt Task and Motion Task.

For further and deeper understanding of the system, the tasks will be basically described in Table 3.2.

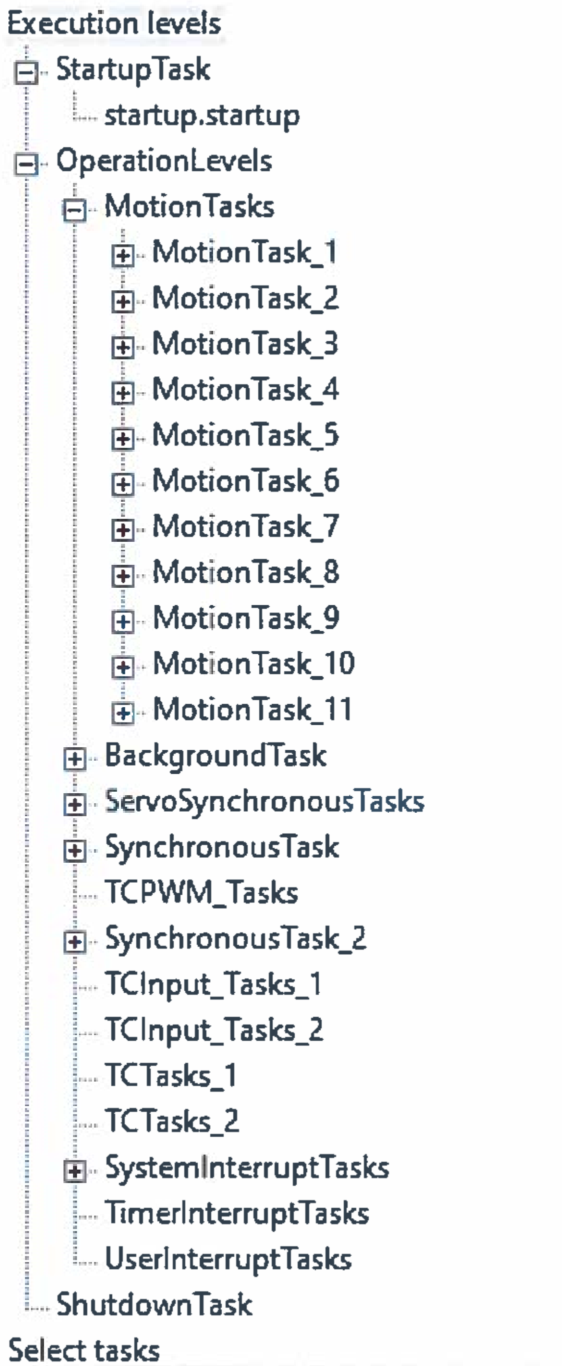

Figure 3.3. Overview of execution system

Table 3.2. Tasks used in execution system

TASK	DESCRIPTION
Start Up Task	The StartupTask is executed once at the transition from STOP mode to RUN mode. It is intended for initialization and resetting of technology objects.
Free Running Tasks	In the round robin execution level, the *Motion Tasks* and *Background Task* are executed by the system in the background in the time-slice procedure.
Time Driven And Synchronous Tasks	Cyclic tasks. They are called cyclically in a certain time frame and are automatically restarted after the execution of the assigned programs.
Event – Driven Tasks	Sequential tasks. They are started and executed once when an event occurs and then terminated. *System Interrupt Task* and *User Interrupt Task* are members of event – driven tasks.
Shutdown Task	The Shutdown Task is executed once at the transition from RUN mode to STOP mode.

In this study the programs only assigned to System Interrupt Task and Motion Tasks. There are four of programs will be modelled and analysed. Before giving some details about these programs, it is better to give some information about the characteristics of System Interrupt Task and Motion Task.

Motion Tasks are intended for programming of sequenes, for programmed motion control or other sequential executions. If needed to give an exmple for the sequential execution, an axis traverses to a target position, waits for an enable signal, and then traverses to the next target position. A motion task starts automatically when the RUN mode has been reached. This means when a program is assigned to Start Up Task is completed, the CPU reaches to RUN mode. A motion task does not have a time monitoring. It can remain active for an indefinite period. A motion task that waits for a synchronous command remains active with regard to its status. When the conditions are fulfilled, the motion task is automatically begun and resumed.

System Interrupt Task is started and executed once when a system event occurs. An internal system task checks approximately every 10 ms whether a configured event has occured and then starts applicable task automatically. If an event occurs which starts the System Interrupt Task, the CPU enters STOP state. System Interrupt Task does not need RUN signal because of its mission is protection. When an event occurs which executes the System Interrupt Task, in any condition of CPU, it must be executed for protection. But in this study the program which assigned to System Interrupt Task needs

RUN signal because of the protection which axis are active during the RUN mode. And also the general consequences of System Interrupt Task will not be examined, instead only stopping and deactivating of axis will be in scope.

Before getting started modelling and analysing of the programs, it is better to mention purposes and missions of these programs in Table 3.3.

Table 3.3. Programs and tasks used in LTM

Program	**Assigned Task**	**Definition**
Emergency Stop	System Interrupt Task	Due to the priority of tasks, System Interrupt Tasks are in higher importance than Motion Tasks. When an emergency situation occurs during the operation process, by only pushing the emergency stop buton, Emergency Program is executed and first stops and then deactivates whole axis motors.
Manual	Motion Task	Allows to the operator of the machine to drive the axis manually in order to make a set up or just detecting a failure situation.
Reference	Motion Task	Used for referencing the axis motors in order to find their own home position which is specified by the operator. Executed only once. If an axis is referenced, there is no need to be executed otherwise referencing process must be done.
Automatic	Motion Task	Normal operation. When the photocell sensor detects the existence of a product, the axis is moved and brings the product under laser marker. After the marking process is done, product is got out of the station.

3.2. Emergency Stop Program

Emergency Stop Program is assigned to System Interrupt Task. This program provides a safety precautions against risky, dangerous and unexpected situations. These situations are observed on site as a daily normal. Preventing these kind of situations are vital because if something dangerous happens on site, an operator may injure or he may even die.

Because of the priority of System Interrupt Task in Execution System, Emergency Stop Program is assigned to this task. Hence when something dangerous happens, this program is executed by only pushing to emergency buton.

This program has three missions to comply.

1) Stopping and deactivating axis directly when an emergency situation occurs. Emergency stop buton is pushed and then the running axis stops and deactivates immediately in whatever task is being executed in the execution system.

2) Resetting the emergency situations and returning to normal run. After emergency situation passes, by returning to emergency stop buton its previous state and pushing the reset buton, this mission is executed. Stopped and deactivated axis return to be activated.

3) Normal run. There is no emergency buton pushed or resetting activities. Whole axis are active.

In Figure 3.4. a basic flowchart is given, in order to be obtained how this program works.

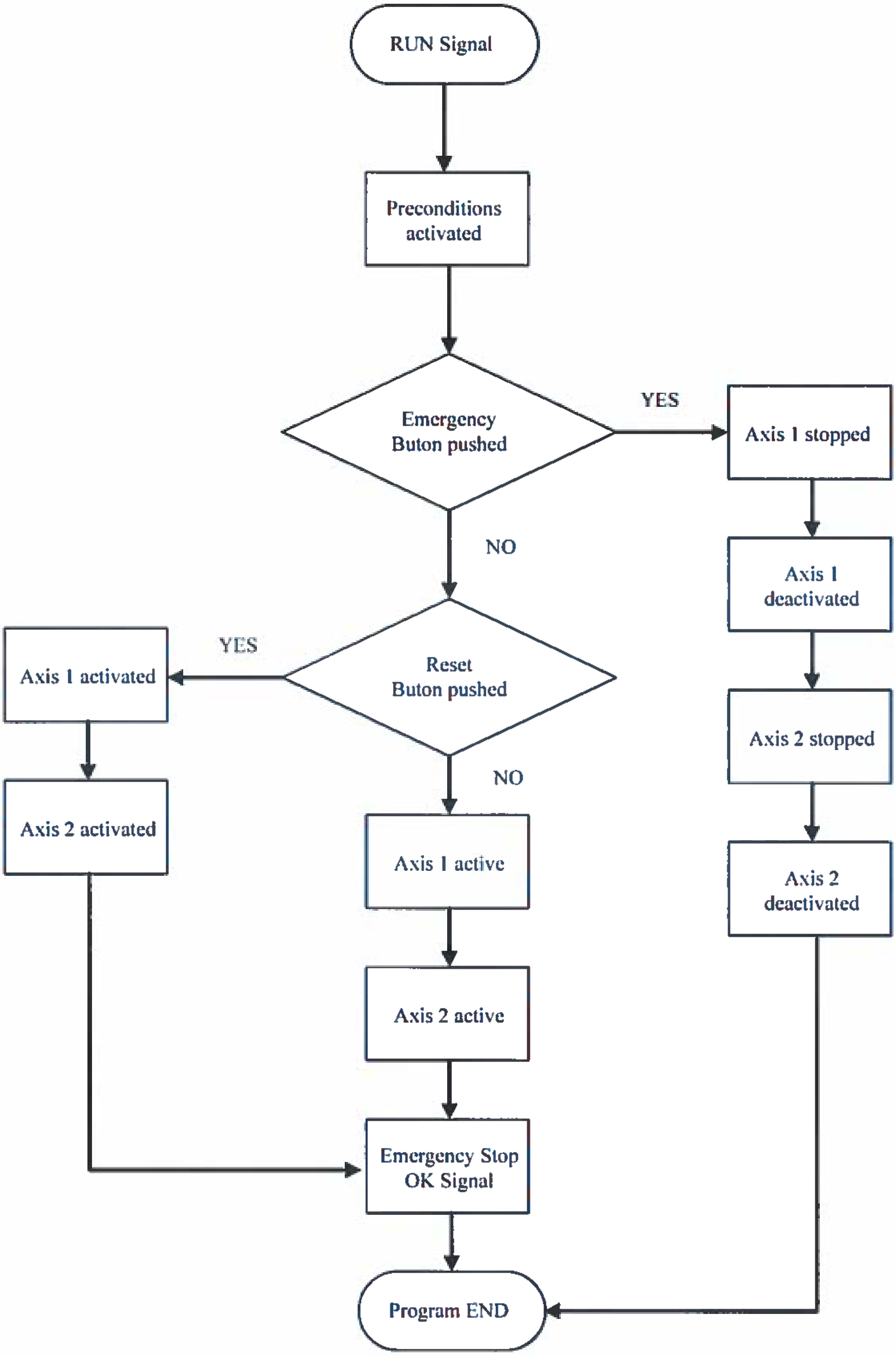

Figure 3.4. Flowchart of emergency stop program

3.2.1. PN Model of the Emergency Stop Program

There are 18 places and 16 transitions in PN model of Emergency Stop Program. All arcs are weighted 1. The places, the transitions and PN model are seen in Table 3.4., Table 3.5. and Figure 3.5. respectively.

Table 3.4. Places of emergeny stop program and their meanings

p_0	Run status. Enables to the start status. Internal marking which is produced by program automatically.
p_1	Start status. Enables to stream the related program. Internal marking which is produced by the program automatically.
p_2	Emergency stop status. Physical input. According to the position of manual switch, it supplies 24 VDC signal to stop the axis.
p_3	*Emergency Stop OFF* status. Physical input. There is no 24 VDC signal in the related input of the system. It means there is no problem. Everything in the system is able to operate properly.
p_4	*Emergency Stop ON* status. Physical input. There is 24 VDC signal in the related input of the system. It means there is a problem. Everything in the system must stop immediately whatever condition in they are.
p_5	Reset status. Physical input. Supplies 24 VDC signal to related input to reset the conditions which occurs after emergency situation happens. If the emergency conditions are fixed and then the reset buton is pressed to enable, the problems are disappeared and the system is able to operate properly again.
p_6	*Emergency Stop NOT OK* status. It means somebody pushed the emergency buton and the 24 VDC signal were cut off.
p_7	Transfer 1 axis status. Internal marking. Not a physical input.
p_8	Transfer 1 axis stop status. After emergency not ok signal occured, the axis turns in to stop status primarly.
p_9	True condition. Just a trespassing condition to enable the Transfer 1 axis enable. Internal marking.
p_{10}	Transfer 1 axis disable status. The driver of related axis in this status cuts off the power.
p_{11}	Transfer 2 axis status. Internal marking. Not a physical input.
p_{12}	Transfer 2 axis stop status.
p_{13}	Transfer 1 axis enable status. The driver of related axis in this status is ready to give the electrical power to motor.
p_{14}	Transfer 2 axis disable status. The driver of related axis in this status cuts off the power.
p_{15}	Transfer 2 axis enable status. The driver of related axis in this status is ready to give the electrical power to motor.
p_{16}	*Emergency Stop OK* status. It means the 24 VDC control signal reaches to the related input.
p_{17}	Program end.

Table 3.5. Transitions of emergency stop program and their meanings

t_0	Activates the p_1, p_2, p_5, p_7 and p_{11} status autoatically when the p_0 has a token.
t_1	Normally open position of *Emergency Buton*. Controlled manually by the operator.
t_2	Normally closed position of *Emergency Buton*. Controlled manually by the operator.
t_3	Activates the p_6 status automatically when the p_4 and p_1 have tokens.
t_4	Checks the p_1, p_3, p_5, p_7 and p_{11} status. If all these events have tokens it is fired automatically to confirm the *Emergency Stop OK* status.
t_5	*Reset ON* buton. Controlled manually by the operator.
t_6	Checks the p_5 and p_6 status. If both events have tokens it is fired automatically to confirm to stop the Transfer 1 axis moves.
t_7	Checks the p_7 and p_8 status. If both events have tokens it is fired automatically to disable the Transfer 1 axis.
t_8	Checks the p_7 and p_9 status. If both events have tokens it is fired automatically to enable the Transfer 1 axis.
t_9	Checks the p_{10} status. If that place has a token it is fired automatically to confirm to stop the Transfer 2 axis moves.
t_{10}	Checks the p_{11} and p_{12} status. If both events have tokens it is fired automatically to disable the Transfer 2 axis.
t_{11}	Checks the p_{11} and p_{13} status. If both events have tokens it is fired automaticaly to enable the Transfer 2 axis.
t_{12}	Checks the p_{14} status. If that place has a token it is fired automatically to end the program.
t_{13}	Checks the p_{15} status. After confirming that the Transfer 2 axis is enable automatically fired to allow the *Emergency Stop OK* signal.
t_{14}	Checks the p_{16} status. If there is a token on that place it means that *Emergency Stop OK* signal exists and transition is automatically fired to end the program.
t_{15}	Checks the p_{17} status. If that place has a token it is fired automatically to return to run status.

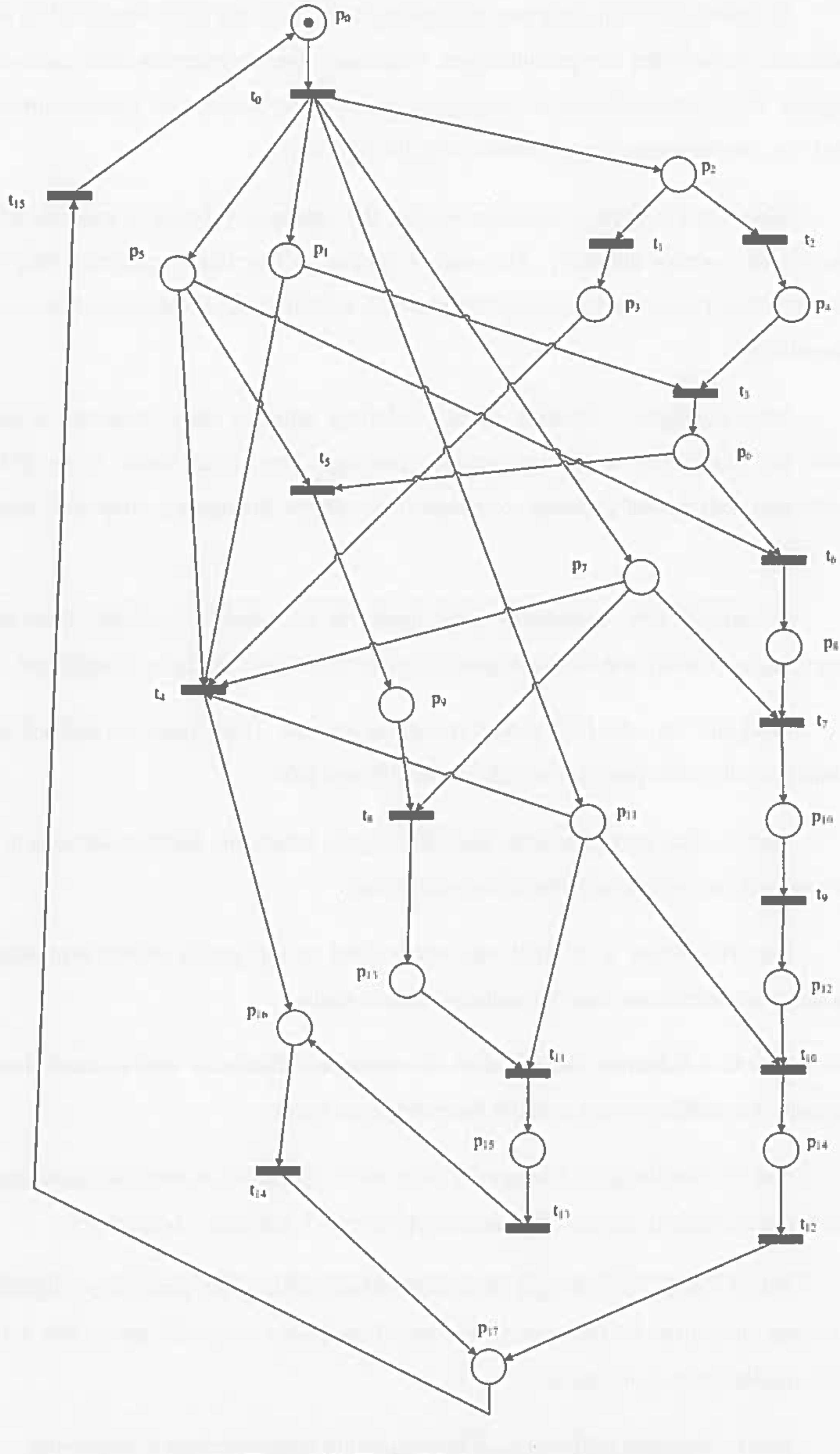

Figure 3.5. PN model of emergency stop program

In Emergency Stop program, everything starts with the RUN signal. RUN signal automatically activates the preconditions which are used to progress other parts of the program. These preconditions are emergency buton, reset buton, axis 1 and 2 commands which are used to activating or deactivating the both axis.

When an emergency situation occurs, the emergency buton is pushed. This is done by the operator manually. The control system decides that Emergency Stop NOT OK status has a token and then axis 1 and axis 2 switch to deactivated and then stopped respectively.

After emergency situation passes, resetting must be done. Operator manually switch the emergency buton its previous position. Then reset buton is pushed and deactivated axis 1 and 2 return to active. This allows Emergency Stop OK status is activated.

In normal run, emergency stop buton is not already pushed. That means everything is in order and works properly. Emergency Stop OK status is activated.

In Figure 3.6., the P.N. model is seen as aparted. These parts are defined as the submissions in entire program which do the different jobs.

Part A : This part represents the RUN signal. When this status is active t_0 is fired automatically in order to activate the preconditions.

Part B : When t_0 is fired, the token goes to the places which represent the preconditions which are used for butons and axis status.

Part C : Selecting the position of emergency buton. t_1 and t_2 transitions are represent the pushing or not pushing the emergency buton.

Part D : By firing t_3, Emergency Stop NOT OK status is reached automatically. This situation leads to deactivating and stopping axis 1 and axis 2 respectively.

Part E : Axis get activated. It is used either resetting the emergency situation or in normal operation. In the end of this part Emergency Stop OK status has a token which implies there is no problem.

Part F : Program END status. The program is ready to return to beginning.

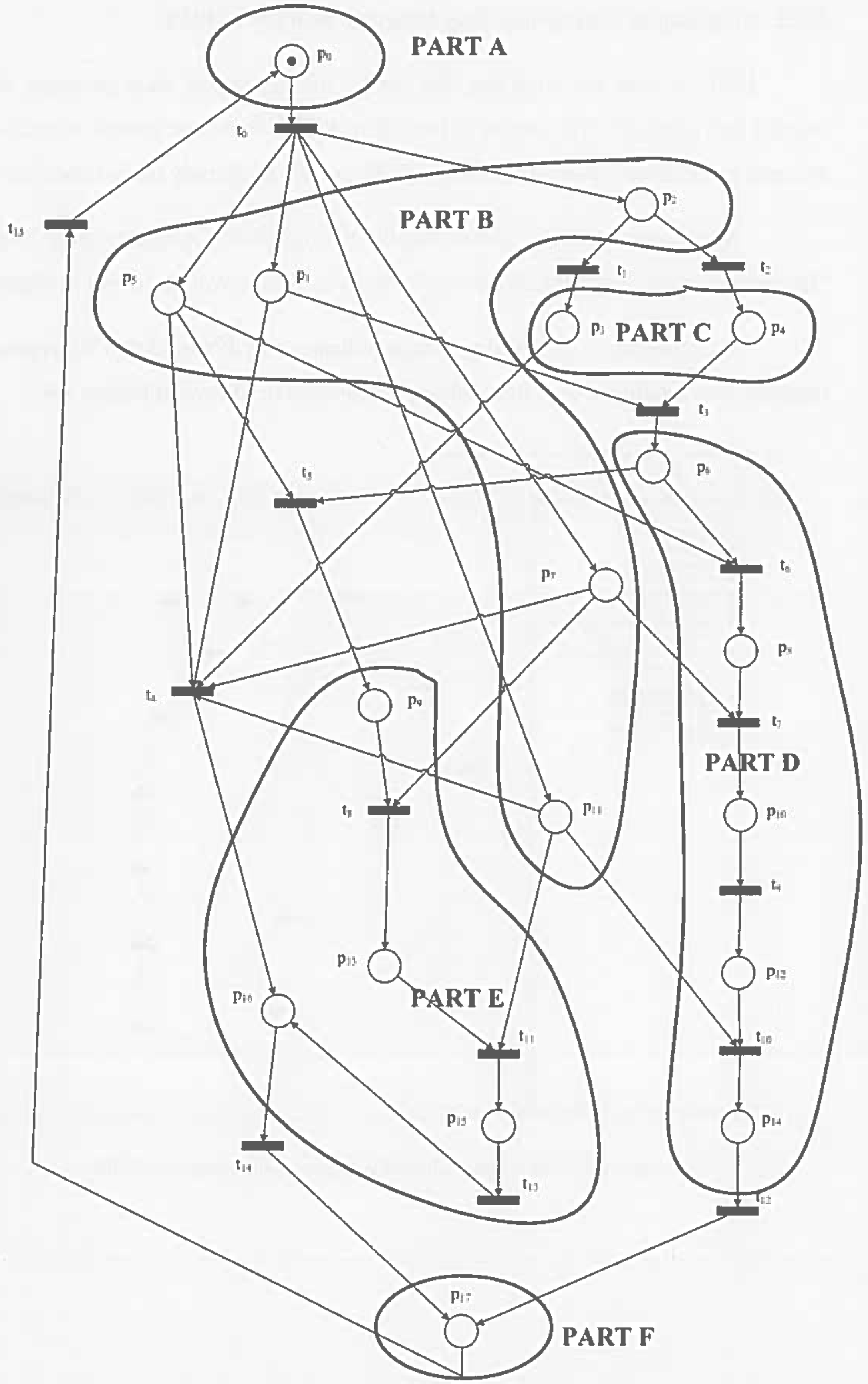

Figure 3.6. Parts of PN model of emergency stop program

3.2.2. Analysing of Emergency Stop Program as a PN in PIPE

PIPE is used for analysing PN model of Emergency Stop program which is showed in Figure 3.7. The model is formed in PIPE. There are several modules which are used for analysing such structures in PIPE. In this study only two of them are used.

- State Space Analysis shows that the PN model of Emergency Stop Program is "Bounded", "Safe" and "Deadlock Free". Analysis results will be shown in Figure 3.8.

- Reachability / Coverability Graph indicates that PN model of Emergency Stop Program is reversible or not. Reachability Graph will be shown in Figure 3.9.

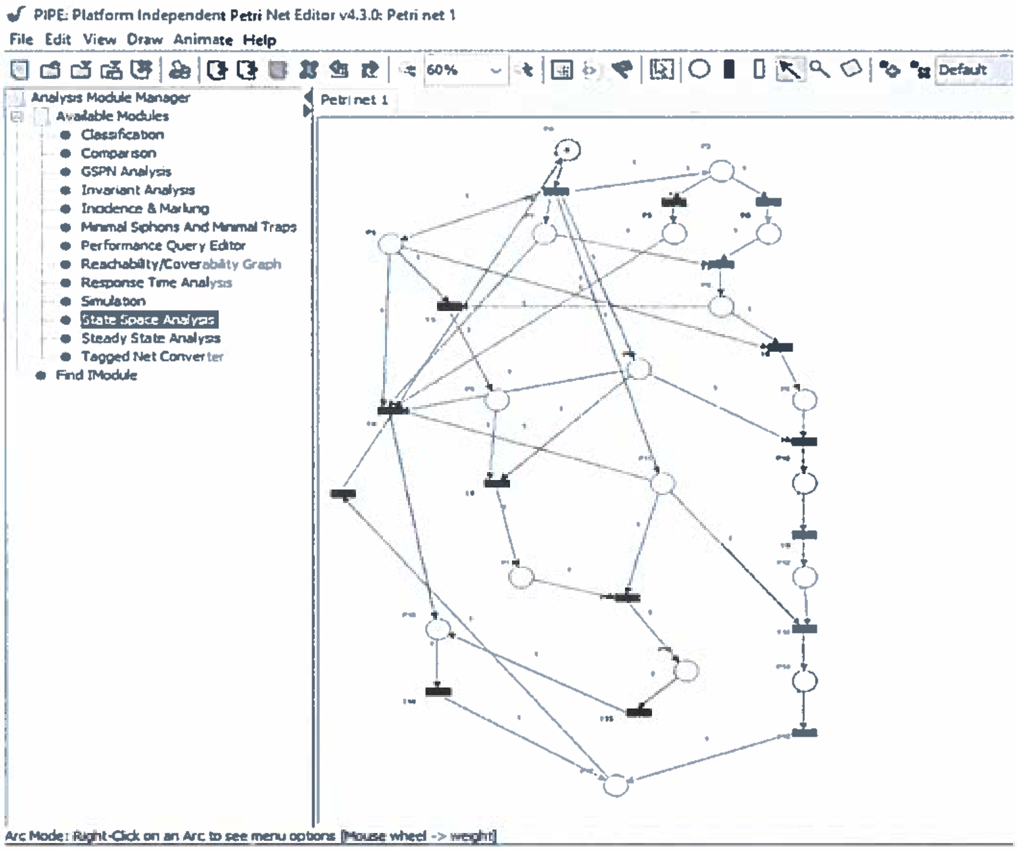

Figure 3.7. PN representation of emergency stop program in PIPE

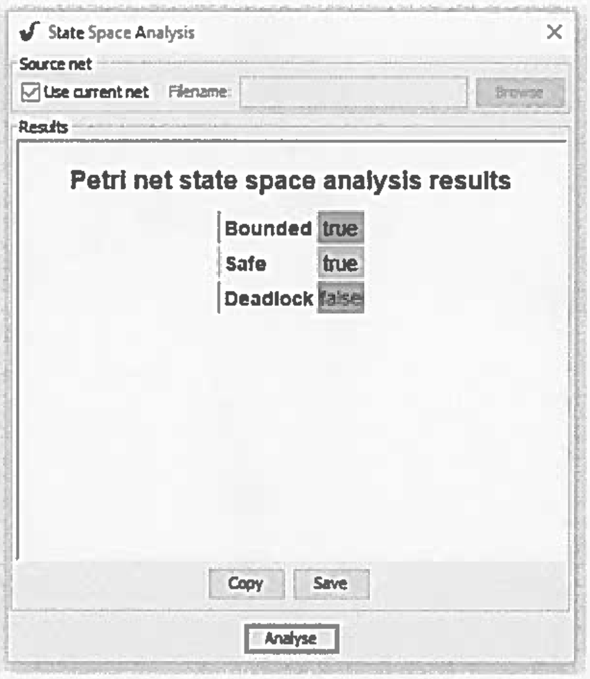

Figure 3.8. Analysis results of the emergency stop program In PIPE

State Space Analysis results show that PN model is bounded, safe and deadlock free.

1) According to bounded and safe in every single enable transition is fired there can be only one token in related places. This result implies that in each cycle of the program, each state can be enable only once. This shows that while the Emergency Stop Program is being executed with no delay from the beginning till the end. There is no extra cycling burden on the CPU.

2) Deadlock free implies that the Emergency Stop Program can be executed in each cycle without stucking into a state. All transitions can be enabled during execution.

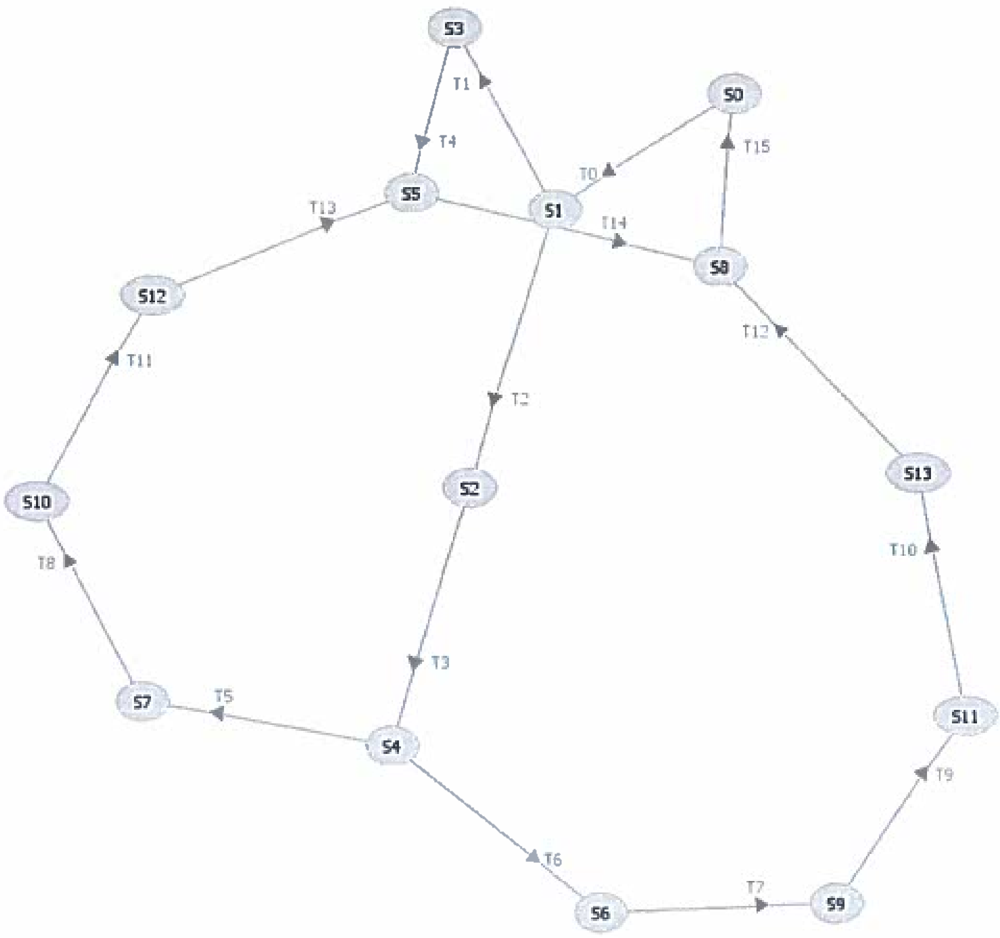

Figure 3.9. Reachability graph of emergency stop program

As can be seen in Figure 3.9, reachability graph is shown. S_0 indicates the initial state of the program which is established according to M marking vector. Other S markings are the possible marking vectors which can be obtained while all possible enabled transitions are fired in order.

$M = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}\}$

$S_0 = \{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$

There are totally 14 possible states until to reach to initial state of Emergency Stop Program. The reachability graph starts with the S_0 tracks 3 possible ways to turn back to its initial marking state.

1) $S_0 \Rightarrow T_0 \Rightarrow S_1 \Rightarrow T_1 \Rightarrow S_3 \Rightarrow T_4 \Rightarrow S_5 \Rightarrow T_{14} \Rightarrow S_8 \Rightarrow T_{15} \Rightarrow S_0$

2) $S_0 \Rightarrow T_0 \Rightarrow S_1 \Rightarrow T_2 \Rightarrow S_2 \Rightarrow T_3 \Rightarrow S_4 \Rightarrow T_6 \Rightarrow S_6 \Rightarrow T_7 \Rightarrow S_9 \Rightarrow T_9 \Rightarrow S_{11} \Rightarrow T_{10} \Rightarrow S_{13} \Rightarrow T_{12} \Rightarrow S_8 \Rightarrow T_{15} \Rightarrow S_0$

3) $S_0 \Rightarrow T_0 \Rightarrow S_1 \Rightarrow T_2 \Rightarrow S_2 \Rightarrow T_3 \Rightarrow S_4 \Rightarrow T_5 \Rightarrow S_7 \Rightarrow T_8 \Rightarrow S_{10} \Rightarrow T_{11} \Rightarrow S_{12} \Rightarrow T_{13} \Rightarrow S_5 \Rightarrow T_{14} \Rightarrow S_8 \Rightarrow T_{15} \Rightarrow S_0$

In every three firing sequence which can be seen in the Reachability Grap in Figure Reach, the processes start with the S_0 initial marking state. After possible transitions fired each process return to initial state. This situation proves that the Emergency Program is reversible. This is so important result for program itself.

After starting, program passes through different ways to reach the end according to the plot of the program itself. During the process there is no deadlock which makes program stuck into a situation. The program is also bounded and even safe which cause no delay which makes program not loose time to execute entire orders. And according to Reachability Graph, the Emergency Stop Program can return to its initial situation which makes enable itself for the next turns.

3.2.3. Findings and Results

State Space and Reachability Graph Analysis indicates that the Emergency Stop Program is Bounded, Safe, Deadlock free and Reversible.

Boundedness and Safeness proves that there can be only one token in any places during the one cycle of the program. This showes that there is no delay during the program streaming.

Deadlock free proves that Emergency Stop Program does not stuck in any places which forbid to reach to end.

Reversibility also shows that Emergency Stop Program can return to its initial status which renew the program cycle.

3.3. Manual Program

Manual Program is assigned to Motion Task. This program gives a possibility to operator of LTM intervening to machine manually. In the cases of failures or unwanted moves, operator can switch to manual mode and then intervene as he wishes. For example in a failure situation an axis does not move. Operator suspects from some point which may cause the axis not to move. He can easily switch the manual mode on and makes the axis move individually by his wish. In that way the operator understands that the axis motor, power and encoder cables and even driver of motor are entirely healthy. So in order to find the reason of failure, it should be focused to other parts of the machine.

When the operator would like to set the position of axis in to a specific position, he switches to manual mode and he can bring the axis in desired point by turning the axis to left or right.

Motion Tasks are the lowest priority tasks among the Execution System in Simotion Scout. So in any urgent cases they are the first to stop. But also motion tasks are the main supplier of what is desired to be done by the program itself.

The position and the turning direction of the axis does not matter in Manual Program. So the only parameter needed to be set is the manual velocity. Before intervening to the machine, operator should enter the manual velocity value to screen as an input. This value is sensed by the control system and CPU allows to move the axis through which side the operator wants to turn the axis with manual velocity.

This program has two missions to comply.

1) Turning the axis to right. During the operator pushes the right side buton, axis continues to turn to right.

2) Turning the axis to left. During the operator pushes the left side buton, axis continues to turn to left.

When an emergency situation occurs, operator pushes the emergency stop buton. This intervention enables the program by-passes the manual move operation and directs the program to end without doing nothing. Axis are stopped by the Emergency Stop Program.

In Figure 3.10. a basic flowchart is given, in order to be obtained how this program works.

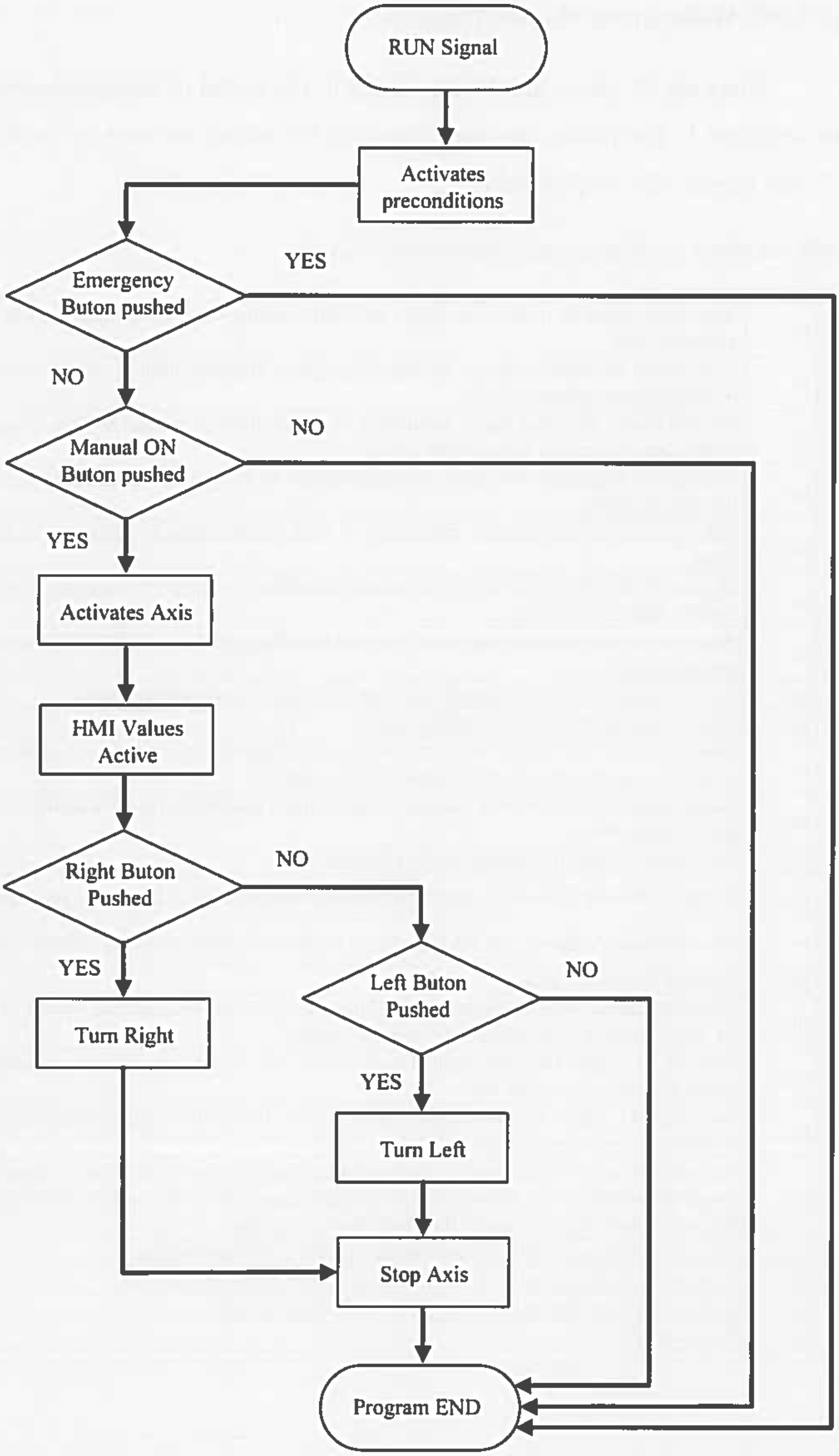

Figure 3.10. Flowchart of manual program

3.3.1. PN Model of the Manual Program

There are 23 places and 18 transitions in PN model of Manual Program. All arcs are weighted 1. The places, the transitions and PN model are seen in Table 3.6., Table 3.7. and Figure 3.11. respectively.

Table 3.6. Places of manual program and their meanings

p_0	Run status. Enables to the start status. Internal marking which is produced by program automatically.
p_1	Start status. Enables to stream the related program. Internal marking which is produced by the program automatically.
p_2	Manual status. Physical input. According to the position of manual switch, it supplies 24 VDC signal to choose ON or OFF status.
p_3	*Emergency Stop NOT OK* status. It means the 24 VDC control signal does not reach to the related input.
p_4	*Emergency Stop OK* status. It means the 24 VDC control signal reaches to the related input.
p_5	Manual Mode ON status. Just a trespassing condition to enable the transfer 1 axis enable. Internal marking.
p_6	Actor drive state monitoring status. Internal marking which is produced by the program automatically.
p_7	Current values on HMI (Human Machine Interface) screen enable status.
p_8	New values on HMI screen enable status.
p_9	Manual Mode OFF status. Just a trespassing condition to proceed the program to end unless the conditions one step before are not suitable.
p_{10}	Actor drive state monitoring "Active" status which means the driver which drives the axis is ready to run.
p_{11}	Axis status. Internal marking. Not a physical input.
p_{12}	Current "Manual Velocity" on HMI screen is ready to send to system as prerequisites.
p_{13}	New "Manual Velocity" on HMI screen is ready to send to system as current values.
p_{14}	Manual Velocity status.
p_{15}	Direction status. Physical input. According to the position of direction switch, it supplies 24 VDC signal to choose RIGHT or LEFT status.
p_{16}	Turn LEFT status. Physical input. There is 24 V DC control signal on the related input to allow the axis turn to left side.
p_{17}	Turn RIGHT status. Physical input. There is 24 V DC control signal on the related input to allow the axis turn to right side.
p_{18}	Axis enable status which means after enabling the driver to ready to use, it transformed single phase 600 V DC power voltage to three phase 380 V AC voltage to be used for the axis motor in any moment the conditions come true.
p_{19}	Axis moves to left status. Internal marking. Not a physical output.
p_{20}	Axis moves to right status. Internal marking. Not a physical output.
p_{21}	Axis stop status. Internal marking. Not a physical output.
p_{22}	Program end.

Table 3.7. Transitions of manual program and their meanings

t_0	Activates the p_1, p_2, p_4, p_6, p_7, p_{11} and p_{15} status autoatically when the p_0 has a token.
t_1	Manual ON buton. Activates the p_5 status automatically when the p_1, p_2 and p_4 have tokens.
t_2	Manual OFF buton. Checks the p_1 and p_2 status. If the conditions come true it is fired automatically.
t_3	*Emergency Buton.* It is activated by operator. When the operator pushes the buton, *Emergency Stop OK* signal is cut off
t_4	Checks the p_5 and p_6 status. If the conditions come true it is fired automatically to make the p_{10} condition is enable.
t_5	Checks the p_{10} and p_{11} status. If the conditions come true it is fired automatically to make the p_{18} condition is enable.
t_6	Checks the p_7 status. If there is no change on the previous HMI screen value it is fired automatically to next step to activate the p_{12} status.
t_7	Checks the p_7 status. If there will be a change on the HMI screen value, it is fired by the operator's intervenion by touching the related zones of the HMI screen.
t_8	Checks the p_8 status. If the new value on the HMI screen is entered it is fired by the operator by pressing the enter buton to approve manually.
t_9	Checks the p_{13} status. If the new value on the HMI screen is approved it is fired automatically to send the new value as the value which is ready to send.
t_{10}	Checks the p_{12} status. If there is a token in that place it is fired automatically as the HMI value to be sent to system.
t_{11}	Checks the p_{15} status. It is fired automatically to allow choosing the direction manually.
t_{12}	Manual switch to allow the axis move to right side.
t_{13}	Manual switch to allow the axis move to left side.
t_{14}	Checks the p_{16} and p_{20} status. It is fired by operator to stop moving right of the axis.
t_{15}	Checks the p_{17} and p_{19} status. It is fired by operator to stop moving left of the axis.
t_{16}	Checks the p_{21} status. When it is reached it is fired automatically to end the program.
t_{17}	Checks the p_{22} status. If that place has a token it is fired automatically to return to run status.
t_{18}	Checks the p_3, p_6, p_9, p_{11}, p_{14}, p_{16} and p_{17} status. It is fired automatically when these events come true to enable the program end status.

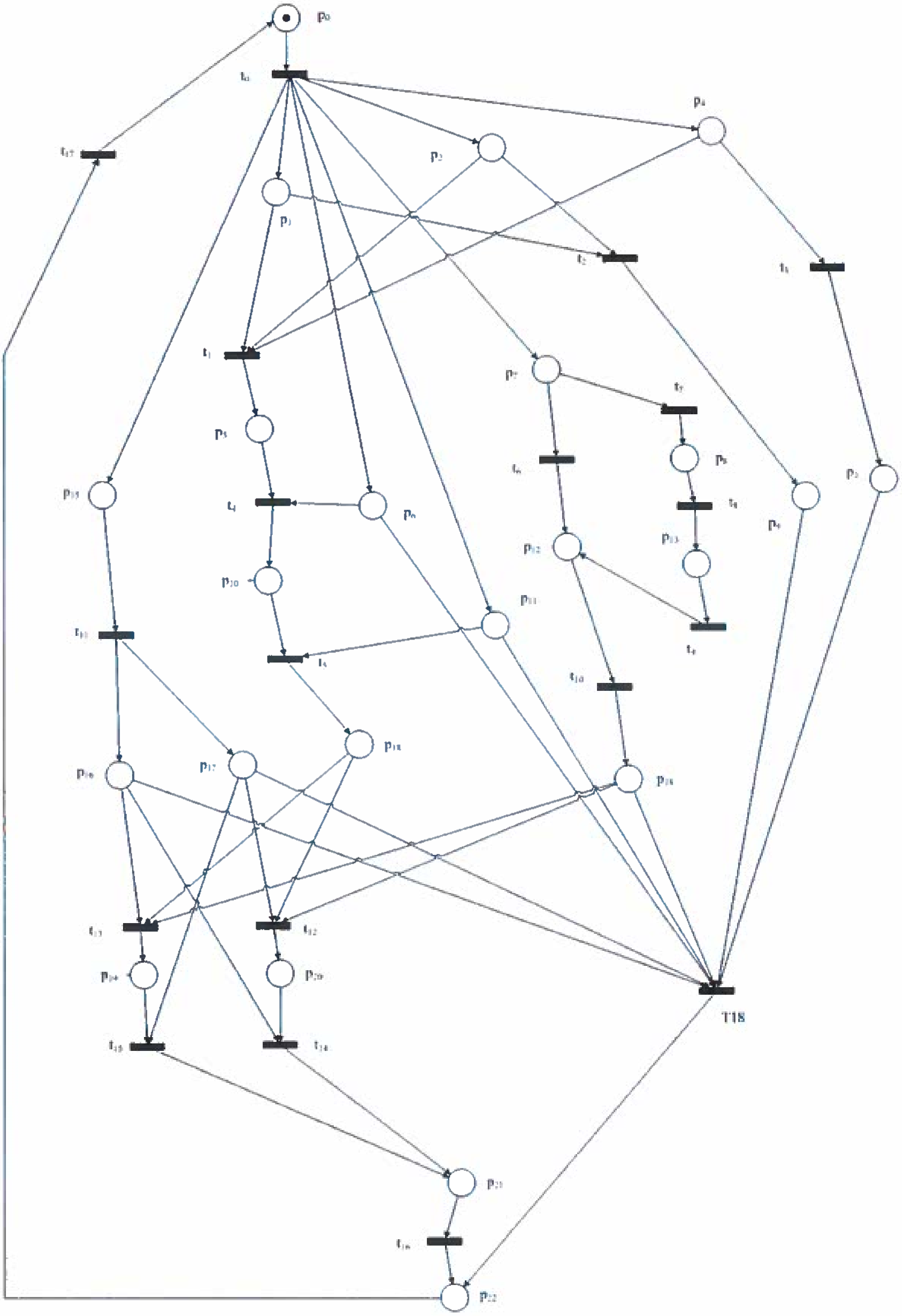

Figure 3.11. PN model of manual program

In Manual Program, all process starts with the RUN signal comes. When RUN status has a token which comes from the Execution System itself for the very first time when the system executes Start Up Task, the bounded preconditions are automatically activated by firing the t_0. These preconditions are emergency buton, manual status buton, direction status buton, actor drive state monitoring status and axis status.

Operator pushes the manual on selection buton by firing t_1 transition. The control system first checks the driver state and turns it into Active. Then axis is enabled. On the other hand operator specifies the manual velocity value. Combining the axis enabled and manual velocity value, the axis is allowed to turn to both sides.

By the choice of operator, switching the right side buton or left side buton, the related axis starts to move until the signal of other side comes.

If an emergency situation occurs, operator directly hits the emergency stop buton. Emergency Stop Program deactivates and stops the axis but in Manual Program, emergency situation directly by-passes the program streaming into end.

In Figure 3.12., the P.N. model is seen as aparted. These parts are defined as the submissions in entire program which do the different jobs.

Part A : This part represents the RUN signal. When this status is active t_0 is fired automatically in order to activate the preconditions.

Part B : When t_0 is fired, the token goes to the places which represent the preconditions which are used for butons and axis status.

Part C : Setting the screen values. In this program, only needed value is manual velocity. In this part current screen value can be used or new value can be specified by the operator.

Part D : An emergency situation occurs, operator pushes the emergency buton. This part directly by-passes whole activity in the program and preceeds to end.

Part E : Axis status. First the driver status turns into active and then axis becomes enable.

Part F : Manual direction changing. By pressing to one side buton, axis moves through that way. When the other side signal comes, the axis stops.

Part G : Program END status. The program is ready to return to beginning.

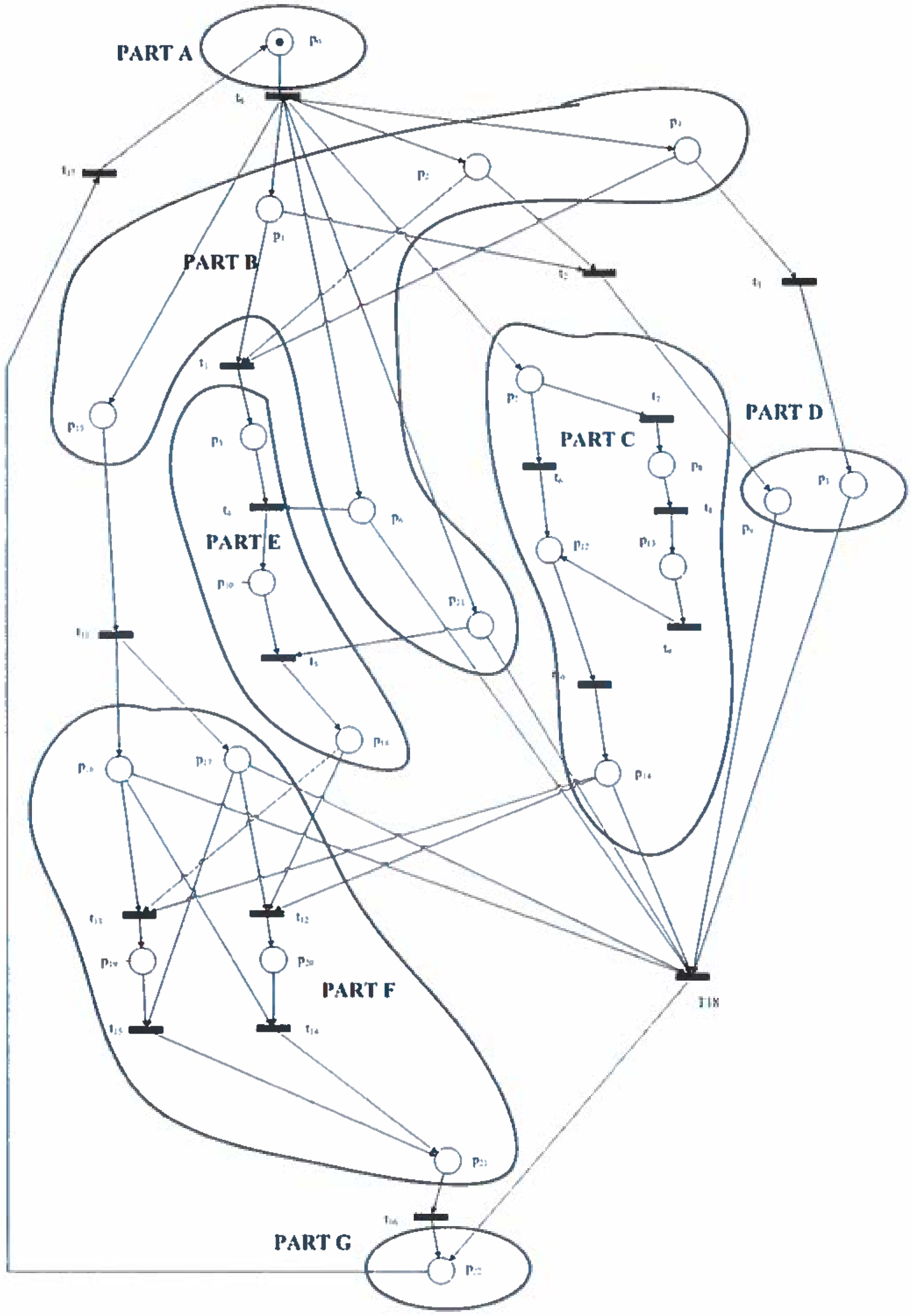

Figure 3.12. Parts of PN model of manual program

3.3.2. Analysing of Manual Program as a PN in PIPE

PIPE is used for analysing PN model of Manual Program which is showed in Figure 3.13. The model is formed in PIPE. There are several modules which are used for analysing such structures in PIPE. In this study only two of them are used.

- State Space Analysis shows that the PN model of Manual Program is "Bounded", "Safe" and "Deadlock Free". Analysis results will be shown in Figure 3.14.

- Reachability / Coverability Graph indicates that PN model of Manual Program is reversible or not. Reachability Graph will be shown in Figure 3.15.

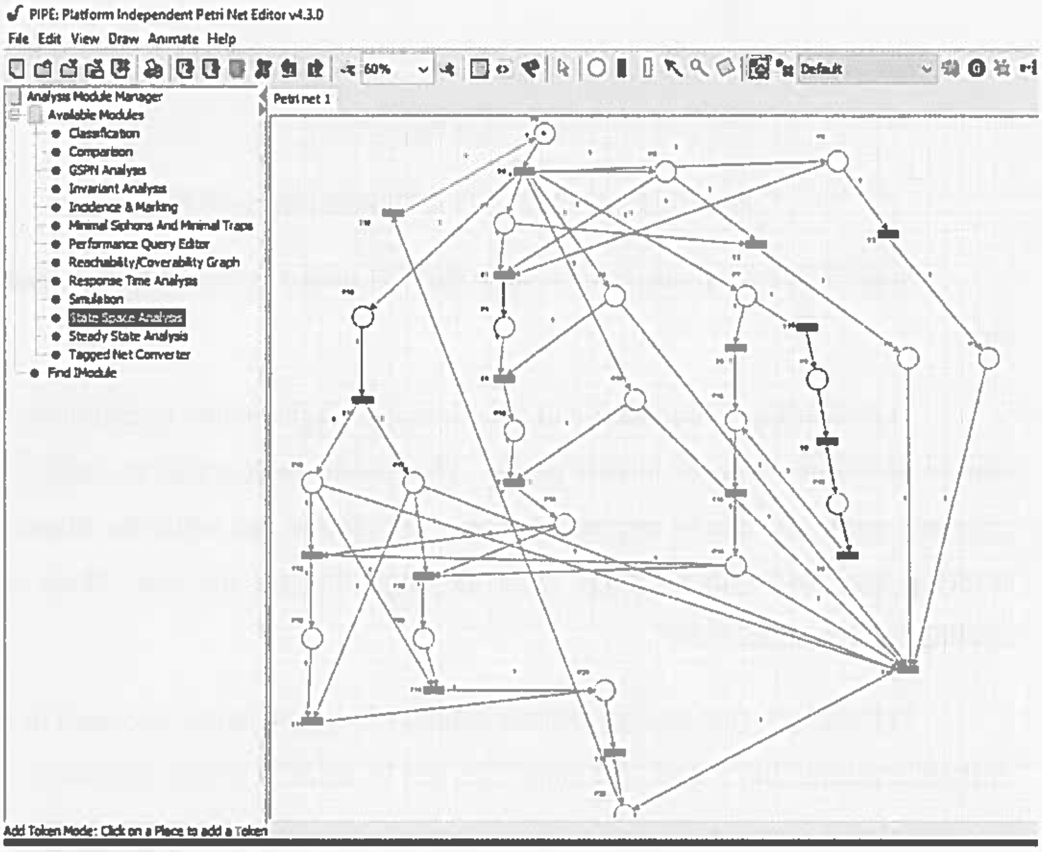

Figure 3.13. PN representation of manual program in PIPE

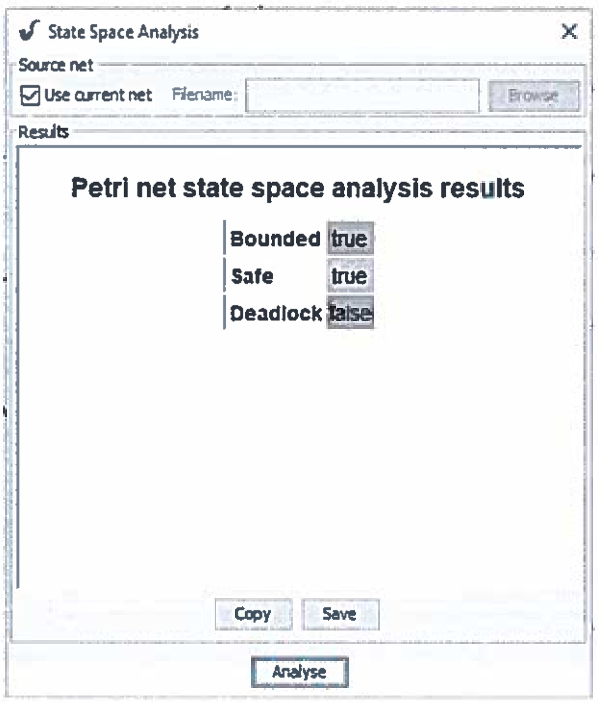

Figure 3.14. Analysis results of manual program in PIPE

State Space Analysis results show that PN model is bounded, safe and deadlock free.

1) According to bounded and safe in every single enable transition is fired there can be only one token in related places. This result implies that in each cycle of the program, each state can be enable only once. This shows that while the Manual Program is being executed with no delay from the beginning till the end. There is no extra cycling burden on the CPU.

2) Deadlock free implies that the Manual Program can be executed in each cycle without stucking into a state. All transitions can be enabled during execution.

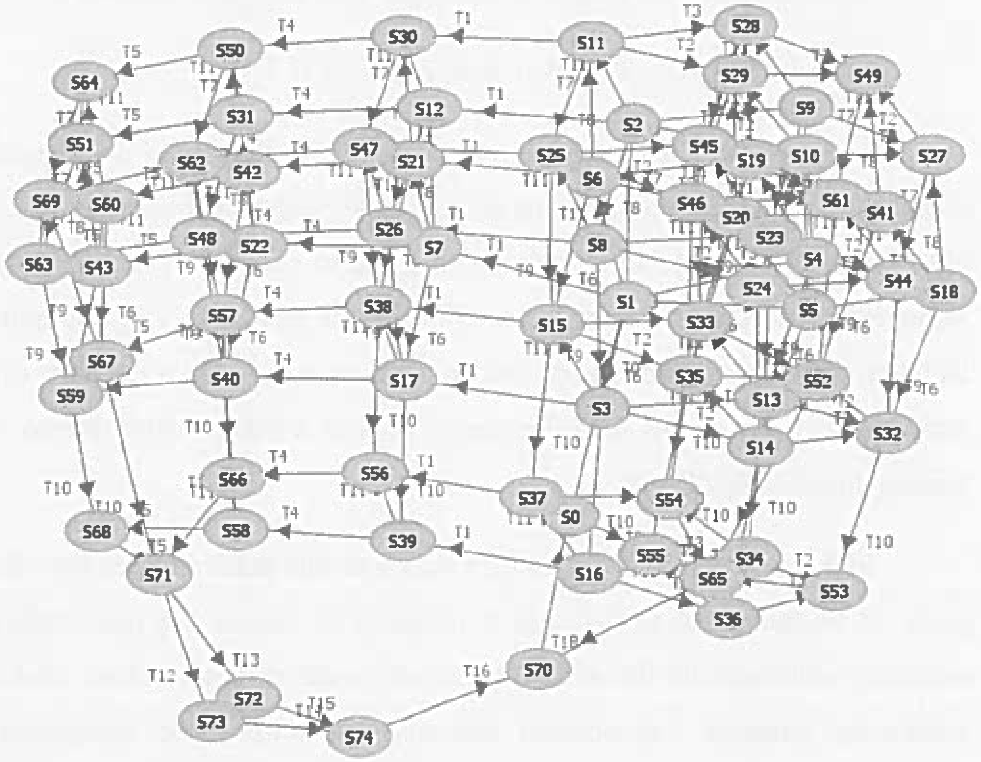

Figure 3.15. Reachability graph of manual program

As can be seen in Figure 3.15., reachability graph is shown. S_0 indicates the initial state of the program which is established according to M marking vector. Other S markings are the possible marking vectors which can be obtained while all possible enabled transitions are fired in order.

$M = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}, p_{21}, p_{22}\}$

$S_0 = \{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$

There are totally 75 possible states until to reach to initial state of Manual Program. It is useless to give all possible tracks to reach to initial state. But some examples can be given to explain the meaning of status because it is obvious from the reachability graph in Figure 3.15. that there are so many states and scenario unable to explain.

So only 2 states are chosen randomly just to be explained. These states are

S_{22} = {0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0}

S_{45} = {0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0}

In S_{22} p_7 has a token which means the HMI values used for Manual program is ready. p_{10} has a token which means driver of the related axis is active which is ready to give 600 V DC voltage to axis motor. p_{11} is also available which represents the axis status which is used to change the enability of the related axis by combining the token which is hold by p_{10}. This means that in order to make enable the axis, driver must be activated and axis status must be ready. p_{15} has a token which means selecting the turning direction is allowed.

In S_{45} p_1 has a token. Having a token in this place implies that the program is ready to be streamed. p_2 has also a token. It is one of the preconditions which is necessary selecting the the status of manual mode. p_3 has a token. That means some emergency situation has occured and operator pushed the emergency buton and Emergency Stop NOT OK status has been activated and is ready to preceed the program to end. p_6 has a token too. This place reprents the driver state. If 3 upper level preconditions are ready which means p_1, p_2 and p_4 have tokens each, by pushing the Manual Mode ON buton (firing t_1), p_5 gets a token. Combining the p_5 and p_6 automatically activates driver by firing t_4. p_{11} is also available which represents the axis status which is used to change the enability of the related axis by combining the token which is hold by p_{10}. p_{13} has a token which means that new screen value has been specified by the operator and this value is ready to send to system. p_{16} and p_{17} both have tokens each. These status show that the related axis is ready to move both direction by only choice of operator.

3.3.3. Findings and Results

State Space and Reachability Graph Analysis indicates that the Manual Program is Bounded, Safe, Deadlock free and Reversible.

Boundedness and Safeness proves that there can be only one token in any places during the one cycle of the program. This showes that there is no delay during the program streaming.

Deadlock free proves that Manual Program does not stuck in any places which forbid to reach to end.

Reversibility also shows that Manual Program can return to its initial status which renew the program cycle.

3.4. Reference Program

Reference Program is assinged to Motion Task. This program is used for finding the home position of related axis which means the zero point or reference point.

Axis motors have encoders which specify the exact position of the motor shaft. In order to specify the position encoder must know its reference point. According to this reference point, when motor takes the moving order from CPU, it goes to exact position.

The purpose of the Reference Program is to help finding the reference point. In order to find the reference point of related axis which is given by the operator, Reference Mode ON must be selected manually by pushing the buton and entered a home position value to screen. Then axis starts to turn. When it reaches to home position, axis stops and produces to outside world which implies that it has found its reference point.

When reference point (home position) is once found, the program streaming ends. Until the axis looses its home position, there is no need to use the Reference Program.

If an emergency situation occurs, operator directly hits the emergency stop buton which produces Emergency Stop NOT OK signal and proceeds the program directly to the end. As can be remembered from the priorities of Execution System, the axis can only be stopped by the Emergency Stop Program because of the mission and priority of the System Interrupt Task which Emergency Stop Program is assinged.

Reference Program is only one mission to comply;

1) Finding the home position of axis which is a reference point of axis encoder for calculating its position.

In Figure 3.16. a flowchart is given, in order to be obtained how this program works basically.

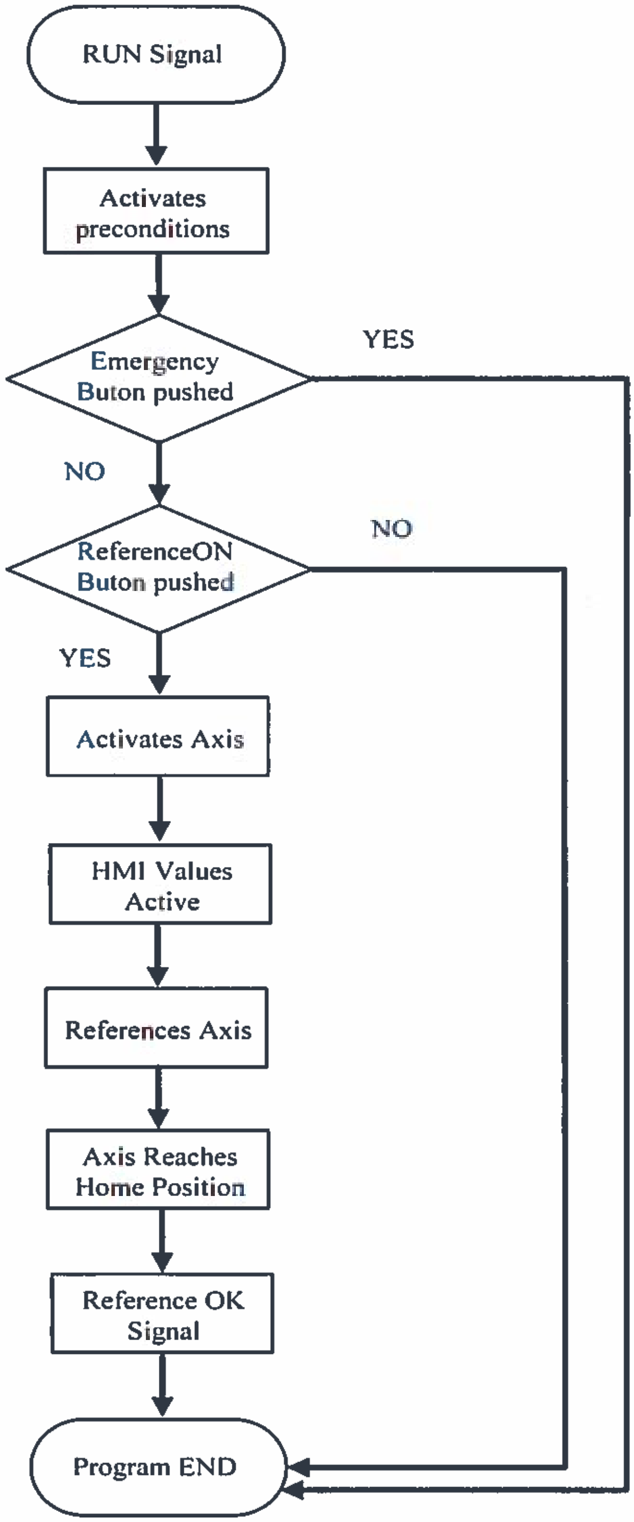

Figure 3.16. Flowchart of reference program

3.4.1. PN Model of the Reference Program

There are 22 places and 18 transitions in PN model of Reference Program. All arcs are weighted 1. The places, the transitions and PN model are seen in Table 3.8., Table 3.9. and Figure 3.17. respectively.

Table 3.8. Places of reference program and their meanings

p_0	Run status. Enables to the start status. Internal marking which is produced by program automatically.
p_1	Start status. Enables to stream the related program. Internal marking which is produced by the program automatically.
p_2	Reference status. Physical input. According to the position of reference switch, it supplies 24 V DC signal to choose ON or OFF status.
p_3	*Emergency Stop NOT OK* status. It is activated by operator by pushing the Emergency Stop buton.
p_4	*Emergency Stop OK* status. It means the 24 V DC control signal reaches to the related input.
p_5	Reference Mode ON status. Just a trespassing condition to proceed the process in to next step. Internal marking.
p_6	Actor drive state monitoring status. Internal marking which is produced by the program automatically.
p_7	Current values on HMI (Human Machine Interface) screen enable status.
p_8	Axis status. Internal marking. Not a physical input.
p_9	Actor drive state monitoring "Active" status which means the driver which drives the axis is ready to run.
p_{10}	Axis enable status which means after enabling the driver to ready to use, it transformed single phase 600 V DC power voltage to three phase 380 V AC voltage to be used for the axis motor in any moment the conditions come true.
p_{11}	New values on HMI screen enable status.
p_{12}	Current "Home Position" on HMI screen is ready to send to system as prerequisites.
p_{13}	Current "Automatic Velocity" on HMI screen is ready to send to system as prerequisites.
p_{14}	New "Home Position" on HMI screen is ready to send to system as current values.
p_{15}	New "Automatic Velocity" on HMI screen is ready to send to system as current values.
p_{16}	Home Position status.
p_{17}	Automatic Velocity status.
p_{18}	Axis reached to home position status.
p_{19}	*Reference OK* status. Internal marking. Not a physical output.
p_{20}	Program end.
p_{21}	Reference Mode OFF. Just a trespassing condition to proceed the process in to next step. Internal marking.

Table 3.9. Transitions of reference program and their meanings

t_0	Activates the p_1, p_2, p_7, p_8 and p_9 status autoatically when the p_0 has a token.
t_1	*Reference ON* buton. Controlled manually by the operator.
t_2	*Reference OFF* buton. Controlled manually by the operator.
t_3	*Emergency* buton. It is activated by operator. When the operator pushes the buton, *Emergency Stop OK* signal is cut off.
t_4	Checks the p_5 and p_6 status. If the conditions come true it is fired automatically to make the p_9 condition is enable. It is used for making Actor Drive Monitoring State is Active.
t_5	Checks the p_7 status. If there is no change on the previous HMI screen value it is fired automatically to next step to activate the p_{12} and p_{13} status.
t_6	Checks the p_7 status. If there will be a change on the HMI screen value, it is fired by the operator's intervenion by touching the related zones of the HMI screen.
t_7	Checks the p_{11} status. If there will be a change on the HMI screen value, it is fired by the operator's intervenion by touching the related zones of the HMI screen.
t_8	Checks the p_8 and p_9 status. If the conditions come true it is fired automatically to make the p_{10} condition is enable. It is used for making the related axis is enable.
t_9	Checks the p_{12} status. If the conditions come true it is fired automatically to make the p_{16} condition is enable. It represents the previous home position is still being used.
t_{10}	Checks the p_{13} status. If the conditions come true it is fired automatically to make the p_{17} condition is enable. It represents the previous automatic velocity is still being used.
t_{11}	Checks the p_{14} status. If the new value on the HMI screen is entered it is fired by the operator by pressing the enter buton to approve manually.
t_{12}	Checks the p_{15} status. If the new value on the HMI screen is entered it is fired by the operator by pressing the enter buton to approve manually.
t_{13}	Checks the p_{10}, p_{16} and p_{17} status. If the conditions come true it is fired automatically to make the p_{18} condition is enable.
t_{14}	Checks the p_{18} status. If the conditions come true it is fired automatically to make the p_{19} condition is enable.
t_{15}	Checks the p_{19} status. If the conditions come true it is fired automatically to make the p_{20} condition is enable.
t_{16}	Checks the p_3, p_6, p_7, p_8, p_{16}, p_{17} and p_{21} status. If the conditions come true it is fired automatically to make the p_{20} condition is enable.
t_{17}	Checks the p_{20} status. If the conditions come true it is fired automatically to make the p_0 condition is enable.

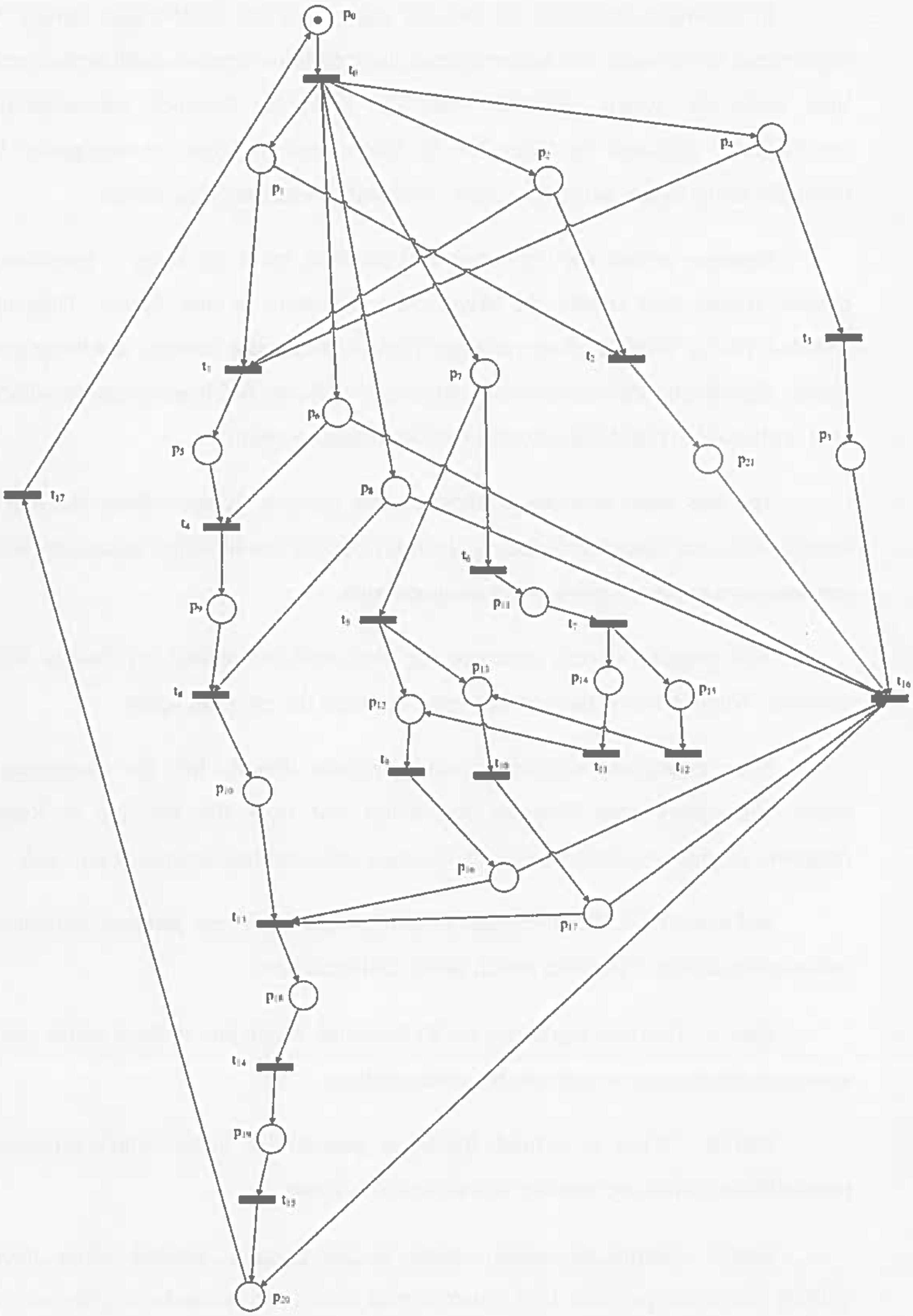

Figure 3.17. PN model of reference program

In Reference Program, all process starts with the RUN signal comes. When RUN status has a token which comes from the Execution System itself for the very first time when the system executes Start Up Task, the bounded preconditions are automatically activated by firing the t_0. These preconditions are emergency buton, reference status buton, actor drive state monitoring status and axis status.

Operator pushes the Reference ON selection buton by firing t_1 transition. The control system first checks the driver state and turns it into Active. Then axis is Enabled. On the other hand operator specifies the automatic velocity and home position values. Combining the axis enabled, automatic velocity and home position values, the axis is allowed to turn to find home position (reference point).

The axis starts to move to find its home position. When it finds that axis stops immediately and system produces a signal for outside world which means that axis has just found its reference point. Then program ends.

This program is only executed for once until the related axis looses its home position. When it is lost, the operator must executes the program again.

If an emergency situation occurs, operator directly hits the emergency stop buton. Emergency Stop Program deactivates and stops the axis but in Reference Program, emergency situation directly by-passes the program streaming into end.

In Figure 3.18., the PN model is seen as aparted. These parts are defined as the submissions in entire program which do the different jobs.

Part A : This part represents the RUN signal. When this status is active t_0 is fired automatically in order to activate the preconditions.

Part B : When t_0 is fired, the token goes to the places which represent the preconditions which are used for butons and axis status.

Part C : Setting the screen values. In this program, needed values automatic velocity and home position. In this part current screen values can be used or new values can be specified by the operator.

Part D : An emergency situation occurs, operator pushes the emergency buton. This part directly by-passes whole activity in the program and proceeds to end.

Part E : Axis status. First the driver status turns into active and then axis becomes enable.

Part F : Reaching the axis into its home position. When the axis reaches to the home position, Reference OK signal is produced.

Part G : Program END status. The program is ready to return to beginning.

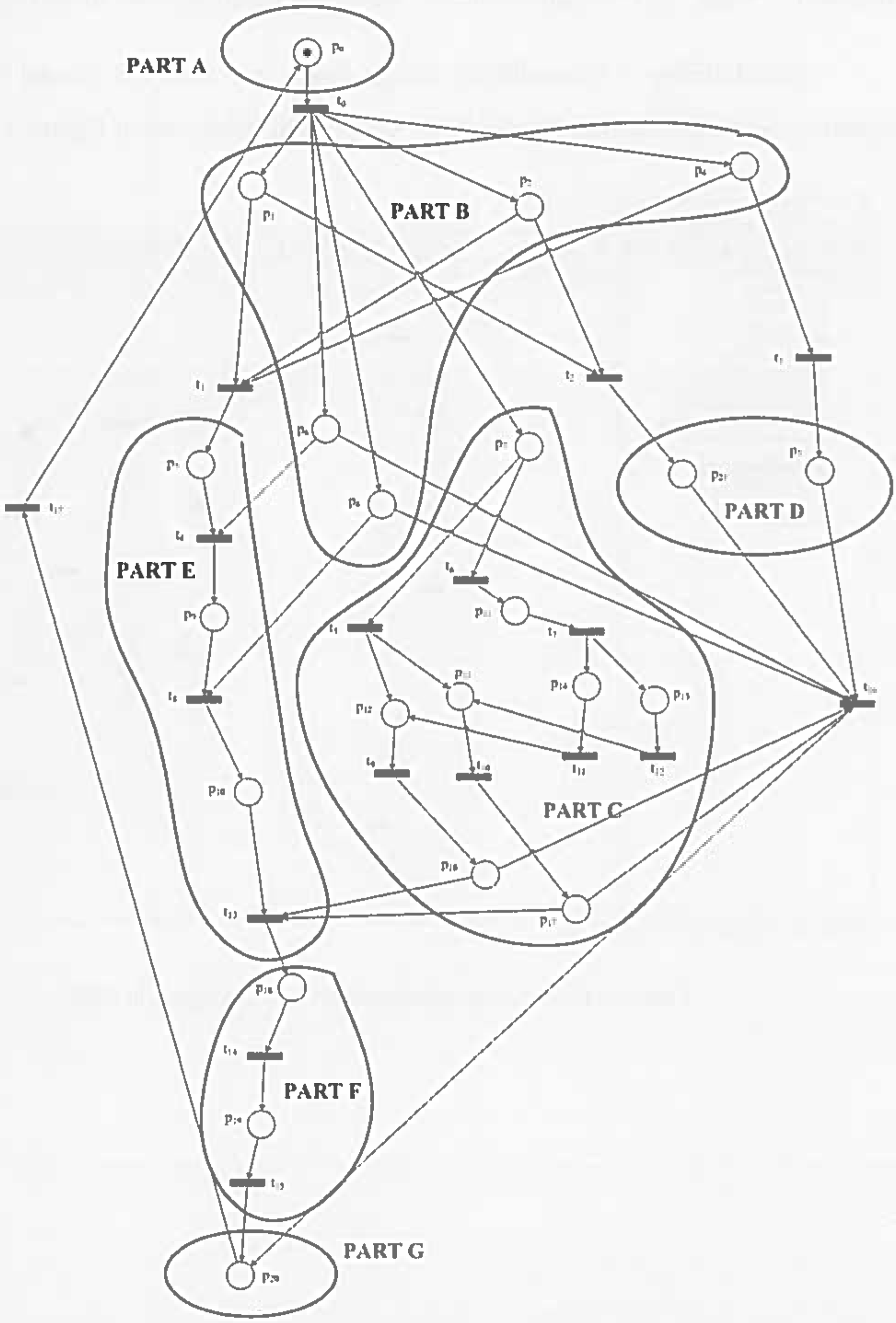

Figure 3.18. Parts of PN model of reference program

3.4.2. Analysing of Reference Program as a PN in PIPE

PIPE is used for analysing PN model of Reference Program which is showed in Figure 3.19. The model is formed in PIPE. There are several modules which are used for analysing such structures in PIPE. In this study only two of them are used.

- State Space Analysis shows that the PN model of Reference Program is "Bounded", "Safe" and "Deadlock Free". Analysis results will be shown in Figure 3.20.

- Reachability / Coverability Graph finds out that PN model of Reference Program is reversible or not. Reachability Graph will be shown in Figure 3.21..

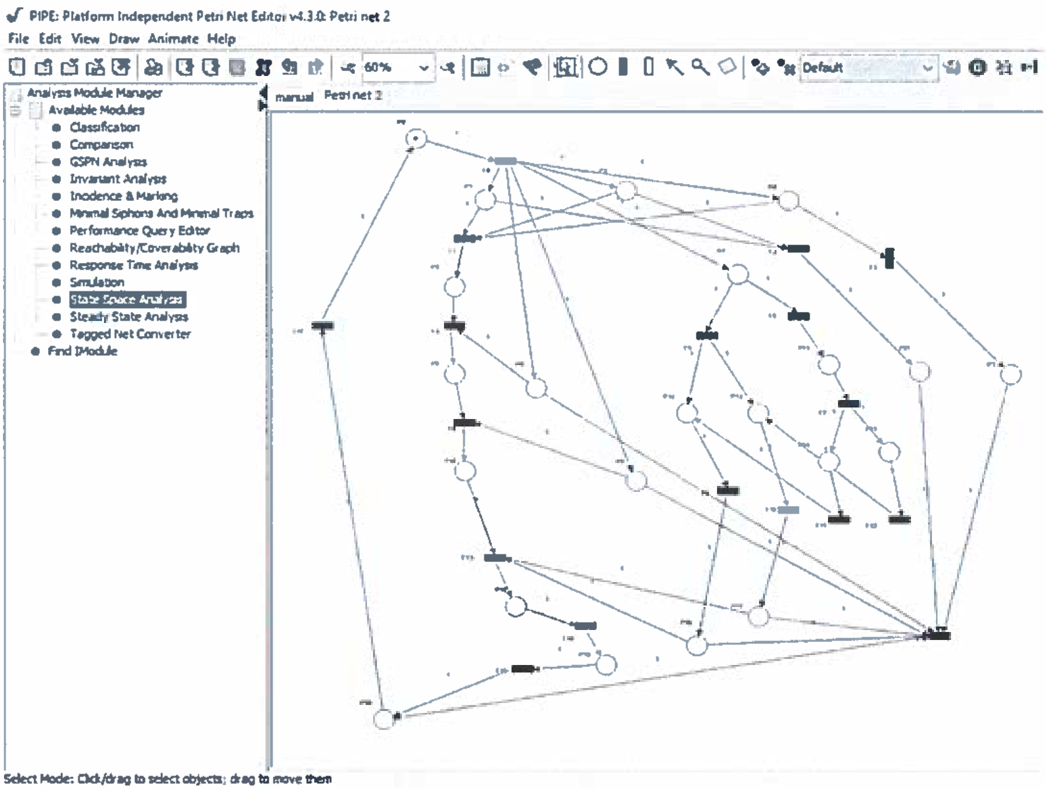

Figure 3.19. PN representation of reference program in PIPE

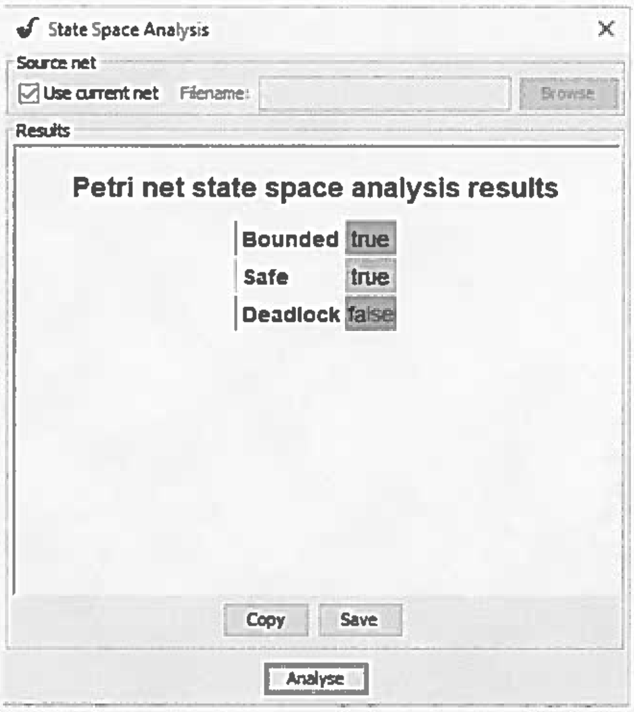

Figure 3.20. Analysis results of reference program in PIPE

State Space Analysis results show that PN model is bounded, safe and deadlock free.

1) According to bounded and safe in every single enable transition is fired there can be only one token in related places. This result implies that in each cycle of the program, each state can be enable only once. This shows that while the Reference Program is being executed with no delay from the beginning till the end. There is no extra cycling burden on the CPU.

2) Deadlock free implies that the Reference Program can be executed in each cycle without stucking into a state. All transitions can be enabled during execution.

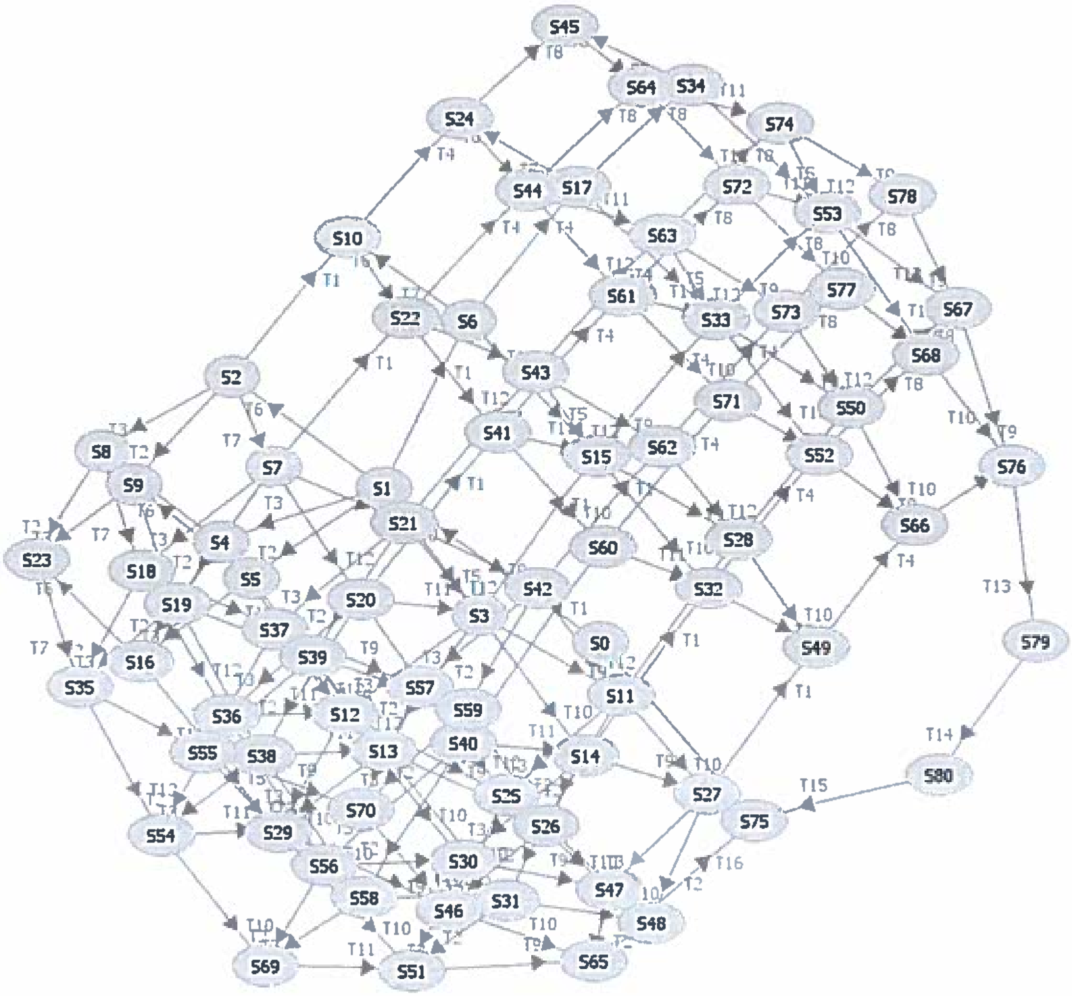

Figure 3.21. Reachability graph of reference program

As can be seen in Figure 3.21., reachability graph is shown. S_0 indicates the initial state of the program which is established according to M marking vector. Other S markings are the possible marking vectors which can be obtained while all possible enabled transitions are fired in order.

$M = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}, p_{21}\}$

$S_0 = \{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$

There are totally 81 possible states until to reach to initial state of Reference Program. It is useless to give all possible tracks to reach to initial state. But some examples can be given to explain the meaning of status because it is obvious from the reachability graph in Figure 3.21. that there are so many states and scenario unable to explain.

So only 2 states are chosen randomly just to be explained. These states are

S_{39} = {0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1}

S_{62} = {0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0}

In S_{39} p_4 has a token which mean Emergency Stop OK status. p_6 has a token which means driver of the related axis is active which is ready to give 600 V DC voltage to axis motor. p_8 has also a token which implies that the axis status which is used to change the enability of the related axis. Having a token in p_{12} means that current home position is used for referencing but a token in p_{15} indicates that new automatic velocity is used. p_{21} has a token which means that the operator chooses the Reference Off mode. So that the program proceeds to end. That situation implies that there is no need to make the axis referenced.

In S_{62} p_5 has a token which means that Reference Mode Selected status. The token in p_6 means that the driver of the related axis is active which is ready to give 600 V DC voltage to axis motor. p_8 has also a token which implies that the axis status which is used to change the enability of the related axis. p_{15} indicates that new automatic velocity is used. p_{16} has a token which means home position status which is used in the system.

3.4.3. Findings and Results

State Space and Reachability Graph Analysis indicates that the Reference Program is Bounded, Safe, Deadlock free and Reversible.

Boundedness and Safeness proves that there can be only one token in any places during the one cycle of the program. This showes that there is no delay during the program streaming.

Deadlock free proves that Reference Program does not stuck in any places which forbid to reach to end.

Reversibility also shows that Reference Program can return to its initial status which renew the program cycle.

3.5. Automatic Program

Automatic Program is the last one which is introduced in this study. Automatic Program is assigned to Motion Task. Automatic program carries out the main objective of LTM. This program is used while the automatic mode is selected during mass production.

In mass production session the glassware products perpetually streams from the previous working station to laser marker. As to be told in section 3.1. there are two conveyor lines on the machine each of them dedicated to a positioner axis.

After glassware products seperated to each conveyor one by one, they are get caught by a photcell sensor which is located just before positioner. When this sensor sees a product it produces a signal to positioner to turn with a specific angle which is decided by the operator.

Anytime the products are caught by the photocell sensor, positioner turns and brings the products under laser marker.

If an emergency situation occurs, operator directly hits the emergency stop buton which produces Emergency Stop NOT OK signal and proceeds the program directly to the end. As can be remembered from the priorities of Execution System, the axis can only be stopped by the Emergency Stop Program because of the mission and priority of the System Interrupt Task which Emergency Stop Program is assinged.

Automatic Program is only one mission to comply;

1) Bringing the glassware products under laser marker during mass production process.

In Figure 3.22. a flowchart is given, in order to be obtained how this program works basically.

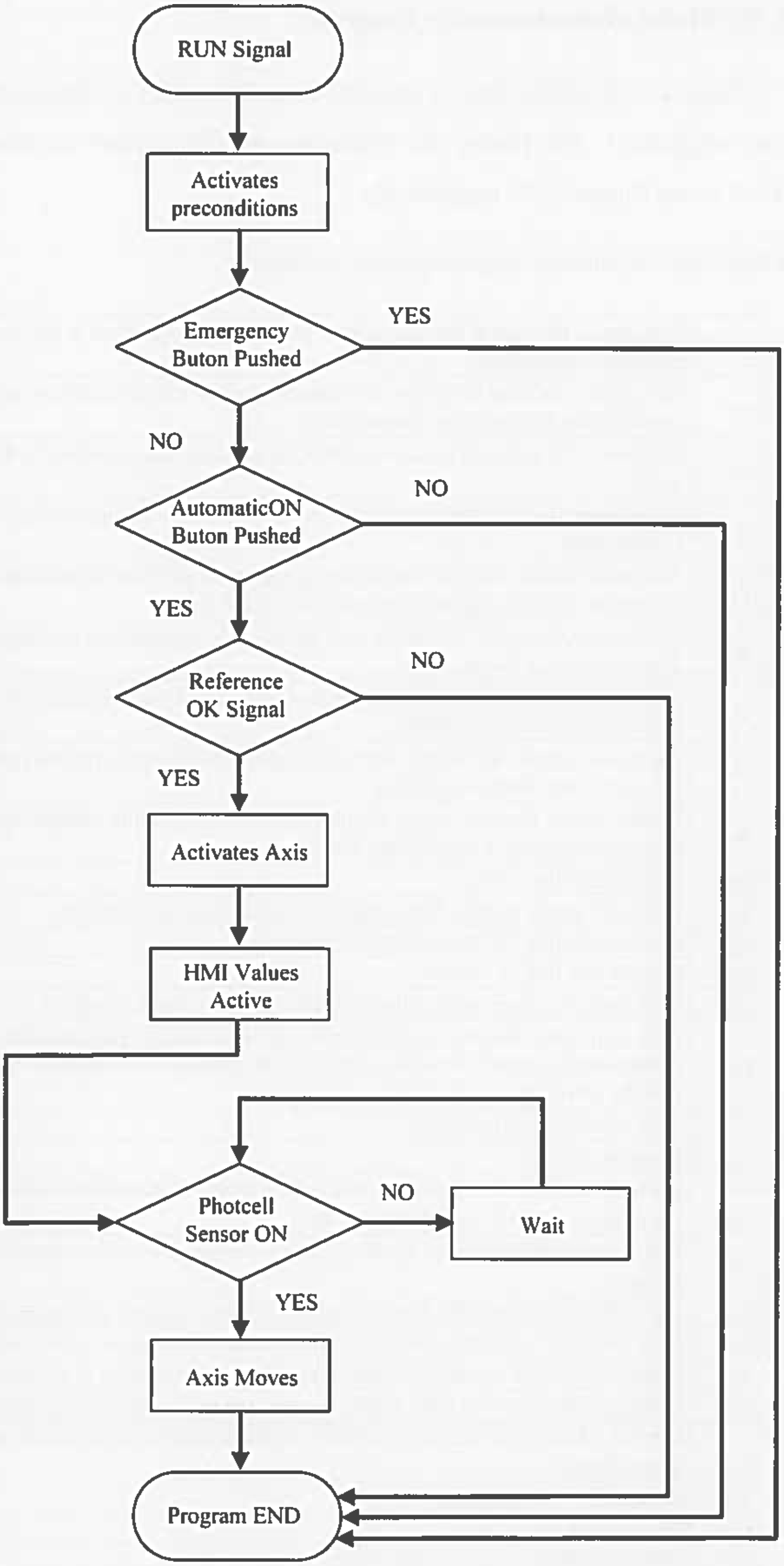

Figure 3.22. Flowchart of automatic program

3.5.1. PN Model of the Automatic Program

There are 28 places and 21 transitions in PN model of Automatic Program. All arcs are weighted 1. The places, the transitions and PN model are seen in Table 3.10. Table 3.11. and Figure 3.23. respectively.

Table 3.10. Places of automatic program and their meanings

p_0	Run status. Enables to the start status. Internal marking which is produced by program automatically.
p_1	Start status. Enables to stream the related program. Internal marking which is produced by the program automatically.
p_2	*Reference OK* status. It means the 24 V DC control signal reaches to the related input.
p_3	*Emergency Stop OK* status. It means the 24 V DC control signal reaches to the related input.
p_4	Automatic status. Physical input. According to the position of automatic switch, it supplies 24 VDC signal to choose ON or OFF status.
p_5	*Emergency Stop NOT OK* status. It is activated by operator by pushing the *Emergency Stop Buton*.
p_6	Automatic Mode ON status. Just a trespassing condition to proceed the process in to next step. Internal marking.
p_7	Automatic Mode OFF status. Just a trespassing condition to proceed the process in to next step. Internal marking.
p_8	Fotosel status. Physical input. Supplies 24 V DC signal into related input to sense there is something in front of the sensor.
p_9	Axis switch flag status.
p_{10}	Photocell sensor enable. Not a physical output. Internal marking.
p_{11}	Axis switch flag "0" status. Internal marking.
p_{12}	Axis switch flag "1" status.
p_{13}	Axis ready to move status. Internal marking. Not a physical output.
p_{14}	Axis stop status. Internal marking. Not a physical output. Expresses the specified movement has ended. Axis moved to specific position with specific velocitiy and specific direction.
p_{15}	Axis is being in move status.
p_{16}	Program end.
p_{17}	Current values on HMI (Human Machine Interface) screen enable status.
p_{18}	New values on HMI screen enable status.
p_{19}	New "Automatic Velocity" on HMI screen is ready to send to system as current values.
p_{20}	New "Direction" on HMI screen is ready to send to system as current values.
p_{21}	New "Position" on HMI screen is ready to send to system as current values.
p_{22}	Current "Position" on HMI screen is ready to send to system as prerequisites.
p_{23}	Current "Direction" on HMI screen is ready to send to system as prerequisites.
p_{24}	Current "Automatic Velocity" on HMI screen is ready to send to system as prerequisites.
p_{25}	Automatic Velocity status.
p_{26}	Direction status.
p_{27}	Position status.

Table 3.11. Transitions of automatic program and their meanings

t_0	Activates the p_1, p_2, p_3, p_4, p_8, p_9 and p_{17} status autoatically when the p_0 has a token.
t_1	*Emergency Buton*. It is activated by operator. When the operator pushes the buton, *Emergency Stop OK* signal is cut off
t_2	Checks the p_{16} status. If that place has a token it is fired automatically to return to run status.
t_3	Checks the p_{24} status. If there is a token in that place it is fired automatically as the HMI value to be sent to system.
t_4	*Automatic ON Buton*. Controlled manually by the operator.
t_5	*Automatic OFF Buton*. Controlled manually by the operator.
t_6	Checks the p_{23} status. If there is a token in that place it is fired automatically as the HMI value to be sent to system.
t_7	Checks the p_{22} status. If there is a token in that place it is fired automatically as the HMI value to be sent to system.
t_8	Checks the p_{19} status. If the new value on the HMI screen is entered it is fired by the operator by pressing the enter buton to approve manually.
t_9	Checks the p_5, p_7, p_8, p_{25}, p_{26}, p_{27}, p_{11} and p_{12} status. It is fired automatically when these events come true to enable the program end status.
t_{10}	Photocell sensor switch on when it sees something in front of it.
t_{11}	Checks the p_9 status. It is fired automatically to activate both p_{11} and p_{12}.
t_{12}	Checks the p_{20} status. If the new value on the HMI screen is entered it is fired by the operator by pressing the enter buton to approve manually.
t_{13}	Checks the p_{10} and p_{11} status. It is fired automatically to activate p_{13}.
t_{14}	Checks the p_{12} and p_{15} status. It is fired automatically to activate p_{14}.
t_{15}	Checks the p_{14} status. It is fired automatically to activate p_{16}.
t_{16}	Checks the p_{13}, p_{25},p_{26} and p_{27} status. It is fired automatically to activate p_{15}.
t_{17}	Checks the p_{17} status. If there will be a change on the HMI screen value, it is fired by the operator's intervenion by touching the related zones of the HMI screen.
t_{18}	Checks the p_{18} status. If there will be a change on the HMI screen value, it is fired by the operator's intervenion by touching the related zones of the HMI screen.
t_{19}	Checks the p_{17} status. If there is no change on the previous HMI screen value it is fired automatically to next step to activate the p_{22}, p_{23} and p_{24} status.
t_{20}	Checks the p_{21} status. If the new value on the HMI screen is entered it is fired by the operator by pressing the enter buton to approve manually.

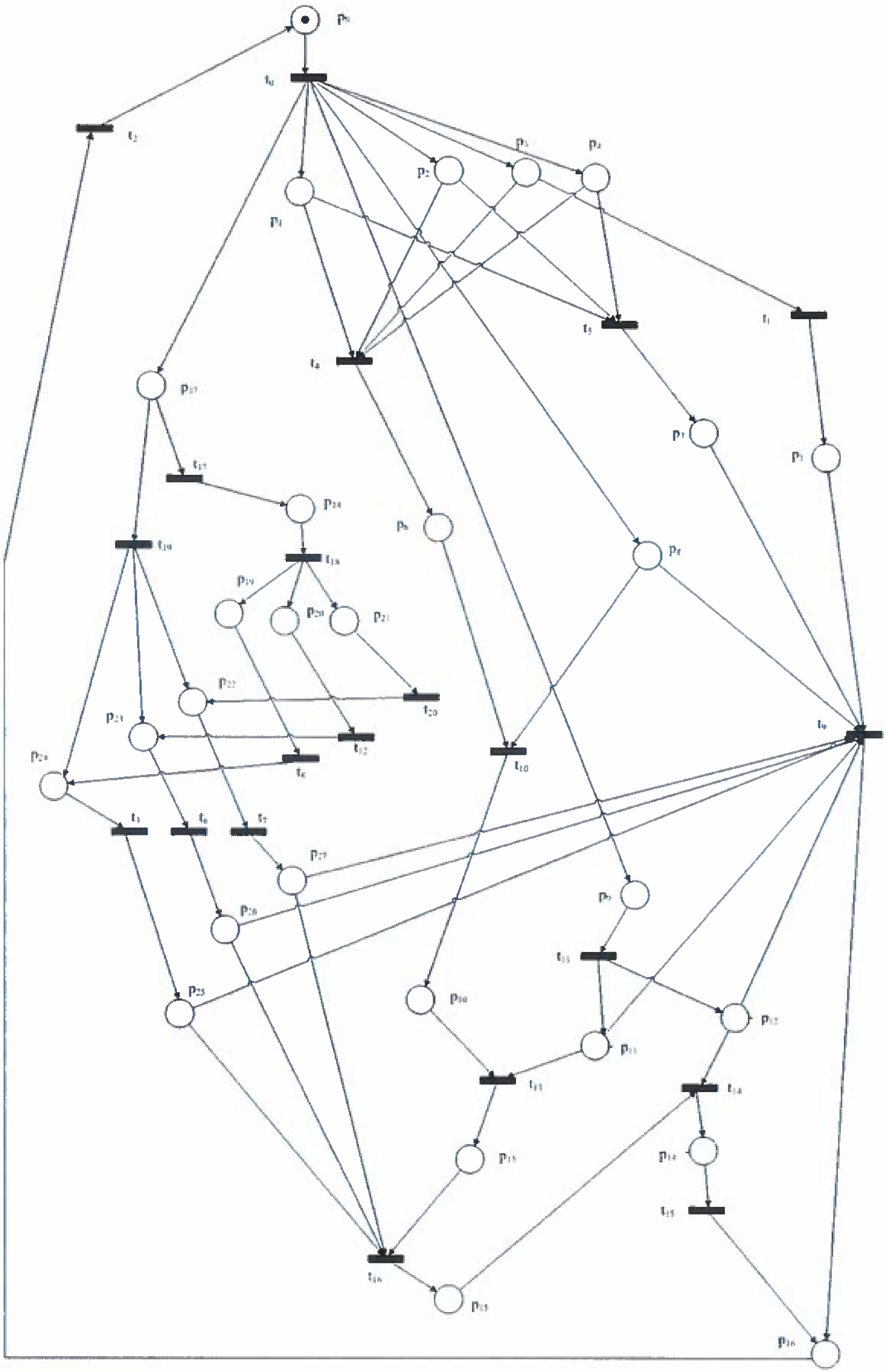

Figure 3.23. PN model of automatic program

In Automatic Program, all process starts with the RUN signal comes. When RUN status has a token which comes from the Execution System itself for the very first time when the system executes Start Up Task, the bounded preconditions are automatically activated by firing the t_0. These preconditions are emergency buton, automatic status buton, reference status, axis switch flag status and photocell sensor status.

In Automatic Program, all process starts with the RUN signal comes. When RUN status has a token which comes from the Execution System itself for the very first time when the system executes Start Up Task, the bounded preconditions are automatically activated by firing the t_0. These preconditions are emergency buton, reference ok status, automatic mode status, photocell sensor status and axis switch flag status.

When operator selects the Automatic Mode ON by firing t_4 system produces Automatic Mode Selected signal. This is not a physical signal (24 V DC) but just a flag which is used internally. Meanwhile photocell sensor is already ready but waits the products come. When it senses a product, it produces a signal which means sensor is enable. Combining the enability of sensor and axis switch flag status, the axis gets ready to move but it only waits the automatic velocity, direction and position values which come from screen values.

So that axis starts to move. This is only an instantaneous moment to bring the product under laser machine. So this is very important to cut the axis is on move signal just after enabling the move signal. When axis stops t_{15} is automaticaly fired and program ends.

If an emergency situation occurs, operator directly hits the emergency stop buton. Emergency Stop Program deactivates and stops the axis but in Automatic Program, emergency situation directly by-passes the program streaming into end.

In Figure 3.24., the PN model is seen as aparted. These parts are defined as the submissions in entire program which do the different jobs.

Part A : This part represents the RUN signal. When this status is active t_0 is fired automatically in order to activate the preconditions.

Part B : When t_0 is fired, the token goes to the places which represent the preconditions which are used for butons and photcell sensor status.

Part C : Setting the screen values. In this program, needed values automatic velocity, direction and position. In this part current screen values can be used or new values can be specified by the operator.

Part D : An emergency situation occurs, operator pushes the emergency buton. This part directly by-passes whole activity in the program and proceeds to end.

Part E : When the glassware product is sensed by the photcell sensor, it urges the axis to move. The positioner brings the product under laser marker and stops.

Part F : Program END status. The program is ready to return to beginning.

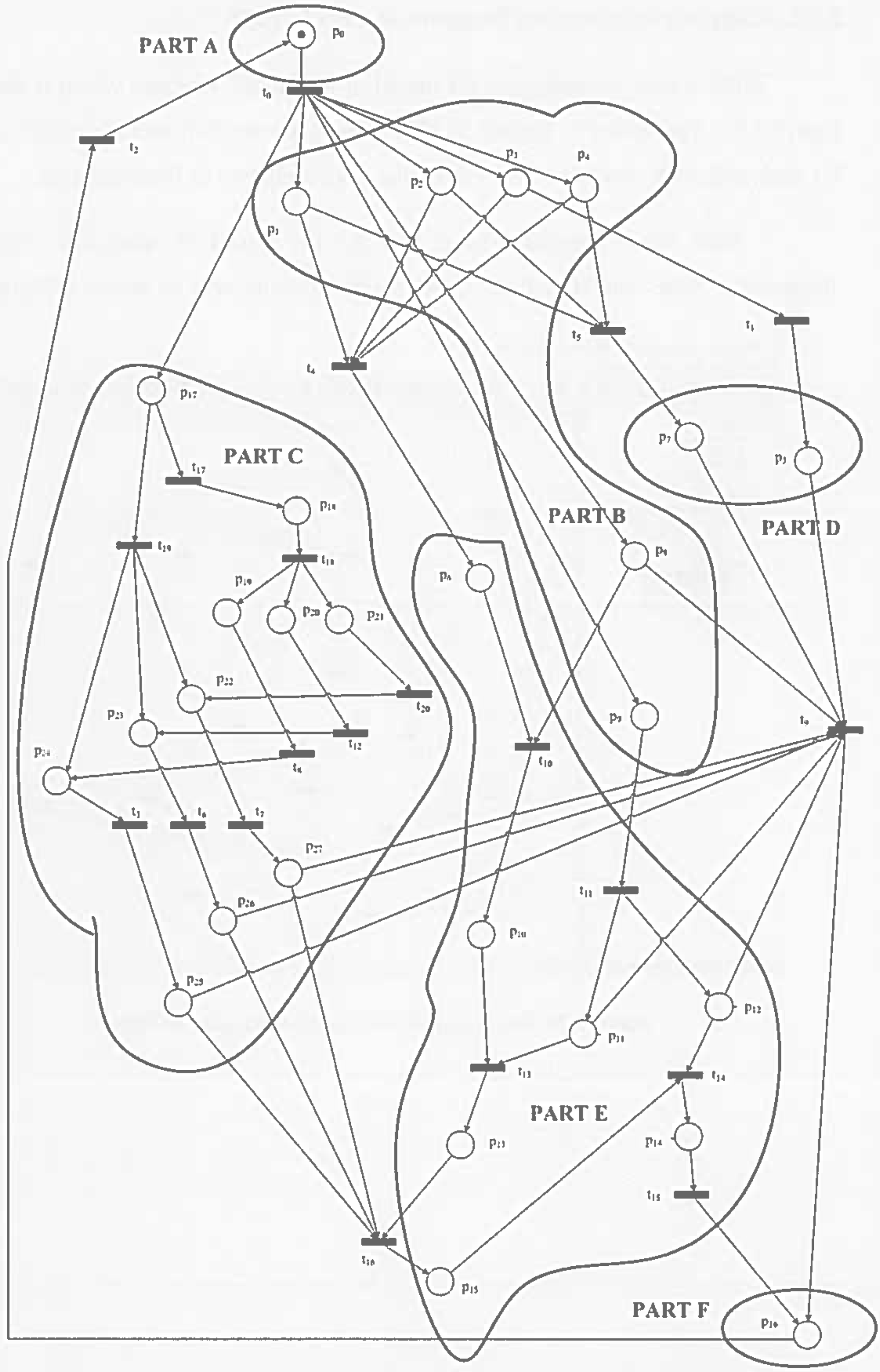

Figure 3.24. Parts of PN model of automatic program

3.5.2. Analysing of Automatic Program as a PN in PIPE

PIPE is used for analysing PN model of Automatic Program which is showed in Figure 3.25. The model is formed in PIPE. There are several modules which are used for analysing such structures in PIPE. In this study only two of them are used.

- State Space Analysis shows that the PN model of Automatic Program is "Bounded", "Safe" and "Deadlock Free". Analysis results will be shown in Figure 3.26.

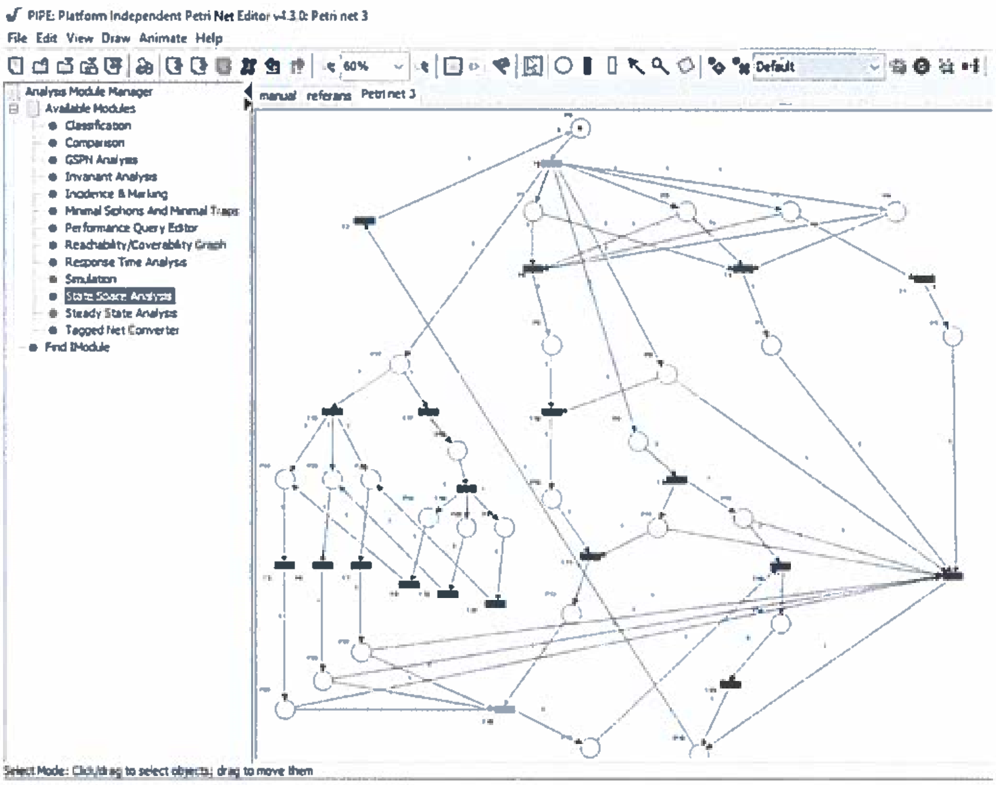

Figure 3.25. PN representation of automatic program in PIPE

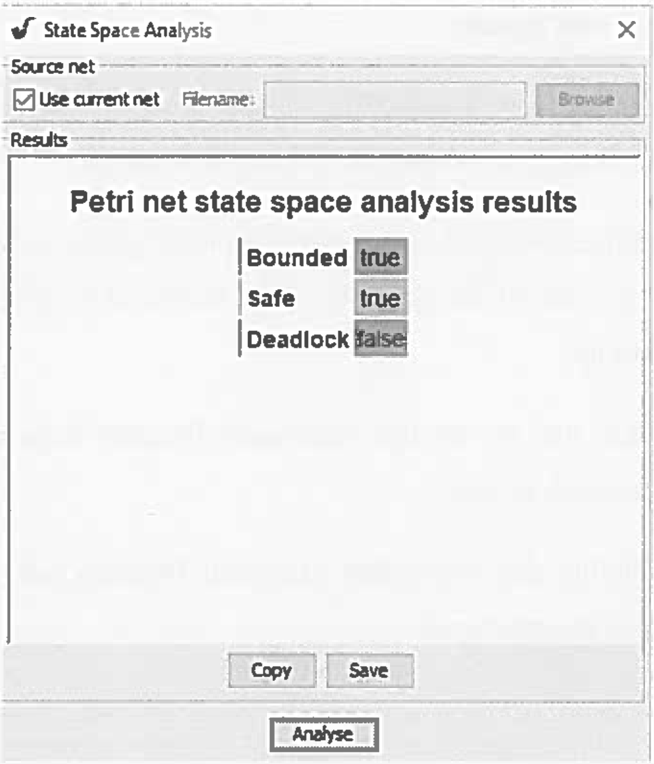

Figure 3.26. Analysis results of automatic program in PIPE

State Space Analysis results show that PN model is bounded, safe and deadlock free.

1) According to bounded and safe in every single enable transition is fired there can be only one token in related places. This result implies that in each cycle of the program, each state can be enable only once. This shows that while the Automatic Program is being executed with no delay from the beginning till the end. There is no extra cycling burden on the CPU.

2) Deadlock free implies that the Automatic Program can be executed in each cycle without stucking into a state. All transitions can be enabled during execution.

3) Rechability Graph Analysis was also made but because of the technical incapabilities of PIPE Reachability Graph could not be displayed. The reason of the matter is the number of states and arcs. There are 381 states and 1298 arcs in the Reachability Graph of Automatic Program which PIPE does not allow to display.

3.5.3. Findings and Results

State Space and Reachability Graph Analysis indicates that the Automatic Program is Bounded, Safe, Deadlock free and Reversible.

Boundedness and Safeness proves that there can be only one token in any places during the one cycle of the program. This showes that there is no delay during the program streaming.

Deadlock free proves that Automatic Program does not stuck in any places which forbid to reach to end.

Reversibility also shows that Automatic Program can return to its initial status which renew the program cycle.

4. CONCLUSIONS

The improvements in the machine building technology keeps going with perpertual speed. Robustness of position controlled systems are getting involved more and more to manufacturing plants. Programming the automation softwares are important as much as the quality and capabilities of equipment.

In this study Laser Transfer Machine which was built on the concept of position control was tried to model and analysis by using PNs. PNs are quite suitable for modelling and analysing such systems because the structure of programs which are the main concern of this study are event based. Causality and the relationship between the inputs and outputs of the system prepared a suitable environment to use PNs in LTM.

PN analysis gave valuable informations about the system. A servo position control program can be modelled by using PN. Some of behavioral characteristics were tested. In each program cycle before reaching to end, it was seen that only one action was done by trespassing the conditions. This showed that the entire programs are safe and there is no delay in program cycle. Deadlock free showed that each program are not stuck in any places in the program cycle hence programs do not lock. Finally reachability graphs showed that in program cycle to all possible places can be reached and in the end program can return to its initial state which means whole programs are reversible.

This study is not a new approach for automation projects. PNs are being used for modelling many other automation projects. In this thesis, motion controlled automation programs are modelled and analysed by using PNs. In future perspective this approach may be used for developing a control method in such kind of projects.

5. REFERENCES

[1] Raisch, J. (2009). Discrete Event And Hybrid Systems. Dublin: Trinity College

[2] Murata, T. (1989). Petri Nets: Properties, Analysis And Applications. Proceedings Of The IEEE, 77(4), 541 – 543, 547 – 548

[3] Cassandras, C.G. and Lafortune, S. (2008). Introduction To Discrete Event Systems. New York: Springer

[4] Giua, A. (1992). Petri Nets As Discrete Event Models For Supervisory Control. New York: Rensselaer Polytechnic Institute

[5] Bogdan, S., Lewis, F.L., Kovacic, Z. and Mireles, J. (2006). Manufacturing Systems Control Design. New York: Springer

[6] Liu, Y. (2009). A Petri Net – Based Ladder Logic Diagram Design Method for the Logic and Sequence Control of Photo Mask Transport. ICIC 2009, LNCS 5754, 774 – 783

[7] İçmez, Y. (2015). Robotik Montaj Sistemlerinin Petri Ağı Temelli Kontrolü. Yayımlanmamış Yüksek Lisans Tezi. Tokat: Gaziosmanpaşa Üniversitesi, Fen Bilimleri Enstitüsü

[8] Behmeranrezai, O. (2019). Petri Ağları Yaklaşımı İle Raylı Sistemler Anklaşman Sistemlerinin PLC İle Gerçeklenmesi Ve Simülasyonu. Yayımlanmamış Yüksek Lisans Tezi. İstanbul: İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü

[9] https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRXhhbXBsZXNfb2ZfTWFya292X2NoYWlucw

[10] https://polen.itu.edu.tr/bitstream/11527/4953/1/2824.pdf

ABOUT AUTHOR

Name Surname : Selçuk ERGİNEL

Languages : Turkish (Native), English (Advance), Russian (Intermediate)

Birth Place & Date : Eskişehir / TURKEY, 30.07.1987

Email : ergnlslck@gmail.com

Education : Yıldız Technical University – Electrical Engineering (2010)

Job Experiences

Şişecam A.Ş. – Electrical and Automation Engineer (2017 - Present) (TURKEY)

Bridgestone Sabancı A.Ş. – Maintenance Engineer (2016 – 2017) (TURKEY)

Eti Gıda A.Ş. – Electrical Engineer (2014 – 2016) (TURKEY)

Vitra Plitka OOO. – Maintenance Chief (2012 – 2013) (RUSSIA)

EEE. A.Ş. – Electrical Engineer (2011 – 2012) (TURKEY)