ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN AND RAMS ANALYSIS OF RAILWAY INTERLOCKING SYSTEMS USING FORMAL METHODS

M.Sc. THESIS

Mustafa BELLEK

Department of Electrical Engineering

Electrical Engineering Programme

DECEMBER 2013

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN AND RAMS ANALYSIS OF RAILWAY INTERLOCKING BASED ON FORMAL METHODS: AN EXAMPLE APPLICATION

M.Sc. THESIS

Mustafa BELLEK

(504111031)

Department of Electrical Engineering

Electrical Engineering Programme

Thesis Advisor: Prof. Dr. Ömer USTA

DECEMBER 2013

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DEMİRYOLU ANKLAŞMAN SİSTEMLERİNİN FORMAL YÖNTEMLER İLE DİZAYNI VE RAMS ANALİZİ: ÖRNEK UYGULAMA

YÜKSEK LİSANS TEZİ

Mustafa BELLEK
(504111031)

Elektrik Mühendisliği Anabilim Dalı
Elektrik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Ömer USTA

ARALIK 2013

Mustafa Bellek, a **M.Sc.** student of ITU **Graduate School of Science Engineering and Technology** student ID **504111031**, successfully defended the **thesis** entitled "**DESIGN AND RAMS ANALYSIS OF RAILWAY INTERLOCKING BASED ON FORMAL METHODS: AN EXAMPLE APPLICATION**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Ömer USTA** İstanbul Technical Universitv	.....
Co-advisor :	**Prof. Dr. M. Turan SÖYLEMEZ** İstanbul Technical Universitv	.....
Jury Members :	**Prof. Dr. Mustafa BAĞRIYANIK** İstanbul Technical University	.....
	Asst. Prof. Özgür T. KAYMAKÇI Yıldız Technical University	.....
	Asst. Prof. İlker Üstoğlu Yıldız Technical Universitv	.....

Date of Submission : 16 December 2013

Date of Defense : 07 February 2013

To my family and friends,

FOREWORD

This report is a continued study of my master thesis [1] written during my study in Technische Universität Dresden, Faculty of Transportation and Traffic Sciences (Fakultät Verkehrswissenschaften "Friedrich List") as an exchange student. The content of the thesis reexamined with considering Turkish signalling methodology. Furthermore, the original content has been prepared in cooperation with Thales Transportation Systems, Germany.

I would like to express my sincere gratitude to Prof. Dr.-Ing. Jochen Trinckauf of TU Dresden for giving me an opportunity to work on this topic. My special thanks to Dr.-Ing. Ulrich Maschek (TU Dresden) for his contribution on my work. I would also like to thank my advisors in ITU, Prof. Dr. Ömer Usta and Prof. Dr. M. Turan Söylemez for their valuable comments and suggestions.

I am very grateful to M. Sc. Qamar Mahboob (TU Dresden) for providing me guidance, resources and supports. I am particularly grateful to my tutor in Thales Transportation Systems, Germany, Dipl.-Ing. Thomas Heinig for sharing his knowledge in railway signalling and continuous help during my research. I also wish to thank Dr.-Ing. Enrico Anders (Thales Transportation Systems, Germany) for his valuable suggestions and fruitful discussions.

I would like to thank Thales Transportation Systems, Germany for providing the financial support during my thesis study.

My special thanks to all my friends in Dresden, Stuttgart and Istanbul.

I am very grateful to Asst. Prof. Deniz YILDIRIM and Prof. Dr. M. Ertuğrul Çelebi for their invaluable favors.

Finally, I would like to express my gratitude to my parents for their support and belief in me.

December 2013

Mustafa BELLEK

Electrical Engineer

TABLE OF CONTENTS

ABBREVIATIONS

PLC	**:** Programmable Logic Controller
FSM	**:** Finite State Machine
RAMS	**:** Reliability, Availability, Maintainability, Safety
Ks	**:** Kombinationssignal (combination signal)
NX	**:** Entrance-Exit Route Setting Method
TCC	**:** Train Control Center
SSI	**:** Solid State Interlocking
CBI	**:** Computer Based Interlocking
PDF	**:** Probability Density Function
SFC	**:** Sequential Function Chart
MTTF	**:** Mean Time to Failure
MTBF	**:** Mean Between to Failures
MTTR	**:** Mean Time to Repair
FTA	**:** Fault-tree Analysis
FMEA	**:** Failure Modes and Effect Analysis
HAZOP	**:** Hazard and Operability Analysis
PHA	**:** Preliminary Hazard Analysis

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

∨ **:** Logical "OR"
∧ **:** Logical "AND"
! **:** Logical "NOT"
$\overline{x}$ **:** Logical inverse of *x*
T#1s **:** Timer defined for 1 second
λ **:** Hazard rate
μ **:** Repair rate

DESIGN AND RAMS ANALYSIS OF RAILWAY INTERLOCKING BASED ON FORMAL METHODS: AN EXAMPLE APPLICATION

SUMMARY

In this thesis study, design and implementation of an example railway interlocking mechanism with formal methods is aimed. German "Ks" signal system is considered as the signalling principle for designed simple interlocking. However, all features of the Ks system are not considered for the purpose of simplification of the study. All basic terms and equipment used in railway signalling are defined in the first chapter. Then, the features of "Ks" signalling system and Turkish signalling system are explained in detail.

In the third chapter, definition of the interlocking is given and the functionality of the interlocking in railways is explained. Most of the definitions in third chapter are excerpted from reference number 2.

In the fourth chapter, formal methods that are also used for designing interlocking system are explained. Then, two widely used formal methods, "Petri Nets" and "Finite State Machines" are discussed. Model of a simple turnstile device is given as an example to show design steps of finite state machines method. Afterwards, two different implementation software are examined with advantages and disadvantages. In the end of the chapter, implementation of example given before is achieved with both programming tools.

In fifth chapter, a model railway station is created. All types operational specifications and characteristics are defined for the model station that includes train types, line types and others. Then, positioning of the signalling equipment on the model station is discussed.

In "Example Interlocking Design" part, the route table of the model station is generated and a route setting mechanism is designed with using finite state machines method. Firstly, control unit of all wayside equipment are modelled and implemented. Afterwards, some basic route setting functions according to route setting rules are modelled with the same method. Finally, the route setting mechanism for the first route defined in the route table is created with developed models. Then, it is implemented with PLC programming software, SilworX, and tested with the same software.

The RAMS analyses are presented in chapter 7. Basic definitions of RAMS are explained and two mostly used methods in RAMS analysis, "Fault Tree Analysis" and "Markov Model" are explained with detailed examples. Finally, a Markov model is created for the model station which is designed in fifth chapter and equations used for RAMS calculations are obtained. The RAMS parameters are estimated.

Final chapter presents results and conclusion of the thesis work. Designed example interlocking and the future works are discussed in this chapter.

DEMİRYOLU ANKLAŞMAN SİSTEMLERİNİN FORMAL YÖNTEMLER İLE DİZAYNI VE RAMS ANALİZİ: ÖRNEK UYGULAMA

ÖZET

Demiryolu sinyalizasyon sistemleri trenlerin güvenli, planlı ve ekonomik bir şekilde işletilmesini sağlayan sistemlerdir.

Geleneksel demiryolu araçları raylar üzerinde çelik ray – çelik tekerlek yöntemi ile yol alırlar. Bu yöntem sayesinde çelik ray ile çelik tekerlek arasındaki sürtünme kuvveti azaltılarak yuvarlanma direnci düşürülmüş olur. Böylelikle trenlerin hareket etmesi için harcanan enerjiden tasarruf edilmiş olur. Fakat bu durum başka bir problemi de beraberinde getirir; Frenleme problemi. Raylar ve tekerlekler arasındaki düşük sürtünme kuvveti fren mesafesinin, makinistlerin görüş mesafesinden daha uzun olmasına neden olur. Bu nedenle trenlerin duruş noktalarından belirli bir mesafe öncesinde fren uygulamaları gerekmektedir. Demiryolu sinyalizasyon sistemlerinin temel amaçlarından biriside fren mesafesini hesaba katarak trenlerin hareket güvenliğini sağlamaktır.

Demiryollarında çeşitli amaçlarla çeşitli cihazlar kullanılır. Örneğin makaslar rayların bağlantısını değiştirerek trenlerin bir raydan başka bir raya geçmesi için kullanılır. Trenler gitmesi gereken güzergâhlarda ilerlerken çok sayıda makasın üzerinden geçerler ve tüm bu makasların güzergâha uygun pozisyona ayarlanmış olması gerekir. Sinyalizasyon sistemleri makas gibi demiryolu cihazların güvenlik kriterleri çerçevesinde otomatik olarak kontrol eder ve güvenliliği garanti eder. Sistemde bu gibi saha ekipmanlarının kontrolü ve güvenli pozisyonda kilitlenmesi işlevleri yerine getiren mekanizma "Anklaşman" olarak adlandırılır.

Anklaşman sistemleri, trenlerin güvenli hareket edebilmesi için demiryollarında kullanılan saha ekipmanlarının uygun ve güvenli durumda kilitlenmesini sağlayan sinyalizasyon sistemlerinin temel bileşenidir. Bu tez çalışmasında örnek bir demiryolu anklaşman mekanizmasının formal yöntemler ile tasarlanması ve uygulanması amaçlanmıştır. Tasarlanan basit anklaşman sistemi için dizayn kriteri olarak Alman "Ks" sinyal sistemi dikkate alınmıştır. Fakat çalışmayı basitleştirmek amacı ile Ks sisteminin tüm özellikleri kapsanmamıştır.

Birinci bölümde genel manada sinyalizasyon sisteminin ve güvenlik kriterlerinin demiryollarındaki önemi istatistiki bilgilerle anlatılmıştır.

İkinci bölümde, demiryolu sinyalizasyon sistemlerinin yapısı ve bu sistemlere neden ihtiyaç duyulduğu açıklanmıştır. Daha sonra sinyalizasyon sistemlerinde kullanılan temel bileşenler ve makas, sinyal lambası, aks sayıcı, vs. gibi temel saha ekipmanları açıklanmıştır.

Farklı ülkeler farklı sinyalizasyon prensiplerine sahiptir. İkinci bölümün devamında Alman Ks sinyal sisteminde ve Türk sinyal sisteminde kullanılan sinyalizasyon

prensipleri tanımlanmıştır. Her iki sistemde kullanılan sinyal lambaları kullanım yerleri ve anlamları ile açıklanmıştır.

Üçüncü bölümde anklaşman terimi açıklandıktan sonra demiryollarındaki karşılığı anlatılmıştır. İlk kullanılan mekanik sistemlerinden günümüzde kullanılan bilgisayar tabanlı modern sistemlere kadar kullanılan farklı yapılardaki anklaşman sistemleri üçüncü bölümde işlenmiştir.

Sinyalizasyon sistemlerinde oluşabilecek her hangi bir hata, trenlerin raydan çıkması veya başka trenler ile çarpışması gibi ölümcül sonuçlar doğuracak ciddi tren kazalarına sebep olabilir. Bu nedenle sinyalizasyon sistemleri tasarlanırken sistemin çalışması esnasında oluşabilecek tüm arızalar düşünülerek bu gibi arıza durumlarında sistemin güvenli duruma geçmesi sağlanır. Hatada güvenlilik şeklinde tanımlanan bu prensip üçüncü bölümde örneklerle açıklanmıştır.

Anklaşman sistemleri tasarlanırken bir takım temel prensipler dikkate alınır. Üçüncü bölümde bu tasarım prensiplerinden bir kısmı, 2 numaralı kaynaktan faydalanılarak açıklanmıştır.

Dördüncü bölümde anklaşman sistemlerinin tasarlanmasında kullanılan formal yöntemler açıklanmıştır. Daha sonra yaygın olarak kullanılan iki yöntem "Petri Ağları" ve "Sonlu Durum Makinaları" tartışılmıştır. Sonlu durum makinaları yönteminin tasarım basamaklarını göstermek amacı ile basit bir turnike cihazının modellenmesi örnek olarak verilmiştir.

Dördüncü bölümün devamında, tasarlanacak modelleri gerçeklemek ve test etmek için iki farklı PLC programlama yazılımı avantaj ve dezavantajları ile incelenmiştir. Ardından, daha önce verilen basit örnek model her iki programlama yazılımıyla da gerçeklenmiştir. İleriki bölümlerde tasarlanacak modeller için kullanılacak olan SilworX yazılımının neden tercih edildiği aynı bölümün sonunda açıklanmıştır.

Beşinci bölümde bir model demiryolu istasyonu tasarlanmıştır. Tasarlanan model istasyon için hat tipleri ve tren tipleri ve tüm işletme karakteristikleri tanımlanmıştır. Daha sonra sinyalizasyon ekipmanlarının konumlandırılması tartışılmıştır.

Altıncı bölümde model istasyon için olası tüm tren güzergâhlarını gösteren bir güzergâh tablosu oluşturulmuştur. Bu tablo anklaşman tasarlanan bölgedeki güzergâhların hangi saha ekipmanlarını kullandığı ve bu saha ekipmanlarının durumunun ne olması gerektiğini gösterir.

Altıncı bölümün devamında sonlu durum makinaları yöntemi kullanılarak ikinci bölümde açıklanan hat boyu ekipmanlarının modelleri oluşturulmuş ve PLC programlama yazılımı SilworX ile gerçeklenmiştir. Daha sonra aynı yöntemle güzergâh tablosu dikkate alınarak bazı güzergah tayin etme fonksiyonları modellenmiştir. Son olarak tasarlanan modeller ile güzergah tablosundaki ilk güzergah için tayin etme mekanizması oluşturulmuştur. Daha sonra bu mekanizma SilworX yazılımı ile gerçeklenmiş ve test edilmiştir.

Bölüm 7'de sistem tasarımında dikkat edilmesi gereken "Güvenilirlik, Emre amadelik, Sürdürülebilirlik ve Güvenlik" kriterleri işlenmiştir. RAMS kriterleri olarak ifade edilen bu kriterlerin hesaplanması ve analizinde yaygın olarak kullanılan iki adet yöntem "Hata Ağacı Yöntemi" ve "Markov Modeli" aynı bölümde açıklanmıştır. Son olarak beşinci bölümde oluşturulan model istasyon için bir Markov modeli tasarlanmış ve bu model ile RAMS analizinde kullanılan denklemler elde edilmiştir. Bu bölümün sonunda RAMS parametreleri elde edilmiştir.

Tez çalışmasında ulaşılan sonuçlar son bölümde gösterilmiştir. Ayrıca bu bölümde tasarlanan anklaşman sistemi ve gelecekte yapılabilecekler tartışılmıştır.

1. INTRODUCTION

Railway transportation is a major form of passenger and freight transport in many countries. People prefer rail transport for their daily journeys and intercity travels. Due to the fact that the rail transportation is safe, fast, easily reachable and comfortable. [2]

Despite of high safety, fatal accidents are still occurring in modern railways [3]. For example, in 2011, there were 2325 persons killed or seriously injured in railway accidents in Europe [4]. Table 1.1 shows the number of persons killed and injured by those accidents in 2011 [4].

Table 1.1 : Number of persons killed and injured by type of accident in Europa [4]

	Number of Persons											
	Killed				Seriously Injured				Total			
	Passengers	Employees	Other	Total	Passengers	Employees	Other	Total	Passengers	Employees	Other	Total
Collisions	9	3	3	15	33	11	5	49	42	14	8	64
Derailments	2	2	0	4	43	2	0	45	45	4	0	49
Accidents involving level crossing	6	0	311	317	24	14	291	329	30	14	602	646
Accidents to persons caused by rolling stock in motion	22	25	856	903	123	36	453	612	145	61	1309	1515
Fires in rolling stock	0	0	0	0	0	0	0	0	0	0	0	0
Others	0	1	2	3	6	20	22	48	6	21	24	51
Total	39	31	1172	1242	229	83	771	1083	268	114	1943	2325

Signalling systems play the most important role in railway safety. Main purpose of the signalling systems is to prevent derailments and collusions between trains. The second objective is to manage the railway traffic and increase the operation capacity.

In railways, several equipment and devices, also called “wayside” or “lineside” equipment, are used for different purposes. All this equipment and devices have to be proper position before permitting a train movement to ensure a safe operation. Signalling system guarantees the safety with locking wayside equipment with each other. This internal locking activity is called “interlocking”.

Furthermore, a failure in the signalling systems can cause serious consequences and any dangerous failure is unacceptable. Whereas, any device or equipment cannot be fully reliable in the real world. For that reason, almost every equipment and devices are produced with respect to fail-safe criteria in railway signalling systems. Fail-safe is a design criteria used to design a device, which may cause some dangerous consequences in the system when it fails. A Fail-safe device guarantees to be system in safe state when a failure in system occurs. Therefore, the safety of the system is ensured.

In modern railway signalling systems, interlocking function is provided by programmable electronic devices such as microprocessor, industrial computer or PLC. These devices are called “interlocking unit”. The software in the interlocking unit has to be developed with special methods to obtain high safety levels. According to the European Standard EN 61508, formal methods can be used to develop an interlocking algorithm.

Formal methods are a kind of mathematical based design techniques for specification, development and verification of software systems. They play an important role in increasing the completeness, consistency or correctness of a specification or implementation because formal methods transfer the principles of mathematical reasoning to the specification and implementation of technical systems [5].

On the other hand, high safety level is not the only essential requirement of the signalling systems. Besides, signalling system must have a certain level of reliability, availability and maintainability rate. All these rates are called RAMS (reliability, availability, maintainability and safety) rates. RAMS is defined to indicate the quality and working performance of the signalling system.

The intent of this thesis is to examine how to design and implement an example railway interlocking system with using formal methods. For that purpose, the general features and characteristics of the modern railway signalling systems will be examined in first

chapters of thesis. Afterwards, the formal methods will be discussed with all steps. Finally, an example interlocking will be designed and implemented for a model railway station with formal methods. German Ks system has been considered as the signalling principle in this study. Because, the most part of the thesis are completed in Germany.

In chapter 7, two widely used methods which used to calculate RAMS parameters of the signalling system will be examined. Then, a simple RAMS analysis will be handled for the model station designed before. Application of fault tree and Markov model to railway risk, safety and reliability is referred to [6] and [7].

2. BASICS OF RAILWAY SIGNALLING

2.1 General Description

Railway vehicles have some different characteristics from other land transportation vehicles. If it is compared with road vehicles; the mass of a train is very high, acceleration and deceleration rates are low and stopping distant is relatively long. A railway vehicle cannot stop safely when an obstacle or another vehicle seen on the way. A train running full speed at a curvy track can be given as an example. Because of the restricted visibility, driver cannot see if there is another vehicle waiting on the same track. Therefore, driver has to be informed in advance with a **movement authority** which guarantees there isn't any other vehicle on the path. Railway signalling system gives the moving authority to driver [8].

On the other hand, there are several equipment and devices used in railways for various purposes such as point machine. It is also required to monitor and control these equipment to ensure they are in correct state and working without failure. All equipment and devices have to be **failure-free**, because any failure occurred in them can lead collision or derailment. Safety is the main purpose of the railway signalling system.

Furthermore, signalling system also increases the **operation capacity**. Because it sets automatically the train's path which wanted to proceed on and allows trains to travel at maximum speed is allowable by the characteristics of the line. Then, the number of journey per day can be offered more frequent and that makes possible to use railway line more efficiently.

To sum up, basic functional principle of railway signalling system can be defined as; it monitors all vehicles on tracks, checks and sets the wayside equipment and gives to trains movement authority to ensure the safety and operational quality.

2.2 Train Control Center

Train control center (TCC) is the monitoring and management office of a railway signalling system. Almost all central equipment of the signalling system are placed in TCC. Figure 2.1 shows a TCC.

Figure 2.1: A Train control center (TCC) and Dispatcher [9]

The person who is responsible to manage the whole railway traffic is called Dispatcher. Dispatcher monitors the traffic flows and gives related commands to signalling system to control it. The interface between signalling system and dispatcher provided by a computer called operator tool. This computer shows the map of whole line controlled by signalling system and accept signalling control commands such as; point control, route setting or route blocking. A sample dispatcher screen can be seen in Figure 2.2.

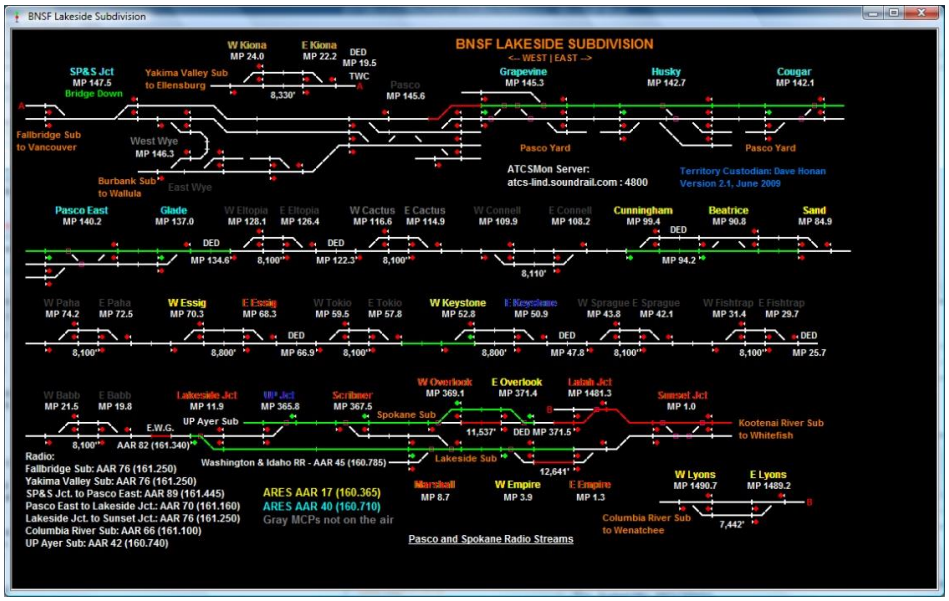

Figure 2.2 : A Sample Dispatcher Screen [10]

2.3 Wayside Equipment

As mentioned in the description of signalling, there are some basic lineside equipment for various purpose in the railways. In this chapter most using lineside equipment are explained.

Signalling systems must be designed to be fail-safe. This means that the failure of any equipment or subsystem must result in a default state which ensures safety in all circumstances. Systems and equipment are therefore designed, manufactured, installed and maintained with safety criteria. The term of fail-safe will be explained in next chapters.

2.3.1 Point machines

Railway vehicles proceed on guided ways called track. The purpose of points is to provide mechanical connection between tracks. It is a movable track element and it makes possible to change existing track of a train with another track according to its position.

Positions of a point are defined as “Normal” and “Reverse” (or “Straight” and “Divergent”). Normal position means the train will continue on the same track. Conversely, if a point in reverse position, that means the train running over it will leave the existing track and pass another track. The third position can be defined as “Intermediate” to indicate the point's condition when it is moving. It is a transition condition between normal and reverse position. Figure 2.3 shows the basic structure of a point.

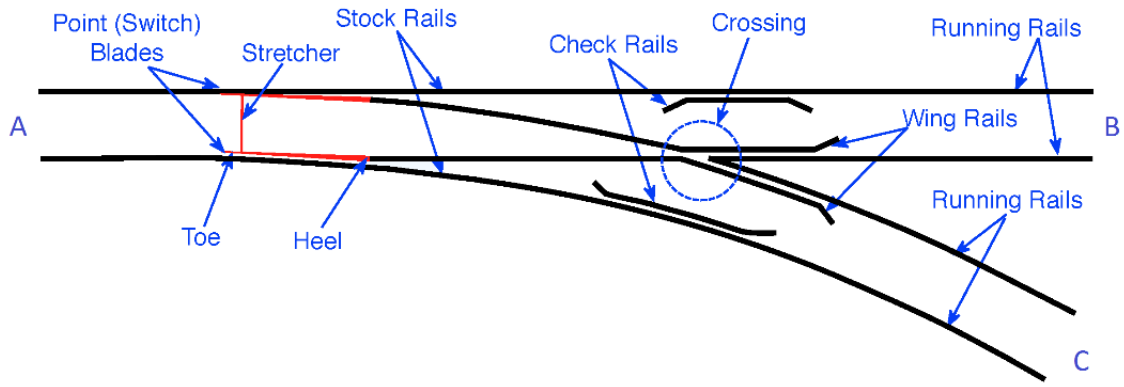

Figure 2.3 : A railway point [11]

Train movements from “A” to “B” or “A” to “C” in the figure called facing movements. These movements are arranged with the point position. On the other hand, a movement from “B” to “A” or “C” to “A” is called trailing move. If the point is in wrong position in a trailing move, the point blades are forced to move to correct position by the wheel flanges of the train. This is the trailing action of the point. Some type of points have a blade locking mechanism and they cannot be trailed. Therefore, wrong blade position of this type of points can cause a derailment.

The movement of the point is provided by the **point machine**. Point machine is an active device for using to control the positioning of a point. There are also position sensors inside the point control mechanism to ensure the actual position of the point. These sensors detect the position of the point blades and provide a feedback to the signalling system continuously.

Railway signalling system monitors and controls the point via position sensors and point machine.

2.3.1.1 Simple point

Simple point is the basic type of points. It has only two end positions: normal and reverse. Figure 2.4 shows a simple point. It is the most used type of point around the world.

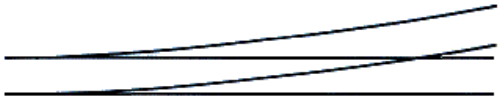

Figure 2.4 : A Simple Point [12]

The trains has to obey a speed restriction when they pass over a point in reverse position. Because point in reverse position is a curvy path and the trains cannot proceed with full speed at curve. The speed restriction is one of the feature of a point. If the radius of a point is large, then trains can pass over it faster. Radius of the points determines the characteristic of the line.

2.3.1.2 Diamond crossing

Diamond Crossing is used for the crossing of two tracks (Figure 2.5). It is not a movable track element but passing over a diamond crossing has to be controlled to prevent any collision.

Figure 2.5 : A Diamond Crossing [12]

2.3.1.3 Slip point

Slip point is a combined form of diamond crossing and simple point. Two types of slip points are used. The first one is single slip point and the other one is double slip point. The differences between two types of slip point can be seen in the following Figure 2.6 and Figure 2.7.

Figure 2.6 : A Single Slip Point. Possible paths: A->B, A->D, C->B [12]

Figure 2.7 : A Double Slip Point. Possible paths: A->B, A->D, C->B, C->D [12]

2.3.1.4 Double point

Double point is used to split a track into three divergent paths. Its structure is more complicated. The only advantage of a double point is it is required small installing

area. Therefore, it is usually only used in a station or depot where space is restricted. It also called “three-way-points”. Possible paths can be seen in Figure 2.8.

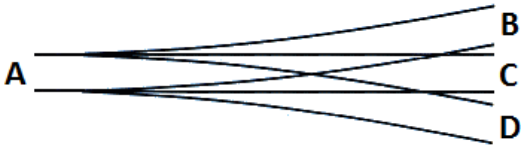

Figure 2.8 : A Double Point. Possible paths: A->B, A->C, A->D [12]

2.3.2 Signals

Signals are the basic equipment provide an interface between technical devices and people. In railway signalling systems signals are used for conveying information from the system to the train driver or workers on the track. The mechanical signals called “Semaphore” were used in the railway signalling in the past but light signals is preferred now. Figure 2.9 shows the general appearance of two type signals.

Figure 2.9 : Sample Railway Signals. Left: Light Signal, right: Semaphore Signal [13]

Most generally conveyed information can be listed as follows:

- Movement authority
- Permitted speed
- Information about the direction of the route
- Position of points
- Commands for brake test [railway signalling and interlocking]

In this study, only signals which used for movement authority and speed restriction are encompassed. The types of the signals will be described in the next topic. Considered types of signals are mostly used types but, there might be some other signal forms for other purposes.

2.3.2.1 Main signal

A main signal is a basic signal controls a train movement along a running line. These signals indicate if the train has to stop or is allowed to continue until the next main signal.

2.3.2.2 Distant signal

When the train driver sees that the main signal shows stop, it may not be possible to stop before passing it because of the long brake distance. Therefore, train driver is informed in advance about the next main signal's aspect. The function of distant signal is fulfill these purpose. The aim of the distant signal is to enable the driver to decelerate in time. Almost every main signal is preceded by a distant signal. In general, it gives two information; "next main signal shows proceed" or "next main signal shows stop".

2.3.2.3 Speed restriction signal

In some part of the railway line, trains aren't allowed to proceed full speed. The geometry of the track or a point in reverse position on the path can be given as some reasons for speed restriction. The train driver can get the information of speed limits with following speed restriction signals (or speed indicators) on the wayside. Speed indicators are mostly located with the main signal or the distant signal. It uses the numbers to indicate the speed limits. If it does not indicate any number (dark), that means there is no speed restriction and the train can proceed with full speed. Generally, the last digit of the speed limit isn't shown and it is always assumed that it is zero. For instance, if speed indicator shows 8, that means the speed limit is 80 km/h.

2.3.2.4 Shunting signal

Movement of trains in a depot or siding is very slow, so the provision of a main signal in that kind of area is not appropriate. In depot area or a vehicle parking area it might be required to do a coupling operation between coaches. Therefore, proceed aspect of

a shunting signal does not means the path is clear. For that reason, another color (usually white) is used for showing proceed in shunting signals.

2.3.3 Track clear detection

Location of every railway vehicles on the track has to be known by signalling system. Following points are the main purpose of track clear detection:

- Before permitting a train movement track clearance has to be confirmed.
- Switching a moveable track element when there is a vehicle over it is very dangerous. For a safe control of moveable track elements, system has to know the occupancy information on the certain area.

Detection of the train's location is achieved by several technics and devices. Mostly used technic is dividing the track to several sections and checking there is an occupancy in these sections. It is a discrete detection and it is provide the system there is an occupancy in the section. However, it is not possible to know where the train exactly in the section is. Track circuit and axle counter system detect the occupancy section by section. The technologies behind them will expressed in next section.

2.3.3.1 Track circuits

Today, most common ways to determine whether a track section is occupied by use of a track circuit. There several type of track circuits based different technics but the oldest and simplest type is the classical track circuits. Its working principle is based on short circuit principle between two rails formed by wheelset of trains in a section. Figure 2.10 and Figure 2.11 illustrate the working principle of the track circuit.

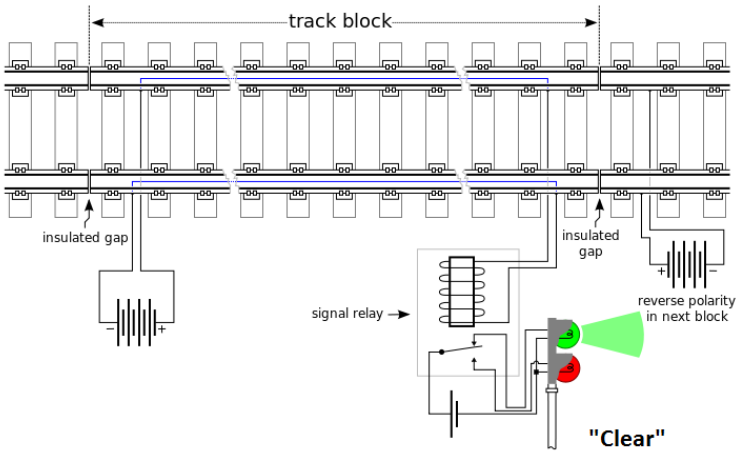

Figure 2.10 : Track Circuit working principle (clear) [14].

To obtain an electrically isolated section, rails divided physically and fitted an isolation material in the cutting point.

It is also possible to obtain isolation between rails by electrical means without physical disruption of the rails. This type of track circuits called “jointless track circuit”.

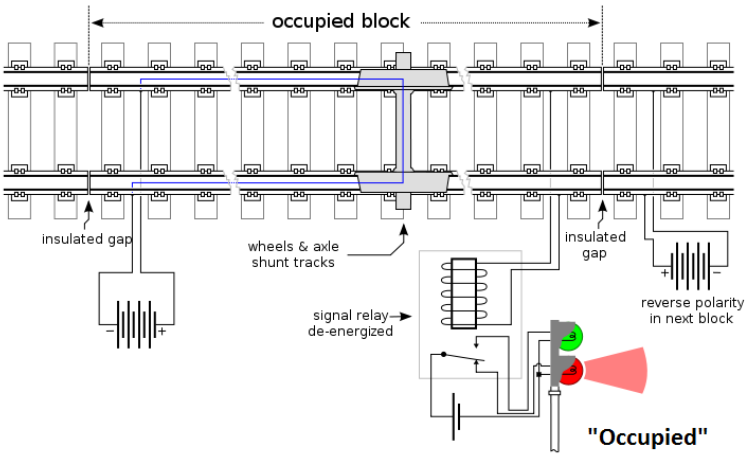

Figure 2.11 : Track Circuit working principle (occupied) [14].

2.3.3.2 Axle counters

Another solution for occupancy detection is axle counting method. In this method occupied status of a block determined by using devices located at the beginning and end of the block that count the number of axles entering and leaving. If the same number of axles leave the block as enter it, the block is assumed to be clear. The logic behind the working principle of axle counters is illustrated in Figure 2.12.

Axle counters provide similar functionality to track circuits. Comparison of track circuits and axle counter can be seen in following Table 2.1.

2.3.4 Derailing devices

Derailing devices are protection equipment used against to accident caused by unintended movements of rail vehicles. Rail vehicles rolling uncontrolled because of any reason may create very dangerous situation for other rolling stocks. Therefore, these devices located on the track which is connecting a depot area or sidings to the main line. Thus, if any rolling stock runs away towards main line, it is derailed by these devices.

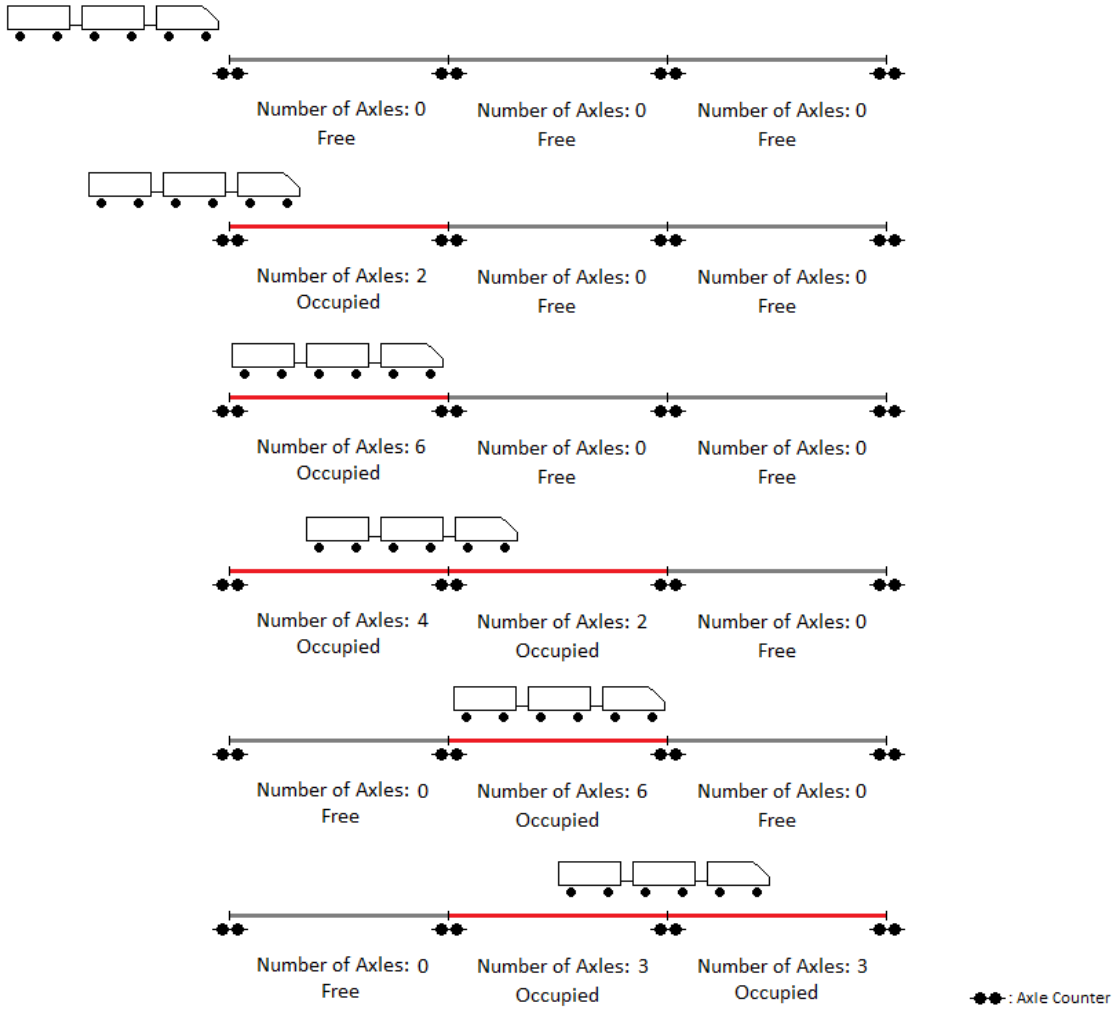

Figure 2.12 : Axle Counter working principle [1]

Table 2.1 : Comparison of Track Circuits and Axle Counters [2]

	Track Circuit	Axle Counter
non-detection of railway vehicles	completely derailed vehicles	vehicles newly put on the track
detection of obstacles	only in few cases	no
detection of broken rails	partly	no
vehicle requirements	electrically conducting wheels and axles	ferromagnetic wheels
track requirements	electrical isolation	no special requirements
treatment of traction return currents	special measures necessary	not necessary
excessive voltage problems (e.g. Lightining)	present, due to earting of many devices to the rails	slight
sensitivity to climatic influences	relatively high	low
length of track sections	electrically limited	unlimited
frequency of dangerous failure	extremely low (if rust on rails is prevented)	extremely low
frequency of hindering failure	relatively high	low
possibility of staff preventing danger	stop trains by short-circuiting the rails	no comparable possibility
usability for other purposes	detection of train reaching (in combination also clearing) a certain point; transmission of block information; train protection + cab signalling	detection of train reaching or clearing certain point.

2.3.4.1 Catch points

Catch point, also called “Trap Point”, is a specific kind of the point. The mechanism of them are almost same but they has different functionality. Catch point is used only as a derailing device in some critical location. Figure 2.13 shows the trap point’s functionality.

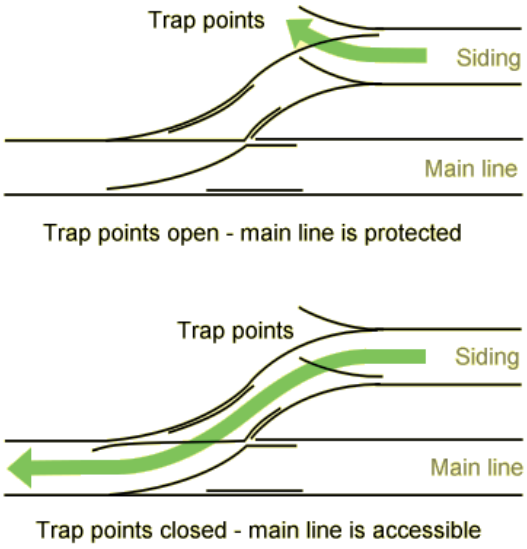

Figure 2.13 : Functionality of a Trap Point [15].

2.3.4.2 Derailer

Derailer is a special device used for the same purpose with catch point. However, it has a special profile and it is mounted above onto the rail head. Derailer is also an active controllable device and it can be moved to upon the rail (Figure 2.14 - b) or aside the rail (Figure 2.14 - a) to enable or block the vehicle passing over it.

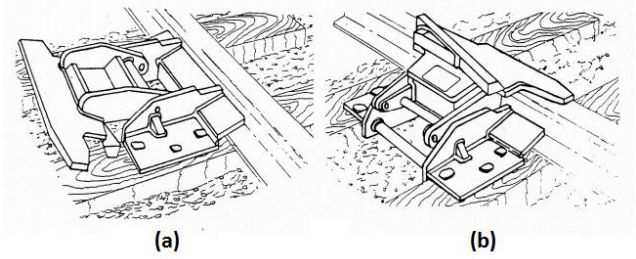

Figure 2.14 : An active controlled Derailer [16].

2.3.5 Level crossings

Normally, railways are isolated from the other vehicle's road. However, in some location they intersect each other. A level crossing is an intersection of a railway and a road. Following illustration (Figure 2.15) shows a level crossing area. Level crossing control is very important in railway signalling to ensure safety.

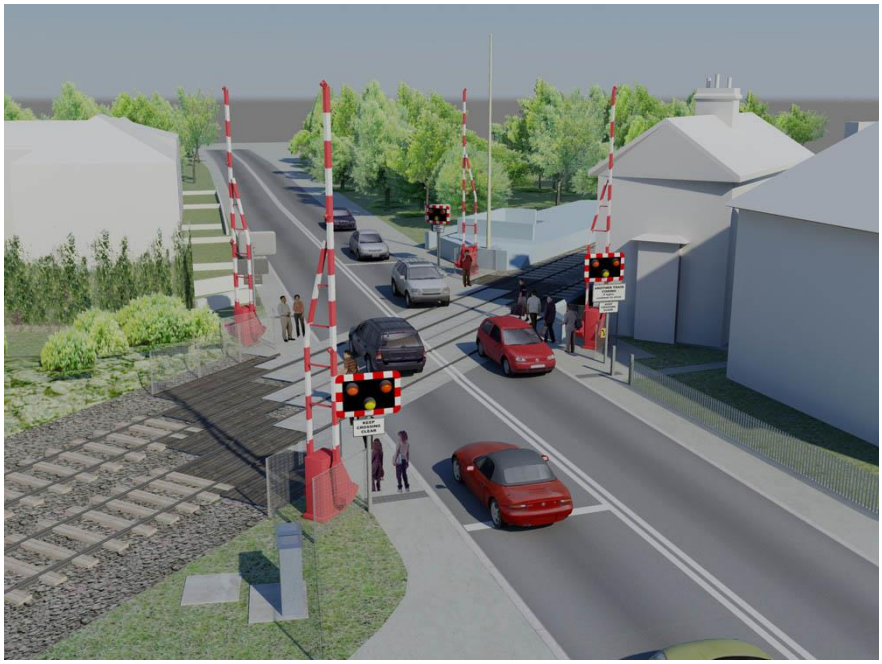

Figure 2.15 : A level crossing area illustration [17].

Level crossing protection is the consequence of having level crossings on a railway line. Its aim is to avoid collisions between trains and road traffic. General protection principle is simple: it has to stop all road traffic before the passing of a train.

2.4 German Ks System

All countries have different type of signalling equipment for different purposes around the world. In Germany, there are also several signal methodology used in different regions such as; Ks, Hp or HI system. Ks system is one of these signalling methodology which using in Germany since 1993 [18]. It is a relatively new signalling system replaced by the old ones. In this study, German Ks system has been considered as the signalling principle. However, all features of the Ks system have not been included. Otherwise, models which will be designed in the next chapters would be too complicated and less understandable.

The most important characteristic of Ks system is that the main signals are used as a combination of conventional main signal and distant signal. The main signal has a “caution” aspect besides “proceed” and “stop” to indicate next main signal's status. Figure 2.16 compares two and three aspect system.

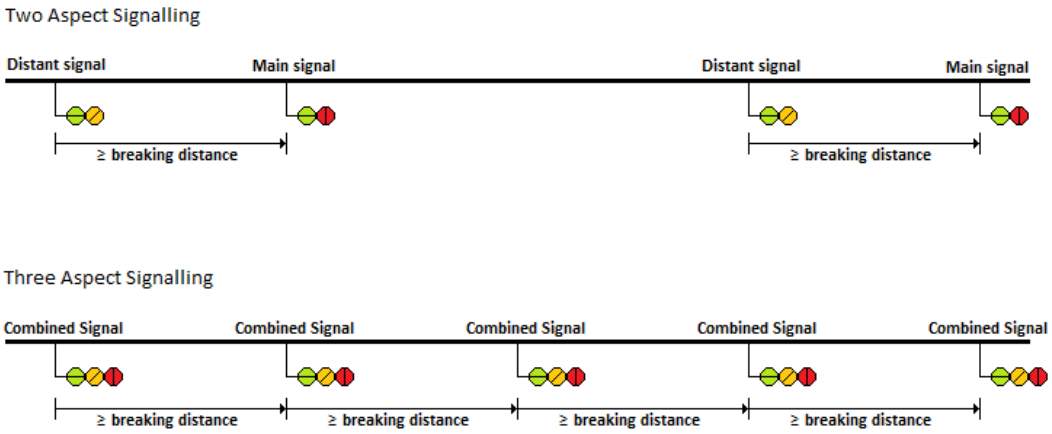

Figure 2.16 : Two and three aspect systems [2]

2.4.1 Main signal

Ks system has 3 main aspects. Figure 2.17 shows general appearance of a main signal.

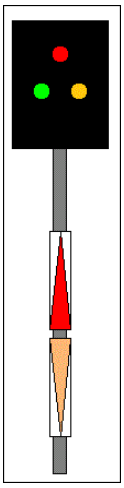

Figure 2.17 : Ks Main Signal [19].

2.4.1.1 Green: proceed

Green light indicates the next two block are clear, proceed with full speed (Figure 2.19). That means next main signal also has been set as yellow or green. Figure 2.18 shows red and green signal sequence.

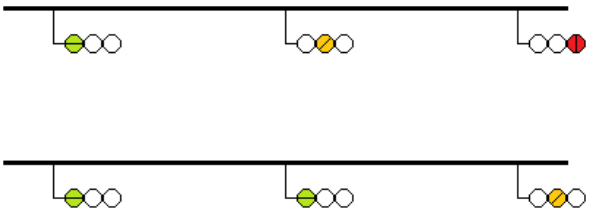

Figure 2.18 : Yellow and green light [2].

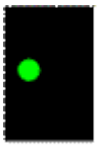

Figure 2.19 : Proceed aspect [19].

2.4.1.2 Yellow: proceed with caution

Yellow light means: proceed but expect stop because next main signal shows stop. See Figure 2.20.

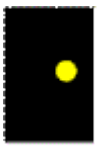

Figure 2.20 : Caution aspect [19].

2.4.1.3 Red: stop

Next signal block is occupied by another vehicle or it has not been set yet. Do not proceed. Figure 2.21 shows the red light aspect.

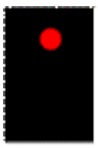

Figure 2.21 : Stop aspect [19].

2.4.1.4 Blinking green: expect speed restriction

If there is a speed limit in the next signal block, main signals shows blinking green (Figure 2.22). It is always used with a speed indicator or speed limit plate.

Figure 2.22 : Expect reduced speed aspect [19].

2.4.2 Distant signal

Distant signal informs driver about aspect of the next main signal. It has only two aspects. Figure 2.23 shows general appearance of a distant signal.

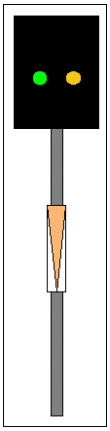

Figure 2.23 : Ks Distant Signal [19].

2.4.2.1 Green: expect proceed or caution

The meaning of green aspect (Figure 2.24) in a distant signal is the next main signal is clear (it is green or yellow).

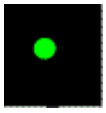

Figure 2.24 : Distant Signal green aspect [19]

2.4.2.2 Yellow: expect stop

If a distant signal shows yellow aspect, that means the next main signal shows stop, apply brakes to stop on time. Following Figure 2.25 is the yellow aspect of a distant signal.

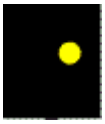

Figure 2.25 : Distant Signal yellow aspect [19].

2.4.2.3 Blinking green: expect speed restriction

The blinking green distant aspect is the same as blinking green main aspect. It is used if the next main signal has a speed limit (Figure 2.26).

Figure 2.26 : Distant Signal blinking green aspect [19].

When there are more than one distant signal in a block, the second signal used as a repeater signal. Little white light in bottom left shows that it is a repeater distant signal. Figure 2.27 and Figure 2.28 are distant repeater signals.

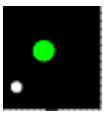

Figure 2.27 : Distant Repeater Signal (1) [19].

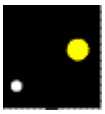

Figure 2.28 : Distant Repeater Signal (2) [19].

If the brake distance is shorter than normal, driver is informed by a little light on the top left side of distant signal. Following Figure 2.29 is a short distance signal.

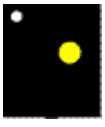

Figure 2.29 : Short distance Distant Signal [19].

2.4.3 Speed restriction signal

Speed limits are shown by a speed board where the allowed speed limit is constant. But maximum speed value can change according to the position of the points in a route.

Speed Restriction signal or speed indicator shows the allowed maximum speed in relevant block. If it is dark, that means there is not any speed limit.

Ks system has two types of speed indicator. One of them is used to show maximum speed value after the main signal. It is located on the top of main signal frame and its color is white. See Figure 2.30.

Figure 2.30 : Main Speed Indicator [19].

Other type of speed indicator shows the maximum speed value for the next signal. It is located just under the main signal frame and it has a yellow color. It is also used with distant signals. See Figure 2.31 and Figure 2.32.

Figure 2.31 : Distant Speed Indicator (1) [19].

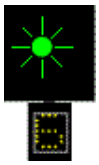

Figure 2.32 : Distant Speed Indicator (2) [19].

If it is necessary, both type of signals can also be used with the same main signal. See Figure 2.33.

Figure 2.33 : Both Speed Indicators with the same main signal [19].

2.4.4 Shunting signal

Shunting signals are used in a depot or another area, where allowed speed limit is very low. There are two shunting signal aspects: Proceed and Stop.

2.4.4.1 White: shunting allowed

Shunting movement is permitted but driver is obliged, not to reach the maximum speed which defined for the shunting movements (Figure 2.34).

Figure 2.34 : Shunting permitted [19].

2.4.4.2 Red: shunting is not allowed

Red light in a shunting signal (Figure 2.35) means shunting movements are not permitted.

Figure 2.35 : Shunting not permitted [19].

Shunting signals can also be combined with the main signal. See Figure 2.36.

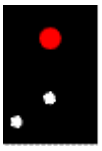

Figure 2.36 : Combination of Shunting and Main Signal [19].

2.5 Turkish Signalling System

In Turkish State Railways, there are mainly three kinds of signal lights: four aspect main signal, three aspect main signal and three aspect dwarf signal. [20]

2.5.1 Four aspects main signal

This type of signals are generally used in the entry of a station or before a point area. Following Figure 2.37 shows a four aspects main signal. Principally, yellow light at the bottom of signal frame indicate that there is at least one point in reverse position.

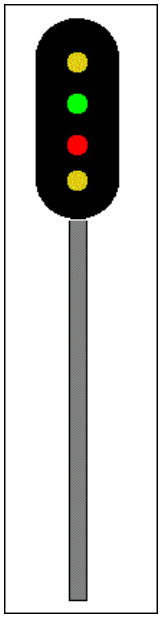

Figure 2.37 : Four aspects main signal [21].

2.5.1.1 Green: proceed

Figure 2.38 shows proceed aspect means the next two block are clear, proceed with full speed.

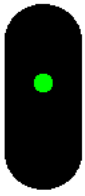

Figure 2.38 : Proceed aspect [21].

2.5.1.2 Yellow: proceed with caution

Yellow light means: proceed but expect stop because next main signal shows stop. See Figure 2.39.

Figure 2.39 : Caution aspect [21].

2.5.1.3 Red: stop

Red light means the signal block is occupied by another vehicle or it has not been set yet. Stop immediately. Figure 2.21 shows the red light aspect.

Figure 2.40 : Stop aspect [21].

2.5.1.4 Yellow - yellow: Proceed with caution and speed restriction

Yellow light at the bottom of the signal frame informs driver there is a point in reverse position. In another words, driver has to proceed with allowed maximum speed for reverse position points. Another yellow aspect which at top of the signal frame has same meaning with single yellow light described before. Following Figure 2.41 shows the yellow over yellow aspect.

Figure 2.41 : Proceed with caution and speed restriction aspect [21].

2.5.1.5 Green - yellow: Proceed with speed restriction

Yellow light at the bottom of the signal frame has the same meaning whit previous yellow over yellow aspect and green light means next two signal block are clear. In another words, green over yellow means proceed with restricted speed because there is a point in reverse position. See Figure 2.42.

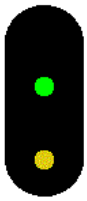

Figure 2.42 : Proceed with speed restriction aspect [21].

2.5.1.6 Red - yellow: Proceed to an occupied block

Red over yellow is a shunting aspect. It means the block is occupied but driver is permitted for shunting movement. Yellow light also indicates there is a point in reverse position. See Figure 2.43.

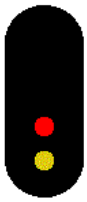

Figure 2.43 : Proceed to an occupied block [21].

2.5.2 Three aspects main signal

Three aspects main signal is used if it is not possible to have a point in reverse position. In that case, it is not needed a yellow signal at the bottom of signal frame. Figure 2.44 shows a general view of a three aspects main signal.

Three aspect main signal only has green, yellow and red aspects and all of them are the same with 4 aspects main signal's green, yellow and red aspects. See the following Figure 2.45, Figure 2.46 and Figure 2.47.

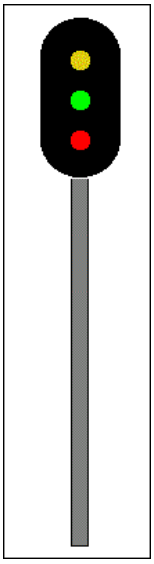

Figure 2.44 : Three aspects main signal [21].

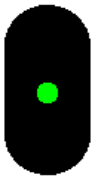

Figure 2.45 : Proceed aspect [21].

Figure 2.46 : Caution aspect [21].

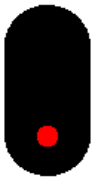

Figure 2.47 : Stop aspect [21].

2.5.3 Three aspects dwarf signal

Three aspects dwarf signal (Figure 2.48) is used if a signal block has always a point in reverse position.

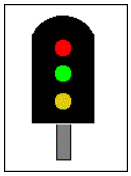

Figure 2.48 : Three aspects short signal [21].

Three possible aspects have the same meanings with three aspects main signal. Green: proceed, yellow: proceed with caution, red: stop. See following Figure 2.49, Figure 2.50 and Figure 2.51. Red – yellow aspect (Figure 2.52) means proceed over an uncontrolled area. After passing that aspect the train will left signalled area.

Figure 2.49 : Proceed on a reverse point [21].

Figure 2.50 : Proceed with caution on a reverse point [21].

Figure 2.51 : Stop [21].

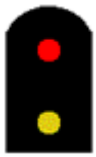

Figure 2.52 : Proceed over an uncontrolled area [21].

Flashing aspects also used in dwarf signals. Flashing green and flashing yellow aspects have the same meanings with constant green and yellow but the difference is the routes start in an uncontrolled area but end in a controlled area. That means there might be another unauthorized vehicle on the route.

Flashing red is used for the train movements in uncontrolled areas which include a controlled point. Flashing red - yellow is used for the routes which is set from

uncontrolled area to another uncontrolled area over a controlled area. Following Figure 2.53 shows the all flashing dwarf signal aspects. A special palate is also used with these signals to indicate they are flashing signals.

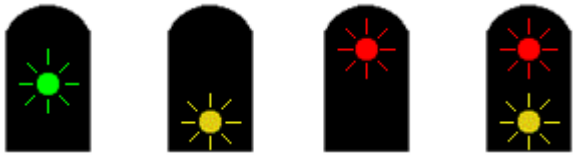

Figure 2.53 : Flashing dwarf signal aspects [21].

3. RAILWAY INTERLOCKING SYSTEMS

3.1 What is Interlocking?

Interlocking is a kind of internal automatic control mechanism which used between two or more devices, equipment or any other phenomenon. In an interlocking system, some status of the devices are defined as a precondition to control a certain device. In another words, devices cannot be controlled directly. It is designed within a system, which can create some hazardous results in a certain status combinations. Interlocking system locks the controlling of critical devices in between and allows only possible safe status sets.

The working mechanism of the interlocking can be explained with a simple example. There is an interlocking system to protect maintenance staff against electrical shock in a maintenance depot of a railway operator company in Istanbul (Istanbul Ulasim A.S.). Some components of railway vehicles are installed over the car body with some high voltage equipment. The maintenance of components can be very dangerous if the high voltage equipment are alive.

An overhead catenary system provides electrical power to trains in the depot. Maintenance staff use a platform to reach top of the trains and electrical power has to be switched off before anybody use this platform. The procedure which defined to work on the trains has to be followed by the maintenance staff when they are working on the train's roof. However, if somebody reaches the train's roof when the catenary line is alive, it may cause injury or death. Therefore, this problem is solved with using an interlocking system between the circuit breaker and the platform.

The electricity on the catenary system is controlled by a circuit breaker, which is equipped with a key. This key is released only when the circuit breaker is switched off and the circuit breaker cannot be switched on without this key as well. On the other side, the platform has a locked door to prevent passage of unauthorized staff. The door can only be opened with a key and it does not release the key when it is unlocked. The

interlocking system is provided by these mechanism. For instance, the staff who wants to work on the trains has to use platforms but there is a locked door front of the platform steps. The only way to unlock the safety door is switching off the circuit breaker and getting the key. Conversely, it is prevented to switch on the circuit breaker when there is somebody on the platform.

To sum up, almost all possible dangerous situations are prevented with an interlocking mechanism between system equipment. In the given example, the platform door and the circuit breaker represent the critical equipment in the system. The key is used as an interlocking tool to interlock the critical equipment.

Nowadays, most of the new developed systems are based on software. However, it is still required some interlocking mechanism in safety critical systems. For this reason, some interlocking algorithms are developed by system engineers to ensure the safety in software based systems. Modern railway interlocking systems can be given as a good example of software based safety critical systems.

3.2 What is the Fail-Safe?

Safety critical systems include some equipment which are very important for the system safety and it is required that these equipment should be always failure-free. Whereas, any device or equipment cannot be fully reliable in the real world. Fail-safe is a design criteria used in the devices may cause some dangerous consequences when it fails to guarantee safety of the system [22]. In railway signalling systems, almost every equipment and devices are produced with respect to fail-safe criteria [23].

A simple example can be given to understand fail-safe logic. For instance, there is a security door in a bank and it should be always monitored whether it is opened. There is also an alarm system which is activated when the door is opened. A simple mechanical switch can support the information of the door's condition. There are only two output: door is open and door is closed. To obtain a fail-safe system, the first question should be "What is the safe situation when the position switch is failed?". If the switch still transmits the "door is closed" information when it fails, the system cannot notice the failure and the door is not being monitored anymore. Thus, anybody can not realize if the door is opened. Therefore, "open state" should be chosen as the fail-safe state. Then, in any failure on the mechanical switch, it will be seen that the

door is opened and the alarm system will be activated. Thus, security staff can realize that there is a failure in the position detection component.

Every equipment and device has a fail-safe procedure in the railway interlocking system. System engineers are also consider the fail-safe procedure of all components used in signalling system when they are designing an interlocking system.

3.3 Railway Interlocking Systems

Railway signalling systems are very critical systems. Any dangerous situation which may occur in the system can cause very dangerous accidents. Therefore, interlocking systems are used to prevent any hazardous cases in the signalling systems. Interlocking mechanism described under the previous topic is implemented to the signalling equipment and it is called the railway interlocking system.

Interlocking is the core system in railway signalling. It ensures that all signalling equipment are in proper status for train movement. Basically, it obtains information about train occupancy and locks the movable wayside elements in correct position for a certain route. Then, it permits movements via signals.

Depending on the technological developments, different kind of interlocking systems are developed until today. The first developed system is the mechanical interlocking. Almost every element were mechanical equipment in the first interlocking system. Movable elements were being controlled by steel wires and there was not any train detection mechanism. Signalling operator who stays in a control tower at the station area checks the presence of the trains, sets the points sequentially and clears the signal by mechanical levers. Interlocking of the wayside equipment is achieved by a device called locking bed (Figure 3.1). It only permits safe possible state combination of the wayside equipment.

Electro-mechanical interlocking systems are developed in the end of 19^{th} century. The central interlocking unit was still a mechanical device but wayside elements was being controlled by electrical or pneumatic actuators.

The next technology used to developing interlocking system was relay based technology. In that technology, mechanical interlocking mechanisms leaved their objects to the complex relay based interlocking circuits.

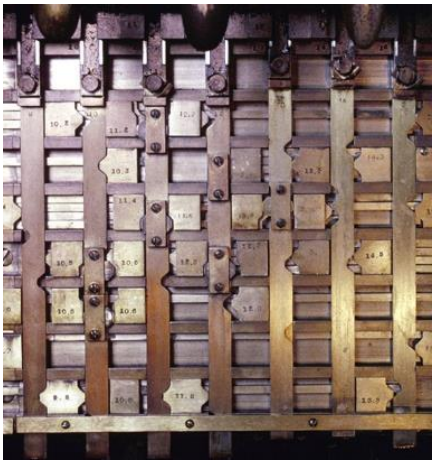

Figure 3.1 : The locking bed mechanism [24].

They were also called “all-electric” signal boxes. Route setting was achieved by selecting start and target signal on the control panel (Figure 3.2). This technique was the first used entrance-exit (NX) method to set a route.

Figure 3.2 : A relay interlocking system and a control panel [24].

The next step was the development of systems with electronic component in the 1980s. The fact that the logic is implemented by software rather than hard-wired circuits in electronic interlocking technology. Modern monitors were used to manage the system instead of old NX panels.

In United Kingdom, the first generation microprocessor-based interlocking called Solid State Interlocking (SSI) is developed. It was the brand new developed technology before the Computer Based Interlocking (CBI) systems.

Nowadays, one of the new trends is to develop interlocking systems which based on PLC devices. Through new developed safe PLC devices, it is possible to develop safe, reliable and flexible PLC based interlocking systems. In this thesis study, an approach to develop PLC based interlocking mechanism is represented.

3.4 Railway Interlocking Basics

Some general basic principles of railway interlocking systems were explained in this chapter.

3.4.1 Path and route

Path is a term used to denote actual possible way on a railway in a certain condition. Some sample paths are shown in Figure 3.3. The railway points set the actual path in a railway.

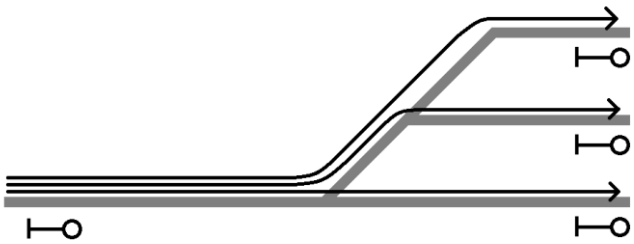

Figure 3.3 : Some possible paths [1].

Paths are arranged and all movable elements on it are locked to safe train movements by interlocking system. This safe path is called as “route” (see Figure 3.4). Every route has a starting and exit signal.

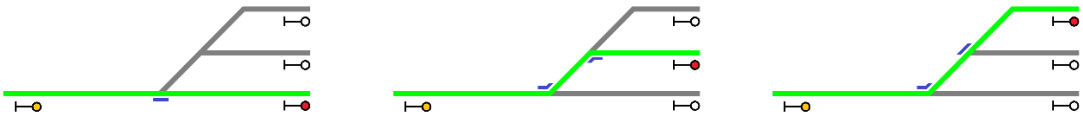

Figure 3.4 : Different Routes [1].

3.4.2 Shunting routes

Railway coaches can be coupled and uncoupled with each other to make a complete train set. Movements of the rolling stocks for this purpose are called shunting movements and routes for the shunting movements are called shunting routes. Usually, shunting routes are defined in a station or a depot area. Shunting signals which have some different aspects from main signals are used to indicate a shunting permission.

Shunting routes have some different procedures from normal routes. These procedures will be explained later.

3.4.3 Local operation area

Shunting movements are controlled from the train control center by a signalling operator. But, for some particular areas, authorization of shunting operation can be given to the local staff. These areas are defined as local operation areas. Once the operation authorization is given to the local personnel, signalling operator does not have any responsibility from any movement in the local area.

3.4.4 Locking functions

There are some special locking functions used in interlocking systems. Some of them are explained in this thesis.

3.4.4.1 Simple element locking

As it is mentioned, all movable elements have to be locked, before any movement permıssion of the train. Locking means that keeping an element in a certain position or condition to prevent any changing on its position or condition until it is unlocked. In other words, a locked equipment cannot be controlled until it is unlocked. For example, locking a point in normal position is defined as a simple locking.

3.4.4.2 Coupled elements

Coupled elements are interlocked to each other directly. Some safe conditions are defined for coupled elements. In movable coupled elements, if any element starts to change its position, other one starts to move immediately to pass defined status according to the new position of first moved element.

Points in the Figure 3.5 are coupled elements. Only two conditions are defined and both have to be in normal or reverse position. A signalling operator sends a moving command to the point located bottom in the figure. It changes position from normal to reverse. Then, other coupled point starts to move without any command from the signalling operator.

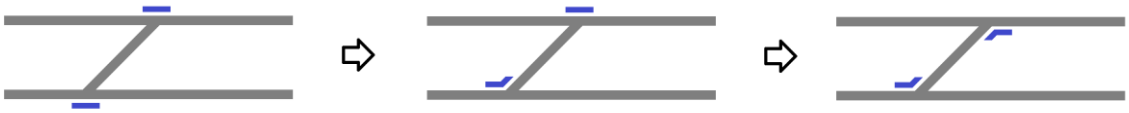

Figure 3.5 : Coupled elements [1].

3.4.4.3 Unidirectional locking

Unidirectional locking is defined for two or more elements. One of them is independent element and the others are dependent elements. Usually, it is defined for the main signal and distant signals.

The signals shown in the Figure 3.6 are locked unidirectionally. The signal on the left hand side is dependent to other signal. Because it is a distant signal and distant signals are always dependent to the main signals.

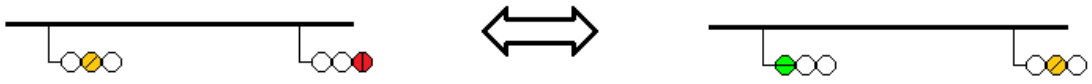

Figure 3.6 : Unidirectional Locking [2].

3.4.4.4 Simple bidirectional locking

Simple bidirectional locking is applied to two or more elements when only one defined condition is allowed. This is a simple interlocking between the elements. Some element's conditions or positions are defined as a precondition to control one of them.

It can be implemented to a signal and point like in the Figure 3.7 given above. In this bidirectional locking, normal position of the point is defined as a precondition before clearing the signal. On the other hand, to move point from normal to reverse position, signal has to show to stop.

Figure 3.7 : Simple Bidirectional Locking [1].

3.4.4.5 Conditional bidirectional locking

Conditional bidirectional locking is very similar to the simple bidirectional locking but it is defined for three or more elements.

In the Figure 3.8, signal can only be cleared if there is a safe path ahead. Bidirectional locking is defined between the signal and the point on the right hand side. When the first point after the signal is in reverse position bidirectional locking is not considered. But if the first point is in normal position bidirectional locking should be considered. Then, it is exactly the same principle with simple bidirectional locking.

Figure 3.8 : Conditional Bidirectional Locking [1].

3.4.5 Flank protection

Flank protection is considered to protect a route against dangerous movements which may come from the flank area. Figure 3.9 shows the flank areas of the given route.

Flank protection can be obtained by blocking points, derailing devices, signals or track sections.

3.4.5.1 Point and derailing device blocking

To protect the route, points and derailers are blocked in proper position as shown in Figure 3.10 and Figure 3.11.

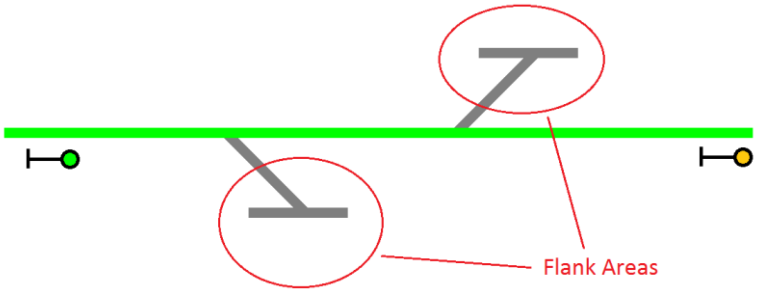

Figure 3.9 : Flank Areas [1]

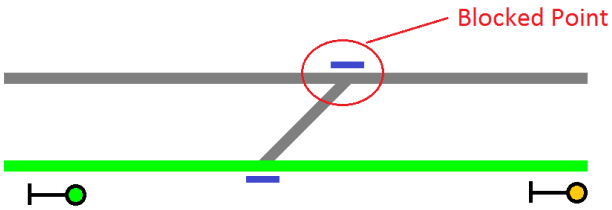

Figure 3.10 : Point blocking for flank protection [1].

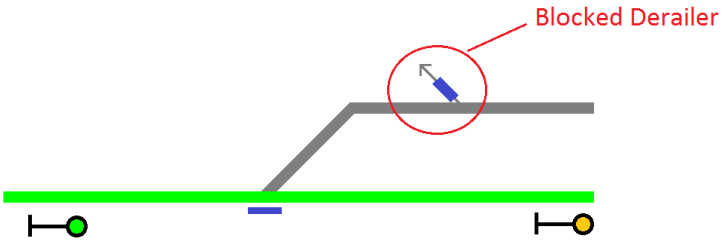

Figure 3.11 : Derailer blocking for flank protection [1].

3.4.5.2 Signal blocking

If there is not any point or derailer device, signals can be blocked for flank movements. Figure 3.12 shows a sample situation.

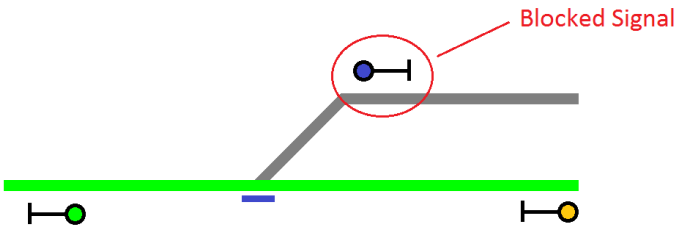

Figure 3.12 : Blocked signal for flank protection [1].

In some cases, points cannot serve flank protection and transfer it to signals or another points. Figure 3.13 and Figure 3.14 can be given as an example for this case.

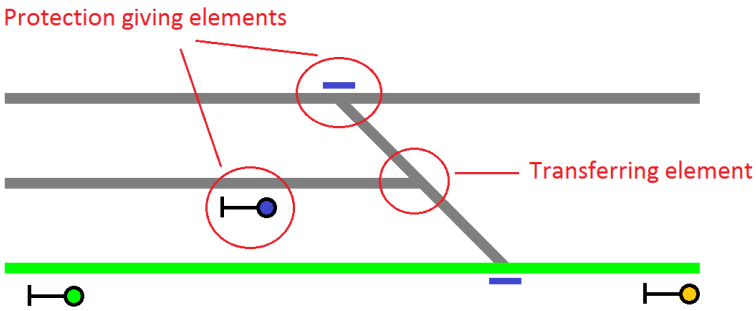

Figure 3.13 : Transferring flank protection (1) [1].

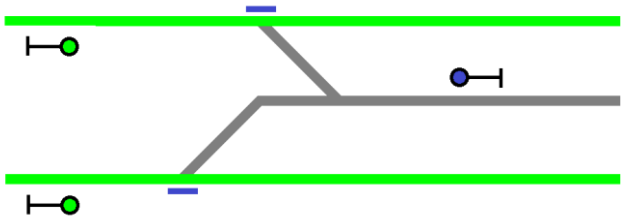

Figure 3.14 : Transferring flank protection (2) [1].

3.4.6 Overlaps

Overlap is another protection method to obtain safe braking distance behind the target signal. Overlap area in a route can be seen in Figure 3.15. When setting a route, it is required to have a free track section after the target signal and it is blocked until the route released. Thus, if driver does not apply brakes on time, there will be still a safe distance before any other vehicle or obstacle.

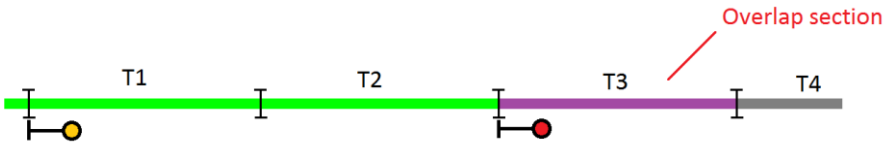

Figure 3.15 : Overlap [1].

Signalling systems, which have a good train protection system does not provide overlap protection. However, it is always used in German railways. Thus, it is considered in this thesis study.

3.4.7 Front protection

Front protection is applied to protect the route against unauthorized vehicles coming from the opposite direction. Point shown in Figure 3.16 is a front protection element. Generally, it is required, when the overlap distance is not so long.

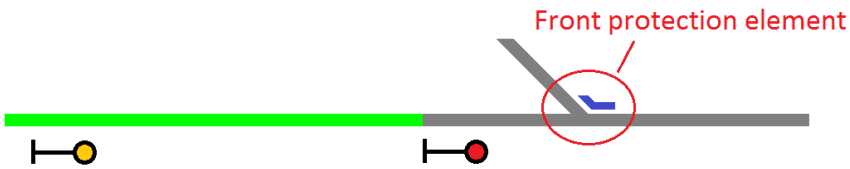

Figure 3.16 : Front protection [1].

3.4.8 Conflicting routes

The routes, which use the same wayside element called conflicting routes. Only one route can be set in a certain time and others have to be blocked. Following Figure 3.17 shows some intersecting routes on different wayside elements.

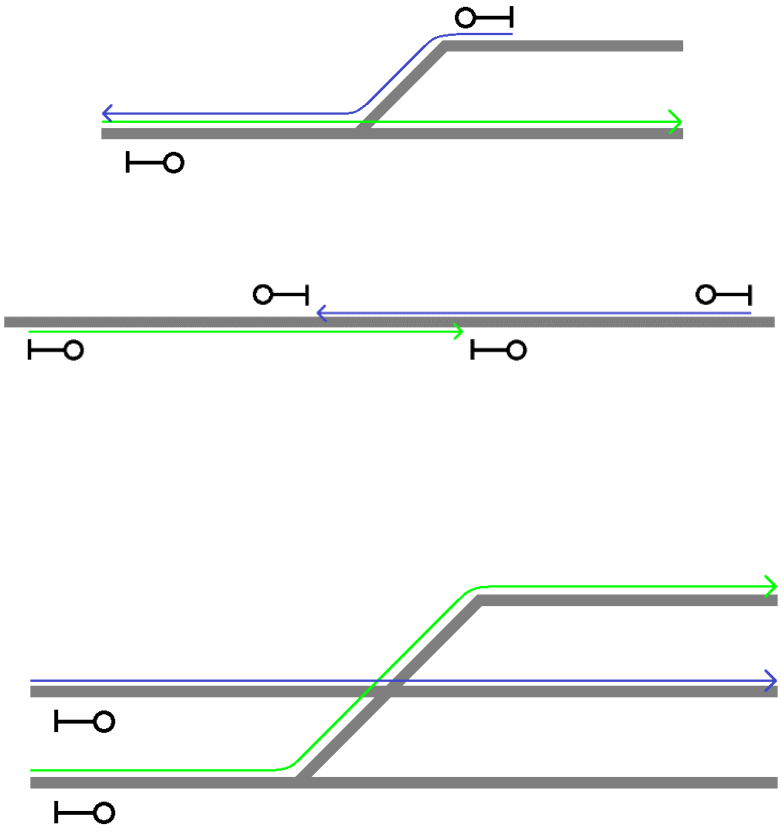

Figure 3.17 : Some conflicting routes [1].

3.4.9 Deadlock situation

Deadlock is a kind of situation, which a number of train block each other and none of them cannot proceed [25]. It is an unfavorable situation and should be prevented. Following Figure 3.18 shows some deadlock situations.

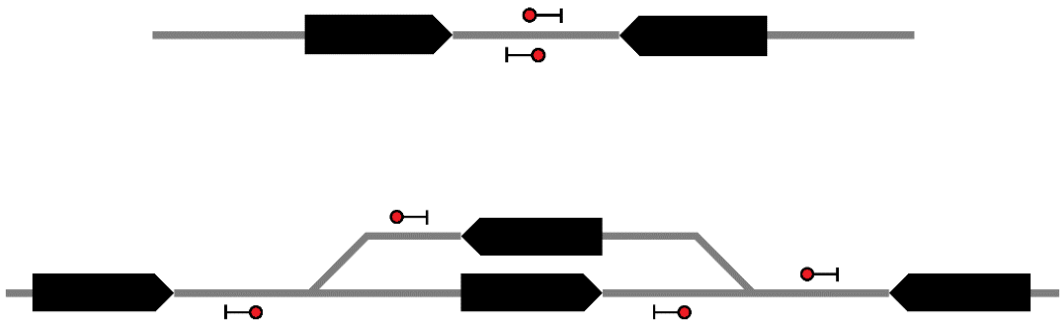

Figure 3.18 : Some deadlock situations [2].

3.4.10 Multi routes

Almost every modern signalling system uses entrance-exit (NX) method to set a route. But when there are more than one defined routes to reach required target signal, it can cause a multi route situation (see Figure 3.19). In that case, interlocking system cannot distinguish the required route.

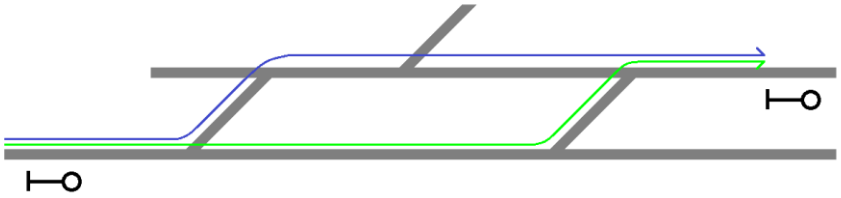

Figure 3.19 : Possible routes to the same signal [1].

To solve this problem, one of the routes is assigned as a priority route. When signalling operator requires a route with NX method, interlocking system sets the priority route. However, if the priority route is not available, it sets alternative route automatically.

Another solution for the same problem is to design an interface for choosing the required route by signalling operator.

3.4.11 Route setting

3.4.11.1 Main route setting

Route setting is one of the basic functions of an interlocking system. Different interlocking systems can have different route setting procedures. Most general and important main route setting steps are explained as follows;

- Conflicting route checking: When a route setting is requested by signalling operator or automatic route setting unit, interlocking system checks, whether there are any conflicting routes which are already set with required route.
- Failure checking: As a second step, interlocking system checks all wayside equipment related the required route. Any failure is detected in the wayside equipment, the route setting request cannot be approved.
- Occupancy checking: There must not be any occupied track section in the route path and overlap. In some routes, some track sections in the flank area also have to be free.
- Setting movable elements: The next step is setting all movable elements (in the route path, flank area and overlap) in their proper positions and locking them. If any element is locked in wrong position before or is not in remote control, it cannot be set and the route setting request is rejected. In addition to that, it is also possible to get a failure message from the elements when they are moving. It also causes request cancellation. Furthermore, any occupancy on the points in the flank protection area is a reason to get a rejection from route setting request.
- Locking the route and signal opening: If all steps are succeeded, the route can be locked and signal can be opened in proper aspect according to condition of the next signal.

To implement these rules, all specifications of the routes have to be defined in the interlocking system. For instance, generation of a table is a method for route definition. Route table will be explained later.

3.4.11.2 Shunting route setting

As distinct from main route setting, shunting routes can be set when there are some occupancy in the path. However, points in wrong position cannot be set, if its track

section is occupied and the route cannot be set. In addition to that, flank protection and overlap are not considered in some interlocking systems. The rest of the rules described for the main routes are almost same in shunting routes.

3.4.12 Route releasing and reversing

After the trains traversed the running path, the route is released. Track sections in the path should follow a set-occupied-free sequence. Set-occupied-free sequence, illustrated in Figure 3.20, is defined to represent a normal train movement in a main route.

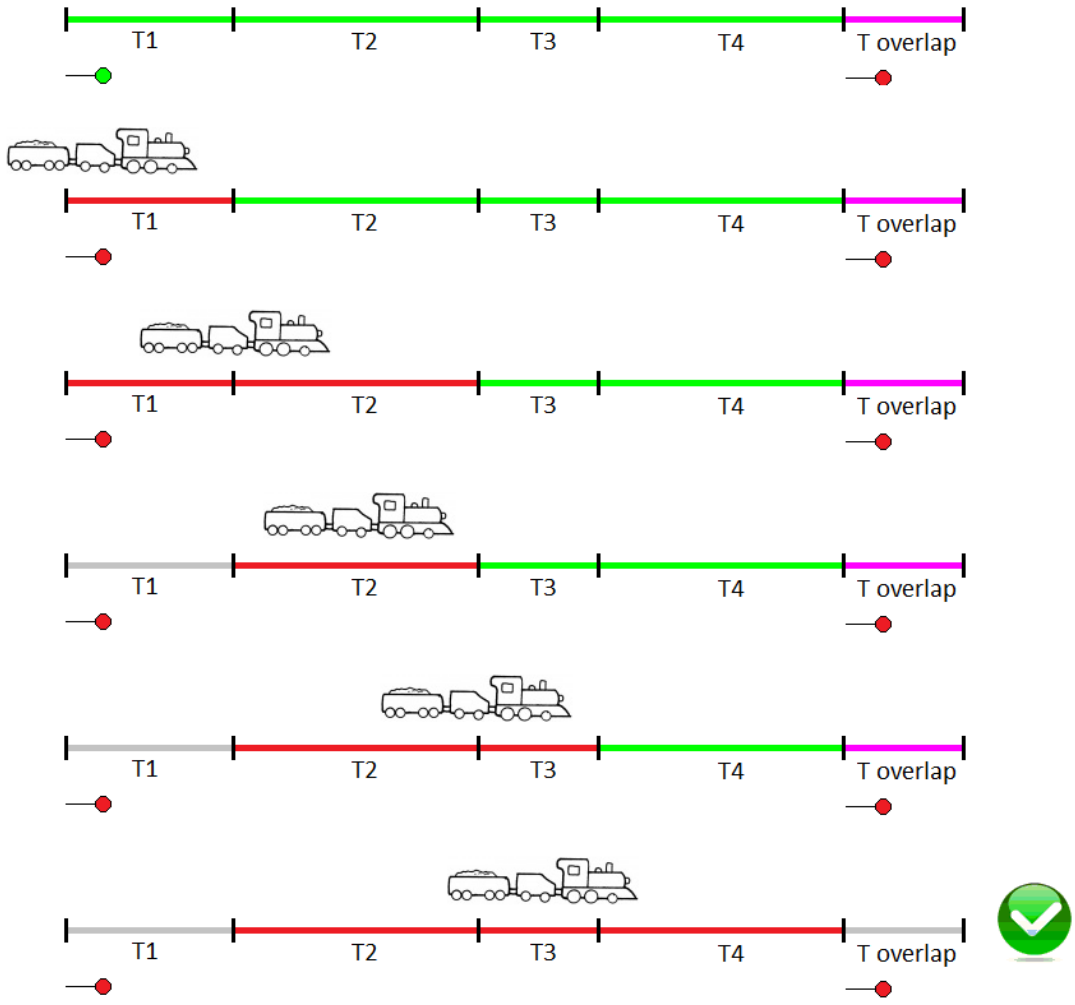

Figure 3.20 : Set-occupied-free sequence [1].

Any different sequences should be detected by interlocking system. Some abnormal train movement cases have been given below. Following Figure 3.21, Figure 3.22, Figure 3.23, Figure 3.24 and Figure 3.25 show some abnormal cases.

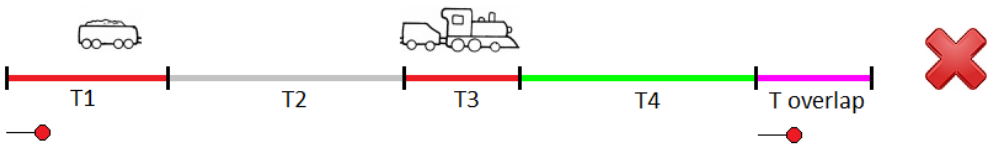

Figure 3.21 : Decoupled wagon case [1].

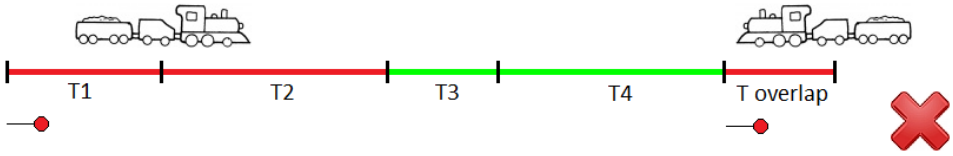

Figure 3.22 : Head-on trains case [1].

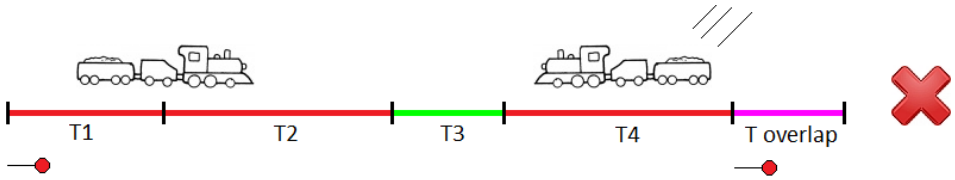

Figure 3.23 : Flying train case [1].

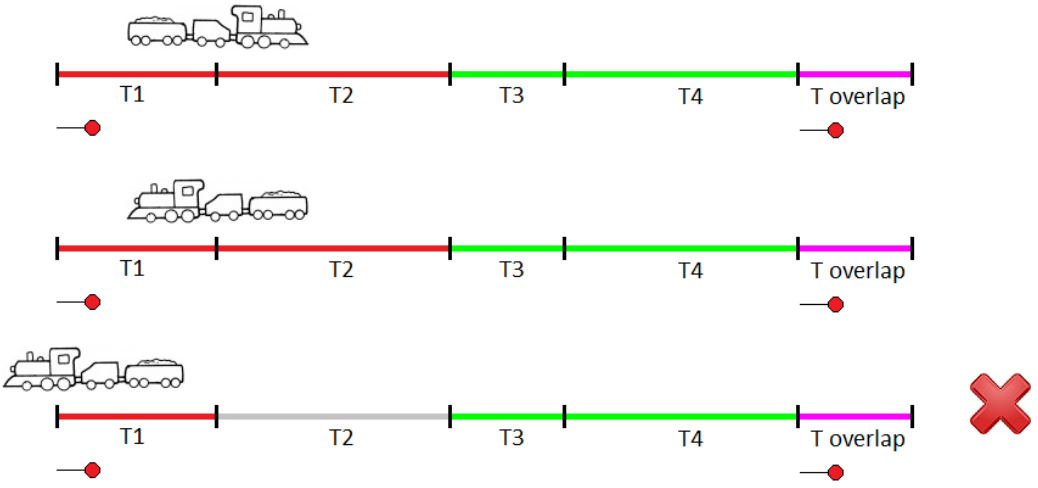

Figure 3.24 : Going back train case [1].

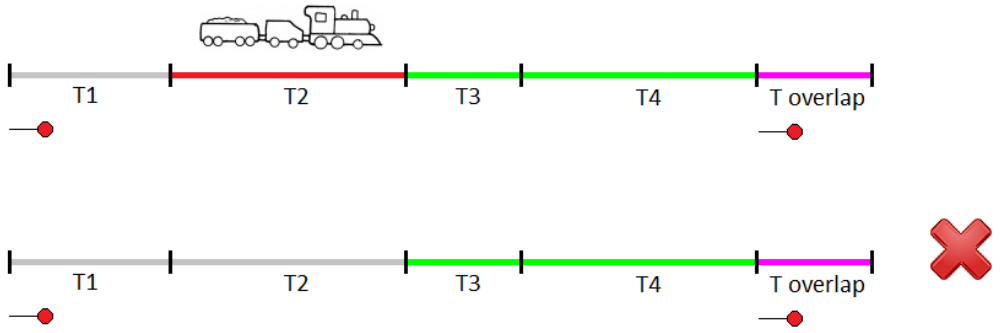

Figure 3.25 : Disappeared train case [1].

Different route releasing conditions can be described for particular routes. Following steps are described as a general main route terminating procedure.

- When the first track section is occupied, start signal has to show “stop” aspect immediately.
- All wayside equipment has to be kept locked until the train leaves the last track section. Then, they can be released. However, for operational reasons some equipment can be unlocked after the train passed over them.
- To keep locked all flank protection objects can cause some operational delays. Therefore, special unlocking procedures are defined for flank protection objects.

Shunting route termination procedure is almost same as the main route termination procedure.

Because of some operational reasons route setting can be reversed easily, if there isn’t any train approaching to the starting signal. Nevertheless, if a train is approaching a route it cannot be cancelled immediately for safety reasons. Interlocking system waits for a while before cancelling the route. If the train does not enter the routing area in a particular time, the route is released.

3.4.13 Route table

Route table, also called interlocking table, is composed to list all specifications of all possible routes in an interlocking systems. It shows the required status of all equipment

in a certain route. When a new signalling system is designed, a route table is created as one of the first steps by system designer and all system is designed based on it.

There is not any general form for route tables. Every system designer use their own route table form. An example route table (Table 3.1) is created for the simple layout shown in Figure 3.26.

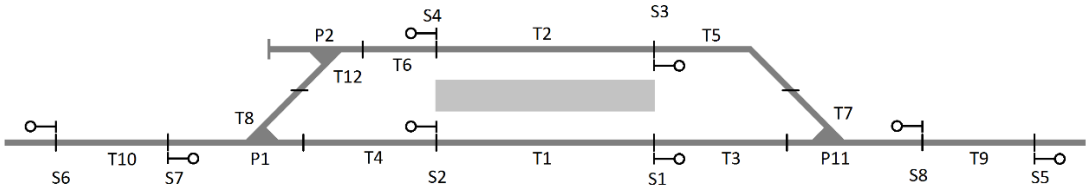

Figure 3.26 : A simple layout [1].

Table 3.1 : Example route table [1]

Route Table	Signals		Track Sections		Points		Flank Protection		Front Protection		Conflicting Routes
	Start	Exit	Path	Flank	Normal	Reverse	Point	Signal	Overlap	Points	
Route 1	S7	S1	T8, T4, T1	T12, T6	P1	-	P2 (N)	-	T3	P11 (R)	R2, R5, R7, R8
Route 2	S7	S3	T8, T12, T6, T2	T4	-	P1, P2	-	S2	T5	P11 (N)	R1, R6, R7, R8
Route 3	S1	S5	T3, T7, T9	T5	P11	-	-	S3	-	-	R4, R5, R6
Route 4	S3	S5	T5, T7, T9	T3	-	P11	-	S1	-	-	R3, R5, R6
Route 5	S8	S2	T7, T3, T1	T5	P11	-	-	S3	T4	P1 (R)	R1, R3, R4, R6
Route 6	S8	S4	T7, T5, T2	T3	-	P11	-	S1	T6	P1 (N)	R2, R3, R4, R5
Route 7	S2	S6	T4, T8, T10	T12, T6	P1	-	P2 (N)	-	-	-	R1, R2, R8
Route 8	S4	S6	T6, T12, T8, T10	T4	-	P1, P2	-	S2	-	-	R1, R2, R7

Created route table shows arrangements of wayside elements for each routes. First column is for route names. Second column is for the starting and exit signals of the routes. For example, it can be easily seen that route 1 starts from the signal 7 and ends signal 1. “Track sections” column shows the track sections on the path and on the flank areas. As it is mentioned before, track sections on flank areas also have to be clear for the safety of routes. “Points” column is created to show names and proper positions of the points on the route. Objects used for flank protection can be seen in the “Flank protection” column. The notations “(N)” and “(R)” represents the required positions of the points (N: normal, R: reverse). In the “Front Protection” column, overlaps and points used for front protection in the related routes are listed. Lastly, conflicting routes are written in the last column.

4. FORMAL METHODS

Formal methods are widely used for development and verification of software and hardware system as mathematical based techniques. According to European Standard EN 61508-7 formal methods described as: *"Formal methods provide a means of developing a description of a system during specification and/or implementation phase. These formal descriptions are mathematical models of the system function and/or structure. Therefore unambiguous system description could be achieved (e.g. any state of an automaton is described by its initial state, inputs and the transition equations of the automaton) which increase understanding of the underlying system."* [5].

Formal methods play an important role to increase the completeness, consistency or correctness of a specification or implementation, because formal methods transfer the principles of mathematical reasoning to the specification and implementation of technical systems [5] [26].

Software developers for railway systems must ensure their software provides high level of assurance. Testing these software usually takes so much time. Nevertheless, regardless how much software testing is performed, it cannot be guaranteed that the safety feature of developed software is fully satisfied [27]. For this reason, it is preferred to use formal methods in their design steps, because mathematical analysis can be performed easily to any design created with formal methods to contribute to the reliability and robustness of it.

Consequently, the goal of using formal methods is to produce an unambiguous and consistent specification that is as completely failure-free and with few contradictions as possible. However it must be simple to verify.

There are several formal methods used for safety critical system design [28] . In this thesis study, only petri nets and Finite State Machines have been explained. Because those two methods are widely used and their graphical design interface are more understandable comparing with other formal methods [29].

4.1 Petri Nets

Petri nets is one of the widely used formal method to model discrete event systems [30]. A Petri net consists of places, transitions, and arcs. Arcs run from a place to a transition or vice versa, never between places or between transitions. The places from which an arc runs to a transition are called the input places of the transition; the places to which arcs run from a transition are called the output places of the transition.

Places may have a number of marks called tokens. A transition is enabled when the input place has enough number of token. When enabled, it is permitted (but not obliged) to fire. If the transition fires, the input places to the transition loses their tokens, and each output place from the transition get the new tokens.

Graphically, places, transitions, arcs, and tokens are represented respectively by: circles, bars, arrows, and dots. See Figure 4.1.

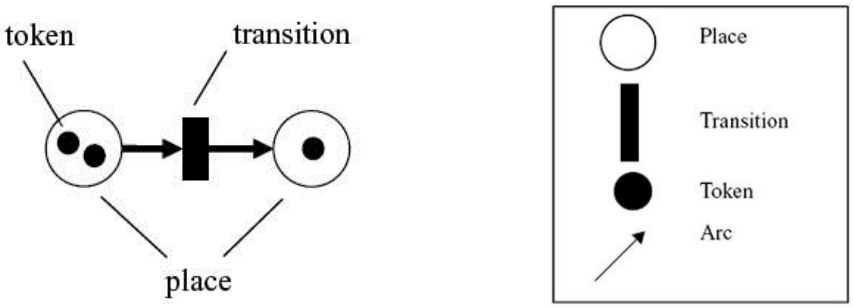

Figure 4.1 : A Simple Petri Net Model [1].

Petri net properties are listed below:

Sequential Execution: Transition t2 can be fired only after the firing of t1. This impose the precedence of constraints "t2 after t1" (Figure 4.2).

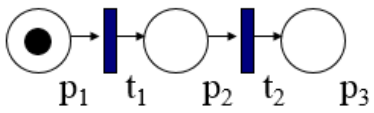

Figure 4.2 : Sequential Execution [1]

Synchronization: Transition t1 is enabled when there is at least one token at each of its input places. See Figure 4.3.

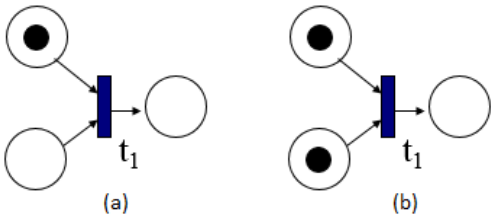

Figure 4.3 : Synchronization. (a): t1 is not enabled, (b): t1 is enabled [1].

Merging: It happens when tokens from several places arrive for service at the same transition. See Figure 4.4.

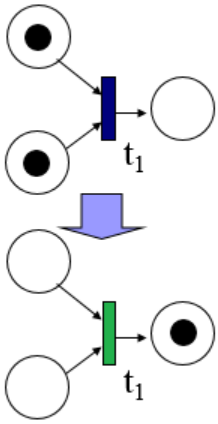

Figure 4.4 : Merging [1].

Concurrency: t1 and t2 are concurrent in the petri net shown in Figure 4.5. With this property, Petri net is able to model systems of distributed control with multiple processes executing concurrently in time.

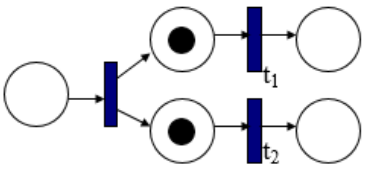

Figure 4.5 : Concurrency [1].

Conflict: t1 and t2 shown in Figure 4.6 are both ready to fire but the firing of any of them leads to the disabling of the other transitions.

The resulting conflict may be resolved in a purely non-deterministic way or in a probabilistic way, by assigning appropriate probabilities to the conflicting transitions. See Figure 4.7.

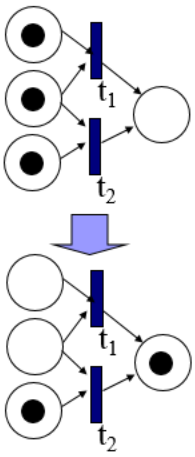

Figure 4.6 : Conflict [1].

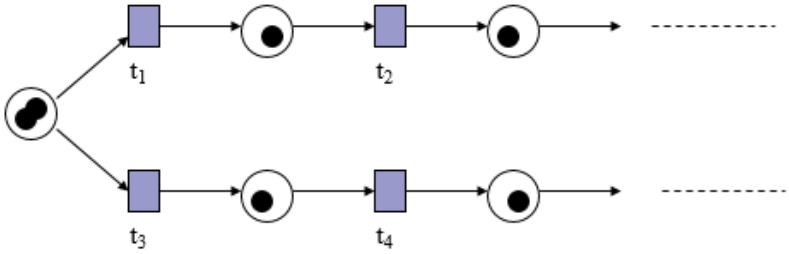

Figure 4.7 : There is a choice of either t1 and t2, or t3 and t4 [1].

The arcs can be defined with a weight value in some models. It is shown with the count of arcs between a place and a transition, or a small number over a single arc. See Figure 4.8.

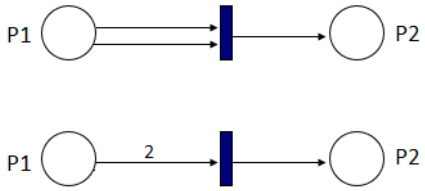

Figure 4.8 : Weight of the arcs [1].

In that case, if the number of token inside the input place equals or greater than weight of the input arc, then the transition is enabled (Figure 4.9). Otherwise, the transition cannot be fired.

It is not required to keep the number of token before and after firing. Following design in Figure 4.10 has 4 tokens on the left hand side and there is not any token on the right hand side. After firing, 3 tokens remain on the left hand side and places on the right hand side have 5 new tokens. The number of tokens is not kept.

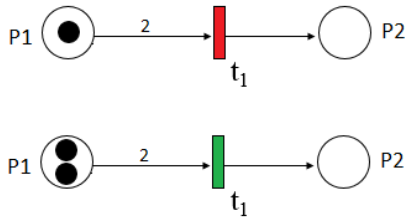

Figure 4.9 : Number of token and weight of the arc [1].

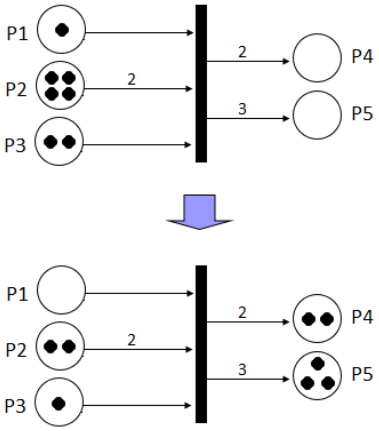

Figure 4.10 : Number of token is not kept [1].

The notation which used to show petri net graphs is given below [31]

$$(P, T, A, w)$$

Where

P is the finite set of *places* (one type of node in the graph)

T is the finite set of *transition* (the other type of node in the graph)

A is the set of arcs from places to transitions and transitions to places in the graph

w: A → {1, 2, 3, …} is the *weight function* on the arcs.

The set of places are represented by

$$P = \{p_1, p_2, \dots, p_n\} \quad \text{and} \quad |P| = n$$

The set of transitions are represented by

$$T = \{t_1, t_2, \dots, t_m\} \quad \text{and} \quad |T| = m$$

A typical arc function is represented by

$$A = \{(p_1, t_1), (t_1, p_1), \dots, (p_k, t_k), (t_k, p_k)\}$$

To show weight function of an arc following notation is used.

$$w(p_a, t_b) = c$$

Where, *c* is an integer shows the weight value.

To describing a petri net graph, it is convenient to use $I(t_j)$ to represent to set of input places to transition t_j. Similarly, $O(t_j)$ represents the set of output places from transition t_j. Thus, we have

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

Simple Example:
Consider a petri net graph defined by

$$P = \{p_1, p_2, p_3\} \qquad T = \{t_1, t_2\} \qquad A = \{(p_1, t_1,), (t_1, p_2), (p_2, t_2), (t_2, p_3)\}$$

$$w(p_1, t_1) = 2 \qquad w(t_1, p_2) = 1 \qquad w(p_2, t_2) = 3 \qquad w = (t_2, p_3) = 1$$

Therefore we have

$$I(t_1) = \{p_1\} \qquad I(t_2) = \{p_2\}$$

$$O(t_1) = \{p_2\} \qquad O(t_2) = \{p_3\}$$

It is so easy to create petri net diagram with these definitions. See Figure 4.11.

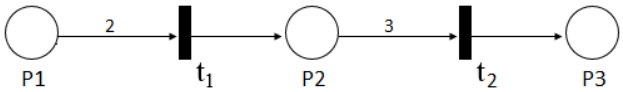

Figure 4.11 : Example Petri Net diagram [1].

4.2 Finite State Machines

FSM is one of the design methods mentioned in European Norms EN 61508-7 and the aim of the finite state machines is described in the same standard as to model, verify, specify or implement the control structure of a system. Furthermore, according to the same standard, finite state machines defined as; *"Many systems can be described in terms of their states, their inputs, and their actions. Thus when in state S1, on receiving input I a system might carry out action A and move to state S2. By describing a system's actions for every input in every state we can describe a system completely. The resulting model of the system is called a finite state machine (or finite state automata). It is often drawn as a so-called state transition diagram showing how the system moves from one state to another, or as a matrix in which the dimensions are state and input, and the matrix cells contain the action and new state resulting from receiving the input when in the given state."* [5].

A finite state machine (FSM) is a mathematical modeling technique which used to design both computer programs and sequential logic circuits [32]. States and transitions are the basic elements of a FSM. States represent the currently condition of the model and transitions represent the actions. The basic terms which used in FSM are described below.

State: A FSM is built around some finite collections of states. When modelling a device with a FSM, the conditions of the device are represented by states. If a point machine model is considered as an example, the conditions of being in normal and reverse position are represented by states. The number of states used in a FSM is finite. States are shown with a circle in a FSM diagram.

Current State: In a FSM, only one state can be active and it is called as the current state. In another words, the machine can only be in a single state in a certain time.

Starting State: Starting state is the initial state of a FSM. When a FSM is started to run, starting state will be current state automatically. Therefore, every finite state machine has a starting state. Initial state is identified by double circles in a FSM diagram.

Transition: Transitions make a connection between two states with some conditions. When the conditions occur, the current state changes one to another one which

connected by the transition. The conditions are defined by transition functions. Transition function is denoted as

$$f: X \times E \rightarrow X$$

Transitions are represented by arrows in the FSM diagrams. See Figure 4.12 for a simple FSM structure.

Event Set: Event set or input set is the set of all possible inputs in a FSM. For instance, in a FSM which designed for modeling a lift, control buttons of the lift can be considered as the input set.

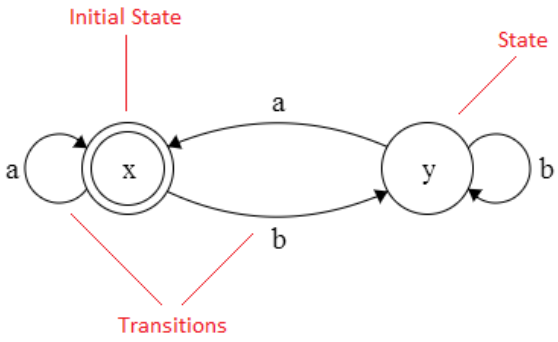

Figure 4.12 : FSM component [1].

The features of a FSM are described with a simple example below in Figure 4.13.

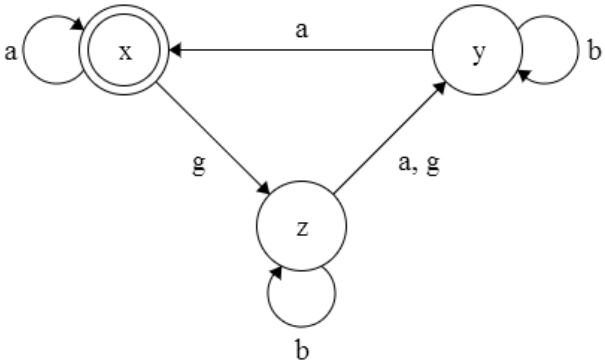

Figure 4.13 : A Finite State Machine diagram [1].

Event set for given FSM is

$$E = \{a, b, g\}$$

Set of states is

$$X = \{x, y, z\}$$

Transfer functions are

$$f(x,a)=x \qquad f(y,a)=x \qquad f(z,b)=z$$

$$f(x,g)=z \qquad f(y,b)=y \qquad f(z,a)=f(z,g)=y$$

The notation $f(y,a)=x$ means that if the model is in state *y*, then upon the occurrence of event *a*, the model will make an instantaneous transition to state *x*.

A FSM is built around some finite collections of states. Each state has a number and name. Event *a* may be occurred for any reason. It could be an external input or an event spontaneously generated by the system modelled by the finite state machine.

Three different cases can be seen on this example. First, an event may occur without changing the state. Transition function $f(x,a)=x$ can be given as an example to these event. When the active state is *x*, event *a* does not change the current state of FSM. In some FSM models, these ineffective transition functions are not defined.

Second case, occurrence of two different event may cause the same transition like $f(z,a)=f(z,g)=y$. Both events change the current state from *z* to *y*. However, it cannot be distinguished which event caused to this transition exactly.

Finally, as it can be easily realized from the diagram, there is not a transition function defined for every event. For instance, $f(y,g)$ is not defined and when the system in state *y*, occurrence of event g doesn't caused any transition.

After defining the basic terms, formal definition of the FSM can be given as

$$G=(X,E,f,\Gamma,x_0)$$

Where:

- X is the finite set of states.
- E is the finite set of events associated with G.
- $f: X\times E\rightarrow X$ is the transition function $f(x,e)=y$ means that there is a transition labeled by event e from state x to state y; in general, y is a partial function on its domain.
- $\Gamma=X\rightarrow 2^E$ is the active event function or feasible event function. $\Gamma(x)$ is the set of all events e for which $f=(x,e)$ is defined and is called the active event set of G at x.

- x_0 is the initial state.

A finite state machine diagram can be transformed to a petri net diagram. Following is a petri net diagram which transformed from Figure 4.13.

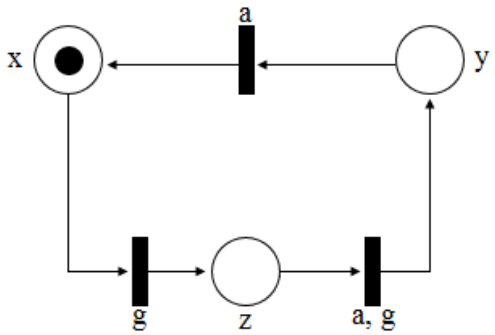

Figure 4.14 : A petri net diagram transformed from Figure 4.13 [1].

If a petri net diagram is designed with only single token and all arcs have 1 weight value, it can also be transformed to a finite state machine.

4.3 Formal Verification

Once designed the FSM diagram, it is easy to obtain mathematical equation from the diagram. Following formulate [33] is used to describe a transition function in a mathematical form.

$$S_i = \sum_{j=1}^{m} s_j \cdot T_{j,i} + s_i \cdot \prod_{k=1}^{n} \bar{T}_{i,k} \qquad \textbf{(4.1)}$$

$$i \neq j, i \neq k$$

Where,

- S_i is the new value of the i^{th} state.
- s_i or s_j are the current value of the related states.
- $T_{j,i}$ is the transition condition from s_j to s_i.
- $\bar{T}_{i,k}$ is the logical inverse of the transition condition from s_i to s_k.
- m is the number of outgoing transition from i^{th} state.
- n is the number of incoming transition to i^{th} state.

Following equations are implemented for given simple example FSM. Transitions which do not change the current state are ignored.

$$X = y \cdot a \ + x \cdot \bar{g}$$
$$Y = z \cdot (a \vee g) + \ y \cdot \bar{a} \qquad \textbf{(4.2)}$$
$$Z = x \cdot g + z \cdot \overline{a \vee g}$$

With using these equation, the current state of the FSM can be calculated for the next time interval. After starting the model, the initial state *x* will be current state. Therefore, the condition of states will be

$$S = \{x, y, z\} = \{1, 0, 0\} \qquad \textbf{(4.3)}$$

If event set is considered as

$$E = \{a, b, g\} = \{0, 0, 1\} \qquad \textbf{(4.4)}$$

It is possible to calculate the next current state with using the equations given above.

$$X = 0 \cdot 0 \ + 1 \cdot \bar{1}$$
$$Y = 0 \cdot (0 \vee 1) + \ 0 \cdot \bar{0} \qquad \textbf{(4.5)}$$
$$Z = 1 \cdot 1 + 0 \cdot \overline{0 \vee 1}$$

$$X = 0$$
$$Y = 0 \qquad \textbf{(4.6)}$$
$$Z = 1$$

New current state is *Z* and the state matrix will be

$$S = \{x, y, z\} = \{0, 0, 1\} \qquad \textbf{(4.7)}$$

It should be noted that, these are logical equations. Hence, “1” should be considered as logical “true” and “0” as logical “false”. Furthermore, the operator “+” means logical “or” and the operator “·” means logical “and”. Therefore, the solution of “1+1” should be “1” (1+1=1). The rest of the possible calculations are

$$0 + 0 = 0 \qquad 1 + 0 = 1 \qquad 0 \cdot 0 = 0 \qquad 1 \cdot 0 = 0 \qquad 1 \cdot 1 = 1$$

Consequently, both methods, petri nets and finite state machines, have some advantages and disadvantages. It is possible to model complicated systems with simple diagrams by using petri nets method. However, it is difficult to build the petri nets diagram. On the other hand, a FSM diagram can be easily built, but it expands so much when modelling the complex systems. In this thesis stud, FSM is used as a design method. Because, it is more understandable according to the petri nets.

4.3.1 An example model

In this topic, a turnstile controller is modeled with finite state machines method. With obtained model a PLC program will be written in the next chapter.

A turnstile shown in Figure 4.15 is a kind of gate which allows one person to pass at a time. In some suburban station, they are used to restrict passage only to people who insert a coin or a ticket. In this example, the working principle of a simple turnstile will be described, a finite state machine model will be designed and the mathematical equations will be obtained from the FSM model.

Using purpose and the working principle of the turnstile are listed below.

- Turnstile is used to prevent the unpaid passages. It allows the passage only to people who insert a coin in it.
- Initially the turnstile should be locked.
- Depositing a coin in a slot on the turnstile unlocks the turnstile arms, allowing a single customer to push through.
- After the customer passes through, the arms are locked again until another coin is inserted.
- Once a coin is inserted to the slot, an internal timer starts and if nobody does not push the arms in a certain time, the turnstile locks itself automatically.
- In an emergency case (fire, natural disaster, etc.), turnstile can be released continuously by an input.

- The slot box has a maximum coin capacity and it is measured with a counter device inside the slot mechanism. When the capacity is reached, turnstile blocks itself and does not allow the passage any more. In blocking state, the arms of turnstile are kept locked.
- The filled slot box is changed with an empty one by the staff. After changing the box, staff reset the counter and the turnstile starts to working normal again.

Figure 4.15 : A Turnstile [34].

According to the statements defined above, inputs or events of the model will be

- e_1 : Coin input
- e_2 : Turnstile pushed
- e_3 : Timer expired
- e_4 : Emergency
- e_5 : Slot is full
- e_6 : Reset

Event set

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6\} \quad \textbf{(4.8)}$$

Defining the states is not so difficult from the statements. However, the important point is that the number of states should be defined optimal. If different states are defined for every situation, that makes the model too complicated. It is not a important criteria for this simple design but in another exhaustive design, large number of states make impossible to analyze the model. Hence, the states for our model are defined below.

- S_1 : Turnstile is locked
- S_2 : Turnstile is unlocked
- S_3 : Turnstile is released continuously
- S_4 : Turnstile is blocked

State set;

$$S = \{S_1, S_2, S_3, S_4\} \quad \textbf{(4.9)}$$

After defining the events and states, the next step will be creating the transitions and the transition functions. Figure 4.16 shows the possible transitions between states.

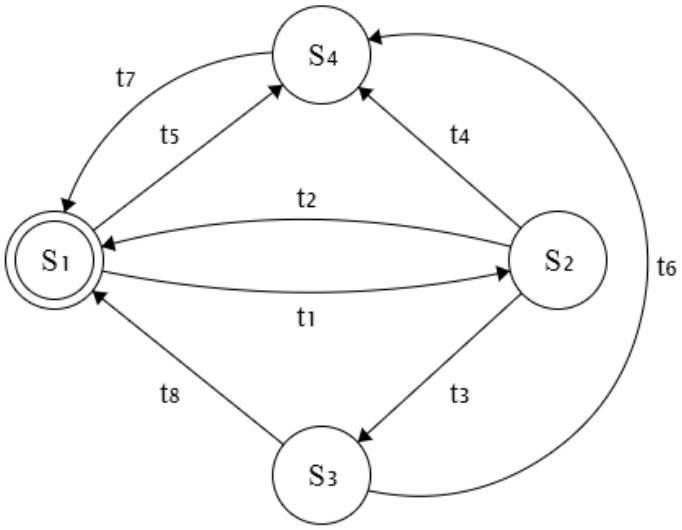

Figure 4.16 : FSM diagram [1].

Definitions of the transition functions;

$t_1 = f(S_1, S_2) = e_1$ (a coin is inserted to the slot)

$t_2 = f(S_2, S_1) = \bar{e}_5 \wedge (e_2 \vee e_3)$ (when slot box is not full, a passenger passed through the turnstile or passage waiting time expired)

$t_3 = f(S_2, S_3) = e_2 \wedge e_5$ (a passenger passed through the turnstile and slot box coin capacity has been exceeded)

$t_4 = f(S_2, S_4) = t_5 = f(S_1, S_4) = t_6 = f(S_3, S_4) = e_4$ (emergency input is true)

$t_7 = f(S_4, S_1) = \bar{e}_4$ (emergency input is false)

$t_8 = f(S_3, S_1) = e_6$ (reset)

Following Figure 4.17 shows the FSM diagram in terms of events.

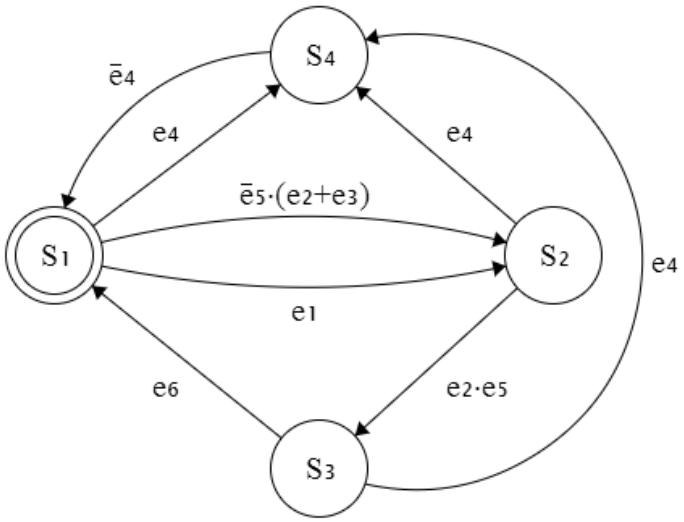

Figure 4.17 : FSM diagram in terms of events [1].

Finally, mathematical equations can be generated with using equation **(4.1)**

$$S_1 = s_2 \cdot t_2 + s_3 \cdot t_8 + s_4 \cdot t_7 + s_1 \cdot \bar{t}_1 \cdot \bar{t}_5$$

$$S_2 = s_1 \cdot t_1 + s_2 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_4$$

$$S_3 = s_2 \cdot t_3 + s_3 \cdot \bar{t}_6 \cdot \bar{t}_8 \qquad \textbf{(4.10)}$$

$$S_4 = s_1 \cdot t_5 + s_2 \cdot t_4 + s_3 \cdot t_6 + s_4 \cdot \bar{t}_7$$

And,

$$S_1 = s_2 \cdot \bar{e}_5 \cdot (e_2 + e_3) + s_3 \cdot e_6 + s_4 \cdot \bar{e}_4 + s_1 \cdot \bar{e}_1 \cdot \bar{e}_4$$

$$S_2 = s_1 \cdot e_1 + s_2 \cdot \overline{\bar{e}_5 \cdot (e_2 + e_3)} \cdot (\overline{e_2 \cdot e_5}) \cdot \bar{e}_4$$

$$S_2 = s_1 \cdot e_1 + s_2 \cdot \left(e_5 + (\overline{e_2 + e_3})\right) \cdot (\overline{e_2 \cdot e_5}) \cdot \bar{e}_4 \text{ (De Morgan's law)}$$

$$S_2 = s_1 \cdot e_1 + s_2 \cdot \left(e_5 + (\bar{e}_2 \cdot \bar{e}_3)\right) \cdot (\bar{e}_2 + \bar{e}_5) \cdot \bar{e}_4 \text{ (De Morgan's law)} \qquad \textbf{(4.11)}$$

$$S_3 = s_2 \cdot (e_2 \cdot e_5) + s_3 \cdot \bar{e}_4 \cdot \bar{e}_6$$

$$S_4 = s_1 \cdot e_4 + s_2 \cdot e_4 + s_3 \cdot e_4 + s_4 \cdot e_4$$

The initial state is S_1(see Figure 4.18). Hence,

$$S_{start} = S_{t_0} = \{1, 0, 0, 0\} \tag{4.12}$$

$$E_{initial} = E_{t_0} = \{0, 0, 0, 0, 0, 0\} \tag{4.13}$$

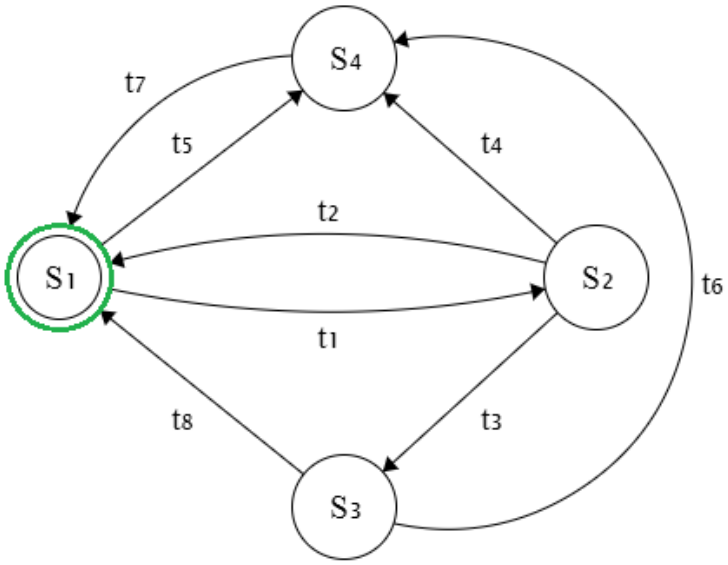

Figure 4.18 : Initial view of the FSM [1].

As a first case, if a passenger inserts a coin, the event matrix will be

$$E_{t_1} = \{1, 0, 0, 0, 0, 0\}$$

New current state in t_1 can be calculated

$$\begin{aligned} S_{1_{t_1}} &= 0 \cdot (0 + 0) + 0 \cdot 0 + 0 \cdot \bar{0} + 1 \cdot \bar{1} \cdot \bar{0} \\ S_{2_{t_1}} &= 1 \cdot 1 + 0 \cdot \left(0 + (\bar{0} \cdot \bar{0})\right) \cdot (\bar{0} + \bar{0}) \cdot \bar{0} \\ S_{3_{t_1}} &= 0 \cdot (0 \cdot 0) + 0 \cdot \bar{0} \cdot \bar{0} \\ S_{4_{t_1}} &= 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 \end{aligned} \tag{4.14}$$

And,

$$\begin{aligned} S_1 &= 0 \\ S_2 &= 1 \\ S_3 &= 0 \end{aligned} \tag{4.15}$$

$$S_4 = 0$$

$$S_{t_1} = \{0, 1, 0, 0\} \tag{4.16}$$

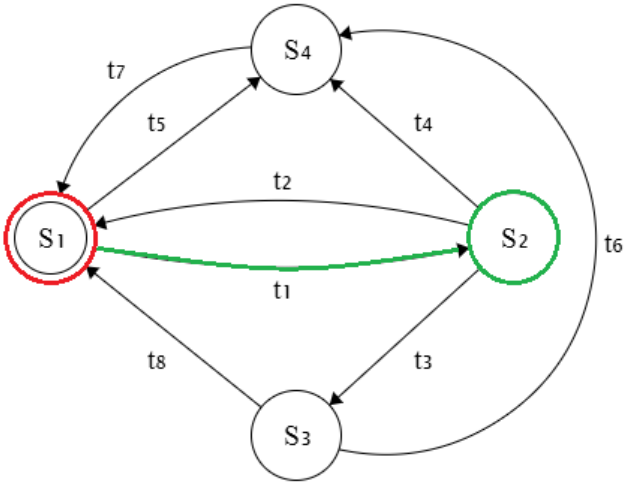

Figure 4.19 : New Current State is S_2 [1].

As it can be seen from the results, new state of the model will be S_2 (see Figure 4.19). Same steps can be repeated to calculate the next step t_2 with assuming that the passenger passed through turnstile on time.

New event matrix will be

$$E_{t_2} = \{0, 1, 0, 0, 0, 0\} \tag{4.17}$$

To find the new state

$$\begin{gathered} S_{1_{t_2}} = 1 \cdot (1 + 0) + 0 \cdot 0 + 0 \cdot \bar{0} + 0 \cdot \bar{0} \cdot \bar{0} \\ S_{2_{t_2}} = 0 \cdot 0 + 1 \cdot \big(0 + (\bar{0} \cdot \bar{0})\big) \cdot (\bar{0} + \bar{0}) \cdot \bar{0} \\ S_{3_{t_2}} = 1 \cdot (1 \cdot 0) + 0 \cdot \bar{0} \cdot \bar{0} \\ S_{4_{t_2}} = 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 \end{gathered} \tag{4.18}$$

And,

$$\begin{gathered} S_{1_{t_2}} = 1 \\ S_{2_{t_2}} = 0 \end{gathered} \tag{4.19}$$

$$S_{3_{t_2}} = 0$$

$$S_{4_{t_2}} = 0$$

$$S_{t_2} = \{1, 0, 0, 0\} \quad \textbf{(4.20)}$$

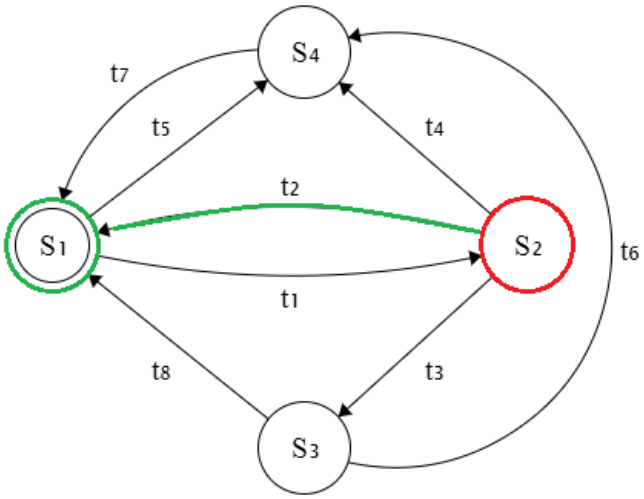

Figure 4.20 : Current State is S_1 again [1].

According to the results, the model passed back into the initial state as shown in Figure 4.20.

Thirdly, the emergency case is simulated. In an the emergency case, new event matrix will be

$$E_{t_3} = \{0, 0, 0, 1, 0, 0\} \quad \textbf{(4.21)}$$

The response of the model to this input matrix will be

$$S_{1_{t_3}} = 0 \cdot (0 + 0) + 0 \cdot 0 + 0 \cdot \bar{1} + 1 \cdot \bar{0} \cdot \bar{1}$$

$$S_{2_{t_3}} = 1 \cdot 0 + 0 \cdot \left(0 + (\bar{0} \cdot \bar{1})\right) \cdot (\bar{0} + \bar{0}) \cdot \bar{0}$$

$$S_{3_{t_3}} = 0 \cdot (0 \cdot 0) + 0 \cdot \bar{1} \cdot \bar{0} \quad \textbf{(4.22)}$$

$$S_{4_{t_3}} = 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1$$

And,

$$S_{1_{t_3}} = 0$$

$$S_{2_{t_3}} = 0$$

$$S_{3_{t_3}} = 0 \qquad \textbf{(4.23)}$$

$$S_{4_{t_3}} = 1$$

$$S_{t_3} = \{0, 0, 0, 1\} \qquad \textbf{(4.24)}$$

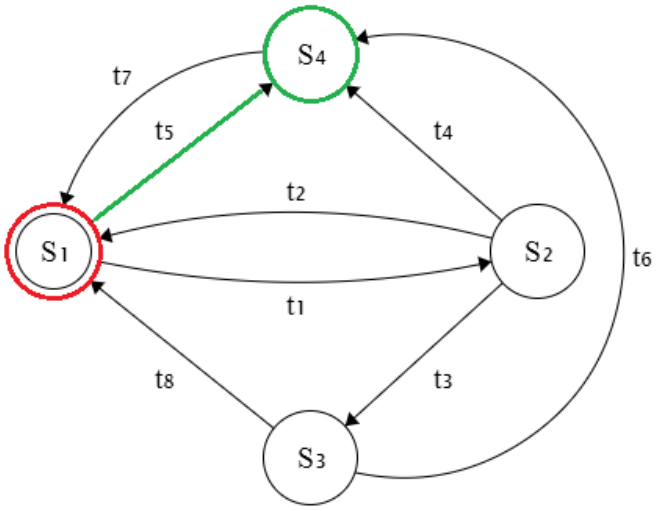

Figure 4.21 : New Current State is S_4 [1].

The new state is S_4 now (Figure 4.21). This is exactly the same result with the expected situation. Because, an emergency signal is received and the turnstile has to pass into the S_4.

Finally, state conditions are calculated for a passenger passage when the turnstile is still in the emergency state.

Event matrix for the new situation

$$E_{t_4} = \{0, 1, 0, 1, 0, 0\} \qquad \textbf{(4.25)}$$

Behavior of the model are obtained as

$$S_{1_{t_4}} = 0 \cdot (1 + 0) + 0 \cdot 0 + 1 \cdot \bar{1} + 0 \cdot \bar{0} \cdot \bar{1}$$

$$S_{2_{t_4}} = 0 \cdot 0 + 0 \cdot \big(0 + (\bar{1} \cdot \bar{0})\big) \cdot (\bar{1} + \bar{0}) \cdot \bar{1} \qquad \textbf{(4.26)}$$

$$S_{2_{t_4}} = 0 \cdot 0 + 0 \cdot (\overline{1+0}) \cdot (\overline{1 \cdot 0}) \cdot \bar{1}$$

$$S_{3_{t_4}} = 0 \cdot (1 \cdot 0) + 0 \cdot \bar{1} \cdot \bar{0}$$

$$S_{4_{t_4}} = 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 1$$

And,

$$\begin{aligned} S_{1_{t_4}} &= 0 \\ S_{2_{t_4}} &= 0 \\ S_{3_{t_4}} &= 0 \\ S_{4_{t_4}} &= 1 \end{aligned} \tag{4.27}$$

$$S_{t_4} = \{0, 0, 0, 1\} \tag{4.28}$$

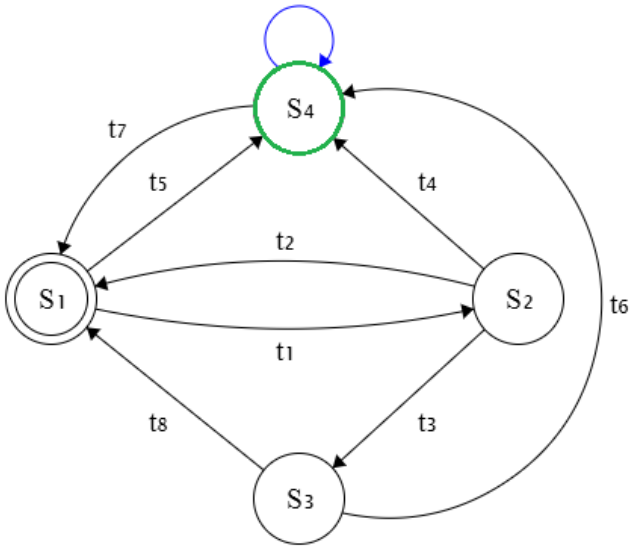

Figure 4.22 : Current state didn't change [1].

The new state matrix shows that the current state will not be changed when a person used the turnstile. See Figure 4.22.

Consequently, four possible scenarios have been simulated and the model has been responded with expected results. Other possible cases can also be simulated, however four scenarios are sufficient to show working principle of a finite state machine. As mentioned before, an implementation of the model with a PLC programming tool will be handled in the next chapter.

4.4 Implementation

Models created with formal methods can be implemented easily with several programming language. In this study, PLC based systems are considered. Therefore, the models created with formal methods will be implemented with PLC programming software in this section.

Once the model is designed and the mathematical equations are obtained, these equations can be defined with the relevant programming language into the implementation software. Obtained equations from the model are simple logical equations and it is not difficult to define them with any programming software. In this chapter, the example model which designed in the previous chapter will be implemented and simulated with two different PLC programming languages.

4.4.1 Ladder diagram

Ladder diagram is a typical technique which used to design relay based system. It is also used in some PLC software as a programming language. Ladder diagram provides a graphical programming logic and it is understandable and easy to design. Siemens Simatic Manager is used as an implementation tool.

To begin with, the basic notations used in ladder logic and their equivalents in the FSM were listed below [35].

---| |--- Normally Open Contact: It represents an event in the FSM.

---| / |--- Normally Closed Contact: It will used as the logical invers of an event in the FSM.

---( ) Output Coil: It represents the States in the FSM

The combinations of contacts and coils will represent the transition functions. Furthermore, some internal functions will be used such as; timer function.

FSM model diagram of the turnstile example designed in the previous chapter is given again in following Figure 4.23. The model will be described as a function block in the software.

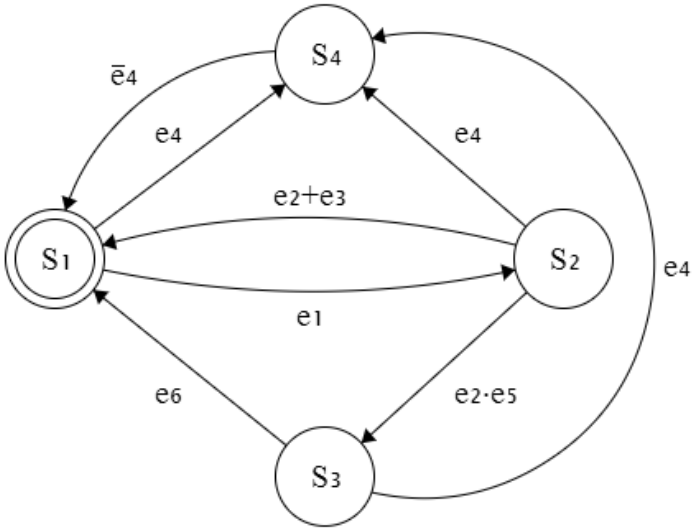

Figure 4.23 : FSM model of the turnstile example [1].

Firstly, the events and the states are created as inputs and outputs (Figure 4.24). $S_{1_{new}}$, $S_{2_{new}}$, $S_{3_{new}}$ and $S_{4_{new}}$ are defined to save the new values of the states.

Symbol Editor - [S7-Programm(1) (Symbole) -- turnstäle\SIMATIC 300(1)\CP...

Tabelle Bearbeiten Einfügen Ansicht Extras Fenster Hilfe

Alle Symbole

	Status	Symbol	Adresse	Datentyp	Kommentar
1		e1	E 0.0	BOOL	Coin input
2		e2	E 0.1	BOOL	Turnstile Pushed
3		e3	M 100.0	BOOL	Timer expired
4		e4	E 0.3	BOOL	Emergency
5		e5	E 0.4	BOOL	Slot is full
6		e6	E 0.5	BOOL	Reset
7		S1	A 10.0	BOOL	Turnstile is locked
8		S2	A 10.1	BOOL	Turnstile is unlocked
9		S3	A 10.2	BOOL	Turnstile is blocked
10		S4	A 10.3	BOOL	Turnstile is released continuously
11		S1_new	M 100.2	BOOL	New value of S1
12		S2_new	M 100.3	BOOL	New value of S2
13		S3_new	M 100.4	BOOL	New value of S3
14		S4_new	M 100.5	BOOL	New value of S4
15		Timer	T 1	TIMER	Internal timer for passage waiting time
16					

NUM

Figure 4.24 : Variable list created in the software [1].

Secondly, transition equations are defined. See following Figure 4.25, Figure 4.26, Figure 4.27 and Figure 4.28.

$$S_1 = s_2 \cdot (e_2 + e_3) + s_3 \cdot e_6 + s_4 \cdot \bar{e}_4 + s_1 \cdot \bar{e}_1 \cdot \bar{e}_4 \qquad \textbf{(4.29)}$$

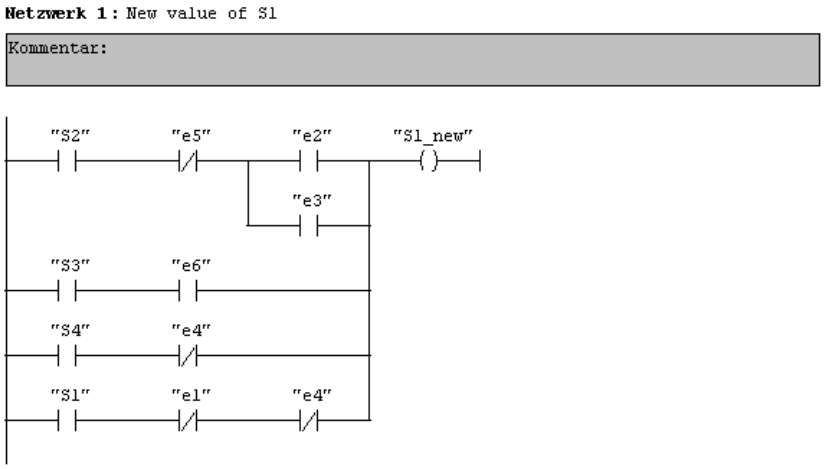

Figure 4.25 : Definition of S_1 transition equation by ladder diagram [1].

$$S_2 = s_1 \cdot e_1 + s_2 \cdot (\bar{e}_2 \cdot \bar{e}_3) \cdot (\bar{e}_2 + \bar{e}_5) \cdot \bar{e}_4 \qquad \textbf{(4.30)}$$

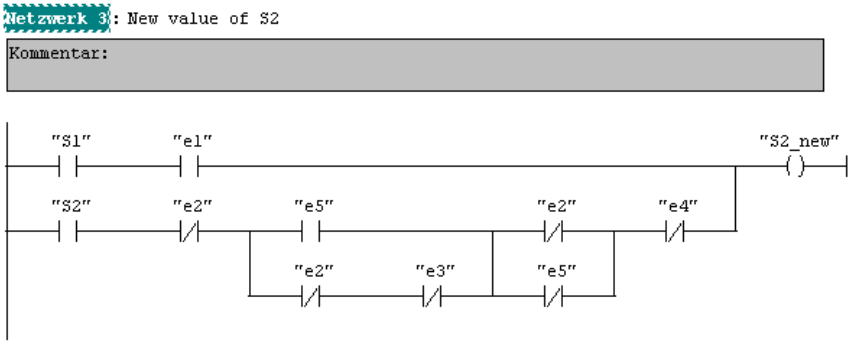

Figure 4.26 : Definition of S_2 transition equation by ladder diagram [1].

$$S_3 = s_2 \cdot (e_2 \cdot e_5) + s_3 \cdot \bar{e}_4 \cdot \bar{e}_6 \qquad \textbf{(4.31)}$$

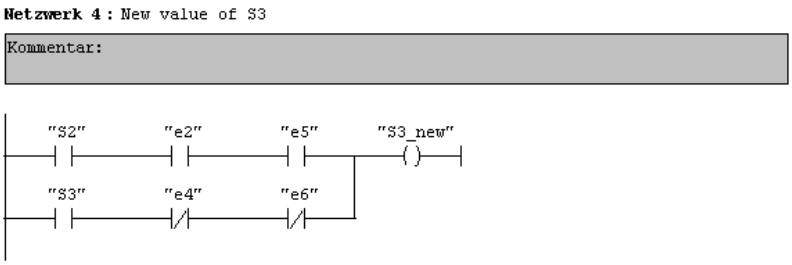

Figure 4.27 : Definition of S_3 transition equation by ladder diagram [1].

$$S_4 = s_1 \cdot e_4 + s_2 \cdot e_4 + s_3 \cdot e_4 + s_4 \cdot e_4 \qquad \textbf{(4.32)}$$

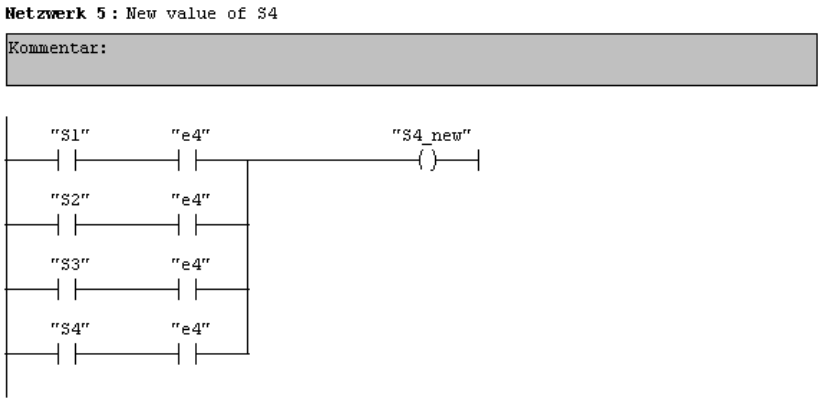

Figure 4.28 : Definition of S_4 transition equation by ladder diagram [1].

These definitions calculate the new values of the states and following codes are written to assign new calculated values (Figure 4.29).

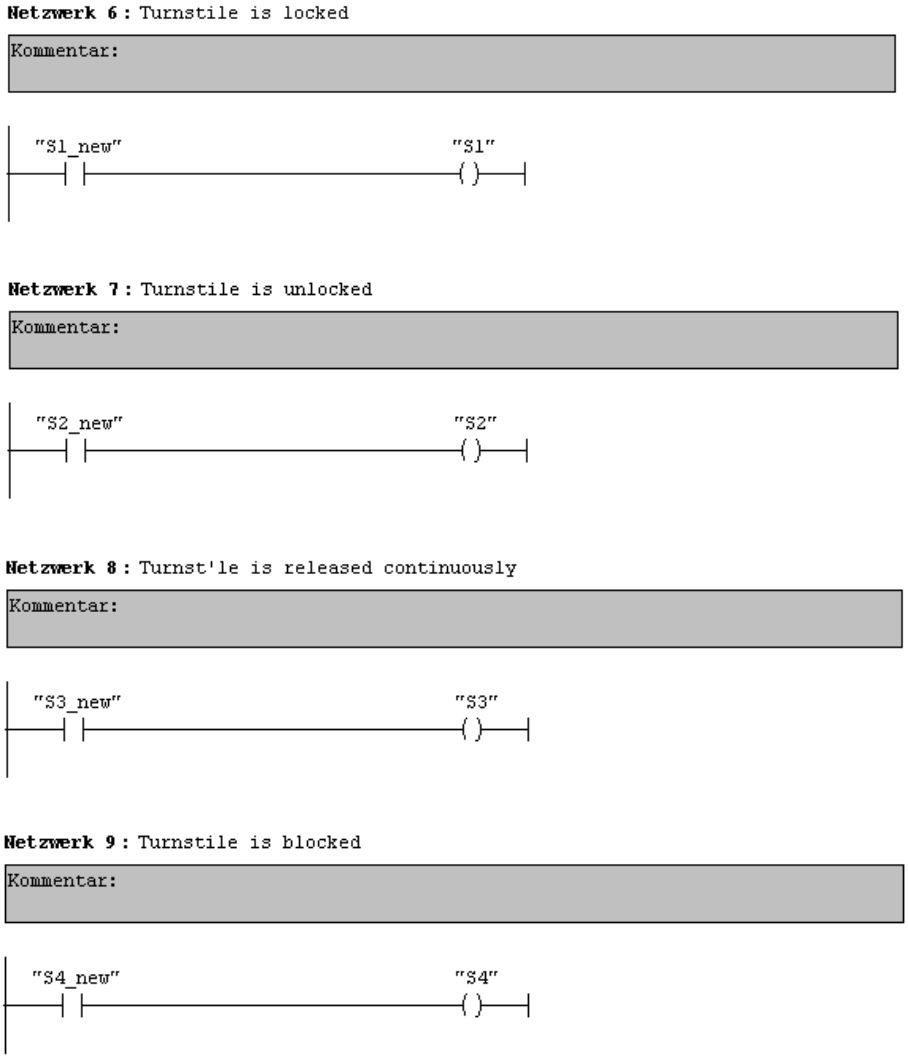

Figure 4.29 : Codes for assign new values to the states [1].

Lastly, a timer is created to obtain a time limit after inserted a coin. It starts when S_2 is activated. If the current states is not chanced in a certain time, timer sets e_3 "true". However, if the current state is changed with S_1, timer will be stopped (Figure 4.30).

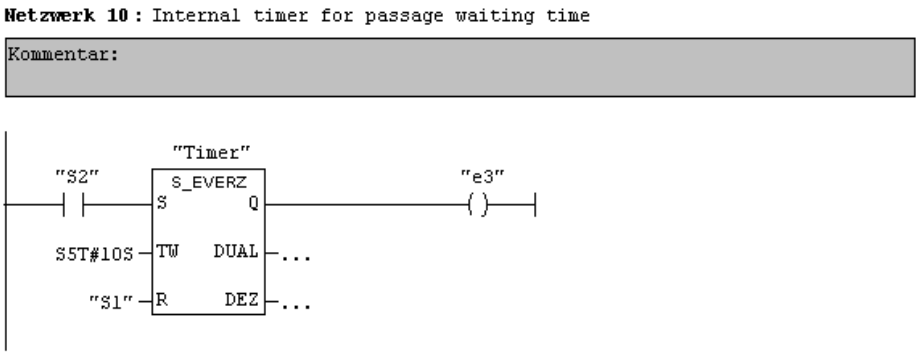

Figure 4.30 : Internal timer to obtain a time limit after inserted a coin [1].

A function block is created with the codes given above (Figure 4.31).

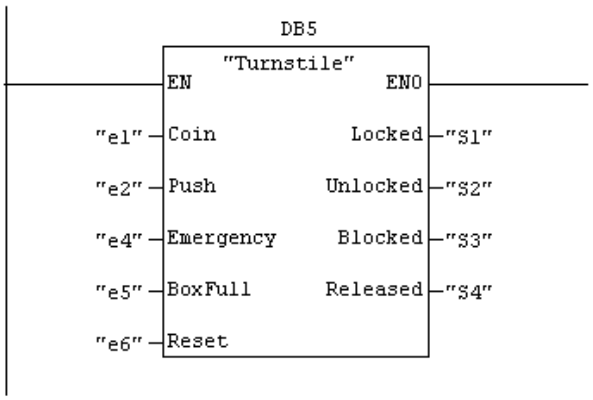

Figure 4.31 : Created function block [1].

Used PLC programming tool has also a simulation option. It can be used to simulate the created program and seen the behavior of developed model without any PLC hardware. All cases are simulated with this tool and model responded all inputs correctly. Here, the responses of the model to the cases which considered in the previous chapter are shown in Figure 4.32, Figure 4.33, Figure 4.34 and Figure 4.35.

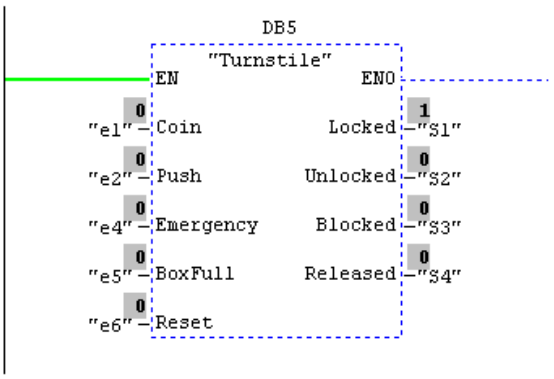

Figure 4.32 : Initial condition of the model [1].

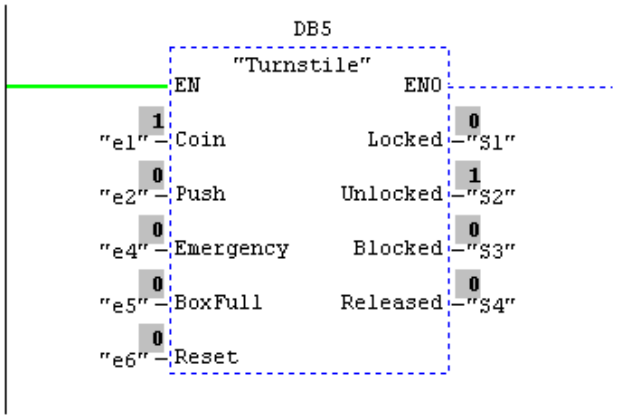

Figure 4.33 : A coin inserted to the slot [1].

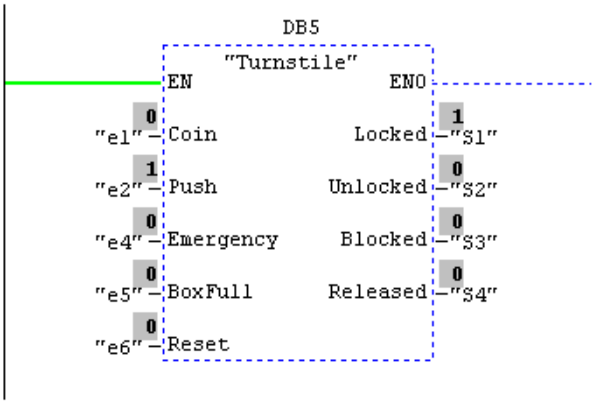

Figure 4.34 : It returns to initial state when the turnstile arms pushed [1].

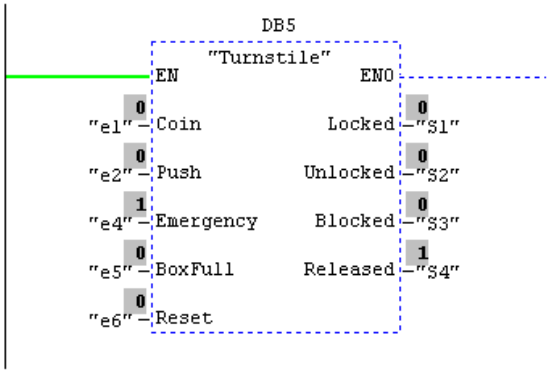

Figure 4.35 : In an emergency input it release the turnstile [1].

4.4.2 Sequential function chart

Sequential function chart (SFC) is a graphical programming language in accordance with IEC 61131-3 [36]. This graphical language is based on the step-transition model. The sequential function chart is used to divide complex tasks into smaller units and define the interaction between these units. In particular, this programming language is suitable for the tasks in which the individual functions are integrated into an overall process.

A SFC program is comprised of Steps, Actions and Transitions. In a SFC network, only one step can be active at a time. A step enabling condition controls when the transition occurs. Only when the step enabling condition is met, new steps can be branched.

One of the main advantage of using SFC is that, its component are almost completely fulfil the component of the finite state machines.

SilworX is chosen as a PLC programming tool to implement the same turnstile model. The basic component used in SFC language are explained firstly.

Step: SFC steps describe states within SFC networks (Figure 4.36). The states in the FSM model will represented with steps in SFC language. Definition of the step in IEC 61131-3 is :*"A step represents a situation in which the behavior of a program organization unit with respect to its inputs and outputs follows a set of rules defined by the associated actions of the step. A step is either active or inactive. At any given moment, the state of the program organization unit is defined by the set of active steps and the values of its internal and output variables."* [37]

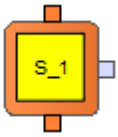

Figure 4.36 : Step symbol [1].

The initial step: initial step is the first step of a step chain and is active at the beginning of a program run (Figure 4.37). It will represent the initial state in the FSM model.

Figure 4.37 : Initial step symbol [1].

Transitions: SFC transitions describe state changes. According to IEC 61131-3, a transition specifies the condition under which the control passes from one or multiple steps preceding the transition to one or multiple steps following the transition along the corresponding connection. It has exactly the same functionality with FSM transition (Figure 4.38).

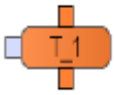

Figure 4.38 : Transition Symbol [1].

The short description of the other component used in the SilwoX were given below.

Standard Input: Events defined in the FSM model will be represented by standard inputs in SFC language (Figure 4.39).

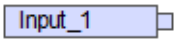

Figure 4.39 : A standard input symbol [1].

The little white circle at the left hand side of the input symbol means that it is an inverted input (Figure 4.40).

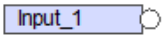

Figure 4.40 : An inverted input [1].

Connector: An SFC connector represents the logical connection between a data source and a data sink. There are two type of connector; input and output connector (Figure 4.41).

Figure 4.41 : Input and output connector symbols [1].

A finite state machine model can be programmed easily with SFC language without any need to mathematical equations. It can be created using only FSM model diagram. The turnstile model is implemented in SilworX by using FSC language.

Firstly, inputs and outputs are created (Figure 4.42).

Global Variables | Blocks | Local Variables | Connectors | Instances | System Variables

	Name	Data type	Initial Value	Description	Additional Comment	Technical Unit	Retain	Constant	Variable Type	Invert	Sequence Number	Activate Value Field
1	e1	BOOL		Coin input			☐	☐	VAR_INPUT	☐	1	☐
2	e2	BOOL		Turnstile pushed			☐	☐	VAR_INPUT	☐	2	☐
3	e3	BOOL		Timer expired			☐	☐	VAR	☐		☐
4	e4	BOOL		Emergency			☐	☐	VAR_INPUT	☐	3	☐
5	e5	BOOL		Slot is full			☐	☐	VAR_INPUT	☐	4	☐
6	e6	BOOL		Reset			☐	☐	VAR_INPUT	☐	5	☐
7	ENO	BOOL	FALSE				☐	☐	VAR_OUTPUT	☐	0	☐
8	S1	BOOL		Locked			☐	☐	VAR_OUTPUT	☐	1	☐
9	S2	BOOL		Unlocked			☐	☐	VAR_OUTPUT	☐	2	☐
10	S3	BOOL		Blocked			☐	☐	VAR_OUTPUT	☐	3	☐
11	S4	BOOL		Released			☐	☐	VAR_OUTPUT	☐	4	☐
12	Time	TIME		Time limit timer			☐	☐	VAR_OUTPUT	☐	5	☐

Figure 4.42 : Described variables [1].

Using created variables, all states are described with regard to FSM diagram (Figure 4.43) as shown in Figure 4.44.

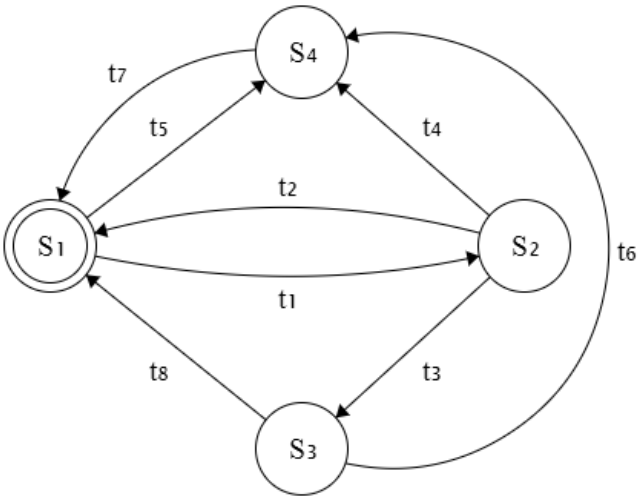

Figure 4.43 : Turnstile FSM diagram [1].

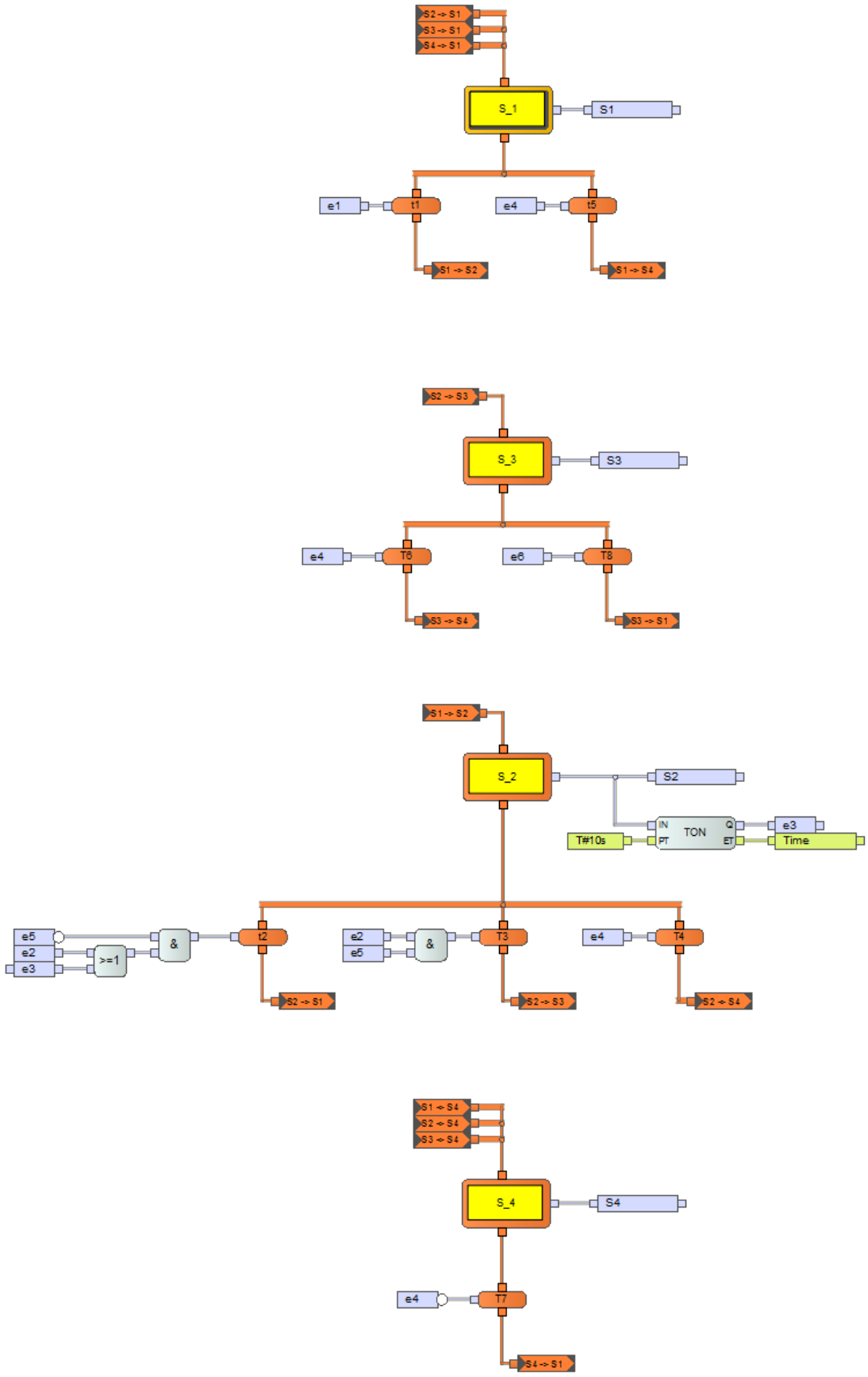

Figure 4.44 : All described states [1].

$$t_1 = f(S_1, S_2) = e_1$$

$$t_2 = f(S_2, S_1) = \bar{e}_5 \wedge (e_2 \vee e_3)$$

$$t_3 = f(S_2, S_3) = e_2 \wedge e_5$$

$$t_4 = f(S_2, S_4) = t_5 = f(S_1, S_4) = t_6 = f(S_3, S_4) = e_4 \quad \textbf{(4.33)}$$

$$t_7 = f(S_4, S_1) = \bar{e}_4$$

$$t_8 = f(S_3, S_1) = e_6$$

Created function block is shown in the following Figure 4.45. Timer value is also assigned as an output to make possible to be monitored.

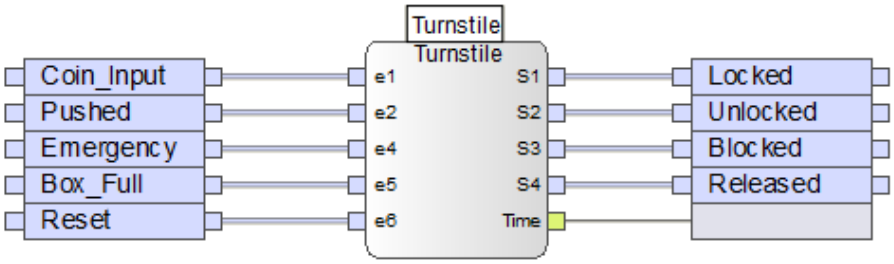

Figure 4.45 : Created function block [1].

SilworX has also an offline simulation option. Created programs in SilworX can be simulated without any needs to a real PLC device. Created modes are tested via offline simulator and it is seen that behavior of the model to the all possible cases is the same with expected results. Status of the model when the simulation has just started can be seen in Figure 4.46.

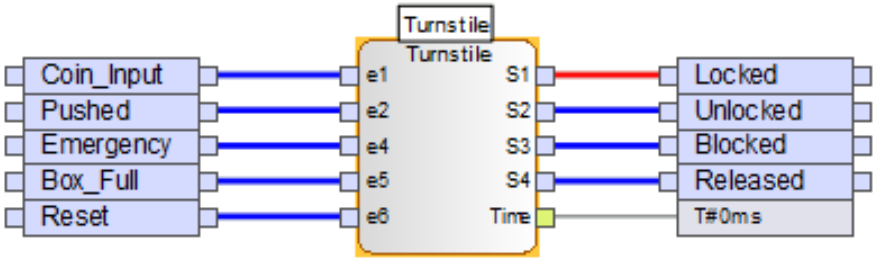

Figure 4.46 : Status of the model when the simulation has just started [1].

To simulate coin insert, e_1 is forced to "true". The actual state of the model is changed with S_2 "unlocked" and the timer began to measure time period. If the passenger pass

through the turnstile, the arms are locked and the timer is stopped. The behavior of the created function block can be seen in Figure 4.47.

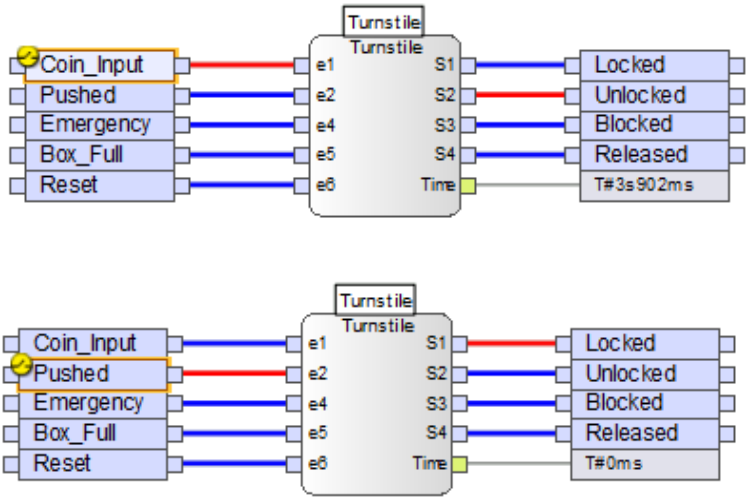

Figure 4.47 : Passenger passage simulation [1].

If the time expired, the model locks the turnstile immediately as seen in Figure 4.48.

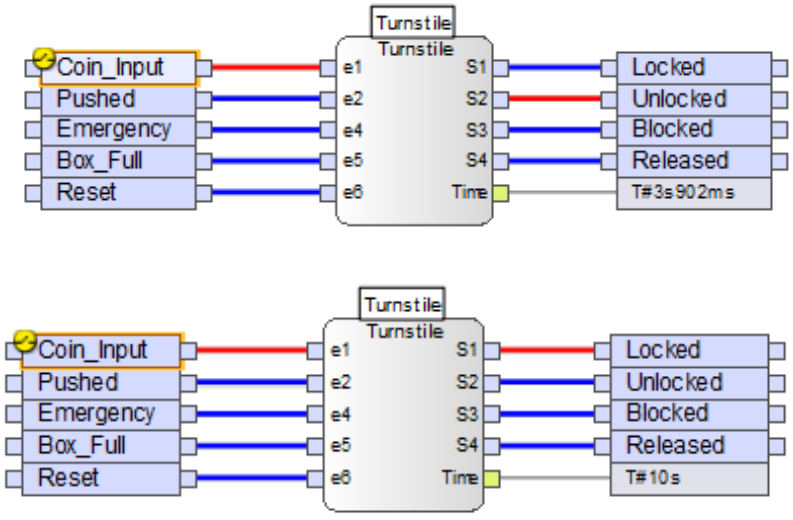

Figure 4.48 : Time limit expire simulation [1].

The situation of the last passage when the coin box gives the “box is full” input is tested below. When the last coin inserted, the model waiting for the passage of people and then blocks the turnstile. See Figure 4.49.

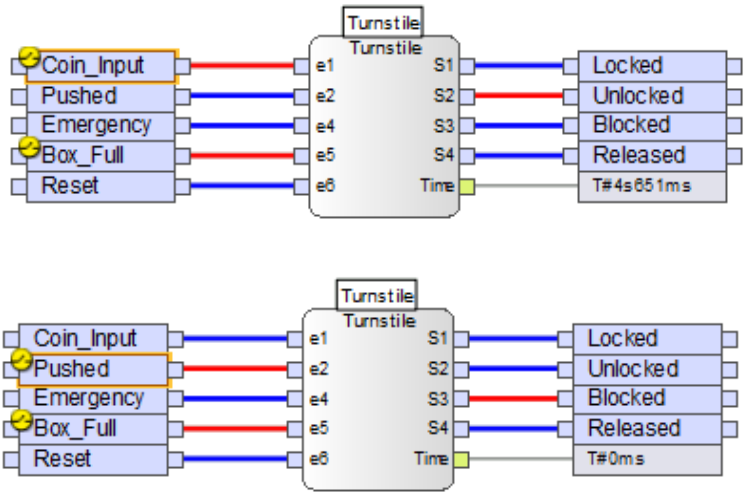

Figure 4.49 : Turnstile blocking simulation [1].

In an emergency case, model release the turnstile mechanism immediately (Figure 4.50).

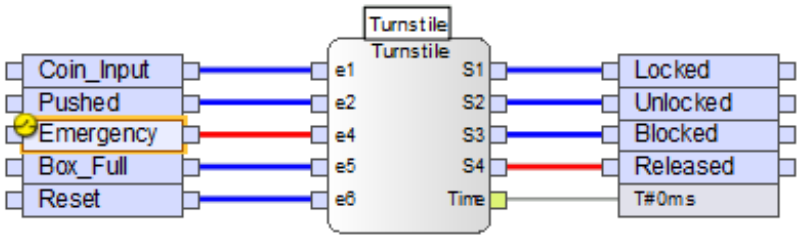

Figure 4.50 : Emergency case simulation [1].

As it can be seen from the implementations, a finite state machine model can be easily defined with a PLC programming software.

As a conclusion, both software and both software language can be used to implement the formal models. However, sequential function charts has a more practical designing interface. In addition to that, it is not necessary to obtain mathematical equations to implement a FSM in SFC language. For that reasons, Hima-SilworX and SFC have been used as implementation tool and programming language in this thesis study.

5. MODEL STATION DESIGN

One of the main topics of this thesis study is creation of an example railway interlocking model. To achieve this purpose, a model railway station is needed. All designs would be developed according to the specifications of the model station. It should be noted that, German design principles are considered in this study.

Some considered criteria when creating the model station are listed below.

- The model station should be a main line station. Because, a metro or tram station is quite simple relatively.
- It must have the same basic specifications with a real railway station. Otherwise, developed model won't be realistic.
- It should include almost all basic wayside equipment.
- It mustn't be so complicated. Because, a developed model for a basic simple station can be adapted to any complicated station.

With respect to these criteria, layout shown in Figure 5.1 is created.

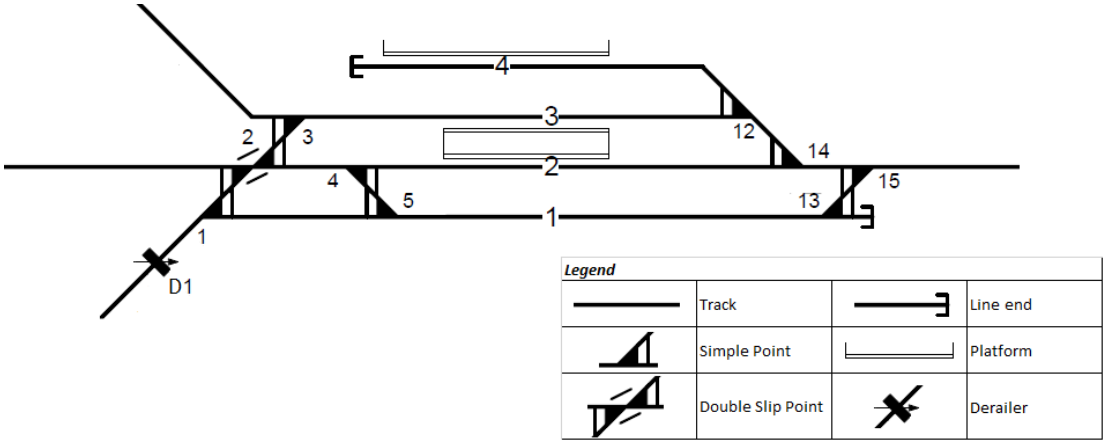

Figure 5.1 : Model Station layout [1]

It is a small size main line station. It is designed as a station in an intersection area of two different line. The parallel line to the station layout is called main line. And the other line which linked to the station from the northwestern is called side line. Main line will be used more frequently than the side line. There is also one more way at the

bottom of station layout. However, it is an entrance to the station form a depot area or an industrial area. Therefore, a derailer located on it to protect the station from any hazardous uncontrolled vehicle may come from the depot area.

There are four tracks and three platforms in the station. First track has no platform, because it is not be designed for the passenger's usage. It can be used a shunting area or a parking track. Furthermore, fourth track is designed as a siding. Trains may enter the platform 4 from the east side and cannot proceed anymore because there is a line end. There are also one more line end at the ride hand side of the first track.

The points are placed to provide trains passing between the tracks. Two types of point are used in the layout. The point 2 located at the west of station is chosen a double slip point. Because the place on that are is considered to be not so wide to using two simple points. The rest of the points are defined as simple point.

5.1 Operational Concept

The operation concept of the model station is described according to a small size station's features. All required operational specifications are described below.

5.1.1 Train types

The train types considered in this study are used in German Railways.

- RB (Regionalbahn), is a type of local passenger train. RB trains usually stop at all stations on a given line to the passengers. The length of the RB vehicles is relatively short and their maximum allowable speeds are less than the other type of trains.
- RE (Regional-Express), is another train type used in Germany and some other countries. Its rank above than RB because it regularly stops only at selected stations on its route and the passenger capacity of RE trains is higher than RB trains. The vehicle length of the RE trains are more long and their maximum speed can be reach 160 km/h.
- IC (Intercity), is higher classification train type. IC trains service more comfortable journeys over long-distances. Their passenger capacity and travelling speed is high, train length is long.

- Freight Trains are used to transport goods from one point to another point. They do not stop any intermediate station in their route except due to the fact that signal authority or abnormal reasons. Freight trains have usually very long vehicle length.

5.1.2 Lines characteristics

As it mentioned before, there are 3 types of line which have different characteristics. See Figure 5.2.

- Main line, has a speed limit 120 km/h and it is used by RE, IC and Freight Trains. RE trains come from both directions every hour and stop at track 2. IC trains come from both direction every two hours. They also use track two but they pass without stop. Freight Trains run over track 2 or track 1 every two hours without stopping.
- Side Line, is used by RB trains and its allowable speed is 80 km/h. RB trains journey frequency is described as one train in every hour. They enter the station on track 3.
- Depot Line, is a gate way between industrial depot and main line. Therefore the Freight Trains and the shunting trains use it. Its speed limit is 25 km/h. Freight Trains run over the depot line and enter the station on track 1. Then, they proceed through the main line.

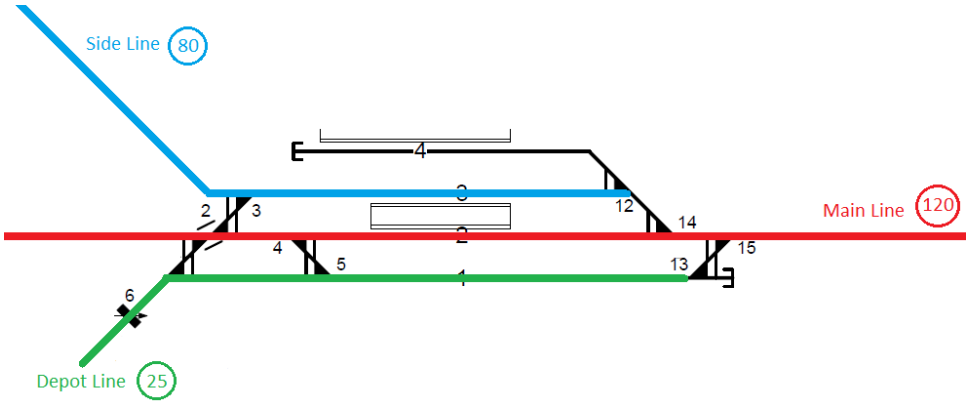

Figure 5.2 : Lines and their speed limits [1].

As it explained in the first chapter, trains cannot proceed with their maximum speed over a point which in reverse position. The speed limits of the points used in the layout are shown in the following Figure 5.3.

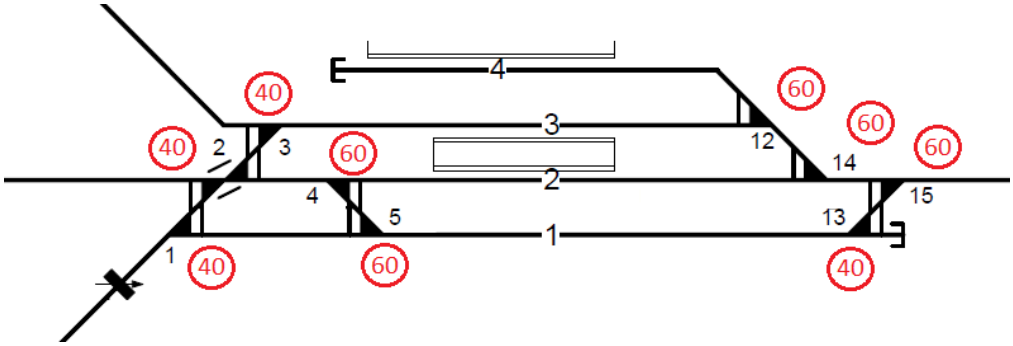

Figure 5.3 : Speed limits of points [1]

5.2 Signalling Design

Signalling equipment are positioned with respect to characteristics of the lines and trains and the operational demands. Therefore, track sections and signals are located with regard to definitions given above.

5.2.1 Signals

Signal's main purpose is to give moving authority to the train drivers. Two adjacent signals compose a signal block and only one railway vehicle can get a moving permission to enter a certain signal block. With regards to this idea, all entrance of the station are equipped a main signal to provide keep under control the station's traffic. In front of the station entry signals, distant signals are located to inform the trains arriving to the station about the main signal aspects in advance. Furthermore, a single main signal is placed at the entrance and exit of all platforms. It is important to ensure platform usage by only related vehicles. Platform exit signals can also be used to give to the trains a departure order. In other words, they can be managed with considering the time schedule.

It should also be made possible to control every shunting movements on the station. With this purpose, some shunting signals are positioned several location in the station. In some points, combined signals are assigned to provide to control of the shunting and normal movements in a single combined signal. See the created design in Figure 5.4.

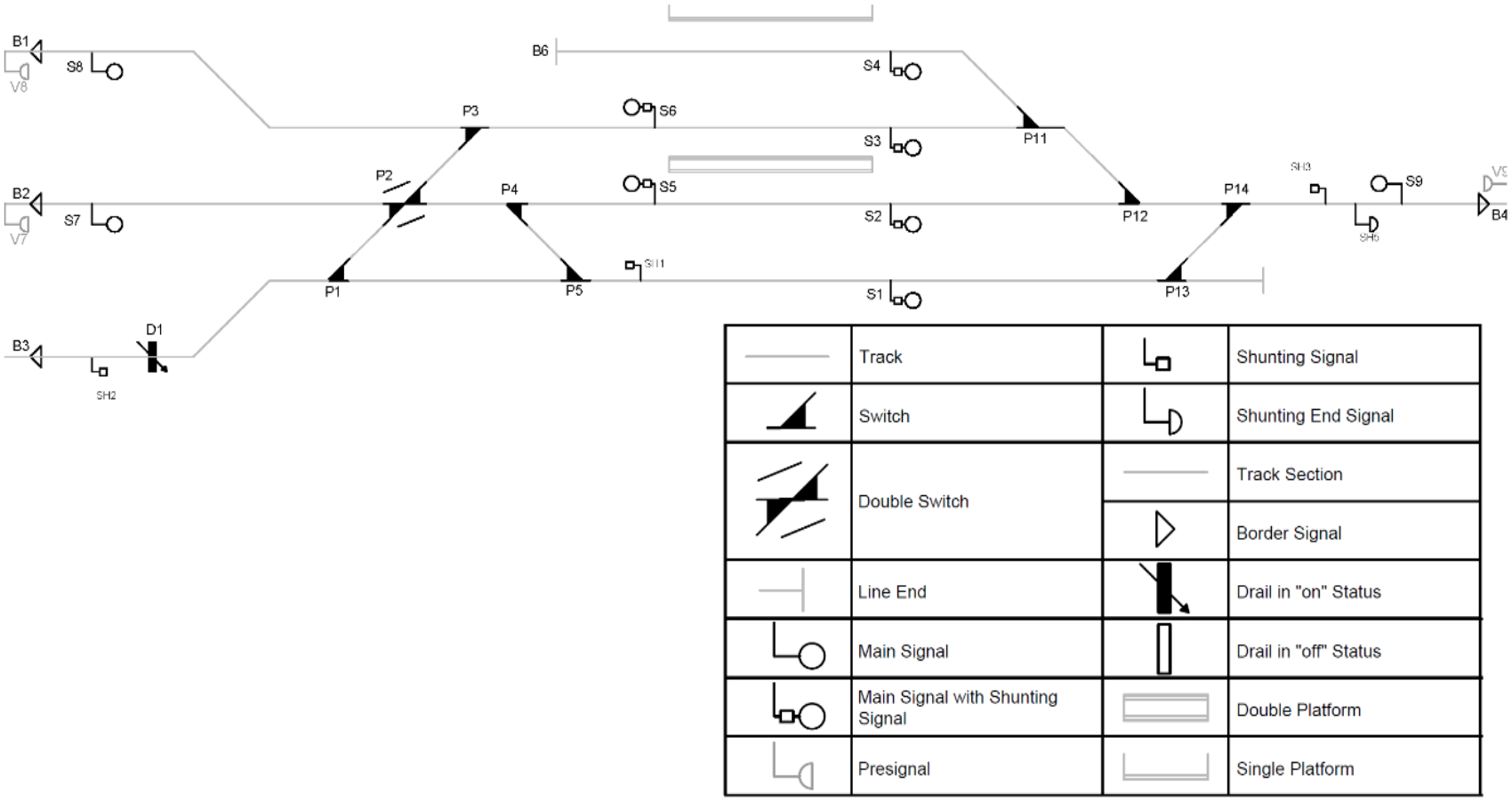

Figure 5.4 : Model station signal plan [1].

5.2.2 Track sections

All tracks are divided to the sections to detect the location of trains. The main criteria when the track sections are located is the safety and the optimal operational conditions. The length of the track section effects the train operations. In other words, if a track section is designed too long, a train occupies whole area and the other vehicles have to wait until the first train leaves the certain section. On the other hand, if track sections are designed too short, it causes more investment and maintenance costs. In addition to that, positioning of track detection areas are related to the signal's position. Hence, positioning of the signals and track sections are always related to each other.

For created model station layout, 21 different track sections were defined. For the purpose of safety and operational efficiency, every point area is separated the other areas by a track section. To provide the overlap protection, especial track sections are defined behind some platform exit signals. Created track section plan of the model station is given in Figure 6.5 and Figure 6.6. Model station specifications are also given as a table in Table 5.1.

Table 5.1 : Model station specifications [1].

Name	Quantity
Track Section	21
Point	9
Main Signal	9
Distant Signal	3
Shunting Signal	4
Derailer	1
Station Track	4
Platform	3

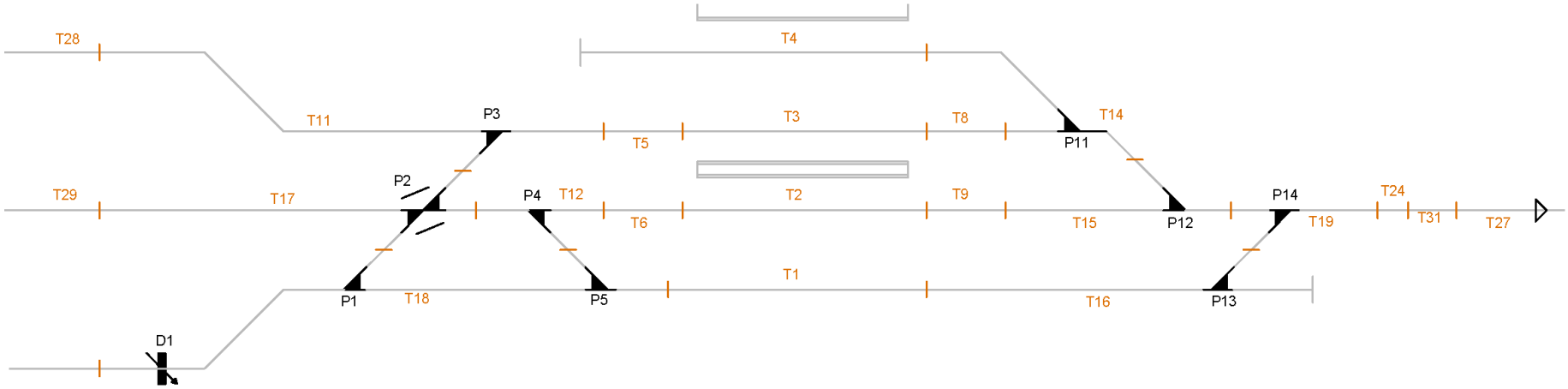

Figure 5.5 : Track section plan of the model station [1].

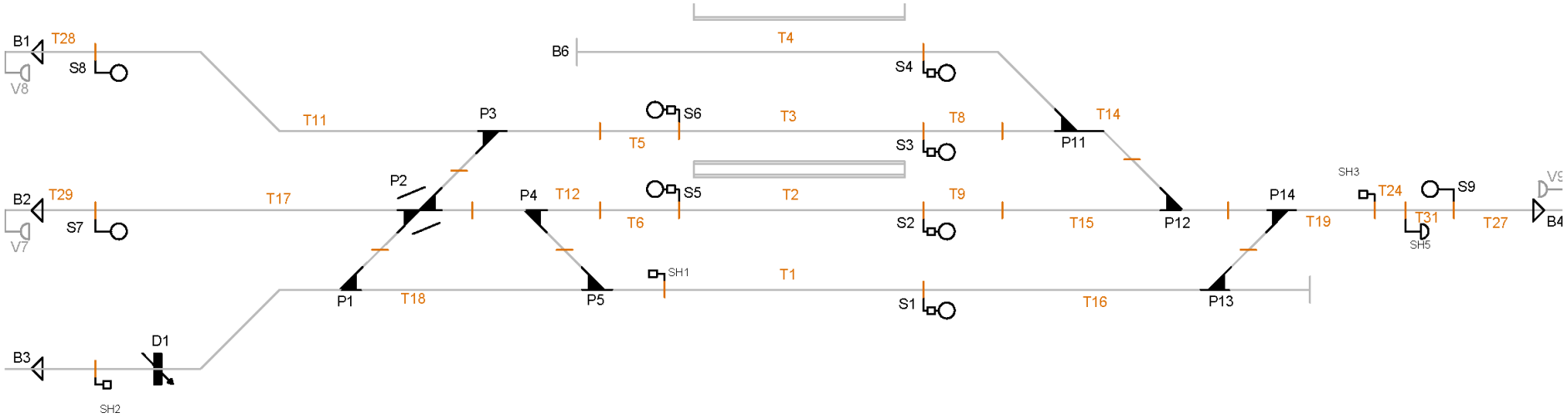

Figure 5.6 : Signal and track section plan [1].

6. EXAMPLE INTERLOCKING DESIGN

6.1 Introduction

This chapter contains developed interlocking functions based on finite state machines method and their implementations in SilworX PLC programming software. Detailed information about the design method and implementation software were given in the previous chapters.

In a complete interlocking system, there are a lot of functions developed to obtain various locking mechanism such as, object control functions, route functions, authorization functions, etc. However, in this thesis study, only object control models and required functions to set a main route are developed and implemented. Automata theory is used as a design method, because automata theory can be more easily designed than Petri Nets Method. It is also easy to implement an automata model with a simulation tool thanks to its simple and clear graphical interface.

Firstly, developed object control models are given. Afterwards, some route setting functions are explained. Finally, a complete route setting mechanism for a certain route in the model station are generated with using developed functions. All developed models are generic models and they can be implemented to any objects or routes in the layout.

6.2 Routes

First of all, possible main and shunting routes in the model station are determined and route table shown in Table 6.1 and Table 6.2 is generated.

6.3 Wayside Equipment Models

The control models for mostly used wayside elements are developed in this chapter. These models are the last controls between the interlocking system and the physical

equipment. All other functions defined in the interlocking software have to use these models to control wayside elements.

Table 6.1 : Route table of the model station [1]

No	Start S.	Target S.	Type	Path	Switches		Drail	Flank Protection					
								Switch		Signal	Track	Drail	Overlap
					Normal	Reverse	Out	Normal	Rev.			In	
1	S7	S2	M	T17, T12, T6, T2	P2a, P2b, P4	-	-	P1, P3, P5	-	-	-	-	T9
2	S7	S3	M	T17, T11, T5, T3	P2a	P2b, P3	-	P1, P5	-	S8, S5	T6, T12	-	T8
3	S7	S1	M	T17, T18, T13, T1	P2a, P2b	P4, P5	-	P3	-	S5, SH2	T6, T18	D1	T10
4	S8	S3	M	T11, T5, T3	P3	-	-	P2b	-	-	-	-	T8
5	S9	S5	M	T31, T24, T19, T15, T9, T2	P14, P12	-	-	P13	-	S3, S4	T7, T8, T14	-	T6
6	S9	S6	M	T31, T24, T19, T15, T14, T8, T3	P14, P11	P12	-	P13	-	S2, S4	T7, T9	-	T5
7	S9	B6	M	T31, T24, T19, T15, T14, T4	P14	P12, P11	-	P13	-	S2, S3	T8, T9	-	-
8	S5	B2	M	T6, T12, T17, T29	P4, P2a, P2b	-	-	P1, P3, P5	-	-	-	-	-
9	S6	B1	M	T5, T11, T28	P3	-	-	P2b	-	-	-	-	-
10	S6	B2	M	T5, T11, T17, T29	P2a	P3, P2b	-	P1, P5	-	S5, S8	T6, T12, T21, T25	-	-
11	S2	B4	M	T9, T15, T19, T24, T31, T27	P12, P14	-	-	P13	-	S3, S4	T7, T8, T14	-	-
12	S3	B4	M	T8, T14, T15, T19, T24, T31, T27	P11, P14	P12	-	P13	-	S2, S4	T7, T9	-	-
13	S4	B4	M	T14, T15, T19, T24, T31, T27	P14	P11, P12	-	P13	-	S2, S3	T8, T9	-	-
14	S1	B4	M	T16, T19, T24, T31, T27	-	P13, P14	-	-	-	S2, S3, S4, SH4	T7, T8, T9, T14, T15	-	-
15	SH2	S1	S	T18, T1	P1, P5	-	D1	P2a, P4	-	-	-	-	-
16	SH2	S2	S	T18, T17, T12, T6, T2	P4, P2b	P1, P2a	D1	P3	-	S7, SH1	-	-	-
17	SH2	S3	S	T18, T17, T11, T5, T3	-	P1, P2a, P2b, P3	D1	-	-	S5, S7, S8, SH1	-	-	-
18	SH3	B6	S	T19, T15, T14, T4	P14	P12, P11	-	P13	-	S2, S3	-	-	-
19	SH3	S6	S	T19, T15, T14, T8, T3	P14, P11	P12	-	P13	-	S2, S4	-	-	-
20	SH3	S5	S	T19, T15, T9, T2	P14, P12	-	-	P13	-	S3, S4	-	-	-
21	SH3	SH1	S	T19, T16, T1	-	P14, P13	-	-	-	S2, S3, S4, SH4	-	-	-
22	S6	B3	S	T5, T11, T17, T18	-	P3, P2b, P2a, P1	D1	-	-	S5, S7, S8, SH1	-	-	-
23	S5	B3	S	T6, T12, T17, T18	P4, P2b,	P2a, P1	D1	P3	-	S7, SH1	-	-	-
24	SH1	B3	S	T18	P5, P1	-	D1	P2a, P4	-	-	-	-	-
25	S2	SH5	S	T9, T15, T19, T24	P12, P14	-	-	P13	-	S3, S4	-	-	-
26	S3	SH5	S	T8, T14, T15, T19, T24	P14, P11	P12	-	P13	-	S2, S4	-	-	-
27	S4	SH5	S	S14, T15, T19, T24	P14	P12, P11	-	P13	-	S2, S3	-	-	-
28	S1	SH5	S	T16, T19, T24	-	P13, P14	-	-	-	S2, S3, S4, SH4	-	-	-

6.3.1 Point control model

Specifications of the point control model are listed below.

- Point has three possible positions: Normal, Reverse or intermediate. Normal and reverse positions are called end positions. The third position, intermediate, is a temporary position which represents the moving action of the point. Position detection of the point is provided by the sensors in the point mechanism.
- If any contradictory inputs provided position sensors is detected, point has to been disabled and kept in error state.
- There should be also a time limit to detect any possible problem when the point is moving. If the point could not complete its moving on time, it should be considered as there is a problem and the point has a failure.

Table 6.2 : Intersecting routes list [1]

Route No	Conflicting Routes
R1	R2, R3, R5, R8, R10, R16, R17, R20, R23, R24
R2	R1, R3, R4, R6, R8, R9, R10, R16, R17, R19, R23, R24
R3	R1, R2, R8, R10, R15, R16, R17, R21, R23, R24, R25
R4	R2, R6, R9, R10, R17, R19, R23
R5	R1, R6, R7, R11, R12, R13, R14, R16, R18, R19, R20, R21, R26, R27, R28, R29
R6	R2, R4, R5, R7, R11, R12, R13, R14, R17, R18, R19, R20, R21, R26, R27, R28, R29
R7	R5, R6, R11, R12, R13, R14, R18, R19, R20, R21, R26, R27, R28, R29
R8	R1, R2, R3, R10, R16, R17, R23, R24
R9	R2, R4, R10, R17, R23
R10	R2, R1, R3, R4, R8, R9, R16, R17, R23, R24
R11	R5, R6, R7, R12, R13, R14, R18, R19, R20, R21, R26, R27, R28, R29
R12	R5, R6, R7, R11, R13, R14, R18, R19, R20, R21, R26, R27, R28, R29
R13	R5, R6, R7, R11, R12, R14, R18, R19, R20, R21, R26, R27, R28, R29
R14	R5, R6, R7, R11, R12, R13, R18, R19, R20, R21, R26, R27, R28, R29, R30
R15	R3, R16, R17, R21, R23, R24, R25
R16	R3, R1, R2, R3, R5, R8, R10, R15, R17, R20, R23, R24, R25
R17	R1, R2, R3, R4, R6, R8, R9, R10, R15, R16, R19, R23, R24, R25
R18	R5, R6, R7, R11, R12, R13, R14, R19, R20, R21, R26, R27, R28, R29
R19	R2, R4, R5, R6, R7, R11, R12, R13, R14, R17, R18, R20, R21, R26, R27, R28, R29
R20	R1, R5, R6, R7, R11, R12, R13, R14, R16, R18, R19, R21, R26, R27, R28, R29
R21	R3, R5, R6, R7, R11, R12, R13, R14, R15, R18, R19, R20, R26, R27, R28, R29, R30
R22	R1, R2, R3, R4, R8, R9, R10, R15, R16, R17, R24, R25
R23	R1, R2, R3, R8, R10, R15, R16, R17, R23, R25
R24	R3, R15, R16, R17, R23, R24
R25	R5, R6, R7, R11, R12, R13, R14, R18, R19, R20, R21, R27, R28, R29
R26	R5, R6, R7, R11, R12, R13, R14, R18, R19, R20, R21, R26, R28, R29
R27	R5, R6, R7, R11, R12, R13, R14, R18, R19, R20, R21, R26, R27, R29
R28	R5, R6, R7, R11, R12, R13, R14, R18, R19, R20, R21, R26, R27, R28, R30

- Point can be controlled in two ways: First one is the remote control. It represents controls by interlocking system. Second way is the local control. Local control is not used in normal operation conditions. In some cases, interlocking system can give the point control authority to local staff for several purpose. Then, interlocking system takes back the point control authority. Otherwise, the point cannot be controlled by local staff without any permission. In real interlocking systems, there are some extra control models and procedures to manage the local control mechanism. However, they are not considered in this study to simplify the model.
- Point can be locked in any end position and it does not approve any command except unlocking command. It should be noted that this is not a physical locking. It is only a logical locking.
- Point cannot be moved if there is an occupancy over it.
- If any failure occurs on the point, it has to be kept in failure status and the control unit can be reset after the failure problem is solved.

The “trail” feature has not been considered because; it would be make the model more complicated. An extra control mechanism can be designed for it as a future work.

Inputs of the model have chosen as below.

- s_nor: Point is in normal position (sensor)
- s_rev: Point is in reverse position (sensor)
- occ: Occupancy (sensor)
- go_n: Go to normal position (command)
- go_r: Go to reverse position (command)
- lock: Lock (command)
- ulock: Unlock (command)
- rem: Remote control authority (command)
- lcl: Local control authority (command)
- reset: Reset the model (command)

States:

- S1: Start (initial state of the model)
- S2: Normal (point is in normal position)
- S3: Reverse (point is in reverse position)
- S4: Intermediate (point is moving)
- S5: Move Reverse (move point to reverse position command)
- S6: Move Normal (move point to normal position command)
- S7: Locked (point locked)
- S8: Error
- S9: Local (point is in local control)

Transition functions between the states:

- t1= s_nor ∧ !s_rev
- t2= !s_nor ∧ s_rev
- t3= !s_nor ∧ !s_rev
- t4= s_nor ∧ s_rev
- t5= go_r ∧ !occ
- t6= go_n ∧ !occ
- t7= !s_nor ∧ !s_rev ∧ (T#10s)
- t8= lok
- t9= ulok ∧ s_nor ∧ !s_rev

- t10= ulok ∧ !s_nor ∧ s_rev
- t11= (!s_nor ∧ !s_rev) ∨ (s_nor ∧ s_rev)
- t12= rst
- t13= (go_n ∨ go_r) ∧ (T#10s)
- t14= lcl
- t15= rem ∧ s_nor ∧ !s_rev
- t16= rem ∧ !s_nor ∧ s_rev
- t17= rem ∧ ((s_nor ∧ s_rev) ∨ (!s_nor ∧ !s_rev))

Finite state machine model has been designed as shown in Figure 6.1 with respect to given specifications.

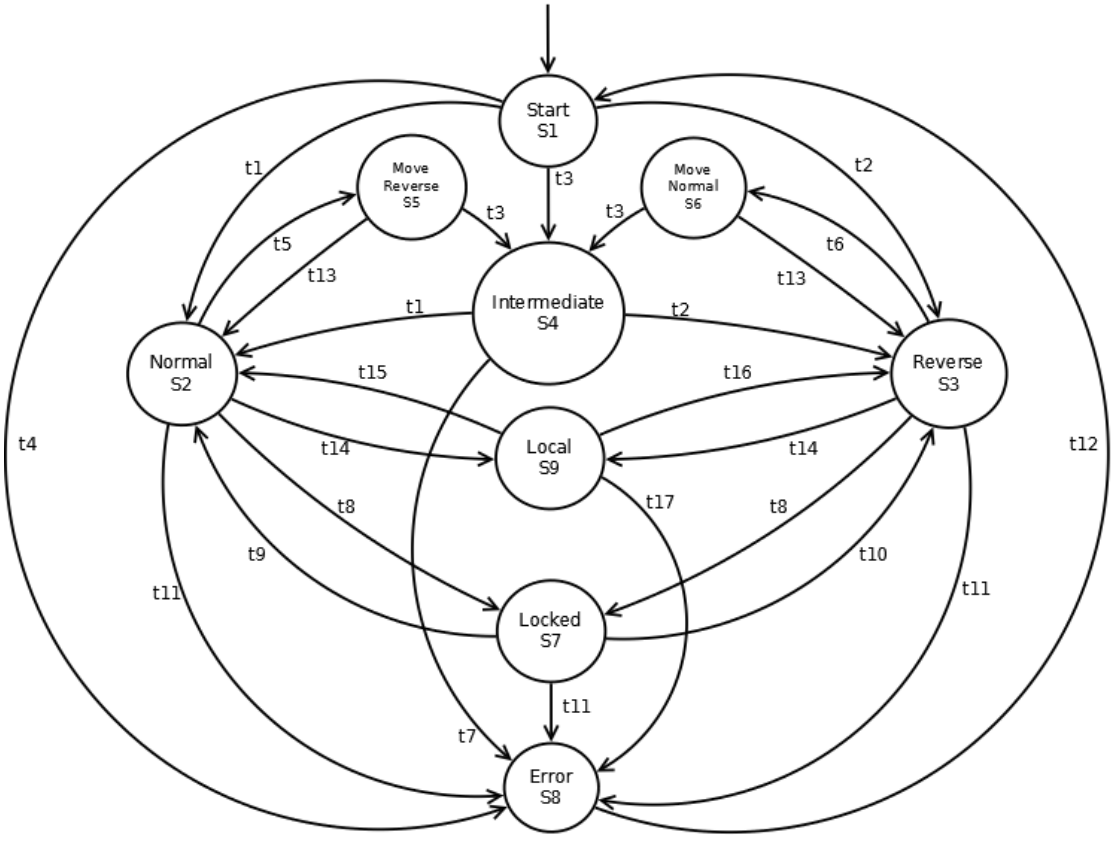

Figure 6.1 : Finite state model of the point [1].

Mathematical equations of the model:

$$S_1 = s_8 \cdot t_{12} + s_1 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_4$$

$$S_2 = s_1 \cdot t_1 + s_4 \cdot t_1 + s_5 \cdot t_{13} + s_9 \cdot t_{15} + s_7 \cdot t_9 + s_2 \cdot \bar{t}_5 \cdot \bar{t}_8 \cdot \bar{t}_{11} \cdot \bar{t}_{14} \quad \textbf{(6.1)}$$

$$S_3 = s_1 \cdot t_2 + s_4 \cdot t_2 + s_6 \cdot t_{13} + s_9 \cdot t_{16} + s_7 \cdot t_{10} + s_3 \cdot \bar{t}_6 \cdot \bar{t}_8 \cdot \bar{t}_{11} \cdot \bar{t}_{14}$$

$$S_4 = s_1 \cdot t_3 + s_5 \cdot t_3 + s_6 \cdot t_3 + s_4 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_7$$

$$S_5 = s_2 \cdot t_5 + s_5 \cdot \bar{t}_3 \cdot \bar{t}_{13}$$

$$S_6 = s_3 \cdot t_6 + s_6 \cdot \bar{t}_3 \cdot \bar{t}_{13}$$

$$S_7 = s_2 \cdot t_8 + s_3 \cdot t_8 + s_7 \cdot \bar{t}_9 \cdot \bar{t}_{10} \cdot \bar{t}_{11}$$

$$S_8 = s_1 \cdot t_4 + s_2 \cdot t_{11} + s_3 \cdot t_{11} + s_4 \cdot t_7 + s_7 \cdot t_{11} + s_9 \cdot t_{17} + s_8 \cdot \bar{t}_{12}$$

$$S_9 = s_2 \cdot t_{14} + s_3 \cdot t_{14} + s_9 \cdot \bar{t}_9 \cdot \bar{t}_{10} \cdot \bar{t}_{11}$$

Created function block of the model in SilworX is shown in Figure 6.2.

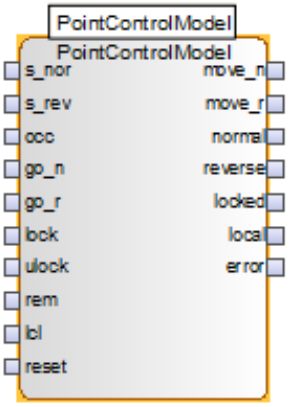

Figure 6.2 : Point control function block [1].

6.3.2 Signal control models

6.3.2.1 Main signal

Main signal specifications:

- Main signal has four different aspects: red, yellow, green and blinking green.
- Actual aspect information is provided by feedback sensors inside the signal device. These sensors are used to be ensure the signal does not have any bulb failure.
- Red aspect is the fail-safe aspect for main signal. In any failure occurs in the signal control, red aspect has to be shown.

- The signal can be blocked in red aspect.
- Any failure in the speed indicator should disable the main signal (if a speed indicator exist with the main signal).
- The signal cannot be cleared if one or more track section in the particular route has an occupancy.
- There should be a command to cancel cleared signal.
- If any occupancy occurs in the track sections, the signal has to show red aspect immediately.
- Signal control model should be able to decide correct aspect according to the status of the next main signal.

Signal model consists of three different sub-models to make it more understandable (see Figure 6.3).

- Signal Main Controller Model
- Signal Aspect Controller Model (decision maker mechanism)
- Signal Lamp Controller Model

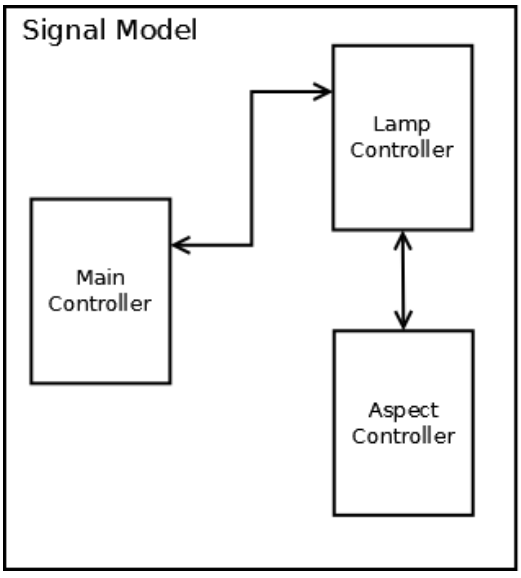

Figure 6.3 : Signal controller sub-units [1].

6.3.2.1.1 Signal main controller model

Inputs:

- occ: Occupancy information of the track sections behind the signal (sensor)
- set: Set signal (command)

- cancel: Cancel signal (command)
- blk: Block signal (command)
- ublk: Unblock signal (command)
- reset: Reset the model (command)
- lc_error: Light controller (sub-unit of signal controller) error info (internal input)
- sr_error: Speed restriction controller error info (internal input)
- lc_set: Light controller (sub-unit of signal controller) set info (internal input)
- sr_set: Speed restriction controller set info (internal input)

States:

- S1: Start (initial state)
- S2: Busy (signal is not ready to be set)
- S3: Free (signal is ready to be set)
- S4: Block (signal is blocked)
- S5: Set Request (signal set command)
- S6: Set (signal has been set)
- S7: Error

Transition functions between the states:

- t1= !occ ∧ !lc_error ∧ !sr_error
- t2= occ ∧ !lc_error ∧ !sr_error
- t3= lc_error ∨ sr_error
- t4= blk
- t5= ublk ∧ !occ
- t6= ublk ∧ occ
- t7= set ∧ !occ ∧ !lc_error ∧ !sr_error
- t8= lc_set ∧ sr_set
- t9= cancel ∨ (!lc_set ∧ S6 ∧ (T#1s)) ∨ (!sr_set ∧ S6 ∧ (T#1s))
- t10= occ
- t11= reset
- t12= lc_error ∨ sr_error ∨ (S5 ∧ (T#2s))

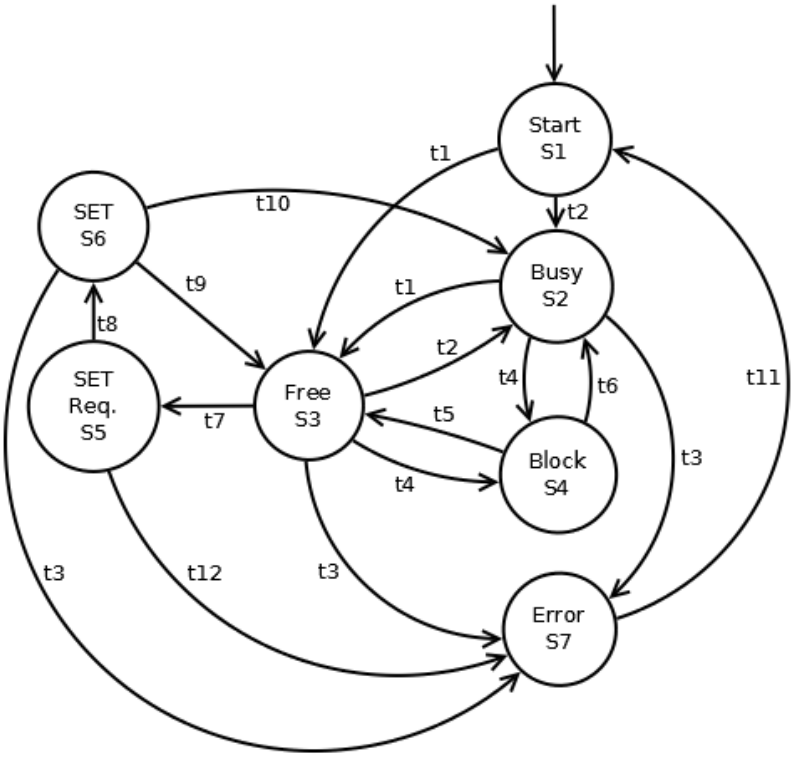

Figure 6.4 : Signal main controller model [1]

Mathematical equations of the model:

$$S_1 = s_7 \cdot t_{11} + s_1 \cdot \bar{t}_1 \cdot \bar{t}_2$$

$$S_2 = s_1 \cdot t_2 + s_6 \cdot t_{10} + s_3 \cdot t_2 + s_4 \cdot t_6 + s_2 \cdot \bar{t}_1 \cdot \bar{t}_3 \cdot \bar{t}_4$$

$$S_3 = s_6 \cdot t_9 + s_1 \cdot t_1 + s_2 \cdot t_1 + s_4 \cdot t_5 + s_3 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_4 \cdot \bar{t}_7$$

$$S_4 = s_2 \cdot t_4 + s_3 \cdot t_4 + s_4 \cdot \bar{t}_5 \cdot \bar{t}_6 \qquad \textbf{(6.2)}$$

$$S_5 = s_3 \cdot t_7 + s_5 \cdot \bar{t}_8 \cdot \bar{t}_{12}$$

$$S_6 = s_5 \cdot t_8 + s_6 \cdot \bar{t}_3 \cdot \bar{t}_9 \cdot \bar{t}_{10}$$

$$S_7 = s_2 \cdot t_3 + s_3 \cdot t_3 + s_5 \cdot t_{12} + s_6 \cdot t_3 + s_7 \cdot \bar{t}_{11}$$

Function block created in SilworX can be seen in Figure 6.5.

6.3.2.1.2 Signal aspect controller model

Aspect controller is a decision maker mechanism and it evaluates the proper aspect with regard to next main signal's aspect.

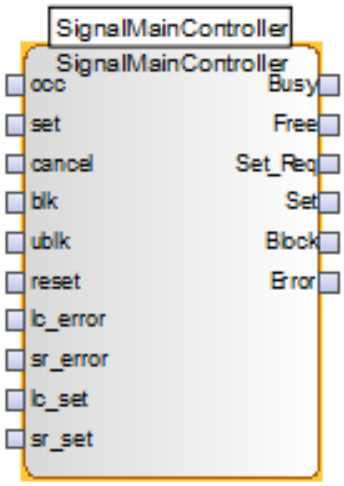

Figure 6.5 : Signal main controller function block [1].

Inputs:

- occ: Occupancy information of the track sections behind the signal (sensor)
- nssi: Next signal set information (internal)
- nsrc: Next signal restricted speed information (internal)
- eva: Evaluate the aspect command (command)

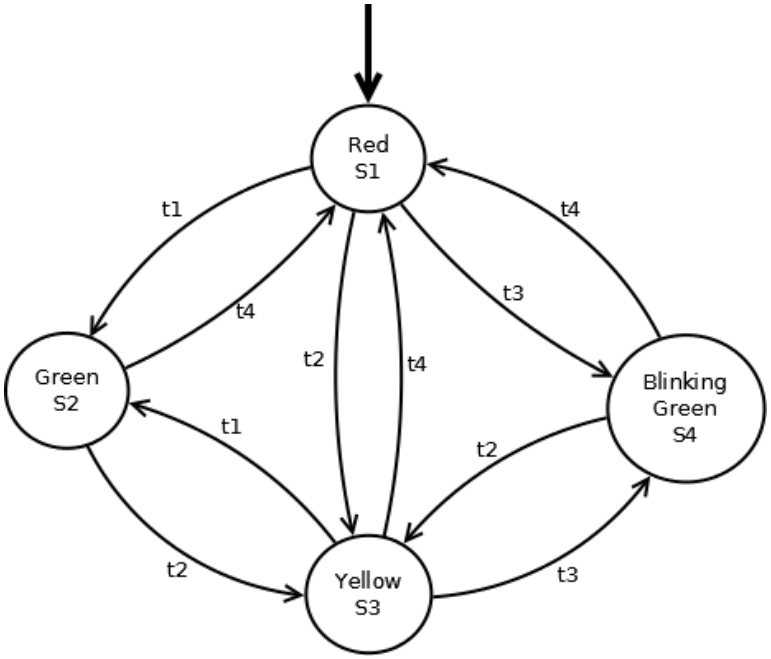

Figure 6.6 : Signal aspect controller model [1].

States:

- S1: Red (initial state)

- S2: Green
- S3: Yellow
- S4: Blinking Green

Transition functions:

- t1= eva ∧ !occ ∧ Nssi ∧ !Nsrc
- t2= eva ∧ !occ ∧ !Nssi
- t3= eva ∧ !occ ∧ Nssi ∧ Nsrc
- t4= !eva ∨ occ

Logical equations:

$$S_1 = s_2 \cdot t_4 + s_3 \cdot t_4 + s_4 \cdot t_4 + s_1 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_3$$
$$S_2 = s_1 \cdot t_1 + s_3 \cdot t_1 + s_2 \cdot \bar{t}_2 \cdot \bar{t}_4$$
$$S_3 = s_1 \cdot t_2 + s_2 \cdot t_2 + s_4 \cdot t_2 + s_3 \cdot \bar{t}_1 \cdot \bar{t}_4 \cdot \bar{t}_3 \quad \textbf{(6.3)}$$
$$S_4 = s_1 \cdot t_3 + s_3 \cdot t_3 + s_4 \cdot \bar{t}_2 \cdot \bar{t}_4$$

Created function block is given in Figure 6.7.

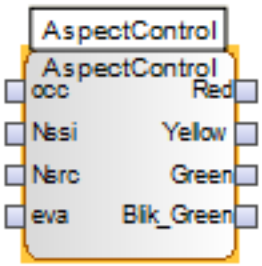

Figure 6.7 : Aspect controller function block [1].

6.3.2.1.3 Signal lamp controller model

Lamp controller is responsible to control signal lamps and detections of the failures in the bulbs.

Inputs:

- r_sens: Red light feedback sensor (sensor)
- y_sens: Yellow light feedback sensor (sensor)
- g_sens: Green light feedback sensor (sensor)
- occ: Occupancy information of the track sections behind the signal (sensor)

- set: Command from signal main controller (command)
- y_req: Yellow request from aspect controller (internal input)
- g_req: Green request from aspect controller (internal input)
- bg_req: Blinking green request from aspect controller (internal input)
- reset: Reset the model (command)

States:

- S1: Start (initial state)
- S2: Red
- S3: Yellow
- S4: Green
- S5: Blinking Green
- S6: Red Command (show red signal command to the signal device)
- S7: Yellow Command (show yellow signal command to the signal device)
- S8: Green Command (show green signal command to the signal device)
- S9: Error (Error state is red aspect)

Created FSM model is shown in Figure 6.8.

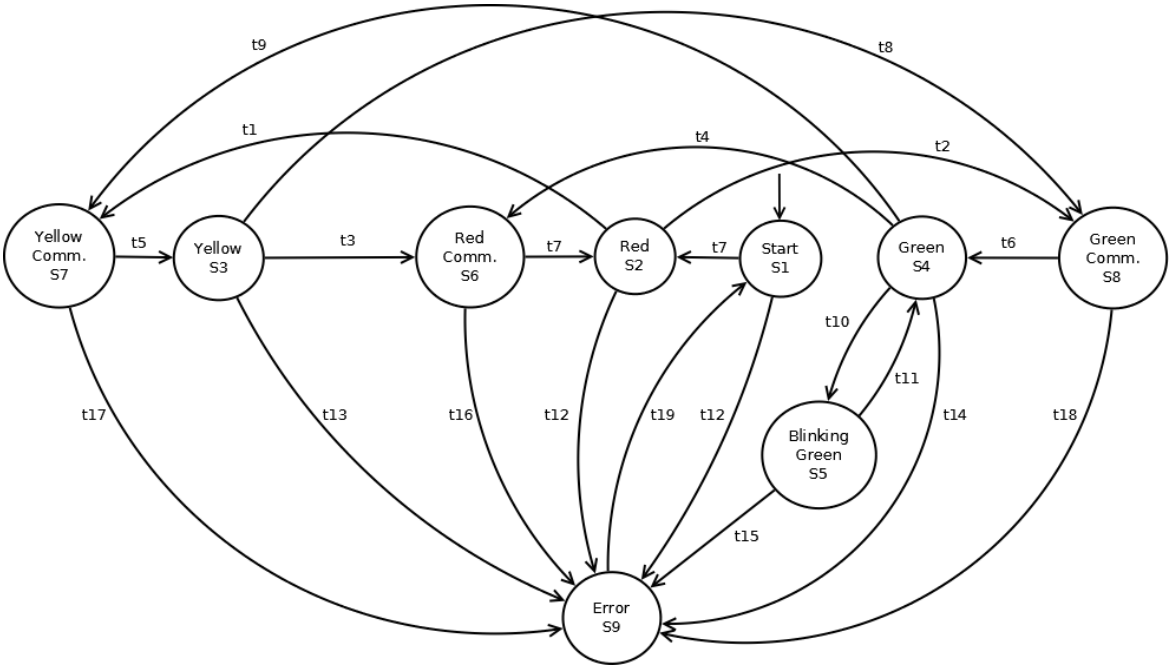

Figure 6.8 : Signal lamp controller model [1].

Transition functions:

- t1= set ∧ y_req ∧ !occ

- t2= set ∧ (g_req ∨ bg_req) ∧ !occ
- t3= occ ∨ !set ∨ (!y_req ∧ (T#100ms))
- t4= occ ∨ !set ∨ (!g_req ∧ (T#100ms))
- t5= y_sens
- t6= g_sens
- t7= r_sens
- t8= set ∧ !y_req ∧ (g_req ∨ bg_req) ∧ !occ
- t9= set ∧ !g_req ∧ y_req ∧ !occ
- t10= set ∧ bg_req ∧ !occ
- t11= !bg_req ∨ !set ∨ occ
- t12= (!r_sens ∧ T#1s) ∨ g_sens ∨ y_sens
- t13= (!y_sens ∧ T#1s) ∨ r_sens ∨ g_sens ∨ (y_req ∧ ((g_req ∨ bg_req) ∧ T#100ms))
- t14= (!g_sens ∧ T#1s) ∨ r_sens ∨ y_sens ∨ (g_req ∧ (y_req ∧ T#100ms))
- t15= (!g_sens ∧ S5 ∧ (T#2s)) ∨ (g_sens ∧ S5 ∧ (T#2s))
- t16= T#1s
- t17= T#1s
- t18= T#1s
- t19= reset

Logical equations:

$$
\begin{aligned}
S_1 &= s_9 \cdot t_{19} + s_1 \cdot \bar{t}_7 \cdot \bar{t}_{12} \\
S_2 &= s_1 \cdot t_7 + s_6 \cdot t_7 + s_2 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_{12} \\
S_3 &= s_7 \cdot t_5 + s_3 \cdot \bar{t}_3 \cdot \bar{t}_8 \cdot \bar{t}_{13} \\
S_4 &= s_5 \cdot t_{11} + s_8 \cdot t_6 + s_4 \cdot \bar{t}_4 \cdot \bar{t}_9 \cdot \bar{t}_{10} \cdot \bar{t}_{14} \\
S_5 &= s_4 \cdot t_{10} + s_5 \cdot \bar{t}_{11} \cdot \bar{t}_{15} \\
S_6 &= s_3 \cdot t_3 + s_4 \cdot t_4 + s_6 \cdot \bar{t}_7 \cdot \bar{t}_{16} \\
S_7 &= s_2 \cdot t_1 + s_4 \cdot t_9 + s_7 \cdot \bar{t}_5 \cdot \bar{t}_{17} \\
S_8 &= s_2 \cdot t_2 + s_3 \cdot t_8 + s_8 \cdot \bar{t}_6 \cdot \bar{t}_{18}
\end{aligned}
\qquad \textbf{(6.4)}
$$

$$S_9 = s_7 \cdot t_{17} + s_3 \cdot t_{13} + s_6 \cdot t_{16} + s_2 \cdot t_{12} + s_1 \cdot t_{12} + s_5 \cdot t_{15} + s_4 \cdot t_{14} + s_8 \cdot t_{18} + s_9 \cdot \bar{t}_{19}$$

Function block can be seen in following Figure 6.9.

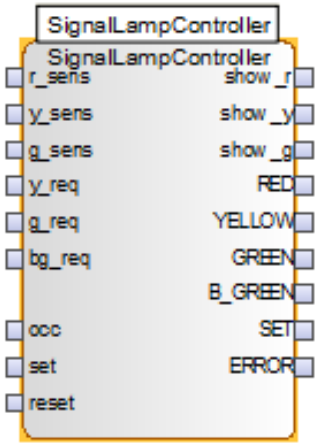

Figure 6.9 : Lamp controller function block [1].

6.3.3 Distant signal

Distant signal specifications:

- Distant signal has three aspects: yellow, green and blinking green.
- Distant signal works always depend on its main signal.
- Actual aspect information is provided by feedback sensors inside the signal device. These sensors are used to be ensure that the signal does not have any bulb failure.
- Yellow aspect is the fail-safe aspect for distant signal. In any failure occurs in the signal control, yellow aspect has to be shown immediately.

Inputs:

- y_sens: Yellow light feedback sensor (sensor)
- g_sens: Green light feedback sensor (sensor)
- set: Main signal set information (internal input)
- src: Main signal speed restriction information (internal input)
- reset : Reset the model (command)

States:

- S1: Yellow (initial state)
- S2: Green
- S3: Blinking Green
- S4: Yellow Command (show yellow output to the signal device)
- S5: Green Command (show green output to the signal device)
- S6: Blinking Green Command (show blinking green output to the signal device)
- S7: Error (error state is yellow aspect)

Figure 6.10 shows created FSM diagram.

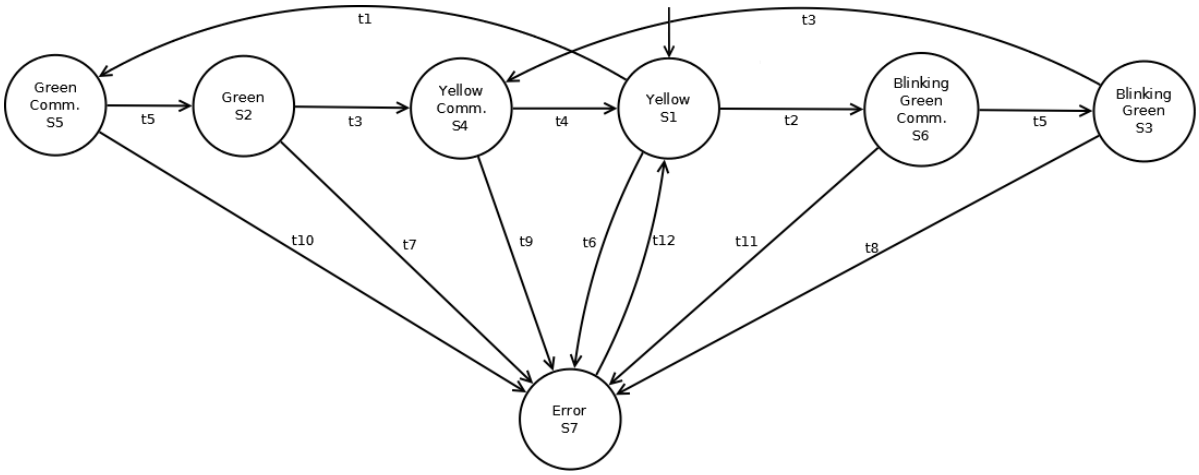

Figure 6.10 : Distant signal control model [1].

Transition functions:

- t1= set ∧ !src
- t2= set ∧ src
- t3= !set
- t4= y_sens
- t5= g_sens
- t6= (!y_sens ∧ S1 ∧ T#1s) ∨ g_sens
- t7= (!g_sens ∧ S2 ∧ T#1s) ∨ y_sens
- t8= (!g_sens ∧ S3 ∧ T#2s) ∨ (g_sens ∧ S3 ∧ T#2s) ∨ y_sens
- t9= S4 ∧ T#1s
- t10= S5 ∧ T#1s
- t11= S6 ∧ T#1s

- t12= reset

Mathematical equations:

$$
\begin{aligned}
S_1 &= s_4 \cdot t_4 + s_7 \cdot t_{12} + s_1 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_6 \\
S_2 &= s_5 \cdot t_5 + s_2 \cdot \bar{t}_3 \cdot \bar{t}_7 \\
S_3 &= s_6 \cdot t_5 + s_3 \cdot \bar{t}_3 \cdot \bar{t}_8 \\
S_4 &= s_2 \cdot t_3 + s_3 \cdot t_3 + s_4 \cdot \bar{t}_4 \cdot \bar{t}_9 \\
S_5 &= s_1 \cdot t_1 + s_5 \cdot \bar{t}_5 \cdot \bar{t}_{10} \\
S_6 &= s_1 \cdot t_2 + s_6 \cdot \bar{t}_5 \cdot \bar{t}_{11} \\
S_7 &= s_5 \cdot t_{10} + s_2 \cdot t_7 + s_4 \cdot t_9 + s_1 \cdot t_6 + s_6 \cdot t_{11} + s_3 \cdot t_8 + s_7 \cdot \bar{t}_{12}
\end{aligned}
\qquad \textbf{(6.5)}
$$

Created function block is given in following Figure 6.11.

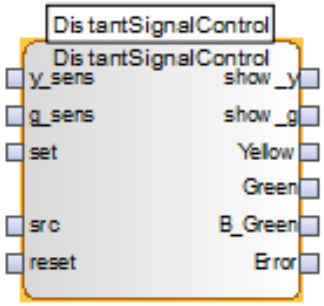

Figure 6.11 : Distant signal control function block [1].

6.3.4 Speed indicator

Speed indicator specifications:

- Several speed aspects can be used in the Speed indicators. However, speed indicators in the model station have only three aspect: “6” and “4” and the dark aspect. “6” means speed limit is 60 km/h and “4” means speed limit is 40 km/h. If there isn’t any speed restriction, speed indicator does not show any number and it means driver can proceed with max allowed line speed if the main signal is cleared.
- Speed indicator is always used with a main or distant signal and it is activated when its main or distant signal is set.

- Actual aspect information is provided by feedback sensors inside the speed indicator device. These sensors are used to be ensure the speed indicator does not have any indicator failure.
- “Dark aspect” is the fail-safe aspect of speed indicator.

Inputs:

- sens6: 60 km/h indicator sensor feedback (sensor)
- sens4: 40 km/h indicator sensor feedback (sensor)
- scr6: Speed restriction 60 km/h (internal input)
- scr4: Speed restriction 40 km/h (internal input)
- set: Set information of its main or distant signal (command)
- reset: Reset the model (command)

States:

- S1: Start (initial state)
- S2: Dark
- S3: 60 km/h light command (show “6” output to the indicator device)
- S4: 40 km/h light command (show “4” output to the indicator device)
- S5: 60 km/h
- S6: 40 km/h
- S7: Error

Created FSM diagram is given in Figure 6.12.

Transitions:

- t2= set ∧ scr6 ∧ !scr4
- t3= set ∧ !scr6 ∧ scr4
- t4= set ∧ !scr6 ∧ !scr4
- t5= sens6
- t6= sens4
- t7= !set
- t8= S3 ∧ T#1s
- t9= S4 ∧ T#1s
- t10= (!sens6 ∨ sens4 ∨ !scr6 ∨ scr4) ∧ S5 ∧ T#1s

- t11= (sens6 ∨ !sens4 ∨ scr6 ∨ !scr4) ∧ S6 ∧ T#1s
- t12= (sens6 ∨ sens4 ∨ scr6 ∨ scr4) ∧ S2 ∧ T#1s
- t13= set ∧ scr6 ∧ scr4
- t14= reset

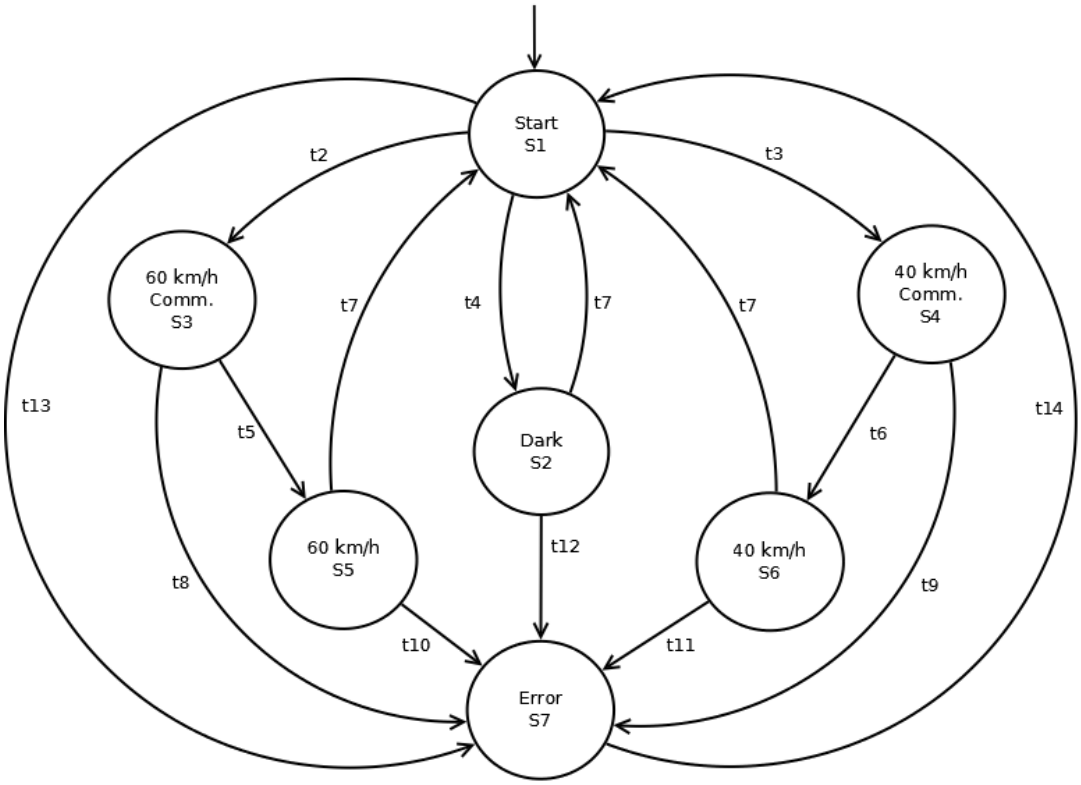

Figure 6.12 : Speed indicator control model [1].

Equations of the model:

$$
\begin{aligned}
S_1 &= s_5 \cdot t_7 + s_2 \cdot t_7 + s_7 \cdot t_{14} + s_6 \cdot t_7 + s_1 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_4 \cdot \bar{t}_{13} \\
S_2 &= s_1 \cdot t_4 + s_2 \cdot \bar{t}_7 \cdot \bar{t}_{12} \\
S_3 &= s_1 \cdot t_2 + s_3 \cdot \bar{t}_5 \cdot \bar{t}_8 \\
S_4 &= s_1 \cdot t_3 + s_4 \cdot \bar{t}_6 \cdot \bar{t}_9 \\
S_5 &= s_3 \cdot t_5 + s_5 \cdot \bar{t}_7 \cdot \bar{t}_{10} \\
S_6 &= s_4 \cdot t_6 + s_2 \cdot \bar{t}_7 \cdot \bar{t}_{11} \\
S_7 &= s_1 \cdot t_{13} + s_3 \cdot t_8 + s_5 \cdot t_{10} + s_2 \cdot t_{12} + s_6 \cdot t_{11} + s_4 \cdot t_9 + s_7 \cdot \bar{t}_{14}
\end{aligned}
\qquad \textbf{(6.6)}
$$

In following Figure 6.13, created function block can be seen.

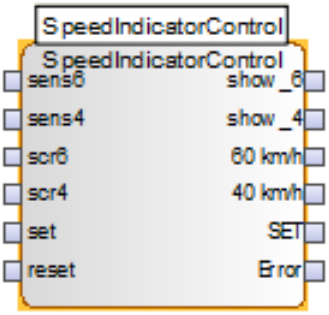

Figure 6.13 : Speed indicator function block [1].

6.3.5 Track clear detector model

Track clear detector model is a general model which is independent to detection technology. Track circuit, axle counter or any other technique can be used with this model. Track clear detector model only needs an occupancy info provided by a track clear detection device.

Inputs:

- occ: Occupancy information from a track clear detection device (sensor)
- set: Set command from interlocking (command)
- cancel: Cancel command from interlocking (command)

States:

- S1: Free
- S2: Occupied (initial state)
- S3: SET

Designed model is given in Figure 6.14.

Transitions:

- t1= occ
- t2= !occ
- t3= set
- t4= cancel

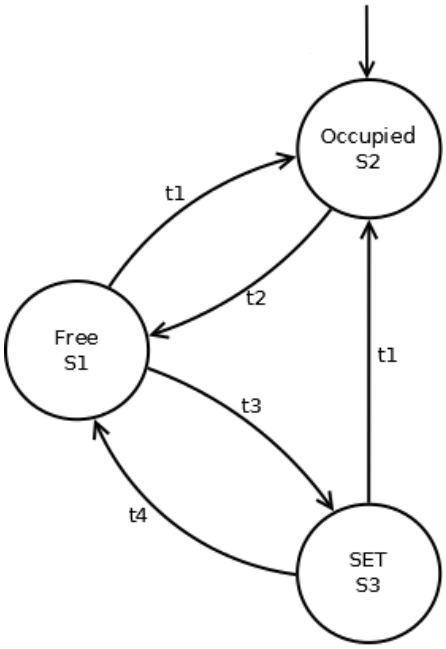

Figure 6.14 : Track clear detector model [1].

Equations:

$$S_1 = s_2 \cdot t_2 + s_3 \cdot t_4 + s_1 \cdot \bar{t}_1 \cdot \bar{t}_3$$

$$S_2 = s_1 \cdot t_1 + s_3 \cdot t_1 + s_2 \cdot \bar{t}_2 \qquad \textbf{(6.7)}$$

$$S_3 = s_1 \cdot t_3 + s_3 \cdot \bar{t}_1 \cdot \bar{t}_4$$

Track clear detector function block created in SilworX is shown in Figure 6.15.

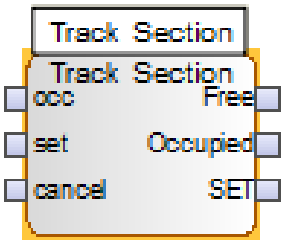

Figure 6.15 : Track clear detector function block [1].

6.3.6 Derailer control model

Derailer characteristics are very similar to point characteristics. They have been listed below:

- Derailer has three possible positions: Active, Passive or intermediate. Active means the derailer is protection mode and it derails the vehicle passing over it.

Passive position represents derailing device is outside the rail and it prevents the vehicle passing. The third position, intermediate, is a temporary position which represents the moving action of the point. Position detection of the point is provided by the sensors in the point mechanism.

- If any contradictory inputs provided position sensors is detected, derailer has to been disabled and kept in error state.
- There should be also a time limit to detect any possible problem when the derailer is moving. If the derailer could not complete its moving on time, it should be considered as there is a problem and the point has a failure.
- Derailer can be controlled in two ways: First one is the remote control. It represents controls by interlocking system. Second way is the local control. Local control is not used in normal operation conditions. In some cases, interlocking system can give the derailer control authority to local staff for several purpose. Then, interlocking system takes back the derailer control authority. Otherwise, the derailer cannot be controlled by local staff without any permission. In real interlocking systems, there are some extra control models and procedures to manage the local control mechanism. However, they are not considered in this study to simplify the model.
- Derailer can be locked in any end position and it does not approve any command except unlocking command. It should be noted that this is not a physical locking. It is only a logical locking.
- Derailer cannot be moved if there is an occupancy over it.
- If any failure occurs in the derailer device, it has to be kept in failure status and the control unit can be reset after the failure problem is solved.

Inputs:

- sens_act: Derailer is in active position (sensor)
- sens_pass: Derailer is in passive position (sensor)
- occ: Occupancy (sensor)
- activate: Go to normal position (command)
- deactivate: Go to reverse position (command)
- lock: Lock (command)
- unlock: Unlock (command)

- remote: Remote control authority (command)
- local: Local control authority (command)
- reset: Reset (command)

States:

- S1: Start (initial state)
- S2: Active (derailer is active)
- S3: Passive (derailer is passive)
- S4: Intermediate (derailer is moving)
- S5: Deactivate Derailer (move derailer to passive position command)
- S6: Activate Derailer (move derailer to active position command)
- S7: Locked (derailer is locked)
- S8: Error
- S9: Local (derailer is in local control)

Generated FSM model (Figure 6.16):

Transitions:

- t1= sens_act ∧ !sens_pass
- t2= !sens_act ∧ !sens_pass
- t3= !sens_act ∧ !sens_pass
- t4= sens_act ∧ sens_pass
- t5= deactivate ∧ !occ
- t6= activate ∧ !occ
- t7= !sens_act ∧ !sens_pass ∧ (T#10s)
- t8= lock
- t9= unlock ∧ sens_act ∧ !sens_pass
- t10= unlock ∧ !sens_act ∧ sens_pass
- t11= (!sens_act ∧ !sens_pass) ∨ (sens_act ∧ sens_pass)
- t12= reset
- t13= (activate ∨ deactivate) ∧ (T#10s)
- t14= local
- t15= remote ∧ sens_act ∧ !sens_pass
- t16= remote ∧ !sens_act ∧ sens_pass

- t17= remote ∧ ((sens_act ∧ sens_pass) ∨ (!sens_act ∧ !sens_pass))

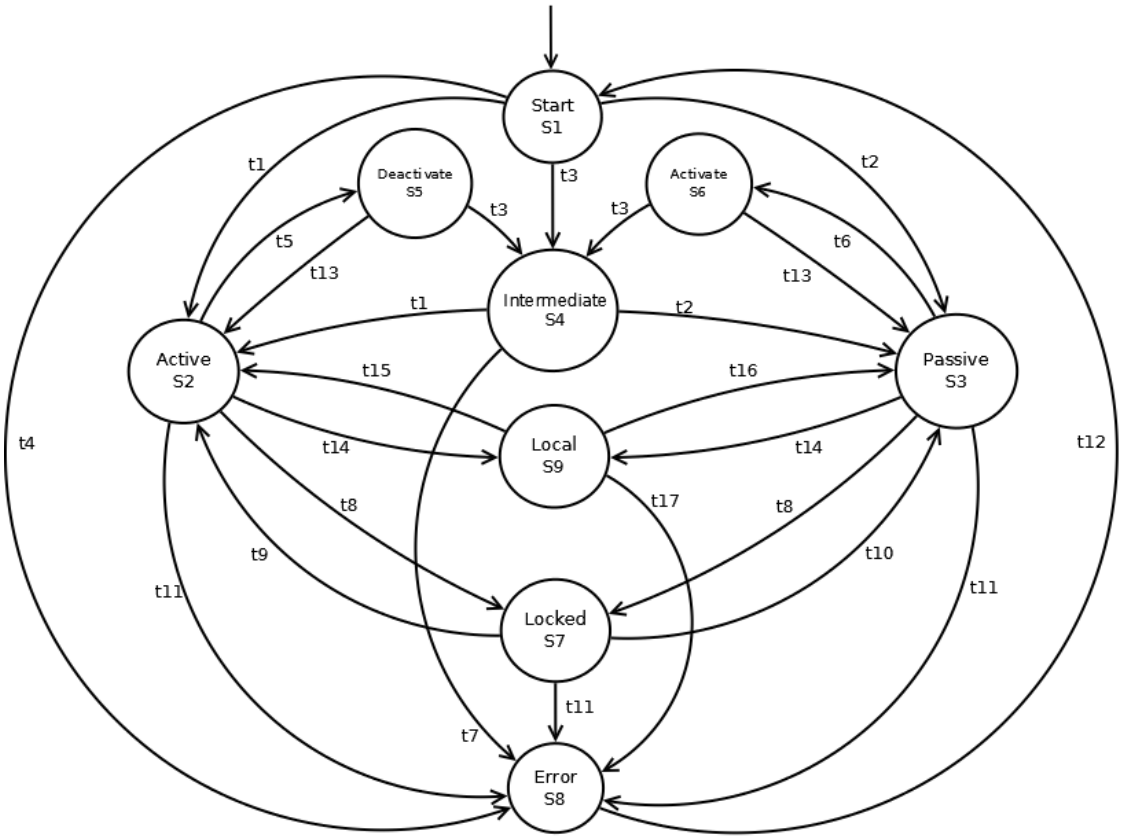

Figure 6.16 : Derailer control model [1].

Equations:

$$S_1 = s_8 \cdot t_{12} + s_1 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_4$$

$$S_2 = s_1 \cdot t_1 + s_4 \cdot t_1 + s_5 \cdot t_{13} + s_9 \cdot t_{15} + s_7 \cdot t_9 + s_2 \cdot \bar{t}_5 \cdot \bar{t}_8 \cdot \bar{t}_{11} \cdot \bar{t}_{14}$$

$$S_3 = s_1 \cdot t_2 + s_4 \cdot t_2 + s_6 \cdot t_{13} + s_9 \cdot t_{16} + s_7 \cdot t_{10} + s_3 \cdot \bar{t}_6 \cdot \bar{t}_8 \cdot \bar{t}_{11} \cdot \bar{t}_{14} \quad \textbf{(6.8)}$$

$$S_4 = s_1 \cdot t_3 + s_5 \cdot t_3 + s_6 \cdot t_3 + s_4 \cdot \bar{t}_1 \cdot \bar{t}_2 \cdot \bar{t}_7$$

$$S_5 = s_2 \cdot t_5 + s_5 \cdot \bar{t}_3 \cdot \bar{t}_{13}$$

$$S_6 = s_3 \cdot t_6 + s_6 \cdot \bar{t}_3 \cdot \bar{t}_{13}$$

$$S_7 = s_2 \cdot t_8 + s_3 \cdot t_8 + s_7 \cdot \bar{t}_9 \cdot \bar{t}_{10} \cdot \bar{t}_{11}$$

$$S_8 = s_1 \cdot t_4 + s_2 \cdot t_{11} + s_3 \cdot t_{11} + s_4 \cdot t_7 + s_7 \cdot t_{11} + s_9 \cdot t_{17} + s_8 \cdot \bar{t}_{12}$$

$$S_9 = s_2 \cdot t_{14} + s_3 \cdot t_{14} + s_9 \cdot \bar{t}_9 \cdot \bar{t}_{10} \cdot \bar{t}_{11}$$

Function block (Figure 6.17):

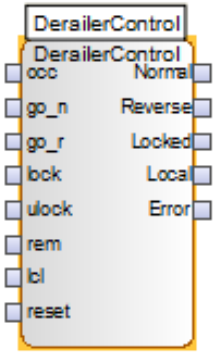

Figure 6.17 : Derailer control model [1].

6.4 Route Setting Model

The functions used for the route mechanism are modelled in this chapter. Four models are generated to obtain a route setting procedure with respect to the route setting rules described in previous chapters. One of them is the main controller and the others are sub-units of the main controller. Figure 6.19 shows the interactions between the models.

- Route Main Controller
- Route Point Controller
- Route Signal Controller
- Route Track Section Controller
- Route Derailer Controller

6.4.1 Route point controller

Route point controller are used to control all points in a certain route. It provides availability and proper position information of all points in a certain route to the route main controller. It can set and lock all points with a single command sent from the route main controller.

Inputs:

- P(1)pos: The proper position info of the first point on the route (sensor)
- P(2)pos: The proper position info of the second point on the route (sensor)
- P(3)pos: The proper position info of the third point on the route (sensor)
- P(4)pos: The proper position info of the fourth point on the route (sensor)
- P(5)pos: The proper position info of the fifth point on the route (sensor)
- P(1)lock: Locking info of the first point on the route (internal input)
- P(2)lock: Locking info of the second point on the route (internal input)
- P(3)lock: Locking info of the third point on the route (internal input)
- P(4)lock: Locking info of the fourth point on the route (internal input)
- P(5)lock: Locking info of the fifth point on the route (internal input)
- available: All points are available for the position changing (internal input)
- set_all: Set all points to the correct position according to the route (command)

States:

- S1: Incorrect (initial state) (at least one point on the route is not in correct position)
- S2: Correct (all points on the route are in correct position)
- S3: Locked (all points on the route are locked)
- S4: Moving Request (move points to the correct position)
- S5: Locking Request (lock all points in proper position)
- S6: Not Available (at least one point on the route is not available)

FSM model can be seen in Figure 6.18.

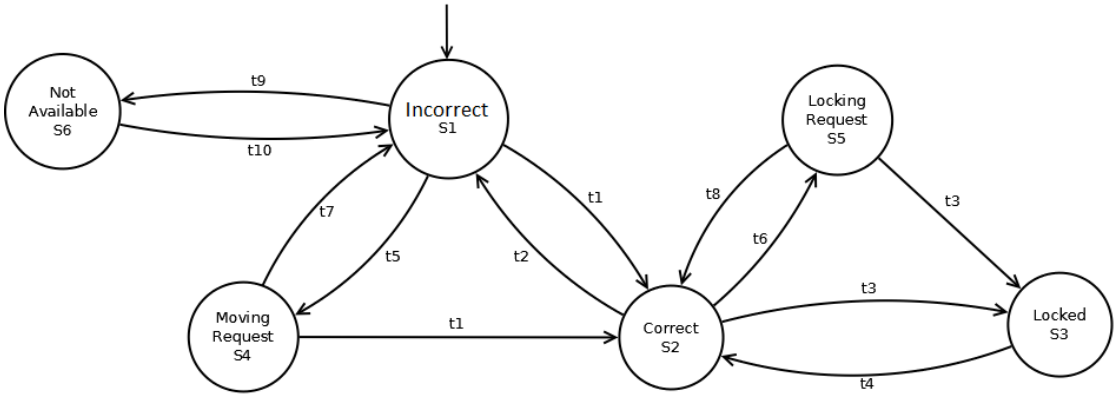

Figure 6.18 : Route points control model [1].

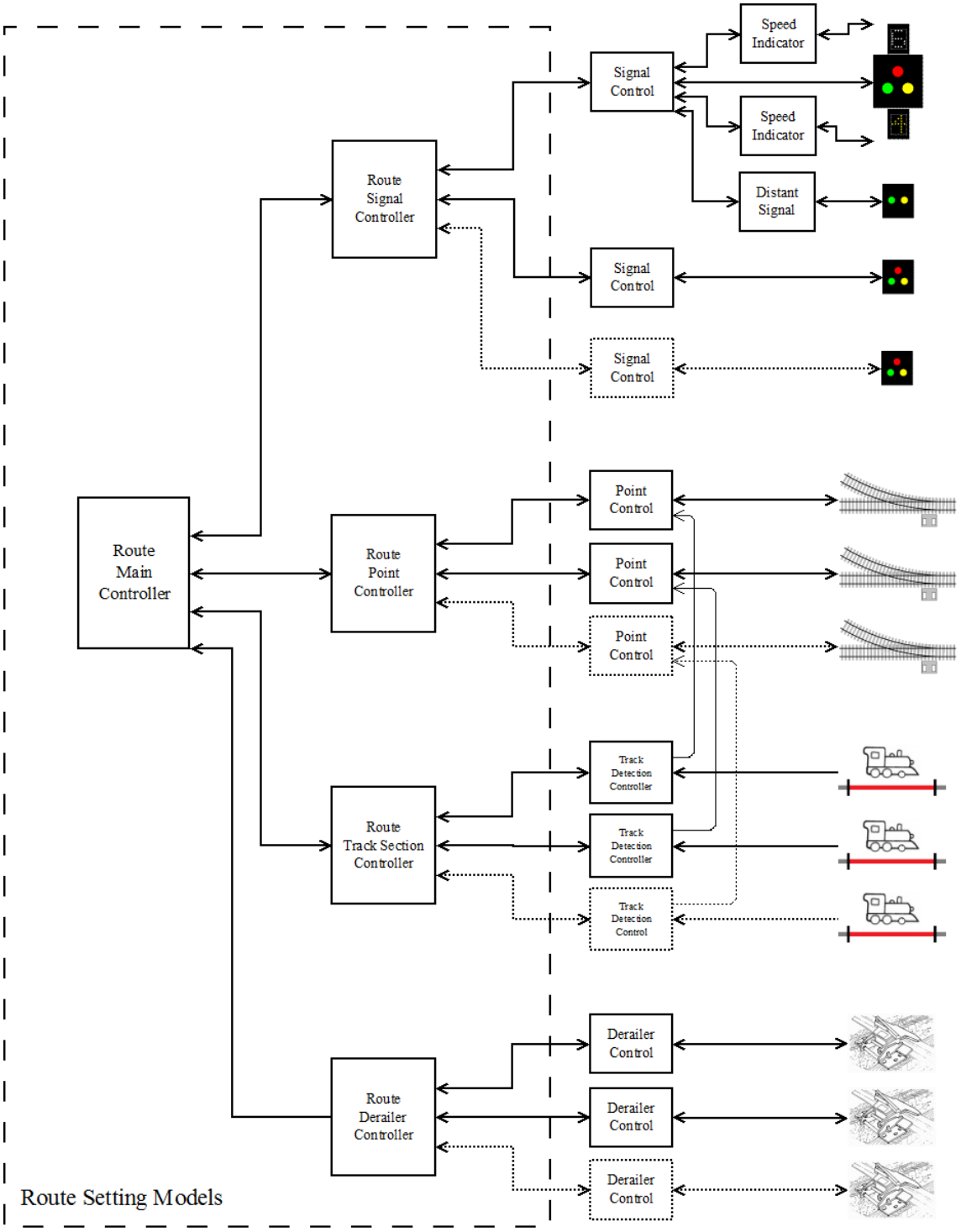

Figure 6.19 : Route setting main and sub-controllers [1].

Transitions:

- t1= P(1)pos ∧ P(2)pos ∧ P(3)pos ∧ P(4)pos ∧ P(5)pos
- t2= !(P(1)pos ∧ P(2)pos ∧ P(3)pos ∧ P(4)pos ∧ P(5)pos)
- t3= P(1)lock ∧ P(2)lock ∧ P(3)lock ∧ P(4)lock ∧ P(5)lock
- t4= !(P(1)lock ∧ P(2)lock ∧ P(3)lock ∧ P(4)lock ∧ P(5)lock)

- t5= available ∧ set_all
- t6= set_all
- t7= S4 ∧ T#15s
- t8= S5 ∧ T#2s
- t9= (!P(1)pos ∧ P(1)lock) ∨ (!P(2)pos ∧ P(2)lock) ∨ (!P(3)pos ∧ P(3)lock) ∨ (!P(4)pos ∧ P(4)lock) ∨ (!P(5)pos ∧ P(5)lock)
- t10= !t9

Equations of the model:

$$
\begin{aligned}
S_1 &= s_5 \cdot t_{10} + s_4 \cdot t_7 + s_2 \cdot t_2 + s_1 \cdot \bar{t}_1 \cdot \bar{t}_7 \cdot \bar{t}_9 \\
S_2 &= s_4 \cdot t_1 + s_1 \cdot t_1 + s_5 \cdot t_8 + s_3 \cdot t_4 + s_2 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_6 \\
S_3 &= s_5 \cdot t_3 + s_2 \cdot t_3 + s_3 \cdot \bar{t}_4 \\
S_4 &= s_1 \cdot t_5 + s_4 \cdot \bar{t}_1 \cdot \bar{t}_7 \\
S_5 &= s_2 \cdot t_6 + s_4 \cdot \bar{t}_3 \cdot \bar{t}_8 \\
S_6 &= s_1 \cdot t_9 + s_6 \cdot \bar{t}_{10}
\end{aligned}
\qquad \textbf{(6.9)}
$$

Function block in SilworX (Figure 6.20):

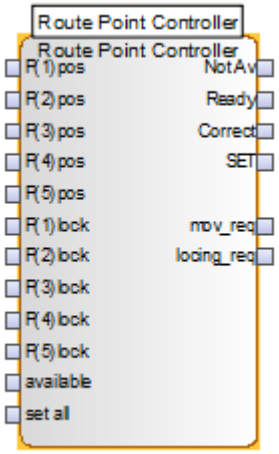

Figure 6.20 : Route point controller function block [1].

6.4.2 Route signal controller

Route signal controller is used to control and monitor start signal, exit signal and all flank protection signals.

Inputs:

- s_ready: Starting signal, target signal, flank protection signals are ready (internal input)
- ss_set: Starting signal is in “set” state (internal input)
- ss_block: Starting signal is in “blocked” state (internal input)
- fs_lock: Flank protection signals are blocked (internal input)
- set: Set starting signal (command)
- block_ss: Block starting signal (command)
- unblock_ss: Unblock starting signal (command)

States:

- S1: Busy (initial state) (at least one signal is not available in the particular route)
- S2: Ready (all signals are in the proper condition for setting)
- S3: SET (starting signal has been set and all flank protection signals have been locked)
- S4: Blocked (starting signal has been blocked in red aspect)
- S5: set_req (set signals to the star signal device and blocking signal to the flank protection signals)

FSM model (Figure 6.21):

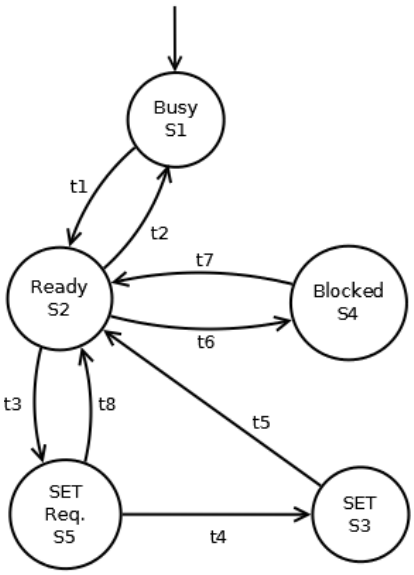

Figure 6.21 : Route signal controller model [1].

Transitions:

- t1= s_ready
- t2= !s_ready
- t3= set
- t4= ss_set ∧ fs_locked
- t5= !ss_set
- t6= block_ss
- t7= unblock_ss
- t8= S5 ∧ T#1s

Equations:

$$S_1 = s_2 \cdot t_2 + s_1 \cdot \bar{t}_1$$

$$S_2 = s_1 \cdot t_1 + s_3 \cdot t_5 + s_4 \cdot t_7 + s_5 \cdot t_8 + s_2 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_6$$

$$S_3 = s_5 \cdot t_4 + s_3 \cdot \bar{t}_5 \qquad \textbf{(6.10)}$$

$$S_4 = s_2 \cdot t_6 + s_4 \cdot \bar{t}_7$$

$$S_5 = s_2 \cdot t_3 + s_5 \cdot \bar{t}_4 \cdot \bar{t}_8$$

Function block is programed in SilworX software (Figure 6.22).

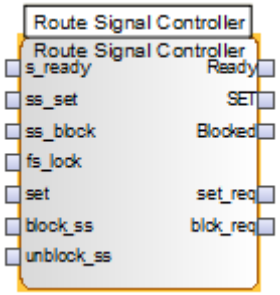

Figure 6.22 : Route signals controller function block [1].

6.4.3 Route track sections controller

This function provides status of all track sections in the route. It can obtain that all track sections in the route are free or there is an occupancy at least in one of them. In addition to that, route track section controller can set all track section with a single command.

Inputs:

- T(1)occ: Occupancy information of the first track section (sensor)
- T(2)occ: Occupancy information of the second track section (sensor)
- T(3)occ: Occupancy information of the third track section (sensor)
- T(4)occ: Occupancy information of the fourth track section (sensor)
- T(5)occ: Occupancy information of the fifth track section (sensor)
- T(6)occ: Occupancy information of the sixth track section (sensor)
- T(1)set: Set information of the first track section (internal input)
- T(2)set: Set information of the second track section (internal input)
- T(3)set: Set information of the third track section (internal input)
- T(4)set: Set information of the fourth track section (internal input)
- T(5)set: Set information of the fifth track section (internal input)
- T(6)set: Set information of the sixth track section (internal input)
- T(overlap)occ: Occupancy information of the overlap section (sensor)
- set_all: Set all track sections (command)
- clear_all: Clear all track sections (command)
- reset: Reset the model (command)

States:

- S1: Ready (initial state) (all track sections in the route are free)
- S2: Occupied (at least one track section in the route is occupied)
- S3: SET (all track sections in the route have been set)
- S4: Partial Set (some track sections in the route have been set and the others not)
- S5: Set Request (setting command to all track section control models)
- S6: Cancel Request (cancel all track sections which have already been)
- S7: Error

Designed model (Figure 6.23):

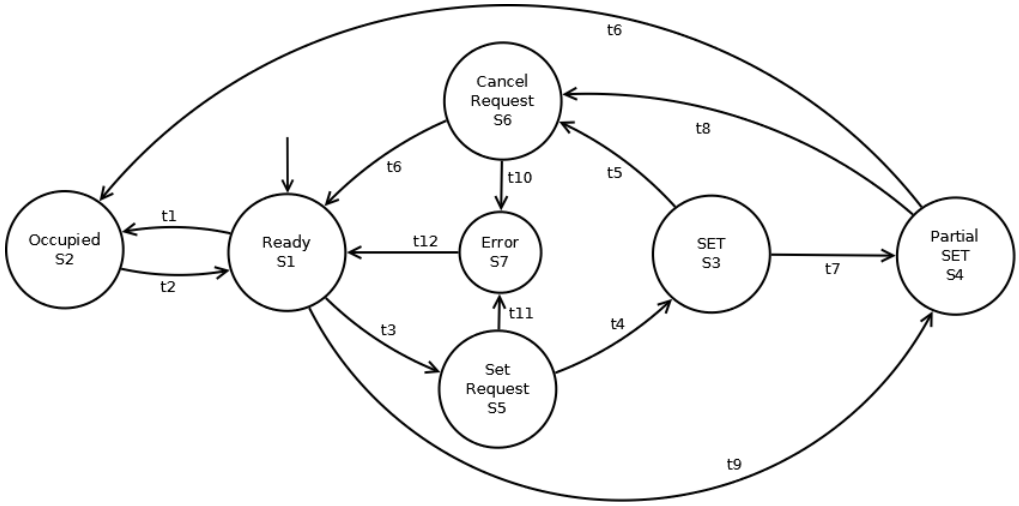

Figure 6.23 : Route track sections model [1].

Transitions:

- t1= T(1)occ ∨ T(3)occ ∨ T(3)occ ∨ T(4)occ ∨ T(5)occ ∨ T(6)occ ∨ T(overlap)occ
- t2= !T(1)occ ∧ !T(3)occ ∧ !T(3)occ ∧ !T(4)occ ∧ !T(5)occ ∧ !T(6)occ ∧ !T(overlap)occ
- t3= set_all
- t4= T(1)set ∧ T(2)set ∧ T(3)set ∧ T(4)set ∧ T(5)set ∧ T(6)set
- t5= clear_all
- t6= !T(1)set ∧ !T(2)set ∧ !T(3)set ∧ !T(4)set ∧ !T(5)set ∧ !T(6)set
- t7= !T(1)set ∨ !T(2)set ∨ !T(3)set ∨ !T(4)set ∨ !T(5)set ∨ !T(6)set
- t8= t2 ∧ clear_all
- t9= t6 ∧ (T(1)set ∨ T(2)set ∨ T(3)set ∨ T(4)set ∨ T(5)set) ∨ T(6)set)
- t10= S6 ∧ T#2s
- t11= S5 ∧ T#2s
- t12= reset

Equations:

$$S_1 = s_2 \cdot t_2 + s_6 \cdot t_6 + s_7 \cdot t_{12} + s_1 \cdot \bar{t}_1 \cdot \bar{t}_3 \cdot \bar{t}_9$$

$$S_2 = s_1 \cdot t_1 + s_4 \cdot t_6 + s_2 \cdot \bar{t}_2 \qquad \textbf{(6.11)}$$

$$S_3 = s_5 \cdot t_4 + s_3 \cdot \bar{t}_5 \cdot \bar{t}_7$$

$$S_4 = s_3 \cdot t_7 + s_1 \cdot t_9 + s_4 \cdot \bar{t}_6 \cdot \bar{t}_8$$

$$S_5 = s_1 \cdot t_3 + s_5 \cdot \bar{t}_4 \cdot \bar{t}_{11}$$

$$S_6 = s_3 \cdot t_5 + s_4 \cdot t_8 + s_6 \cdot \bar{t}_6 \cdot \bar{t}_{10}$$

$$S_7 = s_6 \cdot t_{10} + s_5 \cdot t_{11} + s_7 \cdot \bar{t}_{12}$$

Function block of route track sections controller can be seen in Figure 6.24.

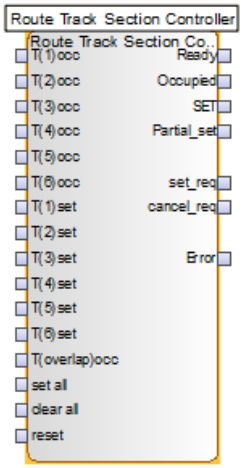

Figure 6.24 : Route track sections controller [1].

6.4.4 Route derailer controller

Route derailer controller is used to control all derailer in a certain route. It provides availability and proper position information of all derailers in a certain route to the route main controller. Route derailer controller can set and lock all derailers with a single command which is sent from the route main controller.

Inputs:

- D(1)pos: The proper position info of the first derailer on the route (internal input)
- D(2)pos: The proper position info of the second derailer on the route (internal input)
- D(3)pos: The proper position info of the third derailer on the route (internal input)

- D(4)pos: The proper position info of the fourth derailer on the route (internal input)
- D(5)pos: The proper position info of the fifth derailer on the route (internal input)
- D(1)lock: Locking info of the first derailer on the route (internal input)
- D(2)lock: Locking info of the second derailer on the route (internal input)
- D(3)lock: Locking info of the third derailer on the route (internal input)
- D(4)lock: Locking info of the fourth derailer on the route (internal input)
- D(5)lock: Locking info of the fifth derailer on the route (internal input)
- available: All derailers are available for the position changing (internal input)
- set_all: Set all derailers to the correct position according to the route (command)

States:

- S1: Incorrect (at least one derailer on the route is not in correct position)
- S2: Correct (all derailers on the route are in correct position)
- S3: Locked (all derailers on the route are locked)
- S4: Moving Request (move derailers to the correct position)
- S5: Locking Request (lock all derailers in proper position)
- S6: Not Available (at least one derailer on the route is not available)

Designed model is shown in Figure 6.25.

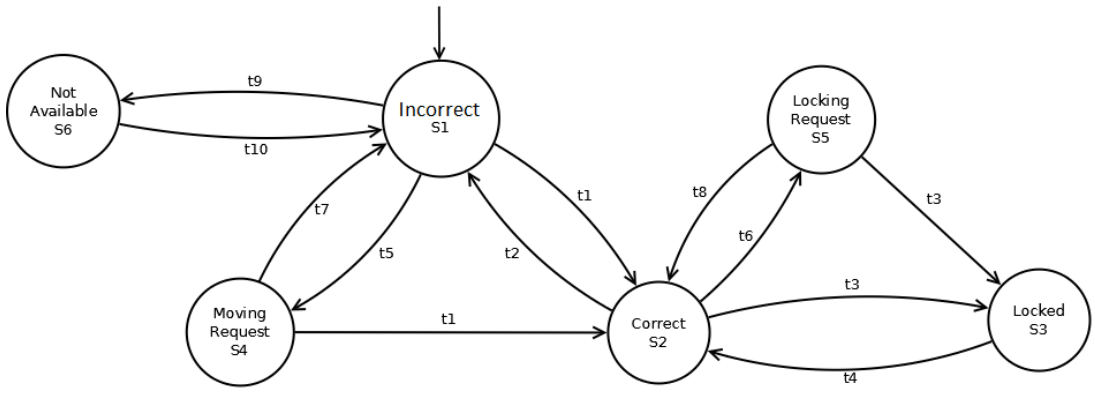

Figure 6.25 : Route derailer control model [1].

Transitions:

- t1= D(1)pos ∧ D(2)pos ∧ D(3)pos ∧ D(4)pos ∧ D(5)pos

- t2= !(D(1)pos ∧ D(2)pos ∧ D(3)pos ∧ D(4)pos ∧ D(5)pos)
- t3= D(1)lock ∧ D(2)lock ∧ D(3)lock ∧ D(4)lock ∧ D(5)lock
- t4= !(D(1)lock ∧ D(2)lock ∧ D(3)lock ∧ D(4)lock ∧ D(5)lock)
- t5= available ∧ set_all
- t6= set_all
- t7= S4 ∧ T#15s
- t8= S5 ∧ T#2s
- t9= (!D(1)pos ∧ D(1)lock) ∨ (!D(2)pos ∧ D(2)lock) ∨ (!D(3)pos ∧ D(3)lock) ∨ (!D(4)pos ∧ D(4)lock) ∨ (!D(5)pos ∧ D(5)lock)
- t10= !t9

Equations of the model:

$$
\begin{aligned}
S_1 &= s_5 \cdot t_{10} + s_4 \cdot t_7 + s_2 \cdot t_2 + s_1 \cdot \bar{t}_1 \cdot \bar{t}_7 \cdot \bar{t}_9 \\
S_2 &= s_4 \cdot t_1 + s_1 \cdot t_1 + s_5 \cdot t_8 + s_3 \cdot t_4 + s_2 \cdot \bar{t}_2 \cdot \bar{t}_3 \cdot \bar{t}_6 \\
S_3 &= s_5 \cdot t_3 + s_2 \cdot t_3 + s_3 \cdot \bar{t}_4 \\
S_4 &= s_1 \cdot t_5 + s_4 \cdot \bar{t}_1 \cdot \bar{t}_7 \\
S_5 &= s_2 \cdot t_6 + s_4 \cdot \bar{t}_3 \cdot \bar{t}_8 \\
S_6 &= s_1 \cdot t_9 + s_6 \cdot \bar{t}_{10}
\end{aligned}
\qquad \textbf{(6.12)}
$$

Function block of route derailer controller in SilworkX can be seen in Figure 6.26.

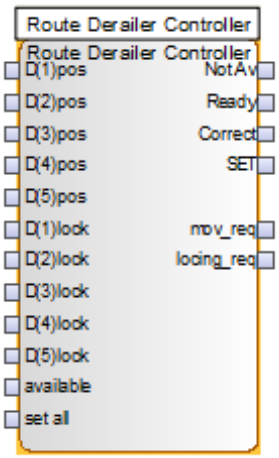

Figure 6.26 : Route derailer controller function block [1].

6.4.5 Route main controller

Route main controller is the main controller for route setting. It uses four sub-models given above.

Inputs:

- set: Set the route (command)
- sig_ready: Signals are ready info from the route signals controller (internal input)
- sw_ready: Points are ready info from the route point controller if it exists in the route (internal input)
- dr_ready: Derailers are ready info from the route derailer controller if it exists in the route (internal input)
- ts_ready: Track sections are ready info from the route track sections controller (internal input)
- ts_set: Track sections have been set (internal input)
- dr_set: Derailer have been set (internal input)
- sw_set: Points have been set (internal input)
- sig_set: Signals have been set (internal input)

States:

- S1: Busy (initial state) (the route is not available)
- S2: Ready (the route is ready to be set)
- S3: Set Request (set command to all sub-controllers)
- S4: Set (the route has been set)

Created FSM model (Figure 6.27):

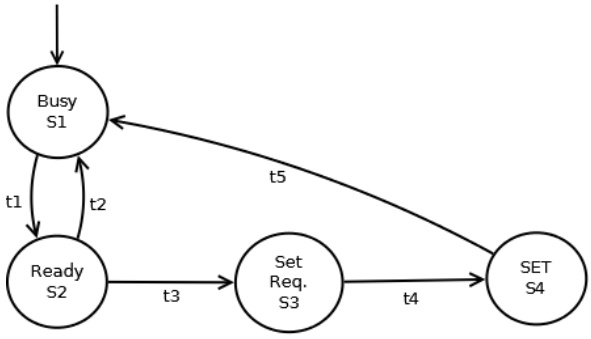

Figure 6.27 : Route main controller model [1].

Transitions:

- t1= sig_ready ∧ sw_ready ∧ dr_ready ∧ ts_ready
- t2= !sig_ready ∨ !sw_ready ∨ !dr_ready ∨ !ts_ready
- t3= set
- t4= tr_set ∧ dr_set ∧ sw_set ∧ sig_set
- t5= !tr_set ∨ !dr_set ∨ !sw_set ∨ !sig_set

Equations:

$$S_1 = s_2 \cdot t_2 + s_4 \cdot t_5 + s_1 \cdot \bar{t}_1$$
$$S_2 = s_1 \cdot t_1 + s_2 \cdot \bar{t}_2 \cdot \bar{t}_3$$
$$S_3 = s_2 \cdot t_3 + s_3 \cdot \bar{t}_4 \quad \textbf{(6.13)}$$
$$S_4 = s_3 \cdot t_4 + s_4 \cdot \bar{t}_5$$

Route main controller function block is shown in following Figure 6.28.

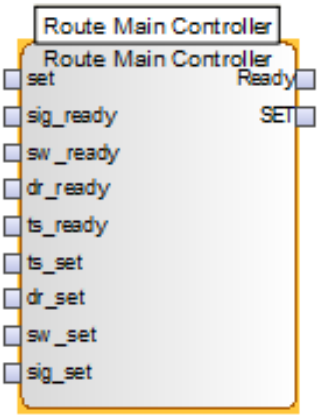

Figure 6.28 : Route main controller function block [1].

6.5 Sample Route Interlocking Design

In this chapter, first route in the route table of the model station will be implemented with using developed FSM models. The specifications and appearance of route 1 can be seen in Figure 6.29 and Figure 6.30.

No	Start S.	Target S.	Type	Path	Switches		Drail	Flank Protection					
					Normal	Reverse	Out	Switch Normal	Switch Rev.	Signal	Track	Drail In	Overlap
1	S7	S2	M	T17, T12, T6, T2	P2a, P2b, P4	-	-	P1, P3, P5	-	-	-	-	T9
2	S7	S3	M	T17, T11, T5, T3	P2a	P2b, P3	-	P1, P5	-	S8, S5	T6, T12	-	T8

Figure 6.29 : Route 1 elements in the route table [1].

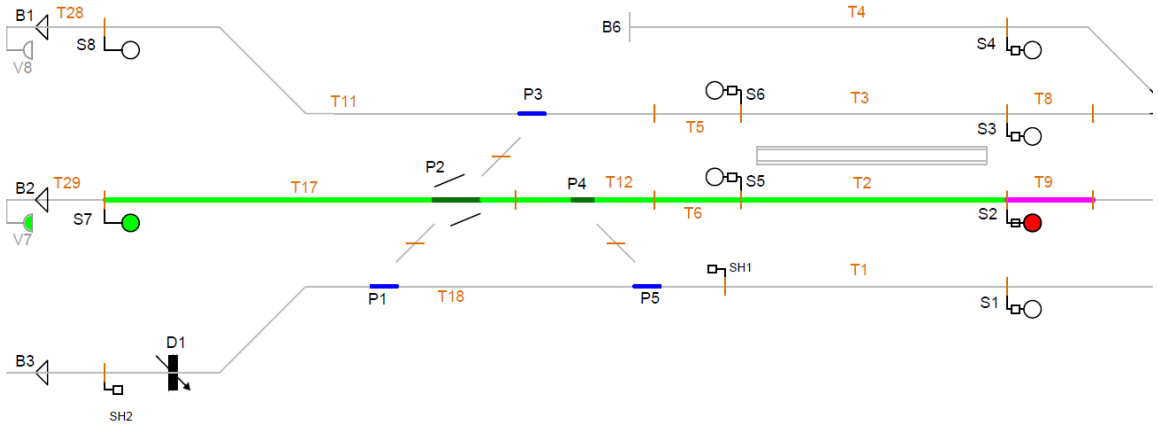

Figure 6.30 : Objects have been created with respect to the route table.

6.5.1 Object models

Track sections:

Track section controls in the route 1 are created using track clear detection control model. The same model is duplicated for all track sections because; it is a generic model. T11 and T18 are not elements of the route 1 but they have to be created to provide occupancy information of points in flank protection.

Track section occupancy inputs are defined as global variables to simulate them manually. Set commands of the track sections are connected with only route 1 because the other routes are not designed. There can be other route's set command connected with an "OR" gate to the same input. See Figure 6.31.

Connectors with relevant names are defined for all outputs of the function blocks to make it reachable from all other functions.

Points:

Figure 6.32 and Figure 6.33 show created point controls in the route 1.

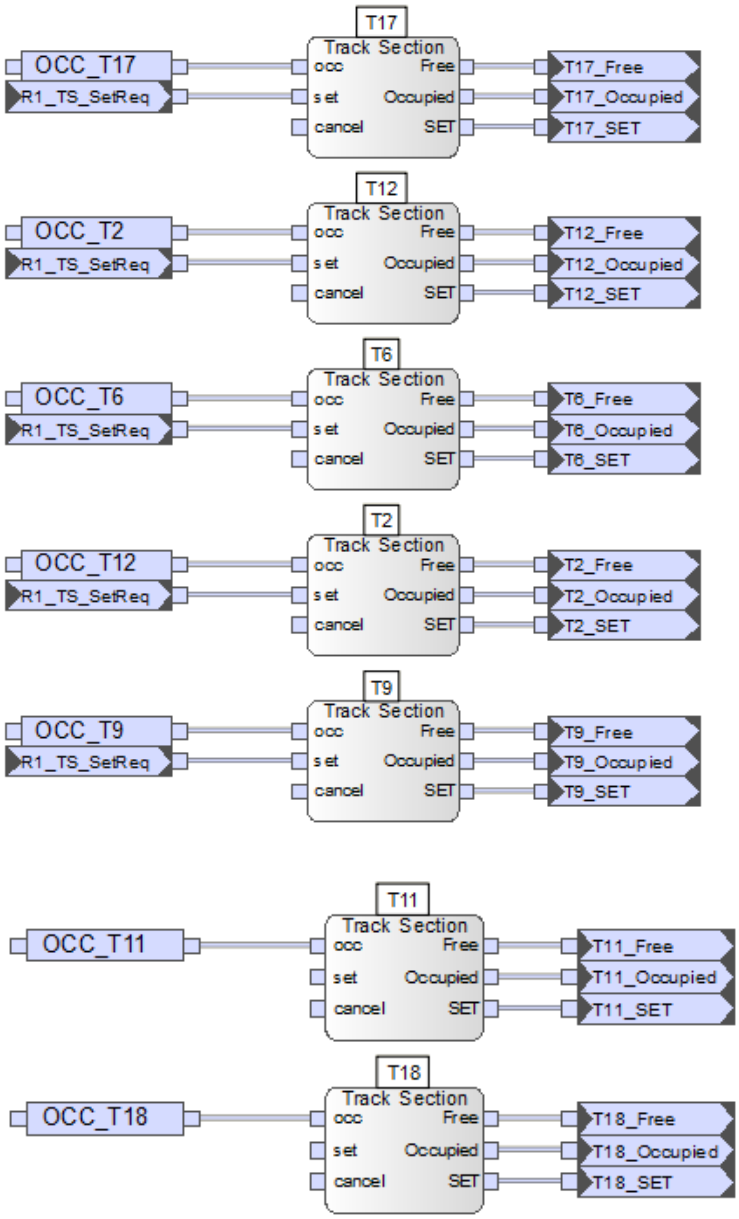

Figure 6.31 : Created track sections in the route 1 [1].

There are 3 points in the route path and 3 more points in the flank area. Their control models are generated. However, there must be real a point machine which connected to them to monitor their workings. Therefore, a simple point simulator is designed to simulate behavior of a simple point. Different point moving time can be defined for every points if they have different moving times. “Change” input in the point simulator can be used to control the point in the local control.

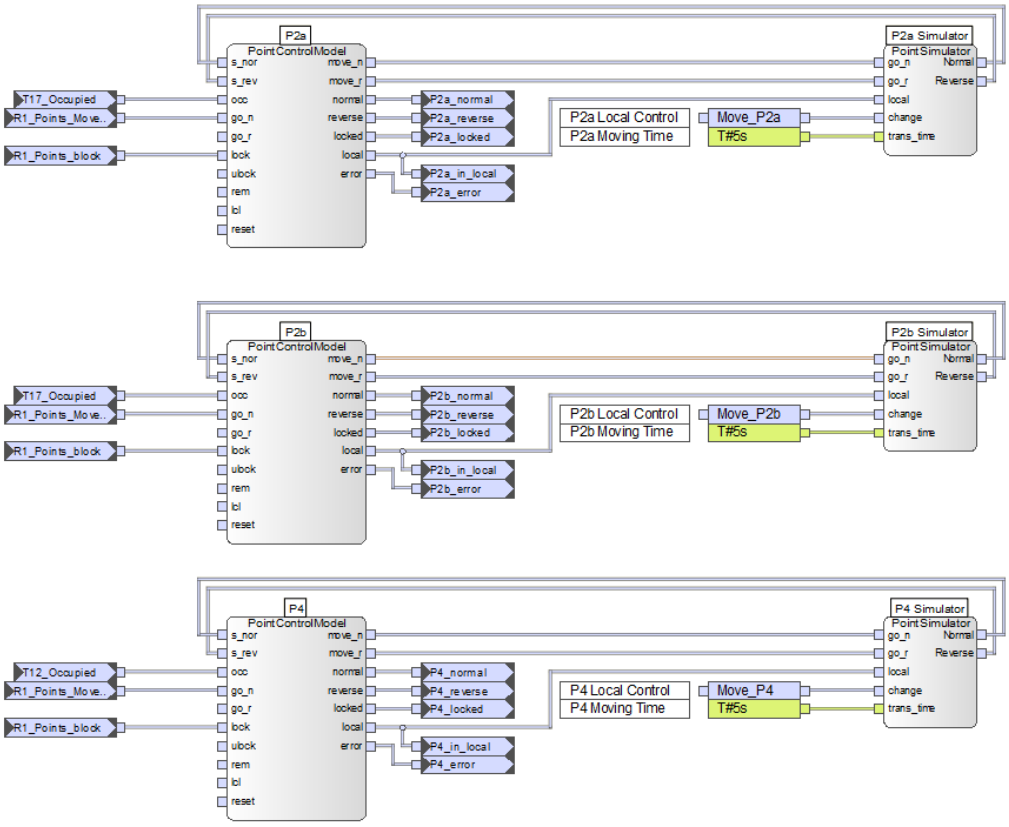

Figure 6.32 : Created point controls in the route 1 (1) [1].

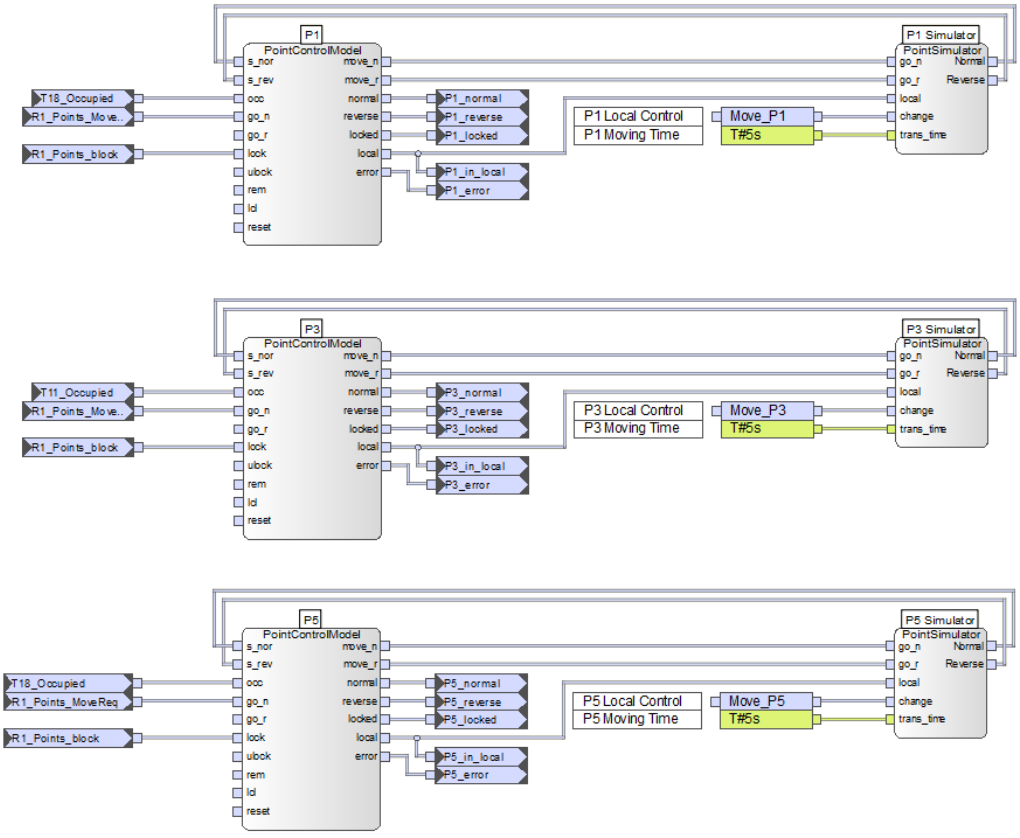

Figure 6.33 : Created point controls in the route 1 (2) [1].

Signals:

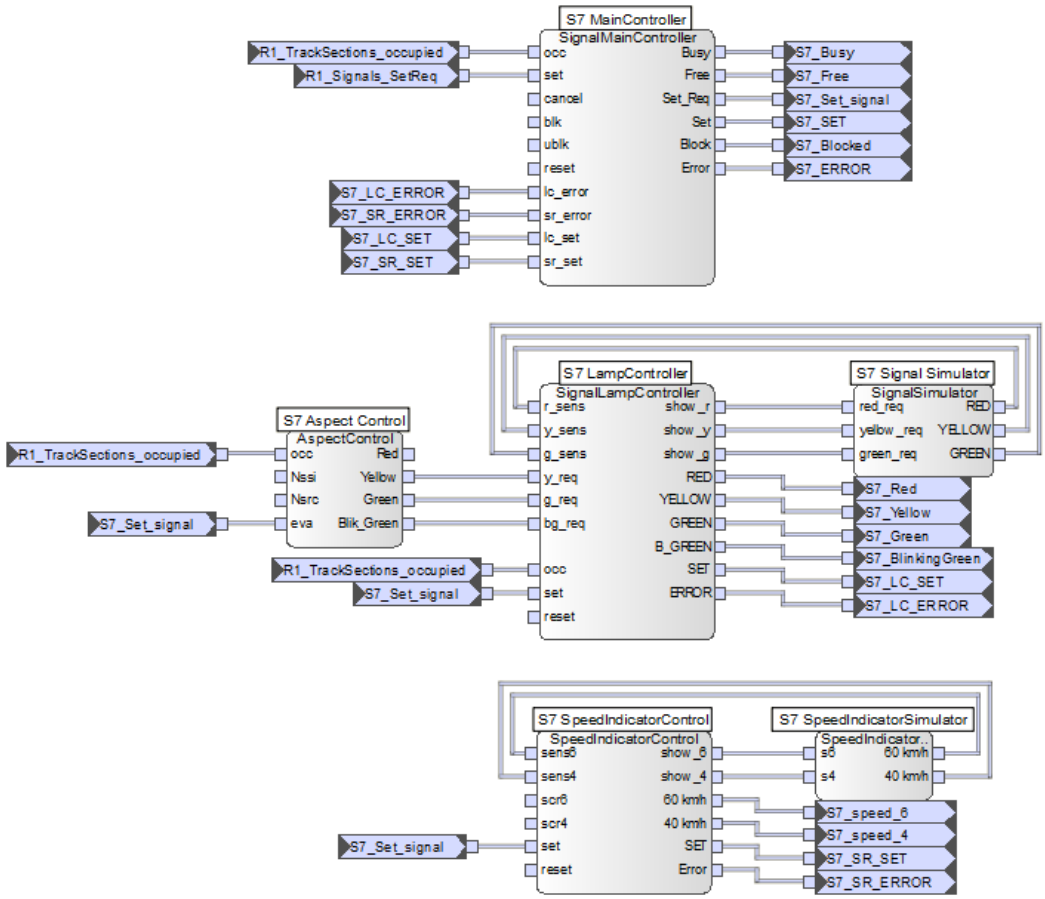

Figure 6.34 : Created starting signal of route 1 [1].

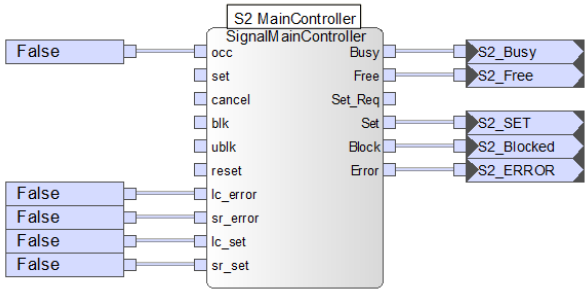

Figure 6.35 : Created exit signal of route 1 [1].

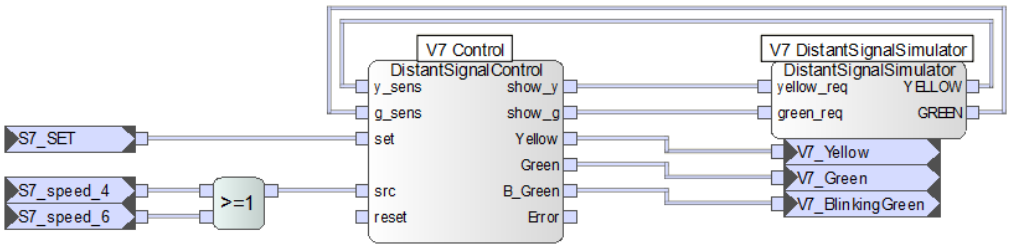

Figure 6.36 : Created distant signal of route 1 [1].

Starting signal (Figure 6.34), exit signal (Figure 6.35) and distant signal (Figure 6.36) are generated with relevant function blocks. However, exit signal is not generated completely because; all features of the exit signal are not necessary for route 1.

Simulation functions are also used to obtain required feedbacks to the signal control models.

6.5.2 Route function models

All route setting functions are generated with proper inputs and outputs. Sequence of the route setting starts with arrangements of the points and continues with track sections setting and signal setting respectively.

Route point controller (Figure 6.37) model is developed for maximum 5 points, but more than 5 points can be connected to it with using logical "AND" gates.

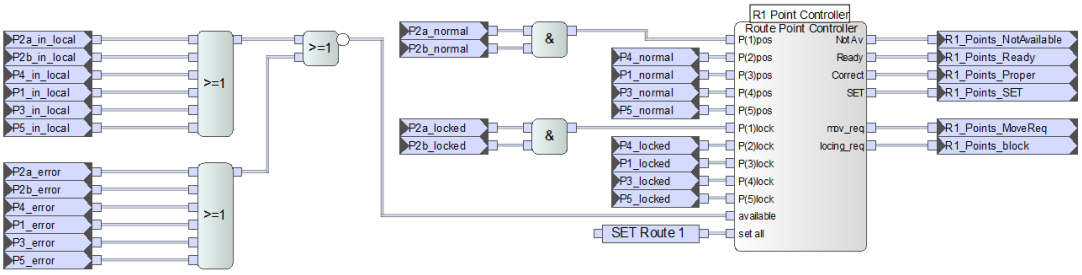

Figure 6.37 : Route 1 point controller [1].

Route track sections controller (Figure 6.38) has 6 track section inputs. When there are less than 6 points, one of them can be duplicated to empty inputs. Route 1 has only 4 track sections. Therefore, T2 is connected also as fifth and sixth track section.

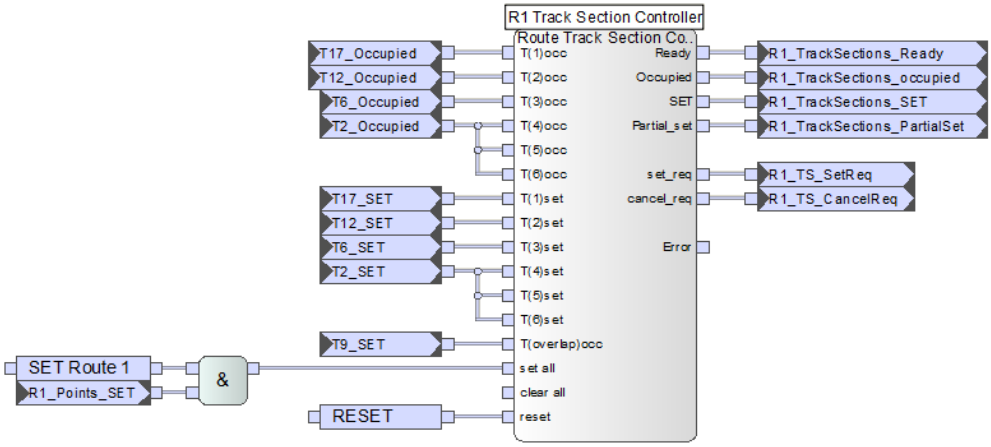

Figure 6.38 : Route 1 track sections controller [1].

Locking states of the signals in flank protection are connected to fs_lock input in the signal controller (Figure 6.39). However, route 1 does not have any flank protection signal. Therefore, a logical constant “True” is connected to fs_lock input.

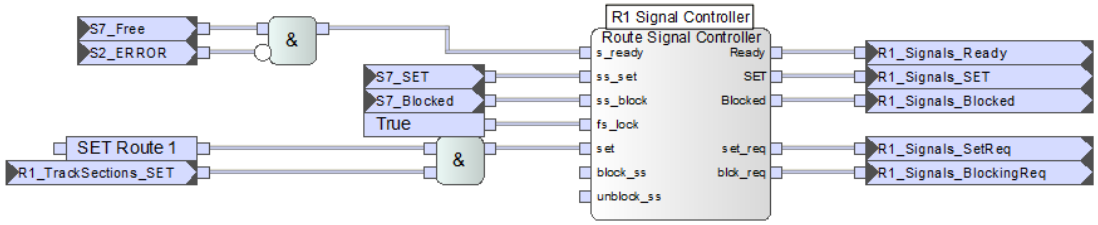

Figure 6.39 : Route 1 signals controller [1].

Route 1 main controller (Figure 6.40) is the main function block to set route 1. It evaluates the conditions of route elements continuously. If all elements are in convenient for route 1, “ready” output is activated and it starts to be available for route setting demand (conflicting routes protection are not considered). In other words, if it is not ready to be set, setting demand would be rejected. Following Figure 6.41 shows a simulation of the situation when there is an occupancy in the T12. Route 1 cannot be set because of an occupancy in T12. Red lines represent “active line” or logical “true” and blue lines represent “passive line” or logical “false” in the simulation.

When the set input is forced, route setting procedure are started. To obey route setting sequence, main controller send a command to the route point controller firstly. After all points are locked in proper positions track circuits are set. Finally, clearing command is sent to the route signal controller. Once the feedback is received from the signal controller, route setting function is completed.

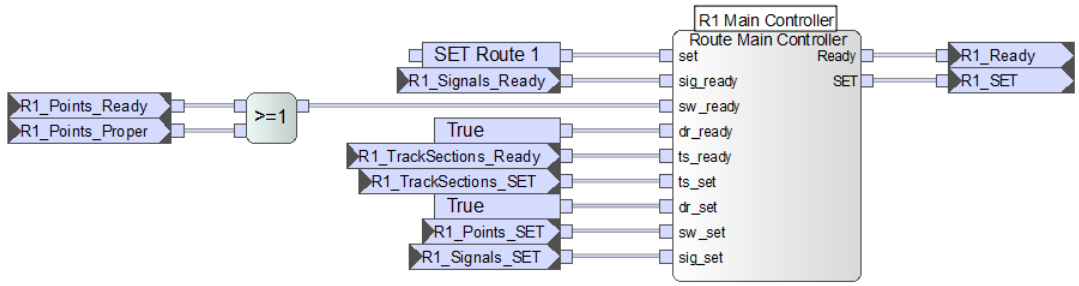

Figure 6.40 : Route 1 main controller [1].

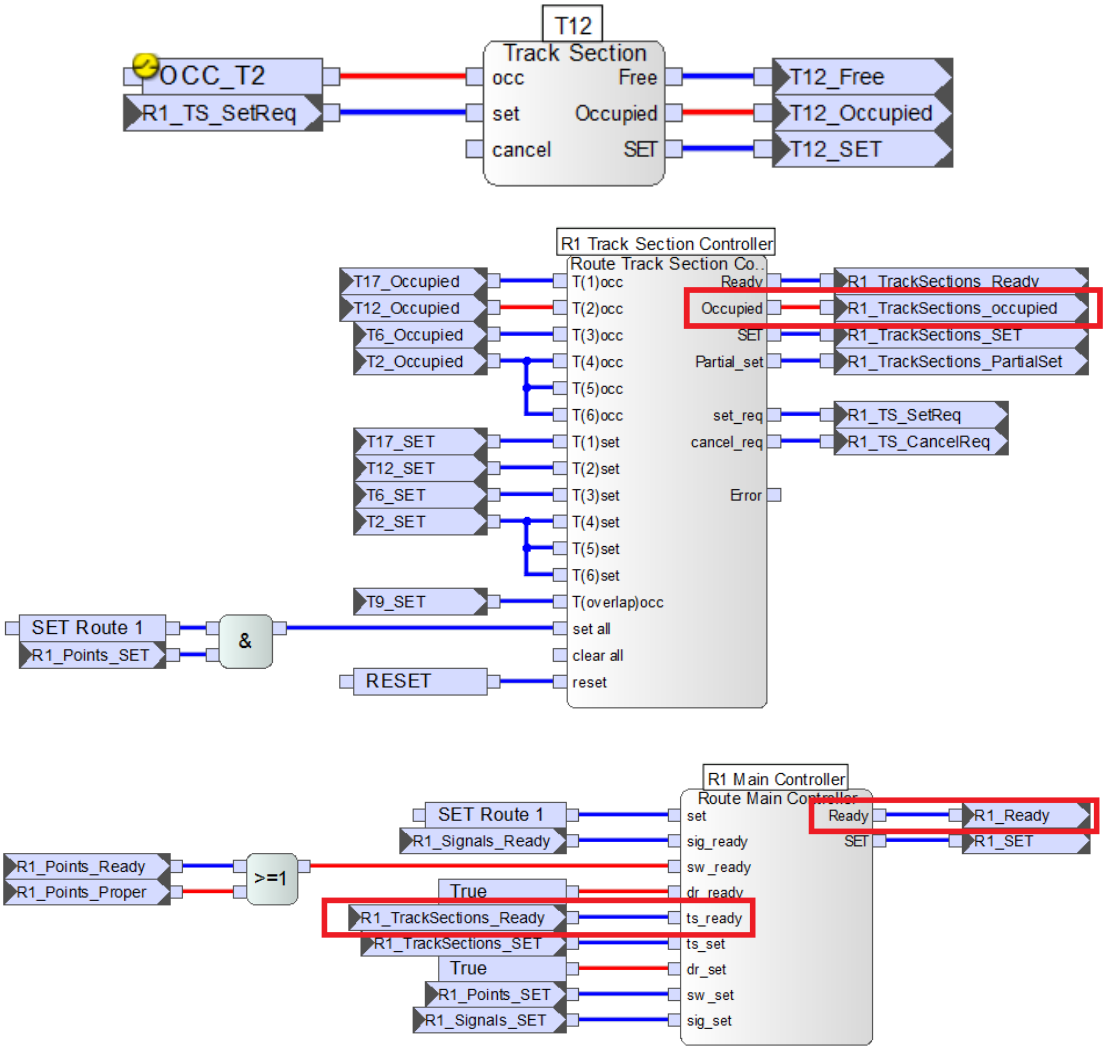

Figure 6.41 : An occupancy situation in T12 [1].

7. RAMS

7.1 Introduction

Reliability, Availability, Maintainability & Safety (RAMS) is defined to indicate the quality and working performance of a system. It is a system characteristic and can be achieved by the application of some particular methods, tools and techniques which performed through whole lifecycle of the system [38]. Figure 7.1 shows the complete lifecycle of a system. It is also called "V diagram".

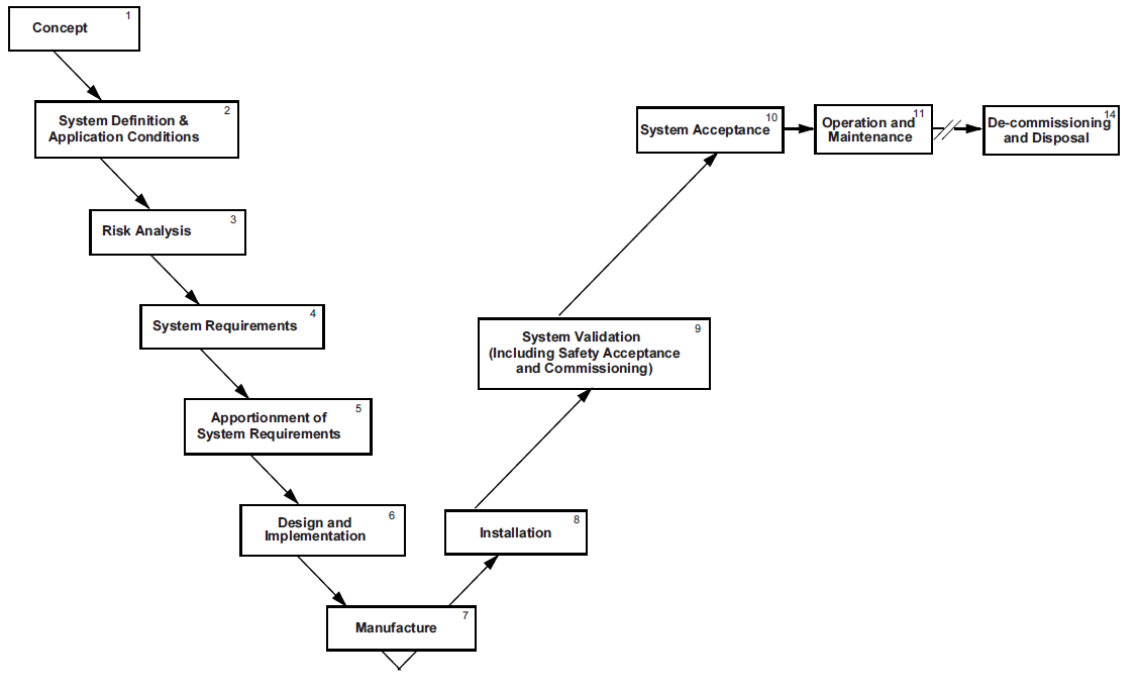

Figure 7.1 : The lifecycle phases of a system [38]

The European Standard EN 50126 explains processes for the specification and demonstration of RAMS requirements. Basic elements of the RAMS are described in the same standard as:

Reliability: probability that an item can perform a required function under given conditions for a given time interval.

Availability: ability of a product to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval assuming that the required external resources are provided.

Maintainability: probability that a given active maintenance action, for an item under given conditions of use can be carried out within a stated time interval when the maintenance is performed under stated conditions and using stated procedures and resources.

Safety: the state of a system freedom from unacceptable risk of harm.

7.1.1 Essential terms related to probability used for RAMS

Probability Distribution Function; A probability distribution is a mapping of all the possible values of a random variable to their corresponding probabilities for a given sample space. It is denoted for discrete systems as

$$\mathrm{P}(X = x) \tag{7.1}$$

Or shortly,

$$\mathrm{P}(x)$$

The probability distribution function is defined to obtain the probability for this random variable to take on a given value. The probability distribution function also known probability density function (PDF) is denoted as

$$f(x) = P(X = x) \tag{7.2}$$

Cumulative Distribution Function; The cumulative distribution function is defined for a random variable and it provides, for each value x, the probability of a result less than or equal to X

$$\mathrm{F}(x) = \mathrm{P}(X \leq x) \tag{7.3}$$

For continuous random variables, it is defined by

$$\mathrm{F}(t) = \int_{-\infty}^{t} f(x)dx \tag{7.4}$$

If the random variable is assumed to be ∞

$$\mathrm{F}(t) = \int_{-\infty}^{\infty} f(x)dx = 1 \qquad \textbf{(7.5)}$$

That means total area under the probability distribution curve is equal to unity.

Failure rate; Failure rate (also known as hazard rate) is described as the failure frequency of a component or system in a certain time. It is often denoted by λ and expressed by

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{-\frac{d}{dt}R(t)}{R(t)} \qquad \textbf{(7.6)}$$

Where

$\lambda(t)$: hazard rate (i.e., time-dependent failure rate)

$f(t)$: failure distribution function

$R(t)$: reliability function

Hazard rate function of an engineering system has a characteristic as can be show in the Figure 7.2. This curve is called bathtub.

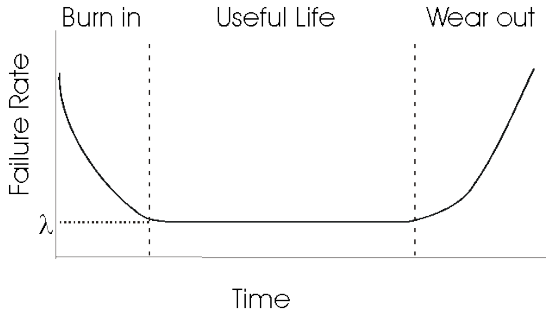

Figure 7.2 : Bathtub curve [39].

As shown in the figure, bathtub curve is divided into three regions: burn-in period, useful-life period and wear-out period. Burn-in period represents the early life of the system and in this area the failure rate is high but rapidly decreasing as defective products are identified and discarded. Some reasons for the occurrence of failure in the burn-in period can be listed as follows.

- Inadequate manufacturing methods

- Substandard control
- Poor quality materials
- Poor processes
- Inadequate debugging
- Human failures

During the useful life period, the hazard rate remains constant. Following reasons can be given for the occurrence of failure in this period.

- Weak safety factors
- Abuse
- Higher random stress than expected
- Undetectable defects
- Natural failures
- Human failures

In the late life of the product, the failure rate increases, as age and wear take their toll on the system. Some failure reasons in the wear-out period are:

- Poor maintenance
- Wear due to aging
- Wear due to friction
- Corrosion

Repair Rate; Repair rate is the number of repairs of a component in a certain time and is represented by μ. Failure rate and repair rate can be defined as follows

$$\lambda = \frac{\text{number of failures of a component in the given period of time}}{\text{total period of time the component was operating}}$$

$$\mu = \frac{\text{number of repairs of a component in the given period of time}}{\text{total period of time the component was repaired}}$$

Reliability Function; Reliability function is a property of any random variable that maps a set of events associated with failure of some system, onto time.

$$R(t) = 1 - F(t) \qquad \textbf{(7.7)}$$

In exponential form,

$$R(t) = e^{-\int_0^t \lambda(t)d(t)} \quad \textbf{(7.8)}$$

Mean Time to Failure; Mean time to failure (MTTF) represents the mean time expected until the first failure in the system. It is defined for non-repairable systems. It can be calculated with reliability function by following equation

$$\mathrm{MTTF} = \int_0^{\infty} \mathrm{R(t)dt} \quad \textbf{(7.9)}$$

Mean Time Between Failures; Mean time between failures (MTBF) is the predicted elapsed time between inherent failures of a system during operation. The difference between MTTF and MTBF is that while MTBF is used for products than that can be repaired and returned to use (repairable systems), MTTF is used for non-repairable systems.

Mean Time to Repair; Mean time to repair (MTTR) is the average time required to repair a component in the system. It is considered for repairable system and it is very important in terms of maintenance optimization.

7.2 RAMS Methods

There are several methods used to calculate RAMS parameters of a system. Some of them are given below:

- Fault-Tree Analysis (FTA)
- Markov Model
- Failure Modes and Effect Analysis (FMEA)
- Hazard and Operability Analysis (HAZOP)
- Interface Safety Analysis
- Preliminary Hazard Analysis (PHA)

In this chapter, Fault-Tree Analysis and Markov Method are used to obtain some particular RAMS parameter. Other methods also can be used but they are chosen because they can be implement easily to the railway systems.

7.2.1 Fault-Tree analysis

Fault-tree analysis (FTA) is a widely used method to evaluate the reliability of a system especially in its design and development phase. Nuclear power generation and railway signalling systems can be given as example industries which used this method. This type of critical systems have some particular undesirable events and they can cause several dangerous consequences. In fault-tree analysis, all basic events which can be lead undesirable events, known as the top event, are described and the logical relationship of them are determined. This logical relationship is easily depicted with using a tree structure and basic logic gates such as AND, OR, etc.

Some important main objectives of the FTA are listed below.

- To identify critical areas and cost-effective improvements.
- To understand the functional relationship of system failures.
- To satisfy jurisdictional requirements.
- To confirm the ability of the system to satisfy its imposed safety related requirements.
- To understand the degree of protection that the design concept provides against failures.

Following prerequisites should be considered before performing a FTA.

- Design, operation and maintenance of the system should be understood clearly.
- Analysis scope and objectives should be defined clearly.
- Clear definition of what constitutes system failure (undesirable event)
- Definition of system physical bounds and system interfaces clearly.
- Identification of associated assumptions
- A comprehensive review of system operational experience.

The first step should be determining of the top event in a FTA. Basic fault events that can cause the occurrence of the top event are generated and connected with logic gates.

Basic event can be a failure of an elementary component or part. The basic fault-event parameters are failure probability, failure rate, unavailability and repair rate.

Basic fault-tree symbols are shown in Figure 7.3.

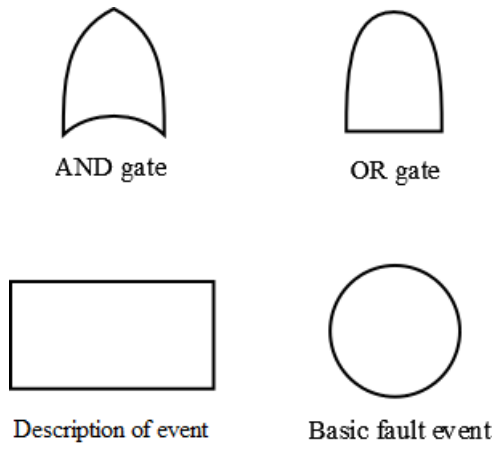

Figure 7.3 : Basic fault-tree symbols [1].

To calculate the occurrence probability of the top event, all basic events occurrence probabilities have to be known. The occurrence probability of an OR gate can be calculated with following equation [40].

$$P(A) = 1 - \prod_{i=1}^{m} \{1 - P(A_i)\} \qquad \textbf{(7.10)}$$

Where,

A : OR gate output fault event

$P(A)$: occurrence probability of the OR gate output fault event A

m : number of OR gate input fault events

$P(A_i)$: probability of occurrence of the OR gate input fault event A_i, for i = 1,2,…,m

Occurrence probability of the AND is given by [40]

$$P(B) = \prod_{i=1}^{k} P(B_i) \qquad \textbf{(7.11)}$$

B : AND gate output fault event

$P(B)$: occurrence probability of the AND gate output fault event B

k : number of AND gate input fault events

$P(A_i)$: probability of occurrence of the AND gate input fault event B_i, for i = 1,2,…,k

An example fault-tree analysis have been implemented to explain these calculations. Assume that there are four light bulbs for lightening a room and a switch is used to control the bulbs. There is also a fuse which protect the system. The probability of "dark room" event can be calculated easily by fault-tree analysis.

The top event is "dark room" and the intermediate events which cause the occurrence of "dark room" are:

- Power supply failure
- Switch failure
- All bulbs have failure

The fault tree of this example can be seen in Figure 7.4.

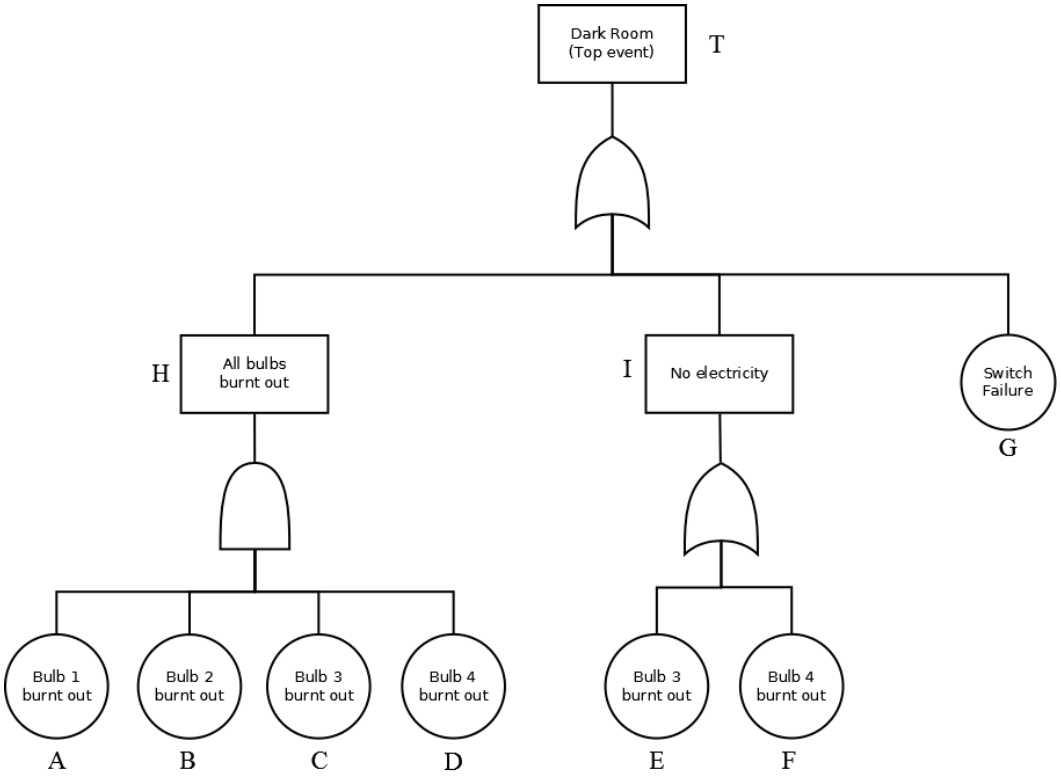

Figure 7.4 : Example fault tree [1].

If the occurrence probabilities of the base event are given as:

- P(A) = 0.07
- P(B) = 0.06
- P(C) = 0.05
- P(D) = 0.04
- P(E) = 0.03

- P(F) = 0.02
- P(G) = 0.01

To calculate intermediate events H and I the equation **(7.10)** and **(7.11)** are used.

$$P(H) = \prod_{i=1}^{4} P(H_i)$$

$$P(H) = \big(P(A)\big) \cdot \big(P(B)\big) \cdot \big(P(C)\big) \cdot \big(P(D)\big)$$

$$P(H) = (0.07) \cdot (0.06) \cdot (0.05) \cdot (0.04)$$

$$P(H) = 0.0000084 \qquad \textbf{(7.12)}$$

Similarly,

$$P(I) = 1 - \prod_{i=1}^{2} \{1 - P(I_i)\}$$

$$P(I) = 1 - \big[\big(1 - P(E)\big) \cdot \big(1 - P(F)\big)\big]$$

$$P(I) = 1 - [(1 - 0.03) \cdot (1 - 0.02)]$$

$$P(I) = 0.0494 \qquad \textbf{(7.13)}$$

Finally, $P(T)$ can be calculated with $P(H)$ and $P(I)$,

$$P(T) = 1 - \big[(1 - P(H)\,) \cdot \big(1 - P(I)\big) \cdot \big(1 - P(G)\big)\big]$$

$$P(T) = 1 - [(1 - 0.0000084) \cdot (1 - 0.0494) \cdot (1 - 0.01)]$$

$$P(T) = 0.0589 \qquad \textbf{(7.14)}$$

Thus, probability of occurrence of top event (“dark room”) is 0.0589.

7.2.2 Markov model

Markov model is another widely used method to perform reliability analysis of several engineering systems including railways. Generally, it is used to model repairable systems with constant failure and repair rates. Markov models can be defined for

continuous and the discrete time [41]. In this chapter, only discrete Markov models are considered.

For a given system, a Markov model consist of a set of all possible states, transitions between those states and the conditions described for the transitions. Conditions of the transitions are generally consist of failures and repairs in reliability analysis. A Markov model is represented with a graph as shown in Figure 7.5. The circles symbolize the states and arrows denoting the transition paths between states. Following graph is a simple Markov model.

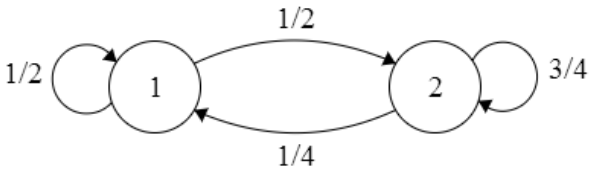

Figure 7.5 : A simple markov model [1].

In this Markov model, the system has only two states designated 1 and 2. The probabilities of remaining in or leaving a particular state in a finite time are also shown in the figure. As it mentioned before, the probabilities in the markov model are assumed to be constant for all times into the future.

If it is assumed that the system is in state 1 in the first time interval, the system can remain in state 1 with a probability of 1/2 or it can move into the state 2 with the same probability (1/2). Similarly, once the system is in state 2, it can remain in it with a probability of 3/4 or it can pass back to state 1 with a probability of 1/4 during the next time interval. This states can also be represented by a matrix called *transition probability matrix*.

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

Where

P : transition probability matrix.

The important point is that the sum of the probabilities defined for a state must be unity. This principle applies equally to all systems no matter what degree of

complexity exist or how many transitions there are between the states. The sum of the probabilities of remaining in or moving out of a state must be unity.

The behavior of the system are illustrated for three time interval in the Figure 7.6. The initial state of the system is state 1, therefore it starts with state 1. The probability of following any one branch of this tree can be evaluated by multiplying the appropriate probabilities of each step of this branch. The calculated branch probabilities can be seen in Table 7.1.

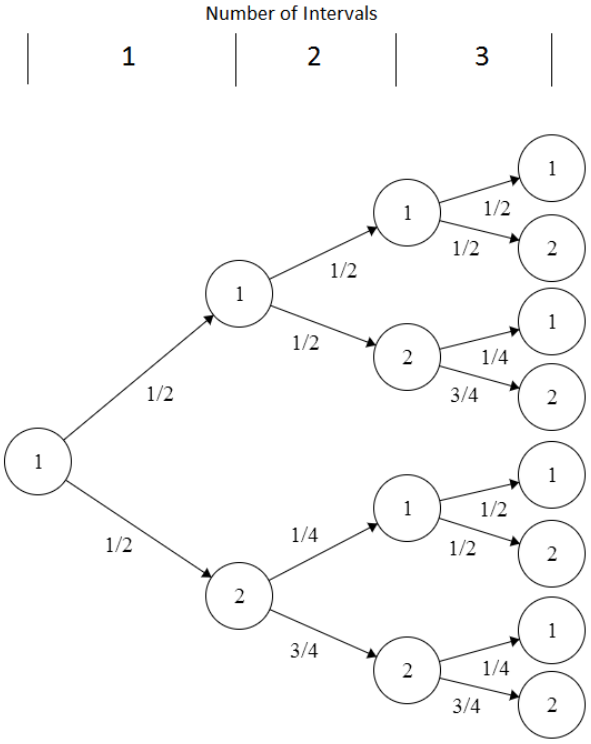

Figure 7.6 : Tree diagram of the system [1].

Table 7.1 : State probabilities of the example markov model [1]

Time Interval	State Probability	
	State 1	State 2
1	1/2=0.5	1/2=0.5
2	3/5=0.375	5/8=0.635
3	11/32=0.344	21/32=0.656

If the state probabilities are calculated for more time interval and the results are represented with a graphic, it will be possible to make some comments about the system characteristic.

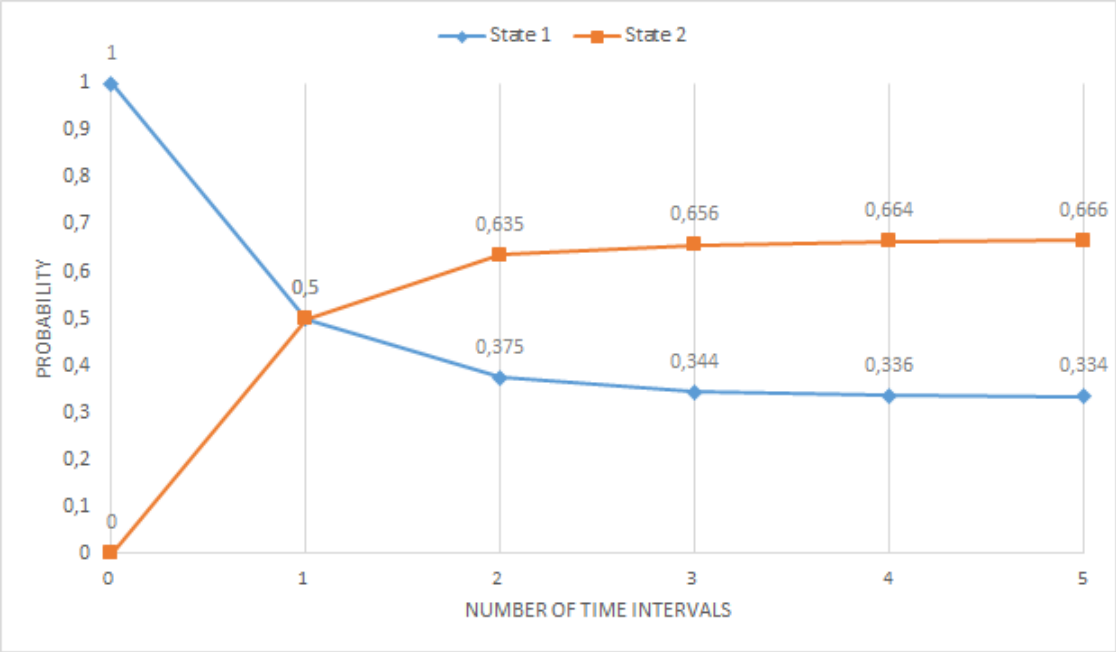

Figure 7.7 : System transient behavior

The characteristic in the **Figure 7.7** [1] are known as the transient behavior or time-dependent values of the state probabilities. As it can be recognized in the **Figure 7.7** the values of the state probabilities trend to a constant or limiting value. This is characteristic of most systems which satisfy the conditions of the Markov approach.

Following Markov model (Figure 7.8) represents operational behavior of a component in a system [40]. State 0 and state 1 are "operating" and "failed" states respectively. And the transitions consist of the failure and repair rates.

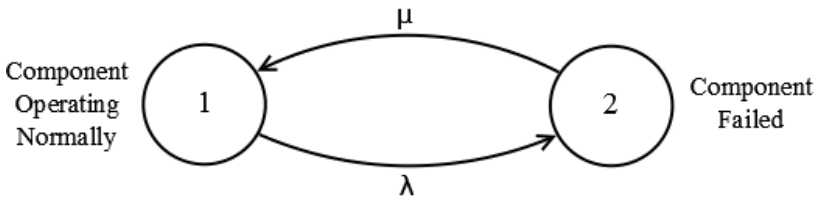

Figure 7.8 : Markov model of a component [1].

Following equations can be defined for the transition probabilities

$$P_0(t+\Delta t) = P_0(t)(1-\lambda\Delta t) + P_1(t)\,\mu\Delta t \qquad \textbf{(7.15)}$$

$$P_1(t+\Delta t) = P_1(t)(1-\mu\Delta t) + P_0(t)\,\lambda\Delta t \qquad \textbf{(7.16)}$$

Where

t : time

$\lambda\Delta t$: probability of component failure in finite time interval Δt

$\mu\Delta t$: probability of component repair in finite time interval Δt

$P_0(t+\Delta t)$: probability of component being in operating state 0 at time $t+\Delta t$

$P_1(t+\Delta t)$: probability of component being in failed state 1 at time $t+\Delta t$

$1-\lambda\Delta t$: probability of no failure in finite time interval Δt

$1-\mu\Delta t$: probability of no repair in finite time interval Δt

P_i : probability that the component is in the state *i* at time *t*, for *i*=0,1.

From the equation **(7.15)**

$$P_0(t+\Delta t) = P_0(t) - P_0(t)\,\lambda\Delta t + P_1(t)\,\mu\Delta t \qquad \textbf{(7.17)}$$

Then,

$$\lim_{\Delta t\to 0}\frac{P_0(t+\Delta t)-P_0(t)}{\Delta t} = -P_0(t)\lambda + P_1(t)\,\mu \qquad \textbf{(7.18)}$$

Thus,

$$\frac{dP_0(t)}{dt} + P_0(t)\,\lambda = P_1(t)\,\mu \qquad \textbf{(7.19)}$$

Similarly, from the equation **(7.16)**,

$$\frac{dP_1(t)}{dt} + P_1(t)\,\mu = P_0(t)\,\lambda \qquad \textbf{(7.20)}$$

at time $t=0$, $P_0(0)=1$ and $P_1(0)=0$.

Laplace transform of these equation,

$$sP_0(s) + P_0(s)\lambda - P_1(s)\,\mu = 1 \qquad \textbf{(7.21)}$$

$$sP_1(s) + P_1(s)\,\mu - P_0(s)\,\lambda = 0 \qquad \textbf{(7.22)}$$

If we rearrange the equation **(7.22)**,

$$P_1(s) = \frac{\lambda}{s+\mu} P_0(s) \qquad \textbf{(7.23)}$$

Using equation **(7.21)** and equation **(7.23)**,

$$P_0(s)(s+\lambda) - \mu\left(\frac{\lambda}{s+\mu} P_0(s)\right) = 1$$

$$P_0(s)\left(\frac{(s+\lambda)\cdot(s+\mu) - \mu\cdot\lambda}{s+\mu}\right) = 1$$

$$P_0(s) = \frac{s+\mu}{(s+\lambda)\cdot(s+\mu) - \mu\cdot\lambda} \qquad \textbf{(7.24)}$$

And using equation **(7.23)** and equation **(7.24)**, $P_1(s)$ will be,

$$P_1(s) = \frac{\lambda}{(s+\lambda)\cdot(s+\mu) - \mu\cdot\lambda} \qquad \textbf{(7.25)}$$

Inverse Laplace transform of **(7.24)** and **(7.25)**,

$$P_0(t) = \frac{\mu}{(\lambda+\mu)} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} \qquad \textbf{(7.26)}$$

$$P_1(t) = \frac{\lambda}{(\lambda+\mu)} - \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} \qquad \textbf{(7.27)}$$

Thus the availability of the system can be defined with equation **(7.26)**

$$A(t) = P_0(t) = \frac{\mu}{(\lambda+\mu)} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} \quad \textbf{(7.28)}$$

Where

$A(t)$: component time-dependent availability

If it is considered that time is equal to infinity,

$$A = \lim_{t\to\infty} A(t) = \frac{\mu}{(\lambda+\mu)} \quad \textbf{(7.29)}$$

Where

A : component steady-state availability

And the unavailability can be defined similarly,

$$UA = \frac{\lambda}{(\lambda+\mu)} \quad \textbf{(7.30)}$$

Where

UA : component steady-state unavailability

For $\mu = 0$, from equation **(7.26)**,

$$R(t) = P_0(t) = e^{-\lambda t} \quad \textbf{(7.31)}$$

Where

$R(t)$: component reliability at time *t*

If equation **(7.31)** is integrated over the time interval [0, ∞], following expression can be obtained for the mean time to failure (MTTF) of the component:

$$MTTF = \int_0^\infty e^{-\lambda t} dt$$

$$MTTF = \frac{1}{\lambda} \quad \textbf{(7.32)}$$

For instance, if the failure rate and the repair rate are considered as equal to 0.0003 failures/hour and 0.0006 repairs/hour, respectively, all parameters of the component for 100-hour mission and the steady-state availability can be calculated.

Reliability of the component:

$$R(t) = e^{-\lambda t} = e^{-0.0003 \cdot 100} = 0.9704 \quad \textbf{(7.33)}$$

MTTF of the component:

$$MTTF = \frac{1}{\lambda} = \frac{1}{0.0003} = 3333.3\ hour \quad \textbf{(7.34)}$$

Availability of the component for 100 hours:

$$A(t) = \frac{\mu}{(\lambda + \mu)} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}$$

$$A(100) = \frac{0.0006}{(0.0003 + 0.0006)} + \frac{0.0003}{0.0003 + 0.0006} e^{-(0.0003+0.0006) \cdot 100}$$

$$A(100) = 0.9713 \quad \textbf{(7.35)}$$

Steady-state availability of the component:

$$A = \frac{\mu}{(\lambda + \mu)} = \frac{0.0006}{(0.0003 + 0.0006)} = 0.6667 \quad \textbf{(7.36)}$$

7.3 Markov Model of Model Station

In this chapter, a simple Markov model for the model station described in the previous chapters has been designed. Furthermore, some RAMS parameters have been obtained according to the Markov model.

Possible failures in railway signalling systems can be categorized by several ways with respect to type of the interlocking system. Also, different system states can be considered according to the failure types. In this study, two type of failures and four

basic states have been considered. Because of the complexity of the model station's layout, component based failures are not considered. The type of the failures are explained and the explanation of the states are given in the Table 7.2.

Tolerable Failure: Tolerable failure is described for the failures which doesn't block the operation of the trains. For instance, a failure in the track detection device can be considered as a tolerable failure. If a track detection device gives an occupancy information when there is any railway vehicle or any obstacle in the section then the signal operator can give the movement authority to the driver by a radio contact or special signal aspect defined in the system.

Significant Failure: This type of failures represent the failures which hinder the operation of the trains. For example, point failures can be given as operation-blocking failures. Because, trains cannot proceed on a failed point.

System status can be seen in the following Table 7.2.

Table 7.2 : Definitions of the system states [1].

States	Explanation
State 0	System operating normally
State 1	System has a tolerable failure
State 1a	System operating in degraded mode
State 2	System has a significant failure

State 0 is defined for the situation of normal operating and system working failure-free. State 1 represents the system status when there is a tolerable failure in the system. When a tolerable failure occurs in the system, the signal operator can confirm this failure, if he or she is ensure that this is a tolerable failure, and the system can be operated in degraded mode. This condition is represented by State 1a. The last state is the significant failure state and the system cannot be operated in this state.

Figure 7.9 shows the generated Markov model of the model station. Failure rates and repair rates are considered as constant values. The mathematical equation of the model can be obtained by the expressions given in the "Markov Model" chapter.

Table 7.3 : Definitions of the transitions [1].

Transitions	Explanation
$\lambda_{0\|1}$	Tolerable failure rate
$\lambda_{0\|2}$	Significant failure rate
$\mu_{1\|0}$	Repair rate of tolerable failure
$\mu_{2\|0}$	Repair rate of significant failure
$\mu_{1\|1a}$	Transition rate from tolerable failure to degraded mode
$\mu_{1a\|0}$	Repair rate of degraded mode

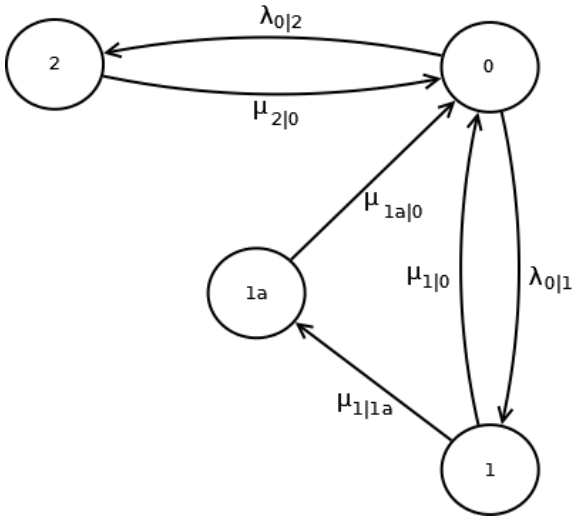

Figure 7.9 : Markov model of the model station [1].

$$\frac{dP_0(t)}{dt} + \left(\lambda_{0|1} + \lambda_{0|2}\right) \cdot P_0(t) = \mu_{0|1} \cdot P_1(t) + \mu_{2|0} \cdot P_2(t) + \mu_{1a|0} \cdot P_{1a}(t) \quad \textbf{(7.37)}$$

$$\frac{dP_1(t)}{dt} + \left(\mu_{1|0} + \mu_{1|1a}\right) \cdot P_1(t) = \lambda_{0|1} \cdot P_0(t) \quad \textbf{(7.38)}$$

$$\frac{dP_2(t)}{dt} + \left(\mu_{2|0}\right) \cdot P_2(t) = \lambda_{0|2} \cdot P_0(t) \quad \textbf{(7.39)}$$

$$\frac{dP_{1a}(t)}{dt} + \left(\mu_{1a|0}\right) \cdot P_{1a}(t) = \mu_{1|1a} \cdot P_1(t) \quad \textbf{(7.40)}$$

At time t=0, $P_0(0) = 1, P_1(0) = 0, P_2(0) = 0, P_{1a}(0) = 0$

If these equations are solved with a calculation tool (e.g Matlab or Mathematica) following solutions are obtained for steady-state ($t = \infty$) [40]

$$P_0 = \frac{\mu_{1a|0} \cdot (\mu_{2|0}) \cdot (\mu_{1|1a} + \mu_{1|0})}{(\mu_{2|0}) \cdot [\mu_{1a|0} \cdot (\mu_{1|1a} + \mu_{1|0}) + \lambda_{0|1} \cdot (\mu_{1a|0} + \mu_{1|1a})] + \lambda_{0|2} \cdot (\mu_{1|1a} + \mu_{1|0}) \cdot (\mu_{1a|0})} \quad \textbf{(7.41)}$$

$$P_1 = \frac{\lambda_{0|2}}{\mu_{2|0}} \cdot P_0 \quad \textbf{(7.42)}$$

$$P_2 = \frac{\lambda_{0|1}}{(\mu_{1a|0} + \mu_{1|0})} \cdot P_0 \quad \textbf{(7.43)}$$

$$P_{1a} = \frac{\lambda_{0|1} \cdot \mu_{1|1a} \cdot \mu_{2|0}}{\mu_{1a|0} \cdot (\mu_{2|0}) \cdot (\mu_{1|1a} + \mu_{1|0})} \cdot P_0 \quad \textbf{(7.44)}$$

Hence, the steady-state availability of the system will be

$$A = P_0$$

$$A = \frac{\mu_{1a|0} \cdot (\mu_{2|0}) \cdot (\mu_{1|1a} + \mu_{1|0})}{(\mu_{2|0}) \cdot [\mu_{1a|0} \cdot (\mu_{1|1a} + \mu_{1|0}) + \lambda_{0|1} \cdot (\mu_{1a|0} + \mu_{1|1a})] + \lambda_{0|2} \cdot (\mu_{1|1a} + \mu_{1|0}) \cdot (\mu_{1a|0})} \quad \textbf{(7.45)}$$

The reliability of the system,

$$R(t) = P_0(t) = e^{-(\lambda_{0|1} + \lambda_{0|2}) \cdot t} \quad \textbf{(7.46)}$$

Lastly, MTTF of the system,

$$MTTF = \int_0^\infty R(t)dt$$

$$MTTF = \int_0^\infty e^{-(\lambda_{0|1}+\lambda_{0|2})\cdot t}dt$$

$$MTTF = \frac{1}{\lambda_{0|1} + \lambda_{0|2}} \qquad \textbf{(7.47)}$$

Using this obtained equation, the parameters of a track detection device can be calculated. Following artificial failure and repair rate data are considered.

$$\lambda_{0|1} = 5{,}15 \times 10^{-6}\ 1/h$$

$$\lambda_{0|2} = 1 \times 10^{-11}\ 1/h$$

$$\mu_{1|0} = 0{,}0833\ 1/h$$

$$\mu_{2|0} = 18000\ 1/h$$

$$\mu_{1|1a} = 18000\ 1/h$$

$$\mu_{1a|0} = 0{,}5208\ 1/h$$

$$t = 5\ years\ (43800\ hours)$$

$$A = \frac{0{,}5208 \cdot (18000) \cdot (18000 + 0{,}0833)}{(18000) \cdot [0{,}5208 \cdot (18000 + 0{,}0833) + 5{,}15 \times 10^{-6} \cdot (0{,}5208 + 18000)] + 10^{-11} \cdot (18000 + 0{,}0833) \cdot (0{,}5208)}$$

$$R(43800) = e^{-(5{,}15\times10^{-6}+1\times10^{-11})\cdot 43800}$$

$$MTTF = \frac{1}{5{,}15 \times 10^{-6} + 1 \times 10^{-11}}$$

Hence,

$$A \cong 0{,}999990111224568 \qquad \textbf{(7.48)}$$

$$R(43800) \cong 0{,}798060844658274 \qquad \textbf{(7.49)}$$

$$MTTF \cong 194174{,}3802\ hours \cong 22{,}2\ years \qquad \textbf{(7.50)}$$

Following figures shows the effect of the repair rate $\mu_{2|0}$ to the steady-state availability with a range starts from 1 to 36000 1/h.

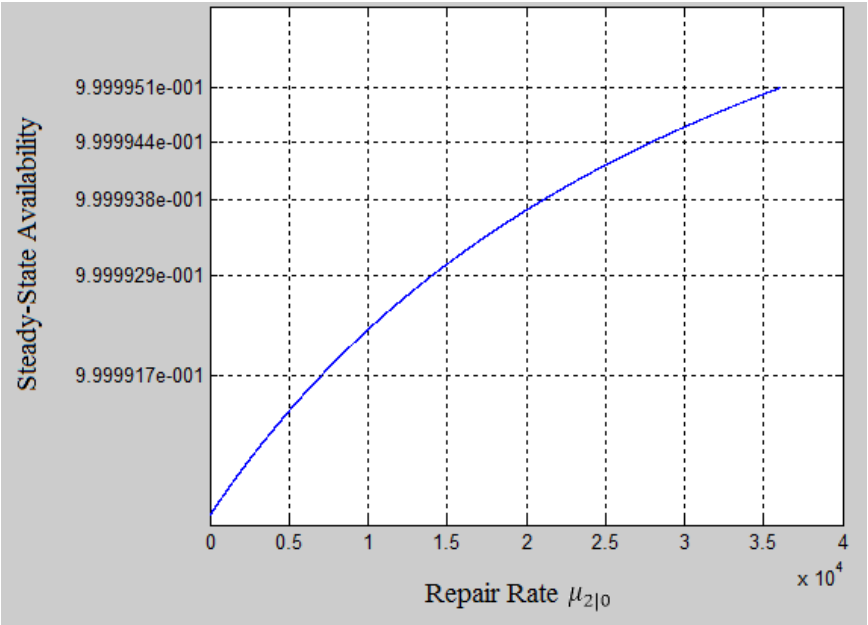

Figure 7.10 : Effect of the repair rate $\mu_{2|0}$ to the steady-state availability [1].

As it can be seen in the Figure 7.10, different repair rates affect availability parameter.

8. CONCLUSION

Railway interlocking is a safety, financial and environmentally critical system. Any failure in it can cause very serious consequences such as loss of human life, severe injuries, huge environmental damages or economic penalties. Therefore, railway interlocking systems are designed with specific development methods [42]. The usage of formal methods in the development of safety critical systems is encouraged. Consequently, applying formal methods in this area is an open issue.

The first main objective of this thesis was to analyze the formal methods in the development of a example interlocking mechanism. For that purpose, all basic terms and equipment used in railway signalling were defined. Then, the definition of the interlocking was given and the functionality of the interlocking in railways was explained. A model station was designed with respect to German signalling principles.

Use of formal methods in the design of railway interlocking is explained. Two widely used formal methods, "Petri Nets" and "Finite State Machines" were discussed with an example. Furthermore, implementation of designed example model was realized with two different software tools.

As a next step, control mechanisms of basic wayside equipment and other mechanisms for route setting function were modelled with FSM. Moreover, generated models were implemented with the PLC programming software, SilworX.

Finally, generated route setting mechanism was tested with the simulator feature of the same software. The behavior of the models was the same with expected results. All processes of route setting mechanism were monitored step by step. In addition to that some failure cases were simulated. The models gave the correct outputs in the failure situations.

The second main objective of this thesis was to make a RAMS analysis for designed model station. Basic definitions of RAMS analysis were discussed with reference to EN 50126. Then, two mostly used methods which indicated in the same standard were examined with detailed examples. Using one of these models, some RAMS parameters

were obtained as a final step. Then, these parameters were implemented to an equipment with artificial data as an example analysis. A graph was discussed to understand the interaction between data and RAMS parameters.

In modern railway signalling systems, interlocking unit is designed with regard to hardware and software redundancy. Hardware redundancy is considered to prevent the dangerous consequences of any failure in the hardware of interlocking unit. On the other hand, software redundancy is used to protect the system against software failures which may occur in the interlocking program. Therefore, diversity of interlocking software is recommended to high safety level.

As a future work, different formal methods can be used to design the same interlocking functions to obtain software diversity. Therefore, small modelling failures in design steps can be prevented. In addition to that, a voting unit can be designed for the interlocking units which based on different formal methods.

The result outcome of this thesis report is not comparable to a real interlocking system. However, the methodology and approach used in this study is based on scientific basis. It is only serves an academic purpose. Therefore, the results must be carefully checked before any further adoption.

REFERENCES

[1] **Bellek, M.** (2013). Design and Rams Analysis Of Railway Interlocking Based On Formal Methods: An Example Application, Master Thesis, Dresden University of Technology, Dresden.

[2] **Theeg G., Maschek U. and Nasedkin a. O.** (2009). Railway Signalling & Interlocking - International Compendium, Hamburg: Eurail Press.

[3] **Ürün Y. and Gülbahar V.** (1972). Demiryolu Sinyalizasyonu ve Yeni Teknikler, Elektrik Mühendisliği Odası Yayın Organı, no. 183, pp. 44-68.

[4] **Url-1** (2013). European Comission Statistics Explained, Eurostat, http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Railway_safety_statistics. date retrieved 28.08.2013.

[5] **European Committee for Electrotechnical Standardization (CENELEC)** (2010), E. 61508-7 Functional safety of electrical/ electronic/programmable electronic safety-related systems, Brussels.

[6] **Mahboob Q., Schöne E., Kunze M., Trinckauf J. and Maschek U.** (2012). Application of Importance Measures to Transport Industry: Computation Using Bayesian Networks and Fault Tree Analysis, QR2MSE 2012: International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, Chengdu (China).

[7] **Mahboob Q., Schöne E., Kunze M., Trinckauf J. and Maschek U.** (2012). Representing Advanced Aspects of Fault Trees Into Bayesian Networks - Modelling Safety in Complex Railway Systems, in PSAM 11 & ESREL 2012: Conference on Probabilistic Safety Assessment and Management & European Safety and Reliability, Helsinki.

[8] **Hotchkiss D.** (1995). European Railway Signalling, London: Institution of Railway Signal Engineers, London.

[9] **Url-2** <https://www.thalesgroup.com/en/content/thales-deliver-prototype-traffic- mana gement-system-network-rail>, date retrieved 30.08.2013.

[10] **Url-3** <http://m8.i.pbase.com/o2/16/639416/1/120200148.a8S2iTVA.Lakeside Sub.jpg >, date retrieved 05.09.2013.

[11] **Url-4** <http://www.railway-technical.com/Turnout%20names.png>, date retrieved 05.09.2013.

[12] **Url-5** <http://www.railway-technical.com/track.shtml>, date retrieved 05.09.2013.

[13] **Url-6** <http://en.wikipedia.org/wiki/Railway_signa> , date retrieved 05.09.2013.

[14] **Url-7** <http://en.wikipedia.org/wiki/Track_circuit>, date retrieved 05.09.2013.

[15] **Url-8** <http://en.wikipedia.org/wiki/Catch_points>, date retrieved 05.08.2013.

[16] **Transportation Safety Board of Canada** (2013). Railway Investigation Report, Tisdale.

[17] **Url-9** <http://www.pds-drivertraining.co.uk/wp-content/uploads/2013/03/pedcrossone.png>, date retrieved 05.09.2013.

[18] **Url-10** <http://www.sh1.org/>, date retrieved 25.08.2013.

[19] **Url-11** <http://www.sh1.org/eisenbahn/shks.htm>, date retrieved 06.09.2013.

[20] **Yıldırım U., Söylemez M. T. and Durmuş M. S.** (2012). Automatic Interlocking Table Generation for Railway Stations using Symbolic Algebra, 13th IFAC Symposium on Control in Transportation Systems, CTS 2012, Sofia.

[21] **Babacan V. K. (2011).** Raylı Sistemlerde Sinyalizasyon Tekniğine Giriş, Raycan raylı sistemler inşaat makine elektrik basın yayın sanayi veticaret limited şirketi, İstanbul.

[22] **Durmuş M. S., Yıldırım U., Kurşun A. and Söylemez M. T.** (2010). Fail-Safe Signalization Design for a Railway Yard: A Level Crossing Case, 10th International Workshop on Discrete Event Systems, Berlin.

[23] **Durmuş M. S., Söylemez M. T. and Avşaroğulları E. (2009).** Coloured Automation Petri Nets Based Interlocking and Signalization Design, 6th IFAC International Workshop on Knowledge and Technology Transfer in/to Developing Countries, Macedonia.

[24] **Url-12** <http://en.wikipedia.org/wiki/Interlocking>, date retrieved 10.09.2013.

[25] **Pachl J.** (2011). Deadlock Avoidance in Railroad Operations Simulations, Transportation Research Board in Washington DC, Washington DC.

[26] **Khan S. A.** (2011). Formal Analysis of Safety Properties of Railway Interlocking System, Ph.D Thesis, University of Central Punjab, Lahore.

[27] **Tretmans J., Wijbrans K. and Chaudron M.** (2000). Software Engineering with Formal Methods: The Development of a Storm Surge Barrier Control System - Revisiting Seven Myths of Formal Methods, Kluwer Academic Publishers, Netherlands.

[28] **Archer M., Heitmeyer C. and Riccobene E.** (2002). Proving Invariants of I/O Automata with TAME, Automated Software Engineering, Catania.

[29] **Durmuş M. S., Yıldırım U., Eriş O. and Söylemez M. T.** (2011). Synchronizing Automata and Petri Net Based Controllers, 7th International Conference on Electrical and Electronics Engineering (ELECO 2011), Bursa.

[30] **Durmuş M. S., Yıldırım U. and Söylemez M. T.** (2012). Automatic Generation of Petri Net Supervisors for Railway Interlocking Design, Australian Control Conference (AUCC 2012), Sydney.

[31] **Cassandras C. G. and Lafortune S.** (2008). Introduction to Discrete Event Systems Second Edition, New York: Springer Science+Business Media, LLC, New York.

[32] **Durmuş M. S., Yıldırım U. and Söylemez M. T.** (2013). The Application of Automation Theory to Railway Signaling Systems: The Turkish National Railway Signaling Project, Pamukkale University Journal of Engineering Sciences (PAJES), pp. 2016-224.

[33] **Eris O.** (2011). Realization of a Railway Interlocking System with PLC, Master Thesis, Istanbul Technical University, Istanbul.

[34] **Url-13** <https://electrobiz.files.wordpress.com/2011/03/tripod_turnstile.jpg>, date retrieved 10.09.2013.

[35] **Siemens** (2010). Ladder Logic (LAD) for S7-300 and S7-400 Programming, Reference Manual, Siemens.

[36] **Champarnaud J.-M., Maurel D. and Ziadi D.** (1998). Automata Implementation, Third International Workshop on Implementing Automata, WIA'98, Rouen.

[37] **International Electrotechnical Commission** (1993). IEC 61131-3 Programmable Controllers, Part 3 Programming Languages, Geneva.

[38] **European Committee for Electrotechnical Standardization (CENELEC)** (1999). EN 50126-1Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process, Brussels.

[39] **Url-14** <http://girdhargopalbansal.blogspot.com.tr/2014/02/software-and-hardware-reliability.html>, date retrieved 15.09.2013.

[40] **Dhillon B. S.** (2011). Transportation Systems Reliability and Safety, Boca Raton: CRC Press, Taylor & Francis Group.

[41] **Billinton R.** (1992). Reliability Evaluation of Engineering Systems - Concepts and Techniques, New York: Plenum Press - New York and London.

[42] **Hartig K., Gerlach J., Soto J. and Busse J.** (2010). Formal Specification and Automated Verification of Safety-Critical Requirements of a Railway Vehicle with Frama-C/Jessie, FORMS/FORMAT 2010, Berlin.

CURRICULUM VITAE

Full Name **:** Mustafa Bellek

Birth Place and Date: Istanbul – 09.10.1987

Graduations **:** Yildiz Technical University, Electrical Engineering – B.Sc., (2005 - 2010)

İstanbul Technical University, Electrical Engineering – M.Sc., (2011- ..)

Technische Universität Dresden, Faculty of Transportation and Traffic Sciences – M.Sc. Exchange Student, (2012 – 2013)

Contact **:** mustafabellek@gmail.com

Work Experiences **:** İstanbul Ulaşım A.Ş, Signalling Installation & Maintenance Engineer, Electrical & Electronics Installation Department, (2011 - ..)

Thales Transportation Systems, M.Sc. Thesis Student, (2013)